Definition of a Digital Signal Processor

A digital signal processor (DSP) is an integrated circuit designed for high-speed data manipulations, and is used in audio, communications, image manipulation, and other data-acquisition and data-control applications.

How Digital Signal Processing Works

To explain how digital signal processing works, you must understand the difference between analog and digital signals. Analog signals, which include sound intensity, pressure, light intensity, etc., are continuously variable. Each of our senses is sensitive to different kinds of analog signals. Our ears are sensitive to sound, our eyes are sensitive to light, and so on. Once we receive a signal, our sensory organs convert it to an electrical signal and send it to our analog computer (the brain). Our brains are very powerful parallel computers whose performance currently is unmatched by any digital computer. Our brains not only analyze the information received, but also make decisions using this data.

Digital signals are those that are transmitted within or between computers, in which information is represented by discrete states – for example, high voltages and low voltages – rather than by continuously variable levels in a continuous stream, as in an analog signal.
How Analog and Digital Signals Work Together

Digital technology such as personal computers (PCs), assist us in many ways: writing documents, spell checking, and drawing. Unfortunately, the world is analog, and electronic analog computers are not as versatile as digital computers. Therefore, in order to make use of the tremendous processing power that digital technology offers us, we must do the following:

- Convert the analog signals into electrical signals, using a transducer (such as a microphone, as shown in the diagram).
- Digitize these signals (i.e., convert them from analog to digital using an analog-to-digital converter (ADC)), as shown in the diagram.

Once the signal is in digital form, our computer can easily process it through a digital signal processor. The DSP specializes in processing these signals, which makes it slightly different from microcomputers, microcontrollers, or general-purpose microprocessors.

After the DSP has processed the signal, the output signal must be converted back to analog form so that we can sense it. This is the digital-to-analog (DAC) conversion stage in the diagram. A loudspeaker, for example, would reproduce analog signals coming from the DAC into sound.

So, we can see that to process the signal digitally, we need to convert it at least twice. Is it worth it? As you will see, it really is, at least until someone designs an analog computer as versatile as a digital one.
**Why Do We Need Digital Signal Processors?**

Why do we need a digital signal processor? Can we not use a general-purpose microprocessor to process signals as well? Let us try to answer this question by giving an example of some arithmetic operations performed by DSPs.

**Add and Subtract**

Add and subtract operations are performed quite simply by general-purpose microprocessors in a single or very few clock cycles. Digital addition is similar to decimal add. Our example shows adding 1 plus 2. The result is the decimal 3.

**Multiply and Divide**

The multiply and divide operations are more complex. A digital multiply operation consists of a series of *shift* and *add* operations. Our example shows a multiplication of 3 by 5. Division, which is more complex, will not be discussed here.

General-purpose microprocessors are quite slow in performing multiply and divide operations. They will typically sequentially execute a series of shift, add, and subtract operations from their microcode to perform a single multiply operation, and may consume many cycles to complete.

The DSP performs multiplication in a single cycle by implementing all shift and add operations in parallel. The circuitry is relatively complex and consumes a considerable
number of transistors. The benefit is very fast multiplication, which is required for processing most digital signals.

When general-purpose DSPs are not fast enough, the signal is either processed using analog circuits (which may have some drawbacks), or in specialized DSP hardware designed only for that task. This eliminates many of the benefits of a programmable DSP.

Digital signal processing, by its nature, requires many calculations of the form:

\[ A = B \times C + D \]

This may appear to be a simple task, but when speed is also required, we find that specialized, dedicated hardware to perform this task is very useful.

Multiply, Accumulate (MAC)

Most DSPs have a specialized instruction that allows them to multiply, add, and save the result in a single cycle. This instruction is usually called MAC (short for Multiply, Accumulate).

We have established that for DSPs, we need specialized hardware that is capable of performing multiply and accumulate functions in the shortest possible time (preferably in a
single cycle). However, the central problem remains. How can we achieve a fast multiply operation? Without a fast multiplier, a worthwhile DSP design would only be a dream.

Designing fast multipliers was one of the greatest challenges in digital design up until the 1980s. In the 1970s, several of the world’s leading research laboratories sought to make fast digital multipliers a reality.

Multiply Times

In 1971, Lincoln Laboratories designed a multiplier using 10,000 integrated circuits, performing the operation in just 600 ns. By the mid-1970s, multiply times of 200 ns were becoming commonplace. This made it possible to design acceptable digital signal processors. These early designs were expensive and bulky, but fast multiplication was determined to be possible.

In the early 1980s, single-chip DSPs with good performance started to appear, and ever since, multiply times have continued to drop. Today’s 16-bit fixed-point devices can achieve multiply times of 5 ns. Given the origins of this technology, this is a remarkable achievement.
Now let us have a closer look at the internal architecture of computers so we can see how this has affected the design of DSP chips.

**Stored Program Machines**

Computers need instructions to operate. At every clock cycle, they must be told what to do. If the instructions are stored, the computer just has to fetch and execute them. Such computers are called *stored program* machines. Our computer typically fetches an instruction and then data, operates on the data, and returns the resulting data to the store.

Stored program machines use two well-known and widely used computer architectures: von Neuman and Harvard. The following diagram shows the structure of the two architectures.

**von Neuman Architecture**

The von Neuman machines store programming and data in the same memory area. In this type of machine, an instruction contains the operation command and the address of the data on which the operation is performed. There are two basic operation units within these machines: the arithmetic logic unit (ALU) and the input/output unit. The ALU performs the core operations: multiply, add, subtract, and many more. It is on these very simple core
operations that complex software, such as word processing software, can be built. The input/output unit manages the flow of external data for the machine.

**Harvard Architecture**

The primary difference between Harvard architecture and von Neuman architecture is that with Harvard, program and data memories are physically separated transmission paths. This enables the machine to transfer instructions and data simultaneously. Such a structure can greatly enhance performance, because instructions and data can be fetched simultaneously. Harvard machines also have ALUs and input/output units.

**Von Neuman and Harvard Architecture History**

The history of these two architectures is very interesting. The Harvard architecture was developed by Howard Aiken in the late 1930s at Harvard University, with the Harvard Mark 1 becoming operational in 1944. The University of Pennsylvania followed in 1946 with the development of the Electronic Numerical Integrator and Calculator (ENIAC™).

John von Neuman, a Hungarian-born mathematician, suggested a simpler and lower cost architecture, namely a single memory for programming and data. This simple solution has set the standard ever since. In 1951, the Institute of Advanced Studies in Princeton built the first von Neuman machine.

**Which Architecture is Best Suited for DSP?**

Common general-purpose personal computers use processors designed with the von Neuman architecture while the Harvard architecture is more commonly used in specialized microprocessors for real-time and embedded applications.

DSPs typically use Harvard architecture, although von Neuman DSPs also exist. Many signal and image processing applications require fast, real-time machines. The drawback to using a true Harvard architecture is that since it uses separate program and data memories, it needs twice as many address and data pins on the chip and twice as much external memory. Unfortunately, as the number of pins or chips increases, so does the price.

Electronic designers, who have had to tackle problems like these before, have come up with an elegant solution: a single data and address bus is used externally, while two (or more) separate buses for program and data are used internally. Timing (multiplexing) handles the separation of program and data information. In one clock cycle, the program information flows on the pins, and in the second cycle, data follows on the same pins. Program and data information is then routed onto separate internal program and data buses. Such machines are called *modified Harvard architecture* processors because the internal architecture is Harvard while the external architecture is von Neuman. The performance of modified
Harvard architecture processors typically compares well with the performance of true Harvard architecture processors because most DSP chips also incorporate multiple internal RAM/ROM cells for high-use instructions and data. This significantly reduces the time used for external sequential program and data access associated with classic von Neuman processors.
Components of a Typical DSP System

Typical DSP systems consist of a DSP chip, memory, possibly an analog-to-digital converter (ADC), a digital-to-analog converter (DAC), and communication channels. Not all DSP systems have the same architecture with the same components. The selection of components in a DSP system depends on the application. For example, a sound system would probably require A/D and D/A converters, whereas an image processing system may not.

DSP Chip

A DSP chip can contain many hardware elements; some of the more common ones are listed below.

**Central Arithmetic Unit**
This part of the DSP performs major arithmetic functions such as multiplication and addition. It is the part that makes the DSP so fast in comparison with traditional processors.

**Auxiliary Arithmetic Unit**
DSPs frequently have an auxiliary arithmetic unit that performs pointer arithmetic, mathematical calculations, or logical operations in parallel with the main arithmetic unit.
Serial Ports
DSPs normally have internal serial ports for high-speed communication with other DSPs and data converters. These serial ports are directly connected to the internal buses to improve performance, to reduce external address decoding problems, and to reduce cost.

Memory
Memory holds information, data, and instructions for DSPs and is an essential part of any DSP system. Although DSPs are intelligent machines, they still need to be told what to do. Memory devices hold a series of instructions that tell the DSP which operations to perform on the data (i.e., information). In many cases, the DSP reads some data, operates on it, and writes it back. Almost all DSP systems have some type of memory device, whether it is on-chip memory or off-chip memory; however, on-chip memory operates faster.

A/D and D/A Converters
Converters provide the translator function for the DSP. Since the DSP can only operate on digital data, analog signals from the outside world must be converted to digital signals. When the DSP provides an output, it may need to be converted back to an analog signal to be perceived by the outside world.

Analog-to-digital converters (ADCs) accept analog input and turn it into digital data that consist of only 0s and 1s. Digital-to-analog converters (DACs) perform the reverse process; they accept digital data and convert it to a continuous analog signal.

Ports
Communication ports are necessary for a DSP system. Raw information is received and processed; then that information is transmitted to the outside world through these ports. For example, a DSP system could output information to a printer through a port. The most common ports are serial and parallel ports. A serial port accepts a serial (single) stream of data and converts it to the processor format. When the processor wishes to output serial data, the port accepts processor data and converts it to a serial stream (e.g., modem connections on PCs). A parallel port does the same job, except the output and input are in parallel (simultaneous) format. The most common example of a parallel port is a printer port on a PC.
Practical Applications for DSP Systems

Since their introduction to the market, DSPs have found a wide variety of applications. They are used in everyday hi-fi systems as well as high-end virtual-reality applications. Generally, DSP is not an expensive technology. Some practical DSP systems are:

- Hi-Fi Equipment
- Toys
- Videophones
- Modems
- Phone Systems
- 3D Graphics Systems
- Image Processing Systems

Hi-Fi Equipment (Music Systems)

DSPs are now being used in sound processors that can create the illusion of three-dimensional sound or modify the acoustics of a room to give the illusion of very large rooms and auditoriums. The result is movie theater quality sound in a home music system.

Toys

Today, DSP technology is integrated in children's toys. Talking toys are commonplace; by pressing the picture of a dog, children can hear it bark. They can also learn their alphabet by singing along with a teaching toy. This clearly demonstrates that DSP technology is not expensive.

Videophones

Videophones will affect the lives of people from all walks of life. They are quickly improving in quality. It is only a matter of time before prices drop and videophones become widely used. DSPs are used for compression and decompression of images in videophones. There are several international standards for compressing moving images. Programmable DSPs are the perfect answer to evolving standards since this may only require a software update.
Modems

As the Internet continues to grow, so has the use of modems. To be able to handle the ever-increasing communications load, modems have become faster and more efficient. DSPs perform vital functions in modems such as modulating the digital bit stream into a signal compatible with a phone line, canceling line echoes, and compressing and decompressing data.

Phone Systems

These days, it is quite common to call a company and be answered by a machine that provides alternatives such as: “Say 1 for sales,” “Say 2 for technical support,” and so on. These phone systems use DSPs to perform the function of voice recognition. DSPs are also commonly used in the communications industry for the add-on features you can get from your telephone company like caller ID, voice messaging, and call back.

3D Graphics Systems

Most flight simulators use 3D real-time graphics to enhance realism. To calculate the necessary details in three dimensions (and to be able to do this 30 times every second) requires very efficient and powerful processors. DSPs are now widely used in virtual-reality applications.

Image Processing Systems

Personal handheld digital cameras are also now becoming widespread. DSPs are used to perform the conversion of charge-coupled device (CCD) chip analog voltages (video) to compressed data, which is then stored digitally in constant storage EEPROM (electrically erasable ROM). The DSP also senses the buttons, controls exposure times, provides the CCD gate timing, and downloads images to the PC.

DSPs are also used extensively in image processing, such as robot vision, machine vision, image compression, and fingerprint recognition. A simple example of an image-processing application is the inspection of printed circuit boards. The system works by recording the image of a working board and comparing (subtracting) it to newly manufactured ones as they pass beneath a CCD camera. These systems also use the efficient multiply and add cycles in DSPs to perform two-dimensional filtering.