

MOTOROLA SEMICONDUCTOR TECHNICAL DATA

DSP56200

Advance Information

Cascadable-Adaptive Finite-Impulse-Response (CAFIR) Digital-Filter Chip

The DSP56200 is a 28-pin HCMOS, algorithm-specific, digital-signal processor (DSP) designed to perform sum-of-products tasks. Two principal algorithms are implemented on the DSP56200 — finite sum of products and adaptive least-mean-squares (LMS). These algorithms make finite impulse response (FIR) and adaptive FIR digital filtering the primary functions of the part. A serial chip-cascading interface enables the user to easily build filters with extended tap lengths and/or increased throughput. Its performance, features, and simple interface make the DSP56200 a natural solution for applications such as echo canceling, noise canceling, convolution, correlation, orthogonal transforms such as sine or cosine transforms, and many other DSP applications.

Key features of the DSP56200 include the following:

- Three Modes of Operation
 - Single FIR Filter
 - Dual FIR Filter (Two Independent FIR Filters)
 - Single Adaptive FIR Filter with dc Tap and Leakage Control
- High-Performance Hardware
 - 24×16 -Bit Multiplication with 40-Bit Accumulation
 - 10.25 MHz Internal Operation
 - Single-Cycle Multiply-Accumulate
 - Low-Power Standby Mode
 - Static 256×24 -Bit Coefficient Random-Access Memory (RAM)
 - Static 256×16 -Bit Data RAM
 - 28-Pin Dual-In-Line Package
- Architecture Optimized for Digital Filtering
 - 24-Bit Coefficients
 - 16-Bit Data
 - Three Execution Units Operate in Parallel
 - Multiple Internal Buses
 - Programmable Filter Length (4 to 256 Taps)
 - Single-Cycle Coefficient Update using LMS Algorithm
 - 16-Bit Rounding Option on Output of Both FIR Filter Modes
- Adaptive Digital Filtering Control
 - Adaptation Disable Capability
 - Programmable Loop Gain in Adaptive Mode
 - Programmable Coefficient Leakage Term
 - DC Tap Option
- Virtually Unlimited Cascadability
 - Larger Number of Filter Taps
 - Higher Sampling Rates
- High Throughput Rates
 - 227-kHz Fixed FIR Filter (32 Taps, 1 DSP56200)
 - 37-kHz Fixed FIR Filter (256 Taps, 1 DSP56200)
 - 115-kHz Adaptive FIR Filter (256 Taps, 8 Cascaded DSP56200s)
 - 19-kHz Adaptive FIR Filter (256 Taps, 1 DSP56200)
 - Many Other Configurations Possible
- Simple Interface to Popular Hosts
 - Microprocessors and Microcomputers
 - General-Purpose DSPs
 - Unused RAM Available for Host Storage

This document contains information on a new product. Specifications and information herein are subject to change without notice.



GENERAL DESCRIPTION

The fundamental operation in digital signal processing is performing a finite sum of products. In many applications, it is difficult to economically justify using a general-purpose user-programmable DSP to perform the finite-sum-of-products function because it is either underutilized in terms of addressing, decision-making, and input/output (I/O), or it simply cannot perform the function fast enough. The DSP56200 is highly pipelined for optimum performance; it can easily be interfaced to a host processor, and it can be cascaded without the need for "glue" logic.

ALGORITHMS

The finite-sum-of-products algorithm is expressed in Equation (1) as

$$y = \sum_{i=0}^{N-1} a(i) b(i) \quad (1)$$

The function performed by Equation (1) depends on $a(i)$ and $b(i)$. If the coefficients, $a(i)$, represent the impulse response, $h(i)$, of a desired filter, and $b(i)$ is one of the N most recent samples, $x(n-i)$, of a real-time signal, $x(nT)$, sampled at T -second intervals (the T will be implied hereafter and n will refer to the time index), then Equation (1) becomes the familiar convolution sum

$$y(n) = \sum_{i=0}^{N-1} h(i) x(n-i) \quad (2)$$

that describes the FIR filter process.¹ Similarly, if $a(i)$ is a sample of a reference signal and $b(i)$ remains $x(n-i)$, then Equation (1) describes a finite length correlation/matched filter process. By setting $a(i)$ equal to $x(i)$ and setting $b(i) = \cos(\omega_i/N)$ or $b(i) = \sin(\omega_i/N)$ where $\omega = 2\pi f_s$ and $f_s = 1/T$, the cosine or sine transform, respectfully, of the real signal, $x(n)$, can be calculated. Thus, the finite-sum-of-products operation is the foundation of signal processing.

The LMS algorithm is used in adaptive FIR filter applications.^{2,3,4} The frequency response, i.e., the $h(i)$ of an adaptive filter, changes as the filter adjusts itself to extract a desired signal contained in the input signal to the adaptive filter. The output of the filter will be an estimate of the desired signal. The adaptive process occurs in two steps. First, an error signal, $e(n)$, is calculated by subtracting the estimate from the desired signal, $d(n)$, as follows:

$$e(n) = d(n) - \sum_{i=0}^{N-1} h(i) x(n-i) \quad (3)$$

The second step involves updating each coefficient in response to the error signal according to the LMS algorithm expressed as

$$h(n+1,i) = h(n,i) + K e(n) \times (n - i) \pm \text{Leakage} \quad (4)$$

where $h(n+1,i)$ is the value of the i^{th} coefficient to be used in the next (i.e., $n+1$) filter operation, and K is a gain factor influencing the rate of convergence of the filter and the residual error.⁵ The leakage term is advantageous when the input does not satisfy the mixing condition over periods of time $>> NT$.^{5,6} Signals that do not exercise all of the coefficients (i.e., signals having fewer degrees of freedom than the order of the filter, N , such as narrowband signals (tones)) do not satisfy the mixing condition. As a result, some of the coefficients will slowly diverge to the maximum-possible value. The leakage correction term prevents this divergence by nudging the coefficient towards zero.⁷ However, the leakage term must be small enough, to have little effect on the convergence when the input does satisfy the mixing conditions: i.e., when the frequency content of the input returns to a wideband state.

MODES

Since the most common application of the DSP56200 is FIR filtering, its three possible modes of operation are referred to as single FIR (SFIR), dual FIR (DFIR), and adaptive FIR (AFIR) filter modes.

The functional block diagram of the DSP56200 in the SFIR filter mode is shown in Figure 1. The input consists of a delay line or shift register of length N , where N can be from 4 to 256; the output is the sum of the products of each element in the shift register, $x(n-i)$, multiplied by its corresponding coefficient, $h(i)$: i.e., the convolution sum. For each new input sample, N multiplications and $N-1$ additions are performed. The shift register memory locations are often referred to as taps since they represent the points at which the delay line can be accessed or "tapped." The last tap data can be accessed in the DSP56200. In the SFIR filter mode, the DSP56200 can be cascaded using its serial cascade interface described in **SERIAL CASCADE INTERFACE**.

In the DFIR filter mode, the DSP56200 implements two independent FIR filters. Both filters must have the same number of taps up to a maximum of 128 each. A single START pulse triggers the filtering; the second filter is activated automatically after the first filter terminates. The DSP56200 cannot be cascaded in this mode.

The functional block diagram of the DSP56200 in a standalone AFIR mode is shown in Figure 2. In this mode, the DSP56200 cycles through its memory twice each sample period: first, to perform the convolution sum as in the SFIR filter mode, and second, to update each of the coefficients per the LMS algorithm. The desired signal, $d(n)$, is subtracted from the filter output to generate the negative of the error signal, $-e(n)$. The error signal is externally fed back to the adaptive filter using the serial cascade interface. The DSP56200 can be cascaded in this mode.

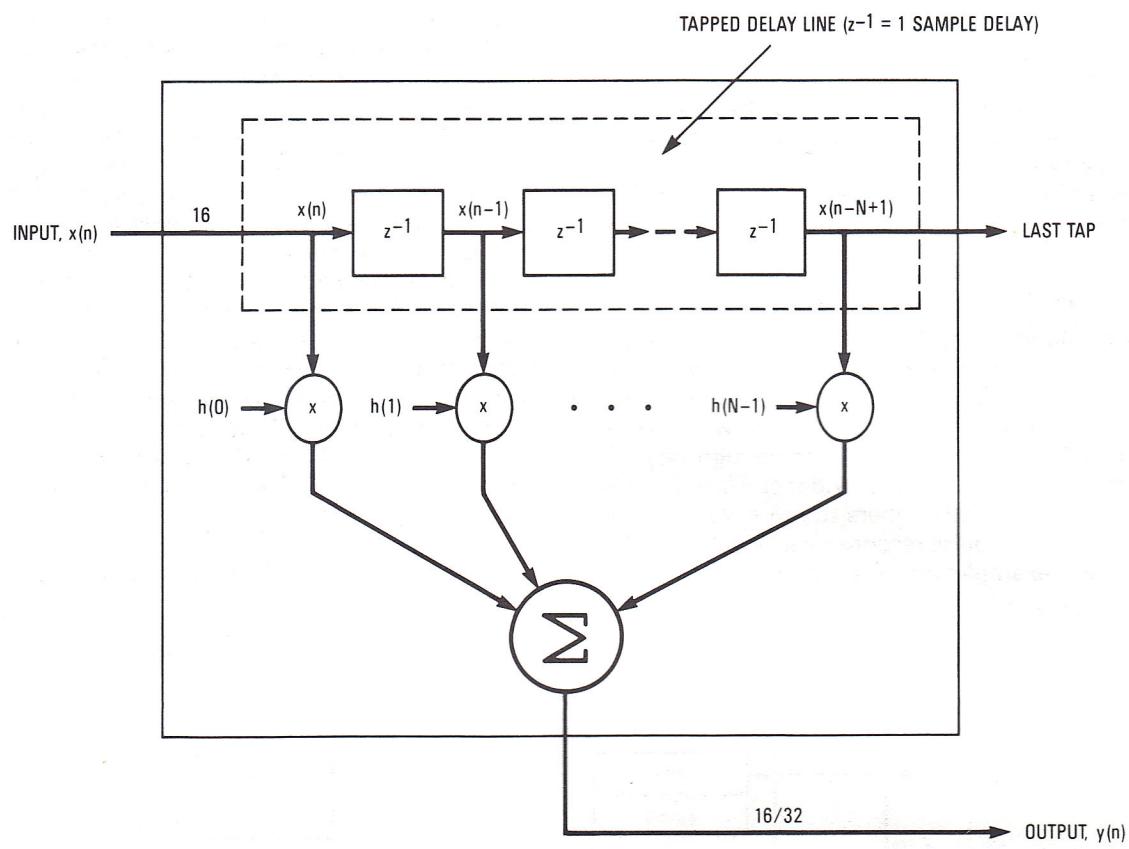


Figure 1. Functional Block Diagram

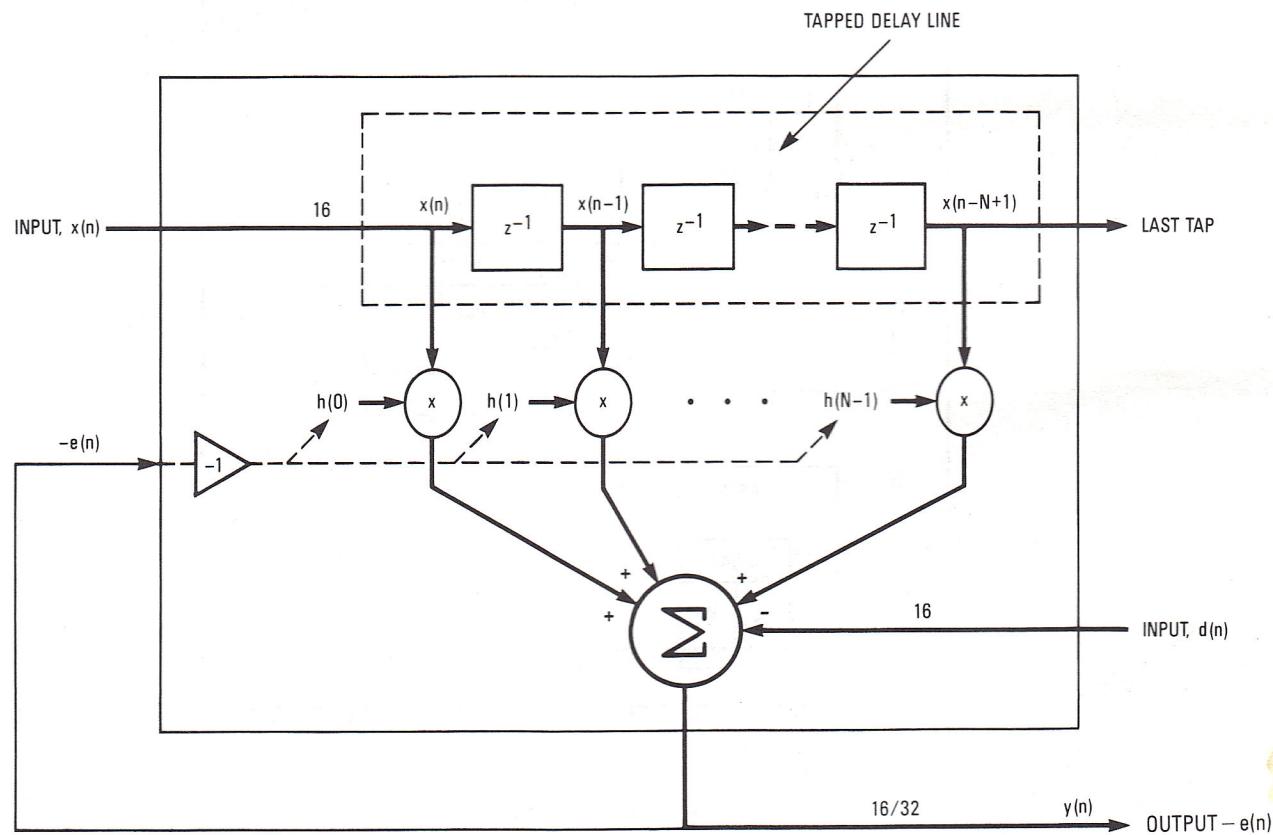


Figure 2. Functional Block Diagram — (AFIR Mode)

ARCHITECTURE

The on-chip resources and signal groups of the DSP56200, shown in Figure 3, consist of two RAMs (a 256×24 -bit coefficient RAM and a 256×16 -bit data RAM), an address generator, an arithmetic unit, an asynchronous parallel interface, a serial cascade interface, and timing and control.

ARITHMETIC UNIT

The key to the accuracy of the DSP56200 is its arithmetic unit. Accuracy is affected by the number of bits in the coefficient and by errors due to rounding. In FIR filter applications, the actual frequency response will deviate from the user's desired response if not enough bits are used to represent the coefficients. Wider coefficient word widths also enable adaptive filters to more closely converge to the desired impulse response, resulting in smaller error terms. For example, on the order of 50 dB of echo

return loss enhancement can be achieved with the DSP56200 in data communication applications. Roundoff errors decrease as the size of the accumulator and coefficients increases. The DSP56200, which uses a 24-bit coefficient, will accept data words having up to 16 bits. The results of the multiply-accumulate (MAC) operation are stored in a 40-bit accumulator. Both the 16-bit data samples and the 24-bit coefficients are represented as signed fractional numbers (see Figure 4).

The arithmetic unit is illustrated in Figure 5. The 24-bit coefficient operand, $h(i)$, is multiplied by a 16-bit data operand, $x(n-i)$, in a fractional 24×16 -bit parallel multiplier. The 40-bit product is accumulated in a 40-bit accumulator. The sum can be transferred to the I/O interfaces as either a truncated 32-bit fraction or as a rounded 16-bit fraction. Nonconvergent rounding, i.e., biased rounding, is performed on the sum by simply adding the rounding constant R_{16} (\$0000080).

To implement the LMS algorithm of Equation (4), the error is first multiplied by the gain, and the convergently

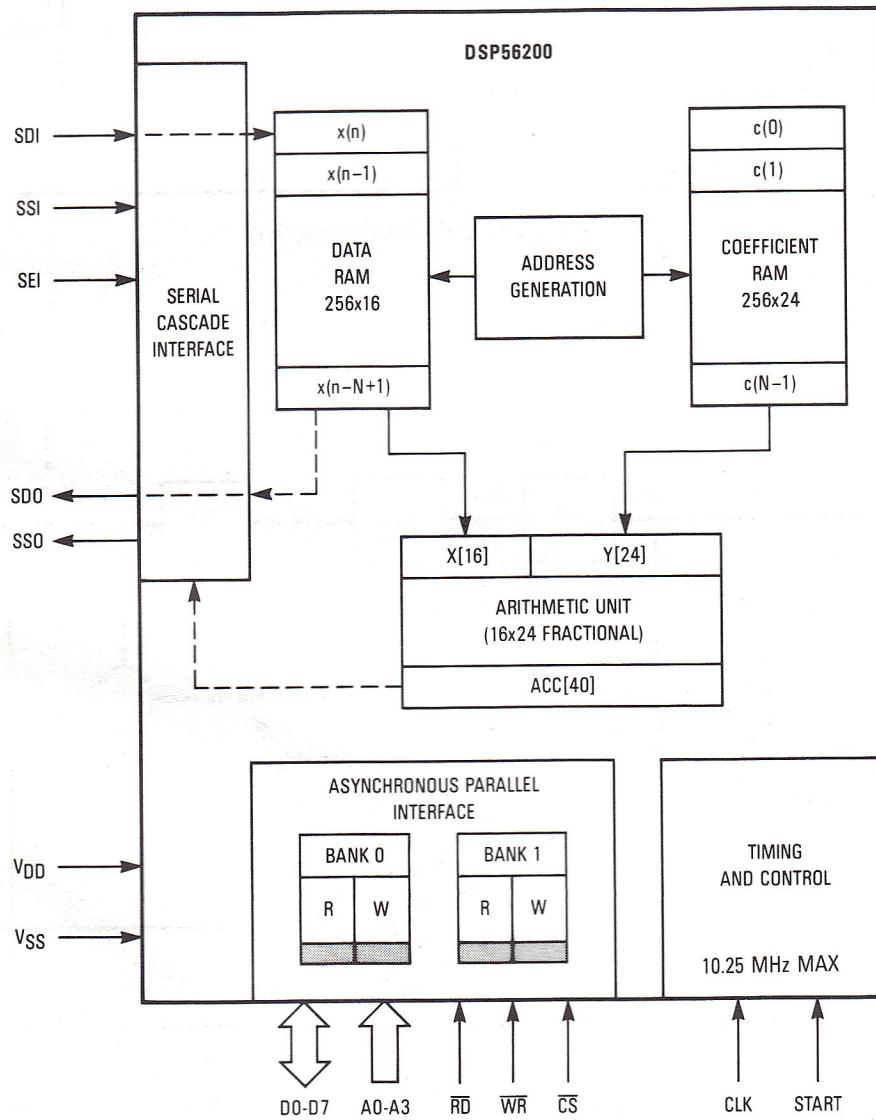


Figure 3. Signal Groups

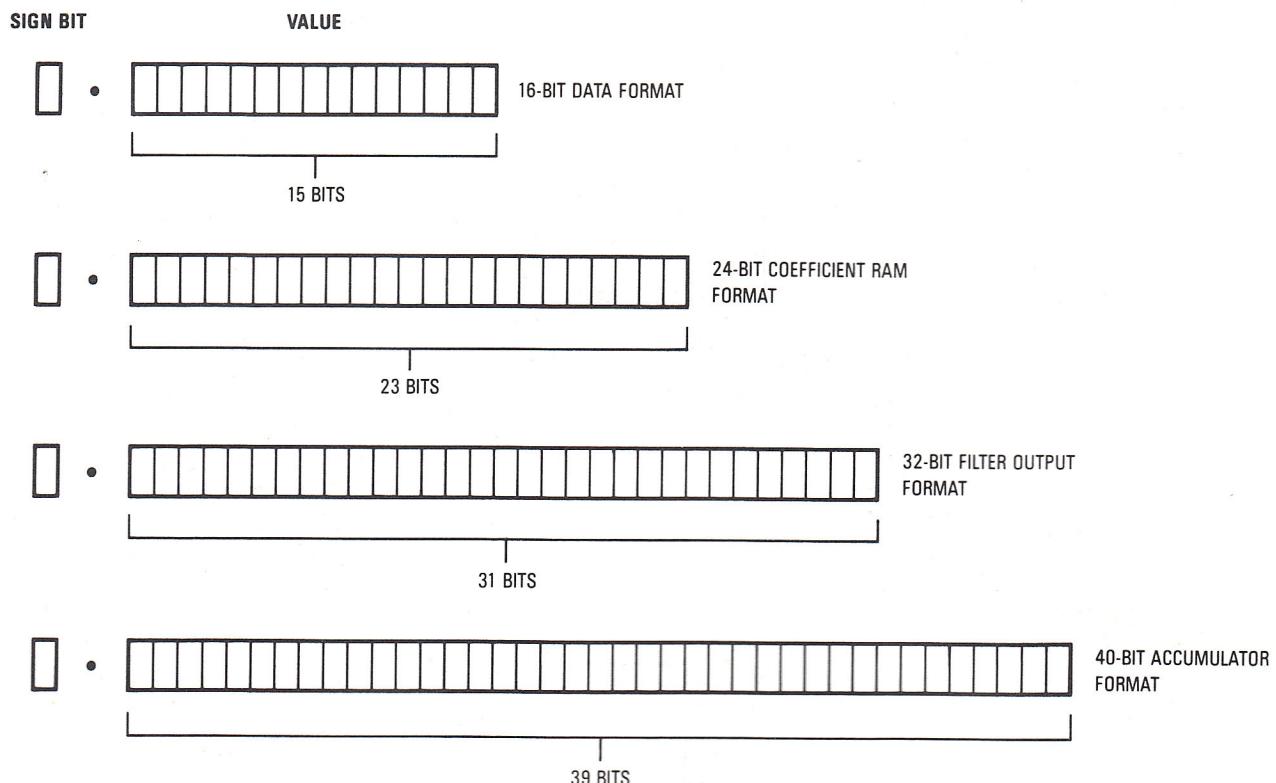


Figure 4. Signed Fractional Twos Complement Data Formats

rounded (unbiased rounding) product, $K_e(n)$, is loaded into the 24-bit input register to the MAC. The coefficient, $h(i)_{old}$, to be updated is latched, and its corresponding data word, $x(n - i)$, is loaded into the 16-bit input register to the MAC. The leakage constant is also latched if the leakage feature is enabled. The leakage constant is formed by loading the user-defined 8-bit leakage value into bits having weightings 2^{-16} through 2^{-23} . The same leakage term is applied to each coefficient. Because the leakage term only affects the least significant byte of the coefficient, it does not prevent the filter from adapting when the input satisfies the mixing condition. The rounding constant for convergent rounding, R_{24} , is concatenated with the leakage constant. After the MAC operation, the accumulator contains the 40-bit updated coefficient. The updated coefficient is stored in the coefficient RAM after it has been convergently rounded to 24 bits, provided an overflow has not been detected. If the accumulator has overflowed, the updated coefficient is not written back to the coefficient memory; the previous coefficient value is retained. Equation (4) expresses a read-modify-write operation on the $h(i)$, where the modify operation is a MAC.

When the dc tap option is activated, \$7FFF is substituted for the last tap data in the convolution sum and LMS coefficient update calculations. The last tap data in the RAM is not overwritten, however, and can be accessed using the RAM address register or the last tap register via the asynchronous parallel interface. The dc tap option can also be used to add a dc offset to the output in the FIR modes.

Scaling can be accomplished by appropriately positioning the coefficient operand, correctly sign extended,

in the coefficient memory if the full 24-bit coefficient is not needed. This option can be used when implementing a correlation function, for example.

ASYNCHRONOUS PARALLEL INTERFACE

The asynchronous parallel interface consists of eight data lines (D0-D7), four address lines (A0-A3), read strobe (RD), write strobe (WR), and a chip select line (CS). The host processor uses this interface to initialize the data and coefficient memories, tap length register, and leakage constant; set up the configuration register; input the most recent data, $x(n)$, and, in the adaptive mode, the desired signal, $d(n)$; and output the sum and last tap data. To the host, this interface looks like a 64-byte memory organized into four register files. The registers and their addresses are shown in Figure 6; the configuration register is shown in Figure 7.

Each register is double buffered, and data is transferred between the interface and the internal engine during the low-to-high transition of each START pulse (see Figure 8). The START pulse is normally synchronized with the system sampling pulse (see **TIMING AND CONTROL**). Conceivably, each register could be accessed every sample period. Due to the double-buffered nature of the registers, the order in which the registers are written between STARTs is unimportant. Any RAM location (i.e., one within the finite sum having an address less than N or one outside the finite sum having an address greater than N) can be accessed asynchronously during a sample period using the RAM access register. This technique facilitates verifying data or coefficients, or using the unused RAM locations for scratchpad system storage.

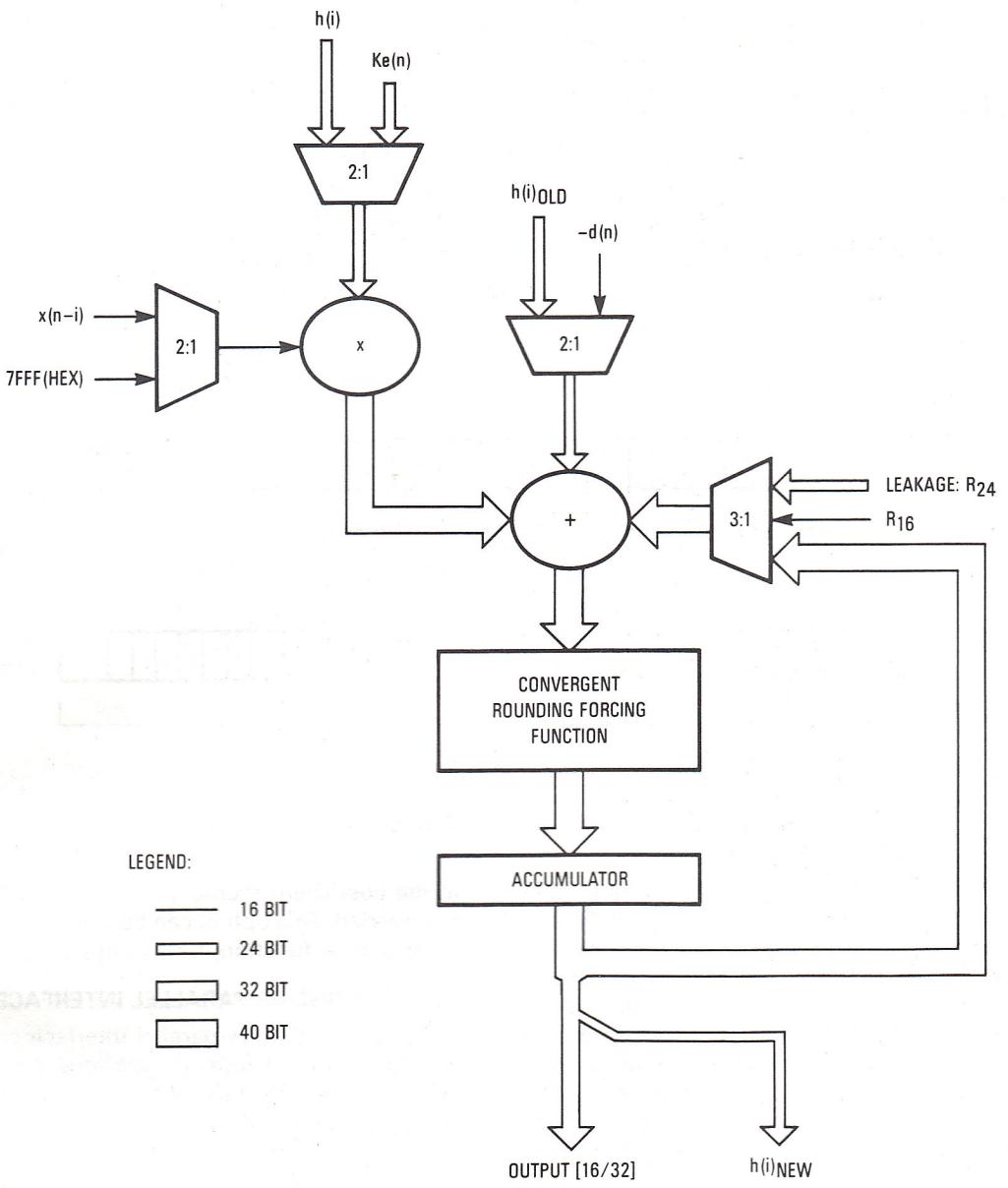


Figure 5. Arithmetic Unit

SERIAL CASCADE INTERFACE

The serial cascade interface was designed to support cascading DSP56200s without glue logic. The last tap data is shifted out the serial data out (SDO) and into serial data in (SDI) of the following DSP56200. The sample delay necessary to logically extend the delay line is built into the DSP56200. The partial-sum data is shifted out the serial sum out (SSO) and into serial sum in (SSI) of the following DSP56200. A true cascade is implemented in that the partial-sum input is added serially to the local partial sum to form the true (cumulative) partial sum, which is output to the next part in the cascade. Therefore, the SSO data from the last part in the cascade is the sum of all partial sums in the cascade. In the AFIR filter mode, this pin is connected to the serial error input (SEI) pins of all parts in the cascade, including the last part.

TIMING AND CONTROL

In most applications, the START pin will be cycled at the system data sampling frequency, and data will be written and read by the host once per START cycle. Some form of interrupt should be provided to the host processor to indicate when a START transition has occurred for the host to provide filter data I/O service. The simplest approach is to connect the START pin signal to the interrupt input of the host processor. Using this approach guarantees that the host will not service the DSP56200 until after the START low-to-high transition has occurred because interrupt service usually has nonzero latency. This method will ensure the DSP56200 is not accessed during the initial phases of the START pulse when parallel transfers between the parallel interface registers and the internal engine are occurring. The maximum clock rate

BANK 0

| HEX ADDRESS | WRITE | HEX ADDRESS | READ |
|-------------|--|-------------|--|
| 0 | X1 - HIGH | 0 | OUTPUT - 3 |
| 1 | X1 - LOW | 1 | OUTPUT - 2 |
| 2 | D - HIGH | 2 | OUTPUT - 1 |
| 3 | D - LOW | 3 | OUTPUT - 0 |
| 4 | K - HIGH | 4 | LAST TAP 1 - HIGH |
| 5 | K - LOW | 5 | LAST TAP 1 - LOW |
| 6 | X2 - HIGH | 6 | LAST TAP 2 - HIGH |
| 7 | X2 - LOW | 7 | LAST TAP 2 - LOW |
| 8 | DATA RAM ACCESS REGISTER - HIGH | 8 | DATA RAM ACCESS REGISTER - HIGH |
| 9 | DATA RAM ACCESS REGISTER - LOW | 9 | DATA RAM ACCESS REGISTER - LOW |
| A | COEFFICIENT RAM ACCESS REGISTER - HIGH | A | COEFFICIENT RAM ACCESS REGISTER - HIGH |
| B | COEFFICIENT RAM ACCESS REGISTER - MID | B | COEFFICIENT RAM ACCESS REGISTER - MID |
| C | COEFFICIENT RAM ACCESS REGISTER - LOW | C | COEFFICIENT RAM ACCESS REGISTER - LOW |
| D | RAM ADDRESS | D | * |
| E | * | E | CONFIGURATION ** |
| F | CONFIGURATION | F | CONFIGURATION ** |

BANK 1

| HEX ADDRESS | WRITE | HEX ADDRESS | READ |
|-------------|----------------|-------------|------------------|
| 0 | LEAKAGE | 0 | * |
| 1 | FIR TAP LENGTH | 1 | * |
| 2 | * | 2 | * |
| 3 | * | 3 | * |
| 4 | * | 4 | * |
| 5 | * | 5 | * |
| 6 | * | 6 | * |
| 7 | * | 7 | * |
| 8 | * | 8 | * |
| 9 | * | 9 | * |
| A | * | A | * |
| B | * | B | * |
| C | * | C | * |
| D | * | D | * |
| E | * | E | CONFIGURATION ** |
| F | CONFIGURATION | F | CONFIGURATION ** |

NOTE:

*Not available for use.

**The configuration register is readable at addresses 0E and 0F (hex).

Figure 6. Register Model

is 10.25 MHz. One tap of a FIR filter can be processed every 97.5 ns; one coefficient can be updated every 97.5 ns.

Pipeline

Three sample periods of pipe delay will occur in the DSP56200. The samples, their respective time indices,

and their status are shown in Figure 9 and described in the following paragraphs for the SFIR mode.

Input Sample $x(n)$

When input sample $x(n)$ is written by the host (or from SDI) to the appropriate input data register, it will be held there until the next START command is issued; it will

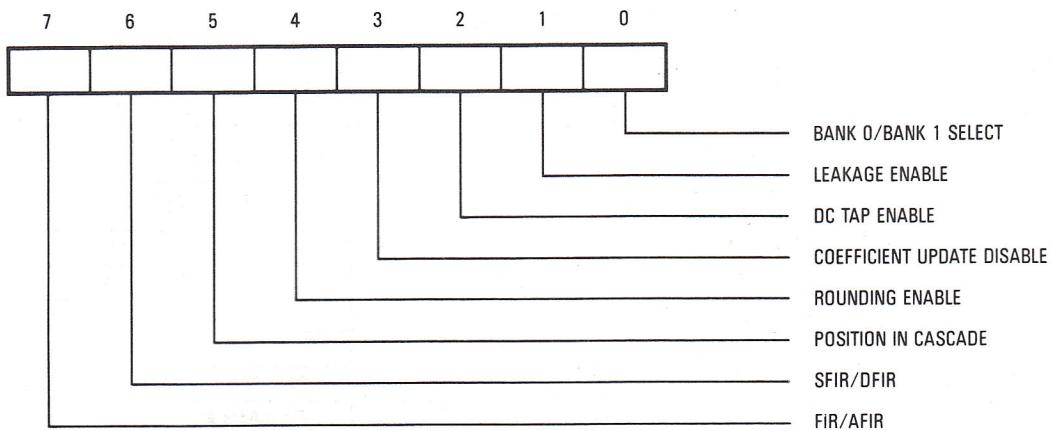


Figure 7. Configuration Register

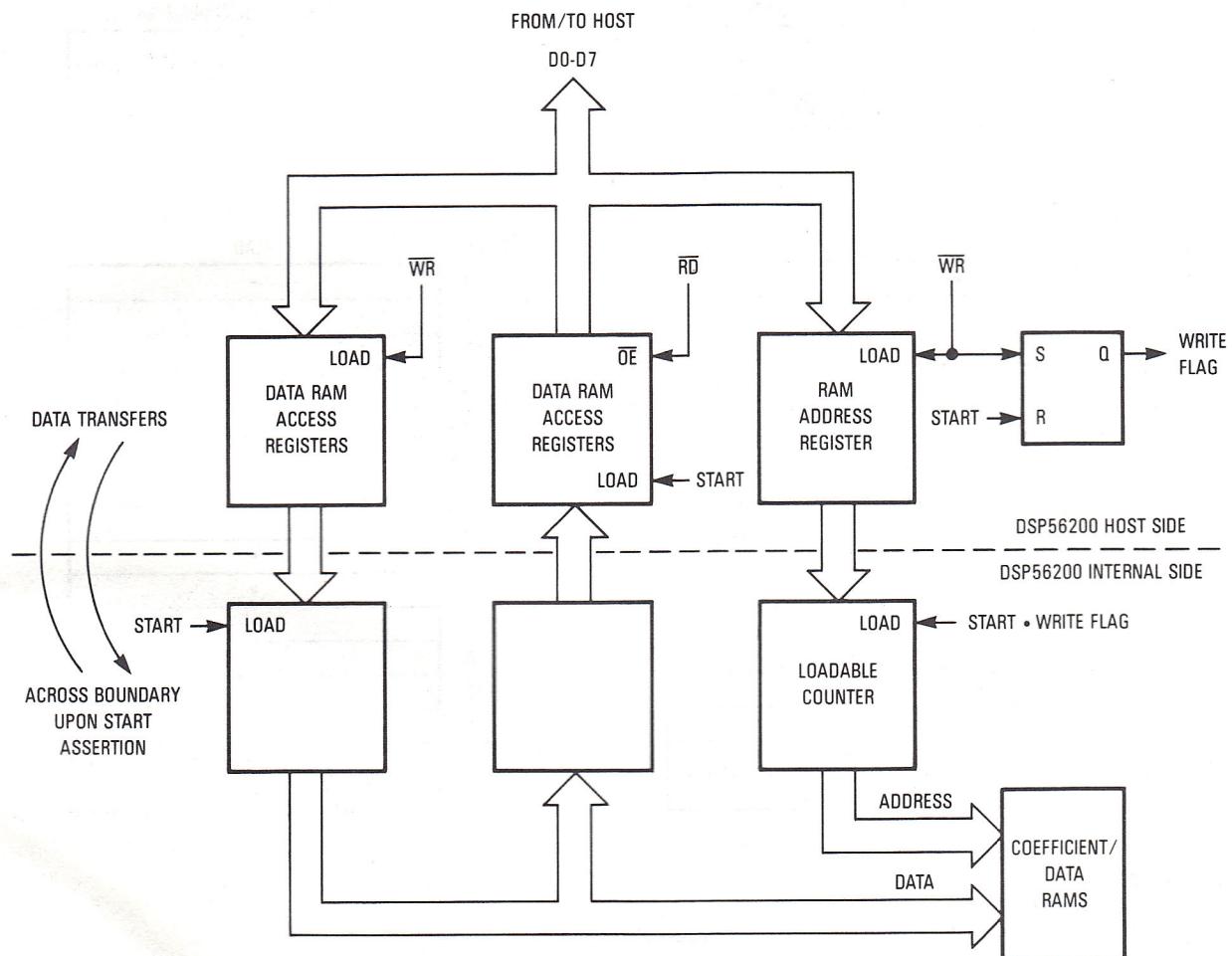


Figure 8. Double-Buffered Asynchronous Parallel Interface

then be transferred to the first location in the X data shift register.

Output Sample $y(n-1)$

While sample $x(n)$ is being held in the appropriate input data register, the first tap in the X data shift register will contain input sample $x(n-1)$, and the FIR calculation will

be done with $x(n-1)$ as the most recent value. The result of the FIR calculation is, therefore, the output associated with $x(n-1)$ through $x(n-1-N)$, namely, $y(n-1)$.

Serial Sum for $y(n-2)$

The serial sum calculation for $y(n-2)$ occurs for both the cascade and standalone modes. In standalone, SSI is

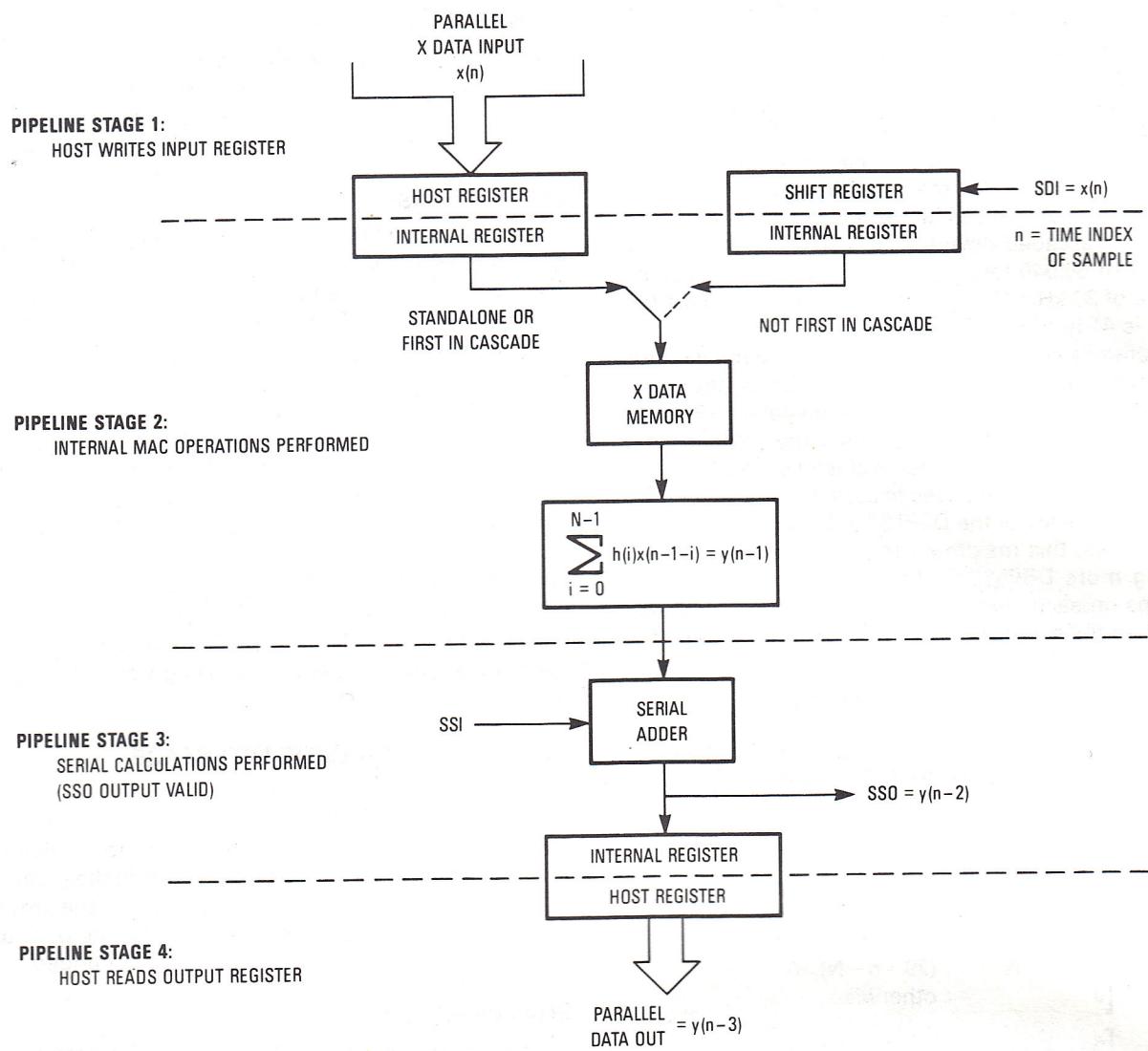


Figure 9. Pipeline Delay

tied to logic zero so that the MAC result is simply added to zero. The sum is then transferred to the asynchronous parallel interface.

Output Sample $y(n-3)$

At the same time that the above states are occurring, the output data register will contain the final result, $y(n-3)$, for the input which was received three sample periods earlier, namely, $x(n-3)$.

PERFORMANCE

Table 1 illustrates examples of DSP56200 systems in which the maximum sampling frequency (in kHz) is given as a function of the number of taps and the number of DSP56200s in cascade.

Table 1 shows that the ability to cascade is advantageous for extending the length of the FIR filter and also useful for increasing the maximum signal frequency that

Table 1. Maximum Sampling Frequency (kHz)

| Mode | Number in Cascade | Total Number of Taps | | | | | |
|------|-------------------|----------------------|-----|-----|-----|-----|------|
| | | 32 | 64 | 128 | 256 | 512 | 1024 |
| SFIR | 1 | 227 | 132 | 71 | 37 | — | — |
| | 4 | | | 222 | 132 | 71 | 37 |
| AFIR | 1 | 123 | 69 | 37 | 19 | — | — |
| | 4 | | | 120 | 69 | 37 | 19 |
| DFIR | 1 | 122 | 68 | 36 | NA | — | — |

can be filtered for a given number of taps. There are bounds on N and the number in cascade, however, given that the sampling rate is to be maximized. For example, since it takes 32 cycles to output the 32-bit serial sum and since there are nine additional cycles of overhead (for each device in the cascade, one additional cycle is required for the start bit in the serial links), the maximum number of devices that can be cascaded is $256 - 41 = 215$. Each of the cascaded devices could contribute 256 taps for a total of 55,040 taps while still maintaining a sampling rate of 37 kHz. The minimum number of cycles for a device is 41 in the SFIR mode.

Four parameters determine the maximum sampling rate of the DSP56200: the number of DSP56200s cascaded together; the number of taps (N) used on each DSP56200 (N should be distributed equally); the clock frequency of the DSP56200; and the selected mode of operation.

The formulas below are used to calculate the maximum sampling frequency of the DSP56200 for a given system. In many cases, this maximum rate can be increased by cascading more DSP56200 chips together and by using fewer taps on each chip. The user's sampling rate must be less than the maximum sampling frequency given by

$$\text{Maximum } f_S = f_{CK} / \# \text{cycles}$$

where

f_{CK} = DSP56200 input clock frequency

and

$$\begin{aligned} \# \text{cycles} &= \begin{cases} 12 + N + q & : \text{SFIR filter mode} \\ 18 + 2N + q & : \text{DFIR filter mode } (\leq 128) \\ 17 + 2N + r & : \text{single AFIR mode} \end{cases} \\ q &= \begin{cases} 29 + n - N & : (29 + n - N) > 0 \\ 0 & : \text{otherwise} \end{cases} \\ r &= \begin{cases} 30 + n - N & : (30 + n - N) > 0 \\ 0 & : \text{otherwise} \end{cases} \end{aligned}$$

n = Number of chips cascaded together

N = Number of taps used on each chip

SIGNAL DESCRIPTION

The DSP56200 is a 28-pin dual-in-line package (DIP) integrated circuit. Its signals can logically be grouped into the following categories: asynchronous parallel interface, cascade interface, and clocks and power. Descriptions of these signals are presented in the following paragraphs.

ASYNCHRONOUS PARALLEL INTERFACE

Data Bus (D0-D7)

These eight pins provide a bidirectional data bus for communication with a host processor. The pins remain in the high-impedance state unless both RD and CS are asserted.

Register Address Pins (A0-A3)

A0-A3 select which register will be addressed when the chip select line is brought low and a read or write operation is performed. These pins operate in conjunction with the least significant bit of the configuration register.

Chip Select (\overline{CS})

This pin (active low) enables accesses to the chip operating registers. When not asserted, the D0-D7 lines will go into the high-impedance state, and all access to the chip is disabled. The CS pin must be high when the START pulse arrives.

Read Strobe (\overline{RD})

When \overline{RD} (active low) is asserted, the contents of the register specified by A0-A3 will be driven onto D0-D7. When RD is high, pins D0-D7 go into the high-impedance state.

Write Strobe (\overline{WR})

This pin (active low) enables host writes to the register specified by A0-A3. Data on D0-D7 must be valid for the specified setup time before the rising edge of WR.

CASCADE INTERFACE

Serial Data Input (SDI)

This pin is used in the cascade mode to receive data from the last tap of the data shift register in the preceding DSP56200 chip. It connects to the SDO pin of the previous chip in cascade. If the chip is first in cascade or is used in standalone mode, this pin should be grounded.

Serial Data Output (SDO)

This pin is used in the cascade mode to pass the last data sample in the data shift register to the next DSP56200 in the cascade. The SDO pin connects to the SDI pin of the next DSP56200. This pin is active in the standalone mode and can be left unconnected. Data is output least significant bit first, starting approximately 1.5 clock periods after the leading edge of the START pulse (see Figure 15). Sixteen bits are always transmitted.

Serial Sum Input (SSI)

This pin is used in the cascade mode to receive the partial sum from SSO of the preceding chip in the cascade. If the chip is first in the cascade or is used in standalone mode, this pin should be grounded.

Serial Sum Out (SSO)

This pin is primarily used in the cascade mode to pass the partial sums to the next DSP56200 in the cascade. The SSO pin is usually connected to the SSI pin of the next chip in the cascade. In the AFIR filter mode, the SSO pin of the last chip in the cascade is connected to the SEI pin on all chips cascaded, including itself. This pin should not be connected to SEI in standalone SFIR or DFIR filter modes. A START bit initiates the serial data transfer and is followed by the least significant bit of the partial sum.

The delay between the START bit and the START pulse depends on the position of the device in the cascade (see Figure 15). Thirty-two bits are always transmitted. This pin is active in the standalone mode and can be left unconnected.

Serial Error Input (SEI)

Used in the AFIR filter mode, this pin provides the means of receiving the error-term output from the last chip in the cascade. In the standalone AFIR filter mode, this pin is tied to the SSO pin. In the cascade AFIR filter mode, this pin is tied to the SSO pin of the last chip in the cascade. This pin should be grounded in the SFIR or DFIR filter modes. Coefficient update does not begin until the SSO START bit (i.e., the error data) has been received.

CLOCKS AND POWER

Clock Input (CLOCK)

This pin accepts the input clock for the DSP56200. The internal and external clocking frequencies are the same, and the maximum frequency for this input is 10.25 MHz. In cascaded systems, all DSP56200s must be driven from the same clock source. When CLOCK is held low and all inputs are tied to full CMOS levels, the device enters a low-power mode.

Start Processing Command (START)

This pin is used to provide a second clock to the chip at the system's sampling rate. This clock must be synchronized with the signal on the CLOCK pin to ensure proper operation. The START pulse duty cycle is not critical; however, the START pulse must be low for at least two clock periods prior to going high, signaling the start of a new cycle, and it must meet the ac electrical specifications (see Figure 15). The DSP56200 will not operate correctly if there are noise spikes on this pin. Power and ground pins should be bypassed as close to the device as possible to prevent ground loops.

V_{CC} (Power) and GND (Ground)

The DSP56200 provides three V_{CC} pins and three GND pins. As with any high-speed logic design, connecting and bypassing all V_{CC} and GND pins on the DSP56200 is crucial. Thus, 0.1 mF ceramic bypass capacitors with short (less than 0.5 inch) leads should be used.

REGISTER MODEL

The DSP56200 is initialized and accessed by the host through a set of control and data transfer registers. In addition, the registers provide access to values in the coefficient and data RAMs, allowing unused memory to be used as auxiliary system storage. All register access occurs through the asynchronous parallel interface.

Registers in the DSP56200 have been divided into two banks of 16 registers (Figure 6). Bank 0 contains the registers commonly accessed during real-time processing. Bank 1 registers are used for initializing the chip. The two banks share one common register, the configuration register, located at hex address, 0F, in each bank. Switching banks is done by complementing the least significant bit of this common register. This bit is not double buffered and does not require a start pulse to become effective. This bit acts as a fifth register address bit. Once the desired bank has been selected, the registers are accessed using A0-A3, RD or WR, and CS. The registers have been ordered so that they can be accessed by the host using a simple autoincrementing address mode.

Upon powerup, the user must initialize the chip's registers and RAMs, which normally involves writing into the configuration register to select the mode of operation and to select bank 1. The FIR tap length register is then written; this action programs the number of taps and also resets the chip. Upon reset, the chip timing is initialized, and the contents of the data RAM are undefined. Next, the configuration register is accessed to change to bank 0. Then the RAMs are usually initialized by writing a valid coefficient into each location of the coefficient RAM and a zero into each location of the data RAM. One data RAM location and one coefficient RAM location can be written each sampling period. The DSP56200 is then ready for real-time filtering (see Figures 10 and 11).

REGISTER DESCRIPTION

X1, X2 REGISTERS

The X1 register is a 16-bit register that functions as the data input register for all three filter modes. Data from the X1 register is copied into the data RAM shift register once per START cycle. In the DFIR filter mode, an additional register, called X2, functions as the second filter's data input register. The X2 register operates in a manner

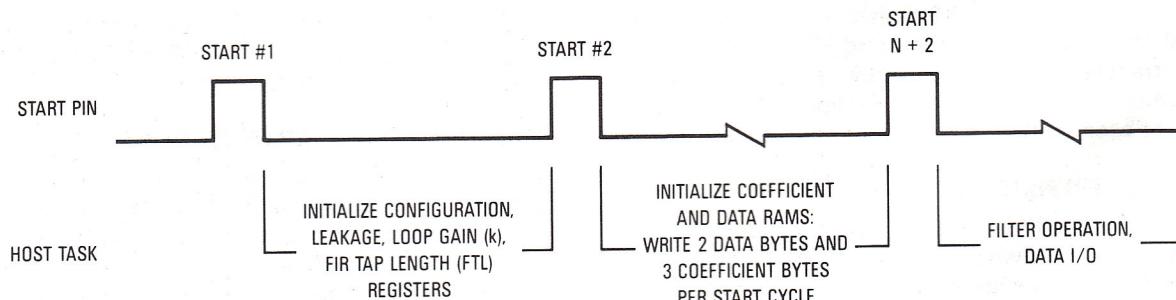
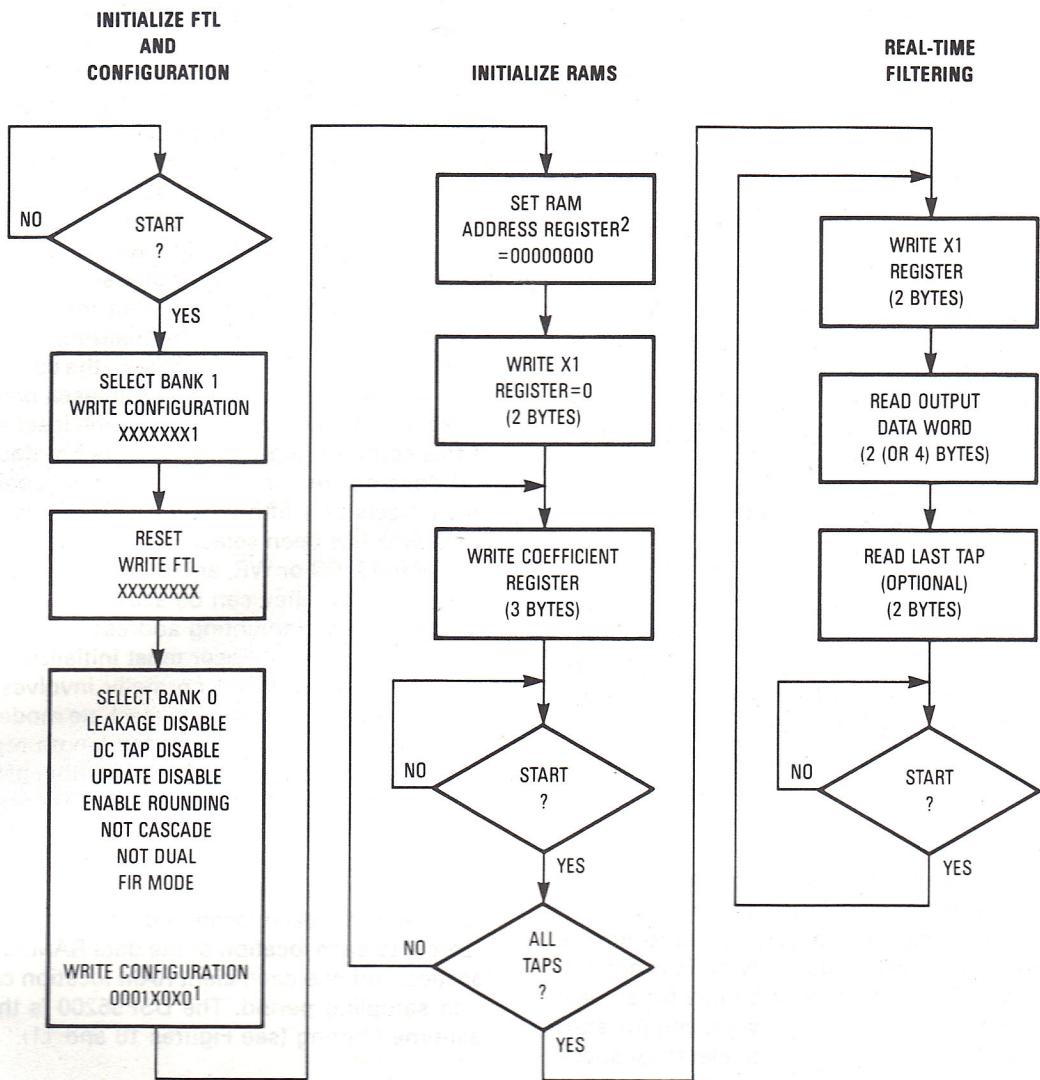


Figure 10. DSP56200 Initialization Sequence



NOTES:

- 1 The parallel interface is double buffered. A START pulse is used to transfer the data from the byte-wide input buffers to the internal registers with one exception. Bit 0 of the configuration register is transferred immediately to change the bank selection.
- 2 DSP56200 autoincrements this register.

Figure 11. Standalone Single FIR Flow Diagram

similar to X1. Data from the X2 register is copied into the data RAM shift register of the second FIR filter after X1 has been transferred; therefore, only one START pulse is required to initiate both filter operations. When X1 or X2 have not been written, their previous values are used.

D REGISTER (AFIR FILTER MODE ONLY)

The D register, a 16-bit register that functions as the reference (echo) input when the DSP56200 is used in the AFIR filter (echo cancellation) mode, is represented as d(n) in the adaptive filtering equations. When several chips are in cascade, the d(n) data must be input to the first chip in the cascade which must have bit 5 set to zero in

its configuration register. This register is not used in the nonadaptive modes of operation. When d(n) has not been written, its previous value is used.

K REGISTER (AFIR FILTER MODE ONLY)

The K register is a 16-bit register used only in the AFIR filter mode. K, the loop gain (convergence parameter) used in the LMS algorithm, is multiplied by the error term e(n) and tap x(n-i) to generate an updated value for h(i). That the loop gain factor K has an optimum value depending on the input signal power⁴ and the FIR tap length used can be shown theoretically.^{4,8} Found empirically, the values in Table 2 are recommended maximum values,

and the user should start with a value less than or equal to the K value shown for the corresponding FIR tap length chosen.

Table 2. Tap Length Versus Loop Gain Factor K

| FIR Tap Length | Maximum Recommended K |
|----------------|-----------------------|
| ≤32 (20 hex) | 0.750000 (6000 hex) |
| ≤64 (40 hex) | 0.375000 (3000 hex) |
| ≤128 (80 hex) | 0.125000 (1000 hex) |
| ≤192 (C0 hex) | 0.078125 (0A00 hex) |
| ≤256 (FF hex) | 0.0703125 (0900 hex) |

CAUTION

K must always be a positive number, i.e., bit 7 of the most significant byte should be zero at all times.

CONFIGURATION REGISTER

The configuration register is used to configure the modes and options of the DSP56200. The bits, shown in Figure 7, are explained in the following paragraphs.

FIR/AFIR determines whether the chip will operate as a fixed FIR filter or as an AFIR filter. This bit (which is set for an AFIR filter) should be set the same for all the DSP56200s in the cascade.

SFIR/DFIR determines whether the filter will be configured as a single or dual FIR filter. In the DFIR mode, the tap lengths of both filters are the same and are controlled by the FIR tap length register. The value of the FIR tap length in DFIR mode is limited to 127 (128 taps). In the DFIR mode, output data bytes 3 and 2 contain channel 1 output; whereas, data bytes 1 and 0 contain channel 2 output data (see Figure 6). Table 3 summarizes the valid modes for the chip as selected by bits 6 and 7 in the configuration register.

Table 3. DSP56200 Modes

| Adaptive/ Nonadaptive (Configuration Bit 7) | SFIR/DFIR (Configuration Bit 6) | Mode |
|--|---------------------------------------|-----------------------|
| 0 | 0 | SFIR Filter |
| 0 | 1 | DFIR Filter |
| 1 | 0 | Single AFIR Filter |
| 1 | 1 | (Operation Undefined) |

Position in Cascade selects whether the chip is configured to operate as standalone/first in cascade, or not first in cascade (see Table 4). Not first in cascade is selected when this bit is set. This bit must be cleared whenever the DSP56200 is used in the DFIR filter mode.

16-Bit Rounding selects whether the filter output will be represented as a 32-bit result or as a rounded 16-bit

Table 4. DSP56200 Cascade Configurations

| Configuration Bit 5 | System Configuration |
|---------------------|---|
| 0 | Single DSP56200 System (Standalone) |
| 0 | First DSP56200 in a Cascaded System |
| 1 | Not First DSP56200 in a Cascaded System |

result in the output register. In the latter case, data bytes 3 and 2 in the output register contain the valid 16-bit rounded result, and data bytes 1 and 0 contain invalid data for SFIR mode. In DFIR mode, both 16-bit outputs are rounded. This bit is set for 16-bit rounding. In cascade, this bit can be set only in the first chip (i.e., bit 5 in the configuration register set to zero) to prevent more than one rounding constant from being added to the sum. Rounding should not be used in the AFIR mode because a relatively large offset would be added to the error signal.

Adaptation Disable is used to disable the LMS algorithm in the AFIR filter mode. When this bit is set, the chip will continue to compute error terms, using the last set of updated filter coefficients. The leakage term is also disabled. In voice-echo-canceling applications, this bit is typically set when "doubletalk" is detected. The bit should be set the same for all DSP56200s in the cascade.

DC Tap Enable is set to turn on the dc tap option. The dc tap looks like a tap in the data shift register with a fixed value of \$7FFF. The dc tap is multiplied by its corresponding (last) coefficient during the FIR filtering phase; it is also used as the data value when updating the last coefficient in AFIR filter mode. The dc tap is normally used in the AFIR filter mode to remove differential dc offset in the converters. The dc tap is substituted for the last tap in the data shift register when it is enabled. The true last tap data is not lost, however, and can be read using the host data RAM access register or the last tap register. When several DSP56200s are in cascade, this bit should be set only in the last chip. The dc tap is also useful in the SFIR and DFIR modes for providing dc offset to the filtered output.

Leakage Enable is set when the use of leakage is desired in the coefficient update calculation. This bit should be set the same for all DSP56200s in cascade.

Register Bank Select selects which register bank is to be accessed by the host processor. The configuration register appears in both bank 0 and bank 1, allowing this bit to always be available for control of register bank selection. This bit is set for access to bank 1 and cleared for access to bank 0. Bit 0 is the only bit that is recognized immediately—that is, it operates asynchronously with respect to the START pulse, thereby allowing access to both banks during the same sampling period.

LEAKAGE REGISTER (AFIR FILTER MODE ONLY)

The leakage register is used in the AFIR filter mode when coefficient update is enabled (bit 4 of the configuration register) and when the leakage enable bit is set. Leakage is an 8-bit magnitude value used to control coefficient drift in adaptive filtering (see **ALGORITHMS**). All devices in the cascade configuration should be programmed with the same leakage value.

FIR TAP LENGTH REGISTER

This 8-bit register determines the number of taps used in the FIR filter. The register is loaded with the number of taps minus one. For example, if a 256-tap filter is desired, this register is loaded with 255. Valid values range from 3 to 255 in SFIR or AFIR filter modes and from 3 to 127 in the DFIR mode. In DFIR mode, both filters must have the same number of taps. For example, if two 14-tap FIR filters are desired, then the FIR tap length register should be set to 13. Writing to this register also resets the chip. Normally, this register is written immediately by the host upon powerup.

RAM ADDRESS REGISTER

This 8-bit register specifies which location will be selected during host accesses to the coefficient and data RAMs. This register, which allows access to taps being used within the filter and to any memory not used in the FIR filter calculation, automatically postincrements once each sampling period. Thus, one memory location in the coefficient RAM and one memory location in the data RAM at the same address can be accessed internally during one sampling period. In the DFIR filter mode, the data samples in the first FIR filter cannot be accessed using the RAM address register. All coefficient RAM locations can be accessed using the RAM address register in all modes, including DFIR. In the SFIR filter modes, the starting address of the memory outside the filter is $FTL + 1$ where FTL denotes FIR tap length; in the DFIR filter mode, the starting address of the unused memory is $2(FTL + 1)$. In the DFIR filter mode, the coefficients for the second filter start at $FTL + 1$.

COEFFICIENT RAM ACCESS REGISTER

This 24-bit register allows the user to read or write any location in the coefficient RAM. The host processor reads a RAM location by writing the desired address into the RAM address register, waiting for two pulses to occur on the START pin (i.e., two sample periods), and then reading the value of the coefficient RAM access register (see Figure 12(a)). The appropriate data transfers are made each start cycle, assuming the default case of host reads. Transfers for host writes are signaled by the host writing the desired value to this register, which sets the write flag (see Figure 8). The actual write operation to the coefficient RAM occurs during the following sampling period (see Figure 12(b)). All locations in the coefficient RAM can be accessed using the RAM address register.

DATA RAM ACCESS REGISTER

This 16-bit register allows the user to read or write any location in the data RAM. The host processor reads a

RAM location by writing the desired address into the RAM address register, waiting for two pulses to occur on the START pin (i.e., two sample periods), and then reading the value of the data RAM access register (see Figure 12(a)). The appropriate data transfers are made each start cycle, assuming the default case of host reads. Transfers for host writes are signaled by the host writing the desired value to this register, which sets the write flag (see Figure 8). The actual write operation to the data RAM occurs during the following sampling period (see Figure 12(b)). If the desired address resides within the FIR filter structure (a circular buffer in the data RAM), the DSP56200 automatically performs a logical-to-physical address conversion and correctly accesses the desired filter tap in the DFIR mode.

LAST TAP 1, LAST TAP 2 REGISTERS

The last tap 1 register provides the user with a copy of the last data sample in the data shift register. In the DFIR filter mode, the last tap 2 register contains the last data sample for the second FIR filter. These registers, provided only for convenience, are not required when cascading chips, since the last tap data sample is also transmitted serially to the next chip in the cascade. These registers, in conjunction with the X1 and X2 registers, are useful for signal power calculations.

OUTPUT DATA REGISTER

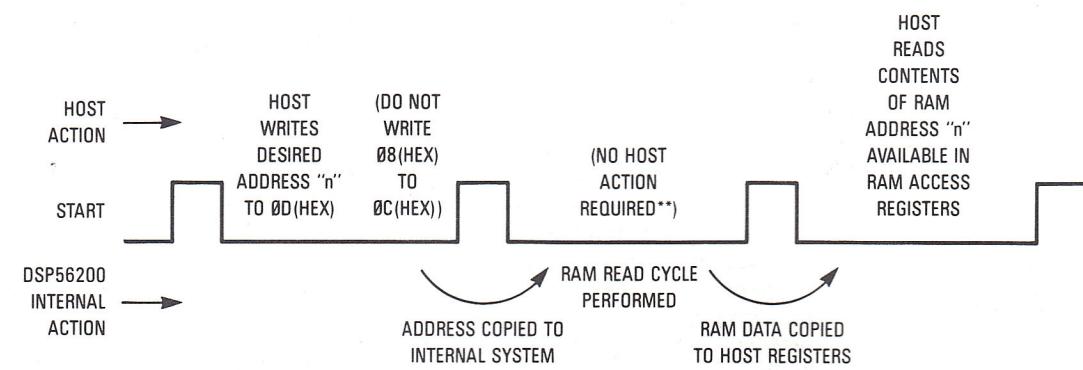
The output-data-register bytes 3 through 0 contain the final FIR or AFIR filter output. In the SFIR filter mode, the result will be four bytes (32 bits) unless the 16-bit rounding mode is set. When the 16-bit rounding mode is set, bytes 3 and 2 contain a valid, rounded, 16-bit result, and bytes 1 and 0 contain invalid data. Only 16 bits of output are available for each channel in the DFIR filter mode. Bytes 3 and 2 contain the output for the first FIR filter; bytes 1 and 0 contain the output for the second FIR filter. Both outputs are rounded to 16 bits if rounding is enabled in the configuration register. Otherwise, the outputs are truncated to 16 bits. In the AFIR filter mode, the output is the negative of the error, $-e(n)$. If several DSP56200s are in cascade, the last chip contains the final output.

APPLICATIONS

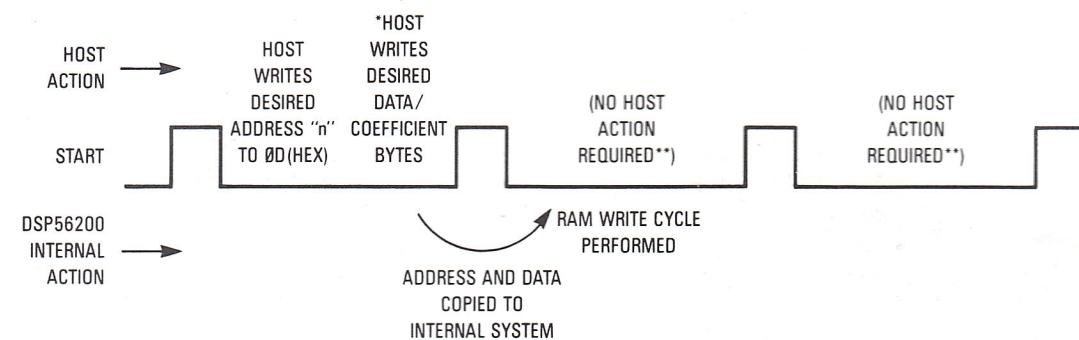
ECHO CANCELLATION USING THE DSP56200

A block diagram of a speakerphone is shown in Figure 13. Two echo cancelers are shown: one to cancel a telephone-line (electrical) echo and another to cancel an acoustic echo. Echo cancellation is necessary to allow full-duplex operation and to prevent the telephone from "singing" or breaking into oscillation.

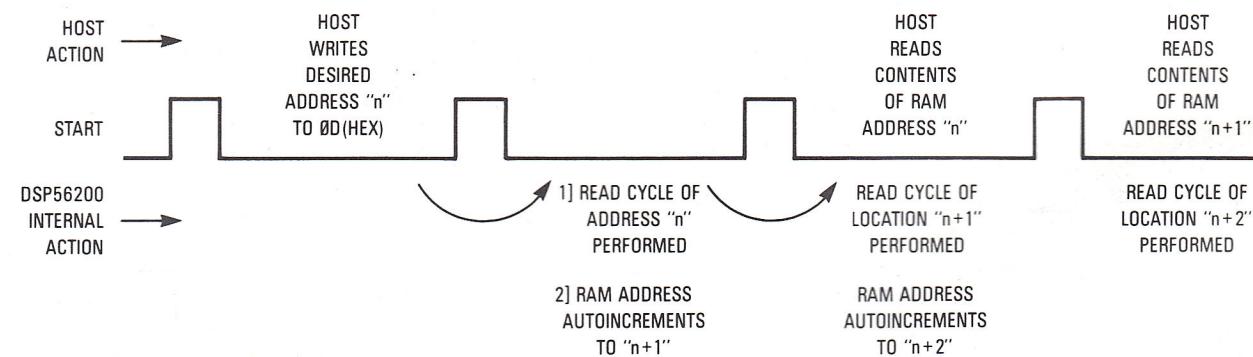
The acoustic-echo-canceler (AEC) composite signal consists of the desired near-end-talker and/or the uncorrelated far-end-talker signal that has been output from the speaker and echoed back to the microphone. The electrical-echo-canceler (EEC) composite signal consists of the desired far-end-talker and/or the uncorrelated near-end-talker signal, which has been echoed into the receive channel due to the impedance mismatch at the 2W/4W



(a) Host Data/Coefficient RAM Read



(b) Host Data/Coefficient RAM Write



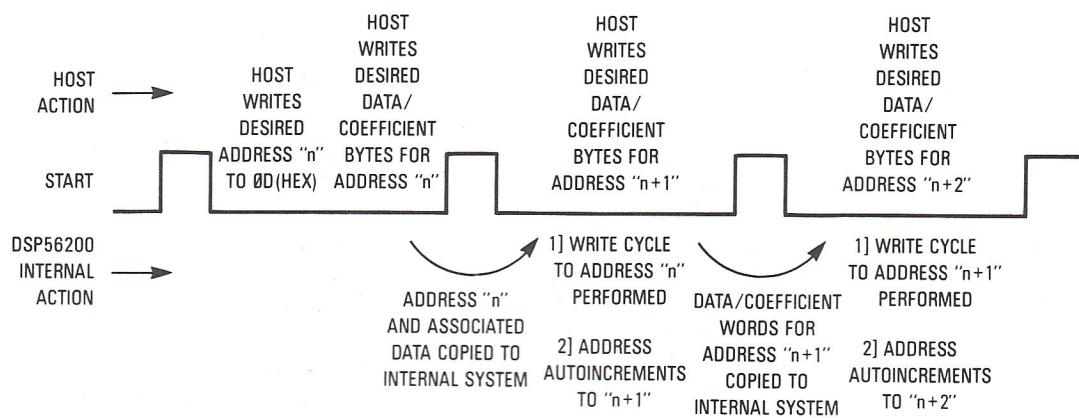
(c) Pipelined Read Cycles

NOTE:

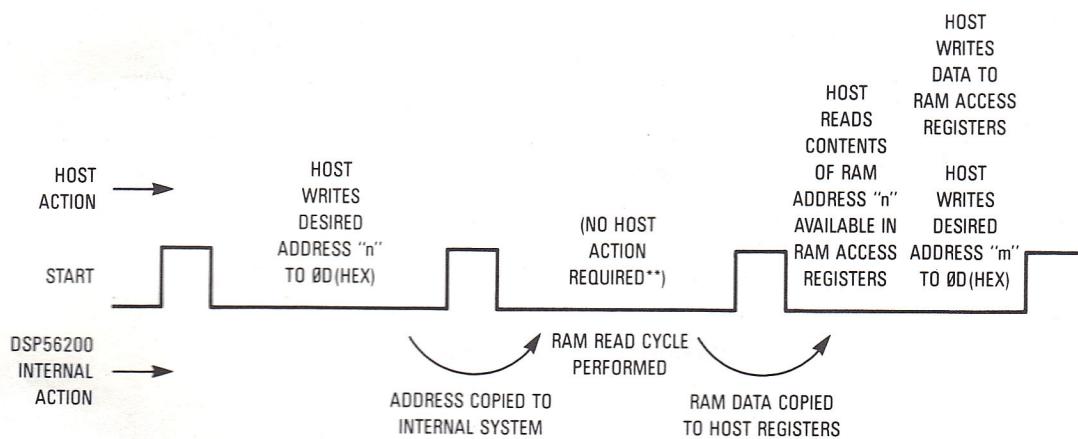
*The user must write entire coefficient (3 bytes) and/or data (2 bytes). Byte writes of coefficient/data are not supported.

**Host action not required, but also not prohibited.

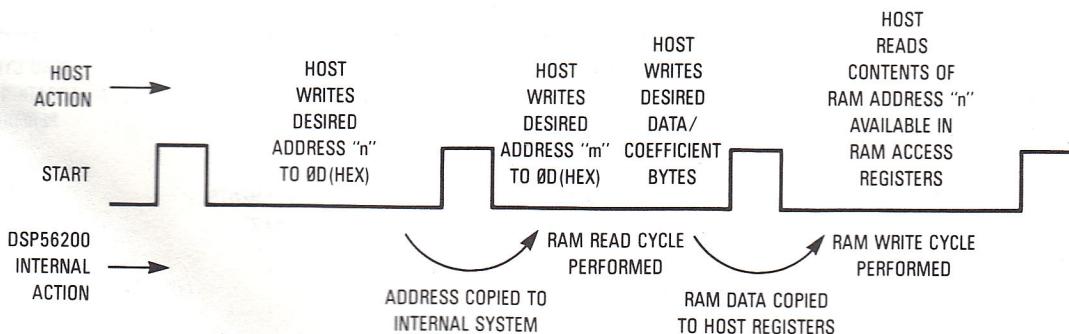
Figure 12. Pipelined Read and/or Write Operations (Sheet 1 of 2)



(d) Pipelined Write Cycles



(e) Overlapped Read/Write Operation Example



(f) Alternate Example of Overlapped Operation

Figure 12. Pipelined Read and/or Write Operations (Sheet 2 of 2)

interface to the telephone line. When either composite signal consists of the echo only, the cancelers will adapt their filter coefficients to extract a replica of their respective echoes from their inputs and subtract the replicas from their composite signals. As described in **ALGORITHM**, the error terms are forced toward zero. When the error term has been minimized, the adaptive filter impulse response is said to have converged to the impulse response of the echo channel. When both talkers are active simultaneously (a state referred to as double-talk), the error term will include the desired signal as well as the uncanceled echo.

When doubletalk is detected in speakerphones, the coefficient update processes are usually suspended because the error signal increases. This larger error signal would result in poorer convergence and less cancellation. The adaption process can be disabled on the DSP56200 by simply setting bit 3 in the configuration register. Suspending the adaption process during doubletalk in speakerphones is allowable because doubletalk is minimal.

In contrast to the voice-echo cancelers in a speakerphone application, data-echo cancelers in V.32 modems operate primarily in the doubletalk mode, and only one echo, the telephone-line echo, must be canceled.⁹ In this case, the adaption process can not be suspended.

These two echo-cancelation applications were selected to highlight two critical design parameters of echo cancelers — the number of taps and the coefficient word size. In the case of the speakerphone, the telephone-line echo is typically less than 32 ms (256 taps at 8 kHz sampling frequency) and can therefore be canceled with one DSP56200. On the other hand, the length of the acoustic echo depends on the size and makeup of the acoustic chamber and is usually greater than 32 ms. Therefore, more than one DSP56200 must be cascaded to cancel the acoustic echoes.

The degree of cancellation depends not only on the number of taps but also on the coefficient word size and update arithmetic. Generally, 3 dB of echo return loss enhancement (ERLE) per coefficient bit is adequate. For voice-echo cancelers, 30 to 40 dB of ERLE is adequate for most applications, which translates into a 16- to 18-bit coefficient-word-length requirement after the loss in the hybrid (assumed to be 12 to 14 dB) is considered. In contrast, for V.32 data-echo cancelers, 50 to 60 dB of ERLE is required if $<10^{-5}$ bit error rates are to be realized, which translates into a 20- to 24-bit coefficient-word-size requirement. For maximum performance, round the updated coefficient prior to writing it back to memory for use in the next convolution sum. The DSP56200 uses convergent rounding when updating coefficients. Finally,

the worst error associated with updating coefficients happens when overflow occurs. The DSP56200 is protected against overflow as described in **ARITHMETIC UNIT**.

In Figure 13, both echo cancelers are interfaced to A/D and D/A subsystems. These subsystems often have dc offsets that should be canceled. A dc offset can be canceled using a DSP56200 with bit 2 in the configuration register set in either the SFIR or AFIR filter modes.

A schematic of a four-chip speakerphone adaptive filter system, including both converter subsystems, four DSP56200s (three connected in cascade), and a DSP56001 controller, is shown in Figure 14.

FOOTNOTES

¹A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, New Jersey: Prentice-Hall, 1975, pp. 195-271.

²B. Widrow, "Adaptive Filters," *Aspects of Network and System Theory*, R. E. Kalman and N. De Claris Eds. New York: Holt, Reinhart and Winston, 1970, pp. 563-587.

³B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, New Jersey: Prentice-Hall, 1985, pp. 193-404.

⁴B. Widrow, *et al.*, "Adaptive Noise Cancelling: Principles and Applications." *Proc. IEEE*, vol. 63, no. 12 (Dec. 1975), pp. 1692-1716.

⁵H. M. Sondhi and D. Mitra, "New Results on the Performance of a Well-Known Class of Adaptive Filters," *Proc. IEEE*, vol. 64, no. 11 (Nov. 1976), pp. 1583-1597.

⁶A. Weiss and D. Mitra, "Digital Adaptive Filters: Conditions for Convergence, Rates of Convergence, Effects of Noise and Errors Arising from the Implementation," *IEEE Trans. on Information Theory*, vol. IT-25 (Nov. 1979), pp. 637-652.

⁷Y. G. Tao, *et al.*, "A Cascadable VLSI Echo Canceller," *IEEE Journal on Sel. Area in Comm.*, vol. SAC-2, no. 2 (March 1984), pp. 297-303.

⁸M. M. Sondhi and D. A. Berkley, "Silencing Echoes on the Telephone Network," *Proc. IEEE*, vol. 68, no. 8 (Aug. 1980) pp. 948-963.

⁹K. H. Mueller, "A New Digital Echo Canceler for Two-Wire Full-Duplex Data Transmission," *IEEE Trans. Communications*, vol. COM-24 (Sept. 1976), pp. 956-962.

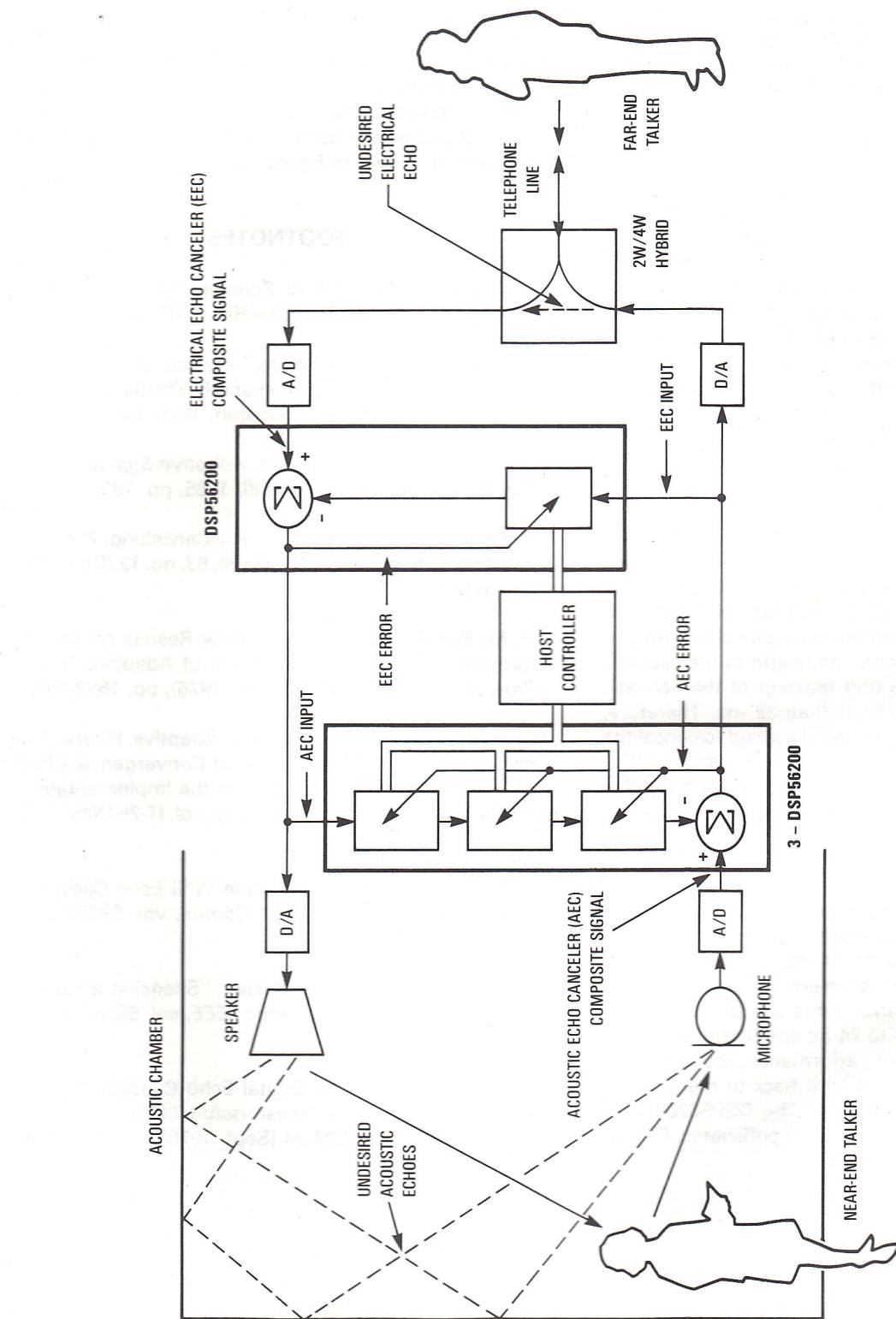


Figure 13. Speakerphone Application — Block Diagram

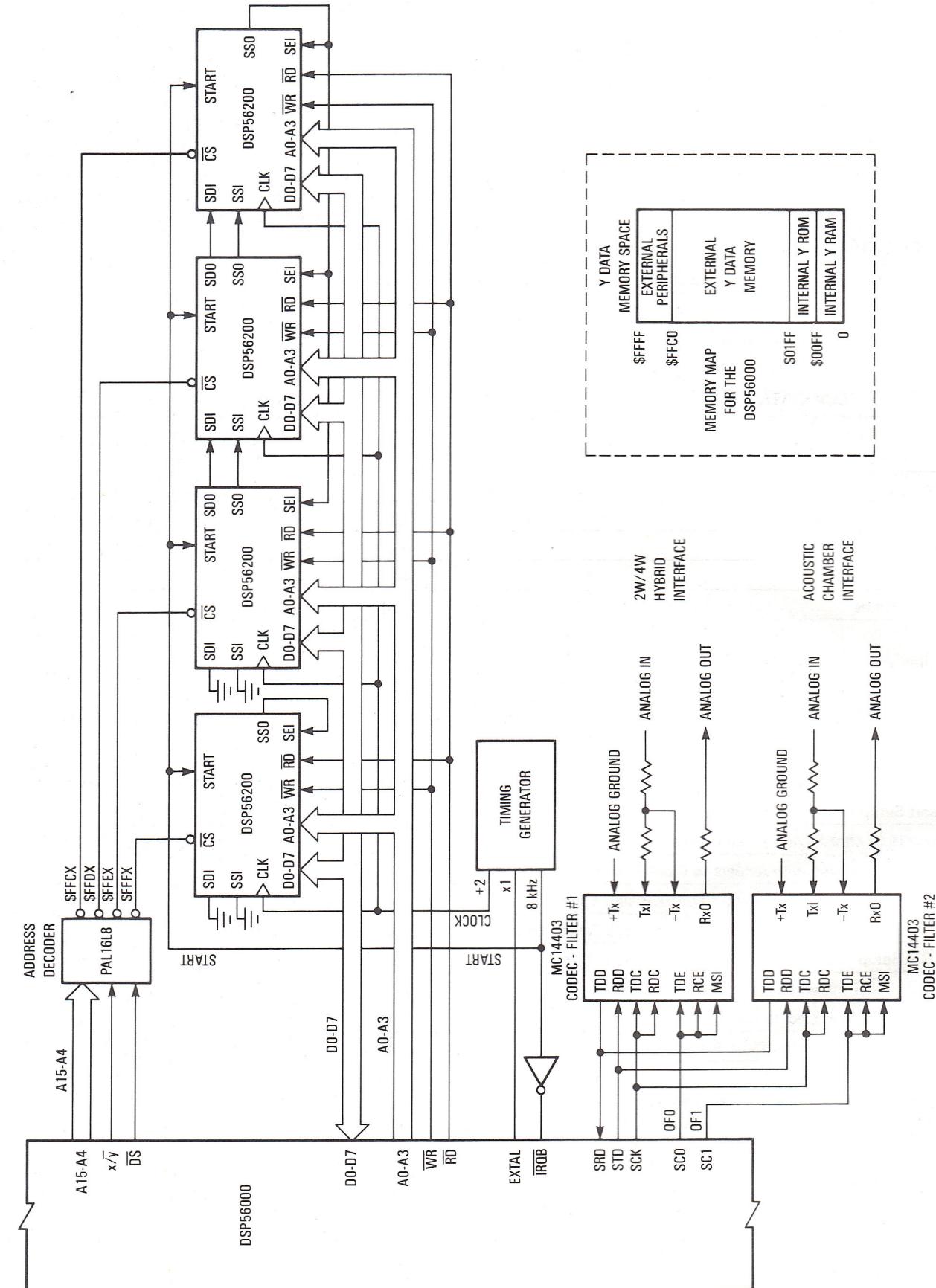
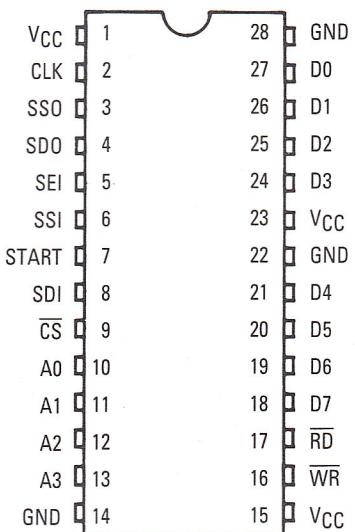


Figure 14. Four-Chip Speakerphone Adaptive Filter System

MECHANICAL DATA

PIN ASSIGNMENT



ORDERING INFORMATION ($T_A = -40^\circ\text{C}$ to 85°C)

| Package Type | Frequency | Order Number |
|---------------------|-----------|--------------|
| Ceramic L Suffix | 10.25 MHz | DSP56200L10 |