

Chapter 5

Program Control Unit

The Program Control Unit (PCU) of the DSP56300 family core coordinates execution of program instructions and instructions for processing interrupts and exceptions. The PCU also controls which of the five DSP56300 core processing states (Normal, Exception, Reset, Wait, or Stop) is currently selected. The PCU functions through a seven-stage instruction pipeline and several programmable registers. This chapter describes the PCU hardware, programming model, and instruction pipeline.

5.1 Overview

The PCU coordinates execution of instructions using three hardware blocks: the Program Address Generator (PAG), the Program Decode Controller (PDC), and the Program Interrupt Controller (PIC). These blocks perform the following functions:

- Fetch instructions
- Decode instructions
- Execute instructions
- Control hardware DO loops and REP
- Process interrupts and exceptions

Operation of the seven-stage pipeline depends on the current core processing state. The seven stages of the pipeline are as follows:

- Fetch-I
- Fetch-II
- Decode
- Address gen-I
- Address gen-II
- Execute-I
- Execute-II

To preserve current operation and status values while processing exceptions and interrupts, the PCU provides a System Stack to store current register contents before executing the exception/interrupt handler program. These contents are restored when control returns to the current program. In addition to these standard program flow-control resources, the PCU provides special support for hardware DO loops and an instruction REPEAT mechanism.

To perform its functions, the PCU uses a number of programmable registers. The organization of these registers forms the programming model for the PCU:

- General configuration and status:
 - Operating Mode Register (OMR)—24-bit, read/write
 - Status Register (SR)—24-bit, read/write
- System Stack configuration and operation:
 - System Stack (SS) register file—hardware stack, 48-bit × 16 locations, read/write
 - System Stack High (SSH) Register—24-bit, read/write
 - System Stack Low (SSL) Register—24-bit, read/write
 - Stack Pointer (SP) Register—24-bit, read/write
 - Stack Counter (SC) Register—5-bit, read/write
 - Stack Size (SZ) Register—24-bit, read/write

Note: The stack Extension Pointer (EP) Register is also used with the System Stack, but is physically part of the Address Generation Unit. For a description of this register, refer to **Chapter 4, Address Generation Unit**.

- Program/Loop/Exception processing control
 - Program Counter (PC) Register—24-bit, read/write
 - Loop Address (LA) Register—24-bit, read/write
 - Loop Counter (LC) Register—24-bit, read/write
 - Vector Base Address (VBA) Register—24-bit, read/write

5.2 PCU Hardware Architecture

The three PCU hardware blocks are:

- Program Address Generator (PAG)—Contains all the hardware needed for program address generation, System Stack, and loop control
- Program Decode Controller (PDC)
 - Decodes the 24-bit instruction loaded into the instruction latch
 - Generates all signals for pipeline control
 - Performs required data transfers between the Data Arithmetic Logic Unit (Data ALU) and memory
- Program Interrupt Controller (PIC)—Arbitrates among all interrupt requests (internal interrupts and the five external interrupts: \overline{IRQA} , \overline{IRQB} , \overline{IRQC} , \overline{IRQD} and \overline{NMI}) and generates the appropriate interrupt vector address

Figure 5-1 shows a block diagram of the PCU.

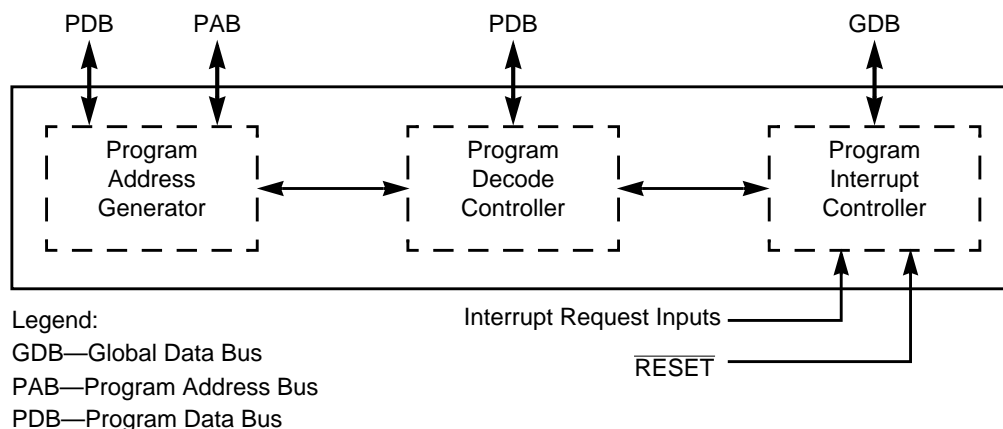


Figure 5-1. PCU Architecture

5.3 Instruction Pipeline

Within the seven-stage pipelined architecture of the PCU, instructions execute concurrently. Execution of a given pipeline stage for one instruction occurs concurrently with execution of other pipeline stages for other instructions. Table 5-1 and Figure 5-2 show that these stages include two fetch stages, one decode stage, two address generation stages, and two execute stages. The pipelined operation is essentially transparent, thus easing programmability. Transparency is achieved by means of interlock hardware present in every execution unit of the processor so that programs written for the DSP56000 family devices execute correctly on the DSP56300 core without any modification. However, code can be optimized to reduce interlocks and improve execution speed.

Table 5-1 Seven-Stage Pipeline

Pipeline Stage	Description
Fetch-I	<ul style="list-style-type: none">■ Address generation for Program Fetch■ Increment PC register
Fetch-II	<ul style="list-style-type: none">■ Instruction word read from memory
Decode	<ul style="list-style-type: none">■ Instruction Decode
AddressGen-I	<ul style="list-style-type: none">■ Address generation for Data Load/Store operations
AddressGen-II	<ul style="list-style-type: none">■ Address pointer update
Execute-I	<ul style="list-style-type: none">■ Read source operands to Multiplier and Adder■ Read source register for memory store operations■ Multiply■ Write destination register for memory load operations
Execute-II	<ul style="list-style-type: none">■ Read source operands for Adder if written by previous ALU operation■ Add■ Write Adder results to the Adder destination operand■ Write Multiplier results to the Multiplier destination operands

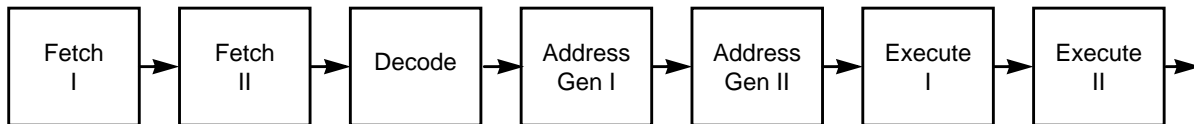


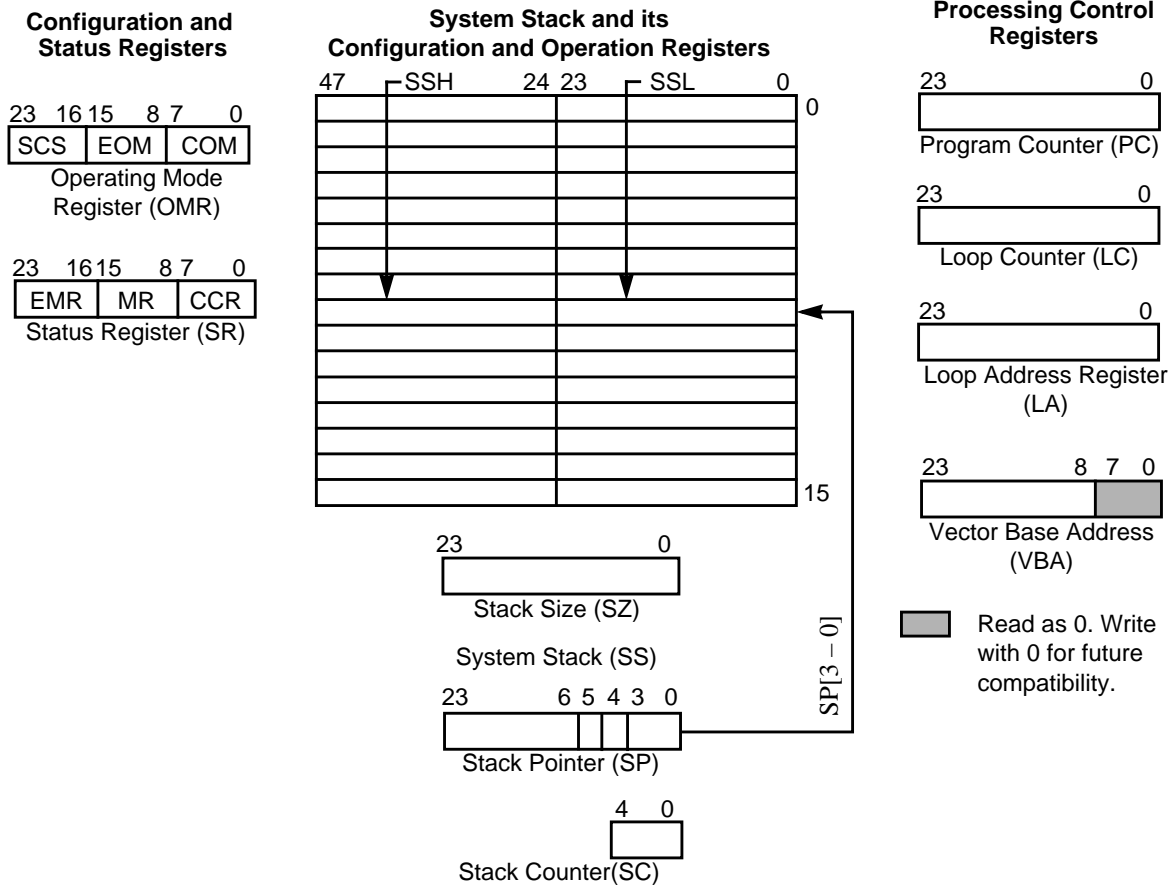
Figure 5-2. Seven-Stage Pipeline

5.4 Programming Model

The PCU programming model comprises three functional areas:

- Configuration and status registers
- System Stack configuration and operation registers
- Program/Loop/Exception processing control registers

Figure 5-3 shows the PCU programming model with the registers and the System Stack. The following paragraphs describe each register.



- Notes:**
1. The Extension Pointer (EP) Register is also used with the System Stack, but it is physically part of the Address Generation Unit (AGU).
 2. SSH and SSL point to the upper and lower halves of the stack location specified by the SP.

Figure 5-3. PCU Programming Model

5.4.1 Configuration and Status Registers

Note: Bits that are listed as reserved in the following sections can be defined for specific devices within the DSP56300 family. Refer to the device-specific user’s manual to determine whether a reserved bit is defined for that device.

The PCU contains two registers that configure and report the current status of the PCU:

- Operating Mode Register (OMR)
- Status Register (SR)

5.4.1.1 Operating Mode Register

The OMR (**Figure 5-4**) is a 24-bit register that is partitioned into the following three bytes:

- OMR[23 – 16], System Stack Control/Status (SCS) Byte: Controls and monitors the stack extension in the data memory. The SCS byte is referenced implicitly by some instructions—such as DO, JSR, and RTI—or directly by the MOVEC instruction.
- OMR[15 – 8], Extended Chip Operating Mode (EOM) Byte: Determines the operating mode of the chip. This byte is affected only by hardware reset and by instructions directly referencing the OMR (that is, ANDI, ORI, and other instructions, such as MOVEC, that specify OMR as a destination).
- OMR[7 – 0], Chip Operating Mode (COM) Byte: Determines the operating mode of the chip. This byte is affected only by hardware reset and by instructions directly referencing the OMR (that is, ANDI, ORI, and other instructions, such as MOVEC, that specify OMR as a destination). During hardware reset, the chip operating mode bits (MD, MC, MB, and MA) are loaded from the external mode select pins MODD, MODC, MODB, and MODA, respectively.

The following sections describe all defined bit functions; however, not all defined functions are implemented on all DSP56300 family devices. Always write non-implemented functions as zeros to ensure future compatibility. Refer to the latest device-specific user's manuals, technical data sheets, and technical bulletins for detailed information about implementation and usage for a particular device.

Stack Control/Status (SCS)						Extended Operating Mode (EOM)						Chip Operating Mode (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
PEN	MSW[1:0]	SEN	WRP	EOV	EUN	XYS	ATE	APD	ABE	BRT	TAS	BE	CDP[1:0]	MS	SD		EBD	MD	MC	MB	MA		

Reserved bit. Read as zero; write with zero for future compatibility

Values after reset:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	*	*	*	*
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

* After reset, these bits reflect the corresponding value of the mode input (that is, MODD, MODC, MODB, or MODA, respectively).

- PEN—Patch Enable
- MSW1—Memory Switch Configuration 1
- MSW0—Memory Switch Configuration 0
- SEN—Stack Extension Enable
- WRP—Stack Extension Wrap Flag
- EOV—Stack Extension Overflow Flag
- EUN—Stack Extension Underflow Flag
- XYS—Stack Extension Space Select
- ATE—Address Trace Enable
- APD—Address Attribution Priority Disable
- ABE—Asynch. Bus Arbitration Enable
- BRT—Bus Release Timing
- TAS— $\overline{\text{TA}}$ Signal Synchronize Select
- BE—Cache Burst Mode Enable
- CDP1—Core-DMA Priority 1
- CDP0—Core-DMA Priority 0
- MS—Memory Switch Mode
- SD—Stop Delay Mode
- EBD—External Bus Disable
- MD—Chip Operating Mode D
- MC—Chip Operating Mode C
- MB—Chip Operating Mode B
- MA—Chip Operating Mode A

Figure 5-4. Operating Mode Register (OMR)

Table 5-2 Operating Mode Register Bit Definitions

Bit Number	Bit Name	Reset Value	Description
23	PEN	0	Patch Enable Enables/Disables the memory patch function, if implemented. Refer to the device-specific user's manual to determine whether and how this function is used on a specific device. Hardware reset clears this bit.
22 – 21	MSW	0	Memory Switch Configuration Determine what portion of the higher locations of internal X and Y data memory are switched to internal program memory when Memory Switch mode is enabled. Memory Switch mode allows reallocation of portions of X and Y data RAM as program RAM. Memory Switch mode is enabled when the Memory Switch bit, OMR[7] is set. For details on how much memory is switched, see the device-specific user's manual for a particular DSP56300 family device. The MSW bits are not available on all members of the DSP56300 family.
20	SEN	0	Stack Extension Enable Enables/ Disables the stack extension in data memory. If SEN is set, the extension is enabled. Hardware reset clears this bit, so the default out of reset is a disabled stack extension.

Table 5-2 Operating Mode Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
19	WRP	0	<p>Stack Extension Wrap Flag</p> <p>During the debugging phase of the software development, this flag can be used to evaluate and increase the speed of software-implemented algorithms. WRP is set when copying from the on-chip hardware stack (System Stack Register file) to the stack extension memory begins. The WRP flag is a <i>sticky bit</i> (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR). Hardware reset clears the WRP flag.</p>
18	EOV	0	<p>Stack Extension Overflow Flag</p> <p>Set when a stack overflow occurs in Stack Extended mode. Extended stack overflow is recognized when a push operation is requested while $SP = SZ$ (Stack Size register), and the Extended mode is enabled by the SEN bit. The EOV flag is a <i>sticky bit</i> (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR). The transition of the EOV flag from zero to one causes a Priority Level 3 (Non-maskable) stack error exception. Hardware reset clears the EOV flag.</p>
17	EUN	0	<p>Stack Extension Underflow Flag</p> <p>Set when a stack underflow occurs in the Stack Extended mode. Stack extended underflow is recognized when a pull operation is requested, $SP = 0$, and the Extended mode is enabled by the SEN bit. The EUN flag is a <i>sticky bit</i> (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR). Transition of the EUN flag from zero to one causes a Priority Level 3 (Non-maskable) stack error exception. Hardware reset clears the EUN flag.</p> <p>NOTE: While the chip is in Extended Stack mode, the UF bit in the SP acts like a normal counter bit.</p>
16	XYS	0	<p>Stack Extension XY Select</p> <p>Determines if the stack extension is mapped onto the X memory space or onto the Y memory space. If YYS is clear, then the stack extension is mapped onto the X memory space. If YYS is set, the stack extension is mapped to the Y memory space. Hardware reset clears the YYS bit.</p>
15	ATE	0	<p>Address Trace Enable</p> <p>Enables Address Trace mode. The Address Trace mode is a debugging tool that reflects internal memory accesses at the external address lines. Refer to device-specific user's manuals and technical data sheets to determine if this feature is implemented for a specific device and how to use it during debugging. Hardware reset clears the ATE bit.</p>

Table 5-2 Operating Mode Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
14	APD	0	<p>Address Attribute Priority Disable Disables the priority assigned to the Address Attribute signals (AA0-AA3). When APD = 0 (default setting), the four Address Attribute signals each have a certain priority: AA3 has the highest priority, AA0 has the lowest priority. Therefore, only one AA signal can be active at one time. This allows continuous partitioning of external memory; however, certain functions, such as using the AA signals as additional address lines, require additional interface hardware. When APD = 1, the priority mechanism is disabled, allowing more than one AA signal to be active simultaneously. Therefore, the AA signals can be used as additional address lines without the need for additional interface hardware. To determine whether this feature is implemented for a particular device, refer to the user's manual and technical data sheets relating to that device. For details on the Address Attribute Registers, see Chapter 9, External Memory Interface (Port A). Hardware reset clears the APD bit.</p>
13	ABE	0	<p>Asynchronous Bus Arbitration Enable Eliminates the setup and hold time requirements (with respect to CLKOUT) for \overline{BB} and \overline{BG}, and substitutes a required non-overlap interval between the deassertion of one \overline{BG} input to a DSP56300 family device and the assertion of a second \overline{BG} input to a second DSP56300 family device on the same bus. When the ABE bit is set, the \overline{BG} and \overline{BB} inputs are synchronized. This synchronization causes a delay between a change in \overline{BG} or \overline{BB} until the receiving device actually accepts the change. Hardware reset clears the ABE bit.</p>
12	BRT	0	<p>Bus Release Timing Selects between fast or slow bus release. If BRT is cleared, a Fast Bus Release mode is selected (that is, no additional cycles are added to the access and \overline{BB} is not guaranteed to be the last Port A pin that is tri-stated at the end of the access). If BRT is set, a Slow Bus Release mode is selected (that is, an additional cycle is added to the access, and \overline{BB} is the last Port A pin that is tri-stated at the end of the access). Hardware reset clears the BRT bit. For details on the bus release modes and their applications, refer to Chapter 9.</p>
11	TAS	0	<p>\overline{TA} Synchronize Select Selects the synchronization method for the input Port A pin, \overline{TA} (Transfer Acknowledge). If TAS is cleared, you are responsible for asserting the \overline{TA} pin synchronized to the chip clock, as described in the device-specific technical data sheet. If TAS is set, the \overline{TA} input assertion is synchronized inside the chip, thus eliminating the need for an off-chip synchronizer. Note that the TAS bit has no effect when the TA pin is deasserted: you are responsible for deasserting the \overline{TA} pin (if additional wait states are desired) before the chip finishes inserting wait states as defined in the BCR (Bus Control Register). See Chapter 9 for details. Hardware reset clears the TAS bit</p>

Table 5-2 Operating Mode Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description										
10	BE	0	<p>Cache Burst Mode Enable Enables/Disables the Burst mode in the memory expansion port during an instruction cache miss. If the bit is cleared, the Burst mode is disabled and only one program word is fetched from the external memory when an instruction cache miss condition is detected. If the bit is set, the Burst mode is enabled, and up to four program words are fetched from the external memory when an instruction cache miss is detected. For details on the Burst mode, see Chapter 8, Instruction Cache. Hardware reset clears the BE bit.</p>										
9 – 8	CDP[1 – 0]	1	<p>Core-DMA Priority Specify the priority between core accesses and DMA accesses to the external bus. Following are the core-DMA priorities for these bits. The CDP[1 – 0] bits are set during hardware reset.</p> <table border="1"> <thead> <tr> <th>CDP1 – 0</th> <th>Core-DMA Priority</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Determined by comparing status register CP[1 – 0] to the active DMA channel priority</td> </tr> <tr> <td>01</td> <td>DMA accesses have higher priority than core accesses</td> </tr> <tr> <td>10</td> <td>DMA accesses have the same priority as the core accesses</td> </tr> <tr> <td>11</td> <td>DMA accesses have lower priority than the core accesses</td> </tr> </tbody> </table>	CDP1 – 0	Core-DMA Priority	00	Determined by comparing status register CP[1 – 0] to the active DMA channel priority	01	DMA accesses have higher priority than core accesses	10	DMA accesses have the same priority as the core accesses	11	DMA accesses have lower priority than the core accesses
			CDP1 – 0	Core-DMA Priority									
			00	Determined by comparing status register CP[1 – 0] to the active DMA channel priority									
			01	DMA accesses have higher priority than core accesses									
			10	DMA accesses have the same priority as the core accesses									
11	DMA accesses have lower priority than the core accesses												
00	Determined by comparing status register CP[1 – 0] to the active DMA channel priority												
01	DMA accesses have higher priority than core accesses												
10	DMA accesses have the same priority as the core accesses												
11	DMA accesses have lower priority than the core accesses												
7	MS	0	<p>Memory Switch Mode Allows some internal memory modules to be switched from Program RAM to data RAM (X, Y, or both) or <i>vice versa</i>. The MS bit is cleared during hardware reset.</p> <p>NOTES:</p> <ol style="list-style-type: none"> For some DSP56300 family chip products, program data placed into the Program RAM/Instruction Cache area changes its placement after the MS bit is set (that is, the Instruction Cache always uses the highest internal Program RAM addresses). For example, this is true in the DSP56301 but not in the DSP56307 or DSP56311. Check your device-specific user's manual. To ensure proper operation, place six NOP instructions after the instruction that changes the MS bit. To ensure proper operation, do not change the MS bit while the Instruction Cache is enabled (CE bit is set in SR). Actual memory configuration is device-specific; refer to the device-specific technical data sheets and user's manuals for implementation information. 										

Table 5-2 Operating Mode Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
6	(SD)	0	Stop Delay Mode Determines the length of the delay invoked when the core exits the Stop state. The STOP instruction suspends core processing indefinitely until a defined event occurs to restart it. If the Stop Delay (SD) mode bit is cleared, a 128 K clock cycle delay is invoked before a STOP instruction cycle continues. However, if the SD bit is set, the delay before the instruction cycle resumes is 16 clock cycles. The long delay allows a clock stabilization period for the internal clock to begin oscillating. When a stable external clock is used, the shorter delay allows faster start-up of the DSP56300 core. The SD bit is cleared during hardware reset.
5		0	Reserved Write to zero for future compatibility.
4	EBD	0	External Bus Disable Disables the external bus controller in order to reduce power consumption when external memories are not used. When the EBD bit is set, the external bus controller is disabled and external memory cannot be accessed. When the EBD bit is cleared, the external bus controller is enabled and external access can be performed. Hardware reset clears the EBD bit.
3 – 0	MD–MA	*	Chip Operating Mode Indicate the operating mode of the DSP56300 core. On hardware reset, these bits are loaded from the external mode select pins, MODD, MODC, MODB, and MODA, respectively. After the DSP56300 core leaves the Reset state, MD, MC, MB, and MA can be changed under program control. *After reset, these bits reflect the corresponding value of the mode input (that is, MODD, MODC, MODB, or MODA, respectively).

5.4.1.2 Status Register (SR)

The Status Register (SR) (**Figure 5-5**) is a 24-bit register that consists of the following three 8-bit special-purpose control registers:

- **Extended Mode Register (EMR) (SR[23 – 16]):** Defines the current system state of the processor. The EMR bits are affected by hardware reset, exception processing, DO FOREVER instructions, ENDDO (end current DO loop) instructions, BRKcc instructions, RTI (return from interrupt) instructions, TRAP instructions, and instructions that specify SR as their destination (for example, MOVEC). During hardware reset, all EMR bits are cleared.
- **Mode Register (MR) (SR[15 – 8]):** Defines the current system state of the processor. The MR bits are affected by hardware reset, exception processing, DO instructions, ENDDO (end current DO loop) instructions, RTI (return from interrupt) instructions, TRAP instructions, and instructions that directly reference

the MR (for example, ANDI, ORI, or instructions, such as MOVEC, that specify SR as the destination). During hardware reset, the interrupt mask bits are set and all other bits are cleared.

- Condition Code Register (CCR) (SR[7 – 0]): Defines the results of previous arithmetic computations. The CCR bits are affected by Data Arithmetic Logic Unit (Data ALU) operations, parallel move operations, instructions that directly reference the CCR (ORI and ANDI), and by instructions that specify SR as a destination (for example, MOVEC). Parallel move operations affect only the S and L bits of the CCR. During hardware reset, all CCR bits are cleared.

The SR is pushed onto the System Stack when:

- Program looping is initialized
- A JSR is performed, including long interrupts

The three 8-bit registers are defined within the SR primarily for compatibility with other Motorola DSPs. Bit definitions in the following paragraphs identify the bits within the SR and not within the subregister.

Extended Mode Register (EMR)								Mode Register (MR)								Condition Code Register (CCR)							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP1 – 0	RM	SM	CE			SA	FV	LF	DM	SC		S1 – 0	I1 – 0			S	L	E	U	N	Z	V	C

Reserved bit. Read as zero. Write with zero for future compatibility

Values after reset:

1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- | | | |
|---------------------------------|--------------------------------|-----------------------|
| CP1 - Core Priority Bit 1 | LF - DO-Loop Flag | S - Scaling Flag |
| CP0 - Core Priority Bit 0 | DM - Double Precision Multiply | L - Limit Flag |
| RM - Rounding Mode | SC - Sixteen-bit Compatibility | E - Extension Flag |
| SM - Arithmetic Saturation Mode | S1 - Scaling Mode Bit 1 | U - Unnormalized Flag |
| CE - Instruction Cache Enable | S0 - Scaling Mode Bit 0 | N - Negative Flag |
| SA - Sixteenth-Bit Arithmetic | I1 - Interrupt Mask Bit 1 | Z - Zero Flag |
| FV - DO-Forever Flag | I0 - Interrupt Mask Bit 0 | V - Overflow Flag |
| | | C - Carry Flag |

Figure 5-5. Status Register (SR)

Table 5-3 Status Register Bit Definitions

Bit Number	Bit Name	Reset Value	Description																																
23 – 22	CP[1 – 0]	1	<p>Core Priority Under the control of CDP[1 – 0] bits in the Operating Mode Register (OMR), the Core Priority bits, CP1 and CP0, specify the priority of core accesses to external memory. These bits are compared against the priority bits of the active DMA channel. If the core priority is greater than the DMA priority, the DMA waits for a free time slot on the external bus. If the core priority is less than the DMA priority, the core waits for a free time slot on the external bus. If the core priority equals the DMA priority, the core and DMA access the external bus in a round robin pattern (for example, ... P, X, Y, DMA, P, X, Y, ...). The core priority bits are set during hardware reset.</p>																																
			<table border="1"> <thead> <tr> <th>Priority Mode</th> <th>Core Priority</th> <th>DMA Priority</th> <th>OMR (CDP [1 – 0])</th> <th>SR (CP[1 – 0])</th> </tr> </thead> <tbody> <tr> <td rowspan="4">Dynamic</td> <td>0 (Lowest)</td> <td rowspan="4">Determined by DCRn (DPR[1 – 0]) for active DMA channel</td> <td>00</td> <td>00</td> </tr> <tr> <td>1</td> <td>00</td> <td>01</td> </tr> <tr> <td>2</td> <td>00</td> <td>10</td> </tr> <tr> <td>3 (Highest)</td> <td>00</td> <td>11</td> </tr> <tr> <td rowspan="3">Static</td> <td colspan="2">core < DMA</td> <td>01</td> <td>xx</td> </tr> <tr> <td colspan="2">core = DMA</td> <td>10</td> <td>xx</td> </tr> <tr> <td colspan="2">core > DMA</td> <td>11</td> <td>xx</td> </tr> </tbody> </table>	Priority Mode	Core Priority	DMA Priority	OMR (CDP [1 – 0])	SR (CP[1 – 0])	Dynamic	0 (Lowest)	Determined by DCRn (DPR[1 – 0]) for active DMA channel	00	00	1	00	01	2	00	10	3 (Highest)	00	11	Static	core < DMA		01	xx	core = DMA		10	xx	core > DMA		11	xx
			Priority Mode	Core Priority	DMA Priority	OMR (CDP [1 – 0])	SR (CP[1 – 0])																												
			Dynamic	0 (Lowest)	Determined by DCRn (DPR[1 – 0]) for active DMA channel	00	00																												
				1		00	01																												
				2		00	10																												
				3 (Highest)		00	11																												
Static	core < DMA		01	xx																															
	core = DMA		10	xx																															
	core > DMA		11	xx																															
21	RM	0	<p>Rounding Mode Selects the type of rounding performed by the Data ALU during arithmetic operations. If the bit is cleared, convergent rounding is selected. If the bit is set, twos-complement rounding is selected. The RM bit is cleared during hardware reset.</p>																																
20	SM	0	<p>Arithmetic Saturation Mode Selects automatic saturation on 48 bits for the results going to the accumulator. A special circuit inside the MAC unit performs the saturation. This bit provides an Arithmetic Saturation mode for algorithms that do not recognize or cannot take advantage of the extension accumulator. The SM bit is cleared during hardware reset.</p>																																

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
19	CE	0	<p>Cache Enable Enables/Disables the operation of the instruction cache controller. If the bit is set, the cache is enabled, and instructions are cached into and fetched from the internal Program RAM. If the bit is cleared, the cache is disabled and the DSP56300 core fetches instructions from external or internal program memory, according to the memory space table of the specific DSP56300 core-based device. The CE bit is cleared during a hardware reset.</p> <p>Note: To ensure proper operation, do not clear Cache Enable mode (CE bit in SR) while Burst mode is enabled (BE bit in OMR is set).</p>
18		0	<p>Reserved Bit Write to zero for future compatibility.</p>
17	SA	0	<p>Sixteen-bit Arithmetic Mode Enables the Sixteen-bit Arithmetic mode of operation. When SA is set, the core uses 16-bit operations instead of 24-bit operations. In this mode, 16-bit data is right-aligned in the 24-bit memory locations, registers, and 24-bit register portions. Shifting, limiting, rounding, arithmetic instructions, and moves are performed accordingly. For details on the operation of Sixteen-bit Arithmetic mode, see Chapter 3.1, Introduction. Hardware reset clears the SA bit.</p>
16	FV	0	<p>DO FOREVER Flag Set when a DO FOREVER loop executes. The FV flag, like the LF flag, is restored from the stack when a DO FOREVER loop terminates. Stacking and restoring the FV flag when initiating and exiting a DO FOREVER loop, respectively, allow the nesting of program loops. When returning from the long interrupt with an RTI instruction, the System Stack is pulled and the value of the FV bit is restored. Hardware reset clears the FV bit.</p>
15	LF	0	<p>DO Loop Flag Enables the detection of the end of a program loop. The LF is restored from stack when a program loop terminates. Stacking and restoring the LF when initiating and exiting a program loop, respectively, allow the nesting of program loops. When returning from the long interrupt with an RTI instruction, the System Stack is pulled and the LF bit value is restored. Hardware reset clears the LF bit.</p>

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
14	DM	0	<p>Double-Precision Multiply Mode Enables the operation of four multiply/MAC operations to implement a double precision algorithm. This algorithm multiplies two 48-bit operands with a 96-bit result. Clearing the DM bit disables the mode. The Double Precision Multiply mode is supported in order to maintain object code compatibility with devices in the DSP56000 family. For a more efficient way of executing double-precision multiply, refer to Chapter 3, Data Arithmetic Logic Unit</p> <p>In Double-Precision Multiply mode, the behavior of the four specific operations listed in the double-precision algorithm is modified. Therefore, do not use these operations (with those specific register combinations) in Double Precision Multiply mode for any purpose other than the double-precision multiply algorithm. All other Data ALU operations (or the four listed operations, but with other register combinations) can be used.</p> <p>The double-precision multiply algorithm uses the Y0 Register at all stages. Therefore, do not change Y0 when running the double-precision multiply algorithm. If the Data ALU must be used in an interrupt service routine, Y0 should be saved with other Data ALU registers to be used and restored before leaving the interrupt routine. The DM bit is cleared during a hardware reset.</p>
13	SC	0	<p>Sixteen-bit Compatibility Mode Enables full compatibility with object code written for the DSP56000 family. When the SC bit is set, MOVE operations to/from any of the following PCU registers clear the eight MSBs of the destination: LA, LC, SP, SSL, SSH, EP, SZ, VBA and SC. If the source is either the SR or OMR, then the eight MSBs of the destination are also cleared. If the destination is either the SR or OMR, then the eight MSBs of the destination are left unchanged. In order to change the value of one of the eight MSBs of the SR or OMR, clear the SC mode bit.</p> <p>The SC mode bit also affects the contents of the Loop Counter Register. If the SC bit is cleared (normal operation), then a loop count value of zero causes the loop body to be skipped, and a loop count value of \$FFFFFF causes the loop to execute the maximum number of $2^{24} - 1$ times. If the SC bit is set, a loop count value of zero causes the loop to be executed 2^{16} times, and a loop count value of \$FFFFFF causes the loop to be executed $2^{16} - 1$ times. The AGU also uses this bit. When SC is set, the 8 MSBs are ignored while checking whether the address is internal or external. Refer to the memory configuration chapter of the device-specific user's manual for a full description of the memory map when this bit is set. A read to/from the AGU registers clears the 8 MSBs.</p> <p>Note: Due to pipelining, a change in the SC bit takes effect only after three instruction cycles. Insert three NOP instructions after the instruction that changes the value of this bit to ensure proper operation.</p>
12		0	<p>Reserved Write to zero for future compatibility.</p>

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description				
11 – 10	S[1 – 0]	0	Scaling Mode The following table shows that the Scaling mode bits, S1 and S0, specify the scaling to be performed in the Data ALU shifter/limiter and the rounding position in the Data ALU MAC unit. The Shifter/limiter Scaling mode affects data read from the A or B accumulator registers out to the X-data bus (XDB) and Y-data bus (YDB). Different scaling modes can be used with the same program code to allow dynamic scaling. One application of dynamic scaling is to facilitate block floating-point arithmetic. The scaling mode also affects the MAC rounding position to maintain proper rounding when different portions of the accumulator registers are read out to the XDB and YDB. Scaling mode bits are cleared at the start of a long Interrupt Service Routine and during a hardware reset.				
			S1	S0	Scaling Mode	Rounding Bit	S Equation
			0	0	No scaling	23	$S = (A46 \text{ XOR } A45) \text{ OR } (B46 \text{ XOR } B45) \text{ OR } S \text{ (previous)}$
			0	1	Scale down	24	$S = (A47 \text{ XOR } A46) \text{ OR } (B7 \text{ XOR } B46) \text{ OR } S \text{ (previous)}$
			1	0	Scale up	22	$S = (A45 \text{ XOR } A44) \text{ OR } (B45 \text{ XOR } B44) \text{ OR } S \text{ (previous)}$
			1	1	Reserved	—	S undefined
9 – 8	I[1 – 0]	1	Interrupt Mask Reflects the current Interrupt Priority Level (IPL) of the processor and indicates the IPL needed for an interrupt source to interrupt the processor. The current IPL of the processor can be changed under software control. The interrupt mask bits are set during hardware reset, but not during software reset. For details about how I1 and I0 are automatically altered during a long interrupt, see Chapter 2, “Core Architecture Overview”.				
			Priority	I1	I0	Exceptions Permitted	Exceptions Masked
			Lowest	0	0	IPL 0, 1, 2, 3	None
				0	1	IPL 1, 2, 3	IPL 0
				1	0	IPL 2, 3	IPL 0, 1
			Highest	1	1	IPL 3	IPL 0, 1, 2

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description			
7	S	0	<p>Scaling Set when a result moves from accumulator A or B to the XDB or YDB buses (during an accumulator-to-memory or accumulator-to-register move) and remains set until explicitly cleared; that is, the S bit is a <i>sticky bit</i>. The logical equations of this bit are dependent on the Scaling mode. The scaling bit is set if the absolute value in the accumulator, before scaling, is ≥ 0.25 and < 0.75. This bit is cleared during a hardware reset.</p>			
6	L	0	<p>Limit Set if the overflow bit is set or if the data shifter/limiter circuits perform a limiting operation. In Arithmetic Saturation mode, the L bit is also set when an arithmetic saturation occurs in the Data ALU result; otherwise, it is not affected. The L bit is cleared only by a hardware reset or by an instruction that specifically clears it (that is, a <i>sticky bit</i>); this allows the L bit to be used as a latching overflow bit. The L bit is affected by data movement operations that read the A or B accumulator registers.</p>			
5	E	0	<p>Extension Indicates when the accumulator extension register is in use. This bit is cleared if all the bits of the integer portion of the 56-bit result are all ones or all zeros; otherwise, this bit is set. As shown below, the Scaling mode defines the integer portion. If the E bit is cleared, then the low-order fraction portion contains all the significant bits; the high-order integer portion is sign extension. In this case, the accumulator extension register can be ignored.</p>			
			S1	S0	Scaling Mode	Integer Portion
			0	0	No Scaling	Bits 55,54.....48,47
			0	1	Scale Down	Bits 55,54.....49,48
			1	0	Scale Up	Bits 55,54.....47,46
4	U	0	<p>Unnormalized Set if the two MSBs of the Most Significant Portion (MSP) of the result are identical; otherwise, this bit is cleared. The MSP portion of the A or B accumulators is defined by the Scaling mode. The U bit is computed as follows.</p>			
			S1	S0	Scaling Mode	U Bit Computation
			0	0	No Scaling	$U = \overline{(\text{Bit } 47 \text{ xor Bit } 46)}$
			0	1	Scale Down	$U = \overline{(\text{Bit } 48 \text{ xor Bit } 47)}$
			1	0	Scale Up	$U = \overline{(\text{Bit } 46 \text{ xor Bit } 45)}$
3	N	0	<p>Negative Set if the MSB of the result is set; otherwise, this bit is cleared.</p>			

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
2	Z	0	Zero Set if the result equals zero; otherwise, this bit is cleared.
1	V	0	Overflow Set if an arithmetic overflow occurs in the 56-bit result; otherwise, this bit is cleared. This bit indicates that the result cannot be represented in the accumulator register (that is, the register overflowed). In Arithmetic Saturation mode, an arithmetic overflow occurs if the Data ALU result is not representable in the accumulator without the extension part (that is, 48-bit accumulator or the 32-bit accumulator in Arithmetic Sixteen-bit mode).
0	C	0	Carry Set if a carry is generated by the MSB resulting from an addition operation. This bit is also set if a borrow is generated in a subtraction operation; otherwise, this bit is cleared. The carry or borrow is generated from bit 55 of the result. The C bit is also affected by bit manipulation, rotate, and shift instructions.

5.4.2 Stack and Stack Extension

The following registers control the operation of the System Stack:

- System Stack High (SSH) and System Stack Low (SSL) registers
- Stack Pointer (SP)
- Stack Counter (SC)
- Stack Size Register (SZ) (used for stack extension)
- Extension Pointer (EP) Register (used for stack extension)

The 24-bit stack Extension Pointer (EP) register points to the stack extension in data memory whenever the stack extension is enabled and move operations to/from the on-chip hardware stack are needed. The EP register is located in the Address Generation Unit (AGU). For details, refer to **Chapter 4, Address Generation Unit**.

5.4.3 System Stack Configuration and Operation Registers

The PCU hardware System Stack is a 16-level by 48-bit separate internal memory that stores the PC and SR contents during subroutine calls and long interrupts. For hardware loops, the System Stack also automatically stores the contents of the LC and LA registers. All other data and control register contents can be stored in the System Stack via software control. Each location in the System Stack is addressable as two 24-bit registers, System Stack High (SSH) and System Stack Low (SSL), to which the four LSBs of the SP register collectively point. The System Stack is extended in the data memory in a space specified

by the stack control registers that monitor System Stack accesses. This hardware copies the Least Recently Used (LRU) location of the System Stack to data memory if the on-chip hardware stack is full and brings data from data memory when the on-chip hardware stack is empty. The main tasks performed by the System Stack include:

- Storing return address and status for subroutine calls (including long interrupts)
- Storing LA, LC, PC and SR for the hardware DO loops

When a subroutine is called (for example, using the JSR instruction), the return address (PC) is automatically stored in the SSH, and the status register (SR) is automatically stored in the SSL. When the RTS instruction initiates a return from the subroutine, the contents of the top location in the SSH are pulled and loaded into the PC, and the SR is not affected. When the RTI instruction initiates a return, the contents of the top location in the System Stack are pulled and loaded into the PC and SR (from SSH and SSL, respectively).

The System Stack is also used to implement no-overhead nested hardware DO loops. When a hardware DO loop is initiated (for example, by using the DO instruction), the previous contents of the LC Register are automatically stored in the SSL, the previous contents of the LA Register are automatically stored in the SSH, and the Stack Pointer (SP) is incremented. After the SP is incremented, the address of the loop's first instruction (PC) is also stored in the SSH, and the SR is stored in the SSL.

Note: Moving data to or from SSH increments or decrements the SP. The SSL does not affect the SP.

The System Stack can be extended into 24-bit wide X or Y data memory via control hardware that monitors the accesses to the System Stack. This extension is enabled by the Stack Extension Enable (SEN) bit in the chip Operating Mode Register (OMR). If this bit is cleared, the extension of the system stack is disabled, and the amount of nesting is determined by the limited size of the hardware stack (that is, 15 available locations; one location is unusable when the stack extension is disabled). The System Stack can accommodate up to 15 long interrupts, seven DO loops, or 15 JSRs, (or equivalent combinations of these) when its extension into data memory is disabled. When the System Stack limit is exceeded (either in Extended or in the Non-extended mode), a nonmaskable stack error interrupt occurs. By enabling the Stack extension, the limits on the level of nesting of subroutines or DO loops can be set to any desired value, subject to available internal/external memory. The XYs bit in the OMR Register determines whether X or Y data memory is used.

When enabled, a stack extension algorithm is applied to all accesses to the stack:

- If an explicit (for example, MOVE to SSH) or implicit (for example, JSR) push operation is performed, then the stack extension control logic examines the stack

after that push has finished. If the on-chip hardware stack is full, the least recently used word is moved into data memory to the location specified by the stack Extension Pointer (EP). The push is always made to the System Stack, and the extension memory space always has the least recently used words moved into it. This always moves one or two 48-bit items or two or four 24-bit words into the next extension memory space to which the stack Extension Pointer (EP) points.

- If an explicit (for example, MOVE from SSH) or implicit (for example, RTS) pull operation is performed, then the stack extension control logic examines the stack after that pull finishes. If the on-chip hardware stack is empty, then the stack is loaded from the location (in data memory) specified by the stack Extension Pointer (EP). For information on stack extension delays, see **Appendix A, *Instruction Timing and Restrictions***.
- External memory can be used for stack extension, and wait states affect it in the same way as they affect any other external memory access.

5.4.3.1 Stack Pointer (SP) Register

The 24-bit Stack Pointer (SP) register indicates the location of the top of the System Stack. The status of the System Stack is also indicated in SP when the Extended mode is disabled (underflow, empty, full, and overflow functions). The SP register is referenced implicitly by some instructions (for example, DO, JSR, RTI, etc.) or directly by the MOVEC instruction. The following paragraphs describe the SP register format, shown in **Figure 5-6**. The SP register is a 24-bit counter that addresses (selects) a 16-location stack with its four LSBs. The possible SP values in the Non-extended mode are shown in Table 4 on page 5-21 in the description for the SE bit.

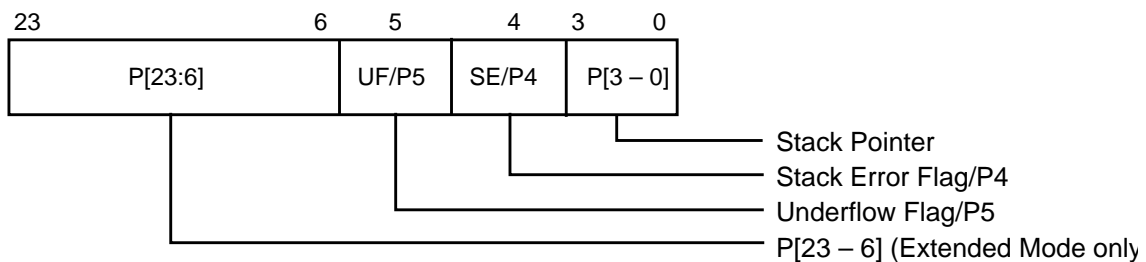


Figure 5-6. Stack Pointer (SP) Register Format

Immediately after hardware reset, the SP bits are cleared ($SP = 0$), so SP points to location 0, indicating that the System Stack is empty. Data is pushed onto the System Stack by incrementing the SP, then writing data to the location to which the SP points (the first push after reset is to location 1). An item is pulled off the stack by copying it from the location to which the SP points and then decrementing SP.

Table 5-4 Stack Pointer (SP) Register Bit Definitions

Bit Number	Bit Name	Reset Value	Description																																																																													
23 – 6	P[23 – 6]	0	P[23 – 6] In extended mode, these bits act as bits 6 through 23 of the Stack Pointer as part of a 24-bit up/down counter.																																																																													
5	UF	0	Underflow Flag / P5 In the Extended mode, UF acts as bit 5 of the Stack Pointer as part of a 24-bit up/down counter. In the Non-extended mode, UF is set when a stack underflow occurs. The stack UF is a <i>sticky bit</i> (that is, once the Stack Error flag is set, the UF does not change state until explicitly written by a MOVE instruction). The combination of “underflow = 1” and “stack error = 0” is an illegal combination and does not occur unless you force it. Also see the description for the Stack Error flag.																																																																													
4	SE	0	<p>Stack Error/P4 In Extended mode, SE acts as bit 4 of the Stack Pointer as part of a 24-bit up/down counter. In the Non-extended mode, it serves as the Stack Error (SE) flag that indicates that a stack error has occurred. The transition of the SE flag from zero to one in the Non-extended mode causes a Priority Level 3 (Non-maskable) stack error exception. When the non-extended stack is completely full, the SP reads 001111, and any operation that pushes data onto the stack causes a stack error exception. The SP reads 010000 (or 010001 if an implied double push occurs). Any implied pull operation with SP equal to zero causes a stack error exception, and the SP reads \$00003F (or \$00003E if an implied double pull occurs). In extended mode, the SP reads \$FFFFFF (or \$FFFFFFE if an implied double pull occurs). During such cases, the stack error bit is set as shown here.</p> <p>NOTE: The stack error flag is a <i>sticky bit</i> which, once set, remains set until you clear it. The overflow/underflow bit remains latched until the first move to SP executes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="7">SP Register Values in Non-extended Mode</th> </tr> <tr> <th>UF</th> <th>SE</th> <th>P3</th> <th>P2</th> <th>P1</th> <th>P0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>Stack Underflow condition after double pull</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>Stack Underflow condition</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Stack Empty (Reset); pull causes underflow</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>Stack Location 1</td> </tr> <tr> <td>0</td> <td>0</td> <td>*</td> <td>*</td> <td>*</td> <td>*</td> <td>Stack Locations 2-13</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>Stack Location 14</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>Stack Location 15; push causes overflow</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Stack Overflow condition</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>Stack Overflow condition after double push</td> </tr> </tbody> </table> <p>*Equal to Stack Locations 2 – 13</p>	SP Register Values in Non-extended Mode							UF	SE	P3	P2	P1	P0	Description	1	1	1	1	1	0	Stack Underflow condition after double pull	1	1	1	1	1	1	Stack Underflow condition	0	0	0	0	0	0	Stack Empty (Reset); pull causes underflow	0	0	0	0	0	1	Stack Location 1	0	0	*	*	*	*	Stack Locations 2-13	0	0	1	1	1	0	Stack Location 14	0	0	1	1	1	1	Stack Location 15; push causes overflow	0	1	0	0	0	0	Stack Overflow condition	0	1	0	0	0	1	Stack Overflow condition after double push
SP Register Values in Non-extended Mode																																																																																
UF	SE	P3	P2	P1	P0	Description																																																																										
1	1	1	1	1	0	Stack Underflow condition after double pull																																																																										
1	1	1	1	1	1	Stack Underflow condition																																																																										
0	0	0	0	0	0	Stack Empty (Reset); pull causes underflow																																																																										
0	0	0	0	0	1	Stack Location 1																																																																										
0	0	*	*	*	*	Stack Locations 2-13																																																																										
0	0	1	1	1	0	Stack Location 14																																																																										
0	0	1	1	1	1	Stack Location 15; push causes overflow																																																																										
0	1	0	0	0	0	Stack Overflow condition																																																																										
0	1	0	0	0	1	Stack Overflow condition after double push																																																																										

Table 5-4 Stack Pointer (SP) Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
3 – 0	P[3 – 0]	0	Stack Pointer Point to the 48-bit entry in the System Stack into which the last push was made. In the Non-extended mode, SP is a physical pointer, P[3 – 0] always having a value less than or equal to the highest physical location in the System Stack. In the extended mode, SP becomes a logical pointer, possibly having a value greater than the highest physical location in the System Stack. However, P[3 – 0] still point to the top of the stack, which is always in the System Stack.

5.4.3.2 Stack Counter (SC) Register

The 5-bit Stack Counter (SC) register monitors how many entries of the hardware stack are in use. The SC is a read/write register and is referenced implicitly by some instructions (for example, DO, JSR, and RTI) or directly by the MOVEC instruction. The stack counter register is cleared during hardware reset. During normal operation, do not write to the SC register. If a task switch is needed, writing a value greater than 14 or smaller than 2 automatically activates the stack extension control hardware. For proper operation, the SC should not be written with values greater than 16.

5.4.3.3 Stack Size (SZ) Register

The 24-bit Stack Size (SZ) register determines the number of data words allocated in memory for the stack in the Extended mode. The necessary value of the SZ register can be determined by $SZ = 15 + \text{software_buffer_size} / 2$, where the buffer size is the number of 24-bit words allocated for the stack extension in data memory. (Fifteen is the maximum number of 48-bit entries that can be occupied in the 16-entry hardware stack at any given time.) The extended stack overflow flag is generated when the value in SP equals the value in SZ and then a push is done.

Note: A stack exception can occur only when the stack is used in Non-extended mode.

The SZ register is not initialized during hardware reset, and must be set, using a MOVEC instruction, prior to enabling the stack extension.

5.4.4 Program, Loop, and Exception Processing Control

The code execution flow control is performed using four registers in the PCU:

- Program Counter (PC)
- Loop Address (LA) Register
- Loop Counter (LC) Register
- Vector Base Address (VBA) Register

5.4.4.1 Program Counter (PC) Register

The Program Counter Register (PC) is a special-purpose 24-bit address register that contains the address of instruction words in the program memory space. The PC can point to instructions, data operands, or addresses of operands. References to this register are always inherent and are implied by most instructions. The PC is stacked when hardware loops are initialized, when a JSR is performed, or when a long interrupt occurs. The PC is the source for the calculation of the real address in all position-independent instructions (such as the instruction BRA).

5.4.4.2 Loop Address (LA) Register

The contents of the 24-bit Loop Address (LA) register indicate the location of the last instruction word in a hardware loop. This register is stacked into the SSH by a DO instruction and is unstacked either by end-of-loop processing or by execution of ENDDO and BRKcc instructions. The LA register, a read/write register, is written by a DO instruction and read by the System Stack when the register is stacked.

5.4.4.3 Loop Counter (LC) Register

The Loop Counter (LC) register is a special read/write 24-bit counter that specifies the number of times a hardware program loop repeats, in the range of 0 to $(2^{24} - 1)$. This register is stacked into the SSL by a DO instruction and unstacked by end-of-loop processing or by execution of ENDDO and BRKcc instructions. The LC is also used in the REP instruction to specify how many times to repeat the repeated instruction.

5.4.4.4 Vector Base Address (VBA) Register

The Vector Base Address Register (VBA) is a 24-bit register. Eight of the bits VBA[7 – 0] are read-only and always cleared. The VBA is used as a base address of the interrupt vector table (discussed in **Chapter 2, Core Architecture Overview**). When a fast or long interrupt executes, VBA[7– 0] are driven from the program interrupt control unit, and bits 23–8 are driven from the VBA. The VBA Register is a read/write register that is referenced implicitly by interrupt processing or directly by the MOVEC instruction. The VBA is cleared during hardware reset.

Chapter 5

Program Control Unit

The Program Control Unit (PCU) of the DSP56300 family core coordinates execution of program instructions and instructions for processing interrupts and exceptions. The PCU also controls which of the five DSP56300 core processing states (Normal, Exception, Reset, Wait, or Stop) is currently selected. The PCU functions through a seven-stage instruction pipeline and several programmable registers. This chapter describes the PCU hardware, programming model, and instruction pipeline.

5.1 Overview

The PCU coordinates execution of instructions using three hardware blocks: the Program Address Generator (PAG), the Program Decode Controller (PDC), and the Program Interrupt Controller (PIC). These blocks perform the following functions:

- Fetch instructions
- Decode instructions
- Execute instructions
- Control hardware DO loops and REP
- Process interrupts and exceptions

Operation of the seven-stage pipeline depends on the current core processing state. The seven stages of the pipeline are as follows:

- Fetch-I
- Fetch-II
- Decode
- Address gen-I
- Address gen-II
- Execute-I
- Execute-II

To preserve current operation and status values while processing exceptions and interrupts, the PCU provides a System Stack to store current register contents before executing the exception/interrupt handler program. These contents are restored when control returns to the current program. In addition to these standard program flow-control resources, the PCU provides special support for hardware DO loops and an instruction REPEAT mechanism.

To perform its functions, the PCU uses a number of programmable registers. The organization of these registers forms the programming model for the PCU:

- General configuration and status:
 - Operating Mode Register (OMR)—24-bit, read/write
 - Status Register (SR)—24-bit, read/write
- System Stack configuration and operation:
 - System Stack (SS) register file—hardware stack, 48-bit × 16 locations, read/write
 - System Stack High (SSH) Register—24-bit, read/write
 - System Stack Low (SSL) Register—24-bit, read/write
 - Stack Pointer (SP) Register—24-bit, read/write
 - Stack Counter (SC) Register—5-bit, read/write
 - Stack Size (SZ) Register—24-bit, read/write

Note: The stack Extension Pointer (EP) Register is also used with the System Stack, but is physically part of the Address Generation Unit. For a description of this register, refer to **Chapter 4, *Address Generation Unit***.

- Program/Loop/Exception processing control
 - Program Counter (PC) Register—24-bit, read/write
 - Loop Address (LA) Register—24-bit, read/write
 - Loop Counter (LC) Register—24-bit, read/write
 - Vector Base Address (VBA) Register—24-bit, read/write

5.2 PCU Hardware Architecture

The three PCU hardware blocks are:

- Program Address Generator (PAG)—Contains all the hardware needed for program address generation, System Stack, and loop control
- Program Decode Controller (PDC)
 - Decodes the 24-bit instruction loaded into the instruction latch
 - Generates all signals for pipeline control
 - Performs required data transfers between the Data Arithmetic Logic Unit (Data ALU) and memory
- Program Interrupt Controller (PIC)—Arbitrates among all interrupt requests (internal interrupts and the five external interrupts: \overline{IRQA} , \overline{IRQB} , \overline{IRQC} , \overline{IRQD} and \overline{NMI}) and generates the appropriate interrupt vector address

Figure 5-1 shows a block diagram of the PCU.

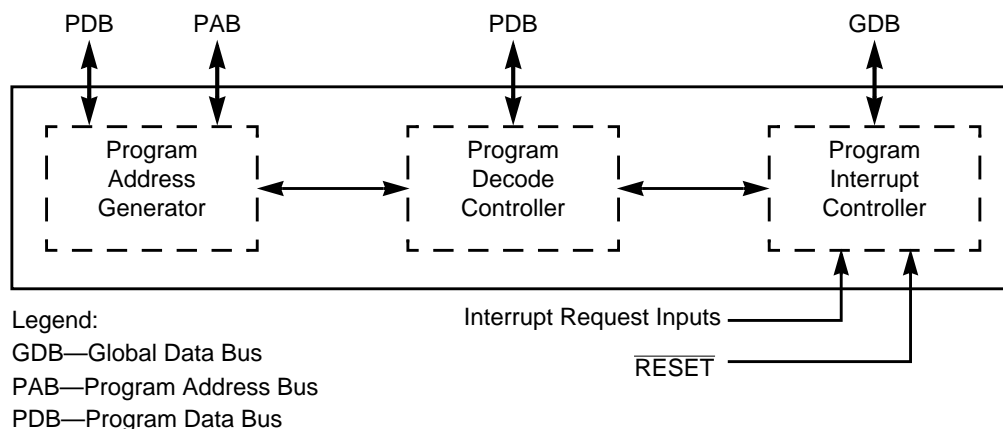


Figure 5-1. PCU Architecture

5.3 Instruction Pipeline

Within the seven-stage pipelined architecture of the PCU, instructions execute concurrently. Execution of a given pipeline stage for one instruction occurs concurrently with execution of other pipeline stages for other instructions. Table 5-1 and Figure 5-2 show that these stages include two fetch stages, one decode stage, two address generation stages, and two execute stages. The pipelined operation is essentially transparent, thus easing programmability. Transparency is achieved by means of interlock hardware present in every execution unit of the processor so that programs written for the DSP56000 family devices execute correctly on the DSP56300 core without any modification. However, code can be optimized to reduce interlocks and improve execution speed.

Table 5-1 Seven-Stage Pipeline

Pipeline Stage	Description
Fetch-I	<ul style="list-style-type: none">■ Address generation for Program Fetch■ Increment PC register
Fetch-II	<ul style="list-style-type: none">■ Instruction word read from memory
Decode	<ul style="list-style-type: none">■ Instruction Decode
AddressGen-I	<ul style="list-style-type: none">■ Address generation for Data Load/Store operations
AddressGen-II	<ul style="list-style-type: none">■ Address pointer update
Execute-I	<ul style="list-style-type: none">■ Read source operands to Multiplier and Adder■ Read source register for memory store operations■ Multiply■ Write destination register for memory load operations
Execute-II	<ul style="list-style-type: none">■ Read source operands for Adder if written by previous ALU operation■ Add■ Write Adder results to the Adder destination operand■ Write Multiplier results to the Multiplier destination operands

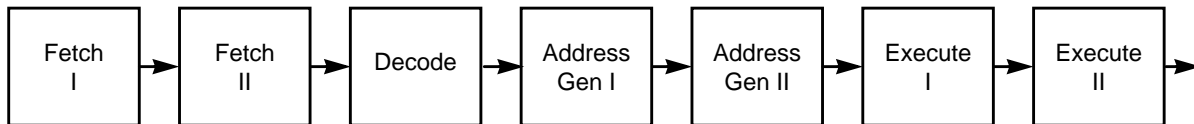


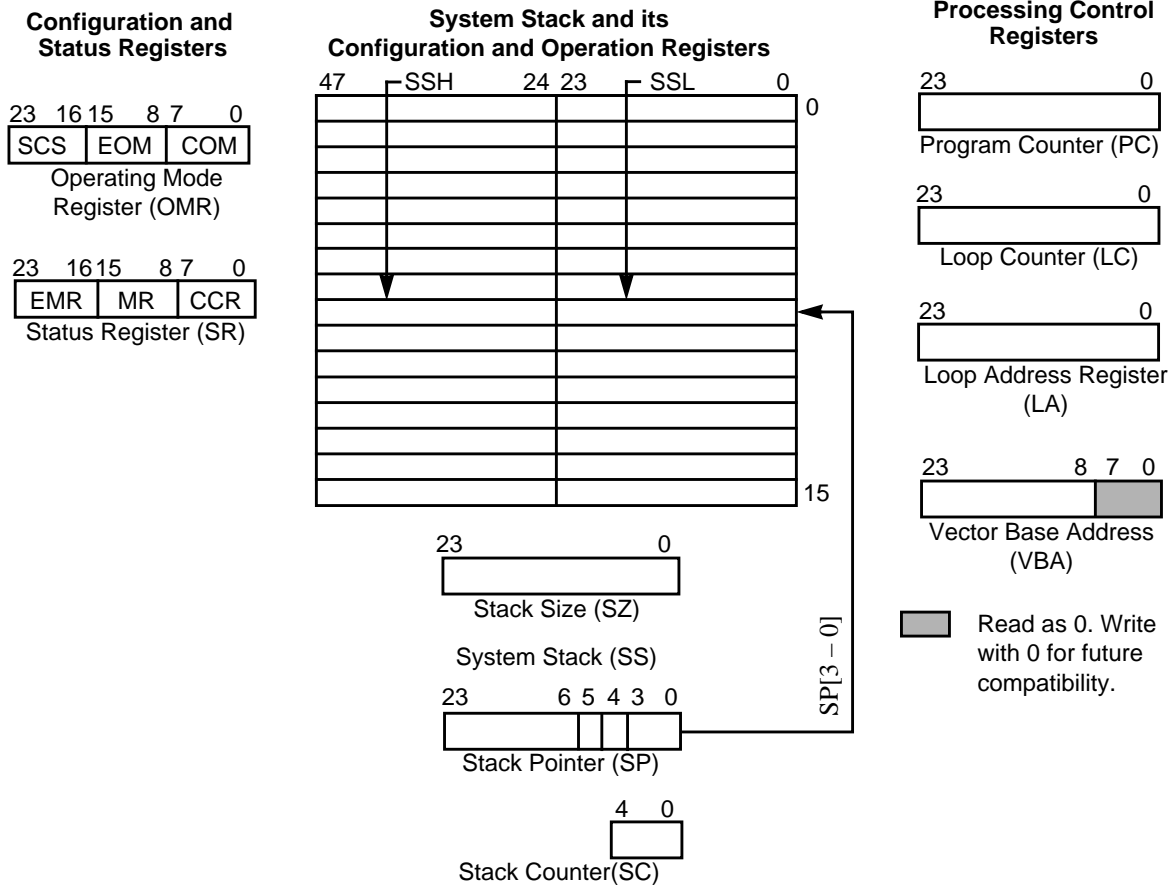
Figure 5-2. Seven-Stage Pipeline

5.4 Programming Model

The PCU programming model comprises three functional areas:

- Configuration and status registers
- System Stack configuration and operation registers
- Program/Loop/Exception processing control registers

Figure 5-3 shows the PCU programming model with the registers and the System Stack. The following paragraphs describe each register.



- Notes:**
1. The Extension Pointer (EP) Register is also used with the System Stack, but it is physically part of the Address Generation Unit (AGU).
 2. SSH and SSL point to the upper and lower halves of the stack location specified by the SP.

Figure 5-3. PCU Programming Model

5.4.1 Configuration and Status Registers

Note: Bits that are listed as reserved in the following sections can be defined for specific devices within the DSP56300 family. Refer to the device-specific user’s manual to determine whether a reserved bit is defined for that device.

The PCU contains two registers that configure and report the current status of the PCU:

- Operating Mode Register (OMR)
- Status Register (SR)

5.4.1.1 Operating Mode Register

The OMR (**Figure 5-4**) is a 24-bit register that is partitioned into the following three bytes:

- OMR[23 – 16], System Stack Control/Status (SCS) Byte: Controls and monitors the stack extension in the data memory. The SCS byte is referenced implicitly by some instructions—such as DO, JSR, and RTI—or directly by the MOVEC instruction.
- OMR[15 – 8], Extended Chip Operating Mode (EOM) Byte: Determines the operating mode of the chip. This byte is affected only by hardware reset and by instructions directly referencing the OMR (that is, ANDI, ORI, and other instructions, such as MOVEC, that specify OMR as a destination).
- OMR[7 – 0], Chip Operating Mode (COM) Byte: Determines the operating mode of the chip. This byte is affected only by hardware reset and by instructions directly referencing the OMR (that is, ANDI, ORI, and other instructions, such as MOVEC, that specify OMR as a destination). During hardware reset, the chip operating mode bits (MD, MC, MB, and MA) are loaded from the external mode select pins MODD, MODC, MODB, and MODA, respectively.

The following sections describe all defined bit functions; however, not all defined functions are implemented on all DSP56300 family devices. Always write non-implemented functions as zeros to ensure future compatibility. Refer to the latest device-specific user's manuals, technical data sheets, and technical bulletins for detailed information about implementation and usage for a particular device.

Stack Control/Status (SCS)						Extended Operating Mode (EOM)						Chip Operating Mode (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
PEN	MSW[1:0]	SEN	WRP	EOV	EUN	XYS	ATE	APD	ABE	BRT	TAS	BE	CDP[1:0]	MS	SD		EBD	MD	MC	MB	MA		

Reserved bit. Read as zero; write with zero for future compatibility

Values after reset:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	*	*	*	*
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

* After reset, these bits reflect the corresponding value of the mode input (that is, MODD, MODC, MODB, or MODA, respectively).

- | | | |
|------------------------------------|---|--------------------------|
| PEN—Patch Enable | ATE—Address Trace Enable | MS—Memory Switch Mode |
| MSW1—Memory Switch Configuration 1 | APD—Address Attribution Priority Disable | SD—Stop Delay Mode |
| MSW0—Memory Switch Configuration 0 | ABE—Asynch. Bus Arbitration Enable | |
| SEN—Stack Extension Enable | BRT—Bus Release Timing | EBD—External Bus Disable |
| WRP—Stack Extension Wrap Flag | TAS— $\overline{\text{TA}}$ Signal Synchronize Select | MD—Chip Operating Mode D |
| EOV—Stack Extension Overflow Flag | BE—Cache Burst Mode Enable | MC—Chip Operating Mode C |
| EUN—Stack Extension Underflow Flag | CDP1—Core-DMA Priority 1 | MB—Chip Operating Mode B |
| XYS—Stack Extension Space Select | CDP0—Core-DMA Priority 0 | MA—Chip Operating Mode A |

Figure 5-4. Operating Mode Register (OMR)

Table 5-2 Operating Mode Register Bit Definitions

Bit Number	Bit Name	Reset Value	Description
23	PEN	0	Patch Enable Enables/Disables the memory patch function, if implemented. Refer to the device-specific user's manual to determine whether and how this function is used on a specific device. Hardware reset clears this bit.
22 – 21	MSW	0	Memory Switch Configuration Determine what portion of the higher locations of internal X and Y data memory are switched to internal program memory when Memory Switch mode is enabled. Memory Switch mode allows reallocation of portions of X and Y data RAM as program RAM. Memory Switch mode is enabled when the Memory Switch bit, OMR[7] is set. For details on how much memory is switched, see the device-specific user's manual for a particular DSP56300 family device. The MSW bits are not available on all members of the DSP56300 family.
20	SEN	0	Stack Extension Enable Enables/ Disables the stack extension in data memory. If SEN is set, the extension is enabled. Hardware reset clears this bit, so the default out of reset is a disabled stack extension.

Table 5-2 Operating Mode Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
19	WRP	0	Stack Extension Wrap Flag During the debugging phase of the software development, this flag can be used to evaluate and increase the speed of software-implemented algorithms. WRP is set when copying from the on-chip hardware stack (System Stack Register file) to the stack extension memory begins. The WRP flag is a <i>sticky bit</i> (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR). Hardware reset clears the WRP flag.
18	EOV	0	Stack Extension Overflow Flag Set when a stack overflow occurs in Stack Extended mode. Extended stack overflow is recognized when a push operation is requested while $SP = SZ$ (Stack Size register), and the Extended mode is enabled by the SEN bit. The EOV flag is a <i>sticky bit</i> (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR). The transition of the EOV flag from zero to one causes a Priority Level 3 (Non-maskable) stack error exception. Hardware reset clears the EOV flag.
17	EUN	0	Stack Extension Underflow Flag Set when a stack underflow occurs in the Stack Extended mode. Stack extended underflow is recognized when a pull operation is requested, $SP = 0$, and the Extended mode is enabled by the SEN bit. The EUN flag is a <i>sticky bit</i> (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR). Transition of the EUN flag from zero to one causes a Priority Level 3 (Non-maskable) stack error exception. Hardware reset clears the EUN flag. NOTE: While the chip is in Extended Stack mode, the UF bit in the SP acts like a normal counter bit.
16	XYS	0	Stack Extension XY Select Determines if the stack extension is mapped onto the X memory space or onto the Y memory space. If YYS is clear, then the stack extension is mapped onto the X memory space. If YYS is set, the stack extension is mapped to the Y memory space. Hardware reset clears the YYS bit.
15	ATE	0	Address Trace Enable Enables Address Trace mode. The Address Trace mode is a debugging tool that reflects internal memory accesses at the external address lines. Refer to device-specific user's manuals and technical data sheets to determine if this feature is implemented for a specific device and how to use it during debugging. Hardware reset clears the ATE bit.

Table 5-2 Operating Mode Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
14	APD	0	<p>Address Attribute Priority Disable Disables the priority assigned to the Address Attribute signals (AA0-AA3). When APD = 0 (default setting), the four Address Attribute signals each have a certain priority: AA3 has the highest priority, AA0 has the lowest priority. Therefore, only one AA signal can be active at one time. This allows continuous partitioning of external memory; however, certain functions, such as using the AA signals as additional address lines, require additional interface hardware. When APD = 1, the priority mechanism is disabled, allowing more than one AA signal to be active simultaneously. Therefore, the AA signals can be used as additional address lines without the need for additional interface hardware. To determine whether this feature is implemented for a particular device, refer to the user's manual and technical data sheets relating to that device. For details on the Address Attribute Registers, see Chapter 9, External Memory Interface (Port A). Hardware reset clears the APD bit.</p>
13	ABE	0	<p>Asynchronous Bus Arbitration Enable Eliminates the setup and hold time requirements (with respect to CLKOUT) for \overline{BB} and \overline{BG}, and substitutes a required non-overlap interval between the deassertion of one \overline{BG} input to a DSP56300 family device and the assertion of a second \overline{BG} input to a second DSP56300 family device on the same bus. When the ABE bit is set, the \overline{BG} and \overline{BB} inputs are synchronized. This synchronization causes a delay between a change in \overline{BG} or \overline{BB} until the receiving device actually accepts the change. Hardware reset clears the ABE bit.</p>
12	BRT	0	<p>Bus Release Timing Selects between fast or slow bus release. If BRT is cleared, a Fast Bus Release mode is selected (that is, no additional cycles are added to the access and \overline{BB} is not guaranteed to be the last Port A pin that is tri-stated at the end of the access). If BRT is set, a Slow Bus Release mode is selected (that is, an additional cycle is added to the access, and \overline{BB} is the last Port A pin that is tri-stated at the end of the access). Hardware reset clears the BRT bit. For details on the bus release modes and their applications, refer to Chapter 9.</p>
11	TAS	0	<p>\overline{TA} Synchronize Select Selects the synchronization method for the input Port A pin, \overline{TA} (Transfer Acknowledge). If TAS is cleared, you are responsible for asserting the \overline{TA} pin synchronized to the chip clock, as described in the device-specific technical data sheet. If TAS is set, the \overline{TA} input assertion is synchronized inside the chip, thus eliminating the need for an off-chip synchronizer. Note that the TAS bit has no effect when the TA pin is deasserted: you are responsible for deasserting the \overline{TA} pin (if additional wait states are desired) before the chip finishes inserting wait states as defined in the BCR (Bus Control Register). See Chapter 9 for details. Hardware reset clears the TAS bit</p>

Table 5-2 Operating Mode Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description										
10	BE	0	<p>Cache Burst Mode Enable Enables/Disables the Burst mode in the memory expansion port during an instruction cache miss. If the bit is cleared, the Burst mode is disabled and only one program word is fetched from the external memory when an instruction cache miss condition is detected. If the bit is set, the Burst mode is enabled, and up to four program words are fetched from the external memory when an instruction cache miss is detected. For details on the Burst mode, see Chapter 8, Instruction Cache. Hardware reset clears the BE bit.</p>										
9 – 8	CDP[1 – 0]	1	<p>Core-DMA Priority Specify the priority between core accesses and DMA accesses to the external bus. Following are the core-DMA priorities for these bits. The CDP[1 – 0] bits are set during hardware reset.</p> <table border="1"> <thead> <tr> <th>CDP1 – 0</th> <th>Core-DMA Priority</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Determined by comparing status register CP[1 – 0] to the active DMA channel priority</td> </tr> <tr> <td>01</td> <td>DMA accesses have higher priority than core accesses</td> </tr> <tr> <td>10</td> <td>DMA accesses have the same priority as the core accesses</td> </tr> <tr> <td>11</td> <td>DMA accesses have lower priority than the core accesses</td> </tr> </tbody> </table>	CDP1 – 0	Core-DMA Priority	00	Determined by comparing status register CP[1 – 0] to the active DMA channel priority	01	DMA accesses have higher priority than core accesses	10	DMA accesses have the same priority as the core accesses	11	DMA accesses have lower priority than the core accesses
			CDP1 – 0	Core-DMA Priority									
			00	Determined by comparing status register CP[1 – 0] to the active DMA channel priority									
			01	DMA accesses have higher priority than core accesses									
			10	DMA accesses have the same priority as the core accesses									
11	DMA accesses have lower priority than the core accesses												
00	Determined by comparing status register CP[1 – 0] to the active DMA channel priority												
01	DMA accesses have higher priority than core accesses												
10	DMA accesses have the same priority as the core accesses												
11	DMA accesses have lower priority than the core accesses												
7	MS	0	<p>Memory Switch Mode Allows some internal memory modules to be switched from Program RAM to data RAM (X, Y, or both) or <i>vice versa</i>. The MS bit is cleared during hardware reset.</p> <p>NOTES:</p> <ol style="list-style-type: none"> For some DSP56300 family chip products, program data placed into the Program RAM/Instruction Cache area changes its placement after the MS bit is set (that is, the Instruction Cache always uses the highest internal Program RAM addresses). For example, this is true in the DSP56301 but not in the DSP56307 or DSP56311. Check your device-specific user's manual. To ensure proper operation, place six NOP instructions after the instruction that changes the MS bit. To ensure proper operation, do not change the MS bit while the Instruction Cache is enabled (CE bit is set in SR). Actual memory configuration is device-specific; refer to the device-specific technical data sheets and user's manuals for implementation information. 										

Table 5-2 Operating Mode Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
6	(SD)	0	Stop Delay Mode Determines the length of the delay invoked when the core exits the Stop state. The STOP instruction suspends core processing indefinitely until a defined event occurs to restart it. If the Stop Delay (SD) mode bit is cleared, a 128 K clock cycle delay is invoked before a STOP instruction cycle continues. However, if the SD bit is set, the delay before the instruction cycle resumes is 16 clock cycles. The long delay allows a clock stabilization period for the internal clock to begin oscillating. When a stable external clock is used, the shorter delay allows faster start-up of the DSP56300 core. The SD bit is cleared during hardware reset.
5		0	Reserved Write to zero for future compatibility.
4	EBD	0	External Bus Disable Disables the external bus controller in order to reduce power consumption when external memories are not used. When the EBD bit is set, the external bus controller is disabled and external memory cannot be accessed. When the EBD bit is cleared, the external bus controller is enabled and external access can be performed. Hardware reset clears the EBD bit.
3 – 0	MD–MA	*	Chip Operating Mode Indicate the operating mode of the DSP56300 core. On hardware reset, these bits are loaded from the external mode select pins, MODD, MODC, MODB, and MODA, respectively. After the DSP56300 core leaves the Reset state, MD, MC, MB, and MA can be changed under program control. *After reset, these bits reflect the corresponding value of the mode input (that is, MODD, MODC, MODB, or MODA, respectively).

5.4.1.2 Status Register (SR)

The Status Register (SR) (**Figure 5-5**) is a 24-bit register that consists of the following three 8-bit special-purpose control registers:

- **Extended Mode Register (EMR) (SR[23 – 16]):** Defines the current system state of the processor. The EMR bits are affected by hardware reset, exception processing, DO FOREVER instructions, ENDDO (end current DO loop) instructions, BRKcc instructions, RTI (return from interrupt) instructions, TRAP instructions, and instructions that specify SR as their destination (for example, MOVEC). During hardware reset, all EMR bits are cleared.
- **Mode Register (MR) (SR[15 – 8]):** Defines the current system state of the processor. The MR bits are affected by hardware reset, exception processing, DO instructions, ENDDO (end current DO loop) instructions, RTI (return from interrupt) instructions, TRAP instructions, and instructions that directly reference

the MR (for example, ANDI, ORI, or instructions, such as MOVEC, that specify SR as the destination). During hardware reset, the interrupt mask bits are set and all other bits are cleared.

- Condition Code Register (CCR) (SR[7 – 0]): Defines the results of previous arithmetic computations. The CCR bits are affected by Data Arithmetic Logic Unit (Data ALU) operations, parallel move operations, instructions that directly reference the CCR (ORI and ANDI), and by instructions that specify SR as a destination (for example, MOVEC). Parallel move operations affect only the S and L bits of the CCR. During hardware reset, all CCR bits are cleared.

The SR is pushed onto the System Stack when:

- Program looping is initialized
- A JSR is performed, including long interrupts

The three 8-bit registers are defined within the SR primarily for compatibility with other Motorola DSPs. Bit definitions in the following paragraphs identify the bits within the SR and not within the subregister.

Extended Mode Register (EMR)								Mode Register (MR)								Condition Code Register (CCR)							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP1 – 0	RM	SM	CE			SA	FV	LF	DM	SC		S1 – 0	I1 – 0			S	L	E	U	N	Z	V	C

 Reserved bit. Read as zero. Write with zero for future compatibility

Values after reset:

1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- | | | |
|---------------------------------|--------------------------------|-----------------------|
| CP1 - Core Priority Bit 1 | LF - DO-Loop Flag | S - Scaling Flag |
| CP0 - Core Priority Bit 0 | DM - Double Precision Multiply | L - Limit Flag |
| RM - Rounding Mode | SC - Sixteen-bit Compatibility | E - Extension Flag |
| SM - Arithmetic Saturation Mode | S1 - Scaling Mode Bit 1 | U - Unnormalized Flag |
| CE - Instruction Cache Enable | S0 - Scaling Mode Bit 0 | N - Negative Flag |
| SA - Sixteenth-Bit Arithmetic | I1 - Interrupt Mask Bit 1 | Z - Zero Flag |
| FV - DO-Forever Flag | I0 - Interrupt Mask Bit 0 | V - Overflow Flag |
| | | C - Carry Flag |

Figure 5-5. Status Register (SR)

Table 5-3 Status Register Bit Definitions

Bit Number	Bit Name	Reset Value	Description																																
23 – 22	CP[1 – 0]	1	<p>Core Priority Under the control of CDP[1 – 0] bits in the Operating Mode Register (OMR), the Core Priority bits, CP1 and CP0, specify the priority of core accesses to external memory. These bits are compared against the priority bits of the active DMA channel. If the core priority is greater than the DMA priority, the DMA waits for a free time slot on the external bus. If the core priority is less than the DMA priority, the core waits for a free time slot on the external bus. If the core priority equals the DMA priority, the core and DMA access the external bus in a round robin pattern (for example, ... P, X, Y, DMA, P, X, Y, ...). The core priority bits are set during hardware reset.</p>																																
			<table border="1"> <thead> <tr> <th>Priority Mode</th> <th>Core Priority</th> <th>DMA Priority</th> <th>OMR (CDP [1 – 0])</th> <th>SR (CP[1 – 0])</th> </tr> </thead> <tbody> <tr> <td rowspan="4">Dynamic</td> <td>0 (Lowest)</td> <td rowspan="4">Determined by DCRn (DPR[1 – 0]) for active DMA channel</td> <td>00</td> <td>00</td> </tr> <tr> <td>1</td> <td>00</td> <td>01</td> </tr> <tr> <td>2</td> <td>00</td> <td>10</td> </tr> <tr> <td>3 (Highest)</td> <td>00</td> <td>11</td> </tr> <tr> <td rowspan="3">Static</td> <td colspan="2">core < DMA</td> <td>01</td> <td>xx</td> </tr> <tr> <td colspan="2">core = DMA</td> <td>10</td> <td>xx</td> </tr> <tr> <td colspan="2">core > DMA</td> <td>11</td> <td>xx</td> </tr> </tbody> </table>	Priority Mode	Core Priority	DMA Priority	OMR (CDP [1 – 0])	SR (CP[1 – 0])	Dynamic	0 (Lowest)	Determined by DCRn (DPR[1 – 0]) for active DMA channel	00	00	1	00	01	2	00	10	3 (Highest)	00	11	Static	core < DMA		01	xx	core = DMA		10	xx	core > DMA		11	xx
			Priority Mode	Core Priority	DMA Priority	OMR (CDP [1 – 0])	SR (CP[1 – 0])																												
			Dynamic	0 (Lowest)	Determined by DCRn (DPR[1 – 0]) for active DMA channel	00	00																												
				1		00	01																												
				2		00	10																												
				3 (Highest)		00	11																												
Static	core < DMA		01	xx																															
	core = DMA		10	xx																															
	core > DMA		11	xx																															
21	RM	0	<p>Rounding Mode Selects the type of rounding performed by the Data ALU during arithmetic operations. If the bit is cleared, convergent rounding is selected. If the bit is set, twos-complement rounding is selected. The RM bit is cleared during hardware reset.</p>																																
20	SM	0	<p>Arithmetic Saturation Mode Selects automatic saturation on 48 bits for the results going to the accumulator. A special circuit inside the MAC unit performs the saturation. This bit is provides an Arithmetic Saturation mode for algorithms that do not recognize or cannot take advantage of the extension accumulator. The SM bit is cleared during hardware reset.</p>																																

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
19	CE	0	<p>Cache Enable Enables/Disables the operation of the instruction cache controller. If the bit is set, the cache is enabled, and instructions are cached into and fetched from the internal Program RAM. If the bit is cleared, the cache is disabled and the DSP56300 core fetches instructions from external or internal program memory, according to the memory space table of the specific DSP56300 core-based device. The CE bit is cleared during a hardware reset.</p> <p>Note: To ensure proper operation, do not clear Cache Enable mode (CE bit in SR) while Burst mode is enabled (BE bit in OMR is set).</p>
18		0	<p>Reserved Bit Write to zero for future compatibility.</p>
17	SA	0	<p>Sixteen-bit Arithmetic Mode Enables the Sixteen-bit Arithmetic mode of operation. When SA is set, the core uses 16-bit operations instead of 24-bit operations. In this mode, 16-bit data is right-aligned in the 24-bit memory locations, registers, and 24-bit register portions. Shifting, limiting, rounding, arithmetic instructions, and moves are performed accordingly. For details on the operation of Sixteen-bit Arithmetic mode, see Chapter 3.1, Introduction. Hardware reset clears the SA bit.</p>
16	FV	0	<p>DO FOREVER Flag Set when a DO FOREVER loop executes. The FV flag, like the LF flag, is restored from the stack when a DO FOREVER loop terminates. Stacking and restoring the FV flag when initiating and exiting a DO FOREVER loop, respectively, allow the nesting of program loops. When returning from the long interrupt with an RTI instruction, the System Stack is pulled and the value of the FV bit is restored. Hardware reset clears the FV bit.</p>
15	LF	0	<p>DO Loop Flag Enables the detection of the end of a program loop. The LF is restored from stack when a program loop terminates. Stacking and restoring the LF when initiating and exiting a program loop, respectively, allow the nesting of program loops. When returning from the long interrupt with an RTI instruction, the System Stack is pulled and the LF bit value is restored. Hardware reset clears the LF bit.</p>

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
14	DM	0	<p>Double-Precision Multiply Mode Enables the operation of four multiply/MAC operations to implement a double precision algorithm. This algorithm multiplies two 48-bit operands with a 96-bit result. Clearing the DM bit disables the mode. The Double Precision Multiply mode is supported in order to maintain object code compatibility with devices in the DSP56000 family. For a more efficient way of executing double-precision multiply, refer to Chapter 3, Data Arithmetic Logic Unit</p> <p>In Double-Precision Multiply mode, the behavior of the four specific operations listed in the double-precision algorithm is modified. Therefore, do not use these operations (with those specific register combinations) in Double Precision Multiply mode for any purpose other than the double-precision multiply algorithm. All other Data ALU operations (or the four listed operations, but with other register combinations) can be used.</p> <p>The double-precision multiply algorithm uses the Y0 Register at all stages. Therefore, do not change Y0 when running the double-precision multiply algorithm. If the Data ALU must be used in an interrupt service routine, Y0 should be saved with other Data ALU registers to be used and restored before leaving the interrupt routine. The DM bit is cleared during a hardware reset.</p>
13	SC	0	<p>Sixteen-bit Compatibility Mode Enables full compatibility with object code written for the DSP56000 family. When the SC bit is set, MOVE operations to/from any of the following PCU registers clear the eight MSBs of the destination: LA, LC, SP, SSL, SSH, EP, SZ, VBA and SC. If the source is either the SR or OMR, then the eight MSBs of the destination are also cleared. If the destination is either the SR or OMR, then the eight MSBs of the destination are left unchanged. In order to change the value of one of the eight MSBs of the SR or OMR, clear the SC mode bit.</p> <p>The SC mode bit also affects the contents of the Loop Counter Register. If the SC bit is cleared (normal operation), then a loop count value of zero causes the loop body to be skipped, and a loop count value of \$FFFFFF causes the loop to execute the maximum number of $2^{24} - 1$ times. If the SC bit is set, a loop count value of zero causes the loop to be executed 2^{16} times, and a loop count value of \$FFFFFF causes the loop to be executed $2^{16} - 1$ times. The AGU also uses this bit. When SC is set, the 8 MSBs are ignored while checking whether the address is internal or external. Refer to the memory configuration chapter of the device-specific user's manual for a full description of the memory map when this bit is set. A read to/from the AGU registers clears the 8 MSBs.</p> <p>Note: Due to pipelining, a change in the SC bit takes effect only after three instruction cycles. Insert three NOP instructions after the instruction that changes the value of this bit to ensure proper operation.</p>
12		0	<p>Reserved Write to zero for future compatibility.</p>

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description				
11 – 10	S[1 – 0]	0	Scaling Mode The following table shows that the Scaling mode bits, S1 and S0, specify the scaling to be performed in the Data ALU shifter/limiter and the rounding position in the Data ALU MAC unit. The Shifter/limiter Scaling mode affects data read from the A or B accumulator registers out to the X-data bus (XDB) and Y-data bus (YDB). Different scaling modes can be used with the same program code to allow dynamic scaling. One application of dynamic scaling is to facilitate block floating-point arithmetic. The scaling mode also affects the MAC rounding position to maintain proper rounding when different portions of the accumulator registers are read out to the XDB and YDB. Scaling mode bits are cleared at the start of a long Interrupt Service Routine and during a hardware reset.				
			S1	S0	Scaling Mode	Rounding Bit	S Equation
			0	0	No scaling	23	S = (A46 XOR A45) OR (B46 XOR B45) OR S (previous)
			0	1	Scale down	24	S = (A47 XOR A46) OR (B7 XOR B46) OR S (previous)
			1	0	Scale up	22	S = (A45 XOR A44) OR (B45 XOR B44) OR S (previous)
			1	1	Reserved	—	S undefined
9 – 8	I[1 – 0]	1	Interrupt Mask Reflects the current Interrupt Priority Level (IPL) of the processor and indicates the IPL needed for an interrupt source to interrupt the processor. The current IPL of the processor can be changed under software control. The interrupt mask bits are set during hardware reset, but not during software reset. For details about how I1 and I0 are automatically altered during a long interrupt, see Chapter 2, “Core Architecture Overview”.				
			Priority	I1	I0	Exceptions Permitted	Exceptions Masked
			Lowest	0	0	IPL 0, 1, 2, 3	None
				0	1	IPL 1, 2, 3	IPL 0
				1	0	IPL 2, 3	IPL 0, 1
			Highest	1	1	IPL 3	IPL 0, 1, 2

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description			
7	S	0	<p>Scaling Set when a result moves from accumulator A or B to the XDB or YDB buses (during an accumulator-to-memory or accumulator-to-register move) and remains set until explicitly cleared; that is, the S bit is a <i>sticky bit</i>. The logical equations of this bit are dependent on the Scaling mode. The scaling bit is set if the absolute value in the accumulator, before scaling, is ≥ 0.25 and < 0.75. This bit is cleared during a hardware reset.</p>			
6	L	0	<p>Limit Set if the overflow bit is set or if the data shifter/limiter circuits perform a limiting operation. In Arithmetic Saturation mode, the L bit is also set when an arithmetic saturation occurs in the Data ALU result; otherwise, it is not affected. The L bit is cleared only by a hardware reset or by an instruction that specifically clears it (that is, a <i>sticky bit</i>); this allows the L bit to be used as a latching overflow bit. The L bit is affected by data movement operations that read the A or B accumulator registers.</p>			
5	E	0	<p>Extension Indicates when the accumulator extension register is in use. This bit is cleared if all the bits of the integer portion of the 56-bit result are all ones or all zeros; otherwise, this bit is set. As shown below, the Scaling mode defines the integer portion. If the E bit is cleared, then the low-order fraction portion contains all the significant bits; the high-order integer portion is sign extension. In this case, the accumulator extension register can be ignored.</p>			
			S1	S0	Scaling Mode	Integer Portion
			0	0	No Scaling	Bits 55,54.....48,47
			0	1	Scale Down	Bits 55,54.....49,48
			1	0	Scale Up	Bits 55,54.....47,46
4	U	0	<p>Unnormalized Set if the two MSBs of the Most Significant Portion (MSP) of the result are identical; otherwise, this bit is cleared. The MSP portion of the A or B accumulators is defined by the Scaling mode. The U bit is computed as follows.</p>			
			S1	S0	Scaling Mode	U Bit Computation
			0	0	No Scaling	$U = \overline{(\text{Bit } 47 \text{ xor Bit } 46)}$
			0	1	Scale Down	$U = \overline{(\text{Bit } 48 \text{ xor Bit } 47)}$
			1	0	Scale Up	$U = \overline{(\text{Bit } 46 \text{ xor Bit } 45)}$
3	N	0	<p>Negative Set if the MSB of the result is set; otherwise, this bit is cleared.</p>			

Table 5-3 Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
2	Z	0	Zero Set if the result equals zero; otherwise, this bit is cleared.
1	V	0	Overflow Set if an arithmetic overflow occurs in the 56-bit result; otherwise, this bit is cleared. This bit indicates that the result cannot be represented in the accumulator register (that is, the register overflowed). In Arithmetic Saturation mode, an arithmetic overflow occurs if the Data ALU result is not representable in the accumulator without the extension part (that is, 48-bit accumulator or the 32-bit accumulator in Arithmetic Sixteen-bit mode).
0	C	0	Carry Set if a carry is generated by the MSB resulting from an addition operation. This bit is also set if a borrow is generated in a subtraction operation; otherwise, this bit is cleared. The carry or borrow is generated from bit 55 of the result. The C bit is also affected by bit manipulation, rotate, and shift instructions.

5.4.2 Stack and Stack Extension

The following registers control the operation of the System Stack:

- System Stack High (SSH) and System Stack Low (SSL) registers
- Stack Pointer (SP)
- Stack Counter (SC)
- Stack Size Register (SZ) (used for stack extension)
- Extension Pointer (EP) Register (used for stack extension)

The 24-bit stack Extension Pointer (EP) register points to the stack extension in data memory whenever the stack extension is enabled and move operations to/from the on-chip hardware stack are needed. The EP register is located in the Address Generation Unit (AGU). For details, refer to **Chapter 4, Address Generation Unit**.

5.4.3 System Stack Configuration and Operation Registers

The PCU hardware System Stack is a 16-level by 48-bit separate internal memory that stores the PC and SR contents during subroutine calls and long interrupts. For hardware loops, the System Stack also automatically stores the contents of the LC and LA registers. All other data and control register contents can be stored in the System Stack via software control. Each location in the System Stack is addressable as two 24-bit registers, System Stack High (SSH) and System Stack Low (SSL), to which the four LSBs of the SP register collectively point. The System Stack is extended in the data memory in a space specified

by the stack control registers that monitor System Stack accesses. This hardware copies the Least Recently Used (LRU) location of the System Stack to data memory if the on-chip hardware stack is full and brings data from data memory when the on-chip hardware stack is empty. The main tasks performed by the System Stack include:

- Storing return address and status for subroutine calls (including long interrupts)
- Storing LA, LC, PC and SR for the hardware DO loops

When a subroutine is called (for example, using the JSR instruction), the return address (PC) is automatically stored in the SSH, and the status register (SR) is automatically stored in the SSL. When the RTS instruction initiates a return from the subroutine, the contents of the top location in the SSH are pulled and loaded into the PC, and the SR is not affected. When the RTI instruction initiates a return, the contents of the top location in the System Stack are pulled and loaded into the PC and SR (from SSH and SSL, respectively).

The System Stack is also used to implement no-overhead nested hardware DO loops. When a hardware DO loop is initiated (for example, by using the DO instruction), the previous contents of the LC Register are automatically stored in the SSL, the previous contents of the LA Register are automatically stored in the SSH, and the Stack Pointer (SP) is incremented. After the SP is incremented, the address of the loop's first instruction (PC) is also stored in the SSH, and the SR is stored in the SSL.

Note: Moving data to or from SSH increments or decrements the SP. The SSL does not affect the SP.

The System Stack can be extended into 24-bit wide X or Y data memory via control hardware that monitors the accesses to the System Stack. This extension is enabled by the Stack Extension Enable (SEN) bit in the chip Operating Mode Register (OMR). If this bit is cleared, the extension of the system stack is disabled, and the amount of nesting is determined by the limited size of the hardware stack (that is, 15 available locations; one location is unusable when the stack extension is disabled). The System Stack can accommodate up to 15 long interrupts, seven DO loops, or 15 JSRs, (or equivalent combinations of these) when its extension into data memory is disabled. When the System Stack limit is exceeded (either in Extended or in the Non-extended mode), a nonmaskable stack error interrupt occurs. By enabling the Stack extension, the limits on the level of nesting of subroutines or DO loops can be set to any desired value, subject to available internal/external memory. The XYs bit in the OMR Register determines whether X or Y data memory is used.

When enabled, a stack extension algorithm is applied to all accesses to the stack:

- If an explicit (for example, MOVE to SSH) or implicit (for example, JSR) push operation is performed, then the stack extension control logic examines the stack

after that push has finished. If the on-chip hardware stack is full, the least recently used word is moved into data memory to the location specified by the stack Extension Pointer (EP). The push is always made to the System Stack, and the extension memory space always has the least recently used words moved into it. This always moves one or two 48-bit items or two or four 24-bit words into the next extension memory space to which the stack Extension Pointer (EP) points.

- If an explicit (for example, MOVE from SSH) or implicit (for example, RTS) pull operation is performed, then the stack extension control logic examines the stack after that pull finishes. If the on-chip hardware stack is empty, then the stack is loaded from the location (in data memory) specified by the stack Extension Pointer (EP). For information on stack extension delays, see **Appendix A, *Instruction Timing and Restrictions***.
- External memory can be used for stack extension, and wait states affect it in the same way as they affect any other external memory access.

5.4.3.1 Stack Pointer (SP) Register

The 24-bit Stack Pointer (SP) register indicates the location of the top of the System Stack. The status of the System Stack is also indicated in SP when the Extended mode is disabled (underflow, empty, full, and overflow functions). The SP register is referenced implicitly by some instructions (for example, DO, JSR, RTI, etc.) or directly by the MOVEC instruction. The following paragraphs describe the SP register format, shown in **Figure 5-6**. The SP register is a 24-bit counter that addresses (selects) a 16-location stack with its four LSBs. The possible SP values in the Non-extended mode are shown in Table 4 on page 5-21 in the description for the SE bit.

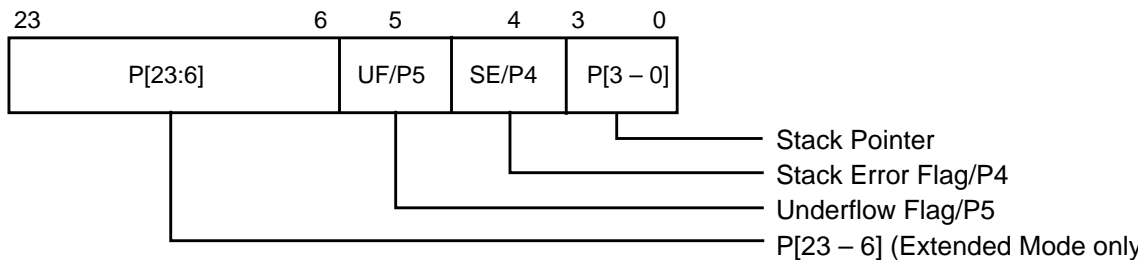


Figure 5-6. Stack Pointer (SP) Register Format

Immediately after hardware reset, the SP bits are cleared ($SP = 0$), so SP points to location 0, indicating that the System Stack is empty. Data is pushed onto the System Stack by incrementing the SP, then writing data to the location to which the SP points (the first push after reset is to location 1). An item is pulled off the stack by copying it from the location to which the SP points and then decrementing SP.

Table 5-4 Stack Pointer (SP) Register Bit Definitions

Bit Number	Bit Name	Reset Value	Description																																																																													
23 – 6	P[23 – 6]	0	P[23 – 6] In extended mode, these bits act as bits 6 through 23 of the Stack Pointer as part of a 24-bit up/down counter.																																																																													
5	UF	0	Underflow Flag / P5 In the Extended mode, UF acts as bit 5 of the Stack Pointer as part of a 24-bit up/down counter. In the Non-extended mode, UF is set when a stack underflow occurs. The stack UF is a <i>sticky bit</i> (that is, once the Stack Error flag is set, the UF does not change state until explicitly written by a MOVE instruction). The combination of “underflow = 1” and “stack error = 0” is an illegal combination and does not occur unless you force it. Also see the description for the Stack Error flag.																																																																													
4	SE	0	<p>Stack Error/P4 In Extended mode, SE acts as bit 4 of the Stack Pointer as part of a 24-bit up/down counter. In the Non-extended mode, it serves as the Stack Error (SE) flag that indicates that a stack error has occurred. The transition of the SE flag from zero to one in the Non-extended mode causes a Priority Level 3 (Non-maskable) stack error exception. When the non-extended stack is completely full, the SP reads 001111, and any operation that pushes data onto the stack causes a stack error exception. The SP reads 010000 (or 010001 if an implied double push occurs). Any implied pull operation with SP equal to zero causes a stack error exception, and the SP reads \$00003F (or \$00003E if an implied double pull occurs). In extended mode, the SP reads \$FFFFFF (or \$FFFFFFE if an implied double pull occurs). During such cases, the stack error bit is set as shown here.</p> <p>NOTE: The stack error flag is a <i>sticky bit</i> which, once set, remains set until you clear it. The overflow/underflow bit remains latched until the first move to SP executes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="7">SP Register Values in Non-extended Mode</th> </tr> <tr> <th>UF</th> <th>SE</th> <th>P3</th> <th>P2</th> <th>P1</th> <th>P0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>Stack Underflow condition after double pull</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>Stack Underflow condition</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Stack Empty (Reset); pull causes underflow</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>Stack Location 1</td> </tr> <tr> <td>0</td> <td>0</td> <td>*</td> <td>*</td> <td>*</td> <td>*</td> <td>Stack Locations 2-13</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>Stack Location 14</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>Stack Location 15; push causes overflow</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Stack Overflow condition</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>Stack Overflow condition after double push</td> </tr> </tbody> </table> <p>*Equal to Stack Locations 2 – 13</p>	SP Register Values in Non-extended Mode							UF	SE	P3	P2	P1	P0	Description	1	1	1	1	1	0	Stack Underflow condition after double pull	1	1	1	1	1	1	Stack Underflow condition	0	0	0	0	0	0	Stack Empty (Reset); pull causes underflow	0	0	0	0	0	1	Stack Location 1	0	0	*	*	*	*	Stack Locations 2-13	0	0	1	1	1	0	Stack Location 14	0	0	1	1	1	1	Stack Location 15; push causes overflow	0	1	0	0	0	0	Stack Overflow condition	0	1	0	0	0	1	Stack Overflow condition after double push
SP Register Values in Non-extended Mode																																																																																
UF	SE	P3	P2	P1	P0	Description																																																																										
1	1	1	1	1	0	Stack Underflow condition after double pull																																																																										
1	1	1	1	1	1	Stack Underflow condition																																																																										
0	0	0	0	0	0	Stack Empty (Reset); pull causes underflow																																																																										
0	0	0	0	0	1	Stack Location 1																																																																										
0	0	*	*	*	*	Stack Locations 2-13																																																																										
0	0	1	1	1	0	Stack Location 14																																																																										
0	0	1	1	1	1	Stack Location 15; push causes overflow																																																																										
0	1	0	0	0	0	Stack Overflow condition																																																																										
0	1	0	0	0	1	Stack Overflow condition after double push																																																																										

Table 5-4 Stack Pointer (SP) Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
3 – 0	P[3 – 0]	0	Stack Pointer Point to the 48-bit entry in the System Stack into which the last push was made. In the Non-extended mode, SP is a physical pointer, P[3 – 0] always having a value less than or equal to the highest physical location in the System Stack. In the extended mode, SP becomes a logical pointer, possibly having a value greater than the highest physical location in the System Stack. However, P[3 – 0] still point to the top of the stack, which is always in the System Stack.

5.4.3.2 Stack Counter (SC) Register

The 5-bit Stack Counter (SC) register monitors how many entries of the hardware stack are in use. The SC is a read/write register and is referenced implicitly by some instructions (for example, DO, JSR, and RTI) or directly by the MOVEC instruction. The stack counter register is cleared during hardware reset. During normal operation, do not write to the SC register. If a task switch is needed, writing a value greater than 14 or smaller than 2 automatically activates the stack extension control hardware. For proper operation, the SC should not be written with values greater than 16.

5.4.3.3 Stack Size (SZ) Register

The 24-bit Stack Size (SZ) register determines the number of data words allocated in memory for the stack in the Extended mode. The necessary value of the SZ register can be determined by $SZ = 15 + \text{software_buffer_size} / 2$, where the buffer size is the number of 24-bit words allocated for the stack extension in data memory. (Fifteen is the maximum number of 48-bit entries that can be occupied in the 16-entry hardware stack at any given time.) The extended stack overflow flag is generated when the value in SP equals the value in SZ and then a push is done.

Note: A stack exception can occur only when the stack is used in Non-extended mode.

The SZ register is not initialized during hardware reset, and must be set, using a MOVEC instruction, prior to enabling the stack extension.

5.4.4 Program, Loop, and Exception Processing Control

The code execution flow control is performed using four registers in the PCU:

- Program Counter (PC)
- Loop Address (LA) Register
- Loop Counter (LC) Register
- Vector Base Address (VBA) Register

5.4.4.1 Program Counter (PC) Register

The Program Counter Register (PC) is a special-purpose 24-bit address register that contains the address of instruction words in the program memory space. The PC can point to instructions, data operands, or addresses of operands. References to this register are always inherent and are implied by most instructions. The PC is stacked when hardware loops are initialized, when a JSR is performed, or when a long interrupt occurs. The PC is the source for the calculation of the real address in all position-independent instructions (such as the instruction BRA).

5.4.4.2 Loop Address (LA) Register

The contents of the 24-bit Loop Address (LA) register indicate the location of the last instruction word in a hardware loop. This register is stacked into the SSH by a DO instruction and is unstacked either by end-of-loop processing or by execution of ENDDO and BRKcc instructions. The LA register, a read/write register, is written by a DO instruction and read by the System Stack when the register is stacked.

5.4.4.3 Loop Counter (LC) Register

The Loop Counter (LC) register is a special read/write 24-bit counter that specifies the number of times a hardware program loop repeats, in the range of 0 to $(2^{24} - 1)$. This register is stacked into the SSL by a DO instruction and unstacked by end-of-loop processing or by execution of ENDDO and BRKcc instructions. The LC is also used in the REP instruction to specify how many times to repeat the repeated instruction.

5.4.4.4 Vector Base Address (VBA) Register

The Vector Base Address Register (VBA) is a 24-bit register. Eight of the bits VBA[7 – 0] are read-only and always cleared. The VBA is used as a base address of the interrupt vector table (discussed in **Chapter 2, Core Architecture Overview**). When a fast or long interrupt executes, VBA[7– 0] are driven from the program interrupt control unit, and bits 23–8 are driven from the VBA. The VBA Register is a read/write register that is referenced implicitly by interrupt processing or directly by the MOVEC instruction. The VBA is cleared during hardware reset.

