Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

DSP56309 USER'S MANUAL

DSP56309UM
Rev. 1, December 2005

freescale™
semiconductor

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations not listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GMBH
Technical Information Center
Schatzbogen 7
81829 München, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
+800 2666 8080

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Order Number: DSP56309UM
Rev. 1
12/2005

freescale™
semiconductor

Contents

3 Memory Configuration

4 Core Configuration

5 Programming the Peripherals

8 Serial Communication Interface (SCI)

B Programming Reference

DSP56309 Overview 1

This manual describes the DSP56309 24-bit digital signal processor (DSP), its memory, operating modes, and peripheral modules. The DSP56309 is an implementation of the DSP56300 core with a unique configuration of internal memory, cache, and peripherals.

Use this manual in conjunction with the *DSP56300 Family Manual (DSP56300FM),* which describes the CPU, core programming models, and instruction set. The *DSP56309 Technical Data (DSP56309)*—referred to as the data sheet—provides DSP56309 electrical specifications, timing, pinout, and packaging descriptions.

You can obtain these documents—and the Freescale DSP development tools—through a local Freescale Semiconductor Sales Office or authorized distributor. To receive the latest information on this DSP, access the Freescale web site at the address listed on the back cover of this manual.

1.1 Manual Organization

This manual contains the following sections and appendices:

- **Chapter 1,** *Overview*. Features list and block diagram, related documentation, organization of this manual, and the notational conventions used.
- **Chapter 2,** *Signals/Connections*. DSP56309 signals and their functional groupings.
- **Chapter 3,** *Memory Configuration*. DSP56309 memory spaces, RAM configuration, memory configuration bit settings, memory configurations, and memory maps.
- **Chapter 4,** *Core Configuration*. Registers for configuring the DSP56300 core when programming the DSP56309, in particular the interrupt vector locations and the operation of the interrupt priority registers; operating modes and how they affect the processor's program and data memories.
- **Chapter 5,** *Programming the Peripherals*. Guidelines on initializing the DSP56309 peripherals, including mapping control registers, specifying a method of transferring data, and configuring for general-purpose input/output (GPIO).
- **Chapter 6,** *Host Interface (HI08)*. Signals, architecture, programming model, reset, interrupts, external host programming model, initialization, and a quick reference to the HI08 programming model.

- **Chapter 7,** *Enhanced Synchronous Serial Interface (ESSI)*. Enhancements, data and control signals, programming model, operating modes, initialization, exceptions, and GPIO.
- **Chapter 8,** *Serial Communication Interface (SCI)*. Signals, programming model, operating modes, reset, initialization, and GPIO.
- **Chapter 9,** *Triple Timer Module*. Architecture, programming model, and operating modes of three identical timer devices available for use as internals or event counters.
- **Appendix A,** *Bootstrap Code*. Bootstrap code and equates for the DSP56309.
- **Appendix B,** *Programming Reference*. Peripheral addresses, interrupt addresses, and interrupt priorities for the DSP56309; programming sheets listing the contents of the major DSP56309 registers for programmer's reference.

1.2 Manual Conventions

This manual uses the following conventions:

- Bits within registers are always listed from most significant bit (MSB) to least significant bit (LSB).
- Bits within a register are indicated AA[n – m], n > m, when more than one bit is involved in a description. For purposes of description, the bits are presented as if they are contiguous within a register. However, this is not always the case. Refer to the programming model diagrams or to the programming sheets to see the exact location of bits within a register.
- When a bit is "set," its value is 1. When a bit is "cleared," its value is 0.
- The word "assert" means that a high true (active high) signal is pulled high to V_{CC} or that a low true (active low) signal is pulled low to ground. The word "deassert" means that a high true signal is pulled low to ground or that a low true signal is pulled high to V_{CC}. See **Table 1-1**.

Table 1-1. High True/Low True Signal Conventions

Signal/Symbol	Logic State	Signal State	Voltage
$\overline{\text{PIN}}$[1]	True	Asserted	Ground[2]
PIN	False	Deasserted	V_{CC}[3]
PIN	True	Asserted	V_{CC}
PIN	False	Deasserted	Ground

Notes:
1. PIN is a generic term for any pin on the chip.
2. Ground is an acceptable low voltage level. See the appropriate data sheet for the range of acceptable low voltage levels (typically a TTL logic low).
3. V_{CC} is an acceptable high voltage level. See the appropriate data sheet for the range of acceptable high voltage levels (typically a TTL logic high).

- Pins or signals that are asserted low (made active when pulled to ground) are indicated like this:
  - In text, they have an overbar: for example, $\overline{\text{RESET}}$ is asserted low.
  - In code examples, they have a tilde in front of their names. In **Example 1-1**, line 3 refers to the $\overline{\text{SS0}}$ signal (shown as `~SS0`).
- Sets of signals are indicated by the first and last signals in the set, for instance HAD[0–7].
- "Input/Output" indicates a bidirectional signal. "Input or Output" indicates a signal that is exclusively one or the other.
- Code examples are displayed in a monospaced font, as shown in **Example 1-1**.

Example 1-1. Sample Code Listing

```
BFSET#$0007,X:PCC; Configure:                          line 1
   ;  MISO0, MOSI0, SCK0 for SPI master                 line 2
   ; ~SS0 as PC3 for GPIO                               line 3
```

- Hexadecimal values are indicated with a $ preceding the value, as follows: $FFFFFF is the X memory address for the core interrupt priority register.
- The word "reset" is used in four different contexts in this manual:
  - the reset signal, written as $\overline{\text{RESET}}$
  - the reset instruction, written as RESET
  - the reset operating state, written as Reset
  - the reset function, written as reset

1.3 Manual Revision History for Revision 1

Revision 1 is completely reorganized compared to Revision 0. This resulted in reordering of information in most chapters and the deletion of the old **Chapters 10 and 11** and **Appendices B and C**. The old **Appendix D** became **Appendix B**. Other significant differences are listed in **Table 1-2**.

Table 1-2. Change History, Revision 0 to Revision 1

Change	Revision 0 Page Number	Revision 1 Page Number
Modified signal definitions. Added note 5 pertaining to GND in **Table 2-1**.	Page 2-3	Page 2-1
Modified signal definitions. Added a note 4 pertaining to GND at the bottom of **Figure 2-1**. Removed overbars from HA10, HRW, AA0–AA3, and TMS.	Page 2-4	Page 2-2
Modified signal definitions. Changed the note at the end of **Table 2-3**.	Page 2-7	Page 2-4
Modified signal definitions. In **Table 2-8**, changed the **State During Reset, Stop, or Wait** descriptions for the $\overline{\text{BR}}$ and $\overline{\text{BB}}$ signals.	Pages 2-11 to 2-12	Pages 2-6 to 2-7

Table 1-2. Change History, Revision 0 to Revision 1 (Continued)

Change	Revision 0 Page Number	Revision 1 Page Number
Modified signal definitions. In **Table 2-11**, changed the title of the third column to **State During Reset**[1,2] Added a new note 1 and changed the old note 1 to note 2. Changed the **State During Reset** of all signals to "Ignored input." Changed the signal description for PB14.	Pages 2-17 to 2-21	Pages 2-10 to 2-12
Modified signal definitions. In **Table 2-12** to **Table 2-15**, deleted the Stop column. Changed the title of the third column to **State During Reset**[1,2] Added a new note 1 and changed the old note 1 to note 2.	Page 2-22 to 2-31	Page 2-13 to 2-19
Operating Mode Register layout and definition. Replaced **Figure 4-2**.	Page 4-17	Page 4-13
In **Section 8.6.4.1**, changed the beginning of the fourth paragraph from "In Synchronous mode" to "In Asynchronous mode."	Page 8-20	Page 8-21
Updated programming sheets. Replaced the programming sheets for the following registers: • **Figure B-2**, Operating Mode Register (OMR) (old **Figure D-2**) • **Figure B-21**, Timer Load, Compare, and Count Registers (TLR, TCPR, and TCR) (old **Figure D-21**)	 Page D-13 Page D-33	 Page B-11 Page B-30

1.4 Features

The Freescale DSP56309, a member of the DSP56300 core family of programmable DSPs, supports wireless infrastructure applications with general filtering operations. Like the other family members, the DSP56309 uses a high-performance, single-clock-cycle- per-instruction engine (code compatible with Freescale's popular DSP56000 core family), a barrel shifter, 24-bit addressing, instruction cache, and DMA controller. The DSP56309 offers 100 million instructions per second (MIPS) performance using an internal 100 MHz clock with 3.3 V core and input/output (I/O) power.

All DSP56300 core family members contain the DSP56300 core and additional modules. The modules are chosen from a library of standard predesigned elements, such as memories and peripherals. New modules can be added to the library to meet customer specifications. A standard interface between the DSP56300 core and the internal memory and peripherals supports a wide variety of memory and peripheral configurations. In particular, the DSP56309 includes a JTAG port integrated with the Freescale OnCE module.

The DSP56309 is intended for use in telecommunication applications, such as multi-line voice/data/fax processing, video conferencing, audio applications, control, and general digital signal processing

1.5 DSP56300 Core

Core features are fully described in the *DSP56300 Family Manual.* This manual, in contrast, documents pinout, memory, and peripheral features. Core features are as follows:

- 100 million instructions per second (MIPS) with a 100 MHz clock at 3.0–3.6 V
- Highly parallel instruction set
- Data arithmetic logic unit (Data ALU)
  - Fully pipelined 24 × 24-bit parallel multiplier-accumulator (MAC)
  - 56-bit parallel barrel shifter (fast shift and normalization; bit stream generation and parsing)
  - Conditional ALU instructions
  - 24-bit or 16-bit arithmetic support under software control
- Program control unit (PCU)
  - Position Independent Code (PIC) support
  - Addressing modes optimized for DSP applications (including immediate offsets)
  - Instruction cache controller
  - Internal memory-expandable hardware stack
  - Nested hardware DO loops
  - Fast auto-return interrupts
- Direct memory access (DMA)
  - Six DMA channels supporting internal and external accesses
  - One-, two-, and three- dimensional transfers (including circular buffering)
  - End-of-block-transfer interrupts
  - Triggering from interrupt lines and all peripherals
- Phase lock loop (PLL)
  - Allows change of low power Divide Factor (DF) without loss of lock
  - Output clock with skew elimination
- Hardware debugging support
  - On-chip emulation (OnCE) module
  - Joint Test Action Group (JTAG) Test Access Port (TAP)
  - Address Trace mode reflects internal program RAM accesses at the external port
- Reduced power dissipation
  - Very low-power CMOS design
  - Wait and stop low-power standby modes
  - Fully-static design specified to operate down to 0 Hz (dc)
  - Optimized power-management circuitry (instruction-dependent, peripheral-dependent, and mode-dependent)

1.6 DSP56300 Core Functional Blocks

The functional blocks of the DSP56300 core are:

- Data arithmetic logic unit (ALU)
- Address generation unit

- Program control unit
- PLL and clock oscillator
- JTAG TAP and OnCE module
- Memory

In addition, the DSP56309 provides a set of internal peripherals, discussed in **Section 1.9**, *Peripherals*, on page 1-12.

1.6.1 Data ALU

The data ALU performs all the arithmetic and logical operations on data operands in the DSP56300 core. These are the components of the data ALU:

- Fully pipelined 24 × 24-bit parallel multiplier-accumulator
- Bit field unit, comprising a 56-bit parallel barrel shifter (fast shift and normalization; bit stream generation and parsing)
- Conditional ALU instructions
- Software-controllable 24-bit, 48-bit, or 56-bit arithmetic support
- Four 24-bit or 48-bit input general-purpose registers: X1, X0, Y1, and Y0
- Six data ALU registers (A2, A1, A0, B2, B1, and B0) that are concatenated into two general-purpose, 56-bit accumulators, A and B, accumulator shifters
- Two data bus shifter/limiter circuits

1.6.1.1 Data ALU Registers

The data ALU registers are read or written over the X data bus and the Y data bus as 16- or 32-bit operands. The source operands for the data ALU can be 16, 32, or 40 bits and always originate from data ALU registers. The results of all data ALU operations are stored in an accumulator. Data ALU operations are performed in two clock cycles in a pipeline so that a new instruction can be initiated in every clock cycle, yielding an effective execution rate of one instruction per clock cycle. The destination of every arithmetic operation can be a source operand for the immediately following operation without penalty.

1.6.1.2 Multiplier-Accumulator (MAC)

The MAC unit comprises the main arithmetic processing unit of the DSP56300 core and performs all of the calculations on data operands. For arithmetic instructions, the unit accepts as many as three input operands and outputs one 56-bit result of the following form: extension:most significant product:least significant product (EXT:MSP:LSP).

The multiplier executes 24-bit × 24-bit parallel, fractional multiplies between twos-complement signed, unsigned, or mixed operands. The 48-bit product is right-justified and added to the 56-bit

contents of either the A or B accumulator. A 56-bit result can be stored as a 24-bit operand. The LSP is either truncated or rounded into the MSP. Rounding is performed if specified.

1.6.2 Address Generation Unit (AGU)

The AGU performs the effective address calculations using integer arithmetic necessary to address data operands in memory and contains the registers that generate the addresses. It implements four types of arithmetic: linear, modulo, multiple wrap-around modulo, and reverse-carry. The AGU operates in parallel with other chip resources to minimize address-generation overhead.

The AGU is divided into halves, each with its own identical address ALU. Each address ALU has four sets of register triplets, and each register triplet includes an address register, offset register, and modifier register. Each contains a 24-bit full adder (called an offset adder). A second full adder (called a modulo adder) adds the summed result of the first full adder to a modulo value that is stored in its respective modifier register. A third full adder (called a reverse-carry adder) is also provided. The offset adder and the reverse-carry adder work in parallel and share common inputs. The only difference between them is that the carry propagates in opposite directions. Test logic determines which of the three summed results of the full adders is output.

Each address ALU can update one address register from its own address register file during one instruction cycle. The contents of the associated modifier register specify the type of arithmetic used in the address register update calculation. The modifier value is decoded in the address ALU.

1.6.3 Program Control Unit (PCU)

The PCU fetches and decodes instructions, controls hardware DO loops, and processes exceptions. Its seven-stage pipeline controls the different processing states of the DSP56300 core. The PCU consists of three hardware blocks:

- *Program decode controller*. Decodes the 24-bit instruction loaded into the instruction latch and generates all signals for pipeline control.
- *Program address generator*. Contains all the hardware needed for program address generation, system stack, and loop control.
- *Program interrupt controller*. Arbitrates among all interrupt requests (internal interrupts, as well as the five external requests $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, $\overline{\text{IRQC}}$, $\overline{\text{IRQD}}$, and $\overline{\text{NMI}}$), and generates the appropriate interrupt vector address.

PCU features include the following:

- Position-independent code support
- Addressing modes optimized for DSP applications (including immediate offsets)
- Instruction cache controller

- Internal memory-expandable hardware stack
- Nested hardware DO loops
- Fast auto-return interrupts
- Hardware system stack

The PCU uses the following registers:

- Program counter register
- Status register
- Loop address register
- Loop counter register
- Vector base address register
- Size register
- Stack pointer
- Operating mode register
- Stack counter register

1.6.4 PLL and Clock Oscillator

The clock generator in the DSP56300 core comprises two main blocks: the PLL, which performs clock input division, frequency multiplication, and skew elimination; and the clock generator, which performs low-power division and clock pulse generation. These features allow you to:

- Change the low-power divide factor without losing the lock
- Output a clock with skew elimination

The PLL allows the processor to operate at a high internal clock frequency using a low-frequency clock input, a feature that offers two immediate benefits:

- A lower-frequency clock input reduces the overall electromagnetic interference generated by a system.
- The ability to oscillate at different frequencies reduces costs by eliminating the need to add additional oscillators to a system.

1.6.5 JTAG TAP and OnCE Module

In the DSP56300 core is a dedicated user-accessible TAP that is fully compatible with the *IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture*. Problems with testing high-density circuit boards led to the development of this standard under the sponsorship of the Test Technology Committee of IEEE and the JTAG. The DSP56300 core implementation supports circuit-board test strategies based on this standard. The test logic includes a TAP with four dedicated signals, a 16-state controller, and three test data registers. A boundary scan

register links all device signals into a single shift register. The test logic, implemented utilizing static logic design, is independent of the device system logic. For details on the JTAG port, consult the *DSP56300 Family Manual*.

The OnCE module interacts with the DSP56300 core and its peripherals nonintrusively so that you can examine registers, memory, or internal peripherals. This facilitates hardware and software development on the DSP56300 core processor. OnCE module functions are provided through the JTAG TAP signals. For details on the OnCE module, consult the *DSP56300 Family Manual*.

1.6.6 Internal Memory

The memory space of the DSP56300 core is partitioned into program, X data, and Y data memory space. The data memory space is divided into X and Y data memory in order to work with the two address ALUs and to feed two operands simultaneously to the data ALU. Memory space includes internal RAM and ROM and can be expanded off-chip under software control. For details on internal memory, see **Chapter 3,** *Memory Configuration*. Program RAM, instruction cache, X data RAM, and Y data RAM size are programmable, as shown in **Table 1-2**.

Table 1-3. Internal Memory

Instruction Cache	Switch Mode	Program RAM Size	Instruction Cache Size	X Data RAM Size	Y Data RAM Size
disabled	disabled	20480 × 24-bit	0	7168 × 24-bit	7168 × 24-bit
enabled	disabled	19456 × 24-bit	1024 × 24-bit	7168 × 24-bit	7168 × 24-bit
disabled	enabled	24576 × 24-bit	0	5120 × 24-bit	5120 × 24-bit
enabled	enabled	23552 × 24-bit	1024 × 24-bit	5120 × 24-bit	5120 × 24-bit

There is an internal 192 × 24-bit bootstrap ROM.

1.6.7 External Memory Expansion

Memory can be expanded externlly as follows:

- Data memory expansion to two 256 K × 24-bit word memory spaces using the standard external address lines
- Program memory expansion to one 256 K × 24-bit words memory space using the standard external address lines

Further features of external memory include the following:

- External memory expansion port
- Simultaneous glueless interface to static random access memory (SRAM) and dynamic random access memory (DRAM)

1.7 Internal Buses

To provide data exchange between the blocks, the DSP56309 implements the following buses:

- Peripheral I/O expansion bus to peripherals
- Program memory expansion bus to program ROM
- X memory expansion bus to X memory
- Y memory expansion bus to Y memory
- Global data bus between PCU and other core structures
- Program data bus for carrying program data throughout the core
- X memory data bus for carrying X data throughout the core
- Y memory data bus for carrying Y data throughout the core
- Program address bus for carrying program memory addresses throughout the core
- X memory address bus for carrying X memory addresses throughout the core
- Y memory address bus for carrying Y memory addresses throughout the core.

The block diagram in **Figure 1-1** illustrates these buses among other components. All internal buses on the DSP56300 family members are 24-bit buses. The program data bus is also a 24-bit bus.

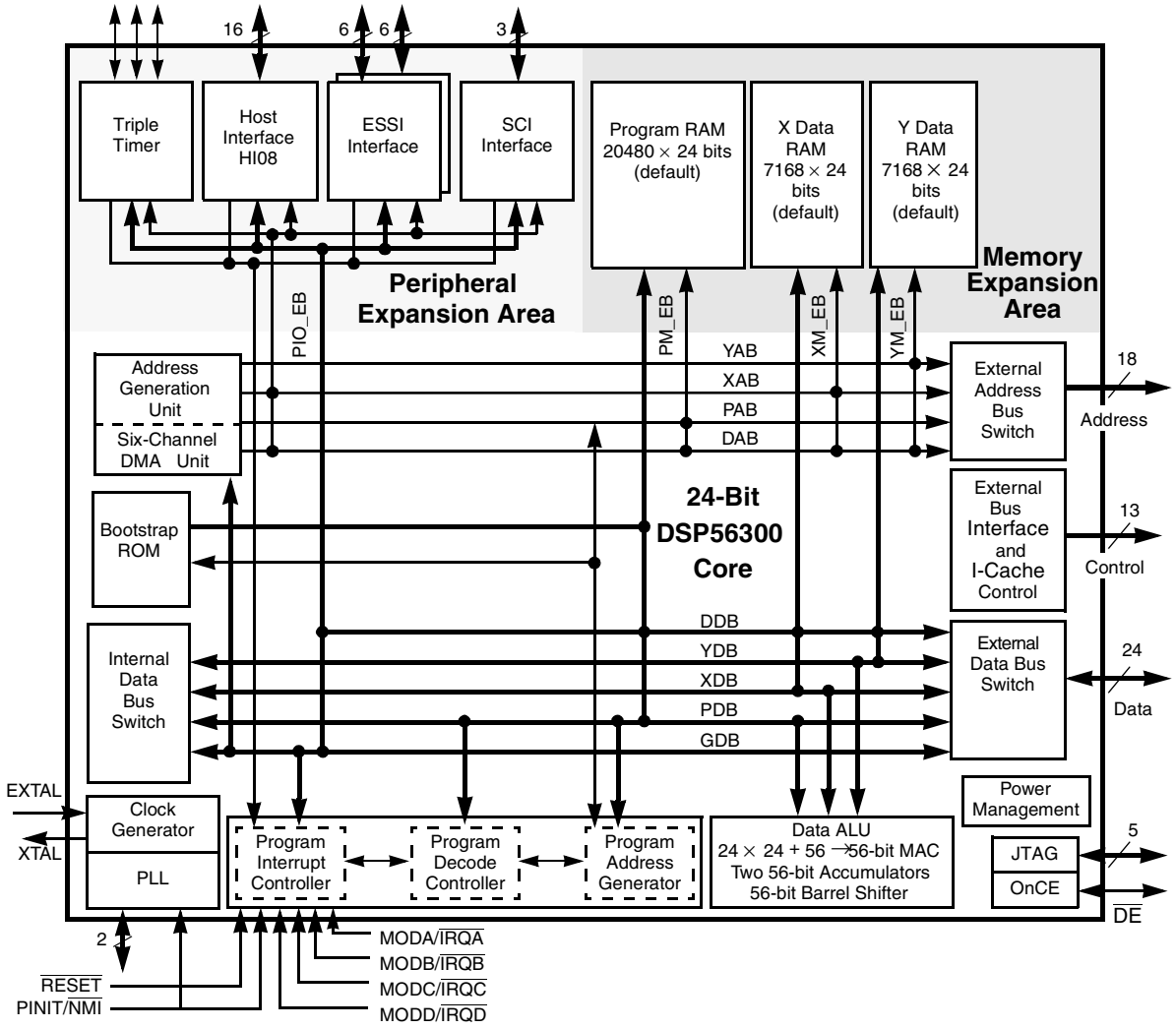

Figure 1-1. DSP56309 Block Diagram

Note: See **Section 1.6.6**, *Internal Memory*, on page 1-9 for memory size details.

1.8 DMA

The DMA block has the following features:

- Six DMA channels supporting internal and external accesses
- One-, two-, and three-dimensional transfers (including circular buffering)
- End-of-block-transfer interrupts
- Triggering from interrupt lines and all peripherals

1.9 Peripherals

In addition to the core features, the DSP56309 provides the following peripherals:

- As many as 34 user-configurable GPIO signals
- HI08 to external hosts
- Dual ESSI
- SCI
- Triple timer module
- Memory switch mode
- Four external interrupt/mode control lines

1.9.1 GPIO Functionality

The GPIO port consists of up to 34 programmable signals, also used by the peripherals (HI08, ESSI, SCI, and timer). There are no dedicated GPIO signals. After a reset, the signals are automatically configured as GPIO. Three memory-mapped registers per peripheral control GPIO functionality. Programming techniques for these registers to control GPIO functionality are detailed in **Chapter 5**, *Programming the Peripherals*.

1.9.2 HI08

The HI08 is a byte-wide, full-duplex, double-buffered parallel port that can connect directly to the data bus of a host processor. The HI08 supports a variety of buses and provides connection with a number of industry-standard DSPs, microcomputers, and microprocessors without requiring any additional logic. The DSP core treats the HI08 as a memory-mapped peripheral occupying eight 24-bit words in data memory space. The DSP can use the HI08 as a memory-mapped peripheral, using either standard polled or interrupt programming techniques. Separate double-buffered transmit and receive data registers allow the DSP and host processor to transfer data efficiently at high speed. Memory mapping allows you to program DSP core communication with the HI08 registers using standard instructions and addressing modes.

1.9.3 ESSI

The DSP56309 provides two independent and identical ESSIs. Each ESSI has a full-duplex serial port for communication with a variety of serial devices, including one or more industry-standard codecs, other DSPs, microprocessors, and peripherals that implement the Freescale SPI. The ESSI consists of independent transmitter and receiver sections and a common ESSI clock generator. ESSI capabilities include the following:

- Independent (asynchronous) or shared (synchronous) transmit and receive sections with separate or shared internal/external clocks and frame syncs
- Normal mode operation using frame sync

- Network mode operation with as many as 32 time slots
- Programmable word length (8, 12, or 16 bits)
- Program options for frame synchronization and clock generation
- One receiver and three transmitters per ESSI

1.9.4 SCI

The SCI provides a full-duplex port for serial communication with other DSPs, microprocessors, or peripherals such as modems. The SCI interfaces without additional logic to peripherals that use TTL-level signals. With a small amount of additional logic, the SCI can connect to peripheral interfaces that have non-TTL level signals, such as the RS-232C, RS-422, etc. This interface uses three dedicated signals: transmit data, receive data, and SCI serial clock. It supports industry-standard asynchronous bit rates and protocols, as well as high-speed synchronous data transmission (up to 12.5 Mbps for a 100 MHz clock). SCI asynchronous protocols include a multidrop mode for master/slave operation with wakeup on idle line and wakeup on address bit capability. This mode allows the DSP56309 to share a single serial line efficiently with other peripherals.

Separate SCI transmit and receive sections can operate asynchronously with respect to each other. A programmable baud-rate generator provides the transmit and receive clocks. An enable vector and an interrupt vector allow the baud-rate generator to function as a general-purpose timer when the SCI is not using it or when the interrupt timing is the same as that used by the SCI.

1.9.5 Timer Module

The triple timer module is composed of a common 21-bit prescaler and three independent and identical general-purpose 24-bit timer/event counters, each with its own memory-mapped register set. Each timer has the following properties:

- A single signal that can function as a GPIO signal or as a timer signal
- Uses internal or external clocking and can interrupt the DSP after a specified number of events (clocks) or signal an external device after counting internal events
- Connection to the external world through one bidirectional signal. When this signal is configured as an input, the timer functions as an external event counter or measures external pulse width/signal period. When the signal is used as an output, the timer functions as either a timer, a watchdog, or a pulse width modulator.

Signals/Connections 2

The DSP56309 input and output signals are organized into functional groups, as shown in **Table 2-1** and illustrated in **Figure 2-1**. The DSP56309 operates from a 3 V supply; however, some of the inputs can tolerate 5 V. A special notice for this feature is added to the signal descriptions of those inputs.

Table 2-1. DSP56309 Functional Signal Groupings

Functional Group		Number of Signals	Description and Page
Power (V_{CC})		18	**Table 2-2** on page -3
Ground (GND)		19	**Table 2-3** on page -3
Clock		2	**Table 2-4** on page -4
PLL		3	**Table 2-5** on page -5
Address bus	Port A[1]	18	**Table 2-6** on page -5
Data bus		24	**Table 2-7** on page -6
Bus control		13	**Table 2-8** on page -6
Interrupt and mode control		5	**Table 2-9** on page -8
HI08	Port B2	16	**Table 2-11** on page -10
ESSI	Ports C and D[3]	12	**Table 2-12** on page -14 **Table 2-13** on page -16
SCI	Port E[4]	3	**Table 2-14** on page -18
Timer[5]		3	**Table 2-15** on page -19
OnCE/JTAG Port		6	**Table 2-16** on page -20

Notes:
1. Port A signals define the external memory interface port, including the external address bus, data bus, and control signals.
2. Port B signals are the HI08 port signals multiplexed with the GPIO signals.
3. Port C and D signals are the two ESSI port signals multiplexed with the GPIO signals.
4. Port E signals are the SCI port signals multiplexed with the GPIO signals.
5. The number of Ground signals listed are for the 144-pin TQFP package. For the 196-ball MAP-BGA package, there are 66 GND connections.