

## SECTION 1 INTRODUCTION

The DSP56001 and DSP56000 are user-programmable, CMOS digital signal processors (DSPs) which are optimized to execute DSP algorithms in as few operations as possible, while maintaining a high degree of accuracy. The architecture has been designed to maximize throughput in data-intensive DSP applications. This design provides a dual-natured, expandable architecture with sophisticated on-chip peripherals and general-purpose I/O. The architecture, on-chip peripherals, and the low power consumption of the DSP56000/DSP56001 have minimized the complexity, cost, and design time needed to add the power of DSP to any design.

The DSP56000 is read-only memory (ROM) based, and is factory programmed with user software for minimum cost in high-volume applications. The DSP56001 is an off-the-shelf random-access memory (RAM) based processor designed to load its program from an external source. The difference between the two processors is their respective on-chip memory resources. A secure version of the DSP56000, which prevents unauthorized access to the internal program memory, is also available.

This manual is written for both the DSP56000 and DSP56001. Normally, the reference will be to the DSP56000/DSP56001. However, when the two processors differ, they will be cited individually.

### 1.1 ORIGIN OF THE DSP56000 ARCHITECTURE

DSP is the arithmetic processing of real-time signals sampled at regular intervals and digitized. Examples of DSP processing include the following:

- Filtering of signals
- Convolution, which is the mixing of two signals
- Correlation, which is a comparison of two signals
- Rectification of a signal
- Amplification of a signal
- Transformation of a signal

All of these functions have traditionally been performed using analog circuits. Only recently has technology provided the processing power necessary to digitally perform these and other functions using DSPs.

Figure 1-1 shows a description of analog signal processing. The circuit in the illustration filters a signal from a sensor using an operational amplifier, and controls an actuator with the result. Since the ideal filter is impossible to design, the engineer must design the filter

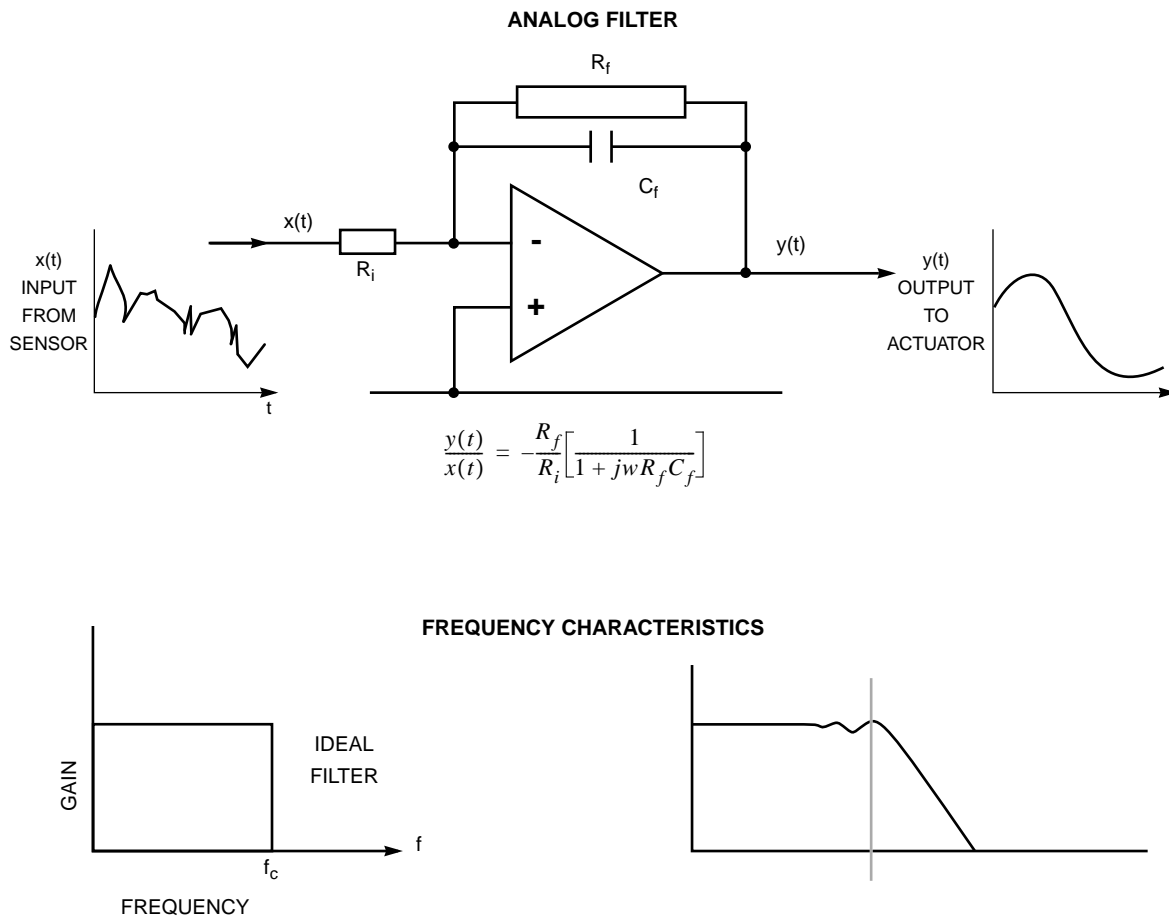


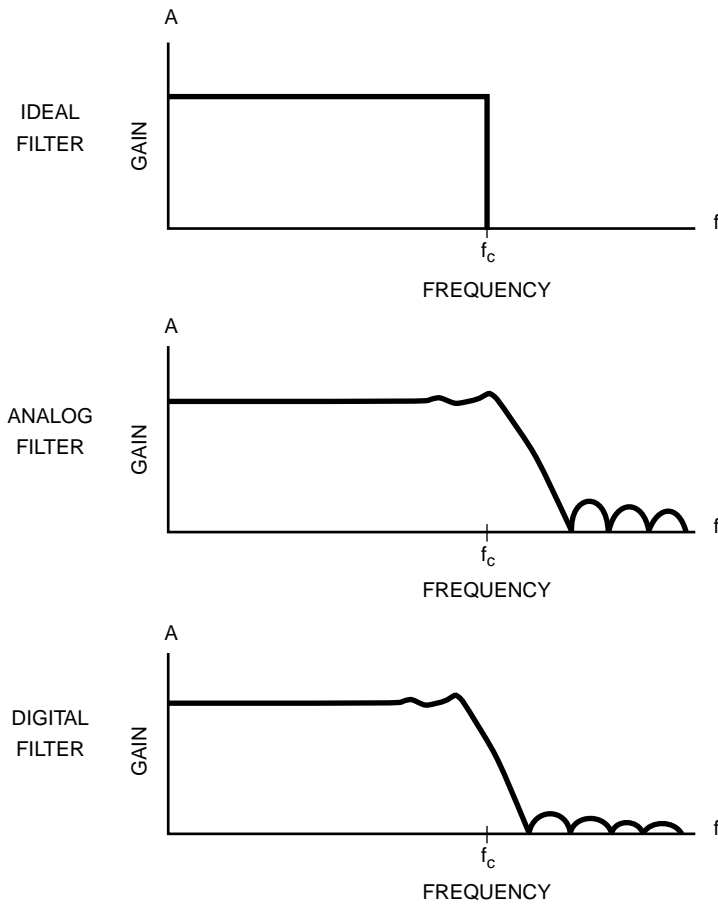
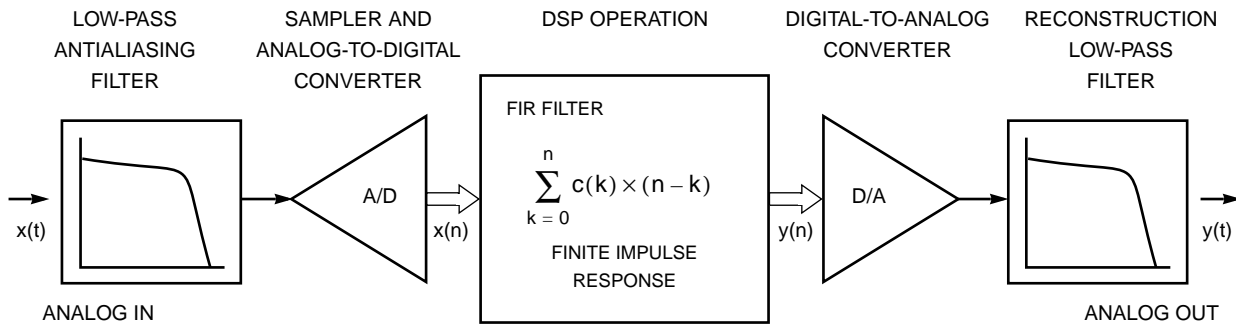
Figure 1-1 Analog Signal Processing

for acceptable response, considering variations in temperature, component aging, power-supply variation, and component accuracy. The resulting circuit typically has low noise immunity, requires adjustments, and is difficult to modify.

The equivalent circuit using a DSP is shown in Figure 1-2. This application requires an analog-to-digital (A/D) converter and digital-to-analog (D/A) converter in addition to the DSP. Even with these additional parts, the component count can be lower using a DSP due to the high integration available with current components.

Processing in this circuit begins by band-limiting the input with an antialias filter, eliminating out-of-band signals that can be aliased back into the pass band due to the sampling process. The signal is then sampled, digitized with an A/D converter, and sent to the DSP.

.The filter implemented by the DSP is strictly a matter of software. The DSP can directly implement any filter that can also be implemented using analog techniques. Also, adaptive filters can be easily implemented using DSP, whereas these filters are extremely difficult to implement using analog techniques.



**Figure 1-2 Digital Signal Processing**

The DSP output is processed by a D/A converter and is low-pass filtered to remove the effects of digitizing. In summary, the advantages of using the DSP include the following:

- Fewer components
- Stable, deterministic performance
- Wide range of applications
- High noise immunity and power-supply rejection
- Self-test can be built in

- No filter adjustments
- Filters with much closer tolerances
- Adaptive filters easily implemented

The DSP56000/DSP5001 was not designed for a particular application but was designed to execute commonly used DSP benchmarks in a minimum time for a single-multiplier architecture. For example, a cascaded, 2<sup>nd</sup>-order, four-coefficient infinite impulse response (IIR) biquad section has four multiplies for each section. For that algorithm, the theoretical minimum number of operations for a single-multiplier architecture is four per section. Table 1-1 shows a list of benchmarks with the number of instruction cycles the DSP56000/DSP56001 uses compared to the number of multiplies in the algorithm.

**Table 1-1 Benchmark Summary in Instruction Cycles**

Benchmark	DSP56000/DSP56001 Number of Cycles	Number of Algorithm Multiplies
Real Multiply	3	1
N Real Multiplies	2N	N
Real Update	4	1
N Real Updates	2N	N
N Term Real Convolution (FIR)	N	N
N Term Real * Complex Convolution	2N	N
Complex Multiply	6	4
N Complex Multiplies	4N	N
Complex Update	7	4
N Complex Updates	4N	4N
N Term Complex Convolution (FIR)	4N	4N
N <sup>th</sup> - Order Power Series	2N	2N
2 <sup>nd</sup> - Order Real Biquad Filter	7	4
N Cascaded 2 <sup>nd</sup> - Order Biquads	4N	4N
N Radix Two FFT Butterflies	6N	4N

These benchmarks and others are used independently or in combination to implement functions. The characteristics of these functions are controlled by the coefficients of the benchmarks being executed. Useful functions using these and other benchmarks include the following:

**Digital Filtering**

- Finite Impulse Response (FIR)
- Infinite Impulse Response (IIR)
- Matched Filters (Correlators)
- Hilbert Transforms
- Windowing
- Adaptive Filters/Equalizers

**Signal Processing**

- Compression (e.g., Linear Predictive Coding of Speech Signals)
- Expansion
- Averaging
- Energy Calculations
- Homomorphic Processing
- Mu-law/A-law to/from Linear Data Conversion

**Data Processing**

- Encryption/Scrambling
- Encoding (e.g., Trellis Coding)
- Decoding (e.g., Viterbi Decoding)

**Numeric Processing**

- Scaler, Vector, and Matrix Arithmetic
- Transcendental Function Computation (e.g., Sin(X), Exp(X))
- Other Nonlinear Functions
- Pseudo-Random-Number Generation

**Modulation**

- Amplitude
- Frequency
- Phase

**Spectral Analysis**

- Fast Fourier Transform (FFT)
- Discrete Fourier Transform (DFT)
- Sine/Cosine Transforms
- Moving Average (MA) Modeling
- Autoregressive (AR) Modeling
- ARMA Modeling

Useful applications are based on combining these and other functions. DSP applications affect almost every area in electronics because any application for analog electronic circuitry can be duplicated using DSP. The advantages in doing so are becoming more compelling as DSPs become faster and more cost effective.

DSPs are also being used as high-speed math processors in many purely digital computer applications. Some typical applications for DSPs are presented in the following list:

**Telecommunication**

- Tone Generation
- Dual-Tone Multifrequency (DTMF)
- Subscriber Line Interface
- Full-Duplex Speakerphone
- Teleconferencing
- Voice Mail
- Adaptive Differential Pulse Code Modulation (ADPCM) Transcoder
- Medium-Rate Vocoders
- Noise Cancelation
- Repeaters
- Integrated Services Digital Network (ISDN) Transceivers
- Secure Telephones

**Data Communication**

- High-Speed Modems
- Multiple Bit-Rate Modems
- High-Speed Facsimile

**Radio Communication**

- Secure Communications
- Point-to-Point Communications
- Broadcast Communications
- Cellular Mobile Telephone

**Computer**

- Array Processors
- Work Stations

- Personal Computers
- Graphics Accelerators

**Image Processing**

- Pattern Recognition
- Optical Character Recognition
- Image Restoration
- Image Compression
- Image Enhancement
- Robot Vision

**Graphics**

- 3-D Rendering
- Computer-Aided Engineering (CAE)
- Desktop Publishing
- Animation

**Instrumentation**

- Spectral Analysis
- Waveform Generation
- Transient Analysis
- Data Acquisition

**Speech Processing**

- Speech Synthesizer
- Speech Recognizer
- Voice Mail
- Vocoder
- Speaker Authentication
- Speaker Verification

**Audio Signal Processing**

- Digital AM/FM Radio
- Digital Hi-Fi Preamplifier
- Noise Cancelation
- Music Synthesis
- Music Processing
- Acoustic Equalizer

**High-Speed Control**

- Laser-Printer Servo
- Hard-Disk Servo
- Robotics
- Motor Controller
- Position and Rate Controller

**Vibration Analysis**

- Electric Motors
- Jet Engines
- Turbines

**Medical Electronics**

- Cat Scanners
- Sonographs
- X-Ray Analysis
- Electrocardiogram
- Electroencephalogram
- Nuclear Magnetic Resonance Analysis

**Digital Video**

- Digital Television
- High-Resolution Monitors

**Radar and Sonar Processing**

- Navigation
- Oceanography
- Automatic Vehicle Location
- Search and Tracking

**Seismic Processing**

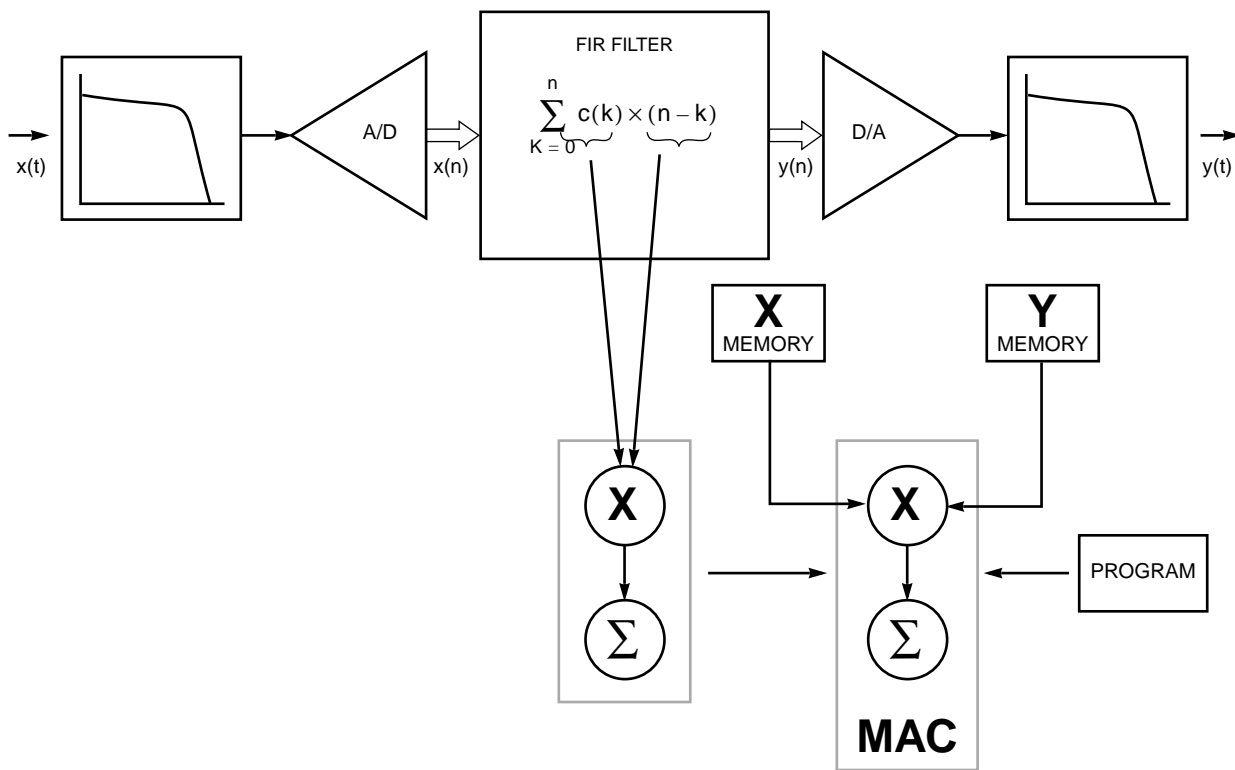
- Oil Exploration
- Geological Exploration

As shown in Figure 1-3, the keys to DSP are as follows:

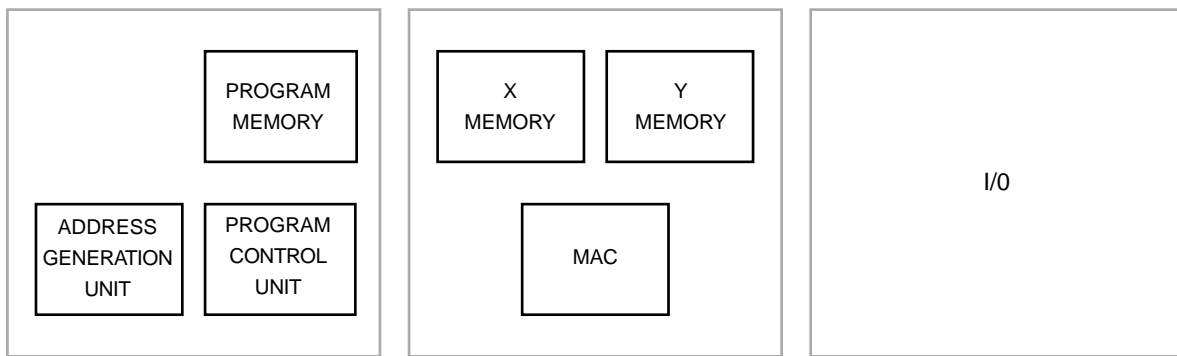
- The Multiply/Accumulate (MAC) operation
- Fetching operands for the MAC
- Program control to provide versatile operation
- Input/Output to move data in and out of the DSP

MAC is the basic operation used in DSP. Figure 1-3 shows how the architecture of the DSP56000/DSP56001 was designed to match the shape of the MAC operation. The two operands, C() and X(), are directed to a multiply operation, and the result is summed. This process is built into the DSP56000/DSP56001 by using two separate memories (X and Y) to feed a single-cycle MAC. The entire process must occur under program control to direct the correct operands to the multiplier and save the accumulator as needed. Since the two memories and the MAC are independent, it is possible to perform two moves (a multiply and an accumulate) in a single operation. As a result, many of the benchmarks shown in Table 1-1 can be executed at or near the theoretical maximum speed for a single-multiplier architecture.

Figure 1-3 shows how the MAC, memories, and program control unit in Figure 1-3 are configured in the DSP56000/DSP56001. Three independent memories and memory buses move two operands to the MAC while concurrently fetching a program instruction. The address generation unit (AGU) is divided into two arithmetic units which independently control the X and Y memories and feed operands to the MAC. Figure 1-3 also features an additional block labeled "I/O". Many other DSPs need external communications circuitry to interface with peripheral circuits (such as A/D converters, D/A converters, or host processors). The DSP56000/DSP56001 provides on-chip serial and parallel interfaces, represented by the I/O block, to simplify this connection problem. Figure 1-4 is a block diagram of the DSP56000 showing all the major



**Figure 1-3 DSP Hardware Origins**



**Figure 1-3 DSP Block Diagram**

blocks with their interconnecting buses. The DSP56000 Family of processors has a dual Harvard architecture optimized for MAC operations

**1.2 SUMMARY OF DSP56000 FAMILY FEATURES**

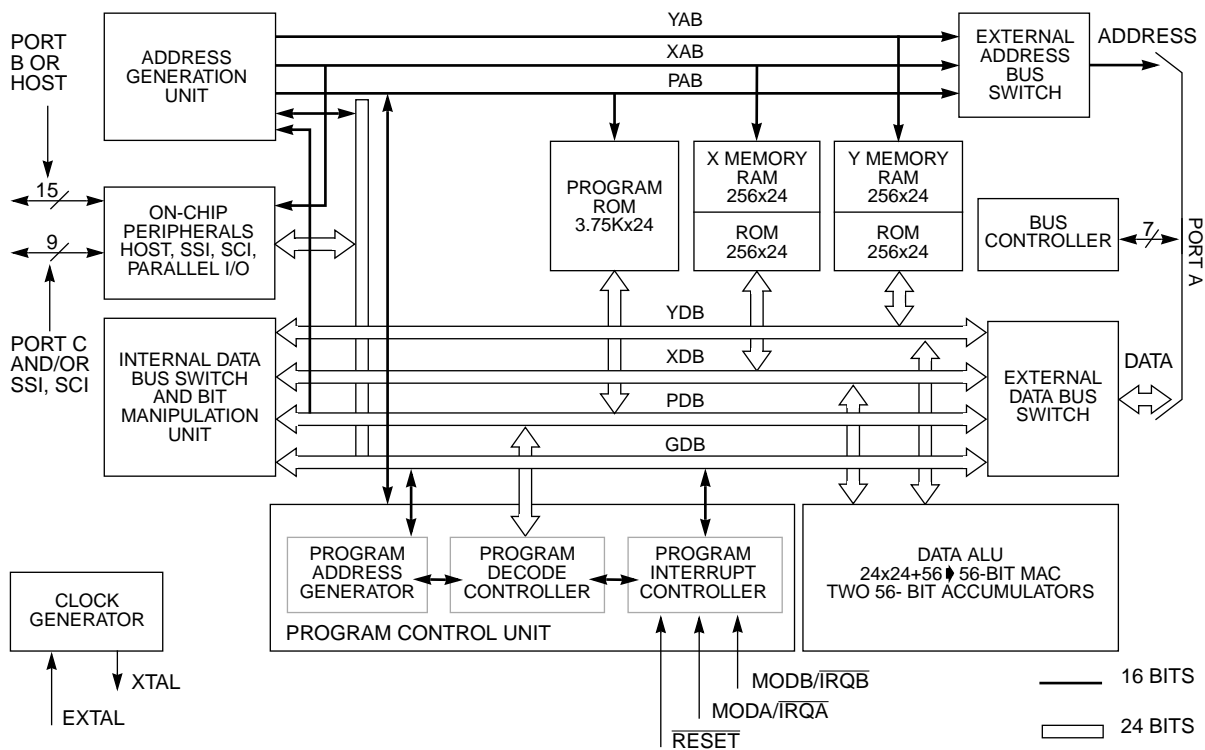
The DSP56000 and DSP56001 are the first two members of Motorola's Family of HCMOS, low-power, general-purpose DSPs. The DSP56001 features 512 words of full-speed, on-chip, program RAM, two preprogrammed data ROMs, and special on-chip bootstrap

hardware to permit convenient loading of user programs into the program RAM. The DSP56001 is an off-the-shelf part, since it has no user-programmable, on-chip ROMs. The DSP56000 features 3.75K words of full-speed, on-chip, program ROM instead of 512 words of program RAM.

The heart of the processor consists of three execution units operating in parallel: the data arithmetic logic unit (ALU), the AGU, and the program control unit. The DSP56000/DSP56001 has MCU-style on-chip peripherals, program memory, data memory, and a memory expansion port. The MPU-style programming model and instruction set allow straightforward generation of efficient, compact code.

The high throughput of the DSP56000/DSP56001 makes it well-suited for communication, high-speed control, numeric processing, computer applications, and audio applications. The main features facilitating this throughput are as follows:

- **Speed** — At 10.25 million instructions per second (MIPS), the DSP56000/DSP56001 can execute a 1024-point complex Fast Fourier Transform (FFT) in 3.23 ms.
- **Precision** — The data paths are 24 bits wide, providing 144 dB of dynamic range; intermediate results held in the 56-bit accumulators can range over 336 dB.
- **Parallelism** — Each on-chip execution unit (AGU, program control unit, data ALU),



**Figure 1-4 DSP56000 Block Diagram**



memory, and peripheral operates independently and in parallel with the other units through a sophisticated bus system. The data ALU, AGUs, and program control unit operate in parallel so that an instruction prefetch, a 24-bit x 24-bit multiplication, a 56-bit addition, two data moves, and two address-pointer updates using one of three types of arithmetic (linear, modulo, or reverse-carry) can be executed in a single instruction cycle. This parallelism allows a four-coefficient IIR filter section to be executed in only four cycles, the theoretical minimum for single-multiplier architecture. At the same time, the two serial controllers can send and receive full-duplex data, and the host port can send/receive simplex data.

- **Integration** — In addition to the three independent execution units, the DSP56000/DSP56001 has six on-chip memories, three on-chip MCU-style peripherals (serial communication interface (SCI), synchronous serial interface (SSI), and host interface), a clock generator, and seven buses (three address and four data), making the overall system low cost, low power, and compact.
- **Invisible Pipeline** — The three-stage instruction pipeline is essentially invisible to the programmer, allowing straightforward program development in either assembly language or a high-level language such as a full Kernighan and Ritchie C.
- **Instruction Set** — The 62 instruction mnemonics are MCU-like, making the transition from programming microprocessors to programming the DSP56000/DSP56001 as easy as possible. The orthogonal syntax supports controlling the parallel execution units. The hardware DO loop instruction and the repeat (REP) instruction make writing straightline code obsolete.
- **DSP56000/DSP56001 Compatibility** — The DSP56001 is identical to the DSP56000 except for the following features:
  - 12-word x 24-bit, on-chip program RAM instead of 3.75K program ROM
  - 32-word x 24-bit bootstrap ROM for loading the program RAM from either a byte-wide, memory-mapped ROM or from the host interface
  - On-chip X and Y data ROMs preprogrammed as positive Mu-law and A-law to linear expansion tables and a full, four-quadrant sine-wave table, respectively
- **Low Power** — As a CMOS part, the DSP56000/DSP56001 is inherently very low power; however, the following features can reduce power consumption to exceptionally low levels:
  - The WAIT instruction shuts off the clock in the central processor portion of the DSP56000/DSP56001.
  - The STOP instruction halts the internal oscillator.
  - Power increases linearly (approximately) with frequency; thus, reducing the clock frequency reduces power consumption.

### 1.3 MANUAL ORGANIZATION

This manual is intended to provide practical information to help the user:

- Understand the operation of the DSP56000 Family
- Interface the DSP56000 Family with additional memory
- Design parallel communication links
- Design serial communication links
- Code DSP algorithms
- Code communication routines
- Code data manipulation algorithms
- Locate additional support

The following list describes the contents of each section and each appendix:

#### Section 2. Architectural Overview and Bus Structure

This section describes each subsystem and the buses interconnecting the major components in the DSP56000/DSP56001.

#### Section 3. Memory

This section describes and differentiates the memory for the DSP56000 and DSP56001. It describes the program memories, data memories, and the operating mode register (OMR) bits controlling the memory maps.

#### Section 4. Data Arithmetic Logic Unit

This section describes in detail the data ALU (one of the three execution units comprising the central processor) and its programming model.

#### Section 5. Address Generation Unit

This section specifically describes the AGU (one of the three execution units comprising the central processor), its programming model, address indirect modes, and address modifiers.

#### Section 6. Program Control Unit

This section describes in detail the program control unit (one of the three execution units comprising the central processor) and its programming model.

#### Section 7. Instruction Set Introduction

This section presents a brief description of the syntax, instruction formats, operand/memory references, data organization, addressing modes, and instruction set. A detailed description of each instruction is given in **APPENDIX A INSTRUCTION SET DETAILS**.

#### Section 8. Processing States

This section describes the five processing states (normal, exception, reset, wait, and stop).

#### Section 9. Port A

The Port A section describes the external memory port, its control register, and its control signals.

#### Section 10. Port B

This section describes the port B parallel I/O, host interface, their registers, and the controls to enable/disable them.

#### Section 11. Port C

This section describes the port C parallel I/O, SCI, SSI, their registers, and the controls to enable/disable them.

#### Appendix A. Instruction Set Details

A detailed description of each DSP56000/DSP56001 instruction, its use, and its affect on the processor are presented.

#### Appendix B. Benchmarks

DSP56000/DSP56001 benchmark results are listed in this appendix.

#### Appendix C. Additional Support

This appendix presents a brief description of current support products and services and information on where to obtain them.

