Chapter 5

Serial I/O

CHAPTER 5. SERIAL I/O

CONTENTS

5. SERIAL I/O

The serial I/O port on the DSP16/DSP16A device provides a serial interface to many codecs and signal processors with few if any external chips. The high-speed, double-buffered port supports back-to-back transmissions. The output buffer empty (OBE) and input buffer full (IBF) flags facilitate the reading and/or writing of the serial I/O port by program or interrupt driven I/O. There are four selectable active clock speeds. A bit-reversal mode provides compatibility with either most significant bit (MSB) first or least significant bit (LSB) first serial I/O formats. A multiprocessor I/O configuration requiring no external chips is provided. Three registers, serial I/O control (sioc), time-division multiplexed slot (tdms), and serial receive/transmit address (srta), allow the modes of operation to be controlled.

Figure 5-1 shows a simplified block-level representation of the serial I/O data path. The double-buffered inputs (isr and ibuf) and outputs (obuf and osr) connect to the internal data bus. The serial I/O uses a register-based implementation. The input and output buffer registers (ibuf and obuf, respectively) are used to input and output the data through the port. Both registers are referenced in the instruction set by the name sdx. Unlike other registers in the DSP16/DSP16A device, the writing of sdx and the reading of sdx are performed on two distinct registers. The ICK, OCK, ILD, and OLD interface is represented by the clock generator block. The signals connected to this block are bidirectional and may be programmed via the sioc register. The ifsr and ofsr provide flag signals for the input and output (IBF and OSE), respectively. The multiprocessor I/O is not represented in Figure 5-1. The signals shown on the lower portion of Figure 5-1 are described in Section 5.3.

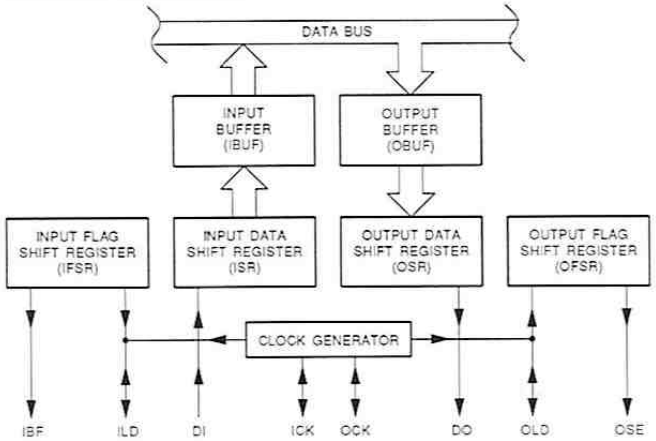

Figure 5-1. Serial I/O Internal Data Path

5.1 SIO OPERATION

The following subsections describe the operation of the SIO input and output sections and the active clock generator.

5.1.1 Input Section

A typically free-running clock (ICK) synchronizes all events occurring within the input section of the SIO. A high-to-low transition of the input load (ILD) signal followed by a rising edge of ICK initiates the start of an input transaction. The first serial data bit is read from DI on the next rising edge of ICK. Eight or sixteen bits later, when the input shift (isr) register fills, this data is transferred to the input buffer (ibuf) register and the input buffer full (IBF) signal is asserted, indicating that the buffer is full. The DSP16/DSP16A device may read the data at this time. The read command is of the type a0 = sdx, a1 = sdx, or *Y* = sdx. The IBF is negated when the input buffer is read. Another serial input may begin before the input buffer read takes place since the port is double-buffered. If the new transfer is completed before the previous input is read, then the new data is transferred to the input buffer, overwriting the old data. The status of IBF may be read from the pioc register, IBF status field (bit 4), or the IBF field (bit 15) – this is the sign bit that can be tested without masking
(i.e., a0 = pioc \ a0 = a0 \ if pl goto loop). The IBF may also be used as an interrupting condition, if the appropriate enable bit in the pioc register is set (pioc register, bit 9).

5.1.2 Output Section

When the DSP16/DSP16A device is reset (power-up or RSTB), the internal status flag output buffer empty (OBE) is set, indicating that the buffer is empty. When data is written to the output buffer by an instruction of the form sdx = a0, sdx = a1, sdx = *Y*, or sdx = *value*, OBE is cleared and the serial output section is ready for a serial transmission. The status of the OBE flag may be read from the pioc register (OBE status field, bit 3). The OBE may be used as an interrupting condition if the appropriate enable bit in the pioc register is set (pioc register, bit 8). A typically free-running clock (OCK) synchronizes all events taking place within the output section. A high-to-low transition of the output load (OLD) signal, followed by a rising edge of OCK, initiates the start of an output transaction. This procedure causes the contents of the output buffer register to be transferred to the output shift (osr) register, the internal flag OBE to be set (indicating the need for more data), and a high-to-low transition of OSE (indicating that the shift register is full). The first serial data bit is placed on the data output (DO) at this time. Eight or sixteen bits later, when the serial output has been completed, the output shift register empty (OSE) signal will be asserted, indicating that the last bit of the serial transmission has been sent. (OSE can be used by external hardware to latch a shift register.) If the output buffer has been reloaded, another transfer begins immediately; otherwise, zeros are sent on the serial output until the buffer is reloaded prior to a high-to-low transition of OLD beginning another transmission. Double-buffering allows the output buffer to be reloaded while data is being shifted out of the output shift register.

A serial address (SADD) transmits simultaneously with DO. This address is the transmit address field of the srta register (see Table 5-4). The SADD output is active low. The SADD may also be used as a second serial output (only) port. The SADD signal is valid whether or not the device is in the multiprocessor mode (See Section 5.6). If SADD is to be used in this manner, the LD field of the sioc register should be set high to synchronize SADD with DO.

5.1.3 Active Clock Generator

The active clock signals for the SIO section are derived from CKI, with a maximum frequency of CKI ÷ 4. A simplified representation of the SIO active clock and load generator is shown in Figure 5-2. In the figure, the switches represent the user-programmable features. A closed switch corresponds to the associated bit in the sioc or tdms register having a value of one.

The five signals ICK, OCK, ILD, OLD, and SYNC can be individually programmed to be either inputs or outputs (passive or active). When using active clocks (generated by the DSP16/DSP16A device), the speed of the clocks can be selected from four speeds: CKI divided by 4, 12, 16, or 20. This selection determines the speed of both ICK and OCK. The speed of ILD and OLD can be selected as either the ICK or OCK signals divided by 16. An active SYNC signal is generated from this same source (ICK or OCK divided by 16) and is further divided by 8 or 16. The resulting SYNC signal is either the signal ICK or OCK divided by 128 or 256. The SYNC signal can be configured to generate an 8 kHz sampling signal for codec applications.

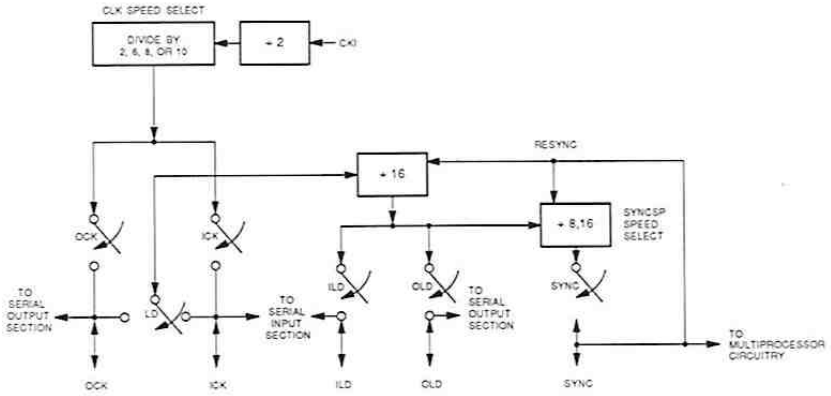

Figure 5-2. SIO Active Clock and Load Generation

5.2 USER-CONTROLLED FEATURES

Programmable modes are controlled by the serial I/O control (sioc) register. Flexibility in programming the functions of the serial I/O port allows the port to interface with a variety of devices with little or no "glue logic." Table 5-1 shows the control bits of the sioc register. During device reset, the sioc register bits are cleared.

Table 5-1. Serial I/O Control (sioc) Register

Bit	9	8	7	6	5	4	3	2	1	0
Field	LD	CLK		MSB	OLD	ILD	OCK	ICK	OLEN	ILEN

Field	Value	Result/Description
LD	0	Active ILD/OLD = ICK÷16, Active SYNC = ICK÷128/256.†
	1	Active ILD/OLD = OCK÷16,‡ Active SYNC = OCK÷128/256.†,‡
CLK	0 0	Active clock = CKI÷4.
	0 1	Active clock = CKI÷12.
	1 0	Active clock = CKI÷16.
	1 1	Active clock = CKI÷20.
MSB	0	LSB first.
	1	MSB first.
OLD	0	OLD is an input (passive mode).
	1	OLD is an output (active mode).
ILD	0	ILD is an input (passive mode).
	1	ILD is an output (active mode).
OCK	0	OCK is an input (passive mode).
	1	OCK is an output (active mode).
ICK	0	ICK is an input (passive mode).
	1	ICK is an output (active mode).
OLEN	0	16-bit output.
	1	8-bit output.
ILEN	0	16-bit input.
	1	8-bit input.

† Either 128 or 256 – see tdms register SYNC field.
‡ Select this mode when using SADD (not necessary if ICK = OCK).

The following section describes each programmable mode in detail.

- **LD** – The LD field (sioc bit 9) allows the active (internally generated) ILD and OLD signals to be derived from either ICK (LD = 0) or OCK (LD = 1). Active ILD and OLD always are derived from the same source.
- **CLK** – The CLK field (sioc bits 8,7) allows one of four active I/O speeds to be selected: a division of the input clock CKI by either 4, 12, 16, or 20. As an example, with a CKI of 8.192 MHz, 24.576 MHz, 32.768 MHz, or 40.960 MHz, using the appropriate divisor of 4, 12, 16, or 20, respectively, results in an active I/O rate of 2.048 MHz. Refer to Table 5-1 for the CLK field encoding.
- **MSB** – The MSB field (sioc bit 6) determines the bit order of the serial transmissions: most significant bit (MSB) first (MSB = 1) or least significant bit (LSB) first (MSB = 0). This mode switch allows compatibility with devices that perform either MSB first or LSB first serial transfers. This mode is also useful when performing μ-law or A-law conversions. A minimal amount of software is required to perform these conversions. Since this field allows the bit order to be switched when an sdx read or write occurs, the MSB field can be switched immediately before and/or after an sdx read or write. If this technique is used in other than an interrupt service routine, care should be taken to insure that the proper mode is in effect in the event of an interrupt.
- **OLD** – The OLD field (sioc bit 5) allows OLD to be either an input (OLD = 0) or an output (OLD = 1).
- **ILD** – The ILD field (sioc bit 4) allows ILD to be either an input (ILD = 0) or an output (ILD = 1).
- **OCK** – The OCK field (sioc bit 3) allows OCK to be either an input (OCK = 0) or an output (OCK = 1).
- **ICK** – The ICK field (sioc bit 2) allows ICK to be either an input (ICK = 0) or an output (ICK = 1).
- **OLEN** – The OLEN field (sioc bit 1) controls the length of the serial output: either 16-bit (OLEN = 0) or 8-bit (OLEN = 1). When the data is sent in the 8-bit mode with the LSB first (MSB = 0), the eight data bits should be placed in the least significant half of obuf; i.e., 0x00DD (D = data). When the data is sent in the 8-bit mode with the MSB first (MSB = 1), the eight data bits should be "packed" in the most significant half of obuf; i.e., 0xDD00 (D = data).
- **ILEN** – The ILEN field (sioc bit 0) controls the length of the serial input: either 16-bit (ILEN = 0) or 8-bit (ILEN = 1). When the data is sent in the 8-bit mode with the LSB first (MSB = 0), the eight data bits are placed in the most significant half of ibuf; i.e., 0xDD00 (D = data). When the data was sent in the 8-bit mode with the MSB first (MSB = 1), the eight data bits are placed in the least significant half of ibuf; i.e., 0x00DD (D = data).

5.3 SERIAL I/O PIN DESCRIPTIONS

The physical serial I/O port consists of eleven signals: four are used for serial input, four are used for serial output, and three are used in multiprocessor and/or TDM applications. Table 5-2 lists each signal with its type, pin number, and description.

Table 5-2. Serial I/O Pins

Symbol	Type*	Pin	Name/Description
DI	I	56	**Data Input.** Serial PCM data latched on rising edge of ICK, either LSB or MSB first, according to the sioc register MSB field.
ICK	I/O†	58	**Input Clock.** Clock for serial PCM input data. In active mode, ICK is an output; in passive mode, ICK is an input, according to the sioc register ICK field.
ILD	I/O†	57	**Input Load.** Falling edge of ILD indicates the beginning of a serial input word. In active mode, ILD is an output; in passive mode, ILD is an input, according to the sioc register ILD field.
IBF	O†	53	**Input Buffer Full.** IBF is asserted when the input buffer is filled and negated by a read of the buffer. IBF is also negated by asserting RSTB.
DO	O†	61	**Data Output.** Serial PCM data output from the output shift register (osr), either LSB or MSB first – according to the sioc register MSB field. DO changes on the rising edges of OCK. DO is 3-stated when DOEN is high.
DOEN	I/O†	64	**Data Output Enable (Active-Low).** An input when not in the multiprocessor mode. DO and SADD are enabled only if DOEN is low. DOEN is an output when in the multiprocessor mode (tdms register MODE field set). In the multiprocessor mode, DOEN indicates a valid time slot for a serial output.
OCK	I/O†	59	**Output Clock.** Clock for serial PCM output data. In active mode, OCK is an output; in passive mode, OCK is an input, according to the sioc register OCK field.
OLD	I/O†	60	**Output Load.** Clock for loading the parallel-to-serial converter from the output buffer (obuf). A falling edge of OLD indicates the beginning of a serial output word. In active mode, OLD is an output; in passive, OLD is an input, according to the sioc register OLD field.
OSE	O†	52	**Output Shift Register Empty.** Indicates the end of a serial transmission. OSE is set either by the emptying of the output shift register or by asserting RSTB. OSE is reset by the DSP16 writing a word (two clock cycles after the falling edge of OLD) to the output shift register. If no new word is written by the DSP16, OSE remains high regardless of activity on OLD.

* I = Input; O = Output.
† 3-stated.

www.DataSheet4U.com

Table 5-2. Serial I/O Pins (continued)			
Symbol	Type*	Pin	Name/Description
SADD	I/O†	63	**Serial Address (Active-Low).** An 8-bit serial bit stream typically used for addressing during multiprocessor communication between multiple DSP16 devices. In multiprocessor mode, SADD is an output when the tdms time slot dictates a serial transmission; otherwise, it is an input. SADD is always an output when not in multiprocessor mode and can be used as a second serial output. SADD is 3-stated when DOEN is high.
SYNC	I/O†	62	**Multiprocessor Synchronization.** Typically used in the multiprocessor mode, a falling edge of SYNC indicates the first word of a TDM I/O stream and causes the resynchronization of the active ILD and OLD generators. SYNC is an output when the tdms register SYNC field is set; otherwise, it is an input. SYNC must be tied low if it is not used as an output. When used as an output, SYNC = ILD/OLD ÷ 8 or 16, depending on the setting of the SYNCSP field of the tdms register. This procedure can be used to generate a slow clock for SIO operation.

* I = Input; O = Output.
† 3-stated.

5.4 CODEC INTERFACE

Figure 5-3 is the schematic showing the connections required to interface the DSP16/DSP16A device to an AT&T T7500 μ-law/A-law Codec. Figure 5-4 shows the connections necessary to interface the DSP16/DSP16A device to an AT&T T7520 or T7522 High-Precision Codec. In both examples, the SYNC signal is actively driving ILD, OLD, and the codec with an 8 kHz signal.

5.5 SERIAL I/O PROGRAMMING EXAMPLE

The program segment shown in this section demonstrates the use of the serial I/O port's interrupt facility. The advantage of using the interrupt on input buffer full (IBF) is that the input data is read in immediately, making careful placement of the sdx read commands within the program unnecessary. This program allows 128 inputs to be read into a buffer while another buffer already loaded with data is used by the program. When the first buffer fills, the two buffers are switched and the process repeats.

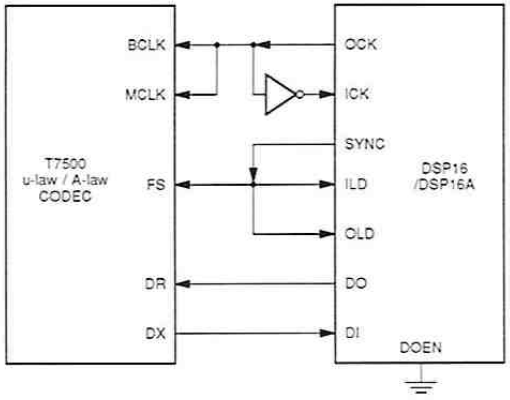

Figure 5-3. *WE*® DSP16/DSP16A to AT&T T7500 Codec Interface

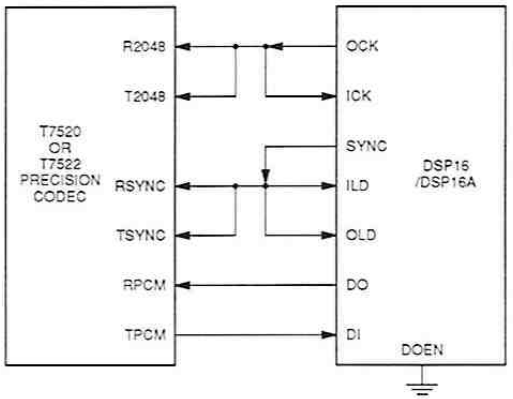

Figure 5-4. *WE*® DSP16/DSP16A to AT&T T7520 or T7522 Codec Interface

5.5.1 Program Segment

```
     /*    Ping pong I/O routine      */

           goto start
intrpt:    *r0++ = sdx
           ireturn
start:     auc = 0x0
           pioc = 0x200     /* interrupt on IBF      */
           sioc = 0x0       /* passive I/O           */
           y = 0x0
           r1 = 0xfe
           *r1 = y            /* initialize flag           */
           r2 = 0xfd          /* temp storage              */
           r0 = 0xff          /* interrupt pointer         */
           r1 = 0xff          /* program I/O pointer       */
           y = 0x17f          /* address of last sample in */
           *r2 = y            /* 128 point buffer          */
           goto loop
mainprg:

     /*    Main program here; prog. must take less time than I/O! */

           a0 = *r1++         /* read in data from buffer */

loop:      a0 = r0            /* r0 is address of input ptr.  */
           y = *r2
           a0 - y             /* check for 128 samples in buffer */
           if ne goto loop    /* loop if not full                */
           r1 = 0xfe
           a0 = *r1           /* get alternate buf flag          */
           a0 = a0            /* set DAU flags                   */
           if eq goto buf     /* alternate between buffers       */
           y = 0x00
           *r1 = y            /* set flag for buf1               */
           r0 = 0xff          /* interrupt pointer to buf1       */
           r1 = 0x17f         /* program I/O pointer to buf2     */
           y = 0x17f
           *r2 = y            /* address of last sample in buf1  */
           goto mainprg
buf:       y = 0x01
           *r1 = y            /* set flag for buf2               */
           r0 = 0x17f         /* interrupt pointer to buf2       */
           r1 = 0xff          /* program I/O pointer to buf1     */
           y = 0x1ff          /* address of last sample in buf2  */
           *r2 = y
           goto mainprg
```


5.6 MULTIPROCESSOR MODE DESCRIPTION

The multiprocessor mode allows up to eight DSP16/DSP16A devices to be connected together in such a way as to provide data transmission to any of the individual DSP16/DSP16As in the system. Two registers associated with the multiprocessor mode are the time division multiplexed slot (tdms) register (see Table 5-3) and the serial receive and transmit address (srta) register (see Table 5-4). This mode requires no external hardware and uses a TDM interface with eight time slots per frame. A serial address on the SADD line is sent simultaneously with data on DO from any one device in a predetermined time slot, and the data is received only by other device(s) having the address specified. Each device has a user-programmable receive address associated with it.

In the multiprocessor mode, the following pins are connected together to form a four-wire bus, as shown in Figure 5-5. The DI and DO form a single-wire data bus referred to as DATA; ICK and OCK form a clock line referred to as CK; the SADD forms a single-wire address bus referred to as ADD; and SYNC provides a synchronization line referred to as SYN. Typically, CK and SYN are specified statically for one particular DSP16/DSP16A device to always generate, although CK may also be generated by an external clock. The signals are generated by the DSP16/DSP16A device having active SYNC and OCK signals, which occur when the tdms register SYNC field is set and the sioc register OCK field is set. The other devices use the SYNC and OCK signals in the passive mode to synchronize operations. All DSPs have their ILD and OLD signals in active mode. A high-to-low transition of SYNC delineates transmit slot 0. Eight words are exchanged within a SYNC frame, so the tdms register should have the SYNCSP field set low when in the multiprocessor mode. This provides 128 active ICK and/or OCK cycles per SYNC frame (8 words × 16-bits/word). The multiprocessor mode is turned on by setting the tdms mode field to one.

In the multiprocessor mode, each device can send data in a unique time slot designated by the tdms register transmit slot field. The tdms register has fully decoded fielding in order to allow for one DSP16/DSP16A device transmitting in more than one time slot. This procedure is useful for multiprocessor systems with less than eight DSP16/DSP16A devices, when a higher bandwidth is necessary between certain devices in that system. Each device also has an address specified by the srta register transmit address field (Table 5-4), used to transmit the information regarding the destination of the data and an address assignment made to it by the receive address field referring to its own identity. In subsequent examples, the srta register receive address will be referred to as the "device number." **Note:** It is possible to assign more than one receive address or a duplicate receive address to a DSP16/DSP16A device, but the examples given assume a unique receive address.

www.DataSheet4U.com

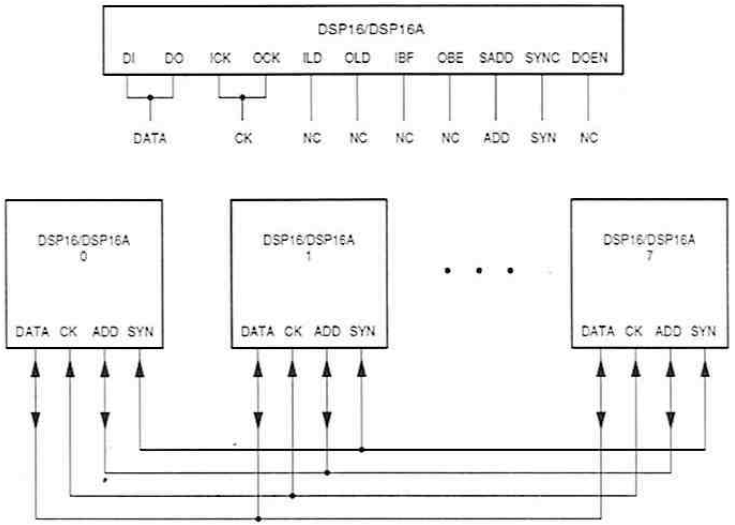

Figure 5-5. DSP16/DSP16A Multiprocessor Connections

If the serial address coming from the bidirectional SADD line of the transmitting device matches the address of one of the other devices, the input is loaded into that device's input buffer and its IBF flag is set at the end of the transmission. In order to read in the new data, an interrupt could take place based on the IBF flag. Note that the address is eight bits wide with eight DSP16/DSP16A devices (maximum) in the multiprocessor configuration. This means there is one address bit per DSP16/DSP16A device. The srta register has one address bit per device in order to allow transmissions to more than one device at a time. A broadcast mode sending data from one device to all others is accomplished by setting all bits high on the transmission field of srta.

Table 5-3. Time-Division Multiplexed Slot (tdms) Register

Bit	9	8	7	6	5	4	3	2	1	0
Field	SYNCSP	MODE	TRANSMIT SLOT							SYNC

Field	Value	Result/Description
SYNCSP	0	SYNC = ICK/OCK† ÷ 128. ‡
	1	SYNC = ICK/OCK† ÷ 256.
MODE	0	Multiprocessor mode off. DOEN is an input (passive mode).
	1	Multiprocessor mode on. DOEN is an output (active mode).
TRANSMIT SLOT	1xxxxxxx	Transmit slot 7.
	x1xxxxxx	Transmit slot 6.
	xx1xxxxx	Transmit slot 5.
	xxx1xxxx	Transmit slot 4.
	xxxx1xxx	Transmit slot 3.
	xxxxx1xx	Transmit slot 2.
	xxxxxx1x	Transmit slot 1.
SYNC	xxxxxxx1	Transmit slot 0. SYNC is an output (active mode).
	xxxxxxx0	SYNC is an input (passive mode).

† See sioc register, LD field.

‡ Select this mode when in multiprocessor mode.

Typically, the time-division multiplexed slot register (tdms) is set up at the beginning of a program and does not change for each of the devices in the multiprocessor system. If the time slot needs to be changed, it is imperative that each processor still have its own unique time slot. The falling edge of SYNC (the TDM frame sync) is used to update all the new time slots except time slot 0, which is updated in the next cycle of SYNC.

During reset, the tdms register resets to all zeros, disabling the multiprocessor mode by default. The srta register is unaltered by reset.

Table 5-4. Serial Receive/Transmit Address (srta) Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RECEIVE ADDRESS								TRANSMIT ADDRESS							

Field	Value	Result/Description
RECEIVE ADDRESS	1xxxxxxx	Receive address 7.
	x1xxxxxx	Receive address 6.
	xx1xxxxx	Receive address 5.
	xxx1xxxx	Receive address 4.
	xxxx1xxx	Receive address 3.
	xxxxx1xx	Receive address 2.
	xxxxxx1x	Receive address 1.
	xxxxxxx1	Receive address 0.
TRANSMIT ADDRESS	1xxxxxxx	Transmit address 7.
	x1xxxxxx	Transmit address 6.
	xx1xxxxx	Transmit address 5.
	xxx1xxxx	Transmit address 4.
	xxxx1xxx	Transmit address 3.
	xxxxx1xx	Transmit address 2.
	xxxxxx1x	Transmit address 1.
	xxxxxxx1	Transmit address 0.

Figure 5-6 shows the operation of a system using eight DSP16/DSP16A devices in a multiprocessor configuration. The settings used for the tdms and srta registers are shown in order to illustrate the current state of these registers during each I/O operation. The following describes the operation shown in Figure 5-6.

Time Slot **Actions**

0 In preparation for time slot 0 (left-most column), the tdms register of device number 7 has been initialized so that it can transmit in time slot zero. This situation also forces the device to generate the frame sync of the I/O stream. The srta register of device 7 has been set so that it can transmit to device 3 and receive address 7. The serial data register (SDX) of device 7 contains the data to be transmitted.

During time slot 0, the data from device 7 is transmitted on the TDM channel. Device 3 recognizes its address on the serial address line (SADD) and accepts the data into its SDX register, which is subsequently read by the command *r0 = sdx. All other devices ignore this transaction, since the transmit address was not theirs.

1 No actions in time slot 1.

2 In preparation for time-slot 2, the tdms register of device 2 has been initialized so that during time slot 2, device 2 will transmit to device 5.

During time slot 2, the data from device 2 is transmitted on the TDM channel. Device 5 recognizes the address on the SADD and accepts the data into its sdx register, which is then read by the command *r1++ = sdx.

3 No actions in time slot 3.

4 No actions in time slot 4.

5 In preparation for time slot 5, device 0 has been initialized so that it will transmit in this time slot to all other devices. Devices 1, 4, and 6, which have not been previously mentioned, are ready to receive data assigned to their respective addresses. Devices 2, 3, 5, and 7, which were initialized earlier, are also ready to receive data.

During time slot 5, the data in device 0 is transmitted on the TDM channel. Every device address is represented on the SADD line, and all devices will accept the data.

6 No actions in time slot 6.

7 No actions in time slot 7.

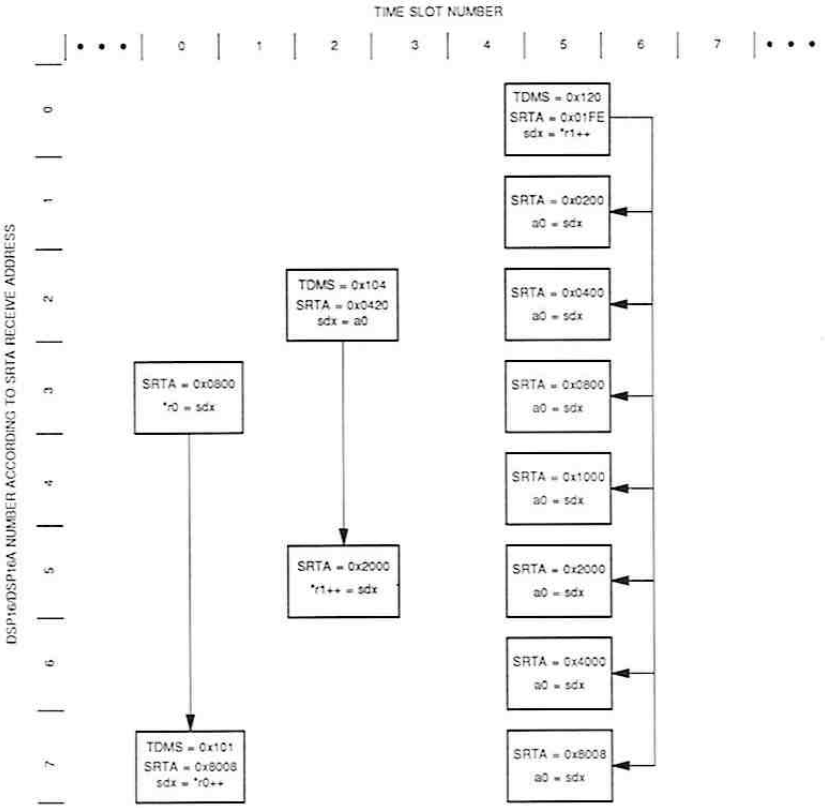

Figure 5-6. DSP16/DSP16A Multiprocessor Communications

5.6.1 Suggested Multiprocessor Configuration

In the suggested configuration, the DSP16/DSP16A device supplying the SYNC signal also supplies the ICK and OCK signals; the remaining DSPs are configured for passive SYNC, ICK, and OCK signals. All DSPs have active ILD and OLD signals.

For the DSP16/DSP16A device with the given transmit slot, the following parameters should be configured as shown:

Parameter	Transmit Slot 0	Transmit Slot 1—7
SYNC	Active	Passive
ICK	Active	Passive
OCK	Active	Passive
ILD	Active	Active
OLD	Active	Active

To achieve the configuration shown above, the following registers in the DSPs should be set as shown:

Register	Transmit Slot 0	Transmit Slot 1—7
sioc	0x23C	0x230
tdms	0x101	0x1XX
srta	0xXXX	0xXXX

Note: An "X" indicates that the number is dependent on the specific application.

The interrupt on IBF must also be enabled in the pioc register of each DSP16/DSP16A device to allow the devices to detect and process an input.

5.7 SERIAL I/O TIMING DIAGRAMS

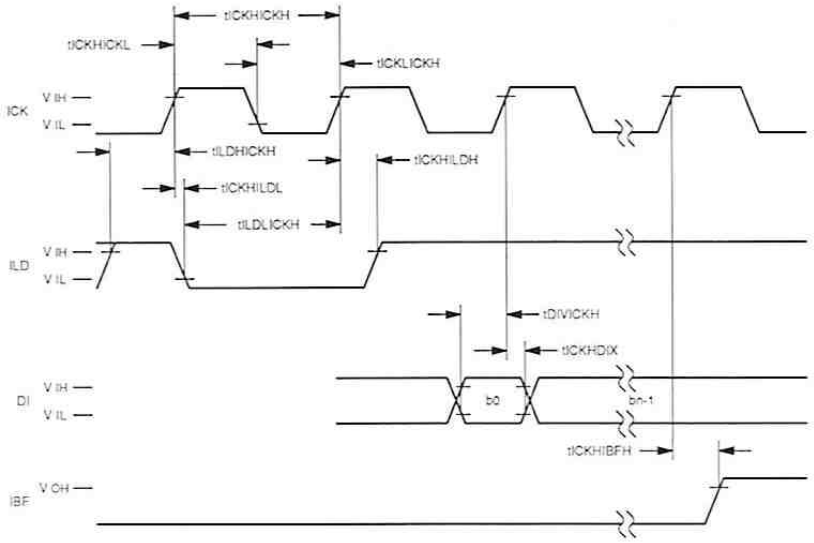

Figure 5-7. Serial Input Timing

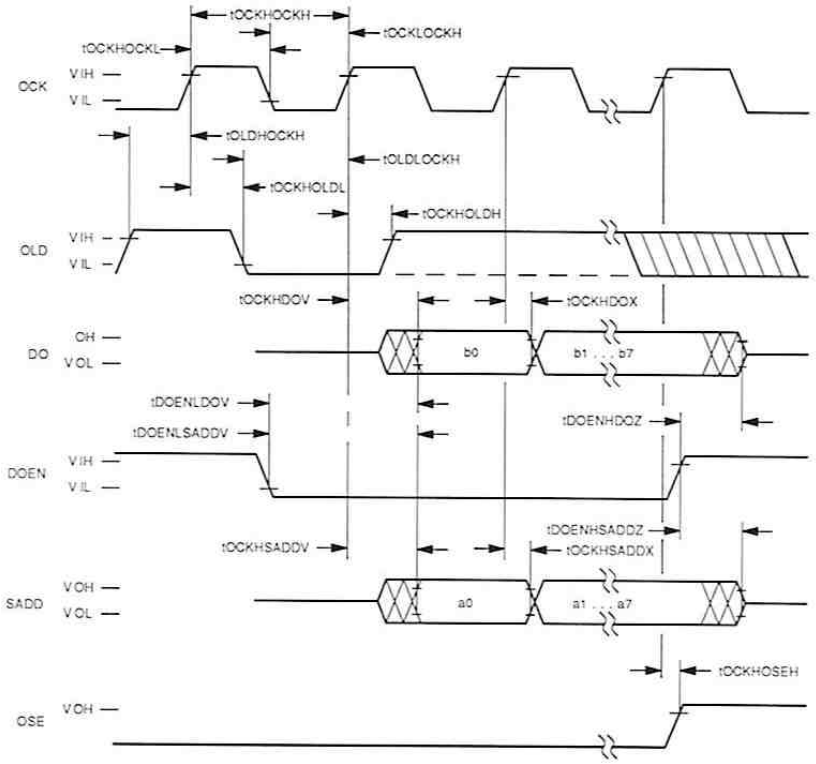

Figure 5-8. Serial Output Timing – 8 Bits

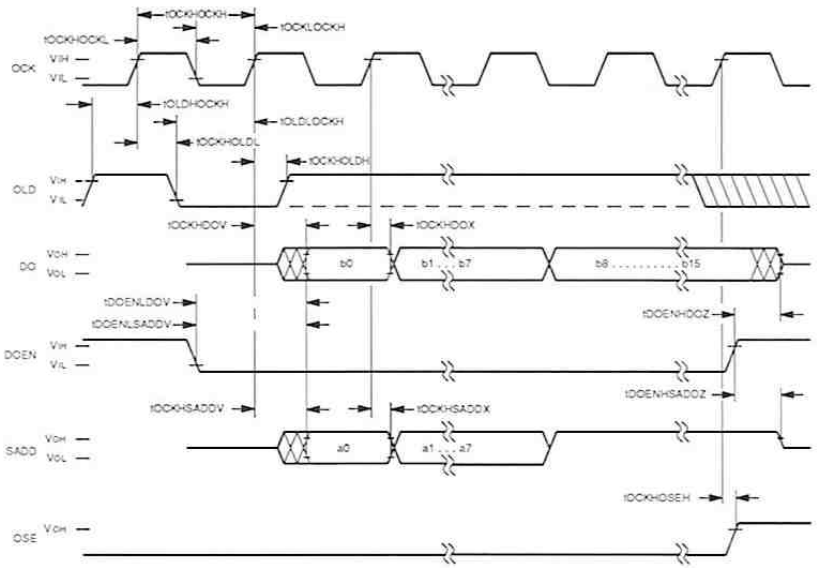

Figure 5-9. Serial Output Timing – 16 Bits

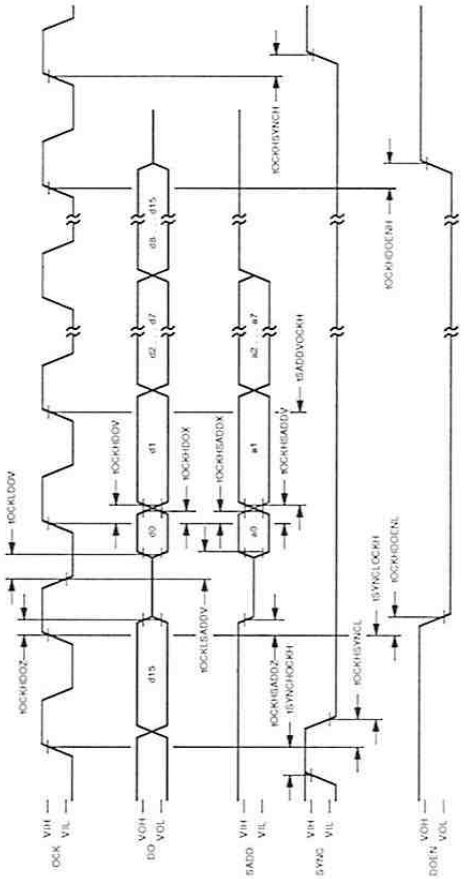

Figure 5-10. Multiprocessor Timing