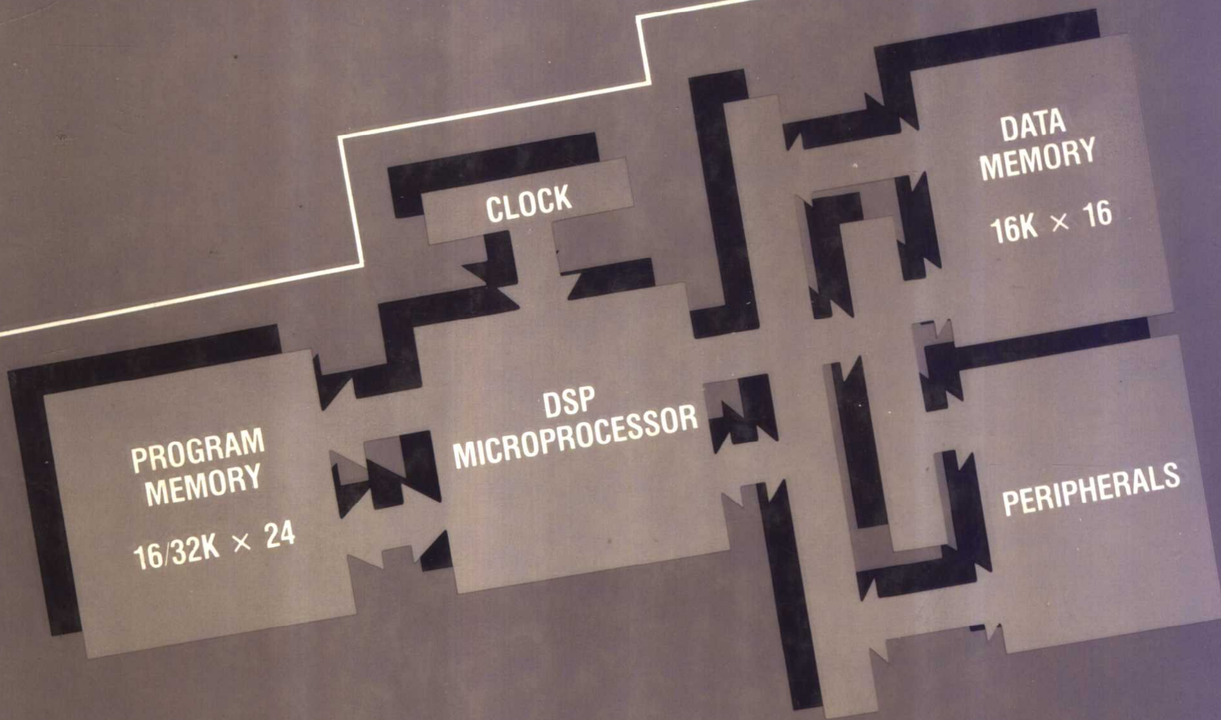
DIGITAL SIGNAL PROCESSING IN VLSI



by Richard J. Higgins

ANALOG DEVICES

DIGITAL SIGNAL PROCESSING IN VLSI

by
Richard J. Higgins
Georgia Institute of Technology

PRENTICE HALL, Englewood Cliffs, NJ 07632

Published by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

Information furnished by Analog Devices, Inc., is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices, Inc., for its use.

Analog Devices, Inc., makes no representation that the interconnection of its circuits as described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting of licenses to make, use, or sell equipment constructed in accordance therewith.

Specifications and prices are subject to change without notice.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-212887-X

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Preface

Do it digitally. These days, this precept goes far beyond personal computers. Inside practically any measuring instrument, communications or audio product, traditional electronic circuits are being emulated in software and firmware. The software drives a new generation of specialized microprocessors whose architecture is optimized to carry out digital signal-processing (DSP) algorithms, typified by the multiply-accumulate computation sequence of digital filters and spectral analysis.

With the gap between the analog and digital worlds bridged by economical analog-to-digital and digital-to-analog converters, the translation of signal processing functions to digital terms is further driven by advantages of stability, programmability, enhanced performance, as well as some digital signal processing algorithms and tricks that have no analog equivalent.

DSP computing processes and processors are having an impact similar to the effect the microprocessor had on computing. Talking calculators, smart measuring instruments, sophisticated music synthesizers, translating machines and seeing aids for the blind are coming into vogue, and even DSP-based smart toys (pioneered by TI's "Speak and Spell") are commonplace.

This book is for users and potential users of DSP who need to massage signals or create systems and are more interested in its applications than in theory (as represented by the classic DSP treatises).

A successful user of DSP techniques, I was driven to DSP by necessity as well as inclination. My lab measures the properties of electronic materials and microelectronic devices. Pulling information out of data by spectral analysis or real-time signal processing gave us a competitive edge in our research. In the mid-1970's I was introduced to digital filter concepts by a former student, Hal Alles, then at Bell Laboratories. Hal's group had constructed a high-performance digital filter and demonstrated it with real-time examples, including speech and music synthesis. While my lab had long been computer-massaging sampled data, I was impressed to see how well you could implement circuit functions by difference equations and solve them *in real time* with compact digital hardware.

This book is for a technical audience that needs to understand DSP algorithms and the special-purpose DSP hardware ICs and software tools developed to carry them out efficiently. Electrical engineers and scientists who are familiar with signals and their manipulation through analog means or through traditional digital computer analysis need to learn about the new tools available through IC chips optimized for DSP algorithms. Computer engineers and computer scientists with good microprocessor background need to learn about real-world signals and signal processing basics in order to design with DSP chips without violating fundamental DSP laws. Readers who have been deterred from DSP by treatises that begin mathematically with the z-transform will find here an introduction to the basics by an intuitive approach, with illustrations based on familiar examples.

In combining theory and practice, the book is a user guide that seeks to follow the tradition established by the *Analog-Digital Conversion Handbook* and other useful handbooks published by Analog Devices. The book is organized in two halves: fundamental DSP basics (Chapters 1-4) and their application (Chapters 5-8). Although the fundamentals half has been subjected to the scrutiny of several DSP experts, our approach is heuristic, with key ideas and rules summarized for rapid use, often without full proofs. The applications half is more motivational than exhaustive, with chip design, software issues, and applications illustrated by selected examples to point the reader to the potential for his or her own application.

Here is a list of the chapters and a brief description of their contents, to establish a perspective on the book's flow:

Chapter 1, Real-World Signal Processing: An introductory survey of digital signal processing, to fill in missing background and provide motivational examples. Builds bridges for the practicing EE via DSP implementations of analog circuit functions, and for the computer expert via software implementation of simple digital filter functions.

Chapter 2, Sampled Signals and Systems: Convolution theorem in Fourier transforms; information theorem and spectral limits in sampled data; the discrete Fourier transform; poles, zeros, and system stability; the z-transform. Provides the equivalent of the first course in DSP for readers without that background, and a ready-reference summary for others.

Chapter 3, The Discrete Fourier Transform and the Fast Fourier-Transform Algorithm: Relation to the continuous transform and to Fourier series; DFT limits (samples, window size); "fast Fourier transform" (FFT) algorithms and speed enhancement; pipelining and parallelism enhancements; avoiding DFT pitfalls: aliasing, leakage, picket fence effect; critical comparison of window weighting functions; spectral analysis examples including decimation (zoom), correlation, convolution; Fourier filtering.

Chapter 4, Digital Filters: Comparison of digital filter types; finite impulse-response (FIR) filter background and design methods; finite wordlength errors; infinite impulse-response (IIR) filter background and design methods; mapping analog to digital designs; CAD design methods for both FIR and IIR filter types.

Chapter 5, The Bridge to VLSI: Tradeoffs between speed, accuracy, and costs; parallelism and pipelining; quantization error, with special emphasis on FFTs and digital filter output error; analog input and output requirements (dynamic range, throughput).

Chapter 6, Real Hardware: Arithmetic-logic units; barrel shifters; address generators; instruction sequencers); system alternatives (single- and multiple-chip processors; bit- and Word-Slice configurations); comparison of alternatives in terms of performance, as well as cost of hardware and software development. Illustration with present day systems (Texas Instruments TMS-320 series; Analog Devices 2100 series); comparison of fixed- and floating-point systems; microprogrammable systems.

Chapter 7, Software Development for the DSP System: Development tools for translating and debugging DSP source code; target system specification; the assembler; the linking loader; instruction-level simulation; circuit emulation. DSP microprocessor applications, illustrated via an FIR filter module and a speech recognition system. Brief comparative introduction of software development (the meta-assembler) for microprogrammed DSP systems.

Chapter 8, DSP Applications: Major elements of a DSP system, using digital transmit-receive (or modulate-detect) prototype; digital detection alternatives (signal averaging, correlation, coherent detection, spectral estimation); digital heterodyning, decimation, and interpolation. Real time detection (examples of matched filters, coherent detection or lock-in, spectrum analyzer, spectral analysis and estimation). Modeling in real time; examples from telecommunications and speech; comparison of digital encoding and data-compression alternatives. Real-time signal generation and music synthesis: fast function generation; musical instrument analysis and synthesis methods; fm synthesizer example. Image processing: machine vision, image enhancement; medical image processing (examples from CAT, PET, and MRI image reconstruction)

ACKNOWLEDGMENTS

This book began during a two year stay in Boston (1983-85), where I divided my time between semiconductor research at the M.I.T. Francis Bitter National Magnet Laboratory and consulting at Analog Devices. I would like to thank the staff of the National Magnet Laboratory for their hospitality during this stimulating period.

The staff of the DSP Division of Analog Devices patiently bore with my many questions. In particular, I would single out Matt Johnson and Bob Fine, who gave generously of their advice, clear explanations, and feedback based on their contact as application engineers with real-world DSP users.

While any book is ultimately the author's responsibility, this book benefited greatly from a succession of technical commentaries at all stages of its development, particularly from Ted Dintersmith and Paul Toldalagi, both critical readers with an eye for clarity. Special credit is due to Analog Devices' super-editor Dan Sheingold, who worked over the author's sometimes turgid prose with an eye for crystal clarity based on his many years' experience in editing Analog Devices' superb technical publications. His contribution was enormous.

Richard J. Higgins

Atlanta, Georgia

Table of Contents

		Page
Contents		v
Preface		ix
1.	**Real-World Signal Processing**	
1.1	Introduction. Chapter overview; What is real-world DSP? MOS-VLSI signal processors with real-time capability	1
1.2	Review of Signals and Signal Processing. Enhancement of signal-to-noise; System models and the transfer function	6
1.3	Spectra—Time and Frequency as Complementary Domains. Time and frequency, two ways to study a system; System prototypes, the second-order system and the delay line; The spectrum and spectral fingerprints	8
1.4	Limitations of Analog Systems. Dynamic-range example, analog vs. digital audio; Analog filter tradeoffs, ideals and reality	16
1.5	Digital Signal Processing. Flexibility, key advantage to DSP; DSP issues and terminology; Sampled data; Throughput expansion, data compression, and pipelining	20
1.6	Non-Recursive Filters—The Moving Average as an Example. Finite impulse-response filters; Digital filters	29
1.7	Recursive Filters From Their Analog Counterparts. Analog feedback filters and their recursive digital counterparts; Introducing recursive filters; The digital filter in block-diagram form; Comparing software- and hardware-multiplier handling of the 1st-order low-pass; How do difference equations relate to the analog system? Impulse response as a probe of system dynamics	34
1.8	Digital Audio Example	47

2. Sampled Signals and Systems

2.1 Introduction. Building blocks of a sampled-data system . . . 53
2.2 Review of Fourier Series and Fourier Transforms. Fourier-series analysis; The Fourier transform and its properties . . 54
2.3 The Convolution Theorem. Fourier transform of a product; Properties of convolutions; The convolution theorem in Fourier transforms, Related signal-spectrum theorems: Parseval, correlation, auto correlation 63
2.4 Limits to Information. Tone bursts and frequency uncertainty; The uncertainty principle of information 72
2.5 Sampled Data. Sampling, aliasing, and anti-aliasing filters . 74
2.6 Fourier Transform of Sampled Data. Special functions in sampled data; Spectral properties of sampled data; Summary of spectral features of sampled data; Example: sinusoidal signal generator 79
2.7 Laplace and z-Transforms. Review of analog filter concepts; Stability facts for analog systems; Poles, spectral response, and filter behavior; The z-transform (equivalent of Fourier/ Laplace transform for sampled data) 88
2.8 Simple Digital Filters. Finite impulse-response (FIR) filters; Numerical analysis filters; Infinite impulse-response (IIR) filters; Recursive filters in numerical analysis 99

3. The Discrete Fourier Transform and the Fast Fourier-Transform Algorithm

3.1 Introduction 117
3.2 The Discrete Fourier Transform 118
3.3 Fast Fourier-Transform Algorithms. Multiply-time bottleneck of conventional DFT; Real-time analysis of speech; The FFT algorithm; Signal flow graphs, decimation-in-time FFT, and bit reversal; Decimation-in-frequency FFT and the inverse FFT; Butterflies for complex signals and spectra; Real-valued signals; FFT program example 121
3.4 Windows and Window Weighting. Why window weighting? Gaussian window; Cosine bell window; The Hamming and Blackman windows; Kaiser-Bessel window; Window weighting summary 139
3.5 Spectral Analysis Example 152
3.6 FFT Applications: Decimation, Convolution, Deconvolution. Spectral consequences of decimation and interpolation; FFTs for rapid convolution; Deconvolution for image enhancement; Fourier filtering and inverse FFT fidelity; Transformations into the spectral domain: multicomponent-exponential example 153

4. Digital Filters

4.1 Digital Filter Overview. Digital filters. when, why, what, how? Comparison of digital filter types; Summary of key digital filter relationships . . . 173

4.2 Finite-Impulse-Response Filters. FIR filter concepts and properties; The Fourier-series approach to FIR filters; The window method of FIR filter design; Computer-aided optimization for FIR filter design; Filter simulation and finite word-length effects; Hardware implementation of FIR filters . . . 182

4.3 Infinite-Impulse-Response Filters. The second-order section as a prototype; Biquads for special purposes; Higher-order filter synthesis; Hardware implementation of the biquad; Mapping analog design to digital design: impulse-invariant and other common transformations; Review of high-performance analog filters; CAD approach to high-performance IIR filters . . . 207

5. The Bridge to VLSI

5.1 Chapter Overview and Issues. Introduction; Computing-speed-performance evaluation; First words about accuracy. 235

5.2 Some VLSI-DSP Design Philosophy. A philosophical framework for VLSI-DSP; Desirable features of a DSP processor; DSP chip families defined . . . 246

5.3 DSP Architecture Issues: Tradeoffs, Pipelining, and Parallelism. On-chip/off-chip tradeoffs and processor performance; Data-flow bottlenecks and their resolution . . . 256

5.4 Finite-Wordlength Arithmetic. Introduction; Arithmetic error sensitivity; Overflow, underflow, and rounding; Filter quantization-error tradeoffs in fixed-point arithmetic; Accuracy in FFT spectral analysis. . . . 272

5.5 Analog I/O Methods: Dynamic-Range and Throughput Tradeoffs. Overview of analog I/O requirements and methods; I/O dynamic range and throughput requirements; Expanded I/O dynamic range with minimum speed loss; Nonlinear I/O digitization: companding and the LOGDAC. 293

6. Real DSP Hardware

6.1 Introduction; Chapter Overview . . . 307

6.2 Key DSP Hardware Elements. Multipliers; Multiplier-accumulators (MAC); Arithmetic logic units (ALU); Shifters: scaling control; Data-address generator; Program sequencer; Memory issues . . . 309

6.3 System Selection. DSP system alternatives; Microcoded systems; Single-chip DSP microprocessor survey........... 352
6.4 DSP Microprocessor: ADSP-2100. Introduction; System configuration; Microprocessor architecture; Instruction set. 368

7. Software Development for the DSP System

7.1 Introduction.................................. 401
7.2 Software Development Tools for the DSP Microprocessor. Development system philosophy and functions; Target system specification phase; Assembler; Linker; Simulator; Prom splitter; Target emulation; DSP system example: real-time isolated-word recognizer 402
7.3 Microcoded System Software Development. Differences between GP and Word-Slice software development; Meta assemblers: from user-defined mnemonics to microcode bits.................................. 437

8. DSP Applications

8.1 Introduction.................................. 447
8.2 Major Elements of a DSP System. The digital transceiver; Digital detection; Digital heterodyning, decimation, and interpolation.................................. 449
8.3 Real-Time Detection. Examples based on correlation principles; Examples based on *a priori* knowledge of the underlying system; Coherent detection or lock-in techniques; Spectrum analyzers; Spectral analysis and estimation.................................. 468
8.4 Modelling in Real Time: Telecommunications and Speech. Why modelling; Telecommunications; Coding of speech .. 508
8.5 Real-Time Signal Generation and Music Synthesis. Introduction and section overview; Fast function generation; What makes music? Analysis of musical sounds; How digital music synthesis is done; Parametric (FM) synthesizer design.................................. 524
8.6 Image Processing. Introduction to image processing; Machine vision: acquisition, enhancement, recognition; Medical image-processing 555

Bibliography xv
Index.................................. xxvii

Chapter One

Real-World Signal Processing

1.1 INTRODUCTION

1.1.1 CHAPTER OVERVIEW

Digital signal-processing (DSP) technology is moving inexorably into measurement, control, graphics, data analysis, and data communications. It is extending the computational capabilities of microcomputers to equal or exceed those of much larger machines, and helping larger machines to perform hitherto unheard of prodigies of number crunching. Finally, DSP circuitry both supplements and supplants analog circuits in signal-conditioning roles.

This chapter, an introductory survey of digital signal processing, is intended to get all our readers on somewhat the same footing. We anticipate that our readers will have diverse backgrounds. Some will have strong analog background but little experience with sampled data; we hope to put at rest the concern that more is lost than gained in sampling or amplitude quantization. Some will have strong digital hardware background but little analog experience; we will introduce you to the principles that underlie filter system behavior. Some with a computer-science background may have little experience with signal-processing hardware or with signals; we will show you what computers, augmented by signal processing technology, can accomplish in the real world. Many other readers already know some DSP from having used the techniques with minicomputers in data acquisition, but may want to extend their knowledge and capabilities to real-time applications of DSP with specialized microprocessors and components.

No previous DSP experience is assumed, except for some general EE circuit background and a firm resolve not to panic at the prospect of brief exposure to mathematical tools necessary for description—even if not for analysis. This introduction, light on the formal math, will emphasize the results you will be able to obtain.

There are five main topics to be covered in this chapter:

1. What is "real-world DSP", and what are typical applications?
2. What is to be gained in going digital? What are the tradeoffs?
3. DSP programmability = flexibility, compared to analog solutions.
4. How to estimate the tradeoffs between hardware and software solutions.
5. VLSI gives new freedom: custom-tailored architecture.

These themes are introduced through spectral analysis, which places time and frequency on equal footing as equivalent representations of information; and through commonly used filters, which enhance signal-to-noise. The benefits are clear: DSP promises flexibility, programmability, multiplexing, performance which is predictable, accurate, repeatable, and adaptable in real time. New DSP issues are understandable: sampling and aliasing; quantization; throughput limitations; program-development effort replacing circuit development. Two familiar examples are brought into digital form: the moving-average filter leads to the finite impulse response (FIR) concept; the first-order low-pass introduces digital feedback or infinite impulse response (IIR). Digital audio applications will be surveyed here and recur throughout the book, since they illustrate DSP computations in an undeniably real-time environment.

This chapter, a survey of possibilities, introduces ideas to be met later in more depth. Formal mathematics is minimized, though we employ some key tools: transfer functions and filter coefficients. Where possible, the analog circuit counterparts will be shown, and typical DSP hardware and software will be introduced.

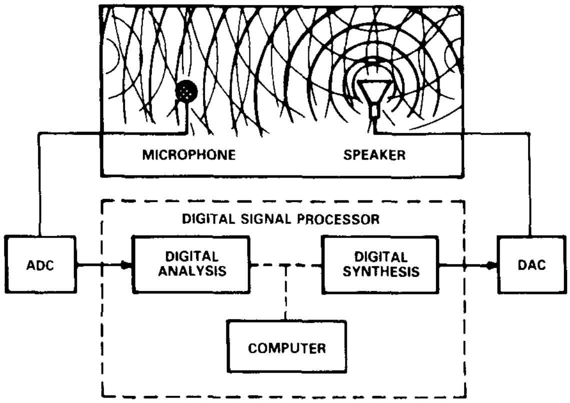
The chapter begins with a review of signals and analog signal processing at (1.2). Spectral analysis is introduced (1.3) by familiar examples, e.g., vibration and resonance. (1.4) What are the limitations of analog signal processing? (1.5) How well can we approximate signals when they are sampled? We survey sampling ideas such as bit rate and quantization. (1.6) The first digital filter is one you already know—the moving average. We develop the moving average in time and also in the frequency domain, as a function of number of "taps" or averaged points. (1.7) The familiar analog first-order low-pass filter will introduce recursive systems (i.e., feedback). This will be compared with a nonrecursive approximation, to show how the number of taps needed to simulate the long-time response can be reduced. While mathematically more complicated, recursive systems are fast and good for adaptive simulation, for example modeling of the vocal tract. (1.8) An in-depth example, digital audio, will leave the reader (we hope) ready to dive into the more challenging DSP background that follows.

1.1.2 WHAT IS REAL-WORLD DSP?

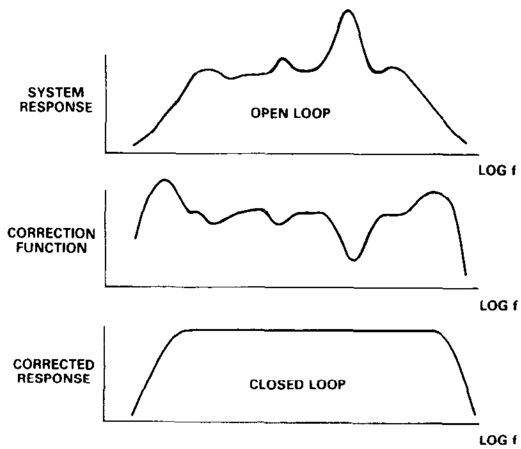
The term usually implies:

1. Contact with analog (continuous) signals as inputs or outputs. Real-world DSP in its simplest form is analog signal processing done digitally.
2. Real-time signal processing as opposed to off-line. The computation keeps pace with the input and output signals. The off-line processing of seismic data from magnetic tape, for example, is not real-time; but filtering an audio signal while it is happening *is* real-time.

An example of a real-world DSP system is shown in Figure 1.1a. Signals are translated from the analog domain to the digital domain via an analog-to-digital



a. A real-world digital signal processing system collects data from the analog domain via an a/d converter and provides it to the analog domain via a d/a converter. This example includes a loudspeaker and microphone as transducers for studying room acoustics in the time and frequency domains.



b. Closed-loop active enhancement of imperfect components (speaker, walls of room) in a digital feedback loop can improve the response curve.

Figure 1.1. Example of digital signal processing.

converter (A/D, ADC), and returned to the analog domain via a digital-to-analog converter (D/A, DAC). Computations embracing both analysis and synthesis are done in binary form in digital hardware and software.

The same system hardware can perform quite diverse applications. For example, with a computer-generated signal producing stimuli via the DAC-plus-speaker and a calibrated microphone-plus-ADC reading the response, the speaker characteristics can be measured; with changes to the program, the acoustics of the room can be characterized. A further program change—closing the feedback loop in software—can optimize the filtering by equalization, as shown in Figure 1.1b, to compensate dynamically for limitations in the speaker or in the room acoustics.

Perhaps 75% of signal processing tasks fall into three categories: *convolution*, *correlation*, and *transformation*.

- *Convolution* filters; it enhances signal-to-noise by selecting frequency bands to pass or suppress.
- *Correlation* compares, suppressing random events and amplifying repeated ones.
- *Transformation* finds frequency content, the characteristic pitch and harmonic-content "fingerprint".

1.1.3 MOS-VLSI SIGNAL PROCESSORS WITH REAL-TIME CAPABILITY

Inexpensive programmable digital signal-processing chips now have the speed and accuracy adequate to process signals in real time. They range from dedicated devices, such as multipliers and multiplier/accumulators, to families of components that are used to form high-performance DSP systems (for example, AMD's Bit-Slice™ and Analog Devices Word-Slice®, to single-chip processors. Examples of single-chip processors available from major manufacturers include: Analog Devices ADSP-2100, Nippon Electric Co. NEC 7720, and Texas Instruments TMS 320 family.

Some of these are in NMOS technology, with power dissipation measurable in watts. However, CMOS has recently evolved to provide competitive speeds with one-tenth as much power dissipation. These chips are quite complex, a characteristic of VLSI (very-large-scale integration). VLSI gives device designers the freedom to custom-design elaborate chip architectures, nevertheless—in practice—the internal complexity that it leads to (apparent in the photomicrograph of an example of such a chip shown in Figure 1.2a) is of no more concern to typical users than the complexity of their own brains or cell structure.

The development of such compact signal processors has significance comparable to the introduction of microcomputers: computing was no longer solely the domain of computer specialists. However, before we can make best

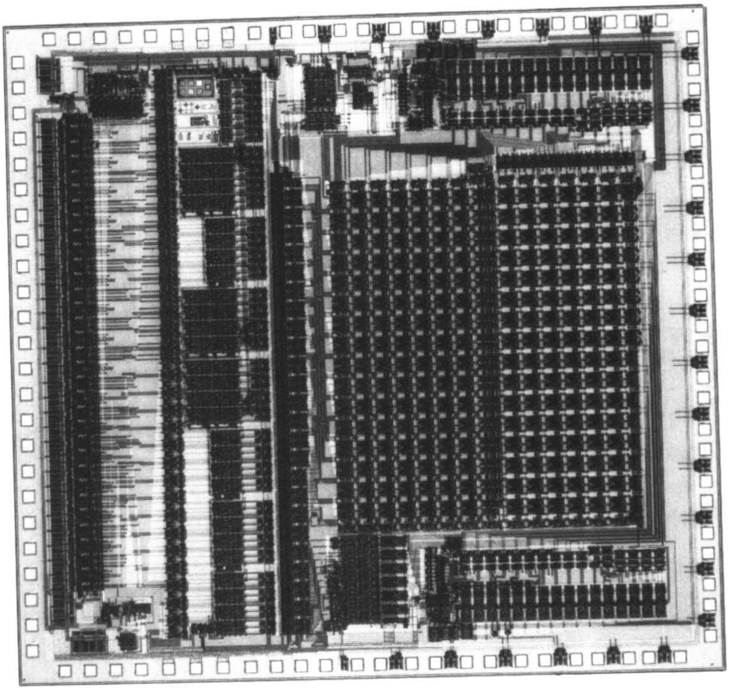


Figure 1.2. VLSI digital signal processing chip. This example is a floating-point CMOS multiplier, capable of executing a full 32 by 32-bit multiply in only 100 ns. The chip can operate programmably in various formats, from fixed point to single- or double-precision IEEE-standard floating point. More-recent members of the family have speeds up to 40 MFLOPS (millions of floating-point operations per second). (Analog Devices)

use of them, we must first understand the fundamentals of digital signal processing—the purpose of this book. Inexpensive, easily available DSP processors can contribute to numerous applications in a variety of fields, for example:

- Instrumentation and Measurement: Spectrum analyzers, Correlators, Signal averagers, Coherent detectors, Filters
- Communications: Voice telecommunications, Data communications, Modems, Encryption
- Digital audio: Speech and music generation, Signal generation, Speech recognition
- Graphics: Computer-graphic art, films, and CAD; Image enhancement and reconstruction, Medical imaging
- Navigation: Radar, Sonar
- Control: Robotics, Machine vision, Guidance, Decision-making
- Seismic investigations
- Computation: Microcomputer and mainframe accelerators; Workstations for science and engineering; Array processors

While these applications have been developing for some years with minicomputers or fixed-purpose hardware, cost and throughput limitations have limited the number of users. Consider digital speech recognition. This application combines digital filters and spectrum analyzers with phoneme recognition algorithms. Putting all this on an integrated circuit chip can bring speech recognition out of the expensive mainframe and into the world—for example, a door-lock which recognizes when you speak that it is you who wants to come in.

1.2 REVIEW OF SIGNALS AND SIGNAL PROCESSING

This section uses the familiar example of analog filters, which selectively enhance signal frequencies and attenuate noise, to introduce some of the nomenclature of signal processing.

1.2.1 ENHANCEMENT OF SIGNAL TO NOISE

Why use filters? Filters are typically used to pick out signals of interest from noise, by making use of their differing frequency characteristics. In Figure 1.3a, if the data consists of a rapidly varying waveform, biased by a slowly varying background and swamped by a low-frequency oscillation, the signal can be passed through a high-pass filter, with the result shown in the bottom trace of Figure 1.3b. On the other hand, if the data is the slow variation being swamped by high frequency noise, pass the signal through a low-pass filter to get the upper trace.

1.2.2 SYSTEM MODELS AND THE TRANSFER FUNCTION

A model of the "system"—in Figure 1.1, it is the room, and perhaps also the speaker and mike—is the starting point for characterizing its behavior. A system's response to a varying input signal, $s(t)$, with a frequency spectrum $S(f)$, can be described, essentially interchangeably, by the response $r(t)$ in the time domain (as a time history) or $R(f)$ in the frequency domain (as a frequency spectrum), as illustrated by Figure 1.4. Key system characteristics operate on the signal to produce the response to the stimulus. In the frequency domain, the operation can be expressed as a simple product; the ratio of response to stimulus is called the *transfer function*, $H(f)$.

$$H(f) = \frac{R(f)}{S(f)} \tag{1.1}$$

We will use lower-case letters for time-domain properties and upper-case letters for the frequency domain; the two domains are intimately connected via the Fourier transform (Chapter 2). The box is a model of the system which must be a convincing, adequate, and self-consistent description of system behavior. We observe what comes out (response) in response to what goes in (stimulus), without looking inside.