Overview and Development of Digital SignalProcessing

Chun-xu Zhang* · Yun-ho Shin**

ABSTRACT

Digital signal processing (DSP) is the process of taking a signal and performing an algorithm on it to analyze, modify, or better identify that signal.[1] To take advantage of DSP advances, one must have at least a basic understanding of DSP theory along with an understanding of the hardware architecture designed to support these new advances. There are several programming techniques that maximize the efficiency of the DSP hardware, as well as a few fundamental concepts used to implement DSP software. This article introduced some of these underlying functions that are the building blocks of complex signal processing functions, and It will touch on the fundamental concepts of DSP theory and algorithms and also provide an overview of the implementation and optimization of DSP software, and discuss the development of DSP.

Key words

DSP, Fundamental Concepts, DSP System, Filters, Optimization Algorithm

Ⅰ. Introduction

With sales of programmable DSPs increasing exponentially, the Digital Signal Processing industry has enjoyed explosive growth over the past decade. In the process, the demands on DSP algorithm software have also mushroomed, to the point where the critical path in the DSP development cycle has shifted from hardware to software. Software directly determines time-to-market, and software-development costs outpace those of hardware many times over, demanding advances to DSP development tools. The current trend in DSP architecture is to widen the data path and augment computational power, often at the expense of issues that significantly impact the ability to utilize its computation. With today's highly parallel architectures and complex application spaces DSP selection has become overly complicated and the addition of special instructions that support intensive algorithms, while maximizing parallelism, has further complicated the issue[1][2].

DSP software development can be divided into several fundamental tasks. They are system, algorithm, library and coding specification; compiling an optimization; simulation, emulation, testing, debugging and integration.

Ⅱ. Underlying Functions

2.1 Systems and Signals

First, I will introduce a few concepts that are common to all systems, emphasizing how these concepts are used in DSP systems. The systems generally used in DSP applications are linear time-invariant systems. The property of linearity is important for a couple of reasons. The most important reason is that the system is not dependant on the order in which processes are applied. For

* Dept. of Electric Comm. Eng. Chonnam National University
접수일자 : 2008. 04. 14
** Dept. of Comm. Eng. Tennessee University
심사완료일자 : 2008. 05. 21

example, it does not matter if we scale our input before or after we run it through a filter; the output is the same. As a result, a complex system can be divided into multiple systems. For example, an eighth-order system can be divided into four second-order systems and still produce the same output. Time invariance is important because we need to know that our system will react in the same way to an input; the output is not dependant on when the input is applied. These two qualities make our system predictable.

Signals and systems are usually graphed to show how the input and output relate to each other. There are two typical ways of viewing the data: in the time domain and in the frequency domain. The time domain is especially handy for applications such as control systems, where response times are important. The frequency domain is useful for viewing filter results to see which frequencies will pass and which will be attenuated.

We usually describe our systems in the time domain in terms of impulse response. An impulse is a stimulus of infinite magnitude and zero duration; its integral is equal to one. Since it is only a single instant of time and spans all frequencies, the impulse makes a good test to show how a system will react to an input. The response to an impulse input, called the impulse response, can be used to describe the system. In the digital domain, an impulse is an input signal whose magnitude is one at time zero and zero everywhere else, as shown in Figure 1. The way a system reacts to an impulse input can also be considered the system's transfer function. The transfer function (or impulse response) can provide us with everything we need to determine how a system will react to an input in the time or frequency domain.

A system response is usually plotted in the frequency domain to show how it will affect signals of different frequencies. When a system response is plotted in the frequency domain, there are two characteristics of concern: magnitude and phase. Magnitude is the ratio of the output's strength and the strength of its input. For example, how much of a radio signal will pass through a filter designed to pass a specific frequency? The phase is how much a frequency of the signal will be changed by the filter, usually lagging or leading. While not important in all applications, the phase can sometimes be particularly important, such as in music or speech applications.

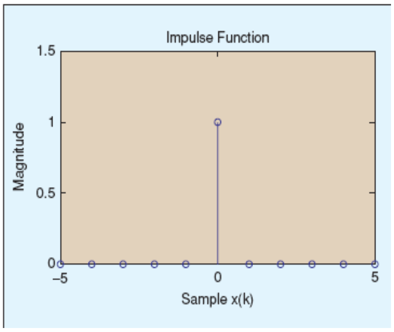

Fig. 1 An input signal in a digital domain with magnitude 1 at time 0.

2.2 DSP Systems

Analog signals are converted to digital signals through a process called sampling. Sampling is the process of taking an analog signal and converting it to discrete numbers. The sampling frequency (or sampling rate) is how many times per second the signal will be sampled. This is important because it restricts the highest frequency that can be present in a signal. Any signal greater than half the sampling rate will be folded into a lower frequency that is less than half the sampling rate; this is known as aliasing.

The inverse of the sampling frequency is known as the sampling period. The sampling period is the amount of time that elapses between the samples of the analog signal. This conversion is conducted by hardware that takes the best approximation of the voltage level and converts it to the nearest digital level that can be represented by the computer. The loss of information during the conversion is referred to as quantization.

2.3 Aliasing

Aliasing is an important concept to understand when working with digital systems. Without understanding this point, numerous unexpected problems can arise when implementing your digital system.
A visual example of aliasing is provided in Figure 2.

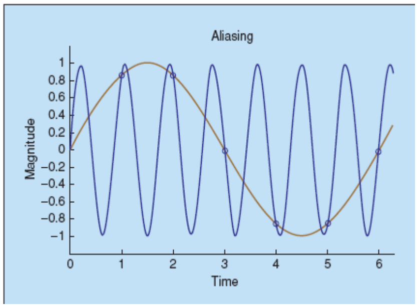

Fig. 2 A visual example of aliasing.

2.4 Basic DSP System

The basic DSP system in Figure 3 consists of an ADC, a DSP, and a digital-to-analog converter (DAC). Typically, systems have analog filters before and after the converters to make the signals more pure. Let's discuss each component in detail[3][4].

Since the signal can contain no frequencies above the Nyquist frequency, steps must be taken to ensure that the high frequencies are eliminated from the signal. This is done with an analog lowpass filter with a cutoff rate set around the Nyquist frequency. This analog filter is known as the antialiasing filter. The signal is then used as input into an ADC where the signal is converted to a digital signal so that the DSP can handle it and process it. The DSP will perform the actions required of it, such as filtering, and then pass the new signal to the DAC. The DAC then converts the digital output back to an analog signal. This analog output usually contains high frequencies introduced by the DAC, so a low-pass filter is needed to smooth the waveform back to its intended shape. This filter is known as a reconstruction filter.

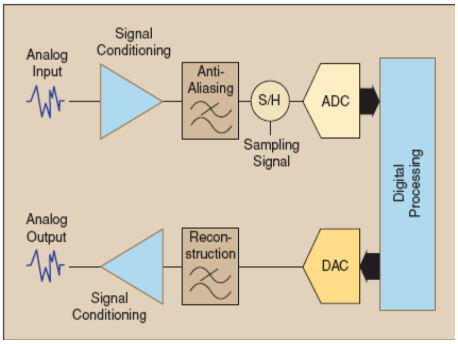

Fig. 3 A basic DSP system.

2.5 Filters

We use filters for many things in our daily lives. We use coffee filters to make our coffee in the morning. We wear sunglasses to filter harmful rays from the environment. We use our digital cell phone to filter the appropriate phone call. We filter out other speech and noise to listen for our name to be called in a busy restaurant. We also use filters at work to search for certain useful information stored on our company mainframes and computers.

All of these signals are composed of different sinusoidal components. Signals are composed of many combinations of sinusoidal periodic functions and these can be used to approximate many different signals that are very useful in the signal processing field[5].

Filtering is a fundamental process in the analog world as well as in the digital world. Almost any filter that can be implemented using analog techniques can also be implemented using digital techniques. Digital filters are by far the most widely used application in a DSP system. Digital filters operate on different frequencies in different ways, allowing some signals to pass while attenuating others. Filters can also change the phase of a signal, which can be important in applications such as digital video.

2.6. Frequency Analysis and the Fourier Transform

Most every application in DSP deals with frequencies in some way[3]. Therefore, a tool is needed that allows conversion from the time domain to the frequency domain, and vice versa. This tool builds on the Fourier transform (FT), an equation to calculate the frequencies in a signal.

Since DSPs always work with discrete,

sampled data, we will use only the discrete forms – the DFT and the FFT – in real-time DSP applications. The DFT is a discrete numerical equivalent using sums instead of integrals. It takes a set of time domain samples as inputs and returns frequencies in the range of negative one-half the sampling frequency to one half the sampling frequency. This allows one to view the spectral content of the signal. To transform the information from the frequency domain back to the time domain, the data is run through an algorithm called the inverse DFT.

2.7 DSP Architecture Optimization for DSP algorithms

Today's DSP architectures are made specifically to maximize throughput of DSP algorithms, such as a DSP filter. Some of the features of a DSP include: [4]

◗ On-chip memory :

Internal memory allows the DSP fast access to algorithm data such as input values, coefficients, and intermediate values.

◗ Special multiply and accumulate (MAC) instruction :

This is used for performing a multiply and accumulate (the crux of a digital filter) in one cycle.

◗ Separate program and data buses :

These allow the DSP to fetch code without affecting the performance of the calculations.

◗ Multiple read buses :

These are used for fetching all the data to feed the MAC instruction in one cycle.

◗ Separate write buses :

These are used for writing the results of the MAC instruction.

◗ Parallel architecture :

DSPs have multiple instruction units so that more than one instruction can be executed per cycle.

◗ Pipelined architecture :

DSPs execute instructions in stages so more than one instruction can be executed at a time. For example, while one instruction is doing a multiply, another instruction can be fetching data with other resources on the DSP chip.

◗ Circular buffers :

These make pointer addressing easier when cycling through coefficients and maintaining past inputs.

◗ Zero overhead looping :

This is special hardware included to take care of counters and branching in loops.

◗ Bit reversed addressing :

This is used for calculating FFTs.

2.8 Number Formatting Issues with DSP Algorithms

As discussed earlier, when converting an analog signal to digital format, the signal has to be truncated due to the limited precision of a DSP. DSPs come in fixed and floating-point format. When working with a floating-point format, this truncation usually is not much of a factor due to its good mix of precision and dynamic range. However, implementing hardware to deal with floating-point formats is harder and more expensive, so most DSPs on the market today have a fixed-point format. But two other problems that can occur when using fixed-point arithmetic are overflow and saturation. However, DSPs help the programmer deal with these problems. One way a DSP does this is by providing guard bits in the accumulator. DSP processors also have a mode that will automatically saturate a result if the overflow flag gets set. This saves the code from having to check the flag and manually saturate the results[6].

III. Development of DSP

At present, internet, wide band connection technology, wireless and line-wire communication technology, family network, wired or Internet protocols (IP) televisions as well as other DSP technologies have become the standard technology at the majority of families.

The real-time processing speed of DSP technology's swift developments play the huge role to electronic technology widespread thorough social life's each domain, more and more affected people daily life aspects.

During the processing speed swift development, the another important factor for the DSP technology unceasingly progresses lie in the more and more miniaturized fabrication technology has got an in-depth development. Now, the 45nm' technic has already appeared, the field is apace approaching the level of "Hypo-nanometer"-technics. The more advanced microminiaturization fabrication technologies helpfully raises the economies of scale benefit, reduces the product cost, also helpfully reduces the power consumption, enables the high-level DSP processing function to apply in the battery capacity limited portable system, also helpfully promotion the high density integration type circuit wafer' development.

While the DSP processing performance continuously enhances and the fabrication technology microminiaturization development, another influence profound technological development tendency appears, namely the DSP development personnel start to integrate the innovation function on the DSP chip. Along with the development of the high integration rate DSP chip and the SoC, the electronic products' cost unceasingly is also reducing, simultaneously unceasingly brings the newest DSP application for the more consumer realm. During the high integration rate may reduce the chip' quantity and the circuit wafer area for the system implementation, impel the price reduction simultaneity. In addition, chip level integration, because reduced the chip quantity which must supply power, therefore is also helpful to further reduces the power loss, thus promoted the motion application development[5].

IV. Conclusions

DSP engineers use a toolbox of algorithms to implement complex signal processing functions. Complex system applications can be built using combinations of supporting algorithms that themselves are composed of combinations of underlying signal processing functions.

Productivity is significantly enhanced with fully integrated DSP code development and debugging environments. On-line help for the DSP's architecture and instruction set also benefit developers productivity. These features and others will help DSP software developers address today's tough market pressures.

References

[1] Overview of Digital Signal Processing

[2] Algorithms, Part II, Robert Oshana, pp.53-58.

[3] Raj Pandey, "Advances in DSP development environment", GO DSP Corporation, Toronto, Ontario, pp.214-1919.

[4] J. Geoffrey Chase, Christopher Pretty, Alex Bedarida and Philippe, "An Applications-Based Approach to Measuring DSP Efficiency", Bettler University of Canterbury, Infineon Technologies Corp.

[5] Xin Zhang, jie Chen, XiaoXu Guo and BaoFeng Li, "Low Power and Concurrency Optimization of the Computational Unit for DSP Processors Based on Software", Microelectronics R&D Center of the Chinese Academy of Sciences, Beijing, 100029, China.
[6] Pradeep Bardia and Michael Stich, "數字技術自身自數字家庭向外擴展"

저자 소개

Chun-xu Zhang

Was born in Beijing, China, on May 13, 1984. She received the B.E degree from the Beijing Institute of Petro and Chemical in 2006, Beijing china, the M.S degree in the Electric Communication Engineering at Chonnam National University from March 2007 to now.

Email : xuxu0513@hotmail.com

Yun-ho Shin

Was born in Kang Rung, South Korea, on September 04, 1976. He received B.E degree in entered the Department of Mass Communication Science of Kyung-Nam University, 1995. Masan Korea, The Department of Mass Communication science of Joong-Ang University, 2003. M.S. Seoul, Korea, the M.E. degree in enconomics from the state University of New-York at Buffalo. In 2007.(M.S. Completion), and degree in the Dept. of Communication and Information from the University of Tennessee of USA, from 2007 to 2008.(Posdoc)

Awards : Mass Communication science.

Email : juwhan78@hanmail.net