SECTION 8

INTERFACING TO DSPs

- Parallel Interfacing to DSP Processors: Reading Data From Memory-Mapped Peripheral ADCs
- Parallel Interfacing to DSP Processors: Writing Data to Memory-Mapped DACs
- Serial Interfacing to DSP Processors
- Interfacing I/O Ports, Analog Front Ends, and Codecs to DSPs
- DSP System Interface

SECTION 8
INTERFACING TO DSPs
Walt Kester, Dan King

INTRODUCTION

As the technology in the rapidly growing field of mixed signal processing evolves, more highly integrated DSP products are being introduced (such as the ADSP-21ESP202) which contain on-chip ADCs and DACs as well as the DSP, thereby eliminating most component-level interface problems. Stand-alone ADCs and DACs are now available with interfaces especially designed for DSP chips, thereby minimizing or eliminating external interface support or *glue* logic. High performance sigma-delta ADCs and DACs are currently available in the same package (called a *CODEC* or COder/DECcoder) such as the AD73311 and AD73322. These products are also designed to require minimum glue logic when interfacing to the most common DSP chips. This section discusses the various data transfer and timing issues associated with the various interfaces.

PARALLEL INTERFACING TO DSP PROCESSORS: *READING* DATA FROM MEMORY-MAPPED PERIPHERAL ADCS

Interfacing an ADC or a DAC to a fast DSP parallel requires an understanding of how the DSP processor reads data from a memory-mapped peripheral (the ADC) and how the DSP processor writes data to a memory-mapped peripheral (the DAC). We will first consider some general timing requirements for reading and writing data. It should be noted that the same concepts presented here regarding ADCs and DACs apply equally when reading and writing from/to external memory.

A block diagram of a typical parallel DSP interface to an external ADC is shown in Figure 8.1. This diagram has been greatly simplified to show only those signals associated with *reading* data from an external memory-mapped peripheral device. The timing diagram for the ADSP-21xx read-cycle is shown in Figure 8.2.

In this example it is assumed that the ADC is sampling at a continuous rate which is controlled by the external sampling clock, not the internal DSP clock. Using a separate clock for the ADC is the preferred method, since the DSP clock may be noisy and introduce jitter in the ADC sampling process, thereby increasing the noise level.

Assertion of the sampling clock at the ADC *convert start* input initiates the conversion process (step 1). The leading (or trailing) edge of this pulse causes the internal ADC sample-and-hold to switch from the sampling mode to the hold mode so that the conversion process can take place. When the conversion is complete, the *conversion complete* output of the ADC is asserted (step 2). The *read* process thus begins when this signal is applied to the *processor interrupt request line* ($\overline{\text{IRQ}}$) of the DSP. The processor then places the address of the peripheral initiating the

interrupt request (the ADC) on the *memory address bus* (A0 - A13) (step 3). At the same time, the processor asserts a *memory select line* ($\overline{\text{DMS}}$ is shown here) (step 4). The two internal address buses of the ADSP-21xx (program memory address bus and data memory address bus) share a single external address bus, and the two internal data buses (program memory data bus and data memory data bus) share a single external data bus. The *boot memory select* ($\overline{\text{BMS}}$), *data memory select* ($\overline{\text{DMS}}$), *program memory select* ($\overline{\text{PMS}}$) and *input/output memory select* ($\overline{\text{IOMS}}$) signals indicate which memory space the external buses are being used for. These signals are typically used to enable an external *address decoder* as shown in Figure 8.1. The output of the address decoder drives the *chip select* input of the peripheral device (step 5).

The *memory read* ($\overline{\text{RD}}$) is asserted t_{ASR} ns after the $\overline{\text{DMS}}$ line is asserted (step 6). The sum of the address decode delay plus the peripheral chip select setup time should be less than t_{ASR} in order to take full advantage of the $\overline{\text{RD}}$ low-time. The $\overline{\text{RD}}$ line remains low for t_{RP} ns. The *memory read* signal is used to enable the three-state parallel data outputs of the peripheral device (step 7). The $\overline{\text{RD}}$ line is connected to the appropriate pin on the peripheral device usually called *output enable* or *read*. The rising edge of the $\overline{\text{RD}}$ signal is used to clock the data on the data bus into the DSP processor (step 8). After the rising edge of the $\overline{\text{RD}}$ signal, the data on the data bus must remain valid for t_{RDH} ns, the data hold time. In the case of most members of the ADSP-21xx family, this specification value is 0ns.

The key timing requirements for the peripheral device are shown in Figure 8.3. Values are given for the ADSP-2189M DSP operating at 75MHz.

ADC TO ADSP-21xx-FAMILY PARALLEL INTERFACE

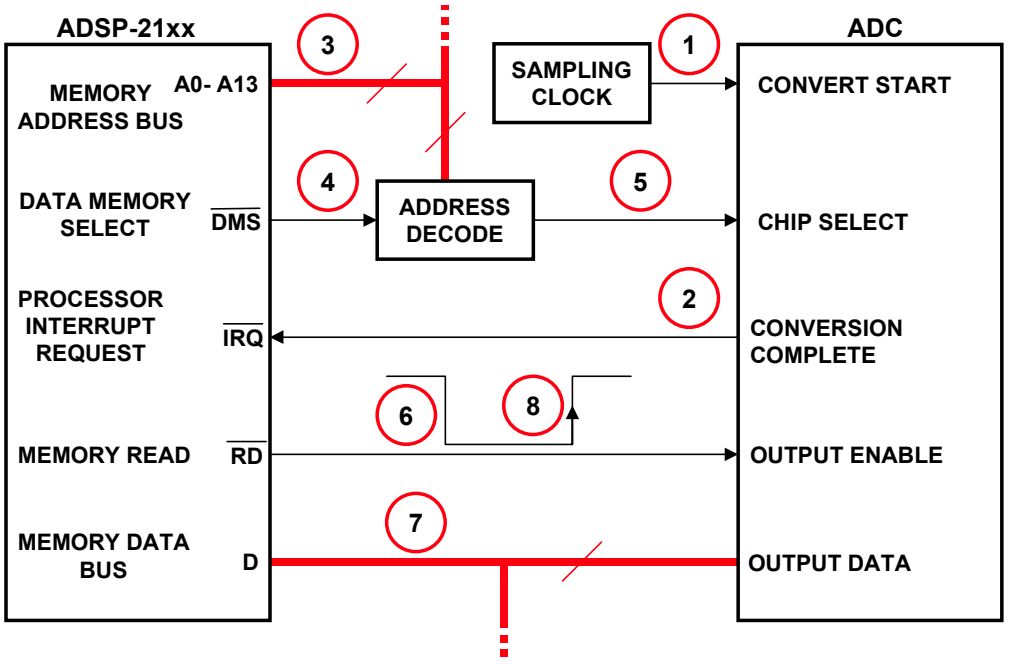


Figure 8.1

ADSP-21xx FAMILY MEMORY READ TIMING

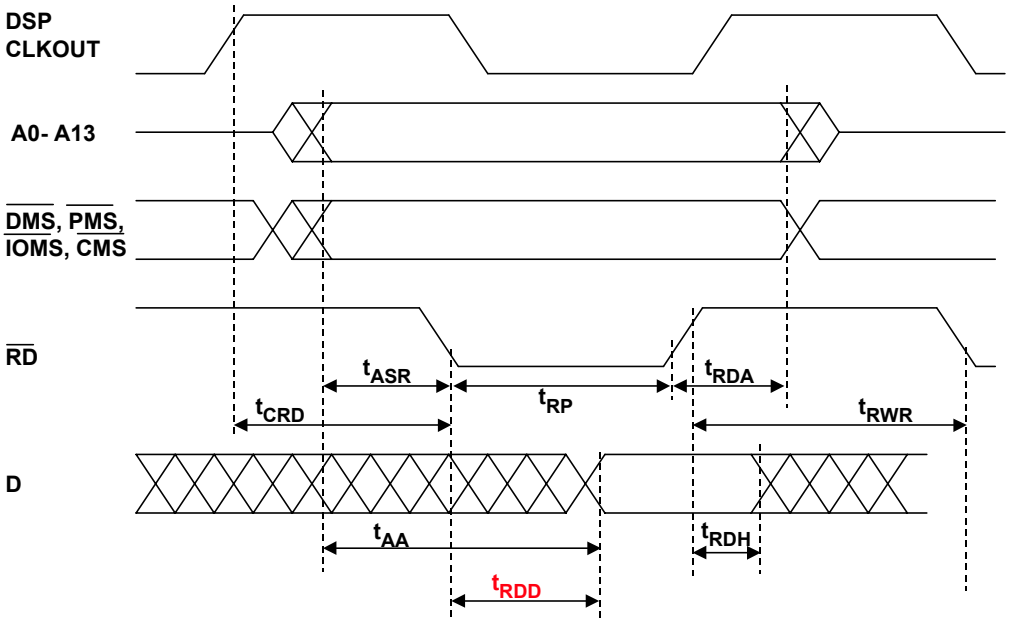


Figure 8.2

PARALLEL PERIPHERAL DEVICE READ INTERFACE KEY REQUIREMENTS

- Peripheral Device Data Outputs Must Be Three-State Compatible
- Address Decode Delay + Peripheral Chip Select Setup Time Must Be Less Than Address and Memory Select Setup Time t_{ASR} (0.325ns min for ADSP-2189M)
- For Zero Wait-State Access, the Time from a Negative-Going Edge of Read Signal ($\overline{RD}$) to Output Data Valid Must be Less than t_{RDD} (1.65ns max for ADSP-2189M Operating at 75MHz) or Software Wait-States Must be Added, or Processor Clock Frequency Reduced
- Output Data from Peripheral must Remain Valid for t_{RDH} from the Rising Edge of Read Signal ($\overline{RD}$) (0ns for ADSP-2189M)
- Peripheral Device Must Accept Minimum Output Enable Pulse Width of t_{RP} (3.65ns for ADSP-2189M Operating at 75MHz) or Software Wait-States Must be Added, or Processor Clock Frequency Reduced

Figure 8.3

The DSP t_{RDD} specification determines the peripheral device data access time requirement. In the case of the ADSP-2189M, t_{RDD} = 1.65ns minimum at 75MHz. If the access time of the peripheral is greater than this, wait states must be added or the processor speed reduced. This is a relatively common situation when interfacing external memory or ADCs to fast DSPs. The relationship between these timing parameters for the ADSP-2189M is given by the equations shown in Figure 8.4. Note that these specifications are dependent on the DSP clock frequency.

ADSP-2189M PARALLEL READ TIMING AT 75MHz

- **t_{CK} = Processor Clock Period (13.3ns)**
- **t_{ASR} = Address and Memory Select Setup Before Read Low = $0.25t_{CK}$ – 3ns Minimum**
- **t_{RDD} = Read Low to Data Valid = $0.5t_{CK}$ – 5ns + # wait states × t_{CK} Maximum**
- **t_{RDH} = Data Hold from Read High = 0ns Minimum**
- **t_{RP} = Read Pulse Width = $0.5t_{CK}$ – 3ns + # wait states × t_{CK} Minimum**

Figure 8.4

The ADSP-2189M can easily be interfaced to slow peripheral devices using its programmable wait state generation capability. Three registers control wait state generation for boot, program, data and I/O memory spaces. You can specify 0 to 15 wait states for each parallel memory interface. Each wait state added increases the allowable external data memory access time by an amount equal to the processor clock period (13.3ns for the ADSP-2189M operating at 75MHz). In this example, the *data memory address*, $\overline{\text{DMS}}$, and $\overline{\text{RD}}$ lines are all held stable for an additional amount of time equal to the duration of the wait states.

The AD7854/AD7854L is a 12 bit, 200/100kSPS ADC which operates in the parallel mode. It operates on a single +3V to +5.5V supply and dissipates only 5.5mW (+3V supply, AD7854L). An automatic power-down after conversion feature reduces this to 650µW.

A functional block diagram of the AD7854/AD7854L is shown in Figure 8.5. The AD7854/AD7854L uses a successive approximation architecture which is based on a charge redistribution (switched capacitor) DAC. A calibration mode removes offset and gain errors. The key interface timing specifications for the AD7854/AD7854L and the ADSP-2189M are compared in Figure 8.6. Specifications for the ADSP-2189M are given for a clock frequency of 75MHz.

Examining the timing specifications shown in Figure 8.6 reveals that for the timing between the devices to be compatible, 5 software wait-states must be programmed into the ADSP-2189M. This increases t_{RDD} to 68.15ns which is greater than the data access time of the AD7854/AD7854L (t_8 = 50ns max.). The read pulse, t_{RP}, is likewise increased to 70.15ns which meets the ADC's read pulse width requirement (t_7 = 70ns min.). Unless the memory-mapped peripheral has an extremely short access time, wait states are generally required, whether interfacing to ADCs, DACs, or external memory.

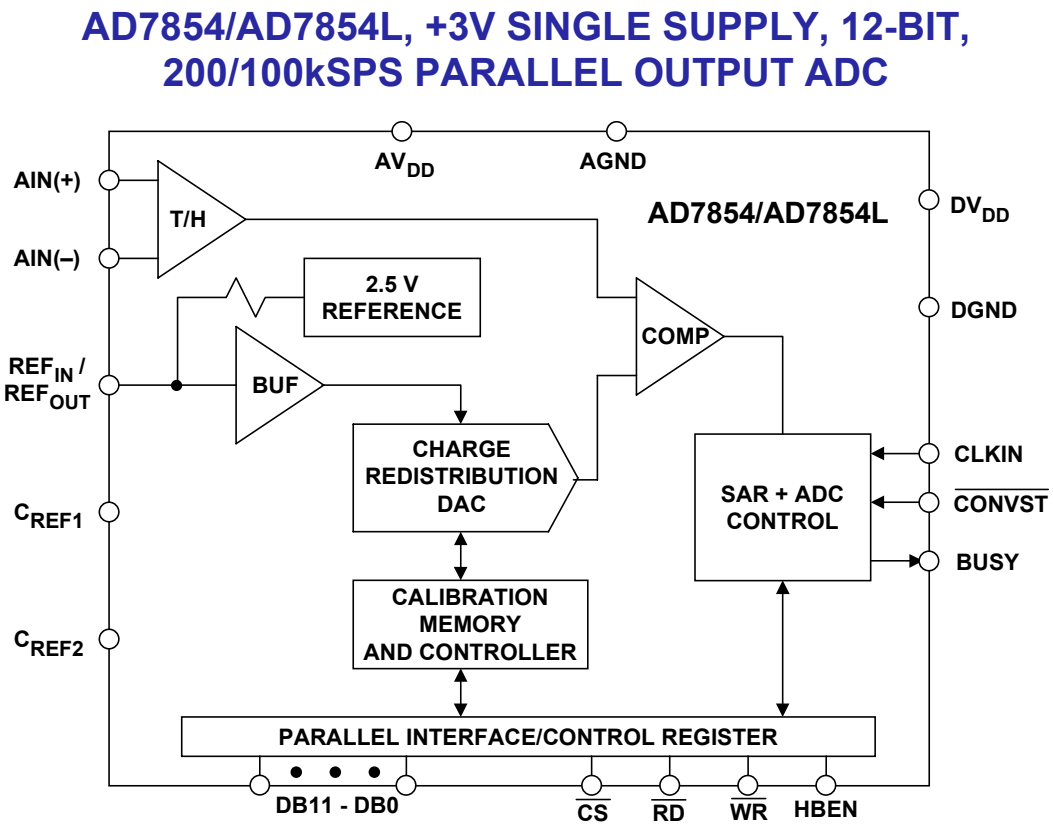


Figure 8.5

ADSP-2189M AND AD7854/AD7854L PARALLEL READ INTERFACE TIMING SPECIFICATION COMPARISON

ADSP-2189M Processor (75MHz)	AD7854/AD7854L ADC
t_{ASR} (Data Address, Memory Select Setup Time Before $\overline{RD}$ Low) = 0.325ns min	t_5 ($\overline{CS}$ to $\overline{RD}$ Setup Time = 0ns min (Must Add Address Decode Time to this Value)
t_{RP} ($\overline{RD}$ Pulse Width) = 3.65ns + # wait states × 13.3ns min = 70.15ns min	t_7 ($\overline{RD}$ Pulse Width) = 70ns min
t_{RDD} ($\overline{RD}$ Low to Data Valid) = 1.65ns + # wait states × 13.3ns min = 68.15ns min	t_8 (Data Access Time After $\overline{RD}$) = 50ns max
t_{RDH} (Data Hold from $\overline{RD}$ High) = 0ns min	t_9 (Bus Relinquish Time After $\overline{RD}$) = 5ns min / 40ns max

NOTES:
(1) Adding 5 wait-states to the ADSP-2189M increases t_{RP} to 70.15ns which is greater than t_7 (70ns) and meets the t_8 (50ns) requirement.
(2) t_9 max (40ns) may cause bus contention if a write cycle immediately follows the read cycle.

Figure 8.6

A simplified interface diagram for the two devices is shown in Figure 8.7. The conversion complete signal from the AD7854/AD7854L corresponds to the BUSY output pin. Notice that the configuration allows the DSP to write data to the AD7854/AD7854L parallel interface control register. This is needed in order to set various options in the AD7854/AD7854L and perform the calibration routines. In normal operation, however, data is read from the AD7854/AD7854L as described above. Writing to external parallel memory-mapped peripherals is discussed later in this section.

Parallel interfaces between other DSP processors and external peripherals can be designed in a similar manner by carefully examining the timing specifications for all appropriate signals for each device. The data sheets for most ADCs contain sufficient information in the application section to interface them to the DSPs.

AD7854/AD7854L ADC PARALLEL INTERFACE TO ADSP-2189M

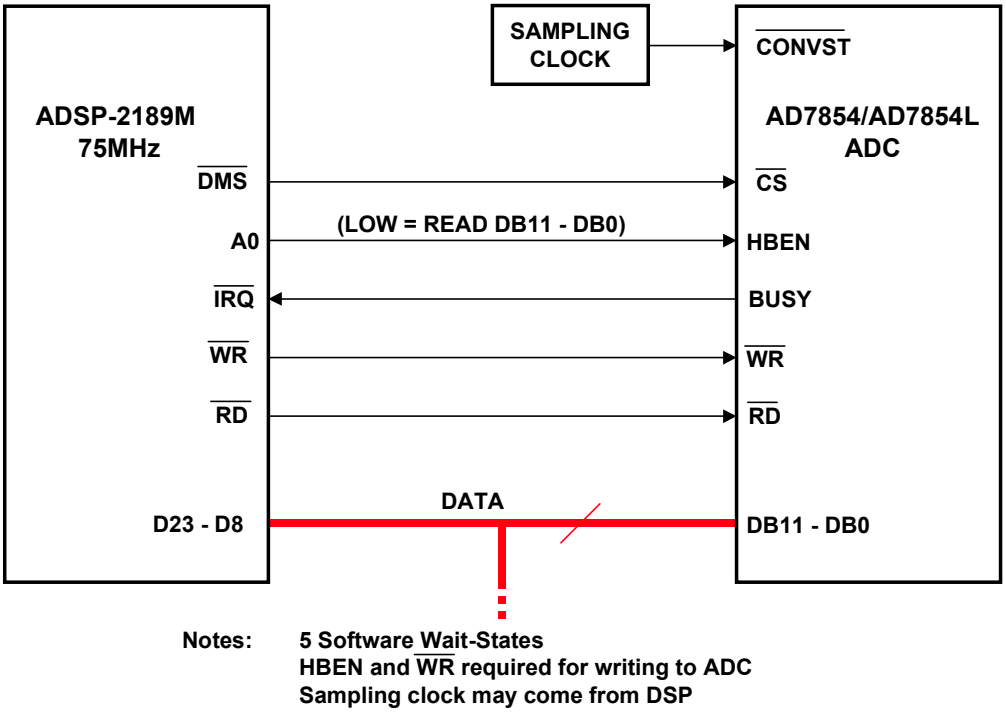


Figure 8.7

PARALLEL INTERFACING TO DSP PROCESSORS: *WRITING* DATA TO MEMORY-MAPPED DACS

A simplified block diagram of a typical DSP interface to a parallel peripheral device (such as a DAC is shown in Figure 8.8. The memory-write cycle timing diagram for the ADSP-21xx-family is shown in Figure 8.9.

In most real-time applications, the DAC is operated continuously from a stable sampling clock. Most DACs for these applications have double buffering: an input latch to handle the asynchronous DSP interface, followed by a second latch (called the DAC latch) which drives the DAC current switches. The DAC latch strobe is derived from an external stable sampling clock. In addition to clocking the DAC latch, the DAC latch strobe is also used to generate a processor interrupt to the DSP which indicates that the DAC is ready for a new input data word.

DAC TO ADSP-21xx-FAMILY PARALLEL INTERFACE

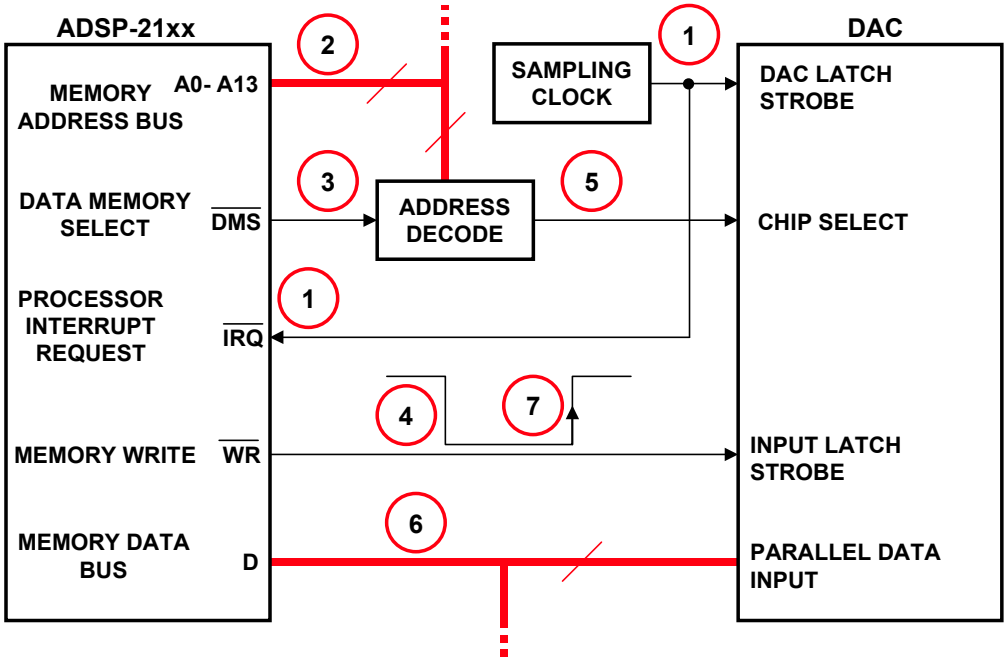


Figure 8.8

ADSP-21xx FAMILY MEMORY WRITE TIMING

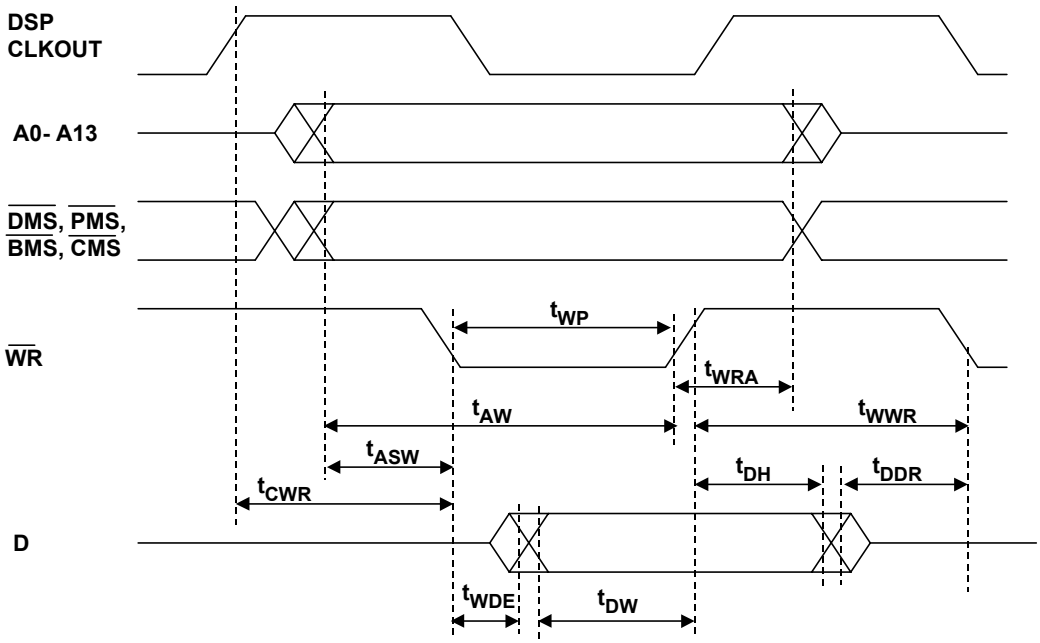


Figure 8.9

The write process is thus initiated by the peripheral device asserting the DSP *interrupt request* line indicating that the peripheral is ready to accept a new parallel data word (step 1). The DSP then places the address of the peripheral device on the *address bus* (step 2) and asserts a *memory select* line ($\overline{\text{DMS}}$ is shown here) (step 3). This causes the output of the address decoder to assert the *chip select* input to the peripheral (step 5). The *write* ($\overline{\text{WR}}$) output of the DSP is asserted t_{ASW} ns after the negative-going edge of the $\overline{\text{DMS}}$ signal (step 4). The width of the $\overline{\text{WR}}$ pulse is t_{WP} ns. Data is placed on the data bus (D) and is valid t_{DW} ns before the $\overline{\text{WR}}$ line goes high (step 6). The positive-going transition of the $\overline{\text{WR}}$ line is used to clock the data on the data bus (D) into the external parallel memory (step 7). The data on the data bus remains valid for t_{DH} ns after the positive-going edge of the $\overline{\text{WR}}$ signal.

The key timing requirements for writing to the peripheral device are shown in Figure 8.10. The key specification is t_{WP} , the write pulse width. All but the fastest peripheral devices will require wait states to be added due to their longer data access times. Figure 8.11 shows the key timing specifications for the ADSP-2189M. Note that they are all related to the processor clock frequency.

PARALLEL PERIPHERAL DEVICES WRITE INTERFACE KEY REQUIREMENTS

- **Address Decode Delay + Peripheral Chip Select Setup Time Must Be Less Than Address and Memory Select Setup Time t_{ASW} (0.325ns for ADSP-2189M Operating at 75MHz)**
- **For Zero Wait-State Access, Input Data *Setup* Time Must be Less Than t_{DW} (2.65ns for ADSP-2189M Operating at 75MHz) or Software Wait-States Must be Added, or Processor Clock Frequency Reduced**
- **Input Data *Hold* Time Must be Less Than t_{DH} (2.325ns for ADSP-2189M Operating at 75MHz)**
- **Peripheral Device Must Accept Input Write Clock Pulse of Width t_{WP} (3.65ns min for ADSP-2189M Operating at 75MHz) or Software-Wait States Must be Added, or Processor Clock Frequency Reduced**

Figure 8.10

ADSP-2189M PARALLEL WRITE TIMING

- t_{CK} = Processor Clock Period (13.3ns)
- t_{ASW} = Address and Memory Select Before $\overline{WR}$ Low
  = $0.25t_{CK}$ – 3ns Minimum
- t_{DW} = Data Setup Before $\overline{WR}$ High = $0.5t_{CK}$ – 4ns + # Wait States × t_{CK}
- t_{DH} = Data Hold After $\overline{WR}$ High = $0.25t_{CK}$ – 1ns
- t_{WP} = $\overline{WR}$ Pulse Width = $0.5t_{CK}$ – 3ns + # Wait States × t_{CK} Minimum

Figure 8.11

The AD5340 is a 12 bit 100kSPS DAC which has a parallel data interface. It operates on a single +2.5V to +5.5V supply and dissipates only 345µW (+3V supply). A power-down mode further reduces the power to 0.24µW. The part incorporates an on-chip output buffer which can drive the output to both supply rails. The AD5340 allows the choice of a buffered or unbuffered reference input. The device has a power-on reset circuit that ensures that the DAC output powers on at 0V and remains there until valid data is written to the part. A block diagram is shown in Figure 8.12. The input is double buffered. The key interface timing specifications for the two devices are compared in Figure 8.13. Specifications for the ADSP-2189M are given for a clock frequency of 75MHz.

AD5340 12-BIT, 100kSPS PARALLEL INPUT DAC

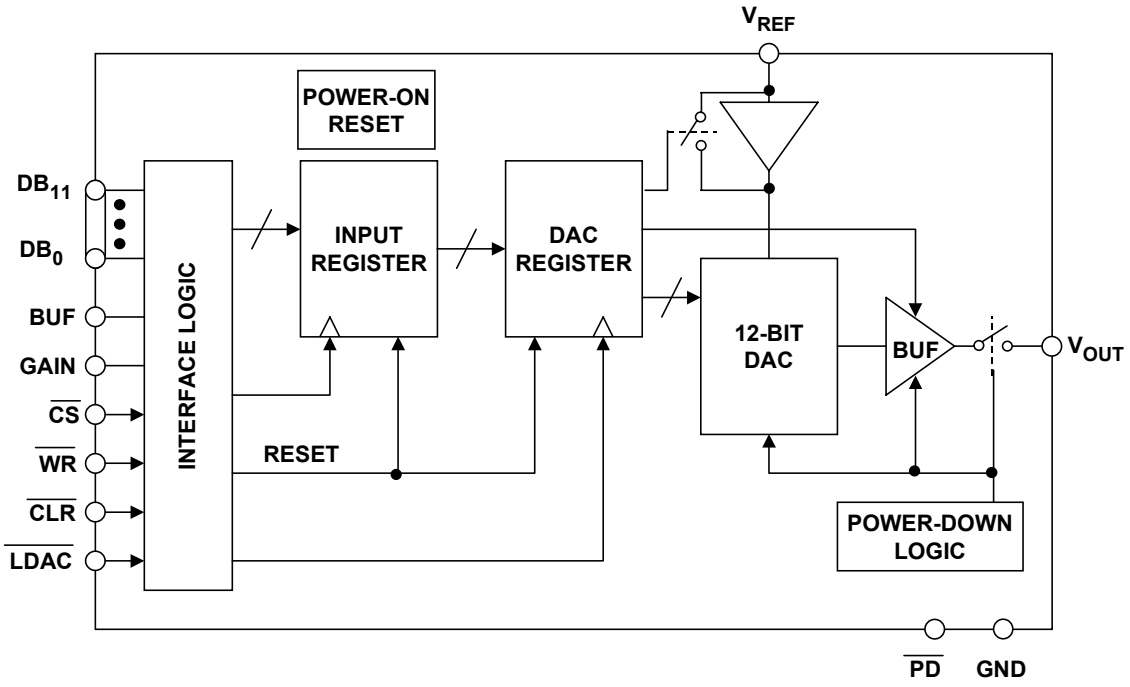


Figure 8.12

ADSP-2189M AND AD5340 PARALLEL WRITE INTERFACE TIMING SPECIFICATIONS

ADSP-2189M PROCESSOR (75MHz)	AD5340 DAC
t_{ASW} (Address and Data Memory Select Setup Before $\overline{WR}$ Low) = 0.325ns min	t_1 ($\overline{CS}$ to $\overline{WR}$ Setup Time) = 0ns min
t_{WP} ($\overline{WR}$ Pulse Width) = 3.65ns + # Wait States × 13.3ns min = 30.25ns min	t_3 ($\overline{WR}$ Pulse Width) = 20ns min
t_{DW} (Data Setup Before $\overline{WR}$ High) = 2.65ns + # Wait States × 13.3ns min = 29.25ns min	t_4 (Data Valid to $\overline{WR}$ Setup Time) = 5ns min
t_{DH} (Data Hold After $\overline{WR}$ High) = 2.325ns min	t_5 (Data Valid to $\overline{WR}$ Hold Time) = 4.5ns min

NOTE: Adding 2 wait states to the ADSP-2189M increases t_{WP} to 30.25ns and t_{DW} to 29.25ns which is greater than t_3 (20ns) and t_4 (5ns) respectively.

Figure 8.13

Examining the timing specifications shown in Figure 8.13 reveals that for the timing between the devices to be compatible, two software wait states must be programmed into the ADSP-2189M. This increases the width of $\overline{WR}$ to 30.25ns which is greater than the minimum required AD5340 write pulse width (20ns). The data setup time of 5ns for the AD5340 is also met by adding two wait states. A simplified interface diagram for the two devices is shown in Figure 8.14.

Parallel interfaces with other DSP processors can be designed in a similar manner by carefully examining the timing specifications for all appropriate signals for each device.

AD5340 DAC PARALLEL INTERFACE TO ADSP-2189M

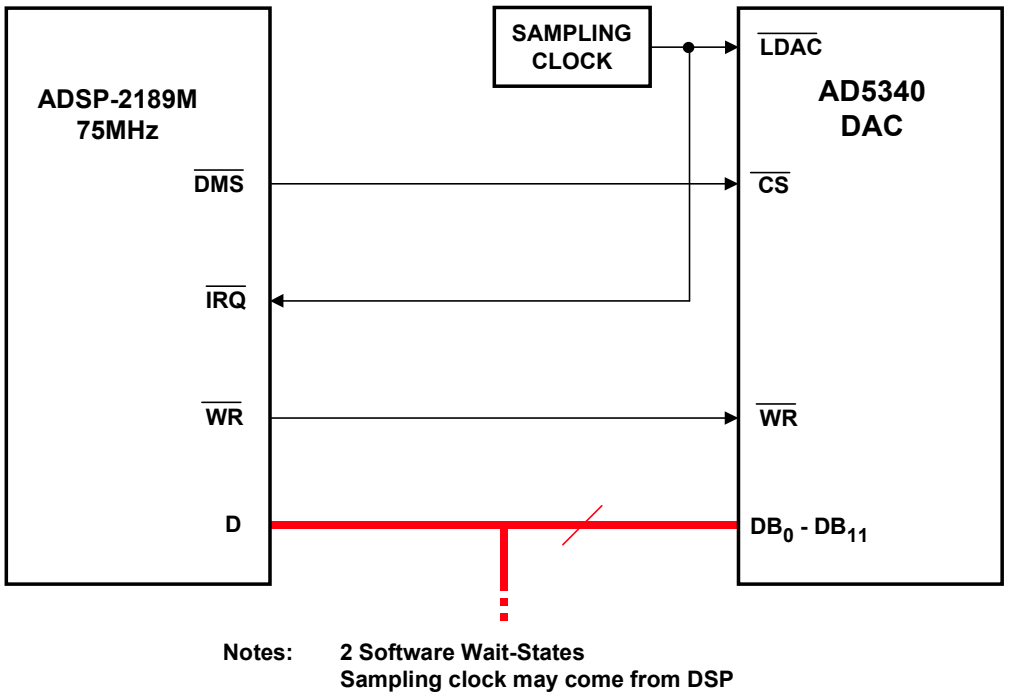


Figure 8.14

SERIAL INTERFACING TO DSP PROCESSORS

DSP processors which have serial ports (such as the ADSP-21xx family) provide a simple interface to peripheral ADCs and DACs. Use of the serial port eliminates the need for using large parallel buses to connect the ADCs and DACs to the DSP. In order to understand serial data transfer better, we will first examine the serial port operation of the ADSP-21xx series.

A block diagram of one of the two serial ports of the ADSP-21xx is shown in Figure 8.15. The *Transmit* (TX) and *Receive* (RX) registers are identified by name in the ADSP-21xx assembly language, and are not memory mapped.

ADSP-21xx FAMILY SERIAL PORT BLOCK DIAGRAM

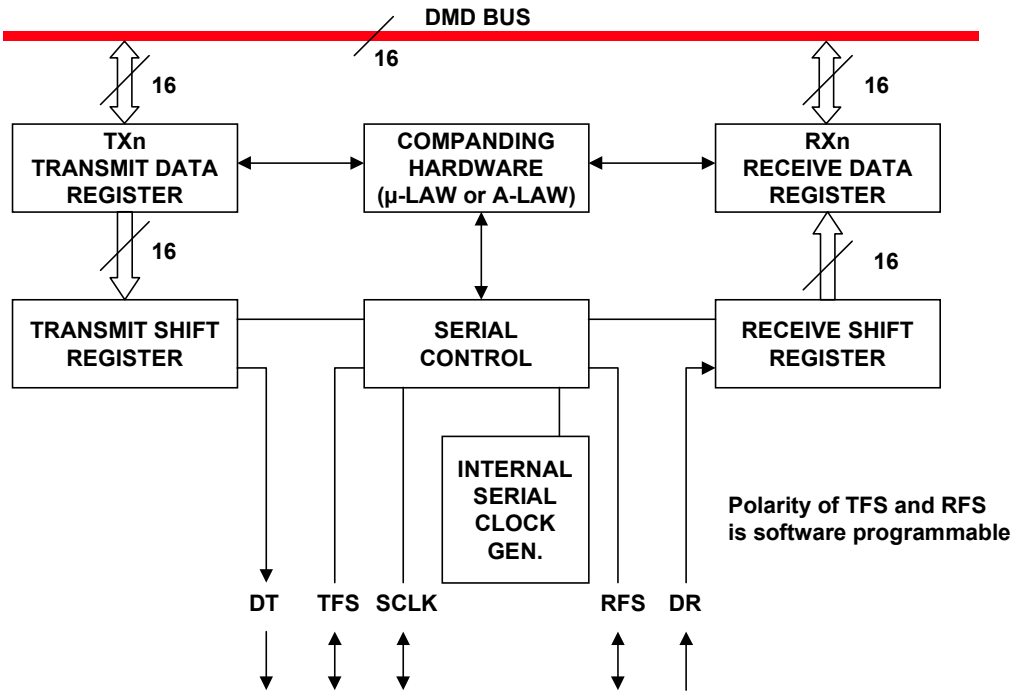


Figure 8.15

ADSP-21xx FAMILY SERIAL PORT FEATURES

- Separate Transmit and Receive Sections for Each Port
- Double-Buffered Transmit and Receive Registers
- Serial Clock Can be Internally or Externally Generated
- Transmit and Receive Frame Sync Signals Can be Internally or Externally Generated
- Serial Data Words of 3 to 16 Bits Supported
- Automatically Generated Processor Interrupts
- Hardware Companding Requires no Software Overhead

Figure 8.16

In the receiving portion of the serial port, the *receive frame sync* (RFS) signal initiates reception. The serial *receive data* (DR) from the external device (ADC) is transferred into the *receive shift register* one bit at a time. The negative-going edge of the *serial clock* (SCLK) is used to clock the serial data from the external device into the *receive shift register*. When a complete word has been received, it is written to the *receive data register* (RX), and the receive interrupt for that serial port is generated. The *receive data register* is then read by the processor.

Writing to the *transmit data register* readies the serial port for transmission. The *transmit frame sync* (TFS) signal initiates transmission. The value in *the transmit data register* (TX) is then written to the internal *transmit shift register*. The data in the *transmit shift register* is sent to the peripheral device (DAC) one bit at a time, and the positive-going edge of the *serial clock* (SCLK) is used to clock the serial *transmit data* (DT) into the external device. When the first bit has been transferred, the serial port generates the transmit interrupt. The *transmit data register* can then be written with new data, even though the transmission of the previous data is not complete.

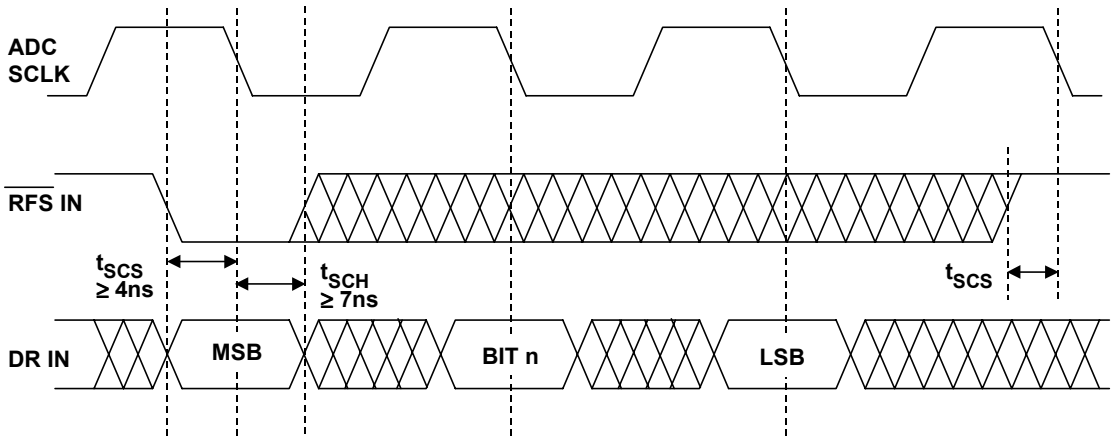
In the *normal* framing mode, the frame sync signal (RFS or TFS) is checked at the falling edge of SCLK. If the framing signal is asserted, data is available (transmit mode) or latched (receive mode) on the *next* falling edge of SCLK. The framing signal is not checked again until the word has been transmitted or received. In the *alternate* framing mode, the framing signal is asserted in the *same* SCLK cycle as the first bit of a word. The data bits are latched on the falling edge of SCLK, but the framing signal is checked only on the first bit. Internally-generated framing signals remain asserted for the length of the serial word. The *alternate* framing mode of the serial port in the ADSP-21xx is normally used to receive data from ADCs and transmit data to DACs.

The serial ports of the ADSP-21xx family are extremely versatile. The TFS, RFS, or SCLK signals can be generated from the ADSP-21xx clock (master mode) or generated externally (slave mode). The polarity of these signals can be reversed with software, thereby allowing more interface flexibility. The port also contains μ-law and A-law companding hardware for voiceband telecommunications applications.

SERIAL ADC TO DSP INTERFACE

A timing diagram of the ADSP-2189M serial port operating in the receive mode (alternate framing) is shown in Figure 8.17. The first negative-going edge of the SCLK to occur after the negative-going edge of the $\overline{\text{RFS}}$, clocks the MSB data from the ADC into the serial input latch. The process continues until all serial bits have been transferred into the serial input latch. The key timing specifications of concern are the serial data setup (t_{SCS}) and hold times (t_{SCH}) with respect to the negative-going edge of the SCLK. In the case of the ADSP-2189M, these values are 4ns and 7ns, respectively . The latest generation ADCs with high speed serial clocks will have no trouble meeting these specifications, even at the maximum serial data transfer rate.

ADSP-2189M SERIAL PORT RECEIVE TIMING



ALTERNATE FRAMING MODE, ADC IS MASTER

Figure 8.17

The AD7853/AD7853L is a 12 bit, 200/100kSPS ADC which operates on a single +3V to +5.5V supply and dissipates only 4.5mW (+3V supply, AD7853L). After each conversion, the device automatically powers down to 25µW. The AD7853/AD7853L is based on a successive approximation architecture and uses a charge redistribution (switched capacitor) DAC. A calibration feature removes gain and offset errors. A block diagram of the device is shown in Figure 8.18.

The AD7853 operates on a 4MHz maximum external clock frequency. The AD7853L operates on a 1.8MHz maximum external clock frequency. The timing diagram for AD7853L is shown in Figure 8.19. The AD7853/AD7853L has modes which configure the $\overline{\text{SYNC}}$ and SCLK as inputs or outputs. In the example shown here they are generated by the AD7853L. The AD7853L serial clock operates at a maximum frequency of 1.8MHz (556ns period). The data bits are valid 330ns after the positive-going edges of SCLK. This allows a setup time of approximately 330ns minimum before the negative-going edges of SCLK, easily meeting the ADSP-2189M 4ns t_{SCS} requirement. The hold-time after the negative-going edge of SCLK is approximately 226ns, again easily meeting the ADSP-2189M 7ns t_{SCH} timing requirement. These simple calculations show that the data and RFS setup and hold requirements of the ADSP-2189M are met with considerable margin.

AD7853/AD7853L +3V SINGLE-SUPPLY 12-BIT 200/100kSPS SERIAL OUTPUT ADC

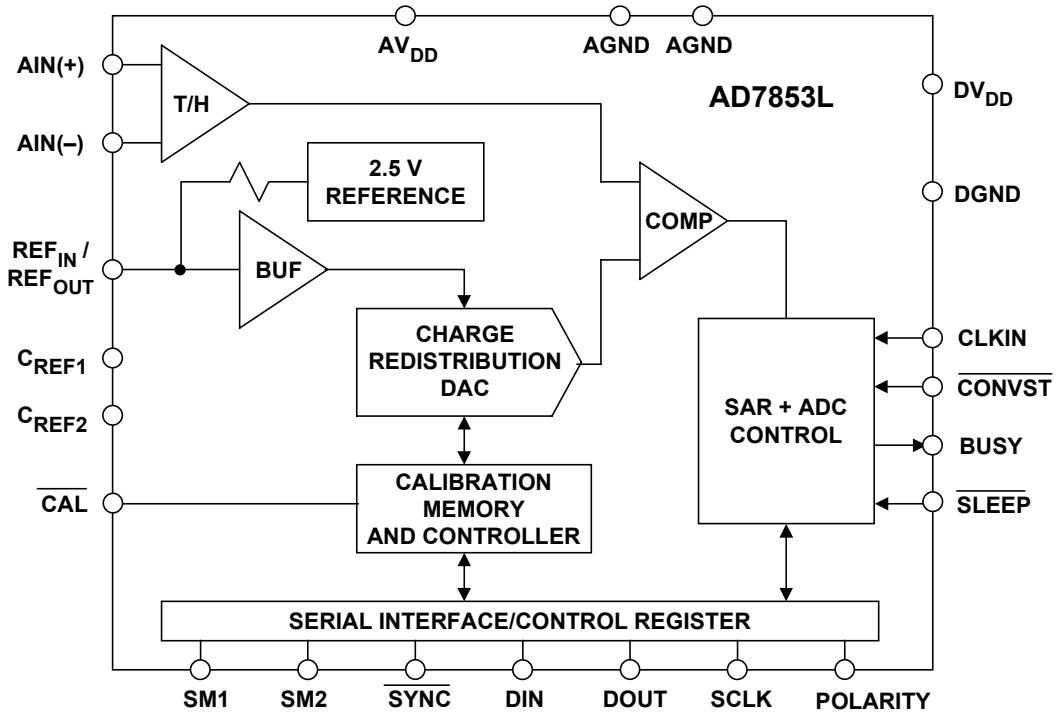


Figure 8.18

AD7853L SERIAL ADC OUTPUT TIMING +3V SUPPLY, SCLK = 1.8MHz

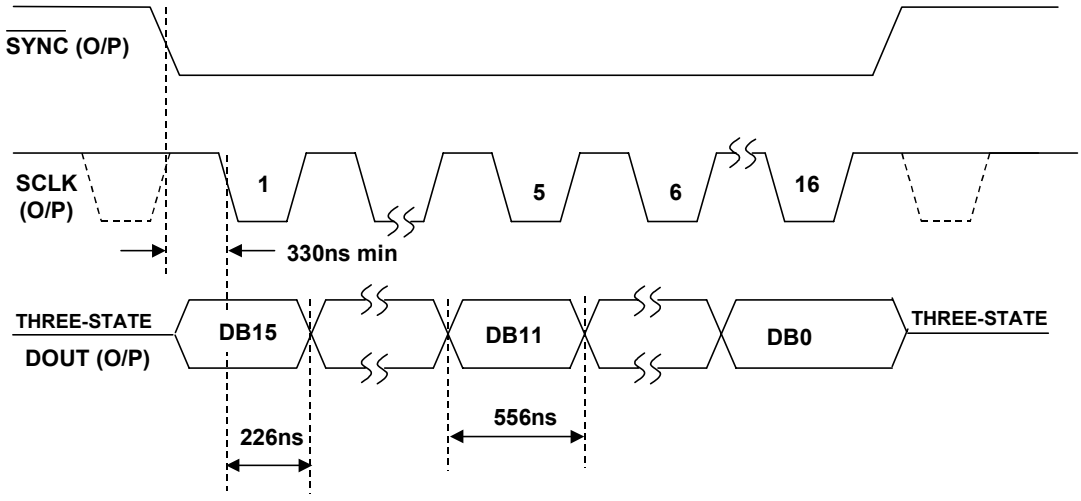


Figure 8.19

Figure 8.20 shows the AD7853L interfaced to the ADSP-2189M connected in a mode to transmit data from the ADC to the DSP (alternate/master mode). The AD7853/AD7853L contains internal registers which can be accessed by writing from the DSP to the ADC via the serial port. These registers are used to set various modes in the AD7853/AD7853L as well as to initiate the calibration routines. These connections are not shown in the diagram.

AD7853/AD7853L SERIAL ADC INTERFACE TO ADSP-2189M

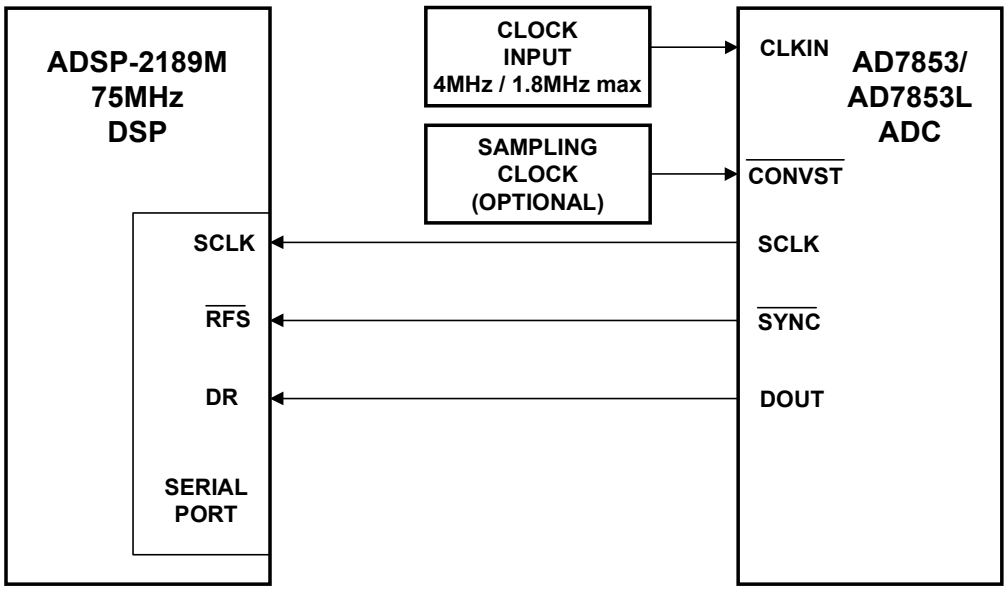


Figure 8.20

SERIAL DAC TO DSP INTERFACE

Interfacing serial input DACs to the serial ports of DSPs such as the ADSP-21xx family is also relatively straightforward and similar to the previous discussion regarding serial output ADCs. The details will not be repeated here, but a simple interface example will be shown.

The AD5322 is a 12-bit, 100kSPS dual DAC with a serial input interface. It operates on a single +2.5V to +5.5V supply, and a block diagram is shown in Figure 8.21. Power dissipation on a +3V supply is 690µW. A power-down feature reduces this to 0.15µW. Total harmonic distortion is greater than 70dB below full scale for a 10kHz output. The references for the two DACs are derived from two reference pins (one per DAC). The reference inputs may be configured as buffered or unbuffered inputs. The outputs of both DACs may be updated simultaneously using the asynchronous $\overline{\text{LDAC}}$ input. The device contains a power-on reset circuit that ensures that the DAC outputs power up to 0V and remain there until a valid write takes place to the device.

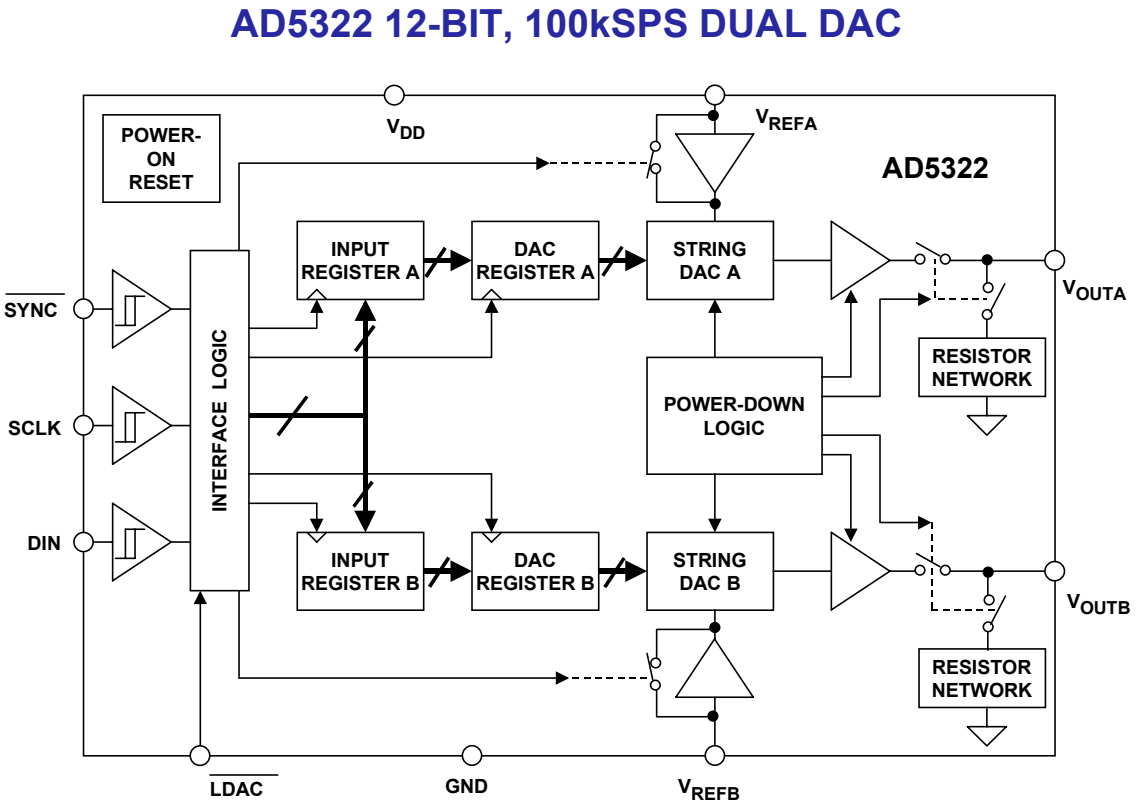


Figure 8.21

Data is normally input to the AD5322 via the SCLK, DIN, and $\overline{\text{SYNC}}$ pins from the serial port of the DSP. When the $\overline{\text{SYNC}}$ signal goes low, the input shift register is enabled. Data is transferred into the AD5322 on the falling edges of the following 16 clocks. A typical interface between the ADSP-2189M and the AD5322 is shown in Figure 8.22. Notice that the clocks to the AD5322 are generated from the ADSP-2189M clock. It is also possible to generate the SCLK and $\overline{\text{SYNC}}$ signals externally to the AD5322 and use them to drive the ADSP-2189M. The serial interface of the AD5322 is not fast enough to handle the ADSP-2189M maximum master clock frequency. However, the serial interface clocks are programmable and can be set to generate the proper timing for fast or slow DACs.

The input shift register in the AD5322 is 16-bits wide. The 16-bit word consists of four control bits followed by 12 bits of DAC data. The first bit loaded determines whether the data is for DAC A or DAC B. The second bit determines if the reference input will be buffered or unbuffered. The next two bits control the operating modes of the DAC (normal, power-down with 1kΩ to ground, power-down with 100kΩ to ground, or power-down with a high impedance output).

AD5322 DAC SERIAL INTERFACE TO ADSP-2189M

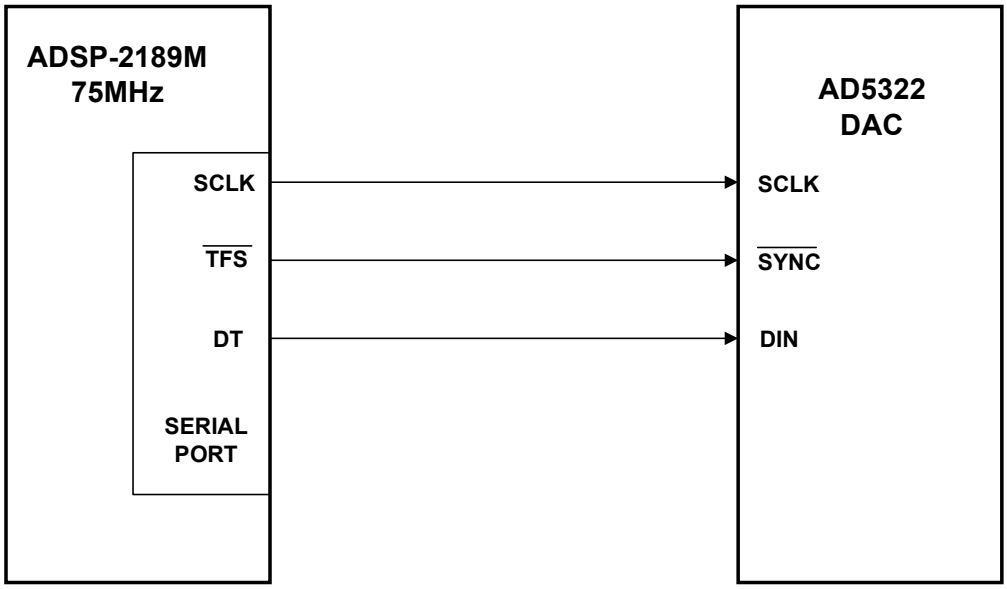


Figure 8.22

INTERFACING I/O PORTS, ANALOG FRONT ENDS, AND CODECS TO DSPS

Since most DSP applications require both an ADC and a DAC, I/O Ports and CODECs have been developed which integrate the two functions on a single chip as well as provide easy-to-use interfaces to standard DSPs. These devices also go by the name of *analog front ends*.

A functional block diagram of the AD73322 is shown in Figure 8.23. This device is a dual analog front end (AFE) with two 16-bit ADCs and two 16-bit DACs capable of sampling at 64KSPS. It is designed for general purpose applications including speech and telephony using sigma-delta ADCs and sigma-delta DACs. Each channel provides 77dB signal-to-noise ratio over a voiceband signal bandwidth.

The ADC and DAC channels feature programmable input/output gains with ranges of 38dB and 21dB, respectively. An on-chip voltage reference is included to allow single supply operation on +2.7V to +5.5V. Power dissipation is 73mW with a +3V supply.

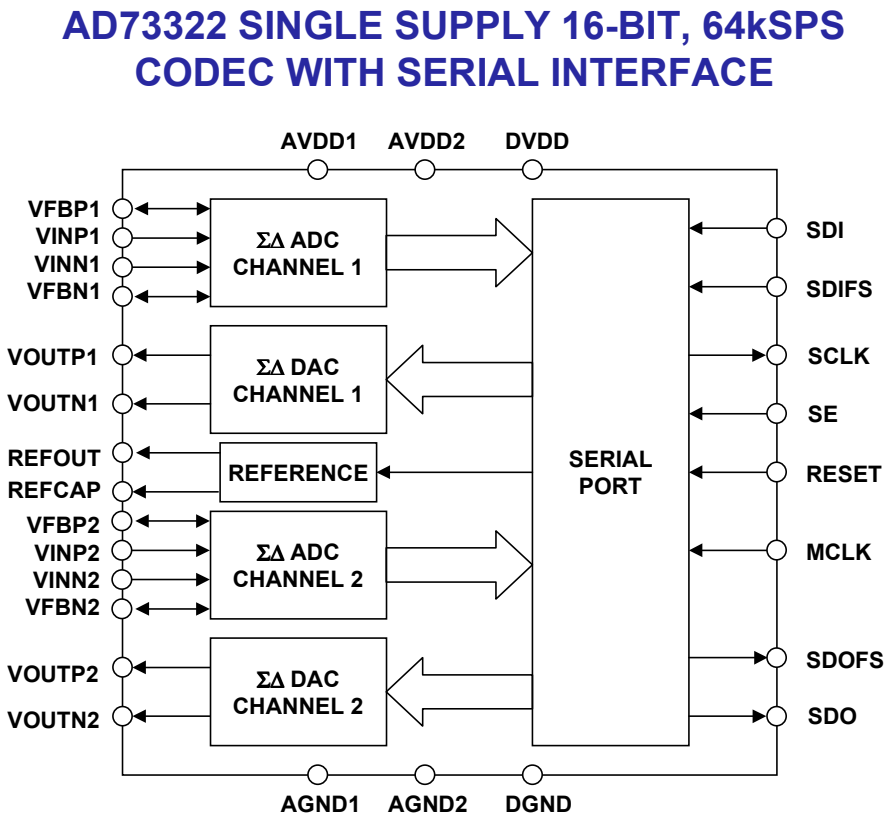


Figure 8.23

The sampling rate of the CODECs is programmable with four separate settings of 64kHz, 32kHz, 16kHz, and 8kHz when operating from a master clock of 16.384MHz. The serial port allows easy interfacing of single or cascaded devices to industry standard DSP engines, such as the ADSP-21xx family. The SPORT transfer rate is programmable to allow interfacing to both fast and slow DSP engines. The interface to the ADSP-218x family is shown in Figure 8.24. The SE pin (SPORT enable) may be controlled from a parallel output pin or a flag pin such as FL1, or where SPORT power down is not required, it can be permanently strapped high using a suitable pull-up resistor. The RESET pin may be connected to the system hardware reset, or it may be controlled with another flag bit.

In the *program* mode, data is transferred from the DSP to the AD73322 control registers to set up the device for desired operation. Once the device has been configured by programming the correct settings to the various control registers, the device may exit the *program* mode and enter the *data* mode. The dual ADC data is transmitted to the DSP in two blocks of 16-bit words. Similarly, the dual DAC data is transmitted from the DSP to the AD73322 in two blocks of 16-bit words. Simplified interface timing is also shown in Figure 8.24.

AD73322 INTERFACE TO ADSP-218x SERIES (DATA TRANSFER MODE)

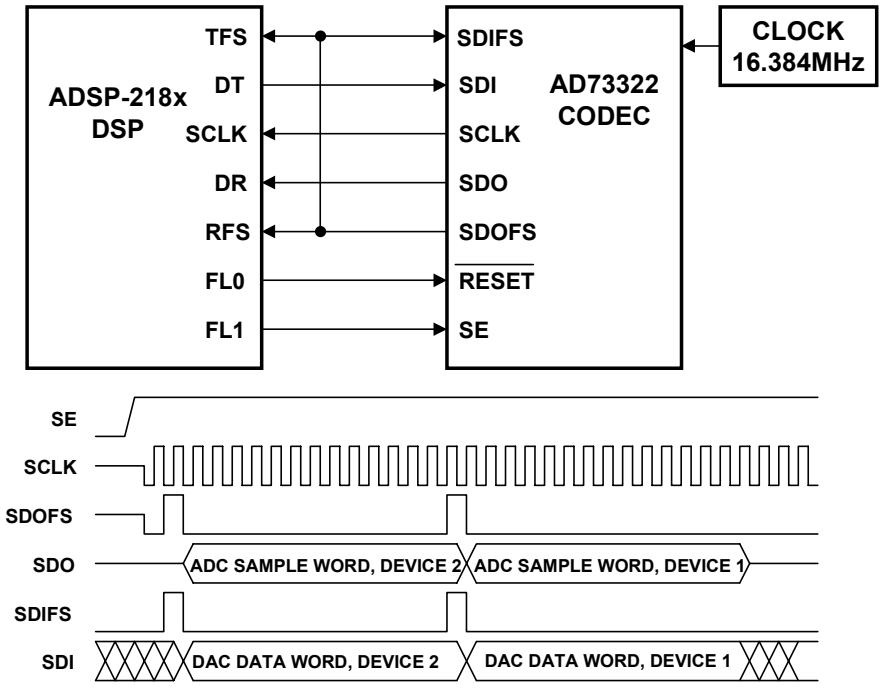


Figure 8.24

The AD73422 is the first product in the dspConverter™ family of products which integrate a dual analog front end (AD73322) and a DSP (52MIPS ADSP-2185L/86L). The entire functionality of the dual-channel CODEC and the DSP fits into a small, 119-ball 14mm by 22mm plastic ball grid array (PBGA) package. The obvious advantage is the saving of circuit board real estate. ADC and DAC signal-to-noise ratios are approximately 77dB over voiceband frequencies.

The AD74222-80 integrates 80K bytes of on-chip memory configured as 16K words (24-bit) of program RAM, and 16K words (16-bit) of data RAM. The AD73422-40 integrates 40K bytes of on-chip memory configured as 8K words (24-bit) of program RAM, and 8K words (16-bit) of data RAM. Power-down circuitry is also provided to meet the low power needs of battery operated portable equipment. The AD73422 operates on a +3V supply and dissipates approximately 120mW with all functions operational.

AD73422 dspConverter™

- Complete Dual CODEC (AD73322) and DSP (ADSP-2185L/86L)
- 14mm by 22mm BGA package
- +3V Single-Supply Operation, 73mW Power Dissipation
- Power-Down Mode
- CODEC:
  - Dual 16-bit Sigma-Delta ADCs and DACs
  - Data Rates: 8, 16, 32, and 64KSPS
  - 77dB SNR
- DSP:
  - 52MIPS
  - ADSP-218x Code Compatible
  - 80K Byte and 40K Byte On-Chip Memory Options

Figure 8.25

HIGH-SPEED INTERFACING

With the advent of ever faster DSP clock rates and newer architectures it has become possible to acquire and process high speed signals. The programmability of DSPs makes it possible to run different algorithms on the same hardware, while providing different system functionality. Figure 8.26 shows a simplified ADSP-21065L system connected to a high-speed ADC and high-speed DAC. The ADC and DAC both have parallel interfaces connected to the External Port of the DSP. With the SHARC family of DSPs there are several ways of connecting the converters to this port. The access to the converters can be done using the *direct memory access* (DMA) controller of the DSP or it can be done under program control using the core of the DSP. Using the DMA places no load on the DSP core so it can continue processing (executing program instructions) while the data is transferred to from the on-chip memory.

The AD9201 is a dual-channel, 10-bit, 20MSPS ADC which operates on a single +2.7V to +5.5V supply and dissipates only 215mW (+3V supply). The AD9201 offers closely matched ADCs needed for many applications such as I/Q communications. Input buffers, an internal voltage reference and multiplexed digital outputs buffers make interfacing to the AD9201 very simple.

The companion part to the AD9201 ADC is the AD9761 DAC. The AD9761 is a dual 10-bit, 20MSPS per channel DAC operating on a single +2.7V to +5.5V supply and dissipating only 200mW (+3V supply). A voltage reference, digital latches and 2x interpolation make the AD9761 useful for I/Q transmitter applications.

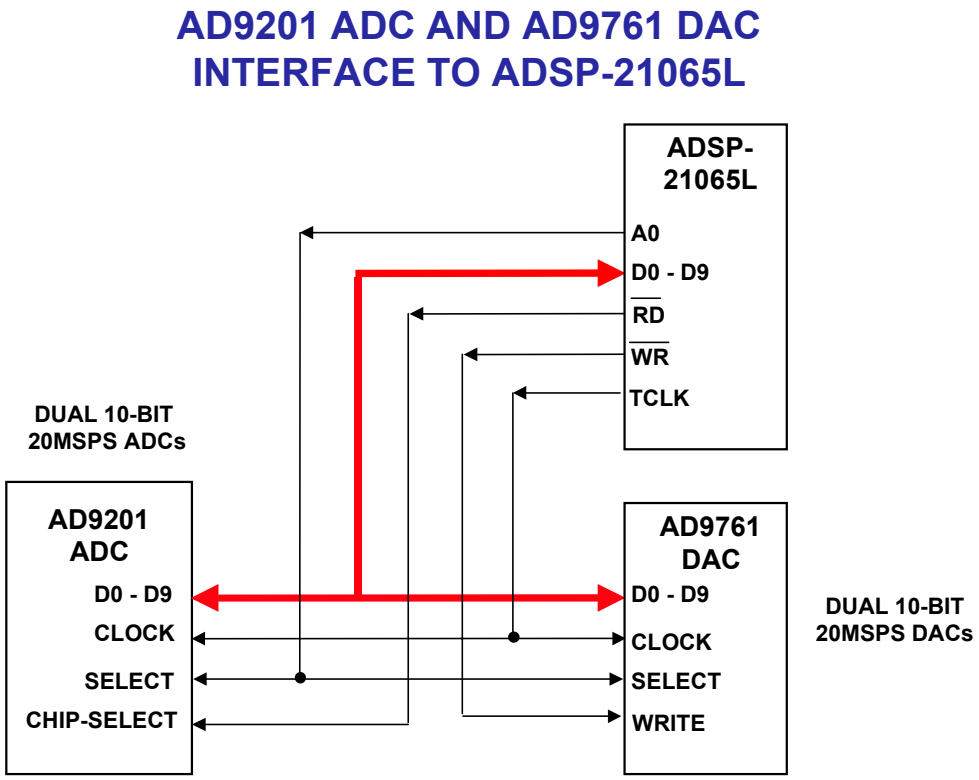


Figure 8.26

DSP SYSTEM INTERFACE

Figure 8.26 shows a simplified ADSP-2189M system using the *full memory mode* configuration with two serial devices, a byte-wide EPROM, and optional external program and data overlay memories. Programmable wait state generation allows the fast processor to connect easily to slower peripheral devices. The ADSP-2189M also provides four external interrupts, seven general-purpose input/output pins and two serial ports. One of the serial ports can be configured as two additional interrupts, a general-purpose input and a general-purpose output pin for a total of six external interrupts, 9 IOs and one serial port. The ADSP-2189M can also be operated in the *host memory mode* which allows access to the full external data bus, but limits addressing to a single address bit. Additional system peripherals can be added in the *host memory mode* through the use of external hardware to generate and latch address signals.

ADSP-2189M SYSTEM INTERFACE: FULL MEMORY MODE

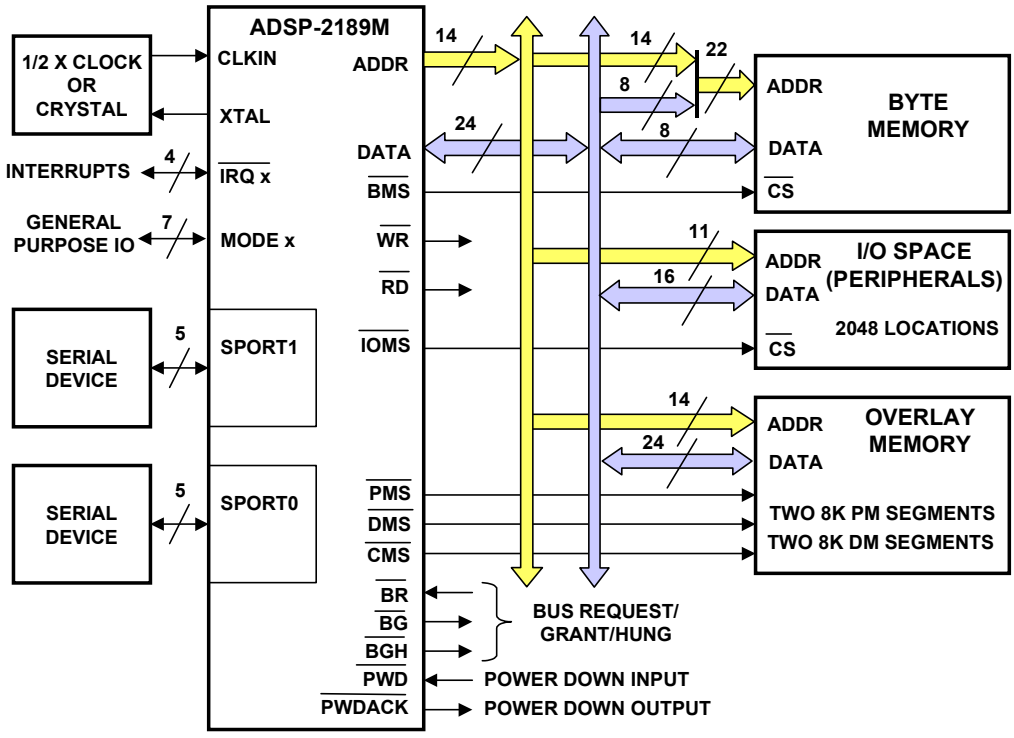


Figure 8.27

REFERENCES

1. Steven W. Smith, **The Scientist and Engineer's Guide to Digital Signal Processing**, Second Edition, 1999, California Technical Publishing, P.O. Box 502407, San Diego, CA 92150. Also available for free download at: http://www.dspguide.com or http://www.analog.com

2. C. Britton Rorabaugh, **DSP Primer**, McGraw-Hill, 1999.

3. Richard J. Higgins, **Digital Signal Processing in VLSI,** Prentice-Hall, 1990.

4. **DSP Designer's Reference (DSP Solutions)** CDROM, Analog Devices, 1999.

5. **DSP Navigators: Interactive Tutorials about Analog Devices' DSP Architectures** (Available for ADSP-218x family and SHARC family): http://www.analog.com/industry/dsp/training/index.html#Navigator

6. **General DSP Training and Workshops**: http://www.analog.com/industry/dsp/training

   The following DSP Reference Manuals and documentation are available for free download from: http://www.analog.com/industry/dsp/tech_docs.html

7. **ADSP-2100 Family Users Manual, 3rd Edition**, Sept., 1995.

8. **ADSP-2100 Family EZ Tools Manual.**

9. **ADSP-2100 EZ-KIT Lite Reference Manual.**

10. **Using the ADSP-2100 Family, Vol. 1, Vol. 2.**

11. **ADSP-2106x SHARC User's Manual, 2nd Edition, July, 1996.**

12. **ADSP-2106x SHARC EZ-KIT Lite Manual.**

13. **ADSP-21065L SHARC User's Manual, Sept. 1, 1998.**

14. **ADSP-21065L SHARC EZ-LAB User's Manual.**

15. **ADSP-21160 SHARC DSP Hardware Reference.**