DSP56301

24-BIT
DIGITAL SIGNAL PROCESSOR
USER'S MANUAL

MOTOROLA

TABLE OF CONTENTS

MEMORY MAPS

CORE CONFIGURATION

PERIPHERALS

Table of Contents (Continued)

**Paragraph
Number**
                                 Title
**Page
Number**

APPENDIX

TABLE OF CONTENTS **MOTOROLA**

8   ENHANCED SYNCHRONOUS SERIAL INTERFACE (ESSI)

9   SERIAL COMMUNICATION INTERFACE (SCI)

LIST of TABLES

1 CHIP DESCRIPTION

This document describes the DSP56301, a new member of the 56300 DSP family of programmable CMOS digital signal processors. The DSP56301 is based on the powerful New DSP Engine (DSP56300) core capable of executing an instruction on every clock cycle, thus yielding a twofold performance increase as compared to the existing 56000 core while maintaining object code compatibility with it.

The DSP56300 core is composed of the Data ALU, Address Generation Unit, Program Controller, Instruction-Cache Controller, Bus Interface Unit, DMA Controller, On-chip Emulator and a PLL based clock oscillator. The DSP56300 core-based family members are DSP chips that contain the DSP56300 core and additional modules. The modules are chosen from a library of standard pre-designed elements such as memories, peripherals etc. New modules may be added to the library as required by customer specifications. A standard interface between the DSP56300 core and the on-chip memory and peripherals supports all memory and peripheral configurations.

The DSP56301 block diagram is shown in Figure 1-1 and its key feature are listed below.

The design priorities for the DSP56301 chip are:

1. Low-cost
2. Low-power dissipation
3. High-performance
4. High integration

DSP56301 Features

High performance DSP56300 core
- 66/80 Million Instructions per Second (Mips) with a 66/80 Mhz clock
- Object Code Compatible with the 56K Core
- Fully pipelined 24 x 24 Bit Parallel Multiplier-Accumulator
- 56 Bit Parallel Barrel Shifter
- 16 Bit Arithmetic Support
- Highly Parallel Instruction Set
- Position Independent Code (PIC) support
- Unique DSP Addressing Modes
- On-Chip Memory-Expandable Hardware Stack
- Nested Hardware Do Loops
- Fast Auto-Return Interrupts
- On-Chip Concurrent Six-Channel DMA Controller
- On-Chip PLL
- On-Chip Emulator (OnCE)
- JTAG port

On-chip memories
- On-Chip 2048 x 24 Bit X Data RAM
- On-Chip 2048 x 24 Bit Y Data RAM
- On-Chip 3072 x 24 Bit Program RAM
- On-Chip 1024 x 24 Bit Instruction Cache/Program RAM
- On-Chip 192 x 24 Bit Bootstrap ROM

Off-chip memory expansion
- Off-Chip Expansion to two 2^{24} 24-Bit Words of Data Memory
- Off-Chip Expansion to 2^{24} 24-Bit Words of Program Memory
- External Memory Expansion Port
- Chip Select Logic for glueless interface to SRAMs and SSRAMs
- On-chip DRAM Controller for glueless interface to DRAMs

On-chip peripherals
- 32-Bit Parallel PCI/Universal Host Interface
- Two Enhanced Synchronous Serial Interfaces (ESSI)
- Serial Communications Interface with Baud Rate Generator (SCI)
- Triple Timer Module
- 42 Programmable General Purpose I/O Pins (GPIO)

Reduced power dissipation
- Very low power CMOS design
- Wait and Stop low power standby modes
- Fully-static logic, operation frequency down to DC.
- Power Management special circuitry
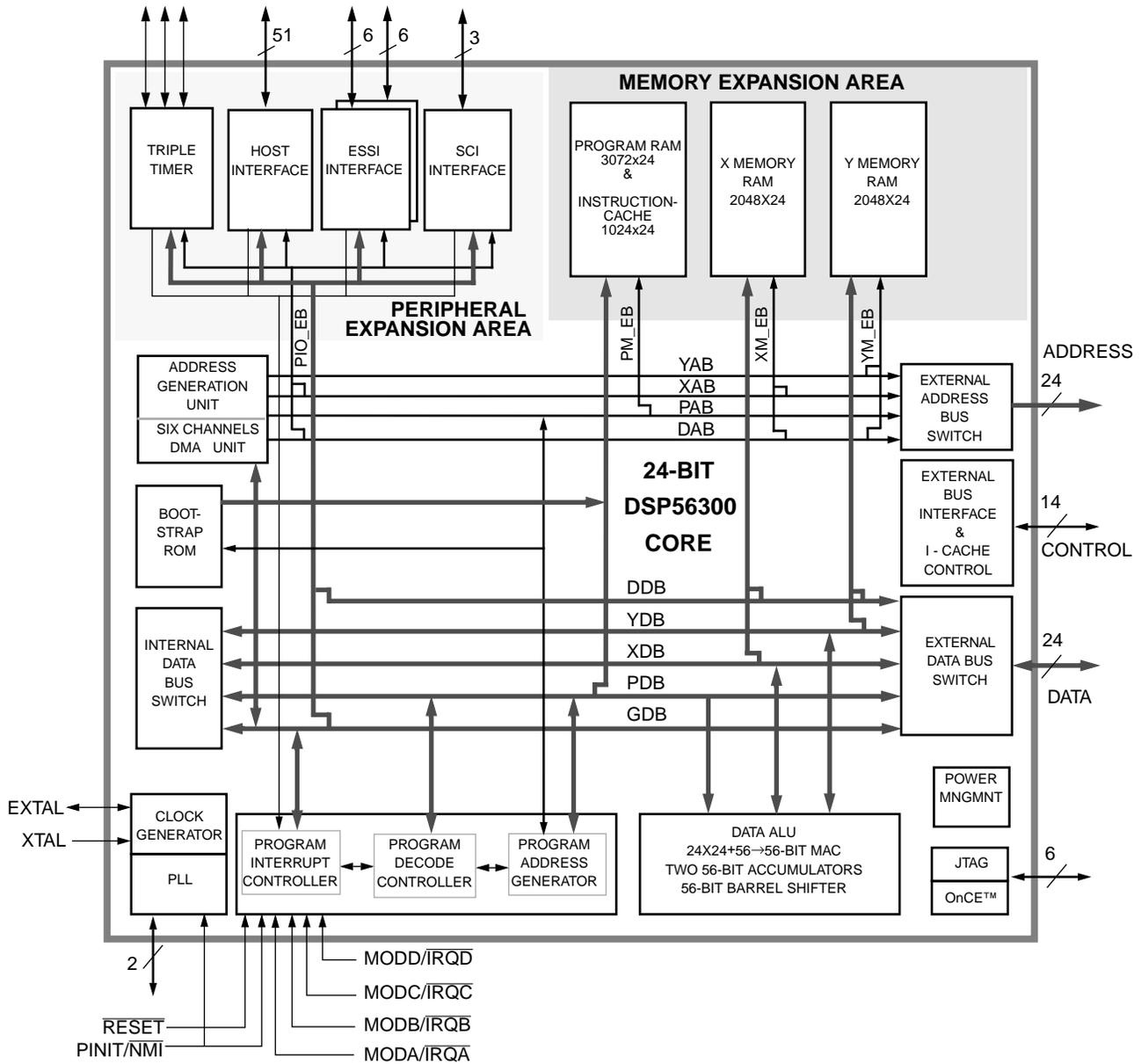
Figure 1-1. DSP56301 Block Diagram

2 PINS DESCRIPTION

2.1 PACKAGE

There are total 208 pins: 148 signal pins, 25 power pins, 26 ground pins, and 9 reserved pins.

2.2 PINOUT

The functional signal groups of the DSP56301 are shown in Figure 2-1 on page 2-8 and in Figure 2-1 on page 2-8 (the difference between the two configuration is the host port functionality) and are described in the following sections.

Although the DSP56301 is operated from a 3.3 volt supply, some of the input pins can tolerate 5 volt. A special notice for this feature is added to the description of those pins.

2.2.1 Interrupt And Mode Control (5 Pins)

$\overline{\text{RESET}}$ (Reset) - Active low, Schmitt trigger input. $\overline{\text{RESET}}$ is internally synchronized to the clock out (CLKOUT). When asserted, the chip is placed in the reset state and the internal phase generator is reset. The Schmitt trigger input allows a slowly rising input (such as a capacitor charging) to reliably reset the chip. If $\overline{\text{RESET}}$ is negated synchronous to the clock out (CLKOUT), exact start-up timing is guaranteed, allowing multiple processors to start-up synchronously and operate together in "lock-step". When the $\overline{\text{RESET}}$ pin is negated, the initial chip operating mode is latched from the MODA, MODB, MODC and MODD pins. $\overline{\text{RESET}}$ pin can tolerate 5V.

MODA/$\overline{\text{IRQA}}$ (Mode Select A/External Interrupt Request A) - Active low Schmitt trigger input, internally synchronized to the clock out (CLKOUT). MODA/$\overline{\text{IRQA}}$ selects the initial chip operating mode during hardware reset and becomes a level sensitive or negative edge triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the operating mode register (OMR) when the $\overline{\text{RESET}}$ pin is negated. If $\overline{\text{IRQA}}$ is asserted synchronous to the clock out (CLKOUT), multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQA}}$ to exit the wait state. If the processor is in the STOP standby state and $\overline{\text{IRQA}}$ is asserted, the processor will exit the STOP state.

MODA/$\overline{\text{IRQA}}$ pin can tolerate 5V.

MODB/$\overline{\text{IRQB}}$ (Mode Select B/External Interrupt Request B) - Active low Schmitt trigger input, internally synchronized to the clock out (CLKOUT). MODB/$\overline{\text{IRQB}}$ selects the initial chip operating mode during hardware reset and becomes a level sensitive or negative edge triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the operating mode register (OMR) when the $\overline{\text{RESET}}$ pin is negated. If $\overline{\text{IRQB}}$ is asserted synchronous to the clock out (CLKOUT), multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQB}}$ to exit the wait state.
MODB/$\overline{\text{IRQB}}$ pin can tolerate 5V

MODC/$\overline{\text{IRQC}}$ (Mode Select C/External Interrupt Request C) - Active low Schmitt trigger input, internally synchronized to the clock out (CLKOUT). MODC/$\overline{\text{IRQC}}$ selects the initial chip operating mode during hardware reset and becomes a level sensitive or negative edge triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the operating mode register (OMR) when the $\overline{\text{RESET}}$ pin is negated. If $\overline{\text{IRQC}}$ is asserted synchronous to the clock out (CLKOUT), multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQC}}$ to exit the wait state.
MODC/$\overline{\text{IRQC}}$ pin can tolerate 5V.

MODD/$\overline{\text{IRQD}}$ (Mode Select D/External Interrupt Request D) - Active low Schmitt trigger input, internally synchronized to the clock out (CLKOUT). MODD/$\overline{\text{IRQD}}$ selects the initial chip operating mode during hardware reset and becomes a level sensitive or negative edge triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the operating mode register (OMR) when the $\overline{\text{RESET}}$ pin is negated. If $\overline{\text{IRQD}}$ is asserted synchronous to the clock out (CLKOUT), multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQD}}$ to exit the wait state.
MODD/$\overline{\text{IRQD}}$ pin can tolerate 5V.

2.2.2   Clock (2 Pins)

EXTAL   (External Clock/Crystal Input) - Interfaces the internal crystal oscillator input to an external crystal or an external clock.

XTAL   (Crystal Output) - This output connects the internal crystal oscillator output to an external crystal. If an external clock is used, XTAL should not be connected.

2.2.3 On-chip Emulator Interface (OnCE)/JTAG Interface (6 Pins)

$\overline{\text{DE}}$ (Debug Event) - This open drain bidirectional active low pin provides, as an input, a means of entering the debug mode of operation from an external command controller, and as an output, a means of acknowledging that the chip has entered the debug mode. This pin when asserted as an input causes the DSP56300 core to finish the current instruction being executed, save the instruction pipeline information, enter the debug mode and wait for commands to be entered from the debug serial input line. This pin is asserted as an output for three clock cycles when the chip enters the debug mode as a result of a debug request or as a result of meeting a breakpoint condition.
$\overline{\text{DE}}$ pin can tolerate 5V.

TCK (Test Clock) - The test clock input TCK pin is the test clock used to synchronize the JTAG test logic
TCK pin can tolerate 5V.

TDI (Test Data Input) - The test data input TDI pin is the serial input for test instructions and data. TDI is sampled on the rising edge of TCK and it has an internal pullup resistor.
TDI pin can tolerate 5V.

TDO (Test data output) - The test data output TDO pin is the serial output for test instructions and data. TDO is three-stateable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK.

TMS (Test Mode Select) - The test mode select input (TMS) pin is used to sequence the test controller's state machine. The TMS is sampled on the rising edge of TCK and it has an internal pullup resistor
TMS pin can tolerate 5V.

$\overline{\text{TRST}}$ (Test Reset) - This active low Schmitt trigger input pin $\overline{\text{TRST}}$ is used to asynchronously initialize the test controller. The $\overline{\text{TRST}}$ has an internal pullup resistor
$\overline{\text{TRST}}$ pin can tolerate 5V.

2.2.4 Expansion Port (Port A) (63 Pins)

A0-A23 (Address Bus) - Three-state. Active high outputs when a bus master, three-stated otherwise, specify the address for external program and data memory accesses. To minimize power dissipation, A0–A23 do not change state when external memory spaces are not being accessed. A0–A23 are three-stated during hardware reset.

D0-D23 (Data Bus) - Three-state, active high, bidirectional input/outputs when a bus master. These pins provide the bidirectional data bus for external

program and data memory accesses. D0–D23 are in the high impedance state when not a bus master. They are also three-stated during hardware reset.

AA(3:0)/$\overline{RAS}$(3:0)(Address Attribute or Row Address Strobe) - Three-state outputs with a programmable polarity. When defined as Address Attribute these signals can be used as chip selects or additional address lines. When defined as $\overline{RAS}$ these signals can be used as Row Address Strobe for DRAM interface. The AA/$\overline{RAS}$ pins are three stated during hardware reset.

$\overline{RD}$ (Read Enable) - Three-state. Active low output when bus master, three-stated otherwise. $\overline{RD}$ is asserted to read external memory on the data bus (D0–D23). $\overline{RD}$ is three-stated during hardware reset.

$\overline{WR}$ (Write Enable) - Three-state. Active low output when bus master, three-stated otherwise. $\overline{WR}$ is asserted to write external memory on the data bus (D0–D23). $\overline{WR}$ is three-stated during hardware reset.

$\overline{TA}$ (Transfer Acknowledge) - Active low input. If the DSP56301 is the bus master and there is no external bus activity or the DSP56301 is not the bus master, the $\overline{TA}$ input is ignored. The $\overline{TA}$ input is a synchronous/ asynchronous (according to TAS bit in the OMR register) "DTACK" function which can extend an external bus cycle indefinitely. Any number of wait states (1, 2,..., infinity) may be added to the wait states inserted by the BCR by keeping $\overline{TA}$ negated. In typical operation, $\overline{TA}$ is negated at the start of a bus cycle, is asserted to enable completion of the bus cycle and is negated before the next bus cycle. The current bus cycle completes one clock period after $\overline{TA}$ is asserted synchronous to CLKOUT. The number of wait states is determined by the $\overline{TA}$ input or by the Bus Control Register (BCR), whichever is longer. The BCR can be used to set the minimum number of wait states in external bus cycles. If $\overline{TA}$ is tied low (asserted) and no wait states are specified in the BCR register, zero wait states will be inserted into external bus cycles.

NOTE1 In order to use the $\overline{TA}$ functionality the BCR must be programmed to at least one wait state. A zero wait state access can not be extended by $\overline{TA}$ negation, otherwise improper operation may result.

NOTE2 $\overline{TA}$ functionality may not be used while performing DRAM type accesses, Otherwise improper operation may result.

$\overline{BR}$ (Bus Request) - Active low output, never three-stated. $\overline{BR}$ is asserted when the CPU or DMA is requesting bus mastership. $\overline{BR}$ is negated when the CPU or DMA no longer needs the bus. $\overline{BR}$ may be asserted or negated independent of whether the DSP56301 is a bus master or a bus slave. Bus "parking" allows $\overline{BR}$ to be negated even though the

DSP56301 is the bus master. See the description of bus "parking" in the $\overline{BB}$ pin description. The BRH bit in the Bus Control Register allows $\overline{BR}$ to be asserted under software control even though the CPU or DMA does not need the bus. $\overline{BR}$ is typically sent to an external bus arbitrator which controls the priority, parking and tenure of each master on the same external bus. $\overline{BR}$ is only affected by CPU or DMA requests for the external bus, never for the internal bus. During hardware reset, $\overline{BR}$ is negated and the arbitration is reset to the bus slave state.

$\overline{BG}$ (Bus Grant) – Active low input. $\overline{BG}$ must be asserted/negated synchronous to the clock out (CLKOUT) for proper operation. $\overline{BG}$ is asserted by an external bus arbitration circuit when the DSP56301 may become the next bus master. When $\overline{BG}$ is asserted, the DSP56301 must wait until $\overline{BB}$ is negated before taking bus mastership. When $\overline{BG}$ is negated, bus mastership is typically given up at the end of the current bus cycle. This may occur in the middle of an instruction which requires more than one external bus cycle for execution. $\overline{BG}$ is ignored during hardware reset.

$\overline{BB}$ (Bus Busy) - Bidirectional active low input/output, must be asserted and negated synchronous to the clock out (CLKOUT). This signal indicates that the bus is active. Only after this signal is negated the pending bus master can become the bus master (and then assert it again). The bus master may keep $\overline{BB}$ asserted after ceasing bus activity regardless of whether $\overline{BR}$ is asserted or negated, this is called "bus parking" and allows the current bus master to reuse the bus without re-arbitration until other device requires the bus. The negation of $\overline{BB}$ is done by an "active pull-up" method i.e. $\overline{BB}$ is driven high and then released and held high by an external pull-up resistor. $\overline{BB}$ is an active low input during hardware reset.

NOTE: $\overline{BB}$ requires an external pullup resistor.

$\overline{BL}$ (Bus Lock) - Active low output, never three-stated. Asserted at the start of an external indivisible Read-Modify-Write (RMW) bus cycle and negated at the end of the write bus cycle. $\overline{BL}$ remains asserted between the read and write bus cycles of the RMW bus sequence. $\overline{BL}$ may be used to "resource lock" an external multi-port memory for secure semaphore updates. The only instructions which automatically assert $\overline{BL}$ are BSET, BCLR or BCHG instruction which accesses external memory. $\overline{BL}$ can also be asserted by setting the BLH bit in the BCR register. $\overline{BL}$ is negated during hardware reset.

Figure 2-1. DSP56301 Functional Signal Groups (w/PCI port)

PORT A ADDRESS BUS

A0-A23
V_{CCA} (6)
GNDA (6)

PORT A DATA BUS

D0-D23
V_{CCD} (4)
GNDD (4)

PORT A BUS CONTROL

AA0-AA3/$\overline{RAS0}$-$\overline{RAS3}$
$\overline{RD}$
$\overline{WR}$
$\overline{BS}$
$\overline{TA}$
$\overline{BR}$
$\overline{BG}$
$\overline{BB}$
$\overline{BL}$
$\overline{CAS}$
BCLK
V_{CCN} (2)
GNDN (2)
Reserved (1)

INTERRUPT AND MODE CONTROL

MODA/$\overline{IRQA}$
MODB/$\overline{IRQB}$
MODC/$\overline{IRQC}$
MODD/$\overline{IRQD}$
$\overline{RESET}$

PLL

PINIT/$\overline{NMI}$
CLKOUT
PCAP
V_{CCP}
GNDP
GND1P

SERIAL COMMUNICATION INTERFACE PORT

TXD
RXD
SCLK

POWER AND CLOCK

EXTAL
XTAL
V_{CCQ} (4)
GNDQ (4)

DSP56301 208 PINS

Port B
Port C
Port D
Port E

OnCE™ ON-CHIP EMULATION/ JTAG PORT

TDI
TCK
TDO
$\overline{DE}$
$\overline{TRST}$
TMS

PCI PORT

HAD(31:0)
HC(3:0)/$\overline{HBE}$(3:0)
$\overline{HTRDY}$
$\overline{HIRDY}$
$\overline{HDEVSEL}$
$\overline{HLOCK}$
HPAR
$\overline{HPERR}$
$\overline{HGNT}$
$\overline{HREQ}$
$\overline{HSERR}$
$\overline{HSTOP}$
HIDSEL
$\overline{HFRAME}$
HCLK
$\overline{HRST}$
$\overline{HINTA}$
V_{CC}/NC
V_{CCH} (6)
GNDH (6)

Port B signals:
HP(48:33,15:0)
HP(19:16)
HP20
HP21
HP22
HP23
HP24
HP25
HP26
HP27
HP28
HP29
HP30
HP31
HP32
HP49
HP50
PVCL

SYNCHRONOUS SERIAL PORT 0

STD0
SRD0
SCK0
SC00-SC02
V_{CCS} (2)
GNDS (2)

SYNCHRONOUS SERIAL PORT 1

STD1
SRD1
SCK1
SC10-SC12

TIMER 0
TIO0

TIMER 1
TIO1

TIMER 2
TIO2

• OnCE™ is a trademark of Motorola Inc.

Figure 2-2. DSP56301 Functional Signal Groups (w/universal bus port)

PORT A ADDRESS BUS

A0-A23
V_{CCA} (6)
GNDA (6)

PORT A DATA BUS

D0-D23
V_{CCD} (4)
GNDD (4)

PORT A BUS CONTROL

AA0-AA3/$\overline{RAS0}$-$\overline{RAS3}$
$\overline{RD}$
$\overline{WR}$
$\overline{BS}$
$\overline{TA}$
$\overline{BR}$
$\overline{BG}$
$\overline{BB}$
$\overline{BL}$
$\overline{CAS}$
BCLK
V_{CCN} (2)
GNDN (2)
Reserved (1)

INTERRUPT AND MODE CONTROL

MODA/$\overline{IRQA}$
MODB/$\overline{IRQB}$
MODC/$\overline{IRQC}$
MODD/$\overline{IRQD}$
$\overline{RESET}$

PLL

PINIT/$\overline{NMI}$
CLKOUT
PCAP
V_{CCP}
GNDP
GND1P

SERIAL COMMUNICATION INTERFACE PORT

TXD
RXD
SCLK

POWER AND CLOCK

EXTAL
XTAL
V_{CCQ} (4)
GNDQ (4)

DSP56301 208 PINS

Port B
Port C
Port D
Port E

OnCE™ ON-CHIP EMULATION/ JTAG PORT

TDI
TCK
TDO
$\overline{DE}$
$\overline{TRST}$
TMS

Port B

HP(48:33,15:8)
HP(7:0,18:16)
HP19
HP20
HP21
HP22
HP23
HP24
HP25
HP26
HP27
HP28
HP29
HP30
HP31
HP32
HP49
HP50
PVCL

HOST PORT

HD(23:0)
HA (10:0)
pullup/pulldown
$\overline{HDBEN}$
HDBDR
$\overline{HSAK}$
$\overline{HBS}$
$\overline{HDAK}$
HDRQ
HAEN
HTA
$\overline{HIRQ}$
$\overline{HWR}$/HRW
$\overline{HRD}$/$\overline{HDS}$
pullup
pullup
$\overline{HRST}$
$\overline{HINTA}$
NC
V_{CCH} (6)
GNDH (6)

SYNCHRONOUS SERIAL PORT 0

STD0
SRD0
SCK0
SC00-SC02
V_{CCS} (2)
GNDS (2)

SYNCHRONOUS SERIAL PORT 1

STD1
SRD1
SCK1
SC10-SC12

TIMER 0

TIO0

TIMER 1

TIO1

TIMER 2

TIO2

- OnCE™ is a trademark of Motorola Inc.

$\overline{BS}$ (Bus Strobe) - Three-state. Active low output when a bus master, three-stated otherwise. Asserted at the start of a bus cycle (for half of a clock cycle) providing an "early bus start" signal for a bus controller. If the external bus is not used during an instruction cycle $\overline{BS}$ remains negated until the next external bus cycle. $\overline{BS}$ is three-stated during hardware reset.

$\overline{CAS}$ (Column Address Strobe) - Active low output when bus master and three stated otherwise (If BME bit in the DRAM Control Register is cleared), $\overline{CAS}$ is used by DRAM to strobe column address. $\overline{CAS}$ is three stated during hardware reset.

BCLK (Bus Clock) - three-state. Active high output when a bus master three stated otherwise. BCLK is used by synchronous SRAM to sample address, data and control signals. BCLK is active only during SSRAM accesses. When active BCLK is synchronized to CLKOUT by the internal Phase Lock Loop, BCLK precedes CLKOUT by 1/4 of a clock cycle. BCLK is three stated during hardware reset

RESERVED (1) There is one reserved pin for use in the expansion port interface and in the peripherals interface. This pin should be left unconnected.

2.2.5 Host Interface (HI32) (52 Pins)

The Host Interface (HI32) provides a fast parallel data port up to 32 bits wide, which may be connected directly to the host bus.
The HI32 supports a variety of standard buses, and provides glue-less connection with the emerging PCI bus standard and with a number of industry standard microcomputers, microprocessors, DSPs and DMA hardware.
The complete HI32 pin functionality is described in the HI32 port chapter.

HP0-HP7 When the HI32 is programmed to interface the PCI bus these pins are lines 0-7 of the Address/Data multiplexed bidirectional three-state bus (HAD0-HAD7).
When the HI32 is programmed to interface any other host bus these pins are lines 3-10 of the input address bus (HA3-HA10).
HP0-HP7 may be programmed as general purpose I/O pins called HIO0-HIO7.
HP0-HP7 can tolerate 5V.

HP8-HP15 When the HI32 is programmed to interface the PCI bus these pins are lines 8-15 of the Address/Data multiplexed bidirectional three-state bus (HAD8-HAD15).
When the HI32 is programmed to interface any other host bus these pins are lines 0-7 of the bidirectional three-state Data bus (HD0-HD7).
HP8-HP15 may be programmed as general purpose I/O pins called HIO8-HIO15.

HP8-HP15 can tolerate 5V.

HP16-HP19　　　When the HI32 is programmed to interface the PCI bus these pins are the Command/Byte enable bidirectional three-state bus (HC0/$\overline{\text{HBE0}}$ - HC3/$\overline{\text{HBE3}}$).
When the HI32 is programmed to interface any other host bus HP16-HP18 pins are lines 0-2 of the Address input bus (HA0-HA2).
HP16-HP19 may be programmed as general purpose I/O pins called HIO16-HIO19.
HP16-HP19 can tolerate 5V.

HP20　　　When the HI32 is programmed to interface the PCI bus this pin is the Target Ready bidirectional three-state pin ($\overline{\text{HTRDY}}$).
When the HI32 is programmed to interface any other host bus this pin is the Host Data Bus Enable output pin (HDBEN).
HP20 may be programmed as general purpose I/O pin called HIO20.
HP20 can tolerate 5V.

HP21　　　When the HI32 is programmed to interface the PCI bus this pin is the Initiator Ready bidirectional three-state pin ($\overline{\text{HIRDY}}$).
When the HI32 is programmed to interface any other host bus this pin is the Host Data Bus Direction output pin (HDBDR).
HP21 may be programmed as general purpose I/O pin called HIO21.
HP21 can tolerate 5V.

HP22　　　When the HI32 is programmed to interface the PCI bus this pin is the Device Select bidirectional three-state pin ($\overline{\text{HDEVSEL}}$).
When the HI32 is programmed to interface any other host bus this pin is the Host Select Acknowledge output pin ($\overline{\text{HSAK}}$).
HP22 may be programmed as general purpose I/O pin called HIO22.
HP22 can tolerate 5V.

HP23　　　When the HI32 is programmed to interface the PCI bus this pin is the Lock bidirectional three-state pin ($\overline{\text{HLOCK}}$).
When the HI32 is programmed to interface any other host bus this pin is the Host Bus Strobe Schmitt trigger input pin ($\overline{\text{HBS}}$).
HP23 may be programmed as general purpose I/O pin called HIO23.
HP23 can tolerate 5V.

HP24　　　When the HI32 is programmed to interface the PCI bus this pin is the Parity bidirectional three-state pin (HPAR).
When the HI32 is programmed to interface any other host bus this pin is the Host DMA Acknowledge Schmitt trigger input pin ($\overline{\text{HDAK}}$).
HP24 can tolerate 5V.

HP25　　　When the HI32 is programmed to interface the PCI bus this pin is the Parity Error bidirectional three-state pin ($\overline{\text{HPERR}}$).

When the HI32 is programmed to interface any other host bus this pin is the Host DMA Request output pin (HDRQ).
HP25 can tolerate 5V.

HP26
When the HI32 is programmed to interface the PCI bus this pin is the Bus Grant input pin ($\overline{\text{HGNT}}$).
When the HI32 is programmed to interface any other host bus this pin is the Host Address Enable input pin (HAEN).
HP26 can tolerate 5V.

HP27
When the HI32 is programmed to interface the PCI bus this pin is the Bus Request three-stated output pin ($\overline{\text{HREQ}}$).
When the HI32 is programmed to interface any other host bus this pin is the Host Transfer Acknowledge three stated output pin (HTA).

HP28
When the HI32 is programmed to interface the PCI bus this pin is the System Error open drain output pin ($\overline{\text{HSERR}}$).
When the HI32 is programmed to interface any other host bus this pin is the Host Interrupt Request programmable open drain output pin ($\overline{\text{HIRQ}}$).
HP28 can tolerate 5V.

HP29
When the HI32 is programmed to interface the PCI bus this pin is the Stop bidirectional three-state pin ($\overline{\text{HSTOP}}$).
When the HI32 is programmed to interface any other host bus this pin is the Host Write/Read-Write Schmitt trigger input pin ($\overline{\text{HWR}}$/HRW).
HP29 can tolerate 5V.

HP30
When the HI32 is programmed to interface the PCI bus this pin is the Initialization Device Select input pin (HIDSEL).
When the HI32 is programmed to interface any other host bus this pin is the Host Read/Data Strobe Schmitt trigger input pin ($\overline{\text{HRD}}$/$\overline{\text{HDS}}$).
HP30 can tolerate 5V.

HP31
When the HI32 is programmed to interface the PCI bus this pin is the cycle Frame bidirectional three-state pin ($\overline{\text{HFRAME}}$).
When the HI32 is programmed to interface any other host bus this pin must be tied to V_{CC} or pullup resistor.
HP31 can tolerate 5V.

HP32
When the HI32 is programmed to interface the PCI bus this pin is the Bus Clock input pin (HCLK).
When the HI32 is programmed to interface any other host bus this pin must be tied to V_{CC} or pullup resistor.
HP32 can tolerate 5V.

HP33-HP48
When the HI32 is programmed to interface the PCI bus these pins are lines 16-31 of the Address/Data Multiplexed bidirectional three-state

Bus (HAD16 - HAD31).
When the HI32 is programmed to interface any other host bus these pins are lines 8-23 of the bidirectional three-state Host Data bus (HD8-HD23).
HP33 - HP48 can tolerate 5V.

HP49   When the HI32 is programmed to interface the PCI bus this pin is the Hardware Reset input pin ($\overline{\text{HRST}}$).
When the HI32 is programmed to interface any other host bus this pin is Hardware Reset Schmitt trigger input pin ($\overline{\text{HRST}}$).
HP49 can tolerate 5V.

HP50   When the HI32 is programmed to interface the PCI bus this pin is the interrupt A open drain output pin ($\overline{\text{HINTA}}$)
In all other modes this pin is an open drain output pin ($\overline{\text{HINTA}}$).
HP50 can tolerate 5V.

PVCL   When the HI32 is connected to a PCI bus that operates in a 3V signaling environment this pin should be connected to V_{CC} (3.3V) in order enable the high voltage clamping required by the PCI specifications. In other cases (including PCI bus with a 5V signaling environment) this pin should be left unconnected.

2.2.6   Enhanced Synchronous Serial Interface 0 (ESSI 0) (6 Pins)

There are two synchronous serial interfaces (ESSI 0 and ESSI 1) that provide a full-duplex serial port for serial communication with a variety of serial devices including one or more industry-standard codecs, other DSPs, microprocessors, and peripherals which implement the Motorola SPI.

STD0   (Serial Transmit Data Pin) - is used for transmitting data from the serial transmit shift register. STD0 is an output when data is being transmitted. STD0 may be programmed as a general purpose I/O pin called PC5 when the ESSI0 STD function is not being used.
STD0 pin can tolerate 5V.

SRD0   (Serial Receive Data Pin) - receives serial data and transfers the data to the ESSI receive shift register. SRD0 is input when data is being received. SRD0 may be programmed as a general-purpose I/O pin called PC4 when the ESSI0 SRD function is not being used.
SRD0 pin can tolerate 5V.

SCK0   (Serial Clock) - is a bidirectional (Schmitt trigger input) pin providing the serial bit rate clock for the ESSI interface. The SCK0 is a clock input or output used by both the transmitter and receiver in synchronous modes or by the transmitter in asynchronous modes. SCK0 may be programmed as a general purpose I/O pin called PC3 when the ESSI0

SCK function is not being used.
SCK0 pin can tolerate 5V.

NOTE: Although an external serial clock can be independent of and asynchronous to the DSP system clock, it must exceed the minimum clock cycle time of 6T (i.e., the system clock frequency must be at least three times the external ESSI clock frequency). The ESSI needs at least three DSP phases (DSP phase = T) inside each half of the serial clock.

SC00 (Serial Control Pin 0) - The function of this pin is determined by the selection of either synchronous or asynchronous mode. For asynchronous mode, this pin will be used for the receive clock I/O (Schmitt trigger input). For synchronous mode, this pin is used either for transmitter1 output or for serial I/O flag 0. SC00 may be programmed as a general purpose I/O pin called PC0 when the ESSI0 SC0 function is not being used.
SC00 pin can tolerate 5V.

SC01 (Serial Control Pin 1) - The function of this pin is determined by the selection of either synchronous or asynchronous mode. For asynchronous mode, this pin is the receiver frame sync I/O. For synchronous mode, this pin is used either for transmitter2 output or for serial I/O flag 1. SC01 may be programmed as a general purpose I/O pin called PC1 when the ESSI0 SC1 function is not being used.
SC01 pin can tolerate 5V.

SC02 (Serial Control Pin 2) - This pin is used for frame sync I/O. SC02 is the frame sync for both the transmitter and receiver in synchronous mode and for the transmitter only in asynchronous mode. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitter (and the receiver in synchronous operation). SC02 may be programmed as a general purpose I/O pin called PC2 when the ESSI0 SC2 function is not being used.
SC02 pin can tolerate 5V.

2.2.7   Enhanced Synchronous Serial Interface 1 (ESSI 1) (6 Pins)

ESSI1 is identical to ESSI0.

STD1 (Serial Transmit Data Pin) - is used for transmitting data from the serial transmit shift register. STD1 is an output when data is being transmitted. STD1 may be programmed as a general purpose I/O pin called PD5 when the ESSI1 STD function is not being used.
STD1 pin can tolerate 5V.

SRD1 (Serial Receive Data Pin) - receives serial data and transfers the data to

the ESSI receive shift register. SRD1 may be programmed as a general-purpose I/O pin called PD4 when the ESSI1 SRD function is not being used.
SRD1 pin can tolerate 5V.

SCK1　　　　(Serial Clock) - is a bidirectional (Schmitt trigger input) pin providing the serial bit rate clock for the ESSI interface. The SCK1 is a clock input or output used by both the transmitter and receiver in synchronous modes or by the transmitter in asynchronous modes. SCK1 may be programmed as a general purpose I/O pin called PD3 when the ESSI1 SCK function is not being used.
SCK1 pin can tolerate 5V.

NOTE:　　Although an external serial clock can be independent of and asynchronous to the DSP system clock, it must exceed the minimum clock cycle time of 6T (i.e., the system clock frequency must be at least three times the external ESSI clock frequency). The ESSI needs at least three DSP phases (DSP phase = T) inside each half of the serial clock.

SC10　　　　(Serial Control Pin 0) - The function of this pin is determined by the selection of either synchronous or asynchronous mode. For asynchronous mode, this pin will be used for the receive clock I/O (Schmitt trigger input). For synchronous mode, this pin is used either for transmitter 1 output or for serial I/O flag 0. SC10 may be programmed as a general purpose I/O pin called PD0 when the ESSI1 SC0 function is not being used.
SC10 pin can tolerate 5V.

SC11　　　　(Serial Control Pin 1) - The function of this pin is determined by the selection of either synchronous or asynchronous mode. For asynchronous mode, this pin is the receiver frame sync I/O. For synchronous mode, this pin is used either for transmitter2 output or for serial I/O flag 1. SC11 may be programmed as a general purpose I/O pin called PD1 when the ESSI1 SC1 function is not being used.
SC11 pin can tolerate 5V.

SC12　　　　(Serial Control Pin 2) - This pin is used for frame sync I/O. SC12 is the frame sync for both the transmitter and receiver in synchronous mode and for the transmitter only in asynchronous mode. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitter (and the receiver in synchronous operation). SC12 may be programmed as a general purpose I/O pin called PD2 when the ESSI1 SC2 function is not being used.
SC12 pin can tolerate 5V.

2.2.8   Serial Communication Interface (SCI) (3 Pins)

The Serial Communication interface (SCI) provides a full duplex port for serial communication to other DSPs, microprocessors, or peripherals such as modems.

TXD   (Serial Transmit Data Pin) - This output transmits data from SCI transmit data register. TXD can be programmed as general purpose I/O pin (PE1) when SCI TXD function is not being used
                       TXD pin can tolerate 5V.

RXD   (Serial Receive Data Pin) - This input receives byte oriented serial data and transfers the data to the SCI receive shift register. RXD can be programmed as general purpose I/O pin (PE0) when the SCI RXD function is not being used.
                       RXD pin can tolerate 5V.

SCLK   (Serial Clock) - This bidirectional (Schmitt trigger input) pin providing the input or output clock used by the transmitter and/or the receiver. SCLK may be programmed as a general purpose I/O pin (PE2) when the SCI SCLK function is not being used.
                       SCLK pin can tolerate 5V.

2.2.9   Triple Timer (3 Pins)

Three identical and independent Timers are implemented in the DSP56301. Each timer can use internal or external clocking and can interrupt the DSP56301 after a specified number of events (clocks) or can signal an external device after counting a specific number of internal events. Each timer connects to the external world through one bidirectional pin TIO.

TIO0   (Timer-0 Schmitt trigger input/output pin) - When TIO0 is used as input, the module is functioning as an external event counter or measures external pulse width or signal period. When TIO0 is used as output the module is functioning as timer and the TIO0 becomes timer pulse. When the TIO0 pin is not used by the timer module, it can be used as a general purpose Input/Output Pin.
                       TIO0 pin can tolerate 5V.

TIO1   (Timer-1 Schmitt trigger Input/output pin) - When TIO1 is used as input, the module is functioning as an external event counter or measures external pulse width or signal period. When TIO1 is used as output the module is functioning as timer and the TIO1 becomes timer pulse. When the TIO1 pin is not used by the timer module, it can be used as a general purpose Input/Output Pin.
                       TIO1 pin can tolerate 5V.

TIO2   (Timer-2 Schmitt trigger Input/output pin) - When TIO2 is used as input, the module is functioning as an external event counter or measures

external pulse width or signal period. When TIO2 is used as output the module is functioning as timer and the TIO2 becomes timer pulse. When the TIO2 pin is not used by the timer module, it can be used as a general purpose Input/Output Pin.
TIO2 pin can tolerate 5V.

2.2.10   Phase-Locked Loop (PLL) (3 Pins)

PCAP   (PLL capacitor) - This input connects the off-chip capacitor for PLL filter. One terminal of the capacitor is connected to PCAP while the other terminal is connected to V_{CCP}

CLKOUT   (Clock Output) - This output pin provides an output clock synchronized to the internal core clock phase.

NOTE 1:   If PLL is enabled and both the multiplication and division factors are equal to one, then CLKOUT is also synchronized to EXTAL.

NOTE 2:   If PLL is disabled, CLKOUT frequency is half the frequency of EXTAL.

PINIT/$\overline{\text{NMI}}$ (PLL Initial/Non Maskable Interrupt) - During the assertion of hardware reset, PINIT/$\overline{\text{NMI}}$ is configured as PINIT and its value is written into the PEN bit of the PLL control register and determines whether the PLL is enabled or disabled. After hardware reset negation and during normal instruction processing, the PINIT/$\overline{\text{NMI}}$ Schmitt trigger pin is configured as $\overline{\text{NMI}}$, a negative edge triggered, non maskable interrupt request, internally synchronized to the clock out (CLKOUT).
PINIT/$\overline{\text{NMI}}$ pin can tolerate 5V.

2.2.11   Power & Ground (51 Pins)

V_{CCQ} (4)   (Quiet Power) - isolated power for the CPU logic. Must be tied to all other chip power pins externally. User must provide adequate external decoupling capacitors.

GNDQ (4)   (Quiet Ground) - isolated ground for the CPU logic. Must be tied to all other chip ground pins externally. User must provide adequate external decoupling capacitors.

V_{CCA} (6)   (Address Bus Power) - isolated power for sections of address bus I/O drivers. Must be tied to all other chip power pins externally. User must provide adequate external decoupling capacitors.

GNDA (6)   (Address Bus Ground) - isolated ground for sections of address bus I/O drivers. Must be tied to all other chip ground pins externally. User must provide adequate external decoupling capacitors.

V_{CCD} (4)   (Data Bus Power) - isolated power for sections of data bus I/O drivers.

Must be tied to all other chip power pins externally. User must provide adequate external decoupling capacitors.

GNDD (4)   (Data Bus Ground) - isolated ground for sections of data bus I/O drivers. Must be tied to all other chip ground pins externally. User must provide adequate external decoupling capacitors.

V_{CCN} (2)   (Bus Control Power) - isolated power for the bus control I/O drivers. Must be tied to all other chip power pins externally. User must provide adequate external decoupling capacitors.

GNDN (2)   (Bus Control Ground) - isolated ground for the bus control I/O drivers. Must be tied to all other chip ground pins externally. User must provide adequate external decoupling capacitors.

V_{CCH} (6)   (Host Power) - isolated power for the HI logic. Must be tied to all other chip power pins externally. User must provide adequate external decoupling capacitors.

GNDH (6)   (Host Ground) - isolated ground for the HI logic. Must be tied to all other chip ground pins externally. User must provide adequate external decoupling capacitors.

V_{CCS} (2)   (ESSIs, SCI and Timers Power) - isolated power for the ESSIs and Timers logic. Must be tied to all other chip power pins externally. User must provide adequate external decoupling capacitors.

GNDS (2)   (ESSIs, SCI and Timers Ground) - isolated ground for the ESSIs and Timers logic. Must be tied to all other chip ground pins externally. User must provide adequate external decoupling capacitors.

V_{CCP} (PLL Power) - V_{CC} dedicated for PLL use. The voltage should be well regulated and the pin should be provided with an extremely low impedance path to the V_{CC} power rail.

GNDP   (PLL Ground) - Gnd dedicated for PLL use. The pin should be provided with en extremely low impedance path to ground. V_{CCP} should be bypassed to GNDP by a 0.1μF capacitor located as close as possible to the chip package.

GND1P   (PLL Ground 1) - Gnd dedicated for PLL use. The pin should be provided with en extremely low impedance path to ground.

3   MEMORY MAPS

3.1   INTRODUCTION

The memory space of the DSP56301 is partitioned into program memory space (P), X data memory space and Y data memory space. The program memory space (P) includes internal PRAM, internal Instruction Cache (that behaves as a PRAM when the cache is disabled), a boot program ROM and an optional off-chip memory expansion. The data memory space is divided into X data memory and Y data memory in order to work with the two address arithmetic logic units (ALUs) and to feed two operands simultaneously to the data ALU. Each data memory space includes internal RAM and an optional off-chip memory expansion.

3.2   DSP56301 MEMORY MAPS

The three independent memory spaces of the DSP56301: X data, Y data, and program, are shown in Figure 3-1.

Figure 3-1. DSP56301 Memory Map
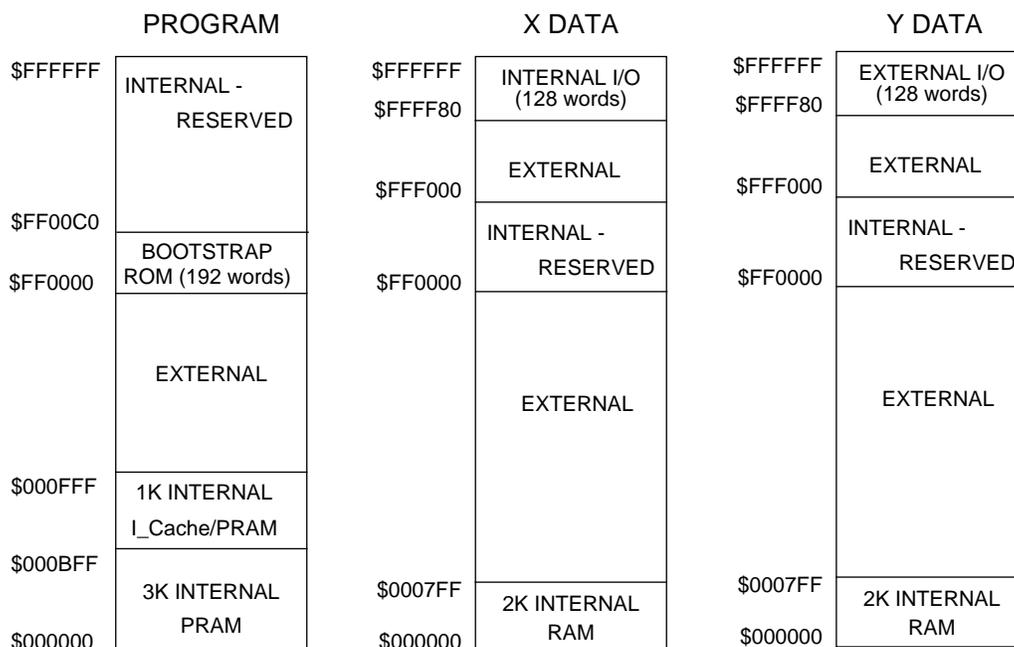
NOTE   If CE bit in SR is set, program memory space $000C00-$000FFF becomes an instruction cache (I-Cache).

3.2.1   On-chip X Data Memory

The on-chip X data RAM is a 24-bit-wide, internal, static memory occupying the lowest 2048 locations (0–2047) in X memory space. The on-chip X data RAM is organized in eight banks, 256 locations each.

The on-chip peripheral registers, and some of the DSP56301 core registers, occupy the top 128 locations of the X data memory ($FFFF80–$FFFFFF). This area is referred to as X-I/O space and it can be accessed by MOVE, MOVEP instructions and by bit oriented instructions (BCHG, BCLR, BSET, BTST, BRCLR, BRSET, BSCLR, BSSET, JCLR, JSET, JSCLR and JSSET).

The X memory space located at locations $FF0000-$FFEFFF is reserved and should not be accessed.

3.2.2   On-chip Y Data Memory

The on-chip Y data RAM is a 24-bit-wide, internal, static memory occupying the lowest 2048 locations (0–2047) in the Y memory space. The on-chip Y data RAM is organized in eight banks, 256 locations each.

The off-chip peripheral registers should be mapped into the top 128 locations ($FFFF80–$FFFFFF) to take advantage of the move peripheral data (MOVEP) instruction and the bit oriented instructions (BCHG, BCLR, BSET, BTST, BRCLR, BRSET, BSCLR, BSSET, JCLR, JSET, JSCLR and JSSET).

The Y memory space located at locations $FF0000-$FFEFFF is reserved and should not be accessed.

3.2.3   On-chip Program Memory

The on-chip program memory consists of a 24-bit-wide, high-speed, static RAM occupying the lowest 4096 locations (0–4095) in the P memory space. The on-chip P data RAM is organized in 16 banks, 256 locations each. Of these on-chip 4096 program words, the upper 1024 program words located at locations $000C00-$000FFF, can be used as an internal Instruction Cache when setting the CE bit in the SR.

The P memory space occupying locations $FF0000-$FF00BF includes the internal Bootstrap ROM. This ROM contains 192 words combining the bootstrap program for the DSP56301.

The P memory space located at locations $FF00C0-$FFFFFF is reserved and should not be accessed.

3.3   DSP56301 INTERNAL I/O MEMORY MAP

The mapping of the DSP56301 internal X-I/O space (the top 128 locations of the X data memory space) appears in the following table:

Table 3-1. Internal I/O Memory Map

Peripheral	Address	Register Name
IPR	$FFFFFF	INTERRUPT PRIORITY REGISTER CORE (IPR-C)
	$FFFFFE	INTERRUPT PRIORITY REGISTER PERIPHERAL (IPR-P)
PLL	$FFFFFD	PLL CONTROL REGISTER (PCTL)
ONCE	$FFFFFC	ONCE GDB REGISTER (OGDB)

Peripheral	Address	Register Name
BIU	$FFFFFB	BUS CONTROL REGISTER (BCR)
	$FFFFFA	DRAM CONTROL REGISTER (DCR)
	$FFFFF9	ADDRESS ATTRIBUTE REGISTER 0 (AAR0)
	$FFFFF8	ADDRESS ATTRIBUTE REGISTER 1 (AAR1)
	$FFFFF7	ADDRESS ATTRIBUTE REGISTER 2 (AAR2)
	$FFFFF6	ADDRESS ATTRIBUTE REGISTER 3 (AAR3)
	$FFFFF5	ID REGISTER (IDR)
DMA	$FFFFF4	DMA STATUS REGISTER (DSTR)
	$FFFFF3	DMA OFFSET REGISTER 0 (DOR0)
	$FFFFF2	DMA OFFSET REGISTER 1 (DOR1)
	$FFFFF1	DMA OFFSET REGISTER 2 (DOR2)
	$FFFFF0	DMA OFFSET REGISTER 3 (DOR3)
DMA0	$FFFFEF	DMA SOURCE ADDRESS REGISTER (DSR0)
	$FFFFEE	DMA DESTINATION ADDRESS REGISTER (DDR0)
	$FFFFED	DMA COUNTER (DCO0)
	$FFFFEC	DMA CONTROL REGISTER (DCR0)
DMA1	$FFFFEB	DMA SOURCE ADDRESS REGISTER (DSR1)
	$FFFFEA	DMA DESTINATION ADDRESS REGISTER (DDR1)
	$FFFFE9	DMA COUNTER (DCO1)
	$FFFFE8	DMA CONTROL REGISTER (DCR1)
DMA2	$FFFFE7	DMA SOURCE ADDRESS REGISTER (DSR2)
	$FFFFE6	DMA DESTINATION ADDRESS REGISTER (DDR2)
	$FFFFE5	DMA COUNTER (DCO2)
	$FFFFE4	DMA CONTROL REGISTER (DCR2)

Peripheral	Address	Register Name
DMA3	$FFFFE3	DMA SOURCE ADDRESS REGISTER (DSR3)
	$FFFFE2	DMA DESTINATION ADDRESS REGISTER (DDR3)
	$FFFFE1	DMA COUNTER (DCO3)
	$FFFFE0	DMA CONTROL REGISTER (DCR3)
DMA4	$FFFFDF	DMA SOURCE ADDRESS REGISTER (DSR4)
	$FFFFDE	DMA DESTINATION ADDRESS REGISTER (DDR4)
	$FFFFDD	DMA COUNTER (DCO4)
	$FFFFDC	DMA CONTROL REGISTER (DCR4)
DMA5	$FFFFDB	DMA SOURCE ADDRESS REGISTER (DSR5)
	$FFFFDA	DMA DESTINATION ADDRESS REGISTER (DDR5)
	$FFFFD9	DMA COUNTER (DCO5)
	$FFFFD8	DMA CONTROL REGISTER (DCR5)
	$FFFFD7	RESERVED
	$FFFFD6	RESERVED
	$FFFFD5	RESERVED
	$FFFFD4	RESERVED
	$FFFFD3	RESERVED
	$FFFFD2	RESERVED
	$FFFFD1	RESERVED
	$FFFFD0	RESERVED
PORT B	$FFFFCF	HOST PORT GPIO DATA REGISTER (DATH)
	$FFFFCE	HOST PORT GPIO DIRECTION REGISTER (DIRH)

Peripheral	Address	Register Name
HI32	$FFFFCD	DSP SLAVE TRANSMIT DATA FIFO (DTXS)
	$FFFFCC	DSP MASTER TRANSMIT DATA FIFO (DTXM)
	$FFFFCB	DSP RECEIVE DATA FIFO (DRXR)
	$FFFFCA	DSP PCI STATUS REGISTER (DPSR)
	$FFFFC9	DSP STATUS REGISTER (DSR)
	$FFFFC8	DSP PCI ADDRESS REGISTER (DPAR)
	$FFFFC7	DSP PCI MASTER CONTROL REGISTER (DPMC)
	$FFFFC6	DSP PCI CONTROL REGISTER (DPCR)
	$FFFFC5	DSP CONTROL REGISTER (DCTR)
	$FFFFC4	RESERVED
	$FFFFC3	RESERVED
	$FFFFC2	RESERVED
	$FFFFC1	RESERVED
	$FFFFC0	RESERVED
PORT C	$FFFFBF	PORT C CONTROL REGISTER (PCRC)
	$FFFFBE	PORT C DIRECTION REGISTER (PRRC)
	$FFFFBD	PORT C GPIO DATA REGISTER (PDRC)

Peripheral	Address	Register Name
ESSI 0	$FFFFBC	ESSI 0 TRANSMIT DATA REGISTER 0 (TX00)
	$FFFFBB	ESSI 0 TRANSMIT DATA REGISTER 1 (TX01)
	$FFFFBA	ESSI 0 TRANSMIT DATA REGISTER 2 (TX02)
	$FFFFB9	ESSI 0 TIME SLOT REGISTER (TSR0)
	$FFFFB8	ESSI 0 RECEIVE DATA REGISTER (RX0)
	$FFFFB7	ESSI 0 STATUS REGISTER (SSISR0)
	$FFFFB6	ESSI 0 CONTROL REGISTER B (CRB0)
	$FFFFB5	ESSI 0 CONTROL REGISTER A (CRA0)
	$FFFFB4	ESSI 0 TRANSMIT SLOT MASK REGISTER A (TSMA0)
	$FFFFB3	ESSI 0 TRANSMIT SLOT MASK REGISTER B (TSMB0)
	$FFFFB2	ESSI 0 RECEIVE SLOT MASK REGISTER A (RSMA0)
	$FFFFB1	ESSI 0 RECEIVE SLOT MASK REGISTER B (RSMB0)
	$FFFFB0	RESERVED
PORT D	$FFFFAF	PORT D CONTROL REGISTER (PCRD)
	$FFFFAE	PORT D DIRECTION REGISTER (PRRD)
	$FFFFAD	PORT C GPIO DATA REGISTER (PDRD)

Peripheral	Address	Register Name
ESSI 1	$FFFFAC	ESSI 1 TRANSMIT DATA REGISTER 0 (TX10)
	$FFFFAB	ESSI 1 TRANSMIT DATA REGISTER 1 (TX11)
	$FFFFAA	ESSI 1 TRANSMIT DATA REGISTER 2 (TX12)
	$FFFFA9	ESSI 1 TIME SLOT REGISTER (TSR1)
	$FFFFA8	ESSI 1 RECEIVE DATA REGISTER (RX1)
	$FFFFA7	ESSI 1 STATUS REGISTER (SSISR1)
	$FFFFA6	ESSI 1 CONTROL REGISTER B (CRB1)
	$FFFFA5	ESSI 1 CONTROL REGISTER A (CRA1)
	$FFFFA4	ESSI 1 TRANSMIT SLOT MASK REGISTER A (TSMA1)
	$FFFFA3	ESSI 1 TRANSMIT SLOT MASK REGISTER B (TSMB1)
	$FFFFA2	ESSI 1 RECEIVE SLOT MASK REGISTER A (RSMA1)
	$FFFFA1	ESSI 1 RECEIVE SLOT MASK REGISTER B (RSMB1)
	$FFFFA0	RESERVED
PORT E	$FFFF9F	PORT E CONTROL REGISTER (PCRE)
	$FFFF9E	PORT E DIRECTION REGISTER (PRRE)
	$FFFF9D	PORT E GPIO DATA REGISTER (PDRE)
SCI	$FFFF9C	SCI CONTROL REGISTER (SCR)
	$FFFF9B	SCI CLOCK CONTROL REGISTER (SCCR)
	$FFFF9A	SCI RECEIVE DATA REGISTER - HIGH (SRXH)
	$FFFF99	SCI RECEIVE DATA REGISTER - MIDDLE (SRXM)
	$FFFF98	SCI RECIEVE DATA REGISTER - LOW (SRXL)
	$FFFF97	SCI TRANSMIT DATA REGISTER - HIGH (STXH)
	$FFFF96	SCI TRANSMIT DATA REGISTER - MIDDLE (STXM)
	$FFFF95	SCI TRANSMIT DATA REGISTER - LOW (STXL)
	$FFFF94	SCI TRANSMIT ADDRESS REGISTER (STXA)
	$FFFF93	SCI STATUS REGISTER (SSR)

Peripheral	Address	Register Name
	$FFFF92	RESERVED
	$FFFF91	RESERVED
	$FFFF90	RESERVED
TRIPLE TIMER	$FFFF8F	TIMER 0 CONTROL/STATUS REGISTER (TCSR0)
	$FFFF8E	TIMER 0 LOAD REGISTER (TLR0)
	$FFFF8D	TIMER 0 COMPARE REGISTER (TCPR0)
	$FFFF8C	TIMER 0 COUNT REGISTER (TCR0)
	$FFFF8B	TIMER 1 CONTROL/STATUS REGISTER (TCSR1)
	$FFFF8A	TIMER 1 LOAD REGISTER (TLR1)
	$FFFF89	TIMER 1 COMPARE REGISTER (TCPR1)
	$FFFF88	TIMER1 COUNT REGISTER (TCR1)
	$FFFF87	TIMER 2 CONTROL/STATUS REGISTER (TCSR2)
	$FFFF86	TIMER 2 LOAD REGISTER (TLR2)
	$FFFF85	TIMER 2 COMPARE REGISTER (TCPR2)
	$FFFF84	TIMER 2 COUNT REGISTER (TCR2)
	$FFFF83	TIMER PRESCALER LOAD REGISTER (TPLR)
	$FFFF82	TIMER PRESCALER COUNT REGISTER (TPCR)
	$FFFF81	RESERVED
	$FFFF80	RESERVED

3.4 DSP56301 MEMORY MAPS IN SIXTEEN-BIT COMPATIBILITY MODE

When the Sixteen-Bit-Compatibility mode bit is set, the DSP56301 memory map is changed to enable 16-bit wide address access to the memory mapped X-I/O, as described in the following figure:

Figure 3-2. DSP56301 Memory Map in Sixteen Bit Compatibility Mode

	PROGRAM		X DATA		Y DATA
$FFFF	EXTERNAL	$FFFF $FF80	INTERNAL I/O (128 words)	$FFFF $FF80	EXTERNAL I/O (128 words)
			EXTERNAL		EXTERNAL
$0FFF	1K INTERNAL I_Cache/PRAM				
$0BFF	3K INTERNAL PRAM	$07FF $0000	2K INTERNAL RAM	$07FF $0000	2K INTERNAL RAM
$0000					
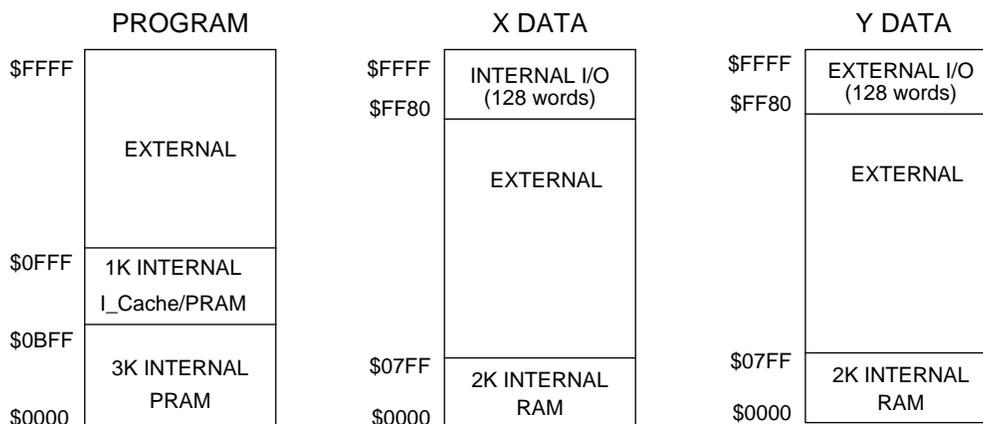
4 CORE CONFIGURATION

4.1 INTRODUCTION

This chapter contains DSP56300 Core configuration details specific to the DSP56301 device. For more information on the described registers or modules, refer to the appropriate chapters in the DSP56300 Core spec.

4.2 CHIP OPERATING MODES

The DSP56301 operating modes determine the chip operating modes and the start-up procedure location when the DSP56301 leaves the reset state. The MODA, MODB, MODC and MODD pins are sampled as the DSP56301 exits the reset state. Table 4-1 depicts the mode assignments.

Table 4-1. DSP56301 Operating Modes

Mode	MOD D	MOD C	MOD B	MOD A	Reset Vector	Description
0	0	0	0	0	$C00000	Expanded mode
1	0 or 1	0	0	1	$FF0000	Bootstrap from byte-wide memory
2	0 or 1	0	1	0	$FF0000	Bootstrap thru SCI
3	0 or 1	0	1	1	-	Reserved
4	0 or 1	1	0	0	$FF0000	Host Bootstrap PCI Mode (32-bit-wide)
5	0 or 1	1	0	1	$FF0000	Host Bootstrap 16-bit-wide UB Mode (ISA)
6	0 or 1	1	1	0	$FF0000	Host Bootstrap 8-bit-wide UB Mode in double-strob pin configuration
7	0 or 1	1	1	1	$FF0000	Host Bootstrap 8-bit-wide UB Mode in single-strob pin configuration
8	1	0	0	0	$008000	Expanded mode

4.2.1 Mode 0: Expanded mode

In this mode the Bootstrap ROM is bypassed and the DSP56301 will start fetching instructions beginning with the address $C00000 assuming that an external memory of SRAM type is used. The accesses will be performed using 31 wait states with no address attributes selected (default area).

4.2.2 Mode 1: Bootstrap from byte-wide external memory

In this mode the hardware reset vector is located at address $FF0000 in the Bootstrap ROM. The program stored in this location, after testing MODA, MODB and MODC, bootstraps thru Port A from an external byte-wide memory, starting at P:$D00000 (bits 7-0). The memory is selected by the Address Attribute AA1 and is accessed with 31 wait states. The boot program concatenates every 3 bytes read from the external memory into a 24-bit wide DSP56301 word.

4.2.3 Mode 2: Bootstrap thru SCI

In this mode the hardware reset vector is located at address $FF0000 in the Bootstrap ROM. The program stored in this location, after testing MODA, MODB and MODC, bootstraps thru the SCI. The boot program concatenates every 3 bytes read from the SCI into a 24-bit wide DSP56301 word.

4.2.4 Mode 3: Reserved

This mode is reserved for future use.

4.2.5 Mode 4: Bootstrap thru HI32 in PCI mode

In this mode the hardware reset vector is located at address $FF0000 in the Bootstrap ROM. The program stored in this location, after testing MODA, MODB and MODC, bootstraps thru HI32 in standard PCI slave configuration. The DSP56301 is written with 24-bit wide words encapsulated in 32-bit wide PCI transfers.

4.2.6 Mode 5: Bootstrap thru HI32 in 16-bit-wide UB mode (ISA)

In this mode the hardware reset vector is located at address $FF0000 in the Bootstrap ROM. The program stored in this location, after testing MODA, MODB and MODC, bootstraps thru HI32 in ISA slave configuration. The DSP56301 is written with 24-bit wide words broken into 16-bit wide ISA transfers.

4.2.7 Mode 6: Bootstrap thru HI32 in 8-bit-wide UB mode in double-strobe pin configuration

In this mode the hardware reset vector is located at address $FF0000 in the Bootstrap ROM. The program stored in this location, after testing MODA, MODB and MODC, boot-

straps thru HI32 in UB slave double-strobe ($\overline{HWR}$, $\overline{HRD}$) configuration. The DSP56301 is written with 24-bit wide words broken into 8-bit wide host bus transfers. This mode may be used for booting from various micro-processors or micro-controllers, as for booting a slave DSP56301 from port A of a master DSP56301.

4.2.8 Mode 7: Bootstrap thru HI32 in 8-bit-wide UB mode in single-strobe pin configuration

In this mode the hardware reset vector is located at address $FF0000 in the Bootstrap ROM. The program stored in this location, after testing MODA, MODB and MODC, boot-straps thru HI32 in UB slave single-strobe (HRW, $\overline{HDS}$) configuration. The DSP56301 is written with 24-bit wide words broken into 8-bit wide host bus transfers. This mode may be used for booting from various micro-processors or micro-controllers.

4.2.9 Mode 8: Expanded mode

In this mode the Bootstrap ROM is bypassed and the DSP56301 will start fetching instructions beginning with the address $008000 assuming that an external memory of SRAM type is used. The accesses will be performed using 31 wait states with no address attributes selected (default area).

4.3   BOOTSTRAP PROGRAM

The Bootstrap Program is factory-programmed in an internal 192-words by 24-bit Bootstrap ROM located in P memory space at locations $FF0000-$FF00BF. The Bootstrap program can load any program RAM segment from an external DSP56301 port.

When exiting reset, the DSP56301 samples MODA, MODB, and MODC pins to determine the reset vector location. In any of the bootstrap modes, program execution begins from internal P memory location $FF0000. The Bootstrap Program tests MODA, MODB and MODC bits in the Operating Mode Register (OMR) to determine the bootstrap mode, and then boots from the specified port. As part of the booting process, the Bootstrap Program reads, from the external port, the number of words to be loaded and the starting address in the DSP56301 P space. At the end of the Bootstrap Program, a jump is performed to the specified starting address in the DSP56301 P space and normal execution begins.

In the course of normal execution, any jump to Program location $FF0000 will activate the Bootstrap program according to the MODA, MODB and MODC values in the OMR.

The Bootstrap Program code is listed in Appendix A-1.

4.4 INTERRUPT SOURCES AND PRIORITIES

4.4.1 Interrupt Sources

The corresponding interrupt starting address for each interrupt source is shown in Table 4-2. These addresses are located in the 256 locations of program memory pointed to by the VBA (Vector Base Address) register in the program control unit.

In the DSP56301 only 46 of the 128 vector addresses are used for specific interrupt sources. The remaining 82 are defined reserved and may be used for Host NMI (IPL = 3) or for Host Command interrupt (IPL = 0-2). If it is known a-priori that certain interrupts will not be used at all, those interrupt vector locations can be used for program or data storage.

Table 4-2. Interrupt Sources

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$00	3	Hardware $\overline{\text{RESET}}$
VBA:$02	3	Stack Error
VBA:$04	3	Illegal Instruction
VBA:$06	3	Debug Request Interrupt
VBA:$08	3	Trap
VBA:$0A	3	Non-Maskable Interrupt ($\overline{\text{NMI}}$)
VBA:$0C	3	Reserved For Future Level-3 Interrupt Source
VBA:$0E	3	Reserved For Future Level-3 Interrupt Source
VBA:$10	0 - 2	$\overline{\text{IRQA}}$
VBA:$12	0 - 2	$\overline{\text{IRQB}}$
VBA:$14	0 - 2	$\overline{\text{IRQC}}$
VBA:$16	0 - 2	$\overline{\text{IRQD}}$
VBA:$18	0 - 2	DMA Channel 0
VBA:$1A	0 - 2	DMA Channel 1
VBA:$1C	0 - 2	DMA Channel 2
VBA:$1E	0 - 2	DMA Channel 3
VBA:$20	0 - 2	DMA Channel 4

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$22	0 - 2	DMA Channel 5
VBA:$24	0 - 2	TIMER 0 Compare
VBA:$26	0 - 2	TIMER 0 Overflow
VBA:$28	0 - 2	TIMER 1 Compare
VBA:$2A	0 - 2	TIMER 1 Overflow
VBA:$2C	0 - 2	TIMER 2 Compare
VBA:$2E	0 - 2	TIMER 2 Overflow
VBA:$30	0 - 2	ESSI0 Receive Data
VBA:$32	0 - 2	ESSI0 Receive Data With Exception Status
VBA:$34	0 - 2	ESSI0 Receive last slot
VBA:$36	0 - 2	ESSI0 Transmit Data
VBA:$38	0 - 2	ESSI0 Transmit Data with Exception Status
VBA:$3A	0 - 2	ESSI0 Transmit last slot
VBA:$3C	0 - 2	Reserved
VBA:$3E	0 - 2	Reserved
VBA:$40	0 - 2	ESSI1 Receive Data
VBA:$42	0 - 2	ESSI1 Receive Data With Exception Status
VBA:$44	0 - 2	ESSI1 Receive last slot
VBA:$46	0 - 2	ESSI1 Transmit Data
VBA:$48	0 - 2	ESSI1 Transmit Data with Exception Status
VBA:$4A	0 - 2	ESSI1 Transmit last slot
VBA:$4C	0 - 2	Reserved
VBA:$4E	0 - 2	Reserved
VBA:$50	0 - 2	SCI Receive Data
VBA:$52	0 - 2	SCI Receive Data with Exception Status
VBA:$54	0 - 2	SCI Transmit Data
VBA:$56	0 - 2	SCI idle Line
VBA:$58	0 - 2	SCI Timer
VBA:$5A	0 - 2	Reserved
VBA:$5C	0 - 2	Reserved
VBA:$5E	0 - 2	Reserved

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$60	0 - 2	Host PCI Transaction Termination
VBA:$62	0 - 2	Host PCI Transaction Abort
VBA:$64	0 - 2	Host PCI Parity Error
VBA:$66	0 - 2	Host PCI Transfer Complete
VBA:$68	0 - 2	Host PCI Master Receive Request
VBA:$6A	0 - 2	Host Slave Receive Request
VBA:$6C	0 - 2	Host PCI Master Transmit Request
VBA:$6E	0 - 2	Host Slave Transmit Request
VBA:$70	0 - 2	Host PCI Master Address Request
VBA:$72	0 - 2 / 3	Host Command / Host NMI (Default)
VBA:$74	0 - 2	Reserved
:	:	:
VBA:$FE	0 - 2	Reserved

NOTE Any Interrupt starting address (including reserved addresses) may be used for Host NMI (IPL = 3) and for Host command interrupt (IPL = 0-2).

4.4.2 Interrupt Priority Levels

There are two interrupt priority registers in the DSP56301: IPR-C is dedicated for DSP56300 Core interrupt sources and IPR-P is dedicated for DSP56301 peripheral interrupt sources. The interrupt priority registers shown in Figure 4-2 and Figure 4-1. Table 4-3 defines the IPL bits.

Table 4-3. Interrupt Priority Level Bits

IPL bits		Interrupts Enabled	Interrupt Priority Level
xxL1	xxL0		
0	0	No	—
0	1	Yes	0
1	0	Yes	1
1	1	Yes	2

Figure 4-1. Interrupt Priority Register P

11	10	9	8	7	6	5	4	3	2	1	0
		T0L1	T0L0	SCL1	SCL0	S1L1	S1L0	S0L1	S0L0	HPL1	HPL0

- HI32 IPL
- ESSI0 IPL
- ESSI1 IPL
- SCI IPL
- TRIPLE TIMER IPL
- reserved

23	22	21	20	19	18	17	16	15	14	13	12

- reserved
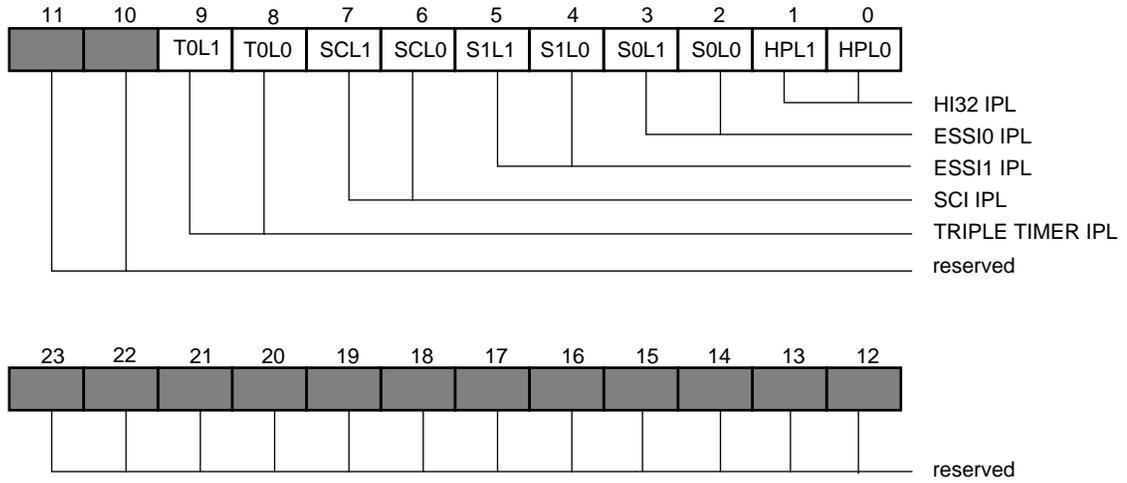
Figure 4-2. Interrupt Priority Register C

11	10	9	8	7	6	5	4	3	2	1	0
IDL2	IDL1	IDL0	ICL2	ICL1	ICL0	IBL2	IBL1	IBL0	IAL2	IAL1	IAL0

- IRQA IPL
- IRQA mode
- IRQB IPL
- IRQB mode
- IRQC IPL
- IRQC mode
- IRQD IPL
- IRQD mode

23	22	21	20	19	18	17	16	15	14	13	12
D5L1	D5L0	D4L1	D4L0	D3L1	D3L0	D2L1	D2L0	D1L1	D1L0	D0L1	D0L0

- DMA0 IPL
- DMA1 IPL
- DMA2 IPL
- DMA3 IPL
- DMA4 IPL
- DMA5 IPL
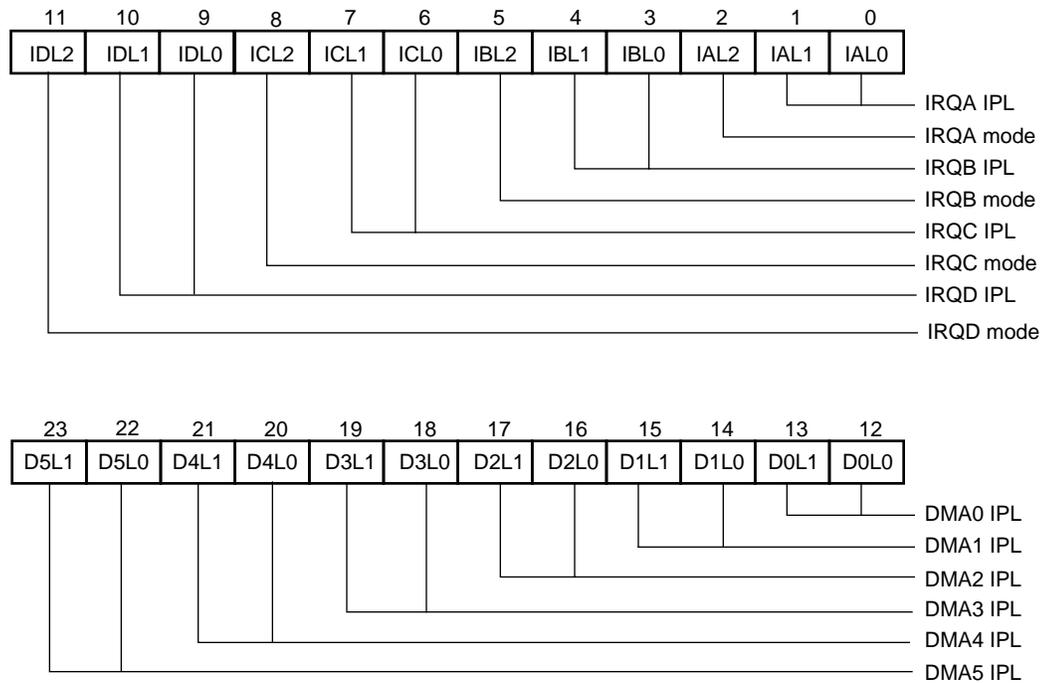
4.4.2.1 Interrupt Sources Priorities within an IPL

If more than one interrupt request is pending when an instruction is executed, the interrupt source with the highest priority level is serviced first. When multiple interrupt requests having the same IPL are pending, a second fixed-priority structure within that IPL determines which interrupt source is serviced. The fixed priority of interrupts sources within an IPL are shown in Table 4-4.

Table 4-4. Interrupt Sources Priorities within an IPL

Priority	Interrupt Source
Level 3 (Nonmaskable)	
Highest	Hardware $\overline{\text{RESET}}$
	Stack Error
	Illegal Instruction
	Debug Request Interrupt
	Trap
	Non-Maskable Interrupt
Lowest	Non-Maskable Host Command Interrupt
Levels 0, 1, 2 (Maskable)	
Highest	$\overline{\text{IRQA}}$ (External Interrupt)
	$\overline{\text{IRQB}}$ (External Interrupt)
	$\overline{\text{IRQC}}$ (External Interrupt)
	$\overline{\text{IRQD}}$ (External Interrupt)
	DMA Channel 0 Interrupt
	DMA Channel 1 Interrupt
	DMA Channel 2 Interrupt

Priority	Interrupt Source
	DMA Channel 3 Interrupt
	DMA Channel 4 Interrupt
	DMA Channel 5 Interrupt
	Host Command Interrupt
	Host PCI Transaction Termination
	Host PCI Transaction Abort
	Host PCI Parity Error
	Host PCI Transfer Complete
	Host PCI Master Receive Request
	Host Slave Receive Request
	Host PCI Master Transmit Request
	Host Slave Transmit Request
	Host PCI Master Address Request
	ESSI0 RX Data with Exception Interrupt
	ESSI0 RX Data Interrupt
	ESSI0 Receive last slot interrupt
	ESSI0 TX Data with Exception Interrupt
	ESSI0 Transmit last slot interrupt
	ESSI0 TX Data Interrupt
	ESSI1 RX Data with Exception Interrupt
	ESSI1 RX Data Interrupt
	ESSI1 Receive last slot interrupt

Priority	Interrupt Source
	ESSI1 TX Data with Exception Interrupt
	ESSI1 Transmit last slot interrupt
	ESSI1 TX Data Interrupt
	SCI Receive Data with Exception Interrupt
	SCI Receive Data
	SCI Transmit Data
	SCI idle Line
	SCI Timer
	TIMER0 Overflow Interrupt
	TIMER0 Compare Interrupt
	TIMER1 Overflow Interrupt
	TIMER1 Compare Interrupt
	TIMER2 Overflow Interrupt
Lowest	TIMER2 Compare Interrupt

4.5   DMA REQUEST SOURCES

The DMA Request Source bits (DRS0-DRS4 bits in the DMA Control/Status registers) encode the source of DMA requests used to trigger the DMA transfers. The DMA request sources may be the internal peripherals or external devices requesting service through the $\overline{IRQA}$, $\overline{IRQB}$, $\overline{IRQC}$ and $\overline{IRQD}$ pins.

DMA Request Source Bits DRS4...DRS0	Requesting Device
00000	External ($\overline{\text{IRQA}}$ pin)
00001	External ($\overline{\text{IRQB}}$ pin)
00010	External ($\overline{\text{IRQC}}$ pin)
00011	External ($\overline{\text{IRQD}}$ pin)
00100	Transfer Done from DMA channel 0
00101	Transfer Done from DMA channel 1
00110	Transfer Done from DMA channel 2
00111	Transfer Done from DMA channel 3
01000	Transfer Done from DMA channel 4
01001	Transfer Done from DMA channel 5
01010	ESSI0 Receive Data (RDF0=1)
01011	ESSI0 Transmit Data (TDE0=1)
01100	ESSI1 Receive Data (RDF1=1)
01101	ESSI1 Transmit Data (TDE1=1)
01110	SCI Receive Data (RDRF=1)
01111	SCI Transmit Data (TDRE=1)
10000	Timer0 (TCF0=1)
10001	Timer1 (TCF1=1)
10010	Timer2 (TCF2=1)
10011-11011	RESERVED
11100	Host Slave Receive Data (SRRQ=1)
11101	Host Master Receive Data (MRRQ=1)
11110	Host Slave Transmit Data (STRQ=1)
11111	Host Master Transmit Data (MTRQ=1)

4.6 OPERATING MODE REGISTER

The Operating Mode Register (OMR) in the DSP56301 is described in the following figure:

Figure 4-3. DSP56301 Operating Mode Register (OMR) Format

SCS								EOM								COM							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SEN	WRP	EOV	EUN	XYS				BRT	TAS	BE	CDP1:0			SD		EBD	MD	MC	MB	MA

SEN	- Stack Extension Enable	BRT	- Bus Release Timing	SD	- Stop Delay
WRP	- Extended Stack Wrap Flag	TAS	- TA Synchronize Select	EBD	- External Bus Disable
EOV	- Extended Stack Overflow Flag	BE	- Burst Mode Enable	MD	- Operating Mode D
EUN	- Extended Stack Underflow Flag	CDP1	- Core-Dma Priority 1	MC	- Operating Mode C
XYS	- Stack Extension Space Select	CDP0	- Core-Dma Priority 0	MB	- Operating Mode B
				MA	- Operating Mode A

- Reserved bit. Read as zero, should be written with zero for future compatibility

4.7 PLL MULTIPLICATION FACTOR

The Multiplication Factor Bits MF0-MF11 in the **PLL Control Register (PCTL)** define the multiplication factor MF that will be applied to the PLL input frequency. The multiplication factor bits (MF0-MF11) are set to a predetermined value during hardware reset; in the DSP56301 this value is $000 which correspond to a multiplication factor MF of 1.

4.8 DEVICE IDENTIFICATION (ID) REGISTER

The Device Identification Register (IDR) is a 24 bit read only factory programmed register used to identify the different DSP56300 core-based family members. This register specifies the derivative number and revision number. This information may be used in testing or by software. Figure 4-4 shows the ID register configuration.

Figure 4-4. Identification Register Configuration

	23 16	15 12	11 0
	Zeros	Rev. Number	Derivative Number
	00000000	0001	001100000001

The Derivative Number (bits 11:0) for the DSP56301 are 001100000001 (i.e. 301 for 56301).
The Revision Number (bits 15:12) for the DSP56301 first silicon is 0001. Therefore the ID register value for DSP56301 first silicon is $001301.

4.9 JTAG IDENTIFICATION (ID) REGISTER

The JTAG Identification (ID) Register is a 32 bit, read only thought JTAG, factory programmed register used to distinguish the component on a board according to the IEEE 1149.1 standard. Figure 4-5 shows the JTAG ID register configuration.

Figure 4-5. JTAG Identification Register Configuration

31 28	27 12	11 1	0
Version Information	Customer Part Number	Manufacturer Identity	1
0001	000110 | 0000000001	00000001 110	1

Motorola's Manufacturer Identity is 00000001110. The Customer Part Number consists of two parts: Motorola Design Center Number (bits 27:22) and a sequence number (bits 21:12). MSIL Design Center Number is 000110. The sequence number for the DSP56301 is 0000000001. The Version Information for DSP56301 first silicon is 0001. Therefore the ID register value for DSP56301 first silicon is $1180101d.

4.10 JTAG BOUNDARY SCAN REGISTER

The boundary scan register (BSR) in the DSP56301 JTAG implementation contains bits for all device signal and clock pins and associated control signals. All DSP56301 bidirectional pins have a single register bit in the boundary scan register for pin data, and are controlled by an associated control bit in the boundary scan register.

The DSP56301 boundary scan register bit definitions are described in the following table:

Table 4-5. DSP56301 BSR Bit Definition

Bit Number	Pin Name	Pin Type	BSR Cell Type
219	$\overline{DE}$	INPUT/OUTPUT	DATA
218	$\overline{DE}$	-	CONTROL
217	SC20	INPUT/OUTPUT	DATA
216	SC20	-	CONTROL
215	SC10	INPUT/OUTPUT	DATA
214	SC10	-	CONTROL
213	SC00	INPUT/OUTPUT	DATA
212	SC00	-	CONTROL
211	STD0	INPUT/OUTPUT	DATA
210	STD0	-	CONTROL
209	SCK0	INPUT/OUTPUT	DATA
208	SCK0	-	CONTROL
207	SRD0	INPUT/OUTPUT	DATA
206	SRD0	-	CONTROL
205	SRD1	INPUT/OUTPUT	DATA
204	SRD1	-	CONTROL
203	SCK1	INPUT/OUTPUT	DATA
202	SCK1	-	CONTROL
201	STD1	INPUT/OUTPUT	DATA
200	STD1	-	CONTROL
199	SC01	INPUT/OUTPUT	DATA
198	SC01	-	CONTROL
197	SC11	INPUT/OUTPUT	DATA
196	SC11	-	CONTROL
195	SC21	INPUT/OUTPUT	DATA

Bit Number	Pin Name	Pin Type	BSR Cell Type
194	SC21	-	CONTROL
193	STX	INPUT/OUTPUT	DATA
192	STX	-	CONTROL
191	$\overline{\text{HINTA}}$	TS	DATA
190	$\overline{\text{HINTA}}$	-	CONTROL
189	SCLK	INPUT/OUTPUT	DATA
188	SCLK	INPUT/OUTPUT	CONTROL
187	SRX	INPUT/OUTPUT	DATA
186	SRX	INPUT/OUTPUT	CONTROL
185	TIO0	INPUT/OUTPUT	DATA
184	TIO0	INPUT/OUTPUT	CONTROL
183	TIO1	INPUT/OUTPUT	DATA
182	TIO1	INPUT/OUTPUT	CONTROL
181	TIO2	INPUT/OUTPUT	DATA
180	TIO2	INPUT/OUTPUT	CONTROL
179	HAD0	INPUT/OUTPUT	DATA
178	HAD0	INPUT/OUTPUT	CONTROL
177	HAD1	INPUT/OUTPUT	DATA
176	HAD1	INPUT/OUTPUT	CONTROL
175	HAD2	INPUT/OUTPUT	DATA
174	HAD2	INPUT/OUTPUT	CONTROL
173	HAD3	INPUT/OUTPUT	DATA
172	HAD3	INPUT/OUTPUT	CONTROL
171	HAD4	INPUT/OUTPUT	DATA
170	HAD4	INPUT/OUTPUT	CONTROL
169	HAD5	INPUT/OUTPUT	DATA

Bit Number	Pin Name	Pin Type	BSR Cell Type
168	HAD5	INPUT/OUTPUT	CONTROL
167	HAD6	INPUT/OUTPUT	DATA
166	HAD6	INPUT/OUTPUT	CONTROL
165	HAD7	INPUT/OUTPUT	DATA
164	HAD7	INPUT/OUTPUT	CONTROL
163	HC/HBE0	INPUT/OUTPUT	DATA
162	HC/HBE0	INPUT/OUTPUT	CONTROL
161	HAD8	INPUT/OUTPUT	DATA
160	HAD8	INPUT/OUTPUT	CONTROL
159	HAD9	INPUT/OUTPUT	DATA
158	HAD9	INPUT/OUTPUT	CONTROL
157	HAD10	INPUT/OUTPUT	DATA
156	HAD10	INPUT/OUTPUT	CONTROL
155	HAD11	INPUT/OUTPUT	DATA
154	HAD11	INPUT/OUTPUT	CONTROL
153	HAD12	INPUT/OUTPUT	DATA
152	HAD12	INPUT/OUTPUT	CONTROL
151	HAD13	INPUT/OUTPUT	DATA
150	HAD13	INPUT/OUTPUT	CONTROL
149	HAD14	INPUT/OUTPUT	DATA
148	HAD14	INPUT/OUTPUT	CONTROL
147	HAD15	INPUT/OUTPUT	DATA
146	HAD15	INPUT/OUTPUT	CONTROL
145	HC/HBE1	INPUT/OUTPUT	DATA
144	HC/HBE1	INPUT/OUTPUT	CONTROL
143	$\overline{\text{HGNT}}$	INPUT	DATA

Bit Number	Pin Name	Pin Type	BSR Cell Type
142	$\overline{\text{HGNT}}$	INPUT	CONTROL
141	HCLK	INPUT	DATA
140	$\overline{\text{HRST}}$	INPUT	DATA
139	$\overline{\text{HRST}}$	INPUT	CONTROL
138	$\overline{\text{HREQ}}$	TS	DATA
137	$\overline{\text{HREQ}}$	TS	CONTROL
136	HPAR	INPUT/OUTPUT	DATA
135	HPAR	INPUT/OUTPUT	CONTROL
134	HSERR	TS	DATA
133	HSERR	TS	CONTROL
132	HPERR	INPUT/OUTPUT	DATA
131	HPERR	INPUT/OUTPUT	CONTROL
130	HLOCK	INPUT/OUTPUT	DATA
129	HLOCK	INPUT/OUTPUT	CONTROL
128	HSTOP	INPUT/OUTPUT	DATA
127	HSTOP	INPUT/OUTPUT	CONTROL
126	$\overline{\text{HDEVSEL}}$	INPUT/OUTPUT	DATA
125	$\overline{\text{HDEVSEL}}$	INPUT/OUTPUT	CONTROL
124	$\overline{\text{HTRDY}}$	INPUT/OUTPUT	DATA
123	$\overline{\text{HTRDY}}$	INPUT/OUTPUT	CONTROL
122	$\overline{\text{HIRDY}}$	INPUT/OUTPUT	DATA
121	$\overline{\text{HIRDY}}$	INPUT/OUTPUT	CONTROL
120	$\overline{\text{HFRAME}}$	INPUT/OUTPUT	DATA
119	$\overline{\text{HFRAME}}$	INPUT/OUTPUT	CONTROL
118	HIDSEL	INPUT	DATA
117	HIDSEL	INPUT	CONTROL

Bit Number	Pin Name	Pin Type	BSR Cell Type
116	HC/HBE2	INPUT/OUTPUT	DATA
115	HC/HBE2	INPUT/OUTPUT	CONTROL
114	HAD16	INPUT/OUTPUT	DATA
113	HAD16	INPUT/OUTPUT	CONTROL
112	HAD17	INPUT/OUTPUT	DATA
111	HAD17	INPUT/OUTPUT	CONTROL
110	HAD18	INPUT/OUTPUT	DATA
109	HAD18	INPUT/OUTPUT	CONTROL
108	HAD19	INPUT/OUTPUT	DATA
107	HAD19	INPUT/OUTPUT	CONTROL
106	HAD20	INPUT/OUTPUT	DATA
105	HAD20	INPUT/OUTPUT	CONTROL
104	HAD21	INPUT/OUTPUT	DATA
103	HAD21	INPUT/OUTPUT	CONTROL
102	HAD22	INPUT/OUTPUT	DATA
101	HAD22	INPUT/OUTPUT	CONTROL
100	HAD23	INPUT/OUTPUT	DATA
99	HAD23	INPUT/OUTPUT	CONTROL
98	HC/HBE3	INPUT/OUTPUT	DATA
97	HC/HBE3	INPUT/OUTPUT	CONTROL
96	HAD24	INPUT/OUTPUT	DATA
95	HAD24	INPUT/OUTPUT	CONTROL
94	HAD25	INPUT/OUTPUT	DATA
93	HAD25	INPUT/OUTPUT	CONTROL
92	HAD26	INPUT/OUTPUT	DATA
91	HAD26	INPUT/OUTPUT	CONTROL

Bit Number	Pin Name	Pin Type	BSR Cell Type
90	HAD27	INPUT/OUTPUT	DATA
89	HAD27	INPUT/OUTPUT	CONTROL
88	HAD28	INPUT/OUTPUT	DATA
87	HAD28	INPUT/OUTPUT	CONTROL
86	HAD29	INPUT/OUTPUT	DATA
85	HAD29	INPUT/OUTPUT	CONTROL
84	HAD30	INPUT/OUTPUT	DATA
83	HAD30	INPUT/OUTPUT	CONTROL
82	HAD31	INPUT/OUTPUT	DATA
81	HAD31	-	CONTROL
80	$\overline{\text{IRQD}}$	INPUT	DATA
79	$\overline{\text{IRQC}}$	INPUT	DATA
78	$\overline{\text{IRQB}}$	INPUT	DATA
77	$\overline{\text{IRQA}}$	INPUT	DATA
76	D23	INPUT/OUTPUT	DATA
75	D22	INPUT/OUTPUT	DATA
74	D21	INPUT/OUTPUT	DATA
73	D20	INPUT/OUTPUT	DATA
72	D19	INPUT/OUTPUT	DATA
71	D18	INPUT/OUTPUT	DATA
70	D17	INPUT/OUTPUT	DATA
69	D16	INPUT/OUTPUT	DATA
68	D15	INPUT/OUTPUT	DATA
67	D[23:12]	-	CONTROL
66	D14	INPUT/OUTPUT	DATA
65	D13	INPUT/OUTPUT	DATA

Bit Number	Pin Name	Pin Type	BSR Cell Type
64	D12	INPUT/OUTPUT	DATA
63	D11	INPUT/OUTPUT	DATA
62	D10	INPUT/OUTPUT	DATA
61	D9	INPUT/OUTPUT	DATA
60	D8	INPUT/OUTPUT	DATA
59	D7	INPUT/OUTPUT	DATA
58	D6	INPUT/OUTPUT	DATA
57	D5	INPUT/OUTPUT	DATA
56	D4	INPUT/OUTPUT	DATA
55	D3	INPUT/OUTPUT	DATA
54	D[11:0]	-	CONTROL
53	D2	INPUT/OUTPUT	DATA
52	D1	INPUT/OUTPUT	DATA
51	D0	INPUT/OUTPUT	DATA
50	A23	TS	DATA
49	A22	TS	DATA
48	A21	TS	DATA
47	A20	TS	DATA
46	A19	TS	DATA
45	A18	TS	DATA
44	A[23:12]	-	CONTROL
43	A17	TS	DATA
42	A16	TS	DATA
41	A15	TS	DATA
40	A14	TS	DATA
39	A13	TS	DATA

Bit Number	Pin Name	Pin Type	BSR Cell Type
38	A12	TS	DATA
37	A11	TS	DATA
36	A10	TS	DATA
35	A9	TS	DATA
34	A8	TS	DATA
33	A7	TS	DATA
32	A6	TS	DATA
31	A[11:0]	-	CONTROL
30	A5	TS	DATA
29	A4	TS	DATA
28	A3	TS	DATA
27	A2	TS	DATA
26	A1	TS	DATA
25	A0	TS	DATA
24	EXTAL	INPUT	DATA
23	$\overline{RD}$	TS	DATA
22	$\overline{WR}$	TS	DATA
21	AA3	TS	DATA
20	AA2	TS	DATA
19	AA3	-	CONTROL
18	AA2	-	CONTROL
17	$\overline{BB}$	-	CONTROL
16	$\overline{BR}$	OUTPUT	DATA
15	$\overline{BG}$	INPUT	DATA
14	$\overline{BB}$	INPUT/OUTPUT	DATA
13	$\overline{RES}$	INPUT	DATA

Bit Number	Pin Name	Pin Type	BSR Cell Type
12	PINIT	INPUT	DATA
11	$\overline{\text{TA}}$	INPUT	DATA
10	$\overline{\text{CAS}}$	TS	DATA
9	BCLK	TS	DATA
8	CLKOUT	OUTPUT	DATA
7	$\overline{\text{RD}},\overline{\text{WR}},$BCLK,$\overline{\text{BS}}$	-	CONTROL
6	$\overline{\text{CAS}}$	-	CONTROL
5	AA1	-	CONTROL
4	AA0	-	CONTROL
3	AA1	TS	DATA
2	AA0	TS	DATA
1	$\overline{\text{BL}}$	OUTPUT	DATA
0	$\overline{\text{BS}}$	TS	DATA

5   GENERAL PURPOSE I/O

5.1   INTRODUCTION

The general purpose I/O capability of the DSP56301 consists of 42 bidirectional pins separated into five different groups, each group being separately controlled. The groups are:

1.   Port B -24 Pins (shared with part of the host interface pins)
2.   Port C - 6 Pins (shared with ESSI0 pins)
3.   Port D - 6 Pins (shared with ESSI1 pins)
4.   Port E - 3 Pins (shared with SCI pins)
5.   Timers -3 Pins (shared with Triple Timer pins)

5.2   PROGRAMMING MODEL

5.2.1   Port B pins and registers

24 pins of the HI32 pin may be configured as a GPIO pin. The GPIO functionality of port B is controlled by three registers: DSP Control Register (DCTR), DSP Host Port GPIO Direction Register (DIRH) and DSP Host Port GPIO Data Register (DATH). These registers are described in Paragraphs 6.1.1, 6.1.13 and 6.1.12.

5.2.2   Port C pins and registers

Each one of the six ESSI0 pins may be configured individually as a GPIO pin. The GPIO functionality of port C is controlled by three registers: Port C Control register (PCRC), Port C Direction register (PRRC) and Port C Data register (PDRC).These registers are described in Paragraphs 8.5.1, 8.5.2 and 8.5.3.

5.2.3   Port D pins and registers

Each one of the six ESSI1 pins may be configured individually as a GPIO pin. The GPIO functionality of port D is controlled by three registers: Port D Control register (PCRD), Port D Direction register (PRRD) and Port D Data register (PDRD).These registers are described in Paragraphs 8.5.1, 8.5.2 and 8.5.3.

5.2.4   Port E pins and registers

Each one of the three SCI pins may be configured individually as a GPIO pin. The GPIO functionality of port E is controlled by three registers: Port E Control register (PCRE), Port E Direction register (PRRE) and Port E Data register (PDRE). These registers are described in Paragraphs 9.5.1, 9.5.2 and 9.5.3.

5.2.5   Triple Timer pins

Each one of the three Timers in the DSP56301 Triple Timer interface has one pin that may be configured individually as a GPIO pin. Each pin is controlled by the appropriate Timer Control register, as described in Paragraph 7.4.5.

6 HOST INTERFACE (HI32)

The Host Interface (HI32) provides a fast parallel host port up to 32 bits wide, which may be connected directly to the host bus.

The HI32 supports a variety of standard buses, and provides glue-less connection with a number of industry standard microcomputers, microprocessors, DSPs and DMA controllers.

The host bus may operate asynchronously to the DSP clock, therefore the HI32 registers are divided into two banks – the 'Host side' bank is accessible to the external host, and the 'DSP side' bank is accessible to the DSP56300 Core. Figure 6-1 on page 6-7 is a block diagram showing the registers in the HI32.

The HI32 supports three classes of interfaces:
- Peripheral Component Interconnect (PCI) bus (PCI Specification Revision 2.0)
- Universal bus interface
- General purpose I/O (GPIO) port

In the PCI mode, the HI32 is a dedicated, bidirectional, target (slave) / initiator (master) parallel port, with a 32-bit wide, data path, up to eight words deep. In this mode, the HI32 may be connected directly to the PCI bus.

In the Universal Bus (UB) modes, the HI32 is a dedicated, bidirectional, slave only parallel port, with a six word deep data path up to 24 bits wide. In this mode, the HI32 may be connected directly to 8-bit data buses, 16-bit data buses (e.g. ISA/EISA, Micro Channel) and 24-bit data buses (e.g. DSP56300 Core based DSP Port A bus).

The host port pin functionality and polarity are controlled by the DSP56300 Core programming the DSP control register (DCTR).

Host port pins that are not used may be programmed by the DSP56300 Core as general purpose I/O pins. The HI32 provides up to 24 general purpose I/O pins.

Below is a brief list of the HI32 features:

Interface - DSP56300 Core Side

Mapping: 11 internal I/O space locations

Word Size: 24 bits

Data Format Conversion:
   PCI Mode:
        Output data alignment of 16 bit words to 16 bit double words (Dwords)
        Output data alignment of 24 bit words to 32 bit Dwords
               (left aligned and zero filled, right aligned and zero extended,
               right aligned and sign extended).
        Input data alignment of 32 bit Dwords to 24 bit words
               (three most significant bytes, three least significant bytes).
        True 32-bit (Dword) input and output data transfers.
               (32-bit PCI bus data to two DSP56300 Core 16-bit words, and vice
               versa)
   UB Mode:
        Output data alignment of 24 bit words to 16 bit words
               (two most significant bytes, two least significant bytes).
        Input data alignment of 16 bit words to 24 bit words
               (left aligned and zero filled, right aligned and zero extended,
               right aligned and sign extended).

Data Buffers: FIFOs, up to eight words deep, on both transmit and receive data paths

Handshake Protocols:
   Software Polled
   Interrupt Driven (Fast or Long)
   Direct Memory Access (up to four DSP56300 Core DMA channels)

GPIO:
   Up to 24 I/O pins (data and pin direction are programmable)

Self Configuration:
   The DSP56300 Core can access, indirectly, the CCMR, CLAT and CBMA HI32
   configuration registers.

Instructions:
   Memory mapped registers allow the standard MOVE instruction to be used.
   The special MOVEP instruction provides for I/O service capability using fast
   interrupts, and provides faster execution with fewer instruction words.

Interface - Host Side

Mapping:
    PCI Mode:

Memory Space:	16K Dword (32-bit wide) locations composed of:

                3 Dword read/write registers
                        (control, status and host command)
                16377 Dword read/write locations corresponding to
                        one Dword input data FIFO, and
                        one Dword output data FIFO.
                4 Dword reserved locations (read only)

Configuration Space:	64 32-bit Dword locations
Universal Bus Mode:	8 locations up to 24-bits wide (of which 4 locations are reserved)

Address Decoding:

PCI Mode:	32 bit internal address decoding
Universal Bus Mode:	11 bit (12 with HAEN) internal address decoding

Word Size: 8, 16, 24 or 32 bits

Data Buffers: FIFOs, six or eight words deep, on transmit and receive data paths

Data Fetch Types in HI32 (slave) to Host Data Transfers:
    Fetch
    Pre-fetch

Semaphores:
    Semaphore flags are supplied for HI32 allocation in a multi-host system

Handshake Protocols:

        Universal Bus mode Handshake Protocols:
            Software Polled
            Interrupt Driven - Data Request ($\overline{\text{HIRQ}}$ pin) and Interrupt A ($\overline{\text{HINTA}}$ pin)
            Data Acknowledge (HTA pin)
            Direct Memory Access (External DMA - HDRQ and $\overline{\text{HDAK}}$ pins)

        PCI mode Handshake Protocols:
            Software Polled
             PCI Interrupt ($\overline{\text{HINTA}}$ pin)
            Data Acknowledge ($\overline{\text{HTRDY}}$ and $\overline{\text{HIRDY}}$ pins)
            Bus Arbitration ($\overline{\text{HREQ}}$ and $\overline{\text{HGNT}}$ pins)

HI32 features in the PCI mode:

1.   Operates as an initiator (master) or target (slave).
2.   Up to 33 Mword/sec zero-wait-state data transfers (with a 33MHz PCI clock and a DSP clock frequency of 66MHz or more).
3.   Supports words 8,16, 24 and 32 bits wide (as defined by the $\overline{\text{HBE3}}$-$\overline{\text{HBE0}}$ lines).

4. Supports output data alignment of 24 bit words to 32 bit Dwords (left aligned and zero filled, right aligned and zero extended, right aligned and sign extended).
5. Supports input data alignment of 32 bit Dwords to 24 bit words (three most significant bytes, three least significant bytes).
6. Supports true 32-bit (Dword) input and output data transfers. (32-bit PCI bus data to two DSP56300 Core 16-bit words, and vice versa)
7. Supports bursts of up to 16384 Dwords when accessed as a memory space mapped target.
8. Generates bursts of up to 64 Dwords or unlimited length (as master).
9. Supports high speed ("fast peripheral") DSP56300 Core DMA transfers (two Core clock cycles per DMA transfer).
10. Supports memory-space and configuration transactions as a target.
11. Supports memory-space, I/O-space and configuration transactions as an initiator.
12. Supports exclusive (locked) accesses.
13. Supports a self configuration mode, for initialization of the configuration registers in a system without an external system configurator.
14. Supports PCI Interrupt Requests (Interrupt A). This interrupt request is software driven.
15. Generates vectored DSP56300 Core interrupts; separately for receive, transmit, transaction termination, error events and host commands.
16. Supports both 3.3V and 5V PCI signalling environments.
17. Supports address insertion in the data written to the HI32.
18. Supports parity generation, detection and reporting.
19. Supports system error generation and reporting.

HI32 features in the Universal Bus modes:

1. Operates as a slave in many standard bus environments (e.g. ISA bus or DSP56300 Core based DSP Port A bus).
2. Transfers data at three clock cycles per transfer (i.e. 22 Mword/sec for a 66MHz DSP clock), when operating synchronously with an DSP56300 Core based DSP host (two wait states per access).
3. Supports high speed ("fast peripheral") DSP56300 Core DMA transfers (two Core clock cycles per DMA transfer).
4. Supports words 8,16, and 24 bits wide.
5. Supports output data alignment of 24 bit words to 16 bit words (two most significant bytes, two least significant bytes).
6. Supports input data alignment of 16 bit words to 24 bit words (left aligned and zero filled, right aligned and zero extended, right aligned and sign extended).
7. Supports an external data buffer for drive and voltage level compatibility with the external bus (e.g. ISA bus).
8. Generates interrupt requests: hardware driven ($\overline{\text{HIRQ}}$) and software driven ($\overline{\text{HINTA}}$).

9.   Generates vectored DSP56300 Core interrupts; separately for receive and transmit events and host commands.

Below is a description of the various HI32 resets.

Type			Entered when	Description
Initiated by the DSP56300 Core	Hardware Reset	HS	The DSP56300 Core $\overline{\text{RESET}}$ pin is asserted.	These resets force the HI32 DSP side state machines, control and status registers to their initial states.
	Software Reset		The RESET instruction is executed.	These resets also activate the Personal Software (PS) reset (see below).
	Personal Software Reset	PS	The DSP56300 Core writes zeros to the HI32 mode bits HM2-HM0 in the DSP control register or the HS reset has been executed	The HI32 terminates the current PCI transaction (if it is an active PCI agent), clears the HACT bit in the DSP status register (DSR) and enters the personal software (PS) reset state. All data paths are cleared. In the personal software reset state, the HI32 is a PCI agent and will respond to all memory space transactions with a retry event. If connected to other buses (e.g. ISA bus, DSP56300 Core based DSP Port A bus, etc.) all outputs are high impedance.
	STOP Reset	ST	The STOP instruction is executed.	This reset forces all host port pins to the disconnected state: all outputs are high impedance, all inputs are electrically disconnected. The host port pins are affected immediately. Note: this reset may be executed only when the HACT bit in the DSP status register (DSR) is zero.
Initiated by the Host	Personal Hardware Reset	PH	The HI32 $\overline{\text{HRST}}$/HRST pin is asserted.	This reset forces the HI32 host side state machines, control registers, and configuration registers to their initial states. All host port pins, except $\overline{\text{HRST}}$/HRST, are forced to the disconnected state: all outputs are high impedance, inputs are electrically disconnected. The DSP side state machines are not affected. The $\overline{\text{HRST}}$/HRST pin is ignored in the self configuration mode.

Figure 6-1 on page 6-7 is a block diagram showing the registers in the HI32. These registers are divided into two banks:

- The "Host side" registers can be accessed by the host bus:
  - control, status, vector and data registers and FIFOs (HCTR, HSTR, HCVR, HTXR, HRXS and HRXM[1])
  - PCI configuration registers (CSTR/CCMR, CDID/CVID, CCCR/CRID, CHTY/CLAT, CBMA and CILP)

- The "DSP side" registers can be accessed by the DSP56300 Core
  - internal I/O space registers (DCTR, DSR, DPCR, DPAR, DPMC, DPSR, DRXR, DTXS and DTXM)

---

1. The HRXM is used by the HI32, as the PCI master, to output data, and cannot actually be accessed by the host bus.

Figure 6-1. HI32 Block Diagram

DSP Side Registers

DCTR	DSP Control Register
DPCR	DSP PCI Control Register
DPMC	DSP PCI Master Control Register
DPAR	DSP PCI Address Register
DSR	DSP Status Register
DPSR	DSP PCI Status Register

DTXM	DSP Master Transmit Data FIFO
DRXR	DSP Receive Data FIFO
DTXS	DSP Slave Transmit Data FIFO
DIRH	DSP GPIO Direction Register
DATH	DSP GPIO Data Register

Host Side Registers

HCTR	Host Interface Control Register
HSTR	Host Interface Status Register
HCVR	Host Command Vector Register
HRXM	Host Master Receive Data FIFO
HRXS	Host Slave Receive Data FIFO
HTXR	Host Transmit Data FIFO

CDID/CVID	Device ID/Vendor ID Configuration Register
CSTR/CCMR	Status/Command Configuration Register
CCCR/CRID	Class Code/Revision ID Configuration Register
CHTY/CLAT	Header Type/Latency Timer Configuration Register
CBMA	Memory Space Base Address Configuration Register
CILP	Interrupt Line -Interrupt Pin Configuration Register

6.1 HI32 - DSP SIDE

The DSP56300 Core views the HI32 as a memory-mapped peripheral occupying 11 24-bit words in data memory space.

The HI32 DSP side programming model is shown in Figure 6-2.

Figure 6-2. HI32 Programming Model - DSP Side

Address	Register
HI32 via programmed base address: $5	DSP Control Register (DCTR)
HI32 via programmed base address: $6	DSP PCI Control Register (DPCR)
HI32 via programmed base address: $7	DSP PCI Master Control Register (DPMC)
HI32 via programmed base address: $8	DSP PCI Address Register (DPAR)
HI32 via programmed base address: $9	DSP Status Register (DSR)
HI32 via programmed base address: $A	DSP PCI Status Register (DPSR)
HI32 via programmed base address: $B	DSP Receive Data FIFO (DRXR)
HI32 via programmed base address: $C	DSP Master Transmit Data FIFO (DTXM)
HI32 via programmed base address: $D	DSP Slave Transmit Data FIFO (DTXS)
HI32 via programmed base address: $E	Host Port GPIO Direction Register (DIRH)
HI32 via programmed base address: $F	Host Port GPIO Data Register (DATH)

The separate host-to-DSP and DSP-to-host data paths are FIFOs to allow the HI32 and the host processor to transfer data efficiently and at high speeds.

Memory mapping allows DSP56300 Core data transfers with the HI32 registers to be accomplished using standard instructions and addressing modes. In addition, the MOVEP instruction allows HI32-to-memory and memory-to-HI32 data transfers without going through an intermediate register.

The on-chip general purpose DMA channels, in the DSP56300 Core, can be programmed to transfer data between the HI32 data FIFOs and other DMA accessible resources at maximum throughput, without DSP56300 Core intervention.

The DSP56300 Core may access the HI32, using either standard polling, interrupt or DMA techniques.

The following paragraphs describe the purpose and operation of each bit in each register of the HI32, visible to the DSP56300 Core. The effects of the different types of reset on these registers are shown.

The HI32 host side programming model is described in Section 6.2.

6.1.1 DSP Control Register (DCTR)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HM2	HM1	HM0	HIRD	HIRH	HRSP	HDRP	HTAP	HRWP	HDSM							HINT	HF5	HF4	HF3	SRIE	STIE	HCIE

Reserved, read as zero and should be written with zero

Bit	Name	Function
0	HCIE	Host Command Interrupt Enable
1	STIE	Slave Transmit Interrupt Enable
2	SRIE	Slave Receive Interrupt Enable
5-3	HF5-HF3	Host Flags
6	HINT	Host Interrupt A
13	HDSM	Host Data Strobe Mode
14	HRWP	Host RD/WR Polarity
15	HTAP	Host Transfer Acknowledge Polarity
16	HDRP	Host DMA Request Polarity
17	HRSP	Host Reset Polarity
18	HIRH	Host Interrupt Request Handshake Mode
19	HIRD	Host Interrupt Request Drive Control
22-20	HM2-HM0	HI32 Mode
23,12-7	reserved	

The DCTR is a 24-bit read/write control register used by the DSP56300 Core to control the HI32 interrupts, flags and the host port pin functionality. The DCTR cannot be accessed by the host processor. All reserved bits are read as zeros and should be programmed as zeros for future compatibility. The bit manipulation instructions are useful for accessing individual bits in the DCTR. The DCTR bits are described in the following paragraphs.

6.1.1.1 Host Command Interrupt Enable (HCIE) Bit 0

The HCIE bit is used to enable a vectored DSP56300 Core interrupt request when the host command pending (HCP) status bit in the DSP status register (DSR) is set. If HCIE is cleared, HCP interrupt requests are disabled. With HCIE set, a host command interrupt request will be generated if HCP is set. The starting address of this interrupt is determined by the host vector HV6-HV0 in the host command vector register (HCVR).

If the host non-maskable interrupt (HNMI) bit is set in the host command vector register (HCVR), HCIE is ignored, and an interrupt is generated if HCP is set regardless of HCIE.

Hardware and software resets clear HCIE.

6.1.1.2 Slave Transmit Interrupt Enable (STIE) Bit 1

The STIE bit is used to enable an DSP56300 Core interrupt request when the slave transmit data request (STRQ) status bit in the DSR is set. If STIE is cleared, STRQ interrupt requests are disabled. If STIE is set an slave transmit data interrupt request will be generated if STRQ is set.

Hardware and software resets clear STIE.

6.1.1.3 Slave Receive Interrupt Enable (SRIE) Bit 2

The SRIE bit is used to enable an DSP56300 Core interrupt request when the slave receive data request (SRRQ) status bit in the DSR is set. If SRIE is cleared, SRRQ interrupt requests are disabled. If SRIE is set, a slave receive data interrupt request will be generated if SRRQ is set.

Hardware and software resets clear SRIE.

6.1.1.4 Host Flags (HF5-HF3) Bits 5-3

The HF5-HF3 bits are used as general purpose flags for DSP-to-host communication. HF5-HF3 may be set or cleared by the DSP56300 Core. HF5-HF3 are visible to the external host in the HSTR register.

Hardware and software resets clear host flags.

> NOTE: There are six host flags: three used by the host to signal the DSP56300 Core (HF2-HF0) and three used by the DSP56300 Core to signal the host processor (HF5-HF3). These are general purpose flags. The host flags do not cause interrupts; they must be polled to see if they have changed. These flags can be used individually or as encoded triads.

6.1.1.5 Host Interrupt A (HINT) Bit 6

The HINT bit controls the $\overline{\text{HINTA}}$ pin. When HINT is set by the DSP56300 Core, the $\overline{\text{HINTA}}$ pin is driven low. When HINT is cleared by the DSP56300 Core, the $\overline{\text{HINTA}}$ pin is released.

Hardware and software resets clear HINT.

6.1.1.6 Host Data Strobe Mode (HDSM) Bit 13

The HDSM bit controls the data strobe mode of the host port pins when in a Universal Bus mode (HM = $2 or $3). If HDSM is cleared, the double-strobe pin mode is selected: $\overline{\text{HWR}}$/HRW pin (HP29) functions as host write strobe $\overline{\text{HWR}}$ and $\overline{\text{HRD}}/\overline{\text{HDS}}$ (HP30) functions as host read strobe $\overline{\text{HRD}}$. If HDSM is set, the single-strobe pin mode is selected: $\overline{\text{HWR}}$/HRW pin functions as host read/write HRW and $\overline{\text{HRD}}/\overline{\text{HDS}}$ functions as host data strobe $\overline{\text{HDS}}$.

The value of HDSM may be changed only when HACT=0 in the DSR.

HDSM is ignored when not in a Universal Bus mode (HM ≠ $2 or $3).

Hardware and software resets clear HDSM.

6.1.1.7 Host Read/Write Polarity (HRWP) Bit 14

The HRWP bit controls the polarity of $\overline{HWR}$/HRW pin, when in single-strobe Universal Bus modes (HM = $2 or $3 and HDSM=1), that is, when $\overline{HWR}$/HRW pin (HP29) functions as the host read/write (HRW) pin.

If HRWP is cleared, the host-to-DSP data transfer direction corresponds to low level of HRW pin, and DSP-to-host data transfer direction corresponds to high level of HRW pin.

If HRWP is set, the host-to-DSP data transfer direction corresponds to high level of HRW pin, and DSP-to-host data transfer direction corresponds to low level of HRW pin.

The value of HRWP may be changed only when HACT=0.

HRWP is ignored when not in a Universal Bus mode or double-strobe host port mode is selected (HM ≠ $2 or $3, or HDSM = 0).

Hardware and software resets clear HRWP.

6.1.1.8 Host Transfer Acknowledge Polarity (HTAP) Bit 15

The HTAP bit controls the polarity of the HTA pin when the HI32 is in a Universal Bus mode (HM = $2 or $3). If HTAP is cleared, the HTA pin is active high and the HI32 will request to extend the access by driving the HTA pin low (i.e. negated). If HTAP is set, the HTA pin is active low and the HI32 will request to extend the access by driving the HTA pin high (i.e. negated).

> NOTE: HTA is driven in the Universal Bus modes (HM = $2 or $3) while the HI32 is being accessed by an external host. If the HI32 is not being accessed, the HTA pin is high impedance.

The value of HTAP may be changed only when HACT=0.

HTAP is ignored when not in a Universal Bus mode (HM ≠ $2 or $3).

Hardware and software resets clear HTAP.

6.1.1.9 Host DMA Request Polarity (HDRP) Bit 16

The HDRP bit controls the polarity of HDRQ pin when the HI32 is in a Universal Bus mode (HM = $2 or $3). If HDRP is cleared, the HDRQ pin is active high and the HI32 will request DMA service by driving the HDRQ pin high (i.e. asserted). If HDRP is set, the HDRQ pin is active low and the HI32 will request DMA service by driving the HDRQ pin low (i.e. asserted).

The value of HDRP may be changed only when HACT=0.

HDRP is ignored when not in a Universal Bus mode (HM ≠ $2 or $3).

Hardware and software resets clear HDRP.

6.1.1.10 Host Reset Polarity (HRSP) Bit 17

The HRSP bit controls the polarity of HRST pin when the HI32 is in a Universal Bus or the GPIO mode (HM = $2, $3 or $4). If HRSP is cleared, the HRST pin is active high and the HI32 will be reset if the HRST pin is high (i.e. asserted). If HRSP is set, the HRST pin is active low and the HI32 will be reset if HRST pin is low (i.e. asserted).

The value of HRSP may be changed only when HACT=0.

HRSP is ignored in the PCI mode (HM = $1).

Hardware and software resets clear HRSP.

6.1.1.11 Host Interrupt Request Handshake Mode(HIRH) Bit 18

The HIRH bit controls the handshake mode of the $\overline{\text{HIRQ}}$ pin when the HI32 is in a Universal Bus mode (HM = $2 or $3). $\overline{\text{HIRQ}}$ is asserted by the HI32 when a host interrupt request (receive and/or transmit) is generated in the HI32. With HIRH cleared, when a host interrupt request is generated, $\overline{\text{HIRQ}}$ is asserted for the number of DSP56300-Core clock cycles specified by the LT7-LT0 bits in the CLAT and then negated. The duration of the $\overline{\text{HIRQ}}$ pulse is given by the following equation:

HIRQ_PULSE_WIDTH = (LT[7:0]_Value + 1) • DSP56300_Core_clock_cycle

If HIRH is set: $\overline{\text{HIRQ}}$ is negated when the interrupt request source is cleared (by the corresponding host data access), masked (by TREQ=0 or RREQ=0) or disabled by the DMA enable bit (DMAE) in the HCTR.

The value of HIRH may be changed only when HACT=0.

HIRH is ignored when not in a Universal Bus mode (HM ≠ $2 or $3).

Hardware and software resets clear HIRH.

6.1.1.12 Host Interrupt Request Drive Control (HIRD) Bit 19

The HIRD bit controls the output drive of $\overline{\text{HIRQ}}$ pin when the HI32 is in a Universal Bus mode (HM = $2 or $3). With HIRD cleared, the $\overline{\text{HIRQ}}$ pin is an open drain output (i.e. driven low when asserted, released (high impedance) when negated). With HIRD set, the $\overline{\text{HIRQ}}$ pin is always driven.

The value of HIRD may be changed only when HACT=0.

HIRD is ignored when not in a Universal Bus mode (HM ≠ $2 or $3).

Hardware and software resets clear HIRD.

> NOTE: Each of the bits HDSM, HRWP, HTAP, HDRP, HRSP, HIRH, and HIRD affect the host port pins directly. To assure proper operation, these pins may be changed only when HACT=0. The HM2-HM0 bits must not be changed together with these bits (i.e. in the same Core write).

6.1.1.13   HI32 Mode (HM2-HM0) Bits 22-20

The HM bits control the operation modes and pin functionality of the HI32 (see Table 6-1). The host port pins in the different modes are shown in Table 6-2.

Table 6-1. HI32 Modes

HM2-HM0	HI32 Mode
0 0 0	Terminate and Reset
0 0 1	PCI
0 1 0	Universal Bus
0 1 1	Enhanced Universal Bus
1 0 0	GPIO
1 0 1	Self Configuration
1 1 0	reserved
1 1 1	reserved

Terminate and Reset (HM=$0):

When HM2-HM0 is written $0:
If the HI32 was in the PCI mode (HM=$1), as an active PCI master: the HI32 generates a master initiated termination; if a selected target in a memory space transaction, the HI32 generates a target-disconnect-C/retry event, thus completing the PCI transaction. When the PCI idle state is subsequently detected, the HI32 clears HACT in the DSR and enters the personal software (PS) reset state. In the personal software reset state all data paths are cleared, and the HI32 will respond to all memory space transactions with a retry event. If the HI32 was not an active target in the PCI mode (HM≠$1) memory space transaction, the HI32 immediately clears HACT in the DSR and enters the personal software (PS) reset state.
Configuration space transactions are not affected by clearing the HM bits.
In the personal software reset the HI32 consumes very little current. This is a low-power state. For even greater power saving, the HI32 may be programmed to the GPIO mode.

PCI Mode (HM=$1):

The HI32 supports:

- Glue-less connection to the standard PCI bus.

- Operation as an initiator (master) or target (slave).

- 24- to 32-bit, 32- to 24-bit data formatting and true 32-bit (Dword) data transfers.

- Memory-space and configuration transactions as a target.

- Memory-space, I/O-space and configuration transactions as an initiator.

Universal Bus Mode (HM=$2):

The HI32 supports:

- Glue-less connection to various external buses (e.g. ISA/EISA, DSP56300 Core based DSP Port A bus).

- 24-bit, 16-bit (with data alignment) and 8-bit buses.

- ISA/EISA bus DMA-type accesses.

- Pins HP22-HP20 are general purpose I/O.

- HP19, HP31 and HP32 are unused and must be forced or pulled up to Vcc.

- When operating with a host bus less than 24 bits wide, the data pins that are not used for transferring data must be forced or pulled up or down to Vcc or to GND respectively. For example: when operating with a 16-bit bus (e.g. ISA bus), HP48-HP41 must be forced or pulled up to Vcc or pulled down to GND.

Enhanced Universal Bus Mode (HM=$3):

The HI32 supports:

- Glue-less connection to various external buses (e.g. ISA/EISA, DSP56300 Core based DSP Port A bus).

- 24-bit, 16-bit (with data alignment) and 8-bit buses.

- ISA/EISA bus DMA-type accesses.

- Two control signals (data direction and data output enable) are output to an optional external data buffer.

- Host select acknowledge output is provided.

- HP19, HP31 and HP32 are unused and must be forced or pulled up to Vcc.

- When operating with a host bus less than 24 bits wide, the data pins that are not used for transferring data must be forced or pulled up or down to Vcc or to GND respectively. For example: when operating with a 16-bit bus (e.g. ISA bus), HP48-HP41 must be forced or pulled up to Vcc or pulled down to GND.

GPIO Mode (HM=$4):

The HI32 supports:

- General purpose I/O (GPIO) port.

- Pins HP23-HP0 are GPIO.

- Pins HP48-HP33, HP30-24 are disconnected.

- HP31 and HP32 are unused and must be forced or pulled up to Vcc.

- Minimum current consumption.

Self Configuration Mode (HM=$5):

The HI32 supports:

- Indirect DSP56300 Core access to the CCMR, CLAT and CBMA HI32 configuration registers.

- All the host port pins are in the disconnected state.

The value of the HM bits may be changed to a non zero value by the DSP56300 Core only when the HI32 is in the personal software reset state (HM=$0, HACT=0), they must not be changed together (i.e. in the same Core write) with any of the following bits: HDSM, HRWP, HTAP, HDRP, HRSP, HIRH, or HIRD.

The combinations HM=$6, HM=$7 are reserved for future expansion and should not be used.

Hardware and software resets clear the HM bits.

Table 6-2. Host Port Pin Functionality

HI32 Port Pin	PCI Bus Mode	Universal Bus Mode [a] — Enhanced Universal Bus Mode	Universal Bus Mode	GPIO Mode
	HM=$1	HM=$3	HM=$2	HM=$4
HP7-0	HAD15-HAD0	HA10-HA3		HIO7-0
HP15-8		HD7-HD0		HIO15-8
HP19-16	HC3/$\overline{\text{HBE3}}$-HC0/$\overline{\text{HBE0}}$	HA2-HA0		HIO18-16
		UNUSED[b]		HIO19
HP20	$\overline{\text{HTRDY}}$	$\overline{\text{HDBEN}}$	HIO20	
HP21	$\overline{\text{HIRDY}}$	HDBDR	HIO21	
HP22	$\overline{\text{HDEVSEL}}$	$\overline{\text{HSAK}}$	HIO22	
HP23	$\overline{\text{HLOCK}}$	$\overline{\text{HBS}}$ [c]		HIO23
HP24	HPAR	$\overline{\text{HDAK}}$ [c]		disconnected
HP25	$\overline{\text{HPERR}}$	HDRQ		
HP26	$\overline{\text{HGNT}}$	HAEN		
HP27	$\overline{\text{HREQ}}$	HTA		disconnected
HP28	$\overline{\text{HSERR}}$	$\overline{\text{HIRQ}}$		
HP29	$\overline{\text{HSTOP}}$	$\overline{\text{HWR}}$/HRW		
HP30	HIDSEL	$\overline{\text{HRD}}$/$\overline{\text{HDS}}$		
HP31	$\overline{\text{HFRAME}}$	UNUSED[d]		
HP32	HCLK	UNUSED[d]		
HP40-33	HAD23-HAD16	HD15-8		disconnected
HP48-41	HAD31-HAD24	HD23-16 Output is high impedance if HRF≠$0. Input is disconnected if HTF≠$0.		disconnected
HP49	$\overline{\text{HRST}}$	HRST		
HP50	$\overline{\text{HINTA}}$			

a. When operating with a host bus less than 24 bits wide, the data pins that are not used for transferring data must be forced or pulled to Vcc or to GND.
b. Must be forced or pulled to Vcc or GND.
c. $\overline{\text{HBS}}$/$\overline{\text{HDAK}}$ should be forced or pulled up to Vcc if not used.
d. Must be forced or pulled up to Vcc.

6.1.1.14   DCTR Reserved Control Bits 23, 12-7

These bits are reserved for future expansion, they are read as zeros and should be written with zeros for upward compatibility.

6.1.2   DSP PCI Control Register (DPCR)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		IAE	RBLE	MWSD	MACE		SERF	MTT	CLRT		TCIE			TTIE		TAIE		PEIE	MAIE		MRIE	MTIE	

☐ Reserved, read as zero and should be written with zero

Bit	Name	Function
1	MTIE	Master Transmit Interrupt Enable
2	MRIE	Master Receive Interrupt Enable
4	MAIE	Master Address Interrupt Enable
5	PEIE	Parity Error Interrupt Enable
7	TAIE	Transaction Abort Interrupt Enable
9	TTIE	Transaction Termination Interrupt Enable
12	TCIE	Transfer Complete Interrupt Enable
14	CLRT	Clear Transmitter
15	MTT	Master Transfer Terminate
16	SERF	$\overline{\text{HSERR}}$ Force
18	MACE	Master Access Counter Enable
19	MWSD	Master Wait State Disable
20	RBLE	Receive Buffer Lock Enable
21	IAE	Insert Address Enable
23-22,17,13, 11-10,8,6,3,0	reserved	

The DPCR is a 24-bit read/write control register used by the DSP56300 Core to control the HI32 PCI interrupts, and interface logic. The DPCR cannot be accessed by the host processor. All reserved bits are read as zeros and should be programmed as zeros for future compatibility. The bit manipulation instructions are useful for accessing individual bits in the DPCR. The DPCR bits are described in the following paragraphs.

6.1.2.1   Master Transmit Interrupt Enable (MTIE) Bit 1

The MTIE bit is used to enable an DSP56300 Core interrupt request when the master transmit data request (MTRQ) status bit in the DPSR is set. If MTIE is cleared, MTRQ interrupt requests are disabled. If MTIE is set a master transmit data interrupt request will be generated if MTRQ is set.

Hardware and software resets clear MTIE.

6.1.2.2   Master Receive Interrupt Enable (MRIE) Bit 2

The MRIE bit is used to enable an DSP56300 Core interrupt request when the master receive data request (MRRQ) status bit in the DSP status register (DPSR) is set. If MRIE

is cleared, master receive data interrupt requests are disabled. If MRIE is set, a master receive data interrupt request will be generated if MRRQ is set.

Hardware and software resets clear MRIE.

6.1.2.3   Master Address Interrupt Enable (MAIE) Bit 4

The MAIE bit is used to enable an DSP56300 Core interrupt request when the HI32 is currently not the PCI transaction initiator, when in the PCI mode (HM=$1). If MAIE is cleared, master address interrupt requests are disabled. If MAIE is set, a master address interrupt request will be generated if the master address request (MARQ) status bit in the DPSR register is set.

Hardware and software resets clear MAIE.

6.1.2.4   Parity Error Interrupt Enable (PEIE) Bit 5

The PEIE bit is used to enable an DSP56300 Core interrupt request when a parity error is detected, when in the PCI mode (HM=$1). If PEIE is cleared, parity error interrupt requests are disabled. If PEIE is set, a parity error interrupt request will be generated if a parity error (address or data) is detected and the address parity error (APER) status bit or the data parity error (DPER) status bit in the DPSR register is set.

Hardware and software resets clear PEIE.

6.1.2.5   Transaction Abort Interrupt Enable (TAIE) Bit 7

The TAIE bit is used to enable an DSP56300 Core interrupt request when in the PCI mode (HM=$1) and the HI32, as a PCI master, has executed a master-abort termination or a target initiated target-abort termination. If TAIE is cleared, transaction abort interrupt requests are disabled. If TAIE is set, a transaction abort interrupt request will be generated if a transaction was terminated due to master-abort (MAB is set in the DPSR) or target-abort (TAB is set).

Hardware and software resets clear TAIE.

6.1.2.6   Transaction Termination Interrupt Enable (TTIE) Bit 9

The TTIE bit is used to enable an DSP56300 Core interrupt request when in the PCI mode (HM=$1) and the HI32, as a PCI master, has executed a time-out termination, or a target initiated disconnect or retry termination. If TTIE is cleared, transaction termination interrupt requests are disabled. If TTIE is set, a transaction termination interrupt request will be generated if a transaction was terminated due to a disconnect (TDIS is set in the DPSR), retry (TRTY is set) or time-out (TO is set).

Hardware and software resets clear TTIE.

6.1.2.7 Transfer Complete Interrupt Enable (TCIE) Bit 12

The TCIE bit is used to enable an DSP56300 Core interrupt request when in the PCI mode (HM=$1) and the host data transfer complete (HDTC) status bit in the DSP PCI status register (DPSR) is set. If TCIE is cleared, transfer complete interrupt requests are disabled. If TCIE is set, a transfer complete interrupt request will be generated if HDTC is set.

Hardware and software resets clear TCIE.

6.1.2.8 Clear Transmitter (CLRT) Bit 14

The CLRT bit is used to clear the HI32 master-to-host bus data path in the PCI mode (HM=$1). When CLRT is set by the DSP56300 Core, the HI32 hardware clears the master DSP-to-host bus data path (i.e. the DTXM-HRXM FIFO is forced empty) - thus setting the PCI Master Transmit Data Request bit (MTRQ) in the DPSR, and then clears CLRT. CLRT cannot be written zero by the DSP56300 Core.

To assure operation, CLRT may be set by the DSP56300 Core, only if

1. MARQ is set in the DPSR (i.e. the DSP56300 Core has not initiated a PCI transaction); and
2. No DSP56300 Core DMA channel is enabled to service HI32 master transmit data DMA requests.

CLRT is ignored when the HI32 is not in the PCI mode (HM≠$1).

Hardware and software resets clear CLRT.

6.1.2.9 Master Transaction Termination (MTT) Bit 15

The MTT bit is used for the generation of a PCI master initiated transaction termination. When the HI32, in the PCI mode (HM=$1), is the active PCI master, if MTT is set, by the DSP56300 Core, a master initiated transaction termination (not master-abort) is generated. MTT is cleared by the HI32 hardware when the PCI bus is in the idle state. MTT cannot be written zero by the DSP56300 Core.

MTT is ignored when the HI32 is not in the PCI mode (HM≠$1).

Hardware and software resets clear MTT.

6.1.2.10 System Error Force (SERF) Bit 16

The SERF bit controls $\overline{\text{HSERR}}$ pin state in the PCI mode (HM=$1). When SERF is set by the DSP56300 Core and the HI32 is the current PCI bus master or a selected target, the $\overline{\text{HSERR}}$ pin is pulsed one PCI clock cycle, if the system error enable (SERE) bit is set in the status/command configuration register (CSTR/CCMR); the signalled system error (SSE) bit is set in the CSTR/CCMR. SERF is cleared by the HI32 hardware after $\overline{\text{HSERR}}$ is asserted. If SERF is cleared, the $\overline{\text{HSERR}}$ pin is controlled by the HI32 hardware (see

$\overline{\text{HSERR}}$ pin definition in Table 6-2, on page 6-16). SERF cannot be written zero by the DSP56300 Core.

SERF is ignored when the SERE bit is cleared or when the HI32 is not an active PCI agent (i.e. HM≠$1 or the HI32 is not the current PCI bus master or a selected target).

Hardware and software resets clear SERF.

6.1.2.11   Master Access Counter Enable (MACE) Bit 18

The MACE bit is used to enable the master access counter.

If MACE is set, the master access counter is enabled and the HI32 as the active PCI master (HM=$1) will terminate the current PCI transaction when the counter reaches the terminal count.

If MACE is cleared, the counter is disabled, and the burst length of transactions initiated by the HI32 are unlimited.

The DSP56300 Core can terminate a transaction initiated by the HI32 by writing one to the MTT bit in the DPCR.

MACE is ignored when the HI32 is not in the PCI mode (HM≠$1).

The value of MACE may be changed only if MARQ=1, or HACT=0.

Hardware and software resets clear MACE.

6.1.2.12   Master Wait State Disable (MWSD) Bit 19

The MWSD bit is used to disable PCI wait states (inserted by negating $\overline{\text{HIRDY}}$), during a data phase.

If MWSD is cleared, the HI32 as the active PCI master (HM=$1) will insert wait states to extend the current data phase if it cannot guarantee the completion of the next data phase. The HI32 will assert $\overline{\text{HIRDY}}$ and complete the current data phase if:

- it can complete the next data phase, or

- it has determined to terminate the transaction due to time-out or completion.

If MWSD is set, the HI32, as the active PCI master (HM=$1) will not insert wait states. If it cannot guarantee the completion of the next data phase, the HI32 will complete the current data phase and terminate the transaction.

MWSD is ignored when the HI32 is not in the PCI mode (HM≠$1).

The value of MWSD may be changed only when HACT=0.

Hardware and software resets clear MWSD.

6.1.2.13   Receive Buffer Lock Enable (RBLE) Bit 20

The RBLE bit is used, in the PCI mode (HM=$1), to assure that the host-to-DSP data path contains data from only one external master at any time. This is accomplished by inhibiting the HI32 from responding to new PCI write transactions to the HTXR until the DSP56300 Core has read all the data written to the HTXR in the last access.

With RBLE set: After a non-exclusive write transaction to the HTXR, or upon $\overline{\text{HLOCK}}$ negation after completion of an exclusive write access to the HTXR; or after the completion of a read transaction initiated by the HI32:

- Forthcoming PCI write accesses to the HTXR will be disconnected (retry or disconnect-C) until the DSP56300 Core writes one to the host data transfer complete (HDTC) bit in the DPSR.
- If the host-to-DSP data path is empty (SRRQ=0 and MRRQ=0), due to DSP56300 Core reads from the DRXR, the HDTC bit will be set. The HI32 will disconnect (retry or disconnect-C) all PCI write accesses to the HTXR until the DSP56300 Core writes one to the HDTC bit to clear it.

If RBLE is cleared the HI32 will not set the HDTC bit.

If the HDTC bit is cleared the HI32 will respond to write PCI transactions according to the status of the host-to-DSP data path.

RBLE is ignored when the HI32 is not in the PCI mode (HM≠$1).

The value of RBLE may be changed only when HACT=0 or HDTC=1.

Hardware and software resets clear RBLE.

6.1.2.14   Insert Address Enable (IAE) Bit 21

The IAE bit is used, in the PCI mode (HM=$1), to insert the PCI transaction address at the head of the incoming data stream in accordance with the value of the host data transfer format (HTF) bits in the HCTR.
If IAE is set the HI32 writes the PCI transaction address to the HTXR, before the data written by the host, if the HI32 is being accessed in a write transaction.
If HTF=$0 (32-bit mode): first, the two least significant bytes of the PCI transaction address are written to the two least significant bytes of the HTXR, then the two most significant bytes of the PCI transaction address (the address is inserted as $00HHHH, $00LLLL, where HHHH = HAD[31:16] and LLLL = HAD[15:0]).
If HTF≠$0: only the two least significant bytes of the PCI transaction address are written to the two least significant bytes of the HTXR (the address is inserted as $00LLLL, where LLLL = HAD[15:0]).
The incoming data is written to the HTXR after the address.

IAE is ignored when the HI32 is not in the PCI mode (HM≠$1).

The value of IAE may be changed only when HACT=0 or HDTC=1.

Hardware and software resets clear IAE.

6.1.2.15 DPCR Reserved Control Bits 23,22,17,13,11,10,8,6,3,0

These bits are reserved for future expansion, they are read as zeros and should be written with zeros for upward compatibility.

6.1.3 DSP PCI Master Control Register (DPMC)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FC1	FC0	BL5	BL4	BL3	BL2	BL1	BL0	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16

Bit	Name	Function
15-0	AR31-AR16	DSP PCI Transaction Address (High)
21-16	BL5-BL0	PCI Data Burst Length
23-22	FC1-FC0	Data Transfer Format Control

The DPMC is a 24-bit read/write register used by the DSP56300 Core to generate the two most significant bytes of the 32-bit PCI transaction address, and to control the burst length and the data transfer format. The DPMC cannot be accessed by the host processor.

The DPMC may be written only if MARQ is set or in the Self Configuration mode. (See Section 6.1 on page 6-8).

The DPMC bits are described in the following paragraphs.

6.1.3.1 DSP PCI Transaction Address (AR31-AR16) Bits 15-0

The AR31-AR16 bits are the two most significant bytes of the 32-bit PCI transaction address. The two least significant bytes of the PCI transaction address are located in the DPAR register (see Section 6.1.4). When the DPAR is written by the DSP56300 Core, while in the PCI mode (HM=$1), the PCI ownership is requested and, when granted, the HI32 will initiate a PCI transaction. The full 32-bit address (AR31-AR16 from the DPMC and AR15-AR0 from the DPAR) is driven to the HAD31-HAD0 pins during the PCI address phase.

Hardware and software resets clear AR31-AR16.

6.1.3.2 DSP PCI Data Burst Length (BL5-BL0) Bits 21-16

The BL5-BL0 bits control the PCI data burst length, the value of the BL5-BL0 bits being the desired number of accesses in the burst, minus one. When the DPAR is written by the DSP56300 Core, while in the PCI mode (HM=$1), the master access counter is initialized with the value of BL5-BL0. The burst length may be programmed from 1 (BL=$00) to 64

(BL=$3F) accesses.

If the master access counter is enabled (MACE=1 in the DPCR) and the HI32 is the active PCI master, the value of the counter is decremented after each data cycle in which data is transferred (i.e. a data phase), until a value of $00 is reached. When the counter value reaches $00, the HI32 PCI master will execute one more data phase and terminate the transaction. A transaction may be terminated before the counter reaches $00 (e.g. a target initiated transaction termination, or the bus grant was taken, or the DSP56300 Core wrote one to MTT). The value of the counter at the end of a transaction is indicated by the RDC5-RDC0 bits in the DSP PCI status register (DPSR).

Hardware and software resets clear BL5-BL0.

6.1.3.3   DSP Data Transfer Format Control (FC1-FC0) Bits 23 and 22

The FC1-FC0 bits define data transfer formats between the HI32 and a PCI agent when in the PCI mode (HM=$1) and the HI32 is a bus master. The data transfer format converter (see Figure 6-1 on page 6-7) operates according to the specified FC1-FC0 (see Table 6-3).

In a PCI DSP-to-host transaction:

If FC=$0 (32-bit data mode):
The two least significant bytes of the first word written to the DTXM and the two least significant bytes of the second word written to the DTXM are output to the HAD31-HAD0 pins. HAD[31:0] = $HHHHLLLL, where LLLL are the two least significant bytes of the first word written to the DTXM, and HHHH are the two least significant bytes of the second word written to the DTXM.

If FC=$1:
The data written to the DTXM is output to the HAD31-HAD0 pins as right aligned and zero extended in the most significant byte.

If FC=$2:
The data written to the DTXM is output to the HAD31-HAD0 pins as right aligned and sign extended in the most significant byte.

If FC=$3:
The data written to the DTXM is output to the HAD31-HAD0 pins as left aligned and zero filled in the least significant byte.

In a PCI host-to-DSP transaction:

If FC=$0 (32-bit data mode):
The two least significant bytes PCI data bytes from the HAD15-HAD0 pins are transferred to the two least significant bytes of the DRXR after which the two most significant bytes, from the HAD32-HAD16 pins, are transferred to the two least significant bytes of the DRXR. Thus, when the DSP56300 Core reads two words from the DRXR, the two least

Table 6-3. HI32 (PCI Master) Data Transfer Formats

FC1	FC0	DSP to PCI Host Data Transfer Format	PCI Host to DSP Data Transfer Format
0	0	The two least significant bytes of two HRXM locations are output.	All 32 PCI data bits are written to the HTXR as two zero extended 16-bit words.
0	1	The three least significant HRXM bytes are output right aligned and zero extended.	The three least significant PCI data bytes are written to the HTXR.
1	0	The three least significant HRXM bytes are output right aligned and sign extended.	The three least significant PCI data bytes are written to the HTXR.
1	1	The three least significant HRXM bytes are output left aligned and zero filled.	The three most significant PCI data bytes are written to the HTXR.

significant bytes of the first word read contain the two least significant bytes of the 32-bit

word written to the HTXR, the two least significant bytes of the second word read contain the two most significant bytes of the 32-bit word.

If FC=$1 or $2:
The three least significant PCI data bytes from the HAD23-HAD0 pins are transferred to the DRXR to be read by the DSP56300 Core.

If FC=$3:
The three most significant PCI data bytes from the HAD31-HAD8 pins are transferred to the DRXR to be read by the DSP56300 Core.

To assure proper operation: FC1-FC0 may be changed only if both the host-to-DSP and the DSP-to-host master data paths are empty. In addition, switching between 32-bit data modes and non-32-bit data modes may be done only in the personal software reset state (HM=$0 and HACT=0).

FC1-FC0 are ignored when not in the PCI mode (HM≠$1).

The DPMC bits are ignored when not in the PCI mode (HM≠$1).

Hardware and software resets clear FC1-FC0.

6.1.4   DSP PCI Address Register (DPAR)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BE3	BE2	BE1	BE0	C3	C2	C1	C0	AR 15	AR 14	AR 13	AR 12	AR 11	AR 10	AR 9	AR 8	AR 7	AR 6	AR 5	AR 4	AR 3	AR 2	AR 1	AR 0

Bit	Name	Function
15-0	AR15-AR0	DSP PCI Transaction Address (Low)
19-16	C3-C0	PCI Bus Command
23-20	$\overline{BE3}$-$\overline{BE0}$	PCI Byte Enables

The DPAR is a 24-bit read/write register used by the DSP56300 Core to generate the two least significant bytes of the 32-bit PCI transaction address, the PCI bus command and the PCI bus byte enables. The DPAR cannot be accessed by the host processor. The two most significant bytes of the PCI transaction address are located in the DSP PCI master control register (DPMC, see Section 6.1.3).

When the DPAR is written by the DSP56300 Core, while in the PCI mode (HM=$1),
- MARQ is cleared,
- when the HI32 can complete the first data phase (i.e. in a write transaction, the DSP-to-host data path is not empty; in a read transaction, the host-to-DSP data path is not full) ownership of the PCI bus is requested and when granted
- the address (from the DPMC and the DPAR) is driven to the HAD31-HAD0 pins and the bus command is driven to the HC3/$\overline{HBE3}$-HC0/$\overline{HBE0}$ pins during the PCI address phase.

The DPAR may be written only if MARQ is set.

In memory space accesses, the AR1-AR0 bits have the following meaning:

AR1	AR0	Burst Order
0	0	Linear incrementing
0	1	PCI Cache line toggle mode (the data must be arranged by the DSP software)
1	X	Reserved

The DPAR bits are ignored when not in the PCI mode (HM≠$1).

Hardware and software resets clear A15-A0.

6.1.4.1 PCI Bus Command (C3-C0) Bits 11-8

The C3-C0 define the PCI bus command. PCI bus commands supported by the HI32 as a PCI master are listed in Table 6-4. When the DPAR is written by the DSP56300 Core, while the HI32 is in the PCI mode (HM=$1), ownership of the PCI bus is requested and, when granted, the address is driven to the HAD31-HAD0 pins and the bus command is driven to the HC3/$\overline{HBE3}$-HC0/$\overline{HBE0}$ pins during the PCI address phase.

Table 6-4. PCI Bus Commands Supported by the HI32 as PCI Master

C3-C0	Command Type
0000	illegal
0001	illegal
0010	**I/O read**
0011	**I/O Write**
0100	illegal
0101	illegal
0110	**Memory Read**
0111	**Memory Write**
1000	illegal
1001	illegal
1010	**Configuration Read**
1011	**Configuration Write**
1100	**Memory Read Multiple**
1101	illegal
1110	**Memory Read Line**
1111	illegal

Illegal C3-C0 values are not supported by the HI32 and should not be used.

Hardware and software resets clear C3-C0.

6.1.4.2　　PCI Byte Enables ($\overline{BE3}$-$\overline{BE0}$) Bits 15-12

The $\overline{BE3}$-$\overline{BE0}$ determine which byte lanes carry meaningful data when in the PCI mode (HM=$1) and the HI32 is a PCI master. $\overline{BE3}$ applies to byte 3, and $\overline{BE0}$ to byte 0. Byte enables are driven to HC3/$\overline{HBE3}$-HC0/$\overline{HBE0}$ pins during the PCI data phases.

The HI32, as master, drives all the HRXM data to the HAD31-HAD0 pins during write transactions, and writes the HAD31-HAD0 pins to the HTXR (in accordance with the FC1-FC0 bits) in read transactions, regardless of the BE3-BE0 value.

Hardware and software resets clear $\overline{BE3}$-$\overline{BE0}$.

6.1.5   DSP Status Register (DSR)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HACT																		HF2	HF1	HF0	SRR Q	STR Q	HCP
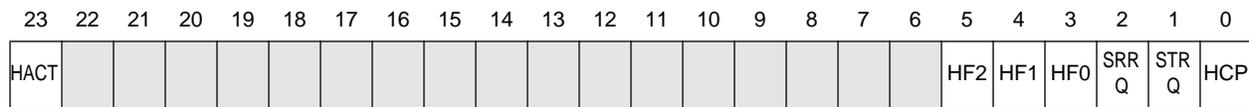
☐   Reserved, read as zero

Bit	Name	Function
0	HCP	Host Command Pending
1	STRQ	Slave Transmit Data Request
2	SRRQ	Slave Receive Data Request
5-3	HF2-HF0	Host Flags
23	HACT	HI32 Active
22-6	reserved	

The DSR is a 24-bit read-only status register used by the DSP56300 Core to examine the status and flags of the HI32. The DSR cannot be accessed by the host processor. The DSR bits are described in the following paragraphs.

6.1.5.1   Host Command Pending (HCP) Bit 0

The HCP bit indicates that the host has set the HC bit and that a host command interrupt is pending. The HCP bit reflects the status of the HC bit in the HCVR. If HCP is set and HCIE is set, a host command interrupt request is generated. HC and HCP are cleared by the HI32 interrupt logic hardware when the HC interrupt request is serviced. The host cannot clear HC.

The personal software reset clears HCP.

6.1.5.2   Slave Transmit Data Request (STRQ) Bit 1

The STRQ bit indicates that the slave transmit data FIFO (DTXS) is not full and may be written by the DSP56300 Core. STRQ functions in accordance with the value of the slave fetch type (SFT) bit in the host control register (HCTR).

In the Fetch mode: the HI32 requests data from the DSP56300 Core (by enabling the STRQ status bit and generating Core interrupt requests or DMA requests if enabled), only after the host has begun a read transaction from the HI32.

In the Pre-Fetch mode: the HI32 requests data from the DSP56300 Core (by enabling the STRQ status bit and generating Core interrupt requests or DMA requests if enabled) whenever the DTXS is not full.

In the PCI mode (HM=$1):

Fetch (SFT= 1): The DSP-to-host data path is a six word deep (three word deep in the

32-bit data format mode, HRF=$0) FIFO buffer. During a read transaction from the DTXS-HRXS FIFO, STRQ reflects the status of the DTXS: STRQ is set if the DTXS is not full. STRQ is cleared when the DSP56300 Core fills the DTXS. If the host is not executing a read transaction from the HRXS, the DSP-to-host data path is forced to the reset state and STRQ is cleared.

In a Universal Bus mode (HM= $2 or $3):

Fetch (SFT= 1): There is no FIFO buffering of the DSP-to-host data path. At the beginning of a read data transfer from the HRXS, STRQ is set. STRQ is cleared when the DSP56300 Core writes to the DTXS. If the host is not reading from the HRXS, the DSP-to-host data path is forced to the reset and STRQ is cleared.

In both the PCI and Universal Bus modes (HM=$1, $2 or $3):

Pre-fetch (SFT= 0): The DSP-to-host data path is a six word deep (three word deep in the 32-bit data format mode, HM=$1 and HRF=$0) FIFO buffer. STRQ reflects the status of the DTXS: STRQ is set if the DTXS is not full. STRQ is cleared when the DSP56300 Core fills the DTXS.

If STRQ is set

- if STIE is set, a slave transmit data interrupt request is generated

- if enabled by an DSP56300 Core DMA channel, a slave transmit data DMA request will be generated.

Hardware, software and personal software resets set STRQ. In the personal software reset state STRQ =0.

6.1.5.3   Slave Receive Data Request (SRRQ) Bit 2

The SRRQ bit indicates that the receive data FIFO (DRXR) contains data written by the host processor to the HI32 slave. When an external host writes data to the host-to-DSP FIFO (HTXR-DRXR), SRRQ is set. SRRQ is cleared if the DRXR is emptied by DSP56300 Core reads; or the data to be read from the DRXR is master data.

If SRRQ is set

- if SRIE is set, a slave receive data interrupt request is generated

- if enabled by an DSP56300 Core DMA channel, a slave receive data DMA request will be generated.

Hardware, software and personal software resets clear SRRQ.

6.1.5.4   Host Flags (HF2-HF0) Bits 5-3

The HF2-HF0 bits in the DSR indicate the state of host flags HF2-HF0 respectively, in the

host control register (HCTR) on the host side. HF2-HF0 can only be changed, albeit indirectly, by the host processor.

In the PCI mode (HM=$1) the HF2-HF0 bits are updated at the end of a transaction.

> NOTE: A potential problem exists when reading the status bits HF2-HF0 as an encoded triad. During personal hardware reset these bits are cleared asynchronously. For example: If HF2-HF0 change from 111 to 000, there is a small probability the DSP56300 Core could read the bits during transition and receive 001 or 110 or other combinations instead of 000. This problem can be avoided if the DSP56300 Core reads these bits twice and checks for consensus.

The personal hardware reset clears HF2-HF0.

6.1.5.5   HI32 Active (HACT) Bit 23

The HACT bit indicates the activity of the HI32. The HACT is cleared in response to HM=$0 (Terminate and Reset) and set by HM = $1, $2, $3, $5.

HACT is cleared in response to Terminate and Reset (HM=$0):

- If HM=$0 is written (Terminate and Reset), while the HI32 is an active PCI bus master or selected target in a memory space transaction, a master initiated termination or target disconnect, respectively, is generated. When the PCI idle state is detected, the HACT status bit in the DSR is cleared.

- If HM=$0 is written (Terminate and Reset), while the HI32 is in a Universal Bus or Self Configuration mode (HM=$2, $3 or $5), the HACT status bit in the DSR is cleared immediately.

When HACT is set, the HI32 is active, and the DCTR mode and polarity bits must NOT be changed.

Hardware and software resets clear HACT.

6.1.5.6   DSR Reserved Status Bits 22-6

These bits are reserved for future expansion and read as zeros.

6.1.6   DSP PCI Status Register (DPSR)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		RDC 5	RDC 4	RDC 3	RDC 2	RDC 1	RDC 0				HDT C	TO	TRTY	TDIS	TAB	MAB	DPE R	APER	MAR Q		MRR Q	MTR Q	MWS

☐   Reserved, read as zero

Bit	Name	Function
0	MWS	PCI Master Wait States
1	MTRQ	PCI Master Transmit Data Request
2	MRRQ	PCI Master Receive Data Request
4	MARQ	PCI Master Address Request
5	APER	PCI Address Parity Error
6	DPER	PCI Data Parity Error
7	MAB	PCI Master Abort
8	TAB	PCI Target Abort
9	TDIS	PCI Target Disconnect
10	TRTY	PCI Target Retry
11	TO	PCI Time Out Termination
12	HDTC	PCI Host Data Transfer Complete
21-16	RDC5-RDC0	Remaining Data Count
23,22,15-13,3	reserved	

The DPSR is a 24-bit read-only status register used by the DSP56300 Core to examine the status and flags of the HI32, when in the PCI mode (HM=$1). The DPSR cannot be accessed by the host processor. The DPSR bits are described in the following paragraphs.

6.1.6.1   PCI Master Wait State (MWS) Bit 0

The MWS bit indicates that the HI32, as master in a PCI transaction, will insert wait states (if enabled, i.e. the MWSD bit in the DPCR is cleared) to extend the current data phase (or the first data phase if the transaction has not been initiated yet) by negated $\overline{\text{HIRDY}}$ as it cannot guarantee completion of the next data phase.

MWS is set:
- In a PCI write transaction, if there is only one word in the HI32-to-host data path.
- In a PCI read transaction, if there is only one empty location in the host-to-DSP data path.

This has many applications. For example, the DSP56300 Core can set MTT, when MWS is set, to terminate a transaction after the transfer of a specific number of words. After MTT is set the HI32 will complete the data phase and terminate the transaction.

Hardware, software and personal software resets clear MWS.

6.1.6.2   PCI Master Transmit Data Request (MTRQ) Bit 1

The MTRQ bit indicates that the DSP master transmit data FIFO (DTXM) is not full and can be written by the DSP56300 Core. MTRQ is cleared if the DTXM is filled by DSP56300 Core writes. MTRQ is set when data is output from the DTXM-HRXM FIFO to the host bus.

If MTRQ is set

- if MTIE is set, a master transmit data interrupt request is generated

- if enabled by an DSP56300 Core DMA channel, a master transmit data DMA request will be generated.

Hardware, software and personal software resets set MTRQ. In the personal software reset state MTRQ = 0.

6.1.6.3   PCI Master Receive Data Request (MRRQ) Bit 2

The MRRQ bit indicates that the DSP receive data FIFO (DRXR) contains data read from the host bus by the HI32 master. When the HI32, as master, reads data from the host bus to the host-to-DSP FIFO (HTXR-DRXR), MRRQ is set. MRRQ is cleared if the DRXR is emptied by DSP56300 Core reads; or the data to be read from the DRXR is slave data.

If MRRQ is set

- if MRIE is set, a master receive data interrupt request is generated

- if enabled by an DSP56300 Core DMA channel, a master receive data DMA request will be generated.

Hardware, software and personal software resets clear MRRQ.

6.1.6.4   Master Address Request (MARQ) Bit 4

The MARQ bit indicates that the HI32 is currently not the initiator of a PCI transaction and the DPAR can be written with the address of the next transaction. When the HI32 with the PCI bus master enable bit (BM) set in the CCMR, is first programmed to the PCI mode (HM=$1) or completes a PCI transaction as a master, MARQ is set and, if MAIE is set, a master address interrupt request is generated. MARQ is cleared by any of the following:

- the DSP56300 Core writes the DPAR

- the PCI bus master enable bit (BM) is cleared in the CCMR

Hardware, software, personal hardware and personal software resets clear MARQ.

6.1.6.5 Address Parity Error (APER) Bit 5

The APER bit indicates that an address parity error has been detected by the HI32 hardware, when in the PCI mode (HM=$1) and the HI32 is a PCI target. At the end of a transaction, if an address parity error has been detected, APER is set and, if PEIE is set, a parity error interrupt request is generated.

If an address parity error has been detected:

- the HI32 target claims the cycles and terminates as though the address was correct.
- if the system error enable (SERE) bit in the status/command configuration register (CSTR/CCMR) is set, the $\overline{\text{HSERR}}$ pin is pulsed one PCI clock cycle, and the signalled system error (SSE) bit is set in the CSTR/CCMR.
- the detected parity error bit (DPE) in the CSTR is set.

APER is cleared when it is written one by the DSP56300 Core.

In personal software reset APER does not reflect new address parity errors.

Hardware and software resets clear APER.

6.1.6.6 Data Parity Error (DPER) Bit 6

The DPER bit indicates that a data parity error has been detected (by the HI32 hardware, or reported by the external host ($\overline{\text{HPERR}}$ asserted)), when in the PCI mode (HM=$1) and the HI32 is a PCI master or selected target. At the end of a transaction, if a data parity error has been detected, DPER is set and, if PEIE is set, a parity error interrupt request is generated. DPER is cleared when it is written one by the DSP56300 Core.

In personal software reset DPER does not reflect new data parity errors.

Hardware and software resets clear DPER.

6.1.6.7 Master Abort (MAB) Bit 7

The MAB bit indicates that a PCI transaction, initiated by the HI32, was terminated with master abort. When a PCI transaction initiated by the HI32 is terminated with master abort, MAB is set and, if TAIE is set, a transaction abort interrupt request is generated. MAB is cleared when written one by the DSP56300 Core.
If a PCI transaction, initiated by the HI32, was terminated with master abort, the received master abort bit (RMA) in the CSTR is also set.

Hardware and software resets clear MAB.

6.1.6.8 Target Abort (TAB) Bit 8

The TAB bit indicates that a PCI transaction, initiated by the HI32, was terminated with target abort. When a PCI transaction initiated by the HI32 is terminated with target abort,

TAB is set and, if TAIE is set, a transaction abort interrupt request is generated. TAB is cleared when written one by the DSP56300 Core.
If a PCI transaction, initiated by the HI32, was terminated with target abort, the received target abort bit (RTA) in the CSTR is also set.

Hardware and software resets clear TAB.

6.1.6.9   Target Disconnect (TDIS) Bit 9

The TDIS bit indicates that a PCI transaction, initiated by the HI32, was terminated with a target initiated disconnect. When a PCI transaction initiated by the HI32 is terminated with disconnect, TDIS is set and, if TTIE is set, a transaction termination interrupt request is generated. TDIS is cleared when written one by the DSP56300 Core.

Hardware and software resets clear TDIS.

6.1.6.10   Target Retry (TRTY) Bit 10

The TRTY bit indicates that a PCI transaction, initiated by the HI32, was terminated with a target initiated retry. When a PCI transaction initiated by the HI32 is terminated with retry, TRTY is set and, if TTIE is set, a transaction termination interrupt request is generated. TRTY is cleared when written one by the DSP56300 Core.

Hardware and software resets clear TRTY.

6.1.6.11   PCI Time Out (TO) Bit 11

The TO bit indicates that a PCI transaction, initiated by the HI32, was terminated due to the negation of the bus grant after the latency timer had expired. When a PCI transaction initiated by the HI32 is terminated due to time-out, TO is set and, if TTIE is set, a transaction termination interrupt request is generated. TO is cleared when written one by the DSP56300 Core.

Hardware and software resets clear TO.

6.1.6.12   Host Data Transfer Complete (HDTC) Bit 12

With the receive buffer lock enable (RBLE) bit in the DSP PCI control register (DPCR) set: the HDTC bit indicates that the host-to-DSP data path is empty.
HDTC is set if SRRQ and MRRQ are cleared (i.e. the host-to-DSP data path is emptied by DSP56300 Core reads) after the termination or completion a non-exclusive PCI write transaction to the HTXR, or the negation of $\overline{HLOCK}$ after the completion of an exclusive write access to the HTXR, or after a read transaction initiated by the HI32. The HI32 will disconnect (retry or disconnect-C) forthcoming write accesses to the HTXR as long as HDTC is set.
HDTC is cleared when written one by the DSP56300 Core. HDTC may be written one by the DSP56300 Core only if it is set.
If the HDTC bit is cleared the HI32 will respond to write PCI transactions according to the

status of the host-to-DSP data path.

Hardware, software and personal software resets clear HDTC.

> NOTE: Each of the bits APER, DPER, MAB, TAB, TDIS, TRTY, TO and HDTC are cleared by writing one to the specific bit. In order to assure that only the desired bit is cleared, the programmer should not use the BSET command. The proper way to clear these bits is to write (MOVE(P) instruction) ones to the bits to be cleared and zeros to all the others.

6.1.6.13   Remaining Data Count (RDC5-RDC0) Bits 21-16

The read-only bits, RDC5-RDC0, indicate the PCI data phases remaining to complete a PCI burst after the HI32 has completed a transaction as a PCI master. The RDC5-RDC0 bits are updated each time a transaction is terminated with the HI32 as a PCI master (MARQ=1).
If the transaction terminated normally, the value of RDC5-RDC0 will be $00 and TO=0, TRTY=0, TDIS=0, TAB=0, MAB=0.
If the master access counter was enabled and the burst was not completed for any reason (typical examples being: the target initiated transaction termination or the HI32 was required to generate a master initiated time-out transaction termination), the value of RDC5-RDC0 will be the remaining number of data phases remaining to complete the burst minus one (i.e. RDC=$2 signifies that there remain three more words to be transferred to complete the burst). The length of the burst is limited by BL5-BL0 in the DPMC.

6.1.6.14   DPSR Reserved Bits 23-22, 15-12 and 3

These bits are reserved for future expansion and are read as zeros.

6.1.7   Host To DSP Data Path

In PCI master data transfers (HM=$1) with FC≠$0, the host-to-DSP data path is a six word deep, 24-bit wide FIFO. The host data is read into the host side of the FIFO (HTXR) as 24-bit words, and the DSP56300 Core reads 24-bit words from the DSP side (DRXR).

In PCI master data transfers (HM=$1) with FC=$0, and PCI target data transfers (HM=$1) with HTF=$0, the host-to-DSP data path operates as a three word deep, 32-bit wide FIFO. The host data is read into the HTXR as 32-bit words, and the DSP56300 Core reads from the DRXR 24-bit words. Each word read by the DSP56300 Core contains 16-bits of data, right aligned and zero extended. The first word read by the DSP56300 Core contains the two least significant bytes of the 32-bit word read into the HTXR. The second word read by the DSP56300 Core contains the two most significant bytes of the 32-bit word read into the HTXR.

In PCI target data transfers (HM=$1) with HTF≠$0 the host-to-DSP data path is a six word deep, 24-bit wide FIFO. The host writes 24-bit words to the HTXR, and the DSP56300 Core reads 24-bit words from the DRXR.

In Universal Bus mode data transfers, the host-to-DSP data path is a five word deep, 24-bit wide FIFO. The host writes 24-bit words to the HTXR, and the DSP56300 Core reads 24-bit words from the DRXR.

The DSP side of the host-to-DSP data FIFO is described below. For a detailed description of the host side see Section 6.2.6 on page 6-62.

6.1.8   DSP Receive Data FIFO (DRXR)

The 24-bit wide DSP receive data register (DRXR) is the output stage of the host-to-DSP data path FIFO used for host-to-DSP data transfers.

The DRXR contains master data (i.e. data read by the HI32 as PCI master from an external target) to be read if MRRQ is set in the DPSR. MRRQ is cleared if the data in the DRXR is slave data or when the host-to-DSP data path FIFO is emptied by DSP56300 Core reads. The DSP56300 Core may set the MRIE bit to cause a host receive data interrupt when MRRQ is set.

The DRXR contains slave data (i.e. data written to the HI32 from the host bus) to be read if SRRQ is set in the DSR. SRRQ is cleared if the data in the DRXR is master data or when the host-to-DSP data path FIFO is emptied by DSP56300 Core reads. The DSP56300 Core may set the SRIE bit to cause a host receive data interrupt when SRRQ is set.

In the 32-bit mode (HM=$1 with FC=$0 or HTF=$0), only the two least significant bytes contain data, the most significant byte is read as zeroes. (See Table 6-3, on page 6-24 and Table 6-8, on page -51).

Hardware, software and personal software resets empty the host-to-DSP data path FIFO (SRRQ and MRRQ are cleared).

6.1.9   DSP To Host Data Path

In PCI master data transfers (HM=$1) with FC$\neq$$0, the master DSP-to-host data path (DTXM-HRXM) is an eight word deep FIFO. The DSP56300 Core writes to the DSP side of the FIFO (DTXM). The data is output to the bus from the host side (HRXM).

In PCI master data transfers (HM=$1) with FC=$0, the master DSP-to-host data path is a four word deep, 32-bit wide FIFO. The DSP56300 Core writes 24-bit words to the DTXM. Each word written by the DSP56300 Core contains 16-bits of significant data, right aligned, the most significant byte is not transmitted. The first word written by the DSP56300 Core contains the two least significant bytes of the 32-bit word to be output from the HRXM. The second word written by the DSP56300 Core contains the two most significant bytes of the 32-bit word be output from the HRXM. Each time a 32-bit word is output from the HRXM, the 32-bits of significant data located in two words written to the DTXM are output.

In PCI target data transfers (HM=$1) with HRF$\neq$$0 and in Universal Bus mode data transfers, the slave DSP-to-host data path (DTXS-HRXS) is a six word deep FIFO. The

DSP56300 Core writes 24-bit words to the DTXS. The data is output, a word at a time, to the bus from the HRXS.

In PCI target data transfers (HM=$1) with HRF=$0, the slave DSP-to-host data path is a three word deep, 32-bit wide FIFO. The DSP56300 Core writes 24-bit words to the DTXS. Each word written by the DSP56300 Core contains 16-bits of significant data, right aligned, the most significant byte is not transmitted. The first word written by the DSP56300 Core contains the two least significant bytes of the 32-bit word to be output from the HRXS. The second word written by the DSP56300 Core contains the two most significant bytes of the 32-bit word be output from the HRXS. Each time the host reads a 32-bit word from the HRXS, the 32-bits of significant data located in two locations of the slave DSP-to-host data path (DTXS and HRXS) are output.

The DSP side of the DSP-to-host data FIFOs are described in the following pages. For a detailed description of the host side see Section 6.2.4 on page 6-61 and Section 6.2.5 on page 6-62.

6.1.10   DSP Master Transmit Data Register (DTXM)

The 24-bit wide DSP master transmit data register (DTXM) is the input stage of the master DSP-to-host data path FIFO used for DSP-to-host master data transfers in the PCI mode (HM=$1).

The DTXM may be written if the MTRQ bit is set in the DPSR. Data should not be written to the DTXM until MTRQ is set to prevent previous data from being overwritten. Filling the DTXM by DSP56300 Core writes (MOVE(P) instructions or DMA transfers) clears MTRQ. The DSP56300 Core may set the MTIE bit to cause a host receive data interrupt when MTRQ is set.

In the PCI mode (HM=$1) the DSP56300 Core can clear the HI32 master-to-host bus data path and empty DTXM by setting the CLRT bit in the DPCR.

In the 32-bit mode (HM=$1 with FC=$0), only the two least significant bytes of the DTXM are transferred. (See Section 6.1.9, above, and Table 6-3, on page 6-24).

Hardware, software and personal software resets empty the DTXM.

6.1.11   DSP Slave Transmit Data Register (DTXS)

The 24-bit wide DSP slave transmit data register (DTXS) is the input stage of the slave DSP-to-host data path FIFO used for DSP-to-host slave data transfers in the PCI mode (HM=$1).

The DTXS may be written if the STRQ bit is set in the DSR. Data should not be written to the DTXS until STRQ is set to prevent previous data from being overwritten. Filling the DTXS by DSP56300 Core writes (MOVE(P) instructions or DMA transfers) clears STRQ. The DSP56300 Core may set the STIE bit to cause a host receive data interrupt when STRQ is set.

In the 32-bit mode (HM=$1 with HRF=$0), only the two least significant bytes of the DTXS are transferred. (See Section 6.1.9, above,Section 6-3 on page 6-24, and Section 6-9 on page 6-54)

Hardware, software and personal software resets empty the DTXS.

6.1.12   DSP Host Port GPIO Data Register (DATH)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT 23	DAT 22	DAT 21	DAT 20	DAT 19	DAT 18	DAT 17	DAT 16	DAT 15	DAT 14	DAT 13	DAT 12	DAT 11	DAT 10	DAT 9	DAT 8	DAT 7	DAT 6	DAT 5	DAT 4	DAT 3	DAT 2	DAT 1	DAT 0

Bit	Name	Function
23-0	DAT23-DAT0	GPIO Pin Data

The DATH is a 24-bit read/write data register used by the DSP56300 Core to read or write data to/from host port pins configured as GPIO. The DATH cannot be accessed by the host processor.

DAT23-DAT0 are used to read or write data from/to the corresponding GPIO pin. The functionality of the DAT23-DAT0 bits is defined in Table 6-5, on page 6-39.

Hardware and software resets clear all DATH bits.

6.1.13   DSP Host Port GPIO Direction Register (DIRH)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR 23	DIR 22	DIR 21	DIR 20	DIR 19	DIR 18	DIR 17	DIR 16	DIR 15	DIR 14	DIR 13	DIR 12	DIR 11	DIR 10	DIR 9	DIR 8	DIR 7	DIR 6	DIR 5	DIR 4	DIR 3	DIR 2	DIR 1	DIR 0

Bit	Name	Function
23-0	DIR23-DIR0	GPIO Pin Direction

The DIRH is a 24-bit read/write register used by the DSP56300 Core to control the direction of the host port pins in GPIO mode. The DIRH cannot be accessed by the host processor.

DIR23-DIR0 are used to define the corresponding GPIO pins as input or output. The functionality of the DIR23-DIR0 bits is defined in Table 6-5, on page 6-39.

Hardware and software resets clear all DIRH bits.

Table 6-5. DATH and DIRH Functionality

DIRx	DATx	
	GPIO pin[a]	non-GPIO pin[a]
0	Read only bit. The value read is the binary value of the pin. The corresponding pin is configured as an input.	Read only bit. Does not contain significant data.
1	Read/write bit. The value written is the value read. The corresponding pin is configured as an output, and is driven with the data written to DATx.	Read/write bit. The value written is the value read.

a. defined by the selected mode

6.2   HI32 - HOST SIDE

The HI32 appears to the host processor as a bank of registers.

In the Universal Bus modes:

- The HI32 occupies eight words in the host processor address space (see Figure 6-5). The PCI configuration registers (CDID/CVID, CSTR/CCMR, CCCR/CRID, CHTY/CLAT, CBMA and CILP) cannot be accessed by the host processor in the Universal Bus modes.

- Due to the fast DSP56300 Core interrupt response, most host microprocessors can read or write data at their maximum programmed non-DMA instruction rate without testing the handshake flags for each transfer. If the full interrupt driven handshake is not needed, the high speed data transfer between the host and the HI32 may be supported with only host data strobe/acknowledge handshake mechanism. DMA hardware may be used with the handshake flags to transfer data without host processor intervention.

- When operating with a host bus less than 24 bits wide, the data pins that are not used for transferring data must be forced or pulled up or down to Vcc or to GND respectively. For example: when operating with a 16-bit bus (e.g. ISA bus), HP48-HP41 must be forced or pulled up to Vcc or pulled down to GND.

In the PCI mode:

- In memory space read/write transactions, the HI32 occupies 16384 Dwords (see Figure 6-3). The HTXR FIFO and HRXS FIFO can be accessed by the host at 16377 Dword locations. These FIFOs appear to the external host as 16377 Dwords of read/write memory. Registers are accessed as 32-bit Dwords.

- HAD1 and HAD0 should be zero during the address phase of a transaction. The HI32 will respond with a target-disconnect transaction termination with the first data phase if HAD1-HAD0≠$0 during the address phase.

- In configuration space read/write transactions, the HI32 occupies 64 Dwords (see Figure 6-4). The configuration registers are accessed as 32-bit Dwords, thus HAD1 ad HAD0 must be zero during the address phase. The HI32 will ignore the transaction if HAD1-HAD0≠$0 during the address phase of a configuration transaction.

- In PCI host-to-DSP data transfers to the HI32 registers (HCTR, HSTR,

HCVR and all configuration space registers): disabled byte lanes (i.e. the corresponding byte enable line is negated) are not written and the corresponding bytes do not contain significant data.

- In HI32 to PCI agent data transfers, all four byte lanes are driven with data, regardless of the value of the byte enables.

- In HCTR, HSTR, HCVR and configuration space register accesses: if all four byte lanes are disabled the HI32 completes the data phase without affecting any flags or data.

- In PCI DSP-to-host data transfers via the HRXS or HRXM, all four byte lanes are driven with data, in accordance with FC1-FC0 or HRF1-HRF0 bits, regardless of the value of the byte enable pins (HC3/$\overline{\text{HBE3}}$-HC0/$\overline{\text{HBE0}}$).

- In PCI host-to-DSP data transfers, data is written to the HTXR FIFO, in accordance with FC1-FC0 or HTF1-HTF0 bits, regardless of the value of the byte enable pins (HC3/$\overline{\text{HBE3}}$-HC0/$\overline{\text{HBE0}}$).

- As a PCI target the HI32 executes the PCI bus command as follows:

HC3/$\overline{\text{HBE3}}$-HC0/$\overline{\text{HBE0}}$	Executed as Command Type
0000	ignored[a]
0001	ignored[a]
0010	ignored[a]
0011	ignored[a]
0100	ignored[a]
0101	ignored[a]
0110	**Memory Read**
0111	**Memory Write**
1000	ignored[a]
1001	ignored[a]
1010	**Configuration Read**
1011	**Configuration Write**
1100	**Memory Read**
1101	ignored[a]
1110	**Memory Read**
1111	**Memory Write**

a. All internal address decoding is ignored and $\overline{\text{DEVSEL}}$ is not asserted.

- The HI32 will not reach dead-lock due to illegal PCI events. Illegal PCI

events bring the HI32 Master and Target state machines to the IDLE state.

The master data transfer format control bits (FC1-FC0 in the DPMC) affect the HTXR-DRXR and DTXM-HRXM data paths only. The target data transfer format control bits (HTF1-HTF0 and HRF1-HRF0 in the HCTR) affect the HTXR-DRXR and DTXS-HRXS data paths only. The data paths to the other host registers (HCTR, HSTR, HCVR, CDID/CVID, CSTR/CCMR, CCCR/CRID, CHTY/CLAT, CBMA and CILP) are not affected by the data transfer format control bits.

The host side registers can be accessed by the host processor. The CCMR, CLAT and CBMA HI32 configuration registers can also be accessed, indirectly, by the DSP56300 Core in the Self Configuration mode (HM=$5 - see Section 6.1.1.13 on page 6-13).

Reserved addresses are read as zeros, and should be written with zeroes for future compatibility.

Host processors may use standard host processor instructions and addressing modes to communicate with the HI32 registers. The host processor may be any of a number of industry standard microcomputers or microprocessors, DMA controllers or standard peripheral bus (e.g. ISA/EISA), because this interface appears to the host like static RAM.

With the host command feature, the host processor can issue vectored interrupt requests to the DSP56300 Core. The host may select any one of 128 DSP56300 Core interrupt routines to be executed by writing a vector address register in the HI32. This flexibility allows the host programmer to execute up to 128 pre-programmed functions inside the DSP. For example, host exceptions can allow the host processor to read or write DSP registers, X-, Y-, or program memory locations, force exception handlers (e.g., SSI, Timer, $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$ exception routines), and perform control and debugging operations if exception routines are implemented in the DSP to perform these tasks. The host processor can also generate non-maskable interrupt requests to the DSP56300 Core using the host commands.

Figure 6-3. Host Side Registers (PCI Memory Address Space[1])

Base Address: $0000 Base Address: $000C	Reserved	(4 Dwords)
Base Address: $0010	HI32 Control Register (HCTR)	
Base Address: $0014	HI32 Status Register (HSTR)	
Base Address: $0018	Host Command Vector Register (HCVR)	
Base Address: $001C Base Address: $FFFC	Host Transmit/Slave Receive Data Register (HTXR/HRXS) (16377 Dwords)	

Figure 6-4. Host Side Registers (PCI Configuration Address Space[2])

$00 (CDID/CVID)	Device ID (CDID)		Vendor ID (CVID)	
$04 (CSTR/CCMR)	Status (CSTR)		Command (CCMR)	
$08 (CCCR/CRID)	Class Code (CCCR)			Revision ID (CRID)
$0C (CLAT)		Header Type (CHTY)	Latency Timer (CLAT)	
$10 (CBMA)	Memory Space Base Address (CBMA)			
$14 $F8	Reserved			(58 Dwords)
$FC (CILP)	MAX_LAT	MIN_GNT	Interrupt Line	Interrupt Pin

Figure 6-5. Host Side Registers (Universal Bus Mode Address Space[3])

Base Address: $0 Base Address: $3	Reserved	(4 Locations)
Base Address: $4	HI32 Control Register (HCTR)	
Base Address: $5	HI32 Status Register (HSTR)	
Base Address: $6	Host Command Vector Register (HCVR)	
Base Address: $7	Host Transmit/Slave Receive Data FIFO (HTXR/HRXS)	

---

1. Addresses shown are in bytes. The base address is defined by the CBMA register
2. Addresses shown are in bytes.
3. Addresses shown are in words (locations). The base address is defined by eight bits of the CBMA register.

6.2.1   HI32 Control Register (HCTR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												TWSD			HS2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS1	HS0		HRF1	HRF0		HTF1	HTF0	SFT	DMAE	HF2	HF1	HF0	RREQ	TREQ	
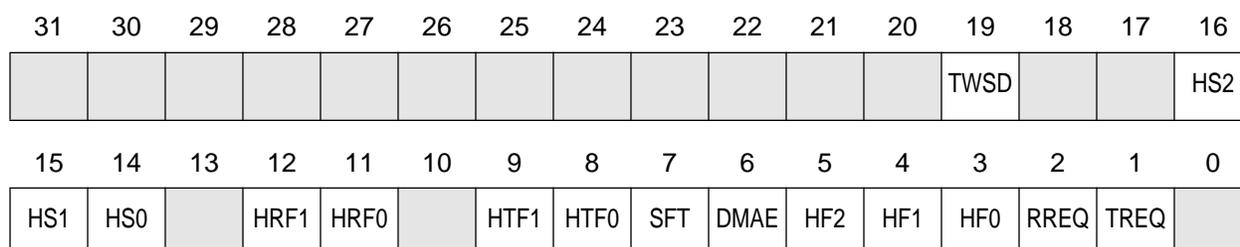
☐ Reserved, read as zero and should be written zero

Bit	Name	Function
1	TREQ	Transmit Request Enable
2	RREQ	Receive Request Enable
5-3	HF2-HF0	Host Flags
6	DMAE	DMA Enable (ISA/EISA)
7	SFT	Slave Fetch Type
9-8	HTF1-HTF0	Host Transmit Data Transfer Format
12-11	HRF1-HRF0	Host Receive Data Transfer Format
16-14	HS2-HS0	Host Semaphores
19	TWSD	Target Wait State Disable
31-20,18-17,13,10,0	reserved	

The HCTR is a 32-bit read/write control register used by the host processor to control the HI32 interrupts, flags, semaphores, data transfer formats and operation modes.

In the PCI mode (HM=$1), the HAD31-HAD0 pins are driven with HCTR data during a read access; and the pins are written to the HCTR in a write access.

In a 24-bit data Universal Bus mode (HM=$2 or $3 and HTF=$0 or HRF=$0), the HD23-HD0 pins are driven with the three least significant HCTR bytes during a read access; HD23-HD0 are written to the three least significant HCTR bytes in a write access.

In a 16-bit data Universal Bus mode (HM=$2 or $3 and HTF≠$0 or HRF≠$0), the HD15-HD0 pins are driven with the two least significant bytes of the HCTR in a read access; HD15-HD0 are written to the two least significant bytes of the HCTR, the most significant portion is zero filled during the HCTR write.

In PCI mode (HM=$1) memory space transactions, the HCTR is accessed if the PCI address is HI32_base_address: $010.

The HCTR is written in accordance with the byte enables (HC3/$\overline{HBE3}$-HC0/$\overline{HBE0}$ pins). Byte lanes that are not enabled are not written and the corresponding bits remain unchanged.

The HCTR bits affect the HI32 logic upon the completion of the transaction in they were written.

When in a Universal Bus mode (HM=$2 or $3), the HCTR is accessed if the HA10-HA3 value matches the HI32 base address (CBMA, see Section 6.2.11 on page 6-70) and the HA2-HA0 value is $4.

The control bits are described in the following paragraphs.

6.2.1.1 Transmit Request Enable (TREQ) Bit 1

The TREQ bit is used to control the $\overline{\text{HIRQ}}$ and HDRQ pins for host transmit data transfers (see Table 6-6), when in a Universal Bus mode (HM=$2 or $3).

If DMA enable bit (DMAE) is cleared, TREQ enables the host interrupt request $\overline{\text{HIRQ}}$ pin when the host transmit data request (HTRQ) status bit in the HI32 status register (HSTR) is set. If TREQ is cleared, HTRQ host interrupt requests are disabled. If TREQ is set, the host interrupt request $\overline{\text{HIRQ}}$ pin will be asserted if HTRQ is set. HDRQ is negated.

If DMAE is set, TREQ enables the host DMA request (HDRQ) pin when the host transmit data request (HTRQ) status bit in the HSTR is set. If TREQ is cleared, HTRQ external DMA requests are disabled. If TREQ is set, the host DMA request HDRQ pin will be asserted if HTRQ is set. $\overline{\text{HIRQ}}$ is negated (high impedance if HIRD=0 in the DCTR).

The personal hardware reset clears TREQ.

6.2.1.2 Receive Request Enable (RREQ) Bit 2

The RREQ bit is used to control the $\overline{\text{HIRQ}}$ and HDRQ pins for DSP-to-host data transfers (see Table 6-6), when in a Universal Bus mode (HM=$2 or $3).

If DMAE is cleared, RREQ enables the host interrupt request ($\overline{\text{HIRQ}}$) pin when the host receive data request (HRRQ) status bit in the HSTR is set. If RREQ is cleared, HRRQ host interrupt requests are disabled. If RREQ is set, the host interrupt request $\overline{\text{HIRQ}}$ pin will be asserted if HRRQ is set. HDRQ is negated.

If DMAE is set, RREQ enables the host DMA request (HDRQ) pin when the host receive data request (HRRQ) status bit in the HSTR is set. If RREQ is cleared, HRRQ host DMA requests are disabled. If RREQ is set, the host DMA request HDRQ pin will be asserted if HRRQ is set. $\overline{\text{HIRQ}}$ is negated (high impedance if HIRD=0 in the DCTR).

The personal hardware reset clears RREQ.

NOTE: In a Universal Bus mode (HM = $2 or $3), when both the TREQ and RREQ control bits (in the HCTR) are cleared, host interrupt request / strobe / acknowledge hardware handshake (using the $\overline{HIRQ}$ / Data Strobe / HTA pins) is disabled. The host may poll the HTRQ, HRRQ status bits or use the host data strobe/acknowledge hardware handshake (using the Data Strobe / HTA pins) (see Table 6-10, on page 6-75).

Table 6-6. $\overline{HIRQ}$ and HDRQ Pin Definition

DMAE	TREQ	RREQ	$\overline{HIRQ}$ Pin	HDRQ pin
0	*0*	*0*	*negated [a] (HRRQ, HTRQ polling)*	*high impedance*
0	*0*	*1*	*HRRQ Host Interrupt Requests Enabled*	*high impedance*
0	*1*	*0*	*HTRQ Host Interrupt Request Enabled*	*high impedance*
0	*1*	*1*	*HRRQ, HTRQ Interrupt Requests Enabled*	*high impedance*
1	*0*	*0*	*negated[a]*	*high impedance*
1	*0*	*1*	*negated[a]*	*HRRQ DMA Request Enabled*
1	*1*	*0*	*negated[a]*	*HTRQ DMA Request Enabled*
1	*1*	*1*	*negated[a]*	*HRRQ, HTRQ Host DMA Requests Enabled*

a. high impedance if HIRD=0 in the DCTR

6.2.1.3   Host Flags (HF2-HF0) Bits 5 and 3

The HF2-HF0 bits are used as general purpose flags for host-to-DSP communication. HF2-HF0 may be set or cleared by the host processor.

The personal hardware reset clears HF2-HF0.

6.2.1.4   DMA Enable (DMAE) Bit 6

The DMAE is used by the host processor to enable the HI32 ISA/EISA DMA-type accesses, when in a Universal Bus mode (HM=$2 or $3) (see Table 6-7)

.

Table 6-7. DMAE Definition

DMAE bit	HAEN pin	ISA/EISA Access Type	$\overline{\text{HIRQ}}$ and HDRQ functionality
0	*0*	*The HI32 responds when it identifies its address (i.e. ISA/EISA I/O-type access)*	$\overline{\text{HIRQ}}$ *is active,* HDRQ *is negated*
0	*1*	*The HI32 will not respond to any access*	$\overline{\text{HIRQ}}$ *is active,* HDRQ *is negated*
1	*0*	*The HI32 responds when it identifies its address (i.e. ISA/EISA I/O-type access)*	HDRQ *is active,* $\overline{\text{HIRQ}}$ *is negated*[a]
1	*1*	*The HI32 responds when* $\overline{\text{HDAK}}$ *is asserted (i.e. ISA/EISA DMA-type access)*	HDRQ *is active,* $\overline{\text{HIRQ}}$ *is negated*[a]

a. *high impedance if HIRD=0 in the DCTR*

If the HAEN pin is driven low by the host, the HI32 responds when it identifies its address (i.e. ISA/EISA I/O-type accesses). The HI32 will not respond to ISA/EISA DMA-type accesses.

If the HAEN pin is high:

- *If DMAE is cleared the HI32 cannot be accessed.*

- *If DMAE is set, the HI32 responds to ISA/EISA DMA-type accesses.*

If DMAE is cleared, the HDRQ pin is negated, $\overline{\text{HIRQ}}$ is active.

If DMAE is set, the $\overline{\text{HIRQ}}$ pin is negated, HDRQ is active. This allows the HI32 to generate host DMA requests during ISA/EISA I/O-type accesses. A typical application would be that the external host writes to the HI32 using a polling procedure, and the external DMA reads from the HI32. An external bus controller arbitrates between the two and sets or clears HAEN accordingly.

If both DMAE and HAEN are set, HTA is released (high impedance), as DMA devices cannot extend DMA cycles (ISA/EISA).

The personal hardware reset clears DMAE.

6.2.1.5 Slave Fetch Type (SFT) Bit 7

The SFT bit defines the fetch mode (data fetch or pre-fetch) as described below.

SFT	Slave Fetch Type
1	*Fetch*
0	*Pre-fetch*

In the Fetch mode: the HI32 requests data from the DSP56300 Core (by enabling the STRQ status bit and generating Core interrupt requests or DMA requests if enabled), only after the host has begun a read transaction from the HI32.

In the Pre-Fetch mode: the HI32 requests data from the DSP56300 Core (by enabling the STRQ status bit and generating Core interrupt requests or DMA requests if enabled) whenever the DTXS is not full.

In the PCI mode (HM=$1):

Fetch (SFT=1):
> *The DSP-to-host data path (DTXS-HRXS) is a six word deep (three word deep if HRF=$0) FIFO buffer. Writing SFT=1 resets the DSP-to-host data path and clears STRQ and HRRQ. During a read transaction from the HRXS, STRQ is set if the DTXS-HRXS FIFO is not full, and cleared when the DSP56300 Core fills the DTXS; HRRQ is cleared if the HRXS is empty, and set if it contains data to be read by an external host. If the host is not executing a read transaction from the HRXS, the DSP-to-host data path is forced to the reset state and STRQ and HRRQ are cleared.*

In a Universal Bus mode (HM= $2 or $3):

Fetch (SFT=1):
> *There is no FIFO buffering of the DSP-to-host data path. Writing SFT=1 resets the DSP-to-host data path and clears the STRQ and the HRRQ. At the beginning of a read data transfer from the HRXS, STRQ is set. STRQ is cleared when the DSP56300 Core writes to the DTXS; HRRQ is cleared if the HRXS is empty, and set if it contains data to be read by an external host. If the host is not reading from the HRXS, the DSP-to-host data path is forced to the reset and STRQ and HRRQ are cleared.*
>
> *NOTE: Any data remaining in the DSP-to-host data path when entering the reset state, is lost.*

In both the PCI and Universal Bus modes (HM=$1, $2 or $3):

Pre-fetch (SFT=0):
> *The DSP-to-host data path is a six word deep (three word deep in the 32-bit data format mode, HM=$1 and HRF=$0) FIFO buffer. STRQ reflects the status of the*

DTXS and HRRQ reflects the status of the HRXS. STRQ is set if the DTXS is not full, and cleared when the DSP56300 Core fills the DTXS. HRRQ is cleared if the HRXS is empty, and set when it contains data to be read by an external host.

The value of SFT may be changed only if the DTXS-HRXS data path is empty.

The personal hardware reset clears SFT.

6.2.1.6   Host Transmit Data Transfer Format (HTF1-HTF0) Bits 9 and 8

The HTF1-HTF0 bits define data transfer formats for host-to-DSP communication. The data transfer format converter (HDTFC) operates according to the specified HTF1-HTF0 (see Table Transmit Data Transfer Format on page 51).

PCI host to DSP data transfer formats (HM=$1):

- *If HTF=$0 (32-bit data mode):*
  All four PCI data bytes from HAD31-HAD0 pins are written to the 32-bit HTXR. The two least significant bytes are transferred to the two least significant bytes of the DRXR FIFO after which the two most significant bytes are transferred to the two least significant bytes of the DRXR FIFO. Thus, when the DSP56300 Core reads two words from the DRXR, the two least significant bytes of the first word read contain the two least significant bytes of the 32-bit word written to the HTXR, the two least significant bytes of the second word read contain the two most significant bytes of the 32-bit word.

- *If HTF=$1 or $2:*
  The three least significant PCI data bytes from the HAD23-HAD0 pins are transferred to the three least significant HTXR bytes and transferred to the DRXR to be read by the DSP56300 Core.

- *If HTF=$3:*
  The three most significant PCI data bytes from the HAD31-HAD8 pins are transferred to the three least significant HTXR bytes and transferred to the DRXR to be read by the DSP56300 Core.

Universal Bus mode host to DSP data transfer formats (HM=$2 or $3):

- *If HTF=$0:*
  The 24-bit data from HD23-HD0 data pins is transferred to the three least significant HTXR bytes and transferred to the DRXR to be read by the DSP56300 Core.

- *If HTF=$1:*
  The 16-bit data from HD15-HD0 data pins is transferred to the three least significant HTXR bytes as right aligned and zero extended and transferred to the DRXR to be read by the DSP56300 Core.

- *If HTF=$2:*
  The 16-bit data from HD15-HD0 data pins is transferred to the three least significant HTXR bytes as right aligned and sign extended and transferred to the DRXR to be read by the DSP56300 Core.

- *If HTF=$3:*
  The 16-bit data from HD15-HD0 data pins is transferred to the three least significant bytes of the HTXR, as left aligned, the least significant byte is zero filled and transferred to the DRXR to be read by the DSP56300 Core.

To assure proper operation,:
- *HTF1-HTF0 may be changed only if the host-to-DSP data path is empty.*
- *Switching between 32-bit data modes and non-32-bit data modes may be done only in the personal software reset state (HM=$0 and HACT=0).*
- *If the HTF1-HTF0 value is not equal to the value of the FC1-FC0 bits in the DPMC: PCI transactions that start in the non-data address space (i.e. the PCI address is less than HI32_base_address:$007) should not extend into the data address space.*

The personal hardware reset clears HTF1-HTF0.

Table 6-8. Transmit Data Transfer Format

HTF 1	HTF 0	Host to DSP Data Transfer Format	
		PCI mode	**Universal Bus mode**
0	0	*All 32 PCI data bits are written to the HTXR as two zero extended 16-bit words.*	*All HD23-HD0 data are written to the HTXR.*
0	1	*The three least significant PCI data bytes are written to the HTXR.*	*HD15-HD0 is written to the HTXR, right aligned and zero extended.*
1	0	*The three least significant PCI data bytes are written to the HTXR.*	*HD15-HD0 are written to the HTXR, right aligned and sign extended.*
1	1	*The three most significant PCI data bytes are written to the HTXR.*	*HD15-HD0 are written to the HTXR, left aligned, and zero filled.*

6.2.1.7   Host Receive Data Transfer Format (HRF1-HRF0) Bits 12 and 11

The HRF1-HRF0 bits define data transfer formats for DSP-to-host communication. The data transfer format converter (HDTFC) operates according to the specified HRF1-HRF0 (See Table Receive Data Transfer Format on page 54).

DSP to PCI host data transfer formats (HM=$1):

- *If HRF=$0 (32-bit data mode):*
  The two least significant bytes of two words written to the DTXS are transferred to the HRXS. The two least significant bytes of the first word written to the DTXS are transferred to the two least significant bytes of the HRXS. The two least significant bytes of the second word written to the DTXS are transferred to the two most significant bytes of the HRXS. All four HRXS bytes are output to the HAD31-HAD0 pins.

- *If HRF=$1:*
  The data written to the DTXS is transferred to the three least significant HRXS bytes and output to the HAD31-HAD0 pins as right aligned and zero extended in the most significant byte.

- *If HRF=$2:*
  The data written to the DTXS is transferred to the three least significant HRXS bytes and output to the HAD31-HAD0 pins as left aligned and zero filled in the least significant byte.

- *If HRF=$3:*
  The data written to the DTXS is transferred to the three least significant HRXS bytes and output to the HAD31-HAD0 pins as right aligned and sign extended in the most significant byte.

Universal Bus mode DSP to host data transfer formats (HM=$2 or $3):

- *If HRF=$0:*
  The data written to the DTXS is transferred to the HRXS and output to the HI32 data pins HD23-HD0.

- *If HRF=$1 or $2:*
  The two least significant bytes of the data written to the DTXS is transferred to the HRXS and output to the HI32 data pins HD15-HD0.

- *If HRF=$3:*
  The two most significant bytes of the data written to the DTXS is transferred to the HRXS and output to the HI32 data pins HD15-HD0.

To assure proper operation, HRF1-HRF0 may be changed only if the DSP-to-host slave data path is empty. In addition, switching between 32-bit data modes and non-32-bit data modes may be done only in the personal software reset state (HM=$0 and HACT=0).

The personal hardware reset clears HRF1-HRF0.

6.2.1.8   Host Semaphores (HS2-HS0) Bits 16 and 14

The HS2-HS0 bits may be used by the host processors for software arbitration of mastership over the HI32. These bits do not affect the HI32 operation and only serve as a read/write semaphore repository. These bits may be used as a mailbox between the external hosts. For example: the semaphores may be used to assist HI32 bus arbitration among several external hosts.
All external host processors that compete for mastership over the HI32 should work according to the same software protocol for handling over the HI32 from one host processor to another.

The personal hardware reset clears HS2-HS0.

6.2.1.9   Target Wait State Disable (TWSD) Bit 19

The TWSD bit is used to disable PCI wait states (which are inserted by negating $\overline{HTRDY}$), during a data phase.

If TWSD is cleared and the HI32 is in the PCI mode (HM=$1):

- *the HI32 as the selected target in a read data phase from the HRXS, will insert PCI wait states if the HRXS is empty (HRRQ=0). Wait states will be inserted until the data is transferred from the DSP side to the HRXS. Up to eight wait states may be inserted before a target initiated transaction termination (disconnect-C/Retry) will be generated.*

- *the HI32 as the selected target in a write data phase to the HTXR, will insert PCI wait states if the HTXR is full (HTRQ=0). Wait states will be inserted until the data is transferred from the HTXR to the DSP side. Up to eight wait states may be inserted before a target initiated transaction termination (disconnect-C/Retry) will be generated.*

- *the HI32 as the selected target in a write data phase to the HCVR, will insert PCI wait states if a host command is pending (HC=1). Wait states will be inserted until the pending host command is serviced. Up to eight wait states may be inserted before a target initiated transaction termination (disconnect-C/Retry) will be generated.*

If TWSD is set and the HI32 is in the PCI mode (HM=$1):

- *the HI32 as the selected target in a read transaction from the HRXS, will generate a target initiated transaction termination (disconnect-C) if the HRXS is empty (HRRQ=0).*

- *the HI32 as the selected target in a write transaction to the HTXR, will generate a target initiated transaction termination (disconnect-C) if the*

Table 6-9. Receive Data Transfer Format

HRF 1	HRF 0	DSP to Host Data Transfer Format	
		PCI mode	**Universal Bus mode**
0	0	The two least significant bytes of two HRXS locations are output.	The three least significant HRXS bytes are output to HD23-HD0.
0	1	The three least significant HRXS bytes are output right aligned and zero extended.	The two least significant HRXS bytes are output to HD15-HD0.
1	0	The three least significant HRXS bytes are output right aligned and sign extended.	The two least significant HRXS bytes are output to HD15-HD0.
1	1	The three least significant HRXS bytes are output left aligned and zero filled.	The two middle HRXS bytes are output to HD15-HD0.

HTXR is full (HTXR=0)

- *the HI32 as the selected target in a write transaction to the HCVR, will generate a target initiated transaction termination (disconnect-C) if a host command is pending (HC=1).*

TWSD is ignored when the HI32 is not in the PCI mode (HM≠$1).

The personal hardware reset clears TWSD.

6.2.1.10   HCTR Reserved Control Bits 31-20, 18-17, 13, 10 and 0

These bits are reserved for future expansion, they are read as zeros and should be written with zeros for upward compatibility.

6.2.2   HI32 Status Register (HSTR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HREQ	HINT	HF5	HF4	HF3	HRRQ	HTRQ	TRDY
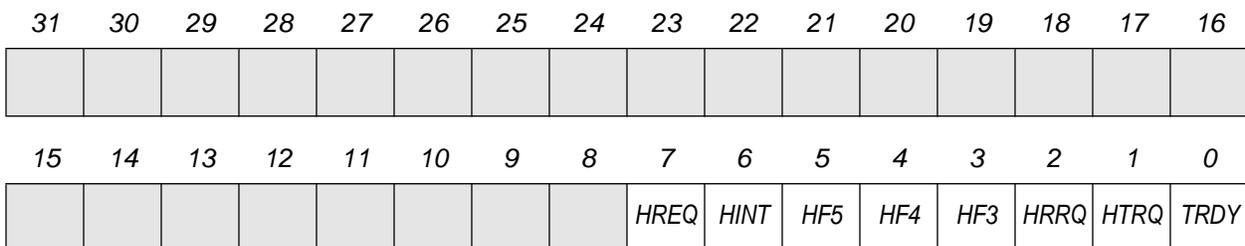
Reserved, read as zero and should be written zero

Bit	Name	Function
0	TRDY	Transmitter Ready
1	HTRQ	Host Transmit Data Request
2	HRRQ	Host Receive Data Request
5-3	HF5-HF3	Host Flags
6	HINT	Host interrupt A
7	HREQ	Host Request
31-8	reserved	

The HSTR is a 32-bit read-only status register used by the host processor to examine the status and flags of the HI32.

When the HSTR is read to the PCI bus (HM=$1), the HAD31-HAD0 pins are driven with the HSTR data during a read access.

In a 24-bit data Universal Bus mode (HM=$2 or $3 and HRF=$0), the HD23-HD0 pins are driven with the three least significant HSTR bytes during a read access.

In a 16-bit data Universal Bus mode (HM=$2 or $3 and HRF≠$0), the HD15-HD0 pins are driven with the two least significant bytes of the HSTR in a read access.

In PCI mode (HM=$1) memory space transactions, the HSTR is accessed if the PCI

address is HI32_base_address: $014.

When in a Universal Bus mode (HM=$2 or $3), the HSTR is accessed if the HA10-HA3 value matches the HI32 base address (CBMA, see Section 6.2.11 on page 6-70) and the HA2-HA0 value is $5.

The status bits are described in the following paragraphs.

6.2.2.1 Transmitter Ready (TRDY) Bit 0

*The TRDY status bit indicates that **both** the HTXR and the DRXR registers are empty. If TRDY is set to one, the data that the host processor writes to the HTXR will be immediately transferred to the DSP side of the HI32. This has many applications. For example: if the host processor issues a host command which causes the DSP56300 Core to read the DRXR, the host processor can be guaranteed that the data it just transferred to the HI32 is what is being received by the DSP56300 Core.*

In order to support high speed data transfers, the HI32 host-to-DSP data path is a six word deep FIFO (five word deep in the Universal Bus modes, three word deep in the 32-bit mode, HM=$1 and HTF=$0). In PCI data transfers with HM=$1 and HTF≠$0, if TRDY is set, the HI32 will not insert wait states in the next six data transfers written by the host to the HTXR. In PCI data transfers with HM=$1 and HTF=$0 (i.e. 32-bit mode), if TRDY is set, the HI32 will not insert wait states in the next three data phases written by the host to the HTXR. In Universal bus mode data transfers, if TRDY is set, the HI32 will not insert wait states in the next five data transfers written by the host to the HTXR.

TRDY is cleared when the HTXR is written by the host processor.

Hardware, software and personal software resets set TRDY.

6.2.2.2 Host Transmit Data Request (HTRQ) Bit 1

The HTRQ bit indicates that the host transmit data FIFO (HTXR) is not full and can be written by the host processor. HTRQ is set when the HTXR data is transferred to the DRXR. HTRQ is cleared when the HTXR is filled by host processor writes.

In the PCI mode:
The HI32 as target in a write data phase to the HTXR, will negate $\overline{HTRDY}$, and insert up to eight PCI wait cycles, if HTRQ is cleared.

In a Universal Bus mode write to the HTXR, the HI32 slave will negate HTA as long as HTRQ is cleared. HTRQ may be used to assert the external $\overline{HIRQ}$ pin if the TREQ bit is set. Regardless of whether the HTRQ host interrupt request is enabled, HTRQ provides valid status so that polling techniques may be used by the host processor.

Hardware, software and personal software resets set HTRQ.

6.2.2.3 Host Receive Data Request (HRRQ) Bit 2

The HRRQ bit indicates that the host slave receive data FIFO (HRXS) contains data from the DSP56300 Core and may be read by the host processor.

In the PCI mode:
The HI32 as a target in a read data phase from the HRXS, will negate $\overline{HTRDY}$, and insert up to eight PCI wait cycles, if HRRQ is cleared.

In a Universal Bus mode read from the HRXS, the HI32 slave will negate $\overline{HTA}$ as long as HRRQ is cleared. HRRQ may be used to assert the $\overline{HIRQ}$ pin if the RREQ bit is set. Regardless of whether the HRRQ host interrupt request is enabled, HRRQ provides valid status so that polling techniques may be used by the host processor.

HRRQ functions in accordance with the value of the slave fetch type (SFT) bit in the HCTR.

Fetch (SFT= 1): HRRQ is always read as zero.

Pre-fetch (SFT= 0):
The DSP-to-host data path is FIFO buffered. HRRQ reflects the status of the HRXS. HRRQ is cleared if the HRXS is empty, and set when data is transferred from the DTXS.

Hardware, software and personal software resets clear HRRQ.

6.2.2.4 Host Flags (HF5-HF3) Bits 5, 4 and 3

The HF5-HF3 bits in the HSTR indicate the state of host flags HF5-HF3 respectively, in the DSP Control Register (DCTR) on the DSP side. HF5-HF3 can be changed, albeit indirectly, only by the DSP56300 Core.

HF5-HF3 are cleared by a hardware or software reset.

6.2.2.5 Host Interrupt A (HINT) Bit 6

The HINT bit reflects the status of the HINT bit in the DSP Control Register (DCTR) and the HINTA pin. HINT is set if the host interrupt A bit is set in the DCTR, and the HINTA pin is driven low. HINT is cleared if the host interrupt A is cleared in the DCTR, and the HINTA pin is driven low.

HINT is cleared by a hardware or software reset.

6.2.2.6 Host Request (HREQ) Bit 7

HREQ is set and cleared in accordance with the following table:

TREQ	RREQ	HREQ
0	0	cleared
0	1	set if HRRQ=1 otherwise cleared
1	0	set if HTRQ=1 otherwise cleared
1	1	set if HTRQ=1 or HRRQ=1 otherwise cleared

The personal hardware reset clears HREQ.

6.2.2.7 HSTR Reserved Status Bits 31-8

These status bits are reserved for future expansion and read as zeros during host read operations.

6.2.3 Host Command Vector Register (HCVR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HNMI								HV6	HV5	HV4	HV3	HV2	HV1	HV0	HC

▢ Reserved, read as zero and should be written zero

Bit	Name	Function
0	HC	Host Command
7-1	HV6-HV0	Host Command Vector
15	HNMI	Host Non Maskable Interrupt
31-16,14-8	reserved	

The HCVR is a 32-bit read/write register used by the host processor to cause the DSP56300 Core to execute a vectored interrupt. The host command feature is independent of any of the data transfer mechanisms in the HI32. It can be used to cause any of the 128 possible interrupt routines in the DSP to be executed.

When the HCVR is read to the PCI bus (HM=$1), the HAD31-HAD0 pins are driven with the HCVR data during a read access; and these pins are written to the HCVR in a write

access.

In a 24-bit data Universal Bus mode (HM=$2 or $3 and HTF=$0 or HRF=$0), the HD23-HD0 pins are driven with the three least significant bytes of the HCVR in a read access; HD23-HD0 are written to the three least significant bytes of the HCVR, the most significant portion is zero filled during the HCVR write.

In a 16-bit data Universal Bus mode (HM=$2 or $3 and HTF≠$0 or HRF≠$0), the HD15-HD0 pins are driven with the two least significant bytes of the HCVR in a read access; HD15-HD0 are written to the two least significant bytes of the HCVR, the most significant portion is zero filled during the HCVR write.

In PCI mode (HM=$1) memory space transactions, the HCVR is accessed if the PCI address is HI32_base_address: $018.

The HCVR is written in accordance with the byte enables (HC3/$\overline{HBE3}$-HC0/$\overline{HBE0}$ pins). Byte lanes that are not enabled are not written and the corresponding bits remain unchanged.

When in a Universal Bus mode (HM=$2 or $3), the HCVR is accessed if the HA10-HA3 value matches the HI32 base address (CBMA, see Section 6.2.11 on page 6-70) and the HA2-HA0 value is $6.

If TWSD is cleared, the HI32 as the selected PCI target (HM=$1) in a write data phase to the HCVR will insert PCI wait states if a host command is pending (HC=1). Wait states will be inserted until the pending host command is serviced. Up to eight wait states may be inserted before a target initiated transaction termination (disconnect-C/Retry) will be generated.

In a Universal Bus mode write to the HCVR, the HI32 will insert wait states if a host command is pending (HC=1). Wait states will be inserted until the pending host command is serviced.

The HCVR bits are described in the following paragraphs.

6.2.3.1   Host Command (HC) Bit 0

The HC bit is used by the host processor to handshake the execution of host command interrupt requests. Normally, the host processor sets HC to request a host command interrupt from the DSP56300 Core. When the host command interrupt request is acknowledged by the DSP56300 Core, the HC bit is cleared by the HI32 hardware. The host processor can read the state of HC to determine when the host command request has been serviced. The host processor cannot clear HC.

Setting HC causes host command pending (HCP) to be set in the DSR. The host can write HC and HV in the same write cycle if desired.

If HC is set:

In the PCI mode:
The HI32 as a target in a write data phase to the HCVR, will negate $\overline{HTRDY}$, and insert up to eight PCI wait cycles, until HC is cleared.

In a Universal Bus mode:
In a write transaction to the HCVR, the HI32 slave will negate HTA, until HC is cleared.

The personal software reset clears HC.

6.2.3.2 Host Vector (HV6-HV0) Bits 7-1

The seven HV bits select the host command interrupt address. When the host command interrupt is recognized by the DSP56300 Core interrupt control logic, the starting address of the interrupt executed is $2 \times$ (HV6-HV0).

The host processor can select any of the 128 possible interrupt routine starting addresses in the DSP by writing the interrupt routine starting address divided by two into HV. This means that the host processor can force any of the existing interrupt routines (SSI, Timer, IRQA, IRQB, etc.) and can use any of the reserved or otherwise unused starting addresses provided they have been pre-programmed in the DSP. Non-maskable interrupts of DSP56300 Core can be forced by the host processor by setting the host non-maskable interrupt (HNMI) bit in the HCVR. When HNMI set is recognized by the HI32 command interrupt logic, the host command interrupt is processed with the highest priority regardless of the current HI32 interrupt priority (as written in the DSP56300 Core peripheral priority register (IPRP)).

> **CAUTION:** *HV6-HV0 should not be used with a value of zero - the reset location, as this location is normally programmed with a JMP instruction. Doing so will cause an improper short interrupt.*

The personal hardware reset sets HV to the default host command vector, which is via programmable (see Section 6.8 on page 6-89).

6.2.3.3 Host Non-Maskable Interrupt (HNMI) Bit 15

The HNMI bit is used by the host processor to force the generation of the host command as non-maskable interrupt request. If HNMI and HC are set, the host command interrupt is processed with the highest priority regardless of the current HI32 interrupt priority (as written in the DSP56300 Core peripheral priority register (IPRP)). If HNMI is cleared and HC is set, the host command interrupt is processed in accordance with the priority programmed in the IPRP register, and can be disabled by clearing HCIE in the DCTR.

The personal hardware reset clears HNMI.

6.2.3.4   HCVR Reserved Bits 31-16, 14-8

These unused bits are reserved for future expansion and should be written with zeros for upward compatibility. They are read by the host processor as zeros.

6.2.4   Host Slave Receive Data Register (HRXS)

The HRXS is the output stage of the slave DSP-to-host data path FIFO used for DSP-to-host data transfers. The HRXS cannot be accessed by the DSP56300 Core.

The HRXS contains valid data when the HRRQ bit is set. Emptying the HRXS by host processor reads clears HRRQ.

The HRXS transfers the data to the HI32 data pins via the data transfer format converter (HDTFC). The value of the HRF bits in the HCTR define which bytes of the HRXS are output to the pins and their alignment. (See Section 6.1.9 on page 66-36 and Section 6-9 on page 6-54).

In PCI mode (HM=$1) memory space read transaction, the HRXS is accessed if the PCI address is between HI32_base_address: $01C and HI32_base_address: $FFFC.

In the PCI mode (HM=$1), HRXS is viewed by the host processor as a 16377 Dword read-only memory.

In PCI DSP-to-host data transfers via the HRXS, all four byte lanes are driven with data, in accordance with HRF1-HRF0 bits, regardless of the value of the byte enable pins (HC3/$\overline{HBE3}$-HC0/$\overline{HBE0}$).

When in a Universal Bus mode (HM=$2 or $3), the HRXS is accessed if the HA10-HA3 value matches the HI32 base address (CBMA, see Section 6.2.11 on page 6-70) and the HA2-HA0 value is $7.

In a 24-bit data Universal Bus mode (HM=$2 or $3 and HRF=$0), the HRXS is viewed by the host processor as a 24-bit read-only register. HD23-HD0 pins are driven with all three bytes of the HRXS in a read access.

In a 16-bit data Universal Bus mode (HM=$2 or $3 and HRF≠$0), the HRXS is viewed by the host processor as a 16-bit read-only register. In a read access, the HD15-HD0 pins are driven with data from the two most significant bytes or two least significant bytes of the HRXS, as defined by the HRF bits in the HCTR.

When HRRQ is set and RREQ in the HCTR is set:
- *the HREQ status bit will be set in the HSTR.*
- *the $\overline{HIRQ}$ pin will be asserted - if DMAE is cleared (in the Universal Bus modes)*
- *the HDRQ pin will be asserted - if DMAE is set (in the Universal Bus modes)*

If TWSD is cleared, the HI32 as the selected PCI target (HM=$1) in a read data phase from the HRXS will insert PCI wait states if the HRXS is empty (HRRQ=0). Wait states will be inserted until the data is transferred from the DSP side to the HRXS. Up to eight wait states may be inserted before a target initiated transaction termination (disconnect-C/Retry) will be generated.

In a Universal Bus mode read from the HRXS the HI32 will insert wait states if the HRXS is empty (HRRQ=0). Wait states will be inserted until the data is transferred from the DSP side to the HRXS.

Hardware, software and personal software resets empty the HRXS (HRRQ is cleared).

6.2.5   Host Master Receive Data Register (HRXM)

The HRXM is the output stage of the master DSP-to-host data path FIFO used for DSP-to-host data transfers. The HRXM cannot be accessed by the DSP56300 Core or the host.

The HRXM transfers the data to the HI32 data pins via the data transfer format converter (HDTFC). The value of the FC bits in the DPMC define which bytes of the HRXM are output to the pins and their alignment. (See Section 6.1.9 on page 66-36 and Section 6-3 on page 66-24).

In the PCI mode (HM=$1) the DSP56300 Core can clear the HI32 master-to-host bus data path and empty HRXM by setting the CLRT bit in the DPCR.

In PCI DSP-to-host data transfers via the HRXM, all four byte lanes are driven with data, in accordance with FC1-FC0 bits, regardless of the value of the byte enable pins ($HC3/\overline{HBE3}$-$HC0/\overline{HBE0}$).

Hardware, software and personal software resets empty the HRXM.

6.2.6   Host Transmit Data Register (HTXR)

The HTXR is the input stage of the host-to-DSP data path FIFO used for host-to-DSP data transfers. The HTXR cannot be accessed by the DSP56300 Core.

The HTXR may be written if the HTRQ bit in the HSTR is set. Data should not be written to the HTXR until HTRQ is set to prevent previous data from being overwritten. Filling the HTXR by host processor writes, clears HTRQ.

The HTXR receives data from the HI32 data pins via the data transfer format converter (HDTFC). The value of the FC bits in the HCTR or the HTF bits in the HCTR define which bytes of the PCI bus are written to the HTXR and their alignment. (See Table 6-3, on page 66-24, Section 6.1.7 on page 66-35 and Table 6-8, on page 6-51).

In the PCI mode (HM=$1):
As the active target, in a memory space write transaction, the HTXR is accessed if the PCI

address is between HI32_base_address: $01C and HI32_base_address: $FFFC (i.e. the HTXR is viewed by the host processor as a 16377 Dword write-only memory).
As the active master, all data read from the target being accessed is written to the HTXR.

In PCI host-to-DSP data transfers, data is written to the HTXR FIFO, in accordance with FC1-FC0 or HTF1-HTF0 bits, regardless of the value of the byte enable pins (HC3/$\overline{HBE3}$-HC0/$\overline{HBE0}$).

In a Universal Bus mode (HM=$2 or $3), the HTXR is accessed if the HA10-HA3 value matches the HI32 base address (CBMA, see Section 6.2.11 on page 6-70) and the HA2-HA0 value is $7.

In a 24-bit data Universal Bus mode (HM=$2 or $3 and HTF=$0), the HTXR is viewed by the host processor as a 24-bit write-only register. HD23-HD0 pins are written to all three bytes of the HTXR in a write access.

In a 16-bit data Universal Bus mode (HM=$2 or $3 and HTF≠$0), the HTXR is viewed by the host processor as a 16-bit write-only register. In a write access, the HD15-HD0 pins are written to the two most significant bytes or least significant bytes of the HTXR, as defined by the HTF bits in the HCTR.

When HTRQ is set and TREQ in the HCTR is set:
  - *the HREQ status bit will be set in the HSTR.*
  - *the $\overline{HIRQ}$ pin will be asserted - if DMAE is cleared (in the Universal Bus modes)*
  - *the HDRQ pin will be asserted - if DMAE is set (in the Universal Bus modes)*

If TWSD is cleared, the HI32 as the selected PCI target (HM=$1) in a write data phase to the HTXR will insert PCI wait states if the HTXR is full (HTRQ=0). Wait states will be inserted until the data is transferred from the HTXR to the DSP side. Up to eight wait states may be inserted before a target initiated transaction termination (disconnect-C/Retry) will be generated.

In a Universal Bus mode write to the HTXR the HI32 will insert wait states if the HTXR is full (HTRQ=0). Wait states will be inserted until the data is transferred from the HTXR to the DSP side.

Hardware, software and personal software resets empty the HTXR (HTRQ is set).

6.2.7 Device/Vendor ID Configuration Register (CDID/CVID)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VID15	VID14	VID13	VID12	VID11	VID10	VID9	VID8	VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0

Bit		Name	Value (hardwired)	Function
CVID	15-0	VID15-VID0	$1057	Vendor ID
CDID	31-16	DID15-DID0	via programmable see Section 6.8 on page 6-89	Device ID

The CDID/CVID is a PCI standard 32-bit read-only register mapped into the PCI configuration space, when in the PCI mode or in mode 0 (HM=$1 or $0). CDID/CVID is accessed if a configuration read command is in progress and the PCI address is $00.

The DID15-DID0 bits identify the DSP. The VID15-VID0 bits identify the manufacturer of the DSP. The contents of CDID/CVID is hardwired and cannot be affected by any type of reset.

The CDID/CVID cannot be accessed by the host when not in the PCI mode (HM≠$1).

6.2.8 Status/Command Configuration Register (CSTR/CCMR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DPE	SSE	RMA	RTA	STA	DST1	DST0	DPR	FBBC							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SERE	WCC	PERR				BM	MSE	

☐ *Not implemented, read as zero, should be written zero*   ☐ *Reserved, read as zero and should be written zero*
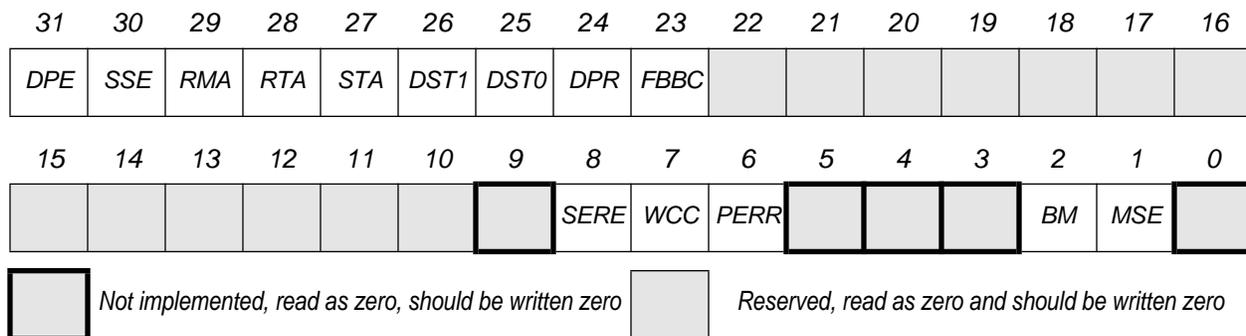
	Bit	Name	Function
CCMR	1	MSE	Memory Space Enable
	2	BM	Bus Master Enable
	6	PERR	Parity Error Response
	7	WCC	Wait Cycle Control (hardwired to zero)
	8	SERE	System Error Enable
	9,5-3, 0	not implemented	
	15-10	reserved	
CSTR	23	FBBC	Fast Back-to-Back Capable (hardwired to one)
	24	DPR	Data Parity Reported
	26-25	DST1-DST0	DEVSEL Timing (hardwired to $1)
	27	STA	Signaled Target Abort
	28	RTA	Received Target Abort
	29	RMA	Received Master Abort
	30	SSE	Signaled System Error
	31	DPE	Detected Parity Error
	22-16	reserved	

The CSTR/CCMR is a PCI standard 32-bit read/write register mapped into the PCI configuration space, when in the PCI mode or in mode 0 (HM=$1 or $0). CSTR/CCMR is accessed if a configuration read/write command is in progress and the PCI address is $04. In the Self Configuration mode (HM=$5): the DSP56300 Core can indirectly access the CCMR. (See "SELF CONFIGURATION MODE" on page 73.)

The CSTR/CCMR is written by the host in accordance with the byte enables. Byte lanes that are not enabled are not written and the corresponding bits remain unchanged.

The CSTR/CCMR cannot be accessed by the host when not in the PCI mode (HM≠$1).

The CSTR/CCMR bits are described in the following paragraphs.

6.2.8.1 Memory Space Enable (MSE) Bit 1

The MSE bit is used to control the HI32 response to the PCI memory space accesses, when in the PCI mode (HM=$1). The HI32 memory space response is disabled if MSE is cleared and enabled if MSE is set.

The personal hardware reset clears MSE.

6.2.8.2 Bus Master Enable (BM) Bit 2

The BM bit is used to control the HI32 ability to act as a master on the PCI bus, when in the PCI mode (HM=$1). If BM is cleared, the HI32 is disabled from acting as a bus master. If BM is set, the HI32 can function as a bus master. This bit affects the MARQ bit in the DSP side status register (DPSR): if BM is cleared, MARQ is also cleared.

The personal hardware reset clears BM.

6.2.8.3 Parity Error Response (PERR) Bit 6

The PERR bit is used to control the HI32 response to parity errors, when in the PCI mode (HM=$1). If PERR is cleared: the HI32 does not drive $\overline{HPERR}$. If PERR is set: if a parity error is detected the HI32 pulses the $\overline{HPERR}$ pin. If a parity error or $\overline{HPERR}$ low is detected, the HI32 sets the DPR bit in the CSTR/CCMR

In both cases the HI32 sets bit 15 (DPE) in the CSTR/CCMR, sets DPER in the DPSR, and generates a parity error interrupt request if PEIE, in the DPCR, is set.

The personal hardware reset clears PERE.

6.2.8.4 Wait Cycle Control (WCC) Bit 7

The WCC bit is hardwired to zero, as the HI32 never executes address stepping.

6.2.8.5 System Error Enable (SERE) Bit 8

The SERE bit is used to enable the $\overline{HSERR}$ pin driving by the HI32, when in the PCI mode (HM=$1). If SERE is cleared, the $\overline{HSERR}$ pin disabled (i.e. high impedance). If SERE is set: if the force system error (SERF) bit in the DPCR is set and the HI32 is an active PCI agent, or an address parity error was detected, the HI32 pulses the $\overline{HSERR}$ pin and sets the signalled system error (SSE) bit in the CSTR.

The personal hardware reset clears SERE.

6.2.8.6 Fast Back-to-Back Capable (FBBC) Bit 23

The FBBC indicates the HI32 supports fast back-to-back transactions as a target, when in the PCI mode (HM=$1). This bit is hardwired to one.

6.2.8.7 Data Parity Reported (DPR) Bit 24

The DPR indicates the data parity error detected, when in the PCI mode (HM=$1). The DPR is set if the HI32 acts as a bus master and detects a data parity error or samples $\overline{HPERR}$ asserted while PERR bit is set in CCMR. The DPR bit is cleared when it is written with one by the host processor.

The personal hardware reset clears DPR.

6.2.8.8 DEVSEL Timing (DST1-DST0) Bits 26 and 25

The DST1-DST0 bits encode the timing of the $\overline{HDEVSEL}$ pin, when in the PCI mode (HM=$1). These bits are hardwired to DST=$1, indicating that the HI32 belongs to the 'medium DEVSEL timing' class of the PCI devices.

6.2.8.9 Signaled Target Abort (STA) Bit 27

The STA indicates a target-abort PCI bus event has been generated. When in the PCI mode (HM=$1) and the HI32, as a target device, terminates a transaction with target-abort, the STA is set. The STA bit is cleared when it is written with one by the host processor.

The personal hardware reset clears STA.

6.2.8.10 Received Target Abort (RTA) Bit 28

The RTA indicates a target-abort PCI bus event has been generated. When in the PCI mode (HM=$1) and the HI32, as a master device, detects that its transaction is terminated with target-abort, the RTA is set. The RTA bit is cleared when it is written with one by the host processor.

The personal hardware reset clears RTA.

6.2.8.11 Received Master Abort (RMA) Bit 29

The RMA indicates a master-abort PCI bus state has been generated. When in the PCI mode (HM=$1) and the HI32, as a master device, terminates its transaction with master-abort, the RMA is set. The RMA bit is cleared when it is written with one by the host processor.

The personal hardware reset clears RMA.

6.2.8.12 Signaled System Error (SSE) Bit 30

The SSE indicates a system error has occurred. When in the PCI mode (HM=$1) and the HI32 asserts $\overline{HSERR}$ pin, the SSE is set. The SSE bit is cleared when it is written with one by the host processor.

The personal hardware reset clears SSE.

6.2.8.13   Detected Parity Error (DPE) Bit 31

The DPE indicates a parity error has been detected by the HI32 hardware. When in the PCI mode (HM=$1) and the HI32 detects either address or data parity error, the DPE is set. The DPE bit is cleared when it is written with one by the host processor.

The personal hardware reset clears DPE.

6.2.8.14   CSTR Reserved Bits 23-16

These unused bits are reserved for future PCI expansion and read by the host processor as zeros.

6.2.8.15   CCMR Reserved Bits 15-10

These bits are reserved for future PCI expansion and should be written with zeros for upward compatibility. They are read by the host processor as zeros.

6.2.8.16   CCMR Not Implemented Bits 9, 5-3

These not implemented bits are reserved for future expansion and should be written with zeros for upward compatibility. They are read by the host processor as zeros.

6.2.9   Class Code/Revision ID Configuration Register (CCCR/CRID)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0	RID7	RID6	RID5	RID4	RID3	RID2	RID1	RID0

	Bit	Name	Value (hardwired)	Function
CRID	7-0	RID7-RID0	*via programmable see Section 6.8 on page 6-89*	*Revision ID*
CCCR	15-8	PI7-PI0		*PCI Device Program Interface*
	23-16	SC7-SC0		*PCI Device Sub-Class*
	31-24	BC7-BC0		*PCI Device Base Class*

The CCCR/CRID is a PCI standard 32-bit read-only register mapped into the PCI configuration space, when in the PCI mode or in mode 0 (HM=$1 or $0). CCCR/CRID is accessed if a configuration read command is in progress and the PCI address is $08.

The RID7-RID0 bits specify the DSP specific identifier (as en extension of Device ID).

The CCCR/CRID cannot be accessed by the host when not in the PCI mode (HM≠$1)

The contents of CCCR/CRID is hardwired and are not affected by any type of reset.

6.2.10   Header Type/Latency Timer Configuration Register (CHTY/CLAT)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								HT7	HT6	HT5	HT4	HT3	HT2	HT1	HT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LT7	LT6	LT5	LT4	LT3	LT2	LT1	LT0								
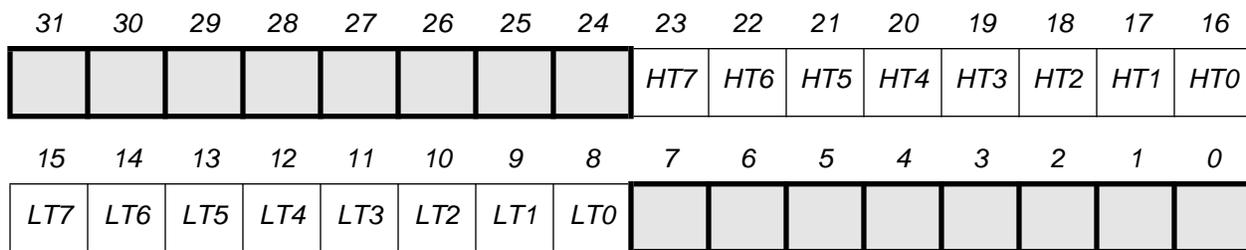
☐   *Not implemented, read as zero and should be written zero*

	Bit	Name	Function
	7-0	not implemented	
CLAT	15-8	LT7-LT0	*Latency Timer (High)*
CHTY	23-16	HT7-HT0	*Header Type (hardwired to $00)*
	31-24	not implemented	

The CHTY/CLAT is a PCI standard 32-bit read/write register mapped into the PCI configuration space, when in the PCI mode or in mode 0 (HM=$1 or $0). The CHTY/CLAT is accessed if a configuration read/write command is in progress and the PCI address is $0C. In the Self Configuration mode (HM=$5): the DSP56300 Core can indirectly access the CLAT (See "SELF CONFIGURATION MODE" on page 73.).

The CHTY/CLAT is written in accordance with the byte enables. Byte lanes that are not enabled are not written and the corresponding bits remain unchanged.

The CHTY/CLAT cannot be accessed by the host when not in the PCI mode (HM≠$1).

The CHTY/CLAT bits are described in the following paragraphs.

6.2.10.1   Header Type (HT7-HT0) Bits 23-16

The read-only bits HT7-HT0 identify the layout of bytes $10-$3F in the configuration space and also whether or not the device contains multiple functions. This byte is hardwired to the value $00.

6.2.10.2   Latency Timer (LT7-LT0) Bits 15-8

The read/write bits LT7-LT0 have two functions:

In the PCI mode (HM=$1): LT7-LT0 specify, in units of PCI bus clock cycles, the value of the latency timer for this PCI bus master.

In the Universal Bus modes (HM=$2,$3) with HIRH cleared: LT7-LT0 specify, in units of DSP56300 Core clock cycles, the duration of the $\overline{HIRQ}$ pulse. The duration of the $\overline{HIRQ}$

pulse is given by the following equation:

$$HIRQ\_PULSE\_WIDTH = (LT[7:0]\_Value + 1) \bullet DSP56300\_Core\_clock\_cycle$$

This bits can be written by the DSP56300 Core in the Self Configuration mode (see Self Configuration Procedure for the Universal Bus Mode on page 73).

The personal hardware reset clears LT7-LT0.

6.2.10.3 CHTY/CLAT Not Implemented Bits 31-24,7-0

These not implemented bits are reserved for future expansion and should be written with zeros for upward compatibility. They are read by the host processor as zeros.

6.2.11 Memory Space Base Address Configuration Register (CBMA)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PM31	PM30	PM29	PM28	PM27	PM26	PM25	PM24	PM23/ GB10	PM22/ GB9	PM21/ GB98	PM20/ GB7	PM19/ GB6	PM18/ GB5	PM17/ GB4	PM16/ GB3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PM15	PM14	PM13	PM12	PM11	PM10	PM9	PM8	PM7	PM6	PM5	PM4	PF	MS1	MS0	MSI

☐ Hardwired to zero

Bit	Name	Function
0	MSI	Memory Space Indicator (Hardwired to zero)
2-1	MS1-MS0	Memory Space (Hardwired to zeros)
3	PF	Pre-fetch (Hardwired to zero)
15-4	PM15-PM4	Memory Base Address Low (Hardwired to zeros)
31-16	PM31-PM16	Memory Base Address High
23-16	GB10-GB3	Universal Bus mode Base Address

The CBMA is a PCI standard 32-bit read/write register mapped into the PCI configuration space, when in the PCI mode or in mode 0 (HM=$1 or $0). The CBMA is accessed if a configuration read/write command is in progress and the PCI address is $10. The CBMA controls the HI32 mapping into the PCI memory space and the Universal Bus mode space. In the Self Configuration mode (HM=$5): the DSP56300 Core can indirectly access the CBMA (See "SELF CONFIGURATION MODE" on page 73.).

The CBMA is written in accordance with the byte enables. Byte lanes that are not enabled are not written and the corresponding bits remain unchanged.

The CBMA cannot be accessed by the host when not in the PCI mode (HM≠$1).

The CBMA bits are described in the following paragraphs.

6.2.11.1 Memory Space Indicator (MSI) Bit 0

The MSI determines that CBMA register maps the HI32 into the PCI memory space. The MSI bit is hardwired to zero and is not affected by any type of reset.

6.2.11.2 Memory Space (MS1-MS0) Bits 2 and 1

The MS1 and MS0 bits encode that CBMA register is 32 bits wide and mapping can be done anywhere in the 32 bit memory space. The MS1 and MS0 are hardwired to zeros and is not affected by any type of reset.

6.2.11.3 Pre-fetch (PF) Bit 3

The PF bit indicates that the data is pre-fetchable or not. PF is hardwired to zero and is not affected by any type of reset.

6.2.11.4 Memory Base Address (PM31-PM16) Bits 31-4

The PM31-PM4 bits define the HI32 base address when it is mapped into the PCI memory space. The PM15-PM4 are hardwired to zero, while PM31-PM16 can be written by the PCI master during system configuration.

The HI32 target occupies 16384 Dwords of the PCI memory space. The HI32 is selected by the 20 most significant PCI address pins HAD31-HAD12, and the twelve least significant address pins HAD11-HAD0 are used to select the HI32 registers on the host side (see Figure 6-3 on page 6-43).

The personal hardware reset clears PM31-PM16.

6.2.11.5 Universal Bus Mode Base Address (GB10-GB3) Bits 23-16

The GB10-GB3 bits define the HI32 base address when it is mapped into the Universal Bus mode space. The remaining CBMA bits are ignored in Universal Bus modes.

The HI32 slave occupies eight locations in the Universal Bus mode space. The HI32 is selected by the eight most significant address pins HA10-HA3, and the three least significant address pins HA2-HA0 are used to select the HI32 registers on the host side.

All reserved register addresses are read as zeros and should be written with zeros for upward compatibility (see Figure 6-5 on page 6-43).

The personal hardware reset clears GB10-GB3.

6.2.12 Interrupt Line - Interrupt Pin Configuration Register (CILP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0	MG7	MG6	MG5	MG4	MG3	MG2	MG1	MG0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0	IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0

☐ Hardwired to zero   ☐ Hardwired to one
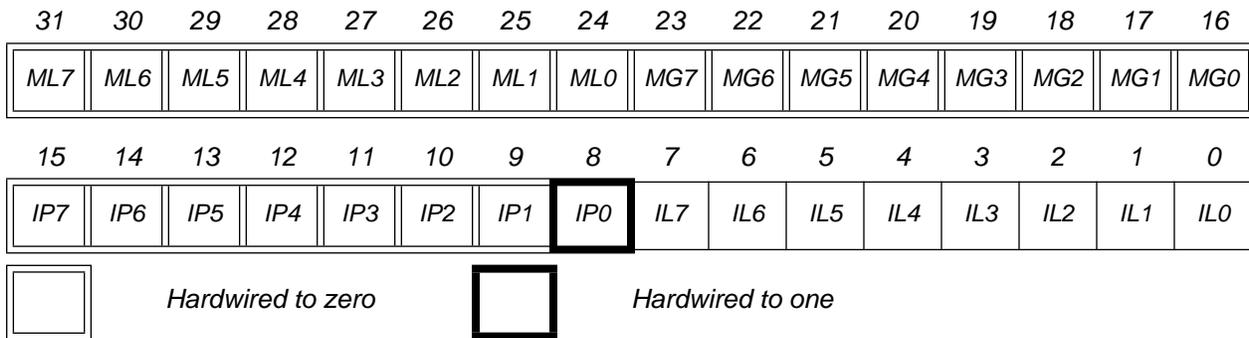
Bit		Name	Value (hardwired)
Interrupt Line	7-0	IL7-IL0	---
Interrupt Pin	15-8	IP7-IP0	$01
MIN_GNT	23-16	MG7-MG0	$00
MAX_LAT	31-24	ML7-ML0	$00

The CILP is a PCI standard 32-bit read-only register mapped into the PCI configuration space, when in the PCI mode or in mode 0 (HM=$1 or $0). CILP is accessed if a configuration read command is in progress and the PCI address is $FC. The CILP register cannot be accessed by the DSP56300 Core.

ML7-ML0: MAX_LAT is used for specifying how often the device needs to gain access to the PCI bus. As the HI32 has no major requirements for the settings of Latency Timers, these bits are hardwired to zero.

MG7-MG0: MIN_GNT is used for specifying how long a burst the device needs. As the HI32 has no major requirements for the settings of Latency Timers, these bits are hardwired to zero.

IP7-IP0: The Interrupt Pin bits specify which interrupt the device uses. A value of 1 corresponds to PCI $\overline{INTA}$.

IL7-IL0: These read/write bits are used to communicate PCI interrupt line routing information. POST software will write the routing information into these bits as it initializes and configures the PCI system.

The CILP cannot be accessed by the host when not in the PCI mode (HM≠$1).

The 24 most significant bits of the CILP register are hardwired and are not affected by any type of reset. The personal hardware reset clears IL7-IL0.

6.3 SELF CONFIGURATION MODE

The Self Configuration mode is used to program the HI32 base address and $\overline{HIRQ}$ pulse width, for operation in the Universal Bus mode; and for programming the configuration registers for operation in a PCI environment without an external system configurator.

In the Self Configuration mode (HM=$5), the DSP56300 Core can indirectly write to all the writable HI32 configuration registers. The DSP56300 Core writes the Dword data to the AR bits of the DPMC and DPAR registers (the remaining bits in these registers are ignored). The two most significant bytes of the Dword are written to the DPMC, the two least significant, to the DPAR. The data is transferred to the configuration register by the HI32 hardware. The registers are written sequentially beginning with the CSTR/CCMR register (location $04). After each write to the DPAR, the data is transferred to the accessed register and an internal pointer is advanced to point to the next Dword location in the configuration space.

> *NOTE: At least one DSP instruction must appear between writing the Self Configuration mode (HM2-HM0 = $5) and first write to the DPAR if the first write is a one DSP clock cycle instruction. (e.g. move immediate and move from external memory are more than one clock cycle)*

6.3.1 Self Configuration Procedure for the PCI Mode

```
M_DCTR    equ   DCTR_ADDR              ; HI32 via programmed address :$5
M_DPMC    equ   DPMC_ADDR              ; HI32 via programmed address :$6
M_DPAR    equ   DPAR_ADDR              ; HI32 via programmed address :$8

movep     #$500000, x: M_DCTR         ; enter self configuration mode
movep     #BASE_ADDRESS, x: M_DPMC    ; CBMA Data (location $10)
movep     #CCMR_DATA, x: M_DPAR       ; write CSTR & CCMR (location $04)
movep     #$0, x: M_DPAR              ; dummy write to location $08
movep     #CLAT_DATA, x: M_DPAR       ; write CLAT (location $0C)
movep     #$0, x: M_DPAR              ; write CBMA (location $10)
movep     #$0, x: M_DCTR              ; return to personal software reset
```

6.3.2 Self Configuration Procedure for the Universal Bus Mode

```
M_DCTR    equ   DCTR_ADDR              ; HI32 via programmed address :$5
M_DPMC    equ   DPMC_ADDR              ; HI32 via programmed address :$6
M_DPAR    equ   DPAR_ADDR              ; HI32 via programmed address :$8

movep     #$500000, x: M_DCTR         ; enter self configuration mode
movep     #BASE_ADDRESS, x: M_DPMC    ; CBMA Data  (location $10)
movep     #$0, x: M_DPAR              ; dummy write to location $04
movep     #$0, x: M_DPAR              ; dummy write to location $08
movep     #HIRQ__DURATION, x: M_DPAR  ; write CLAT (location $0C)
movep     #$0, x: M_DPAR              ; write CBMA (location $10)
```

6.4 HOST PORT PINS

HI32 Port Pin	PCI mode	Enhanced General Bus Mode	General Bus Mode	GPIO Mode
HP0	HAD0	HA3		HIO0
HP1	HAD1	HA4		HIO1
HP2	HAD2	HA5		HIO2
HP3	HAD3	HA6		HIO3
HP4	HAD4	HA7		HIO4
HP5	HAD5	HA8		HIO5
HP6	HAD6	HA9		HIO6
HP7	HAD7	HA10		HIO7
HP8	HAD8	HD0		HIO8
HP9	HAD9	HD1		HIO9
HP10	HAD10	HD2		HIO10
HP11	HAD11	HD3		HIO11
HP12	HAD12	HD4		HIO12
HP13	HAD13	HD5		HIO13
HP14	HAD14	HD6		HIO14
HP15	HAD15	HD7		HIO15
HP16	HC0/$\overline{HBE0}$	HA0		HIO16
HP17	HC1/$\overline{HBE1}$	HA1		HIO17
HP18	HC2/$\overline{HBE2}$	HA2		HIO18
HP19	HC3/$\overline{HBE3}$	UNUSED (Must be forced or pulled to Vcc or GND.)		HIO19
HP20	$\overline{HTRDY}$	$\overline{HDBEN}$	HIO20	
HP21	$\overline{HIRDY}$	$\overline{HDBDR}$	HIO21	
HP22	$\overline{HDEVSEL}$	$\overline{HSAK}$	HIO22	
HP23	$\overline{HLOCK}$	$\overline{HBS}$ (Schmitt trigger buffer on input - should be forced or pulled to Vcc if not used)		HIO23
HP24	HPAR	$\overline{HDAK}$ (Schmitt trigger buffer on input - should be forced or pulled to Vcc if not used)		disconnected
HP25	$\overline{HPERR}$	HDRQ		disconnected
HP26	$\overline{HGNT}$	HAEN		disconnected
HP27	$\overline{HREQ}$	HTA		disconnected
HP28	$\overline{HSERR}$	$\overline{HIRQ}$		disconnected
HP29	$\overline{HSTOP}$	HWR/HRW (Schmitt trigger buffer on input)		disconnected
HP30	HIDSEL	HRD/HDS (Schmitt trigger buffer on input)		disconnected
HP31	$\overline{HFRAME}$	UNUSED (Must be forced or pulled up to Vcc.)		
HP32	HCLK	UNUSED (Must be forced or pulled up to Vcc.)		
HP33	HAD16	HD8 (should be pulled to Vcc or GND if not used)		disconnected
HP34	HAD17	HD9 (should be pulled to Vcc or GND if not used)		disconnected
HP35	HAD18	HD10 (should be pulled to Vcc or GND if not used)		disconnected
HP36	HAD19	HD11 (should be pulled to Vcc or GND if not used)		disconnected
HP37	HAD20	HD12 (should be pulled to Vcc or GND if not used)		disconnected
HP38	HAD21	HD13 (should be pulled to Vcc or GND if not used)		disconnected
HP39	HAD22	HD14 (should be pulled to Vcc or GND if not used)		disconnected
HP40	HAD23	HD15 (should be pulled to Vcc or GND if not used)		disconnected
HP41	HAD24	HD16[a] (should be forced or pulled to Vcc or GND if not used)		disconnected
HP42	HAD25	HD17[a] (should be forced or pulled to Vcc or GND if not used)		disconnected
HP43	HAD26	HD18[a] (should be forced or pulled to Vcc or GND if not used)		disconnected
HP44	HAD27	HD19[a] (should be forced or pulled to Vcc or GND if not used)		disconnected
HP45	HAD28	HD20[a] (should be forced or pulled to Vcc or GND if not used)		disconnected
HP46	HAD29	HD21[a] (should be forced or pulled to Vcc or GND if not used)		disconnected
HP47	HAD30	HD22[a] (should be forced or pulled to Vcc or GND if not used)		disconnected
HP48	HAD31	HD23[a] (should be forced or pulled to Vcc or GND if not used)		disconnected
HP49	$\overline{HRST}$	HRST (Schmitt trigger buffer on input)		
HP50	$\overline{HINTA}$			

a. HD23-HD16 Output is high impedance if HRF≠$0. Input is disconnected if HTF≠$0.

Table 6-10. Host Port Pins - Detailed Description *(Sheet 1 of 8)*

HI32 Port Pin[a]	HI32 Mode			
	PCI	*Enhanced Universal*	*Uni-versal*	*GPIO*
HP7-HP0	*HAD15-HAD0* *Address/Data Multiplexed Bus* *Tri-state, bidirectional bus.* *During the first clock cycle of a transaction HAD31-HAD0 contain the physical byte address (32 bits). During subsequent clock cycles, HAD31-HAD0 contain data.*	*HA10-HA3* *Address Bus* *Input pin.* *This bus selects the HI32 register to be accessed. HA10-HA3 select the HI32 and HA2-HA0 select the particular register of the HI32 to be accessed.*		*HIO15-HIO0* *GPIO[b]*
HP15-HP8		*HD7-HD0* *Data Bus* *Tri-state, bidirectional bus.* *Used to transfer data between the host processor and the HI32.* *This bus is released (disconnected) when the HI32 is not selected by HA10-HA0. The HD23-HD0 pins are driven by the HI32 during a read access, and are inputs to the HI32 during a write access.* *HD23-HD16 outputs are high impedance if HRF≠$0.* *HD23-HD16 inputs are disconnected if HTF≠$0.*		
HP18-HP16	*HC3/HBE3-HC0/HBE0* *Bus Command/Byte Enables* *Tri-state, bidirectional bus.* *During the address phase of a transaction, HC3/HBE3-HC0/HBE0 define the bus command. During the data phase HC3/HBE3-HC0/HBE0 are used as byte enables. The byte enables determine which byte lanes carry meaningful data.*	*HA2-HA0* *Address Bus* *Input pin.* *This bus selects the HI32 register to be accessed. HA10-HA3 select the HI32 and HA2-HA0 select the particular register of the HI32 to be accessed.*		*HIO18-HIO16* *GPIO[b]*
HP19		*UNUSED* *Must be forced or pulled to Vcc or GND.*		*HIO19* *GPIO[b]*

Table 6-10. Host Port Pins - Detailed Description *(Sheet 2 of 8)*

HI32 Port Pin[a]	HI32 Mode			
	PCI	Enhanced Universal	Uni-versal	GPIO
HP20	$\overline{HTRDY}$ Target Ready Sustained tri-state bidirectional pin.[d] Indicates the target agent's ability to complete the current data phase of the transaction. $\overline{HTRDY}$ is used in conjunction with $\overline{HIRDY}$. A data phase is completed on any clock both $\overline{HIRDY}$ and $\overline{HTRDY}$ are sampled asserted. $\overline{HTRDY}$ is asserted if: • during a data read valid data is present on HAD31-HAD0 (HRRQ=1 in the HSTR). • during a data write it indicates the HI32 is ready to accept data (HTRQ=1 in the HSTR). • during a vector write it indicates the HI32 is ready to accept a new host command (HC=0 in the HCVR). Wait cycles are inserted until both $\overline{HIRDY}$ and $\overline{HTRDY}$ are asserted together.	HDBEN Host Data Bus Enable Output pin. Asserted during HI32 accesses. When asserted the external (optional) data transceiver outputs are enabled. When negated the external transceiver outputs are high impedance.	HIO20 GPIO[b]	
HP21	$\overline{HIRDY}$ Initiator Ready Sustained tri-state bidirectional pin.[d] Indicates the initiating agent's ability to complete the current data phase of the transaction. $\overline{HIRDY}$ is used in conjunction with $\overline{HTRDY}$. A data phase is completed on any clock both $\overline{HIRDY}$ and $\overline{HTRDY}$ are sampled asserted. Wait cycles are inserted until both $\overline{HIRDY}$ and $\overline{HTRDY}$ are asserted together. The HI32 negates $\overline{HIRDY}$ if it cannot complete the next data phase.	HDBDR Host Data Bus Direction Output pin. HDBDR is driven high on write data transfers and driven low on read data transfers. This pin is normally high.	HIO21 GPIO[b]	

Table 6-10. Host Port Pins - Detailed Description *(Sheet 3 of 8)*

HI32 Port Pin[a]	HI32 Mode			
	PCI	Enhanced Universal	Universal	GPIO
HP22	$\overline{HDEVSEL}$ Device Select Sustained tri-state bidirectional pin.[d] When actively driven, indicates the driving device has decoded its address as a target of the current access. As an input it indicates whether any device on the bus has been selected.	HSAK Host Select Acknowledge Active low output pin. Used to acknowledge the host processor that the HI32 has identified its address as a slave. HSAK is asserted when the HI32 is the selected slave; otherwise HSAK is released.	HIO22 GPIO[b]	
HP23	$\overline{HLOCK}$ Lock Sustained tri-state bidirectional pin.[d] $\overline{HLOCK}$ indicates an atomic operation that may require multiple transactions to complete. When $\overline{HLOCK}$ is asserted, non-exclusive transactions to the HI32 will be 'retried' (i.e. this is an entire resource lock).	$\overline{HBS}$ (Bus Strobe) Schmitt trigger input pin. Asserted at the start of a bus cycle (for half of a clock cycle) providing an "early bus start" signal. This enables the HI32 to respond ($\overline{HTA}$ valid) earlier. $\overline{HBS}$ should be forced or pulled up to Vcc if not used (e.g. ISA bus).	HIO23 GPIO[b]	
HP24	HPAR Parity Tri-state, bidirectional pin. Even parity across HAD31-HAD0 and HC3/$\overline{HBE3}$-HC0/$\overline{HBE0}$. The master drives HPAR during address and write data phases; the target drives HPAR during read data phases.	$\overline{HDAK}$ Host DMA Acknowledge Schmitt trigger input pin. $\overline{HDAK}$ indicates that the external DMA channel is accessing the HI32. The HI32 is selected as a DMA device if $\overline{HDAK}$ and HWR or HRD (in the double-strobe mode) or $\overline{HDAK}$ and HDS (in the single-strobe mode) are asserted. $\overline{HDAK}$ should be forced or pulled up to Vcc if not used.		Disconnected
HP25	$\overline{HPERR}$ Parity Error Sustained tri-state bidirectional pin.[d] Used for reporting of data parity errors. $\overline{HPERR}$ must be driven active (by the agent receiving data) two clocks following the data (i.e. one clock following the HPAR signal) when a data parity error is detected.	HDRQ DMA Request Output Pin. Used to support ISA/EISA-type DMA data transfers. HDRQ is asserted by the HI32 when a DMA request (receive and/or transmit) is generated in the HI32. HDRQ is negated when the DMA request source is cleared ($\overline{HDAK}$ is asserted), masked (by RREQ=0 or TREQ=0) or disabled (DMAE=0). The polarity of HDRQ pin is controlled by HDRP bit in the DCTR.		Disconnected

Table 6-10. Host Port Pins - Detailed Description *(Sheet 4 of 8)*

HI32 Port Pin[a]	HI32 Mode			
	PCI	Enhanced Universal	Uni-versal	GPIO
HP26	$\overline{HGNT}$ Bus Grant Input pin. Indicates to the HI32 that it has been granted mastership of the bus. If not used this pin should be forced or pulled up to Vcc.	HAEN Host Address Enable Input pin. Enables ISA/EISA DMA / I/O type accesses. When high, the HI32 will respond to DMA cycles only (if DMAE=1 in the DCTR, if DMAE=0 the HI32 will ignore the access). When low, the HI32 responds when it identifies its address (i.e. ISA/EISA DMA / I/O type-space accesses).		*Disconnected*
HP27	$\overline{HREQ}$ Bus Request Tri-state, Output pin. Indicates to the arbiter that the HI32 desires use of the bus. $\overline{HREQ}$ is negated in the same PCI clock that the HI32 asserts $\overline{HFRAME}$. As during the STOP reset $\overline{HREQ}$ is high impedance, an external pull-up should be connected if it is connected to the PCI bus arbiter.	HTA Host Transfer Acknowledge Tri-state, Output pin. Used for high speed data transfer between the HI32 and an external host, when the host uses a non-interrupt driven handshake mechanism. If the HI32 negates HTA at the beginning of the host access, the host should extend the access as long as HTA is negated. The polarity of the HTA pin is controlled by HTAP in the DCTR. The HTA pin is asserted if: • during a data read valid data is present on HD23-HD0 (HRRQ=1 in the HSTR). • during a data write it indicates the HI32 is ready to accept data (HTRQ=1 in the HSTR). • during a vector write it indicates the HI32 is ready to accept a new host command (HC=0 in the HCVR).		*Disconnected*

Table 6-10. Host Port Pins - Detailed Description *(Sheet 5 of 8)*

HI32 Port Pin[a]	HI32 Mode			
	PCI	*Enhanced Universal*	*Uni-versal*	*GPIO*
HP28	$\overline{HSERR}$ System Error Active low, open drain output pin[c]. Used for reporting address parity errors and other errors where the result will be catastrophic. Asserted for a single PCI clock by the HI32.	$\overline{HIRQ}$ Host Interrupt Request Active low, output pin[c]. Used by the HI32 to request service from the host processor. $\overline{HIRQ}$ may be connected to an interrupt request pin of a host processor, a transfer request of a DMA controller or a control input of external circuitry. $\overline{HIRQ}$ is initially asserted by the HI32 when an interrupt request is enabled (TREQ=1 or RREQ=1) and the corresponding data path is ready for a data transfer. If the HIRH bit in the DCTR is cleared: $\overline{HIRQ}$ assertion is a pulse who's width is controlled by the CLAT register. If HIRH is set: $\overline{HIRQ}$ is negated at the begining of a corresponding host data access (read or write), or masked (by TREQ=0 or RREQ=0) or disabled (DMAE=1). $\overline{HIRQ}$ will be asserted again, after the host access (regardless of the HIRH value), if enabled and the corresponding data path is ready for a data transfer. The $\overline{HIRQ}$ drive (driven or open drain) is controlled by the HIRD bit in the DCTR.		*Disconnected*
HP29	$\overline{HSTOP}$ Stop Sustained tri-state bidirectional pin.[d] Indicates the current target is requesting the master to stop the current transaction.	$\overline{HWR/HRW}$ Host Write/Read-Write Schmitt trigger input pin. When in the double-strobe mode of the HI32 (HDSM=0), this pin functions as host write input strobe (HWR). The host processor initiates a write access by asserting HWR. Data input is latched with the rising edge of HWR. When in the single-strobe mode of the HI32 (HDSM=1), this pin functions as host read-write (HRW) input. It selects the direction of data transfer for each host processor access: from the HI32 to the host processor when HRW is asserted and from the host processor to the HI32 when HRW is negated. The polarity of the HRW pin is controlled by HRWP bit in the DCTR. **NOTE**: The simultaneous assertion of HRD and HWR is illegal.		*Disconnected*

Table 6-10. Host Port Pins - Detailed Description *(Sheet 6 of 8)*

HI32 Port Pin[a]	HI32 Mode			
	PCI	*Enhanced Universal*	*Uni-versal*	*GPIO*
HP30	*HIDSEL* *Initialization Device Select* *Input pin.* *Used as a chip select in lieu of the upper 21 address lines during configuration read and write transactions.*	*HRD/HDS* *Host Read/Data Strobe* *Schmitt trigger input pin.* *When in the double-strobe mode of the HI32 (HDSM=0), this pin functions as the host read strobe (HRD). The host processor initiates a read access by asserting HRD. Data output may be latched with the rising edge of HRD.* *When in the single-strobe mode of the HI32 (HDSM=1), this pin functions as the host data strobe (HDS). The host processor initiates a read access by asserting HDS with HRW asserted. Data output may be latched with the rising edge of HDS. The host processor initiates a write access by asserting HDS with HRW negated. Data input is latched by the HI32 with the rising edge of HDS.* **NOTE***: The simultaneous assertion of HRD and HWR is illegal.*	*Disconnected*	
HP31	$\overline{HFRAME}$ *Cycle Frame* *Sustained tri-state bidirectional pin.[d]* *Driven By the current master to indicate the beginning and duration of an access.* $\overline{HFRAME}$ *is negated in the final data phase of the transaction.*	*UNUSED* *Must be forced or pulled up to Vcc.*		
HP32	*HCLK* *Bus Clock* *Input pin.* *Provides timing for all transactions on PCI. All other PCI signals are sampled on the HCLK rising edge.*	*UNUSED* *Must be forced or pulled up to Vcc.*		

Table 6-10. Host Port Pins - Detailed Description *(Sheet 7 of 8)*

HI32 Port Pin[a]	HI32 Mode			
	PCI	*Enhanced Universal*	*Universal*	*GPIO*
HP40-HP33	*HAD31-HAD16* *Address/Data Multiplexed Bus* *Tri-state, bidirectional bus.* *During the first clock of a transaction HAD31-HAD0 contain the physical byte address (32 bits). During subsequent clock HAD31-HAD0 contain data.*	*HD15-HD8* *Data Bus* *Tri-state, bidirectional bus.* *Used to transfer data between the host processor and the HI32.* *This bus is released (disconnected) when the HI32 is not selected by HA10-HA0. The HD15-HD0 pins are driven by the HI32 during a read access, and are inputs to the HI32 during a write access.* *When operating with a host bus less than 16 bits wide, the HD15-HD8 pins that are not used for transferring data must be pulled to Vcc or to GND. For example: when operating with a 8-bit bus, HP40-HP33 must be pulled up to Vcc or pulled down to GND.* **NOTE***: It is recommended to pull these unused data lines to GND, as pulling these lines to Vcc will set the corresponding bits in the HCTR, when the external host writes to this register.*		*Disconnected*
HP48-HP41		*HD23-HD16* *Data Bus* *Tri-state, bidirectional bus.* *Used to transfer data between the host processor and the HI32.* *This bus is released (disconnected) when the HI32 is not selected by HA10-HA0. The HD23-HD16 pins are driven by the HI32 during a read access, and are inputs to the HI32 during a write access.* *HD23-HD16 outputs are high impedance if HRF≠$0.* *HD23-HD16 inputs are disconnected if HTF≠$0.* *When operating with a host bus less than 24 bits wide, the data pins that are not used for transferring data must be forced or pulled to Vcc or to GND. For example: when operating with a 16-bit bus (e.g. ISA bus), HP48-HP41 must be forced or pulled up to Vcc or pulled down to GND.* **NOTE***: It is recommended to force or pull these unused data lines to GND, as forcing or pulling these lines to Vcc will set the corresponding bits in the HCTR, when the external host writes to this register.*		*Disconnected*

Table 6-10. Host Port Pins - Detailed Description *(Sheet 8 of 8)*

HI32 Port Pin[a]	HI32 Mode			
	PCI	Enhanced Universal	Universal	GPIO
HP49	$\overline{HRST}$ Hardware Reset Input pin. Forces the HI32 PCI sequencer to the initial state. All pins are forced to the disconnected state. $\overline{HRST}$ is asynchronous to HCLK.	HRST Hardware Reset Schmitt trigger input pin. Forces the HI32 to its initial state. All pins are forced to the disconnected state. The polarity of the HRST pin is controlled by HRSP bit in the DCTR.		
HP50	$\overline{HINTA}$ Host Interrupt A Active low, open drain output pin[c]. Used by the HI32 to request service from the host processor. $\overline{HINTA}$ may be connected to an interrupt request pin of a host processor, a control input of external circuitry, or be used as a general purpose open-drain output. $\overline{HINTA}$ is asserted by the HI32 when HINT, in the DCTR, is set by the DSP56300 Core. $\overline{HINTA}$ is released (high impedance) when HINT, in the DCTR, is cleared by the DSP56300 Core. $\overline{HINTA}$ is asynchronous to HCLK.			

a. This list does not include Vcc and Ground supply pins.

b. The GPIO pin is controlled by the corresponding bits in the GPIO data (DATH) and GPIO direction (DIRH) registers.

c. Open Drain output pin is driven, when asserted, by the HI32. When negated the pin is released (high impedance). This enables using a multi-slave configuration. An external pull-up, must be connected externally for proper operation.

d. Sustained Tri-State is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives this pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving a sustained tri-state signal any sooner that one clock after the previous owner tri-states it. A pull-up resistor is required to sustain the inactive state until another agent drives it.

6.5 HI32 CONTROL, STATUS AND CONFIGURATION REGISTERS

Legend

- (n) = in Self Configuration mode the nth DPAR write accesses this register
- 0 = hardwired to zero
- (shaded) = reserved – hardwired to zero
- 1 (black) = hardwired to one

DSP Side Registers

NAME	CORE Access	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCTR	$5																		HINT	HF5	HF4	HF3	SRIE	STIE	HCIE
DPCR	$6			IAE	RBLE	MWSD	MACE		SERF	MTT	CLRT		TCIE			TTIE		TAIE		PEIE	MAIE		MRIE	MTIE	
DPMC	$7	FC1	FC0	HM2	HM1	HM0	HIRD	HIRH	HRSP	HDRP	HTAP	HRWP	HDSM												
DPAR	$8	BE3	BE2	BE1	BE0	C3	C2	C1	C0	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
DPMC (AR)	$7		FC0	BL5	BL4	BL3	BL2	BL1	BL0	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
DSR	$9	HACT									HF2									HF2	HF1	HF0	SRRQ	STRQ	HCP
DPSR	$A												HDTC	TO	TRTY	TDIS	TAB	MAB	DPER	APER	MARQ		MRRQ	MTRQ	MWS
DTXM	$B	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
DTXS	$C	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
DRXR	$D	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
DIRH	$E	DI23	DI22	DI21	DI20	DI19	DI18	DI17	DI16	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
DATH	$F	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

(DSR also shows HF1, HF2 at bits 4, 5.)

Host Side Registers

NAME	PCI UB Access	CORE Access	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCTR	$10 / $4		HS2	HS1	HS0		HRF1	HRF0			HTF1	HTF0	SFT	DMAE	HF2	HF1	HF0	RREQ	TREQ	HC
HSTR	$14 / $5											HREQ	HINT	HF5	HF4	HF3	HRRQ	HTRQ	TRDY
HCVR	$18 / $6		HNMI									HV6	HV5	HV4	HV3	HV2	HV1	HV0	HC
HTXR/HRXS	$1C–FFFC / $7		D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(HTXR/HRXS is a full 32-bit data register, D0–D31.)

Host Side Registers (PCI Configuration Space)

NAME	PCI UB Access	CORE Access	31	30	29	28	27	26	25	24	23	…	12	…	8	7	6	5	4	3	2	1	0
CDID / CVID	$0		Via Programmable (see Section 6.8 on page 6-89)										1		0	0	1	0	1	0	1	1	1
CSTR / CCMR	$4	(1)	DPE	SSE	RMA	RTA	STA	0	1	DPR	1		0		SERE	0	PERR	0	0	0	BM	MSE	0
CCCR / CRID	$8	$8	Via Programmable (see Section 6.8 on page 6-89)																				
CHTY / CLAT	$c	(3)	0	0	0	0	0	0	0	0	0		0		LT0	0	0	0	0	0	0	0	0
CBMA	$10	(4)	PM31	PM30	PM29	PM28	PM27	PM26	PM25	PM24	PM23 / GB10		PM12		PM16 / GB3	0	0	0	0	0	0	0	0

(CHTY/CLAT latency bits LT0–LT7 occupy bits 8–15; CBMA base bits PM16–PM31 are labeled GB3–GB10 at bits 16–23.)

6.6 HI32 PROGRAMMING MODEL - QUICK REFERENCE

								Reset Type		
	HI32 Registers - **Quick Reference** *(Sheet 1 of 5)*									

<table>
<tr><td colspan="12" align="center">HI32 Registers - Quick Reference <i>(Sheet 1 of 5)</i></td></tr>
<tr><td rowspan="2"></td><td colspan="6"><i>Bit</i></td><td rowspan="2"><i>Comments</i></td><td colspan="3"><i>Reset Type</i></td></tr>
<tr><td><i>Reg</i></td><td><i>Num</i></td><td><i>Mnem.</i></td><td><i>Name</i></td><td><i>Val</i></td><td><i>Function</i></td><td><i>HS</i></td><td><i>PH</i></td><td><i>PS</i></td></tr>
<tr><td colspan="10" align="center"><i>DSP SIDE</i></td></tr>
<tr><td rowspan="15"><i>DCTR</i></td><td><i>0</i></td><td><i>HCIE</i></td><td><i>Host Command Interrupt Enable</i></td><td><i>0</i>
<i>1</i></td><td><i>HCP interrupt disabled</i>
<i>HCP interrupt enabled</i></td><td></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>1</i></td><td><i>STIE</i></td><td><i>Slave Transmit Interrupt Enable</i></td><td><i>0</i>
<i>1</i></td><td><i>STRQ interrupt disabled</i>
<i>STRQ interrupt enabled</i></td><td></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>2</i></td><td><i>SRIE</i></td><td><i>Slave Receive Interrupt Enable</i></td><td><i>0</i>
<i>1</i></td><td><i>SRRQ interrupt disabled</i>
<i>SRRQ interrupt enabled</i></td><td></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>5-3</i></td><td><i>HF5-HF3</i></td><td><i>Host Flags</i></td><td></td><td></td><td></td><td><i>$0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>6</i></td><td><i>HINT</i></td><td><i>Host Interrupt A</i></td><td><i>0</i>
<i>1</i></td><td><i>HINTA pin is high impedance</i>
<i>HINTA pin is driven low</i></td><td></td><td><i>0</i></td><td></td><td></td></tr>
<tr><td><i>13</i></td><td><i>HDSM</i></td><td><i>Host Data Strobe Mode</i></td><td><i>0</i>
<i>1</i></td><td><i>HWR + HRD (double data strobe)</i>
<i>HRW + HDS (single data strobe)</i></td><td><i>may be changed only in PS reset</i></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>14</i></td><td><i>HRWP</i></td><td><i>Host RD/WR Polarity</i></td><td><i>0</i>
<i>1</i></td><td><i>HRW (0=WRITE, 1=READ)</i>
<i>HRW(0=READ, 1=WRITE)</i></td><td><i>may be changed only in PS reset</i></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>15</i></td><td><i>HTAP</i></td><td><i>Host Transfer Acknowledge Polarity</i></td><td><i>0</i>
<i>1</i></td><td><i>HTA</i>
<i>HTA</i></td><td><i>may be changed only in PS reset</i></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>16</i></td><td><i>HDRP</i></td><td><i>Host DMA Request Polarity</i></td><td><i>0</i>
<i>1</i></td><td><i>HDRQ</i>
<i>HDRQ</i></td><td><i>may be changed only in PS reset</i></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>17</i></td><td><i>HRSP</i></td><td><i>Host Reset Polarity</i></td><td><i>0</i>
<i>1</i></td><td><i>HRST</i>
<i>HRST</i></td><td><i>may be changed only in PS reset</i></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>18</i></td><td><i>HIRH</i></td><td><i>Host Interrupt Request Handshake Mode</i></td><td><i>0</i>
<i>1</i></td><td><i>HIRQ pulsed</i>
<i>HIRQ - full handshake</i></td><td><i>may be changed only in PS reset</i>
<i>HIRQ pulse width is defined by CLAT</i></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>19</i></td><td><i>HIRD</i></td><td><i>Host Interrupt Request Drive Control</i></td><td><i>0</i>
<i>1</i></td><td><i>HIRQ - open drain</i>
<i>HIRQ - driven</i></td><td><i>may be changed only in PS reset</i></td><td><i>0</i></td><td>-</td><td>-</td></tr>
<tr><td><i>22-20</i></td><td><i>HM2-HM0</i></td><td><i>HI32 Mode</i></td><td><i>000</i>
<i>001</i>
<i>010</i>
<i>011</i>
<i>100</i>
<i>101</i>
<i>11x</i></td><td><i>Terminate and Reset</i>
<i>PCI</i>
<i>GenBus</i>
<i>Enhanced GenBus</i>
<i>GPIO</i>
<i>Self Configuration</i>
<i>Reserved</i></td><td><i>may be changed to non-zero value only in PS reset</i></td><td><i>$0</i></td><td>-</td><td>-</td></tr>
</table>

Reg	Num	Mnem.	Name	Val	Function	Comments	HS	PH	PS
								Reset Type	
DPCR	1	MTIE	Master Transmit Interrupt Enable	0 1	MTRQ interrupt disabled MTRQ interrupt enabled		0	-	-
	2	MRIE	Master Receive Interrupt Enable	0 1	MRRQ interrupt disabled MRRQ interrupt enabled		0	-	-
	4	MAIE	Master Address Interrupt Enable	0 1	A/DPER interrupt disabled A/DPER interrupt enabled		0	-	-
	5	PEIE	Parity Error Interrupt Enable	0 1	MARQ interrupt disabled MARQ interrupt enabled		0	-	-
	7	TAIE	Transaction Abort Interrupt Enable	0 1	M/TAB interrupt disabled M/TAB interrupt enabled		0	-	-
	9	TTIE	Transaction Termination Int. En.	0 1	TO/DIS/RTY interrupt disabled TO/DIS/RTY interrupt enabled		0	-	-
	12	TCIE	Transfer Complete Interrupt Enable	0 1	HDTC interrupt disabled HDTC interrupt enabled		0	-	-
	14	CLRT	Clear Transmitter	0 1	inactive empty master transmitter path	may be set only if MARQ=1 cleared by hardware	0	-	-
	15	MTT	Master Transfer Terminate	0 1	inactive terminate current PCI transaction	may be set only if MWS=1 cleared by hardware	0	-	-
	16	SERF	$\overline{HSERR}$ Force	0 1	inactive generate a PCI system error	cleared by hardware	0	-	-
	18	MACE	Master Access Counter Enable	0 1	unlimited burst lengh burst length is limited by the BL value		0	-	-
	19	MWSD	Master Wait State Disable	0 1	HI32 master inserts wait states HI32 master releases bus	may be set only if MARQ=1	0	-	-
	20	RBLE	Receive Buffer Lock Enable	0 1	HI32 responds to new accesses HI32 retries accesses after write accesses	may be changed only in PS reset	0	-	-
	21	IAE	Insert Address Enable	0 1	HI32 does not insert address HI32 inserts address in incoming data	may be changed only in PS reset	0	-	-
DPMC	15-0	AR31-AR16	DSP PCI Transaction Address (High)			may be written only if MARQ=1	$0000	-	-
	21-16	BL5-BL0	PCI Data Burst Length			may be written only if MARQ=1	$0	-	-
	23-22	FC1-FC0	Data Transfer Format Control	00 01 10 11	Transmit Receive 32 bit mode 32 bit mode 3 Right, zero ext. 3 LSBs 3 Right, sign ext. 3 LSBs 3 Left, zero filled 3 MSBs	may be written only if MARQ=1	$0	-	-
DPAR	15-0	AR15-AR0	DSP PCI Transaction Address (Low)			may be written only if MARQ=1	$0000	-	-
	19-16	C3-C0	PCI Bus Command			may be written only if MARQ=1	$0	-	-
	23-20	$\overline{BE3}$-$\overline{BE0}$	PCI Byte Enables			may be written only if MARQ=1	$0	-	-

Reg	Num	Mnem.	Name	Val	Function	Comments	HS	PH	PS
DSR	0	HCP	Host Command Pending	0 1	no host command pending host command pending	cleared when the HC interrupt request is serviced	-	-	0
	1	STRQ	Slave Transmit Data Request	1 0	slave transmit FIFO is not full slave transmit FIFO is full	cleared if the DTXS is filled by Core writes	1[(a)]	-	1[(a)]
	2	SRRQ	Slave Receive Data Request	0 1	slave receive FIFO is empty slave receive FIFO is not empty	cleared if the DRXR is emptied by Core reads; or the data to be read from the DRXR is master data.	0	-	0
	5-3	HF2-HF0	Host Flags				-	$0	-
	23	HACT	HI32 Active	0 1	HI32 is in persnal reset (PS) HI32 is active		0	-	0
DPSR	0	MWS	PCI Master Wait States	0 1	HI32 is asserting $\overline{HIRDY}$ HI32 is negating $\overline{HIRDY}$		0	-	0
	1	MTRQ	PCI Master Transmit Data Req.	1 0	master transmit FIFO is not full master transmit FIFO is full	cleared if the DTXM is filled by Core writes	1[(a)]	-	1[(a)]
	2	MRRQ	PCI Master Receive Data Req.	0 1	master receive FIFO is empty master receive FIFO is not empty	cleared if the DRXR is emptied by Core reads; or the data to be read from the DRXR is slave data.	0	-	0
	4	MARQ	PCI Master Address Request	1 0	Core may initiate new transaction Core may not initiate new transaction		0	0	0
	5	APER	PCI Address Parity Error	0 1	HI32 target has not detected an address parity error HI32 target has detected an address parity error	cleared by writing 1	0	-	-
	6	DPER	PCI Data Parity Error	0 1	a data parity error has not occurred a data parity error has occurred	cleared by writing 1	0	-	-
	7	MAB	PCI Master Abort	0 1	a master abort has not occurred a master abort has occurred	cleared by writing 1	0	-	-
	8	TAB	PCI Target Abort	0 1	a target abort has not occurred a target abort has occurred	cleared by writing 1	0	-	-
	9	TDIS	PCI Target Disconnect	0 1	a target disconnect has not occurred a target disconnect has occurred	cleared by writing 1	0	-	-
	10	TRTY	PCI Target Retry	0 1	a target retry has not occurred a target retry has occurred	cleared by writing 1	0	-	-
	11	TO	PCI Time Out Termination	0 1	a time-out termination has not occurred a time-out termination has occurred	cleared by writing 1	0	-	-
	12	HDTC	PCI Host Data Transfer Complete	0 1	HI32 is transfering data to the Core HI32 has completed transfer of data to the Core, and will disconnect write accesses to the HTXR	cleared by writing 1 may be written 1 only if HDTC=1	0	-	0
	21-16	RDC5-RDC0	Remaining Data Count				-	-	-
DRXR	23-0		DSP Receive Data FIFO				*empty*		
DTXM	23-0		DSP Master Transmit Data FIFO				*empty*		
DTXS	23-0		DSP Slave Transmit Data FIFO				*empty*		
DATH	23-0	DAT23-DAT0	GPIO Pin Data				$000000	-	-
DIRH	23-0	DIR23-DIR0	GPIO Pin Direction	[0] [1]	Input Output		$000000	-	-

Reg	Num	Mnem.	Name	Val	Function	Comments	HS	PH	PS
					Bit			*Reset Type*	
					Host Side				
HCTR	1	TREQ	*Transmit Request Enable*	0 1	*HTRQ interrupt disabled* *HTRQ interrupt enabled*		-	*0*	-
	2	RREQ	*Receive Request Enable*	0 1	*HRRQ interrupt disabled* *HRRQ interrupt enabled*		-	*0*	-
	5-3	HF2-HF0	*Host Flags*				-	*0*	-
	6	DMAE	*DMA Enable (ISA/EISA)*	0 1	*HI32 does not support DMA transfers* *HI32 supports ISA-DMA type transfers*		-	*0*	-
	7	SFT	*Slave Fetch Type*	0 1	*Pre-fetch* *Fetch*		-	*0*	-
	9-8	HTF1-HTF0	*Host Transmit Data Transfer Format*	00 01 10 11	*PCI UB* *32 bit mode 24 bit mode* *3 LSBs 2 Right, zero ext.* *3 LSBs 2 Right, sign ext.* *3 MSbs 2 Left, zero filled*		-	*$0*	-
	12-11	HRF1-HRF0	*Host Receive Data Transfer Format*	00 01 10 11	*PCI UB* *32 bit mode 24 bit mode* *3 Right, zero ext. 2 LSBs* *3 Right, sign ext. 2 LSBs* *3 Left, zero filled 2 midlle bytes*		-	*$0*	-
	16-14	HS2-HS0	*Host Semaphores*				-	*0*	-
	19	TWSD	*Target Wait State Disable*	0 1	*HI32 target will insert up to 8 w.s.* *HI32 target will not insert wait states*		-	*0*	-
HSTR	0	TRDY	*Transmitter Ready*	1 0	*transmit FIF O (6 deep) is empty* *transmit FIFO is not empty*		*1*	-	*1*
	1	HTRQ	*Host Transmit Data Request*	1 0	*host transmit FIFO is not full* *host transmit FIFO is full*		*1*	-	*1*
	2	HRRQ	*Host Receive Data Request*	0 1	*host receive FIFO is empty* *host receive FIFO is not empty*		*0*	-	*0*
	5-3	HF5-HF3	*Host Flags*				*0*	-	-
	6	HINT	*Host Interrupt A*	0 1	*$\overline{HINTA}$ pin is high impedance* *$\overline{HINTA}$ pin is driven low*		*0*	-	-
	7	HREQ	*Host Request*	0 1	*$\overline{HIRQ}$ pin is negated* *$\overline{HIRQ}$ pin is asserted (if enabled)*		-	*0*	-
HCVR	0	HC	*Host Command*	0 1	*no host command pending* *host command pending*	*cleared when the HC interrupt request is serviced*	-	-	*0*
	7-1	HV6-HV0	*Host Command Vector*			*default vector via programmable (Section 6.8 on page 6-89)*	-	*default vector*	-
	15	HNMI	*Host Non Maskable Int. Req.*	0 1	*a maskable interrupt request* *a non-maskable interrupt request*		-	*0*	-
HRXM	31-0		*Host Master Receive Data FIFO*				*empty*		
HRXS	31-0		*Host Slave Receive Data FIFO*				*empty*		
HTXR	31-0		*Host Transmit Data FIFO*				*empty*		
CVID CDID	15-0	VID15-VID0	*Vendor ID*	$1057		*hardwired $1057*	-	-	-
	31-16	DID15-DID0	*Device ID*			*via programmable (Section 6.8 on page 6-89)*	-	-	-

HI32 Registers - Quick Reference *(Sheet 5 of 5)*									
	Bit					*Comments*	*Reset Type*		
Reg	*Num*	*Mnem.*	*Name*	*Val*	*Function*		*HS*	*PH*	*PS*
CCMR CSTR	1	MSE	Memory Space Enable	0 1	memory space response is disabled memory space response is enabled		-	0	-
	2	BM	Bus Master Enable	0 1	HI32 PCI bus master disabled HI32 PCI bus master enabled		-	0	-
	6	PERR	Parity Error Response	0 1	HI32 does not drive $\overline{HPERR}$ HI32 drives $\overline{HPERR}$ if a parity error is detected		-	0	-
	7	WCC	Wait Cycle Control	0	HI32 never executes address stepping	hardwired 0	-	-	-
	8	SERE	System Error Enable	0 1	HI32 does not drive $\overline{HSERR}$ HI32 may drive $\overline{HSERR}$		-	0	-
	23	FBBC	Fast Back-to-Back Capable	1	HI32 supports fast back-to-back transactions as a target	hardwired 1	-	-	-
	24	DPR	Data Parity Reported	0 1	no parity error detected HI32 master parity error detected or $\overline{HPERR}$ asserted	cleared by writing 1	-	0	-
	26-25	DST1-DST0	DEVSEL Timing	01	medium DEVSEL timing	hardwired 01	-	-	-
	27	STA	Signaled Target Abort	0 1	HI32 has not generated a target-abort event HI32 target, generated a target-abort event	cleared by writing 1	-	0	-
	28	RTA	Received Target Abort	0 1	HI32 has not received a target-abort event HI32 master, received a target-abort event	cleared by writing 1	-	0	-
	29	RMA	Received Master Abort	0 1	HI32 has not received a master-abort event HI32 master, terminates a transaction with master-abort	cleared by writing 1	-	0	-
	30	SSE	Signaled System Error	0 1	HI32 not asserted $\overline{HSERR}$ HI32 asserted $\overline{HSERR}$	cleared by writing 1	-	0	-
	31	DPE	Detected Parity Error	0 1	no parity error detected parity error detected	cleared by writing 1	-	0	-
CRID CCCR	7-0	RID7-RID0	Revision ID			via programmable *(Section 6.8 on page 6-89)*	-	-	-
	15-8	PI7-PI0	PCI Device Program Interface				-	-	-
	23-16	SC7-SC0	PCI Device Sub-Class				-	-	-
	31-24	BC7-BC0	PCI Device Base Class				-	-	-
CLAT CHTY	15-8	LT7-LT0	Latency Timer				-	$00	-
	23-16	HT7-HT0	Header Type	$0		hardwired $0	-	-	-
CBMA	0	MSI	Memory Space Indicator	0	HI32 is a memory mapped agent	hardwired 0	-	-	-
	2-1	MS1-MS0	Memory Space	$0	32 bits wide and mapping can be done anywhere	hardwired $0	-	-	-
	3	PF	Prefetch	0	HI32 data is not pre-fetchable (in the PCI sense)	hardwired 0	-	-	-
	15-4	PM15-PM4	Memory Base Address Low	$00	64Kbytes occupancy of PCI memory space	hardwired $00	-	-	-
	31-16	PM31-PM16	Memory Base Address High				-	$0000	-
	23-15	GB10-GB3	Genbus Base Address				-	$00	-
CILP	7-0	IP-7-IP0	Interrupt Pin		PCI interrupt line routing information				
	15-8	IL7-IL0	Interrupt Line	$01	$\overline{INTA}$ is supported	hardwired $01			
	23-16	MG7-MG0	MAX_GNT	$00	Min Grant	hardwired $00			
	31-24	ML7-ML0	MAX_LAT	$00	Max Latency	hardwired $00			

a. STRQ. MTRQ are zero in the personal software reset state.

6.7　INTERRUPT VECTORS

Table 6-11.　Interrupt Vectors

Vector / 2	Interrupt	Activated by	Priority
Via Programmable	*Host Command (default)*	*HC (HCVR)*	*Highest*
Via Programmable Base	*PCI Transaction Termination*	*TO, TRTY, TDIS (DPSR)*	
Via Programmable Base + 1	*PCI Transaction Abort*	*TAB, MAB (DPSR)*	
Via Programmable Base + 2	*PCI Parity Error*	*DPER, APER (DPSR)*	
Via Programmable Base + 3	*PCI Transfer Complete*	*HDTC (DPSR)*	
Via Programmable Base + 4	*PCI Master Receive*	*MRRQ (DPSR)*	
Via Programmable Base + 5	*Slave Receive*	*SRRQ (DSR)*	
Via Programmable Base + 6	*PCI Master Transmit*	*MTRQ (DPSR)*	
Via Programmable Base + 7	*Slave Transmit*	*STRQ (DSR)*	
Via Programmable Base + 8	*PCI Master Address*	*MARQ (DPSR)*	*Lowest*

6.8　VIA PROGRAMMING

Below is a table of the DSP56301 via-programmable registers:

Register	Bits	Value	Meaning
CDID	*CDID15-CDID0*	*$1801*	*DSP56301*
CCCR	*CCCR23-CCCR0*	*$048000*	*$04: (Multimedia Device)* *$80: (Other Multimedia Device)* *$00: (Default Program Interface)*
CRID	*CRID7-CRID0*	*$01*	*Rev A*

6.9 EXAMPLES OF HOST TO HI32 CONNECTIONS

6.9.1 Example of Connection to PCI Bus

PCI Bus (initiator/target)		DSP56301 (target/initiator) HI32
AD31-AD0	⟷	HAD31-HAD0
C3/$\overline{\text{BE3}}$-C0/$\overline{\text{BE0}}$	⟷	HC3/$\overline{\text{HBE3}}$-HC0/$\overline{\text{HBE0}}$
PAR	⟷	HPAR
$\overline{\text{FRAME}}$	⟷	$\overline{\text{HFRAME}}$
$\overline{\text{IRDY}}$	⟷	$\overline{\text{HIRDY}}$
$\overline{\text{TRDY}}$	⟷	$\overline{\text{HTRDY}}$
$\overline{\text{DEVSEL}}$	⟷	$\overline{\text{HDEVSEL}}$
$\overline{\text{STOP}}$	⟷	$\overline{\text{HSTOP}}$
$\overline{\text{PERR}}$	⟷	$\overline{\text{HPERR}}$
$\overline{\text{SERR}}$	⟵	$\overline{\text{HSERR}}$
$\overline{\text{REQ}}$	⟵	$\overline{\text{HREQ}}$
$\overline{\text{GNT}}$	⟶	$\overline{\text{HGNT}}$
IDSEL	⟶	HIDSEL
$\overline{\text{RST}}$	⟶	$\overline{\text{HRST}}$
CLK	⟶	HCLK
$\overline{\text{LOCK}}$	⟶	$\overline{\text{HLOCK}}$
$\overline{\text{INTA}}$	⟵	$\overline{\text{HINTA}}$
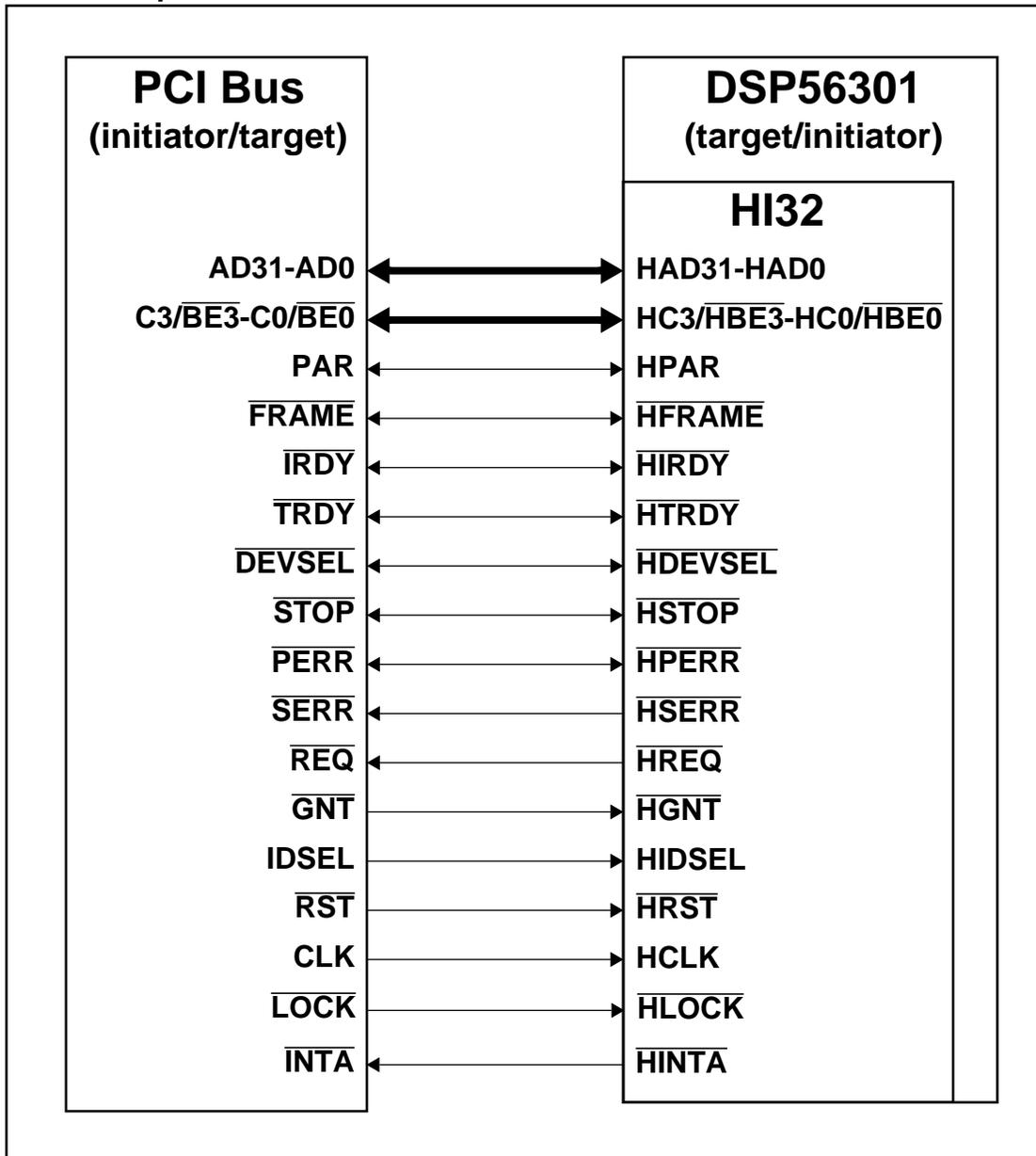
6.9.2 Example of Connection to 16-bit Data Bus (ISA/EISA Bus)

Host (master) ISA		DSP56301 (slave) HI32
AEN	▷	HAEN
$\overline{SBHE}$	▷	HA[10]
SA[0]	▷	HA[9]
SA[9:4]	▷	HA[8:3]
SA[3:1]	▷	HA[2:0]
		$\overline{HDBEN}$
		HDBDR
D[15:0]	BUF	HD[15:0]
CHRDY	◁ *	HTA
$\overline{IOWC}$	▷	$\overline{HWR}$
$\overline{IORC}$	▷	$\overline{HRD}$
$\overline{IO16}$	◁ *	$\overline{HSAK}$
$\overline{IRQ}$	◁ *	$\overline{HIRQ}$
DRQ	◁	HDRQ
$\overline{DAK}$	▷	$\overline{HDAK}$
RESDRV	▷	HRST
	Vcc	$\overline{HBS}$
	Vcc	HP31
	Vcc	HP32
	Vcc	HP19
		HD[23:16]

* Open Collector
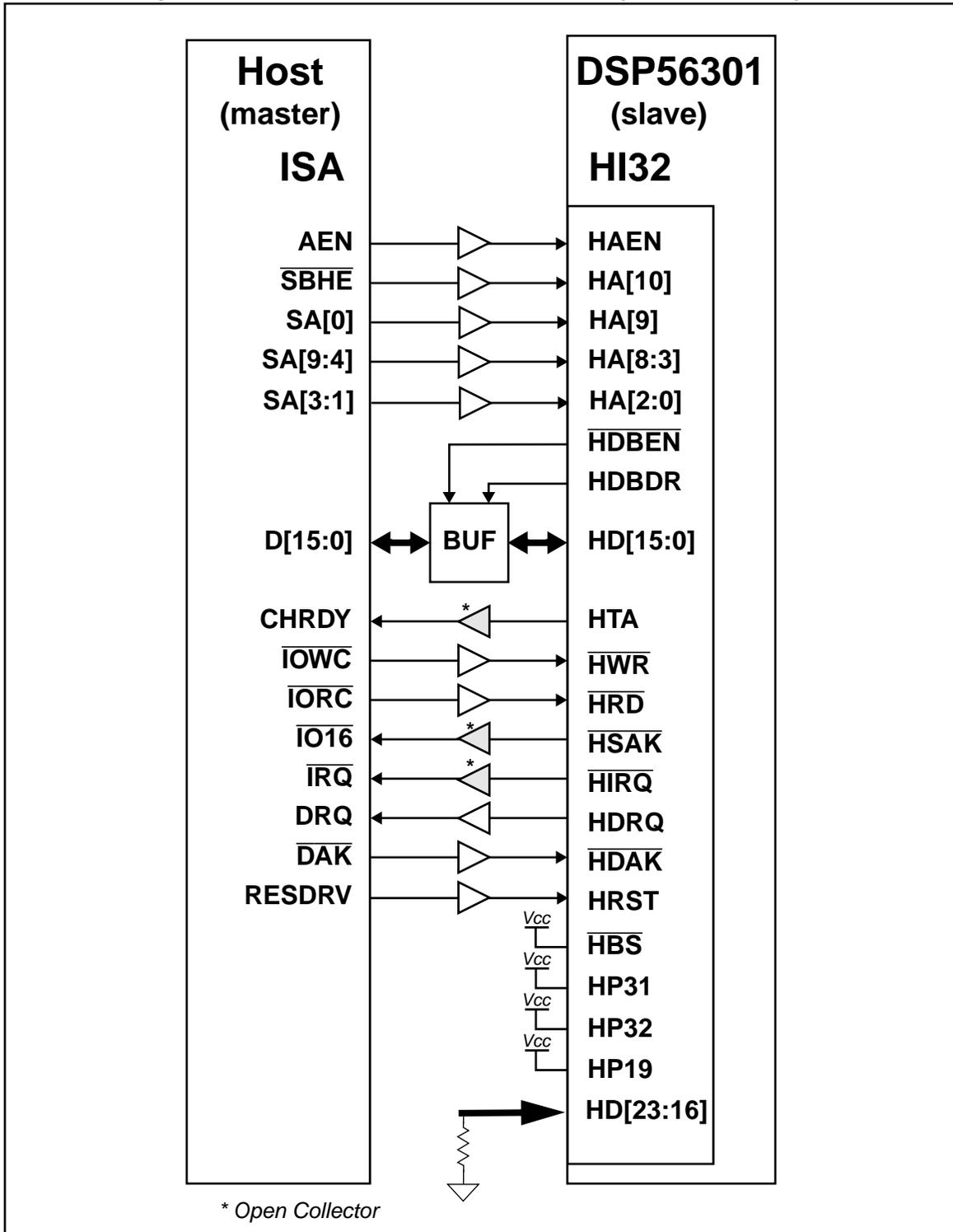
NOTE HI32 may be externally buffered to drive the current required by the

---

ISA/EISA standard. HI32 inputs should be externally buffered if the other ISA/EISA agents are not "3 Volt friendly" as defined in the PCI specifications.

6.9.3 Example of Connection to DSP56300 Core Port A Bus
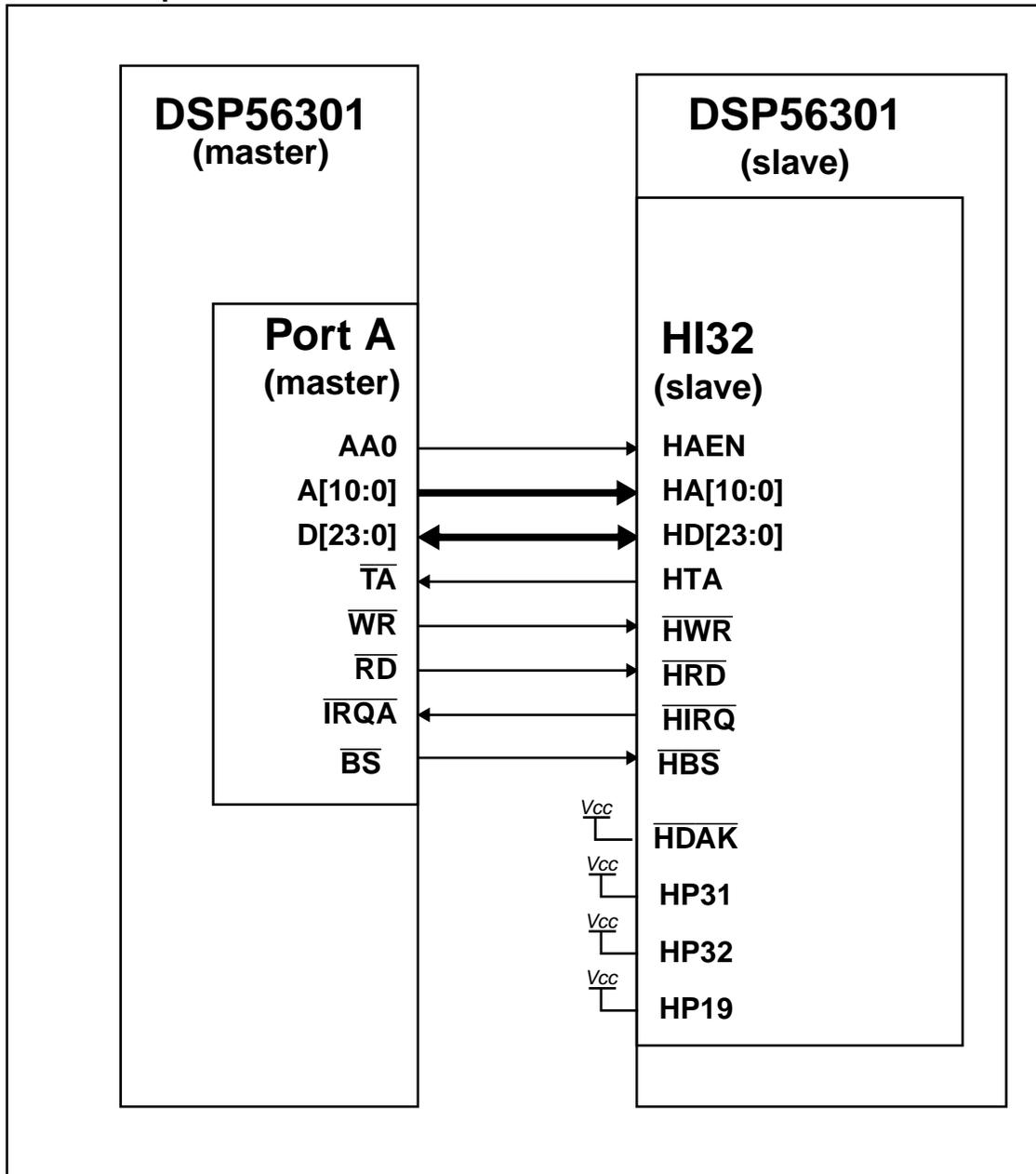
NOTE: If the HI32's DSP and the host DSP use the same EXTAL clock, the HI32 can operate synchronously at its maximum throughput of three clock cycles/word (e.g. for a 66MHz clock the HI32 throughput is 22 Mwords/sec = 66 Mbytes/sec)

7   TRIPLE TIMER MODULE

7.1   INTRODUCTION

This section describes a peripheral module for the DSP56300 core family, composed of a common 21-bit prescaler and three independent and identical general purpose 24-bit timer/event counters, each one having its own register set.

Each timer can use internal or external clocking and can interrupt the DSP56301 after a specified number of events (clocks) or can signal an external device after counting internal events. Each timer can also be used to trigger DMA transfers after a specified number of events (clocks) occurred. Each timer connects to the external world through one bidirectional pin TIO. When TIO is configured as input, the timer functions as an external event counter or can measure external pulse width/signal period. When TIO is used as output the timer is functioning as either a timer, a watchdog or a Pulse Width Modulator. When the TIO pin is not used by the timer it can be used as a General Purpose Input/Output Pin.

7.2   TRIPLE TIMER MODULE - PROGRAMMING MODEL

The registers comprising the Triple Timer Module are shown in Figure 7-1 through Figure 7-4.

Table 7-1. Triple Timer Module Programming Model - Timer 0

TCR0 - Timer Count Register for timer 0
TLR0 - Timer Load Register for timer 0
TCPR0 - Timer Compare Register for timer 0
TCSR0 - Timer Control/Status Register for timer 0

Table 7-2. Triple Timer Module Programming Model - Timer 1

TCR1 - Timer Count Register for timer 1
TLR1 - Timer Load Register for timer 1
TCPR1 - Timer Compare Register for timer 1
TCSR1 - Timer Control/Status Register for timer 1

Table 7-3. Triple Timer Module Programming Model - Timer 2

TCR2 - Timer Count Register for timer 2
TLR2 - Timer Load Register for timer 2
TCPR2 - Timer Compare Register for timer 2
TCSR2 - Timer Control/Status Register for timer 2

Table 7-4. Common Prescaler Registers

TPLR - Timer Prescaler Load Register
TPCR - Timer Prescaler Count Register

7.3   TRIPLE TIMER MODULE ARCHITECTURE

7.3.1   Triple Timer Module Block Diagram

Figure 7-2 shows a block diagram of the triple timer module. It includes a 24-bit Timer Prescaler Load Register (TPLR), a 24-bit Timer Prescaler Count Register (TPCR), a 21-bit counter and three timers. (Each one of the three timers may use the Prescaled Clock as its clock source).

7.3.2   Prescaler Counter

This is a 21-bit counter which is decremented on each rising edge of the prescaler input clock. The counter is enabled when at least one of the three timers is both enabled ($TE_i = 1$) and using the prescaler output as its source ($PCE_i = 1$).

7.3.3   Timer Prescaler Load Register (TPLR)

The Timer Prescaler Load Register is a 24-bit read/write register that controls the prescaler divide factor and the source for the prescaler input clock. The control bits are described in the following paragraphs (see Figure 7-1).

Figure 7-1. Timer Prescaler Load Register
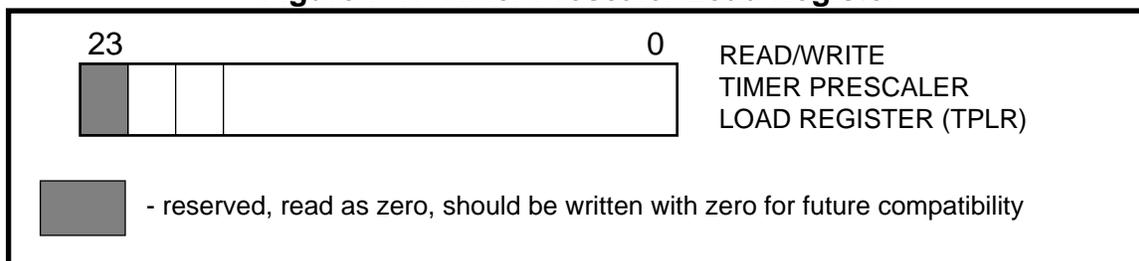
7.3.3.1 Prescaler Preload Value (PL0-PL20) Bits 0-20

These 21 bits contain the prescaler preload value. This preload value is loaded into the prescaler counter whenever either the counter reaches the value of zero or the counter switches state from disabled to enabled.

If PL0-PL20 = N then the prescaler will count N+1 source clock cycles before generating a prescaled clock pulse. Therefore the prescaler divide factor is preload value + 1.

PL0-PL20 bits are cleared by hardware $\overline{\text{RESET}}$ and software RESET (RESET instruction).

Figure 7-2. Triple Timer Module Block Diagram
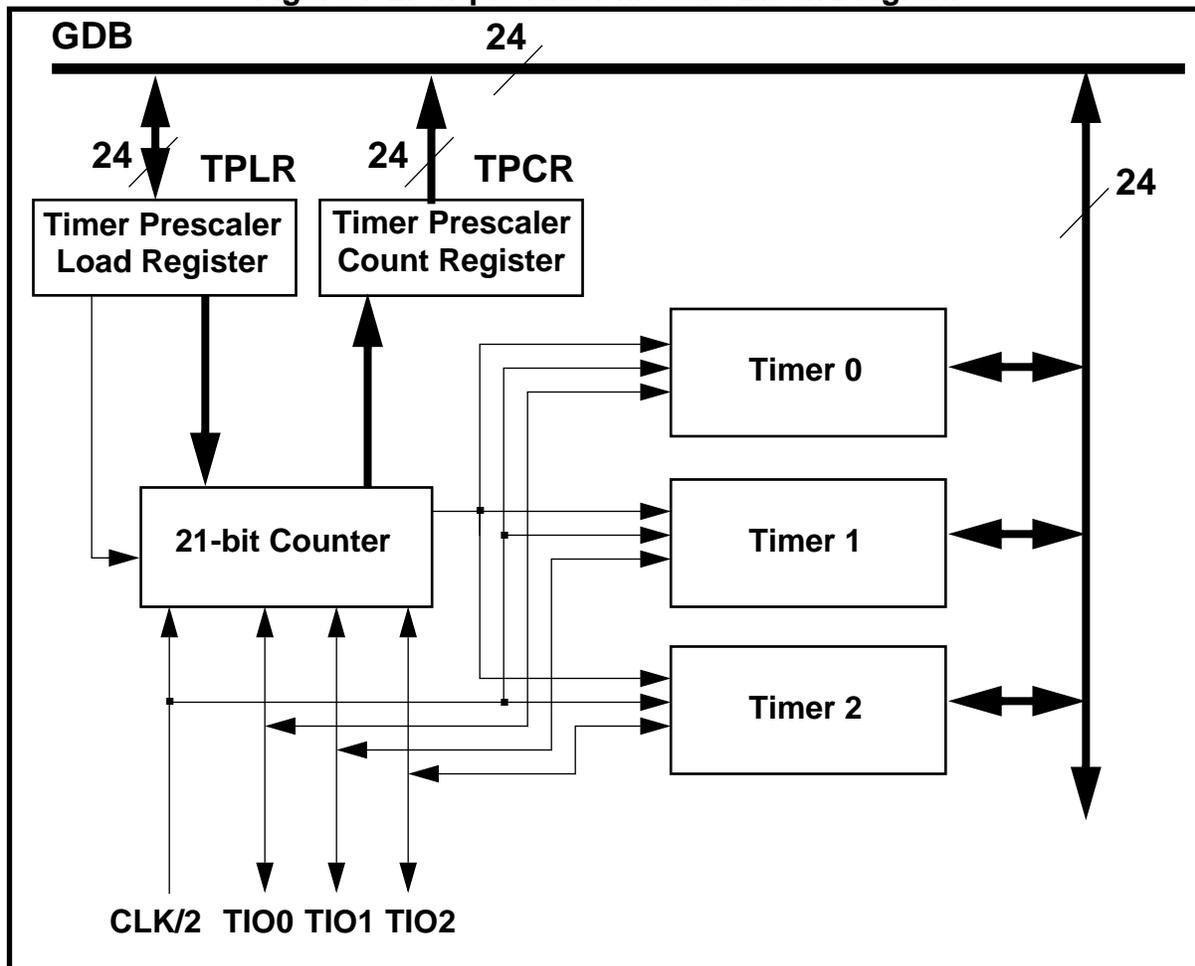
7.3.3.2 Prescaler Source (PS0-PS1) Bits 21-22

The two Prescaler Source bits control the source of the prescaler clock. Table 7-5 summarizes the functionality of the PS bits. The internal clock CLK/2 (the DSP56301 clock divided by two) is selected when bits PS0-PS1 are cleared. The other combinations select one of the TIO pins as the source clock for the prescaler, regardless of the operating mode of the appropriate timer.

PS0-PS1 bits are cleared by hardware $\overline{\text{RESET}}$ and software RESET (RESET instruction).

NOTE1: If the prescaler source clock is external, the prescaler counter will be incremented by the transitions on the TIO pin. The external clock is internally synchronized to the internal clock and its frequency should be lower than the internal operating frequency divided by 4 (CLK/4).

NOTE2: The PS1-PS0 bits should be changed only when the prescaler counter is disabled to ensure proper functionality.

Table 7-5. Prescaler Source Selection

PS1	PS0	PRESCALER CLOCK SOURCE
0	0	Internal CLK/2
0	1	TIO0
1	0	TIO1
1	1	TIO2

7.3.3.3   TPLR Reserved bit (Bit 23)

This reserved bit is read as zero and should be written with zero for future compatibility.

7.3.4   Timer Prescaler Count Register (TPCR)

The Timer Prescaler Count Register is a 24-bit read only register that reflects the current value in the prescaler counter. The register bits are described in the following paragraphs (see Figure 7-3).

Figure 7-3. Timer Prescaler Count Register
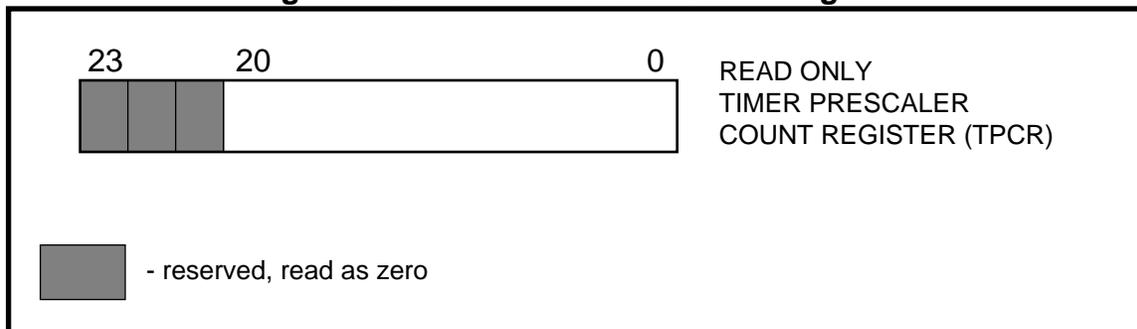
7.3.4.1   Prescaler Counter Value (PC0-PC20) Bits 0-20

These 21 bits contain the current value in the prescaler counter.

7.3.4.2   TPCR Reserved bits (Bits 21,22,23)

These reserved bit are read as zero.

7.4 TIMER ARCHITECTURE

7.4.1 Timer Block Diagram

Figure 7-4 shows a block diagram of a timer. It includes a 24-bit counter, a 24-bit read-write Timer Control and Status Register (TCSR), a 24-bit read only Timer Count Register (TCR), a 24-bit write only Timer Load Register (TLR), a 24-bit read-write Timer Compare Register (TCPR), and logic for clock selection and interrupt/DMA trigger generation. The DSP56301 views each timer as a memory mapped peripheral occupying four 24-bit words in the X data memory space. The user may use standard polled or interrupt programming techniques. The programming model is shown in Figure 7-5

7.4.2 Timer Count Register (TCR)

The Count Register is a 24-bit read-only Register (TCR). In Timer and Watchdog Modes the counter contents can be read at any time by reading the TCR register. In Measurement Modes the TCR will be loaded with the current value of the counter on the appropriate edge of the input signal and its value can be read to determine the width, period or delay of leading edge of the input signal (incoming on the TIO pin).

7.4.3 Timer Load Register (TLR)

The Load Register is a 24-bit write-only Register (TLR). In all modes the counter is preloaded with the TLR value after the Timer Enable bit is set (TE=1) and a first event occurs.
In Timer Modes, if timer reload mode is set (TRM=1), the counter is reloaded each time after it has reached the value contained by the Timer Compare Register and the new event occurs.
In Measurement Modes, if timer reload mode is set (TRM=1), the counter is reloaded with the TLR value on each appropriate edge of the input signal, after the Timer Enable bit is set (TE=1).
In PWM Modes, if timer reload mode is set (TRM=1), the counter is reloaded each time after it has overflowed and the new event occurs.
In Watchdog Modes, if timer reload mode is set (TRM=1), the counter is reloaded each time after it has reached the value contained by the Timer Compare Register and the new event occurs. In this mode, the counter is also reloaded whenever the TLR is written with a new value while the Timer Enable bit is set (TE = 1).
In all modes, if TRM is cleared (TRM=0), the counter operates as free running counter.

7.4.4 Timer Compare Register (TCPR)

The compare register is a 24-bit read-write register which contains the value to be compared to the counter value. The counter value is compared against the value contained by the TCPR on every timer clock after the Timer Enable bit is set (TE=1). When the compare matches, the TCF bit is set and, if interrupts are enabled (TCIE=1), an interrupt is generated. In Measurement Modes the TCPR is ignored.

7.4.5   Timer Control/Status Register (TCSR)

The control/status register is a 24-bit read/write register that controls the timer and reflects its status. The control and status bits are described in the following paragraphs (see Figure 7-5).

Figure 7-4. 24 bit Timer Module Block Diagram
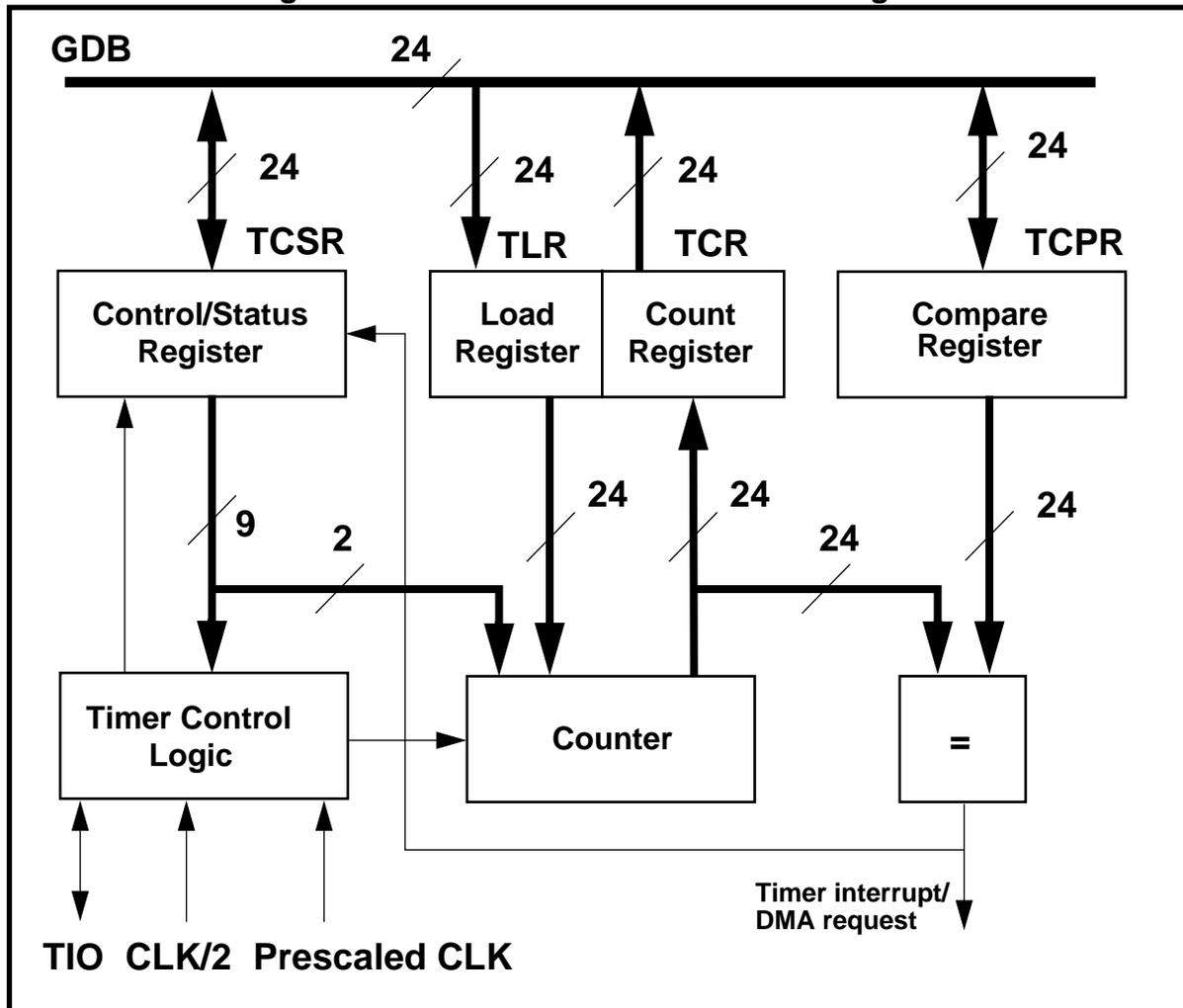
7.4.5.1   Timer Enable (TE) Bit 0

The Timer Enable is used to enable or disable the timer. Setting the TE bit (TE=1) will enable the timer and clear the timer counter. The counter will start counting according to the mode defined by TC3-TC0. Clearing the TE bit will disable the timer.
TE is cleared by hardware $\overline{\text{RESET}}$ and software RESET (RESET instruction).

NOTE: When all the three timers are disabled and not in GPIO mode, all three
TIO pins becomes tri-state. In order to prevent undesired spikes on the
TIO pins (when switching from tri-state into active state) it is recommend-
ed to use external pull-ups or pull downs resistors tied to TIO pins.

7.4.5.2 Timer Overflow Interrupt Enable (TOIE) Bit 1

The Timer Overflow Interrupt Enable bit is used to enable the timer overflow interrupts. The overflow interrupt will be generated after the counter wraparound occurred, i.e. the counter value has changed from $FFFFFF to $000000 when a new event occurred. Setting TOIE (TOIE=1) will enable the overflow interrupts. When the bit is cleared (TOIE=0) the overflow interrupts are disabled.
TOIE is cleared by hardware $\overline{RESET}$ and software RESET (RESET instruction).

Figure 7-5. Timer Module Programming Model

7	6	5	4	3	2	1	0	
TC3	TC2	TC1	TC0		TCIE	TOIE	TE	

15	14	13	12	11	10	9	8	READ/WRITE
PCE		DO	DI	DIR		TRM	INV	TIMER CONTROL/STATUS REGISTER (TCSR)

23	22	21	20	19	18	17	16	
		TCF	TOF					

23	0	WRITE ONLY TIMER LOAD REGISTER (TLR)

23	0	READ ONLY TIMER COUNT REGISTER (TCR)

23	0	READ/WRITE TIMER COMPARE REGISTER (TCPR)

▮ - reserved, read as zero, should be written with zero for future compatibility
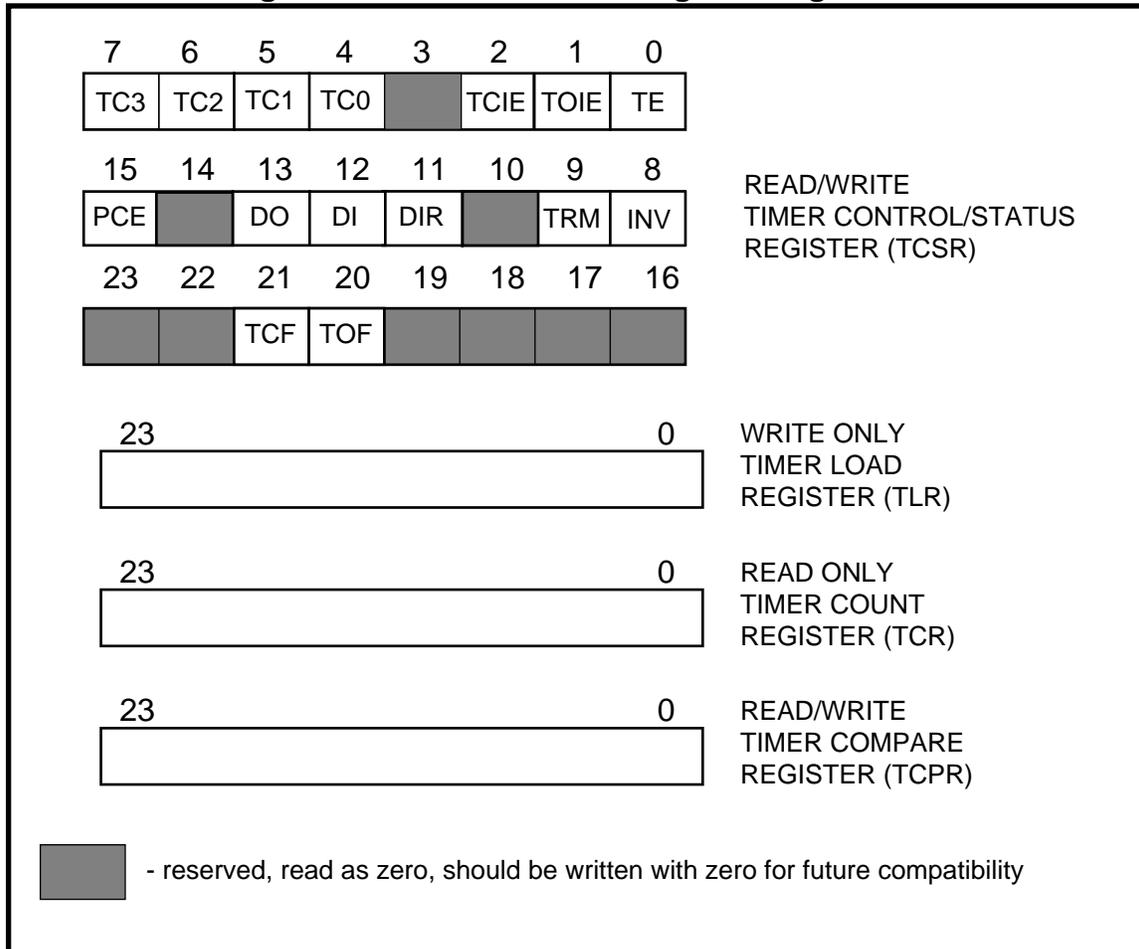
7.4.5.3 Timer Compare Interrupt Enable (TCIE) Bit 2

The Timer Compare Interrupt Enable bit is used to enable the timer compare interrupts. The compare interrupt will be generated after the counter matches the compare register (in the Timer, PWM or Watchdog Modes). If TCPR is loaded with N, an interrupt will occur after (N-M+1) events, where M is TLR value. Setting TCIE (TCIE=1) will enable the compare interrupts. When the bit is cleared (TCIE=0) the compare interrupts are disabled. TCIE is cleared by hardware $\overline{RESET}$ and software RESET (RESET instruction).

7.4.5.4 Timer Control (TC0-TC3) Bits 4-7

The four Timer Control bits control the source of the timer clock, the behavior of the TIO pin and the timer mode of operation. Table 7-6 summarizes the functionality of the TC bits. A detailed description of the timer operating modes is given in Section 7.5.
TC0-TC3 bits are cleared by hardware $\overline{\text{RESET}}$ and software RESET (RESET instruction).

NOTE1:   If the clock is external, the counter will be incremented by the transitions on the TIO pin. The external clock is internally synchronized to the internal clock and its frequency should be lower than the internal operating frequency divided by 4 (CLK/4).

NOTE2:   The TC3-TC0 bits should be changed only when TE=0 (i.e. timer disabled) to ensure proper functionality.

Table 7-6. Timer Control Bits

TC3	TC2	TC1	TC0	TIO	CLOCK	MODE
0	0	0	0	GPIO *	Internal	Timer
0	0	0	1	Output	Internal	Timer Pulse
0	0	1	0	Output	Internal	Timer Toggle
0	0	1	1	Input	External	Event Counter
0	1	0	0	Input	Internal	Input Width
0	1	0	1	Input	Internal	Input Period
0	1	1	0	Input	Internal	Capture
0	1	1	1	Output	Internal	Pulse Width Modulation (PWM)
1	0	0	0	-	-	Reserved
1	0	0	1	Output	Internal	Watchdog Pulse
1	0	1	0	Output	Internal	Watchdog Toggle
1	0	1	1	-	-	Reserved
1	1	0	0	-	-	Reserved
1	1	0	1	-	-	Reserved
1	1	1	0	-	-	Reserved
1	1	1	1	-	-	Reserved

- * - the GPIO function is enabled only if TC3-TC0 are all 0 (zero)

7.4.5.5 Inverter (INV) Bit 8

The Inverter bit affects the polarity of the external incoming signal on the TIO pin (when TIO is programmed as input) and affects the polarity of the pulse generated on the TIO pin (when TIO is programmed as output). In the Timer Modes if TIO is programmed as input and INV=0, the 0 to 1 transitions on the TIO input pin will increment the counter; if INV=1, the 1 to 0 transitions on the TIO input pin will increment the counter. In the Input Width Mode the INV bit determines whether the high pulse or the low pulse is measured and in the Input Period Mode the INV bit determines whether the period is measured between rising or falling edges. If TIO is programmed as output and INV=0, the pulse generated by the timer will be of positive polarity; if INV=1, the pulse generated by the timer will be inverted.

INV is cleared by hardware $\overline{RESET}$ and software RESET (RESET instruction).

NOTE1: The INV bit affects both the timer and the GPIO modes of operation.

NOTE2: The INV bit should be changed only when TE=0 (i.e. timer disabled), or in GPIO mode of operation, to ensure proper functionality.

NOTE3: The polarity of the prescaler source clock, when the TIO is used as input to the prescaler, is not affected by the corresponding INV bit.

7.4.5.6 Timer reload mode (TRM) Bit 9

The timer reload mode bit controls the counter preload operation.

In Timer and Watchdog Modes the counter is preloaded with the TLR value after the Timer Enable bit is set (TE=1) and a first event occurs. If timer reload mode is set (TRM=1), the counter is reloaded each time after it has reached the value contained by the Timer Compare Register and the new event occurs. In PWM Mode the counter is reloaded each time counter wraparound has occurred (overflow) and the new event occurs. In Measurement Modes the counter is preloaded with the TLR value (if TRM=1) on each appropriate edge of the input signal, after the Timer Enable bit is set (TE=1). If TRM is cleared (TRM=0), the counter operates as free running counter, incrementing on each incoming event.

TRM is cleared by hardware $\overline{RESET}$ and software RESET (RESET instruction).

7.4.5.7 Direction (DIR) Bit 11

The direction bit determines the behavior of the TIO pin when used as general purpose Input/Output pin. When DIR=0 the TIO pin is an input and when DIR=1 the TIO pin is an output. The TIO pin can be used as a general purpose Input/Output (GPIO) pin only when TC3-TC0 are all zero. If one of TC3, TC2, TC1 or TC0 is not 0 the GPIO function is disabled and the DIR pin has no effect.

DIR is cleared by hardware $\overline{RESET}$ and software RESET (RESET instruction).

7.4.5.8   Data Input (DI) Bit 12

The DI bit reflects the value of TIO pin according to the INV bit. Reading the DI bit will read the TIO pin if INV=0, or the inverted TIO pin if INV=1.

7.4.5.9   Data Output (DO) Bit 13

When the GPIO mode is enabled (TC3-TC0 are all zero) and DIR=1 the TIO pin acts as data output. Writing the DO bit will write the data to the TIO pin. If the INV bit is set the data on the TIO pin will be inverted. When GPIO mode is disabled writing the DO bit will have no effect.
DO is cleared by hardware $\overline{\text{RESET}}$ and software RESET (RESET instruction).

7.4.5.10   Prescaled Clock Enable (PCE) Bit 15

The prescaled clock enable bit is used to select the prescaled clock as the timer source clock. When PCE is cleared the timer uses either internal (CLK/2) or external (TIO) source clock as determined by the timer operating mode. When PCE is set the prescaler output is used as the timer source clock for the counter regardless of the timer operating mode. PCE is cleared by hardware $\overline{\text{RESET}}$ and software RESET (RESET instruction).

NOTE1:   The PCE bit should be changed only when TE=0 (i.e. timer disabled) to ensure proper functionality.

NOTE2:   The source clock for the prescaler is determined only by the Prescaler Source bits (PS0-PS1) of the Timer Prescaler Load Register (TPLR). Therefore a timer may be clocked by prescaled clock derived from the TIO of another timer.

7.4.5.11   Timer Overflow Flag (TOF) Bit 20

The Timer Overflow Flag bit when set indicates that counter wraparound has occurred. The Timer Overflow Flag bit is cleared when writing a one into the TOF bit. Writing a zero into the TOF bit has no effect. The bit is cleared also when the timer overflow interrupt is serviced (timer overflow interrupt acknowledge).
TOF is cleared by hardware $\overline{\text{RESET}}$ and software RESET (RESET instruction).

7.4.5.12   Timer Compare Flag (TCF) Bit 21

In the Timer, PWM and Watchdog Modes the Timer Compare Flag bit when set indicates that (N-M+1) events are counted, where N is the value in the compare register and M is TLR value. In the Measurement Modes the TCF bit when set indicates that the measurement has been completed.

The Timer Compare Flag bit is cleared when writing a one into the TCF bit. Writing a zero into the TCF bit has no effect. The bit is cleared also when the timer compare interrupt is serviced (timer compare interrupt acknowledge).
TCF is cleared by hardware $\overline{\text{RESET}}$ and software RESET (RESET instruction).

NOTE: Writing a zero in the TOF or TCF bit can be done with the Bit Test and Clear (BCLR) instruction. The state of the tested bit will be stored in the carry bit of the status register (SR).

7.4.5.13   TCSR Reserved bits (Bits 3, 10, 14, 16-19, 22, 23)

These reserved bits are read as zero and should be written with zero for future compatibility.

7.5   TIMER FUNCTIONAL DESCRIPTION (MODES OF OPERATION)

This section describes the various modes of operation of the timer module.

7.5.1   Timer Modes

7.5.1.1   Timer mode, internal clock, no output (mode 0)

This mode is defined by TC3-TC0 equal 0000. In this mode the counter is cleared after TE is set and loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. When the counter matches the value contained by the TCPR, the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated. At the next timer clock the counter is loaded with TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented on each timer clock. If counter wraparound has occurred the TOF bit is set and, if the TOIE is set, an overflow interrupt is generated. This process is repeated until the timer is disabled (TE=0). The counter contents can be read at any time by reading the TCR register.

7.5.1.2   Timer mode, internal clock, output pulse enable (mode 1)

This mode is defined by TC3-TC0 equal 0001. In this mode the counter is cleared after TE is set and loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. When the counter matches the value contained by the TCPR, the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated. At the next timer clock the counter is loaded with TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented on each timer clock. This process is repeated until the timer is disabled (TE=0). Each time the counter matches the TCPR value a pulse will be output on the TIO pin with the width equal to timer clock period. The pulse polarity is determined by the INV bit. If counter wraparound has occurred the TOF bit is set and, if the TOIE is set, an overflow interrupt is generated. The counter contents can be read at any time by reading the TCR register.

NOTE: After TE=1 the TIO pin output value is set equal to INV bit, to guarantee the correct first pin transition.

7.5.1.3   Timer mode, internal clock, output toggle enable (mode 2)

This mode is defined by TC3-TC0 equal 0010. In this mode the counter is cleared after TE is set and loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. When the counter matches the value of the TCPR, the TIO output pin will be toggled, the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated. At the next timer clock the counter is loaded with TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented on each timer clock. This process is repeated until the timer is disabled (TE=0). The TIO polarity is determined by the INV bit. On the first match the TIO output will be set if INV=0 or cleared if INV=1. If counter wraparound has occurred the TOF bit is set and, if the TOIE is set, an overflow interrupt is generated. The counter contents can be read at any time by reading the TCR register.

NOTE: After TE=1 the TIO pin output value is set equal to INV bit, to guarantee the correct first pin transition.

7.5.1.4   Timer mode, external clock, event counter (mode 3)

This mode is defined by TC3-TC0 equal 0011. In this mode the counter is cleared after TE is set and loaded with the TLR value on the first transitions on the source clock which can be either the TIO input pin or the prescaled clock input. The following transitions will increment the counter. When the counter matches the value contained by TCPR, the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated. At the next transitions the counter is loaded with TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented with each transitions on the source clock. This process is repeated until the timer is disabled (TE=0). The INV bit determines whether 0 to 1 transitions (INV=0) or 1 to 0 transitions (INV=1) will increment the counter. If counter wraparound has occurred the TOF bit is set and, if the TOIE is set, an overflow interrupt is generated. The counter contents can be read at any time by reading the TCR register. The external clock is internally synchronized to the internal clock and its frequency should be lower than the internal operating frequency divided by 4 (CLK/4).

7.5.2   Measurement Modes

7.5.2.1   Pulse width measurement mode (mode 4)

This mode is defined by TC3-TC0 equal 0100. In this mode the counter is cleared after TE is set and, after the first appropriate transition (as determined by the INV bit) occurring on TIO input pin, it is loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. When the first edge of opposite polarity occurs on TIO the counter stops, the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated. The contents of the counter is loaded into the TCR and the user's program can read its value that represents the widths of the TIO pulse. On the first timer clock, following the next transition that occurs on TIO input pin, the counter is loaded with

TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented on each timer clock, accumulating measurements results. This process is repeated until the timer is disabled (TE=0). If counter wraparound has occurred the TOF bit is set and, if the TOIE is set, an overflow interrupt is generated. In this mode TIO acts as a gating signal for the internal timer clock. The INV bit determines whether the counting is enabled when TIO is high (INV=0) or when TIO is low (INV=1).

7.5.2.2   Period measurement mode (mode 5)

This mode is defined by TC3-TC0 equal 0101. In this mode the counter is cleared after TE is set and, after the first appropriate transition (as determined by the INV bit) occurring on TIO input pin, it is loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. On each following transition of the same polarity that occurs on TIO the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated. The contents of the counter is loaded in the TCR and the user's program can read its value and the user's program can read the TCR to determine the distance between TIO edges. On the next timer clock the counter is loaded with TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented on each timer clock, accumulating measurements results. This process is repeated until the timer is disabled (TE=0). If counter wraparound has occurred the TOF bit is set and, if the TOIE is set, an overflow interrupt is generated. The INV bit determines whether the period is measured between consecutive 0 to 1 transitions of TIO (INV=0) or between consecutive 1 to 0 transitions of TIO (INV=1).

7.5.2.3   Capture mode (mode 6)

This mode is defined by TC3-TC0 equal 0110. In this mode the counter is cleared after TE is set and loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. If counter wraparound has occurred the TOF bit is set and, if the TOIE is set, an overflow interrupt is generated. At the first transition of external clock the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated. The contents of the counter is loaded into the TCR and the user's program can read its value that represents the delay of the leading detected edge in relation to the setting of the TE bit. The counting is stopped. The INV bit determines whether the period is measured between the setting of TE bit and the transitions of TIO from 0 to 1 (INV=0) or from 1 to 0 (INV=1).

7.5.2.4   Measurement modes exactness

Since the measurement modes use the internal clock to increment the counter, but use the external signal for gating the count, synchronization is needed. The synchronization process may effect the measurement exactness up to a single selected (internal or prescaled) clock cycle.

7.5.3   PWM Modes

7.5.3.1   Pulse Width Modulation mode, internal clock, output toggle enable (mode 7)

This mode is defined by TC3-TC0 equal 0111. In this mode the counter is cleared after TE is set and loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. When the counter matches the value of the TCPR, the TIO output pin will be toggled, the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated and the count is continued. When counter wraparound has occurred the TIO output pin will be toggled, the TOF bit in TCSR is set and, if the TOIE is set, an overflow interrupt is generated. At the next timer clock the counter is loaded with TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented on each timer clock. This process is repeated until the timer is disabled (TE=0). The TIO polarity is determined by the INV bit. On the first transaction the TIO output will be set if INV=0 or cleared if INV=1. The counter contents can be read at any time by reading the TCR register.
The value in the TLR determines the output period ($FFFFFF - TLR). The value in the TCPR determines the duty cycle of the output signal (TLR - TCPR vs. $FFFFFF - TCPR). Therefore, the ensure correct functionality, the values in TLR and TCPR should not be the same.

NOTE: After TE=1 the TIO pin output value is set equal to INV bit, to guarantee the correct first pin transition.

7.5.4   Watchdog Modes

7.5.4.1   Watchdog mode, internal clock, output pulse enable (mode 9)

This mode is defined by TC3-TC0 equal 1001. In this mode the counter is cleared after TE is set and loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. When the counter matches the value of the TCPR, the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated and the count is continued. At the next timer clock the counter is loaded with TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented on each timer clock. This process is repeated until the timer is disabled (TE=0). The counter will be reloaded whenever the TLR is written with a new value while the Timer Enable bit is set (TE = 1). When counter wraparound has occurred the TOF bit in TCSR is set and, if the TOIE is set, an overflow interrupt is generated. At the same time a pulse will be output on the TIO pin with the width equal to timer clock period. The pulse polarity is determined by the INV bit. The counter contents can be read at any time by reading the TCR register.

NOTE: In this mode, the internal hardware will preserve the TIO value and direction for additional 2.5 internal clock cycles after reset was activated. This will ensure a valid length reset when the TIO is used as input to the RESET pin.

7.5.4.2   Watchdog mode, internal clock, output toggle enable (mode 10)

This mode is defined by TC3-TC0 equal 1010. In this mode the counter is cleared after TE is set and loaded with the TLR value on the first timer clock derived either from the DSP56301 clock divided by two (CLK/2) or from the prescaled clock input. The following timer clocks will increment the counter. When the counter matches the value of the TCPR, the TCF bit in TCSR is set and, if the TCIE is set, a compare interrupt is generated and the count is continued. At the next timer clock the counter is loaded with TLR value (if TRM is set) and the count is resumed. If TRM is cleared, the counter continues to be incremented on each timer clock. This process is repeated until the timer is disabled (TE=0). The counter will be reloaded whenever the TLR is written with a new value while the Timer Enable bit is set (TE = 1). When counter wraparound has occurred the TIO output pin will be toggled, the TOF bit in TCSR is set and, if the TOIE is set, an overflow interrupt is generated. The TIO polarity is determined by the INV bit. On the first transaction the TIO output will be set if INV=0 or cleared if INV=1. The counter contents can be read at any time by reading the TCR register.

NOTE1:   After TE=1 the TIO pin output value is set equal to INV bit, to guarantee the correct first pin transition.

NOTE2:   In this mode, the internal hardware will preserve the TIO value and direction for additional 2.5 internal clock cycles after reset was activated. This will ensure a valid length reset when the TIO is used as input to the RESET pin.

7.5.5   Reserved Modes

Modes 8,11,12,13,14 and 15 are reserved.

7.5.6   Special Cases

7.5.6.1   Timer behavior during WAIT and STOP

During the execution of the WAIT instruction the timer clocks are active thus the timer activity will continue undisturbed. When reaching the final event, if the timer interrupt is enabled, an interrupt will be generated, the processor will leave the WAIT state and service the interrupt.

During the execution of the STOP instruction the timer clocks are disabled, the timer activity is stopped and the TIO pins are disconnected. If, for example, the TIO pin is used as input the changes that occur while in STOP will be ignored. In order to ensure correct behavior the timer should be disabled before executing the STOP instruction.

7.5.7   DMA trigger

Each timer can also be used to trigger DMA transfers. For that a DMA channel has to be programmed to be triggered by the specific timer. The timer will issue a DMA trigger on every compare event, in all modes of operation. It is the user's responsibility to guarantee

that a new trigger will not be generated before the DMA channel has completed the transfers associated with the previous trigger, otherwise the trigger will not be captured by the DMA channel.

8   ENHANCED SYNCHRONOUS SERIAL INTERFACE

(ESSI)

8.1   INTRODUCTION

There are two independent and identical Synchronous Serial Interfaces in the DSP56301: ESSI0 and ESSI1. For simplicity a single generic ESSI will be described.

The enhanced synchronous serial interface (ESSI) provides a full-duplex serial port for serial communication with a variety of serial devices including one or more industry-standard codecs, other DSPs, microprocessors, and peripherals which implement the Motorola SPI. The ESSI consists of independent transmitter and receiver sections and a common ESSI clock generator.

ESSI Block Diagram is shown in Figure 8-1. This interface is named synchronous because all serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is similar in that it is also intended for periodic transfers; however, it will support up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data. This mode can be used to transfer data serially at high speed when the data becomes available. This mode offers a subset of the SPI protocol.

8.2   ESSI DATA AND CONTROL PINS

Three to six pins are required for operation, depending on the operating mode selected. STD, SC0, SC1 pins are fully synchronized if they are programmed as transmit data pins.

8.2.1   Serial Transmit Data Pin (STD)

STD is used for transmitting data from TX0 serial transmit shift register. STD is an output when data is being transmitted from TX0 shift register. With an internally generated bit clock, the STD pin becomes high impedance after the last data bit has been transmitted for a full clock period, assuming another data word does not follow immediately. If a data word follows immediately, there will not be a high-impedance interval.
STD may be programmed as a general-purpose pin (P5) when the ESSI STD function is not being used.

8.2.2 Serial Receive Data Pin (SRD)

SRD receives serial data and transfers the data to the ESSI receive shift register. SRD may be programmed as a general-purpose I/O pin (P4) when the ESSI SRD function is not being used.

Figure 8-1. ESSI Block Diagram
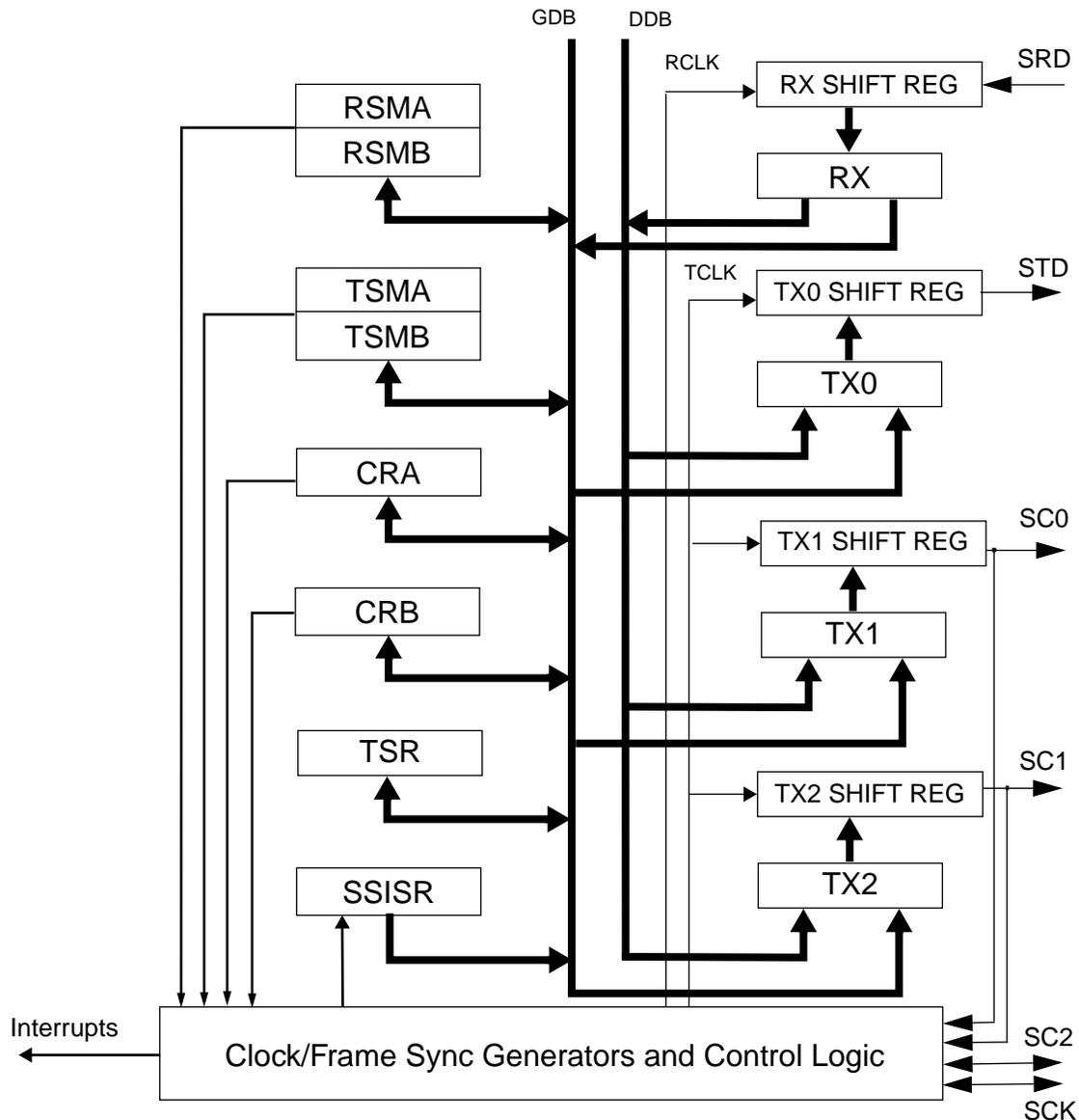
8.2.3 Serial Clock (SCK)

SCK is a bidirectional pin providing the serial bit rate clock for the ESSI interface. The SCK is a clock input or output used by all the enabled transmitters and receiver in synchronous modes or by all the enabled transmitters in asynchronous modes (see Table 8-1 on page 4).

SCK may be programmed as a general-purpose I/O pin (P3) when the ESSI SCK function is not being used.

NOTE 1   The ESSI can operate with more than one active transmitter only in synchronous mode.

NOTE 2   Although an external serial clock can be independent of and asynchronous to the DSP system clock, it must exceed the minimum clock cycle time of three Clkout cycles time. (i.e., the DSP56301 clock frequency must be at least three times the external ESSI clock frequency and each ESSI phase must exceed the minimum of 1.5 Clkout cycles)

8.2.4   Serial Control Pin (SC0)

The function of this pin is determined by the selection of either synchronous or asynchronous mode (see Table 8-3 on page 13). For asynchronous mode, this pin will be used for the receive clock I/O. For synchronous mode, this pin is used for transmitter data out pin of transmit shift register number 1 or for serial flag I/O. A typical application of flag I/O would be multiple device selection for addressing in codec systems. If this pin is configured as serial flag pin its direction is determined by the SCD0 bit in the CRB. When configured as an output, this pin will be either serial output flag 0, based on control bit OF0 in CRB, or a receive shift register clock output. When configured as an input, this pin may be used either as serial input flag 0, which will control status bit IF0 in the SSISR, or as a receive shift register clock input.
When this pin is configured as a transmit data pin its direction is always output regardless of SCD0 bit value, and is fully synchronized with the other transmit data pins (STD and SC1).
SC0 may be programmed as a general-purpose I/O pin (P0) when the ESSI SC0 function is not being used.

8.2.5   Serial Control Pin (SC1)

The function of this pin is determined by the selection of either synchronous or asynchronous mode (see Table 8-3 on page 13). In asynchronous mode (such as a single codec with asynchronous transmit and receive), this pin is the receiver frame sync I/O. For synchronous mode, this pin is used for transmitter data out pin of transmit shift register number 2 or for drive enable transmitter#0 signal or for serial flag SC1 and operates like the previously described SC0. SC0 and SC1 are independent serial I/O flags but may be used together for multiple serial device selection. SC0 and SC1 can be used unencoded to select up to two codecs or may be decoded externally to select up to four codecs. If this pin is configured as serial flag pin its direction is determined by the SCD1 bit in the CRB. When configured as an output, this pin will be either a serial output flag, based on control bit OF1, the transmitter#0 drive enable signal or it will make the receive frame sync signal available. When configured as an input, this pin may be used as a serial input flag, which will control status bit IF1 in the ESSI status register, or as a receive frame sync from an external source.
When this pin is configured as a transmit data pin its direction is always output regardless

of SCD1 bit value, and is fully synchronized with the other transmit data pins (STD and SC0).

SC1 may be programmed as a general-purpose I/O pin (P1) when the ESSI SC1 function is not being used.

Table 8-1. ESSI Clock Sources

SYN	SCKD	SCD0	R Clock Source	RX Clock Out	T Clock Source	TX Clock Out
Asynchronous						
0	0	0	EXT, SC0	–	EXT, SCK	–
0	0	1	INT	SC0	EXT, SCK	–
0	1	0	EXT, SC0	–	INT	SCK
0	1	1	INT	SC0	INT	SCK
Synchronous						
1	0	0/1	EXT, SCK	–	EXT, SCK	–
1	1	0/1	INT	SCK	INT	SCK

8.2.6   Serial Control Pin (SC2)

This pin is used for frame sync I/O. SC2 is the frame sync for both the transmitter and receiver in synchronous mode and for the transmitter only in asynchronous mode (see Table 8-3 on page 13). The direction of this pin is determined by the SCD2 bit in CRB. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitter (and the receiver in synchronous operation).

SC2 may be programmed as a general-purpose I/O pin (P2) when the ESSI SC2 function is not being used.

8.3   ESSI PROGRAMMING MODEL

The ESSI can be viewed as two control registers, one status register, three transmit data registers, a receive data register, two transmit slot mask registers, two receive slot mask registers and special-purpose time slot register. The following paragraphs give detailed

descriptions and operations of each of the bits in the ESSI registers.
The ESSI contains also the GPIO functionality, described at Chapter 8.5.

Figure 8-2. ESSI Control Register A (CRA)

11	10	9	8	7	6	5	4	3	2	1	0
PSR				PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0

23	22	21	20	19	18	17	16	15	14	13	12
	SSC1	WL2	WL1	WL0	ALC		DC4	DC3	DC2	DC1	DC0

Figure 8-3. ESSI Control Register B (CRB)

11	10	9	8	7	6	5	4	3	2	1	0
CKP	FSP	FSR	FSL1	FSL0	SHFD	SCKD	SCD2	SCD1	SCD0	OF1	OF0

23	22	21	20	19	18	17	16	15	14	13	12
REIE	TEIE	RLIE	TLIE	RIE	TIE	RE	TE0	TE1	TE2	MOD	SYN

Figure 8-4. ESSI Status Register (SSISR)

7	6	5	4	3	2	1	0
RDF	TDE	ROE	TUE	RFS	TFS	IF1	IF0

Figure 8-5. ESSI Transmit Slot Mask Register A (TSMA)

11	10	9	8	7	6	5	4	3	2	1	0
TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0

23	22	21	20	19	18	17	16	15	14	13	12
								TS15	TS14	TS13	TS12

Figure 8-6. ESSI Transmit Slot Mask Register B (TSMB)

11	10	9	8	7	6	5	4	3	2	1	0
TS27	TS26	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16

23	22	21	20	19	18	17	16	15	14	13	12
								TS31	TS30	TS29	TS28

Figure 8-7. ESSI Receive Slot Mask Register A (RSMA)

11	10	9	8	7	6	5	4	3	2	1	0
RS11	RS10	RS9	RS8	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0

23	22	21	20	19	18	17	16	15	14	13	12
								RS15	RS14	RS13	RS12

Figure 8-8. ESSI Receive Slot Mask Register B (RSMB)

11	10	9	8	7	6	5	4	3	2	1	0
RS27	RS26	RS25	RS24	RS23	RS22	RS21	RS20	RS19	RS18	RS17	RS16

23	22	21	20	19	18	17	16	15	14	13	12
								RS31	RS30	RS29	RS28

Reserved bit - read as zero should be written with zero for future compatibility

8.3.1 ESSI Control Register A (CRA)

CRA is one of two 24-bit read/write control registers used to direct the operation of the ESSI. The CRA controls the ESSI clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. The CRA control bits are described in the following paragraphs (see Figure 8-2).

8.3.1.1 CRA Prescale Modulus Select (PM7–PM0) Bits 7–0

The PM7–PM0 bits specify the divide ratio of the prescale divider in the ESSI clock generator. A divide ratio from 1 to 256 (PM=0 to $FF) may be selected. The bit clock output is available at the transmit clock (SCK) and/or the receive clock (SC0) pins of the DSP. The

bit clock output is also available internally for use as the bit clock to shift the transmit and receive shift registers. Careful choice of the crystal oscillator frequency and the prescaler modulus will allow the industry-standard codec master clock frequencies of 2.048 MHz, 1.544 MHz, and 1.536 MHz to be generated. Hardware and software reset clear PM0–PM7.

8.3.1.2 CRA Prescaler Range (PSR) Bit 11

The PSR controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When PSR is set, the fixed prescaler is bypassed. When PSR is cleared, the fixed divide-by-eight prescaler is operational. The maximum internally generated bit clock frequency is fosc/4, the minimum internally generated bit clock frequency is fosc/2/8/256=fosc/4096. Hardware and software reset clear PSR.

NOTE   The combination PSR = 1 and PM7-PM0 = $00 is forbidden and may cause synchronization problems if used.

8.3.1.3 CRA Frame Rate Divider Control (DC4–DC0) Bits 16–12

The DC4–DC0 bits control the divide ratio for the programmable frame rate dividers used to generate the frame clocks. In network mode, this ratio may be interpreted as the number of words per frame minus one. In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (DC=00000 to 11111) for normal mode and 2 to 32 (DC=00001 to 11111) for network mode.
A divide ratio of one (DC=00000) in network mode is a special case (on demand mode). In normal mode, a divide ratio of one (DC=00000) provides continuous periodic data word transfers. A bit-length sync (FSL1=1, FSL0=0) must be used in this case. Hardware and software reset clear DC4–DC0.

8.3.1.4 CRA Alignment Control (ALC) Bit 18

The ESSI is designed for 24 bit fractional data, thus shorter data words are left aligned to the MSB (bit 23). Several applications use 16 bit fractional data, thus shorter data words are left aligned to bit 15. Alignment Control (ALC) support this applications.
If ALC is set, received words are left aligned to bit 15 in the receive shift register and transmitted words are supposed to reside left aligned to bit 15 in the transmit/s shift register.
If ALC is cleared, received words are left aligned to bit 23 in the receive shift register and transmitted word are supposed to reside left align to bit 23 in the transmit/s shift register. Hardware and software reset clear ALC**.**

NOTE   While in ALC is set, 24 and 32 bits words use are forbidden, and word length control should be 8, 12 or 16, otherwise results are unpredictable.

8.3.1.5 CRA Word Length Control (WL2- WL0) Bits 21-19

The WL2, WL1 and WL0 bits are used to select the length of the data words being

transferred via the ESSI. Word lengths of 8, 12, 16, 24 or 32 bits may be selected according to the assignment described in Table 8-2. - ESSI word length selection

The data register of the ESSI are of 24 bit length and therefore the ESSI transmits 32 bit words by either duplicating the last bit 8 more times (WL2-WL0 = 100), or duplicating the first bit 8 more times (WL2-WL0 = 101).

Hardware and software reset clear WL2, WL1 and WL0.

Table 8-2. ESSI word length selection

WL2	WL1	WL0	Number of Bits/Word
0	0	0	8
0	0	1	12
0	1	0	16
0	1	1	24
1	0	0	32 (valid data in the first 24 bits)
1	0	1	32 (valid data in the last 24 bits)
1	1	0	Reserved
1	1	1	Reserved

8.3.1.6   CRA Reserved Bits 8-10,17,18,22,23

Bits 8-10,17, 22, 23 in the CRA are reserved bits they read as zeros and must be written with zero for future compatibility.

8.3.1.7   CRA Select SC1 as Transmitter#0 drive enable (SSC1) Bit 22

The SSC1 controls the functionality of SC1 pin. If the ESSI is configured as synchronous mode (SYN=1), and transmitter#2 is disable (TE2=0), and SSC1 is set while SC1 pin is configured as output (SCD1=1), the SC1 pin will function as the driver enable of transmitter#0 to enable the use of an external buffer on transmitter#0 output (STD). If on same conditions SSC1 is clear then the pin will function as the serial I/O flag.

8.3.2   ESSI Control Register B (CRB)

The CRB is one of two 24-bit read/write control registers used to direct the operation of the ESSI (see Figure 8-3). CRB controls the ESSI multifunction pins, SC2, SC1, and SC0, which can be used as clock inputs or outputs, frame synchronization pins, transmit data pins or serial I/O flag pins. The serial output flag control bits and the direction control bits for the serial control pins are in the ESSI CRB. Interrupt enable bits for the receiver and

the transmitter are provided in this control register. The number of enabled transmitters are enabled in this register (0,1,2 or 3 transmitters can be enabled). Operating modes are also selected in this register. Hardware and software reset clear all the bits in the CRB. The relationships between the ESSI pins (SC0, SC1, SC2, and SCK) and some of the CRB bits are summarized in Table 8-3 on page 13. The ESSI CRB bits are described in the following paragraphs.

8.3.2.1 CRB Serial Output Flag 0 (OF0) Bit 0

When the ESSI is in the synchronous clock mode and transmitter #1 is disabled (TE1 = 0) SC0 pin is configured as ESSI flag 0. The serial control direction zero bit (SCD0) when set, indicates that the SC0 pin is an output, then data present in OF0 will be written to SC0 at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode. Hardware and software reset clear OF0.

8.3.2.2 CRB Serial Output Flag 1 (OF1) Bit 1

When the ESSI is in the synchronous clock mode and transmitter #2 is disabled (TE2 = 0) SC1 pin is configured as ESSI flag 1. The serial control direction one bit (SCD1) when set, indicates that the SC1 pin is an output, then data present in OF1 will be written to SC1 at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode. Hardware and software reset clear OF1.

The normal sequence for setting output flags when transmitting data (by transmitter #0 through STD pin only) is: wait for TDE (TX0 empty) to be set, first write the flags, and then write the transmit data to the TX register. OF0 and OF1 are double buffered so that the flag states appear on the pins when the TX data is transferred to the transmit shift register (i.e., the flags are synchronous with the data). Hardware and software reset clear OF1.

NOTE The optional serial output pins timing (SC0, SC1, and SC2) are controlled by the frame timing and are not affected by TE2, TE1, TE0 or RE.

8.3.2.3 CRB Serial Control 0 Direction (SCD0) Bit 2

In synchronous mode (SYN=1), when transmitter #1 is disabled (TE1=0), or in asynchronous mode (SYN=0), SCD0 controls the direction of the SC0 I/O pin. When SCD0 is cleared, SC0 is an input; when SCD0 is set, SC0 is an output. Hardware and software reset clear SCD0. When TE1 is set SCD0 is ignored and the SC0 pin is always an output.

8.3.2.4 CRB Serial Control 1 Direction (SCD1) Bit 3

In synchronous mode (SYN=1), when transmitter #2 is disabled (TE2=0), or in asynchronous mode (SYN=0) SCD1 controls the direction of the SC1 I/O pin. When SCD1 is cleared, SC1 is an input; when SCD1 is set, SC1 is an output. Hardware and software reset clear SCD1. When TE2 is set SCD1 is ignored and the SC1 pin is always an output.

8.3.2.5 CRB Serial Control 2 Direction (SCD2) Bit 4

SCD2 controls the direction of the SC2 I/O pin. When SCD2 is cleared, SC2 is an input; when SCD2 is set, SC2 is an output. Hardware and software reset clear SCD2.

8.3.2.6 CRB Clock Source Direction (SCKD) Bit 5

SCKD selects the source of the clock signal used to clock the transmit shift register in the asynchronous mode and all the transmit shift register/s and the receive shift register in the synchronous mode. In asynchronous mode when SCKD is set, the internal clock source becomes the bit clock for the transmit shift register and word length divider and is the output on the SCK pin. When SCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCK pin, and an external clock source may drive this pin. Hardware and software reset clear SCKD.

8.3.2.7 CRB Shift Direction (SHFD) Bit 6

This bit causes the transmit shift register/s to shift data out MSB first when SHFD equals zero or LSB first when SHFD equals one. Received data is shifted in MSB first when SHFD equals zero or LSB first when SHFD equals zero. Hardware reset and software reset clear SHFD.

8.3.2.8 CRB Frame Sync Length (FSL0 and FSL1) Bits 7 and 8

These bits select the length of frame sync to be generated or recognized. If FSL1 equals to zero and FSL0 equals to zero, a word-length frame sync is selected for both TX and RX that is the length of the data word defined by bits WL2, WL1 and WL0. If FSL1 equals to one and FSL0 equals to zero, a 1-bit clock period frame sync is selected for both TX and RX. When FSL0 equals to one, the TX and RX frame syncs are different lengths. The encoding of FSL1 and FSL0 is described in the following table. Hardware reset and software reset clear FSL0 and FSL1.

FSL1	FSL0	Frame Sync Length
0	0	WL bit clock for both TX/RX
0	1	One-bit clock for TX and WL bit clock for RX
1	0	One-bit clock for both TX/RX
1	1	One-bit clock for RX and WL bit clock for TX

8.3.2.9 CRB Frame Sync Relative Timing (FSR) Bit 9

FSR determines the relative timing of the receive and transmit frame sync signal as referred to the serial data lines, for a word length frame sync only. When FSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When FSR is set the word length frame sync occurs one serial clock cycle earlier (i.e together with the last bit of the previous data word). Hardware reset and software reset

clear FSR.

8.3.2.10 CRB Frame Sync Polarity (FSP) Bit 10

FSP determines the polarity of the receive and transmit frame sync signals. When FSP is cleared the frame sync signal polarity is positive (i.e the frame start is signed by the high level of the frame sync pin). When FSP is set the frame sync signal polarity is negative (i.e the frame start is signed by the low level of the frame sync pin). Hardware reset and software reset clear FSP.

8.3.2.11 CRB Clock Polarity (CKP) Bit 11

The clock polarity bit controls on which bit clock edge data and frame sync are clocked out and latched in. If CKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the receive bit clock. If CKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the receive clock is used to latch the data and frame sync in. Hardware reset and software reset clear CKP.

8.3.2.12 CRB Synchronous /Asynchronous (SYN) Bit 12

SYN controls whether the receive and transmit functions of the ESSI occur synchronously or asynchronously with respect to each other. When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals. Only in the synchronous mode more than one transmitter can be enabled. Hardware reset and software reset clear SYN.

8.3.2.13 CRB ESSI Mode Select (MOD) Bit 13

MOD selects the operational mode of the ESSI. When MOD is cleared, the normal mode is selected; when MOD is set, the network mode is selected. In the normal mode, the frame rate divider determines the word transfer rate – one word is transferred per frame sync during the frame sync time slot. In network mode, a word is (possibly) transferred every time slot. For more details, see Section 8.4. Hardware and software reset clear MOD.

8.3.2.14 CRB ESSI Transmit #2 Enable (TE2) Bit 14

TE2 enables the transfer of data from TX2 to the transmit shift register #2. TE2 is functional only in synchronous mode (SYN =1) and ignored in asynchronous mode (SYN = 0). When TE2 is set (in synchronous mode) and a frame sync is detected, the transmit #2 portion of the ESSI is enabled for that frame. When TE2 is cleared (in synchronous mode), transmitter #2 will be disabled after completing transmission of data currently in the ESSI transmit shift register. Any data present in TX2 will not be transmitted (i.e., data can be written to TX2 with TE2 cleared; TDE will be cleared, but data will not be transferred to the transmit shift register #2).

The normal transmit enable sequence is to write data to one or more transmit data

registers (or TSR) before setting TEx. The normal transmit disable sequence is to clear TEx TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE2 and setting it again will disable the transmitter #2 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SC1 pin will remain in the high-impedance state. Keeping TE2 cleared until the start of the next frame will cause SC1 pin to act as serial I/O flag, (from the start of the frame) - in both normal and network mode.Hardware reset and software reset clear TE2.

The on-demand mode transmit enable sequence can be the same as the normal mode, or TE2 can be left enabled.

NOTE   TE2 does not affect the generation of frame sync or output flags.

8.3.2.15   CRB ESSI Transmit #1 Enable (TE1) Bit 15

TE1 enables the transfer of data from TX1 to the transmit shift register #1. TE1 is functional only in synchronous mode (SYN =1) and ignored in asynchronous mode (SYN = 0). When TE1 is set (in synchronous mode) and a frame sync is detected, the transmit #1 portion of the ESSI is enabled for that frame. When TE1 is cleared (in synchronous mode), the transmitter #1 will be disabled after completing transmission of data currently in the ESSI transmit shift register. Any data present in TX1 will not be transmitted (i.e., data can be written to TX1 with TE1 cleared; TDE will be cleared, but data will not be transferred to the transmit shift register #1).

The normal mode transmit enable sequence is to write data to one or more transmit data registers (or TSR) before setting TEx. The normal transmit disable sequence is to clear TEx TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE1 and setting it again will disable the transmitter #1 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SC0 pin will remain in the high-impedance state. Keeping TE1 cleared until the start of the next frame will cause SC0 pin to act as serial I/O flag, (from the start of the frame) - in both normal and network mode.Hardware reset and software reset clear TE1.

The on-demand mode transmit enable sequence can be the same as the normal mode, or TE1 can be left enabled.

NOTE   TE1 does not affect the generation of frame sync or output flags.

8.3.2.16   CRB ESSI Transmit #0 Enable (TE0) Bit 16

TE0 enables the transfer of data from TX0 to the transmit shift register #0. When TE0 is set and a frame sync is detected, the transmit #0 portion of the ESSI is enabled for that frame. When TE0 is cleared, the transmitter #0 will be disabled after completing transmission of data currently in the ESSI transmit shift register. The STD output is three-stated, and any data present in TX0 will not be transmitted (i.e., data can be written to TX0 with TE0 cleared; but data will not be transferred to the transmit shift register #0).

The normal mode transmit enable sequence is to write data to one or more transmit data registers (or TSR) before setting TEx. The normal transmit disable sequence is to clear

TEx TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE0 and setting it again will disable the transmitter #0 after completing transmission of the current data word until the beginning of the next frame. During that time period, the STD pin will remain in the high-impedance state. Hardware reset and software reset clear TE0.

The on-demand mode transmit enable sequence can be the same as the normal mode, or TE0 can be left enabled.

NOTE   Transmitter #0 is the only transmitter that can operate in asynchronous mode (SYN=0).

NOTE   TE0 does not affect the generation of frame sync or output flags.

Table 8-3. Mode and Pin Definition Table

Control Bits					ESSI PINS					
SYN	TE0	TE1	TE2	RE	SC0	SC1	SC2	SCK	STD	SRD
0	0	X	X	0	U	U	U	U	U	U
0	0	X	X	1	RXC	FSR	U	U	U	RD
0	1	X	X	0	U	U	FST	TXC	TD0	U
0	1	X	X	1	RXC	FSR	FST	TXC	TD0	RD
1	0	0	0	1/0	F0	F1/T0D	FS	XC	U	RD/U
1	0	0	1	1/0	F0	TD2	FS	XC	U	RD/U
1	0	1	0	1/0	TD1	F1/T0D	FS	XC	U	RD/U
1	0	1	1	1/0	TD1	TD2	FS	XC	U	RD/U
1	1	0	0	1/0	F0	F1/T0D	FS	XC	TD0	RD/U
1	1	0	1	1/0	F0	TD2	FS	XC	TD0	RD/U
1	1	1	0	1/0	TD1	F1/T0D	FS	XC	TD0	RD/U
1	1	1	1	1/0	TD1	TD2	FS	XC	TD0	RD/U

- • TXC - Transmitter Clock
- • RXC - Receiver Clock
- • XC - Transmitter/Receiver Clock (Synchronous Operation)
- • FST - Transmitter Frame Sync
- • FSR - Receiver Frame Sync
- • FS - Transmitter/Receiver Frame Sync (Synchronous Operation)
- • TD0 - Transmit Data pin #0
- • TD1 - Transmit Data pin #1
- • TD2 - Transmit Data pin #2

- • T0D - Transmitter#0 drive enable if SSC1=1 & SCD1=1
- • RD - Receive Data
- • F0 - Flag 0
- • F1 - Flag 1 if SSC1=0
- • U - Unused (may be used as GPIO pin)
- • X - Don't Care

8.3.2.17   CRB ESSI Receive Enable (RE) Bit 17

When RE is set, the receive portion of the ESSI is enabled. When this bit is cleared, the receiver will be disabled by inhibiting data transfer into RX. If data is being received while this bit is cleared, the remainder of the word will be shifted in and transferred to the ESSI receive data register.

RE must be set in the normal mode and on-demand mode to receive data. In network mode, the operation of clearing RE and setting it again will disable the receiver after reception of the current data word until the beginning of the next data frame. Hardware and software reset clear RE.

NOTE   RE does not affect the generation of a frame sync.

8.3.2.18   CRB ESSI Transmit Interrupt Enable (TIE) Bit 18

The DSP will be interrupted when TIE and the TDE flag in the ESSI status register are set. When TIE is cleared, this interrupt is disabled. Writing data to all the data register of the enabled transmitters or to TSR will clear TDE, thus clearing the interrupt.

Transmit interrupts with exception have higher priority than normal transmit data interrupts, therefore if exception occurs (TUE is set) and TEIE is set the ESSI will request an SSI transmit data with exception interrupt from the interrupt controller.

Hardware and software reset clear TIE.

8.3.2.19   CRB ESSI Receive Interrupt Enable (RIE) Bit 19

When RIE is set, the DSP will be interrupted when RDF in the ESSI status register is set. When RIE is cleared, this interrupt is disabled. Reading the receive data register will clear RDF, thus clearing the pending interrupt.

Receive interrupts with exception have higher priority than normal receive data interrupts, therefore if exception occurs (ROE is set) and REIE is set the ESSI will request an SSI receive data with exception interrupt from the interrupt controller.

Hardware and software reset clear RIE.

8.3.2.20   CRB ESSI Transmit Last slot Interrupt Enable (TLIE) Bit 20

TLIE enables an interrupt at the beginning of last slot of a frame in network mode. When TLIE is set the DSP will be interrupted at the start of the last slot in a frame in network mode regardless of the transmit mask register setting. When TLIE is cleared the transmit last slot interrupt is disabled. Hardware and software reset clear TLIE. TLIE is disabled when DC= $0 (on demand mode).

The use of the transmit last slot interrupt is described in paragraph 8.4.3 - ESSI

Exceptions.

8.3.2.21   CRB ESSI Receive Last slot Interrupt Enable (RLIE) Bit 21

RLIE enables an interrupt after the last slot of a frame ended in network mode only. When RLIE is set the DSP will be interrupted after the last slot in a frame ended regardless of the receive mask register setting. When RLIE is cleared the receive last slot interrupt is disabled. Hardware and software reset clear RLIE. RLIE is disabled when DC = $0 (on demand mode). The use of the Receive last slot interrupt is described in paragraph 8.4.3 - ESSI Exceptions.

8.3.2.22   CRB ESSI Transmit Exception Interrupt Enable (TEIE) Bit 22

When TEIE is set, the DSP will be interrupted when both TDE and TUE in the ESSI status register are set. When TEIE is cleared, this interrupt is disabled. Reading the status register followed by writing to all the data registers of the enabled transmitters will clear TUE, thus clearing the pending interrupt. Hardware and software reset clear TEIE

8.3.2.23   CRB ESSI Receive Exception Interrupt Enable (REIE) Bit 23

When REIE is set, the DSP will be interrupted when both RDF and ROE in the ESSI status register are set. When REIE is cleared, this interrupt is disabled. Reading the status register followed by reading the receive data register will clear ROE, thus clearing the pending interrupt. Hardware and software reset clear REIE

8.3.3   ESSI Status Register (SSISR)

The SSISR (see Figure 8-4) is an 8-bit read-only status register used by the DSP to read the status and serial input flags of the ESSI. When the SSISR is read to the internal data bus, the register contents occupy the low-order byte of the data bus, and the remaining bits are read as zeros. The status bits are described in the following paragraphs.

8.3.3.1   SSISR Serial Input Flag 0 (IF0) Bit 0

The ESSI latches data present on the SC0 pin during reception of the first received bit after frame sync is detected. IF0 bit is updated with this data when the receive shift register is transferred into the receive data register. The IF0 bit is enabled only when SC0 is programmed as ESSI in the Port Control Register and SYN is set, TE1 is cleared and SCD0 is cleared, indicating that SC0 is an input flag and the synchronous mode is selected; otherwise, IF0 reads as a zero when it is not enabled. Hardware, software, ESSI individual, and STOP reset clear IF0.

8.3.3.2   SSISR Serial Input Flag 1 (IF1) Bit 1

The ESSI latches data present on the SC1 pin during reception of the first received bit after frame sync is detected. IF1 bit is updated with this data when the receiver shift register is transferred into the receive data register. The IF1 bit is enabled only when SC1 is programmed as ESSI in the Port Control Register and SYN is set, TE2 is cleared and SCD1 is cleared, indicating that SC1 is an input flag and the synchronous mode is

selected; otherwise, IF1 reads as a zero when it is not enabled. Hardware, software, ESSI individual, and STOP reset clear IF1.

8.3.3.3 SSISR Transmit Frame Sync Flag (TFS) Bit 2

When set, TFS indicates that a transmit frame sync occurred in the current time slot. TFS is set at the start of the first time slot in the frame and cleared during all other time slots. Data written to a transmit data register during the time slot when TFS is set will be transmitted (in network mode), if the transmitter is enabled, during the second time slot in the frame. TFS is useful in network mode to identify the start of a frame. TFS is cleared by hardware, software, ESSI individual, or STOP reset. TFS is valid only if at least one transmitter is enabled (TE0, TE1 or TE2 equal 1).

NOTE In normal mode, TFS will always be read as a one when transmitting data because there is only one time slot per frame – the "frame sync" time slot.

8.3.3.4 SSISR Receive Frame Sync Flag (RFS) Bit 3

When set, RFS indicates that a receive frame sync occurred during reception of the word in the serial receive data register. This indicates that the data word is from the first time slot in the frame. When RFS is clear and a word is received, it indicates (only in the network mode) that the frame sync did not occur during reception of that word.
RFS is cleared by hardware, software, ESSI individual, or STOP reset. RFS is valid only if the receiver is enabled (RE=1).

NOTE In normal mode, RFS will always read as a one when reading data because there is only one time slot per frame – the "frame sync" time slot.

8.3.3.5 SSISR Transmitter Underrun Error Flag (TUE) Bit 4

TUE is set when at least one of the enabled serial transmit shift registers is empty (no new data to be transmitted) and a transmit time slot occurs. When a transmit underrun error occurs, the previous data (which is still present in the TX registers that were not written) will be retransmitted.
In the normal mode, there is only one transmit time slot per frame. In the network mode, there can be up to 32 transmit time slots per frame.
If TEIE is set, a DSP transmit underrun error interrupt request will be issued when TUE is set. Hardware, software, ESSI individual, and STOP reset clear TUE. TUE is also cleared by reading the SSISR with TUE set, followed by writing to all the enabled transmit data registers or to TSR.

8.3.3.6 SSISR Receiver Overrun Error Flag (ROE) Bit 5

This flag is set when the serial receive shift register is filled and ready to transfer to the receiver data register (RX) and RX is already full (i.e., RDF=1). If REIE is set, a DSP receiver overrun error interrupt request will be issued when ROE is set. Hardware,

software, ESSI individual, and STOP reset clear ROE. ROE is also cleared by reading the SSISR with ROE set, followed by reading the RX.

8.3.3.7 SSISR ESSI Transmit Data Register Empty (TDE) Bit 6

This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift register/s; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TDE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TDE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, a DSP transmit data interrupt request will be issued when TDE is set. Hardware, software, ESSI individual, and STOP reset clear TDE.

8.3.3.8 SSISR ESSI Receive Data Register Full (RDF) Bit 7

RDF is set when the contents of the receive shift register are transferred to the receive data register. RDF is cleared when the DSP reads the receive data register or cleared by hardware, software, ESSI individual, or STOP reset. If RIE is set, a DSP receive data interrupt request will be issued when RDF is set.

8.3.4 ESSI Receive Shift Register

This 24-bit shift register receives the incoming data from the serial receive data pin. Data is shifted in by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. Data is assumed to be received MSB first if SHFD equals zero and LSB first if SHFD equals one. Data is transferred to the ESSI receive data register after 8, 12, 16, 24, or 32 serial clock cycles were counted, depending on the word-length control bits in the CRA.

8.3.5 ESSI Receive Data Register (RX)

RX is a 24-bit read-only register that accepts data from the receive shift register as it becomes full. The data read will occupy the most significant portion of the receive data register according to ALC setting (When ALC=0, the m.s.b is bit 23, or bit 15 when ALC=1). The unused bits (least significant portion, and 8 most significant bits when ALC=1) will read as zeros. The DSP is interrupted whenever RX becomes full if the associated interrupt is enabled.

8.3.6 ESSI Transmit Shift Registers

These three 24-bit shift registers contain the data being transmitted. Data is shifted out to the serial transmit data pins by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. The number of bits shifted out before the shift registers are considered empty and may be written to again can be 8, 12, 16, 24 or 32 bits (determined by the word-length control bits in CRA). The data to be transmitted occupies the most significant portion of the shift registers according to ALC setting (When ALC=0,

the m.s.b is bit 23, or bit 15 when ALC=1). The unused portion of the registers is ignored. Data is shifted out of these registers MSB first if SHFD equals zero and LSB first if SHFD equals one.

8.3.7   ESSI Transmit Data Registers (TX2,TX1,TX0)

TX2, TX1 and TX0 are 24-bit write-only registers. Data to be transmitted is written into these registers and is automatically transferred to the transmit shift registers. The data written (8, 12, 16, or 24 bits) should occupy the most significant portion of the TXx according to ALC setting (When ALC=0, the m.s.b is bit 23, or bit 15 when ALC=1). The unused bits (least significant portion, and 8 most significant bits when ALC=1) of the TXx are don't care bits. The DSP is interrupted whenever the TXx becomes empty if the transmit data register empty interrupt has been enabled.

8.3.8   ESSI Time Slot Register (TSR)

TSR is effectively a null data register that is used when the data is not to be transmitted in the available transmit time slot. For the purposes of timing, TSR is a write-only register that behaves like an alternative transmit data register, except that, rather than transmitting data, the transmit data pins of all the enabled transmitters are in the high-impedance state for that time slot.

8.3.9   Transmit Slot Mask Registers (TSMA, TSMB)

The Transmit Slot Mask Registers are two 16-bit read/write registers. When the TSMA or TSMB is read to the internal data bus, the register contents occupy the two low-order bytes of the data bus, and the high-order byte is zero filled. They are used by the transmitter/s in network mode to determine for each slot whether to transmit a data word and generate a transmitter empty condition (TDE=1), or to three-state the transmitter/s data pin/s. TSMA and TSMB (see Figure 8-5 and Figure 8-6) should be seen as only one 32-bit register, TSM. Bit number N in TSM (TS**) is an enable/disable control bit for transmission in slot number N.

When bit number N in TSM is cleared, all the transmit data pins of the enabled transmitters are three-stated during transmit time slot number N. The data is still transferred from the Transmit Data Register/s to the transmit shift register/s and the Transmit Data Empty flag (TDE) is not set. Also the Transmitter Underrun Error flag is not set. This means that during a disabled slot, no Transmitter Empty interrupt is generated. The DSP is interrupted only for enabled slots. Data that is written to the Transmit Data Register/s when servicing this request is transmitted in the next enabled transmit time slot.

When bit number N in TSM register is set, the transmit sequence is as usual: data is transferred from TX register/s to the shift register/s, it is transmitted during slot number N, and TDE flag is set.

Using the slot mask in TSM does not conflict with using TSR. Even if a slot is enabled in TSM, the user may chose to write to TSR instead of writing to the transmit data registers TXx. This will cause all the transmit data pins of the enabled transmitters to be three-stated during the next slot.

Data written to the TSM will affect next frame transmission. The frame being transmitted

is not affected by this data and would comply to last TSM setting. Data read from TSM will return last written data.

After hardware or software reset, the TSM register is preset to $FFFFFFFF, which means that all 32 possible slots are enabled for data transmission.

8.3.10   Receive Slot Mask Registers (RSMA, RSMB)

The Receive Slot Mask Registers are two 16-bit read/write registers. When the RSMA or RSMB is read to the internal data bus, the register contents occupy the two low-order bytes of the data bus, and the high-order byte is zero filled. They are used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. RSMA and RSMB (see Figure 8-7 and Figure 8-8) should be seen as only one 32-bit register, RSM. Bit number N in RSM (RS**) is an enable/disable control bit for receiving data in slot number N.

When bit number N in RSM register is cleared, the data from receive data pin SRD is shifted into the Receive Shift Register during slot number N. The data is not transferred from the Receive Shift Register to the Receive Data Register and the Receiver Data Full flag (RDF) is not set. Also the Receiver Overrun Error flag is not set. This means that during a disabled slot, no Receiver Full interrupt is generated. The DSP is interrupted only for enabled slots.

When bit number N in RSM is set, the receive sequence is as usual: data which is shifted into the receive shift register is transferred to the Receive Data register and RDF flag is set.

Data written to the RSM will affect next received frame. The frame being received is not affected by this data and would comply to last RSM setting. Data read from RSM will return last written data.

After hardware or software reset, the RSM register is preset to $FFFFFFFF, which means that all 32 possible slots are enabled for data reception.

8.4   OPERATING MODES

ESSI operating mode are selected by the ESSI control registers (CRA and CRB). The main operating mode are described in the following paragraphs.

8.4.1   ESSI After Reset

Hardware or software reset clears the port control register bits and the port direction control register bits, which configure all I/O as general-purpose input. The ESSI is reset while all ESSI pins are programmed as general-purpose I/O and is active only if at least one of the ESSI I/O pins is programmed as ESSI pin.

8.4.2   ESSI Initialization

The correct way to initialize the ESSI is as follows:

1. Hardware, software, ESSI individual, or STOP reset
2. Program ESSI control and time slot registers
3. Write data to all the enabled transmitters.
4. Configure at least one pin as ESSI pin

During program execution, PC6-PC0 bits in the GPIO port control register (PCR) may be cleared causing the ESSI to stop serial activity and enter the individual reset state. All status bits of the interface will be set to their reset state; however, the contents of CRA and CRB are not affected. This procedure allows the DSP program to reset each interface separately from the other internal peripherals. During individual reset internal DMA accesses to the data registers of the ESSI are not valid and data read will be unexpected.

The DSP program must use an ESSI reset when changing the ESSI control registers (except for TEIE, REIE, TLIE, RLIE, TIE, RIE, TE2, TE1, TE0, or RE) to ensure proper operation of the interface.

8.4.3   ESSI Exceptions

The ESSI can generate six different exceptions (ordered from the highest to the lowest priority)**:**

1. ESSI Receive Data with Exception Status – occurs when the receive exception interrupt is enabled, the receive data register is full, and a receiver overrun error has occurred. ROE is cleared by first reading the SSISR and then reading RX.
2. ESSI Receive Data – occurs when the receive interrupt is enabled, the receive data register is full, and no receive error conditions exist. Reading RX clears the pending interrupt. This error-free interrupt can use a fast interrupt service routine for minimum overhead.
3. ESSI Receive Last slot interrupt occurs after the last slot of the frame ended (in network mode only) regardless of the receive mask register setting. The Receive Last slot interrupt may be used for resetting the receive mask slot register, reconfigure the DMA channels and reassignment of data memory pointers. Using the Receive last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame will be serviced with the new setting without synchronization problems. Note that the maximum Receive last slot interrupt service time, should not exceed N-1 ESSI bits service time (Where N is the number of bits in a slot).
4. ESSI Transmit Data with Exception Status – occurs when the transmit exception interrupt is enabled, at least one transmit data register of the enabled transmitters is empty, and a transmitter underrun error has occurred. TUE is cleared by first reading the SSISR and then writing to all the transmit data registers of the enabled transmitters, or to the TSR to clear the pending interrupt.
5. ESSI Transmit Last slot interrupt occurs at the start of the last slot of the frame in network mode regardless of the transmit mask register setting.

The Transmit Last slot interrupt may be used for resetting the transmit mask slot register, reconfigure the DMA channels and reassignment of data memory pointers. Using the Transmit last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame will be serviced with the new setting without synchronization problems.Note that the maximum Transmit last slot interrupt service time, should not exceed N-1 ESSI bits service time (Where N is the number of bits in a slot).

6. ESSI Transmit Data – occurs when the transmit interrupt is enabled, and at least one of the enabled transmit data registers is empty, and no transmitter error conditions exist. Writing to all the TX registers of the enabled transmitters, or to the TSR will clear this interrupt. This error-free interrupt may use a fast interrupt service routine for minimum overhead (if no more than two transmitters are used).

8.4.4   Operating Modes – Normal, Network, and On-Demand

The ESSI has three basic operating modes and many data/operation formats. These modes can be programmed by several bits in the ESSI control registers.
The data/operation formats available to the ESSI are selected by setting or clearing control bits in the CRA and CRB. These control bits are WL2, WL1, WL0, MOD, SYN, FSL1, FSL0, FSR, FSP, CKP and SHFD.

8.4.4.1   Normal/Network/On-Demand Mode Selection

Selecting between the normal mode and network mode is accomplished by clearing or setting the MOD bit in the CRB. For normal mode, the ESSI functions with one data word of I/O per frame (per enabled transmitter). For the network mode, 2 to 32 time slots per frame may be selected. During each frame, 0 to 32 data words of I/O may be received/transmitted (from each enabled transmitter). In either case, the transfers are periodic. The normal mode is typically used to transfer data to/from a single device. Network mode is typically used in time division multiplexed (TDM) networks of codecs or DSPs with multiple words per frame.
Setting the MOD bit in the CRB, as for network mode, and setting the frame rate divider to zero (DC=00000) selects the on demand mode.This special case will not generate a periodic frame sync. A frame sync pulse will be generated only when data is available to transmit. The frame sync signal indicates the first time slot in the frame. The on-demand mode requires that the transmit frame sync be internal (output) and the receive frame sync be external (input). Therefore, for simplex operation, the synchronous mode could be used; however, for full-duplex operation, the asynchronous mode must be used. Data transmission that is data driven is enabled by writing data into each TX. Although the ESSI is double buffered, only one word can be written to each TX, even if the transmit shift register is empty. The receive and transmit interrupts, function as usual using TDE and RDF; however, transmit underruns are impossible for on-demand transmission and are disabled. This mode is useful for interfacing to codecs requiring a continuous clock.

8.4.4.2 Synchronous/Asynchronous Operating Modes

The transmit and receive sections of this interface may be synchronous or asynchronous – i.e., the transmitter and receiver may use common clock and synchronization signals or they may have their own separate clock and sync signals (asynchronous operating mode). The SYN bit in CRB selects synchronous or asynchronous operation. Since the ESSI is designed to operate either synchronously or asynchronously, separate receive and transmit interrupts are provided.

When SYN equals zero, the ESSI TX and RX clocks and frame sync sources are independent. If SYN equals one, the ESSI TX and RX clocks and frame sync come from the same source (either external or internal).

Transmitter #1 and transmitter #2 can operate only in synchronous mode.

Data clock and frame sync signals can be generated internally by the DSP or may be obtained from external sources. If internally generated, the ESSI clock generator is used to derive bit clock and frame sync signals from the DSP internal system clock. The ESSI clock generator consists of a selectable fixed prescaler and a programmable prescaler for bit rate clock generation and also a programmable frame-rate divider and a word-length divider for frame-rate sync-signal generation.

8.4.4.3 Frame Sync Selection

The transmitter and receiver can operate totally independent of each other. The transmitter can have either a bit-long or word-long frame-sync signal format, and the receiver can have the same or opposite format. The selection is made by programming FSL0 and FSL1 in the CRB.

1. If FSL1 equals zero, the RX frame sync is asserted during the entire data transfer period. This frame sync length is compatible with Motorola codecs, SPI serial peripherals, serial A/D and D/A converters, shift registers, and telecommunication PCM serial I/O.
2. If FSL1 equals one, the RX frame sync pulses active for one bit clock immediately before the data transfer period. This frame sync length is compatible with Intel and National components, codecs, and telecommunication PCM serial I/O.

The ability to mix frame sync lengths is useful in configuring systems in which data is received from one type device (e.g., codec) and transmitted to a different type device.

FSL0 controls whether RX and TX have the same frame sync length. If FSL0 equals zero, RX and TX have the same frame sync length, which is selected by FSL1. If FSL0 equals one, RX and TX have different frame sync lengths, which are selected by FSL1. FSL0 is ignored when the SYN bit is set.

FSR controls the relative timing of the word length frame sync as referred to the data word. When FSR is cleared the word length frame sync is generated (or expected) with the first bit of the data word. When FSR is set the word length frame sync is generated (or expected) with the last bit of the previous word. FSR is ignored when a bit length frame sync is selected.

FSP controls the polarity of the frame sync. When FSP is cleared the polarity of the frame

sync is positive i.e. the frame sync signal is asserted high. When FSP is set the polarity of the frame sync is negative i.e. the frame sync is asserted low.

The ESSI receiver looks for a receive frame sync leading edge (trailing edge if FSP is set) only when the previous frame is completed. If the frame sync goes high before the frame is completed (or before the last bit of the frame is received in the case of a bit frame sync or a word length frame sync with FSR set), the current frame sync will not be recognized, and the receiver will be internally disabled until the next frame sync. Frames do not have to be adjacent – i.e., a new frame sync does not have to immediately follow the previous frame. Gaps of arbitrary periods can occur between frames. All the enabled transmitters will be three-stated during these gaps.

8.4.4.4   Shift Direction Selection

Some data formats, such as those used by codecs, specify MSB first other data formats, such as the AES-EBU digital audio, specify LSB first. To interface with devices from both systems, the shift registers in the ESSI are bidirectional. The MSB/LSB selection is made by programming SHFD in the CRB.

If SHFD equals zero, data is shifted into the receive shift register MSB first and shifted out of the transmit shift register MSB first. If SHFD equals one, data is shifted into the receive shift register LSB first and shifted out of the transmit shift register LSB first.

8.4.5   Flags

Two ESSI pins (SC1 and SC0) are available as serial I/O flags. Their operation can be controlled by SYN, SCD0, SCD1, SSC1, TE1 and TE2 bits in the CRB/CRA.The control bits (OF1 and OF0) and status bits (IF1 and IF0) are double buffered to/from SC1 and SC0. Double buffering the flags keeps them in sync with TX and RX.

The flags are available in the synchronous mode only (SYN=1). Each flag can be separately programmed. Flag 0 is enabled when transmitter #1 is disabled (TE1=0), its direction is selected by SCD0, SCD0=1 as output and SCD0=0 as input. In the same way flag1 is enabled when transmitter #2 is disabled (TE2=0), the pin is not configured as transmitter drive enable (SSC1=0) and its direction is selected by SCD1, SCD1=1 as output and SCD1=0 as input.

When programmed as input, SC0 and SC1 value, respectively, are latched at the same time as the first bit is sampled of the receive data word. Since the input was latched, the signal on the input flag pin (SC0 and SC1), can change without affecting the input flag until the first bit of the next receive data word. When received data word is latched by RX, the latched values are then latched by the SSISR IF0 and IF1 bits respectively and can be read by software.

When programed as output, SC0 and SC1 value, is driven by the value, from the CRB OF0 and OF1 bits respectively, latched when the TX is transferred to the transmit shift register. The value on SC0 or SC1, will be stable from the same time the first bit of the transmit data word is transmitted until the first bit of the next transmit data word is transmitted. Software can change the CRB OF0 and OF1 values and thus controlling the SC0 and SC1 pin values for each transmitted word.

8.5 GPIO - pins and registers

The GPIO functionality of port ESSI is controlled by three registers: Port Control register (PCR), Port Direction register (PRR) and Port Data register (PDR).

8.5.1 Port Control Register (PCR)

The read/write 16 bit Port Control Register controls the functionality of ESSI GPIO pins. When the PCR is read to the internal data bus, the register contents occupy the two low-order bytes of the data bus, and the remaining bits are read as zeros. When written the most significant byte should be written zero for future compatibility.

Figure 8-9. Port Control Register (PCR)
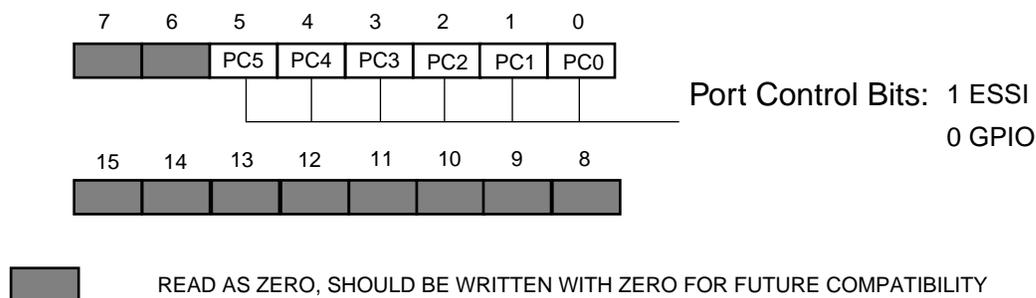
Each of PC(5:0) bits controls the functionality of the corresponding port pin. When a PC[i] bit is set, the corresponding port pin is configured as a ESSI pin. When a PC[i] bit is cleared, the corresponding port pin is configured as GPIO pin.

Hardware and software reset clear all PCR bits.

8.5.2 Port Direction Register (PRR)

The read/write 16 bit Port Direction Register controls the direction of ESSI GPIO pins. When the PRR is read to the internal data bus, the register contents occupy the two low-order bytes of the data bus, and the remaining bits are read as zeros. When written the most significant byte should be written zero for future compatibility.

Figure 8-10. Port Direction Register (PRR)
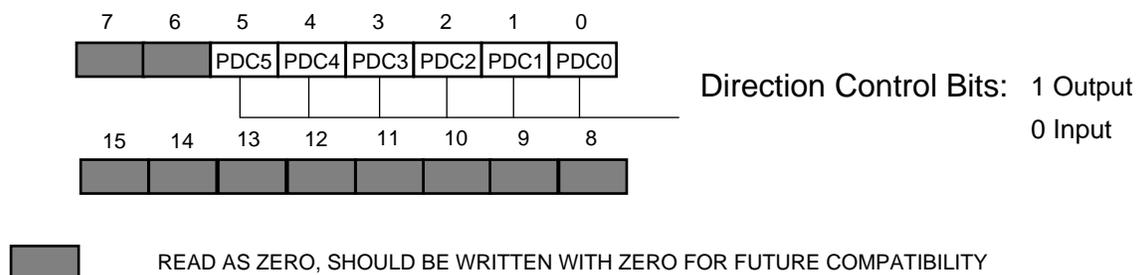
When port pin[i] is configured as GPIO, PDC[i] controls the port pin direction. When PDC[i] is set, the GPIO port pin[i] is configured as output. When PDC[i] is cleared the GPIO port pin[i] is configured as input.

Hardware and software reset clear all PRR bits.

The following table describe the port pin configurations.

Table 8-4. Port Control Register and Port Direction Register bits functionality

PC[i]	PDC[i]	Port Pin[i] Function
1	X	ESSI
0	0	GPIO input
0	1	GPIO output

8.5.3   Port Data register (PDR)

The read/write 16 bit Port Data Register is used to read or write data to/from ESSI GPIO pins.Bits PD(5:0) are used to read or write data from/to the corresponding port pins if they are configured as GPIO (by PC(5:0) bits in PCR). If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit will reflect the value present on this pin. If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit will be reflected on the this pin.When the PDR is read to the internal data bus, the register contents occupy the two low-order bytes of the data bus, and the remaining bits are read as zeros. When written the most significant byte should be written zero for future compatibility.

Figure 8-11. Port Data Register (PDR)

7	6	5	4	3	2	1	0
		PD5	PD4	PD3	PD2	PD1	PD0

15	14	13	12	11	10	9	8

READ AS ZERO, SHOULD BE WRITTEN WITH ZERO FOR FUTURE COMPATIBILITY
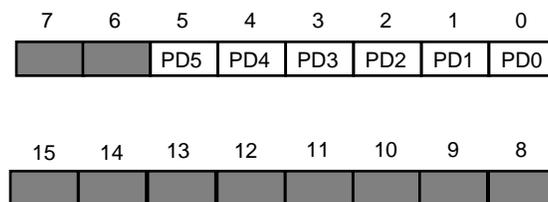
Hardware and software reset clear all PDR bits.

9 SERIAL COMMUNICATION INTERFACE (SCI)

9.1 INTRODUCTION

The SCI provides a full-duplex port for serial communication to other DSPs, microprocessors, or peripherals such as modems. The communication can be TTL-level signals or, with additional external logic, RS232C, RS422, etc.

This interface uses three dedicated pins: transmit data (TXD), receive data (RXD), and SCI serial clock (SCLK). It supports industry-standard asynchronous bit rates and protocols as well as high-speed synchronous data transmission (up to 8.25 Mbps for a 66-MHz clock). The asynchronous protocols include a multidrop mode for master/slave operation with Wakeup On Idle Line and Wakeup On Address Bit capability.

The SCI consists of separate transmit and receive sections whose operation can be asynchronous with respect to one another. A programmable baud-rate generator provides the transmit and receive clocks. An enable vector and an interrupt vector have been included so that the baud-rate generator can function as a general-purpose timer when it is not being used by the SCI or when the interrupt timing is the same as that used by the SCI.

9.2 SCI I/O Pins

Each of the three SCI pins can be configured as either a general-purpose I/O or as a specific SCI pin. Each pin is independent of the others (e.g. if only TXD is needed, RXD and SCLK can be programmed for general-purpose I/O). However, at least one of the three pins must be selected as an SCI pin to release the SCI from reset.

SCI interrupts may be enabled by programming the SCI control registers before any of the SCI pins are programmed as SCI functions. In this case, only one transmit interrupt can be generated because the transmit data register is empty. The timer and timer interrupt will operate when one or more of the SCI pins is programmed as an SCI pin.

9.2.1 Receive Data (RXD)

This input receives byte-oriented serial data and transfers the data to the SCI receive shift register. Asynchronous input data is sampled on the positive edge of the receive clock (1X SCLK) if SCKP equals zero. RXD may be programmed as a general-purpose I/O pin (PE0) when the SCI RXD function is not being used.

9.2.2   Transmit Data (TXD)

This output transmits serial data from the SCI transmit shift register. Data changes on the negative edge of the asynchronous transmit clock (SCLK) if SCKP equals zero. This output is stable on the positive edge of the transmit clock. TXD may be programmed as a general-purpose I/O pin (PE1) when the SCI TXD function is not being used.

9.2.3   SCI Serial Clock (SCLK)

This bidirectional pin provides an input or output clock from which the transmit and/or receive baud rate is derived in the asynchronous mode and from which data is transferred in the synchronous mode. SCLK may be programmed as a general-purpose I/O pin (PE2) when the SCI SCLK function is not being used.   This pin may be programmed as PE2 when data is being transmitted on TXD since, in the asynchronous mode, the clock need not be transmitted. There is no connection between programming the PE2 pin as SCLK and data coming out the TXD pin because SCLK is independent of SCI data I/O.

9.3   SCI PROGRAMMING MODEL

The registers comprising the SCI are shown in Figure 9-1 and Figure 9-2. These registers are the SCI control register (SCR), SCI status register (SSR), SCI clock control register (SCCR), SCI receive data registers (SRX), SCI transmit data registers (STX), and the SCI transmit data address register (STXA). The SCI programming model can be viewed as three types of registers:

1.   control – SCR and SCCR in Figure 9-1;
2.   status – SSR in Figure 9-1; and
3.   data transfer – SRX, STX, and STXA in Figure 9-2.

The SCI contains also the GPIO functionality, described at Chapter 9.5.
The following paragraphs describe each bit in the programming model.

9.3.1   SCI Control Register (SCR)

The SCR is a 24 -bit read/write register that controls the serial interface operation. seventeen of the 24 bits are currently defined. Each bit is described in the following paragraphs.

9.3.1.1   SCR Word Select (WDS0, WDS1, WDS2) Bits 0, 1, and 2

The three word-select bits (WDS0, WDS1, WDS2) select the format of the transmit and receive data. The formats include three asynchronous and one multidrop asynchronous mode, as well as an 8-bit synchronous (shift register) mode. The asynchronous modes are compatible with most UART-type serial devices. Standard RS232C communication links are supported by these modes.

Figure 9-1. SCI Programming Model – Control and Status Registers

SCI Control Register (SCR)

7	6	5	4	3	2	1	0
WOMS	RWU	WAKE	SBK	SSFTD	WDS2	WDS1	WDS0

15	14	13	12	11	10	9	8
SCKP	STIR	TMIE	TIE	RIE	ILIE	TE	RE

23	22	21	20	19	18	17	16
							REIE

SCI Status Register (SSR)

7	6	5	4	3	2	1	0
R8	FE	PE	OR	IDLE	RDRF	TDRE	TRNE

15	14	13	12	11	10	9	8

Reserved bit - read as zero should be written with zero for future compatibility

SCI Clock Control Register (SCCR)

7	6	5	4	3	2	1	0
CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0

15	14	13	12	11	10	9	8
TCM	RCM	SCP	COD	CD11	CD10	CD9	CD8

The multidrop asynchronous modes are compatible with the MC68681 DUART, the M68HC11 SCI interface, and the Intel 8051 serial interface.

The synchronous data mode is essentially a high-speed shift register used for I/O expansion and stream-mode channel interfaces. Data synchronization is accomplished by the use of a gated transmit and receive clock that is compatible with the Intel 8051 serial interface mode 0. These formats are indicated in Table 9-1. The word-select bits are cleared by hardware reset.

When odd parity is selected, the transmitter will count the number of ones in the data word; if the total is not an odd number, the parity bit is made equal to one and thus

produces an odd number. If the receiver counts an even number of ones, an error in transmission has occurred. When even parity is selected, an even number must result from the calculation performed at both ends of the line or an error in transmission has occurred. The three word-select bits are cleared by hardware and software reset.

Table 9-1. Word Formats

WDS 2	WDS 1	WDS 0	Word Formats
0	0	0	8-Bit Synchronous Data (shift register mode)
0	0	1	Reserved
0	1	0	10-Bit Asynchronous (1 start, 8 data, 1 stop)
0	1	1	Reserved
1	0	0	11-Bit Asynchronous (1 start, 8 data, 1 even parity, 1 stop)
1	0	1	11-Bit Asynchronous (1 start, 8 data, 1 odd parity, 1 stop)
1	1	0	11-Bit Multidrop (1 start, 8 data, 1 data type, 1 stop)
1	1	1	Reserved

9.3.1.2   SCR SCI Shift Direction (SSFTD) Bit 3

The SCI data shift registers can be programmed to shift data in/out either LSB first - if SSFTD equals zero; or MSB first - if SSFTD equals one. The parity and data type bits do not change their position in the frame, and remain adjacent to the stop bit. SSFTD is cleared by hardware and software reset

9.3.1.3   SCR Send Break (SBK) Bit 4

A break is an all-zero word frame – a start bit zero and a character of all zeros (including any parity), and a stop bit zero: i.e., 10 or 11 zeros depending on the WDS mode selected. If SBK is set and then cleared, the transmitter completes transmission of the current frame, sends 10 or 11 zeros, and reverts to idle or sending data. If SBK remains set, the transmitter will continually send whole frames of zeros (10 or 11 bits with no stop bit). At the completion of the break code, the transmitter sends at least one high bit before transmitting any data to guarantee recognition of a valid start bit. Break can be used to signal an unusual condition, message, etc. by forcing a frame error, which is caused by a missing stop bit.   Hardware and software reset clear SBK.

9.3.1.4   SCR Wakeup Mode Select (WAKE) Bit 5

When WAKE equals zero, the Wakeup On Idle Line mode is selected. In the Wakeup On Idle Line mode, the SCI receiver is re-enabled by an idle string of at least 10 or 11

(depending on WDS mode) consecutive ones. The transmitter's software must provide this idle string between consecutive messages. The idle string cannot occur within a valid message because each word frame contains a start bit that is a zero.

When WAKE equals one, the wakeup on address bit mode is selected. In the Wakeup On Address Bit mode, the SCI receiver is re-enabled when the last (eighth or ninth) data bit received in a character (frame) is one. The ninth data bit is the address bit (R8) in the 11-bit multidrop mode; the eighth data bit is the address bit in the 10-bit asynchronous and 11-bit asynchronous with parity modes. Thus, the received character is an address that has to be processed by all sleeping processors – i.e., each processor has to compare the received character with its own address and decide whether to receive or ignore all following characters. WAKE is cleared by hardware and software reset.

9.3.1.5   SCR Receiver Wakeup Enable (RWU) Bit 6

When RWU equals one and the SCI is in an asynchronous mode, the wakeup function is enabled – i.e., the SCI is put to sleep, and will be woken by the event defined by the WAKE bit. In the Sleep state, all receive flags, except IDLE, and interrupts are disabled. When the receiver wakes up, RWU is cleared by the wakeup hardware. The programmer may also clear the RWU bit to wake up the receiver.

RWU can be used by the programmer to ignore messages that are for other devices on a multidrop serial network. Wakeup On Idle Line (WAKE=0) or Wakeup On Address Bit (WAKE=1) must be chosen.

1.   When WAKE equals zero and RWU equals one, the receiver will not respond to data on the data line until an idle line is detected.
2.   When WAKE equals one and RWU equals one, the receiver will not respond to data on the data line until a data byte with bit 9 equal to one is detected.

When the receiver wakes up, the RWU bit is cleared, and the first byte of data is received. If interrupts are enabled, the CPU will be interrupted, and the interrupt routine will read the message header to determine if the message is intended for this DSP.

1.   If the message is for this DSP, the message will be received, and RWU will again be set to one to wait for the next message.
2.   If the message is not for this DSP, the DSP will immediately set RWU to one. Setting RWU to one causes the DSP to ignore the remainder of the message and wait for the next message.

RWU is cleared by hardware and software reset. RWU is ignored in the synchronous mode.

9.3.1.6   SCR Wired-OR Mode Select (WOMS) Bit 7

When the WOMS bit is set, the SCI TXD driver is programmed to function as an open-drain output and may be wired together with other TXD pins in an appropriate bus

configuration such as a master-slave multidrop configuration. An external pullup resistor is required on the bus. When the WOMS is cleared, the TXD pin uses an active internal pullup. This bit is cleared by hardware and software reset.

9.3.1.7 SCR Receiver Enable (RE) Bit 8

When RE is set, the receiver is enabled. When RE is cleared, the receiver is disabled, and data transfer is inhibited to the receive data register (SRX) from the receive shift register. If RE is cleared while a character is being received, the reception of the character will be completed before the receiver is disabled. RE does not inhibit RDRF or receive interrupts. RE is cleared by a hardware and software reset.

9.3.1.8 SCR Transmitter Enable (TE) Bit 9

When TE is set, the transmitter is enabled. When TE is cleared, the transmitter will complete transmission of data in the SCI transmit data shift register; then the serial output is forced high (idle). Data present in the SCI transmit data register (STX) will not be transmitted. STX may be written and TDRE will be cleared, but the data will not be transferred into the shift register. TE does not inhibit TDRE or transmit interrupts. TE is cleared by a hardware and software reset.
Setting TE will cause the transmitter to send a preamble of 10 or 11 consecutive ones (depending on WDS). This procedure gives the programmer a convenient way to ensure that the line goes idle before starting a new message. To force this separation of messages by the minimum idle line time, the following sequence is recommended:

1.   Write the last byte of the first message to STX.
2.   Wait for TDRE to go high, indicating the last byte has been transferred to the transmit shift register.
3.   Clear TE and set TE back to one. This queues an idle line preamble to immediately follow the transmission of the last character of the message (including the stop bit).
4.   Write the first byte of the second message to STX.

In this sequence, if the first byte of the second message is not transferred to the STX prior to the finish of the preamble transmission, then the transmit data line will simply mark idle until STX is finally written.

9.3.1.9 SCR Idle Line Interrupt Enable (ILIE) Bit 10

When ILIE is set, the SCI interrupt occurs when IDLE is set. When ILIE is cleared, the IDLE interrupt is disabled. ILIE is cleared by hardware and software reset.
An internal flag, the shift register idle interrupt (SRIINT) flag, is the interrupt request to the interrupt controller. SRIINT is not directly accessible to the user.
When a valid start bit has been received, an idle interrupt will be generated if both IDLE (SCI Status Register bit 3) and ILIE equals one. The idle interrupt acknowledge from the interrupt controller clears this interrupt request. The idle interrupt will not be asserted again until at least one character has been received. The result is as follows:

1. The IDLE bit shows the real status of the receive line at all times.
2. Idle interrupt is generated once for each idle state, no matter how long the idle state lasts.

9.3.1.10 SCR SCI Receive Interrupt Enable (RIE) Bit 11

The RIE bit is used to enable the SCI receive data interrupt. If RIE is cleared, receive data interrupts are disabled, and the RDRF bit in the SCI status register must be polled to determine if the receive data register is full. If both RIE and RDRF are set, the SCI will request an SCI receive data interrupt from the interrupt controller.
Receive interrupts with exception have higher priority than normal receive data interrupts, therefore if exception occurs (PE, FE or OR are set) and REIE is set the SCI will request an SCI receive data with exception interrupt from the interrupt controller. RIE is cleared by hardware and software reset.

9.3.1.11 SCR SCI Transmit Interrupt Enable (TIE) Bit 12

The TIE bit is used to enable the SCI transmit data interrupt. If TIE is cleared, transmit data interrupts are disabled, and the transmit data register empty (TDRE) bit in the SCI status register must be polled to determine if the transmit data register is empty. If both TIE and TDRE are set, the SCI will request an SCI transmit data interrupt from the interrupt controller. TIE is cleared by hardware and software reset.

9.3.1.12 SCR Timer Interrupt Enable (TMIE) Bit 13

The TMIE bit is used to enable the SCI timer interrupt. If TMIE is set (enabled), the timer interrupt requests will be made to the interrupt controller at the rate set by the SCI clock register. The timer interrupt is automatically cleared by the interrupt acknowledge from the interrupt controller. This feature allows DSP programmers to use the SCI baud rate generator as a simple periodic interrupt generator if the SCI is not in use, if external clocks are used for the SCI, or if periodic interrupts are needed at the SCI baud rate. The SCI internal clock is divided by 16 (to match the $1 \times$ SCI baud rate) for timer interrupt generation. This timer does not require that any SCI pins be configured for SCI use to operate. TMIE is cleared by hardware and software reset.

9.3.1.13 SCI Timer Interrupt Rate (STIR) Bit 14

This bit controls a divide by 32 in the SCI Timer interrupt generator. When this bit is cleared, the divide by 32 is inserted in the chain. When the bit is set, the divide by 32 is bypassed. This bit is cleared by hardware and software reset. To ensure proper operation of the timer, STIR may not be change during timer operation (TMIE=1).

9.3.1.14 SCR SCI Clock Polarity (SCKP) Bit 15

The clock polarity, sourced or received on the clock pin (SCLK), can be inverted using this bit, eliminating the need for an external inverter. When SCKP equals zero, the clock polarity is positive; when SCKP equals one, the clock polarity is negative. In the synchronous mode, positive polarity means that the clock is normally positive and

transitions negative during data valid; whereas, negative polarity means that the clock is normally negative and transitions positive during valid data. In the asynchronous mode, positive polarity means that the rising edge of the clock occurs in the center of the period that data is valid; negative polarity means that the falling edge of the clock occurs during the center of the period that data is valid. SCKP is cleared on hardware and software reset.

9.3.1.15 SCR SCI Receive with Exception Interrupt Enable (REIE) Bit 16

The REIE bit is used to enable the SCI receive data with exception interrupt. If REIE is cleared, receive data with exception interrupt is disabled. If both REIE and RDRF are set, and PE, FE, OR are not all zero the SCI will request an SCI receive data with exception interrupt from the interrupt controller. REIE is cleared by hardware and software reset.

9.3.2 SCI Status Register (SSR)

The SSR is an 8-bit read-only register used by the DSP CPU to determine the status of the SCI. When the SSR is read onto the internal data bus, the register contents occupy the low-order byte of the data bus and all high-order portions are zero filled. The status bits are described in the following paragraphs.

9.3.2.1 SSR Transmitter Empty (TRNE) Bit 0

The TRNE flag is set when both the transmit shift register and data register are empty to indicate that there is no data in the transmitter. When TRNE is set, data written to one of the three STX locations or to the STXA will be transferred to the transmit shift register and be the first data transmitted. TRNE is cleared when TDRE is cleared by writing data into the transmit data register (STX) or the transmit data address register (STXA), or when an idle, preamble, or break is transmitted. The purpose of this bit is to indicate that the transmitter is empty; therefore, the data written to STX or STXA will be transmitted next – i.e., there is no word in the transmit shift register presently being transmitted. This procedure is useful when initiating the transfer of a message (i.e., a string of characters). TRNE is set by the hardware, software, SCI individual, and stop reset.

9.3.2.2 SSR Transmit Data Register Empty (TDRE) Bit 1

The TDRE bit is set when the SCI transmit data register is empty. When TDRE is set, new data may be written to one of the SCI transmit data registers (STX) or transmit data address register (STXA). TDRE is cleared when the SCI transmit data register is written. TDRE is set by the hardware, software, SCI individual, and stop reset.
In the SCI synchronous mode, when using the internal SCI clock, there is a delay of up to 5.5 serial clock cycles between the time that STX is written until TDRE is set, indicating the data has been transferred from the STX to the transmit shift register. There is a two to four serial clock cycle delay between writing STX and loading the transmit shift register; in addition, TDRE is set in the middle of transmitting the second bit. When using an external serial transmit clock, if the clock stops, the SCI transmitter stops. TDRE will not be set until the middle of the second bit transmitted after the external clock starts. Gating the external clock off after the first bit has been transmitted will delay TDRE indefinitely.

In the SCI asynchronous mode, the TDRE flag is not set immediately after a word is transferred from the STX or STXA to the transmit shift register nor when the word first begins to be shifted out. TDRE is set two cycles of the 16 × clock after the start bit – i.e., two 16 × clock cycles into to transmission time of the first data bit.

9.3.2.3   SSR Receive Data Register Full (RDRF) Bit 2

The RDRF bit is set when a character is transferred to the SCI receive data register from the SCI receive shift register (regardless of the error bits condition). RDRF is cleared when the SCI receive data register is read or by the hardware, software, SCI individual, and stop reset.

9.3.2.4   SSR Idle Line Flag (IDLE) Bit 3

IDLE is set when 10 (or 11) consecutive ones are received. IDLE is cleared by a start-bit detection. The IDLE status bit represents the status of the receive line. The transition of IDLE from zero to one can cause an IDLE interrupt (ILIE). IDLE is cleared by the hardware, software, SCI individual, and stop reset.

9.3.2.5   SSR Overrun Error Flag (OR) Bit 4

The OR flag is set when a byte is ready to be transferred from the receive shift register to the receive data register (SRX) that is already full (RDRF=1). The receive shift register data is not transferred to the SRX. The OR flag indicates that character(s) in the receive data stream may have been lost. The only valid data is located in the SRX. OR is cleared when the SCI status register is read, followed by a read of SRX. The OR bit clears the FE and PE bits – i.e., overrun error has higher priority than FE or PE. OR is cleared by the hardware, software, SCI individual, and stop reset.

9.3.2.6   SSR Parity Error (PE) Bit 5

In the 11-bit asynchronous modes, the PE bit is set when an incorrect parity bit has been detected in the received character. It is set simultaneously with RDRF for the byte which contains the parity error – i.e., when the received word is transferred to the SRX. If PE is set, it does not inhibit further data transfer into the SRX. PE is cleared when the SCI status register is read, followed by a read of SRX. PE is also cleared by the hardware, software, SCI individual, or stop reset. In the 10-bit asynchronous mode, the 11-bit multidrop mode, and the 8-bit synchronous mode, the PE bit is always cleared since there is no parity bit in these modes. If the byte received causes both parity and overrun errors, the SCI receiver will only recognize the overrun error.

9.3.2.7   SSR Framing Error Flag (FE) Bit 6

The FE bit is set in the asynchronous modes when no stop bit is detected in the data string received. FE and RDRE are set simultaneously – i.e., when the received word is transferred to the SRX. However, the FE flag inhibits further transfer of data into the SRX until it is cleared. FE is cleared when the SCI status register is read followed by reading the SRX. The hardware, software, SCI individual, and stop reset also clear FE. In the 8-bit

synchronous mode, FE is always cleared. If the byte received causes both framing and overrun errors, the SCI receiver will only recognize the overrun error.

9.3.2.8   SSR Received Bit 8 (R8) Address Bit 7

In the 11-bit asynchronous multidrop mode, the R8 bit is used to indicate whether the received byte is an address or data. R8 is not affected by reading the SRX or status register. The hardware, software, SCI individual, and stop reset clear R8.

9.3.3   SCI Clock Control Register (SCCR)

The SCCR is a 16-bit read/write register, which controls the selection of the clock modes and baud rates for the transmit and receive sections of the SCI interface. The control bits are described in the following paragraphs. The SCCR is cleared by hardware reset. The basic points of the clock generator are as follows:

1. The SCI logic always uses a 16 $\times$ internal clock in the asynchronous modes and always uses a 2 $\times$ internal clock in the synchronous mode. The maximum internal clock available to the SCI peripheral block is the oscillator frequency divided by 4. With a 66-MHz DSP56301 processor, this gives a maximum data rate of 1031.25 Kbps for asynchronous data and 8.25 Mbps for synchronous data. These maximum rates are the same for internally or externally supplied clocks.
2. The 16 $\times$ clock is necessary for the asynchronous modes to synchronize the SCI to the incoming data.
3. For the asynchronous modes, the user must provide a 16 $\times$ clock if he wishes to use an external baud rate generator (i.e., SCLK input).
4. For the asynchronous modes, the user may select either 1 $\times$ or 16 $\times$ for the output clock when using internal TX and RX clocks (TCM=0 and RCM=0).
5. The transmit data on the TXD pin changes on the negative edge of the 1$\times$ serial clock and is stable on the positive edge (SCKP=0). For SCKP equals one, the data changes on the positive edge and is stable on the negative edge.
6. The receive data on the RXD pin is sampled on the positive edge (if SCKP=0) or on the negative edge (if SCKP=1) of the 1 $\times$ serial clock.
7. For the asynchronous mode, the output clock is continuous.
8. For the synchronous mode, a 1 $\times$ clock is used for the output or input baud rate. The maximum 1 $\times$ clock is the crystal frequency divided by 8.
9. For the synchronous mode, the clock is gated.
10. For the synchronous mode, the transmitter and receiver are synchronous with each other.

9.3.3.1   SCCR Clock Divider (CD11–CD0) Bits 11–0

The CD11–CD0 bits specify the divide ratio of the prescale divider in the SCI clock generator. A divide ratio from 1 to 4096 (CD11–CD0=$000 0 to $FFF) may be selected.

Hardware and software reset clear CD11–CD0.

9.3.3.2 SCCR Clock Out Divider (COD) Bit 12

The clock output divider is controlled by COD and the SCI mode. If the SCI mode is synchronous, the output divider is fixed at divide by 2; if the SCI mode is asynchronous, and

1. If COD equals zero and SCLK is an output (i.e., TCM and RCM=0), the SCI clock is divided by 16 before being output to the SCLK pin; thus, the SCLK output is a $1 \times$ clock.
2. If COD equals one and SCLK is an output, the SCI clock is fed directly out to the SCLK pin; thus, the SCLK output is a $16 \times$ baud clock.

The COD bit is cleared by hardware and software reset.

9.3.3.3 SCCR SCI Clock Prescaler (SCP) Bit 13

The SCI SCP bit selects a divide by 1 (SCP=0) or divide by 8 (SCP=1) prescaler for the clock divider. The output of the prescaler is further divided by 2 to form the SCI clock. Hardware and software reset clear SCP.

9.3.3.4 SCCR Receive Clock Mode Source Bit (RCM) Bit 14

RCM selects internal or external clock for the receive. RCM when equals zero selects the internal clock; RCM when equals one selects the external clock from the SCLK pin. Hardware and software reset clear RCM.

9.3.3.5 SCCR Transmit Clock Source Bit (TCM) Bit 15

The TCM bit selects internal or external clock for the transmitter. TCM equals zero selects the internal clock; TCM equals one selects the external clock from the SCLK pin. Hardware and software reset clear TCM.

9.3.4 SCI Data Registers

The SCI data registers are divided into two groups: receive and transmit. There are two receive registers – a receive data register (SRX) and a serial-to-parallel receive shift register. There are also two transmit registers – a transmit data register (called either STX or STXA) and a parallel-to-serial transmit shift register.

9.3.4.1 SCI Receive Registers

Data bits received on the RXD pin are shifted into the SCI receive shift register. When a complete word has been received, the data portion of the word is transferred to the byte-wide SRX. This process converts the serial data to parallel data and provides double buffering. Double buffering provides flexibility to the programmer and increased throughput since the programmer can save (and process) the previous word while the current word is being received.

Figure 9-2. SCI Programming Model - Data Registers


```
23          16 15        8 7          0
┌─────────────┬──────────┬──────────┐
│    SRX      │          │          │   SCI RECIEVE DATA REGISTER HIGH (READ ONLY)
├─────────────┼──────────┼──────────┤
│             │   SRX    │          │   SCI RECIEVE DATA REGISTER MID (READ ONLY)
├─────────────┼──────────┼──────────┤
│             │          │   SRX    │   SCI RECEIVE DATA REGISTER LOW (READ ONLY)
└─────────────┴──────────┴──────────┘
```

RXD → [SCI RECEIVE DATA SHIFT REGISTER]

NOTE: SRX is the same register decoded at three different addresses.

(a) Receive Data Register

```
23          16 15        8 7          0
┌─────────────┬──────────┬──────────┐
│    STX      │          │          │   SCI TRANSMIT DATA REGISTER HIG (WRITE ONLY)
├─────────────┼──────────┼──────────┤
│             │   STX    │          │   SCI TRANSMIT DATA REGISTER MID (WRITE ONLY)
├─────────────┼──────────┼──────────┤
│             │          │   STX    │   SCI TRANSMIT DATA REGISTER LOW (WRITE ONLY)
└─────────────┴──────────┴──────────┘
```

SCI TRANSMIT DATA SHIFT REGISTER [ ] → TXD

```
23          16 15        8 7          0
┌─────────────┬──────────┬──────────┐
│             │          │   STXA   │   SCI TRANSMITDATA ADDRESS REGISTER
└─────────────┴──────────┴──────────┘       (WRITE ONLY)
```

NOTES:
1. Bytes are masked on the fly.
2. STX is the same register decoded at three different addresses.

(b) Transmit Data Register

The SRX can be read at three locations as SRXL, SRXM and SRXH: When SRXL is read, the contents of the SRX are placed in the lower byte of the data bus and the remaining bits on the data bus are read as zeros. Similarly, when SRXM is read, the contents of SRX are placed in the middle byte of the bus, and when SRXH is read, the contents of SRX are placed in the high byte with the remaining bits are read as zeros. Mapping SRX as described allows three bytes to be efficiently packed into one 24-bit word by ORing three data bytes read from the three addresses.

The length and format of the serial word is defined by the WDS0, WDS1, and WDS2 control bits in the SCI control register. In the synchronous modes, the start bit, the eight data bits with LSB first, the address/data indicator bit and/or the parity bit, and the stop bit are received in that order if SSFTD equals zero. If SSFTD equals one, the data bits are transmitted MSB first. The clock source is defined by the receive clock mode (RCM) select bit in the SCR. In the synchronous mode, the synchronization is provided by gating the clock. In either mode, when a complete word has been clocked in, the contents of the shift register can be transferred to the SRX and the flags; RDRF, FE, PE, and OR are changed appropriately. Because the operation of the SCI receive shift register is transparent to the DSP, the contents of this register are not directly accessible to the programmer.

9.3.4.2   SCI Transmit Registers

The transmit data register is a one byte-wide register mapped into four addresses as STXL, STXM, STXH and STXA. In the asynchronous mode, when data is to be transmitted, STXL, STXM and STXH are used. When STXL is written, the low byte on the data bus is transferred to the STX; when STXM is written, the middle byte is transferred to the STX; and when STXH is written, the high byte is transferred to the STX. This structure makes it easy for the programmer to unpack the bytes in a 24-bit word for transmission. TDXA should be written in the 11-bit asynchronous multidrop mode when the data is an address and it is desired that the ninth bit (the address bit) be set. When STXA is written, the data from the low byte on the data bus is stored in the register. The address data bit will be cleared in the 11-bit asynchronous multidrop mode when any of STXL, STXM or STXH is written. When either STX (STXL,STXM or STXH) or STXA is written, TDRE is cleared.

The transfer from either STX or STXA to the transmit shift register occurs automatically, but not immediately, when the last bit from the previous word has been shifted out – i.e., the transmit shift register is empty. Like the receiver, the transmitter is double buffered. However, there will be a two to four serial clock cycle delay between when the data is transferred from either STX or STXA to the transmit shift register and when the first bit appears on the TXD pin. (A serial clock cycle is the time required to transmit one data bit). The transmit shift register is not directly addressable, and a dedicated flag for this register does not exist. Because of this fact and the two to four cycle delay, two bytes cannot be written consecutively to STX or STXA without polling, as the second byte might overwrite the first byte. The TDRE flag should always be polled prior to writing STX or STXA to prevent overruns unless transmit interrupts have been enabled. Either STX or STXA is usually written as part of the interrupt service routine. Of course, the interrupt will only be generated if TDRE equals one. The transmit shift register is indirectly visible via the TRNE bit in the SSR.

In the synchronous modes, data is synchronized with the transmit clock, which may have either an internal or external source as defined by the TCM bit in the SCCR. The length and format of the serial word is defined by the WDS0, WDS1, and WDS2 control bits in the SCR. In the asynchronous modes, the start bit, the eight data bits (with the LSB first if SSFTD=0 and the MSB first if SSFTD=1), the address/data indicator bit or parity bit, and the stop bit are transmitted in that order.

The data to be transmitted can be written to any one of the three STX addresses. If SCKP equals one and SSHTD equals one, the SCI synchronous mode is equivalent to the SSI operation in the 8-bit data on-demand mode.

9.3.5   Preamble, Break, and Data Transmission Priority

It is possible that two or three transmission commands are set simultaneously:

1.   A preamble (TE was toggled).
2.   A break (SBK was set or was toggled).
3.   There is data for transmission (TDRE=0).

After the current character transmission, if two or more of these commands are set, the transmitter will execute them in the following order:

1.   Preamble
2.   Break
3.   Data

9.4   OPERATING MODES

9.4.1   Register Contents After Reset

There are four different methods of resetting the SCI.

1) Hardware and 2) software resets clear the port control register bits, which configure all I/O as general-purpose input. The SCI will remain in the reset state while all SCI pins are programmed as general-purpose I/O (CC2, CC1, and CC0=0); the SCI will become active only when at least one of the SCI I/O pins is programmed as not general-purpose I/O.

3) During program execution, the CC2, CC1, and CC0 bits may be cleared (individual reset), which will cause the SCI to stop serial activity and enter the reset state. All SCI status bits will be set to their reset state; however, the contents of the interface control register are not affected, allowing the DSP program to reset the SCI separately from the other internal peripherals. During individual reset internal DMA accesses to the data registers of the SCI are not valid and data read will be unexpected.

4) Executing the STOP instruction halts operation of the SCI until the DSP is restarted, causing the SSR to be reset. No other SCI registers are affected by the STOP instruction. Table 9-2. illustrates how each type of reset affects each register in the SCI.

Table 9-2. SCI Registers after Reset

Register Bit	Bit Mnemonic	Bit Number	Reset Type			
			HW Reset	SW Reset	IR Reset	ST Reset
SCR	REIE	16	0	0	–	–
	SCKP	15	0	0	–	–
	TMIE	14	0	0	–	–
	TMIE	13	0	0	–	–
	TIE	12	0	0	–	–
	RIE	11	0	0	–	–
	ILIE	10	0	0	–	–
	TE	9	0	0	–	–
	RE	8	0	0	–	–
	WOMS	7	0	0	–	–
	RWU	6	0	0	–	–
	WAKE	5	0	0	–	–
	SBK	4	0	0	–	–
	SSFTD	3	0	0	–	–
	WDS (2–0)	2–0	0	0	–	–
SSR	R8	7	0	0	0	0
	FE	6	0	0	0	0
	PE	5	0	0	0	0
	OR	4	0	0	0	0
	IDLE	3	0	0	0	0
	RDRF	2	0	0	0	0
	TDRE	1	1	1	1	1
	TRNE	0	1	1	1	1
SCCR	TCM	15	0	0	–	–
	RCM	14	0	0	–	–
	SCP	13	0	0	–	–
	COD	12	0	0	–	–
	CD (11–0)	11–0	0	0	–	–
SRX	SRX (23–0)	23–16, 15–8, 7–0	–	–	–	–
STX	STX (23–0)	23–0	–	–	–	–
SRSH	SRS (8–0)	8–0	–	–	–	–
STSH	STS (8–0)	8–0	–	–	–	–

NOTES:

SRSH – SCI receive shift register, STSH – SCI transmit shift register
HW – Hardware reset is caused by asserting the external RESET pin.
SW – Software reset is caused by executing the RESET instruction.
IR – Individual reset is caused by clearing PCRE (bits 0–2) (configured for general-purpose I/O).
ST – Stop reset is caused by executing the STOP instruction.
1 – The bit is set during the xx reset.
0 – The bit is cleared during the xx reset.
– – The bit is not changed during the xx reset

9.4.2 SCI Initialization

The correct way to initialize the SCI is as follows:

1. Hardware or software reset.
2. Program SCI control registers.
3. Configure SCI pins (at least one) as not general-purpose I/O.

If interrupts are to be used, the pins must be selected, and interrupts must be enabled and unmasked before the SCI will operate. The order does not matter; any one of these three requirements for interrupts can be used to finally enable the SCI. Figure 9-1 shows the meaning of the individual bits in the SCR and SCCR. Synchronous applications usually require exact frequencies, which require that the crystal frequency be chosen carefully. An alternative to selecting the system clock to accommodate the SCI requirements is to provide an external clock to the SCI.

9.4.3 SCI Exceptions

The SCI can cause five different exceptions in the DSP. These exceptions are as follows (ordered from the highest to the lowest priority):

1. SCI Receive Data with Exception Status – caused by receive data register full with a receiver error (parity, framing, or overrun error). Clearing the pending interrupt is done by reading the SCI status register, followed by a read of SRX. A long interrupt service routine should be used to handle the error condition. This interrupt is enabled by SCR bit 16 (REIE).
2. SCI Receive Data – caused by receive data register full. Reading SRX clears the pending interrupt. This error-free interrupt may use a fast interrupt service routine for minimum overhead. This interrupt is enabled by SCR bit 11 (RIE).
3. SCI Transmit Data – caused by transmit data register empty. Writing STX clears the pending interrupt. This error-free interrupt may use a fast interrupt service routine for minimum overhead. This interrupt is enabled by SCR bit 12 (TIE).
4. SCI Idle Line – caused by the receive line entering the idle state (10 or 11 bits of ones). This interrupt is latched and then automatically reset when the interrupt is accepted. This interrupt is enabled by SCR bit 10 (ILIE).
5. SCI Timer – caused by the baud rate counter reaching zero. This interrupt is automatically reset when the interrupt is accepted. This interrupt is enabled by SCR bit 13 (TMIE).

9.5 GPIO - pins and registers

The GPIO functionality of port SCI is controlled by three registers: Port Control register (PCR), Port Direction Register (PRR) and Port Data register (PDR).

9.5.1 Port Control Register (PCR)

The read/write 16 bit Port Control Register controls the functionality of SCI GPIO pins. When the PCR is read to the internal data bus, the register contents occupy the two low-order bytes of the data bus, and the remaining bits are read as zeros. When written the most significant byte should be written zero for future compatibility.

Figure 9-3. Port Control Register (PCR)
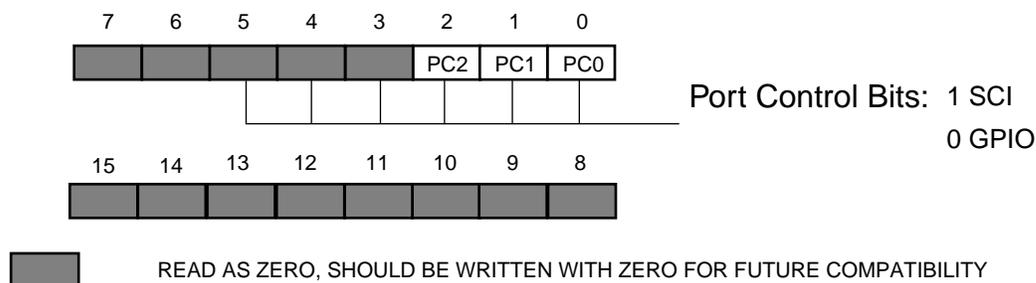
Each of PC(2:0) bits controls the functionality of the corresponding port pin. When a PC[i] bit is set, the corresponding port pin is configured as a SCI pin. When a PC[i] bit is cleared, the corresponding port pin is configured as GPIO pin.

Hardware and software reset clear all PCR bits.

9.5.2 Port Direction Register (PRR)

The read/write 16 bit Port Direction Register controls the direction of SCI GPIO pins. When the PRR is read to the internal data bus, the register contents occupy the two low-order bytes of the data bus, and the remaining bits are read as zeros. When written the most significant byte should be written zero for future compatibility.

Figure 9-4. Port Direction Register (PRR)
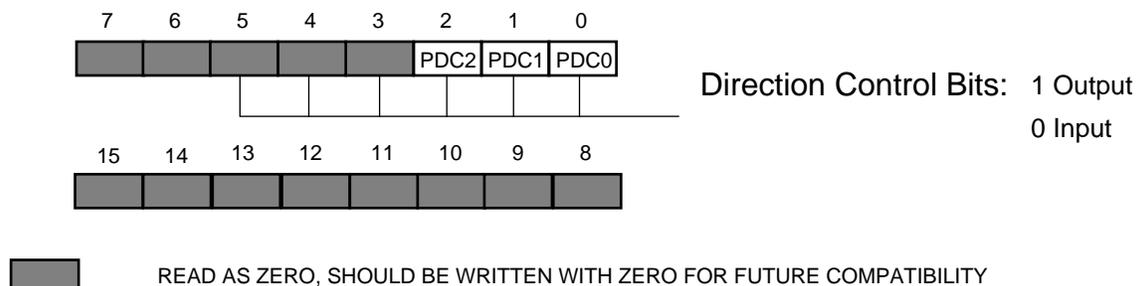
When port pin[i] is configured as GPIO, PDC[i] controls the port pin direction. When PDC[i] is set, the GPIO port pin[i] is configured as output. When PDC[i] is cleared the GPIO port pin[i] is configured as input.

Hardware and software reset clear all PRR bits.

The following table describe the port pin configurations.

Table 9-3. Port Control Register and Port Direction Register bits functionality

PC[i]	PDC[i]	Port Pin[i] Function
1	X	SCI
0	0	GPIO input
0	1	GPIO output

9.5.3   Port Data register (PDR)

The read/write 16 bit Port Data Register is used to read or write data to/from SCI GPIO pins.Bits PD(2:0) are used to read or write data from/to the corresponding port pins if they are configured as GPIO (by PC(2:0) bits in PCR). If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit will reflect the value present on this pin. If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit will be reflected on the this pin.When the PDR is read to the internal data bus, the register contents occupy the two low-order bytes of the data bus, and the remaining bits are read as zeros. When written the most significant byte should be written zero for future compatibility.

Figure 9-5. Port Data Register (PDR)

7	6	5	4	3	2	1	0
					PD2	PD1	PD0

15	14	13	12	11	10	9	8

☐ READ AS ZERO, SHOULD BE WRITTEN WITH ZERO FOR FUTURE COMPATIBILITY

Hardware and software reset clear all PDR bits.

Appendix A PROGRAMS

A-1   BOOTSTRAP PROGRAM

```
; BOOTSTRAP CODE FOR DSP56301 - (C) Copyright 1994 Motorola Inc.
; Revised April, 17 1995.
;
; Bootstrap through the Host Interface, External EPROM or SCI.
;
; This is the Bootstrap program contained in the DSP56301 192-word Boot
; ROM. This program can load any program RAM segment from an external
; EPROM, from the Host Interface or from the SCI serial interface.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MC:MB:MA=000, then the Boot ROM is bypassed and the DSP56301 will start
; fetching instructions beginning with the address $C00000 assuming that
; an external memory of SRAM type is used. The accesses will be performed
; using 31 wait states with no address attributes selected (default area).
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MC:MB:MA=001, then it loads a program RAM segment from consecutive
; byte-wide P memory locations, starting at P:$D00000 (bits 7-0).
; The memory is selected by the Address Attribute AA1 and is accessed with
; 31 wait states.
; The EPROM bootstrap code expects to read 3 bytes
; specifying the number of program words, 3 bytes specifying the address
; to start loading the program words and then 3 bytes for each program
; word to be loaded. The number of words, the starting address and the
; program words are read least significant byte first followed by the
; mid and then by the most significant byte.
; The program words  will be condensed into 24-bit words and stored in
; contiguous PRAM memory locations starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MC:MB:MA=010, then it loads the program RAM from the SCI interface.
; The number of program words to be loaded and the starting address must
; be specified. The SCI bootstrap code expects to receive 3 bytes
; specifying the number of program words, 3 bytes specifying the address
; to start loading the program words and then 3 bytes for each program
; word to be loaded. The number of words, the starting address and the
; program words are received least significant byte first followed by the
; mid and then by the most significant byte. After receiving the
```

```
; program words, program execution starts in the same address where
; loading started. The SCI is programmed to work in asynchronous mode
; with 8 data bits, 1 stop bit and no parity. The clock source is
; external and the clock frequency must be 16x the baud rate.
; After each byte is received, it is echoed back through the SCI
; transmitter.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MC:MB:MA=100, then it loads the program RAM from the Host
; Interface programmed to operate in the PCI target (slave) mode.
; The HOST PCI bootstrap code expects to read a 32-bit word
; specifying the number of program words, a 32-bit word specifying the address
; to start loading the program words and then a 32-bit word for each program
; word to be loaded. The program words  will be stored in
; contiguous PRAM memory locations starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
; The Host Interface bootstrap load program may be stopped by
; setting the Host Flag 0 (HF0). This will start execution of the loaded
; program from the specified starting address.
;
; The HOST Processor must first configure the Host Interface as PCI slave
; and then start writing data to the Host Interface. The HOST Processor must
; program the HCTR HTF1-HTF0 bits with one of the combinations: 01, 10, 11.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MC:MB:MA=101, then it loads the program RAM from the Host
; Interface programmed to operate in the ISA (slave) mode.
; Using self configuration mode, the base address in CBMA is initially written
; with $2f which corresponds to an ISA HTXR address of $2fe (Serial Port 2 Modem
; Status read only register).
;
; The HOST ISA  bootstrap code expects to read 32 consecutive times the "magic
; number" $0037. Subsequently the bootstrap code expects to read a 16-bit word
; which is the designated ISA Port Address; this address is written into the
; CBMA. The HOST Processor must poll for the Host Interface to be reconfigured.
; This must be done by reading the HSTR and verifying that the value $0013 is
; read. From this moment the HOST Processor may start writing data to the
; Host Interface.
; The HOST ISA  bootstrap code expects to read a 24-bit word
; specifying the number of program words, a 24-bit word specifying the address
; to start loading the program words and then a 24-bit word for each program
; word to be loaded. The program words  will be stored in
; contiguous PRAM memory locations starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
; The Host Interface bootstrap load program may be stopped by
; setting the Host Flag 0 (HF0). This will start execution of the loaded
; program from the specified starting address.
```

```
; Note that this ISA connection implies 16 bit data width access only and
; that the number of 16-bit wide words that are transferred must be even.
;
; The code to be loaded into 56301 is stored in the HOST Processor 16-bit
; wide memory in the following format:
;
;    | M0  | L0  |
;    | L1  | H0  |
;    | H1  | M1  |
;    | M2  | L2  |
;
; The boot program will convert every three 16-bit wide host words to two 24-bit
; wide 56301 opcodes in the following format:
;
;    | H0  | M0  | L0  |
;    | H1  | M1  | L1  |
;
; Required 56301 to ISA connection:
;
;   HA[10]  <- SBHE_             ; selects HI32 (base address 10011111)
;   HA[9]   <- SA[0]            ; selects HI32 (base address 10011111)
;   HA[8:3] <- SA[9:4]         ; selects HI32 (base address 10011111)
;   HA[2:0] <- SA[3:1]         ; selects HTXR registers
;   HD[15:0] - SD[15:0]        ; Data bus
;   HD[23:16] - Not connected  ; High Data Bus - Should be pulled up or down
;   HDBEN_  -> OE_              ; Output enable of transcievers
;   HDBDR   -> DIR             ; Direction of transcievers
;   HSAK_   -> IO16_           ; 16 bit data word
;   HBS_    <- Vcc             ; Bus Strobe disabled
;   HAEN    <- AEN             ; DMA cycle enable
;   HTA     -> CHRDY           ; Channel ready
;   HWR_    <- IOWC_           ; IO/DMA write strobe
;   HRD_    <- IORC_           ; IO/DMA read strobe
;   HRST    <- inverted RSTDRV ; invert ISA reset
;
;
; NOTE: the Host Processor must program the Host Interface to operate in the zero
;       fill mode (HTF1-HTF0 = 01 in HCTR).
;
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MC:MB:MA=110, then it loads the program RAM from the Host
; Interface programmed to operate in the Universal Bus (UB) mode,
; in double-strob pin configuration.
; The HOST UB bootstrap code expects accesses that are byte wide.
; The HOST UB bootstrap code expects to read 3 bytes forming a 24-bit word
; specifying the number of program words, 3 bytes forming a 24-bit word
; specifying the address to start loading the program words and then 3 bytes
; forming 24-bit words for each program word to be loaded.
; The program words  will be stored in contiguous PRAM memory locations
; starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
```

```
; The Host Interface bootstrap load program may be stopped by setting the
; Host Flag 0 (HF0). This will start execution of the loaded program from
; the specified starting address.
;
; The user must externally decode the port address with active low logic and
; connect the select line to HAEN; all the address lines shall be pulled down
; except for A3, A2 and A1 that select the HOST Interface registers.
;
; When booting through the Host Interface it is recommended that the Host
; boot program will verify that the Host Interface is operational, by
; reading the status register (HSTR) and confirm that TRDY=1.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MC:MB:MA=111, then it loads the program RAM from the Host
; Interface programmed to operate in the Universal Bus (UB) mode,
; in single-strob pin configuration.
; Other than the single-strob pin configuration, this mode is identical to
; the double-strob pin configuration UB mode (MC:MB:MA=110).

BOOT    equ     $D00000         ; this is the location in P memory
                                ; on the external memory bus
                                ; where the external byte-wide
                                ; EPROM would be located
AARV    equ     $D00409         ; AAR1 selects the EPROM as CE~
                                ; mapped as P from $D00000 to
                                ; $DFFFFF, active low


M_SSR   EQU     $FFFF93         ; SCI Status Register
M_STXL  EQU     $FFFF95         ; SCI Transmit Data Register (low)
M_SRXL  EQU     $FFFF98         ; SCI Receive Data Register (low)
M_SCCR  EQU     $FFFF9B         ; SCI Clock Control Register
M_SCR   EQU     $FFFF9C         ; SCI Control Register
M_PCRE  EQU     $FFFF9F         ; Port E Control register
M_DCTR  EQU     $FFFFC5         ; DSP CONTROL REGISTER (DCTR)
M_DPMC  EQU     $FFFFC7         ; DSP PCI MASTER CONTROL REGISTER (DPMC)
M_DPAR  EQU     $FFFFC8         ; DSP PCI ADDRESS REGISTER (DPAR)
M_DSR   EQU     $FFFFC9         ; DSP STATUS REGISTER (DSR)
M_DRXR  EQU     $FFFFCB         ; DSP RECEIVE DATA FIFO (DRXR)
M_AAR1  EQU     $FFFFF8         ; Address Attribute Register 1


        ORG PL:$ff0000,PL:$ff0000    ; bootstrap code starts at $ff0000


START
        clr a #$0a,X0           ; clear a and load X0 with constant 0a0000
        jclr #2,omr,EPRSCILD    ; If MC:MB:MA=0xx, go load from EPROM/SCI
        jclr #1,omr,IHOSTLD     ; If MC:MB:MA=10x, go load from PCI/ISA HOST
        jclr #0,omr,UB2HOSTLD   ; If MC:MB:MA=110, go load from double-strob UB
Host
                                ; If MC:MB:MA=111, go load from single-strob UB Host
```

```
;========================================================================
; This is the routine that loads from the Host Interface in UB (UNIVERSAL) mode,
; with single-strob pin configuration (RD/WR,DS).
; MC:MB:MA=111 - Host UB

UB1HOSTLD
; Activate the UB mode
        movep #$3e2000,X:M_DCTR ; HM=$3 (UB)
                                ; HIRD=1 (HIRQ_ pin - drive high enabled)
                                ; HIRH=1 (HIRQ_ pin - handshake enabled)
                                ; HRSP=1 (HRST pin - active low)
                                ; HTAP=0 (HTA  pin - active high)
                                ; HDSM=1 (Double-strob pin mode disabled)

        bra <UB_CONT            ; Continue as in double-strob mode


;========================================================================
; This is the routine that loads from the Host Interface in UB (UNIVERSAL) mode,
; with double-strob pin configuration (RD,WR).
; MC:MB:MA=110 - Host UB

UB2HOSTLD
; Activate the UB mode
        movep #$3e0000,X:M_DCTR ; HM=$3 (UB)
                                ; HIRD=1 (HIRQ_ pin - drive high enabled)
                                ; HIRH=1 (HIRQ_ pin - handshake enabled)
                                ; HRSP=1 (HRST pin - active low)
                                ; HTAP=0 (HTA  pin - active high)
                                ; HDSM=0 (Double-strob pin mode enabled)


UB_CONT
        do #6,_LOOP0            ; read # of words and start address
        jclr #2,X:M_DSR,*       ; Wait for SRRQ to go high (i.e. data ready)
        movep X:M_DRXR,a2       ;
        asr #8,a,a              ; Shift 8 bit data into A1
_LOOP0                         ;
        move a1,r0             ; starting address for load
        move a1,r1             ; save it in r1
                               ; a0 holds the number of words
; Download P memory through UB


        do a0,_LOOP1           ; Load instruction words
        do #3,_LOOP2           ; for each byte
_LBLA
        jset #2,X:M_DSR,_LBLB  ; Wait for SRRQ to go high (i.e. data ready)
        jclr #3,X:M_DSR,_LBLA  ; If HF0=1, stop loading new data.
        enddo                  ; Must terminate the do loop
        enddo                  ; Must terminate the do loop
        bra <_LOOP1
_LBLB
        movep X:M_DRXR,a2      ; Store 16-bit data in accumulator
        asr #8,a,a            ; Shift 8 bit data into A1
_LOOP2                        ; and go get another 24-bit word.
```

```
        move a1,p:(r0)+           ; Store 24-bit data in P mem.
_LOOP1                            ; and go get another 24-bit word.
        bra <FINISH               ; finish bootstrap


;========================================================================
IHOSTLD
        jclr #0,omr,PCIHOSTLD    ; If MC:MB:MA=100, go load from PCI HOST


;========================================================================
; This routine loads from the Host Interface in ISA (UNIVERSAL) mode.
; MC:MB:MA=101 - Host ISA

; Using self configuration mode, the base address in CBMA is written with
; $2f which corresponds to an ISA HTXR address of $2fe (Serial Port 2 Modem
; Status read only register).

ISAHOSTLD
        movep #$5a0000,X:M_DCTR ; Configure HI32 as Self-Config
        movep #$00002f,X:M_DPMC ; write to DPMC
        rep #4
        movep X0,X:M_DPAR       ; write to DPAR (CSTR+CCMR,CCCR+CRID,CLAT,CBMA)
                                ; completing 32 bit write
; Switch to ISA mode
        movep X0,X:M_DCTR        ; Software personal reset
        move #$010020,y1         ; width 16, offset 32
                                ; (also used as replacment to needed NOP after
sw reset!)
        movep #$3a0000,X:M_DCTR ; HM=$3 (UB)
                                ; HIRD=1 (HIRQ_ pin - drive high enabled)
                                ; HIRH=0 (HIRQ_ pin - handshake disabled)
                                ; HRSP=1 (HRST pin - active low)
                                ; HDRP=0 (HDRQ pin - active high)
                                ; HTAP=0 (HTA  pin - active high)
                                ; HDSM=0 (Data-strob pin mode enabled)

; read the "magic sequence" 32 consecutive words with value $37
_LBLC
        do #32,_LOOP3            ;
        jclr #2,X:M_DSR,*         ; Wait for SRRQ to go high (i.e. data ready)
        movep X:M_DRXR,A1        ; Store 24-bit data into A1
        and #$00ffff,A          ; Mask upper byte
        cmp #$37,A              ; Compare the 24-bit dat to $000037
        jeq _LBLD              ; If dat = $37 then go back to loop
        enddo                 ; else break the loop and retry
        bra <_LBLC
_LBLD
        nop
_LOOP3

; read new CBMA value ("ISA base address")
        jclr #2,X:M_DSR,*        ; Wait for SRRQ to go high (i.e. data ready)
```

```
        movep X:M_DRXR,A1        ; Store 24-bit data into A1


; Switch to Self Configuration mode
        movep X0,X:M_DCTR        ; Software personal reset
        movep A1,X:M_DPMC        ; write to DPMC
                                 ; (also used as replacment to needed NOP after
sw reset!)
        movep #$5a0000,X:M_DCTR ; Configure HI32 as Self-Config
        rep #4
        movep X0,X:M_DPAR        ; write to DPAR (CSTR+CCMR,CCCR+CRID,CLAT,CBMA)


; Switch to ISA mode
        movep X0,X:M_DCTR        ; Software personal reset
        move #$010010,x1         ; width 16, offset 16
                                 ; (also used as replacment to needed NOP after
sw reset!)
        movep #$3a0010,x:M_DCTR ; HM=$3 (UB)
                                 ; HIRD=1 (HIRQ_ pin - drive high enabled)
                                 ; HIRH=0 (HIRQ_ pin - handshake disabled)
                                 ; HRSP=1 (HRST pin - active low)
                                 ; HDRP=0 (HDRQ pin - active high)
                                 ; HTAP=0 (HTA  pin - active high)
                                 ; HDSM=0 (Double-strob pin mode enabled)
                                 ; HF4 =1 (turn on flag 4 for handshake)

        jclr #2,X:M_DSR,*        ; Wait for SRRQ to go high (i.e. data ready)
        movep X:M_DRXR,a0        ; Store number of words
        jclr #2,X:M_DSR,*        ; Wait for SRRQ to go high (i.e. data ready)
        movep X:M_DRXR,x0        ; Store starting address
        jclr #2,X:M_DSR,*        ; Wait for SRRQ to go high (i.e. data ready)
        movep X:M_DRXR,y0        ; Store starting address
        insert x1,x0,a           ; concatenate next 16-bit word
        insert y1,y0,a           ; concatenate next 16-bit word
        move a1,r0               ; start to p-mem
        move a0,a1               ; number of words to transfer


; Download P memory through UB
        lsr a   r0,r1            ; divide loop count by 2 and save r0

        do a1,_LOOP4             ; Load instruction words
_LBLE
        jset #2,X:M_DSR,_LBLF    ; Wait for SRRQ to go high (i.e. data ready)
        jclr #3,X:M_DSR,_LBLE    ; If HF0=1, stop loading new data.
        enddo                    ; Must terminate the do loop
        bra <_LOOP4
_LBLF
        movep X:M_DRXR,a0        ; Store 16-bit data in accumulator
_LBLG
        jset #2,X:M_DSR,_LBLH    ; Wait for SRRQ to go high (i.e. data ready)
        jclr #3,X:M_DSR,_LBLG    ; If HF0=1, stop loading new data.
        enddo                    ; Must terminate the do loop
        bra <_LOOP4
_LBLH
        movep X:M_DRXR,x0        ; Store 16-bit data in register
```

```
_LBLI
        jset #2,X:M_DSR,_LBLJ   ; Wait for SRRQ to go high (i.e. data ready)
        jclr #3,X:M_DSR,_LBLI   ; If HF0=1, stop loading new data.
        enddo                   ; Must terminate the do loop
        bra <_LOOP4
_LBLJ
        movep X:M_DRXR,y0       ; Store 16-bit data in register
        insert x1,x0,a          ; concatenate next 16-bit word
        insert y1,y0,a          ; concatenate next 16-bit word
        move a0,p:(r0)+         ; Store 24-bit data in P mem.
        move a1,p:(r0)+         ; Store 24-bit data in P mem.
_LOOP4                          ; and go get another 24-bit word.
        bra <FINISH             ; finish bootstrap

;============================================================================
; This is the routine that loads from the Host Interface in PCI mode.
; MC:MB:MA=100 - Host PCI

PCIHOSTLD
        bset #20,X:M_DCTR       ; Configure HI32 as PCI
        jclr #2,X:M_DSR,*       ; Wait for SRRQ to go high (i.e. data ready)
        movep X:M_DRXR,a0       ; Store number of words
        jclr #2,X:M_DSR,*       ; Wait for SRRQ to go high (i.e. data ready)
        movep X:M_DRXR,r0       ; Store starting address
        move r0,r1             ; save r0

        do a0,_LOOP5            ; Load instruction words
_LBLK
        jset #2,X:M_DSR,_LBLL   ; Wait for SRRQ to go high (i.e. data ready)
        jclr #3,X:M_DSR,_LBLK   ; If HF0=1, stop loading data. Else check SRRQ.
        enddo                   ; Must terminate the do loop
        bra <_LOOP5
_LBLL
        movep X:M_DRXR,P:(R0)+  ; Store 24-bit data in P mem.
_LOOP5                          ; and go get another 24-bit word.
                                ; finish bootstrap
        bra <FINISH             ;

;============================================================================
EPRSCILD
        jclr #1,omr,EPROMLD     ; If MC:MB:MA=001, go load from EPROM

;============================================================================
; This is the routine that loads from the SCI.
; MC:MB:MA=010 - external SCI clock

SCILD
        movep #$0302,X:M_SCR    ; Configure SCI Control Reg
        movep #$C000,X:M_SCCR   ; Configure SCI Clock Control Reg
        movep #7,X:M_PCRE       ; Configure SCLK, TXD and RXD
```

```
        do #6,_LOOP6            ; get 3 bytes for number of
                                ; program words and 3 bytes
                                ; for the starting address
        jclr #2,X:M_SSR,*       ; Wait for RDRF to go high
        movep X:M_SRXL,A2       ; Put 8 bits in A2
        jclr #1,X:M_SSR,*       ; Wait for TDRE to go high
        movep A2,X:M_STXL       ; echo the received byte
        asr #8,a,a
_LOOP6
        move a1,r0              ; starting address for load
        move a1,r1             ; save starting address

        do a0,_LOOP7            ; Receive program words
        do #3,_LOOP8
        jclr #2,X:M_SSR,*       ; Wait for RDRF to go high
        movep X:M_SRXL,A2       ; Put 8 bits in A2
        jclr #1,X:M_SSR,*       ; Wait for TDRE to go high
        movep a2,X:M_STXL       ; echo the received byte
        asr #8,a,a
_LOOP8
        movem a1,p:(r0)+        ; Store 24-bit result in P mem.
_LOOP7
        bra <FINISH            ; Boot from SCI done

;========================================================================
; This is the routine that loads from external EPROM.
; MC:MB:MA=001

EPROMLD
        move #BOOT,r2          ; r2 = address of external EPROM
        movep #AARV,X:M_AAR1   ; aar1 configured for SRAM types of access


        do #6,_LOOP9           ; read number of words and starting address
        movem p:(r2)+,a2       ; Get the 8 LSB from ext. P mem.
        asr #8,a,a             ; Shift 8 bit data into A1
_LOOP9                         ;
        move a1,r0             ; starting address for load
        move a1,r1             ; save it in r1
                              ; a0 holds the number of words

        do a0,_LOOP10          ; read program words
        do #3,_LOOP11          ; Each instruction has 3 bytes
        movem p:(r2)+,a2       ; Get the 8 LSB from ext. P mem.
        asr #8,a,a             ; Shift 8 bit data into A1
_LOOP11                        ; Go get another byte.
        movem a1,p:(r0)+       ; Store 24-bit result in P mem.
_LOOP10                        ; and go get another 24-bit word.
                              ; Boot from EPROM done
;========================================================================
FINISH

; This is the exit handler that returns execution to normal
; expanded mode and jumps to the RESET vector.
```

```
        andi #$0,ccr            ; Clear CCR as if RESET to 0.
        jmp (r1)                ; Then go to starting Prog addr.

; End of bootstrap code. Number of program words: 186
```

A-2   XI/O EQUATES

```
;****************************************************************************
;
;     EQUATES for DSP56301 I/O registers and ports
;
;****************************************************************************

        page    132,55,0,0,0
        opt     mex

ioequ   ident   1,0


;--------------------------------------------------------------------------
;
;       EQUATES for I/O Port Programming
;
;--------------------------------------------------------------------------

;       Register Addresses

M_DATH   EQU     $FFFFCF         ; Host port GPIO data Register
M_DIRH   EQU     $FFFFCE         ; Host port GPIO direction Register
M_PCRC   EQU     $FFFFBF         ; Port C Control Register
M_PRRC   EQU     $FFFFBE         ; Port C Direction Register
M_PDRC   EQU     $FFFFBD         ; Port C GPIO Data Register
M_PCRD   EQU     $FFFFAF         ; Port D Control register
M_PRRD   EQU     $FFFFAE         ; Port D Direction Data Register
M_PDRD   EQU     $FFFFAD         ; Port D GPIO Data Register
M_PCRE   EQU     $FFFF9F         ; Port E Control register
M_PRRE   EQU     $FFFF9E         ; Port E Direction Register
M_PDRE   EQU     $FFFF9D         ; Port E Data Register
M_OGDB   EQU     $FFFFFC         ; OnCE GDB Register

;--------------------------------------------------------------------------
;
;       EQUATES for Host Interface
;
;--------------------------------------------------------------------------

;       Register Addresses
```

```
M_DTXS     EQU     $FFFFCD          ; DSP SLAVE TRANSMIT DATA FIFO (DTXS)
M_DTXM     EQU     $FFFFCC          ; DSP MASTER TRANSMIT DATA FIFO (DTXM)
M_DRXR     EQU     $FFFFCB          ; DSP RECEIVE DATA FIFO (DRXR)
M_DPSR     EQU     $FFFFCA          ; DSP PCI STATUS REGISTER (DPSR)
M_DSR      EQU     $FFFFC9          ; DSP STATUS REGISTER (DSR)
M_DPAR     EQU     $FFFFC8          ; DSP PCI ADDRESS REGISTER (DPAR)
M_DPMC     EQU     $FFFFC7          ; DSP PCI MASTER CONTROL REGISTER (DPMC)
M_DPCR     EQU     $FFFFC6          ; DSP PCI CONTROL REGISTER (DPCR)
M_DCTR     EQU     $FFFFC5          ; DSP CONTROL REGISTER (DCTR)

;          Host Control Register Bit Flags

M_HCIE     EQU     0                ; Host Command Interrupt Enable
M_STIE     EQU     1                ; Slave Transmit Interrupt Enable
M_SRIE     EQU     2                ; Slave Receive Interrupt Enable
M_HF35     EQU     $38              ; Host Flags 5-3 Mask
M_HF3      EQU     3                ; Host Flag 3
M_HF4      EQU     4                ; Host Flag 4
M_HF5      EQU     5                ; Host Flag 5
M_HINT     EQU     6                ; Host Interrupt A
M_HDSM     EQU     13               ; Host Data Strobe Mode
M_HRWP     EQU     14               ; Host RD/WR Polarity
M_HTAP     EQU     15               ; Host Transfer Acknowledge Polarity
M_HDRP     EQU     16               ; Host Dma Request Polarity
M_HRSP     EQU     17               ; Host Reset Polarity
M_HIRP     EQU     18               ; Host Interrupt  Request Polarity
M_HIRC     EQU     19               ; Host Interupt Request Control
M_HM0      EQU     20               ; Host Interface Mode
M_HM1      EQU     21               ; Host Interface Mode
M_HM2      EQU     22               ; Host Interface Mode
M_HM       EQU     $700000          ; Host Interface Mode Mask

;          Host PCI Control Register Bit Flags

M_PMTIE    EQU     1                ; PCI Master Transmit Interrupt Enable
M_PMRIE    EQU     2                ; PCI Master Receive Interrupt Enable
M_PMAIE    EQU     4                ; PCI Master Address Interrupt Enable
M_PPEIE    EQU     5                ; PCI Parity Error Interrupt Enable
M_PTAIE    EQU     7                ; PCI Transaction Abort Interrupt Enable
M_PTTIE    EQU     9                ; PCI Transaction Termination Interrupt Enable
M_PTCIE    EQU     12               ; PCI Transfer Complete Interrupt Enable
M_CLRT     EQU     14               ; Clear Transmitter
M_MTT      EQU     15               ; Master Transfer Terminate
M_SERF     EQU     16               ; HSERR~ Force
M_MACE     EQU     18               ; Master Access Counter Enable
M_MWSD     EQU     19               ; Master Wait States Disable
M_RBLE     EQU     20               ; Receive Buffer Lock Enable
M_IAE      EQU     21               ; Insert Address Enable

;          Host PCI Master Control Register Bit Flags

M_ARH      EQU     $00FFFF          ; DSP PCI Transaction Address (High)
M_BL       EQU     $3F0000          ; PCI Data Burst Length
M_FC       EQU     $C00000          ; Data Transfer Format Control
```

```
;         Host PCI Address Register Bit Flags


M_ARL     EQU     $00FFFF          ; DSP PCI Transaction Address (Low)
M_C       EQU     $0F0000          ; PCI Bus Command
M_BE      EQU     $F00000          ; PCI Byte Enables


;         DSP Status Register Bit Flags


M_HCP     EQU     0                ; Host Command pending
M_STRQ    EQU     1                ; Slave Transmit Data Request
M_SRRQ    EQU     2                ; Slave Receive Data Request
M_HF02    EQU     $38              ; Host Flag 0-2 Mask
M_HF0     EQU     3                ; Host Flag 0
M_HF1     EQU     4                ; Host Flag 1
M_HF2     EQU     5                ; Host Flag 2


;         DSP PCI Status Register Bit Flags


M_MWS     EQU     0                ; PCI Master Wait States
M_MTRQ    EQU     1                ; PCI Master Transmit Data Request
M_MRRQ    EQU     2                ; PCI Master Receive Data Request
M_MARQ    EQU     4                ; PCI Master Address Request
M_APER    EQU     5                ; PCI Address Parity Error
M_DPER    EQU     6                ; PCI Data Parity Error
M_MAB     EQU     7                ; PCI Master Abort
M_TAB     EQU     8                ; PCI Target Abort
M_TDIS    EQU     9                ; PCI Target Disconnect
M_TRTY    EQU     10               ; PCI Target Retry
M_TO      EQU     11               ; PCI Time Out Termination
M_RDC     EQU     $3F0000          ; Remaining Data Count Mask (RDC5-RDC0)
M_RDC0    EQU     16               ; Remaining Data Count  0
M_RDC1    EQU     17               ; Remaining Data Count  1
M_RDC2    EQU     18               ; Remaining Data Count  2
M_RDC3    EQU     19               ; Remaining Data Count  3
M_RDC4    EQU     20               ; Remaining Data Count  4
M_RDC5    EQU     21               ; Remaining Data Count  5
M_HACT    EQU     23               ; Hi32 Active


;-------------------------------------------------------------------------
;
;         EQUATES for Serial Communications Interface (SCI)
;
;-------------------------------------------------------------------------


;         Register Addresses

M_STXH    EQU     $FFFF97          ; SCI Transmit Data Register (high)
M_STXM    EQU     $FFFF96          ; SCI Transmit Data Register (middle)
M_STXL    EQU     $FFFF95          ; SCI Transmit Data Register (low)
```

```
M_SRXH    EQU     $FFFF9A          ; SCI Receive Data Register (high)
M_SRXM    EQU     $FFFF99          ; SCI Receive Data Register (middle)
M_SRXL    EQU     $FFFF98          ; SCI Receive Data Register (low)
M_STXA    EQU     $FFFF94          ; SCI Transmit Address Register
M_SCR     EQU     $FFFF9C          ; SCI Control Register
M_SSR     EQU     $FFFF93          ; SCI Status Register
M_SCCR    EQU     $FFFF9B          ; SCI Clock Control Register

;         SCI Control Register Bit Flags

M_WDS     EQU     $7               ; Word Select Mask (WDS0-WDS3)
M_WDS0    EQU     0                ; Word Select 0
M_WDS1    EQU     1                ; Word Select 1
M_WDS2    EQU     2                ; Word Select 2
M_SSFTD   EQU     3                ; SCI Shift Direction
M_SBK     EQU     4                ; Send Break
M_WAKE    EQU     5                ; Wakeup Mode Select
M_RWU     EQU     6                ; Receiver Wakeup Enable
M_WOMS    EQU     7                ; Wired-OR Mode Select
M_SCRE    EQU     8                ; SCI Receiver Enable
M_SCTE    EQU     9                ; SCI Transmitter Enable
M_ILIE    EQU     10               ; Idle Line Interrupt Enable
M_SCRIE   EQU     11               ; SCI Receive Interrupt Enable
M_SCTIE   EQU     12               ; SCI Transmit Interrupt Enable
M_TMIE    EQU     13               ; Timer Interrupt Enable
M_TIR     EQU     14               ; Timer Interrupt Rate
M_SCKP    EQU     15               ; SCI Clock Polarity
M_REIE    EQU     16               ; SCI Error Interrupt Enable (REIE)

;         SCI Status Register Bit Flags

M_TRNE    EQU     0                ; Transmitter Empty
M_TDRE    EQU     1                ; Transmit Data Register Empty
M_RDRF    EQU     2                ; Receive Data Register Full
M_IDLE    EQU     3                ; Idle Line Flag
M_OR      EQU     4                ; Overrun Error Flag
M_PE      EQU     5                ; Parity Error
M_FE      EQU     6                ; Framing Error Flag
M_R8      EQU     7                ; Received Bit 8 (R8) Address

;         SCI Clock Control Register

M_CD      EQU     $FFF             ; Clock Divider Mask (CD0-CD11)
M_COD     EQU     12               ; Clock Out Divider
M_SCP     EQU     13               ; Clock Prescaler
M_RCM     EQU     14               ; Receive Clock Mode Source Bit
M_TCM     EQU     15               ; Transmit Clock Source Bit

;-----------------------------------------------------------------------
;
;         EQUATES for Synchronous Serial Interface (SSI)
;
;-----------------------------------------------------------------------
```

```
;
;        Register Addresses Of SSI0
M_TX00   EQU     $FFFFBC          ; SSI0 Transmit Data Register 0
M_TX01   EQU     $FFFFBB          ; SSIO Transmit Data Register 1
M_TX02   EQU     $FFFFBA          ; SSIO Transmit Data Register 2
M_TSR0   EQU     $FFFFB9          ; SSI0 Time Slot Register
M_RX0    EQU     $FFFFB8          ; SSI0 Receive Data Register
M_SSISR0 EQU     $FFFFB7          ; SSI0 Status Register
M_CRB0   EQU     $FFFFB6          ; SSI0 Control Register B
M_CRA0   EQU     $FFFFB5          ; SSI0 Control Register A
M_TSMA0  EQU     $FFFFB4          ; SSI0 Transmit Slot Mask Register A
M_TSMB0  EQU     $FFFFB3          ; SSI0 Transmit Slot Mask Register B
M_RSMA0  EQU     $FFFFB2          ; SSI0 Receive Slot Mask Register A
M_RSMB0  EQU     $FFFFB1          ; SSI0 Receive Slot Mask Register B

;        Register Addresses Of SSI1
M_TX10   EQU     $FFFFAC          ; SSI1 Transmit Data Register 0
M_TX11   EQU     $FFFFAB          ; SSI1 Transmit Data Register 1
M_TX12   EQU     $FFFFAA          ; SSI1 Transmit Data Register 2
M_TSR1   EQU     $FFFFA9          ; SSI1 Time Slot Register
M_RX1    EQU     $FFFFA8          ; SSI1 Receive Data Register
M_SSISR1 EQU     $FFFFA7          ; SSI1 Status Register
M_CRB1   EQU     $FFFFA6          ; SSI1 Control Register B
M_CRA1   EQU     $FFFFA5          ; SSI1 Control Register A
M_TSMA1  EQU     $FFFFA4          ; SSI1 Transmit Slot Mask Register A
M_TSMB1  EQU     $FFFFA3          ; SSI1 Transmit Slot Mask Register B
M_RSMA1  EQU     $FFFFA2          ; SSI1 Receive Slot Mask Register A
M_RSMB1  EQU     $FFFFA1          ; SSI1 Receive Slot Mask Register B

;        SSI Control Register A Bit Flags

M_PM     EQU     $FF              ; Prescale Modulus Select Mask (PM0-PM7)
M_PSR    EQU     11               ; Prescaler Range
M_DC     EQU     $1F000           ; Frame Rate Divider Control Mask (DC0-DC7)
M_ALC    EQU     18               ; Alignment Control (ALC)
M_WL     EQU     $380000          ; Word Length Control Mask (WL0-WL7)
M_SSC1   EQU     22               ; Select SC1 as TR #0 drive enable (SSC1)

;        SSI Control Register B Bit Flags

M_OF     EQU     $3               ; Serial Output Flag Mask
M_OF0    EQU     0                ; Serial Output Flag 0
M_OF1    EQU     1                ; Serial Output Flag 1
M_SCD    EQU     $1C              ; Serial Control Direction Mask
M_SCD0   EQU     2                ; Serial Control 0 Direction
M_SCD1   EQU     3                ; Serial Control 1 Direction
M_SCD2   EQU     4                ; Serial Control 2 Direction
M_SCKD   EQU     5                ; Clock Source Direction
M_SHFD   EQU     6                ; Shift Direction
M_FSL    EQU     $180             ; Frame Sync Length Mask (FSL0-FSL1)
M_FSL0   EQU     7                ; Frame Sync Length 0
```

```
M_FSL1    EQU    8                    ; Frame Sync Length 1
M_FSR     EQU    9                    ; Frame Sync Relative Timing
M_FSP     EQU    10                   ; Frame Sync Polarity
M_CKP     EQU    11                   ; Clock Polarity
M_SYN     EQU    12                   ; Sync/Async Control
M_MOD     EQU    13                   ; SSI Mode Select
M_SSTE    EQU    $1C000               ; SSI Transmit enable Mask
M_SSTE2   EQU    14                   ; SSI Transmit #2 Enable
M_SSTE1   EQU    15                   ; SSI Transmit #1 Enable
M_SSTE0   EQU    16                   ; SSI Transmit #0 Enable
M_SSRE    EQU    17                   ; SSI Receive Enable
M_SSTIE   EQU    18                   ; SSI Transmit Interrupt Enable
M_SSRIE   EQU    19                   ; SSI Receive Interrupt Enable
M_STLIE   EQU    20                   ; SSI Transmit Last Slot Interrupt Enable
M_SRLIE   EQU    21                   ; SSI Receive Last Slot Interrupt Enable
M_STEIE   EQU    22                   ; SSI Transmit Error Interrupt Enable
M_SREIE   EQU    23                   ; SSI Receive Error Interrupt Enable

;       SSI Status Register Bit Flags

M_IF      EQU    $3                   ; Serial Input Flag Mask
M_IF0     EQU    0                    ; Serial Input Flag 0
M_IF1     EQU    1                    ; Serial Input Flag 1
M_TFS     EQU    2                    ; Transmit Frame Sync Flag
M_RFS     EQU    3                    ; Receive Frame Sync Flag
M_TUE     EQU    4                    ; Transmitter Underrun Error FLag
M_ROE     EQU    5                    ; Receiver Overrun Error Flag
M_TDE     EQU    6                    ; Transmit Data Register Empty
M_RDF     EQU    7                    ; Receive Data Register Full

;       SSI Transmit Slot Mask Register A

M_SSTSA   EQU    $FFFF                ; SSI Transmit Slot Bits Mask A (TS0-TS15)

;       SSI Transmit Slot Mask Register B

M_SSTSB   EQU    $FFFF                ; SSI Transmit Slot Bits Mask B (TS16-TS31)

;       SSI Receive Slot Mask Register A

M_SSRSA   EQU    $FFFF                ; SSI Receive Slot Bits Mask A (RS0-RS15)

;       SSI Receive Slot Mask Register B

M_SSRSB   EQU    $FFFF                ; SSI Receive Slot Bits Mask B (RS16-RS31)



;------------------------------------------------------------------------
;
;       EQUATES for Exception Processing
;
;------------------------------------------------------------------------
```

```
;       Register Addresses

M_IPRC   EQU     $FFFFFF          ; Interrupt Priority Register Core
M_IPRP   EQU     $FFFFFE          ; Interrupt Priority Register Peripheral


;       Interrupt Priority Register Core (IPRC)

M_IAL    EQU     $7               ; IRQA Mode Mask
M_IAL0   EQU     0                ; IRQA Mode Interrupt Priority Level (low)
M_IAL1   EQU     1                ; IRQA Mode Interrupt Priority Level (high)
M_IAL2   EQU     2                ; IRQA Mode Trigger Mode
M_IBL    EQU     $38              ; IRQB Mode Mask
M_IBL0   EQU     3                ; IRQB Mode Interrupt Priority Level (low)
M_IBL1   EQU     4                ; IRQB Mode Interrupt Priority Level (high)
M_IBL2   EQU     5                ; IRQB Mode Trigger Mode
M_ICL    EQU     $1C0             ; IRQC Mode Mask
M_ICL0   EQU     6                ; IRQC Mode Interrupt Priority Level (low)
M_ICL1   EQU     7                ; IRQC Mode Interrupt Priority Level (high)
M_ICL2   EQU     8                ; IRQC Mode Trigger Mode
M_IDL    EQU     $E00             ; IRQD Mode Mask
M_IDL0   EQU     9                ; IRQD Mode Interrupt Priority Level (low)
M_IDL1   EQU     10               ; IRQD Mode Interrupt Priority Level (high)
M_IDL2   EQU     11               ; IRQD Mode Trigger Mode
M_D0L    EQU     $3000            ; DMA0 Interrupt priority Level Mask
M_D0L0   EQU     12               ; DMA0 Interrupt Priority Level (low)
M_D0L1   EQU     13               ; DMA0 Interrupt Priority Level (high)
M_D1L    EQU     $C000            ; DMA1 Interrupt Priority Level Mask
M_D1L0   EQU     14               ; DMA1 Interrupt Priority Level (low)
M_D1L1   EQU     15               ; DMA1 Interrupt Priority Level (high)
M_D2L    EQU     $30000           ; DMA2 Interrupt priority Level Mask
M_D2L0   EQU     16               ; DMA2 Interrupt Priority Level (low)
M_D2L1   EQU     17               ; DMA2 Interrupt Priority Level (high)
M_D3L    EQU     $C0000           ; DMA3 Interrupt Priority Level Mask
M_D3L0   EQU     18               ; DMA3 Interrupt Priority Level (low)
M_D3L1   EQU     19               ; DMA3 Interrupt Priority Level (high)
M_D4L    EQU     $300000          ; DMA4 Interrupt priority Level Mask
M_D4L0   EQU     20               ; DMA4 Interrupt Priority Level (low)
M_D4L1   EQU     21               ; DMA4 Interrupt Priority Level (high)
M_D5L    EQU     $C00000          ; DMA5 Interrupt priority Level Mask
M_D5L0   EQU     22               ; DMA5 Interrupt Priority Level (low)
M_D5L1   EQU     23               ; DMA5 Interrupt Priority Level (high)


;       Interrupt Priority Register Peripheral (IPRP)

M_HPL    EQU     $3               ; Host Interrupt Priority Level Mask
M_HPL0   EQU     0                ; Host Interrupt Priority Level (low)
M_HPL1   EQU     1                ; Host Interrupt Priority Level (high)
M_S0L    EQU     $C               ; SSI0 Interrupt Priority Level Mask
M_S0L0   EQU     2                ; SSI0 Interrupt Priority Level (low)
```

```
        M_S0L1    EQU    3                    ; SSI0 Interrupt Priority Level (high)
        M_S1L     EQU    $30                  ; SSI1 Interrupt Priority Level Mask
        M_S1L0    EQU    4                    ; SSI1 Interrupt Priority Level (low)
        M_S1L1    EQU    5                    ; SSI1 Interrupt Priority Level (high)
        M_SCL     EQU    $C0                  ; SCI  Interrupt Priority Level  Mask
        M_SCL0    EQU    6                    ; SCI  Interrupt Priority Level  (low)
        M_SCL1    EQU    7                    ; SCI  Interrupt Priority Level  (high)
        M_T0L     EQU    $300                 ; TIMER Interrupt Priority Level Mask
        M_T0L0    EQU    8                    ; TIMER Interrupt Priority Level (low)
        M_T0L1    EQU    9                    ; TIMER Interrupt Priority Level (high)


        ;----------------------------------------------------------------------
        ;
        ;         EQUATES for TIMER
        ;
        ;----------------------------------------------------------------------


        ;         Register Addresses Of TIMER0

        M_TCSR0   EQU    $FFFF8F              ; TIMER0 Control/Status Register
        M_TLR0    EQU    $FFFF8E              ; TIMER0 Load Reg
        M_TCPR0   EQU    $FFFF8D              ; TIMER0 Compare Register
        M_TCR0    EQU    $FFFF8C              ; TIMER0 Count Register


        ;         Register Addresses Of TIMER1

        M_TCSR1   EQU    $FFFF8B              ; TIMER1 Control/Status Register
        M_TLR1    EQU    $FFFF8A              ; TIMER1 Load Reg
        M_TCPR1   EQU    $FFFF89              ; TIMER1 Compare Register
        M_TCR1    EQU    $FFFF88              ; TIMER1 Count Register


        ;         Register Addresses Of TIMER2

        M_TCSR2   EQU    $FFFF87              ; TIMER2 Control/Status Register
        M_TLR2    EQU    $FFFF86              ; TIMER2 Load Reg
        M_TCPR2   EQU    $FFFF85              ; TIMER2 Compare Register
        M_TCR2    EQU    $FFFF84              ; TIMER2 Count Register
        M_TPLR    EQU    $FFFF83              ; TIMER Prescaler Load Register
        M_TPCR    EQU    $FFFF82              ; TIMER Prescalar Count Register


        ;      Timer Control/Status Register Bit Flags

        M_TE      EQU    0                    ; Timer Enable
        M_TOIE    EQU    1                    ; Timer Overflow Interrupt Enable
        M_TCIE    EQU    2                    ; Timer Compare Interrupt Enable
        M_TC      EQU    $F0                  ; Timer Control Mask (TC0-TC3)
        M_INV     EQU    8                    ; Inverter Bit
        M_TRM     EQU    9                    ; Timer Restart Mode
        M_DIR     EQU    11                   ; Direction Bit
        M_DI      EQU    12                   ; Data Input
        M_DO      EQU    13                   ; Data Output
```

```
M_PCE      EQU     15                ; Prescaled Clock Enable
M_TOF      EQU     20                ; Timer Overflow Flag
M_TCF      EQU     21                ; Timer Compare Flag


;          Timer Prescaler Register Bit Flags


M_PS       EQU     $600000           ; Prescaler Source Mask
M_PS0      EQU     21
M_PS1      EQU     22


;          Timer Control Bits
M_TC0      EQU     4                 ; Timer Control 0
M_TC1      EQU     5                 ; Timer Control 1
M_TC2      EQU     6                 ; Timer Control 2
M_TC3      EQU     7                 ; Timer Control 3



;-------------------------------------------------------------------------
;
;          EQUATES for Direct Memory Access (DMA)
;
;-------------------------------------------------------------------------

;          Register Addresses Of DMA
M_DSTR     EQU     $FFFFF4           ; DMA Status Register
M_DOR0     EQU     $FFFFF3           ; DMA Offset Register 0
M_DOR1     EQU     $FFFFF2           ; DMA Offset Register 1
M_DOR2     EQU     $FFFFF1           ; DMA Offset Register 2
M_DOR3     EQU     $FFFFF0           ; DMA Offset Register 3



;          Register Addresses Of DMA0

M_DSR0     EQU     $FFFFFEF          ; DMA0 Source Address Register
M_DDR0     EQU     $FFFFFEE          ; DMA0 Destination Address Register
M_DCO0     EQU     $FFFFFED          ; DMA0 Counter
M_DCR0     EQU     $FFFFFEC          ; DMA0 Control Register

;          Register Addresses Of DMA1

M_DSR1     EQU     $FFFFFEB          ; DMA1 Source Address Register
M_DDR1     EQU     $FFFFFEA          ; DMA1 Destination Address Register
M_DCO1     EQU     $FFFFFE9          ; DMA1 Counter
M_DCR1     EQU     $FFFFFE8          ; DMA1 Control Register

;          Register Addresses Of DMA2

M_DSR2     EQU     $FFFFFE7          ; DMA2 Source Address Register
M_DDR2     EQU     $FFFFFE6          ; DMA2 Destination Address Register
M_DCO2     EQU     $FFFFFE5          ; DMA2 Counter
M_DCR2     EQU     $FFFFFE4          ; DMA2 Control Register
```

```
;         Register Addresses Of DMA4

M_DSR3   EQU     $FFFFE3          ; DMA3 Source Address Register
M_DDR3   EQU     $FFFFE2          ; DMA3 Destination Address Register
M_DCO3   EQU     $FFFFE1          ; DMA3 Counter
M_DCR3   EQU     $FFFFE0          ; DMA3 Control Register


;         Register Addresses Of DMA4


M_DSR4   EQU     $FFFFDF          ; DMA4 Source Address Register
M_DDR4   EQU     $FFFFDE          ; DMA4 Destination Address Register
M_DCO4   EQU     $FFFFDD          ; DMA4 Counter
M_DCR4   EQU     $FFFFDC          ; DMA4 Control Register


;         Register Addresses Of DMA5

M_DSR5   EQU     $FFFFDB          ; DMA5 Source Address Register
M_DDR5   EQU     $FFFFDA          ; DMA5 Destination Address Register
M_DCO5   EQU     $FFFFD9          ; DMA5 Counter
M_DCR5   EQU     $FFFFD8          ; DMA5 Control Register


;         DMA Control Register

M_DSS    EQU     $3               ; DMA Source Space Mask (DSS0-Dss1)
M_DSS0   EQU     0                ; DMA Source Memory space 0
M_DSS1   EQU     1                ; DMA Source Memory space 1
M_DDS    EQU     $C               ; DMA Destination Space Mask (DDS-DDS1)
M_DDS0   EQU     2                ; DMA Destination Memory Space 0
M_DDS1   EQU     3                ; DMA Destination Memory Space 1
M_DAM    EQU     $3F0             ; DMA Address Mode Mask (DAM5-DAM0)
M_DAM0   EQU     4                ; DMA Address Mode 0
M_DAM1   EQU     5                ; DMA Address Mode 1
M_DAM2   EQU     6                ; DMA Address Mode 2
M_DAM3   EQU     7                ; DMA Address Mode 3
M_DAM4   EQU     8                ; DMA Address Mode 4
M_DAM5   EQU     9                ; DMA Address Mode 5
M_D3D    EQU     10               ; DMA Three Dimensional Mode
M_DRS    EQU     $F800            ; DMA Request Source Mask (DRS0-DRS4)
M_DCON   EQU     16               ; DMA Continuous Mode
M_DPR    EQU     $60000           ; DMA Channel Priority
M_DPR0   EQU     17               ; DMA Channel Priority Level (low)
M_DPR1   EQU     18               ; DMA Channel Priority Level (high)
M_DTM    EQU     $380000          ; DMA Transfer Mode Mask (DTM2-DTM0)
M_DTM0   EQU     19               ; DMA Transfer Mode 0
M_DTM1   EQU     20               ; DMA Transfer Mode 1
M_DTM2   EQU     21               ; DMA Transfer Mode 2
M_DIE    EQU     22               ; DMA Interrupt Enable bit
M_DE     EQU     23               ; DMA Channel Enable bit


;         DMA Status Register

M_DTD    EQU     $3F              ; Channel Transfer Done Status MASK (DTD0-DTD5)
```

```
M_DTD0    EQU    0                ; DMA Channel Transfer Done Status 0
M_DTD1    EQU    1                ; DMA Channel Transfer Done Status 1
M_DTD2    EQU    2                ; DMA Channel Transfer Done Status 2
M_DTD3    EQU    3                ; DMA Channel Transfer Done Status 3
M_DTD4    EQU    4                ; DMA Channel Transfer Done Status 4
M_DTD5    EQU    5                ; DMA Channel Transfer Done Status 5
M_DACT    EQU    8                ; DMA Active State
M_DCH     EQU    $E00             ; DMA Active Channel Mask (DCH0-DCH2)
M_DCH0    EQU    9                ; DMA Active Channel 0
M_DCH1    EQU    10               ; DMA Active Channel 1
M_DCH2    EQU    11               ; DMA Active Channel 2


;-------------------------------------------------------------------------
;
;         EQUATES for Phase Locked Loop (PLL)
;
;-------------------------------------------------------------------------

;         Register Addresses Of PLL

M_PCTL    EQU    $FFFFFD          ; PLL Control Register

;         PLL Control Register

M_MF      EQU    $FFF             ; Multiplication Factor Bits Mask (MF0-MF11)
M_DF      EQU    $7000            ; Division Factor Bits Mask (DF0-DF2)
M_XTLR    EQU    15               ; XTAL Range select bit
M_XTLD    EQU    16               ; XTAL Disable Bit
M_PSTP    EQU    17               ; STOP Processing State Bit
M_PEN     EQU    18               ; PLL Enable Bit
M_PCOD    EQU    19               ; PLL Clock Output Disable Bit
M_PD      EQU    $F00000          ; PreDivider Factor Bits Mask (PD0-PD3)


;-------------------------------------------------------------------------
;
;         EQUATES for BIU
;
;-------------------------------------------------------------------------

;         Register Addresses Of BIU


M_BCR     EQU    $FFFFFB          ; Bus Control Register
M_DCR     EQU    $FFFFFA          ; DRAM Control Register
M_AAR0    EQU    $FFFFF9          ; Address Attribute Register 0
M_AAR1    EQU    $FFFFF8          ; Address Attribute Register 1
M_AAR2    EQU    $FFFFF7          ; Address Attribute Register 2
M_AAR3    EQU    $FFFFF6          ; Address Attribute Register 3
M_IDR     EQU    $FFFFF5          ; ID Register
```

```
;        Bus Control Register

M_BA0W    EQU     $1F               ; Area 0 Wait Control Mask (BA0W0-BA0W4)
M_BA1W    EQU     $3E0              ; Area 1 Wait Control Mask (BA1W0-BA14)
M_BA2W    EQU     $1C00             ; Area 2 Wait Control Mask (BA2W0-BA2W2)
M_BA3W    EQU     $E000             ; Area 3 Wait Control Mask (BA3W0-BA3W3)
M_BDFW    EQU     $1F0000          ; Default Area Wait Control Mask (BDFW0-BDFW4)
M_BBS     EQU     21                ; Bus State
M_BLH     EQU     22                ; Bus Lock Hold
M_BRH     EQU     23                ; Bus Request Hold


;        DRAM Control Register

M_BCW     EQU     $3                ; In Page Wait States Bits Mask (BCW0-BCW1)
M_BRW     EQU     $C               ; Out Of Page Wait States Bits Mask (BRW0-BRW1)
M_BPS     EQU     $300              ; DRAM Page Size Bits Mask (BPS0-BPS1)
M_BPLE    EQU     11                ; Page Logic Enable
M_BME     EQU     12                ; Mastership Enable
M_BRE     EQU     13                ; Refresh Enable
M_BSTR    EQU     14                ; Software Triggered Refresh
M_BRF     EQU     $7F8000           ; Refresh Rate Bits Mask (BRF0-BRF7)
M_BRP     EQU     23                ; Refresh prescaler


;        Address Attribute Registers

M_BAT     EQU     $3                ; External Access Type and Pin Definition
                                    ;   Bits Mask (BAT0-BAT1)
M_BAAP    EQU     2                 ; Address Attribute Pin Polarity
M_BPEN    EQU     3                 ; Program Space Enable
M_BXEN    EQU     4                 ; X Data Space Enable
M_BYEN    EQU     5                 ; Y Data Space Enable
M_BAM     EQU     6                 ; Address Muxing
M_BPAC    EQU     7                 ; Packing Enable
M_BNC     EQU     $F00              ; No. of Addr Bits to Compare Mask (BNC0-BNC3)
M_BAC     EQU     $FFF000           ; Address to Compare Bits Mask (BAC0-BAC11)


;        control and status bits in SR

M_CP      EQU     $C00000           ; mask for CORE-DMA priority bits in SR
M_CA      EQU     0                 ; Carry
M_V       EQU     1                 ; Overflow
M_Z       EQU     2                 ; Zero
M_N       EQU     3                 ; Negative
M_U       EQU     4                 ; Unnormalized
M_E       EQU     5                 ; Extension
M_L       EQU     6                 ; Limit
M_S       EQU     7                 ; Scaling Bit
M_I0      EQU     8                 ; Interupt Mask Bit 0
M_I1      EQU     9                 ; Interupt Mask Bit 1
M_S0      EQU     10                ; Scaling Mode Bit 0
M_S1      EQU     11                ; Scaling Mode Bit 1
M_SC      EQU     13                ; Sixteen_Bit Compatibility
M_DM      EQU     14                ; Double Precision Multiply
```

```
M_LF        EQU     15                ; DO-Loop Flag
M_FV        EQU     16                ; DO-Forever Flag
M_SA        EQU     17                ; Sixteen-Bit Arithmetic
M_CE        EQU     19                ; Instruction Cache Enable
M_SM        EQU     20                ; Arithmetic Saturation
M_RM        EQU     21                ; Rounding Mode
M_CP0       EQU     22                ; bit 0 of priority bits in SR
M_CP1       EQU     23                ; bit 1 of priority bits in SR


;       control and status bits in OMR
M_CDP       EQU     $300              ; mask for CORE-DMA priority bits in OMR
M_CDP0      EQU     8                 ; bit 0 of priority bits in OMR
M_CDP1      EQU     9                 ; bit 1 of priority bits in OMR
M_XYS       EQU     16                ; Stack Extension space select bit in OMR.
M_EUN       EQU     17                ; Extensed stack UNderflow flag in OMR.
M_EOV       EQU     18                ; Extended stack OVerflow flag in OMR.
M_WRP       EQU     19                ; Extended WRaP flag in OMR.
M_SEN       EQU     20                ; Stack Extension Enable bit in OMR.
```

A-3   INTERRUPT EQUATES

```
;***************************************************************************
;
;      EQUATES for DSP56301 interrupts
;
;***************************************************************************

        page    132,55,0,0,0
        opt     mex

intequ  ident   1,0

        if      @DEF(I_VEC)
        ;leave user definition as is.
        else
I_VEC   EQU     $0
        endif


;-----------------------------------------------------------------------
; Non-Maskable interrupts
;-----------------------------------------------------------------------
I_RESET EQU     I_VEC+$00         ; Hardware RESET
I_STACK EQU     I_VEC+$02         ; Stack Error
I_ILL   EQU     I_VEC+$04         ; Illegal Instruction
I_DBG   EQU     I_VEC+$06         ; Debug Request
I_TRAP  EQU     I_VEC+$08         ; Trap
I_NMI   EQU     I_VEC+$0A         ; Non Maskable Interrupt


;-----------------------------------------------------------------------
```

```
; Interrupt Request Pins
;-------------------------------------------------------------------------
I_IRQA   EQU     I_VEC+$10         ; IRQA
I_IRQB   EQU     I_VEC+$12         ; IRQB
I_IRQC   EQU     I_VEC+$14         ; IRQC
I_IRQD   EQU     I_VEC+$16         ; IRQD


;-------------------------------------------------------------------------
; DMA Interrupts
;-------------------------------------------------------------------------
I_DMA0   EQU     I_VEC+$18         ; DMA Channel 0
I_DMA1   EQU     I_VEC+$1A         ; DMA Channel 1
I_DMA2   EQU     I_VEC+$1C         ; DMA Channel 2
I_DMA3   EQU     I_VEC+$1E         ; DMA Channel 3
I_DMA4   EQU     I_VEC+$20         ; DMA Channel 4
I_DMA5   EQU     I_VEC+$22         ; DMA Channel 5


;-------------------------------------------------------------------------
; Timer Interrupts
;-------------------------------------------------------------------------
I_TIM0C  EQU     I_VEC+$24         ; TIMER 0 compare
I_TIM0OF EQU     I_VEC+$26         ; TIMER 0 overflow
I_TIM1C  EQU     I_VEC+$28         ; TIMER 1 compare
I_TIM1OF EQU     I_VEC+$2A         ; TIMER 1 overflow
I_TIM2C  EQU     I_VEC+$2C         ; TIMER 2 compare
I_TIM2OF EQU     I_VEC+$2E         ; TIMER 2 overflow


;-------------------------------------------------------------------------
; ESSI Interrupts
;-------------------------------------------------------------------------
I_SI0RD  EQU     I_VEC+$30         ; ESSI0 Receive Data
I_SI0RDE EQU     I_VEC+$32         ; ESSI0 Receive Data With Exception Status
I_SI0RLS EQU     I_VEC+$34         ; ESSI0 Receive last slot
I_SI0TD  EQU     I_VEC+$36         ; ESSI0 Transmit data
I_SI0TDE EQU     I_VEC+$38         ; ESSI0 Transmit Data With Exception Status
I_SI0TLS EQU     I_VEC+$3A         ; ESSI0 Transmit last slot
I_SI1RD  EQU     I_VEC+$40         ; ESSI1 Receive Data
I_SI1RDE EQU     I_VEC+$42         ; ESSI1 Receive Data With Exception Status
I_SI1RLS EQU     I_VEC+$44         ; ESSI1 Receive last slot
I_SI1TD  EQU     I_VEC+$46         ; ESSI1 Transmit data
I_SI1TDE EQU     I_VEC+$48         ; ESSI1 Transmit Data With Exception Status
I_SI1TLS EQU     I_VEC+$4A         ; ESSI1 Transmit last slot


;-------------------------------------------------------------------------
; SCI Interrupts
;-------------------------------------------------------------------------
I_SCIRD  EQU     I_VEC+$50         ; SCI Receive Data
I_SCIRDE EQU     I_VEC+$52         ; SCI Receive Data With Exception Status
I_SCITD  EQU     I_VEC+$54         ; SCI Transmit Data
I_SCIIL  EQU     I_VEC+$56         ; SCI Idle Line
I_SCITM  EQU     I_VEC+$58         ; SCI Timer


;-------------------------------------------------------------------------
; HOST Interrupts
```

```
;----------------------------------------------------------------------
I_HPTT   EQU     I_VEC+$60       ; Host PCI Transaction Termination
I_HPTA   EQU     I_VEC+$62       ; Host PCI Transaction Abort
I_HPPE   EQU     I_VEC+$64       ; Host PCI Parity Error
I_HPTC   EQU     I_VEC+$66       ; Host PCI Transfer Complete
I_HPMR   EQU     I_VEC+$68       ; Host PCI Master Receive
I_HSR    EQU     I_VEC+$6A       ; Host Slave Receive
I_HPMT   EQU     I_VEC+$6C       ; Host PCI Master Transmit
I_HST    EQU     I_VEC+$6E       ; Host Slave Transmit
I_HPMA   EQU     I_VEC+$70       ; Host PCI Master Address
I_HCNMI  EQU     I_VEC+$72       ; Host Command/Host NMI (Default)


;----------------------------------------------------------------------
; INTERRUPT ENDING ADDRESS
;----------------------------------------------------------------------
I_INTEND EQU     I_VEC+$7F       ; last address of interrupt vector space
```

Order this document by DSP56301UM/AD

OnCE is a trade mark of Motorola, Inc.