

# Audio and Other Waveforms

Stephen A. Edwards

Columbia University

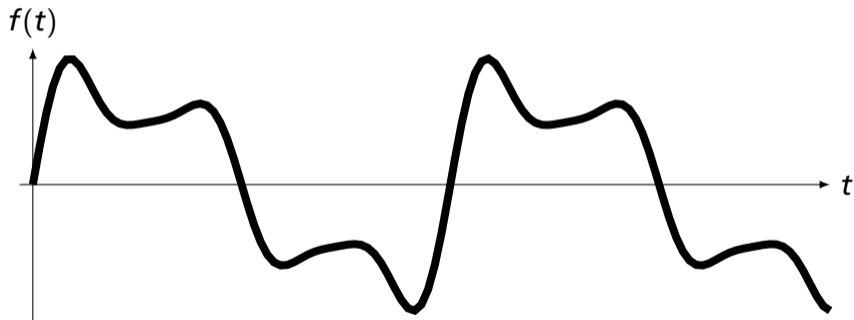
Spring 2020

# Waveforms

Time-varying scalar value

Commonly called a “signal” in the control-theory literature

Sound: air pressure over time



Raster video: brightness over time

Speed over time, position over time, etc.

# The Fourier Series

*Any periodic function can be expressed as a sum of harmonics*

For a smooth function  $f(t)$  with period  $T$ , i.e.,

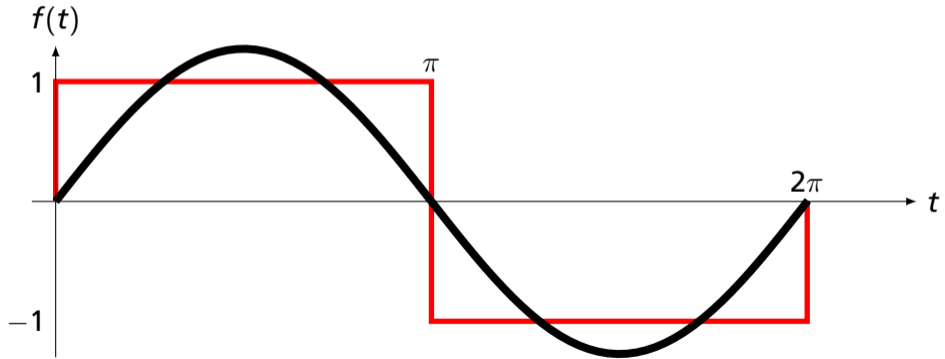
$$f(t) = f(t + T),$$

there exists coefficients  $a_n, b_n$  such that

$$f(t) = a_0 + \sum_{m=1}^{\infty} a_m \cos \frac{2\pi mt}{T} + b_m \sin \frac{2\pi mt}{T}$$

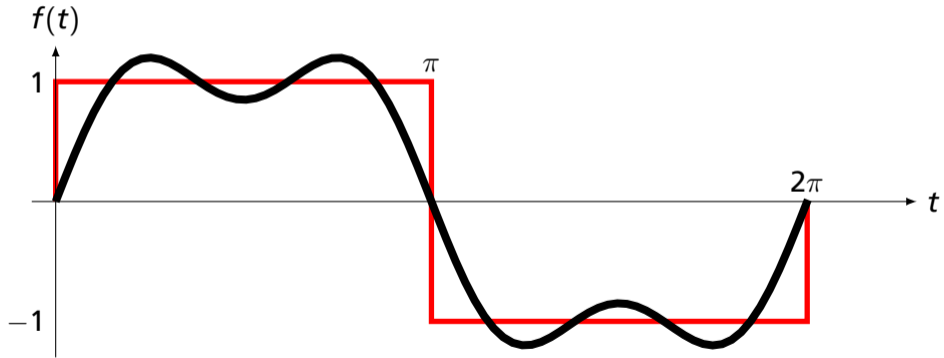
# The Fourier Series for a Square Wave

$$f(t) = \frac{4}{\pi} \left( \sin t \right)$$



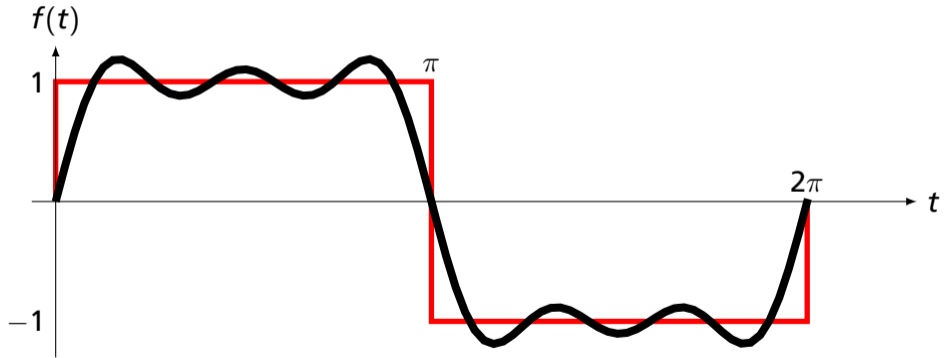
# The Fourier Series for a Square Wave

$$f(t) = \frac{4}{\pi} \left( \sin t + \frac{\sin 3t}{3} \right)$$



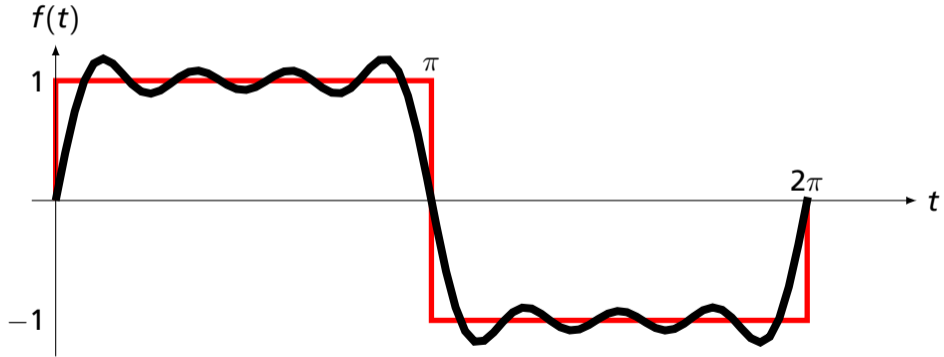
# The Fourier Series for a Square Wave

$$f(t) = \frac{4}{\pi} \left( \sin t + \frac{\sin 3t}{3} + \frac{\sin 5t}{5} \right)$$



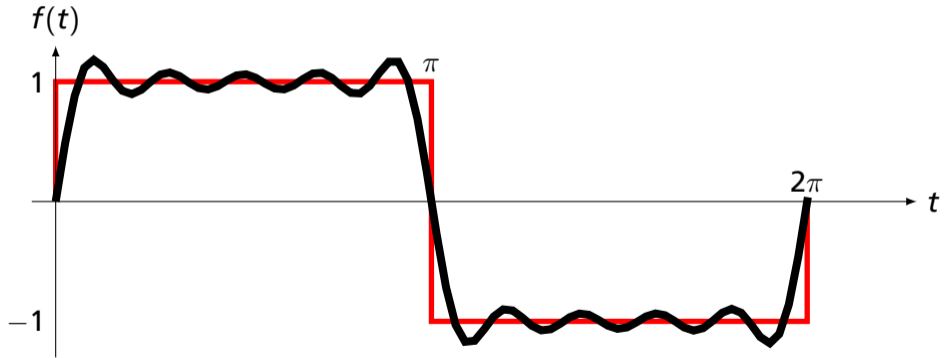
# The Fourier Series for a Square Wave

$$f(t) = \frac{4}{\pi} \left( \sin t + \frac{\sin 3t}{3} + \frac{\sin 5t}{5} + \frac{\sin 7t}{7} \right)$$



# The Fourier Series for a Square Wave

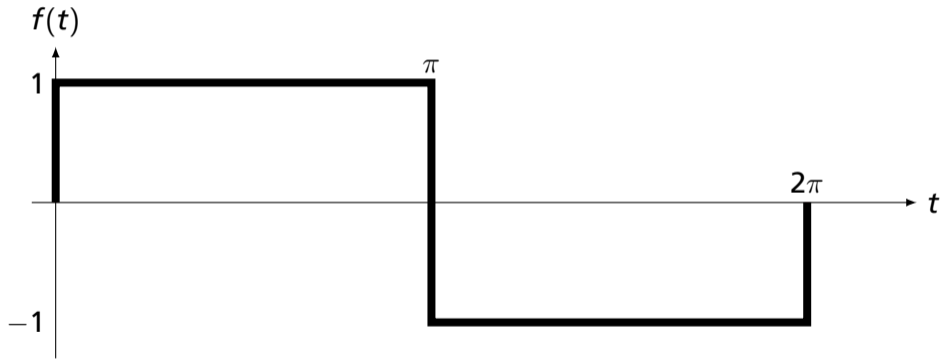
$$f(t) = \frac{4}{\pi} \left( \sin t + \frac{\sin 3t}{3} + \frac{\sin 5t}{5} + \frac{\sin 7t}{7} + \frac{\sin 9t}{9} \right)$$





# The Fourier Series for a Square Wave

$$f(t) = \frac{4}{\pi} \left( \sin t + \frac{\sin 3t}{3} + \frac{\sin 5t}{5} + \frac{\sin 7t}{7} + \frac{\sin 9t}{9} + \dots \right)$$

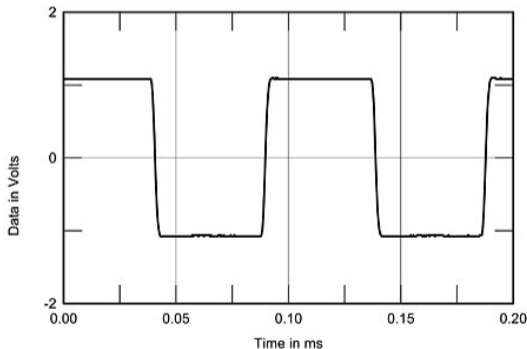


## Bandwidth-Limited Signals

Basic observation: nothing changes infinitely fast

Bounding the rate of change sets the bandwidth of a signal

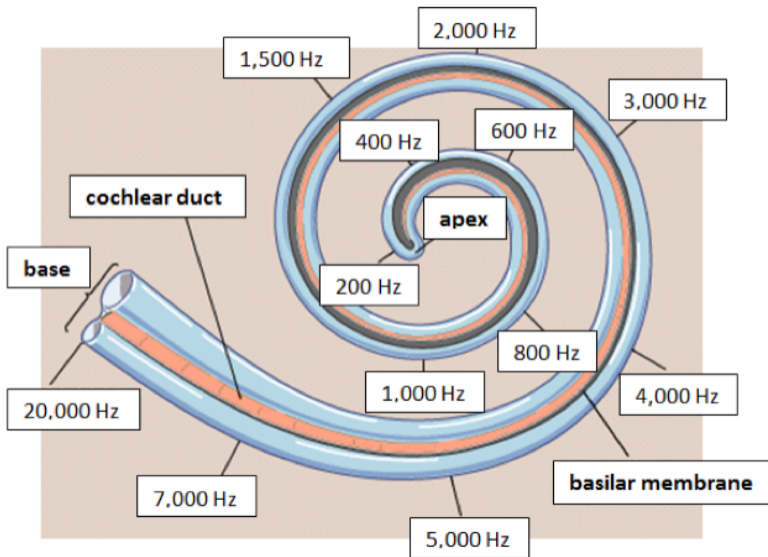
Hertz or Hz: "per second"



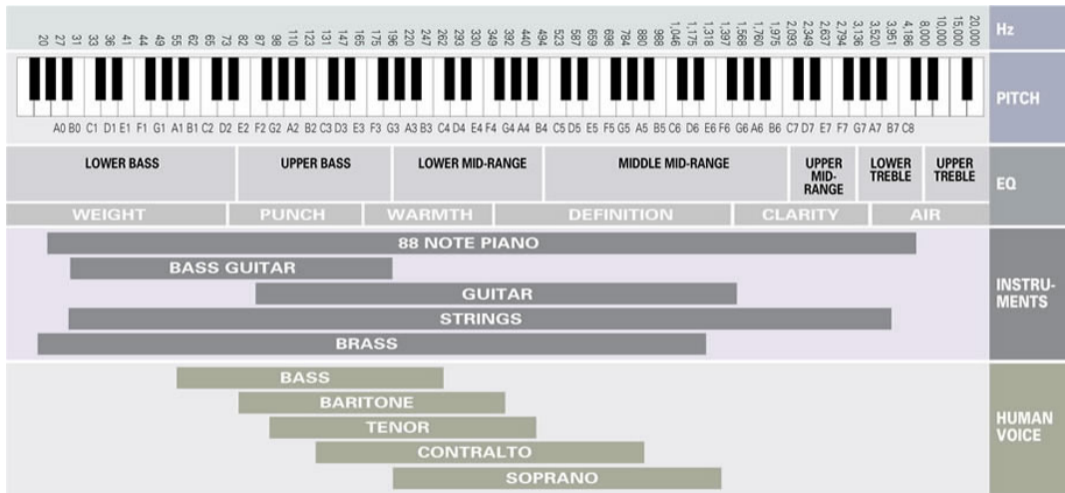
Source: Stereophile magazine: Marantz SM-11S1, A \$4000 audiophile amplifier rated 5 Hz–120 kHz. Small-signal 10kHz squarewave into 8 ohms.

# The Bandwidth of Sound

Human ears are almost a Fourier transform



# Human Hearing

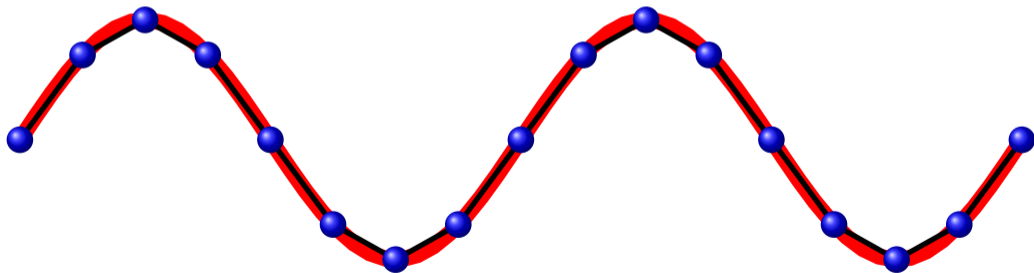


Empirically, humans hear 20 Hz–20 kHz

Highest frequency limit tends to decrease with age

# Nyquist Theorem

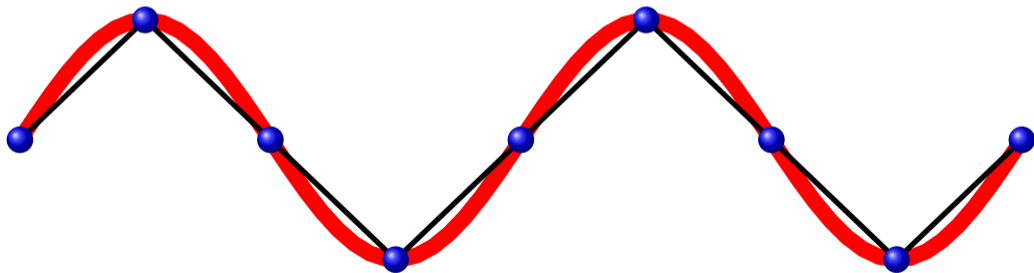
To reconstruct a bandwidth-limited signal from samples, you need to sample at least twice the maximum frequency.



Sampling at  $8 \times f$

# Nyquist Theorem

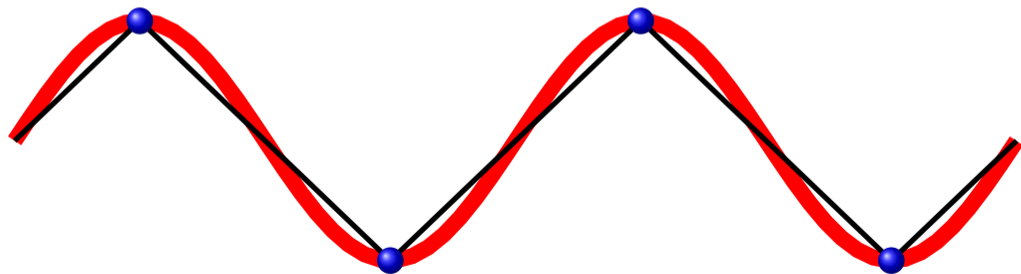
To reconstruct a bandwidth-limited signal from samples, you need to sample at least twice the maximum frequency.



Sampling at  $4 \times f$

# Nyquist Theorem

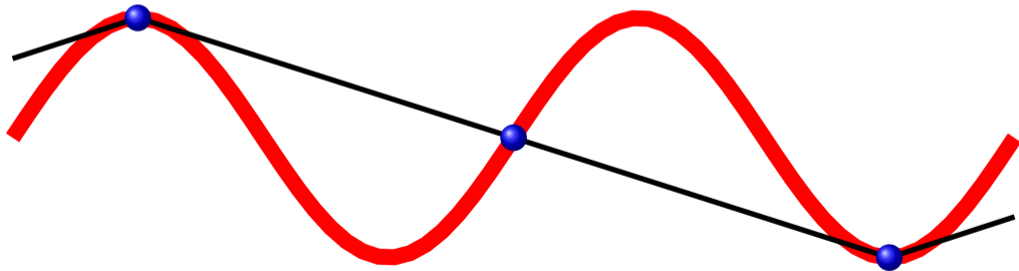
To reconstruct a bandwidth-limited signal from samples, you need to sample at least twice the maximum frequency.



Sampling at  $2 \times f$

# Nyquist Theorem

To reconstruct a bandwidth-limited signal from samples, you need to sample at least twice the maximum frequency.

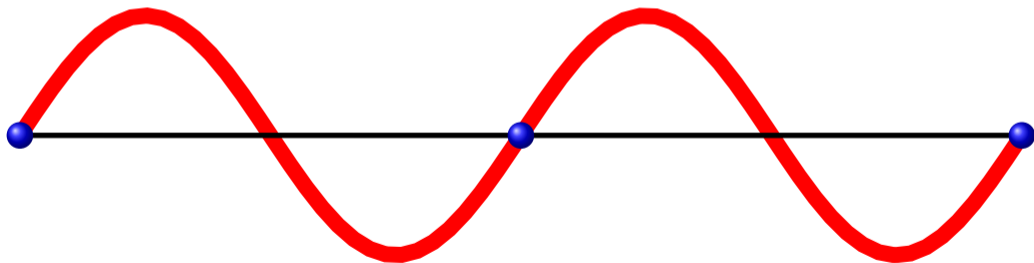


Sampling at  $\frac{4}{3} \times f$



# Nyquist Theorem

To reconstruct a bandwidth-limited signal from samples, you need to sample at least twice the maximum frequency.

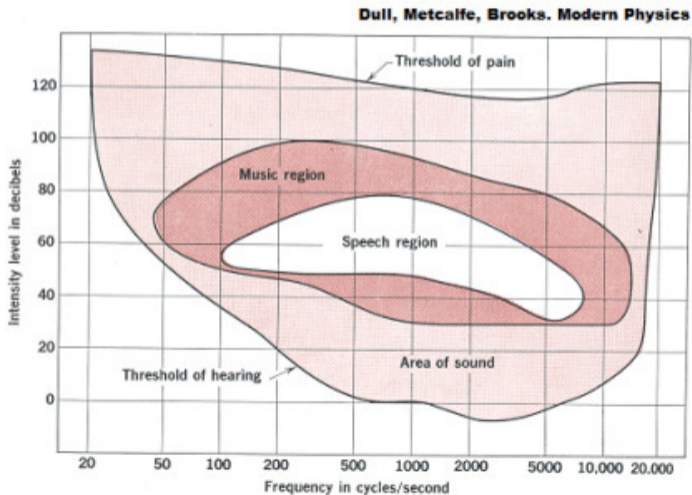


Sampling at  $1 \times f$

# Audio Sampling Rates

CD-quality audio: 44.1 kHz

Telephone-quality audio: 8 kHz



## Signal-to-Noise Ratio

*You can't always get what you want  
But if you try sometimes you might find  
You get what you need*

—The Rolling Stones

Signals are never pure: there's always something that makes them deviate from the ideal.

Signal-to-Noise ratio:

$$SNR = \frac{\text{Signal Power}}{\text{Noise Power}}$$

Usually measured using a log scale, i.e.,

$$dB = 10 \log_{10} \frac{P_{\text{signal}}}{P_{\text{noise}}}$$

## Human Hearing dB, SNR, and bits

<b>Colt .45 Pistol (25 feet)</b>	<b>140</b>
<b>Threshold of Pain</b>	<b>130</b>
	<b>120</b>
<b>Underground Train</b>	<b>110</b>
	<b>100</b>
<b>Average Home Hi-Fi Level</b>	<b>90</b>
<b>Average Factory</b>	<b>80</b>
	<b>70</b>
<b>Average Conversation</b>	<b>60</b>
	<b>50</b>
<b>Average Office</b>	<b>40</b>
<b>Residential Ambient Noise</b>	<b>30</b>
	<b>20</b>
<b>Quiet Whisper (5 feet)</b>	<b>10</b>
<b>Threshold of Hearing</b>	<b>0</b>
<b>0.0002 Dyne/Sq. cm</b>	

$$n \times 6.02 + 1.76 = \text{SNR in dB}$$

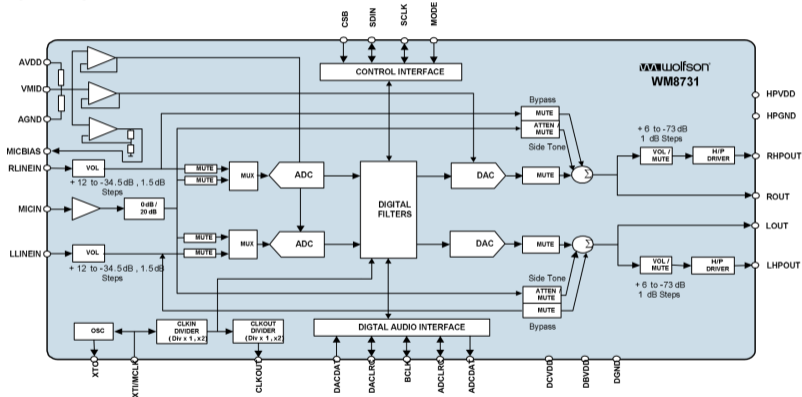
CD samples:

16 bits = 98 dB

Near the limit of human hearing

# The CODEC on the DE1-SoC: Wolfson WM8731

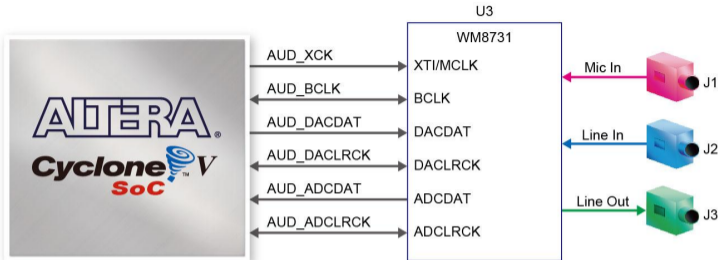
enCOder/DECOder: analog-to-digital converter (ADC) + digital-to-analog converter (DAC)



Two 24-bit ADCs; two 24-bit DACs + headphone amp

Sampling rates: 8 kHz – 96 kHz, 16–24 bit words

# DE1-SoC Interface to the Audio Codec



**Pin Assignment of Audio CODEC**

Signal Name	FPGA Pin No.	Description	I/O Standard
AUD_ADCLCK	PIN_K8	Audio CODEC ADC LR Clock	3.3V
AUD_ADCDAT	PIN_K7	Audio CODEC ADC Data	3.3V
AUD_DACLCK	PIN_H8	Audio CODEC DAC LR Clock	3.3V
AUD_DACDAT	PIN_J7	Audio CODEC DAC Data	3.3V
AUD_XCK	PIN_G7	Audio CODEC Chip Clock	3.3V
AUD_BCLK	PIN_H7	Audio CODEC Bit-stream Clock	3.3V
I2C_SCLK	PIN_J12 or PIN_E23	I2C Clock	3.3V
I2C_SDAT	PIN_K12 or PIN_C24	I2C Data	3.3V

I<sup>2</sup>C bus for configuration: data format, volume levels, etc.  
Synchronous serial protocol (data + L/R + bit clock) for data

# WM8731 Serial Protocol

One of four communication modes, set by I<sup>2</sup>C registers:

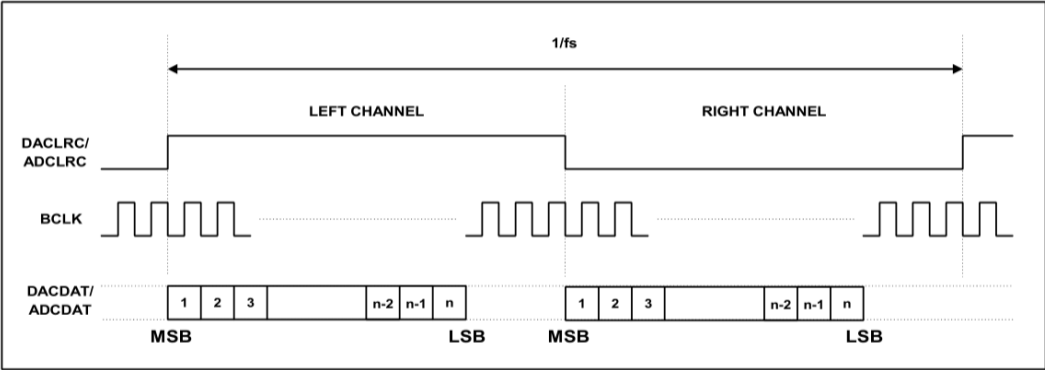
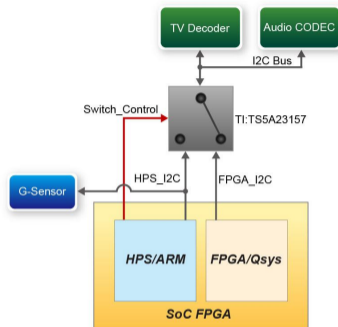


Figure 26 Left Justified Mode

# DE1-SoC I<sup>2</sup>C Multiplexer



Control mechanism for the I<sup>2</sup>C multiplexer

## Pin Assignment of I<sup>2</sup>C Bus

<b>Signal Name</b>	<b>FPGA Pin No.</b>	<b>Description</b>	<b>I/O Standard</b>
FPGA_I2C_SCLK	PIN_J12	FPGA I2C Clock	3.3V
FPGA_I2C_SDAT	PIN_K12	FPGA I2C Data	3.3V
HPS_I2C1_SCLK	PIN_E23	I2C Clock of the first HPS I2C concontroller	3.3V
HPS_I2C1_SDAT	PIN_C24	I2C Data of the first HPS I2C concontroller	3.3V
HPS_I2C2_SCLK	PIN_H23	I2C Clock of the second HPS I2C concontroller	3.3V
HPS_I2C2_SDAT	PIN_A25	I2C Data of the second HPS I2C concontroller	3.3V



## Storing Waveforms

If you store each sample,

$$\frac{\text{samples}}{\text{second}} \times \frac{\text{bits}}{\text{sample}} \times \text{channels} = \frac{\text{bits}}{\text{second}}$$

Total memory consumption:

$$\frac{\text{bits}}{\text{seconds}} \times \text{seconds} = \text{bits}$$

E.g., CD-quality audio: 44.1 kHz, 16 bits/sample, 2 channels

$$44.1 \text{ kHz} \times 16 \times 2 = 1.4 \text{ Mbps} = 175 \text{ KB/s}$$

A 74-minute CD:

$$1.4 \text{ Mbps} \times 60 \frac{\text{seconds}}{\text{minute}} \times 74 \text{ minutes} \times \frac{\text{byte}}{8 \text{ bits}} = 783 \text{ MB}$$

## Reducing Memory: Sample Less; Use Fewer Bits

74 minutes of CD-quality audio

(16 bits/sample, stereo, 44.1 kHz)

$$44.1 \text{ kHz} \times 32 \text{ bits} \times 60 \text{ sec/min} \times 74 \text{ min} \div 8 \text{ bits/byte} = 783 \text{ MB}$$

74 minutes of telephone-quality audio:

(8 bits/sample, mono, 8 kHz)

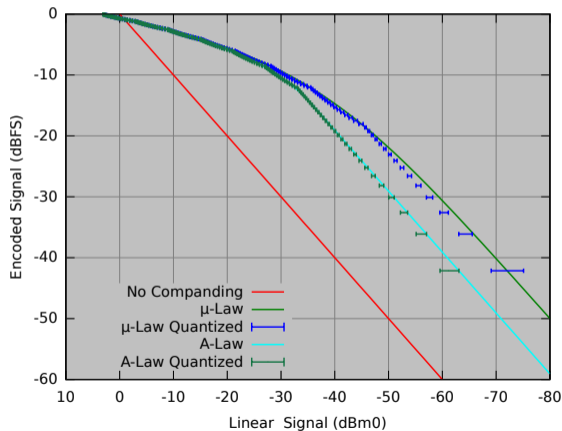
$$8 \text{ kHz} \times 8 \text{ bits} \times 60 \text{ sec/min} \times 74 \text{ min} \div 8 \text{ bits/byte} = 35 \text{ MB}$$

# Reducing Memory: Lossy Compression (Comanding)

$\mu$ -law and A-law compression

Logarithmic encoding of 12 bit samples in 8 bits

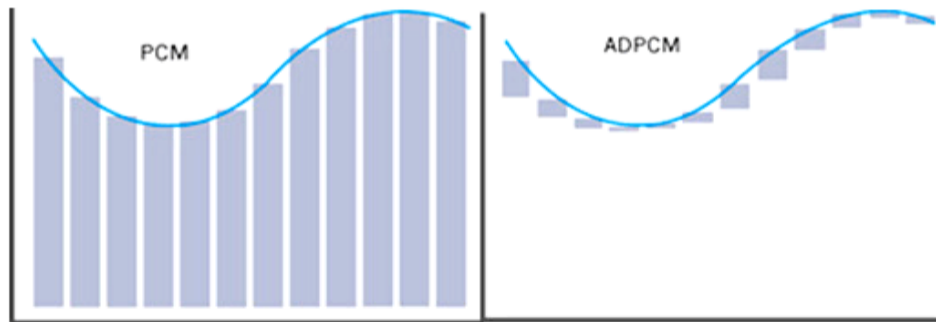
Trades dynamic range for quantization noise



# ADPCM: Adaptive Predictive Pulse Code Modulation

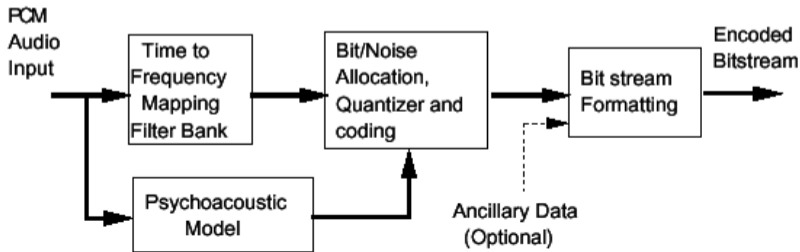
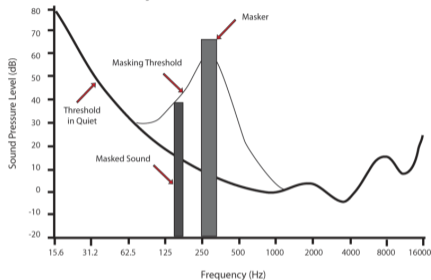
Uses 4 bits/sample to reconstruct 8-bit samples

Encodes the *difference* between the next sample and its predicted value



# MPEG Layer 3 Compression: Perceptual Coding

Carefully reproduce what we hear well and worry less about what we can't (soft sounds masked by loud ones)



# Sound Synthesis: Analog

Modular analog sound synthesis c. 1968

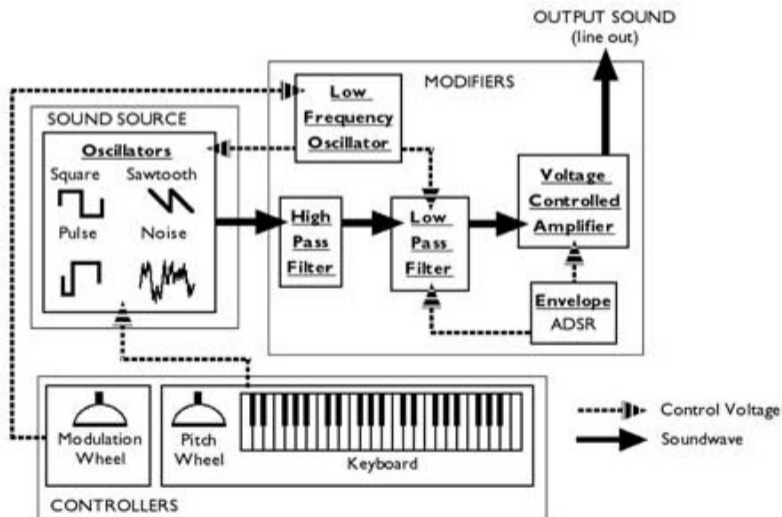
Oscillators + noise sources + envelope generators + filters



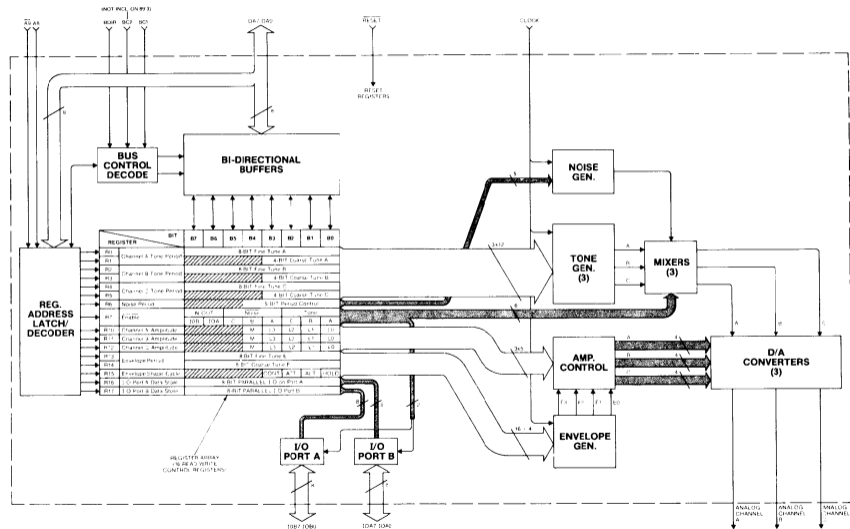
Moog synthesizer

# Subtractive Synthesis

Start with a saw, square, or triangle wave, then filter

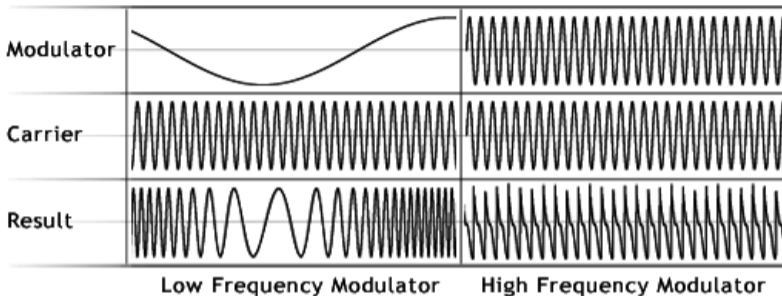


# The AY-3-8912 Programmable Sound Generator





# FM Synthesis



What does it sound like? Any pop music from the 1980s

## Summary of Audio Waveform Generation

- Direct sampling (Pulse Code Modulation)  
Consider sampling frequency, bits/sample
- Lossy Compression  
Comanding ( $\mu$ -law, A-law)  
ADPCM  
Perceptual Coding (MP3 et al.)
- Synthesis  
Subtractive (oscillators, filters, envelopes)  
FM (Carrier  $\times$  modulator, envelopes)  
Wavetable/sampling (sound snippets + note events)

## Representing Images

Same story; two dimensional waveforms

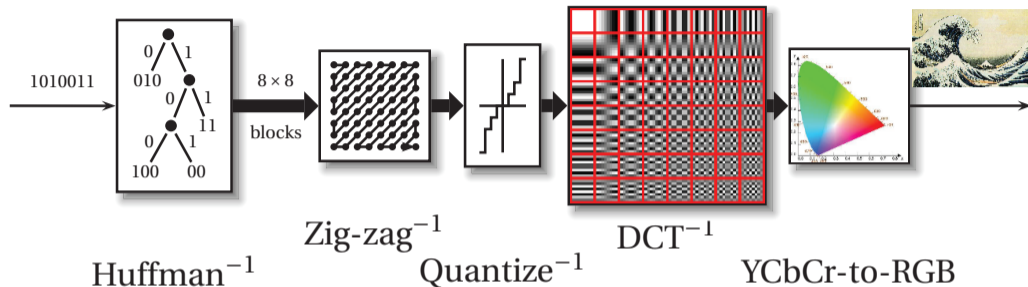
E.g., a single frame of VGA/standard definition television:

$$640 \times 480 \times 24 \frac{\text{bits}}{\text{pixel}} = 900 \text{ KB}$$

HD is terrifying:

$$1920 \times 1080 \times 24 \frac{\text{bits}}{\text{pixel}} = 5.9 \text{ MB}$$

# JPEG: Still Image Compression



Colorspace conversion

Space-to-frequency domain conversion

Quantization

Zig-zag encoding

Huffman encoding