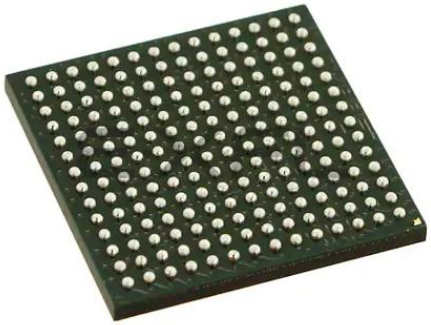


Welcome to E-XFL.COM

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/xc56309vl100a

Table 2-8. External Bus Control Signals (Continued)

Signal Name	Type	State During Reset, Stop, or Wait	Signal Description
$\overline{BR}$	Output	Output (deasserted) State during Stop/Wait depends on the BRH bit setting: • BRH = 0: Output, deasserted. • BRH = 1: Maintains last state (that is, if asserted, remains asserted)	**Bus Request** Asserted when the DSP requests bus mastership and deasserted when the DSP no longer needs the bus. $\overline{BR}$ can be asserted or deasserted independently of whether the DSP56309 is a bus master or a bus slave. Bus "parking" allows $\overline{BR}$ to be deasserted even though the DSP56309 is the bus master (see the description of bus "parking" in the $\overline{BB}$ signal description). The Bus Request Hold (BRH) bit in the BCR allows $\overline{BR}$ to be asserted under software control, even though the DSP does not need the bus. $\overline{BR}$ is typically sent to an external bus arbitrator that controls the priority, parking and tenure of each master on the same external bus. $\overline{BR}$ is affected only by DSP requests for the external bus, never for the internal bus. During hardware reset, $\overline{BR}$ is deasserted and the arbitration is reset to the bus slave state.
$\overline{BG}$	Input	Ignored Input	**Bus Grant** Must be asserted/deasserted synchronous to CLKOUT for proper operation. An external bus arbitration circuit asserts $\overline{BG}$ when the DSP56309 becomes the next bus master. When $\overline{BG}$ is asserted, the DSP56309 must wait until $\overline{BB}$ is deasserted before taking bus mastership. When $\overline{BG}$ is deasserted, bus mastership is typically given up at the end of the current bus cycle. This may occur in the middle of an instruction that requires more than one external bus cycle for execution.
$\overline{BB}$	Input/ Output	Ignored Input	**Bus Busy** Indicates that the bus is active and must be asserted and deasserted synchronous to CLKOUT. Only after $\overline{BB}$ is deasserted can the pending bus master become the bus master (and then assert the signal again). The bus master can keep $\overline{BB}$ asserted after ceasing bus activity, regardless of whether $\overline{BR}$ is asserted or deasserted. This is called "bus parking" and allows the current bus master to reuse the bus without re-arbitration until another device requires the bus. $\overline{BB}$ is deasserted by an "active pull-up" method (that is, $\overline{BB}$ is driven high and then released and held high by an external pull-up resistor). $\overline{BB}$ requires an external pull-up resistor.
$\overline{CAS}$	Output	Tri-stated	**Column Address Strobe** When the DSP is the bus master, DRAM uses $\overline{CAS}$ to strobe the column address. Otherwise, if the Bus Mastership Enable (BME) bit in the DRAM Control Register is cleared, the signal is tri-stated.
$\overline{BCLK}$	Output	Tri-stated	**Bus Clock** When the DSP is the bus master, BCLK is active when the OMR[ATE] is set. When BCLK is active and synchronized to CLKOUT by the internal PLL, BCLK precedes CLKOUT by one-fourth of a clock cycle.
$\overline{BCLK}$	Output	Tri-stated	**Bus Clock Not** When the DSP is the bus master, $\overline{BCLK}$ is the inverse of the BCLK signal. Otherwise, the signal is tri-stated.

Table 2-11. Host Interface (Continued)

Signal Name	Type	State During Reset[1,2]	Signal Description
HA0 $\overline{\text{HAS}}$/HAS PB8	Input Input Input or Output	Ignored input	**Host Address Input 0** When the HI08 is programmed to interface with a non-multiplexed host bus and the HI function is selected, this signal is line 0 of the Host Address bus. **Host Address Strobe** When the HI08 is programmed to interface with a multiplexed host bus and the HI function is selected, this signal is the Host Address Strobe (HAS) Schmitt-trigger input. The polarity of the address strobe is programmable, but is configured active-low ($\overline{\text{HAS}}$) following reset. **Port B 8** When the HI08 is configured as GPIO through the HPCR, this signal is individually programmed through the HDDR. This input is 5 V tolerant.
HA1 HA8 PB9	Input Input Input or Output	Ignored input	**Host Address Input 1** When the HI08 is programmed to interface with a non-multiplexed host bus and the HI function is selected, this signal is line 1 of the Host Address bus. **Host Address 8** When the HI08 is programmed to interface with a multiplexed host bus and the HI function is selected, this signal is line 8 of the Host Address bus. **Port B 9** When the HI08 is configured as GPIO through the HPCR, this signal is individually programmed through the HDDR. This input is 5 V tolerant.
HA2 HA9 PB10	Input Input Input or Output	Ignored input	**Host Address Input 2** When the HI08 is programmed to interface with a non-multiplexed host bus and the HI function is selected, this signal is line 2 of the Host Address bus. **Host Address 9** When the HI08 is programmed to interface with a multiplexed host bus and the HI function is selected, this signal is line 9 of the Host Address bus. **Port B 10** When the HI08 is configured as GPIO through the HPCR, this signal is individually programmed through the HDDR. This input is 5 V tolerant.

2.9 Enhanced Synchronous Serial Interface 1 (ESSI1)

Table 2-13. Enhanced Synchronous Serial Interface 1 (ESSI1)

Signal Name	Type	State During Reset[1, 2]	Signal Description
SC10 PD0	Input or Output	Ignored input	**Serial Control 0** Functions in either Synchronous or Asynchronous mode. For Asynchronous mode, this signal is the receive clock I/O (Schmitt-trigger input). For Synchronous mode, this signal is either for Transmitter 1 output or Serial I/O Flag 0. **Port D 0** The default configuration following reset is GPIO. For PD0, signal direction is controlled through the Port D Direction Register (PRRD). This signal is configured as SC10 or PD0 through the Port D Control Register (PCRD). This input is 5 V tolerant.
SC11 PD1	Input/Output Input or Output	Ignored input	**Serial Control 1** Functions in either Synchronous or Asynchronous mode. For Asynchronous mode, this signal is the receiver frame sync I/O. For Synchronous mode, this signal is either Transmitter 2 output or Serial I/O Flag 1. **Port D 1** The default configuration following reset is GPIO. For PD1, signal direction is controlled through PRRD. This signal is configured as SC11 or PD1 through PCRD. This input is 5 V tolerant.
SC12 PD2	Input/Output Input or Output	Ignored input	**Serial Control Signal 2** The frame sync for both the transmitter and receiver in Synchronous mode, and for the transmitter only in Asynchronous mode. When configured as an output, this signal is the internally generated frame sync signal. When configured as an input, this signal receives an external frame sync signal for the transmitter (and the receiver in synchronous operation). **Port D 2** The default configuration following reset is GPIO. For PD2, signal direction is controlled through PRRD. This signal is configured as SC12 or PD2 through PCRD. This input is 5 V tolerant.

Table 4-1. DSP56309 Operating Modes (Continued)

Mode	MODD	MODC	MODB	MODA	Reset Vector	Description
8	1	0	0	0	$008000	**Expanded mode** Bypasses the bootstrap ROM, and the DSP56309 starts fetching instructions beginning at address $008000. Memory accesses are performed using SRAM memory access type with 31 wait states and no address attributes selected.
9	1	0	0	1	$FF0000	**Bootstrap from byte-wide memory** The bootstrap program it loads a program RAM segment from consecutive byte-wide P memory locations, starting at P:$D00000 (bits 7-0). The memory is selected by the Address Attribute AA1 and is accessed with 31 wait states. The EPROM bootstrap code expects to read 3 bytes specifying the number of program words, 3 bytes specifying the address to start loading the program words and then 3 bytes for each program word to be loaded. The number of words, the starting address and the program words are read least significant byte first followed by the mid and then by the most significant byte. The program words are condensed into 24-bit words and stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started.
A	1	0	1	0	$FF0000	**Bootstrap through SCI** The DSP is configured to load the program RAM from the SCI interface. The number of program words to be loaded and the starting address must be specified. The SCI bootstrap code expects to receive 3 bytes specifying the number of program words, 3 bytes specifying the address to start loading the program words and then 3 bytes for each program word to be loaded. The number of words, the starting address and the program words are received least significant byte first followed by the mid and then by the most significant byte. After receiving the program words, program execution starts in the same address where loading started. The SCI is programmed to work in asynchronous mode with 8 data bits, 1 stop bit and no parity. The clock source is external and the clock frequency must be 16x the baud rate. After each byte is received, it is echoed back through the SCI transmitter.
B	1	0	1	1	$FF0000	Reserved
C	1	1	0	0	$FF0000	**HI08 bootstrap in ISA/DSP563xx mode** The HI08 is configured to load the program RAM from the Host Interface programmed to operate in the ISA mode. The HOST ISA bootstrap code expects to read a 24-bit word specifying the number of program words, a 24-bit word specifying the address to start loading the program words and then a 24-bit word for each program word to be loaded. The program words are stored in contiguous P RAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started. The Host Interface bootstrap load program may be stopped by setting the Host Flag 0 (HF0). This starts execution of the loaded program from the specified starting address.

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.

direction register (PRRE), and Port E data register (PDRE). **Chapter 8**, *Serial Communication Interface (SCI)*, discusses these registers.

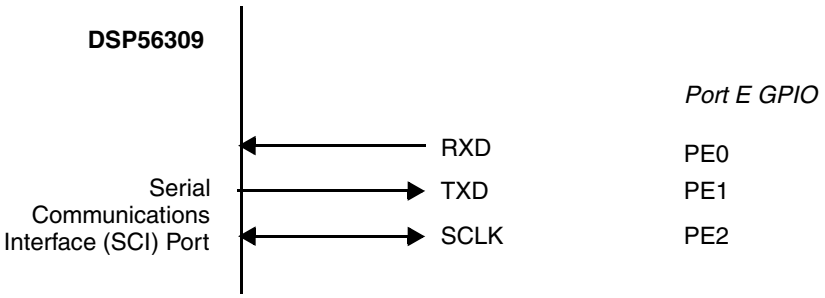


Figure 5-5. Port E Signals

5.5.5 Triple Timer Signals and Registers

Each of the three triple timer interface signals (TIO[0–2]) not used as a timer signal can be configured as a GPIO signal. Each signal is controlled by the appropriate timer control status register (TCSR[0–2]). **Chapter 9**, *Triple Timer Module*, discusses these registers.

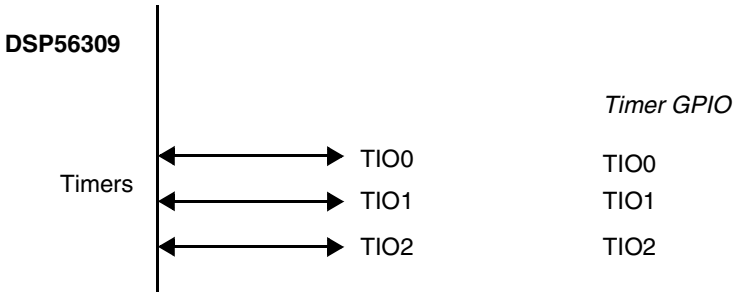


Figure 5-6. Triple Timer Signals

Note: When the DSP enters Stop mode, the HI08 pins are electrically disconnected internally, thus disabling the HI08 until the core leaves Stop mode. Do *not* issue a STOP command via the HI08 unless some other mechanism for exiting this mode is provided.

6.4.3 Core DMA Access

The DSP56300 family Direct Memory Access (DMA) controller permits transfers between internal or external memory and I/O without any core intervention. A DMA channel can be set up to transfer data to/from the HTX and HRX data registers, freeing the core to use its processing power on functions other than polling or interrupt routines for the HI08. DMA may well be the best method to use for data transfers, but it requires that one of the six DMA channels be available for use. Two HI08 DMA sources are possible, as **Table 6-4** shows. Refer to the *DSP56300 Family Manual* to learn about DMA accesses.

Table 6-4. DMA Request Sources

Requesting Device	DCRx[15–11] = DRS[4–0]
Host Receive Data Full (HRDF = 1)	10011
Host Transmit Data Empty (HTDE = 1)	10100

Note: DMA transfers do not access the host bus. The host must determine when data is available in the host-side data registers using an appropriate polling mechanism.

6.4.4 Host Requests

A set of signal lines allow the HI08 to request service from the host. The request signal lines normally connect to the host interrupt request pins (IRQx) and indicate to the host when the DSP HI08 port requires service. The HI08 can be configured to use either a single Host Request (HREQ) line for both receive and transmit requests or two signal lines, a Host Transmit Request (HTRQ) and a Host Receive Request (HRRQ), for each type of transfer.

Host requests are enabled on both the DSP-side and host-side. On the DSP side, the HPCR Host Request Enable bit (HPCR[4] = HREN) is set to enable host requests. On the host side, clearing the ICR Double Host Request bit (ICR[2] = HDRQ) configures the HI08 to use a single request line (HREQ). Setting the ICR[2] = HDRQ bit enables both transmit and request lines to be used. Further, the host uses the ICR Receive Request Enable bit (ICR[0] = RREQ) and the ICR Transmit Request Enable bit (ICR[1] = TREQ) to enable receive and transmit requests,

6.6.4 Host Data Register (HDR)

The HDR register holds the data value of the corresponding bits of the HI08 signals configured as GPIO signals. The functionality of D*xx* depends on the corresponding HDDR bit (that is, DR*xx*).The host processor can not access the Host Data Register (HDR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Figure 6-9. Host Data Register (HDR) (X:$FFFFC8)

Table 6-10. HDR and HDDR Functionality

HDDR	HDR	
DRxx	D*xx*	
	GPIO Signal[1]	Non-GPIO Signal[1]
0	Read-only bit—The value read is the binary value of the signal. The corresponding signal is configured as an input.	Read-only bit—Does not contain significant data.
1	Read/write bit— The value written is the value read. The corresponding signal is configured as an output and is driven with the data written to Dxx.	Read/write bit— The value written is the value read.
1. Defined by the selected configuration.		

6.6.5 Host Base Address Register (HBAR)

In multiplexed bus modes, HBAR selects the base address where the host-side registers are mapped into the host bus address space. The address from the host bus is compared with the base address as programmed in the Base Address Register. An internal chip select is generated if a match is found. **Figure 6-11** shows how the chip-select logic uses HBAR.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3

—Reserved bit, read as 0, write to 0 for future compatibility.

Figure 6-10. Host Base Address Register (HBAR) (X:$FFFFC5)

Table 6-11. Host Base Address Register (HBAR) Bit Definitions

Bit Number	Bit Name	Reset Value	Description
15–8		0	Reserved. Write to 0 for future compatibility.
7–0	BA[10–3]	$80	**Base Address** Reflect the base address where the host-side registers are mapped into the bus address space.

Table 6-12. Host Port Control Register (HPCR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
13	HCSP	0	**Host Chip Select Polarity** If the HCSP bit is cleared, the host chip select (HCS) signal is configured as an active low input and the HI08 is selected when the HCS signal is low. If the HCSP signal is set, HCS is configured as an active high input and the HI08 is selected when the HCS signal is high.
12	HDDS	0	**Host Dual Data Strobe** If the HDDS bit is cleared, the HI08 operates in single-strobe bus mode. In this mode, the bus has a single data strobe signal for both reads and writes. If the HDDS bit is set, the HI08 operates in dual strobe bus mode. In this mode, the bus has two separate data strobes: one for data reads, the other for data writes. See **Figure 6-13** on page -19 and **Figure 6-14** on page -19 for details on dual and single strobe modes.
11	HMUX	0	**Host Multiplexed Bus** If HMUX is set, the HI08 operates in multiplex mode, latching the lower portion of a multiplexed address/data bus. In this mode the internal address line values of the host registers are taken from the internal latch. If HMUX is cleared, it indicates that the HI08 is connected to a non-multiplexed type of bus. The values of the address lines are then taken from the HI08-dedicated address signals.
10	HASP	0	**Host Address Strobe Polarity** If HASP is cleared, the host address strobe (HAS) signal is an active low input, and the address on the host address/data bus is sampled when the HAS signal is low. If HASP is set, HAS is an active-high address strobe input, and the address on the host address or data bus is sampled when the HAS signal is high.
9	HDSP	0	**Host Data Strobe Polarity** If HDSP is cleared, the data strobe signals are configured as active low inputs, and data is transferred when the data strobe is low. If HDSP is set, the data strobe signals are configured as active high inputs, and data is transferred when the data strobe is high. The data strobe signals are either HDS by itself or both HRD and HWR together.
8	HROD	0	**Host Request Open Drain** Controls the output drive of the host request signals. In the single host request mode (that is, when HDRQ is cleared in ICR), if HROD is cleared and host requests are enabled (that is, if HREN is set and HEN is set in the host port control register (HPCR)), then the HREQ signal is always driven by the HI08. If HROD is set and host requests are enabled, the HREQ signal is an open drain output. In the double host request mode (that is, when HDRQ is set in the ICR), if HROD is cleared and host requests are enabled (that is, if HREN is set and HEN is set in the HPCR), then the HTRQ and HRRQ signals are always driven. If HROD is set and host requests are enabled, the HTRQ and HRRQ signals are open drain outputs.
7		0	Reserved. Write to 0 for future compatibility.
6	HEN	0	**Host Enable** If HEN is set, the HI08 operates as the host interface. If HEN is cleared, the HI08 is not active, and all the HI08 signals are configured as GPIO signals according to the value of the HDDR and HDR.

Table 7-3. ESSI Control Register A (CRA) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
16–12	DC[4–0]	0	**Frame Rate Divider Control** Control the divide ratio for the programmable frame rate dividers that generate the frame clocks. In Network mode, this ratio is the number of words per frame minus one. In Normal mode, this ratio determines the word transfer rate. The divide ratio ranges from 1 to 32 (DC = 00000 to 11111) for Normal mode and 2 to 32 (DC = 00001 to 11111) for Network mode. A divide ratio of one (DC = 00000) in Network mode is a special case known as On-Demand mode. In Normal mode, a divide ratio of one (DC = 00000) provides continuous periodic data word transfers. A bit-length frame sync must be used in this case; you select it by setting the FSL[1–0] bits in the CRA to (01). **Figure 7-4** shows the ESSI frame sync generator functional block diagram.
11	PSR	0	**Prescaler Range** Controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit extends the range of the prescaler when a slower bit clock is needed. When PSR is set, the fixed prescaler is bypassed. When PSR is cleared, the fixed divide-by-eight prescaler is operational, as in **Figure 7-3**. This definition is reversed from that of the SSI in other DSP56000 family members. The maximum allowed internally generated bit clock frequency is the internal DSP56309 clock frequency divided by 4; the minimum possible internally generated bit clock frequency is the DSP56309 internal clock frequency divided by 4096. **Note:** The combination PSR = 1 and PM[7–0] = $00 (dividing F_{core} by 2) can cause synchronization problems and thus should not be used.
10–8		0	Reserved. Write to 0 for future compatibility.
7–0	PM[7–0]	0	**Prescale Modulus Select** Specify the divide ratio of the prescale divider in the ESSI clock generator. A divide ratio from 1 to 256 (PM = $0 to $FF) can be selected. The bit clock output is available at the transmit clock signal (SCK) and/or the receive clock (SC0) signal of the DSP. The bit clock output is also available internally for use as the bit clock to shift the transmit and receive shift registers. **Figure 7-3** shows the ESSI clock generator functional block diagram. F_{core} is the DSP56309 core clock frequency (the same frequency as the enabled CLKOUT signal). Careful choice of the crystal oscillator frequency and the prescaler modulus can generate the industry-standard CODEC master clock frequencies of 2.048 MHz, 1.544 MHz, and 1.536 MHz.

7.5.2 ESSI Control Register B (CRB)

CRB is one of two read/write control registers that direct the operation of the ESSI (see **Figure 7-5**). The CRB bit definitions are presented in **Table 7-4**. CRB controls the ESSI multifunction signals, SC[2–0], which can be used as clock inputs or outputs, frame synchronization signals, transmit data signals, or serial I/O flag signals.

23	22	21	20	19	18	17	16	15	14	13	12
REIE	TEIE	RLIE	TLIE	RIE	TIE	RE	TE0	TE1	TE2	MOD	SYN
11	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CKP	FSP	FSR	FSL1	FSL0	SHFD	SCKD	SCD2	SCD1	SCD0	OF1	OF0

(ESSI0 X:$FFFFB6, ESSI1 X:$FFFFA6)

Figure 7-5. ESSI Control Register B (CRB)

The CRB contains the serial output flag control bits and the direction control bits for the serial control signals. Also in the CRB are interrupt enable bits for the receiver and the transmitter. Bit settings of the CRB determines how many transmitters are enabled: 0, 1, 2, or 3. The CRB settings also determine the ESSI operating mode. Either a hardware $\overline{\text{RESET}}$ signal or a software RESET instruction clears all the bits in the CRB. **Table 7-2**, *Mode and Signal Definitions,* on page 7-4 summarizes the relationship between the ESSI signals SC[2–0], SCK, and the CRB bits.

The ESSI has two serial output flag bits, OF1 and OF0. The normal sequence follows for setting output flags when transmitting data (by transmitter 0 through the STD signal only).

1. Wait for TDE (TX0 empty) to be set.
2. Write the flags.
3. Write the transmit data to the TX register

Bits OF0 and OF1 are double-buffered so that the flag states appear on the signals when the TX data is transferred to the transmit shift register. The flag bit values are synchronized with the data transfer. The timing of the optional serial output signals SC[2–0] is controlled by the frame timing and is not affected by the settings of TE2, TE1, TE0, or the receive enable (RE) bit of the CRB.

The ESSI has three transmit enable bits (TE[2–0]), one for each data transmitter. The process of transmitting data from TX1 and TX2 is the same. TX0 differs from these two bits in that it can also operate in Asynchronous mode. The normal transmit enable sequence is to write data to one or more transmit data registers (or the Time Slot Register (TSR)) before you set the TE bit. The normal transmit disable sequence is to set the Transmit Data Empty (TDE) bit and then to clear the TE, Transmit Interrupt Enable (TIE), and Transmit Exception Interrupt Enable (TEIE) bits. In Network mode, if you clear the appropriate TE bit and set it again, then you disable the corresponding transmitter (0, 1, or 2) after transmission of the current data word. The transmitter remains disabled until the beginning of the next frame. During that time period, the

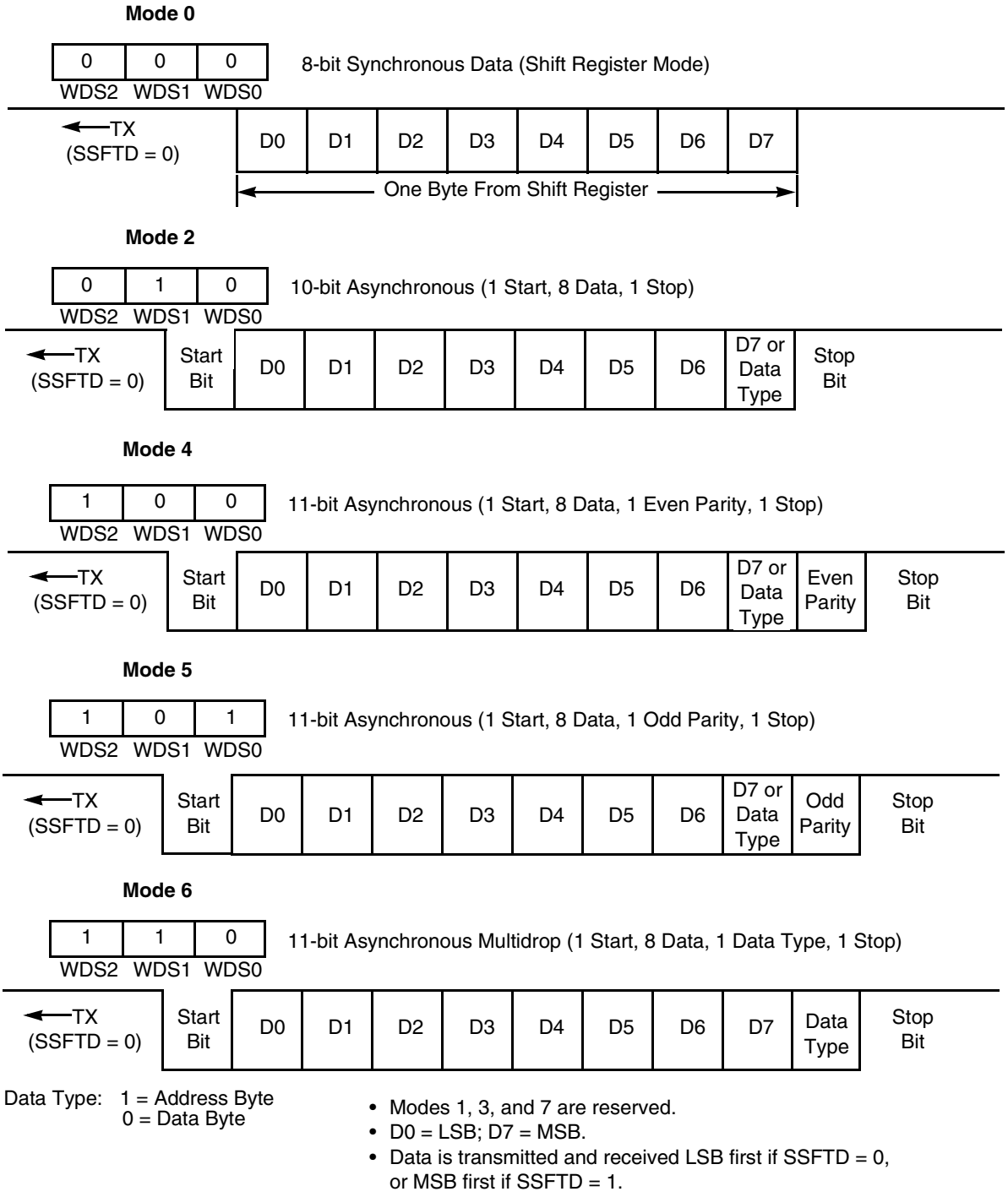
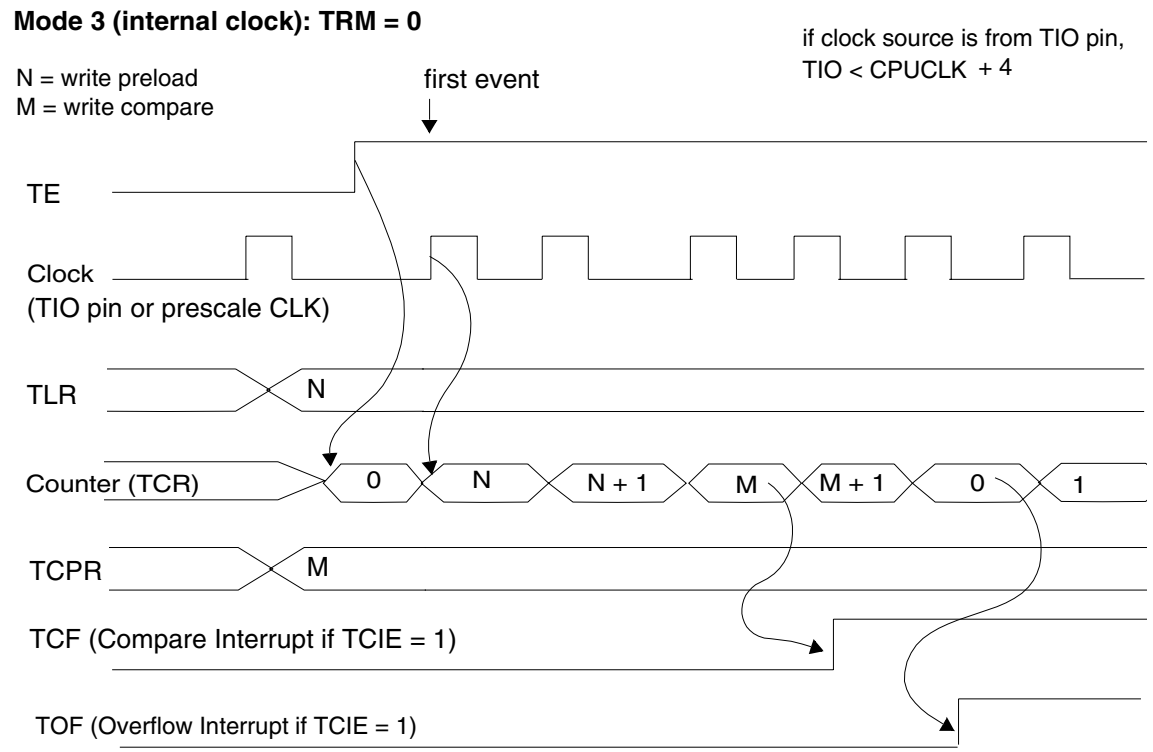


Figure 8-2. SCI Data Word Formats (SSFTD = 0), 2

8.6.1 SCI Control Register (SCR)

The SCR is a read/write register that controls the serial interface operation.



NOTE: If INV = 1, counter is clocked on 1-to-0 clock transitions, instead of 0-to-1 transitions.

Figure 9-10. Event Counter Mode, TRM = 0

9.3.2 Signal Measurement Modes

The following signal measurement and pulse width modulation modes are provided:

- Measurement input width (Mode 4)
- Measurement input period (Mode 5)
- Measurement capture (Mode 6)
- Pulse width modulation (PWM) mode (Mode 7)

The external signal synchronizes with the internal clock that increments the counter. This synchronization process can cause the number of clocks measured for the selected signal value to vary from the actual signal value by plus or minus one counter clock cycle.

Table 9-3. Timer Control/Status Register (TCSR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
2	TCIE	0	**Timer Compare Interrupt Enable** Enables/disables the timer compare interrupts. When set, TCIE enables the compare interrupts. In the timer, pulse width modulation (PWM), or watchdog modes, a compare interrupt is generated after the counter value matches the value of the TCPR. The counter starts counting up from the number loaded from the TLR and if the TCPR value is M, an interrupt occurs after (M – N + 1) events, where N is the value of TLR. When cleared, the TCSR[TCIE] bit disables the compare interrupts.
1	TOIE	0	**Timer Overflow Interrupt Enable** Enables timer overflow interrupts. When set, TOIE enables overflow interrupt generation. The timer counter can hold a maximum value of $FFFFFF. When the counter value is at the maximum value and a new event causes the counter to be incremented to $000000, the timer generates an overflow interrupt. When cleared, the TOIE bit disables overflow interrupt generation.
0	TE	0	**Timer Enable** Enables/disables the timer. When set, TE enables the timer and clears the timer counter. The counter starts counting according to the mode selected by the timer control (TC[3–0]) bit values. When clear, TE bit disables the timer. **Note:** When all three timers are disabled and the signals are not in GPIO mode, all three TIO signals are tri-stated. To prevent undesired spikes on the TIO signals when you switch from tri-state into active state, these signals should be tied to the high or low signal state by pull-up or pull-down resistors.

Table 9-4. Inverter (INV) Bit Operation

Mode	TIO Programmed as Input		TIO Programmed as Output	
	INV = 0	INV = 1	INV = 0	INV = 1
0	GPIO signal on the TIO signal read directly.	GPIO signal on the TIO signal inverted.	Bit written to GPIO put on TIO signal directly.	Bit written to GPIO inverted and put on TIO signal.
1	Counter is incremented on the **rising** edge of the signal from the TIO signal.	Counter is incremented on the **falling** edge of the signal from the TIO signal.	—	—
2	Counter is incremented on the **rising** edge of the signal from the TIO signal.	Counter is incremented on the **falling** edge of the signal from the TIO signal.	Initial output put on TIO signal directly.	Initial output inverted and put on TIO signal.
3	Counter is incremented on the **rising** edge of the signal from the TIO signal.	Counter is incremented on the **falling** edge of the signal from the TIO signal.	—	—
4	Width of the **high** input pulse is measured.	Width of the **low** input pulse is measured.	—	—
5	Period is measured between the **rising** edges of the input signal.	Period is measured between the **falling** edges of the input signal.	—	—

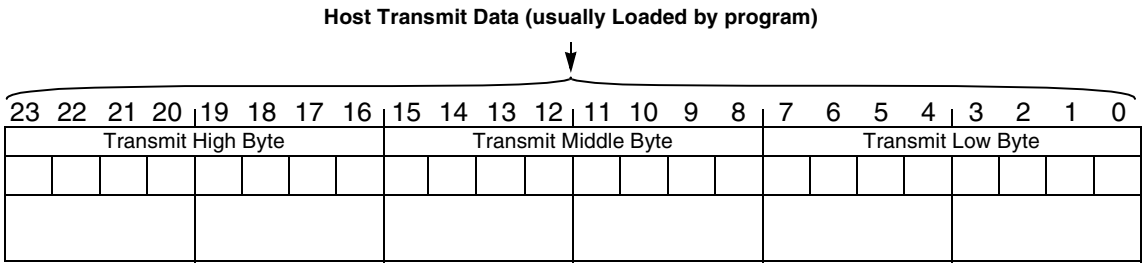
```
                                        ; to be written
        movep   x:M_HRX,r0
        move    r0,r1
        do      a0,HI08LOOP             ; set a loop with the downloaded length
HI08LL
        jset    #HRDF,x:M_HSR,HI08NW    ; If new word was loaded then jump to
                                        ; 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:_____

Sheet 1 of 5

HOST



Host Transmit Data Register (HTX) **X:$FFFFC7 Write Only**

Reset = empty

Figure B-9. Host Transmit Data Register

G

H

R

T