MANTECH Publications

Journal of Research in Electrical Circuits and Systems
Volume 1 Issue 1, January-April 2016

Comparison between Fixed- And Floating-Point DSPs in Their Respective Numeric Representations of Data

[1]T. Selvam, [2]Shiv Raju

[1]Assistant Professor, [2]Student (M. Tech)

Department of Electrical Engineering

Jaya Engineering College, Thiruninravur

E-mail: tselvam.eee@jec.ac.in[1]

Abstract

System developers, especially those who are new to digital signal processors (DSPs), are sometimes uncertain whether they need to use fixed- or floating-point DSPs for their systems. Both fixed- and floating-point DSPs are designed to perform the high speed computations that underlie real-time signal processing. Both feature system-on-a-chip (SOC) integration with on-chip memory and a variety of high-speed peripherals to ensure fast throughput and design flexibility. Tradeoffs of cost and ease of use often heavily influenced the fixed- or floating-point decision in the past. Today, though, selecting either type of DSP depends mainly on whether the added computational capabilities of the floating-point format are required by the application. In this paper we will discuss the comparison between Fixed- and Floating-Point DSPs in their respective numeric representations of data.

Keywords: *Digital Signal Processing, Fixed Point, Floating Point, Fixed Point Vs Floating Point*

INTRODUCTION

Digital signal processors (DSPs) are essential for real-time processing of real-world digitized data, performing the high-speed numeric calculations necessary to enable a broad range of applications – from basic consumer electronics to sophisticated industrial instrumentation. Software programmable for maximum flexibility and supported by easy-to-use, low-cost development tools, DSPs enable

designers to build innovative features and differentiating value into their products, and get these products to market quickly and cost-effectively.

There are many considerations that system developers weigh when selecting digital signal processors for their applications. Among the key factors to consider are the computational capabilities required for the application, processor and system costs, performance attributes, and ease of development. Balancing these factors together, designers can identify the DSP that is best suited for an application.

FIXED POINT VS FLOATING POINT

Digital signal processing can be separated into two categories - fixed point and floating point. These designations refer to the format used to store and manipulate numeric representations of data. Fixed-point DSPs are designed to represent and manipulate integers – positive and negative whole numbers – via a minimum of 16 bits, yielding up to 65,536 possible bit patterns (2^{16}). Floating-point DSPs represent and manipulate rational numbers via a minimum of 32 bits in a manner similar to scientific notation, where a number is represented with a mantissa and an exponent (e.g., A x 2B, where 'A' is the mantissa and 'B' is the exponent), yielding up to 4,294,967,296 possible bit patterns (2^{32}).

The term 'fixed point' refers to the corresponding manner in which numbers are represented, with a fixed number of digits after, and sometimes before, the decimal point. With floating-point representation, the placement of the decimal point can 'float' relative to the significant digits of the number. For example, a fixed-point representation with a uniform decimal point placement convention can represent the numbers 123.45, 1234.56, 12345.67, etc, whereas a floating-point representation could in addition represent 1.234567, 123456.7, 0.00001234567, 1234567000000000, etc. As such, floating point can support a much wider range of values than fixed point, with the ability to represent very small numbers and very large numbers.

With fixed-point notation, the gaps between adjacent numbers always equal a value of one, whereas in floating-point notation, gaps between adjacent numbers are not uniformly spaced – the gap between any two numbers is approximately ten million times smaller than the value of the numbers (ANSI/IEEE Std. 754 standard format),

with large gaps between large numbers and small gaps between small numbers.

Figure 1 illustrates the primary trade-offs between fixed and floating point DSPs. Fixed point arithmetic is much faster than floating point in general purpose computers. However, with DSPs the speed is about the same, a result of the hardware being highly optimized for math operations. The internal architecture of a

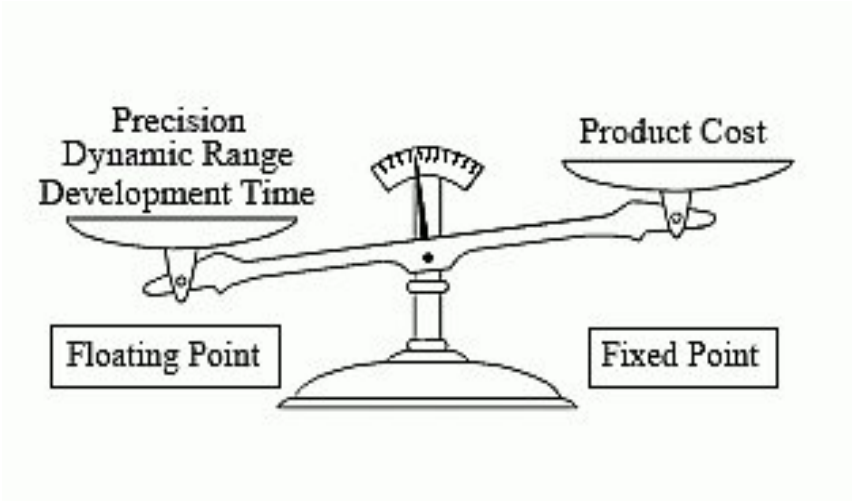

Figure 1 Fixed versus floating point. Fixed point DSPs are generally cheaper while floating point devices have better precision, higher dynamic range, and a shorter development cycle.

floating point DSP is more complicated than for a fixed point device. All the registers and data buses must be 32 bits wide instead of only 16; the multiplier and ALU must be able to quickly perform floating point arithmetic, the instruction set must be larger (so that they can handle both floating and fixed point numbers), and so on. Floating point (32 bit) has better precision and a higher dynamic range than fixed point (16 bit) . In addition, floating point programs often have a shorter development cycle, since the programmer doesn't generally need to worry about issues such as overflow, underflow, and round-off error.

On the other hand, fixed point DSPs have traditionally been cheaper than floating point devices. Nothing changes more rapidly than the price of electronics; anything you find in a book will be out-of-

date before it is printed. Nevertheless, cost is a key factor in understanding how DSPs are evolving, and we need to give you a general idea.

COST VERSUS EASE OF USE

The much greater computational power offered by floating-point DSPs is normally the critical element in the fixed- or floating-point design decision. However, in the early 1990s, when TI released its first floating-point DSP products, other factors tended to obscure the fundamental mathematical issue. Floating-point functions require more internal circuitry, and the 32-bit data paths were twice as wide as those of fixed-point DSPs, which at that time integrated only a single 16-bit data path. These factors, plus the greater number of pins required by the wider data bus, meant a larger die and larger package that resulted in a significant cost premium for the new floating-point devices. Fixed-point DSPs therefore were favored for high-volume applications like digitized voice and telecom concentration cards, where unit manufacturing costs had to be kept low. Offsetting the cost issue at that time was ease of use. TI floating-point DSPs were among the first DSPs to support the C language, while fixed-point DSPs still needed to be programmed at the assembly code level. In addition, real arithmetic could be coded directly into hardware operations with the floating-point format, while fixed-point devices had to implement real arithmetic indirectly through software routines that added development time and extra instructions to the algorithm. Because floating-point DSPs were easier to program, they were adopted early on for low-volume applications where the time and cost of software development were of greater concern than unit manufacturing costs. These applications were found in research, development prototyping, military applications such as radar, image recognition, three-dimensional graphics accelerators for workstations and other areas. Today the early differences in cost and ease of use, while not altogether erased, are considerably less pronounced. Scores of transistors can now fit into the same space required by a single transistor a decade ago, leading to SOC integration that reduces the impact of a single DSP core on die size and expense. Many DSP-based products, such as TI's broadband, camera imaging, wireless baseband and OMAP™ wireless application platforms, leverage the advantages of rescaling by integrating more than a single core in a product targeted at a specific market. Fixed-point DSPs continue to benefit more from cost reductions of scale in

manufacturing, since they are more often used for high-volume applications; however, the same reductions will apply to floating point DSPs when high-volume demand for the devices appears. Today, cost has increasingly become an issue of SOC integration and volume, rather than a result of the size of the DSP core itself. The early gap in ease of use has also been reduced. TI fixed-point DSPs have long been supported by outstandingly efficient C compilers and exceptional tools that provide visibility into code execution. The advantage of implementing real arithmetic directly in floating-point hardware still remains; but today advanced mathematical modeling tools, comprehensive libraries of mathematical functions, and off-the-shelf algorithms reduce the difficulty of developing complex applications—with or without real numbers—for fixed-point devices. Overall, fixed-point DSPs still have an edge in cost and floating-point DSPs in ease of use, but the edge has narrowed until these factors should no longer be overriding in the design decision.

FLOATING-POINT ACCURACY

As the cost of floating-point DSPs has continued to fall, the choice of using a fixed- or floating-point DSP boils down to whether floating-point math is needed by the application data set. In general, designers need to resolve two questions: What degree of accuracy is required by the data set? and How predictable is the data set? The greater accuracy of the floating-point format results from three factors. First, the 24-bit word width in TI C67x™ floating-point DSPs yields greater precision than the C62x™ 16-bit fixed-point word width, in integer as well as real values. Second, exponentiation vastly increases the dynamic range available for the application. A wide dynamic range is important in dealing with extremely large data sets and with data sets where the range cannot be easily predicted. Third, the internal representations of data in floating-point DSPs are more exact than in fixed-point, ensuring greater accuracy in end results. The final point deserves some explanation. Three data word widths are important to consider in the internal architecture of a DSP. The first is the I/O signal word width, already discussed, which is 24 bits for C67x floating-point, 16 bits for C62x fixed-point, and can be 8, 16, or 32 bits for C64x™ fixed-point DSPs. The second word width is that of the coefficients used in multiplications. While fixed-point coefficients are 16 bits, the same as the signal data in C62x DSPs, floating-point coefficients can be 24 bits or 53 bits of precision, depending whether

single or double precision is used. The precision can be extended beyond the 24 and 53 bits in some cases when the exponent can represent significant zeroes in the coefficient. Finally, there is the word width for holding the intermediate products of iterated multiply accumulate (MAC) operations. For a single 16-bit by 16-bit multiplication, a 32-bit product would be needed, or a 48-bit product for a single 24-bit by 24-bit multiplication. (Exponents have a separate data path and are not included in this discussion.) However, iterated MACs require additional bits for overflow headroom. In C62x fixedpoint devices, this overflow headroom is 8 bits, making the total intermediate product word width 40 bits (16 signal + 16 coefficient + 8 overflow). Integrating the same proportion of overflow headroom in C67x floating-point DSPs would require 64 intermediate product bits (24 signal + 24 coefficient + 16 overflow), which would go beyond most application requirements in accuracy. Fortunately, through exponentiation the floating-point format enables keeping only the most significant 48 bits for intermediate products, so that the hardware stays manageable while still providing more bits of intermediate accuracy than the fixed-point format offers. These word widths are summarized in Table 1 for several TI DSP architectures.

VIDEO AND AUDIO DATA SET REQUIREMENTS

The advantages of using the fixed- and floating-point formats can be illustrated by contrasting the data set requirements of two common signal-processing applications: video and audio. Video has a high sampling rate that can amount to tens or even hundreds of megabits per second (Mbps) in pixel data, depending on the application. Pixel data is usually represented in three words, one for each of the red, green and blue (RGB) planes of the image. In most systems, each color requires 8 to 12 bits, though advanced applications may use up to 14 bits per color. Key mathematical operations of the industry-standard MPEG video compression algorithms include discrete cosine transforms (DCTs) and quantization, and there is limited filtering.

Table 1. Word widths for TI DSPs

TI DSP(s)	Format	Word Width		
		Signal I/O	Coefficient	Intermediate result
C25x	fixed	16	16	40
C5x™/C62x™	fixed	16	16	40
C64x™	fixed	8/16/32	16	40
C3x™	floating	24 (mantissa)	24	32
C67x™(SP)	floating	24 (mantissa)	24	24/53
C67x(DP)	floating	53	53	53

Audio, by contrast, has a more limited data flow of about 1 Mbps that results from 24 bits sampled at 48 kilosamples per second (ksps). A higher sampling rate of 192 ksps will quadruple this data flow rate in the future, yet it is still significantly less than video. Operations on audio data include infinite impulse response (IIR) and intensive filtering. Video thus has much more raw data to process than audio. DCTs and quantization are handled effectively using integer operations, which together with the short data words make video a natural application for C62x and C64x fixed-point DSPs. The massive parallelism of the C64x makes it a excellent platform for applications that run multiple video channels, and some C64x DSP products have been designed with on-chip video interfaces that provide seamless data throughput.

Video may have a larger data flow, but audio has to process its data more accurately. While the eye is easily fooled, especially when the image is moving, the ear is hard to deceive. Although audio has usually been implemented in the past using fixed-point devices, high-fidelity audio today is transistioning to the greater accuracy of the floating-point format. Some C67x DSP products further this trend by integrating a multichannel audio serial port (McASP) in order to make audio system design easier. As the newest audio innovations become increasingly common in consumer electronics, demand for floating-point DSPs will also rise,

helping to drive costs closer to parity with fixed-point DSPs.

The wider words (24-bit signal, 24-bit coefficient, 53-bit intermediate product) of C67x DSPs provide much greater accuracy in audio output, resulting in higher sound quality. Sampling sound with 24 bits of accuracy yields 144 dB of dynamic range, which provides more than adequate coverage for the full amplitude range needed in sound reproduction. Wide coefficients and intermediate products provide a high degree of accuracy for internal operations, a feature that audio requires for at least two reasons.

First, audio typically use cascaded IIR filters to obtain high performance with minimal latency., But, in doing so, each filtering stage propagates the errors of previous stages. So a high degree of precision in both the signal and coefficients are required to minimize the effects of these propagated errors. Second, signal accuracy must be maintained, even as it approaches zero (this is necessary because of the sensitivity of the human ear). The floating-point format by its nature aligns well with the sensitivity of the human ear and becomes more accurate as floating point numbers approach 0. This is the result of the exponent's keeping track of the significant zeros after the binary point and before the significant data in the mantissa. This is in contrast to a fixed point system for very small fractional numbers. All of these aspects of floating-point real arithmetic are essential to the accurate reproduction of audio signals.

OTHER APPLICATION AREAS

The data sets of other types of applications also lend themselves better to either fixedor floating-point computations. Today, one of the heaviest uses of DSPs is in wired and wireless communications, where most data is transmitted serially in octets that are then expanded internally for 16-bit processing based on integer operations. Obviously, this data set is extremely well-suited for the fixed-point format, and the enormous demand for DSPs in communications has driven much of fixed-point product development and manufacturing.

Floating-point applications are those that require greater computational accuracy and flexibility than fixed-point DSPs offer. For example, image recognition used for medicine is similar to audio in requiring a high degree of accuracy. Many levels of signal input from light, x-rays, ultrasound and other sources must be defined and processed to create output

images that provide useful diagnostic information. The greater precision of C67x signal data, together with the device's more accurate internal representations of data, enable imaging systems to achieve a much higher level of recognition and definition for the user.

Radar for navigation and guidance is a traditional floating-point application since it requires a wide dynamic range that cannot be defined ahead of time and either uses the divide operator or matrix inversions. The radar system may be tracking in a range from 0 to infinity, but need to use only a small subset of the range for target acquisition and identification. Since the subset must be determined in real time during system operation, it would be all but impossible to base the design on a fixed-point DSP with its narrow dynamic range and quantization effects.

Wide dynamic range also plays a part in robotic design. Normally, a robot functions within a limited range of motion that might well fit within a fixed-point DSP's dynamic range. However, unpredictable events can occur on an assembly line. For instance, the robot might weld itself to an assembly unit, or something might unexpectedly block its range of motion. In these cases, feedback is well out of the ordinary operating range, and a system based on a fixed-point DSP might not offer programmers an effective means of dealing with the unusual conditions. The wide dynamic range of a floatingpoint DSP, however, enables the robot control circuitry to deal with unpredictable circumstances in a predictable manner.

A DATA SET DECISION

In recent years, as the world of digital signal processing has become much larger, DSPs have become application-driven. SOC integration means that, along with application-specific peripherals, different cores can be integrated on the same device, enabling DSP products to be tailored for the requirements of specific markets. In this environment, floating-point capabilities have become another element in the overall DSP product mix.

CONCLUSION

There are still some differences in cost and ease of use between fixed- and floating point DSPs, but these have become less significant over time. The critical feature for designers is the greater mathematical flexibility and accuracy of the floating-point format. For application data sets that require real arithmetic, greater precision

and a wider dynamic range, floating-point DSPs offer the best solution. Application data sets that do not require these computational features can normally use fixed-point DSPs. Once the data set requirements have been determined, it should no longer be difficult to decide whether to use a fixed- or floating-point DSP.

In summary, floating-point DSPs are optimized for specialized, computationally intensive applications, whereas fixed-point DSPs are optimized for high-volume, general purpose applications. Development costs can be higher for fixed point, owing to the relative difficulty of algorithm implementation, but the cost of the final product will often be reduced. Product costs for applications that leverage floating-point DSPs can be higher, owing to processor cost and lower manufacturing volumes, but designers will realize ease-of-development benefits and greater overall system precision. Ultimately, the data set requirements associated with the target application will dictate the need for fixed-point or floating-point processing.

REFERENCES

I. "Digital Signal Processing", by Salilvahanan. Pp. 128-137.

II. "Digital Signal Processing A Filtering Approach (English) 1st Edition" by Cengage Learning. Pp 234-246.

III. "Fundamentals Of Digital Signal Processing" by Wiley India. Pp. 233-239.

IV. "Essentials of Digital Signal Processing Using MATLAB (English) 3rd Edition" by Vijay k Ingle. Pp. 163-174.

V. "Multirate Digital Signal Processing" by 'Rabiner, Lawrence R.' pp. 278-295.

VI. "Digital Signal Processing System Design LabVIEW - Based Hybird Programming (With CD) 2nd Edition" by Nasser Kehtarnawaz.

VII. "Digital Signal Processing" by Emmanuel Ifeachor, Barry Jervis. 2002. Pp. 234-256.

VIII. "Digital Signal Processing (Principles And Implementations)" by Jay M. Joshi,Jigar H. Shah. 2011. Pp. 147-158.

IX. www. analog.com.

X. "Fundamentals of Analog and Digital Signal Processing" by Li Tan. 2008. Pp. 246-275.

XI. "Practical Digital Signal Processing" by Edmund Lai – 2003. Pp. 137-142.

XII. "A Simple Approach to Digital Signal Processing" by Craig Marven, Gillian Ewers – 1996. Pp. 246-278.