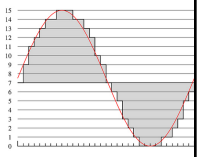
Penn Engineering

ESE



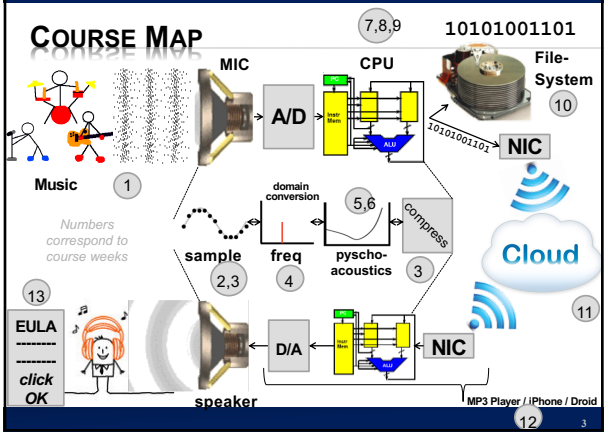
Lecture #2 – A2D

ESE 150 – Digital Audio Basics

Lecture Topics

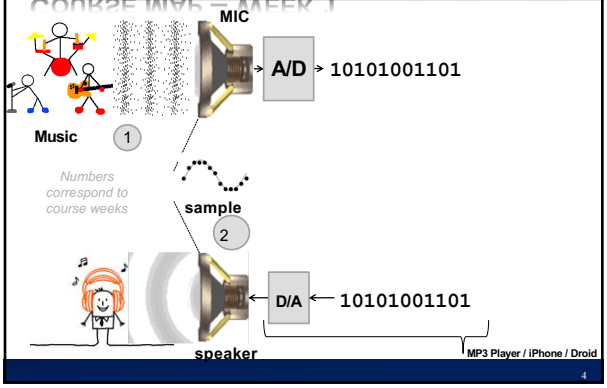
- Where are we on course map?
- Part 1: Sound / Sound Pressure
  - Continuous, discrete, ADC, DAC
- Part 2: Sampling & Quantization
  - Infinite, Continuous signals → Finite, Discrete data in bits

Course Map



Numbers correspond to course weeks

Course Map – Week 1

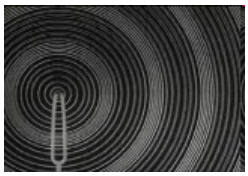


Numbers correspond to course weeks

Sound Waves

Introduction to Sound

- Sound is a pressure wave

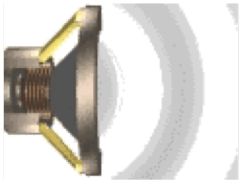


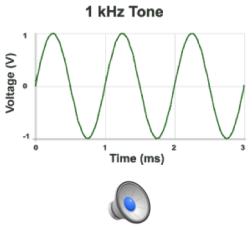
http://www.archive.org/details/SoundWavesAn

Week 1: Introduction to Sound Waves

Cycle = 1 iteration of sine wave
Hertz (Hz) = 1 cycle per second

1kHz = 1000 cycles/s

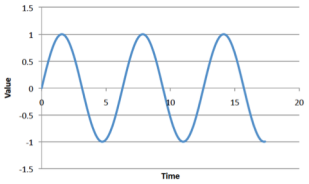




Source: http://www.mediacollege.com/audio/01/sound-waves.html

Preclass 1 and 2

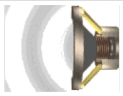
- **Frequency of sine wave?**



ms

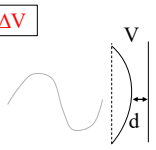
- **Relationship between period and frequency?**

Week 1: Pressure to Voltage



- Microphones convert pressure to voltage
  - (speakers/headphones voltage to pressure)
  - Physical position to voltage

$\Delta d \rightarrow \Delta C \rightarrow \Delta V$



- Reason as parallel place capacitor
  - ESE 112 or PHYS 151

$$C = \frac{\varepsilon A}{d}$$

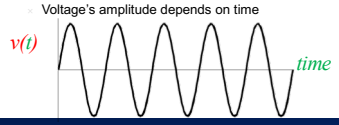
$$Q = CV \qquad V = \frac{Q}{C}$$

Voltage is a function of distance (pressure)

Signals

We need to define some terms

- **What is a signal?**
  - Something that carries information
  - A description of how one parameter depends on another
    - Common Engineering Example:
      - Voltage that varies with time
        - E.g. Amplitude of voltage changes as time moves forward
      - Time = ***independent*** variable (x-axis): time
        - Depends on nothing!
      - Voltage = ***dependent*** variable (y-axis): *v(t)*
        - Voltage's amplitude depends on time

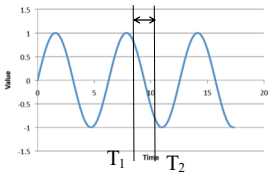


We need to define some terms

- **Most signals encountered in nature…**
  - …are "**continuous**" / analog
    - Continuous range of values (any real #)
    - Examples: 1) Light intensity that changes with distance
    - 2) Voltage that varies over time *v(t)*
      - *We will see in lab this week: MUSIC signal represented with voltage*
    - 3) Chemical reaction rate that depends on temperature
  - as opposed to "**non-continuous**" / discrete signals
    - Only a discrete range of values possible (limited subset of real #s)
    - How a computer must represent signals
      - Fundamental unit of information: **bit**
      - Cannot represent all possible real #'s
      - Uses binary digit (bit) to represent #'s:
        - 1-bit, represents 2 things…2-bits, represents 4 things
        - What's the generalization? (n-bits → how many things?)

Big Question

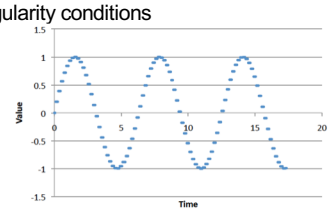
- **How represent and process *continuous* information on a digital computer with *finite* memory?**
  - Note: continuous means signal may take on infinite number of values between any T_1 and T_2



14

Connect the Dots

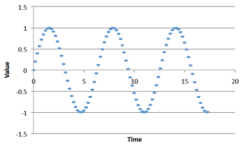
- **Intuition, with enough dots, not hard to "connect-the-dots" to reconstruct (understand) the continuous signal.**
  - What is the continuous signal here? (preclass 3)
  - Assumes certain regularity conditions
  - What is enough?
  - Not unlike calculus
  - Limit as $\Delta x \rightarrow 0$
  - Discrete sum approaches Integral



15

Definitions

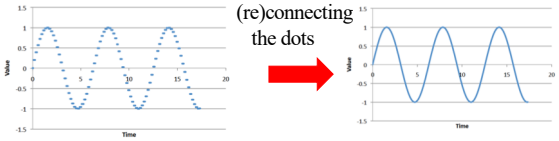
- **Analog-to-Digital (ADC) Conversion**
  - Process of converting *continuous* signal to *discrete* signal
  - Going from analog to digital "domain"
  - Often called: digitization
  - Use a subset of real #'s to represent all real #'s
    - Involves a lot of approximation (lots of room for error!)
- **…collecting** the dots



16

Definitions

- **Digital-to-Analog (DAC) Conversion**
  - Process of converting *discrete* signal to *continuous* signal
  - Going from digital to analog "domain"
  - Converting "bits" to a continuous waveform
    - Our MP3/Music players do this all the time (will do in lab2)



17

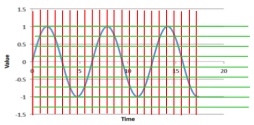
Part 2

Sampling & Quantization

18

ADC – Sampling & Quantization

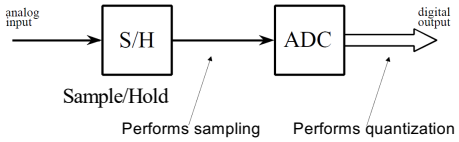
- **Analog-to-Digital (ADC) Conversion**
  - Converting analog (continuous) signal to digital signal
  - Digitization process has two important aspects:
    - *1) Sampling*
      - Converting ***independent*** variable of signal from continuous to discrete
      - e.g.: breaking continuous *time* down into intervals
      - Pick Δx
      - Look at value ever 1 ms



    - *2) Quantization*
      - Converting ***dependent*** variable of signal from continuous to discrete
      - e.g.: breaking continuous *voltage* down into levels
      - Round value to nearest 0.25 volts

19

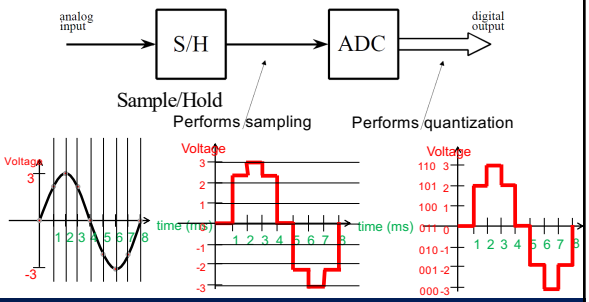
ADC – Broken into two parts



Figures from reading: *The Scientist and Engineer's Guide to Digital Signal Processing*, By Steven W. Smith

20

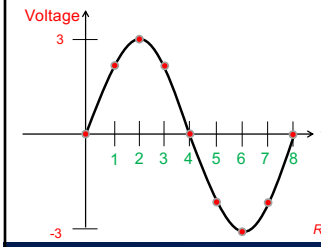
ADC – Broken into two parts



21

ADC – Sampling

- **Analog-to-Digital (ADC) Conversion**
  - ***Sampling***: breaking independent variable (time) into intervals
  - Example: Let's *sample* our continuous signal @ 1 ms intervals:



What is our sampling rate?
(How many samples per second?)

1000 samples per second
1 kiloSample / s

What is frequency of this signal?

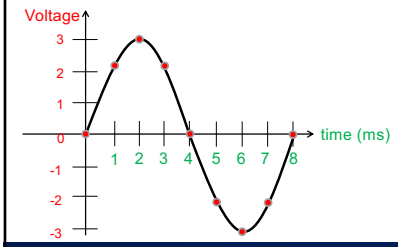
$$freq = \frac{1}{period} = \frac{1}{8ms} = 125\ Hz$$

Spoiler alert:
Relationship between sample rate & frequency!

22

ADC – Quantization

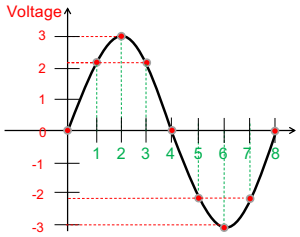
- **Analog-to-Digital (ADC) Conversion**
  - ***Quantization***: breaking dependent variable (voltage) into levels
  - Ex: Let's quantize our range of voltages into 7 levels (1 Volt each)



23

ADC – Sampling & Quantization

- **Analog-to-Digital (ADC) Conversion**
  - **Let's collect our samples at the quantized levels**



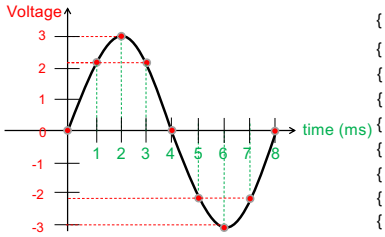
Samples:
{ 0 ms, 0 Volts }
{ 1 ms, 2 Volts }
{ 2 ms, 3 Volts }
{ 3 ms, 2 Volts }
{ 4 ms, 0 Volts }
{ 5 ms, -2 Volts }
{ 6 ms, -3 Volts }
{ 7 ms, -2 Volts }
{ 8 ms, 0 Volts }

Notice, we are rounding! Error is inherent in this process

24

ADC – Digital Conversion / Encoding

- **Analog-to-Digital (ADC) Conversion**
  - **We've converted something continuous into discrete form**
  - **How do we get it to "digital form"?** **We *encode* it…(map to another format**

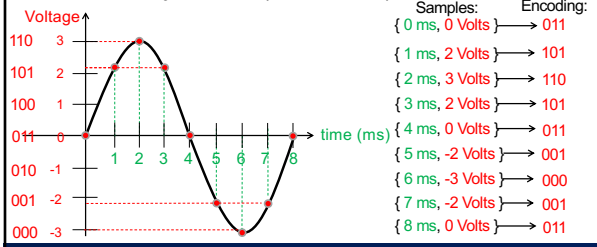


Samples:
{ 0 ms, 0 Volts }
{ 1 ms, 2 Volts }
{ 2 ms, 3 Volts }
{ 3 ms, 2 Volts }
{ 4 ms, 0 Volts }
{ 5 ms, -2 Volts }
{ 6 ms, -3 Volts }
{ 7 ms, -2 Volts }
{ 8 ms, 0 Volts }

25

ADC – Digital Conversion / Encoding

- **Analog-to-Digital (ADC) Conversion**
  - **We have 7 discrete voltages, # of bits to represent 7 things?**
  - **3-bits! Why? $2^{3\text{-bits}} = 8$ (1 unused state)**



Samples:	Binary Encoding:
{ 0 ms, 0 Volts }	→ 011
{ 1 ms, 2 Volts }	→ 101
{ 2 ms, 3 Volts }	→ 110
{ 3 ms, 2 Volts }	→ 101
{ 4 ms, 0 Volts }	→ 011
{ 5 ms, -2 Volts }	→ 001
{ 6 ms, -3 Volts }	→ 000
{ 7 ms, -2 Volts }	→ 001
{ 8 ms, 0 Volts }	→ 011

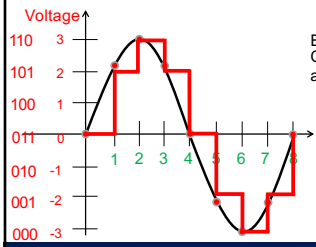
Encoding: mapping data from one form to another (not always conversion)

ADC – Storing the Data

- **Analog-to-Digital (ADC) Conversion**
  - **What do we store? Just the encoded bits:**
    - **Our digitized signal: {011, 101, 110, 101, 011, 001, 000, 001, 011}**
    - **It is now discrete & in digital format, store bits in MP3 player!**
  - **Why can we avoid storing the time?**
    - **It's repetitive! Just store sampling rate: 1 kilo-samples/ sec**
    - **Later, if we wish to restore signal, each "sample" occurred at 1ms**
  - **In this example:**
    - **Sampling rate: 1 k-samples/sec**
    - **Resolution: 3-bits**
    - **Our digitized signal: {011, 101, 110, 101, 011, 001, 000, 001, 011}**

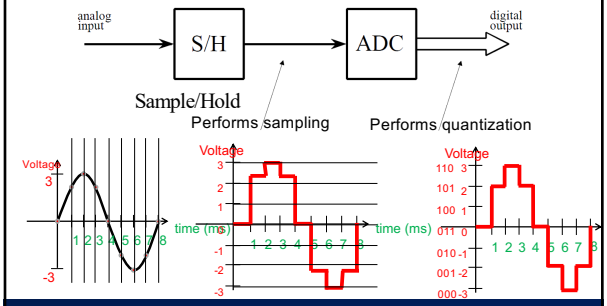
ADC – An approximation at Best

- **Analog-to-Digital (ADC) Conversion**
  - **Continuous analog signal overlaid with discrete digital signal**
  - **At best an approximation of original signal**



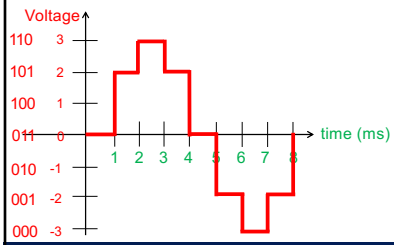
Ever heard an audiophile complain: CDs/MP3s/Digital music doesn't sound as "real" or as good old records? Now you know why!

ADC – Broken into two parts



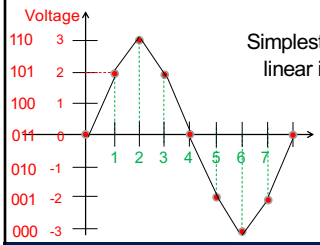
ADC – An approximation at Best

- **Digital-to-Analog (DAC) Conversion**
  - Process of converting *discrete* signal to *continuous* signal
    - **How to get back to original signal from bits?**



ADC – An approximation at Best

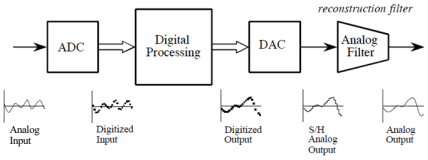
- **Digital-to-Analog (DAC) Conversion**
  - Process of converting *discrete* signal to *continuous* signal
    - **How to get back to original signal from bits?**



Simplest – connect the dots linear interpolation between points.

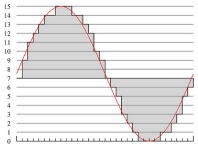
With more sophistication (and typical circuits) can make smoother links.

ADC / DAC – THE FULL PICTURE



Figures from reading: *The Scientist and Engineer's Guide to Digital Signal Processing*, By Steven W. Smith

PCM



- **Quantization & Sampling Technique described:**
  - Called Pulse-Code-Modulation (PCM)
    - Patented in 1943
    - PCM process is the ADC process
    - Developed for telecommunications

NEXT STEPS

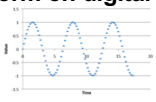
- **Lab Today:** sample sound waveforms
- **Monday:** look more formally and quantitatively at quantization and errors from quantization
- **Next Wednesday:** Start looking at discrete sampling rates

LEARN MORE

- **ESE215 – basic analog circuitry, RLC circuits, simple filters**
  - Including why typical circuits give smoother (not linear) connection of dots
- **ESE568 – Mixed Signal Integrated Circuits**
  - Build A2D, D2A

BIG IDEAS

- **Approximate continuous waveform on digital media by**
  - Discretize in all dimension
  - For audio: in time and amplitude
    - Sample in time; quantize voltage



- **Allows us to store audio signal as sequence of bits**
- **Reconstruct by "connecting-the-dots"**
  - If our dots are frequent enough to represent the signal

ADMIN

- **Reading for today, next Wednesday on syllabus**
- **Lab Today**
  - Lab kit pickup today 5:30pm-6:45pm Levine Lobby
  - Lab partners posted
    - If you cannot pick up a kit, you have a partner who can
  - Read lab, work prelab (includes download software)
- **Remember feedback**
- **TA Office Hours this week**
  - R2pm, F3-5pm, Sun. 2-4pm

REFERENCES

- **S. Smith, "The Scientists and Engineer's Guide to Digital Signal Processing," 1997.**
- **Wikipedia,** http://en.wikipedia.org/wiki/Analog-to-digital_converter
- **Wikipedia:** http://en.wikipedia.org/wiki/Pulse-code_modulation

39