14

VLSI in Digital Signal Processing

David J. Beebe

The hardware used to implement the digital techniques discussed previously is the main focus of this chapter. We will first discuss digital signal processors (DSPs) and the functions they must perform. Next, two commercially available DSPs are described. Current high-performance VLSI architectures for signal processing are introduced including parallel processing, bit-serial processing, systolic arrays, and wavefront arrays. A portable ECG and a digital hearing aid are used as examples of VLSI applications in medicine. Finally, the emerging integration of VLSI and biomedical sensors is briefly discussed.

14.1 DIGITAL SIGNAL PROCESSORS

Until about 25 years ago, most signal processing was performed using specialized analog processors. As digital systems became available and digital processing algorithms could be implemented, the digital processing of signals became more widespread. Initially, digital signal processing was performed on general-purpose microprocessors such as the Intel 8088. While this certainly allowed for more sophisticated signal analysis, it was quite slow and was not useful for real-time applications. A more specialized design was needed.

14.1.1 Processor requirements and elements

Digital signal processors are really just specialized microprocessors. Microprocessors are typically built to be used for a wide range of general-purpose applications. In addition, microprocessors normally run large blocks of software, such as operating systems, and usually are not used for real-time computation.

A digital signal processor, on the other hand, is designed to perform a fairly limited number of functions, but at very high speeds. The digital signal processor must be capable of performing the computations necessary to carry out the techniques described in previous chapters. These include transformation to the frequency

domain, averaging, and a variety of filtering techniques. In order to perform these operations, a typical digital signal processor would include the following elements:

1. Control processor
2. Arithmetic processor
3. Data memory
4. Timing control
5. Systems

In early digital signal processing systems, the implementation of the elements shown schematically in Figure 14.1 involved many chips or ICs (integrated circuits). Today all the elements can be realized on a single VLSI chip.

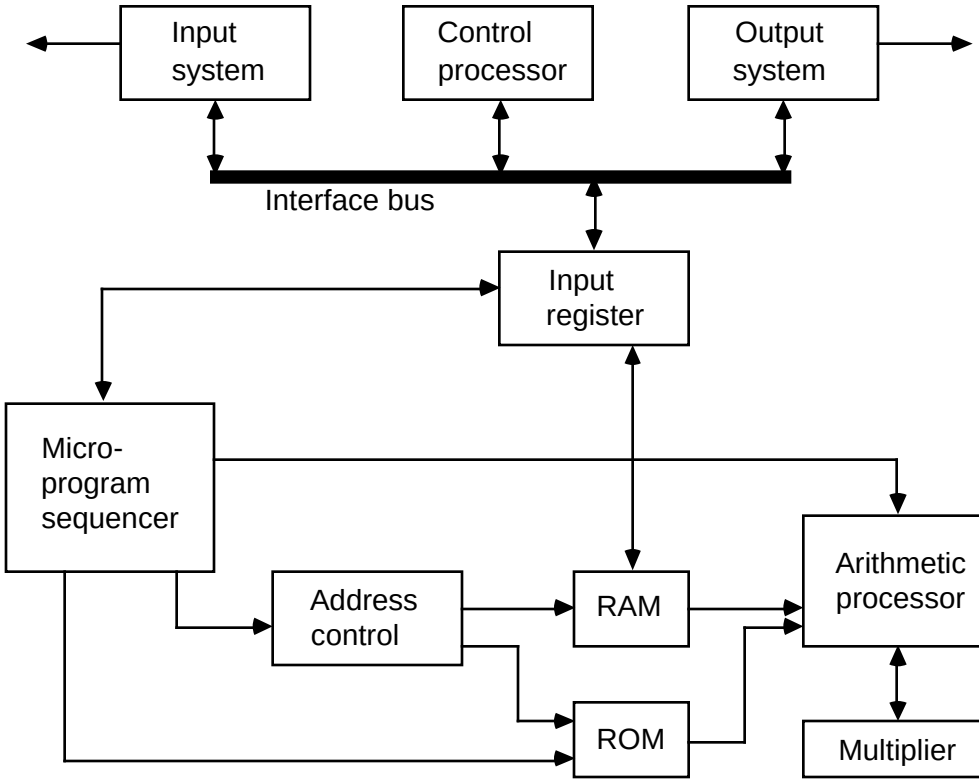

Figure 14.1 A typical digital signal processor modular design.

14.1.2 Single-chip digital signal processors

Some of the earliest attempts to incorporate the elements shown in Figure 14.1 on a single chip took place in the late 1970s. By the early 1980s, several companies including Bell Labs, Texas Instruments, NEC, and Intel had commercially available single-chip digital signal processors. In the last few years, the use of VLSI have made DSPs easier to use and more affordable.

Comparing the performance of DSPs is not always a straightforward procedure. While MIPS (million instructions per second) or MFlops (million floating-point operations per second) are often used when comparing microprocessor speed, this is not well suited to DSPs. A common benchmark for comparing the performance of DSPs is the multiply and accumulate (MAC) time. The MAC time generally reflects the maximum rate at which instructions involving both multiplication and accumulation can be issued. More meaningful benchmarks would be computations such as FFTs and digital filters. However, comparing these is tricky, because the benchmarks are not always completely described and often do not match those of the competition (Lee, 1988). Figure 14.2 shows the MAC times for some common DSPs.

The following sections describe two DSPs. First we discuss the TMS320 family made by Texas Instruments. This has been the most widely used DSP family. Second we describe the DSP56001 by Motorola. For a rough comparison of speed, consider that a Motorola 68000 microprocessor can handle 270,000 multiplications per second, while the DSP56001 is capable of 10,000,000 multiplications per second (Mo, 1991). That is an increase in speed of 37 times.

Company	Part	Date	MAC (ns)	Bits in mult.
AT&T	DSP1	1979	800	16
Texas Inst.	TMS32010	1982	390	16
Fujitsu	MB8764	1983	100	16
NEC	μPD77220	1986	100	24
Motorola	DSP56001	1987	74	24
AT&T	DSP16A	1988	33	16
Texas Inst.	TMS320C30	1988	60	24
Motorola	DSP96001	1989	75	32

Figure 14.2 Comparison of MAC times for several popular DSPs (adapted from Lee, 1988).

TMS320

TMS320 refers to a family of microprocessors introduced by Texas Instruments in 1982 and designed for application in real-time digital signal processing. The major feature of the TMS320 is a large on-chip array multiplier. In most general-purpose microprocessors, the multiplication is done in microcode. While this provides great versatility, it makes for long MAC times. An on-chip hardware multiplier greatly reduces the MAC times. On a typical microprocessor, approximately 10 percent of the chip area performs arithmetic functions, while on the TMS320, 35 percent of the chip area is dedicated to these functions. This large computing area is consistent with the numerically intensive nature of digital signal processing algorithms. The arithmetic and logic unit operates with 16/32-bit logic and the parallel hardware multiplication performs 16 × 16-bit two's complement multiplication in 200 ns. This high multiplication rate supports high-speed FFT com-

putations. Programming is done in assembly language. The maximum clock rate is 20 MHz (Quarmby, 1985; Yuen *et al.*, 1989).

DSP56001

The DSP56001 was introduced in 1987 and has quickly gained widespread use in audio equipment, scientific instrumentation, and other applications. The chip is capable of 10 million multiplications, 10 million additions, 20 million data movements, and 10 million loop operations per second. To achieve these high speeds, the chip has a highly parallel architecture with a hardware array multiplier, two ALUs (arithmetic logic units), two independent on-chip memory spaces, and an on-chip program memory. Most important, the DSP56001 utilizes a parallel and pipelined architecture in which several independent units operate simultaneously (Mo, 1991). The Motorola DSP is also unique in its ability to perform a MAC in just one cycle, with the result available by the next cycle. Its features include 512 words of on-chip program RAM, 24-bit data paths providing a dynamic range of 144 dB, a data ALU, address arithmetic units and program controller operating in parallel, and a MAC time of 74 ns (Lee, 1988; Motorola, 1988).

14.2 HIGH-PERFORMANCE VLSI SIGNAL PROCESSING

Only recently has it become feasible to perform digital signal processing in real time. This is because the implementation of digital processing techniques requires high levels of computational throughput, particularly for real-time applications. This demand combined with the continually increasing levels of performance of VLSI has led to the development of VLSI digital signal processors such as the TMS320 and the DSP56001 discussed above. The trend in DSP design is toward more algorithm-based architectures. In other words, the ease with which VLSI design can be done today leads the designer to more specialized architectures.

As discussed previously, the most useful digital signal processing techniques include FFT computing, FIR and IIR digital filters. The implementation of these techniques requires only three types of operations. The required operations are storage, multiplication, and addition. The small number of operations required suggest the use of a repetitive modular architecture. This is indeed the case. The limited number of different operations required and the way in which VLSI technology is fabricated have led to several VLSI-oriented special-purpose architectures for digital signal processing applications. These architectures use multiprocessing and parallel processing, array processors, reduced instruction set computer (RISC), and pipelining to achieve very high-speed processing rates.

The terminology used to describe and classify VLSI architectures is by no means standard. Various attempts have been made to establish a useful taxonomy, including Flynn's terminology based on instruction and data streams (Flynn, 1966). In the following discussion, we have tried to use the most commonly used terms. However, be aware that the literature is peppered with various terminology used to describe the similar architectures.

Parallel processing or multiprocessing uses multiple processors that cooperate to solve problems through concurrent execution. Pipelining is just an extension of multiprocessing that optimizes resource utilization and takes advantage of dependencies among computations (Fortes and Wah, 1987). Array processor typically refers to a two-dimensional array of processors that the data flows through. Pipelining is used to route the data through the array in the most efficient manner.

14.2.1 Parallel processing

The application of parallel processing to signal processing generally consist of two types (Yuen *et al.*, 1989). Figure 14.3 illustrates these architectures.

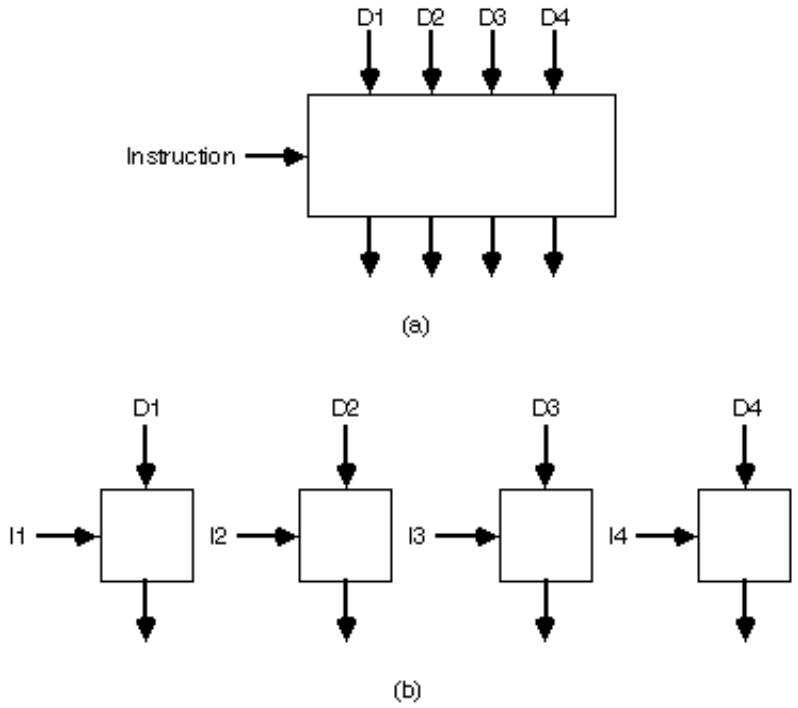

Figure 14.3 Classification of parallel processor architectures. (a) Single Instruction Multiple Data (SIMD), (b) Multiple Instruction Multiple Data (MIMD).

1. The single instruction multiple data (SIMD) operation in which the computational field consists of a number of data elements acted upon by a single operational instruction.

2. The multiple instruction multiple data (MIMD) operation in which a number of instruction streams act on multiple data elements. This method is used in image processing work.

Parallelism is achieved at the mathematical level by applying the residue number system (RNS) in architectural form (Szabo and Tanaka, 1967). In this system, the

computational field is decomposed into a set of independent subfields. Computations in these subfields are performed in parallel in a SIMD-like structure. RNS provides for high-speed mathematical operations since addition and subtraction have no interdigit carries or borrows and multiplication does not require the generation of partial products.

14.2.2 Bit-serial processing

In bit-serial processors, operations are performed on only 1 bit in each word at a time. Bit-serial organization is usually applied to many words at once. Hence, the term bit-slice or bit-column arose. Figure 14.4 shows a simple bit-slice scheme.

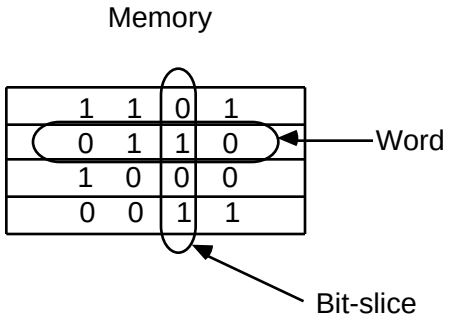

Figure 14.4 A simple bit-slice organization. The bit-slice of data is operated on, so one bit from each word is operated on simultaneously.

In bit-serial architecture, the digital signals are transmitted bit sequentially along single wires rather than simultaneously on a parallel bus (Denyer and Renshaw, 1985). This approach has several advantages over the parallel approach. First, communications within and between VLSI chips is more efficient. This is an important point given the communication-dominated operations involved in signal processing. Second, bit-serial architecture leads to an efficient pipelining structure at the bit level which leads to faster computations.

There are also several advantages to the bit-serial approach in terms of doing the actual VLSI chip layout. Bit-serial networks are easily routed on the chip since there is no need to make parallel connections to a bus. Also, since all signals enter and leave the chip via single pins, the number of input/output pins is reduced. Finally, bit-level pipelining distributes both memory and processing elements on the chip in a modular and regular fashion. This greatly facilitates ease of design, layout, and the application of silicon compilers (Yuen et al., 1989).

14.2.3 Systolic arrays

A systolic array takes a bit-serial architecture and applies pipelining principles in an array configuration. The array can be at the bit-level, at the word-level, or at both levels. The name systolic array arose from analogy with the pumped circula-

tion of the bloodstream. In the systolic operation, the data coefficients and other information are systematically fed into the system with the results “pumped out” for additional processing. A high degree of parallelism is obtained by pipelining data through multiple processors, typically in a two-dimensional fashion. Once data enters the array, it is passed to any processor that needs it without any stores to memory. Figure 14.5 illustrates this data flow. The data flow is synchronized via a global clock and explicit timing delays (Duncan, 1990).

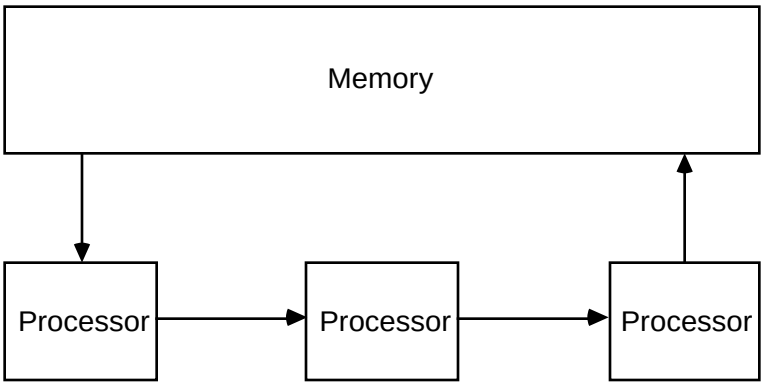

Figure 14.5 Systolic flow of data to and from memory.

Systolic arrays are particularly well suited to VLSI implementation. In order to take full advantage of the ever increasing density of VLSI, chip layouts must be simple, regular, and modular. Systolic arrays use simple processing elements and interconnection patterns that are replicated along one or two dimensions on the chip. In fact, most connections involve only nearest neighbor communication. The general architecture for a systolic array is shown in Figure 14.6. The chip resembles a grid in which each point is a processor and each line is the link between them.

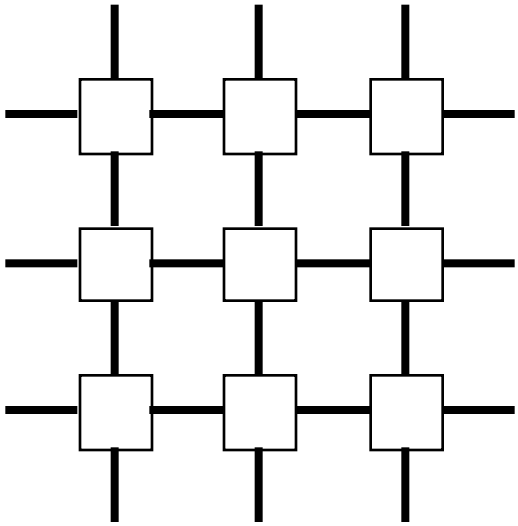

Figure 14.6 A simple systolic array configuration. Each box represents a processor or transputer and each line represents the link between the processors.

14.2.4 Wavefront arrays

One of the newest architectures is the wavefront array. Developed by Kung, the wavefront architecture is similar to the systolic array in that both are characterized by modular processors and regular, local interconnection networks. The difference is that in the wavefront array, the global clock and time delays are replaced by asynchronous handshaking. This eliminates problems of clock skew, fault tolerance, and peak power (Duncan, 1990).

The details of these architectures is beyond the scope of this text; the intent here is to give the reader an overview of the current capabilities and applications for VLSI digital signal processing (see Kung et al., 1985 and Kung, 1988 for details of VLSI architectures).

14.3 VLSI APPLICATIONS IN MEDICINE

VLSI devices are currently used in a wide variety of medical products ranging from magnetic resonance imaging systems to conventional electrocardiographs to Holter monitors to instruments for analyzing blood.

14.3.1 Portable ECG

The portable ECG machine was made possible largely due to progress in VLSI design. Today's portable ECG machines, such as the Elite (Siemens Burdick, Inc.), are fully functional 12-lead ECG machines equal in every respect to larger machines with the exception of paper size and mass storage capacity. A quick comparison points out the dramatic results that can be obtained with VLSI. The

portable ECG is 15 times smaller, 10 times lighter, and half as costly as the full-size machine. It consumes 90 percent less power and uses 60 percent fewer parts (Einspruch and Gold, 1989).

14.3.2 Digital hearing aid

A traditional analog hearing aid consists of the parallel connection of bandpass filters. To provide accurate compensation, a large number of filters are needed. Size, complexity, and cost all limit the number of filters. A hearing aid with a large number of adjustable components is also difficult to fit and adjust to the needs of each individual patient.

With the advent of general-purpose DSPs and application-specific integrated circuits (ASIC), it has become feasible to implement a hearing aid using digital technology. The digital implementation utilizes many of the ideas discussed previously including A/D converters, D/A converters, and digital filtering. The biggest advantage of the digital design is that the transfer function of the filter used to compensate for hearing loss is independent of the hardware. The compensation is performed in software and thus is extremely flexible and can be tailored to the individual. The digital hearing aid is more reliable and the fitting process can be totally automated (Mo, 1988).

14.4 VLSI SENSORS FOR BIOMEDICAL SIGNALS

A recent spinoff from standard microelectronics technology has been solid-state sensors and smart sensors. Integrated circuit processing techniques have been used for some time to fabricate a variety of sensing devices. Currently under development is a new generation of smart or VLSI sensors. These sensors combine the sensing element with signal processing circuitry on a single substrate. Figure 14.7 shows the elements of a generic VLSI sensor.

Integrated solid-state sensors or smart sensors contain four parts: (1) custom thin films for transduction, structural support or isolation, or encapsulation; (2) microstructures formed using micromachining techniques; (3) interface circuitry; and (4) signal processors (Einspruch and Gold, 1989). A sensor built by Najafi and Wise to measure the extracellular potentials generated within neurons illustrates the possibilities of combining VLSI with current sensor technologies.

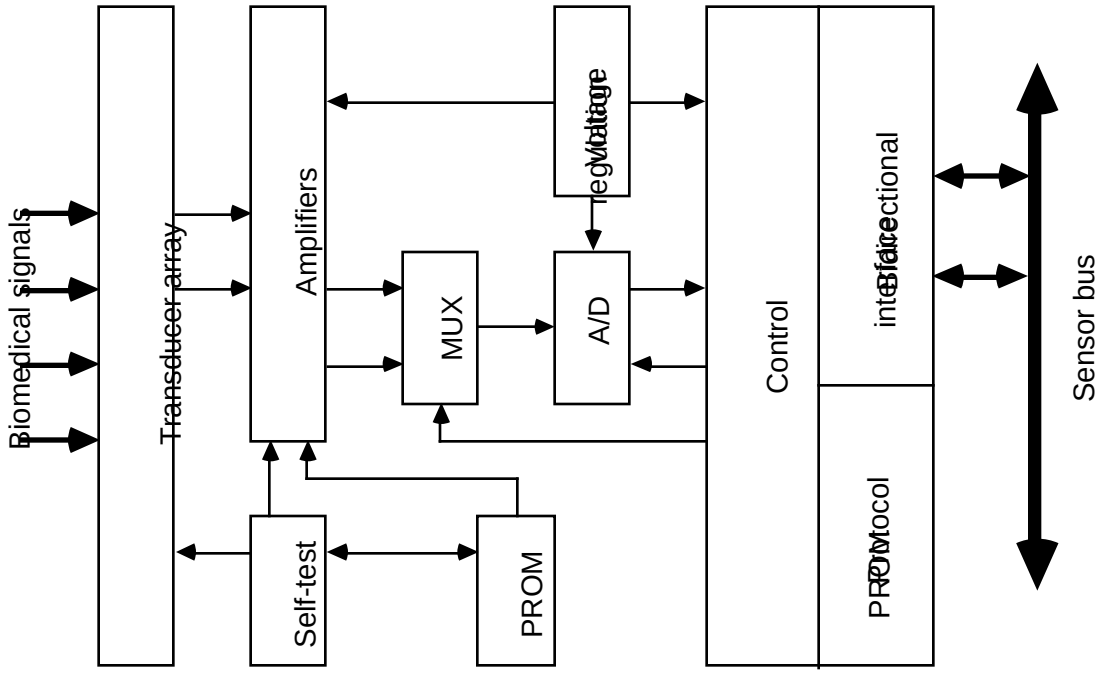

Figure 14.7 Block diagram of a generic VLSI sensor. The sensor is addressable, self-testing and provides a standard digital output (adapted from Einspruch and Gold, 1989).

Figure 14.8 shows the overall structure of the sensor. The electronics on the sensor includes functional circuits such as amplifiers, an analog multiplexer, a shift register, and a clock. Indeed most of the sampling functions discussed in Chapter 3 are integrated right onto the sensor. The sensor is 3–4 mm long, 200 μm wide and 30 μm thick.

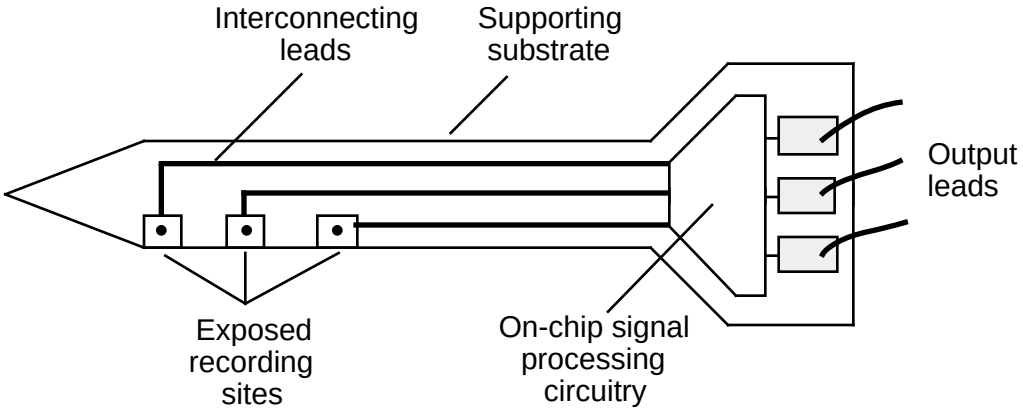

Figure 14.8 The overall structure of the smart sensor built by Najafi and Wise (adapted from Einspruch and Gold, 1989).

14.5 VLSI TOOLS

The most commonly used VLSI design tools in the academic environment are the Berkeley VLSI tools. The system allows the designer to lay out the design at the device level and simulate the design for correct operation. A typical design would begin with a functional block diagram. One next moves progressively lower and lower toward the actual device layout. The device level layout is done in MAGIC, a VLSI layout editor. MAGIC is analogous to a mechanical CAD program. The designer actually draws each and every transistor in the design just as it will be fabricated. Obviously this can be a painstaking task since a typical chip may contain several hundred thousand transistors. MAGIC contains many features that simplify the design and often portions of the layout are redundant. As portions of the design are completed, programs such as CRYSTAL (a VLSI timing analyzer) and ESIM (an event-driven switch level simulator) are used to test the logic and timing operation of that portion of the design. This is an important point! It is important to thoroughly test each portion of the layout as it is completed rather than waiting until the whole design is done. Continual testing will help ensure that the design will be functional when all the pieces are connected together in the final layout.

Once the entire layout is complete and it has been successfully simulated, it is fabricated. The MOSIS (MOS implementation system) fabrication facility at the University of Southern California is often used for low-volume experimental work. Finally, the device is tested and revised as necessary.

14.6 CHOICE OF CUSTOM, ASIC, OR OFF-THE-SHELF COMPONENTS

When approaching an instrument design, one must decide between simply using off-the-shelf chips, ASIC (application specific integrated circuits), or fully custom chips. In reality, the finished design will usually contain some combination of these approaches. Ten years ago, using off-the-shelf chips was the only option. However, in recent years it has become feasible for even small companies to design their own ASIC or fully custom chips via tools similar to those discussed above. The design choices are based on the needs of the particular project and the availability of suitable off-the-shelf chips. The design time and cost generally increase as one moves from a design containing only off-the-shelf chips to a fully custom design.

14.7 REFERENCES

Denyer, P. and Renshaw, D. 1985. *VLSI Signal Processing: A Bit-serial Approach*. Reading, MA: Addison-Wesley.

Duncan, R. 1990. A survey of parallel computer architectures. *Computer*, **23**(2): 5–16.

Einspruch, N. G. and Gold, R. D. 1989. *VLSI in Medicine.* San Diego: Academic Press.

Flynn, M. J. 1966. Very high speed computing systems. *Proc. IEEE*, **54**: 1901–09.
Fortes, J. A. B. and Wah, B. W. 1987. Systolic arrays—from concept to implementation. *Computer*, **20**(7) :12–17.
Kung, S. Y. 1988. *VLSI Array Processors.* Englewood Cliffs, NJ: Prentice Hall.
Kung, S. Y., Whitehouse, H. J., and Kailath, T. 1985. *VLSI and Modern Signal Processing.* Englewood Cliffs, NJ: Prentice Hall.
Lee, E. A. 1988. Programmable DSP architectures: Part 1. *IEEE ASSP Magazine*. 4–19.
Mo, F. 1991. Basic digital signal processing concepts for medical applications. *Biomedical Science and Technology*. **1**(1): 12–18.
Motorola. 1988. Technical Data #DSP56001/D, Motorola Literature Distribution, P.O. Box 20912, Phoenix, Arizona, 85036.
Quarmby, D. (ed.) 1985. *Signal Processor Chips.* Englewood Cliffs, NJ: Prentice Hall.
Szabo, N. S. and Tanaka, R. I. 1967. *Residue Arithmetic and Its Applications to Computer Technology*. New York: McGraw-Hill.
Yuen, C. K., Beauchamp, K.G., and Fraser, D. 1989. *Microprocessor Systems in Signal Processing*. San Diego: Academic Press.

14.8 STUDY QUESTIONS

14.1 Describe the difference between general-purpose microprocessors and DSPs.
14.2 Why are MAC times used as benchmarks for DSPs instead of the usual MIPS or MFlops?
14.3 Discuss why VLSI is well suited to the design of DSPs.
14.4 Define the following terms: (1) parallel processing, (2) pipelining, (3) array processors, (4) SIMD, (5) MIMD.
14.5 Describe the difference between systolic arrays and wavefront arrays. Which would operate more efficiently?
14.6 List at least four advantages of using VLSI-based designs for medical applications.