SECTION 2
ARCHITECTURAL OVERVIEW AND BUS STRUCTURE

The DSP56000/DSP56001 architecture has been designed to maximize throughput in data-intensive digital signal processor (DSP) applications. This objective has resulted in a dual-natured, expandable architecture with sophisticated on-chip peripherals and general purpose I/O. The architecture is dual natured in that there are two independent, expandable data memory spaces, two address generation units (AGUs), and a data arithmetic logic unit (ALU) which has two accumulators and two shifter/limiter circuits.

The duality of the architecture facilitates writing software for DSP applications. For example, data is naturally partitioned into X and Y spaces for graphics and image-processing applications, into coefficient and data spaces for filtering applications, and into real and imaginary spaces for performing complex arithmetic.

The major components of the DSP56000/DSP56001 are as follows:

- Data Buses
- Address Buses
- Data ALU
- AGU
- X Data Memory
- Y Data Memory
- Program Control Unit
- Program Memory
- Input/Output:
  - Memory Expansion (Port A)
  - General-Purpose I/O (Ports B and C)
  - Host Interface
  - Serial Communication Interface (SCI)
  - Synchronous Serial Interface (SSI)

Figure 2-1 shows these components for the DSP56000. Figure 2-2 shows these components for the DSP56001. The processors differ only in the on-chip memory resources.The following paragraphs give a brief description for each component.

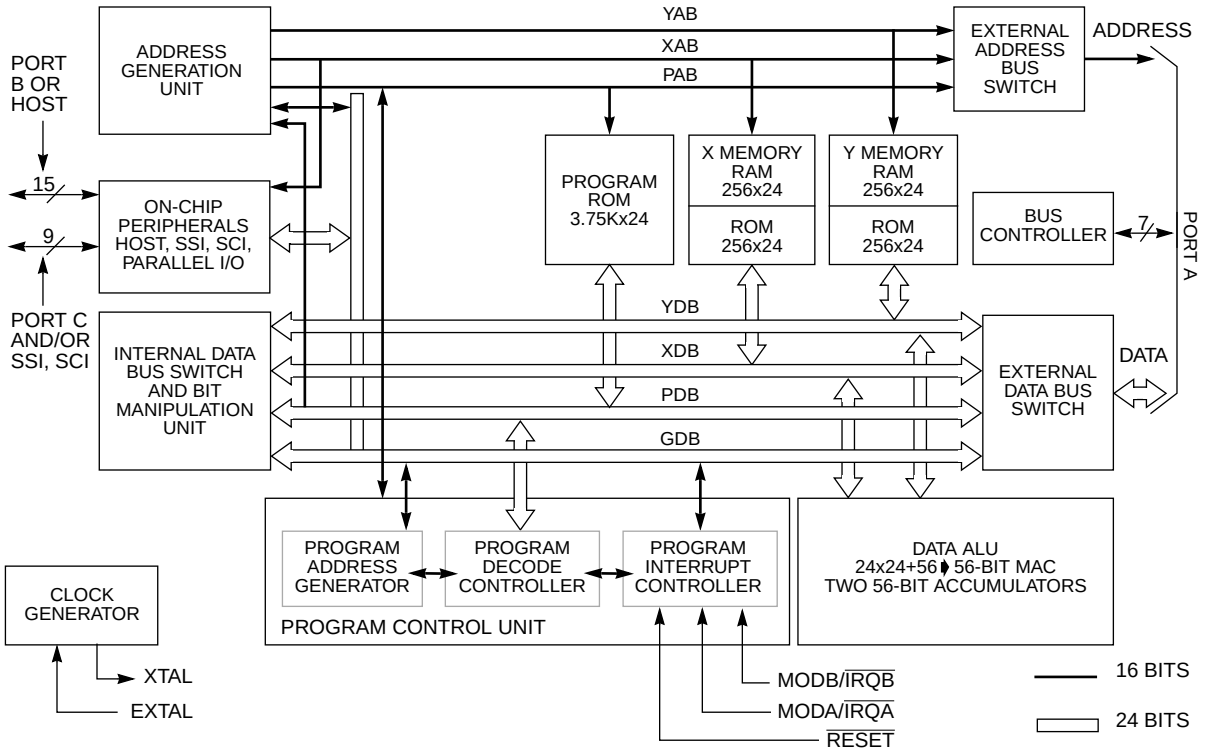

Figure 2-1 DSP56000 Block Diagram

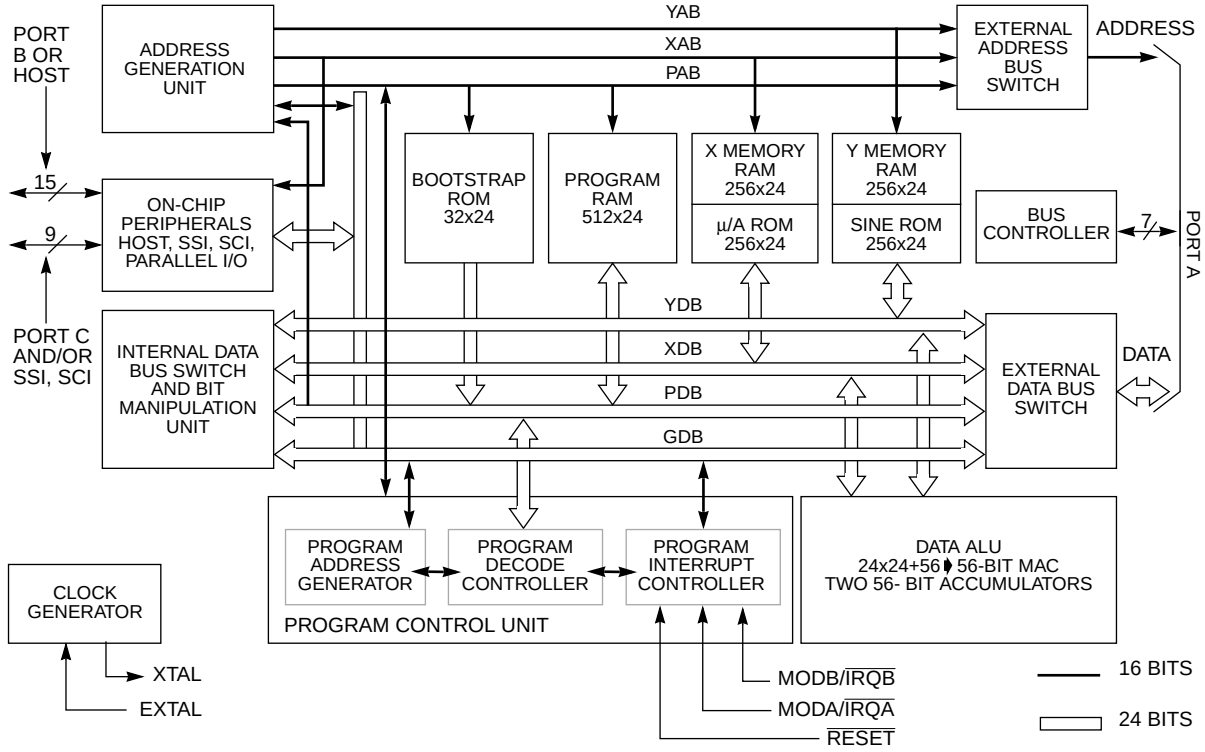

Figure 2-2 DSP56001 Block Diagram

2.1 DATA BUSES

The DSP56000/DSP56001 is organized around the registers of a central processor composed of three independent execution units: the program control unit, the AGU, and the Data ALU.

Data movement on the chip occurs over four, bidirectional, 24-bit buses: the X data bus (XDB), the Y data bus (YDB), the program data bus (PDB), and the global data bus (GDB). The X and Y data buses may also be treated by certain instructions as one 48-bit data bus by concatenation of XDB and YDB. Data transfers between the data ALU and the X data memory or Y data memory occur over XDB and YDB, respectively. XDB and YDB are kept local on the chip to maximize speed and minimize power dissipation. All other data transfers, such as I/O transfers with peripherals, occur over the GDB. Instruction word prefetches occur in parallel over the PDB. The bus structure supports general register-to-register, register-to-memory, and memory-to-register data movement and can transfer up to two 24-bit words and one 56-bit word in the same instruction cycle. Transfers between buses occur in the internal bus switch.

2.2 ADDRESS BUSES

Addresses are specified for internal X data memory and Y data memory on two, unidirectional, 16-bit buses — X address bus (XAB) and Y address bus (YAB). Program memory addresses are specified on the bidirectional program address bus (PAB). External memory spaces are addressed via a single 16-bit, unidirectional address bus driven by a three-input multiplexer that can select the XAB, the YAB, or the PAB. Only one external memory access can be made in an instruction cycle. There is no speed penalty if only one external memory space is accessed in an instruction cycle. If two or three external memory spaces are accessed in a single instruction, there will be a one- or two-instruction-cycle execution delay, respectively. A bus arbitrator controls external access.

2.2.1 Internal Bus Switch

Transfers between buses occur in the internal bus switch. The internal bus switch, which is similar to a switch matrix, can connect any two internal buses without adding any pipeline delays. This flexibility simplifies programming.

2.2.2 Bit Manipulation Unit

The bit manipulation unit is physically located in the internal bus switch block because the internal data bus switch can access each memory space. The bit manipulation unit performs bit manipulation operations on memory locations, address registers, control registers, and data registers over the XDB, YDB, and GDB.

2.3 Data ALU

The data ALU has been designed to process signals which have a wide dynamic range. Special circuitry handles data overflows and roundoff errors.

The data ALU performs all of the arithmetic and logical operations on data operands. It consists of four 24-bit input registers, two 48-bit accumulator registers, two 8-bit accumulator extension registers, an accumulator shifter, two data bus shifter/limiter circuits, and a parallel, single-cycle, nonpipelined multiply-accumulator (MAC) unit. Data ALU operations use fractional twos-complement arithmetic.

Data ALU registers may be read or written over XDB and YDB as 24- or 48-bit operands. The data ALU can perform any of the following operations in a single instruction cycle — multiplication, multiply-accumulate with positive or negative accumulation, convergent rounding, multiply-accumulate with positive or negative accumulation and convergent rounding, addition, subtraction, a divide iteration, a normalization iteration, shifting, and logical operations.

Data ALU source operands, which may be 24, 48, or, in some cases, 56 bits, always originate from data ALU registers. Arithmetic operations always have a 56-bit result stored in an accumulator. Logical operations are performed on 24-bit operands and yield 24-bit results in one of the two accumulators.

The 24-bit data word provides 144 dB of dynamic range, which is sufficient for most real-world applications, since the majority of data converters are 16 bits or less — and certainly not greater than 24 bits. The 56-bit accumulation inside the data ALU provides 336 dB of internal dynamic range so no loss of precision occurs due to intermediate processing.

The data shifter/limiter circuits perform special postprocessing on data read from the ALU accumulator registers A and B out to the XDB or YDB. The data shifters can shift data one bit to the left or one bit to the right as well as pass the data unshifted. Each data shifter has a 24-bit output with overflow indication. The data shifters are controlled by the scaling mode bits in the status register. These shifters permit dynamic scaling of fixed-point data without modifying the program code, which allows block floating-point algorithms to be implemented in a regular fashion. For example, fast Fourier transform (FFT) routines can use this feature to selectively scale each butterfly pass.

“Overflow” occurs when a source operand requires more bits for accurate representation than are available in the destination. To minimize error due to overflow, the DSP56000 writes the maximum (or ‘‘limited’’) signed value the destination can assume when an overflow condition is detected.

In the DSP56000/DSP56001, the data ALU accumulators A and B have extension registers that are used when more than 48-bit accuracy is needed. Therefore, when the extension registers are in use, and either A or B is the source being read over XDB or YDB, data limiters place a ‘‘limited’’ value on XDB or YDB. Such limiting is performed on

the contents of A or B after the contents have been shifted in the shifter. Two limiters allow two-word operands to be limited independently in the same instruction cycle. The two limiters can also be concatenated to form one 48-bit data limiter for long-word operands.

2.4 ADDRESS GENERATION UNIT

All of the address storage and address calculations necessary to indirectly address data operands in memory occur in the AGU. This unit operates in parallel with other chip resources to minimize address generation overhead. The AGU contains eight address registers (R0–R7), eight offset registers (N0–N7), and eight modifier registers (M0–M7). Rn are 16-bit registers that may contain an address or data. The contents of each Rn may be sent to the XAB (65,536 locations), YAB (65,536 locations), or PAB (65,536 locations); thus, 196,608 24-bit data words can be directly addressed. Nn and Mn, which are 16-bit registers normally used in updating or modifying Rn registers, can also be used to store 16-bit data. The AGU registers may be read or written via the GDB as 16-bit operands.

The AGU has two identical address arithmetic units that can generate two 16-bit addresses every instruction cycle — one for any two of the XAB, YAB, or PAB buses. Each of the arithmetic units can perform three types of arithmetic: linear, modulo, and reverse-carry.

2.5 X DATA MEMORY

The on-chip X data random-access memory (RAM), a 24-bit-wide internal static memory, occupies the lowest 256 (0 - 255) locations in X memory space. The on-chip X data read-only memory (ROM) occupies locations 256–511. On the DSP56001, the X data ROM has been programmed as positive Mu-law (128 locations) and A-law (128 locations) 24-bit companding tables useful in telecommunication applications. On the DSP56000, the X data ROM is user defined.

Three on-chip peripherals exist on the DSP56000/DSP56001: an 8-bit parallel host microprocessor unit/direct memory access (MPU/DMA) interface, an SCI, and an SSI. The on-chip peripherals occupy the top 64 locations in X data memory space. Addresses are received from the XAB, and data transfers to the data ALU occur on the XDB. X data memory may be expanded off-chip so that a total of 65,536 locations can be addressed.

2.6 Y DATA MEMORY

The on-chip Y data RAM, a 24-bit-wide internal static memory, occupies the lowest 256 (0 - 255) locations in Y memory space. The on-chip Y data ROM occupies locations 256–511. On the DSP56001, the Y data ROM has been programmed as a full, four-quadrant, 24-bit sine table. On the DSP56000, the Y data ROM is user defined. The off-chip peripheral registers should be mapped into the top 64 locations in Y data memory space.

Addresses are received from the YAB, and data transfers to the data ALU occur on the YDB. Y memory may be expanded off-chip so that a total of 65,536 locations can be

addressed.

2.7 PROGRAM MEMORY

The on-chip program memory consists of a 3.75K-word by 24-bit ROM for the DSP56000 or a 512-word by 24-bit RAM for the DSP56001. Addresses are received from the program control logic (usually the program counter). The interrupt vector addresses for the on-chip resources are located in the bottom 64 locations of program memory. Program memory may be expanded off-chip so that a total of 65,536 locations can be addressed.

Bootstrap ROM is a 32-word by 24-bit factory-programmed ROM used only in the bootstrap mode (operating mode 1). It is available only on the DSP56001; it is not available on the DSP56000. More detailed information on bootstrap ROM is discussed in the DSP56001 Advance Information Data Sheet (ADI1290).

2.8 PROGRAM CONTROL UNIT

The program control unit performs instruction prefetch, instruction decoding, hardware DO loop control, and exception processing. It contains a 15-level by 32-bit system stack memory and the following six directly addressable registers: the program counter (PC), loop address (LA), loop counter (LC), status register (SR), operating mode register (OMR), and stack pointer (SP). The 16-bit PC can address 65,536 locations in program memory space.

2.9 INPUT/OUTPUT

The I/O capability of the DSP56000/DSP56001 is extensive and advanced.Its structure facilitates interfacing into a variety of system configurations, including multiple DSP56000/DSP56001 systems (with or without a host processor), global bus systems with bus arbitration, and many serial configurations, all with minimal additional ‘‘glue’’ logic.

Each I/O interface, which has its own control, status, and data registers, is treated as memory-mapped I/O by the DSP56000/DSP56001. Each interface has several dedicated interrupt vector addresses and control bits to enable/disable interrupts, which minimizes the overhead associated with servicing the device. The interrupt sources can be programmed to one of three maskable priority levels.

The I/O structure consists of a flexible, 47-pin expansion port (Port A) and 24 additional I/O pins. These pins may operate as general-purpose I/O pins, called port B and port C, or they may be allocated to on-chip peripherals (MPU/DMA, SCI, and SSI) under software control.

Port B is a 15-bit I/O interface that may function as general-purpose I/O pins or as host MPU/DMA interface pins.

Port C is a 9-bit I/O interface that may be used as general-purpose I/O pins or as SCI and

2.9.1 Expansion Port (Port A)

DSP56000/DSP56001 expansion port is designed to synchronously interface over a common 24-bit data bus which has a wide variety of memory and peripheral devices. These devices include high-speed static RAMs, slower memory devices, and other DSPs and MPUs in master/slave configurations. This variety is possible because the expansion bus timing is programmable.

Two pins can be defined with a control bit to operate as either master processor controls (called “bus strobe” and “bus wait” in this configuration) or as slave processor controls (called “bus request” and “bus grant”).

The expansion bus timing can also be controlled by a bus control register (BCR). The BCR controls the timing of bus interface signals RD and WR, as well as the data output lines. Each of the four memory spaces, X data, Y data, program data, and I/O, has its own 4-bit register in the BCR that can be programmed for inserting up to 15 wait states (one wait state is equal to a clock period or equivalently one-half of an instruction cycle). Thus, external bus timing can be tailored to match the speed requirements of the different memory spaces.

2.9.2 General-Purpose I/O (Ports B and C)

Each Port B and Port C pin may be programmed as a general-purpose I/O pin or as a dedicated, on-chip peripheral pin under software control. A 9-bit port C control register (PCC) allows each port C pin to be programmed for one of these two functions. The port control register associated with port B (PBC) contains only one bit, which programs all 15 pins. Also associated with each general-purpose port is a data direction register, which programs the direction of each pin, and a data register for data I/O. All these registers are memory mapped and read/write, which makes the use of bit manipulation instructions extremely effective.

2.9.3 Host Interface

The host interface is a byte-wide, full-duplex, parallel port that can be connected directly to the data bus of a host processor. The host processor may be any of a number of industry-standard microcomputers or MPUs, another DSP, or DMA hardware. To control data transfers the DSP56000/DSP56001 host interface has an 8-bit, bidirectional data bus: H0–H7 (PB0–PB7); and seven dedicated control lines: HA0, HA1, HA2, HR/$\overline{W}$, $\overline{HEN}$, $\overline{HREQ}$, and $\overline{HACK}$ (PB8–PB14).

The host interface appears as a memory-mapped peripheral occupying eight bytes in the host-processor address space. Separate double buffered transmit and receive data registers allow the DSP56000/DSP56001 and host processor to efficiently transfer data at high speed. Standard, host-processor data move instructions and addressing modes facilitate communication with the host interface. Handshake flags are provided for polled or interrupt-driven data transfers with the host processor. DMA hardware may be used with the handshake flags to efficiently transfer data without using address lines HA0–HA2.

One of the most innovative features of the host interface is the host command feature. With this feature, the host processor can issue vectored exception requests to the

DSP56000/DSP56001. The host may select any one of 32 DSP56000/DSP56001 exception routines to be executed by writing a vector address register in the host interface. This flexibility allows the host programmer to execute up to 32 preprogrammed functions inside the DSP56000/DSP56001. For example, host exception routines allow the host processor to read or write DSP56000/DSP56001 registers, X, Y, or program memory locations, force exception handlers (e.g., SSI, SCI, $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$ exception routines), and perform control and debugging operations.

2.9.4 Serial Communication Interface

The SCI provides a full-duplex port for 8-bit data serial communication to other DSPs, MPUs, or peripherals such as modems. The communication can be either direct or over RS232C-type lines. 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 (up to 2.5 Mbits/sec) synchronous data transmission.

The asynchronous protocols include a multidrop mode for master/slave operation. The SCI consists of separate transmit and receive sections having operations that can be asynchronous with respect to each other by using the internal clock for one and an external clock for the other. A programmable baud-rate generator is included to generate the transmit and receive clocks. An enable and interrupt vector are included so that the baud-rate generator can function as a general-purpose timer when it is not being used by the SCI peripheral.

2.9.5 Synchronous Serial Interface

The SSI is a flexible, full-duplex serial interface that allows the DSP56000/DSP56001 to communicate with a variety of serial devices, including one or more industry-standard codecs, other DSPs, MPUs, and peripherals.

The user can independently define the following characteristics of the SSI: the number of bits per word, the protocol, the clock, and the transmit/receive synchronization.

The user can select three modes: normal, on-demand, and network. The normal mode is typically used to interface with devices on a regular or periodic basis. The data-driven on-demand mode is intended to be used to communicate with devices on a nonperiodic basis. The network mode provides time slots in addition to a bit clock and frame synchronization pulse.

The SSI functions with a range of 2 to 32 words of I/O per frame in the network mode. This mode is typically used in star or ring time division multiplex (TDM) networks with other DSP56000s and/or codecs. The clock can be programmed to be continuous or gated. Since the transmitter and receiver sections of the SSI are independent, they can be programmed to be synchronous (using a common clock) or asynchronous with respect to each other.

The SSI supports a subset of the Motorola SPI. The SSI requires up to six pins, depending on its operating mode. The most common minimum configuration is three pins: transmit data (STD), receive data (SRD), and clock (SCK).

2.10 SIGNAL DESCRIPTION

The DSP56000/DSP56001 is available in an 88-pin pin-grid array package or surface mount. The input and output signals are organized into the following seven functional groups which are shown in Figure 2-1:

1. Port A Address and Data Buses
2. Port A Bus Control
3. Interrupt and Mode Control
4. Power and Clock
5. Host Interface or Port B I/O
6. SCI or Port C I/O
7. SSI or Port C I/O

The signals are discussed in the following paragraphs.

2.10.1 Port A Address and Data Bus

The following signals relate to the Port A address and data bus.

2.10.1.1 Address (A0–A15)

These three-state output pins specify the address for external program and data memory accesses. To minimize power dissipation, A0–A15 do not change state when external memory spaces are not being accessed.

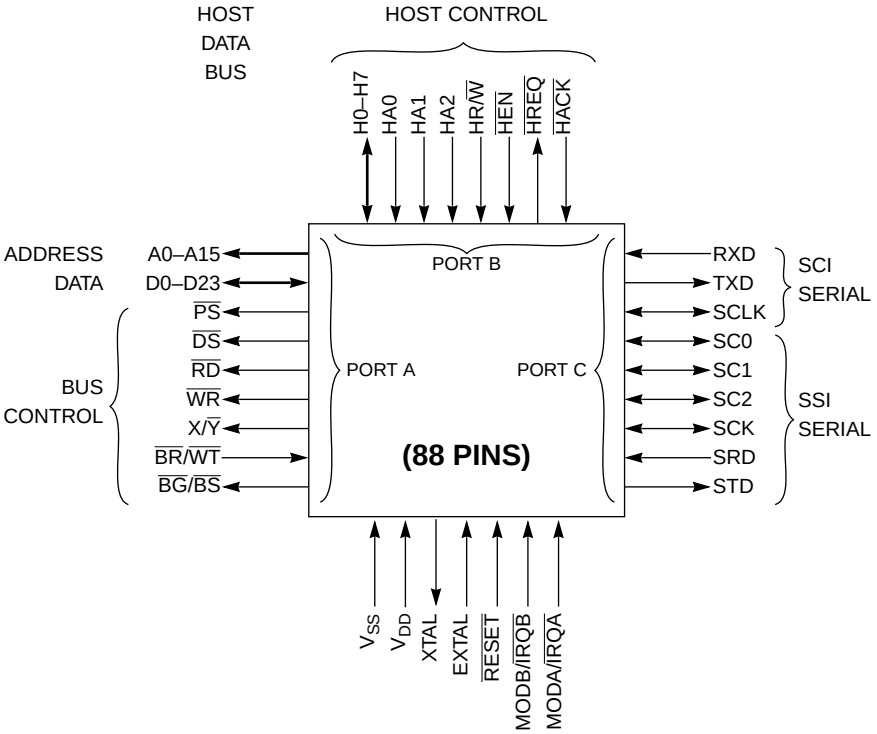

Figure 2-1 DSP56000/DSP56001 Functional Signal Groups

2.10.1.2 Data (D0–D23) These pins provide the bidirectional data bus for external program and data memory accesses. D0–D23 are in the high-impedance state when the bus grant signal is asserted.

2.10.2 Port A Bus Control The Port A bus control signals are discussed in the following paragraphs.

2.10.2.1 Program Memory Select ($\overline{PS}$) This three-state output is asserted only when external program memory is referenced (see Table 2-1)**.**

Table 2-1 Program and Data Memory Select Encoding

$\overline{PS}$	$\overline{DS}$	X/$\overline{Y}$	External Memory Reference
1	1	1	No Activity
1	0	1	X Data Memory on Data Bus
1	0	0	Y Data Memory on Data Bus
0	1	1	Program Memory on Data Bus (Not Exception)
0	1	0	External Exception Fetch: Vector or Vector +1 (Development Mode Only)
0	0	X	Reserved
1	1	0	Reserved

2.10.2.2 Data Memory Select ($\overline{DS}$) This three-state output is asserted only when external data memory is referenced (see Table 2-1).

2.10.2.3 X/$\overline{Y}$ Select (X/$\overline{Y}$) This three-state output selects which external data memory space (X or Y) is referenced by $\overline{DS}$ (see Table 2-1l).

2.10.2.4 Read Enable ($\overline{RD}$) This three-state output is asserted to read external memory on the data bus (D0–D23).

2.10.2.5 Write Enable ($\overline{WR}$) This three-state output is asserted to write external memory on the data bus (D0–D23).

2.10.2.6 Bus Request/Wait ($\overline{BR}$/$\overline{WT}$) The bus request input ($\overline{BR}$) allows another device such as a processor or DMA controller to become the master of the external data bus (D0–D23) and external address bus (A0–A15). When bit 7 of the operating mode register (OMR)

is clear and $\overline{BR}$ is asserted, the DSP56000/DSP56001 will always release Port A, including A0–A15, D0–D23, and the bus control pins by placing them in the high-impedance state after execution of the current instruction has been completed.

If OMR bit 7 is set, $\overline{BR}$/$\overline{WT}$ acts as an input that allows an external device to force wait states during an external Port A operation for as long as $\overline{WT}$ is asserted.

2.10.2.7 Bus Grant/Bus Strobe ($\overline{BG}$/$\overline{BS}$) If OMR bit 7 is clear, this output is asserted to grant an external bus request after Port A has been released. If OMR bit 7 is set, this pin assumes bus strobe and is asserted when the DSP accesses Port A.

2.10.3 Interrupt and Mode Control

The signals described in the following paragraphs are the interrupt and mode control signals for the DSP56000/DSP56001.

2.10.3.1 Mode Select A/External Interrupt Request A (MODA/$\overline{IRQA}$) and Mode Select B/External Interrupt Request B (MODB/$\overline{IRQB}$) These two inputs have dual functions: 1) to select the initial chip operating mode and 2) to receive an interrupt request from an external source.

MODA and MODB are read and internally latched in the DSP when the processor exits the reset state. After leaving the reset state, the MODA and MODB pins automatically change to external interrupt requests, $\overline{IRQA}$ and $\overline{IRQB}$.

After leaving the reset state, the chip operating mode can be changed by software. $\overline{IRQA}$ and $\overline{IRQB}$ can be programmed to be level sensitive or negative edge triggered. When edge triggered, triggering occurs at a voltage level and is not directly related to the fall time of the interrupt signal. However, as the fall time of the interrupt signal increases, theprobability for noise on $\overline{IRQA}$ or $\overline{IRQB}$ to generate multiple interrupts also increases.

2.10.3.2 Reset ($\overline{RESET}$) This Schmitt-trigger input pin is used to reset the DSP56000/DSP56001. When $\overline{RESET}$ is asserted, the DSP56000/DSP56001 is initialized and placed in the reset state. When $\overline{RESET}$ is deasserted, the initial chip operating mode is latched from the MODA and MODB pins. When coming out of $\overline{RESET}$, deassertion occurs at a voltage level and is not directly related to the rise time of the $\overline{RESET}$ signal; however, the probability of noise on $\overline{RESET}$ generating multiple resets increases with increasing rise time of the $\overline{RESET}$ signal.

2.10.4 Power and Clock

The power and clock signals are presented in the following paragraphs.

2.10.4.1 Power (V_{CC}), Ground (GND)

There are five sets of power and ground pins: two pairs for internal logic; one power, and two ground for Port A address and control pins; one power and two ground for Port A data pins; and one pair for peripherals.

2.10.4.2 External Clock/Crystal Input (EXTAL)

EXTAL interfaces the internal crystal oscillator input to an external crystal or an external clock.

2.10.4.3 Crystal Output (XTAL)

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

2.10.5 Host Interface

The following paragraphs discuss the host interface signals.

2.10.5.1 Host Data Bus (H0–H7)

This bidirectional data bus transfers data between the host processor and the DSP56000/DSP56001. This bus is an input unless enabled by a host processor read. It is high impedance when $\overline{\text{HEN}}$ is deasserted. H0–H7 can be programmed as general-purpose parallel I/O pins (PB0–PB7) when the host interface is not being used.

2.10.5.2 Host Address (HA0–HA2)

These inputs provide the address selection for each host interface register. HA0–HA2 can be programmed as general-purpose parallel I/O pins (PB8–PB10) when the host interface is not being used.

2.10.5.3 Host Read/Write (HR/$\overline{\text{W}}$)

This input selects the direction of data transfer for each host processor access. HR/$\overline{\text{W}}$ can be programmed as a general-purpose I/O pin (PB11) when the host interface is not being used.

2.10.5.4 Host Enable ($\overline{\text{HEN}}$)

This input enables a data transfer on the host data bus. When $\overline{\text{HEN}}$ is asserted and HR/$\overline{\text{W}}$ is high, H0–H7 become outputs and DSP56000/DSP56001 data may be read by the host processor. When $\overline{\text{HEN}}$ is asserted and HR/$\overline{\text{W}}$ is low, H0–H7 become inputs, and host data is latched inside the DSP. When $\overline{\text{HEN}}$ is deasserted, the host data bus is three-stated. Normally, a chip select signal derived from host address decoding and an enable clock are used to generate $\overline{\text{HEN}}$. $\overline{\text{HEN}}$ can be programmed as a general-purpose I/O pin (PB12) when the host interface is not being used.

2.10.5.5 Host Request ($\overline{\text{HREQ}}$)

This open-drain output signal is used by the DSP56000/DSP56001 host interface to request service from the host processor, DMA controller, or a simple external controller. $\overline{\text{HREQ}}$ can be programmed as a general-purpose (not open-drain) I/O pin (PB13) when the host interface is not being used.

2.10.5.6 Host Acknowledge ($\overline{\text{HACK}}$)

This input has two functions: 1) to provide a host acknowledge handshake signal for DMA transfers and 2) to receive a host interrupt acknowledge compatible with M68000 Family processors. $\overline{\text{HACK}}$ may be programmed as a general-purpose I/O pin (PB14) when the host interface is not being used.

2.10.6 Serial Communications Interface

The following signals relate to the SCI.

2.10.6.1 Receive Data (RXD)

This input receives byte-oriented serial data and transfers the data to the SCI receive shift register. RXD can be programmed as a general-purpose I/O pin (PC0) when the SCI RXD function is not being used.

2.10.6.2 Transmit Data (TXD)

This output transmits serial data from the SCI transmit shift register. TXD can be programmed as a general-purpose I/O pin (PC1) when the SCI TXD function is not being used.

2.10.6.3 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 can be programmed as a general-purpose I/O pin (PC2) when the SCI SCLK function is not being used.

2.10.7 Synchronous Serial Interface

The SSI signals are presented in the following paragraphs.

2.10.7.1 Serial Clock Zero (SC0)

The SSI uses this bidirectional pin for control by the SSI as a flag or receiver clock. SC0 can be programmed as a general-purpose I/O pin (PC3) when the SSI SC0 function is not being used.

2.10.7.2 Serial Control One (SC1)

The SSI uses this bidirectional pin to control flag or frame synchronization. SC1 can be programmed as a general-purpose I/O pin (PC4) when the SSI SC1 function is not being used.

2.10.7.3 Serial Control Two (SC2)

The SSI uses this bidirectional pin to control frame synchronization only. SC2 can be programmed as a general-purpose I/O pin (PC5) when the SSI SC2 function is not being used.

2.10.7.4 SSI Serial Clock (SCK)

This bidirectional pin provides the serial bit rate clock for the SSI. SCK can be programmed as a general-purpose I/O pin (PC6) when SCK is not being used.

2.10.7.5 SSI Receive Data (SRD)

This input pin receives serial data into the SSI receive shift register. SRD can be programmed as a general-purpose I/O pin (PC7) when SRD is not being used.

2.10.7.6 SSI Transmit Data (STD)

This output pin transmits serial data from the SSI transmit shift register. STD can be programmed as a general-purpose I/O pin (PC8) when the SSI STD function is not being used.