Preliminary Data Sheet
February 1998

microelectronics group

Lucent Technologies
Bell Labs Innovations

DSP1609/*FlashDSP*®1609F Digital Signal Processor

1 Features

- For 5 V operation:
  - — 10.0 ns instruction cycle time (100 MIPS) (see page 104)
  - — 12.5 ns instruction cycle time (80 MIPS) (for the DSP1609)
- For 3.3 V operation:
  - — 12.5 ns instruction cycle time (80 MIPS) (for the *FlashDSP*1609F)
- Power-saving features:
  - — Low-power 0.35 µm CMOS technology; fully static design
  - — Active power: 9.5 mW/MIPS at 5.0 V
    3.3 mW/MIPS at 3.3 V
  - — Low-power stopclk: 175 µW at 5.0 V
    66 µW at 3.3 V
- For DSP1609F: 24 Kwords internal flash ROM; For DSP1609: 24 Kwords internal ROM
- 2 Kwords internal RAM
- 16 x 16-bit multiplication and 36-bit accumulation in one instruction cycle
- Two 36-bit accumulators
- Instruction cache for high-speed, program-efficient, zero-overhead looping
- One external vectored interrupt
- Two 64 Kword address spaces
- Programmable phase-locked loop
- Three 8-bit and one 4-bit I/O ports for flexible status or control pins for 44-pin PLCC package
- Two 8-bit I/O ports for 28-pin SOJ package
- Two interrupt timers and one watchdog timer
- For DSP1609: 28-pin SOJ or 44-pin PLCC package; for *FlashDSP*1609F: 44-pin PLCC
- High- and low-frequency clock options
- Synchronous serial interface unit
- Object code upward compatible with DSP1600 Digital Signal Processor family
- Supported by DSP1609 support tools
- Full-speed in-circuit emulation HDS (HD-supported)
- One dual-channel serial I/O port
- One bit manipulation unit
- DRAM control interface

2 Description

The DSP1609 is a 16-bit, fixed-point digital signal processor (DSP) based on the DSP1600 core. It is programmable to perform a wide variety of fixed-point signal processing functions. A member of the DSP1600 family, the DSP1609 includes a mix of peripherals specifically intended to support processing-intensive but cost-sensitive applications. In addition to the core, the DSP1609 consists of the following peripheral blocks: a programmable phase-locked loop (PLL), synchronous serial interface unit (SSI), four I/O ports (IOPs), two timer units, a watchdog timer, one dual-channel serial I/O interface (SIO), and a JTAG interface; as well as 2 Kwords of RAM. The DSP1609 is part of a low-cost, high-performance solution for consumer product applications.

The DSP1609 is available in the following packages:

- 28-pin SOJ (not available for the *FlashDSP*1609F) (See Figure 1 on page 8.)
- 44-pin PLCC (See Figure 2 on page 9.)

The DSP1609/*FlashDSP*1609F achieves high throughput without programming restrictions or latencies due to its parallel pipelined architecture. The processor has an arithmetic unit capable of a 16 x 16-bit multiplication and 36-bit accumulation, or a 32-bit ALU operation in one instruction cycle. Data is accessed from memory through two independent addressing units.

A fully static, low-power, 0.35 µm CMOS design and a low-power standby mode support power-sensitive equipment applications. A single external crystal allows the use of a high-frequency and a low-frequency clock. Under program control, the DSP1609/*FlashDSP*1609F can be switched between the high-frequency and low-frequency clock options. When switched to the low-frequency clock, the power is reduced and can be further reduced using a stop-clock mode.

The *FlashDSP*1609F device is the development platform for the DSP1609. To support full-speed in-circuit emulation, the *FlashDSP*1609F device includes an internal HDS module and an internal flash ROM for program development.

Table of Contents

Contents Page

Table of Contents (continued)

Contents **Page**

Table of Contents (continued)

Contents **Page**

Table of Contents (continued)

Tables

Table of Contents (continued)

Tables

Table of Contents (continued)

Tables

Figures

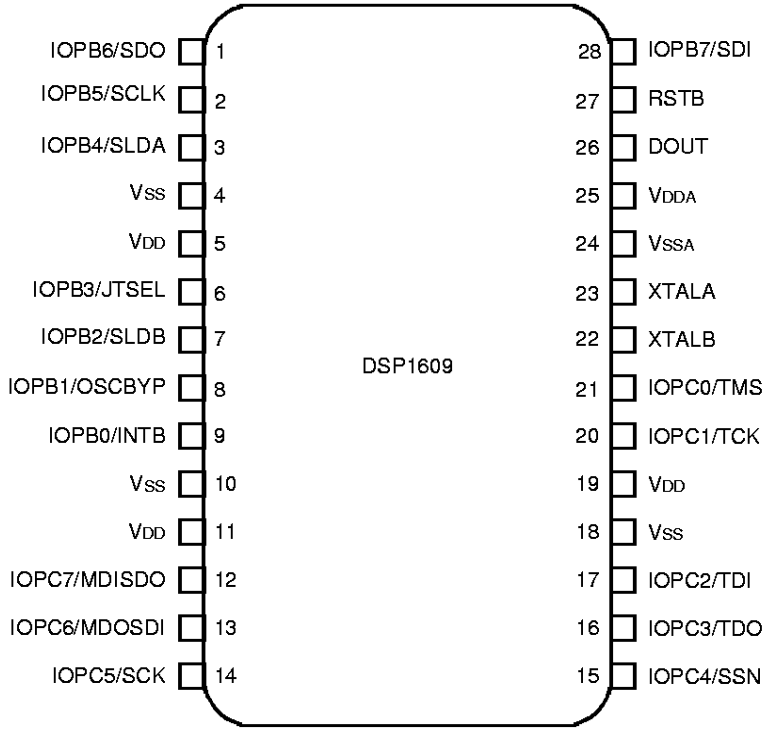
2 Description (continued)

The DSP1609 is identical to the *FlashDSP* 1609 (1609F) except that it contains mask-programmable internal ROM in place of flash ROM.

3 Pin Information

The DSP1609 is available in the following packages:

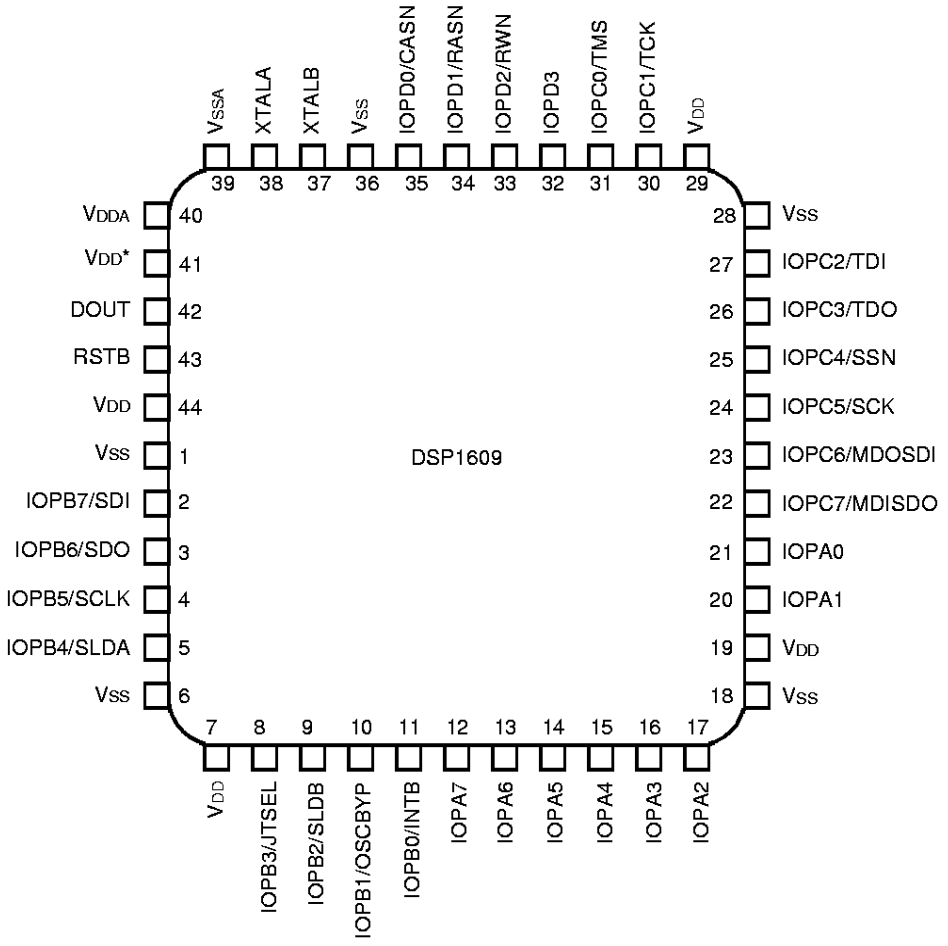
- 28-pin SOJ (See Figure 1.)
- 44-pin PLCC (See Figure 2 on page 9.)



5-6212 (F)

Figure 1. DSP1609 28-Pin SOJ Pin Diagram

3 Pin Information (continued)



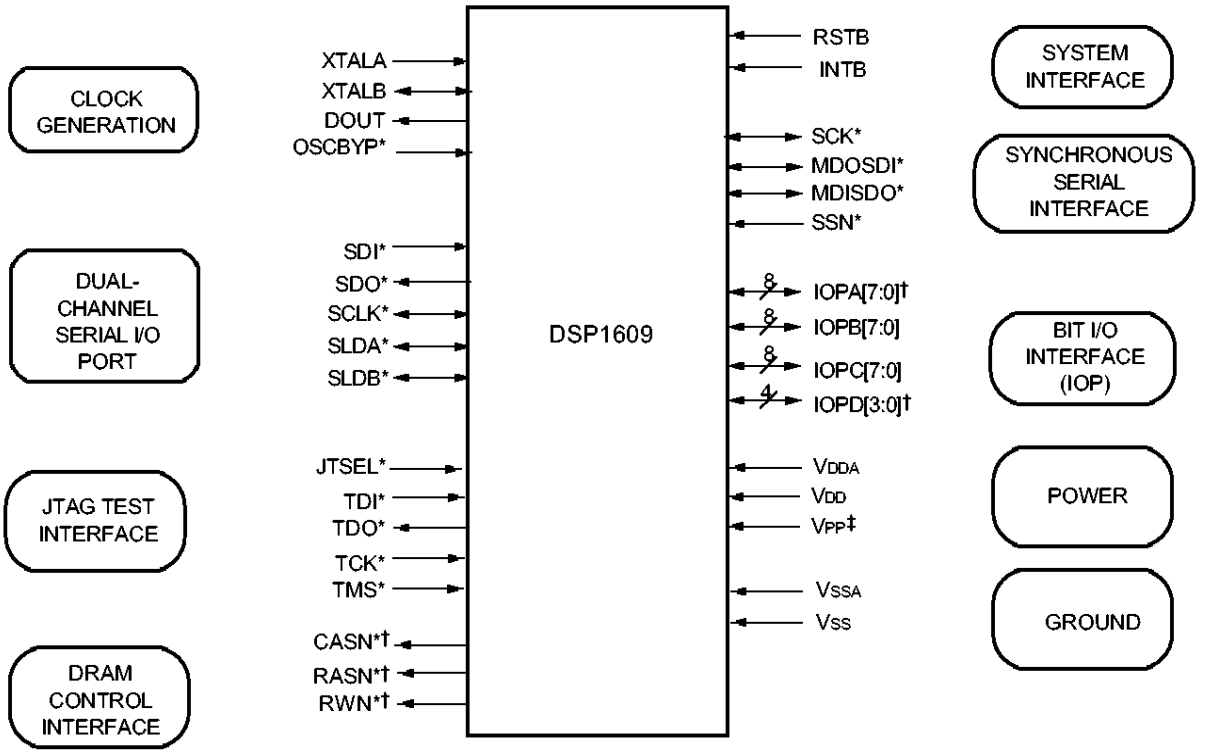
* VPP on *FlashDSP*1609F.

5-6213 (F)

Figure 2. DSP1609 44-Pin PLCC Pin Diagram

3 Pin Information (continued)

Figure 3 shows DSP1609 pins organized into their functional groups. Tables 1 and 2 and Sections 3.1 through 3.5 describe these pins.



* Multiplexed with IOP port pins.
† The DSP1609 28-pin SOJ package has no IOPA[7:0], IOPD[3:0], RASN, CASN, and RWN.
‡ The DSP1609 has no VPP.

5-4006(C).a

Figure 3. DSP1609 Pinout by Group

For each DSP1609/DSP1609F pin listed in Table 1:

- Each entry in the Type column is one of the following:
  - —I (input)
  - —I/O (input/output)
  - —O (output)
  - —OD (open-drain output)
  - —TO (3-state output)
- Each entry in the Active column is one of the following:
  - —High
  - —Low
  - —Pgm (programmable polarity)
  - —Neg (negative edge triggered)

3 Pin Information (continued)

Table 1. DSP1609 Pinout

Symbol	Pin Number		Type	Active	Pin Description
	28-Pin SOJ	44-Pin PLCC			
DOUT	26	42	O	High	Processor clock; digital output/clock output.
INTB*	9	11	I	Neg	External interrupt, negative edge triggered; multiplexed with IOPB0.
IOPA[7:0]	—	12—17, 20, 21	I/O	Pgm	I/O port A (bits 7—0).
IOPB[7:0]	28, 1—3, 6—9	2—5, 8—11	I/O	Pgm	I/O port B (bits 7—0).
IOPC[7:0]	12—17, 20, 21	22—27, 30, 31	I/O	Pgm	I/O port C (bits 7—0).
IOPD[3:0]	—	32—35	I/O	Pgm	I/O port D (bits 3—0).
JTSEL*	6	8	I	High	JTAG select. (JTSEL is multiplexed with IOPB3.)
MDISDO*	12	22	I/O	—	SSI master data in/slave data out; multiplexed with IOPC7.
MDOSDI*	13	23	I/O	—	SSI master data out/slave data in; multiplexed with IOPC6.
OSCBYP*	8	10	I	High	Oscillator bypass. (OSCBYP is multiplexed with IOPB1.)
RSTB	27	43	I/O	Low	Reset.
SCK*	14	24	I/O	—	SSI serial clock; multiplexed with IOPC5.
SSN*	15	25	I	Low	SSI serial slave select; multiplexed with IOPC4.
TCK*	20	30	I	—	Test clock (JTAG). (TCK is multiplexed with IOPC1.)
TDI*	17	27	I	High	Test data in (JTAG). (TDI is multiplexed with IOPC2.)
TDO*	16	26	O	High	Test data out (JTAG). (TDO is multiplexed with IOPC3.)
TMS*	21	31	I	High	Test mode select (JTAG). (TMS is multiplexed with IOPC0.)
CASN*	—	35	O	Low	Column address select. (CASN is multiplexed with IOPD0.)
RASN*	—	34	O	Low	Row address select. (RASN is multiplexed with IOPD1.)
RWN	—	33	O	High	Read/write not. (RWN is multiplexed with IOPD2.)
XTALA	23	38	I	—	Crystal oscillator connection.
XTALB	22	37	I/O		
SDI*	28	2	I	—	SIO data in; multiplexed with IOPB7.
SDO*	1	3	O	—	SIO data out; multiplexed with IOPB6.
SCLK*	2	4	I/O	—	SIO shift clock; multiplexed with IOPB5.
SLDA*	3	5	I/O	—	SIO channel A load clock; multiplexed with IOPB4.
SLDB*	7	9	I/O	—	SIO channel B load clock; multiplexed with IOPB2.

* Multiplexed with IOP pins.

3 Pin Information (continued)

Table 2. DSP1609 Power Supply, Ground, and Unconnected Pins

Symbol	Pin Number		Pin Description
	28-Pin SOJ	44-Pin PLCC	
VDD	5, 11, 19	7, 19, 29, 41*, 44	Supply for digital circuitry (5 V for DSP1609 or 3.3 V for DSP1609F).
VSS	4, 10, 18	1, 6, 18, 28, 36	Ground for digital circuitry.
VDDA	25	40	Supply for phase-locked loop (should be connected to VDD).
VSSA	24	39	Ground for phase-locked loop (should be connected to VSS).
VPP*	—	41*	Programming supply for flash ROM.

* Pin 41 is a VDD pin in the DSP1609, but a VPP pin in the *FlashDSP*1609F.

3.1 System Interface

3.1.1 XTALA, XTALB

Crystal Oscillator. Input/Output. The external crystal is connected between these two pins, which places the crystal in the feedback loop of the on-chip crystal oscillator. The output of the crystal oscillator is used as a reference for the phase-locked loop (PLL). The frequency of the oscillator input must be 4.096 MHz.

3.1.2 RSTB

Reset. Bidirectional (Schmitt trigger). Negative assertion. A high-to-low transition on RSTB causes the DSP1609/*FlashDSP*1609F to enter the reset state. Upon deassertion, the DSP1609/*FlashDSP*1609F begins execution from location 0x0000.

Note: This input contains a Schmitt trigger providing a hysteresis between positive- and negative-going transitions. When the watchdog timer is triggered, RSTB is pulled low.

3.1.3 INTB

Interrupt. Input (Schmitt trigger). Negative edge triggered. External interrupt to the DSP1609/*FlashDSP* 1609F. This input contains a Schmitt trigger providing a hysteresis between positive and negative-going transitions. An interrupt is posted when a negative-going transition is detected. Also, note that there is no acknowledge pin. External hardware must guarantee that the interrupt service routine has completed before issuing another interrupt on INTB.

Any activity on INTB is **not** recognized during the interrupt service routine and must be reissued after the interrupt service routine has completed.

3.1.4 DOUT

Digital Output. Output. The default of the pin is to output CLKLOW, the internal low-frequency clock. DOUT is selectable through the **clkc** register (see Table 37 on page 57). The following options are selectable: internal low-power oscillator, an input crystal clock, a free-running clock, a wait-stated clock, a logic 0, or a logic 1.

3.2 Synchronous Serial Interface (SSI)

The SSI is configurable for either master mode or slave mode, selectable by the **ssic** register.

3.2.1 SCK

SSI Clock. Bidirectional. If SSI is configured in master mode, this pin is an output providing a clock to the slave devices. If SSI is configured in slave mode, this pin is an input that takes in the serial clock.

3.2.2 MDOSDI

Master Data Out/Slave Data In. Bidirectional. If SSI is configured in master mode, this pin is the serial data output. If SSI is configured in slave mode, this pin is the serial data input. The SSI is either master or slave, according to the **ssic** register MSTR field (see Table 48 on page 64).

3 Pin Information (continued)

3.2 Synchronous Serial Interface (SSI) (continued)

3.2.3 MDISDO

Master Data In/Slave Data Out. Bidirectional. If SSI is configured in master mode, this pin is the serial data input. If SSI is configured in slave mode, this pin is the serial data output. The SSI is either master or slave, according to the **ssic** register MSTR field (see Table 48 on page 64).

3.2.4 SSN

Serial Select. Input. Negative assertion. When SSI is configured as a slave, the assertion of SSN signals the slave that it is being addressed to transfer data with the master device. If this pin is asserted while SSI is configured as a master, a mode fault is detected by SSI. Alternatively, in master mode, this pin can be configured to act as a general-purpose IOP pin.

3.3 I/O Port Interface (IOP)

3.3.1 IOPA[7:0]

I/O Port A. Bidirectional. These pins may be individually configured to be inputs or outputs. As outputs, each bit may be set, cleared, or toggled under program control. As inputs, they may be read. Note that IOPA is not available in the 28-pin SOJ package.

3.3.2 IOPB[7:0]

I/O Port B. Bidirectional. These pins may be individually configured to be inputs or outputs. As outputs, each bit may be set, cleared, or toggled under program control. As inputs, they may be read. Note that IOPB pins are multiplexed as follows:

- IOPB0/INTB
- IOPB1/OSCBYP
- IOPB2/SLDB
- IOPB3/JTSEL
- IOPB4/SLDA
- IOPB5/SCLK
- IOPB6/SDO
- IOPB7/SDI

3.3.3 IOPC[7:0]

I/O Port C. Bidirectional. These pins may be individually configured to be inputs or outputs. As outputs, each bit may be set, cleared, or toggled under program control. As inputs, they may be read. Note that IOPC pins are multiplexed with other pins as follows:

- IOPC0/TMS
- IOPC1/TCK
- IOPC2/TDI
- IOPC3/TDO
- IOPC4/SSN
- IOPC5/SCK
- IOPC6/MDOSDI
- IOPC7/MDISDO

3.3.4 IOPD[3:0]

I/O Port D. Bidirectional. These pins may be individually configured to be inputs or outputs. As outputs, each bit may be set, cleared, or toggled under program control. As inputs, they may be read. Note that the IOPD pins are multiplexed as follows:

- IOPD0/CASN
- IOPD1/RASN
- IOPD2/RWN

Note: IOPD is not available in the 28-pin SOJ package

3.4 JTAG Test Mode Interface

3.4.1 TCK

Test Clock. Input. JTAG serial shift clock that clocks data into TDI and out of TDO and controls the JTAG port by latching TMS into the state machine controller.

3.4.2 TMS

Test Mode Select. Input. JTAG mode control signal that controls the state of the JTAG controller. TMS is sampled on the rising edge of TCK. This pin has an internal pull-up resistor that is typically 68 kΩ.

3 Pin Information (continued)

3.4 JTAG Test Mode Interface (continued)

3.4.3 TDI

Test Data Input. Input. JTAG serial input of all serial-scanned data and instructions that is sampled on the rising edge of TCK. This pin has an internal pull-up resistor that is typically 68 kΩ.

3.4.4 TDO

Test Data Output. 3-state Output. JTAG serial output of all serial-scanned data and status bits. TDO changes on the falling edge of TCK.

3.4.5 JTSEL

JTAG Select. Input. JTSEL is multiplexed with IOPB3. This pin is sampled on the rising edge of RSTB. If sampled high, the four multiplexed JTAG pins will be active in place of IOPC[3:0]. If sampled low, the IOPC[3:0] pins will behave as normal IOP signals.

3.5 DRAM Control Interface

3.5.1 CASN

Column Address Select. Output. Negative assertion. See Section 4.6 for DRAM access operation.

3.5.2 RASN

Row Address Select. Output. Negative assertion. See Section 4.6 for DRAM access operation.

3.5.3 RWN

Read/Write Not. Output. When a logic 1, a read access is in progress; when a logic 0, a write access is in progress. See Section 4.6 for DRAM access operation.

3.6 PWR/GND

3.6.1 VDDA

5.0 V or 3.3 V PLL Supply. VDDA is the positive supply for the analog blocks. Separate bypass capacitors should be connected from VDDA to VSSA. (PLL supply is 5.0 V for the DSP1609 and 3.3 V for the *FlashDSP* 1609F.)

3.6.2 VSSA

PLL Ground. VSSA is the ground return for the analog blocks.

3.6.3 VSS

Ground. The DSP1609 has three ground pins for the 28-pin SOJ and five ground pins for the 44-pin PLCC.

3.6.4 VDD

5.0 V or 3.3 V Supply. There are three power pins for the 28-pin SOJ and five 5.0 V power pins for the 44-pin PLCC. The *FlashDSP*1609F has four 3.3 V power pins for the 44-pin PLCC.

3.6.5 VPP

Flash ROM Power. VPP is the high-voltage pin required on the development device during flash ROM programming and should be connected to 7 V when erasing or programming the flash ROM. Otherwise, it should be connected to VDD.

4 Hardware Architecture

The DSP1609 device is a 16-bit, fixed-point digital signal processor (DSP). The DSP1609 consists of a DSP1600 core together with internal memory and peripherals.

To minimize pin count, the 28-pin DSP1609 multiplexes the following package pins:

- IOPB0 is multiplexed with INTB.
- IOPB1 is multiplexed with OSCBYP.
- IOPB2 is multiplexed with SLDB.
- IOPB3 is multiplexed with JTSEL.
- IOPB4 is multiplexed with SDLA.
- IOPB5 is multiplexed with SCLK.
- IOPB6 is multiplexed with SDO.
- IOPB7 is multiplexed with SDI.
- IOPC0 is multiplexed with TMS.
- IOPC1 is multiplexed with TCK.
- IOPC2 is multiplexed with TDI.
- IOPC3 is multiplexed with TDO.
- IOPC4 is multiplexed with SSN.
- IOPC5 is multiplexed with SCK.
- IOPC6 is multiplexed with MDOSDI.
- IOPC7 is multiplexed with MDISDO.

In addition, the 44-pin DSP1609 and *FlashDSP*1609F multiplex the following package pins:

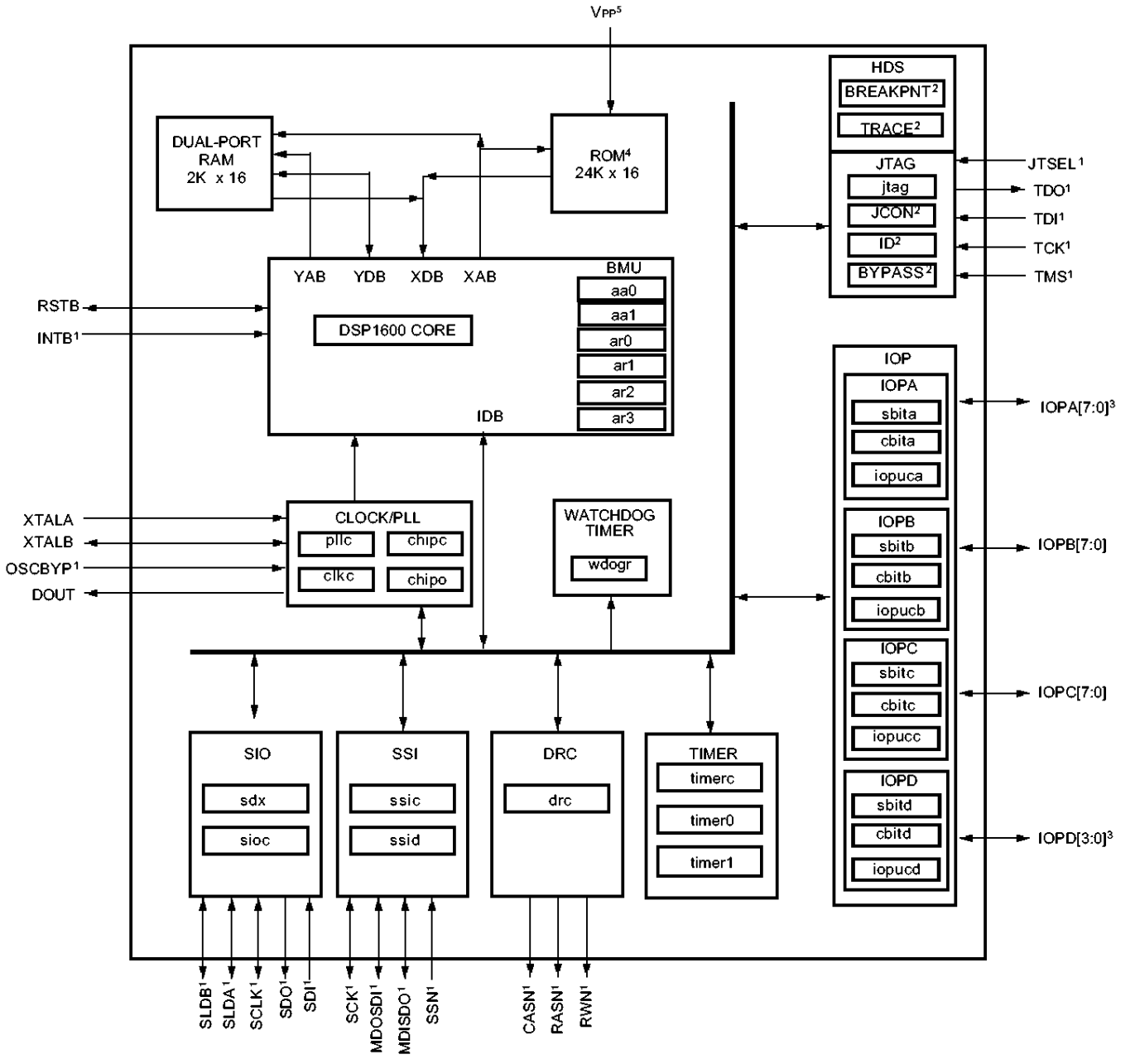
- IOPD0 is multiplexed with CASN.
- IOPD1 is multiplexed with RASN.
- IOPD2 is multiplexed with RWN.

4.1 DSP1609 Architectural Overview

Figure 4 on page 16 shows a block diagram of the DSP1609, which consists of the modules described in Sections 4.1.1 through 4.1.12. Table 3, on page 17, shows a legend for the DSP1609 block diagram. Certain modules contain internal registers that are illustrated (not to scale) in Figure 4. The DSP 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).

4 Hardware Architecture (continued)

4.1 DSP1609 Architectural Overview (continued)



1. These pins are multiplexed with IOP pins.
2. These registers are accessible only through the JTAG pins.
3. Power allocation:
   28-pin SOJ: Digital—three V_{SS} pins and three V_{DD} pins; Analog—one V_{SSA} and one V_{DDA} (for PLL).
   44-pin PLCC: Digital—five V_{SS} pins and five V_{DD} pins; Analog—one V_{SSA} and one V_{DDA} (for PLL).
4. Flash ROM for *FlashDSP*1609F. (For the DSP1609, flash ROM is replace with ROM.)
5. Power supply to flash ROM for *FlashDSP*1609F. (For the 44-pin DSP1609, V_{PP} is replaced with an additional V_{DD} pin.)

Figure 4. DSP1609 Block Diagram

4 Hardware Architecture (continued)

4.1 DSP1609 Architectural Overview (continued)

Table 3. DSP1609 Block Diagram Legend

Symbol	Name
BMU	Bit manipulation unit.
BREAKPNT	Four instruction breakpoint registers.
BYPASS	JTAG bypass register.
cbit<a—d>*	Control registers for IOP<A—D>.
chipc	Chip control register—controls miscellaneous functions.
chipo	Chip option register—configures the watchdog timer and other miscellaneous functions.
clkc	Clock control register.
DPRAM	Internal dual-port random-access memory.
drc	DRAM control register.
flash ROM†	Internal flash read-only memory (IFROM).
flashc	Flash ROM control register—controls flash programming function.
HDS	Hardware development system module.
ID	JTAG device identification register.
IDB	Internal data bus.
IOP<A—D>*	Input/output port units IOPA, IOPB, IOPC, and IOPD.
IOPUC<A—D>	Pull-up/pull-down control register for IOP<A—D>.
JCON	JTAG control register.
jtag	16-bit serial/parallel register.
pllc	Phase-lock loop control register.
ROM‡	Internal read-only memory.
sbit<a—d>*	Status registers for IOP<A—D>.
sdx	SIO data register.
SIO	Dual-channel serial I/O port.
sioc	SIO control register.
SSI	Synchronous serial interface unit.
ssic	SSI control register.
ssid	SSI data register.
TIMER	Programmable timer unit.
timer<0—1>	Timer running count registers.
timerc	Timer control register.
TRACE	Program discontinuity trace buffer.
wdogr	Watchdog timer register.
XAB	X space (program space) address bus.
XDB	X space data bus.
YAB	Y space (data space) address bus.
YDB	Y space data bus.

* IOPA and IOPD are not available for the 28-pin SOJ package.
† For *FlashDSP* 1609F only.
‡ For DSP1609 only.

4 Hardware Architecture (continued)

4.1 DSP1609 Architectural Overview (continued)

4.1.1 DSP1600 Core

The DSP1600 core is the heart of the DSP1609 device. The core consists of a data arithmetic unit (DAU), two address arithmetic units (XAAU and YAAU), an instruction cache, and a control section. The core provides support for internal dual-port RAM, and features vectored interrupts and a trap mechanism. (For more information, see Section 4.2 on page 20.)

4.1.2 Dual-Port RAM (DPRAM)

This module contains two banks of zero wait-state memory in the DSP1609. It consists of 2K 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 restriction. The DSP1600 core automatically performs the required multiplexing. In the event that references to both ports of a single bank of DPRAM 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 external memory into DPRAM and then be executed without wait-states. DPRAM is also useful for improving convolution performance in cases where the coefficients are adaptive. Because 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 through the JTAG port.

4.1.3 Read-Only Memory (ROM)

The *FlashDSP*1609F contains 24K of 16-bit words of zero wait-stated, user-programmable flash ROM for program and fixed coefficients. The DSP1609 contains 24K of 16-bit words of conventional ROM. A mask-programmable secure option is available on production chips, that prohibits reading out the ROM contents externally. This is accomplished by prohibiting the selection of memory map 3.

4.1.4 Timers

Two interrupt timers are provided in the DSP1609. They are used to provide an interrupt at the expiration of a programmed interval. The interrupt may be a single interrupt or a repetitive interrupt. TIMER0 runs on either the free-running core clock (CLKFREE) or low-frequency clock (CLKLOW), has a clock prescaler, and supports over nine orders of magnitude of interval selection. TIMER1 runs on either the free-running core clock (CLKFREE) or low-frequency clock (CLKLOW), has a prescaler, and supports over nine orders of magnitude of interval selection. The timers may be stopped and restarted at any time. For more information, see Section 4.9.

4.1.5 Watchdog Timer

A watchdog timer can be used to protect from catastrophic loss of control of the DSP. It can be programmed for one of three time-out intervals. The watchdog timer clock is selectable as the divided-down (divide by 128 only) input clock or the internal ring oscillator output. For more information, see Section 4.10.

4.1.6 Input/Output Ports (IOP)

Three 8-bit IOP units (IOPA, IOPB, IOPC) and one 4-bit IOP unit (IOPD) for the 44-pin device provide convenient and efficient monitoring and control of 28 individually configurable pins. The 28-pin DSP1609 has only two IOP units (IOPB and IOPC). When configured as outputs, the pins can be individually set, cleared, toggled, or left unchanged. When configured as inputs, the entire port can be read (those configured as inputs or multiplexed, as well as those configured as outputs). Note that some of the pins of the IOP units are multiplexed with other functions. See Section 4.8.

4.1.7 JTAG

The JTAG section contains logic that implements the JTAG/*IEEE* * P1149.1 standard four-signal test port. No boundary-scan register is included. The JTAG port provides a mechanism for the DSP1600 core to communicate with remote test equipment or a remote hardware development system (DSP1600 HDS or *FlashDSP*1600 HDS). The JTAG port also supports program, memory, and register upload/download, and execution start/stop.

A 4-bit instruction register, a bypass register, and a device identification register have been implemented (see Table 42 on page 60). The instructions for accessing the ID and BYPASS registers are 0xE (1110) and 0xF (1111), respectively. There is no separate TRST input pin.

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

4 Hardware Architecture (continued)

4.1 DSP1609 Architectural Overview (continued)

4.1.8 Synchronous Serial Interface Units (SSI)

One independent SSI unit is compatible with the SPI interface of the *Motorola** 68HC11 microncontroller. All the features of the 68HC11 SPI interface are supported except the open collector mode of operation and slave-initiated transfers, which are not supported. The DSP core can be interrupted upon completion of an SSI data transfer. The SSI pins are multiplexed with IOPC, under control of a programmable bit in the SSI. For more information, see Section 4.7.

4.1.9 Clock Generation

The clock structure for the DSP1609 incorporates a wide range of options, including a crystal oscillator, a PLL, a low-frequency divider, and an internal ring oscillator.

The DSP core receives a 2X DSP core clock (two times MIPS rate) that is divided by two inside the core to generate the DSP's non-wait-stated and wait-stated clocks.

It includes one low-power internal ring oscillator and a crystal oscillator that connects to an external crystal. The **clkc** register bits 7—3 (see Table 37 on page 57) determine which clock is selected to run the core and peripherals.

The DSP1609 28-pin SOJ and 44-pin PLCC packages each provide an output clock pin, DOUT. As shown in Table 4, **clkc** register bits 8—11 select one of the DOUT MUX pin output functions.

Table 4. DOUT Pin Output Functions

clkc Register Bits 11—8	DOUT MUX[3:0] Pin Output Function
0000	Ring oscillator (CLKRING) selected to DOUT
0001	Input crystal/clock (CLKIN) selected to DOUT
0010	Low-frequency clock (CLKLOW) selected to DOUT
0011	Wait-stated DSP (CLKWAIT) clock selected to DOUT
0100	Free-running DSP (CLKFREE) clock selected to DOUT
0101	Reserved
0110	Logic 0 selected to DOUT
0111	Logic 1 selected to DOUT
1xxx	Reserved

4.1.10 Dual-Channel Serial I/O Port (SIO)

The SIO provides a serial interface to codecs or other external devices.

See Section 4.11 for more information.

4.1.11 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.

4.1.12 DRAM Control Interface (DRC)

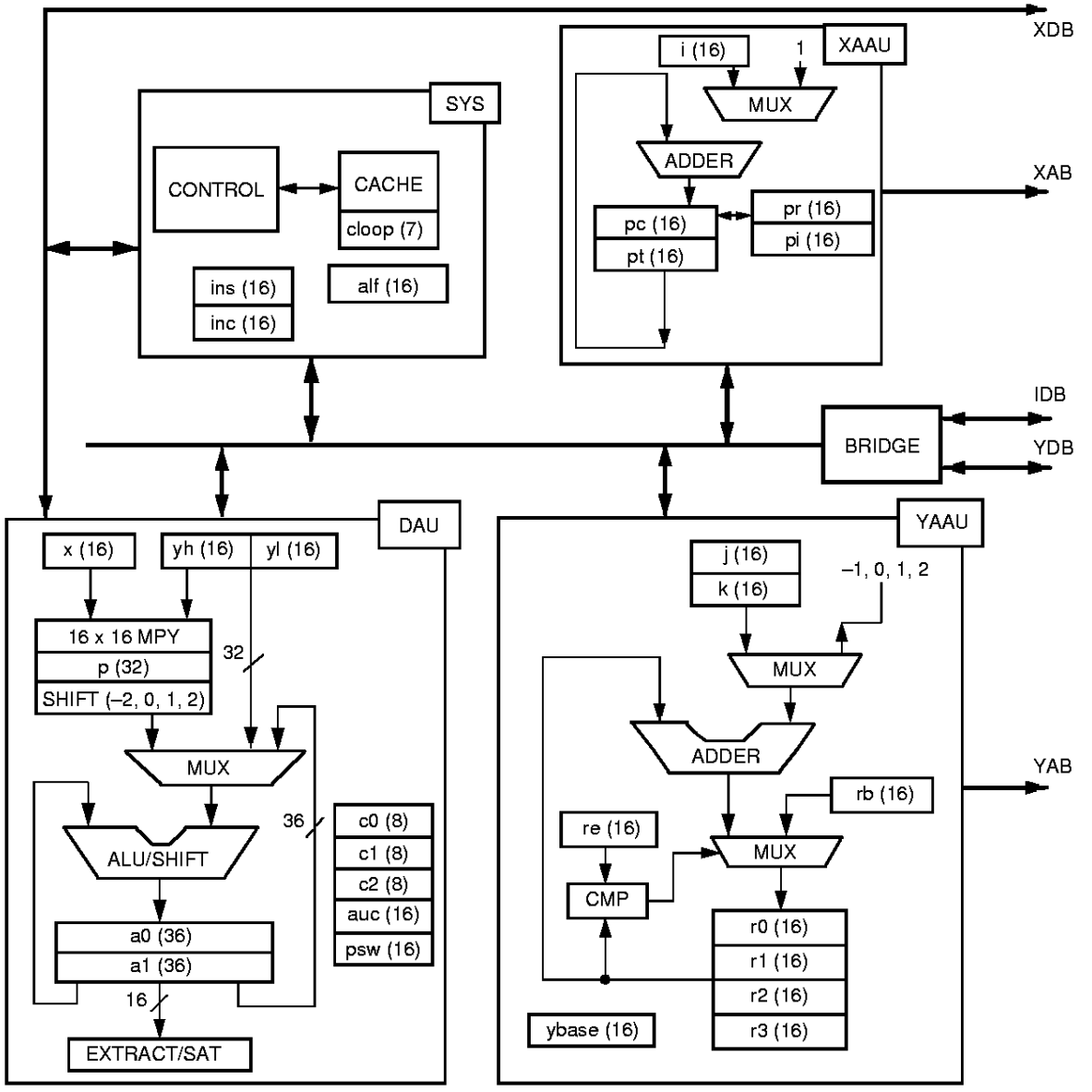
The DRC provides CASN, RASN, and RWN signals during refresh operations. DRAM read/write operations must be programmed through IOPD<0—2>. The data bus and address bus must be configured to use the other IOP ports as the data and address bus to the external DRAM.

* *Motorola* is a registered trademark of Motorola, Inc.

4 Hardware Architecture (continued)

4.2 DSP1600 Core Architectural Overview

Figure 5 shows a block diagram of the DSP1600 core, which consists of the modules described in Sections 4.2.1 through 4.2.4. Table 5, on page 21, contains a legend for the DSP1600 core block diagram.



5-1741e(F)

Figure 5. DSP1600 Core Block Diagram

4 Hardware Architecture (continued)

4.2 DSP1600 Core Architectural Overview (continued)

Table 5. DSP1600 Core Block Diagram Legend

Symbol	Name
16 x 16 MPY	16-bit by 16-bit multiplier
a0—a1	Accumulators 0 and 1 (16-bit halves specified as **a0**, **a0l**, **a1**, and **a1l**)*
alf	Low-power standby mode and memory map control
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
IDB	Internal data bus
inc	Interrupt control
ins	Interrupt status
j	Increment register
k	Increment register
MUX	Multiplexer
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**.

4 Hardware Architecture (continued)

4.2 DSP1600 Core Architectural Overview (continued)

4.2.1 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 sequences downloading through JTAG of self-test programs to internal DPRAM.

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

4.2.2 Data Arithmetic Unit (DAU)

The data arithmetic unit (DAU) contains a 16 x 16 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-bit 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-bit or 32-bit microprocessor for logical and control operations.

The user also has access to two additional DAU registers. The **psw** register contains status information from the DAU. The arithmetic control register, **auc**, is used to configure some of the features of the DAU.

4.2.3 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 pointer 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 internal DPRAM.

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 a Y space memory address and also specifies one of 16 registers for source or destination.

4.2.4 X Space Address Arithmetic Unit (XAAU)

The XAAU supports high-speed, register-indirect instruction/coefficient memory addressing with postmodification of the register. The **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.

All of the XAAU registers and the adder for increments are 16 bits wide. The XAAU decodes the 16-bit instruction/coefficient address and produces signals for the appropriate X memory segment. The addressable X segments are internal ROM and internal DPRAM.

The locations of these memory segments depend upon the memory map selected. A security mode can be selected by mask option. This prevents unauthorized access to the contents of internal ROM.

4.3 Interrupts, Trap, and Low-Power Standby Mode

The DSP1609 supports vectored interrupts and a trap. The device has eight internal hardware sources of program interrupt and one external interrupt pin. As shown in Table 6 on page 23, each source of an interrupt and trap has been assigned a unique vector address.

Vectored interrupts are enabled in the **inc** register (see Table 39 on page 58) and monitored in the **ins** register (see Table 40 on page 58).

4 Hardware Architecture (continued)

4.3 Interrupts, Trap, and Low-Power Standby Mode (continued)

Table 6. Interrupt Vectors

Source	Vector	Priority	Issued By
No interrupt	0x0	—	—
Software interrupt	0x2	1 (lowest)	icall
SW3	0x1	2	—
JINT	0x42	3	JTAG interrupt
TIME0	0x4	4	TIMER0
SW4	0x8	5	—
IOPA	0xc	6	IOPA interrupt
IBFB (1609F) or SIOB (1609)	0x10	7	SIO (Channel B) input buffer full
INTB	0x14	8	INTB pin
OBEB (1609F) or SW5 (1609)	0x18	9	SIO (Channel B) output buffer empty
No interrupt	0x1c	—	—
SSI	0x20	10	SSI
SW6	0x24	11	—
SW7	0x28	12	—
IBF (1609F) or SIOA (1609)	0x2c	13	SIO (Channel A) input buffer full
OBE (1609F) or SW1 (1609)	0x30	14	SIO (Channel A) output buffer empty
TIME1	0x34	15	TIMER1
SW2	0x38	16	—
Trap from JTAG	0x3	17	JTAG/HDS

4.3.1 Interruptibility

Vectored interrupts are serviced only after an interruptible instruction or the completion of a prior interrupt service routine. If more than one vectored interrupt is asserted at the same time, the interrupts are serviced sequentially according to their assigned priorities. 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 because 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 on an interruptible instruction or a prior interrupt service routine) before an enabled vectored interrupt can be acted on. An interrupt cannot be serviced if it is not enabled. Polled interrupt service can be implemented by disabling the interrupt in the **inc** register and polling the **ins** register for the expected event.

4.3.2 Vectored Interrupts

A logic 1 written to any bit of the **inc** enables (or unmasks) the associated interrupt. If the bit is cleared to a logic 0, the interrupt is disabled (or masked). The occurrence of an interrupt that is not masked causes the program execution to transfer to the memory location pointed to by that interrupt's vector address, assuming no other interrupt is being serviced.

The occurrence on an interrupt that is masked causes no automatic processor action, but it does set the corresponding status bit in the **ins** register. If an interrupt occurs while it is masked, it is latched. Subsequently, unmasking the interrupt causes it to be serviced as soon as the processor reaches an interruptible state. The status of the interrupt sources is readable in the **ins** register even if the interrupt is masked in the **inc** register.

4.3.3 External Interrupt Pin (INTB)

INTB is a negative edge-triggered interrupt pin. External hardware must guarantee that the interrupt service routine has completed before issuing another interrupt on INTB. Any activity on INTB is **not** recognized during the interrupt service routine and must be reissued after the completion of the interrupt service routine. Because there is no interrupt acknowledge pin on the DSP1609, the interrupt service routine must communicate with the external hardware through other means. One possible way of performing this task would be through the use of IOP pins.

4 Hardware Architecture (continued)

4.3 Interrupts, Trap, and Low-Power Standby Mode (continued)

4.3.4 Clearing Interrupts

The IOPA interrupt is cleared by reading the **sbita** register. The JTAG interrupt (JINT) is cleared by reading the **jtag** register.

Other interrupts are cleared by either of the following methods:

- After a vectored interrupt has been serviced, it is cleared when the **ireturn** instruction is issued, leaving set any other vectored interrupts that are pending.
- In the **ins** register, writing a 1 to the bit that is associated with the interrupt. Writing a 0 to the **ins** register does nothing.

Once an interrupt is cleared, the corresponding bit in the **ins** register is cleared.

4.3.5 Power-Saving Modes

Three primary power-saving modes are available. These modes are described below in order of increasing power savings:

Low-Frequency Clock Mode—Switching the core clock to the lowest frequency possible for as long as possible is the easiest way to conserve power, since power consumption is directly linked to the instruction cycle rate. The low-frequency clock source to the DSP is controlled by register bits **clkc**[7:4]. The appropriate selection of the low-frequency clock depends on the application requirements. If an application requires an accurate real-time clock, the internal ring oscillator cannot be used.

Low-Power Standby Mode—Setting the AWAIT bit in the **alf** register causes the processor to go into a standby mode in which program execution is halted and the internal wait-stated DSP clock is turned off. The occurrence of any interrupt that is enabled in the **inc** register is processed and returns the processor to the previous state, allowing program execution to continue.

Stop Clock Mode—Setting the STOPCLK bit in the **clkc** register causes the processor to go into a sleep mode in which program execution is halted and the clock to the DSP core is turned off. An IOPA interrupt, an INTB pin interrupt, or a Timer0 or Timer1 time-out interrupt when running off of the low-frequency clock clears the STOPCLK condition and allows program execution to continue. (The desired pin interrupts must be selected with the appropriate multiplexing controls in order to use these interrupts for exiting stop clock mode.) The settings of the **inc** register have no effect on the clearing of STOPCLK.

4.4 Memory Maps and Wait-States

The DSP implements a modified Harvard architecture that has separate internal 16-bit address and data buses for the instruction/coefficient (X) and data (Y) memory spaces. The DSP1609 contains 24 Kwords of flash ROM and two banks of 2 Kwords of RAM.

The instruction/coefficient memory map is configurable to provide application flexibility.

Table 7 shows the instruction/coefficient memory maps available for the DSP1609/*FlashDSP*1609F.

The data memory maps are divided into segments as shown in Table 8. The selection of a segment is automatic depending on the address in the YAAU.

The internal RAM can be accessed by both the Y space and the X space. If the same bank of internal RAM is accessed from both memory spaces simultaneously, one wait-state is added, and the Y space is accessed first.

4.4.1 Instruction/Coefficient Memory Map Selection

In determining which memory map to use, the processor evaluates the state of 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 0xC000. If LOWPR is high, internal dual-port RAM begins at address 0x0. LOWPR also moves IROM from 0x0 in MAP1 to 0x4000 in MAP3.

The Lucent development system tools using the JTAG port can independently set the memory map. Specifically, during a trap from JTAG, the memory map is forced to MAP1. The map selection made prior to the JTAG trap is restored when the trap service routine has completed execution.

4 Hardware Architecture (continued)

4.4 Memory Maps and Wait-States (continued)

4.4.1 Instruction/Coefficient Memory Map Selection (continued)

Whenever the device is reset using the RSTB pin, the default memory map is MAP1. A reset through the JTAG port does not reinitialize the **alf** register, so the previous memory map is retained.

4.4.2 Data Memory Map Selection

Table 8 shows the data memory maps.

Table 7. Instruction/Coefficient Memory Map (X Memory Space)

Decimal Address	Hexadecimal Address in PC, pt, pi, pr	MAP1 (LOWPR = 0)*	MAP3† (LOWPR = 1)
0 2047	0x0000 0x07FF	ROM	RAM1, 2
2048 8191	0x0800 0x1FFF		Reserved
8192 16,383	0x2000 0x3FFF		
16,384 24,575	0x4000 0x5FFF		ROM
24,576 32,767	0x6000 0x7FFF	Reserved	
32,768 40,959	0x8000 0x9FFF		
40,960 49,151	0xA000 0xBFFF		Reserved
49,152 51,199	0xC000 0xC7FF	RAM1, 2	
51,200 57,343	0xC800 0xDFFF	Reserved	
57,344 65,535	0xE000 0xFFFF		

* LOWPR is an alf register bit. The Lucent development system tools can independently set the memory map.
† MAP3 is not available if the mask-programmable secure option is selected.

Table 8. Data (Y) Memory Map

Decimal Address	Address in r0, r1, r2, r3	Segment
0 — 2047	0x0000 — 0x07FF	RAM1, 2
2048 — 65,535	0x0800 — 0xFFFF	Reserved

4 Hardware Architecture (continued)

4.5 Clock Generation

There are multiple options for clock generation in the DSP1609. The device includes a crystal oscillator to be used with an external crystal, but can also accept a direct-driven input clock. (In real applications, the crystal oscillator clock is always selected by default.) An internal independently enabled PLL is also provided, allowing the input clock to be multiplied up to provide a high-frequency 2X core clock of up to 140 MHz. If the PLL is not used, then a 2X clock (2 times the MIP rate) must be provided externally. This 2X clock, whether provided externally or generated internally by the PLL, is internally divided by two to generate the DSP core clock. An internal low-power oscillator is also provided which can be used to clock the DSP core for power savings.

4.5.1 Functional Overview

The clock generator for the DSP1609/*FlashDSP*1609F incorporates a wide range of options, including a crystal oscillator, PLL, low-frequency divider, and an internal ring oscillator. The clock generates the outputs shown in Table 9.

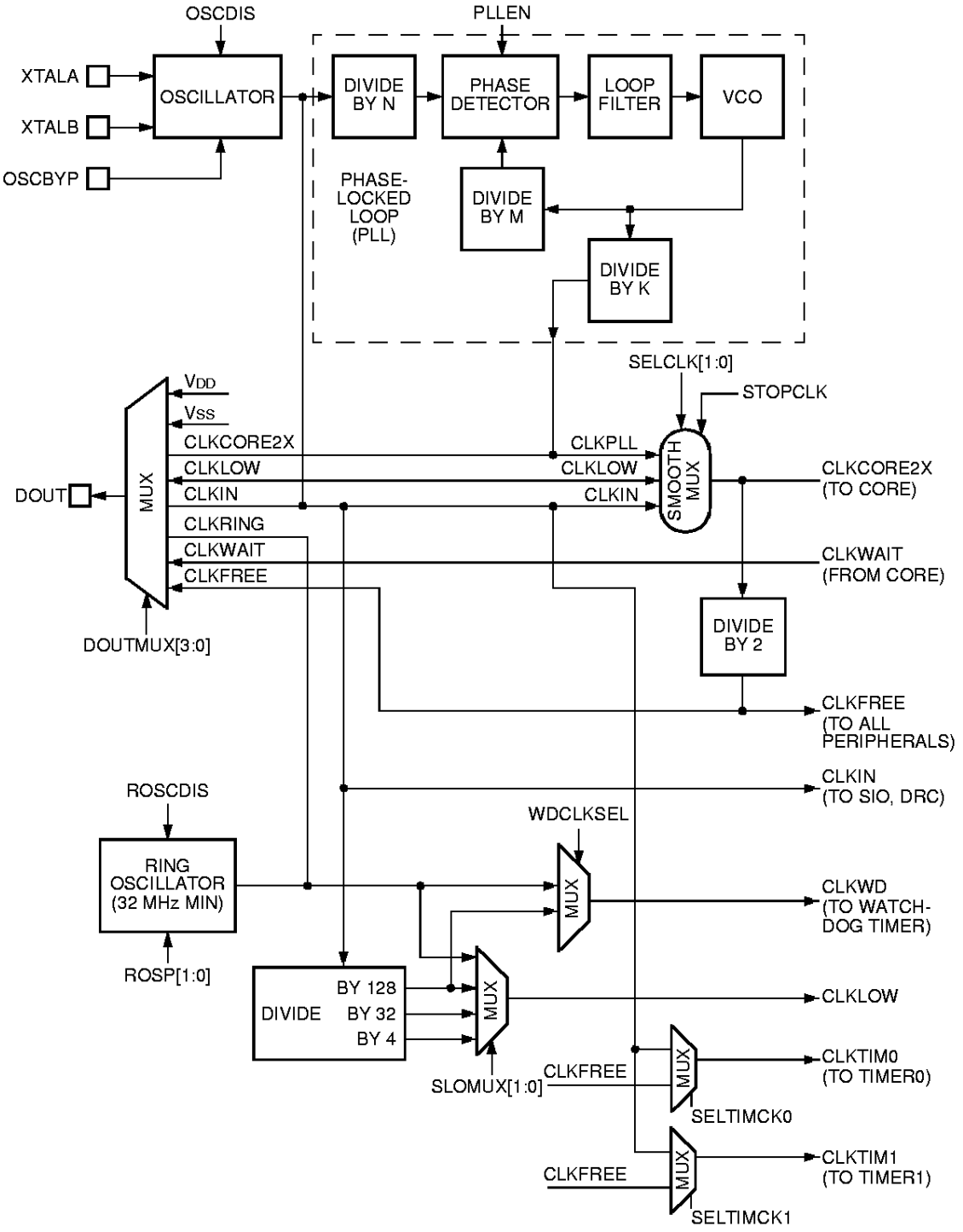
Table 9. Clock Options

Clock Option	Signal	Generated by/Output
Input Crystal/ Clock	CLKIN	Output of the oscillator. The crystal oscillator circuit provides the device input clock and is controlled by the OSCBYP input pin.
Ring Oscillator	CLKRING	The internal ring oscillator defaults to a minimum of 32 kHz. Even though the internal oscillator frequency is not exact, it may be measured against the crystal oscillator output and coarsely adjusted by setting the ROSP[1:0] bits in the **chipo** register (see Table 36). The **clkc** register (see Table 37) controls most of the configuration settings in the clock generator.
Wait-stated DSP Clock	CLKWAIT	The wait-stated clock generated by the core.
PLL Clock	CLKPLL	The PLL consists of the phase detector, loop filter, voltage-controlled oscillator (VCO), M divider, K divider, and N divider and provides output clock frequencies ranging from 50 MHz to 140 MHz. The value of M, K, and N are set in the **pllc** register (see Table 44).
DSP 2X Core Clock	CLKCORE2X	Generated by the PLL, low-frequency clock, input clock, or ring oscillator and output to clock the core.
Stop Clock	—	A power-saving mode that completely turns off CLKCORE2X. Any reset (powerup, pin, or JTAG), IOPA interrupt, INTB, TIMER1 or TIMER0 interrupt clears STOPCLK.
Free-running DSP Clock	CLKFREE	Generated by the core clock (the 2X core clock divided by two) and output to all the peripherals. The timers count cycles of CLKFREE divided by a prescaler that is set in the **timerc** register (see Table 50). CLKFREE is also the MIPS rate of the DSP.
Watchdog Clock	CLKWD	Generated by the input clock, divided by 128, or the ring oscillator and output to the watchdog timer. The watchdog timer clock is selectable between CLKIN divided by 128 or CLKRING. Its time-out period is specified by bits WDEN[1:0] in the **chipo** register (see Table 36).
DOUT	—	Provides digital output depending on the setting of DOUTMUX[3:0] in the **clkc** register (see Table 37).
Timer 0 Clock	CLKTIM0	Multiplexor control signal: SELTIMCKO, from the **timerc** register (see Table 50).
TImer 1 Clock	CLKTIM1	Multiplexor control signal: SELTIMCKI, from the **timerc** register (see Table 50).

4 Hardware Architecture (continued)

4.5 Clock Generation (continued)

4.5.1 Functional Overview (continued)



5-6144 (F)

Figure 6. Clock Generation Overview

4 Hardware Architecture (continued)

4.5 Clock Generation (continued)

4.5.2 Core Clock Switching

When any of the smooth MUX control bits are changed, because of synchronization issues, a delay is incurred before the clock switch takes place. Five conditional flags can be checked via software to determine when the clock change actually takes place. These five conditionals are pllon, plloff, slowon, slowoff, stopclk. These five flags indicate that:

- CLKPLL currently is or is not the output of the smooth MUX.
- CLKLOW currently is or is not the output of the smooth MUX.
- STOPCLK main clock for DSP core turned off.

When the SELCLK control bits (**clkc**[5:4]) are changed, resulting in the smooth MUX switching the source for the core clock, there is a maximum time for the clock switch to occur (see Table 10). The latencies are defined in terms of input clock cycles (CLKIN period) and/or low-frequency clock cycles (CLKLOW period). In addition, a stabilization time is required for the entire clock path of the new source clock before that clock is selected via the SELCLK control bits (see Table 11).

Table 10. Clock Switch Latencies

Source Clock Before Switch	Source Clock After Switch	Maximum Clock Cycles for Switch
CLKIN	CLKPLL	[((N + 1) • 2.5) + ((K + 1)/(M + 2)) + 1.5]CLKIN Period
CLKIN	CLKLOW	1.5 CLKLOW Period
CLKIN	CLKCORE2X stopped	1.5 CLKIN Period
CLKLOW	CLKIN	1.5 CLKLOW Period + 1.5 CLKIN Period
CLKLOW	CLKPLL	[((N + 1) • 2.5) + ((K + 1)/(M + 2)) + 1.5]CLKIN Period + 1.5 CLKLOW Period
CLKLOW	CLKCORE2X stopped	1.5 CLKLOW Period
CLKPLL	CLKIN	[((N + 1) • 3) + 0.5](CLKIN Period)
CLKPLL	CLKLOW	1.5 CLKLOW Period
CLKPLL	CLKCORE2X stopped	[((N + 1) • 2.5) + ((K + 1)/(M + 2)) + 1.5]CLKIN Period
CLKCORE2X stopped	CLKIN	1.5 CLKIN Period
CLKCORE2X stopped	CLKLOW	1.5 CLKLOW Period
CLKCORE2X stopped	CLKPLL	[((N + 1) • 2.5) + ((K + 1)/(M + 2)) + 1.5]CLKIN Period

Table 11. Core Clock Stabilization Requirements

Source Clock to be Selected	Condition	Required Stabilization Time Prior to Selecting Source Clock
CLKIN*	Oscillator just enabled	10 ms (preliminary)
CLKRING	Ring oscillator just enabled	100 µs (preliminary)
CLKPLL	PLL just enabled	50 µs (preliminary)
CLKPLL	Any change to **pllc** register bits	50 µs (preliminary)
CLKLOW	Any change to SLOWMUX[1:0] bits	1 instruction cycle

* Assumes that OSCBYP = 0.

4 Hardware Architecture (continued)

4.6 DRAM Control Interface

This DRAM controller (DRC) provides the refresh signals, i.e., CASN and RASN, to support external DRAM in the 44-pin DSP1609/*FlashDSP*1609F. Meanwhile, the normal read/write operations to access DRAM can be accomplished through programming IOPs. The period of refresh signals generated by DRC can be programmed to various values, from 2 to 1024 times the period of clock XTALA. If XTALA is connected to a 4.096 MHz crystal input, the refresh period will be from 0.4882 µs to 249.96 µs with a resolution of 0.4882 µs.

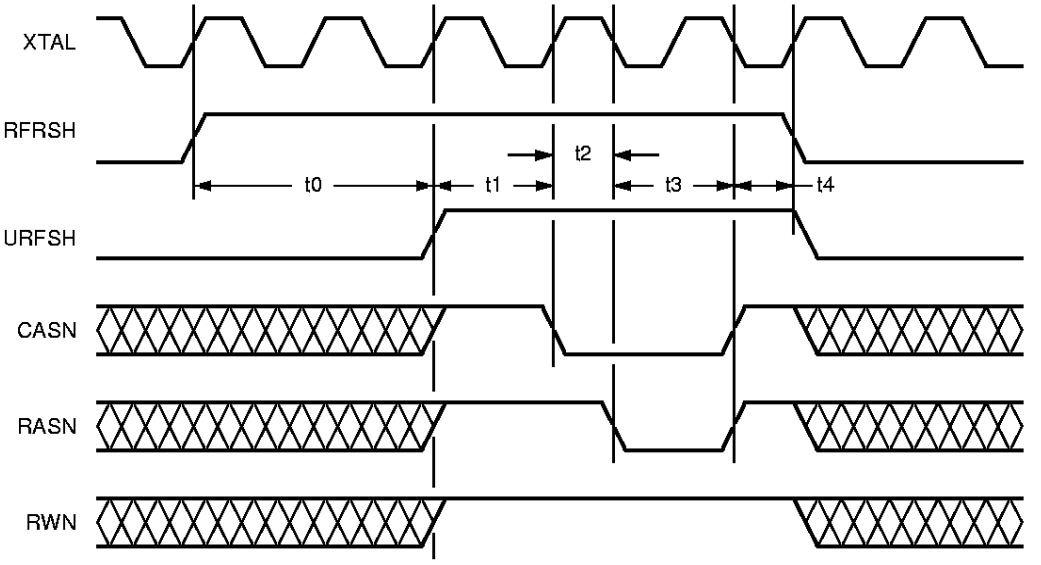
In the refresh operation, a flag signal RFRSH generated by DRC is set to 1 and sent to DSP CORE to prevent an IOP instruction being issued to IOPD. Meanwhile, a MUX control signal is set and sent to IOPD so that IOP0 and IOP1 become signals CASN and RASN, and IOPD2 (RWN) is pulled up to 1. Furthermore, the flag RFRSH must be set to 1 before the MUX control signal is set to 1 because the delay (t0) between signal RFRSH and MUX control signal has to be larger than the time needed to perform one DRAM read or write. This gap can be programmed to be 0 to 63 times of the period of crystal input. When the MUX control is disabled, IOPD is in normal I/O port mode in which IOPD can be programmed to access DRAM.

4.6.1 DRC Refresh Timing

The time intervals in Figure 7 and Figure 8 are specified in Table 12. The timing diagram is shown in Figure 7.

Table 12. DRC Refresh Timing

Interval	Value
t0	FLSEL[5:0] • period of XTALA.
t1	A period of XTALA.
t2	One-half of a period of XTALA.
t3	A period of XTALA.
t4	One-half of a period of XTALA.
t5	RFSHSEL[8:0] • 2 • period of XTALA.



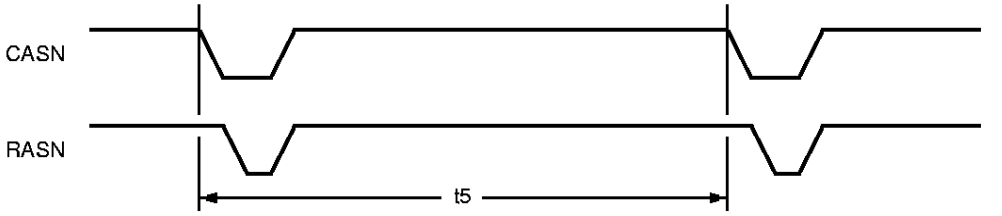
5-6145 (F)

Figure 7. DRAM Refresh Timing

4 Hardware Architecture (continued)

4.6 DRAM Control Interface (continued)

4.6.1 DRC Refresh Timing (continued)



5-6146 (F)

Figure 8. DRAM Refresh Period

4.6.2 DRC Refresh Programming Procedure

Determine the **drc** register value (see Table 38 on page 58). DRCEN should be set to 1 to turn on the DRC. RFSHSEL is the round-off of the refresh period (t5 in µs • 4.096 • 2). For example, RFSHSEL is 0x7B when the required refresh period (t5) is 60.06 µs. FLSEL is the ceiling of time (t0 in µs) to execute one read/write operation • 4.096. The read/write operation is performed by programming IOPD0 (CASN), IPD1 (RASN), IOPD2 (RWN) and other IOP ports that are chosen to be the data bus and address bus, these operations may vary on different types of DRAMs and depend on the configuration of the DSP1609. The time to execute one read/write operation must be calculated from the number of cycles needed by the IOP read/write operation for each configuration. For example, if the time for read/write operation is 0.976 µs, the FLSEL should be set to 4.

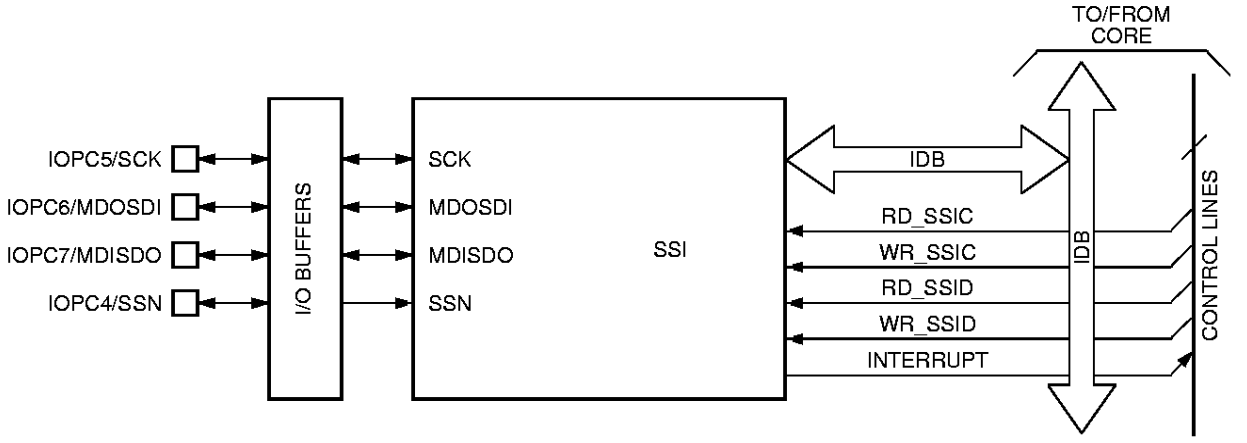
```
   #include "1609.h"
   .rsect ".ram"
   goto start
   start: auc=0
   drc=0x887B                /*Refresh period is 60.06 µs, FLSEL is "4" */

/*read operation */
   loop:
   if rfrsh goto loop       /*check rfrsh if DRAM is under refreshing */
   program CASN (IOPD0), RASN (IOPD1), RWN (IOPD2), other IOP ports for DATA bus and
   address bus to perform read operation

/*write operation */
   if rfrsh goto loop      /*check rfrsh if DRAM is under refreshing */
   program CASN (IOPD0), RASN (IOPD1), RWN (IOPD2), other IOP ports for DATA bus and
   address bus to perform write operation
```

4 Hardware Architecture (continued)

4.7 Synchronous Serial Interface (SSI)

The SSI can be programmed in the master or slave mode. The control word can be written to a 16-bit control register, **ssic**, to configure the SSI. The byte to be transmitted is written into the 8-bit data register, **ssid**. As a master, the SSI initiates serial transmission of a byte. This occurs whenever the master writes a byte to the data register (provided there is no transmission in progress). As a slave, it waits until the master initiates the start of transfer. In either case, transmitting and receiving occur simultaneously at the master and slave.

The byte from the data register of the master is shifted into the data register of the slave and vice versa. The two registers operate as a 16-bit circular shift register, wherein the most significant bit (MSB) of the master data register is simultaneously shifted into the least significant bit (LSB) of the slave data register and the MSB of the slave data register is shifted into the LSB of the master data register. The DSP1609 reads the received byte as a 16-bit word. The lower byte is the received data, and it is sign extended into the upper byte. The shift clock is provided by the master. Depending on the polarity of the shift clock and the phase of the transfer relative to the shift clock, the master operates in one of four different modes of transfer. These modes are under program control and must match the mode used by the peripheral device with which the SSI communicates. In addition, there are seven different clock rates that the master can choose from as the rate at which the bits are to be shifted out. The status of the transfer is indicated by three status bits, included as read-only bits in the control register. These indicate end of transfer and any errors that may have occurred.



Note: Table 13 on page 32 describes the SSI pins.

5-6147 (F)

Figure 9. SSI Interconnections

4 Hardware Architecture (continued)

4.7 Synchronous Serial Interface (SSI) (continued)

Table 13. SSI Pin Descriptions

Pin	Description
SCK	**SSI Clock**. Bidirectional. If SSI is configured in master mode, this pin is an output providing a clock to the slave devices. If SSI is configured in slave mode, this pin is an input that takes in the serial clock. This pin is multiplexed with IOPC5 (if the **ssic** register disables the SSI (bit 15 = 0), then this pin is used for IOPC5).
MDOSDI	**Master Data Out/Slave Data In**. Bidirectional. If the SSI is configured in master mode, this pin is the serial data output. If the SSI is configured in slave mode, this pin is the serial data input. This pin is multiplexed with IOPC6 (if the **ssic** register disables the SSI (bit 15 = 0), then this pin is used for IOPC6).
MDISDO	**Master Data In/Slave Data Out**. Bidirectional. If SSI is configured in master mode, this pin is an input and samples serial data supplied by the master device. If SSI is configured in slave mode, this pin is the serial data output. This pin is multiplexed with IOPC7 (if the **ssic** register disables the SSI (bit 15 = 0), then this pin is used for IOPC7).
SSN	**Serial Select**. Input. Negative assertion. When SSI is configured as a slave, the assertion of SSN signals the slave that is being addressed to transfer data with the master device. If this pin is asserted while SSI is configured as a master, a mode fault will be detected by SSI. This pin is multiplexed with IOPC4 (if the **ssic** register disables the SSI (bit 15 = 0), **or** if the **ssic** register configures the SSI for master mode (bit 14 = 1) and the SSN input is disabled (bit 10 = 0), then this pin is used for IOPC4).

The SSI has two programmable registers, one 16-bit register for control and one 8-bit register for data. The 16-bit control register, **ssic**, enables and configures the SSI for serial communication in the desired mode. It has control fields (the EN, MSTR, SPOL, SPHA, SDOEN, SSNEN, and SCLK bits of the **ssic** register) and status fields (the SDONE, WCOLL, and MODF bits of the **ssic** register). The 8-bit data register, **ssid**, is used for writing the byte to be transmitted and reading the received byte.

Table 48 on page 64 shows the register structure of the SSI control registers **ssic**. Brief descriptions of the function of each control bit are also given in the tables. Table 49 on page 65 shows the register structure of the SSI data register **ssid**.

4.7.1 SSI Operation

The SPOL and SPHA bits in the SSI control register determine the mode of data transfer. Both of these bits control the type of shift clock (SCK) generated. SPOL controls the polarity of SCK and SPHA determines the phase at which the serial transfer begins. The latter leads to a fundamentally different type of transfer with implications in situations where back-to-back byte transfer is required. The transfer formats may be different for different peripheral devices but remain unchanged during a transfer between the master and the slave device. The SSI is flexible enough to allow any desired configuration that conforms to the HC11 specifications.

The SSI generates an interrupt whenever a byte is successfully shifted in and copied to the read data buffer or when a mode-fault occurs. The interrupts are enabled by the appropriate bits in the **inc** register and their status is posted in the **ins** register.

The SSI control register includes three bits, SDONE, WCOLL, and MODF, that can only be read by the processor. Their values are written by the SSI. SDONE is a status flag that indicates the end of a transfer. The WCOLL flag indicates a write was attempted to the SSI data register while a transfer was in progress. The MODF bit indicates a mode fault. This occurs when the DSP1609 SSI is configured as a master and its SSN line is pulled low. Another error condition that can occur when the DSP1609 is configured as a slave and a transfer is aborted by pulling SSN high before the transfer is complete. This error condition is not indicated by the status flags.

4 Hardware Architecture (continued)

4.8 I/O Ports (IOP)

The DSP1609 contains three 8-bit IOP units and one 4-bit IOP unit, IOP<A—D>. Each IOP controls the directions of eight bidirectional control I/O pins, IOP<A—D>[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. See Table 14 for IOP operation.

4.8.1 IOP Operation

The lower half of each **sbit<a—d>** register (see Table 34 on page 54) contains the current value (VALUE[7:0]) of that register's eight bidirectional pins. The upper half of each **sbit<a—d>** register (DIR[7:0]) controls the direction of each pin independently. A logic 1 configures the corresponding pin as an output; a logic 0 configures it as an input. Reset clears the upper half of each **sbit<a—d>**, configuring all IOP<A—D> pins as inputs.

The **cbit<a—d>** registers (see Table 34 on page 54) each contain two 8-bit fields, MODE[7:0] and DATA[7:0]. Reset clears the values of DATA[7:0]. The meaning of a bit in either field depends on whether it has been configured as an input or an output in the corresponding **sbit<a—d>** register. If a pin has been configured to be an output, the meanings are MODE and DATA. For an input, MODE and DATA are used on some pins as multiplex controls, or for IOPA interrupt control. Table 14 shows the functionality of the MODE and DATA bits based on the direction selected for the associated IOP pin.

Table 14. IOP Operation

DIR[n*]	MODE[n*]	DATA[n*]	Action on IOP[n*]
1 (Output)	0	0	Clear
1 (Output)	0	1	Set
1 (Output)	1	0	No Change
1 (Output)	1	1	Toggle
0 (Input)	Don't Care	Don't Care	Input

* For IOP<A—C>: $0 \le n \le 7$; for IOPD: $0 \le n \le 3$.

If an IOP 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.

4.8.2 IOPA Interrupt Circuitry

The IOPA port has an interrupt generation capability to help perform functions like a keypad scan. Each IOPA bit can individually be configured to generate the IOPA interrupt or not. To configure a bit for interrupt, it should be set as an input, with its mode bit set to1. For example, to configure IOPA[1] for interrupt, set (**cbita**[9] = 1) and (**sbita**[9] = 0).

The interrupt generation logic compares the current state of the IOPA pins against a shadowed state which is loaded with the IOPA pins every time the **sbita** register is read. Whenever the current state and the shadowed state differ, an interrupt is generated. Which IOPA pin caused the interrupt can be determined by reading the **sbita** register. Reading sbita is also necessary to clear the interrupt and reload the shadowed state. Before any pins are enabled for interrupt, **sbita** should be read once to set the initial shadowed state to a defined value.

4 Hardware Architecture (continued)

4.8 I/O Ports (IOP) (continued)

4.8.3 Pin Multiplexing Control

Table 15. IOP Pin Multiplexing

IOP Pin	Multiplexed Signal	Condition for Selection of Multiplexed Signal
IOPA0	—	—
IOPA1	—	—
IOPA2	—	—
IOPA3	—	—
IOPA4	—	—
IOPA5	—	—
IOPA6	—	—
IOPA7	—	—
IOPB0	INTB	(**cbitb**[8] = 0) AND (**sbitb**[8] = 0)
IOPB1	OSCBYP	Latched on RSTB rising edge
IOPB2	SLDB	(**sioc**[15] = 1) AND (**sioc**[14] = 1)
IOPB3	JTSEL	Latched on RSTB rising edge
IOPB4	SLDA	**sioc**[15] = 1
IOPB5	SCLK	**sioc**[15] = 1
IOPB6	SDO	**sioc**[15] = 1
IOPB7	SDI	**sioc**[15] = 1
IOPC0	TMS	JTSEL* = 1
IOPC1	TCK	JTSEL* = 1
IOPC2	TDI	JTSEL* = 1
IOPC3	TDO	JTSEL* = 1
IOPC4	SSN	((**ssic**[15] = 1) AND (**ssic**[14] = 0)) OR (**ssic**[10] = 1)
IOPC5	SCK	**ssic**[15] = 1
IOPC6	MDOSDI	**ssic**[15] = 1
IOPC7	MDISDO	**ssic**[15] = 1
IOPD0	CASN	**drc**[15] = 1†
IOPD1	RASN	**drc**[15] = 1†
IOPD2	RWN	**drc**[15] = 1†
IOPD3	—	—

* JTSEL refers to the state of the JTSEL pin during the most recent device reset (RSTB held active-low).
† Refer to Section 4.6, DRAM Control Interface for details.

4 Hardware Architecture (continued)

4.9 Timers

The DSP1609 contains two timers, TIMER0 and TIMER1. TIMER0 is composed of three main blocks: the timer control register, the prescaler, and the timer itself. The timer control register, **timerc**, (see Table 50 on page 65) sets up the operational state of the timer and prescaler. The prescaler is a programmable divider that can be set to a count of 2 to 65,536. It provides a wide range of time delay. The timer itself is a 16-bit binary counter that can be preloaded with any 16-bit number. If enabled, the timer counts down at the programmed rate and generates an interrupt upon reaching zero.

If the TIMER0 interrupt is generated (see Table 6 on page 23), program control jumps to location 0x0004 where typically a branch to an interrupt service routine should be placed. Writing the **timer0** register sets the initial count into the timer and loads the period register with the same value for repeated count cycles.

The following functions are programmable in the **timerc** register:

- SELTIMCK*n* selects the clock source for the timers. If zero, the timer counts off of the low-frequency clock, CLKLOW. If one, the timer counts off of the free-running clock, CLKFREE.
- COUNT*n* starts TIMER*n* counting when set.
- RELOAD*n* enables repeated counts of TIMER*n* when set. If zero, TIMER*n* counts down once and stops. If one, TIMER*n* automatically reloads the previous starting value from the period register into the **timer*n*** register, and recommences counting down.
- PRESCALE*n* encodes the value of the divider of the clock going to the counter. For each timer it ranges from CLKTIM*n*/2 to CLKTIM*n*/65,536, where CLKTIM*n* is the clock source for each timer.

The timer interrupt can be individually enabled or disabled through the **inc** register. The timer can be stopped and started by software and can be reloaded with a new delay at any time. Its current value can also be read by software. When the DSP is reset, the timer is guaranteed to be in an inactive state.

The timer is normally run with two data move instructions, one to write the **timer0** register with the initial count, and the second to write the **timerc** register with initial values. Setting COUNTEN0 starts the counting of TIMER0.

When the DSP is reset, the control bits of the **timerc** register and the timer itself are initialized to 0. This sets the prescaler to CLKTIM0/2, turns off the reload feature, disables timer counting, selects the low-frequency clock to the timer, and initializes the timer value to its inactive state. The act of resetting the device does not cause a timer interrupt. The period register is not initialized on reset.

TIMER1 is the same as TIMER0 and is automatically powered down when TIMER*n* is not enabled.

4.10 Watchdog Timer

The watchdog timer allows for protection from catastrophic loss of control of the DSP by the software system. It can be programmed for one of three time-out intervals. The watchdog timer clock is nominally 32 kHz (selectable as the divided-down input clock—divided by 128 only—or the internal ring oscillator output).

When the watchdog timer is enabled, it can be programmed to time-out based on counting out one of three periods. This time period is the watchdog input clock period multiplied by 2^{12}, 2^{14}, or 2^{16}. It is specified by the WDEN[1:0] bits in the **chipo** register (see Table 36 on page 56).

When the watchdog timer is enabled and counting, the current watchdog count value is not viewable; however, the timer can be reset at any time by writing any value to the watchdog reset register, **wdogr**. If the watchdog timer ever counts out the specified time between **wdogr** writes, it initiates a chip reset and asserts the RSTB pin low. The fact that a watchdog chip reset has occurred may be determined by reading the WDRST bit in the **chipc** register (see Table 35 on page 55), which is set after such a reset takes place.

4 Hardware Architecture (continued)

4.11 Dual-Channel Serial I/O Port (SIO) for *FlashDSP* 1609F

The SIO interfaces to the external world through five pins: Serial Data Input (SDI), Serial Data Output (SDO), Serial Clock (SCLK), Serial Load A (SLDA), and Serial Load B (SLDB). The SIO pins are multiplexed with IOPB pins. When the SIO is enabled, the IOPB[7:4] pins are configured as SIO pins, i.e., SDI, SDO, SCLK, and SLDA; otherwise, these pins function normally as IOP pins. When the SIO is enabled in dual-channel mode (see Table 47, SIO Control Register (SIOC) Fields, on page 63), the IOPB2/SLDB pin is configured for SLDB operation; otherwise, this pin normally functions as the IOPB2 pin.

Serial data enters the SIO through the SDI pin and exits through the SDO pin. SCLK is the clock that controls shifting data in and out of the SIO. Data is shifted in and latched on the falling edge of SCLK, while data is shifted out on the rising edge of SCLK. SLDA and SLDB provide the strobes necessary for initiating serial transfers. For single-channel mode, only the SLDA pin is necessary; however, for dual-channel operation, SLDA and SLDB initiate transfers from channels A and B, respectively.

The SIO can be configured to support many serial protocols; however, all serial transfers are restricted to most significant bit (MSB) format. That is, the most significant bit in the data is always transmitted first. The serial input/output control (**sioc**) register configures the SIO for different transfer modes.

The SIO can be configured to operate in either active or passive mode, depending on the setting of the ACTIVE bit. In active mode, the SIO is the master and controls the serial interface by generating the serial clock and load pulses. In passive mode, the SIO is the slave, and control signals are supplied by the external serial device.

By default, the SIO is configured to support a single channel; however, the SIO can also be configured in a dual-channel mode that allows it to support two external serial devices. In this mode, serial transfers are time-division multiplexed between two channels; i.e., first serial data is transferred to/from channel A, and then serial data is transferred to/from channel B. Setting the DUAL bit in the **sioc** register configures the SIO for dual-channel operation.

SLDA and SLDB are the strobes that initiate serial transfers on the corresponding channels. In single-channel mode, only SLDA is used to initiate serial transfers. Both SLDA and SLDB can be configured to assert on either high-to-low or low-to-high transitions through the LDTA and LDTB bits.

In active mode, SLDA and SLDB can be configured to be delayed by one SCLK cycle. When the DLP bit is set, the load pulses occur at the same time as the first bit is being shifted in and out of the SIO. The DLP bit has no effect when the SIO is configured for passive-mode operation.

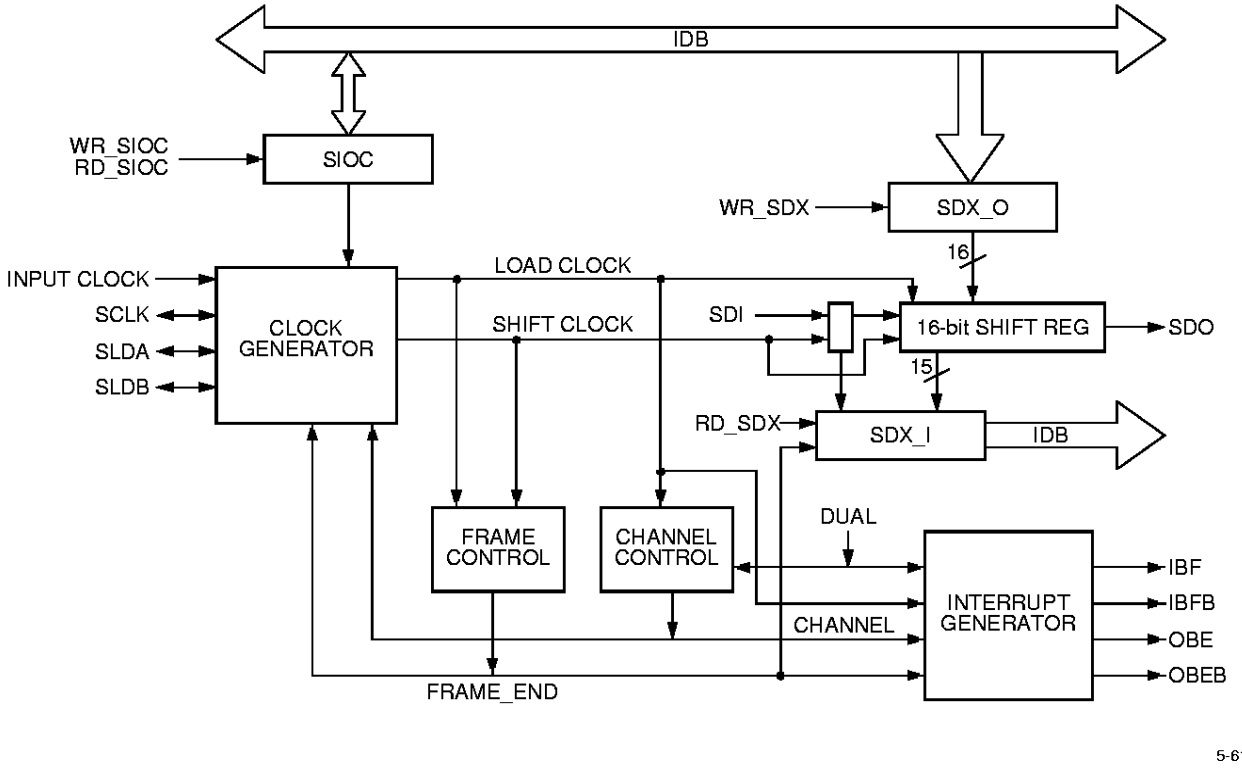
In active mode, SCLK is selectable between CLKIN and CLKIN/2. The load pulse frequencies are selectable as SCLK divided by 256 or 512.

4 Hardware Architecture (continued)

4.11 Dual-Channel Serial I/O Port (SIO) for *FlashDSP*1609F (continued)

4.11.1 *FlashDSP*1609F SIO Architecture

Figure 10 shows the block diagram of the *FlashDSP*1609F SIO.



5-6148 (F)

Figure 10. *FlashDSP*1609F SIO Block Diagram

The SIO data register (**sdx**) is composed of an output buffer (sdx_o) and input buffer (sdx_i). Data to be shifted out of the SIO is transferred from the DSP1600 core to the sdx_o buffer through the internal data bus (IDB). This data is then transferred from the sdx_o buffer to a 17-bit shift register for shifting data out onto the SDO pin. The 17-bit shift register duals as both the input and output shift registers. The extra bit is to handle the case of overlap between input and output serial transmissions. When a full 16 bits of serial data have been shifted in through the SDI pin, this data is latched into the sdx_i buffer. This data can then be transferred to the DSP1600 core through the IDB.

The SIO is configured by loading the **sioc** register through the IDB. The Clock Generator block uses this information to generate the Load Clock and Shift Clock signals for handling different serial transmission protocols. The Shift Clock signal is the clock to the shift register. The Load Clock signal is the strobe for initiating a serial transfer; this signal latches the contents of the sdx_o buffer to the shift register, clears the FRAME_END signal to enable the Shift Clock, and causes either the OBE or OBEB interrupts to be generated.

The Frame Control block contains a 4-bit binary counter that detects when 16 shifts have occurred. At the end of 16 shift cycles, the FRAME_END signal is pulsed, which stops the Shift Clock, transfers the serial input data to the sdx_i buffer, and causes either the IBF or IBFB interrupts to be generated.

The Channel Control block keeps track of which channel, A or B, that the SIO is communicating with in dual-channel mode. Whenever the Load Clock signal is strobed, the current channel state is toggled to indicate that serial transmission will occur with the other channel.

4 Hardware Architecture (continued)

4.11 Dual-Channel Serial I/O Port (SIO) for *FlashDSP* 1609F (continued)

4.11.2 *FlashDSP*1609F SIO Operation

Figure 11 illustrates that the operation of the SIO is active, single-channel mode for different load pulses. The strobe signal SLDA initiates a serial transfer.

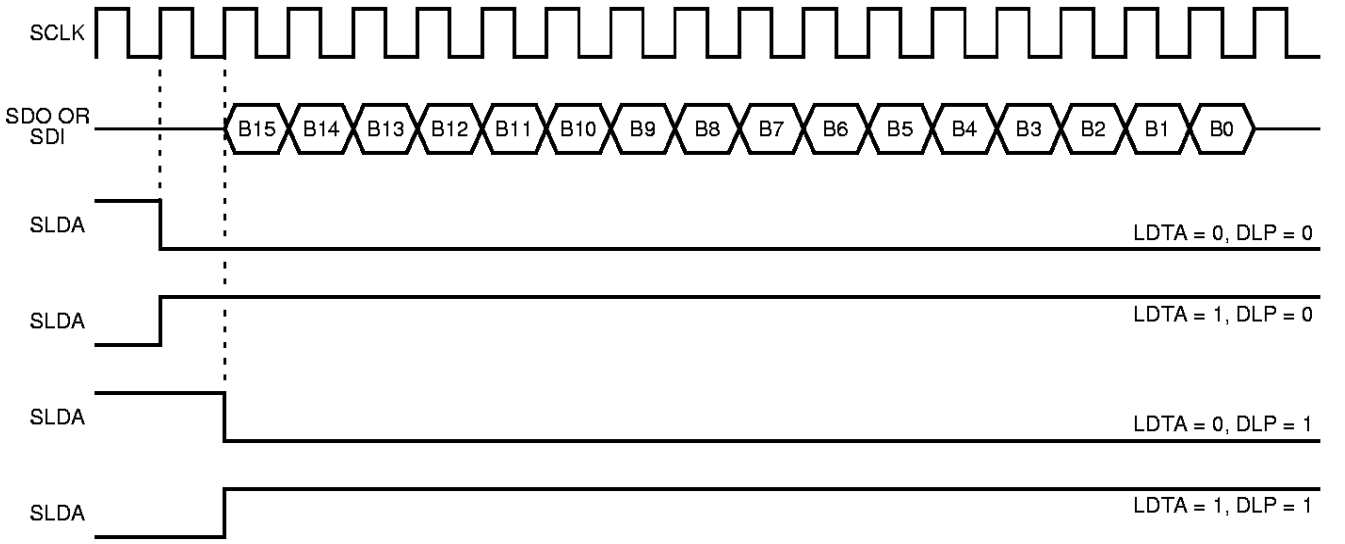


Figure 11. Serial Transfer in Active, Single-Channel Mode

By default, the SIO generates a load pulse that occurs one SCLK cycle before data is shifted in and out of the SIO; however, the load pulses can be configured to occur at the same time as when the serial transmission occurs, by setting the DLP bit. When the load pulse occurs, data to be shifted out appears at the SDO pin at the next rising edge of SCLK. Data to be shifted in should be applied to the SDI pin on the rising edge of SCLK and will be latched on the falling edge of SCLK.

When the serial transfer begins, an Output Buffer Empty (OBE) interrupt is generated. Similarly, at the end of a serial transfer, an Input Buffer Full (IBF) interrupt is generated. The user should generate code that services these interrupts before the next serial transmission occurs in order to ensure that a proper data is transmitted. If new serial data is not written to the **sdx** register when OBE is set, the previous contents of this register are shifted out. Similarly when IBF is set, if the **sdx** register is not read before the end of the next serial transfer, whatever data was applied to the SDI pin will replace the current contents of the **sdx** register on the next IBF interrupt.

The SIO generates a load pulse that has a 50% duty cycle. Since the load pulse frequency is either SCLK/256 or SCLK/512 in active mode, back-to-back transmissions are not possible in this mode, and the next serial transmission will occur either 240 or 496 SCLK cycles later.

The operation of the SIO in passive mode is similar to that of Figure 11. The delayed load pulse option is not supported in passive mode. Back-to-back serial transfers are possible in passive mode with appropriate load pulse signals, as shown in Figure 12.

4 Hardware Architecture (continued)

4.11 Dual-Channel Serial I/O Port (SIO) for *FlashDSP*1609F (continued)

4.11.2 *FlashDSP*1609F SIO Operation (continued)

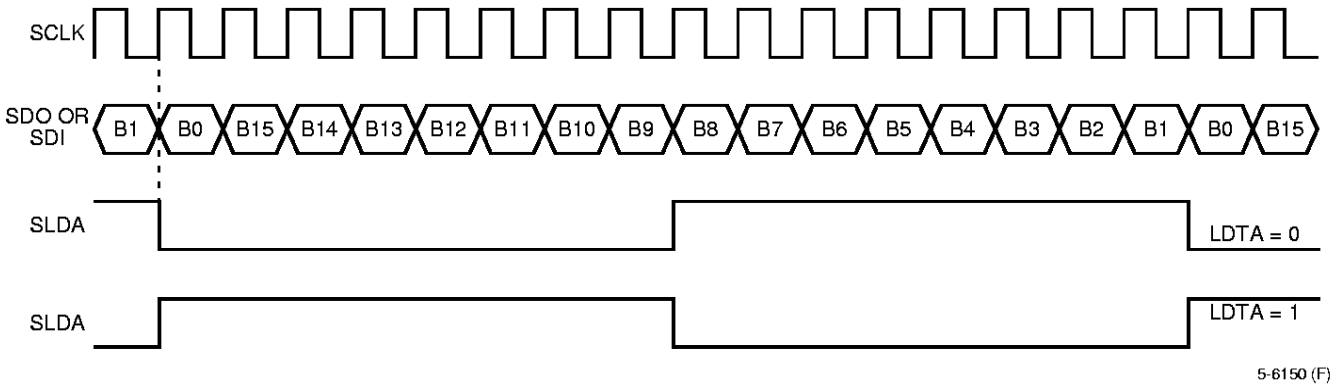


Figure 12. Back-To-Back Serial Transfer in Passive, Single-Channel Mode

Figure 13 illustrates the operation of the SIO in passive, dual-channel mode. The SCLK signal is not shown for simplicity. In dual-channel mode, SLDA is the strobe for initiating serial transfers through channel A, and SLDB is the strobe for initiating serial transfers through channel B, SIO operation is the same as described previously in single-channel mode; however, the serial transfers are now time-division multiplexed between channels A and B.

In addition to the interrupts IBF and OBE, there are two interrupts, IBFB and OBEB, to indicate when serial data from channel B needs to be serviced.

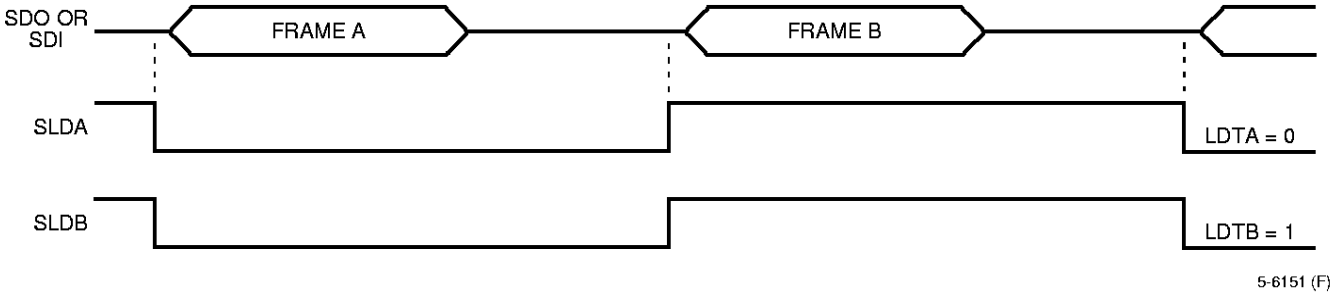


Figure 13. Serial Transfer in Passive, Dual-Channel Mode

Figure 14 shows the operation of the SIO in active, dual-channel mode with delayed load. SLDA and SLDB are in phase with each other and need to be externally inverted. Transfers begin with channel A, and SLDA starts out in the low phase.

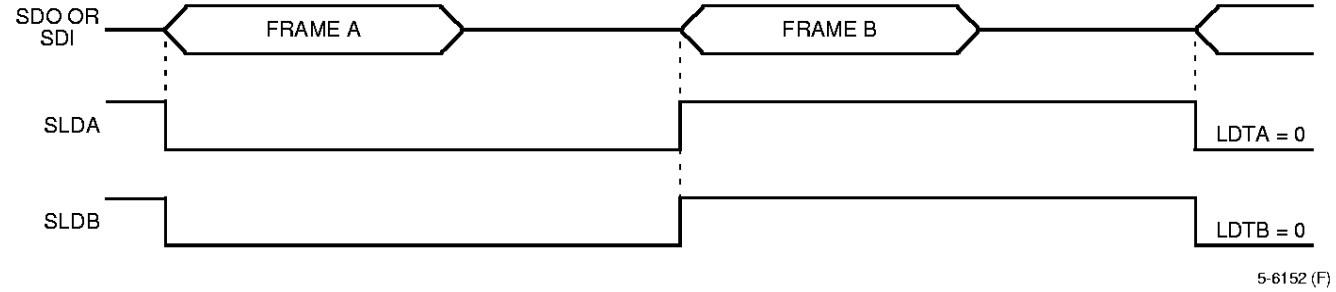


Figure 14. Serial Transfer in Active, Dual-Channel Mode with Delayed Load

4 Hardware Architecture (continued)

4.12 Dual-Channel Serial I/O Port (SIO) for DSP1609

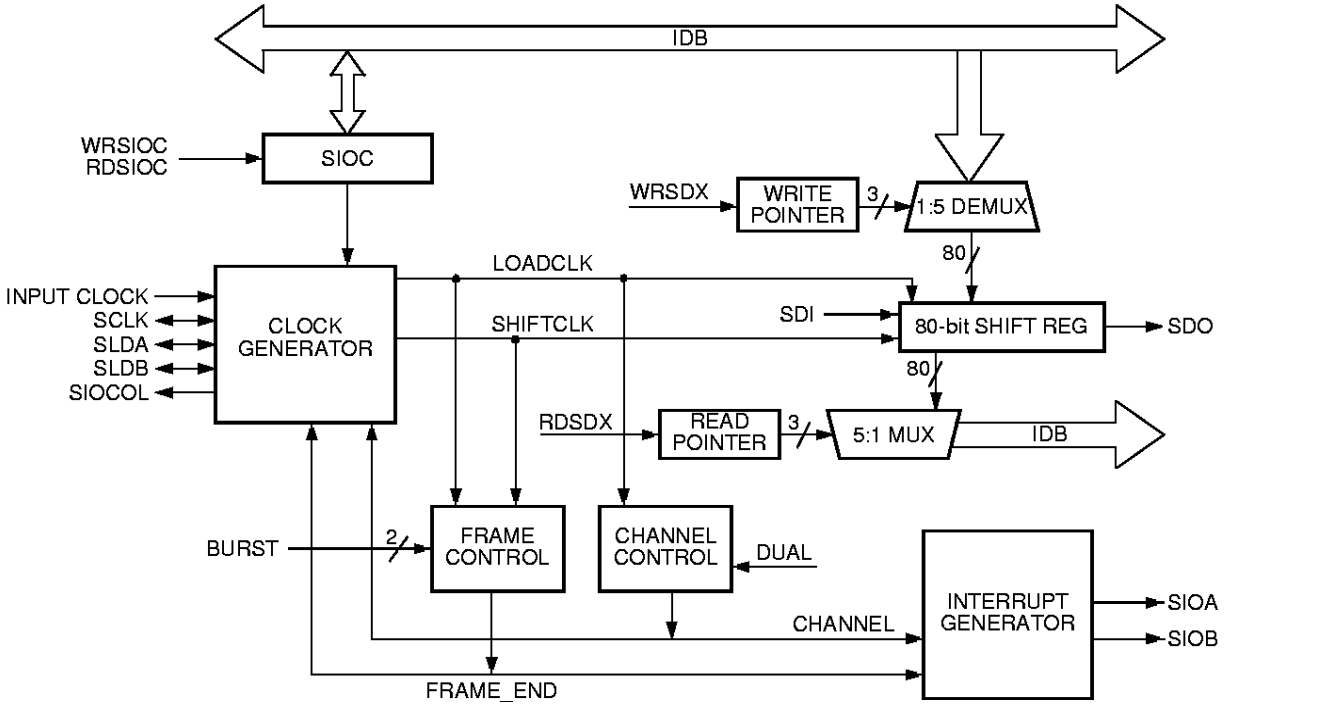
The SIO interfaces to the external world through five pins: Serial Data Input (SDI), Serial Data Output (SDO), Serial Clock (SCLK), Serial Load A (SLDA), and Serial Load B (SLDB). The SIO pins are multiplexed with IOPB pins. When the SIO is enabled, the IOPB[7:4] pins are configured as SIO pins, i.e., SDI, SDO, SCLK, and SLDA; otherwise, these pins function normally as IOP pins. When the SIO is enabled in dual-channel mode (see Table 47, SIO Control Register (SIOC) Fields, on page 63), the IOPB2/SLDB pin is configured for SLDB operation; otherwise, this pin normally functions as the IOPB2 pin.

Serial data enters the SIO through the SDI pin and exits through the SDO pin. SCLK is the clock that controls shifting data in and out of the SIO. Data is shifted in and latched on the falling edge of SCLK, while data is shifted out on the rising edge of SCLK. SLDA and SLDB provide the strobes necessary for initiating serial transfers. For single-channel mode, only the SLDA pin is necessary; however, for dual-channel operation, SLDA and SLDB initiate transfers from channels A and B, respectively.

The SIO can be configured to support many serial protocols; however, all serial transfers are restricted to most significant bit (MSB) format. That is, the most significant bit in the data is always transmitted first.

4.12.1 DSP1609 SIO Architecture

Figure 15 shows the block diagram of the DSP1609 SIO.



5-6153 (F)

Figure 15. DSP1609 SIO Block Diagram

The SIO data register (**sdx**) is composed of an 80-bit shift register, multiplexing logic, and read/write pointers. The shift register is divided into five individually addressable 16-bit registers, **sdx**[0:4]. The write pointer selects one of these 16-bit registers to be loaded with serial output data. The DSP1600 core transfers data to the SIO through the internal data bus (IDB). Data is loaded starting with the last register (**sdx**[4]) and ending with the first register (**sdx**[0]).

4 Hardware Architecture (continued)

4.12 Dual-Channel Serial I/O Port (SIO) for DSP1609 (continued)

4.12.2 DSP1609 SIO Operation

Figure 16 illustrates the operation of the SIO for one setting in both cases.

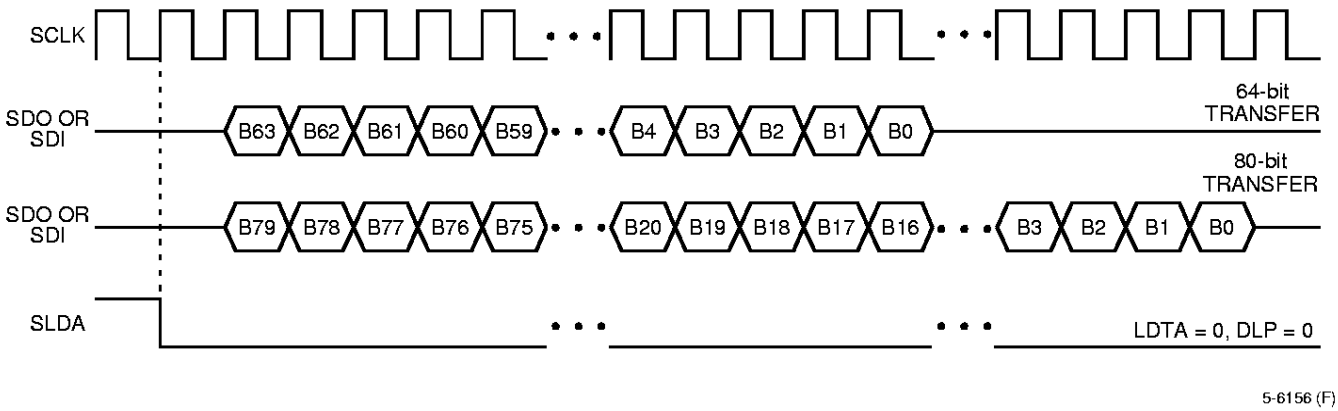


Figure 16. 64-Bit and 80-Bit Serial Transfer in Active, Single-Channel Mode

At the end of a serial transfer, a SIOA interrupt is generated to the DSP1600 core. The user should generate code that services this interrupt before the next serial transmission occurs in order to ensure that proper data is transmitted. Since the SIO does not buffer the serial data, the user needs to read the serial input data before writing new output data for 64-bit and 80-bit transfers; otherwise, the output data will clobber the input data in the shift register. For 16-bit transfers, it does not matter whether a write is performed before a read, since data transfers occur from opposite ends of the shift register.

The read/write pointers point to the 16-bit segment in the shift register from/to which serial data is transferred. Table 16 shows the reset state of the pointers and the addressing sequence for multiple reads/writes. At the beginning of every serial frame, the SIO resets the pointers to their default address. The pointers are designed to wrap around at the end of every sequence. The pointers continue to operate even when the SIO is disabled, allowing the sdx register to be initialized with data.

Table 16. SIO Read/Write Pointer Operation

Pointer	Reset Address	Addressing Sequence
Write	sdx4	sdx4, sdx3, sdx2, sdx1, sdx0, sdx4, . . .
Read (16-bit)	sdx0	sdx0, sdx1, sdx2, sdx3, sdx4, sdx0, . . .
(64-bit)	sdx3	sdx3, sdx2, sdx1, sdx0, sdx3, . . .
(80-bit)	sdx4	sdx4, sdx3, sdx2, sdx1, sdx0, sdx4, . . .

Keep in mind that the pointers retain their state even when the SIO is halted. If the sdx register is written just before the SIO is halted, serial output data may be written to an unintended 16-bit segment when the sdx register is re-initialized and the SIO restarted.

For dual-channel operation, serial transfer begins with channel A when the SIO is first enabled and SLDA starts out in the high phase. The SIOA interrupt indicates that serial input data was received for channel A and that the user should load serial output data to be sent out in the next frame for channel B. The SIOB interrupt indicates that data was received for channel B and the user must load the data for channel A.

4 Hardware Architecture (continued)

4.12 Dual-Channel Serial I/O Port (SIO) for DSP1609 (continued)

4.12.3 DSP1609 SIO Programming Examples

The following programming segment configures the SIO in passive, single-channel mode. Serial input data is stored in DPRAM at memory location serial_in, and serial output data is stored at memory location serial_out.

```
sioa: *r2=sdx         /* store SIO input data in location serial_in */
      sdx=*r3         /* transfer serial output data to SIO         */
      ireturn
start:auc=0
      . . .
      r2=serial_in    /* set pointer to serial input location       */
      r3=serial_out   /* set pointer to serial output location      */
      inc=0x0001      /* enable SIOA interrupt                      */
      *r3=a0          /* transfer serial output data to memory      */
      sioc=0x8000     /* enable SIO in passive, single-channel mode */
      . . .
/* other code */
      . . .
serial_in: int
serial_out: int
```

The following programming segment illustrates the application of the SIO in single-channel mode with 64-bit serial transfers. For 64-bit and 80-bit transfers, the serial input data must be read before new serial output data is written.

```
sioa: r2=serial_in    /* transfer serial input data to memory       */
      *r2++=sdx
      *r2++=sdx
      *r2++=sdx
      *r2++=sdx
      r3=serial_out   /* transfer serial output data to SIO         */
      sdx=*r3++
      sdx=*r3++
      sdx=*r3++
      sdx=*r3++
      ireturn
start:auc=0
      . . .
      r2=serial_in    /* set pointer to serial input location       */
      r3=serial_out   /* set pointer to serial output location      */
      inc=0x0001      /* enable SIOA interrupt                      */
      sioc=0x8000     /* enable SIO in passive, single-channel mode */
      . . .
/* other code */
      . . .
serial_in: 4* int
serial_out: 4* int
```

4 Hardware Architecture (continued)

4.13 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 45, psw (Processor Status Word) Register, on page 61, and Table 32, alf (Standby and Memory Map) Register, on page 53. The **ar<0—3>** registers can also be used as general-purpose registers.

The BMU instruction are detailed in Section 5.1.6, F4 BMU Instructions. For a thorough description of the BMU, see the *DSP1611/17/18/27 Digital Signal Processor* Information Manual.

5 Software Architecture

5.1 Instruction Set

The DSP1609 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 DSP1609 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.

The operators in Table 17 are used to describe the instructions in Sections 5.1.1 through 5.1.8.

Table 17. Instruction Set Operators

Symbol	Meaning
•	Denotes any of the following: 16-by-16 multiplication for a 32-bit product. Register-indirect addressing when used as a prefix to an address register. Direct addressing when used as a prefix to an immediate.
+	36-bit addition[1].
–	36-bit subtraction[1].
>>	Arithmetic right shift.
<<	Arithmetic left shift.
>>>	Logical right shift.
<<<	Logical left shift.
|	36-bit bitwise OR[1].
&	36-bit bitwise AND[1].
^	36-bit bitwise EXCLUSIVE OR[1].
:	Compound address swapping, accumulator shuffling.
~	One's complement.

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

5.1.1 F1 Multiply/ALU Instructions

Note that the function statements and transfer statements in Table 18 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.

5 Software Architecture (continued)

5.1 Instruction Set (continued)

5.1.1 F1 Multiply/ALU Instructions (continued)

Table 18. F1 Multiply/ALU Instructions

F1 Function Statement	Transfer Statement[1]	Transfer Statement Cycles[2]	
		Not Using Cache	Using Cache
p = x • y[3]	y = Y, x = X	2	1
aD = p, p = x • y[3]	y = aT, x = X	2	1
aD = aS + p, p = x • y[3]	y[I] = Y	1	1
aD = aS – p, p = x • y[3]	aT[I] = Y	1	1
aD = p	x = Y	1	1
aD = aS + p	Y	1	1
aD = aS – p	Y = y[I]	2	2
aD = y	Y = aT[I]	2	2
aD = aS + y	Z:y, x = X	2	2
aD = aS – y	Z:y[I]	2	2
aD = aS & y	Z:aT[I]	2	2
aD = aS | y	—	1	1
aD = aS ^ y	—	1	1
aS – y	—	1	1
aS & y	—	1	1

1. The [I] is an optional argument that specifies the low 16 bits of aT or **y**.
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.
3. **p = x • y** becomes a single-cycle squaring operation if the **auc** bit 7 is set. With bit 7 set, a transfer statement of the form **y = Y** loads the **x** register and the **y** register with the same number, so **p = x • y** results in the square.

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.

Table 19. Replacement Table for F1 Multiply/ALU Instructions

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

5 Software Architecture (continued)

5.1 Instruction Set (continued)

5.1.2 F2 Special Function Instructions

All forms of the special function instructions require one word of program memory and execute in one instruction cycle.

The special functions in Table 20 can be executed conditionally, 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 preceding special function instructions can be executed unconditionally by writing them directly. For example, **a0 = a1**.

Table 20. F2 Special Function Instructions

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

Table 21. Replacement Table for F2 Special Function Instructions

Replace	Value	Meaning
aD, aS	a0, a1	One of the two DAU accumulators.
CON	mi, pl, eq, ne, gt, le, lvs, lvc, mvs, mvc, c0ge, c0lt, c1ge, c1lt, heads, tails, true, false, npint, njint, pllon, plloff, slowon, slowoff, stopclk, rfrsh, oddp, evenp, mns1, nmns1	See Table 24 for definitions.

5 Software Architecture (continued)

5.1 Instruction Set (continued)

5.1.3 Control Instructions

Table 22 shows control instructions and their required numbers of instruction cycles and program-memory words. Required instruction cycles and program-memory words vary according to whether each instruction is executed unconditionally or conditionally. Control instructions cannot be executed from the cache.

Table 22. Control Instructions

Control Instructions	Executed Unconditionally		Executed Conditionally	
	Number of Cycles	Number of Words	Number of Cycles	Number of Words
goto JA[1] goto pt call JA[1] call pt return (goto pr)	2	1	3	2
ireturn (goto pi)[2]	2	1	—	—

1. 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 increments to the next page and the jump is to the next page rather than the desired current page.
2. The **ireturn** instruction can only be executed unconditionally.

With the exception of **ireturn**, the control instructions in Table 22 can be executed conditionally. For example:

if le goto 0x0345

Table 23. Replacement Table for Control Instructions

Replace	Value	Meaning
CON	mi, pl, eq, ne, gt, le, lvs, lvc, mvs, mvc, c0ge, c0lt, c1ge, c1lt, heads, tails, true, false, npint, njint, plloff, pllon, slowoff, slowon, stopclk, rfrsh, oddp, evenp, mns1, nmns1	See Table 24 for definitions of mnemonics.
JA	12-bit value	Least significant 12 bits of absolute address within the same 4 Kword memory section.

5 Software Architecture (continued)

5.1 Instruction Set (continued)

5.1.4 Conditional Mnemonics (Flags)

Please note the following:

- Testing the state of the counter (**c0** or **c1**) automatically increments the counter by one.
- The pseudorandom sequence generator (PSG) may be reset by writing any value to the **pi** register, except during an interrupt service routine. While in an interrupt service routine, writing to the **pi** register updates the register and does not reset the PSG. If not in an interrupt service routine, writing to the **pi** register resets the PSG. (The **pi** register is updated, but written with the contents of the PC on the next instruction.) Interrupts must be disabled when writing to the **pi** register. If an interrupt is taken after the **pi** write, but before **pi** is updated with the PC value, the **ireturn** instruction does not return to the correct location. However, if the RAND bit in the **auc** register is set, writing the **pi** register never resets the PSG. A random rounding function can be implemented with either heads or tails.

Table 24. DSP1609 Conditional Mnemonics

Test	Meaning	Test	Meaning
pl	Result is nonnegative (sign bit is bit 35).	mi	Result is negative.
eq	Result is equal to 0.	ne	Result is not equal to 0.
gt	Result is greater than 0.	le	Result is less than or equal to 0.
lvs	Logical overflow set.[1]	lvc	Logical overflow clear.
mvs	Mathematical overflow set.[2]	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.
npint	Not PINT (used by JTAG).	njint	Not JINT (used by JTAG).
pllon	DSP core clock is currently phase-locked loop (CLKPLL).	plloff	DSP core clock is not currently the phase-locked loop (CLKPLL).
slowon	DSP core clock is currently the low-frequency clock (CLKLOW).	slowoff	DSP core clock is not currently the low-frequency clock (CLKLOW).
stopclk	STOPCLK is set but the 2X core clock (CLKCORE2X) has not yet stopped.	rfrsh	IOPD<0—2> is controlled by DRC.
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).

1. Result is not representable in the 36-bit accumulators (36-bit overflow).
2. Bits 35:31 are not the same (32-bit overflow).

5 Software Architecture (continued)

5.1 Instruction Set (continued)

5.1.5 F3 ALU Instructions

These instructions, shown in Table 25, are implemented in the DSP1600 core. F3 ALU instructions allow accumulator two-operand operations with either another accumulator, the **p** register, or a 16-bit immediate operand. The result is placed in a destination accumulator that can be independently specified. All operations are done with the full 36 bits. For the accumulator with accumulator operations, both inputs are 36 bits. For the accumulator 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.

Table 25. F3 ALU Instructions

Cachable (1 cycle)	Not Cachable (2 cycle)
aD = aS + aT	aD = aSh + IM16
aD = aS – aT	aD = aSh – IM16
aD = aS & aT	aD = aSh & IM16
aD = aS | aT	aD = aSh | IM16
aD = aS ^ aT	aD = aSh ^ IM16
aS – aT	aSh – IM16
aS & aT	aSh & IM16
aD = aS + p	aD = aSl + IM16
aD = aS – p	aD = aSl – IM16
aD = aS & p	aD = aSl & IM16
aD = aS | p	aD = aSl | IM16
aD = aS ^ p	aD = aSl ^ IM16
aS – p	aSl – IM16
aS & p	aSl & IM16

Table 26. Replacement Table for F3 ALU Instructions

Replace	Value	Meaning
aD, aT, aS	a0 or a1	One of the two accumulators.
IM16	16-bit value	Long immediate data: sign-, zero-, or one-extended as appropriate.
aSh	a0h or a1h	Upper half of the accumulator.
aSl	a0l or a1l	Lower half of the accumulator.

5.1.6 F4 BMU Instructions

The bit manipulation unit in the DSP1609/*FlashDSP* 1609F provides a set of efficient bit manipulation operations on accumulators. It contains four auxilliary registers, **ar<0—3>** (**arM**, **M** = 0, 1, 2, 3), two alternate accumulators (**aa0—aa1**), which can be shuffled with the working set, and four flags (oddp, evenp, mns1, and nms1). 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. All BMU instructions require 1 word of program memory unless otherwise noted. Please refer to the *DSP1611/17/18/27 Digital Signal Processor* Information Manual for further discussion of the BMU instructions.

5 Software Architecture (continued)

5.1 Instruction Set (continued)

5.1.6 F4 BMU Instructions (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 = $\overline{aS}$ >> aS	Arithmetic right shift by aS (36-bit, sign filled in); 2-cycle.
aD = aS >>> IM16	Logical right shift by immediate (32-bit, 0s filled in); 2-cycle, 2-word.
aD = aS >>> arM	Logical right shift by **arM** (32-bit, 0s filled in); 1-cycle.
aD = $\overline{aS}$ >>> aS	Logical right shift by aS (32-bit, 0s filled in); 2-cycle.
aD = aS << IM16	Arithmetic left shift[1] by immediate (36-bit, 0s filled in); 2-cycle, 2-word.
aD = aS << arM	Arithmetic left shift[1] by **arM** (36-bit, 0s filled in); 1-cycle.
aD = $\overline{aS}$ << aS	Arithmetic left shift[1] by aS (36-bit, 0s filled in); 2-cycle.
aD = aS <<< IM16	Logical left shift by immediate (36-bit, 0s filled in); 2-cycle, 2-word.
aD = aS <<< arM	Logical left shift by **arM** (36-bit, 0s filled in); 1-cycle.
aD = $\overline{aS}$ <<< aS	Logical left shift by aS (36-bit, 0s filled in); 2-cycle.

1. 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	Shufle 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 27. Replacement Table for 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 auxilliary BMU registers.

5 Software Architecture (continued)

5.1 Instruction Set (continued)

5.1.7 Cache Instructions

Each cache instruction requires one program-memory word. Table 28 shows cache instructions and their required numbers of instruction cycles. Control instructions and long immediate values cannot be stored inside the cache.

Table 28. Cache Instructions

Cache Instructions	Number of Cycles
do	1
redo	2

Cache instruction formats are as follows:

```
do K {
INSTR_1
INSTR_2
.
.
.
INSTR_NI
}

redo K
```

Table 29. Replacement Table for Cache Instructions

Replace	Instruction Encoding	Meaning
K	cloop[1]	Number of times the instructions are to be executed taken from bits 6:0 of the **cloop** register.
	1 to 127	Number of times the instructions are to be executed is encoded in the instruction.
NI	1 to 15	1 to 15 instructions can be included.

1. 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:

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

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), 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**.

5 Software Architecture (continued)

5.1 Instruction Set (continued)

5.1.8 Data Move Instructions

Table 30 shows data move instructions and their required numbers of program-memory words and instruction cycles. 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.

Table 30. Data Move Instructions

Data Move Instructions	Number of Words	Number of Cycles
R = IM16	2	2
SR = IM9	1	1
aT[I] = R R = aS[I] Y = R R = Y Z:R DR = *(OFFSET) *(OFFSET) = DR	1	2

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.

Table 31. Replacement Table for Data Move Instructions

Replace	Value	Meaning
R	Any of the registers in Table 78 on page 72[1]	—
DR	**r<0—3>**, **a0**[I], **a1**[I], **y**[I], **p**[I], **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.
OFFSET	5-bit value from instruction 11-bit value from 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 data.

1. Some registers are write only or read only.

5 Software Architecture (continued)

5.2 Register Settings

The following tables, listed alphabetically, describe the programmable registers of the DSP1609.

Note: Some tables in this section use the following abbreviations:
X = don't care
W = write only

Table 32. alf (Standby and Memory Map) Register

Bit	15	14	13—8	7	6	5	4	3—0
Field	AWAIT	LOWPR	Res	NMNS1	MNS	EVENP	ODDP	Res

Bit	Field	Description
15	AWAIT	0 = normal DSP operation. 1 = DSP enters standby, powerdown mode.
14	LOWPR	Memory map selection: 0 = select memory MAP1. 1 = select memory MAP3.
13—8	Res	Reserved—read as zero, write as zero.
7	NMNS1	0 = BMU result is minus one in 2's complement form. 1 = BMU result is not minus one in 2's complement form.
6	MNS1	0 = BMU result is not minus one in 2's complement form. 1 = BMU result is minus one in 2's complement form.
5	EVENP	0 = BMU result has odd parity. 1 = BMU result has even parity.
4	ODDP	0 = BMU result has even parity. 1 = BMU result has odd parity.
3—0	Res	Reserved—read as zero, write as zero.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 33. auc (Arithmetic Unit Control) Register

Bit	15—9	8	7	6—4	3—2	1—0
Field	Res	RAND	X = Y =	CLR	SAT	ALIGN

Bit	Field	Value	Description
15—9	Res	—	Reserved—read as zero, write as zero.
8	RAND	0	Pseudorandom number generator (PNG) reset by writing the **pi** register only outside an interrupt service routine.
		1	PNG never reset by writing the **pi** register.
7	X = Y =	0	Normal operation.
		1	Transfer statements **y = Y** load both the **x** and the **y** registers. All instructions that load the high half of the **y** register also load the **x** register. This allows single-cycle squaring (**p** = **x** • **y**).
6—4	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).
3—2	SAT	1X	**a1** saturation on overflow is disabled (enabled when 0).
		X1	**a0** saturation on overflow is disabled (enabled when 0).
1—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	**a0**, **a1** ← **p** x 2 (and zeros written to the LSB).

Table 34. cbit<a—d> (IOP Control Bit) and sbit<a—d> (IOP Status Bit) Registers

cbit<a—d> Registers

Bit	15—8	7—0
Field	MODE[7:0]	DATA[7:0]

sbit<a—d> Registers

Bit	15—8	7—0
Field	DIR[7:0]	VALUE[7:0]*

cbit<a—d> and sbit<a—d> Register Fields

DIR[n]†	MODE[n]†	DATA[n]†	Action on IOP[n]†
1 (Output)	0	0	Clear
1 (Output)	0	1	Set
1 (Output)	1	0	No change
1 (Output)	1	1	Toggle
0 (Input)	x‡	x‡	Input

* Read only. Any value written to this field is ignored.
† $0 \le n \le 7$. For IOPD $0 \le n \le 3$.
‡ See Table 15, IOP Pin Multiplexing, on page 34 and Section 4.8.2, IOPA Interrupt Circuitry, on page 33.

Note: Because the **cbit** and **sbit** registers have interrelated fields, Table 34 duplicates the information in Table 46.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 35. chipc Register Fields

Bit	15—6	5	4—1	0
Field	Reserved	WIOPUR	Reserved	WDRST

Bit	Field	Description
15—6	Reserved	Reserved; write with zeros, read as zeros.
5	WIOPUR	Enable write of IOP pull-up resistor control registers: 0 = write **cbit<a—d>** registers as normal. 1 = write **IOPUC<A—D>** registers in place of **cbit<a—d>**.
4—1	Reserved	Reserved; write with zeros, read as zeros.
0	WDRST	Watchdog reset status: 0 = no watchdog reset has occurred. 1 = watchdog reset has occurred.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 36. chipo Register Fields

Bit	15—12	11	10—9	8—7	6	5—4	3	2—0
Field	Reserved	LOCK	Reserved	ROSP	LOCKOSCON	WDEN	WDCLKSEL	Reserved

Bit	Field	Description
15—12	Reserved	Write with zeros; read as zeros.
11	LOCK[1]	Lock all bits in the chipo register: 0 = chipo register contents may be changed. 1 = chipo register contents may not be modified until after next reset sequence.
10—9	Reserved	Write with zeros; read as zeros.
8—7	ROSP[1:0]	Ring oscillator speed: 00 = minimum of 32 kHz. 01 = increase default speed approximately 36%. 10 = decrease default speed approximately 28%. 11 = decrease default speed approximately 50%.
6	LOCKOSCON	Lock oscillator ON, crystal oscillator override: 0 = allows the OSCDIS bit (**clkc**[1]) to control oscillator. 1 = forces the oscillator to produce a clock; takes precedence over the OSCDIS bit (**clkc**[1]).
5—4	WDEN[1:0]	Watchdog timer enable bits: 00 = watchdog timer disabled. 01 = time-out after 2^{12} watchdog clock cycles. 10 = time-out after 2^{14} watchdog clock cycles. 11 = time-out after 2^{16} watchdog clock cycles.
3	WDCLKSEL	Watchdog clock select: 0 = select (InputClock/128) as watchdog clock source. 1 = select internal ring oscillator as watchdog clock source; also prevents ROSCDIS (**clkc**[2]) from disabling ring oscillator.
2—0	Reserved	Write with zeros; read as zeros.

1. The **chipo** register should be initialized once immediately after reset and can be locked to prevent any further changes to its contents by setting the LOCK bit. Once LOCK bit has been set, the **chipo** register cannot be modified until a reset sequence occurs.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 37. clkc Register Fields

Bit	15—13	12	11—8	7—6	5—4	3	2	1	0
Field	Res	PFIVOLT	DOUT MUX	SLOMUX	SELCLK	STOPCLK	ROSCDIS	OSCDIS	PLLEN

Bit	Field	Description
15—13	Reserved	Write with 0, read as 0.
12	PFIVOLT	Voltage select: 0 = 3 V. 1 = 5 V.
11—8	DOUT MUX[3:0]	DOUT source selection: 0000 = logic 0 selected to DOUT. 0001 = input crystal/clock (CLKIN) selected to DOUT. 0010 = low-frequency clock (CLKLOW) selected to DOUT. 0011 = wait-stated DSP clock (CLKWAIT) selected to DOUT. 0100 = free-running DSP clock (CLKFREE) selected to DOUT. 0101 = slow, smooth MUX output selected to DOUT (CLKCORE2X). 0110 = logic 0 selected to DOUT. 0111 = logic 1 selected to DOUT.
7—6	SLOMUX[1:0]	Low-frequency clock source selection (CLKLOW): 00 = chip input clock (CLKIN) divided by 4. 01 = chip input clock (CLKIN) divided by 32. 10 = chip input clock (CLKIN) divided by 128. 11 = internal ring oscillator (CLKRING).
5—4	SELCLK[1:0][1]	Select the 2X core clock (CLKCORE2X) (as long as STOPCLK = 0): 00 = select input clock (CLKIN). X1 = select PLL output (CLKPLL). 10 = select low-frequency clock (CLKLOW) as selected by SLOMUX[1:0].
3	STOPCLK	Stop DSP core clock (CLKCORE2X): 0 = clock runs as normal. 1 = disable 2X DSP input core clock.
2	ROSCDIS	Ring oscillator disable: 0 = enable the internal ring oscillator clock (CLKRING). 1 = disable the internal ring oscillator clock (CLKRING).
1	OSCDIS[2]	Oscillator disable (this bit should only be set if the oscillator is not bypassed, i.e., OSCBYP pin = 0): 0 = enable the crystal oscillator clock. 1 = disable the crystal oscillator clock.
0	PLLEN[3]	Phase-locked loop enable: 0 = disable PLL circuitry. 1 = enable PLL to operate.

1. It is recommended that the 2 bits of the SELCLK field be changed one at a time to prevent the possibility of an intermediate clock switch before switching to the desired destination clock. For example, to switch the clock selection back and forth between the low-frequency clock (CLKLOW) and the high-frequency PLL output (CLKPLL), the more significant bit of SELCLK (**clkc**[5]) should be set to one, allowing the less significant bit of SELCLK (**clkc**[4]) to be toggled on and off without the possibility of an intermediate clock switch.
2. This bit should not be set if the internal crystal oscillator is bypassed, i.e., OSCBYP pin = 1. Also note that when this bit is set, it will not take effect until after CLKRING is selected and is sourcing CLKCORE2X.
3. This bit is overridden if it is cleared to zero and if the PLL output (CLKPLL) is selected as the 2X core clock (if SELCLK[0] = 1).

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 38. drc Register Fields

Bit	15	14—9	8—0
Field	DRCEN	FLSEL	RFSHSEL

Bit	Field	Description
15	DRCEN	0 = disable DRC. 1 = enable DRC.
14—9	FLSEL[5:0]	Delay between setting both rfrsh and URFSH high = FLSEL[5:0] • XTAL period
8—0	RFSHSEL[8:0]	RFRSH Period = XTAL period • RFSHSEL[8:0] • 2

Table 39. inc (Interrupt Control) Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field[1]	JINT[2]	SW7	SW6	SSI	—	OBEB (SW5)	INTB	IBFB (SIOB)	IOPA	SW4	TIME0	SW3	TIME1	SW2	OBE (SW1)	IBF (SIOA)

1. DSP1609 settings are shown in parentheses.
2. JINT is a JTAG interrupt and is controlled by the HDS. It may be made unmaskable by the Lucent development system tools.

Encoding: A zero in a bit of **inc** disables an interrupt; a one enables the interrupt. For more information about the fields in Table 39, see Table 6 on page 23.

Table 40. ins (Interrupt Status) Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field[1]	JINT	SW7	SW6	SSI	—	OBEB (SW5)	INTB	IBFB (SIOB)	IOPA	SW4	TIME0	SW3	TIME1	SW2	OBE (SW1)	IBF (SIOA)

1. DSP1609 settings are shown in parentheses.

Encoding: A zero in a bit of **ins** indicates no interrupt. A one indicates an interrupt has been recognized and is pending or being serviced. If a one is written to bits 0, 3, 5, 8, 9, or 12 of **ins** (also 1 or 10 for DSP1609F), the corresponding interrupt is cleared. To clear IOPA, the **sbita** register must be read. For more information about the fields in Table 40, see Table 6 on page 23.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 41. IOPUC<a—d> Register Fields[1]

Bit	15—8	7—0
Field	IOPU[7:0]	IOPD[7:0]

Bit	Field	Description
15—8	IOPU[7:0]	IOPU[7:0] pull-up control: 0 = no pull-up resistor. 1 = pull-up IOP pin when configured as an input.
7—0	IOPD[7:0]	IOPL[7:0] pull-down control: 0 = no pull-down resistor. 1 = pull-down IOP pin when configured as an input.

1. **ipucd**[15:12] and **iopucd**[7:4] are reserved. These bits should only be written with 0.

Notes:
For all IOP ports, when both IPU*n* = 1 and IOPL*n* = 1, a strong pull-up option is enabled for the corresponding IOP.
IOPB1 and IOPB3 have no internal pull-up or pull-down capability.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 42. JTAG ID Register (32-bit)

Bit	31	30	29—28	27—19	18—12	11—0
Field	SLEWR	SECURE	BOPT	ROMCODE	0x09	0x03B

Bit	Field	Mask-Programmable Features
31	SLEWR	Slew rate control: Specifies the slew rate for the output buffers for all IOP pins. 0 = standard buffers. 1 = slow buffers.
30	SECURE	Security: 0 = unsecured. 1 = secured.
29—28	BOPT	Buffer options: Specifies what value of series resistance is placed in all IOP pins. 00 = no resistance with output buffers. 01 = 100 Ω of resistance in output buffers. 10 = 300 Ω of resistance in output buffers. 11 = 500 Ω of resistance in output buffers.
27—19	ROMCODE	The user's ROMCODE ID (in hexadecimal), as calculated by the following formula using the letter codes from Table 43: ROMCODE ID = 0x[(20 x first letter) + second letter]
18—12	0x09	Device ID.
11—0	0x03B	Manufacturer's ID (Lucent Technologies).

Table 43. JTAG ROMCODE Letter Values

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

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 44. pllc Register Fields

Bit	15—14	13—11	10—9	8—6	5—0
Field	LF	ICP	N	K	M

Bit	Field	Description
15—14	LF[1:0]	PLL Loop Filter Control. TBD settings based on desired vco frequency[1].
13—11	ICP[2:0]	PLL Charge Pump Control.
10—9	N[1:0]	PLL Input Clock Divider. 2-bit positive integer from 0 to 3.
8—6	K[2:0]	PLL Output Divider. 3-bit positive integer from 0 to 7.
5—0	M[5:0]	PLL Multiplier. 6-bit positive integer from 0 to 63.

1. $$VCO(FREQUENCY) = \frac{CLKIN(M+2)}{(N+1)}$$

Keep N = 0 when possible, VCO range: 50—150 MHz.

$$MIPSrate = \frac{CLKIN(M+2)}{2((N+1)\bullet(K+1))}$$

Table 45. psw (Processor Status Word) Register

Bit	15—12	11—10	9	8—5	4	3—0
Field	DAU Flags	Res	a1[V]	a1[35:32]	a0[V]	a0[35:32]

Bit	Field	Value	Description
15—12	DAU Flags[1]	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.
11—10	Res	—	Reserved—read as zero, write as zero.
9	a1[V]	W	Accumulator 1 (**a1**) overflow when set.
8—5	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.
4	a0[V]	W	Accumulator 0 (**a0**) overflow when set.
3—0	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.

1. The DAU flags are set by multiply/ALU (F1), conditionals (F2), ALU (F3), or BMU (F4) operations involving the accumulators.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 46. sbit<a—d> (IOP Status Bit) and cbit<a—d> (IOP Control Bit) Registers

sbit<a—d> Registers

Bit	15—8	7—0
Field	DIR[7:0]	VALUE[7:0]*

cbit<a—d> Registers

Bit	15—8	7—0
Field	MODE[7:0]	DATA[7:0]

sbit<a—d> and cbit<a—d> Register Fields

DIR[n]†	MODE[n]†	DATA[n]†	Action on IOP[n]†
1 (Output)	0	0	Clear
1 (Output)	0	1	Set
1 (Output)	1	0	No change
1 (Output)	1	1	Toggle
0 (Input)	x	x	Input

* Read-only. Any value written to this field is ignored.
† $0 \le n \le 7$, except $0 \le n \le 3$ for sbitd and cbitd.

Note: Because the **sbit** and **cbit** registers have interrelated fields, Table 46 duplicates the information in Table 34.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 47. SIO Control Register (SIOC) Fields

Bit	15	14	13	12	11	10	9	8—7	6	5—4	3—2	1—0
Field	EN	DUAL	ACTIVE	LDTA	LDTB	DLP	Res	BURST	SIOCOL	Res	SLD	SCLK

Bit	Field	Description
15	EN	Enable SIO: 0 = disable SIO. 1 = enable SIO.
14	DUAL	Dual-channel mode select: 0 = single-channel mode. 1 = dual-channel mode.
13	ACTIVE	Active mode select: 0 = SIO operates in passive mode. 1 = SIO operates in active mode.
12	LDTA	SLDA mode select: 0 = SLDA active-low. 1 = SLDA active-high.
11	LDTB	SLDB mode select: 0 = SLDB active-low. 1 = SLDB active-high.
10	DLP	Load pulse delay (active mode only): 0 = so not delay load pulse. 1 = delay load pulse by one SCLK cycle.
9	Res	Reserved—write with zero.
8—7	BURST	Burst transfer mode select (DSP1609 only): 00 = 16-bit word transfer. 01 = 64-bit word transfer. 10 = 80-bit word transfer. 11 = 80-bit word transfer.
6	SIOCOL	Shift collision error (DSP1609 only): 0 = no error. 1 = error (write or read **sdx** register during shifting).
5—4	Res	Reserved—write with zero.
3—2	SLD	SLDA and SLDB frequency in active mode: 00 = SCLK/256. 01 = SCLK/512. 1x = reserved.
1—0	SCLK	SCLK frequency in active mode: 00 = CLKIN. 01 = CLKIN/2. 1x = reserved.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 48. SSI Control Register (SSIC) Fields

Bit	15	14	13	12	11	10	9—8	7	6	5	4—3	2—0
Field	EN	MSTR	SPOL	SPHA	SDOEN	SSNEN	Res	SDONE	WCOLL	MODF	Res	SCLK

Bit	Field	Description
15	EN	Enable SSI: 0 = disable SSI. 1 = enable SSI.
14	MSTR	SSI master/slave: 0 = configure SSI as a slave. 1 = configure SSI as a master.
13	SPOL	SCK polarity: 0 = SSI clock is idle at a logic 0. 1 = SSI clock is idle at a logic 1.
12	SPHA	SCK phase: 0 = data transmitted on SCK idle level. 1 = data transmitted on SCK active level.
11	SDOEN	SSI slave data out enable: 0 = MDISDO disabled in slave mode. 1 = MDISDO enabled in slave mode.
10	SSNEN	SSN input enable: 0 = SSN input disabled in master mode. 1 = SSN input enabled in master mode.
9—8	Res	Reserved—must be written with logic 0.
7	SDONE[1]	SSI transfer done: 0 = SSI transfer not completed. 1 = SSI transfer completed.
6	WCOLL[1]	Write collision error: 0 = no error. 1 = write to SSI data register (ssid) during a transaction.
5	MODF[1]	Mode fault error: 0 = no error. 1 = SSN asserted while SSI is in master mode.
4—3	Res	Reserved—must be written with logic 0.
2—0	SCLK	SSI clock frequency in master mode: 000 = CLKFREE/2. 001 = CLKFREE/4. 010 = CLKFREE/8. 011 = CLKFREE/16. 100 = CLKFREE/32. 101 = CLKFREE/64. 110 = CLKFREE/128. 111 = CLKFREE/128.

1. These fields are read only.

5 Software Architecture (continued)

5.2 Register Settings (continued)

Table 49. SSI Data Register (SSID) Fields

Bit	15—8	7—0
Field	Sign extension of SSI data[1]	SSI data

1. For a read operation, these bits are a sign extension of the data, i.e., the same as bit 7. For a write operation, these bits are ignored.

Table 50. Timerc (Timer Control) Register (TIMER0 and TIMER1)

Bit	15	14	13	12	11—8	7	6	5	4	3—0
Field	SELTIMCK1	Res	RELOAD1	COUNTEN1	PRESCALE1	SELTIMCK0	Res	RELOAD0	COUNTEN0	PRESCALE0

Bit	Field	Description
15	SELTIMCK1	Timer1 clock source: 0 = low-frequency clock (CLKLOW). 1 = free-running DSP clock (CLKFREE).
14	Reserved	Reserved—read as zero, write as zero.
13	RELOAD1	Timer1 reload option: 0 = Timer1—count down and stop. 1 = Timer1—count down and repeat count cycle.
12	COUNTEN1	Timer1 halt: 0 = Timer1—hold current count. 1 = Timer1—resume count toward zero.
11—8	PRESCALE1	Timer1 count frequency selection: 0000 = CLKTIM1/2 0110 = CLKTIM1/128 1100 = CLKTIM1/8192 0001 = CLKTIM1/4 0111 = CLKTIM1/256 1101 = CLKTIM1/16384 0010 = CLKTIM1/8 1000 = CLKTIM1/512 1110 = CLKTIM1/32768 0011 = CLKTIM1/16 1001 = CLKTIM1/1024 1111 = CLKTIM1/65536 0100 = CLKTIM1/32 1010 = CLKTIM1/2048 0101 = CLKTIM1/64 1011 = CLKTIM1/4096
7	SELTIMCK0	Timer0 clock source: 0 = low-frequency clock (CLKLOW). 1 = free-running DSP clock (CLKFREE).
6	Reserved	Write with zero.
5	RELOAD0	Timer0 reload option: 0 = Timer0—count down and stop. 1 = Timer0—repeat count cycle.
4	COUNTEN0	Timer0 halt: 0 = Timer0—hold current count. 1 = Timer0—resume count toward zero.
3—0	PRESCALE0	Timer0 count frequency selection: 0000 = CLKTIM0/2 1000 = CLKTIM0/512 0001 = CLKTIM0/4 1001 = CLKTIM0/1024 0010 = CLKTIM0/8 1010 = CLKTIM0/2048 0011 = CLKTIM0/16 1011 = CLKTIM0/4096 0100 = CLKTIM0/32 1100 = CLKTIM0/8192 0101 = CLKTIM0/64 1101 = CLKTIM0/16384 0110 = CLKTIM0/128 1110 = CLKTIM0/32768 0111 = CLKTIM0/256 1111 = CLKTIM0/65536

5 Software Architecture (continued)

5.3 Reset States

Table 51. Register States After Reset

Register	Bits 15—0	Register	Bits 15—0	Register	Bits 15—0
a0	•••• •••• •••• ••••	drc	0000 0000 0000 0000	r1	•••• •••• •••• ••••
a0l	•••• •••• •••• ••••	flashc	0000 0000 0000 0000	r2	•••• •••• •••• ••••
a1	•••• •••• •••• ••••	IOPUCA[1]	CCCC CCCC CCCC CCCC	r3	•••• •••• •••• ••••
a1l	•••• •••• •••• ••••	IOPUCB[1]	CCCC CCCC CCCC CCCC	rb	0000 0000 0000 0000
alf	00•• •••• •••• ••••	IOPUCC[1]	CCCC CCCC CCCC CCCC	re	0000 0000 0000 0000
ar0	•••• •••• •••• ••••	IOPUCD[1]	XXXX CCCC XXXX CCCC	sbita	0000 0000 PPPP PPPP
ar1	•••• •••• •••• ••••	inc	0000 0000 0000 0000	sbitb	0000 0000 PPPP PPPP
ar2	•••• •••• •••• ••••	ins[2]	0110 0000 0101 0100	sbitc	0000 0000 PPPP PPPP
ar3	•••• •••• •••• ••••	i	•••• •••• •••• ••••	sbitd	XXXX 0000 XXXX PPPP
auc	0000 0000 0000 0000	j	•••• •••• •••• ••••	sdx	•••• •••• •••• ••••
c0	•••• •••• •••• ••••	jtag	•••• •••• •••• ••••	sioc	0000 0000 0000 0000
c1	•••• •••• •••• ••••	k	•••• •••• •••• ••••	ssic	0100 0000 0000 0000
c2	•••• •••• •••• ••••	p	•••• •••• •••• ••••	ssid	•••• •••• •••• ••••
cbita	0000 0000 •••• ••••	PC	0000 0000 0000 0000	timer0	0000 0000 0000 0000
cbitb	0000 0000 •••• ••••	pi	SSSS SSSS SSSS SSSS	timer1	0000 0000 0000 0000
cbitc	0000 0000 •••• ••••	pl	•••• •••• •••• ••••	timerc	0000 0000 0000 0000
cbitd	XXXX 0000 XXXX ••••	pllc	•••• •••• •••• ••••	wdogr	•••• •••• •••• ••••
chipc	0000 0000 0000 000C	pr	•••• •••• •••• ••••	x	•••• •••• •••• ••••
chipo	CCCC 0CCC CCCC CCCC	psw	•••• 00•• •••• ••••	y	•••• •••• •••• ••••
clkc	0000 0000 0000 0000	pt	•••• •••• •••• ••••	ybase	•••• •••• •••• ••••
cloop	•••• •••• •••• ••••	r0	•••• •••• •••• ••••	yl	•••• •••• •••• ••••

1. These registers are shown as upper-case because they are not directly program-accessible. They are accessed indirectly via write operations to the **cbit**<a—d> registers.
2. Reset state for **ins** in DSP1609 is 0110 0100 0101 0110.

Bit code: Indicates that this bit:

- • is unknown on powerup reset and unaffected by all other resets.
- 0 is set to logic zero by all types of resets.
- 1 is set to logic one by all types of resets.
- C is not affected by a pin (RSTB) reset or watchdog reset; however, powerup reset and JTAG reset clear the bit to zero.
- P reflects the value on its corresponding input pin.
- S shadows the program counter (PC).
- X may not be written and is read as zero.

5 Software Architecture (continued)

5.4 Instruction Set Formats

This section defines the hardware-level encoding of the DSP1609 instructions.

5.4.1 Multiply/ALU Instructions

Table 52. 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

Table 53. 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

Table 54. Format 2: Multiply/ALU Read/Write Group

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

Table 55. Format 2a: Multiply/ALU Read/Write Group

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

5.4.2 Special Function Instructions

Table 56. 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

Table 57. Format 3a: F3 ALU Operations

Field	T					D	S	F3				SRC2		$\overline{aT}$	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 58. 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

5 Software Architecture (continued)

5.4 Instruction Set Formats (continued)

5.4.3 Control Instructions

Table 59. 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

Table 60. 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

Table 61. Format 6: Conditional Branch Qualifier

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

Note: A branch instruction immediately follows the qualifier.

5 Software Architecture (continued)

5.4 Instruction Set Formats (continued)

5.4.4 Data Move Instructions

Table 62. 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

Table 63. Format 8: Data Move (16-Bit Immediate Operand—2 Words)

Field	T					D	R					Reserved			
	16-bit Immediate Operand (IM16)														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 64. Format 9: Short Immediate Group

Field	T					I	9-bit Short Immediate Operand (IM9)								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 65. Format 9a: Direct Addressing

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

5.4.5 Cache Instructions

Table 66. 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

5 Software Architecture (continued)

5.4 Instruction Set Formats (continued)

5.4.6 Field Descriptions

T Field: Specifies the type of instruction.

Table 67. T Field

T	Operation	ALU Type	Format
0000x	goto JA		4
00010	short IM9 j, k, rb, re		9
00011	short IM9 r0, r1, r2, r3		9
00100	Y = a1[l]	F1[1]	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[2]	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[3]	3a
11001	y = a0, x = X		1
11010	Conditional branch qualifier		6
11011	y = a1, x = X	F1	1
11100	Y = a0[l]	F1	1
11101	Z:y, x = X	F1	2
11110	Bit 5 = 0, F4 ALU (BMU)		3b
11110	Bit 5 = 1, direct addressing		9a
11111	y = Y, x = X	F1	1

1. See Table 73 on page 71.
2. See Table 74 on page 71.
3. See Table 75 on page 71.

$\overline{aT}$ Field: Specifies a transfer accumulator.

Table 68. $\overline{aT}$ Field

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

B Field: Specifies the type of branch instruction.

Table 69. B Field

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

CON Field: Specifies the condition for special functions and conditional control instructions.

Table 70. CON Field

CON	Condition	CON	Condition
00000	mi	10000	gt
00001	pl	10001	le
00010	eq	10010	pllon
00011	ne	10011	slowon
00100	lvs	10100	plloff
00101	lvc	10101	slowoff
00110	mvs	10110	stopclk
00111	mvc	10111	evenp
01000	heads	11000	mns1
01001	tails	11001	nmns1
01010	c0ge	11010	npint
01011	c0lt	11011	njint
01100	c1ge	11100	oddp
01101	c1lt	11101	rfrsh
01110	true	11110	Reserved
01111	false	11111	Reserved

D Field: Specifies a destination register.

Table 71. D Field

D	Register
0	Accumulator 0
1	Accumulator 1

5 Software Architecture (continued)

5.4 Instruction Set Formats (continued)

5.4.6 Field Descriptions (continued)

DR Field: Specifies the data register.

Table 72. 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

F1 Field: Specifies the multiply/ALU function.

Table 73. F1 Field

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	

F2 Field: Specifies the special function to be performed.

Table 74. F2 Field

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

F3 Field: Specifies the operation in an F3 ALU instruction.

Table 75. F3 Field

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}

5 Software Architecture (continued)

5.4 Instruction Set Formats (continued)

5.4.6 Field Descriptions (continued)

Table 76. 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

I Field: Specifies a register for short immediate data move instructions.

Table 77. I Field

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

JA Field: 12-bit jump address.

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

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

R Field: Specifies the register for data move instructions.

Table 78. R Field

R	Condition	R	Condition
000000	r0	100000	inc
000001	r1	100001	ins
000010	r2	100010	pllc
000011	r3	100011	clkc
000100	j	100100	cloop
000101	k	100101	Reserved
000110	rb	101101	Reserved
000111	re	100111	wdogr
001000	pt	101000	cbita, iopuca[1]
001001	pr	101001	sbita
001010	pi	101010	chipo
001011	i	101011	jtag
001100	p	101100	sdx
001101	pl	101101	drc
001110	sioc	101110	cbitd, iopucd[1]
001111	sbitd	101111	flashc
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	cbitb, iopucb[1]
010111	c2	110111	sbitb
011000	ar0	111000	cbitc, iopucc[1]
011001	ar1	111001	sbitc
011010	ar2	111010	timer1
011011	Reserved	111011	ar3
011100	ssid	111100	Reserved
011101	ssic	111101	Reserved
011110	chipc	111110	Reserved
011111	ybase	111111	alf

1. These registers are double-mapped and accessed based on the value of the WIOPUR bit in the **chipc** register.

5 Software Architecture (continued)

5.4 Instruction Set Formats (continued)

5.4.6 Field Descriptions (continued)

R/W Field: A 1 specifies a read, DR = *(OFFSET). A 0 specifies a write, *(OFFSET) = DR.

S Field: Specifies a source accumulator.

Table 79. S Field

S	Register
0	Accumulator 0
1	Accumulator 1

SI Field: Specifies when the conditional branch qualifier instruction should be interpreted as a software interrupt instruction. (Reserved for Lucent hardware development system use.)

Table 80. SI Field

SI	Operation
0	Not a software interrupt
1	Software interrupt

SRC2 Field: Specifies operands in an F3 ALU instruction.

Table 81. SRC2 Field

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

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

Table 82. X Field

X	Operation
Two-Operand Multiply/ALU	
0	*pt++
1	*pt++i
One-Operand Multiply/ALU	

Table 82. X Field(continued)

0	aTl, yl
1	aTh, yh

Y Field: Specifies the form of register-indirect addressing with postmodification.

Table 83. Y Field

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

Z Field: Specifies the form of register-indirect compound addressing with postmodification.

Table 84. Z Field

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

6 Device Requirements and Characteristics

This section describes DSP1609/*FlashDSP* 1609F device requirements and characteristics.

6.1 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.

Absolute maximum ratings are the limiting conditions that can be applied to all variations of circuit and environmental conditions without the occurrence of permanent damage.

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

Table 85. Maximum Package Rating Parameters and Values

Parameter	Min	Max	Unit
Voltage on Any Pin with Respect to Ground	V_{SS} – 0.5	V_{DD} + 0.5	V
Power Dissipation	—	0.75	W
Ambient Temperature	–40	85	°C
Storage Temperature	–65	150	°C

6.2 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.

Lucent employs a human-body model for ESD-susceptibility testing. Because 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 V are the most common and are the values used in the Lucent human-body model test circuit. The breakdown voltage for the DSP1609 is greater than 2000 V.

6 Device Requirements and Characteristics (continued)

6.3 Recommended Operating Conditions

Table 86. Recommended Voltage and Temperature

Minimum Instruction Cycle Time (T_{MIN})	Temperature Class	Supply Voltage V_{DD} (V)		Ambient Temperature T_A (°C)	
		Min	Max	Min	Max
12.5 ns	Commercial	4.5	5.5	0	70
12.5 ns	Industrial	4.5	5.5	–40	85
12.5 ns	Industrial	3.0	3.6	–40	85

6.4 Decoupling Requirements

Install a high-quality ceramic 0.01 pF capacitor between each V_{DD} pin and ground. Install each capacitor as close as possible to the DSP1609 package. Also, install an additional 0.47 µF—1.0 µF capacitor at only one of the V_{DD} pins.

6.5 Package Thermal Considerations

The recommended operating temperature specified in Table 86 is based on the maximum power, package type, and maximum junction temperature. The following equations describe the relationship between these parameters. If the application's maximum power is less than the worst-case value, this relationship determines a higher maximum ambient temperature or the maximum temperature measured at top dead center of the package.

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

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

where T_A is the still-air ambient temperature and T_{TDC} is the temperature measured by a thermocouple at the top dead center of the package.

Maximum Junction Temperature (T_J) in 28-pin SOJ . . . . . 125 °C

28-pin SOJ Maximum Thermal Resistance in Still-Air-Ambient (Θ_{JA}) . . . . . 70 °C/W

Maximum Junction Temperature (T_J) in 44-pin PLCC . . . . . 125 °C

44-pin PLCC Maximum Thermal Resistance in Still-Air-Ambient (Θ_{JA}) . . . . . 49 °C/W

44-pin PLCC Maximum Thermal Resistance, Junction to Top Dead Center (Θ_{J-TDC}) . . . . . 11 °C/W

WARNING: Due to package thermal constraints, proper precautions in the user's application should be taken to avoid exceeding the maximum junction temperature of 125 °C. Otherwise, the device will be affected adversely.

7 Electrical Requirements and Characteristics

The following electrical requirements and characteristics are preliminary and subject to change. Electrical requirements refer to conditions imposed on the user for proper operation of the device. Electrical characteristics refer to the behavior of the device under conditions specified in Section 6, Device Requirements and Characteristics. Tables 87, 88, and 89 describe the valid electrical parameters of these conditions.

Table 87. Electrical Requirements

Parameter	Symbol	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		Min	Max	Min	Max	Min	Max	
Input Voltage (except clocks, INTB, RSTB):								
Low	V_{IL}	—	0.8	—	—	—	0.8	V
High	V_{IH}	2.0	—	—	—	2.0	—	V
Input Voltage (INTB):								
Low	V_{IL}	—	0.6	—	—	—	0.6	V
High	V_{IH}	V_{DD} – 0.5	—	—	—	V_{DD} – 0.3	—	V
Input Voltage (RSTB) and XTALA[1]:								
Low	V_{IL}	—	0.5	—	—	—	0.5	V
High	V_{IH}	V_{DD} – 0.5	—	—	—	V_{DD} – 0.3	—	V

1. Internal crystal oscillator bypass mode only.

Note: For information about input buffer power dissipation, see Section 7.2.

Table 88. Electrical Characteristics

Parameter	Symbol	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		Min	Max	Min	Max	Min	Max	
Output Low Voltage:								
Low (I_{OL} = 2.0 mA)	V_{OL}	—	0.4	—	—	—	0.4	V
Low (I_{OL} = 50 µA)	V_{OL}	—	0.2	—	—	—	0.2	V
Output High Voltage:								
High (I_{OH} = –2.0 mA)	V_{OH}	V_{DD} – 0.7	—	—	—	V_{DD} – 0.7	—	V
High (I_{OH} = –50 µA)	V_{OH}	V_{DD} – 0.2	—	—	—	V_{DD} – 0.2	—	V
Output 3-state Current, V_{DD} = V_{DD} (max):								
Low (V_{IL} = 0 V)	I_{OZL}	–10	—	—	—	–10	—	µA
High [V_{IH} = V_{DD} (max)]	I_{OZH}	—	10	—	—	—	10	µA
Schmitt Trigger Hysteresis:								
INTB	V_{IHL}	300	600	—	—	300	600	mV
RSTB	V_{IHL}	1.8	2.2	—	—	1.8	2.2	V
Input Capacitance	C_I	—	10	—	—	—	10	pF

7 Electrical Requirements and Characteristics (continued)

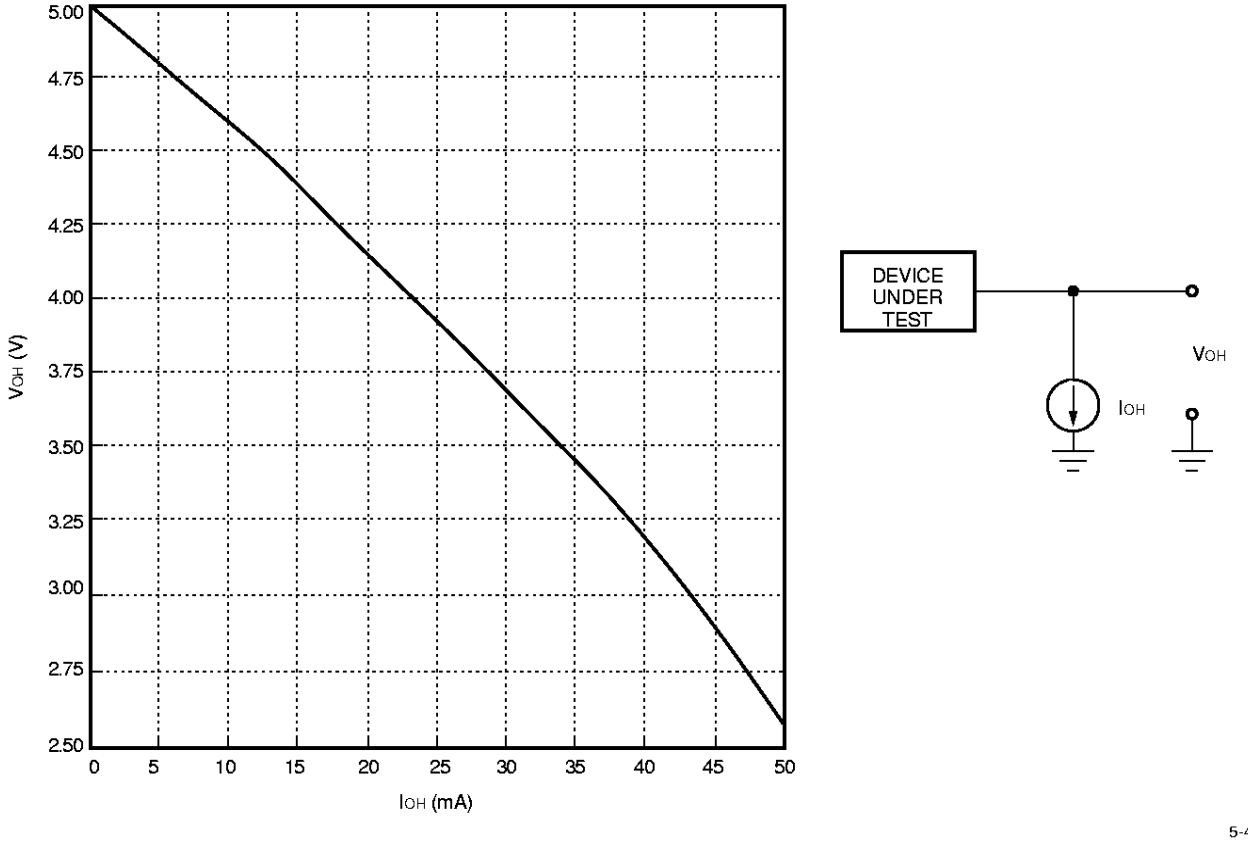


Figure 17. Plot of V_{OH} vs. I_{OH} Under Typical Operating Conditions

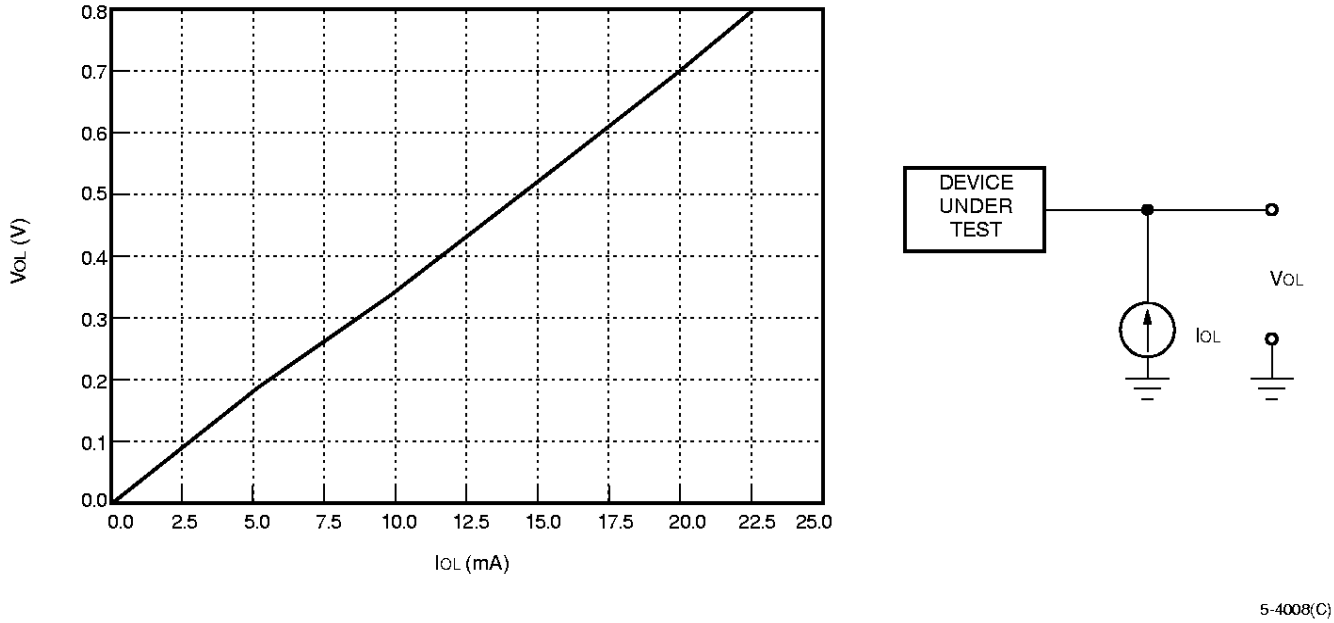


Figure 18. Plot of V_{OL} vs. I_{OL} Under Typical Operating Conditions

7 Electrical Requirements and Characteristics (continued)

Table 89. PLL Electrical Specifications, VCO Frequency Ranges

Parameter	Symbol	Min	Max	Unit
VCO frequency range[1]:				
V_{DD} = 3 V ± 10%	fVCO	50	150	MHz
V_{DD} = 5 V ± 10%	fVCO	50	200	MHz
Input Jitter at CKI	—	—	200	ps-rms

1. The M and N counter values in the **pllc** register must be set so that the VCO will operate in the appropriate range,

where $VCOfreq = CLKINfreq \times \frac{(m+2)}{(N+1)}$.

Choose the lowest value of N and then the appropriate value of M for internal MIPS rate = $\frac{(InputClock(m+2))}{2((K+1)(N+1))}$

Table 90. PLL Electrical Specifications and Register Settings

Voltage (Volts)	M pllc[5:0]	K pllc[8:6]	N pllc[10:9]	ICP pllc[13:11]	LF pllc[15:14]	PFIVOLT clkc[12]	Typical Lock Time (µs) (See Note.)
*FlashDSP*1609F							
3.0—3.6	>28	0—7	0	101	10	0	16—24
3.0—3.6	19—28	0—7	0	110	01	0	10—20
3.0—3.6	<19	0—7	0	111	00	0	8—15
DSP1609							
3.0—3.6	>28	0—7	0—1	101	11	0	8—20
3.0—3.6	19—28	0—7	0—1	110	10	0	8—20
3.0—3.6	<19	0—7	0—1	110	01	0	8—20
4.5—5.5	>28	0—7	0—1	110	11	1	8—20
4.5—5.5	19—28	0—7	0—1	111	10	1	8—20
4.5—5.5	<19	0—7	0—1	111	01	1	8—20

Note: Lock-in time represents the time following assertion of the PLLEN bit of the **pllc** register during which the PLL output clock is unstable. The DSP must operate from the CLKIN input clock or from CLKLOW while the PLL is locking.

7.1 Typical Power Dissipation

Power dissipation is highly dependent on program activity and the frequency of operation. Table 91 lists typical power dissipation on a module-by-module basis and consists of preliminary data that is subject to change. For analog modules, this data assumes the recommended external circuitry is being used. For digital modules, it refers to internal power only, i.e., Table 91 does not account for activity on digital I/O pins. The total analog and internal digital power can be calculated by adding the sum of the modules that have been enabled. For example, the lowest power dissipation can be achieved by turning everything off except for the internal low-power oscillator and setting the stop clock bit. In this case, the total power would be 62 µW at 3.1 V.

7 Electrical Requirements and Characteristics (continued)

7.1 Typical Power Dissipation (continued)

Table 91. Internal Module Power Dissipation

DSP Module	Enabled When	Instruction Rate (CLKFREE)	Typical Module Power Dissipation	
			5.0 V (mW)	3.3 V (mW)
Internal Low-power Oscillator	**clkc**[2] = 0	NA	0.175	0.062
4.096 MHz Crystal Oscillator	**clkc**[1] = 0 and OSCBYP pin = GND	NA	6	0.83
4 MHz Direct Clock Input (internal oscillator bypassed)	OSCBYP pin = V_{DD}	NA	2.5	0.52
SSI Enabled	**ssic**[15] = 1	NA	0.032 • $F_{CLKFREE}$[1]	0.012 • $F_{CLKFREE}$[1]
PLL Enabled	**clkc**[0] = 1	NA	1.65 + (0.092 • Fvco[1])	0.75 + (0.036 • Fvco[1])
Timer1 Enabled (32 kHz)	**timerc**[12] = 1	NA	<0.020	<0.005
Timer1 Enabled (1 MHz)	**timerc**[12] = 1	NA	0.150	0.046
Timer0 Enabled	**timerc**[6] = 0 and **timerc**[4] = 1	16 kHz	<0.020	<0.005
		512 kHz	0.030	0.012
		2 MHz	0.230	0.080
		>4 MHz	0.100 • $F_{CLKFREE}$[1]	0.031 • $F_{CLKFREE}$[1]
DRC Enabled	**drc**[15] = 1	—	—	—
SIO Enabled	**sioc**[15] = 1	—	—	—
Non-Stopclock Sleep Mode	**alf**[15] = 1 and **clkc**[3] = 0	16 kHz	<0.020	<0.005
		512 kHz	0.37	0.127
		2 MHz	0.795	0.493
		>4 MHz	0.678 • $F_{CLKFREE}$[1]	0.244 • $F_{CLKFREE}$[1]
Active[2]	**alf**[15] = 1 and **clkc**[3] = 0	16 kHz	0.21	0.071
		512 kHz	5.71	2.21
		2 MHz	25.2	8.74
		>4 MHz	9.4 • $F_{CLKFREE}$[1]	3.7 • $F_{CLKFREE}$[1]
Flash ROM (DSP1609F only)	Always	NA	NA	29.15

1. $F_{CLKFREE}$ and F_{VCO} refer to the frequency of the free-running clock and the frequency of the PLL VCO, both in units of megahertz (MHz).
2. Active power is measured while the DSP is executing a single-cycle, double memory fetch FIR filter from cache.

7 Electrical Requirements and Characteristics (continued)

7.2 Input and I/O Buffer Power Dissipation

Power dissipation due to the input and I/O buffers is highly dependent on the input voltage level. At full CMOS levels, essentially no dc current is drawn. However, for levels near the threshold of 1.4 V, high current can flow. Therefore, all unused input pins should be tied inactive to V_{DD} or V_{SS}, and all unused I/O pins should be tied inactive through the pull-up or pull-down options.

Note: This especially needs to be done for IOPA and IOPD in the 28-pin SOJ option, as these pins are not bonded-out and are floating.

Tables 92 and 93 show the input buffer power dissipation for 43 inputs biased at dc level, V_{IN}, with V_{DD} at 5.0 V.

Table 92. Input Buffer Maximum Power Dissipation

V_{IN} (V)	5.0	3.6	2.8	2.4	2.0	1.4	0.8	0.4	0
PD (mW)	<1.0	13.5	140	180	180	High	80	7.5	<1.0

WARNING: The device needs to be clocked for at least twelve f_{XTALB} cycles during reset after powerup; otherwise, high current may flow.

Table 93. Schmitt Trigger Input Buffer Maximum Power Dissipation

V_{IN} (V)	5.0	3.6	2.8	2.4	2.0	1.4	0.8	0.4	0
PD (mW)	<1.0	13.5	140	180	180	180	80	7.5	<1.0

8 Timing Requirements and Characteristics

Requirements are restrictions on the external device connected to the DSP1609. Characteristics are properties of the DSP1609. For details, see Section 6, Device Requirements and Characteristics, and Section 7, Electrical Requirements and Characteristics.

The characteristics listed are valid under the following conditions:

- V_{SS} = 0 V, V_{DD} = 5.0 V ± 0.5 V, T_A = 0 °C to 70 °C, C_{LOAD} = 50 pF, T_{MIN} = 12.5 ns
- V_{SS} = 0 V, V_{DD} = 3.3 V ± 0.3 V, T_A = –40 °C to +85 °C, C_{LOAD} = 50 pF, T_{MIN} = 12.5 ns

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

- For all rising edge outputs, $dt/dC_L \leq 0.06$ ns/pF for $0 \leq C_L \leq 100$ pF at 2.0 V
- For all falling edge outputs, $dt/dC_L \leq 0.05$ ns/pF for $0 \leq C_L \leq 100$ pF at 0.8 V

For example, the derating for a time delay that includes a rising edge with an external load of 20 pF is as follows:

$$\Delta t = (C_L - C_{LOAD})\, dt/dC_L = (20 - 50)\, 0.06 = -1.8 \text{ ns}$$

Output characteristics assume standard slew rate buffers. See Table 42 on page 59 for information on mask options, for slow buffers option all the output times should be increased by the additional time given by the equation:

$$\Delta t = 0.4 \bullet CL$$

Where CL is the load capacitance in picofarads, and Δt is the additional output delay in nanoseconds. For example, for load capacitance of 50 pF, with slow buffers option, all output timing characteristics should be increased by 20 ns.

Table 94. Test Conditions

Test Conditions for Inputs	Test Conditions for Outputs
Rise and fall times of 4 ns or less.	C_{LOAD} = 50 pF.
Timing reference levels for delays = V_{IH}, V_{IL}.	Timing reference levels for delays = V_{OH}, V_{OL}. 3-state delays are measured to the high-impedance state of the output driver.

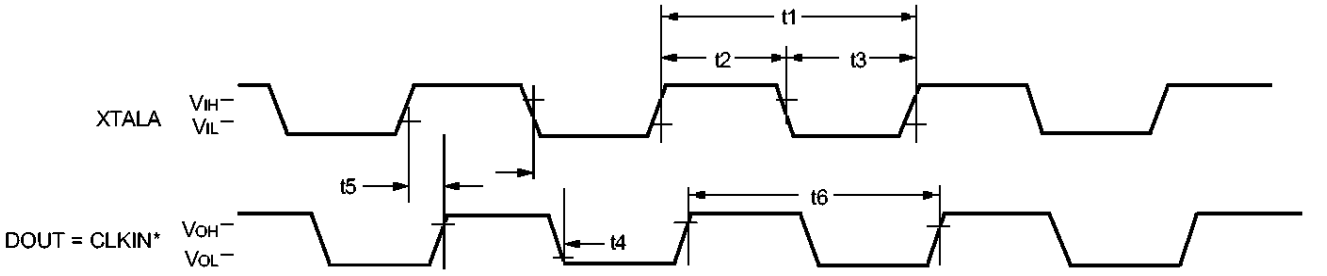
For definitions of the V_{IH}, V_{IL}, V_{OH}, and V_{OL} labels, see Table 87 on page 76. In this section's timing diagrams, these labels distinguish inputs and outputs.

8.1 Input Clock Options

The input clocks to the DSP1609 allow for the use of either CMOS level input signals or an internal crystal oscillator with an external crystal. This allows the user to select either both clock inputs to use the internal crystal oscillator or XTALA may be driven with a CMOS level signal. In the latter case, the OSCBYP pin should be pulled high externally, while XTAB is terminated by a 20 pF capacitor.

8 Timing Requirements and Characteristics (continued)

8.2 DSP Clock Generation



* Input clock; the DOUT pin may be selected to be the input clock, see the **clkc** register.

Figure 19. I/O Clock Timing Diagram

Note: For Tables 95 to 110, TMIN is the minimum instruction cycle time (see Section 8.1 on page 81).

Table 95. Timing Requirements for Input Clock

Ref	Parameter	DSP1609				DSP1609F		Unit
		VDD = 4.5 V to 5.5 V		VDD = 3.0 V to 3.6 V		VDD = 3.0 V to 3.6 V		
		TMIN = 12.5 ns		TMIN =		TMIN = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t1*	Clock In Period (low to low)	12.5	—†			12.5	—†	ns
t2	Clock In Low Time (low to high)	6	—			6	—	ns
t3	Clock In High Time (high to low)	6	—			6	—	ns

* The clock input frequency should normally be set to 4.096 MHz for proper PLL operation.
† Device is fully static; t1 is tested at 125 ns, and memory hold time is tested at 0.1 s.

Table 96. Timing Requirements for Input Clock and Output Clock

Ref	Parameter	DSP1609				DSP1609F		Unit
		VDD = 4.5 V to 5.5 V		VDD = 3.0 V to 3.6 V		VDD = 3.0 V to 3.6 V		
		TMIN = 12.5 ns		TMIN =		TMIN = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t4	Clock Out High Delay (high to high)	—	21			—	21	ns
t5	Clock Out Low Delay (low to low)	—	21			—	21	ns
t6	Clock Out Period (high to low)	12.5	—			12.5	—	ns

8 Timing Requirements and Characteristics (continued)

8.3 Reset Synchronization

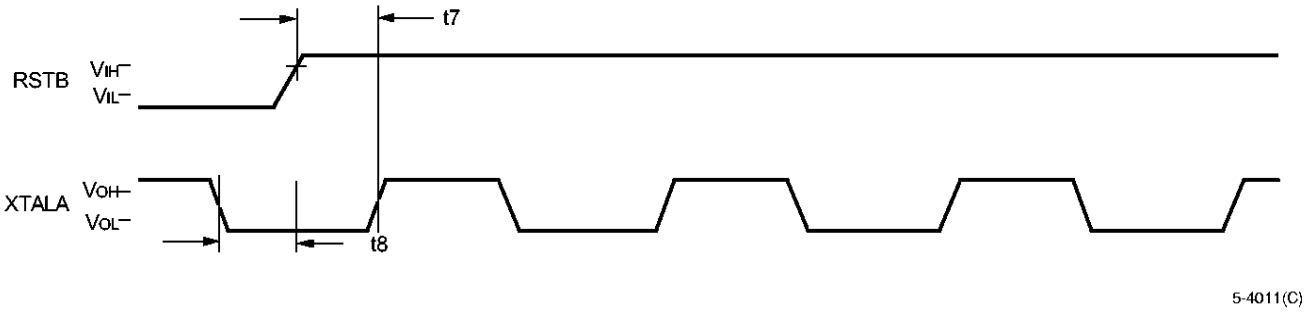


Figure 20. Reset Synchronization Timing

Table 97. Timing Requirements and Characteristics for Reset Synchronization Timing

Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t7	Reset Setup (high to high)	10	—			10	—	ns
t8	Clock Low to Reset High	12.5	—			12.5	—	ns

8 Timing Requirements and Characteristics (continued)

8.4 JTAG I/O Specifications

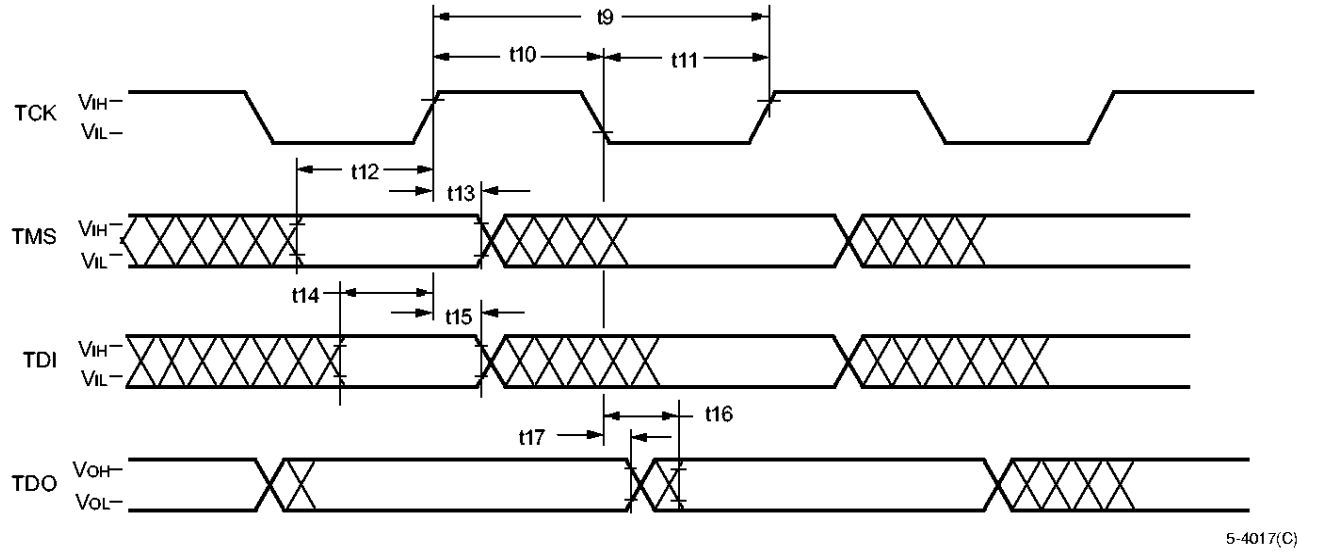


Figure 21. JTAG Timing Diagram

Table 98. Timing Requirements for JTAG Input

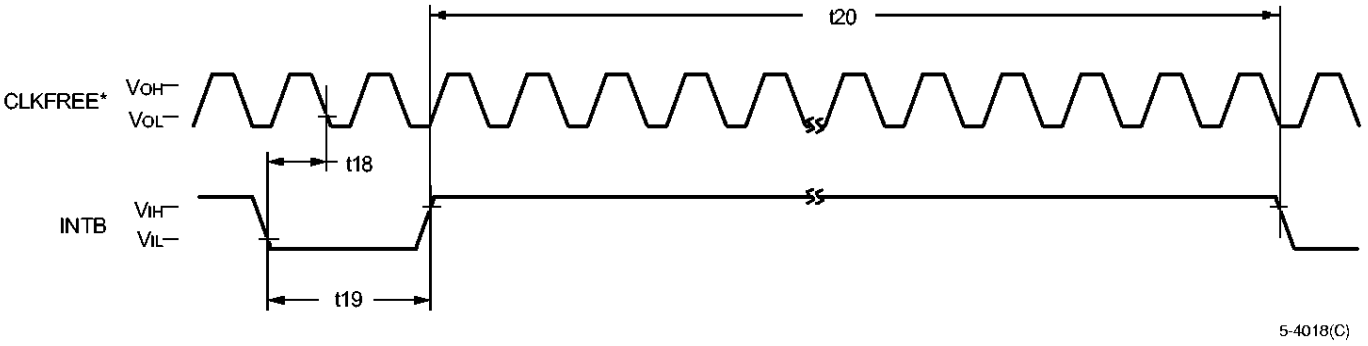
Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		Min	Max	Min	Max	Min	Max	
t9	TCK Period (high to high)	50	—			50	—	ns
t10	TCK High Time (high to low)	22	—			22	—	ns
t11	TCK Low Time (low to high)	22	—			22	—	ns
t12	TMS Setup Time (valid to high)	7	—			7	—	ns
t13	TMS Hold Time (high to invalid)	0	—			0	—	ns
t14	TDI Setup Time (valid to high)	7	—			7	—	ns
t15	TDI Hold Time (high to invalid)	0	—			0	—	ns

Table 99. Timing Characteristics for JTAG Output

Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		Min	Max	Min	Max	Min	Max	
t16	TDO Delay (low to valid)	—	22			—	22	ns
t17	TDO Hold (low to invalid)	0	—			0	—	ns

8 Timing Requirements and Characteristics (continued)

8.5 Interrupt



* CLKFREE is the free-running clock sourced to DOUT.

Figure 22. Interrupt Timing Diagram

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

Table 100. Timing Requirements for Interrupt

Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t18*	Interrupt Setup (low to low)	17	—			17	—	ns
t19	INTB Assertion Time (low to high)	2T	—			2T	—	ns
t20†	INTB Deassertion Time (high to low)	2T	—			2T	—	ns

* Only necessary for synchronized timing, no requirement for asynchronous assertion.
† In general, INTB deassertion time should be long enough to allow for the interrupt service routine to be completed.

Note: T = the period of the free-running clock, CLKFREE.

8 Timing Requirements and Characteristics (continued)

8.6 Input/Output Ports (IOP)

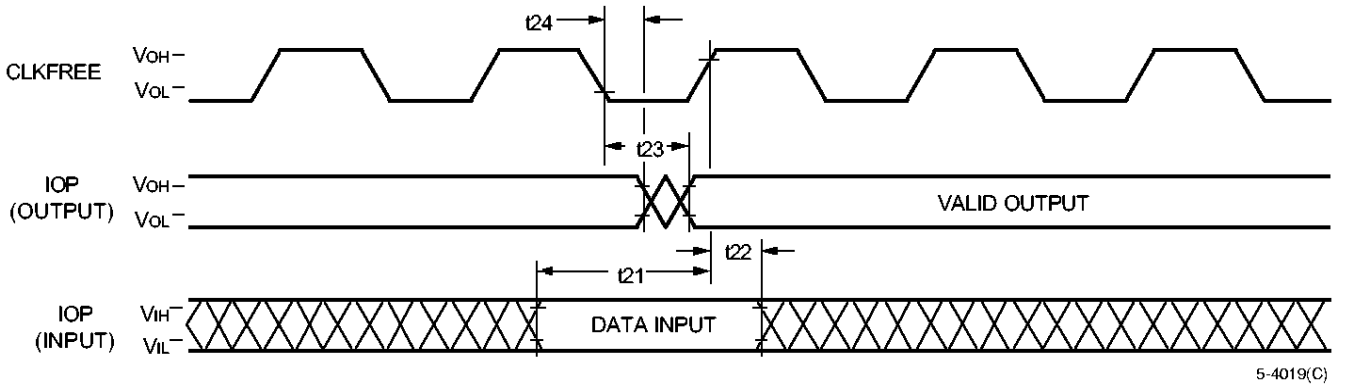


Figure 23. Write Outputs Followed by Read Inputs (cbit = Immediate; a1 = sbit)

Table 101. Timing Requirements for IOP Input Read

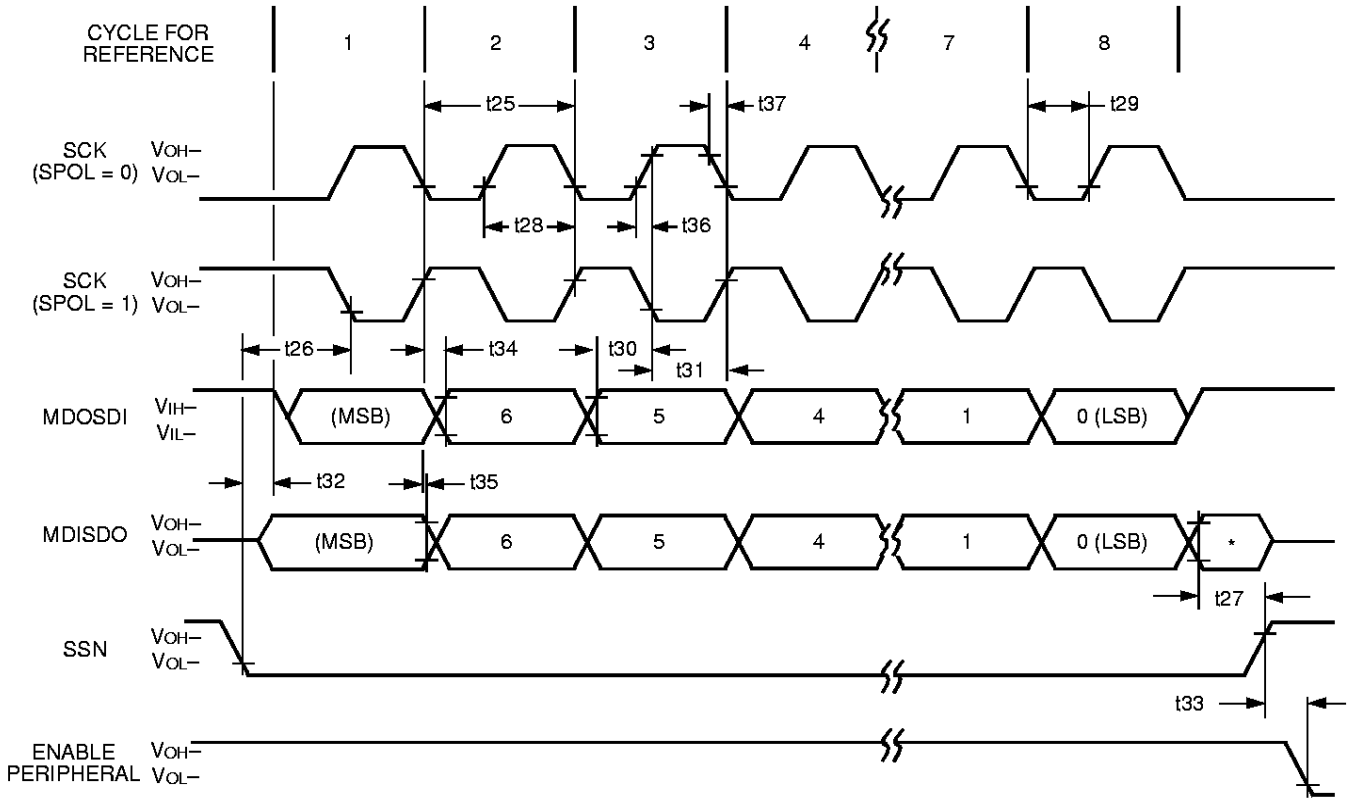
Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t21	IOP Input Setup Time (valid to high)	20	—			20	—	ns
t22	IOP Input Hold Time (high to invalid)	—	0			—	0	ns

Table 102. Timing Characteristics for IOP Output

Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t23	IOP Output Valid Time (low to valid)	—	4			—	4	ns
t24	IOP Output Hold Time (low to invalid)	–4	—			–4	—	ns

8 Timing Requirements and Characteristics (continued)

8.7 Synchronous Serial Interface (SSI) Specifications



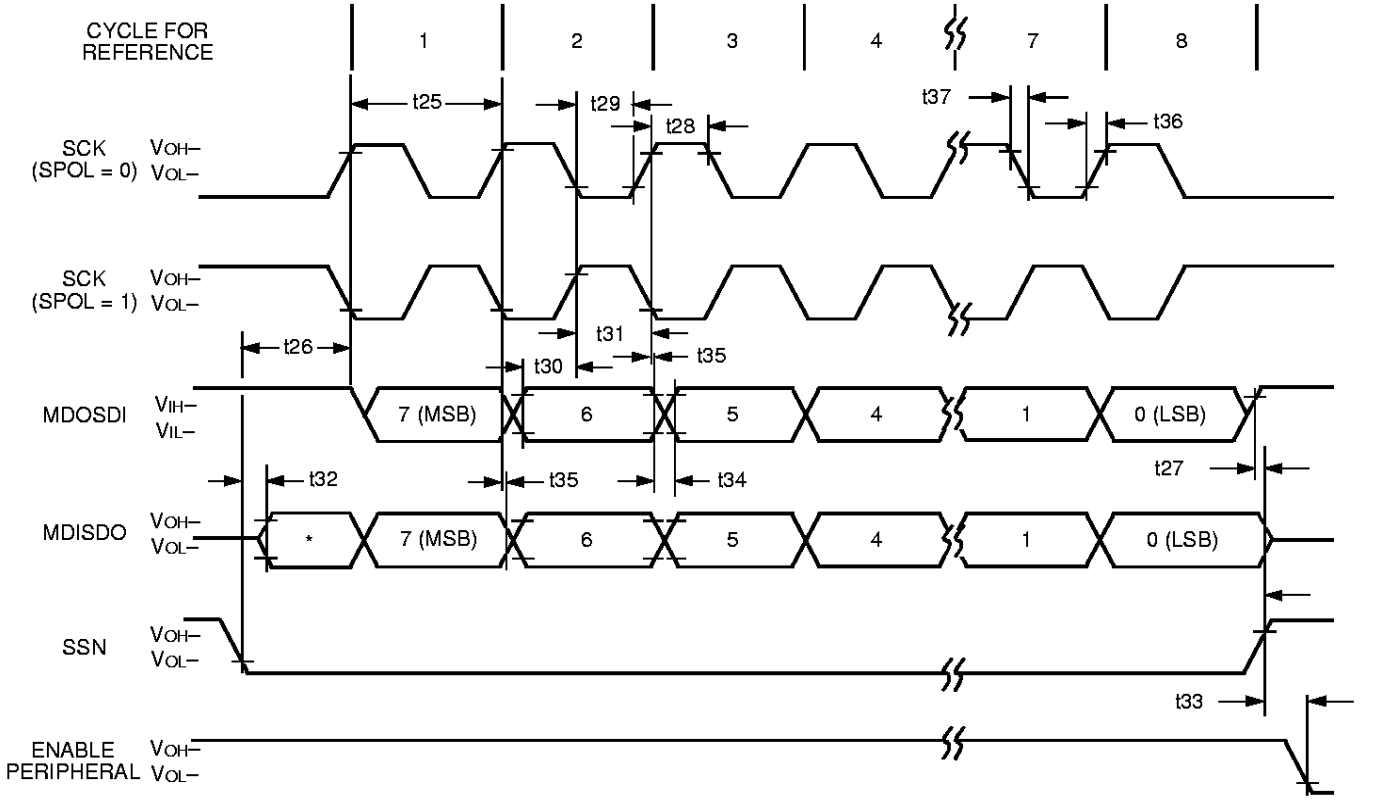
* Not defined.

12-3357 (F)

Figure 24. SSI Transfer Timing (SPHA = 0)

8 Timing Requirements and Characteristics (continued)

8.7 Synchronous Serial Interface (SSI) Specifications (continued)



* Not defined.

12-3358 (F)

Figure 25. SSI Transfer Timing (SPHA = 1)

8 Timing Requirements and Characteristics (continued)

8.7 Synchronous Serial Interface (SSI) Specifications (continued)

Table 103. Timing Characteristics for SSI*

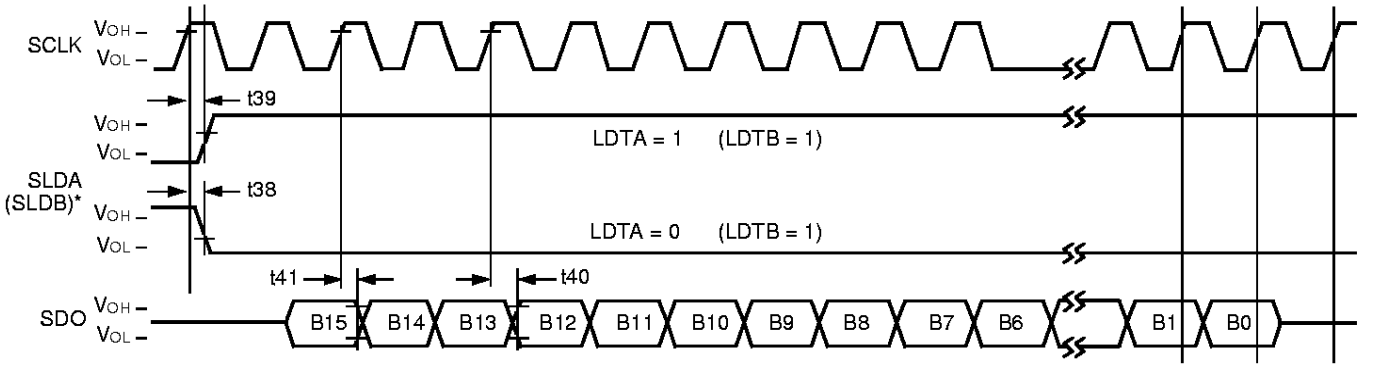
Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t25	Master Cycle Time	2T†	—			2T†	—	ns
	Slave Cycle Time	80	—			80	—	ns
t26	Slave Enable Lead Time	5	—			5	—	ns
t27	Slave Disable Idle Time	5	—			5	—	ns
t28	Master Clock High Time	T – 25	—			T – 25	—	ns
	Slave Clock High Time	40	—			40	—	ns
t29	Master Clock Low Time	T – 20	—			T – 20	—	ns
	Slave Clock Low Time	40	—			40	—	ns
t30‡	Input Data Setup Time (slave mode)	12	—			12	—	ns
t31‡	Input Data Hold Time (slave mode)	0	—			0	—	ns
t32	Slave Data Out Access Time	0	60			0	60	ns
t33	Slave Data Out Disable Time	0	12			0	12	ns
t34	Output Data Valid After Clock	—	80			—	80	ns
t35	Output Data Hold Time After Clock	0	—			0	—	ns
t36	Output Rise Time	1	25			1	25	ns
	Input Rise Time	—	100			—	100	ns
t37	Output Fall Time	1	25			1	25	ns
	Input Fall Time	—	100			—	100	ns

* These specifications are derived from the 68HC11 specification. Some of the timing definitions may change later to be compatible with the other timing definitions in this document.
† T is the period of CLKFREE.
‡ t30 and t31 are given for slave operation. For master mode, input data setup time is 20 ns minimum and input data hold time is still 0 ns.

8 Timing Requirements and Characteristics (continued)

8.8 Serial I/O Specifications

The load clocks (SLDA and SLDB) have 50% duty cycle and are offset from each other by half a cycle.



* SLDB for channel B in dual-channel mode.

5-4031A (F)

Figure 26. SIO Active Output Timing Diagram

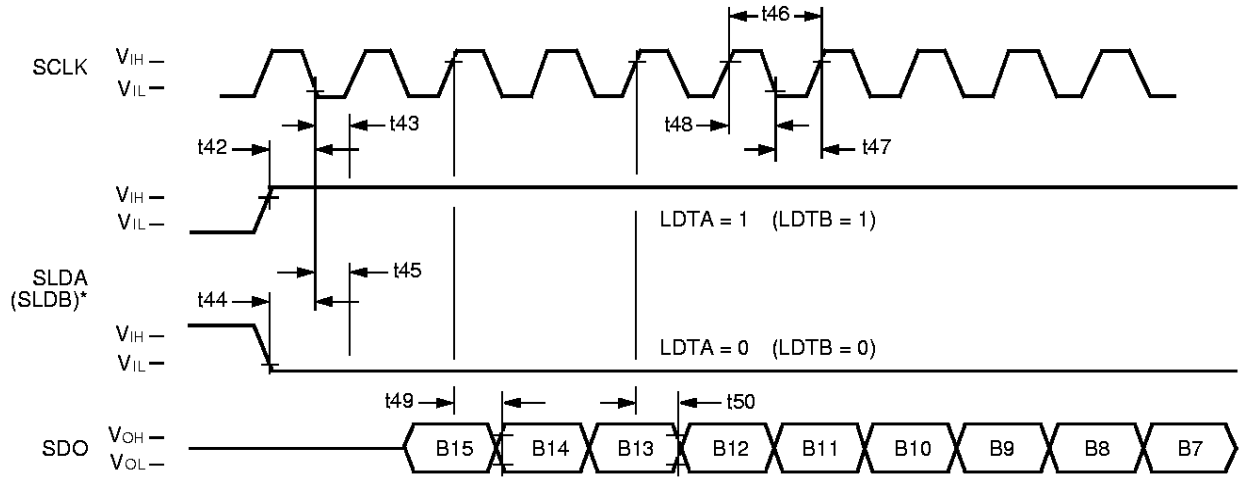
Table 104. Timing Characteristics for Serial Active Output*

Ref	Parameter	DSP1609				DSP1609F		Unit
		VDD = 4.5 V to 5.5 V		VDD = 3.0 V to 3.6 V		VDD = 3.0 V to 3.6 V		
		TMIN = 12.5 ns		TMIN =		TMIN = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t38	SLDA Delay (high to low)	—	16			—	16	ns
t39	SLDA Delay (high to low)	—	16			—	16	ns
t40	Data Delay (high to valid)	—	22			—	22	ns
t41	Data Hold (high to invalid)	3	—			3	—	ns

* Capacitance load on SCKA and SDO equals 100 pF.

8 Timing Requirements and Characteristics (continued)

8.8 Serial I/O Specifications (continued)



* SLDB for channel B in dual-channel mode.

5-4032 (F)

Figure 27. SIO Passive Output Timing Diagram

Table 105. Timing Requirements for Serial Passive Output

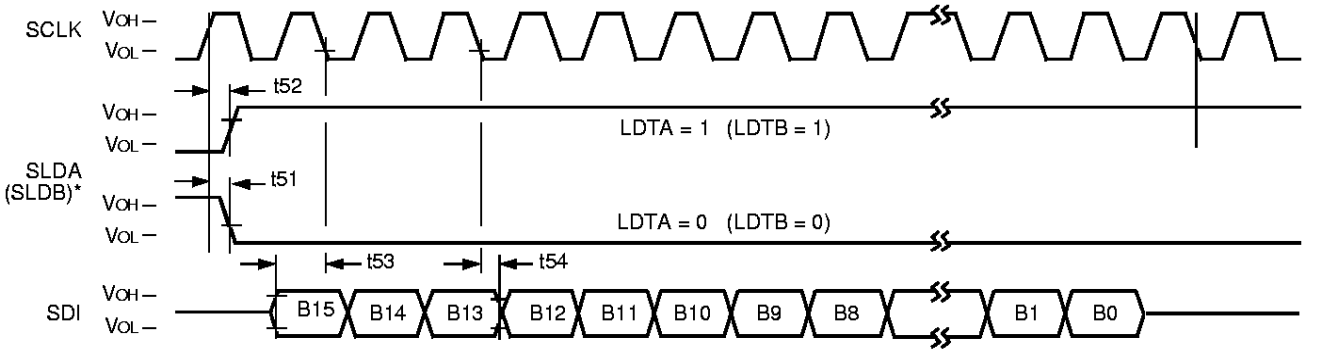
Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t42	SLDA Setup (high to low)	4	—			4	—	ns
t43	SLDA Hold (low to invalid)	4	—			4	—	ns
t44	SLDA Setup (high to low)	4	—			4	—	ns
t45	SLDA Hold (low to invalid)	4	—			4	—	ns
t46	Clock Period (high to high)	50	—			50	—	ns
t47	Clock Low Time (low to high)	22	—			22	—	ns
t48	Clock High Time (high to low)	22	—			22	—	ns

Table 106. Timing Characteristics for Serial Passive Output

Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t49	Data Delay (high to valid)	—	22			—	22	ns
t50	Data Hold (high to invalid)	4	—			4	—	ns

8 Timing Requirements and Characteristics (continued)

8.8 Serial I/O Specifications (continued)



* SLDB for channel B in dual-channel mode.

5-4033a (F)

Figure 28. SIO Active Input Timing Diagram

Table 107. Timing Characteristics for Serial Active Input Clocks

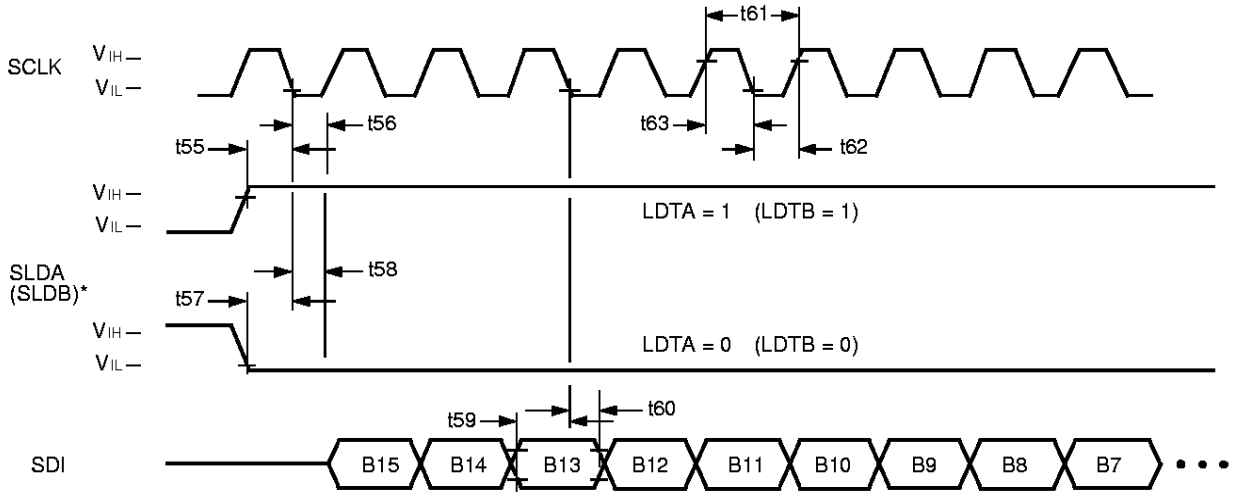
Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t51	SLDB Delay (high to low)	—	13			—	13	ns
t52	SLDB Delay (high to high)	—	14			—	14	ns

Table 108. Timing Requirements for Serial Active Input

Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t53	Data Setup (valid to low)	2	—			2	—	ns
t54	Data Hold (low to invalid)	4	—			4	—	ns

8 Timing Requirements and Characteristics (continued)

8.8 Serial I/O Specifications (continued)



* SLDB for channel B in dual-channel mode.

5-4035a (F)

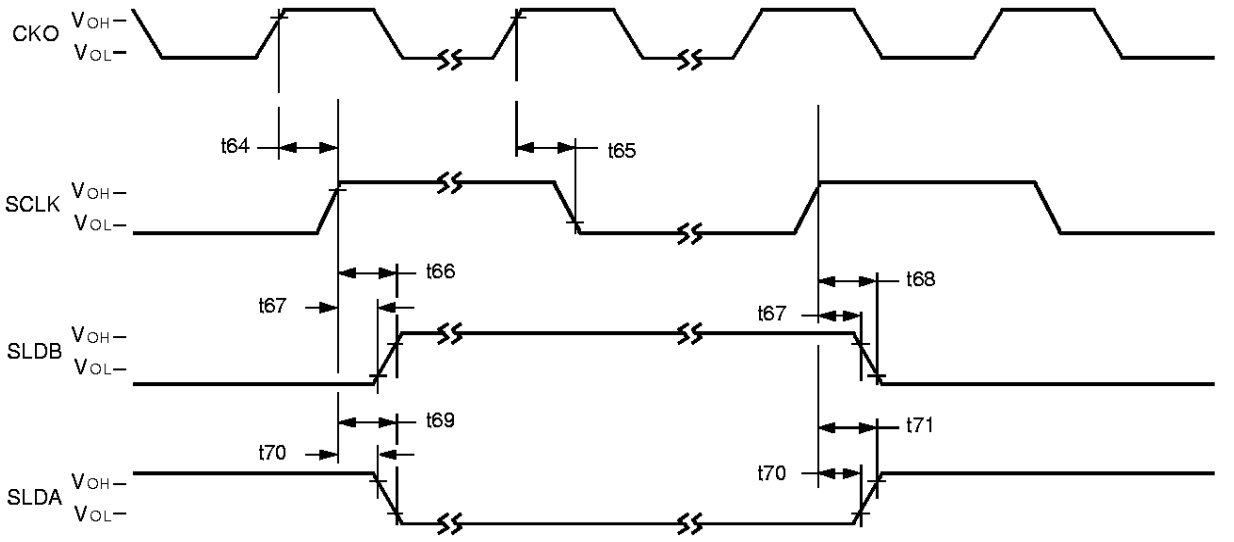
Figure 29. SIO Passive Input Timing Diagram

Table 109. Timing Characteristics for Serial Passive Input

Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t55	SLDA Setup (high to low)	4	—			4	—	ns
t56	SLDA Hold (low to invalid)	4	—			4	—	ns
t57	SLDA Setup (low to low)	4	—			4	—	ns
t58	SLDA Hold (low to invalid)	4	—			4	—	ns
t59	Data Setup (valid to low)	2	—			2	—	ns
t60	Data Hold (low to invalid)	4	—			4	—	ns
t61	Clock Period (high to high)	50	—			50	—	ns
t62	Clock Low Time (low to high)	22	—			22	—	ns
t63	Clock High Time (high to low)	22	—			22	—	ns

8 Timing Requirements and Characteristics (continued)

8.8 Serial I/O Specifications (continued)



* SLDB for channel B in dual-channel mode.

5-4035a (F)

Figure 30. Serial I/O Active Clocks Timing Diagram

Table 110. Timing Characteristics for Signal Generation

Ref	Parameter	DSP1609				DSP1609F		Unit
		V_{DD} = 4.5 V to 5.5 V		V_{DD} = 3.0 V to 3.6 V		V_{DD} = 3.0 V to 3.6 V		
		T_{MIN} = 12.5 ns		T_{MIN} =		T_{MIN} = 12.5 ns		
		Min	Max	Min	Max	Min	Max	
t64	SCLK Delay (high to high)	—	16			—	16	ns
t65	SCLK Delay (high to low)	—	16			—	16	ns
t66	SLDB Delay (high to high)	—	38			—	38	ns
t67	SLDB Hold (high to invalid)	2	—			2	—	ns
t68	SLDB Delay (high to low)	—	38			—	38	ns
t69	SLDA Delay (high to low)	—	38			—	38	ns
t70	SLDA Hold (high to invalid)	2	—			2	—	ns
t71	SLDA Delay (high to high)	—	38			—	38	ns

9 Crystal Oscillator Electrical Requirements and Characteristics

This section describes electrical requirements and characteristics for high- and low-frequency crystal oscillator circuits.

9.1 Crystal Oscillator

If the option for using the external crystal is chosen, the electrical requirements and characteristics described in this section apply.

9.1.1 Crystal Oscillator Power Dissipation

The typical power dissipation at 4.096 MHz of the internal high-frequency crystal oscillator circuit below is 6 mW at 5 V and 4 mW at 3 V.

9.1.2 Crystal Oscillator External Components

The crystal oscillator is enabled by connecting a crystal across XTALA and XTALB, along with one external capacitor from each of these pins to ground (see Figure 31). For most applications, 15 pF external capacitors are recommended; however, larger values may be necessary if precise frequency tolerance is required (see Section 9.2, Frequency Accuracy Considerations). The crystal should be either fundamental or overtone mode, parallel resonant, with a power dissipation 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 Figures 32 and 33 for the crystal frequency.

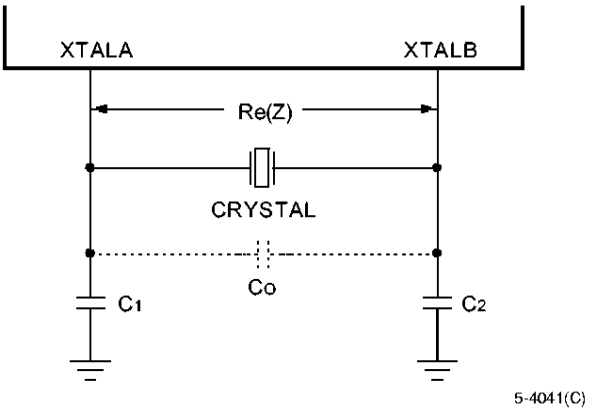


Figure 31. Fundamental Crystal Configuration

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

- Keep crystal and external capacitors as close to XTALA and XTALB pins as possible to minimize board stray capacitance.
- Keep high-frequency digital signals such as DOUT away from XTALA and XTALB traces to avoid coupling.

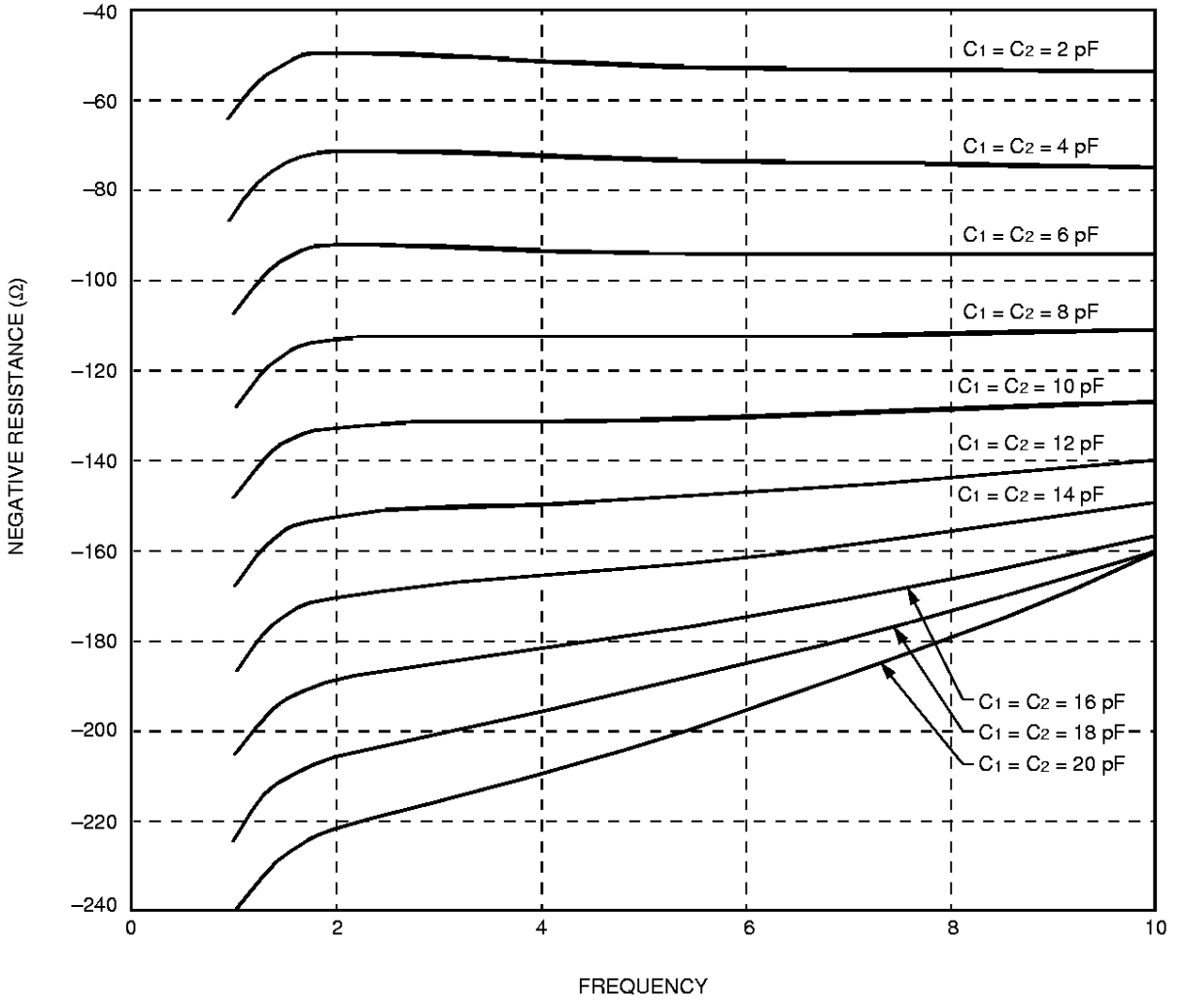
9 Crystal Oscillator Electrical Requirements and Characteristics (continued)

9.1 Crystal Oscillator (continued)

9.1.3 Crystal Oscillator Negative Resistance Curves

Figure 32 shows worst-case negative resistance curves for a high-frequency crystal oscillator operating with a 5 V power supply (V_{DD} = 4.5 V to 5.5 V). These worst-case conditions are as follows:

- Maximum temperature = 120 °C
- Minimum V_{DD} = 4.5 V
- Maximum C_O = 7 pF



Note: V_{DD} = 5 V.

5-6154 (F)

Figure 32. 5 V Crystal Oscillator Negative Resistance Curves

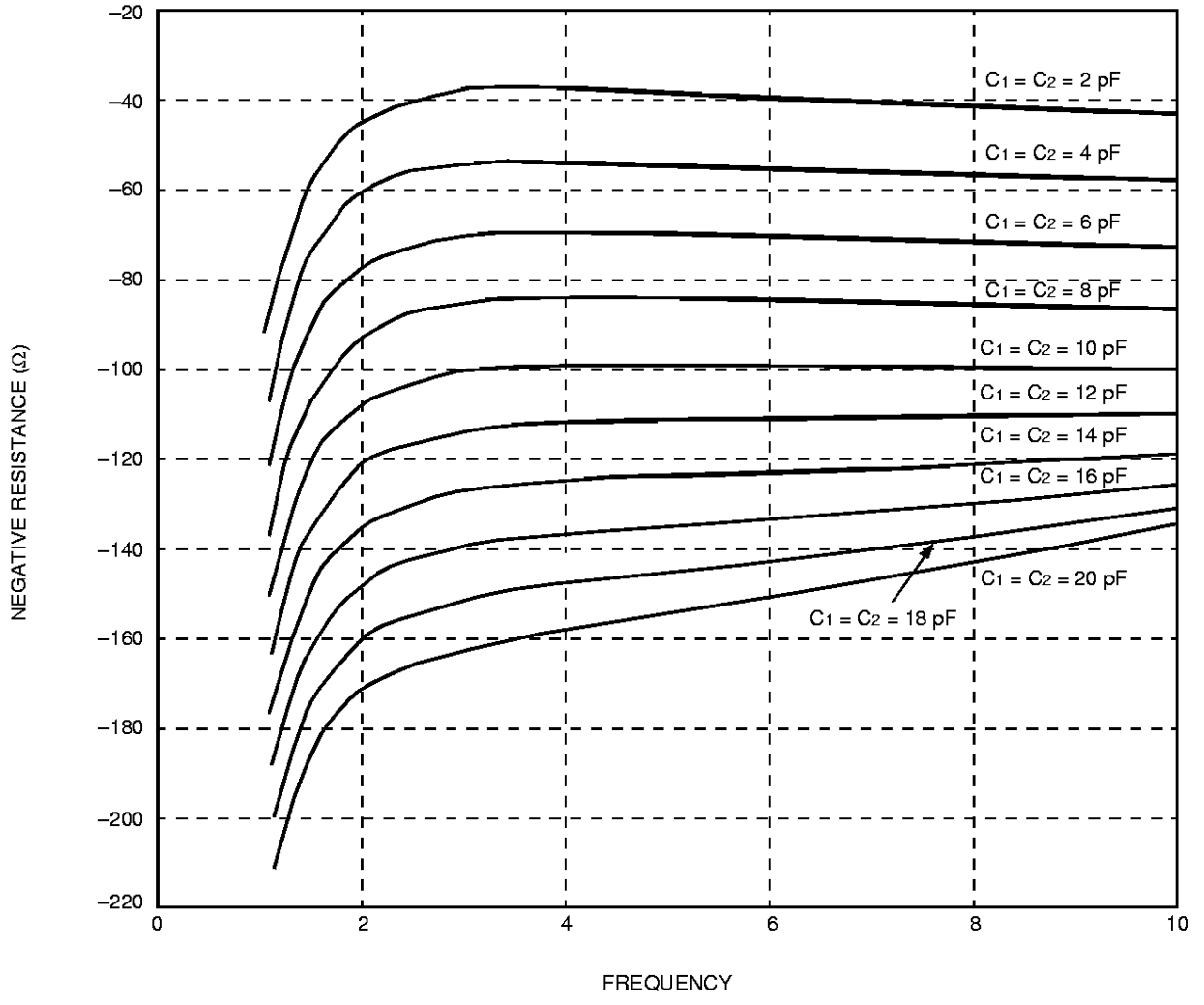
9 Crystal Oscillator Electrical Requirements and Characteristics (continued)

9.1 Crystal Oscillator (continued)

9.1.3 Crystal Oscillator Negative Resistance Curves (continued)

Figure 33 shows worst-case negative resistance curves for a crystal oscillator operating with a 3.3 V power supply (V_{DD} = 3.0 V to 3.6 V). These worst-case conditions are as follows:

- Maximum temperature = 120 °C
- Minimum V_{DD} = 3.0 V
- Maximum C_O = 7 pF



Note: V_{DD} = 3 V.

5-6155 (F)

Figure 33. 3.3 V Crystal Oscillator Negative Resistance Curves

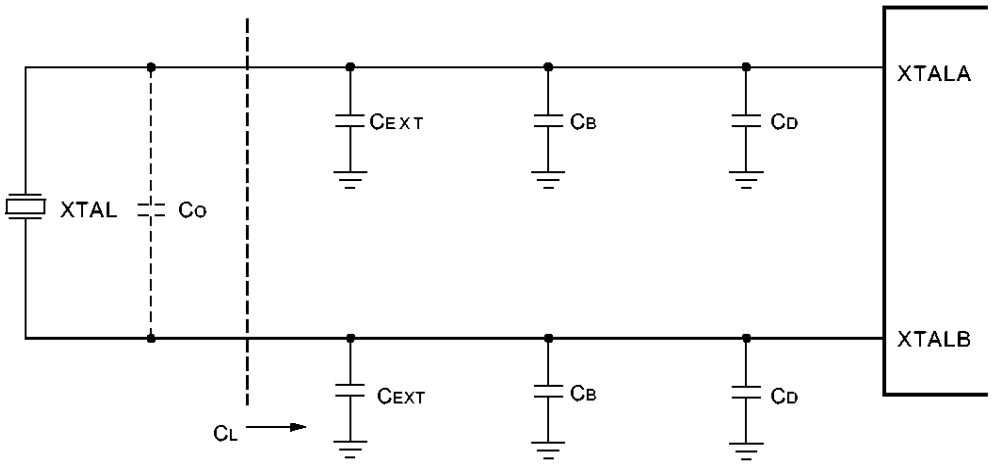
9 Crystal Oscillator Electrical Requirements and Characteristics (continued)

9.2 Frequency Accuracy Considerations

For most applications, clock frequency errors in the hundreds of parts per million (ppm) can be tolerated with no adverse effect. However, for applications where precise frequency tolerance on the order 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 is not 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 crystal operating frequency is measured. Variations in this load capacitance due to temperature and manufacturing variations cause variations in the operating frequency of the oscillator. Figure 34 illustrates some of the sources of this variation.

C_{EXT} = External load capacitor (one each required for XTALA and XTALB)
C_D = Parasitic capacitance of the DSP1609 itself
C_B = Parasitic capacitance of the printed-wiring board
C_O = Parasitic capacitance of crystal (not part of C_L, but still a source of frequency variation)



5-4045(C)

Figure 34. Components of Load Capacitance for Crystal Oscillator

The load capacitance, C_L, must be specified to the crystal vendor. The crystal manufacturer cuts the crystal so that the frequency of oscillation is 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 34, 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, 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 parasitic capacitance, the frequency also deviates. This change in frequency as a function of load capacitance is known as pullability, which is expressed in units of ppm/pF.

9 Crystal Oscillator Electrical Requirements and Characteristics (continued)

9.2 Frequency Accuracy Considerations (continued)

For small deviations of a few pF, pullability can be determined by the following equation.

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

where: C_O= parasitic capacitance of crystal
C_1 = motional capacitance of crystal (usually around 1 fF—25 fF, value can be obtained from crystal vendor)
C_L = total load capacitance seen by crystal

Note: For a given crystal, making C_L as large as possible can reduce pullability and improve frequency stability, while still maintaining sufficient negative resistance to ensure start-up according to the curves shown in Figures 32 and 33.

Because the exact values of the parasitic capacitance in a crystal-based oscillator system are unknown, 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 15 pF, each external capacitor would have to be slightly less than 30 pF to account for parasitic capacitance. Suppose, for instance, that a crystal for which C_L = 15 pF is specified is plugged into the system and it is determined empirically that the best frequency accuracy occurs with C_{EXT} = 28 pF. This would mean that the equivalent board and device parasitic capacitance from each lead to ground would be 2 pF.

As an example, suppose it is desired to design a 4.096 MHz, 5.0 V system with ±100 ppm frequency accuracy. The parameters for a typical high-accuracy, custom, 4.096 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_O)	7 pF max

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 32 on page 96, external capacitors plus strays can be made as large as 30 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 30 pF per lead, or 40 pF total. Thus, a load capacitance, C_L = 15 pF would be specified to the crystal manufacturer.

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

$$\text{pullability (ppm/pF)} = \frac{(C_1)(10^6)}{2(C_O + C_L)^2} = \frac{(0.015)(10^6)}{2(7 + 15)^2} = 15.5 \text{ ppm/pF}$$

If 2% external capacitors are used, the frequency deviation due to this variation is equal to:

(0.02)(15 pF)(15.5 ppm/pF) = 4.7 ppm

Note: To simplify analysis, C_{EXT} is considered to be 40 pF. In practice, it would be slightly less than this value to account for strays. Also, temperature and aging tolerance on the capacitors has been neglected.

9 Crystal Oscillator Electrical Requirements and Characteristics (continued)

9.2 Frequency Accuracy Considerations (continued)

Typical capacitance variation of oscillator circuit in the DSP1609 itself across process, temperature, and supply voltage is ±1 pF. Thus, the expected frequency variation due to the DSP1609 is:

(1 pF)(3.4 ppm/pF) = 3.4 ppm

Other frequency accuracy considerations include the following:

- Approximate variation in parasitic capacitance of crystal = ±0.5 pF
- Frequency shift due to variation in C_O = (0.5 pF)(3.4 ppm/pF) = 1.7 ppm
- Approximate variation in parasitic capacitance of printed-circuit board = ±1.5 pF
- Frequency shift due to variation in board capacitance = (1.5 pF)(3.4 ppm/pF) = 5.1 ppm

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

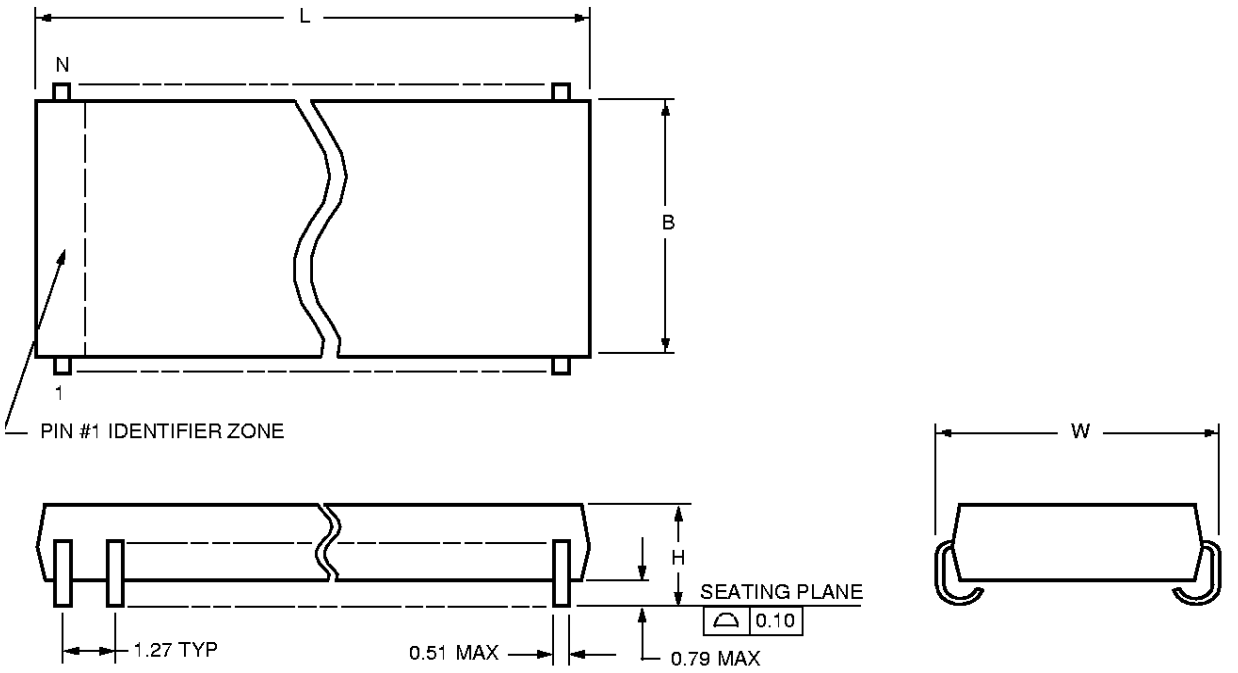
Initial Tolerance of Crystal =	10.0 ppm
Temperature Tolerance of Crystal =	25.0 ppm
Aging Tolerance of Crystal =	6.0 ppm
Load Capacitor Variation =	2.8 ppm
DSP1609 Circuit Variation =	3.4 ppm
C_O Variation =	1.7 ppm
Board Variation =	5.1 ppm
Total =	54.0 ppm

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

10 Outline Diagrams

10.1 28-Pin SOJ Outline Diagram

Dimensions are in millimeters.

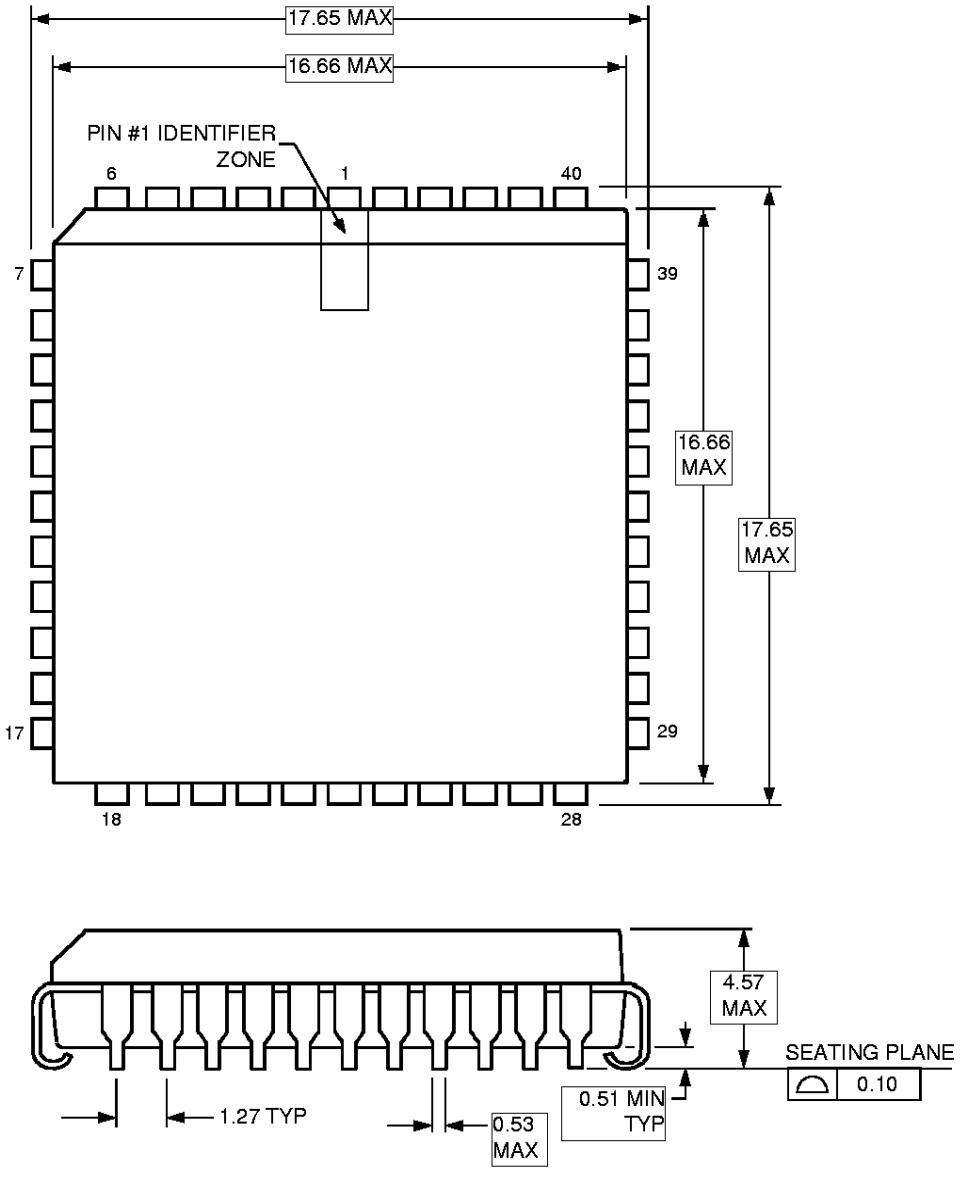


5-4413 (F)

10 Outline Diagrams (continued)

10.2 44-Pin PLCC

Dimensions are in millimeters.



5-2506 (F)

11 Ordering Information

This section describes the DSP1609/*FlashDSP*1609F device coding and mask-programmable options.

11.1 Device Coding

Table 111 defines each DSP1609/*FlashDSP*1609F part number character.

Table 111. DSP1609/*FlashDSP*1609F Device Coding

Part Number Character Positions (Left to Right)							
1—4	5	6—7	8	9—10	11—12	13	14
1609 Family	Package (E = SOJ, M = PLCC)	ROM Size (Number of 16-bit Kwords)	Designator (P = No Designation, F = Flash ROM)	ROM Code ID (AA—ZZ) or Flash Pinout Option	Instruction Cycle Time (ns)	Temperature Class (– or Blank = Commercial, I = Industrial)	Supply Voltage [Blank (V_{DD} = 4.5 V to 5.5 V) or T (V_{DD} = 3.0 V to 3.6 V)]
1609	E or M	24	P	—	12	– or Blank	Blank
					12	– or Blank	T
1609	M	24	F	—	12	– or Blank	T
					12	– or Blank	T

11.2 Mask-Programmable Options

The DSP1609 contains a ROM that is mask-programmable (see Table 42 on page 60). Encoding a custom ROM selects the following programming options:

- The JTAG fields SLEWR, SECURE, BOPT, and ROMCODE.
- ROM security. This option protects the internal ROM contents by not allowing access to the instruction/coefficient memory map 3. Restricting access to memory map 3 makes it impossible to access IROM space when running from IRAM space.

12 Product Preview: DSP1609

12.1 DSP1609 Features

- For 5 V operation:
  — 10.0 ns instruction cycle time (100 MIPS) at 5.0 ± 0.25 V, 0—70 °C
  — 11.11 ns instruction cycle time (90 MIPS) at 5.0 ± 0.5 V, 0—85 °C

12.2 DSP1609 Description

The DSP1609 is a 16-bit, fixed-point digital signal processor (DSP) based on the DSP1600 core. It is programmable to perform a wide variety of fixed-point signal processing functions. A member of the DSP1600 family, the DSP1609 includes a mix of peripherals specifically intended to support processing-intensive but cost-sensitive applications. In addition to the core, the DSP1609 consists of the following peripheral blocks: a programmable phase-locked loop (PLL), synchronous serial interface unit (SSI), four I/O ports (IOPs), two timer units, a watchdog timer, one dual-channel serial I/O interface (SIO), and a JTAG interface; as well as 2 Kwords of RAM. The DSP1609 is part of a low-cost, high-performance solution for consumer product applications.

The DSP1609 is available in the following packages:

- 28-pin SOJ
- 44-pin PLCC

The DSP1609 achieves high throughput without programming restrictions or latencies due to its parallel pipelined architecture. The processor has an arithmetic unit capable of a 16 x 16-bit multiplication and 36-bit accumulation, or a 32-bit ALU operation in one instruction cycle. Data is accessed from memory through two independent addressing units.

A fully static, low-power, 0.30 µm CMOS design and a low-power standby mode support power-sensitive equipment applications. A single external crystal allows the use of a high-frequency and a low-frequency clock. Under program control, the DSP1609 can be switched between the high-frequency and low-frequency clock options. When switched to the low-frequency clock, the power is reduced and can be further reduced using a stop-clock mode.

The *FlashDSP*1609F device is the development platform for the DSP1609. To support in-circuit emulation, the *FlashDSP*1609F device includes an internal HDS module and an internal flash ROM for program development.

For additional information, contact your Microelectronics Group Account Manager or the following:

INTERNET: **http://www.lucent.com/micro**
E-MAIL: **docmaster@micro.lucent.com**
U.S.A.: Microelectronics Group, Lucent Technologies Inc., 555 Union Boulevard, Room 30L-15P-BA, Allentown, PA 18103
1-800-372-2447, FAX 610-712-4106 (In CANADA: **1-800-553-2448**, FAX 610-712-4106)
ASIA PACIFIC: Microelectronics Group, Lucent Technologies Singapore Pte. Ltd., 77 Science Park Drive, #03-18 Cintech III, Singapore 118256
Tel. (65) 778 8833, FAX (65) 777 7495
JAPAN: Microelectronics Group, Lucent Technologies Japan Ltd., 7-18, Higashi-Gotanda 2-chome, Shinagawa-ku, Tokyo 141, Japan
Tel. (81) 3 5421 1600, FAX (81) 3 5421 1700
EUROPE: Data Requests: MICROELECTRONICS GROUP DATALINE: **Tel. (44) 1189 324 299**, FAX (44) 1189 328 148
Technical Inquiries: GERMANY: **(49) 89 95086 0** (Munich), UNITED KINGDOM: **(44) 1344 865 900** (Bracknell UK),
FRANCE: **(33) 1 41 45 77 00** (Paris), SWEDEN: **(46) 8 600 7070** (Stockholm), FINLAND: **(358) 9 4354 2800** (Helsinki),
ITALY: **(39) 2 6601 1800** (Milan), SPAIN: **(34) 1 807 1441** (Madrid)

February 1998
DS97-298DCON

microelectronics group

Lucent Technologies
Bell Labs Innovations