

An Intuitive Approach to Digital Signal Processing

Kenneth W. Schachter

Senior Member of the Technical Staff

The World Leader in DSP Solutions



[1] Hello. My name is Ken Schachter and I'm a senior member of the technical staff. Welcome to An Intuitive Approach to Digital Signal Processing.

Presentation Goals

“Something in it for everyone”

- **Very Familiar with the Topic**
 - **Serve as a review == Show a technique to help explain abstract and difficult concepts to others**
- **Less Familiar with the Topic**
 - **To help become more knowledgeable and show the value in understanding the fundamental concepts**

[2] I realize that some of you may be very familiar with this topic. For that group, I'll like this to serve as a review. Better yet, if I can give you a technique to explain this to other people with less experience in this signal processing area, then I've accomplished my task for that group. Those who are less familiar with this topic, I would like to spark your interest to learn more on your own.

Course Objectives

- **After completing this course, you will be able to:**
 - **Define signal processing and related terms**
 - **Explain the relationship between time and frequency domain**
 - **Explain the Sampling Theorem**
 - **Describe components of a Digital Signal Processing System**
 - **Recognize general signal processing applications**

[3] After completing this course, you should be able to define signal processing and related terms; explain the relationship between time and frequency domain; explain the sampling theorem; describe components of a digital signal processing system; and recognize general signal processing applications.

Digital Signal Processing

Presentation Outline

- Introduction
- Signal Fundamentals
- Signal Processing System
- Signal Processing Applications
 - Filtering
 - Spectral Analysis
 - Other Topics
- Summary

[4] The outline of this course consists of an introduction; signal fundamentals; signal processing system; signal processing applications; and a summary.

Digital Signal Processing

Presentation Outline

- **Introduction**
- **Signal Fundamentals**
- **Signal Processing System**
- **Signal Processing Applications**
 - **Filtering**
 - **Spectral Analysis**
 - **Other Topics**
- **Summary**

[5] In the introduction section we will cover the general terms that are used in signal processing.

Introduction

- Digital Signal Processing
- Signal Processing and Signal Processor
- The Signal
- Analog and Digital Signals
- Digitized Signals and Digital Data
- Real-Time and Non Real-Time

[6] We'll start with a definition of digital signal processing; then look at the differences between signal processing and processors. We'll take a look at the signal itself; discuss analog and digital signals; digitized signals and digitized data; and the relationship between real time and non-real time.

Digital Signal Processing

A method of processing real-world signals (represented by a sequence of numbers) using mathematical techniques to perform transformations or extract information



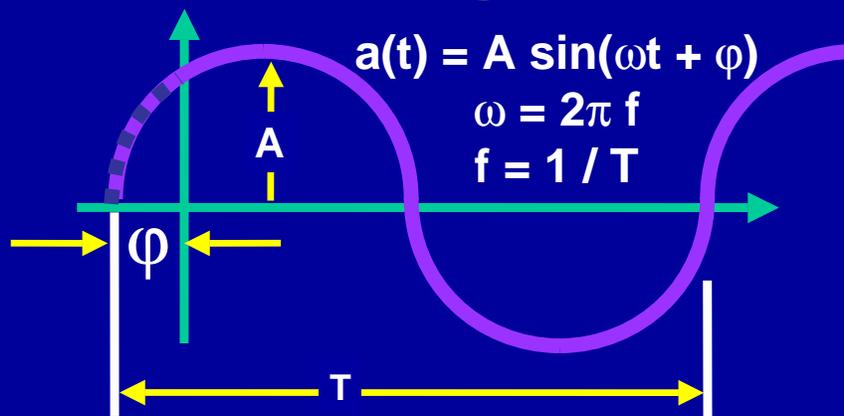
[7] One can define digital signal processing a method of processing real world signals (represented by a sequence of numbers) using mathematical techniques to perform transformations or extract information.

Signal Processing and Signal Processor

- Signal Process**ING**
 - Scientific and mathematical concepts
- Signal Process**OR**
 - A device or system which performs signal processing

[8] Signal processing or signal processors – two words that are sometimes used interchangeably. However, signal processing is nothing more than the scientific and mathematical concepts, where a processor is a device or a system that performs processing.

The Signal



- An **analog** signal is a function that is defined over a continuous range of time and in which the amplitude may have a continuous range of values
- A **digital** signal is a function in which both time and amplitude will have discrete quantized finite values

[9] Let's look at the signal. This is a real world signal. It happens to be a simple sine wave. It's found every day in life. It has some characteristics we are very interested in, the first being the A value or amplitude. If I was to talk with you and my voice was to get louder, the A value would increase.

[10] The T value is the period; $1/T$ is the frequency. As my pitch increases, the frequency increases, or the number of cycles per second of this waveform would be in a unit of time of one second.

[11] This is an analog signal. It is continuous over time and continuous over a range of amplitudes. As opposed to a digital signal where we might be interested in just discrete instances of time and discrete amplitude levels.

Analog and Digital Signals



- **Analog**
 - Real world signals
 - ◆ sound, light, temperature, pressure
- **Digital**
 - Numerical representation of the signal

[12] Analog and digital signals. The analog signal is a real world signal that you and I experience every day – sound, light, temperature, pressure. A digital signal is a numerical representation of the analog signal. It may be easier and more cost effective to process these signals in the digital world. We live in the analog world. In the real world, we can convert these signals into digital signals through our analog-to-digital conversion process, process the signals, and if needed, bring the signals back out to the analog world through the digital-to-analog converter. We will look at the analog-to-digital and digital-to-analog conversion functions in the next few models.

Digitized Signals and Digital Data

- **Digitized Signals**
 - Numerical representation of a real-world signals
 - ◆ e.g., digitized waveform
- **Digital Data**
 - Data containing character information which may not necessarily represent a signal
 - ◆ e.g., data processing with alphabetical letters, decimal digits and symbols (ASCII)

[13] Digitized signals and digital data. Though they might look a like at first, they are two different things. First, we have a digitized signal. That is a real-world signal represented by a sequence of numbers. Digital data might be a series of zeros and ones representing a character on a keyboard, for example. Though they might look alike with zeros and ones, it is the digitized signal that carries the life and meaning.

Real-Time and Non Real-Time

- **Real-Time**
 - Processing keeps pace with the input and output signal
 - ◆ e.g., performing task as it is happening
- **Non Real-Time**
 - Processing is performed off-line
 - ◆ e.g., data is stored and processed at a later time

[14] Real-time and non real-time. Real-time is when we take a sample off the signal, process it, and output the value before the next signal sample comes in. Non real-time is when we take the samples in, process them, then output them later and perform all our operations off-line. We've looked at some of the various definitions within this section. The first one we looked at was signal processing and signal processors. We're interested in processing. The next thing we looked at was signals and digital data. We will concentrate on digitized signals. Here, we're looking at real-time and non real-time. We're going to concentrate on real-time systems.

Digital Signal Processing

Presentation Outline

- Introduction
- **Signal Fundamentals**
- Signal Processing System
- Signal Processing Applications
 - Filtering
 - Spectral Analysis
 - Other Topics
- Summary



[15] Signal fundamentals.

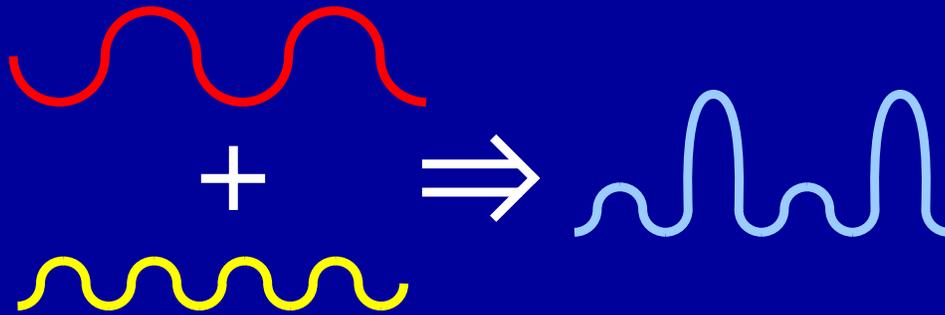
Signal Fundamentals

- Analog Signals
- Time Domain
- Frequency Domain
- Relationship of Time and Frequency
- Signals Viewed in Time and Frequency
- Sampling Theorem
- Sampling Frequency and Sampling Period

[16] In the signal fundamentals section, we'll look at the analog signal; time domain; frequency domain; the relationships between time and frequency; why you might want to view a signal in time and frequency; and sampling theorem, which we'll use to sample frequency and sampling periods.

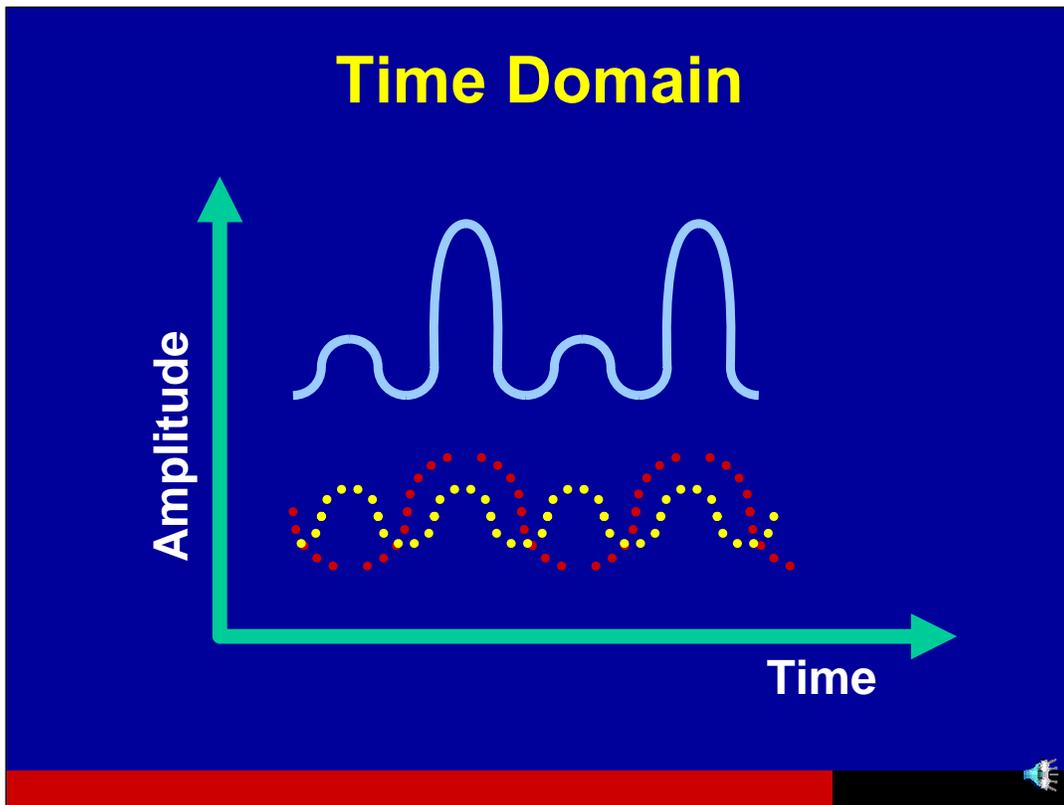
Analog Signals

Fourier - Any waveform can be generated by adding together sine waves.



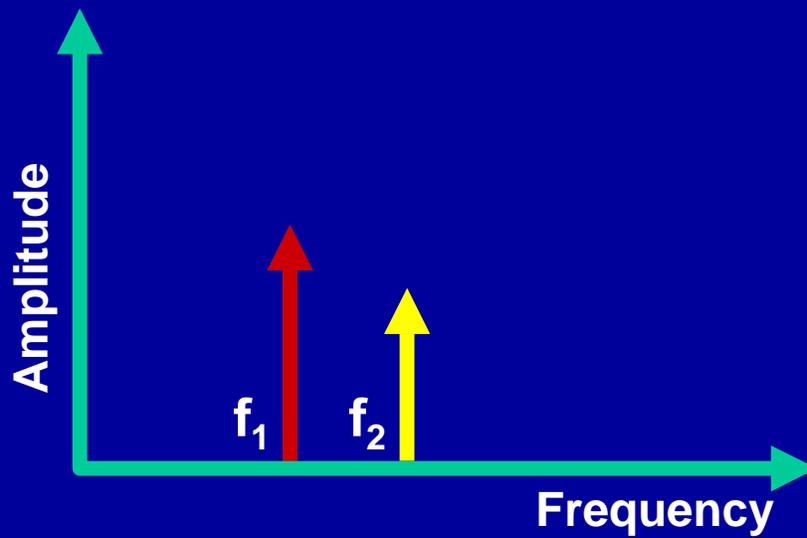
A complex signal can be decomposed into the sum of sine waves.

[17] Before we stated that any waveform can be generated by adding together simple sine waves. Or you can take a complex waveform and decompose it into the sum of simple sine waves.



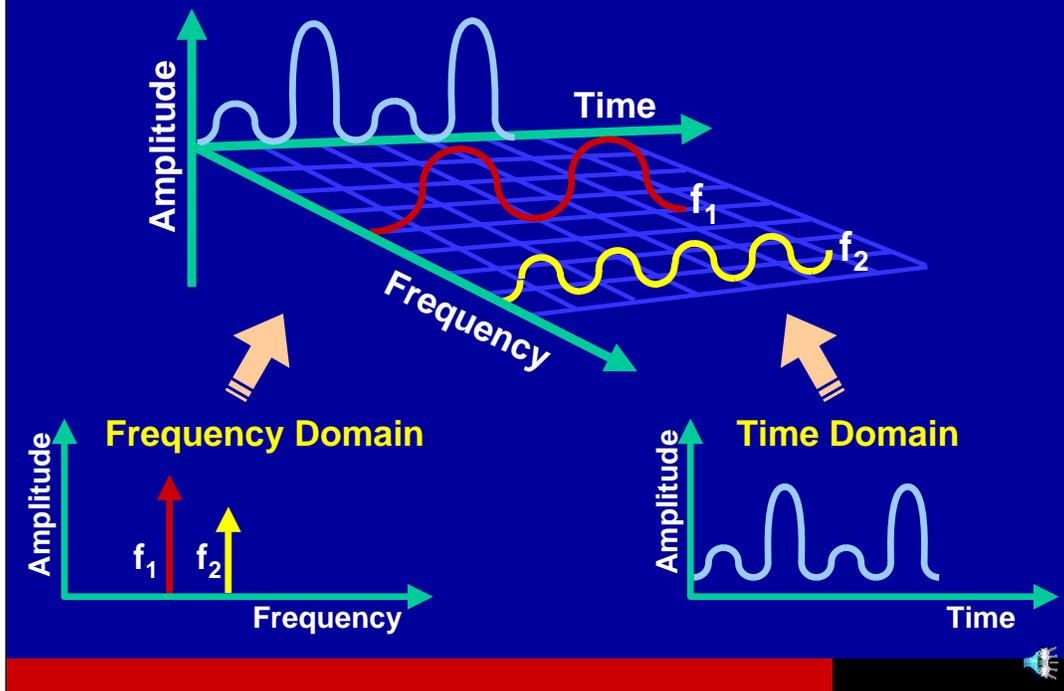
[18] When we take these signals and add them together point-to-point in time, we refer to this as the time domain. In time domain, we view the signal in amplitude versus time. The two sine waves on the bottom of the figure are added together to construct the signal on the top.

Frequency Domain



[19] We can also view the signal in what is called the “frequency domain.” In the frequency domain, we view the signals in amplitude versus frequency. This picture shows us that a complex signal is made up of two frequencies, f_1 and f_2 , added together.

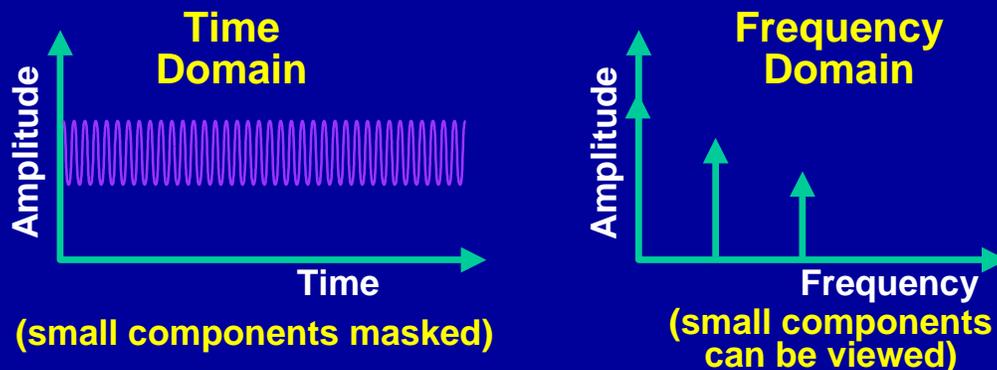
Relationship of Time and Frequency



[20] Now we understand time and frequency. How do the two relate to each other? Looking at the right side, the two sine waves are added together to form a complex waveform. If you were to look in through the left picture, we would see two spikes representing the two sine waves in amplitude. In the frequency domain, I'm looking at the amplitude versus frequency; in the time domain, amplitude versus time.

Signals Viewed in Time and Frequency

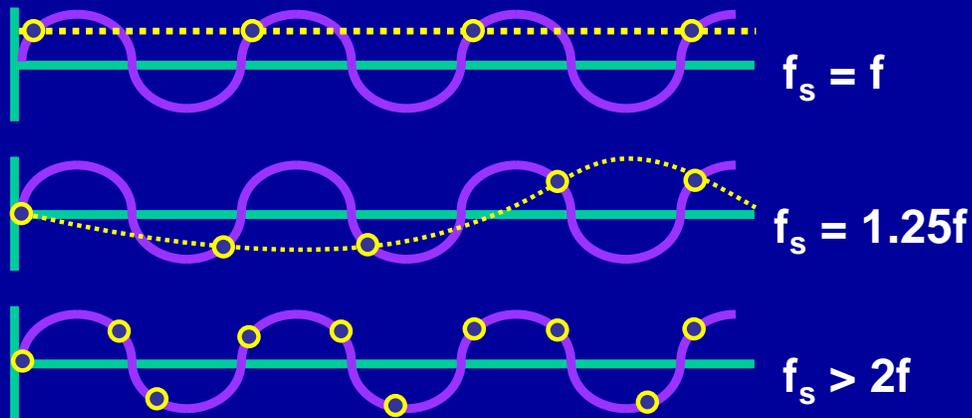
Small components are masked by larger components in the Time Domain, but can be viewed in the Frequency Domain.



[21] Now we see how time and frequency relate to each other. Why would you want to view the signal in terms of time or frequency? If you look at the figure on the left, it looks like a pure sine wave. But if you look at it in the frequency domain, we will find that there are actually two signals added together to construct the signal you see on the left. The smaller components are masked out by the larger components in the time domain. However, they are easily viewed in the frequency domain. There are many times when the frequency domain can give us more information about the signal than the time domain.

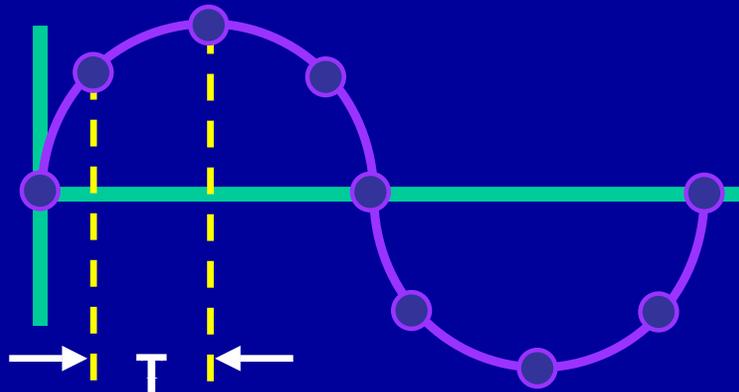
Sampling Theorem

A signal must be sampled at a rate which is at least twice the highest frequency component of interest.



[22] Next, we'll look at the Sampling Theorem. A signal must be sampled at least twice the highest frequency component of interest to retain all the information in the signal. To look at the top figure, the signal is sampled at the frequency. As a result, we have a straight line. Increasing the sampling frequency to 1.5 times the frequency of the signal shows us that we have another waveform stitched in-between. This is known as an alias. If I was to raise the sampling frequency to greater than twice the highest frequency component of interest, we would have enough dots on the waveform to reconstruct the waveform.

Sampling Frequency and Sampling Period



$$\text{Sampling Period (T)} = \frac{1}{\text{Sampling Frequency (} f_s \text{)}}$$

[23] This is why old signal processing systems work. We have samples on a waveform, and the time between the samples is where we process the signal. We will try to show the relationship between sampling frequency and sampling period. The sampling period is equal to 1 over the sampling frequency. That is, given the sampling frequency between these two points, I can figure out the time between each sample.

Sampling Frequency and Sampling Period - Example

- $F_s = 8 \text{ kHz} = 8,000 \text{ cycles per second}$
- $T = 1 / 8 \text{ kHz} = 125 \mu\text{s} = 0.000125 \text{ seconds}$
- **Instruction Cycle Time**
 - ▶ $100 \text{ ns} = 0.000000100$
 - ▶ $50 \text{ ns} = 0.000000050$
 - ▶ $25 \text{ ns} = 0.000000025$

**Sampling Period (T) / Instruction Cycle Time =
Number of Instructions per Sample**

- ▶ $125 \mu\text{s} / 100 \text{ ns} = 1,250 \text{ Instructions per Sample}$
- ▶ $125 \mu\text{s} / 50 \text{ ns} = 2,500 \text{ Instructions per Sample}$
- ▶ $125 \mu\text{s} / 25 \text{ ns} = 5,000 \text{ Instructions per Sample}$

[24] As an example, if I was to sample my signal at 8 kHz, or 8000 times a second, my sampling period would be $1/8000$, or 125 microseconds. 125 microseconds sounds like a very short amount of time. However, our processors work in the areas of 100, 50, 25, and 10 nanosecond range – much quicker units of time. Assuming a single cycle instruction cycle time, if I was to take 125 microseconds and divide it by 100 nanoseconds, I can execute 1250 instructions per sample.

[25] That might be sufficient for some systems, however, others might require more instructions in a given amount of time. So we can move to faster and faster processors. Today, it is not unusual to find instruction cycle times in the order of 10 nanoseconds. In this type of system, or an 8 kHz sampling frequency of 10 nanoseconds, I can execute 12,500 instructions.

Digital Signal Processing

Presentation Outline

- Introduction
- Signal Fundamentals
- **Signal Processing System**
- Signal Processing Applications
 - Filtering
 - Spectral Analysis
 - Other Topics
- Summary

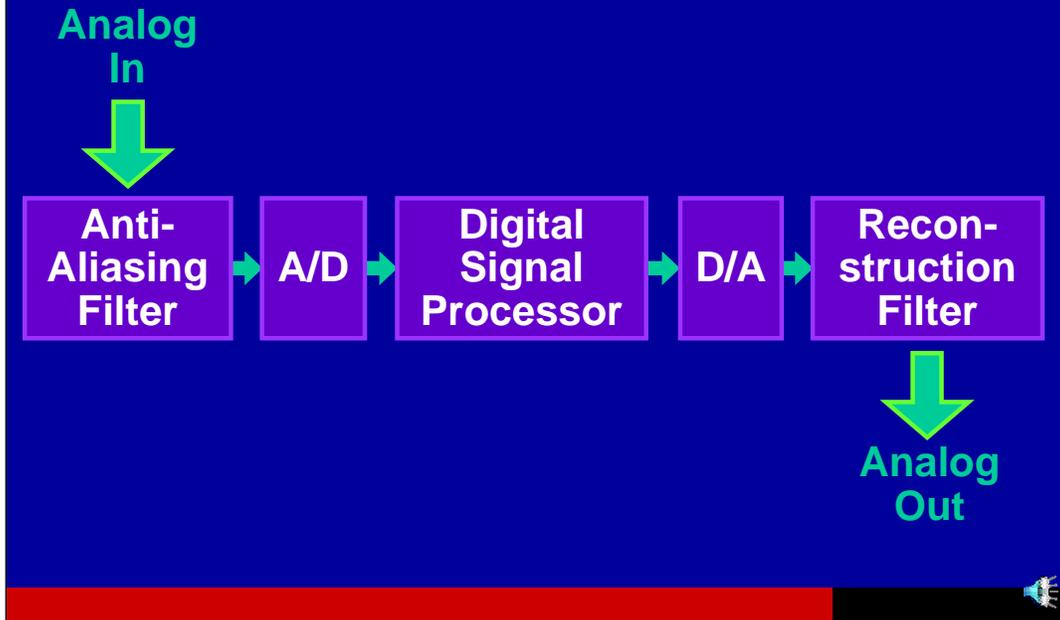
[26] The next area we will look at is signal processing system.

Signal Processing System

- **Digital Signal Processing System Block Diagram**
 - Anti-aliasing filter
 - Analog to digital converter
 - Digital signal processor
 - Digital to analog converter
 - Reconstruction filter
- **Converting Analog Signals to Digital Signals**
- **Converting Digital Signals to Analog Signals**
- **DSP System Block Diagram - Reviewed**

[27] In the signal processing system, we'll look at a generic signal processing system block diagram; discuss converting analog signals to digital signals; and digital signals to analog signals; then review the signal processing block diagram.

Digital Signal Processing System Block Diagram

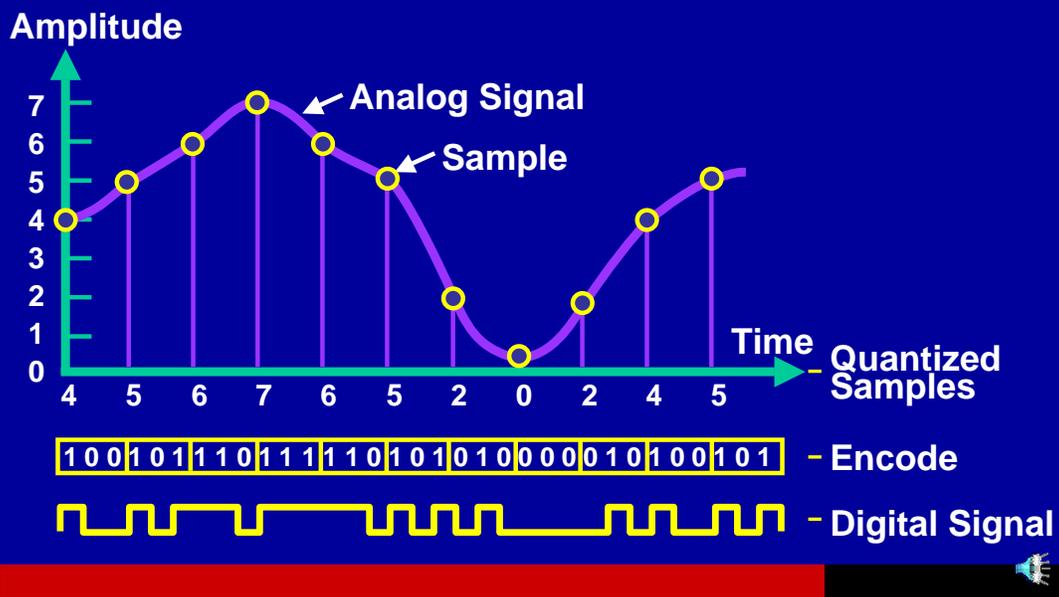


[28] The digital signal processing system block diagram consists of five generic blocks. The first block, the anti-aliasing filter, band limits our signal. It removes the frequency components above a certain point that we are not interested in. From there, we take our analog signal and digitize it through our A/D block, or our analog-to-digital converter.

[29] The digital signal processor then performs its functions on that signal. Some systems might just require numerical information -- this is where they stop -- and output the results in numerical form. Other systems require bringing the information back out to the real world, or the analog world.

[30] That information is then fed into the digital-to-analog converter block, and from there, into a reconstruction filter. When we discuss the digital-to-analog conversion, I will point out the purpose of the reconstruction filter.

Converting Analog Signals to Digital Signals



[31] Converting analog signals to digital signals consists of three steps. First, we need to sample, quantize, and encode the signal. We've already discussed how to sample a signal. We already know we need to sample at least twice the highest frequency component of interest. Now we need to quantize it. We want to find out if the sample point is closer to a 5 or a 6. If it is closer to a 5, we make it a 5. If it's closer to 6, the layer being coded in a binary sequence.

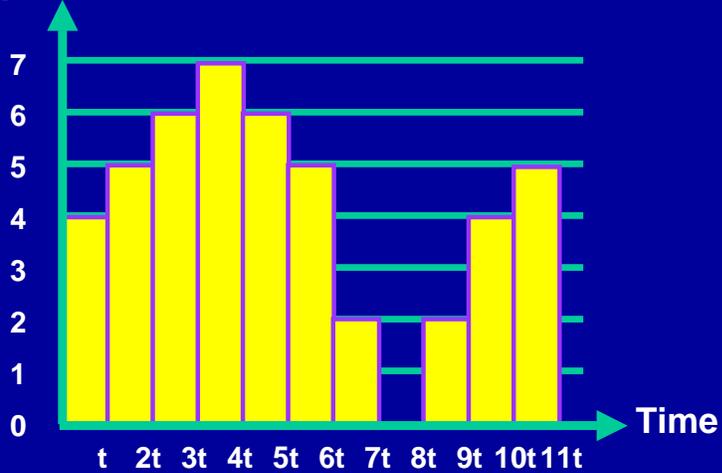
[32] The top signal is fully represented by the digital signal shown in the bottom of the slide. That digital signal represents our analog signal.

Converting Digital Signals to Analog Signals

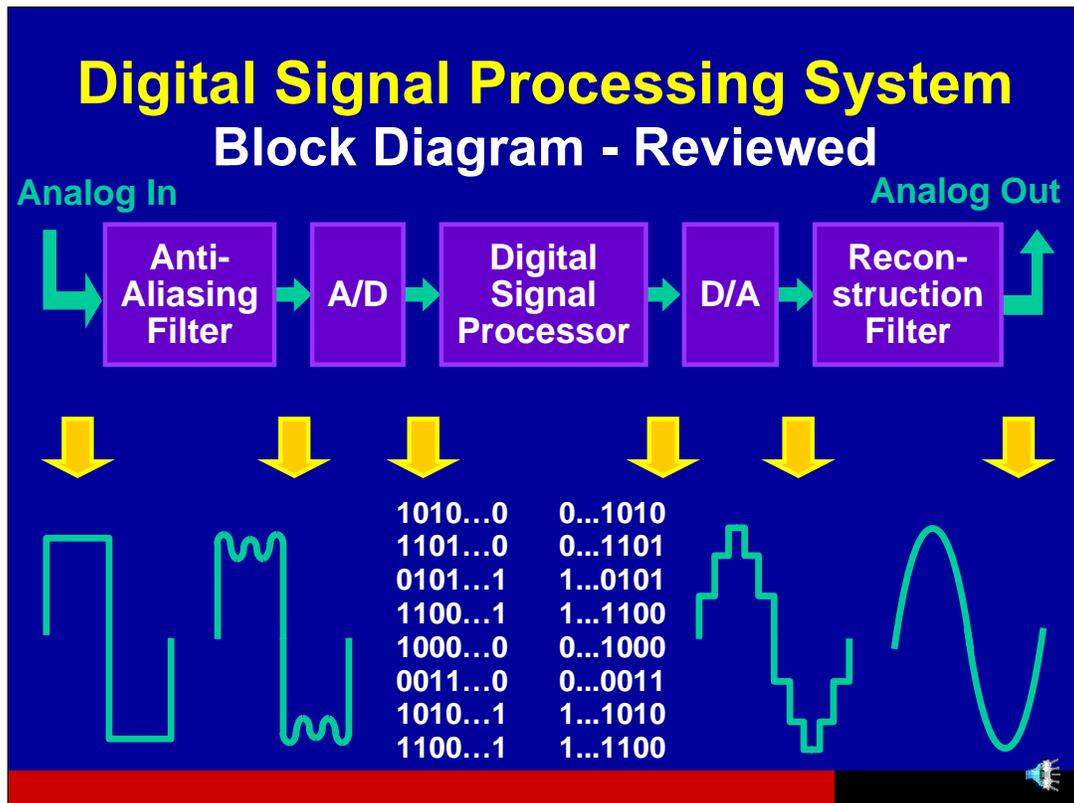
100 = 4
101 = 5
110 = 6
111 = 7
110 = 6
101 = 5
010 = 2
000 = 0
010 = 2
100 = 4
101 = 5



Amplitude



[33] Converting the signal back from a digital signal to an analog signal is a very simple process. All we do is find out our amplitude levels and hold it for a specific unit of time. Looking at this figure, it looks like a staircase. It has a lot of high energy levels, or a lot of high frequency components associated with it. We put a smoothing filter or a reconstruction filter to smooth it off.



[34] Next, we will review the block diagram, except this time, we will put a signal into our block diagram. The input signal will be a square wave. As we stated earlier, any signal can be made up by adding up simple sine waves. A square wave is no exception. By adding up the odd harmonics of sine waves, we can create a square wave.

[35] Once I take that square wave and feed it through the anti-aliasing filter, I've already eliminated some of the upper frequency components. As a result, I have a square wave that is a little bit sloppier. On the top and bottom, you can see the waveforms have little squiggles in them. That is then fed into the analog-to-digital converter to give me a digital representation of that analog signal.

[36] In the digital signal processing block, I process the signal, and output my results to a digital-to-analog converter. The output results have a lot of high energy levels as you can see from that staircase. I smooth it off with a reconstruction filter, and output my waveform.

Digital Signal Processing

Presentation Outline

- Introduction
- Signal Fundamentals
- Signal Processing System
- Signal Processing Applications
 - Filtering
 - Spectral Analysis
 - Other Topics
- Summary



[37] The next area we will look at is signal processing applications.

Signal Processing Applications

- Limitations of Analog Systems
- Advantages of Digital Systems
- Filtering
 - Digital Filtering
 - ◆ Finite Impulse Response (FIR)
 - ◆ Infinite Impulse Response (IIR)
- Spectral Analysis
 - Converting from time domain to frequency domain
 - ◆ FFT
- Other Topics

[38] We'll discuss the limitations of analog systems; the advantages of digital systems; filtering; spectral analysis; and other topics.

[39] When you look at most signal processing applications, they usually fall into one of two different areas of filtering or spectral analysis.

Limitations of Analog Systems

- **Stability problems**
 - Component aging (values change over time)
 - Components sensitive to environmental conditions (thermal aging)
- **Predictability problems**
 - Component to component variations
- **Accuracy problems**
 - Internally generated noise
- **Difficult to make system characteristic changes**
- **Board layout considerations**
- **Difficult to implement some functions**

[40] There are many limitations of working with an analog system. The first one is stability problems. Components age and change values over temperature and various environmental conditions. As far as predictability, you can't get two components that are exactly alike. They have to be tweaked and trimmed. We have accuracy problems. Difficult to make system changes. To add changes to our system, we have to remove the components and replace them with different valued components. Board layouts become a consideration we must take into account, as well as it is difficult, sometimes impossible, to implement some functions.

Advantages of Digital Systems

- No component aging (will not drift out of tolerance)
- Not sensitive to environmental conditions
- No adjusting or tweaking components
- Easy to time-share processor among different functions
- Sophisticated functions can be implemented
- More compact design
- More reliable than analog approach

[41] In digital systems, you relieve some of those limitations. To begin with, there is no component aging. All we are concerned about is zeros and ones. We make changes to our system very easily. We change those zeros and ones, also known as coefficients in our system, who are not sensitive to any environmental conditions. There is no need for adjusting or tweaking of any components. It is much easier to time-share systems. Sophisticated functions that were almost impossible to implement in the analog world can be implemented easily in the digital world. It is a more compact design. It becomes a more reliable system.

Signal Processing Applications

Filtering

smooths, removes noise from, selects particular signal components from, or predicts future values of incoming signal

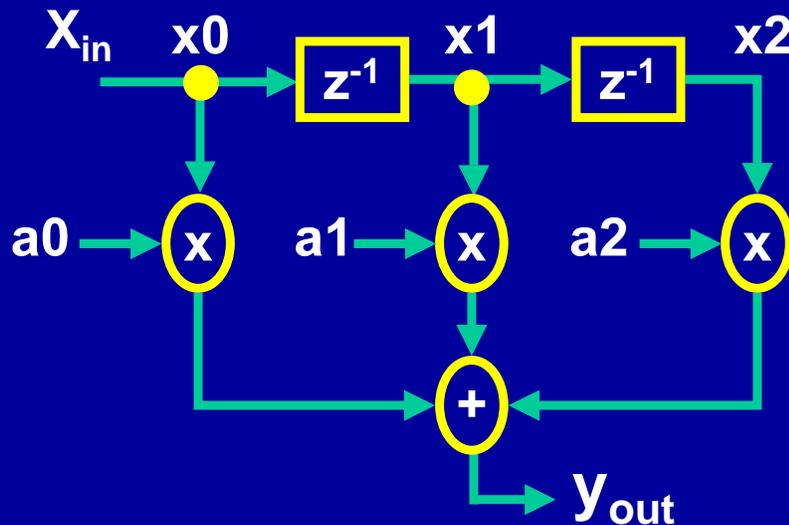
Digital Filters:

Finite Impulse Response (FIR)

Infinite Impulse Response (IIR)

[42] Next, we need to look at filtering. Filters are used to smooth or remove noise, or find a particular frequency component of interest within a signal. Digital filters can be divided into two broad categories: finite impulse response filters, or FIRs, and infinite impulse response filters, or IIRs.

Fundamental Building Block Digital Filter: FIR Structure

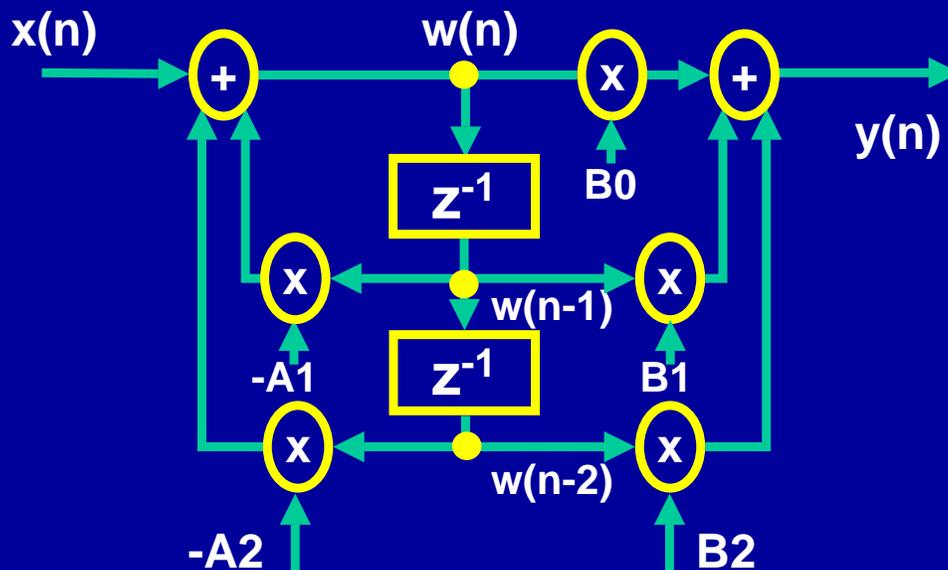


[43] Looking at the structure of a digital filter, a finite impulse response filter, consists of a couple of components. There are blocks with the Xs in them – they are multiplication blocks. There are plus signs which are additions or adds. Then z^{-1} which means wait a unit of time. If we would have samples on this system, they would be at the x_0 's, x_1 's, and x_2 's, x_2 being the oldest sample; x_0 being the newest sample. The a value is a closed coefficient.

[44] I take my first sample in, x_0 , multiply it with a_0 . My next oldest sample is x_1 multiplied with a_1 , and x_2 multiplied with a_2 . Add them together and output the results. When the new sample comes in, the oldest sample falls off the system; x_1 moves over to x_2 's spot; and x_0 moves to x_1 's spot to make room for the new sample. Then the process repeats itself.

[45] If I want to average numbers going through this system, and maybe I selected a_0 , a_1 , and a_2 to equal a third, a third, and a third, I would take the average of the signal coming through. Or a low pass filter. Taking the three numbers, adding them together, and dividing by three, is the same thing as taking each individual number and multiplying it by a third.

Fundamental Building Block Digital Filter: IIR Structure



[46] Looking at the IIR filter, it looks a little more complex than the FIR filter. However, if you look at the right side, it is nothing more than an FIR filter. The left side has feedback associated with it. There are different characteristics associated with FIR and IIR filters. Those characteristics are used to determine which would be proper for a particular application.

Digital Filter Application Example



- Multi-band filters detect the energy in each band
- Detected energy is compared with predetermined statistics of particular signal signature

[47] As an example of a digital filter, let's look at a 5-band filter bank. Taking an analog signal in, digitizing it through our A/D converter, then filtering out various frequency components, and using a logic section to determine the amplitudes that match the specific frequency characteristics of the signal we're looking for.

[48] As an example, breaking glass. If we were to break a sheet of glass and we were interested in specific frequency components, we would filter out those components, determine the amplitude levels, and then, through our decision logic, decide was the glass actually broken.

Signal Processing Applications

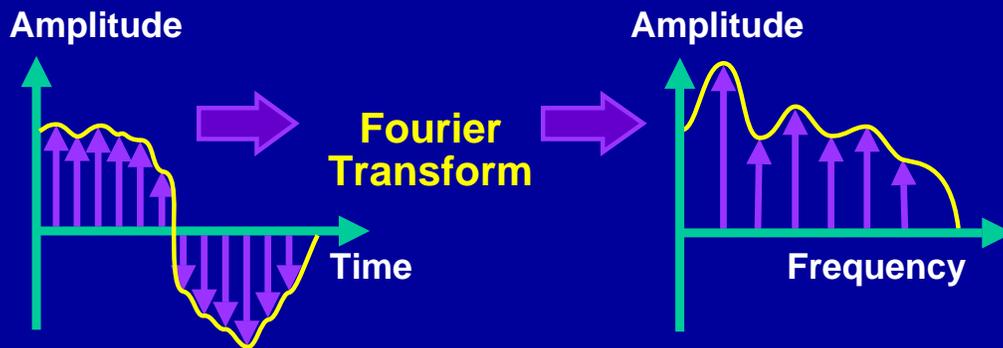
Spectral Analysis

Determines the weights corresponding to each frequency in the spectrum

The FFT:
Converting from Time Domain to
Frequency Domain

[49] Next, we'll look at spectral analysis. Spectral analysis determines the weights corresponding to each frequency in the spectrum. We will also look at something called the "FFT," or converting from the time domain to the frequency domain.

Converting from Time Domain to Frequency Domain



[50] In converting signals from time to frequency, for example, taking the input of a signal on the left side, we look at the signal in the time domain – amplitude versus time; process it through a Fourier transform; then output our results in the frequency domain, or amplitude versus frequency.

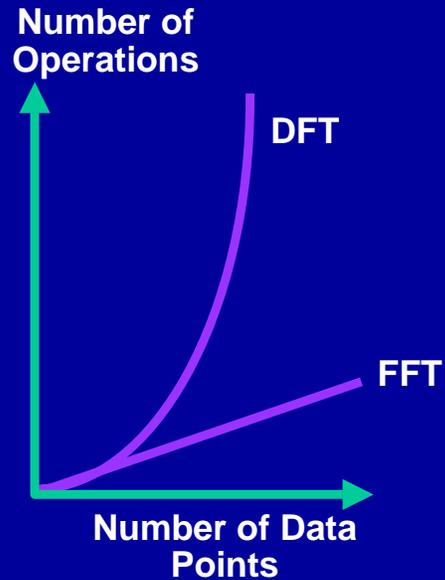
Fast Fourier Transform

- **FFT**
 - An efficient algorithm for computing the DFT (Discrete Fourier Transform); it is used to transform data samples between the Time Domain and Frequency Domain
- **DFT**
 - requires N^2 number of complex arithmetic operations
- **FFT**
 - requires $N \log_2 N$ number of complex arithmetic operations

[51] A Fast Fourier Transform is nothing more than an algorithm that implements a discrete Fourier Transform. In fact, it is an algorithm that gives us the exact results the Discrete Fourier Transform would give us. However, using the FFT reduces the number of complex arithmetic operations we need to perform to get the same results.

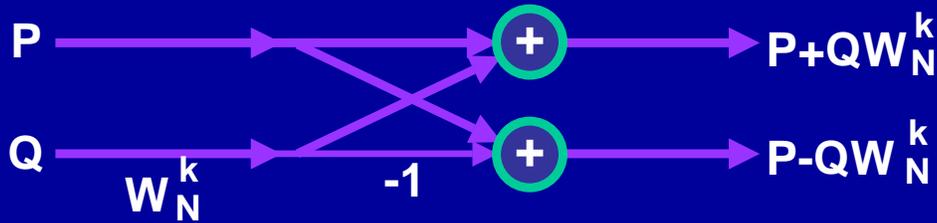
Fast Fourier Transform

N	DFT	FFT
16	256	64
32	1,024	160
64	4,096	384
128	16,384	896
256	65,536	2,048
512	262,144	4,608
1,024	1,048,576	10,240
2,048	4,194,304	22,528
4,096	16,777,216	49,152



[52] As an example, let's take 256 points in the time domain and convert them over to the frequency domain. If I was to use a Discrete Fourier Transform, it would take me 65,000 complex operations to perform this function. However, if I was to use a Fast Fourier Transform, it would take about 2000 complex operations – a lot more efficient. We can do it much quicker using a Fast Fourier Transform and get the same exact results.

Fundamental Building Block: Fast Fourier Transform

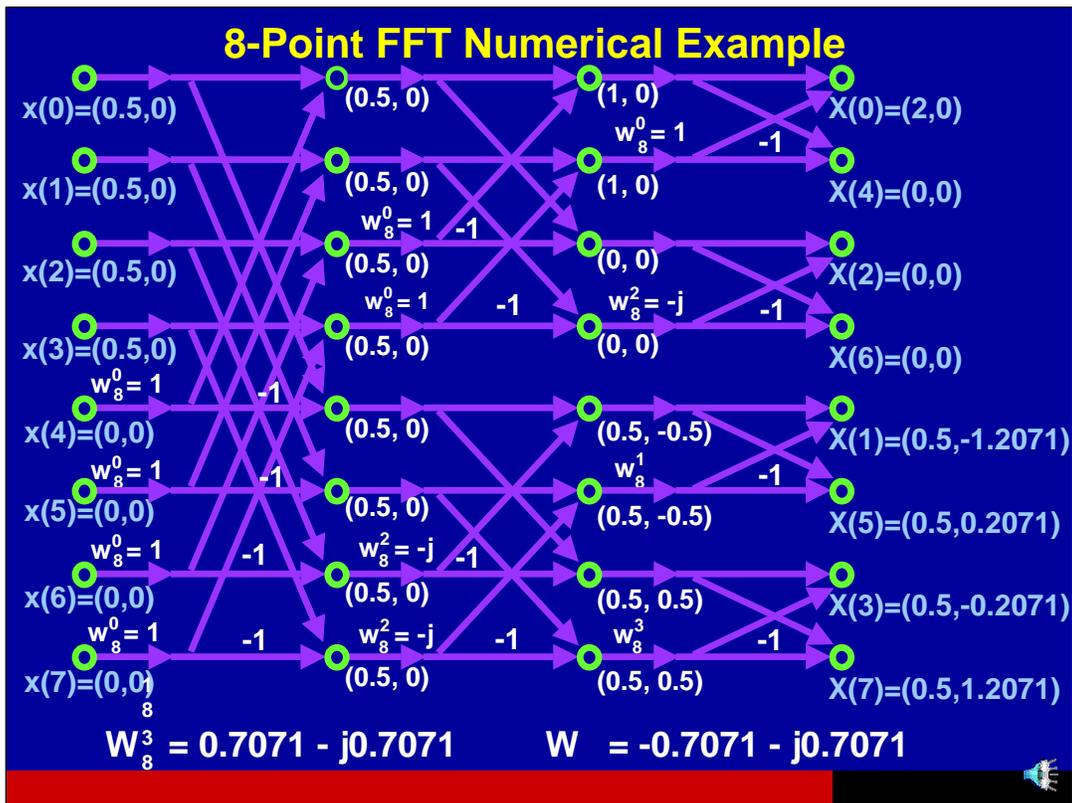


Radix - 2 DIT FFT Butterfly

$$W^k = e^{-j(2\pi/N)k} = \cos(x) - j\sin(x)$$

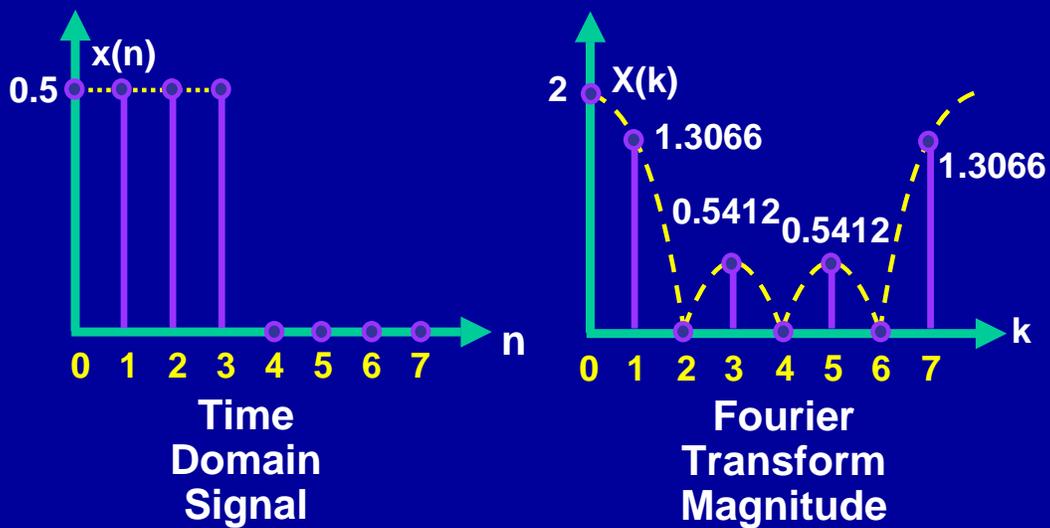
$$\text{where } x = (2\pi/N)k \text{ and } j = \sqrt{-1}$$

[53] Looking at the fundamental building block of the Fast Fourier Transform, we would place values on the left side in time, process them through the Fast Fourier Transform block, and output our values on the right side in terms of frequency.



[54] Taking the structure and expanding it further to an 8-point FFT, let's look at this example. On the input, we put a real square wave coming in. That is .5, .5, .5, .5, 0, 0, 0, 0 in amplitude. On the output, we have our frequency values. If you notice, there are two numbers associated on the output. These are complex numbers. They consist of a real and imaginary part. By using complex numbers, we can also capture the phase relationship of the sine waves.

Time Domain Signal and Magnitude FFT



[55] Taking the output and plotting the magnitude response, we will find out that that square wave was made up of the odd harmonics of sine waves. Additionally, you'll notice that in the frequency domain, we see a mirror image of the first half reflected back in the second half. Therefore, for the number of points I'm putting into my FFT in this example, I'm getting half the number of points of useful information on the output in the frequency domain.

Fast Fourier Transform Fundamentals and Characteristics

- **FFT is an algorithm to implement the DFT**
 - It is an algorithm and not an approximation, therefore the reduction of computations is not made at a reduction of accuracy
- **FFT has two complex data input arrays (real and imaginary)**
- **Imaginary input array can be zero filled if not used**

[56] FFT is nothing more than an algorithm that implements the Discrete Fourier Transform, but it gives us the exact results. FFTs are much more efficient in handling conversions from time to frequency. FFTs also have two complex data inputs, however, if we are looking at real signals, we can zero fill the imaginary portion.

Fast Fourier Transform Fundamentals and Characteristics

- Output data is complex (two arrays)
- Frequency spectrum output covers half the sampling frequency
- Only first half output data is used (second half is a mirror image)
- Output frequency resolution = sampling rate/number of points

[57] The output always consists of two arrays. That gives us the phase information.

[58] The frequency spectrum output covers one-half the sampling frequency. We've already discussed sampling frequency earlier. Only the first half gives us useful information. As we've pointed out, the second half is a mirror image.

[59] The output frequency resolution is equal to the sampling rate divided by the number of points. If we wanted to get a finer picture in the frequency domain, we just increase the number of points, but this is at the expense of complication and time.

FFT Application Example Block Diagram



- **Fast Fourier Transform performs analysis of signal in the Frequency Domain**
- **Detected frequencies are compared with predetermined spread of frequency variations over time to check for a match**

[60] Looking at that same example of breaking glass using a Fast Fourier Transform, we take our signal, convert it to a digital signal through our analog-to-digital converter block, process it through the FFT, and compare it to a bunch of templates. Here, we are comparing the glass breaking, and instead of using a 5-band filter bank, we are using a Fast Fourier Transform, and I can have, perhaps, 256 points, 1024 points, 64 points, to give us the picture in the frequency domain.

Signal Processing Applications

Other Topics



[61] Let's take a look at other topics.

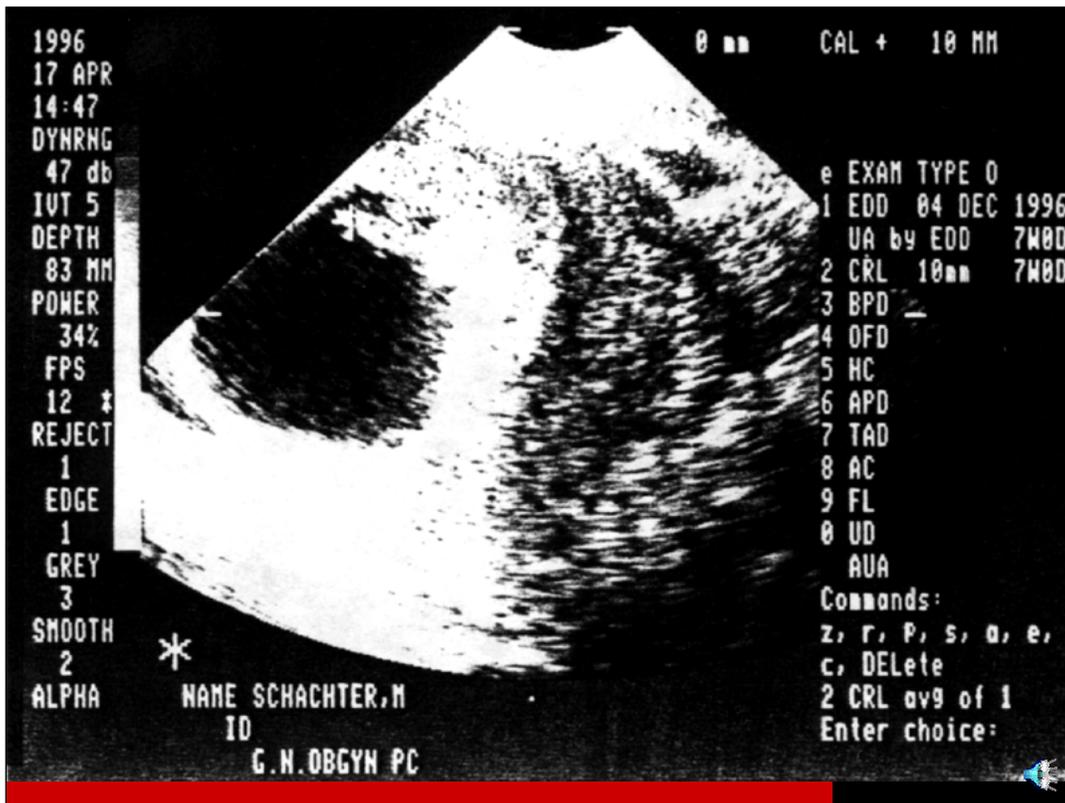
Typical Applications of Signal Processing

- General-Purpose DSP
- Graphics / Imaging
- Instrumentation
- Voice / Speech
- Control
- Military
- Telecommunications
- Automotive
- Consumer
- Industrial
- Medical

[62] Everything we've heard to this point in terms of time, frequency, sampling period, sampling frequency, filtering, and spectral analysis, applies to all these general areas of signal processing. Perhaps we just want to filter out noise from an audio signal. Filtering we've already looked at. We might want to look at signals in the frequency domain. Voice and speech processing makes use of speech recognition, for example, of taking a signal, looking at the frequency components, and matching it to find out the speaker.

[63] Control systems use signal processing. Telecommunications – many times we'll find out there might be an echo on the line. If we take our signal processing and model the echo, but invert it and feed it back, we know from previous examples those signals will cancel each other out. So we have echo cancellation.

[64] As for medical applications, let's look at a real world example.



[65] Ultrasonography. In ultrasonography, we take a sound wave, bounce it off an object, take the reflections, and form a two-dimensional picture.

Characteristics of a Digital Signal Processor

- **Fast Instruction Cycle Time**
 - real-time processing
- **Dedicated Hardware Multiplier**
 - single cycle multiplication
 - ◆ (rather than consecutive additions)
- **Multiple Bus Architecture**
 - large number of simultaneous inputs and outputs
 - ◆ (avoids bottleneck in processing)

[66] Up to this point, we've looked at signal processing, but we'd like to touch on the characteristics of a signal processor. Since we know a little about signal processing and want to work with real-time signals, we want things to be fast, we're interested in digitized signals – let's look at how that all feeds together into a digital signal processor.

[67] First, which is obvious, we want a fast instruction cycle time. We want to perform things in real time. The next thing we've noticed on the various block diagrams, there are a lot of multiplications, so we need to perform multiplications very quickly, so we have dedicated hardware multipliers on the signal processor to perform single-cycle multiplications.

[68] We've added multiple buses on board as well to perform operations in parallel.

Characteristics of a Digital Signal Processor

- **Extensive Pipelining**
 - executes parts of several instructions in a single cycle
- **Special Instructions**
 - combines several operations into a single cycle

[69] In addition, we have an extensive pipelining and special instructions tailored for handling signal processing applications.

Digital Signal Processing

Presentation Outline

- Introduction
- Signal Fundamentals
- Signal Processing System
- Signal Processing Applications
 - Filtering
 - Spectral Analysis
 - Other Topics
- Summary

[70] Next, we'll look at our summary.

Summary

- Future Directions
- Conclusion

[71] Our summary consists of future directions and a conclusion.

Future Directions

- **Faster Processor Speeds**
 - implement more functions in a given amount of time
- **Higher Levels of Integration**
 - implement more sophisticated and complex functions
- **More Powerful Instructions**
 - more operations in a given processor cycle

[72] We can expect to see faster processor speeds in the future. Faster processor speed translates into implementing more functions in a given amount of time. Also, we'll see higher levels of integration as well as more powerful instructions.

Future Directions

- **More Advanced Algorithms**
 - drive more advanced device architectures
- **New Applications**
 - more DSPs being used in non “classical” DSP systems

[73] And, we'll see more advanced algorithms being developed. It's these advanced algorithms that will drive directions of our future architectures.

[74] In addition, we're finding newer applications for DSPs. They are not just going into the classic signal processing systems we've looked at, but into other areas that require high-performance microprocessors.

Conclusion

**Signal Processing is the
Past, Present, and Future**

[75] In conclusion, signal processing has been our past; it is our present; and it will be our future. Thank you very much.

Copyright Page

The World Leader in DSP Solutions

