Welcome to E-XFL.COM

Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Obsolete
Type	Fixed Point
Interface	Host Interface, SSI, SCI
Clock Rate	100MHz
Non-Volatile Memory	ROM (576B)
On-Chip RAM	24kB
Voltage - I/O	3.30V
Voltage - Core	3.30V
Operating Temperature	-40°C ~ 105°C (TJ)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/xc56309ag100a

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

The PCU uses the following registers:

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

1.6.4 PLL and Clock Oscillator

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

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

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

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

1.6.5 JTAG TAP and OnCE Module

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

Table 4-2. Status Register Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
16	FV	0	**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 program loops to be nested. When returning from the long interrupt with an RTI instruction, the system stack is pulled and the value of the FV bit is restored.
15	LF	0	**Do Loop Flag** When a program loop is in progress, enables the detection of the end of the 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 program loops to be nested. When returning from the long interrupt with an RTI instruction, the System Stack is pulled and the LF bit value is restored.
14	DM	0	**Double-Precision Multiply Mode** Enables four multiply/MAC operations to implement a double-precision algorithm that multiplies two 48-bit operands with a 96-bit result. Clearing the DM bit disables the mode. The Double-Precision Multiply mode is supported to maintain object code compatibility with devices in the DSP56000 family. For a more efficient way of executing double precision multiply, refer to the chapter on the Data Arithmetic Logic Unit in the *DSP56300 Family Manual*. 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. 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 the interrupt routine terminates.
13	SC	0	**Sixteen-Bit Compatibility Mode** Affects addressing functionality, enabling full compatibility with object code written for the DSP56000 family. When SC 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. To change the value of one of the eight MSBs of the SR or OMR, clear SC. SC also affects the contents of the Loop Counter Register. If SC 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 execute 2^{16} times, and a loop count value of $FFFFFF causes the loop to execute $2^{16} - 1$ times. **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.

Table 4-2. Status Register Bit Definitions (Continued)

<table>
<tr><th>Bit Number</th><th>Bit Name</th><th>Reset Value</th><th>Description</th></tr>
<tr><td>5</td><td>E</td><td>1</td><td>Extension
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. 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. If the E bit is set, it indicates that the accumulator extension register is in use.
<table>
<tr><th>S1</th><th>S0</th><th>Scaling Mode</th><th>Integer Portion</th></tr>
<tr><td>0</td><td>0</td><td>No scaling</td><td>Bits 55–47</td></tr>
<tr><td>0</td><td>1</td><td>Scale down</td><td>Bits 55–48</td></tr>
<tr><td>1</td><td>0</td><td>Scale up</td><td>Bits 5–46</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td><td>Undefined</td></tr>
</table></td></tr>
<tr><td>4</td><td>U</td><td>0</td><td>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.
<table>
<tr><th>S1</th><th>S0</th><th>Scaling Mode</th><th>Integer Portion</th></tr>
<tr><td>0</td><td>0</td><td>No scaling</td><td>$U = \overline{(\text{Bit 47 XOR Bit 46})}$</td></tr>
<tr><td>0</td><td>1</td><td>Scale down</td><td>$U = \overline{(\text{Bit 48 XOR Bit 47})}$</td></tr>
<tr><td>1</td><td>0</td><td>Scale up</td><td>$U = \overline{(\text{Bit 46 XOR Bit 45})}$</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td><td>U undefined</td></tr>
</table></td></tr>
<tr><td>3</td><td>N</td><td>0</td><td>Negative
Set if the MSB of the result is set; otherwise, this bit is cleared.</td></tr>
<tr><td>2</td><td>Z</td><td>0</td><td>Zero
Set if the result equals zero; otherwise, this bit is cleared.</td></tr>
<tr><td>1</td><td>V</td><td>0</td><td>Overflow
Set if an arithmetic overflow occurs in the 56-bit result; otherwise, this bit is cleared. V 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).</td></tr>
<tr><td>0</td><td>C</td><td>0</td><td>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.</td></tr>
</table>

Table 4-8. Bus Control Register (BCR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
9–5	BA1W[4–0]	11111 (31 wait states)	**Bus Area 1 Wait State Control** Defines the number of wait states (one through 31) inserted into each external SRAM access to Area 1 (DRAM accesses are not affected by these bits). Area 1 is the area defined by AAR1. **Note:** Do not program the value of these bits as zero, since SRAM memory access requires at least one wait state. When four through seven wait states are selected, one additional wait state is inserted at the end of the access. When selecting eight or more wait states, two additional wait states are inserted at the end of the access. These trailing wait states increase the data hold time and the memory release time and do not increase the memory access time.
4–0	BA0W	11111 (31 wait states)	**Bus Area 0 Wait State Control** Defines the number of wait states (one through 31) inserted in each external SRAM access to Area 0 (DRAM accesses are not affected by these bits). Area 0 is the area defined by AAR0. **Note:** Do not program the value of these bits as zero, since SRAM memory access requires at least one wait state. When selecting four through seven wait states, one additional wait state is inserted at the end of the access. When selecting eight or more wait states, two additional wait states are inserted at the end of the access. These trailing wait states increase the data hold time and the memory release time and do not increase the memory access time.

4.6.2 DRAM Control Register (DCR)

The DRAM controller is an efficient interface to dynamic RAM devices in both random read/write cycles and Fast Access mode (Page mode). An internal DRAM controller controls the page hit circuit, the address multiplexing (row address and column address), the control signal generation ($\overline{\text{CAS}}$ and $\overline{\text{RAS}}$) and the refresh access generation ($\overline{\text{CAS}}$ before $\overline{\text{RAS}}$) for a variety of DRAM module sizes and access times. The internal DRAM controller configuration is determined by the DRAM Control Register (DCR). The DRAM Control Register (DCR) is a 24-bit read/write register that controls and configures the external DRAM accesses. The DCR bits are shown in **Figure 4-7**.

Note: To prevent improper device operation, you must guarantee that all the DCR bits except BSTR are not changed during a DRAM access.

Table 4-10. Address Attribute Registers (AAR[0–3]) Bit Definitions

Bit Number	Bit Name	Reset Value	Description
1–0	BAT	0	**Bus Access Type** Read/write bits that define the type of external memory (DRAM or SRAM) to access for the area defined by the BAC[11–0],BYEN, BXEN, and BPEN bits. The encoding of BAT[1–0] is: 00 = Reserved 01 = SRAM access 10 = DRAM access 11 = Reserved When the external access type is defined as a DRAM access (BAT[1–0] = 10), AA/$\overline{\text{RAS}}$ acts as a Row Address Strobe (RAS) signal. Otherwise, it acts as an Address Attribute signal. External accesses to the default area always execute as if BAT[1–0] = 01 (that is, SRAM access). If Port A is used for external accesses, the BAT bits in the AAR3–0 registers must be initialized to the SRAM access type (that is, BAT = 01) or to the DRAM access type (that is BAT = 10). To ensure proper operation of Port A, this initialization must occur even for an AAR register that is not used during any Port A access. Note that at reset, the BAT bits are initialized to 00.

4.7 DMA Control Registers 5–0 (DCR[5–0])

The DMA Control Registers (DCR[5–0]) are read/write registers that control the DMA operation for each of their respective channels. All DCR bits are cleared during processor reset.

23	22	21	20	19	18	17	16	15	14	13	12
DE	DIE	DTM2	DTM1	DTM0	DPR1	DPR0	DCON	DRS4	DRS3	DRS2	DRS1
11	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DRS0	D3D	DAM5	DAM4	DAM3	DAM2	DAM1	DAM0	DDS1	DDS0	DSS1	DSS0

Figure 4-9. DMA Control Register (DCR)

Table 4-11. DMA Control Register (DCR) Bit Definitions

Bit Number	Bit Name	Reset Value	Description
23	DE	0	**DMA Channel Enable** Enables the channel operation. Setting DE either triggers a single block DMA transfer in the DMA transfer mode that uses DE as a trigger or enables a single-block, single-line, or single-word DMA transfer in the transfer modes that use a requesting device as a trigger. DE is cleared by the end of DMA transfer in some of the transfer modes defined by the DTM bits. If software explicitly clears DE during a DMA operation, the channel operation stops only after the current DMA transfer completes (that is, the current word is stored into the destination).
22	DIE	0	**DMA Interrupt Enable** Generates a DMA interrupt at the end of a DMA block transfer after the counter is loaded with its preloaded value. A DMA interrupt is also generated when software explicitly clears $\overline{\text{DE}}$ during a DMA operation. Once asserted, a DMA interrupt request can be cleared only by the service of a DMA interrupt routine. To ensure that a new interrupt request is not generated, clear DIE while the DMA interrupt is serviced and before a new DMA request is generated at the end of a DMA block transfer—that is, at the beginning of the DMA channel interrupt service routine. When DIE is cleared, the DMA interrupt is disabled.

In GPIO mode, two additional registers (HDDR and HDR) are related to the HI08 peripheral. The separate receive and transmit data paths are double buffered for efficient, high speed asynchronous transfers. The host-side transmit data path (host writes) is also the DSP-side receive path; the host-side receive data path (host reads) is also the DSP-side transmit path. The Receive (RXH:RXM:RXL) and Transmit Data Registers (TXH:TXM:TXL) use the same host address. During host writes to these addresses, the data is transferred to the Transmit Data Registers while reads are performed from the Receive Data Registers.

6.4 Operation

The HI08 is a slave-only device, so the host is the master of all bus transfers. In host-to-DSP transfers, the host writes data to the Transmit Data Registers (TXH:TXM:TXL). In DSP-to-host transfers the host reads data from the Receive Data Registers (RXH:RXM:RXL). The DSP side has access only to the Host Receive Data Register (HRX) and the Host Transmit Data Register (HTX). Data automatically moves between the host-side data registers and the DSP-side data registers when it is available. This double-buffered mechanism allows for fast data transfers but creates a “pipeline” that can either stall communication (if the pipeline is either full or empty) or cause erroneous data transfers (new data to be overwritten or old data to be read twice). The HI08 port has several handshaking mechanisms to counter these buffering effects.

Suppose the host is writing several pieces of data to the HI08 port. The host first uses one of the handshaking protocols to determine whether any data previously written to the Transmit Data Registers (TXH:TXM:TXL) has successfully transferred to the DSP side. If the host-side Transmit Data Registers (TXH:TXM:TXL) are empty, the host writes the data to these registers. The transfer to the DSP-side Host Receive Data Register (HRX) occurs only if HRX is empty (that is, the DSP has read it). The DSP core then uses an appropriate handshaking protocol to move data from the HRX to the receiving buffer or register. Without handshaking, the host might overwrite data not transferred to the DSP side or the DSP might receive stale data.

Similarly, when the host performs multiple reads from the HI08 port Receive Data Registers (RXH:RXM:RXL), the DSP side uses an appropriate handshaking protocol to determine whether any data previously written to the Host Transmit Register (HTX) has successfully transferred to the host-side registers. If HTX is empty, the DSP writes the data to this register. Data transfers to the host-side Receive Data Registers (RXH:RXM:RXL) occur only if they are empty (that is, the host has read them). The host can then use any of the available handshaking protocols to determine whether more data is ready to be read. The DSP56309 HI08 port offers the following handshaking protocols for data transfers with the host:

- Software polling
- Interrupts
- Core DMA access
- Host requests

Table 6-9. Host Status Register (HSR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
4–3	HF[1–0]	0	**Host Flags 0, 1** General-purpose flags for host-to-DSP communication. These bits reflect the status of host flags HF[1–0] in the ICR on the host side. These two general-purpose flags can be used individually or as encoded pairs in a simple host-to-DSP communication protocol, implemented in both the DSP and the host processor software.
2	HCP	0	**Host Command Pending** Reflects the status of the CVR[HC] bit. When set, it indicates that a host command interrupt is pending. HI08 hardware clears HC and HCP when the DSP core services the interrupt request. If the host clears HC, HCP is also cleared.
1	HTDE	0	**Host Transmit Data Empty** Indicates that the host transmit data register (HTX) is empty and can be written by the DSP core. HTDE is set when the HTX register is transferred to the RXH:RXM:RXL registers. The host processor can also set HTDE using the initialize function. HTDE is cleared when the DSP core writes to HTX.
0	HRDF	0	**Host Receive Data Full** Indicates that the host receive data register (HRX) contains data from the host processor. HRDF is set when data is transferred from the TXH:TXM:TXL registers to the HRX register. The host processor can also clear HRDF using the initialize function.

6.6.3 Host Data Direction Register (HDDR)

The HDDR controls the direction of the data flow for each of the HI08 signals configured as GPIO. Even when the HI08 functions as the host interface, its unused signals can be configured as GPIO signals. For information on the HI08 GPIO configuration options, see **Section 6.2**, *Host Port Signals*, on page 6-3. If Bit DR***xx*** is set, the corresponding HI08 signal is configured as an output signal. If Bit DR***xx*** is cleared, the corresponding HI08 signal is configured as an input signal. Hardware and software reset clear the HDDR bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

Figure 6-8. Host Data Direction Register (HDDR) (X:$FFFFC8)

Table 6-17. Interface Status Register (ISR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
0	RXDF	0	**Receive Data Register Full** Indicates that the receive byte registers (RXH:RXM:RXL) contain data from the DSP56309 to be read by the host processor. RXDF is set when the HTX is transferred to the receive byte registers. RXDF is cleared when the host processor reads the receive data register (RXL or RXH according to HLEND bit). The host processor can clear RXDF using the initialize function. RXDF can assert the external HREQ signal if the RREQ bit is set. Regardless of whether the RXDF interrupt is enabled, RXDF indicates whether the RX registers are full and data can be latched out (so that the host processor can use polling techniques).

6.7.4 Interrupt Vector Register (IVR)

The IVR is an 8-bit read/write register that typically contains the interrupt vector number used with MC68000 family processor vectored interrupts. Only the host processor can read and write this register. The contents of the IVR are placed on the host data bus, H[7–0], when both the HREQ and HACK signals are asserted. The contents of this register are initialized to $0F by a hardware or software reset. This value corresponds to the uninitialized interrupt vector in the MC68000 family.

7	6	5	4	3	2	1	0
IV7	IV6	IV5	IV4	IV3	IV2	IV1	IV0

Figure 6-18. Interrupt Vector Register (IVR)

6.7.5 Receive Data Registers (RXH:RXM:RXL)

The host processor views the receive byte registers as three 8-bit read-only registers: the receive high register (RXH), the receive middle register (RXM), and the receive low register (RXL). They receive data from the high, middle, and low bytes, respectively, of the HTX register and are selected by the external host address inputs (HA[2–0]) during a host processor read operation. The memory address of the receive byte registers are set by ICR[HLEND]. If ICR[HLEND] is set, the RXH is located at address $7, RXM at $6, and RXL at $5. If ICR[HLEND] is cleared, the RXH is located at address $5, RXM at $6, and RXL at $7.

When data is transferred from the HTX register to the receive byte register at host address $7, the ISR Receive Data Register Full (RXDF) bit is set. The host processor can program the RREQ bit to assert the external HREQ signal when ISR[RXDF] is set. This indicates that the HI08 has a full word (either 8, 16, or 24 bits) for the host processor. The host processor can program the RREQ bit to assert the external HREQ signal when ISR[RXDF] is set. Assertion of the HREQ signal informs the host processor that the receive byte registers have data to be read. When the host reads the receive byte register at host address $7, the ISR[RXDF] bit is cleared.

Table 6-19. HI08 Programming Model, DSP Side (Continued)

Register	Bit					Reset Type		
	Bit No.	Bit	Name	Value	Function	HW/ SW	Indivi- dual	STOP
HPCR cont.	4	HREN	Host Request Enable	0 1	HDRQ = 0 HDRQ = 1 HREQ/HTRQ = GPIO HREQ/HTRQ HACK/HRRQ = GPIO HREQ/HTRQ = HREQ,HREQ/HTRQ HACK/HRRQ = HTRQ, HRRQ	0	—	—
	5	HAEN	Host Acknowledge Enable	0 1	HDRQ = 0 HDRQ=1 HACK/HRRQ = GPIO HREQ/HTRQ HACK/HRRQ = GPIO HACK/HRRQ = HACK HREQ/HTRQ HACK/HRRQ = HTRQ, HRRQ	0	—	—
	6	HEN	Host Enable	0 1	Host Port = GPIO Host Port Active	0	—	—
	7	—	Reserved	0	Reserved	0	—	—
	8	HROD	Host Request Open Drain	0 1	HREQ/HTRQ/HRRQ = driven HREQ/HTRQ/HRRQ = open drain	0		
	9	HDSP	Host Data Strobe Polarity	0 1	HDS/HRD/HWR active low HDS/HRD/HWR active high	0	—	—
	10	HASP	Host Address Strobe Polarity	0 1	HAS active low HAS active high	0	—	—
	11	HMUX	Host Multiplexed Bus	0 1	Separate address and data lines Multiplexed address/data	0	—	—
	12	HDDS	Host Dual Data Strobe	0 1	Single Data Strobe (HDS) Double Data Strobe (HWR, HRD)	0	—	—
	13	HCSP	Host Chip Select Polarity	0 1	HCS active low HCS active high	0	—	—
	14	HRP	Host Request Polarity	0 1	HREQ/HTRQ/HRRQ active low HREQ/HTRQ/HRRQ active high	0	—	—
	15	HAP	Host Acknowledge Polarity	0 1	HACK active low HACK active high	0	—	—

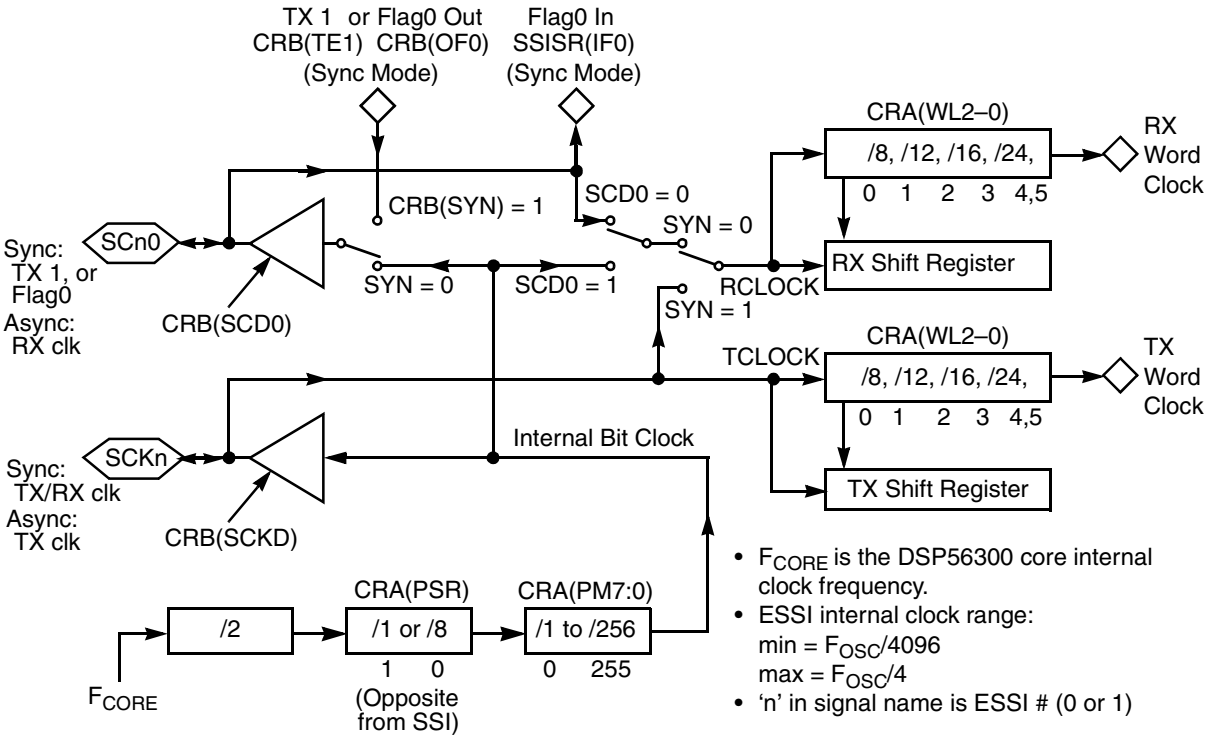


Figure 7-3. ESSI Clock Generator Functional Block Diagram

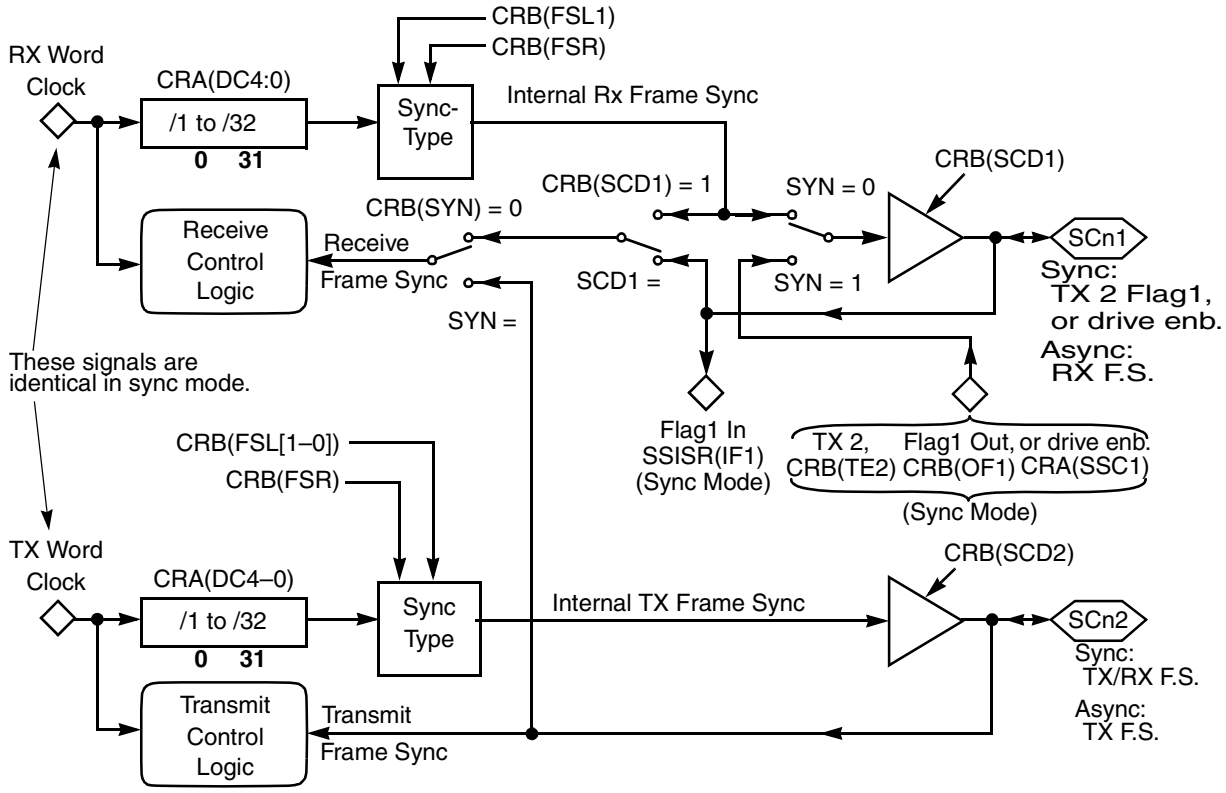


Figure 7-4. ESSI Frame Sync Generator Functional Block Diagram

Table 7-4. ESSI Control Register B (CRB) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
17	RE	0	**Receive Enable** Enables/disables the receive portion of the ESSI. When RE is cleared, the receiver is disabled: data transfer into RX is inhibited. If data is being received while this bit is cleared, the remainder of the word is shifted in and transferred to the ESSI receive data register. RE must be set in both Normal and On-Demand modes for the ESSI to receive data. In Network mode, clearing RE and setting it again disables the receiver after reception of the current data word. The receiver remains disabled until the beginning of the next data frame. **Note:** The setting of the RE bit does not affect the generation of a frame sync.
16	TE0	0	**Transmit 0 Enable** Enables the transfer of data from TX0 to Transmit Shift Register 0. TE0 is functional when the ESSI is in either synchronous or Asynchronous mode. When TE0 is set and a frame sync is detected, the transmitter 0 is enabled for that frame. When TE0 is cleared, transmitter 0 is disabled after the transmission of data currently in the ESSI transmit shift register. The STD output is tri-stated, and any data present in TX0 is not transmitted. In other words, data can be written to TX0 with TE0 cleared; the TDE bit is cleared, but data is not transferred to the transmit shift register 0. The transmit enable sequence in On-Demand mode can be the same as in Normal mode, or TE0 can be left enabled. **Note:** Transmitter 0 is the only transmitter that can operate in Asynchronous mode (SYN = 0). The setting of the TE0 bit does not affect the generation of frame sync or output flags.
15	TE1	0	**Transmit 1 Enable** Enables the transfer of data from TX1 to Transmit Shift Register 1. TE1 is functional only when the ESSI is in Synchronous mode and is ignored when the ESSI is in Asynchronous mode. When TE1 is set and a frame sync is detected, transmitter 1 is enabled for that frame. When TE1 is cleared, transmitter 1 is disabled after completing transmission of data currently in the ESSI transmit shift register. Any data present in TX1 is not transmitted. If TE1 is cleared, data can be written to TX1; the TDE bit is cleared, but data is not transferred to transmit shift register 1. If the TE1 bit is kept cleared until the start of the next frame, it causes the SC0 signal to act as serial I/O flag from the start of the frame in both Normal and Network mode. The transmit enable sequence in On-Demand mode can be the same as in Normal mode, or the TE1 bit can be left enabled. **Note:** The setting of the TE1 bit does not affect the generation of frame sync or output flags.

8.1.3.1 Transmitting Data and Address Characters

To send data, the 8-bit data character must be written to the STX register. Writing the data character to the STX register sets the ninth bit in the frame to zero, which indicates that this frame contains data. To send an 8-bit address, the address data is written to the STXA register, and the ninth bit in the frame is set to one, indicating that this frame contains an address.

8.1.3.2 Wired-OR Mode

Building a multidrop bus network requires connecting multiple transmitters to a common wire. The Wired-OR mode allows this to be done without damaging the transmitters when the transmitters are not in use. A protocol is still needed to prevent two transmitters from simultaneously driving the bus. The SCI multidrop word format provides an address field to support this protocol.

8.1.3.3 Idle Line Wakeup

A wakeup mode frees a DSP from reading messages intended for other processors. The usual operational procedure is for each DSP to suspend SCI reception (the DSP can continue processing) until the beginning of a message. Each DSP compares the address in the message header with the DSP's address. If the addresses do not match, the SCI again suspends reception until the next address. If the address matches, the DSP reads and processes the message and then suspends reception until the next address. The Idle Line Wakeup mode wakes up the SCI to read a message before the first character arrives.

8.1.3.4 Address Mode Wakeup

The purpose and basic operational procedure for Address Mode Wakeup is the same as for Idle Line Wakeup. The difference is that Address Mode Wakeup re-enables the SCI when the ninth bit in a character is set to one (if cleared, this bit marks a character as data; if set, an address). As a result, an idle line is not needed, which eliminates the dead time between messages.

8.2 I/O Signals

Each of the three SCI signals (RXD, TXD, and SCLK) can be configured as either a GPIO signal or as a specific SCI signal. Each signal is independent of the others. For example, if only the TXD signal is needed, the RXD and SCLK signals can be programmed for GPIO. However, at least one of the three signals must be selected as an SCI signal to release the SCI from reset.

To enable SCI interrupts, program the SCI control registers before any of the SCI signals are programmed as SCI functions. In this case, only one transmit interrupt can be generated because the Transmit Data Register is empty. The timer and timer interrupt operate regardless of how the SCI pins are configured, either as SCI or GPIO.

In Asynchronous mode, the start bit, the eight data bits, the address/data indicator bit or the parity bit, and the stop bit are received, respectively. Data bits are sent LSB first if SSFTD is cleared, and MSB first if SSFTD is set. In Synchronous mode, a gated clock provides synchronization. In either Synchronous or Asynchronous mode, when a complete word is clocked in, the contents of the shift register can be transferred to the SRX and the flags; RDRF, FE, PE, and OR are changed appropriately. Because the operation of the receive shift register is transparent to the DSP, the contents of this register are not directly accessible to the programmer.

8.6.4.2 SCI Transmit Register (STX)

The transmit data register is a one-byte-wide register mapped into four addresses as STXL, STXM, STXH, and STXA. In Asynchronous mode, when data is to be transmitted, STXL, STXM, and STXH are used. When STXL is written, the low byte on the data bus is transferred to the STX. When STXM is written, the middle byte is transferred to the STX. When STXH is written, the high byte is transferred to the STX. This structure makes it easy for the programmer to unpack the bytes in a 24-bit word for transmission. TDXA should be written in 11-bit asynchronous multidrop mode when the data is an address and the programmer wants to set the ninth bit (the address bit). When STXA is written, the data from the low byte on the data bus is stored in it. The address data bit is cleared in 11-bit asynchronous multidrop mode when any of STXL, STXM, or STXH is written. When either STX (STXL, STXM, or STXH) or STXA is written, TDRE is cleared.

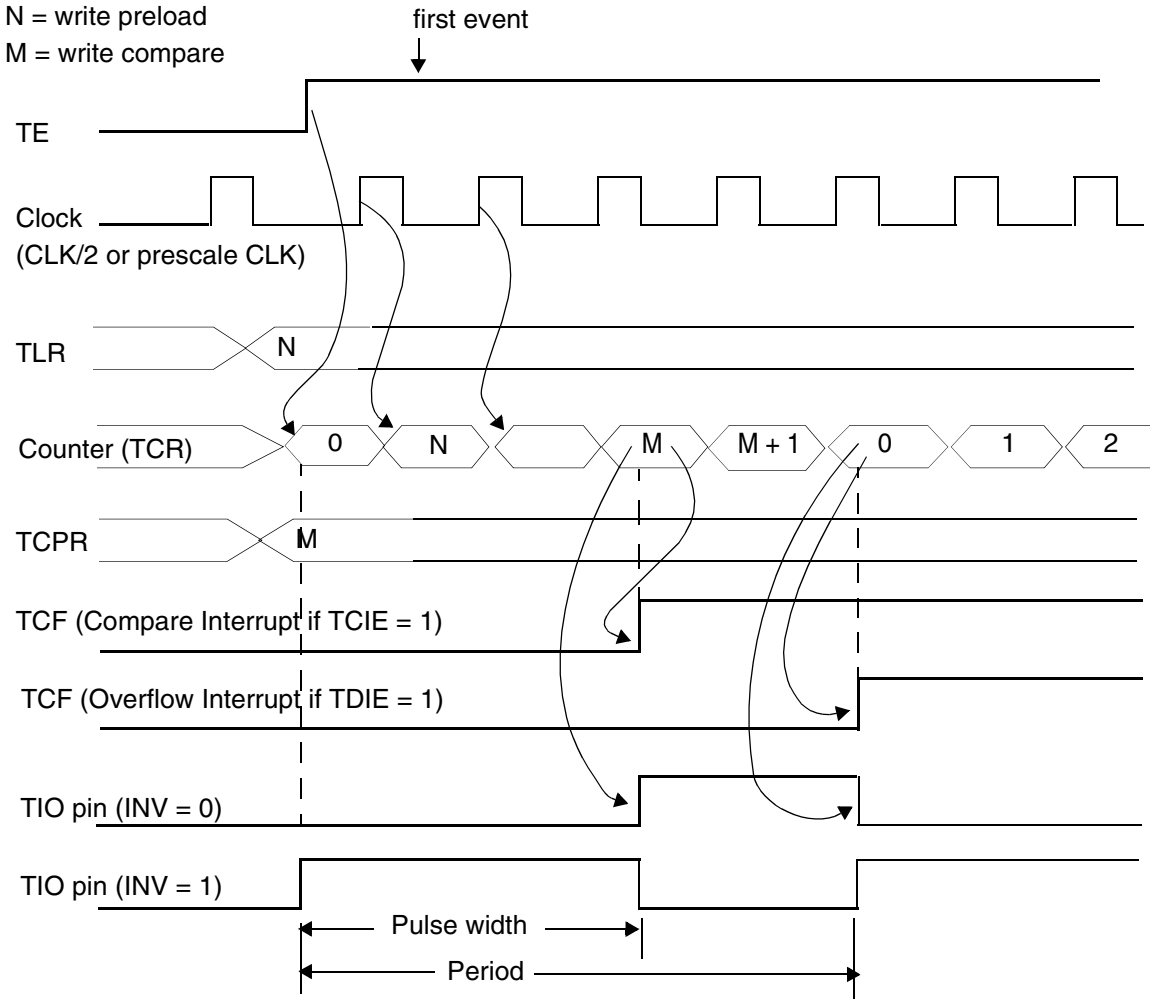
The transfer from either STX or STXA to the transmit shift register occurs automatically, but not immediately, after the last bit from the previous word is shifted out; that is, the transmit shift register is empty. Like the receiver, the transmitter is double-buffered. However, a delay of two to four serial clock cycles occurs between when the data is transferred from either STX or STXA to the transmit shift register and when the first bit appears on the TXD signal. (A serial clock cycle is the time required to transmit one data bit.)

The transmit shift register is not directly addressable, and there is no dedicated flag for this register. Because of this fact and the two- to four-cycle delay, two bytes cannot be written consecutively to STX or STXA without polling, because the second byte might overwrite the first byte. Thus, you should always poll the TDRE flag prior to writing STX or STXA to prevent overruns unless transmit interrupts are enabled. Either STX or STXA is usually written as part of the interrupt service routine. An interrupt is generated only if TDRE is set. The transmit shift register is indirectly visible via the SSR[TRNE] bit.

In Synchronous mode, data is synchronized with the transmit clock. That clock can have either an internal or external source, as defined by the TCM bit in the SCCR. The length and format of the serial word is defined by the WDS0, WDS1, and WDS2 control bits in the SCR. In Asynchronous mode, the start bit, the eight data bits (with the LSB first if SSFTD = 0 and the MSB first if SSFTD = 1), the address/data indicator bit or parity bit, and the stop bit are transmitted in that order. The data to be transmitted can be written to any one of the three STX

Period = $FFFFFF - TLR + 1
Duty cycle = ($FFFFFF - TCPR)
Ensure that TCPR > TLR for correct functionality

Mode 7 (internal clock): TRM = 0



NOTE: On overflow, TCR is loaded with the value of TLR.

Figure 9-17. Pulse Width Modulation Toggle Mode, TRM = 0

9.3.4 Watchdog Modes

The following watchdog timer modes are provided:

- Watchdog Pulse
- Watchdog Toggle

9.4.7 Timer Count Register (TCR)

The TCR is a 24-bit read-only register. In timer and watchdog modes, the contents of the counter can be read at any time from the TCR register. In measurement modes, the TCR is loaded with the current value of the counter on the appropriate edge of the input signal, and its value can be read to determine the width, period, or delay of the leading edge of the input signal. When the timer is in measurement mode, the TIO signal is used for the input signal.

Application:_____ Date:_____

_____ Programmer:_____

GPIO

Port C (ESSI0)

PCn = 1 →Port Pin configured as ESSI
PCn = 0 →Port Pin configured as GPIO

23···6	5	4	3	2	1	0
* 0 * 0	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0

Port C Control Register (PCRC) **X:$FFFFBF Read/Write**
Reset = $000000

PDCn = 1 →Port Pin is Output
PDCn = 0 →Port Pin is Input

23···6	5	4	3	2	1	0
* 0 * 0	PRC5	PRC4	PRC3	PRC2	PRC1	PRC0

Port C Direction Register (PRRC) **X:$FFFFBE Read/Write**
Reset = $000000

if port pin n is GPIO input, then PDn reflects the value on port pin n

if port pin n is GPIO output, then value written to PDn is reflected on port pin n

23···6	5	4	3	2	1	0
* 0 * 0	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0

Port C GPIO Data Register (PDRC) **X:$FFFFBD Read/Write**
Reset = $000000

*= Reserved, Program as 0

Figure B-23. Port C Registers (PCRC, PRRC, PDRC)

J

L

M

N

O

P

R