Digital Signal Processing Hardware and Software

Six Ways to Implement DSP

All DSP involves multiplication and addition in some combination and sequence. The six common ways to implement those calculations are:

Personal computers

Embedded controllers

Enhanced embedded processors

Embedded logic

PLDs and ASICs

Programmable DSP chips

We will discuss these six methods in this section.

Personal Computer DSP

Literally, any computer can do DSP calculations. Because of the widespread availability of personal computers (PCs) as well as their exceptional performance, they can be used in many DSP applications. As long as the signal frequencies and sampling rates are slow enough, a PC can easily do all the processing needed. To implement any filter, FFT applications, or other use, all you do is write the necessary software and run it on the PC. With clock speeds of 1 to 3 GHz today, most PCs are quite adept at DSP. One good example is the digital audio output generated on a PC. All of it is done with software DSP.

On the other hand, PCs are often overkill for many applications. Simpler, cheaper, and smaller hardware platforms are available. Furthermore, PCs are not fast enough for real time processing in many applications.

Embedded Controllers

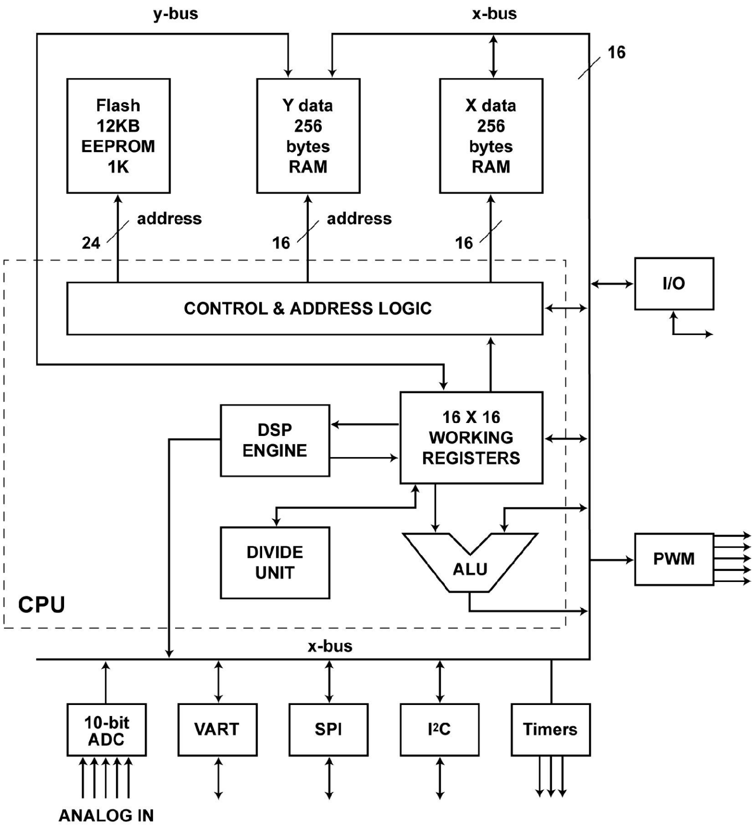
An embedded controller is a single chip computer with processor, memory, and I/O on chip. They are the heart of almost every electronic product today. The embedded controller is mainly used as a controller to direct the sequence of operations in a device and to control and monitor peripherals like input buttons and keyboards and output displays such as LCDs.

Embedded controllers can also do math. They all add and multiply with only a few exceptions. What keeps them from being widely used in DSP is that their architecture and lack of a multiply and accumulate function make them too slow for most applications.

However, that is changing. Some manufacturers are already adding DSP features like multiply and add instructions and other architectural features that speed up the kind of operations needed in DSP algorithms. Embedded processors from ARM, Intel, MIPS, Freescale, and others are examples.

Enhanced Embedded Controller Example

An example of a single chip embedded controller designed for standard operations as well as selected DSP functions is Microchip Technology’s dsPIC30F2010. Microchip’s PIC series of embedded controllers are the most widely used in embedded applications because of their small size, low power, and low cost. A simplified block diagram of this chip is shown here. All memory and I/O circuits are on-chip with the CPU.



dsPIC30F2010 Features

A key feature of the dsPIC30F2010 chip is an on-chip analog-to-digital converter (ADC). It has a 10-bit resolution and can sample at a rate up to 500k samples/second. It also has a 6-input analog multiplexer.

The dsPIC30F2010 is used primarily in filters but is also useful in DSP motor control applications where the output speed control to the motor is a PWM signal whose characteristics are computed from motor speed, torque and other sensor inputs and the control algorithm.

A list of other features is available in the Learning Resources tab of this module.

Embedded Logic

Embedded logic refers to individual logic circuits like gates and flip flops that have been put together to implement one specific DSP algorithm. It is entirely logic and no software. This special circuit does one specific function like filtering. For example, some cell phones use a special digital filter designed with CMOS logic circuits to provide filtering at one point in the cell phone receiver. No special external chip is required as the DSP circuits designed for that function are made as the part of the larger cell phone system chip.

PLDs and ASICs

Since most DSP must be very fast, programmable logic devices (PLDs) are sometimes used to implement DSP functions. For example, a programmable array logic (PAL) or complex programmable logic device (CPLD) is sometimes used to create a filter.

With field programmable gate arrays (FPGA) getting less expensive, these too can be used to create special larger DSP operations. These work well with the more complex operations like FFTs. The big benefit of the FPGA is that it is reprogrammable making design faster and simpler and providing for updated performance or features at a later time.

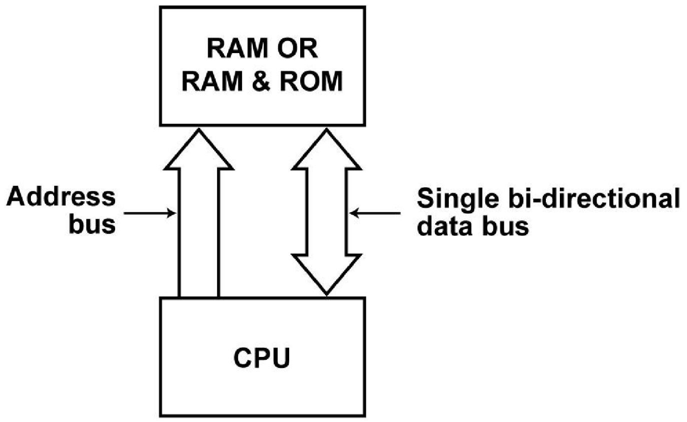
Applications specific integrated circuits (ASICs) are also used to implement DSP functions when very high volume is expected in the final product. The circuit can be optimized for a particular operation. These chips are fixed so they cannot be changed but their cost and higher performance often justify their use.

Programmable DSP

Most DSP is implemented with special embedded controller chips designed for DSP. Their organization and operation are optimized to do all those special operations demanded by DSP algorithms. These chips also contain both random access (read/write) memory (RAM) as well as one or more forms of read-only memory (ROM). The DSP program is stored in the ROM while the sampled data to be processed is stored in RAM. The chip also contains input/output (I/O) interface circuits for getting programs and data in and out.

Special development software that runs on a PC is used to write the programs that implement the desired processing algorithms. The C language is the most often used but assembly language can also be used.

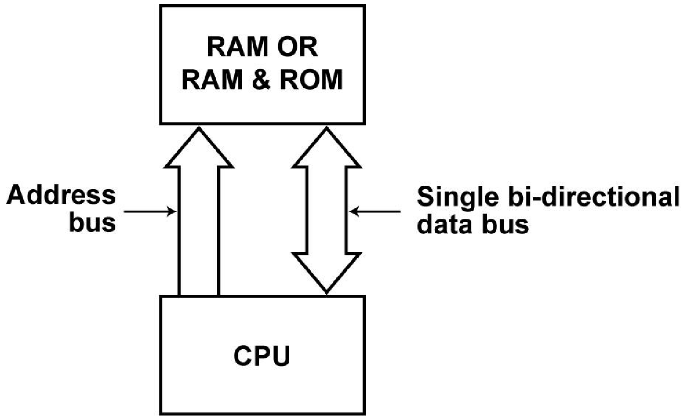
DSP Chip Architecture



Most microprocessors use what is called the von Neumann architecture. This arrangement provides a single bus between the memory and the central processing unit (CPU). Both the program and the data are stored in the same memory space and must be accessed via the single bus. Processing is performed by sequentially accessing the instructions in the program one at a time and executing them. As each instruction is executed, some will require access to data stored in the memory.

This figure shows the single data path between the CPU and memory.

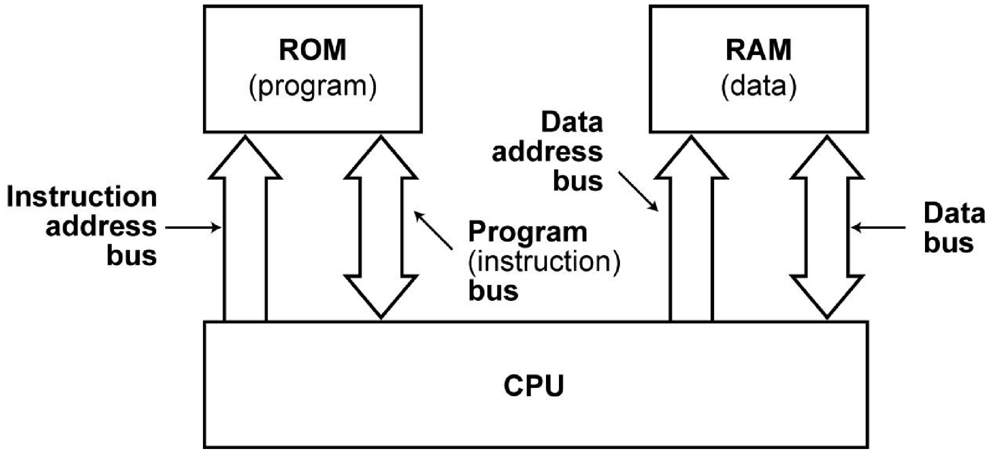
Bottleneck



Memory is typically a mix of both RAM and ROM but both memories share the same address space and the single data in/out bus. After an instruction fetch over the data bus, the CPU may also receive data from or send data to the RAM over the same data bus.

The only thing that makes this arrangement practical and workable is the very high speed with which we can access the instructions and data over the bus. Yet with some operations, this access arrangement limits processing speed. It is sometimes referred to as the von Neumann bottleneck.

Harvard Architecture



DSP chips use a different arrangement. Two memories and two buses are provided, one for the program and the other for the data as shown here. This arrangement is called the Harvard architecture. It permits simultaneous access to both instructions and data thereby greatly speeding up all processing operations. Virtually all DSP chips and enhanced embedded controllers use Harvard architecture or some variation thereof.

Other DSP Features

Besides the Harvard architecture, DSP chips have other features designed to simplify and speed up the unique operations required in DSP algorithms. Fast multipliers, a multiply and accumulate (MAC) instruction and operation, large barrel shifters that can shift sequential data right or left, circular buffers which permit rapid access to the same data in the same sequence again and again, and special address generation units are included. Some DSP chips can also perform these operations on both fixed point data as well as floating point data. Fixed point data and floating point data will be discussed later.

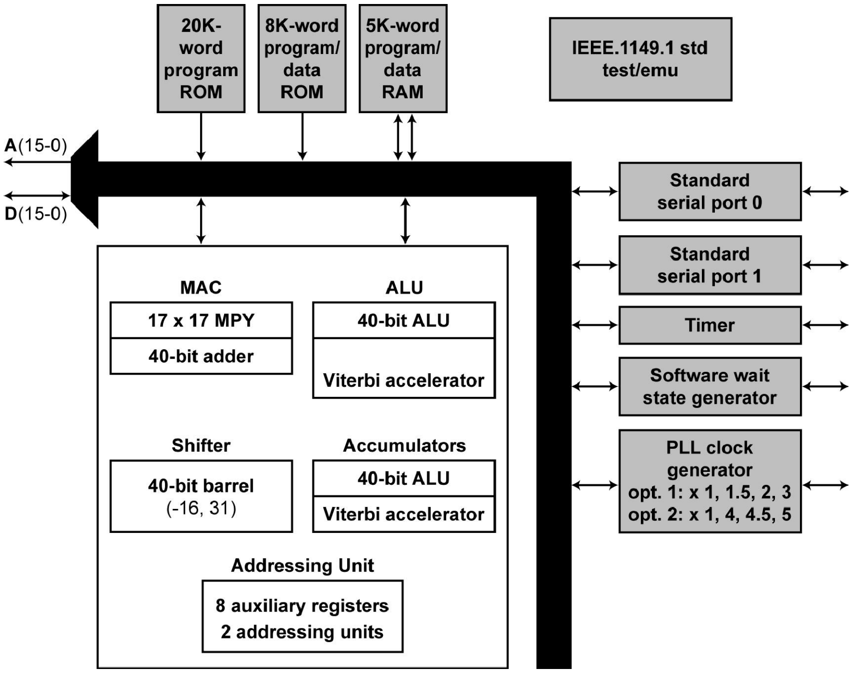
Most commercial DSP chips are made by Analog Devices, Freescale Semiconductor (formerly Motorola), and Texas Instruments.

A Typical DSP Chip

Texas Instruments (TI) produces over 80% of all DSP chips in use. They have been making DSP chips since the late 1970s and have the most extensive line of chips for almost any application. Their chips are designated by the part number TMS320Cxx. The C means CMOS technology and the xx represents the specific version. There are many versions available with different clock speeds and memory sizes. Many have on-chip memories but other models use external RAM and/or ROM.

TI TMS320C54 DSP

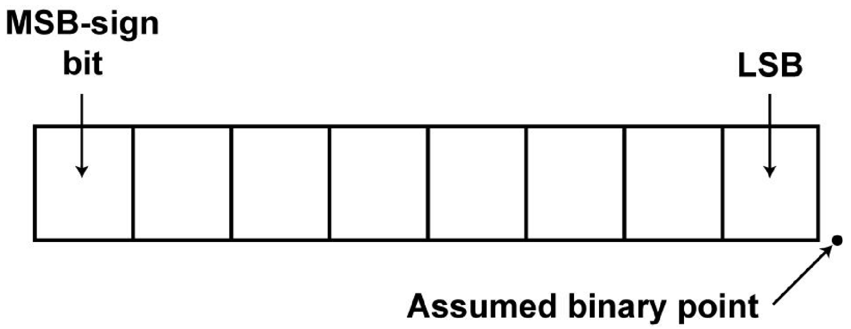
A good example of a popular programmable DSP chip is the Texas Instruments (TI) TMS320C54. There are many models and variations. The figure shows a simplified block diagram of the version called TMS320LC541. The L indicates a low power consumption version for battery powered applications. It has a basic 16-bit word size and there are 16-bit data and address buses labeled A(15-0) and D(15-0) to external devices if needed.



Fixed Point Data Representation

Most microprocessors represent data in what is known as 2's complement binary. A fixed length word size is defined to represent the data. Common fixed word sizes are 8, 16, and 32-bits. Keep in mind that the range of values that can be represented by a fixed point word is $(2^N - 1)$ where N is the number of bits. For an 8-bit word, the range is 0 to 255. For a 16-bit word, the range is 0 to 65,536. For a 32-bit word, the range is 0 to 4,292,967,296. Designers must choose a DSP that has the range and precision to represent the data to be processed.

Representing Negative Numbers

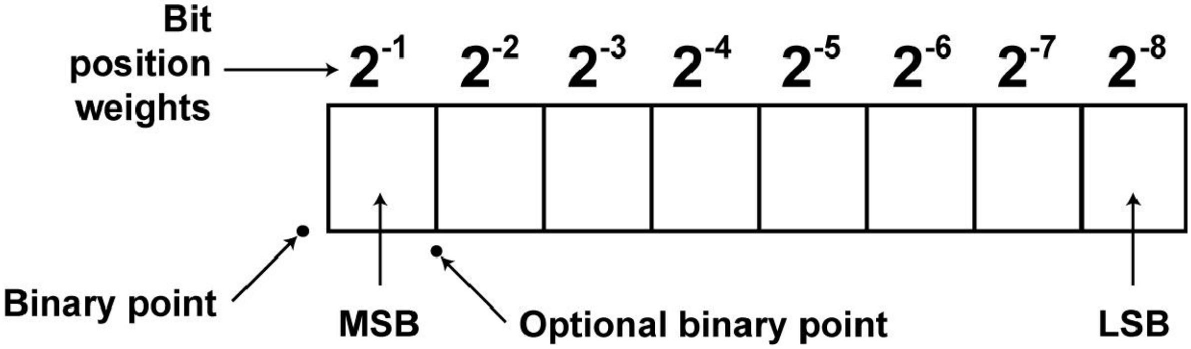


In 2’s complement number representation, the most significant bit (MSB) is designated as a sign bit. If the bit is 0, the remaining bits indicate the magnitude value. This is an integer value.

If the sign bit is 1 it means the value is negative and represented in 2’s complement form. Remember that the 2’s complement value is determined by complementing all bits and adding one to the least significant bit (LSB).

For an 8-bit word, the range of values that can be represented is +127 to -128.

Fractional Numbers



In some applications, fractional values must be used. This is done by simply defining a binary point to the left of the MSB as here. In this way the weights of the bit positions from left to right are 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, $2^{-7,}$ 2^{-8} or 0.5, 0.25, 0.125, 0.0625, 0.03125, and so on.

By putting the binary point between the MSB and next most significant bit, negative fractional values can be represented.

Dynamic Range

Determining what word size to use depends upon the range of binary sample values coming from the analog-to-digital converter (ADC). That word size determines the dynamic range which is a measure of the smallest value to the largest value. It is usually expressed in decibels (dB).

For example, if we use an 8-bit ADC with a 5 volt maximum value the minimum value will be $5/2^N$ where N is the number of bits. This translates into a minimum value of 5/256 = .01953 volts or 19.53 mV. This produces a dynamic range of:

$$dB = 20\log(max/min) = 20\log(5/.01953) = 20\log(256)$$

$$dB = 20\,(2.4) = 48\ dB$$

This is a small dynamic range and in some applications, a range of 100 dB or more is needed. This means more bits in the ADC and more bits in the samples to be processed.

Precision and Errors

Most DSP today is carried out with 16-bit DSP chips. That usually gives sufficient dynamic range for most applications. And that is the maximum number of bits usually available from fast ADCs. ADCs with 12 and 14-bits are more common.

However, some applications may require even greater dynamic range. Furthermore, larger binary values can help to minimize the errors that creep into the processes. Coefficients in digital filters must be represented with as great a precision as is available. The reason for this is that as multiple sequential multiplications and additions take place, they are usually accompanied by truncation and rounding errors. For each new multiplication and addition, the amount of error increases. The overall result is a sequence that contains errors that result in a filter response not as designed or a spectral analysis that is flawed. The solution is longer binary numbers or the use of floating point number notation.

Floating Point Numbers

Floating point numbers are the binary equivalent of scientific notation. Remember that scientific notation is a way to represent numbers with a value between 1 and 10 multiplied by some power of 10 factor. For instance, the numbers 78500 and 0.0000000062 are expressed in scientific notation as:

$78500 = 7.85 \times 10^{4}$

$0.0000000062 = 6.2 \times 10^{-9}$

Floating point representation is similar. It requires two binary words, one to represent the numerical value and another to represent the power of 2 as the multiplier.

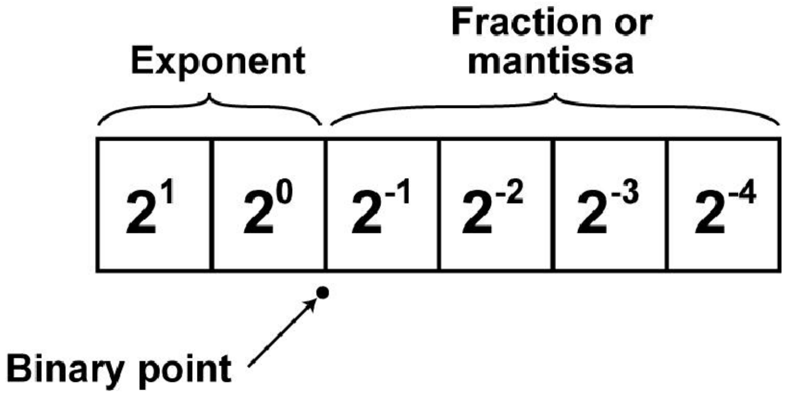
$\text{Value} = M \times 2^{N}$

M is called the mantissa, or fraction, while N is the exponent.

Floating Point Example

The figure here illustrates a simple 6-bit floating point number. Two bits are designated for the integer exponent and 4-bit for the fractional mantissa.

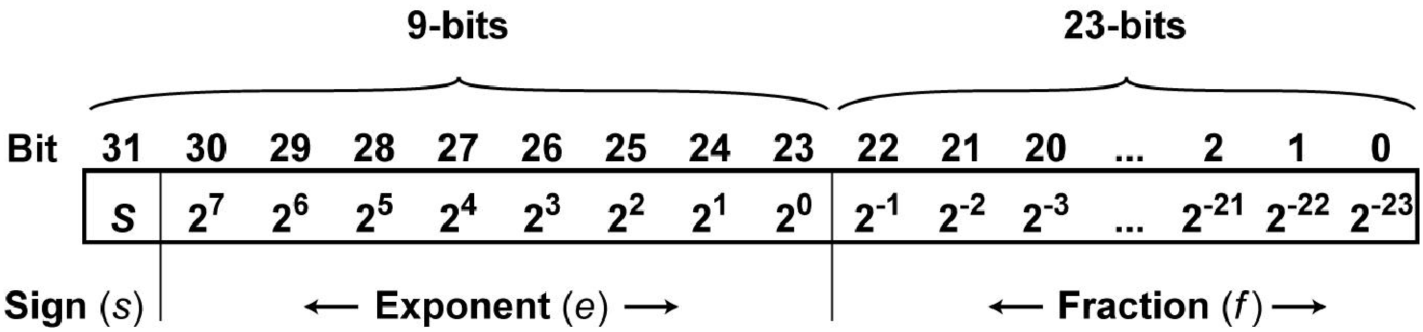
Assuming a value of 10.1010, then the value represented is 2.5.



$$\underbrace{10}_{2}.\underbrace{1010}_{0.5 + 0.125 = 0.625}$$

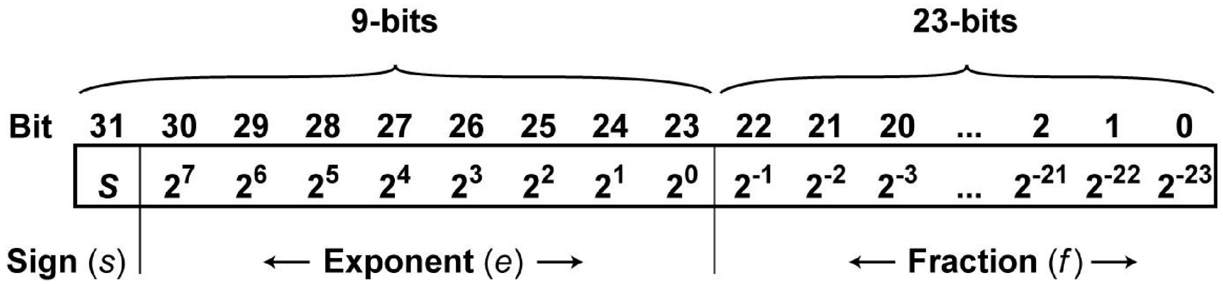
$$\begin{aligned} \text{Value} &= 0.625 \times 2^2 \\ &= 0.625 \times 4 = 2.5 \end{aligned}$$

Floating Point Format



Over the years, large scientific computers have used a variety of formats to represent floating point numbers. Today, most use the Institute of Electrical and Electronic Engineers (IEEE) P754 standard format which uses a 32-bit word as shown here.

Floating Point Format Example



9-bits

23-bits

Bit	31	30	29	28	27	26	25	24	23	22	21	20	...	2	1	0
	S	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	...	2^{-21}	2^{-22}	2^{-23}

Sign (*s*)

← **Exponent** (*e*) →

← **Fraction** (*f*) →

Here 9-bits are used for the exponent and 23 bits for the fraction. As you can imagine, this word can represent values from extremely small to extremely large.

Some DSP operations require floating point representation to get the precision and range needed for the application. Most DSP chips are fixed point devices but more sophisticated versions with floating point arithmetic logic units are available.

Test your knowledge

Digital Signal Processing
Knowledge Probe 4
Digital Signal Processing and Hardware

Click on Course Materials at the top of the page.
Then choose **Knowledge Probe 4**.