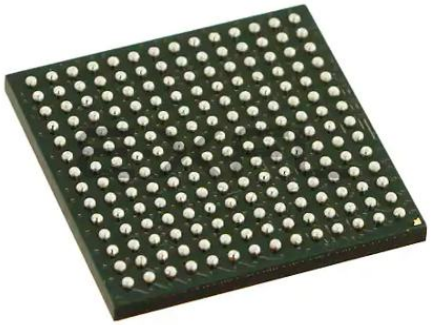

Welcome to E-XFL.COM

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	196-LBGA
Supplier Device Package	196-LBGA (15x15)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/xc56309vl100ar2

- 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

3.6 RAM Configuration Summary

The RAM configurations for the DSP56309 are listed in **Table 3-1**.

Table 3-1. DSP56309 RAM Configurations

Bit Settings		Memory Sizes (in K)			
MS	CE	Program RAM	X data RAM	Y data RAM	Cache
0	0	20	7	7	0
0	1	19	7	7	1
1	0	24	5	5	0
1	1	23	5	5	1

The actual memory locations for Program RAM and the Instruction Cache in the Program memory space are determined by the MS and CE bits, and their addresses are given in **Table 3-2**.

Table 3-2. DSP56309 RAM Address Ranges by Configuration

MS	CE	Program RAM Location	Cache Location
0	0	$0000–$4FFF	N/A
0	1	$0000–$4BFF	$4C00–$4FFF (internal location not accessible; address range assigned to external Program Memory)
1	0	$0000–$5FFF	N/A
1	1	$0000–$5BFF	$5C00–$5FFF (internal location not accessible; addressed assigned to external Program Memory)

3.7 Memory Maps

The following figures describe each of the memory space and RAM configurations defined by the settings of the SC, MS, and CE bits. The figures show the configuration and the table describes the bit settings, memory sizes, and memory locations.

4.2 Bootstrap Program

The bootstrap program is factory-programmed in an internal 192-word by 24-bit bootstrap ROM located in program memory space at locations $FF0000–$FF00BF. The bootstrap program can load any program RAM segment from an external byte-wide EPROM, the SCI, or the host port. The bootstrap program code is listed in **Appendix 4**, *Core Configuration*.

Upon exit from the Reset state, the DSP56309 samples the MODA–MODD signal lines and loads their values into OMR[MA–MD]. The mode input signals (MODA–MODD) and the resulting MA, MB, MC, and MD bits determine which bootstrap mode the DSP56309 enters (see **Table 4-1**).

Note: To stop the bootstrap in any HI08 bootstrap mode, set the Host Flag 0 (HF0). The loaded user program begins executing from the specified starting address.

You can invoke the bootstrap program options (except modes $0 and $8) at any time by writing the appropriate values to the MA, MB, MC, and MD bits in the OMR and jumping to the bootstrap program entry point, $FF0000. Software can set the mode selection bits directly in the OMR. Bootstrap modes 0 and 8 are the normal DSP56309 functioning modes. The other bootstrap modes select different specific bootstrap loading source devices. Refer to **Appendix A**, *Bootstrap Program* for details on the bootstrap program.

In these modes, the bootstrap program expects the following data sequence when downloading the user program through an external port:

1. Three bytes that specify the number of (24-bit) program words to load.
2. Three bytes that specify the (24-bit) start address where the user program loads in the DSP56309 program memory.
3. The user program (three bytes for each 24-bit program word).

Note: The three bytes for each data sequence are loaded LSB first.

When the bootstrap program finishes loading the specified number of words, it jumps to the specified starting address and executes the loaded program.

4.3 Central Processor Unit (CPU) Registers

Two CPU registers must be configured to initialize operation. The Status Register (SR) selects various arithmetic processing protocols and contains several status reporting flag bits. The Operating Mode Register (OMR) configures several system operating modes and characteristics.

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

Bit Number	Bit Name	Reset Value	Description
23	BRH	0	**Bus Request Hold** Asserts the $\overline{BR}$ signal, even if no external access is needed. When BRH is set, the $\overline{BR}$ signal is always asserted. If BRH is cleared, the $\overline{BR}$ is asserted only if an external access is attempted or pending.
22		0	Reserved. Write to zero for future compatibility.
21	BBS	0	**Bus State** This read-only bit is set when the DSP is the bus master and is cleared otherwise.
20–16	BDFW	11111 (31 wait states)	**Bus Default Area Wait State Control** Defines the number of wait states (one through 31) inserted into each external access to an area that is not defined by any of the AAR registers. The access type for this area is SRAM only. These bits should not be programmed 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.
15–13	BA3W	1 (7 wait states)	**Bus Area 3 Wait State Control** Defines the number of wait states (one through seven) inserted in each external SRAM access to Area 3 (DRAM accesses are not affected by these bits). Area 3 is the area defined by AAR3. **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. This trailing wait state increases the data hold time and the memory release time and does not increase the memory access time.
12–10	BA2W	111 (7 wait states)	**Bus Area 2 Wait State Control** Defines the number of wait states (one through seven) inserted into each external SRAM access to Area 2 (DRAM accesses are not affected by these bits). Area 2 is the area defined by AAR2. **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. This trailing wait state increases the data hold time and the memory release time and does not increase the memory access time.

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.

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

<table>
<tr><th>Bit Number</th><th>Bit Name</th><th>Reset Value</th><th colspan="3">Description</th></tr>
<tr><td rowspan="10">18–17 cont.</td><td rowspan="10">DPR</td><td rowspan="10"></td><td>OMR - CDP[1–0]</td><td>CP[1–0]</td><td>Core Priority</td></tr>
<tr><td>00</td><td>00</td><td>0 (lowest)</td></tr>
<tr><td>00</td><td>01</td><td>1</td></tr>
<tr><td>00</td><td>10</td><td>2</td></tr>
<tr><td>00</td><td>11</td><td>3 (highest)</td></tr>
<tr><td>01</td><td>xx</td><td>DMA accesses have higher priority than core accesses</td></tr>
<tr><td>10</td><td>xx</td><td>DMA accesses have the same priority as core accesses</td></tr>
<tr><td>11</td><td>xx</td><td>DMA accesses have lower priority than core accesses</td></tr>
<tr><td colspan="3">• If DMA priority > core priority (for example, if CDP = 01, or CDP = 00 and DPR > CP), the DMA performs the external bus access first and the core waits for the DMA channel to complete the current transfer.
• If DMA priority = core priority (for example, if CDP = 10, or CDP = 00 and DPR = CP), the core performs all its external accesses first and then the DMA channel performs its access.
• If DMA priority < core priority (for example, if CDP=11, or CDP = 00 and DPR < CP), the core performs its external accesses and the DMA waits for a free slot in which the core does not require the external bus.
• In Dynamic Priority mode (CDP = 00), the DMA channel can be halted before executing both the source and destination accesses if the core has higher priority. If another higher-priority DMA channel requests access, the halted channel finishes its previous access with a new higher priority before the new requesting DMA channel is serviced.</td></tr>
<tr><td>16</td><td>DCON</td><td>0</td><td colspan="3">DMA Continuous Mode Enable
Enables/disables DMA Continuous mode. When DCON is set, the channel enters the Continuous Transfer mode and cannot be interrupted during a transfer by any other DMA channel of equal priority. DMA transfers in the continuous mode of operation can be interrupted if a DMA channel of higher priority is enabled after the continuous mode transfer starts. If the priority of the DMA transfer in continuous mode (that is, DCON = 1) is higher than the core priority (CDP = 01, or CDP = 00 and DPR > CP), and if the DMA requires an external access, the DMA gets the external bus and the core is not able to use the external bus in the next cycle after the DMA access even if the DMA does not need the bus in this cycle. However, if a refresh cycle from the DRAM controller is requested, the refresh cycle interrupts the DMA transfer. When DCON is cleared, the priority algorithm operates as for the DPR bits.</td></tr>
</table>

— HRW/$\overline{\text{HRD}}$ read/write select (HRW) or read strobe ($\overline{\text{HRD}}$)
— $\overline{\text{HDS}}$/$\overline{\text{HWR}}$ data strobe ($\overline{\text{HDS}}$) or write strobe ($\overline{\text{HWR}}$)
— $\overline{\text{HCS}}$/HA10 host chip select ($\overline{\text{HCS}}$) or host address line (HA10)
— $\overline{\text{HREQ}}$/$\overline{\text{HTRQ}}$ host request ($\overline{\text{HREQ}}$) or host transmit request ($\overline{\text{HTRQ}}$)
— $\overline{\text{HACK}}$/$\overline{\text{HRRQ}}$ host acknowledge ($\overline{\text{HACK}}$) or host receive request ($\overline{\text{HRRQ}}$)

Note: The signals in the above list that are shown as asserted low (for example, $\overline{\text{HRD}}$) all have programmable polarity. The default value following reset is shown in the above list.

- Mapping:
  — HI08 registers are mapped into eight consecutive locations in the host's external bus address space.
  — The HI08 acts as a memory or I/O-mapped peripheral for microprocessors, microcontrollers, and so forth.
- Transfer modes:
  — Mixed 8-bit, 16-bit, and 24-bit data transfers, DSP-to-host and host-to-DSP
  — Host command
- Handshaking protocols:
  — Software polled
  — Interrupt-driven (Interrupts are compatible with most processors, including the MC68000, 8051, HC11, and Hitachi H8.)
- Data word: 8 bits
- Dedicated interrupts:
  — Separate request lines for each interrupt source
  — Special host commands force DSP core interrupts under host processor control. These commands are useful for
    - Real-time production diagnostics
    - Creation of a debugging window for program development
    - Host control protocols
- Interface capabilities:
  — Glueless interface (no external logic required) to
    - HC11
    - Hitachi H8
    - 8051 family
    - Thomson P6 family
  — Minimal glue logic (pull-ups, pull-downs) required to interface to
    - ISA bus
    - Freescale 68K family
    - Intel X86 family

6.3 Overview

The HI08 is partitioned into two register banks, as **Figure 6-1** shows. The host-side register bank is accessible only to the host, and the DSP-side register bank is accessible only to the DSP core. For the host, the HI08 appears as eight byte-wide locations mapped in its external address space. The DSP-side registers appear to the DSP core as six 24-bit registers mapped into internal I/O X memory space and therefore accessible via standard DSP56300 instructions and addressing modes.

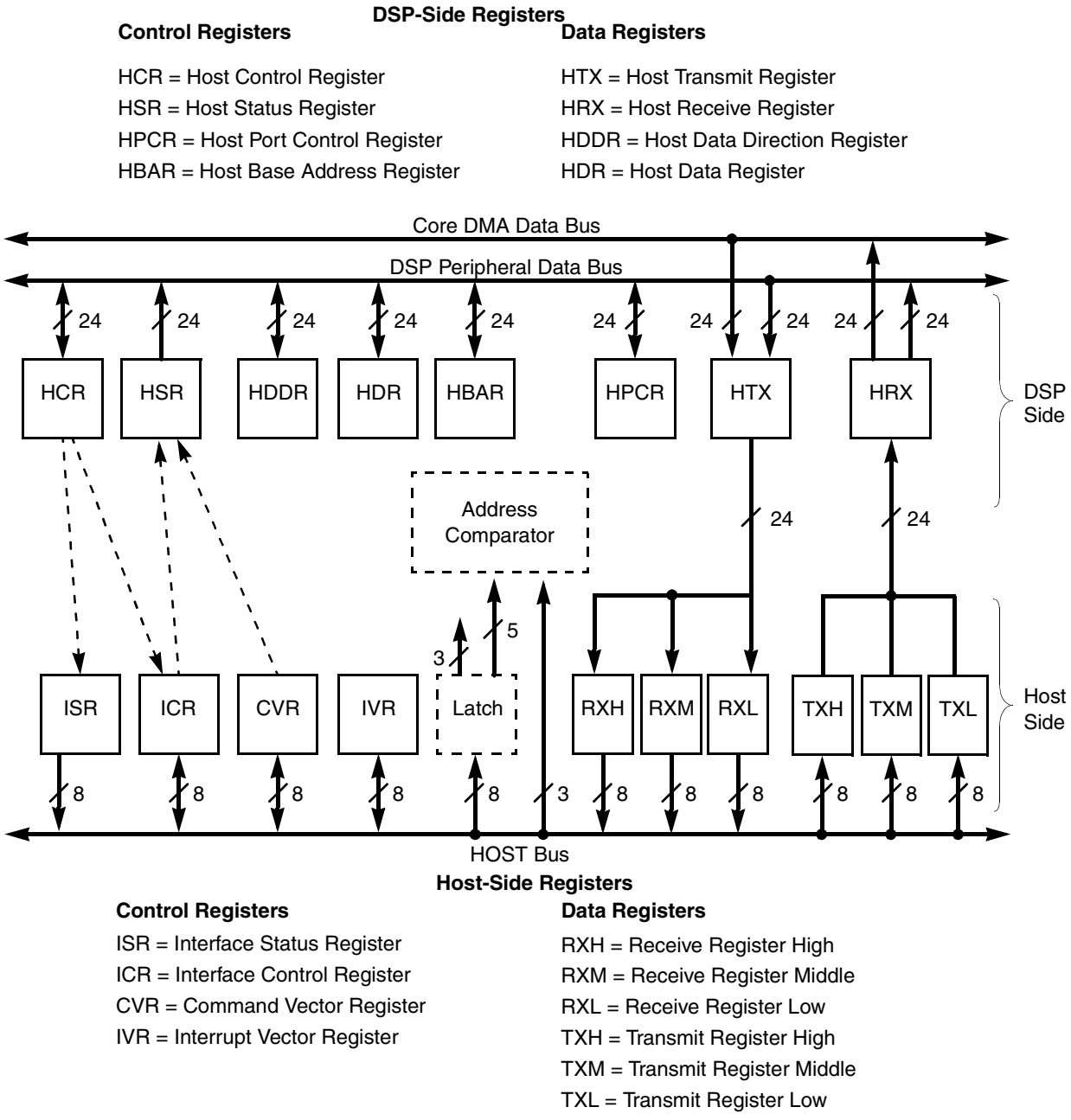

Figure 6-1. HI08 Block Diagram

Table 7-5. ESSI Status Register (SSISR) Bit Definitions

Bit Number	Bit Name	Reset Value	Description
23–8		0	Reserved. Write to 0 for future compatibility.
7	RDF	0	**Receive Data Register Full** Set when the contents of the receive shift register transfer to the receive data register. RDF is cleared when the DSP reads the receive data register. If RIE and RDF are set, a DSP receive data interrupt request is issued.
6	TDE	0	**Transmit Data Register Empty** Set when the contents of the transmit data register of every enabled transmitter are transferred to the transmit shift register. It is also set for a TSR disabled time slot period in Network mode (as if data were being transmitted after the TSR has been written). When TDE is set, TDE data is written to all the TX registers of the enabled transmitters or to the TSR. The TDE bit is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If the TIE bit is set, a DSP transmit data interrupt request is issued when TDE is set.
5	ROE	0	**Receiver Overrun Error Flag** Set when the serial receive shift register is filled and ready to transfer to the receive data register (RX) but RX is already full (that is, the RDF bit is set). If the REIE bit is set, a DSP receiver overrun error interrupt request is issued when the ROE bit is set. The programmer clears ROE by reading the SSISR with the ROE bit set and then reading the RX.
4	TUE	0	**Transmitter Underrun Error Flag** TUE is set when at least one of the enabled serial transmit shift registers is empty (that is, there is no new data to be transmitted) and a transmit time slot occurs. When a transmit underrun error occurs, the previous data (which is still present in the TX registers not written) is retransmitted. In Normal mode, there is only one transmit time slot per frame. In Network mode, there can be up to 32 transmit time slots per frame. If the TEIE bit is set, a DSP transmit underrun error interrupt request is issued when the TUE bit is set. The programmer can also clear TUE by first reading the SSISR with the TUE bit set, then writing to all the enabled transmit data registers or to the TSR.
3	RFS	0	**Receive Frame Sync Flag** When set, the RFS bit indicates that a receive frame sync occurred during the reception of a word in the serial receive data register. In other words, the data word is from the first time slot in the frame. When the RFS bit is cleared and a word is received, it indicates (only in Network mode) that the frame sync did not occur during reception of that word. RFS is valid only if the receiver is enabled (that is, if the RE bit is set). In Normal mode, RFS is always read as 1 when data is read because there is only one time slot per frame, the frame sync time slot.
2	TFS	0	**Transmit Frame Sync Flag** When set, TFS indicates that a transmit frame sync occurred in the current time slot. TFS is set at the start of the first time slot in the frame and cleared during all other time slots. If the transmitter is enabled, data written to a transmit data register during the time slot when TFS is set is transmitted (in Network mode) during the second time slot in the frame. TFS is useful in Network mode to identify the start of a frame. TFS is valid only if at least one transmitter is enabled that is, when TE0, TE1, or TE2 is set). In Normal mode, TFS is always read as 1 when data is being transmitted because there is only one time slot per frame, the frame sync time slot.

Table 7-5. ESSI Status Register (SSISR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
1	IF1	0	**Serial Input Flag 1** The ESSI latches any data on the SC1 signal during reception of the first received bit after the frame sync is detected. IF1 is updated with this data when the data in the receive shift register transfers into the receive data register. IF1 is enabled only when SC1 is an input flag and Synchronous mode is selected; that is, when SC1 is programmed as ESSI in the port control register (PCR), the SYN bit is set, and the TE2 and SCD1 bits are cleared. If it is not enabled, IF1 is cleared.
0	IF0	0	**Serial Input Flag 0** The ESSI latches any data on the SC0 signal during reception of the first received bit after the frame sync is detected. The IF0 bit is updated with this data when the data in the receive shift register transfers into the receive data register. IF0 is enabled only when SC0 is an input flag and the Synchronous mode is selected; that is, when SC0 is programmed as ESSI in the port control register (PCR), the SYN bit is set, and the TE1 and SCD0 bits are cleared. If it is not enabled, the IF0 bit is cleared.

7.5.4 ESSI Receive Shift Register

The 24-bit Receive Shift Register (see **Figure 7-12** and **Figure 7-13**) receives incoming data from the serial receive data signal. The selected (internal/external) bit clock shifts data in when the associated frame sync I/O is asserted. Data is received MSB first if SHFD is cleared and LSB first if SHFD is set. Data transfers to the ESSI Receive Data Register (RX) after 8, 12, 16, 24, or 32 serial clock cycles are counted, depending on the word length control bits in the CRA.

7.5.5 ESSI Receive Data Register (RX)

The Receive Data Register (RX) is a 24-bit read-only register that accepts data from the receive shift register as it becomes full, according to **Figure 7-12** and **Figure 7-13**. The data read is aligned according to the value of the ALC bit. When the ALC bit is cleared, the MSB is bit 23, and the least significant byte is unused. When the ALC bit is set, the MSB is bit 15, and the most significant byte is unused. Unused bits are read as 0. If the associated interrupt is enabled, the DSP is interrupted whenever the RX register becomes full.

7.5.6 ESSI Transmit Shift Registers

The three 24-bit transmit shift registers contain the data being transmitted, as in **Figure 7-12** and **Figure 7-13**. Data is shifted out to the serial transmit data signals by the selected (whether internal or external) bit clock when the associated frame sync I/O is asserted. The word-length control bits in CRA determine the number of bits that must be shifted out before the shift registers are considered empty and can be written again. Depending on the setting of the CRA, the number of bits to be shifted out can be 8, 12, 16, 24, or 32. Transmitted data is aligned according to the value of the ALC bit. When ALC is cleared, the MSB is Bit 23 and the least significant byte is unused. When ALC is set, the MSB is Bit 15 and the most significant byte is

unused. Unused bits are read as 0. Data shifts out of these registers MSB first if the SHFD bit is cleared and LSB first if SHFD is set.

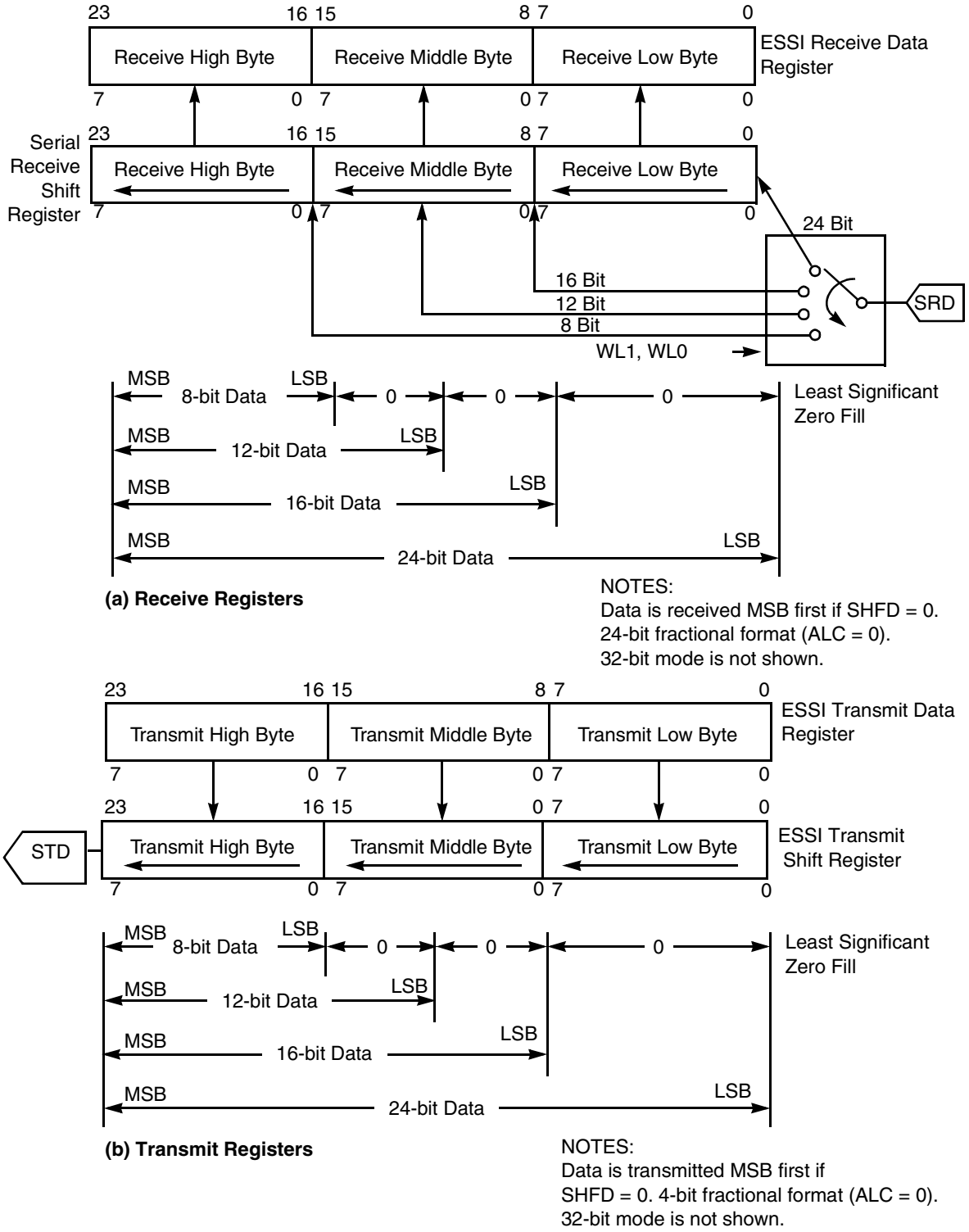

Figure 7-12. ESSI Data Path Programming Model (SHFD = 0)

Table 8-4. SCI Status Register (SSR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
4	OR	0	**Overrun Error Flag** Set when a byte is ready to be transferred from the receive shift register to the receive data register (SRX) that is already full (RDRF = 1). The receive shift register data is not transferred to the SRX. The OR flag indicates that character(s) in the received data stream may have been lost. The only valid data is located in the SRX. OR is cleared when the SCI status register is read, followed by a read of SRX. The OR bit clears the FE and PE bits; that is, overrun error has higher priority than FE or PE. A hardware $\overline{\text{RESET}}$ signal, a software RESET instruction, an SCI individual reset, or a STOP instruction clears OR.
3	IDLE	0	**Idle Line Flag** Set when 10 (or 11) consecutive ones are received. IDLE is cleared by a start-bit detection. The IDLE status bit represents the status of the receive line. The transition of IDLE from 0 to 1 can cause an IDLE interrupt (ILIE).
2	RDRF	0	**Receive Data Register Full** Set when a valid character is transferred to the SCI receive data register from the SCI receive shift register (regardless of the error bits condition). RDRF is cleared when the SCI receive data register is read.
1	TDRE	1	**Transmit Data Register Empty** Set when the SCI transmit data register is empty. When TDRE is set, new data can be written to one of the SCI transmit data registers (STX) or the transmit data address register (STXA). TDRE is cleared when the SCI transmit data register is written. Either a hardware $\overline{\text{RESET}}$ signal, a software RESET instruction, an SCI individual reset, or a STOP instruction sets TDRE. In Synchronous mode, when the internal SCI clock is in use, there is a delay of up to 5.5 serial clock cycles between the time that STX is written until TDRE is set, indicating the data has been transferred from the STX to the transmit shift register. There is a delay of 2 to 4 serial clock cycles between writing STX and loading the transmit shift register; in addition, TDRE is set in the middle of transmitting the second bit. When using an external serial transmit clock, if the clock stops, the SCI transmitter stops. TDRE is not set until the middle of the second bit transmitted after the external clock starts. Gating the external clock off after the first bit has been transmitted delays TDRE indefinitely. In Asynchronous mode, the TDRE flag is not set immediately after a word is transferred from the STX or STXA to the transmit shift register nor when the word first begins to be shifted out. TDRE is set 2 cycles (of the 16 × clock) after the start bit; that is, 2 (16 × clock) cycles into the transmission time of the first data bit.
0	TRNE	1	**Transmitter Empty** This flag bit is set when both the transmit shift register and transmit data register (STX) are empty, indicating that there is no data in the transmitter. When TRNE is set, data written to one of the three STX locations or to the transmit data address register (STXA) is transferred to the transmit shift register and is the first data transmitted. TRNE is cleared when a write into STX or STXA clears TDRE or when an idle, preamble, or break is transmitted. When set, TRNE indicates that the transmitter is empty; therefore, the data written to STX or STXA is transmitted next. That is, there is no word in the transmit shift register being transmitted. This procedure is useful when initiating the transfer of a message (that is, a string of characters).

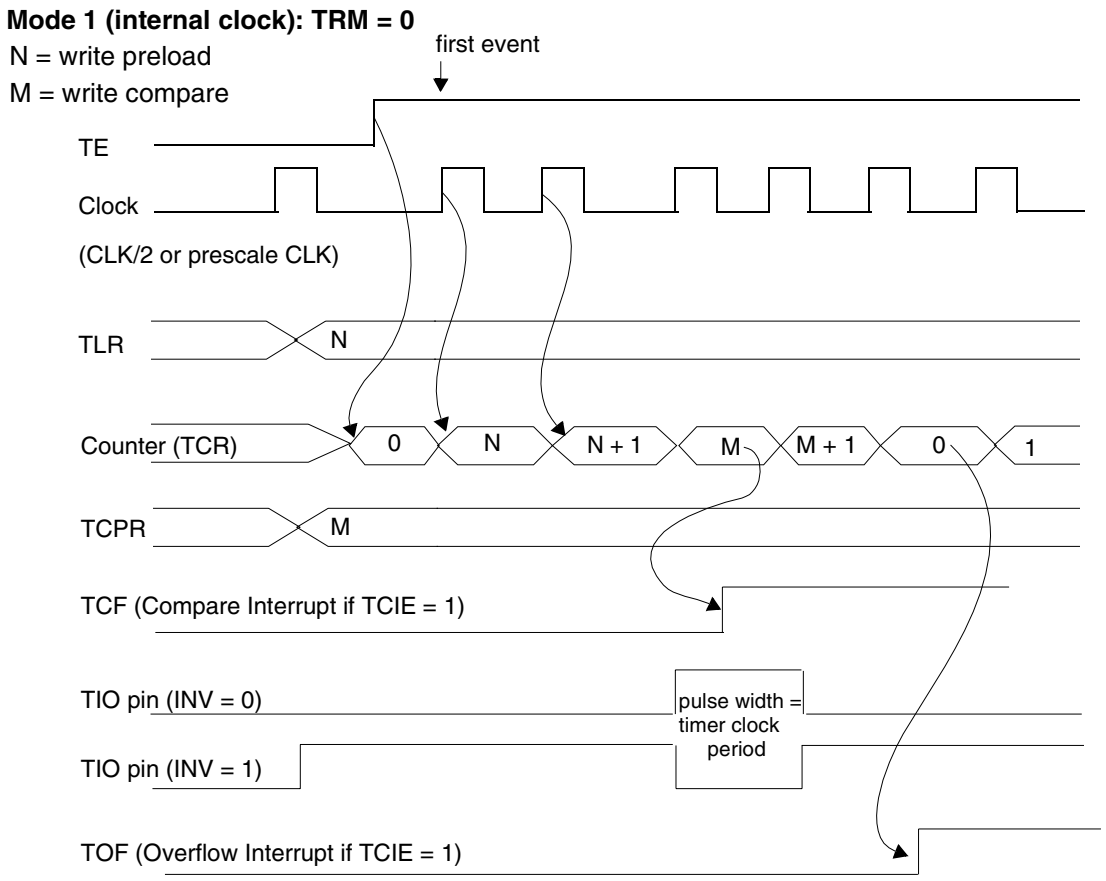

Figure 9-6. Pulse Mode (TRM = 0)

9.3.1.3 Timer Toggle (Mode 2)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	Mode	Name	Function	TIO	Clock
0	0	1	0	2	Toggle	Timer	Output	Internal

In Mode 2, the timer periodically toggles the polarity of the TIO signal. When the timer is enabled, the TIO signal is loaded with the value of the TCSR[INV] bit. When the counter value matches the value in the TCPR, the polarity of the TIO output signal is inverted. TCSR[TCF] is set, and a compare interrupt is generated if the TCSR[TCIE] bit is set. If the TCSR[TRM] bit is set, the counter is loaded with the value of the TLR when the next timer clock is received, and the count resumes. If the TRM bit is cleared, the counter continues to increment on each timer clock. This process repeats until the timer is cleared (disabling the timer). The TCPR[TLR] value sets the delay between starting the timer and toggling the TIO signal. To generate output signals with a delay of X clock cycles between toggles, set the TLR value to X/2, and set the TCSR[TRM] bit. This process repeats until the timer is disabled (that is, TCSR[TE] is cleared).

9.3.5 Special Cases

The following special cases apply during wait and stop state.

- *Timer behavior during wait*. Timer clocks are active during the execution of the **wait** instruction and timer activity is undisturbed. If a timer interrupt is generated, the DSP56309 leaves the wait state and services the interrupt.
- *Timer behavior during stop*. During execution of the **stop** instruction, the timer clocks are disabled, timer activity stops, and the TIO signals are disconnected. Any external changes that happen to the TIO signals are ignored when the DSP56309 is in stop state. To ensure correct operation, disable the timers before the DSP56309 is placed in stop state.

9.3.6 DMA Trigger

Each timer can also trigger DMA transfers if a DMA channel is programmed to be triggered by a timer event. The timer issues a DMA trigger on every event in all modes of operation. To ensure that all DMA triggers are serviced, provide for the preceding DMA trigger to be serviced before the DMA channel receives the next trigger.

9.4 Triple Timer Module Programming Model

The timer programming model in **Figure 9-20** shows the structure of the timer registers.

9.4.1 Prescaler Counter

The prescaler counter is a 21-bit counter that decrements on the rising edge of the prescaler input clock. The counter is enabled when at least one of the three timers is enabled (that is, one or more of the timer enable bits are set) and is using the prescaler output as its source (that is, one or more of the PCE bits are set).

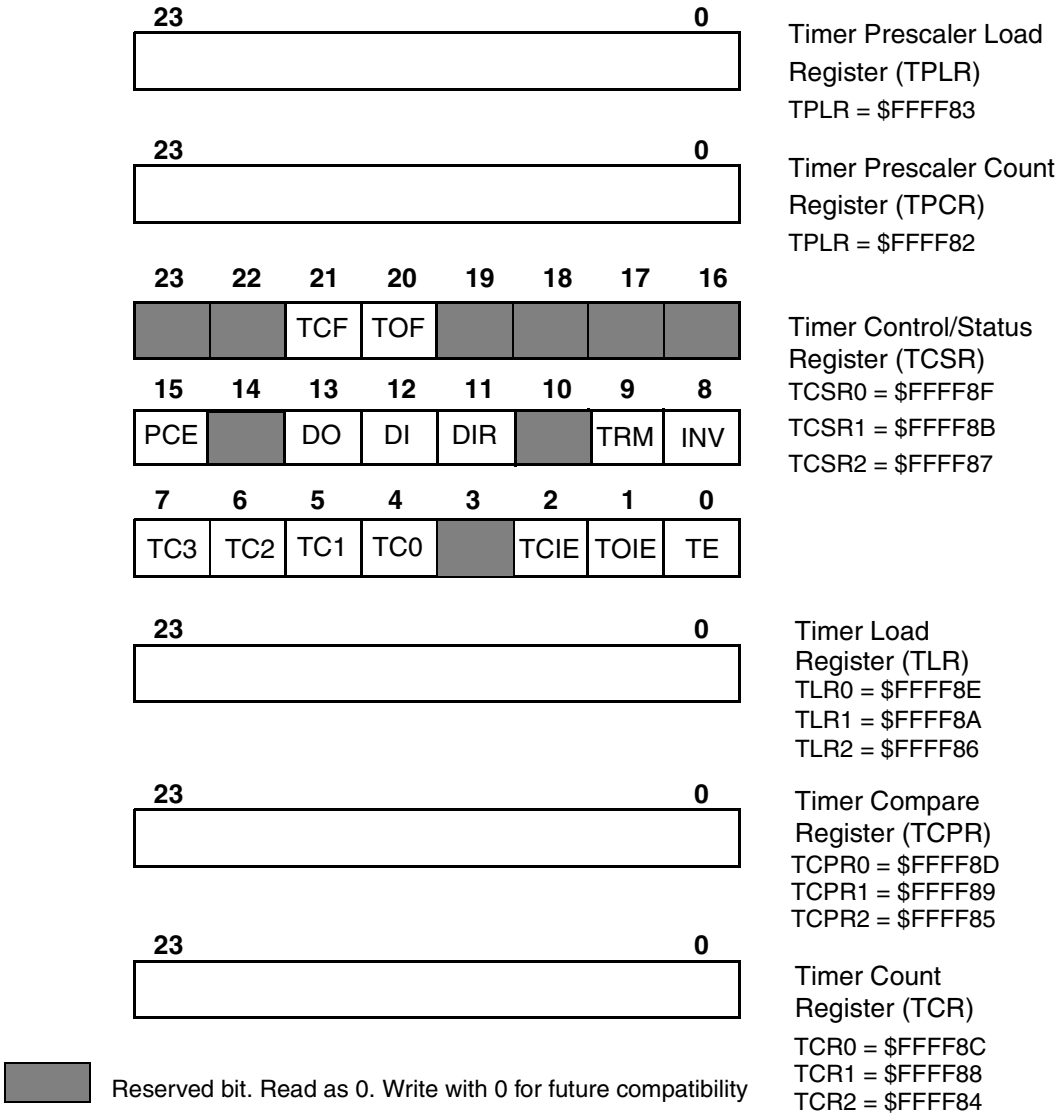

Figure 9-20. Timer Module Programming Model

9.4.2 Timer Prescaler Load Register (TPLR)

The TPLR is a read/write register that controls the prescaler divide factor (that is, the number that the prescaler counter loads and begins counting from) and the source for the prescaler input clock.

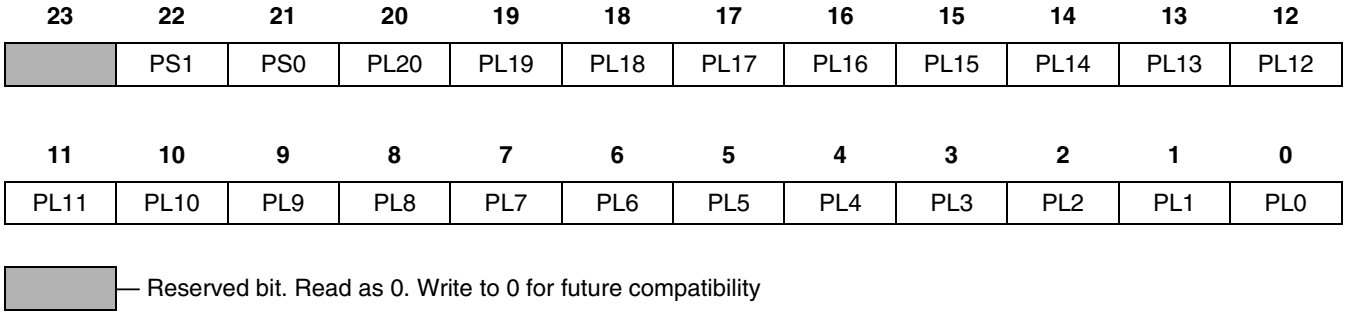

Figure 9-21. Timer Prescaler Load Register (TPLR)

```
;  2 - HC11          - Single strobe non-multiplexed bus with positive strobe
;                    pulse single negative request.
;  4 - i8051         - Dual strobes multiplexed bus with negative strobe pulses
;                    dual negative request.
;  5 - MC68302       - Single strobe non-multiplexed bus with negative strobe
;                    pulse single negative request.
;=========================================================================

MC68302HOSTLD
        movep   #%0000000000111000,x:M_HPCR
                                ; Configure the following conditions:
                                ; HAP   = 0 Negative host acknowledge
                                ; HRP   = 0 Negative host request
                                ; HCSP  = 0 Negatice chip select input
                                ; HD/HS = 0 Single strobe bus (R/W~ and DS)
                                ; HMUX  = 0 Non multiplexed bus
                                ; HASP  = 0 (address strobe polarity has no
                                ;           meaning in non-multiplexed bus)
                                ; HDSP  = 0 Negative data stobes polarity
                                ; HROD  = 0 Host request is active when enabled
                                ; spare = 0 This bit should be set to 0 for
                                ;           future compatability
                                ; HEN   = 0 When the HPCR register is modified
                                ;           HEN should be cleared
                                ; HAEN  = 1 Host acknowledge is enabled
                                ; HREN  = 1 Host requests are enabled
                                ; HCSEN = 1 Host chip select input enabled
                                ; HA9EN = 0 (address 9 enable bit has no
                                ;           meaning in non-multiplexed bus)
                                ; HA8EN = 0 (address 8 enable bit has no
                                ;           meaning in non-multiplexed bus)
                                ; HGEN  = 0 Host GPIO pins are disabled
        bra     <HI08CONT
OMR1IS0
        jset #0,omr,HC11HOSTLD  ; If MD:MC:MB:MA=1101, go load from HC11 Host
                                ; If MD:MC:MB:MA=1100, go load from ISA HOST

ISAHOSTLD
        movep   #%0101000000011000,x:M_HPCR
                                ; Configure the following conditions:
                                ; HAP   = 0 Negative host acknowledge
                                ; HRP   = 1 Positive host request
                                ; HCSP  = 0 Negatice chip select input
                                ; HD/HS = 1 Dual strobes bus (RD and WR)
                                ; HMUX  = 0 Non multiplexed bus
                                ; HASP  = 0 (address strobe polarity has no
                                ;           meaning in non-multiplexed bus)
                                ; HDSP  = 0 Negative data stobes polarity
                                ; HROD  = 0 Host request is active when enabled
                                ; spare = 0 This bit should be set to 0 for
                                ;           future compatability
                                ; HEN   = 0 When the HPCR register is modified
                                ;           HEN should be cleared
                                ; HAEN  = 0 Host acknowledge is disabled
                                ; HREN  = 1 Host requests are enabled
                                ; HCSEN = 1 Host chip select input enabled
                                ; HA9EN = 0 (address 9 enable bit has no
                                ;           meaning in non-multiplexed bus)
```

```
                                ; HA8EN = 0 (address 8 enable bit has no
                                ;           meaning non-multiplexed bus)
                                ; HGEN  = 0 Host GPIO pins are disabled
        bra     <HI08CONT
HC11HOSTLD
        movep   #%0000001000011000,x:M_HPCR
                                ; Configure the following conditions:
                                ; HAP   = 0 Negative host acknowledge
                                ; HRP   = 0 Negative host request
                                ; HCSP  = 0 Negatice chip select input
                                ; HD/HS = 0 Single strobe bus (R/W~ and DS)
                                ; HMUX  = 0 Non multiplexed bus
                                ; HASP  = 0 (address strobe polarity has no
                                ;           meaning in non-multiplexed bus)
                                ; HDSP  = 1 Negative data stobes polarity
                                ; HROD  = 0 Host request is active when enabled
                                ; spare = 0 This bit should be set to 0 for
                                ;           future compatability
                                ; HEN   = 0 When the HPCR register is modified
                                ;           HEN should be cleared
                                ; HAEN  = 0 Host acknowledge is disabled
                                ; HREN  = 1 Host requests are enabled
                                ; HCSEN = 1 Host chip select input enabled
                                ; HA9EN = 0 (address 9 enable bit has no
                                ;           meaning in non-multiplexed bus)
                                ; HA8EN = 0 (address 8 enable bit has no
                                ;           meaning in non-multiplexed bus)
                                ; HGEN  = 0 Host GPIO pins are disabled
        bra     <HI08CONT
I8051HOSTLD
        movep   #%0001110000011110,x:M_HPCR
                                ; Configure the following conditions:
                                ; HAP   = 0 Negative host acknowledge
                                ; HRP   = 0 Negatice host request
                                ; HCSP  = 0 Negatice chip select input
                                ; HD/HS = 1 Dual strobes bus (RD and WR)
                                ; HMUX  = 1 Multiplexed bus
                                ; HASP  = 1 Positive address strobe polarity
                                ; HDSP  = 0 Negative data stobes polarity
                                ; HROD  = 0 Host request is active when enabled
                                ; spare = 0 This bit should be set to 0 for
                                ;           future compatability
                                ; HEN   = 0 When the HPCR register is modified
                                ;           HEN should be cleared
                                ; HAEN  = 0 Host acknowledge is disabled
                                ; HREN  = 1 Host requests are enabled
                                ; HCSEN = 1 Host chip select input enabled
                                ; HA9EN = 1 Enable address 9 input
                                ; HA8EN = 1 Enable address 8 input
                                ; HGEN  = 0 Host GPIO pins are disabled

HI08CONT
        bset    #HEN,x:M_HPCR           ; Enable the HI08 to operate as host
                                        ; interface (set HEN=1)
        jclr    #HRDF,x:M_HSR,*         ; wait for the program length to be
                                        ; written
        movep   x:M_HRX,a0
        jclr    #HRDF,x:M_HSR,*         ; wait for the program starting address
```

```
                                        ; to be written
        movep   x:M_HRX,r0
        move    r0,r1
        do      a0,HI08LOOP             ; set a loop with the downloaded length
HI08LL
        jset    #HRDF,x:M_HSR,HI08NW    ; If new word was loaded then jump to
                                        ; read that word
        jclr    #HF0,x:M_HSR,HI08LL     ; If HF0=0 then continue with the
                                        ; downloading
        enddo                           ; Must terminate the do loop
        bra     <HI08LOOP
HI08NW
        movep   x:M_HRX,p:(r0)+         ; Move the new word into its destination
                                        ; location in the program RAM
        nop                             ; pipeline delay
HI08LOOP
        bra     <FINISH
;=========================================================================
EPRSCILD
        jclr #1,omr,EPROMLD     ; If MD:MC:MB:MA=1001, go load from EPROM
;       jset #0,omr,SCILD       ; If MD:MC:MB:MA=1011, reserved, default to SCI

;=========================================================================
; This is the routine that loads from the SCI.
; MD:MC:MB:MA=1010 - external SCI clock

SCILD
        movep #$0302,X:M_SCR    ; Configure SCI Control Reg
        movep #$C000,X:M_SCCR   ; Configure SCI Clock Control Reg
        movep #7,X:M_PCRE       ; Configure SCLK, TXD and RXD

        do #6,_LOOP6            ; get 3 bytes for number of
                                ; program words and 3 bytes
                                ; for the starting address
        jclr #2,X:M_SSR,*       ; Wait for RDRF to go high
        movep X:M_SRXL,A2       ; Put 8 bits in A2
        jclr #1,X:M_SSR,*       ; Wait for TDRE to go high
        movep A2,X:M_STXL       ; echo the received byte
        asr #8,a,a
_LOOP6
        move a1,r0              ; starting address for load
        move a1,r1              ; save starting address

        do a0,_LOOP7            ; Receive program words
        do #3,_LOOP8
        jclr #2,X:M_SSR,*       ; Wait for RDRF to go high
        movep X:M_SRXL,A2       ; Put 8 bits in A2
        jclr #1,X:M_SSR,*       ; Wait for TDRE to go high
        movep a2,X:M_STXL       ; echo the received byte
        asr #8,a,a
_LOOP8
        movem a1,p:(r0)+        ; Store 24-bit result in P mem.
        nop                     ; pipeline delay
_LOOP7
        bra <FINISH             ; Boot from SCI done

;=========================================================================
; This is the routine that loads from external EPROM.
```

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)