Help Harvard Engineers Without Borders win an $8k Grant

ES50 TF, Will Jameson

Please follow this link

Former ES50 Student, Aisha Lee

http://wshe.es/ekTjhgYK

Current ES50 Student, Tomoya Hasegawa

ES50 Instructor, Chris Lombardo

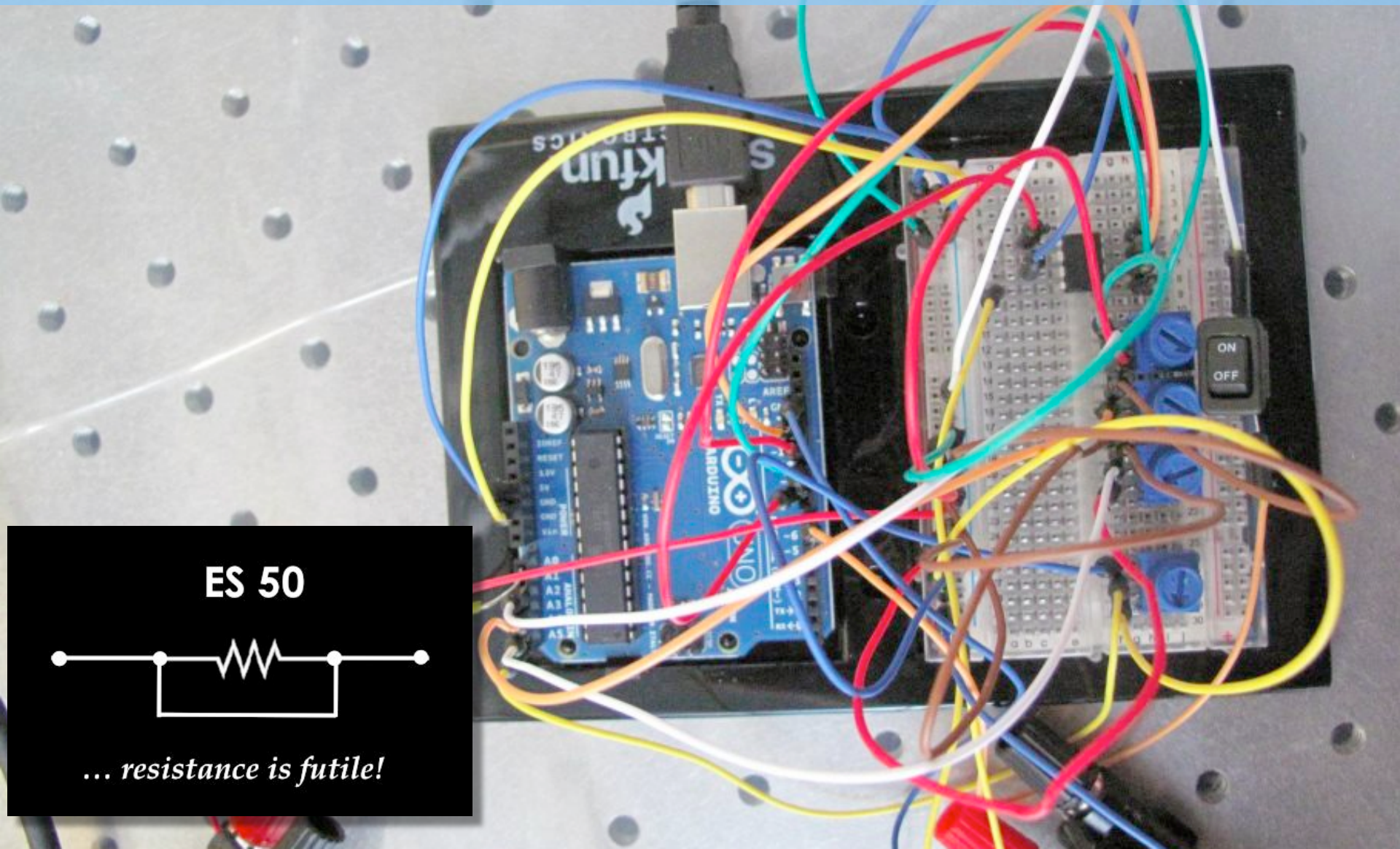
Thank you for your support of clean water in the Dominican Republic

HARVARD
School of Engineering and Applied Sciences

ENGINEERS WITHOUT BORDERS-USA
HARVARD UNIVERISTY CHAPTER

ES 50: Introduction to Electrical Eng.

Explore EE, satisfy a Gen Ed requirement & have fun



ES 50

… resistance is futile!

Lecture 10: Quantization & Digital Electroinics

October 19, 2015

HARVARD
School of Engineering and Applied Sciences

Upcoming Events

Lab 7 - Project Proposal

- Not a typical lab, please choose your final project team (3-4 people)
- You will brainstorming and beginning to write you final project proposal
- You may go to an alternative lab time so your group can meet together BUT email the lab TFs for the lab time you intend to go to ahead of time

Make up Labs

- Able to make up labs 1-6 during the week of Oct 26-30
- Google form will be sent out via CANVAS
- You must sign up in order to make up a lab

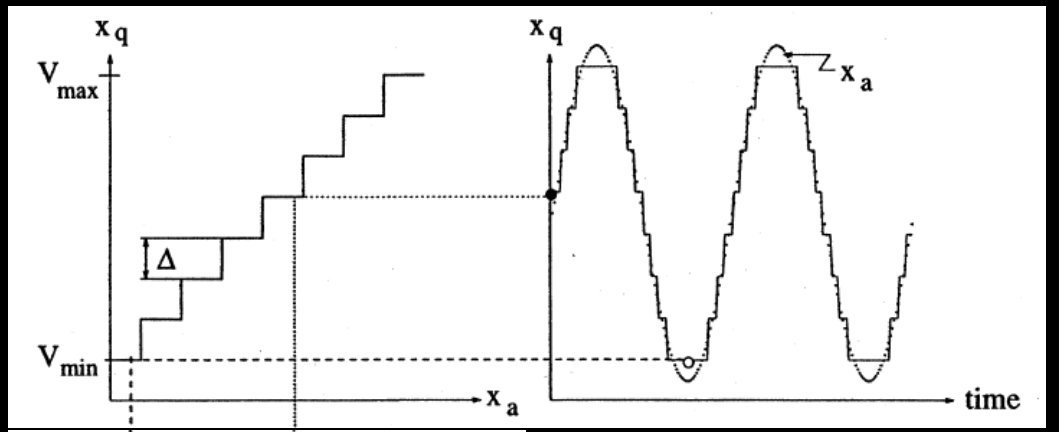
Plan for today…

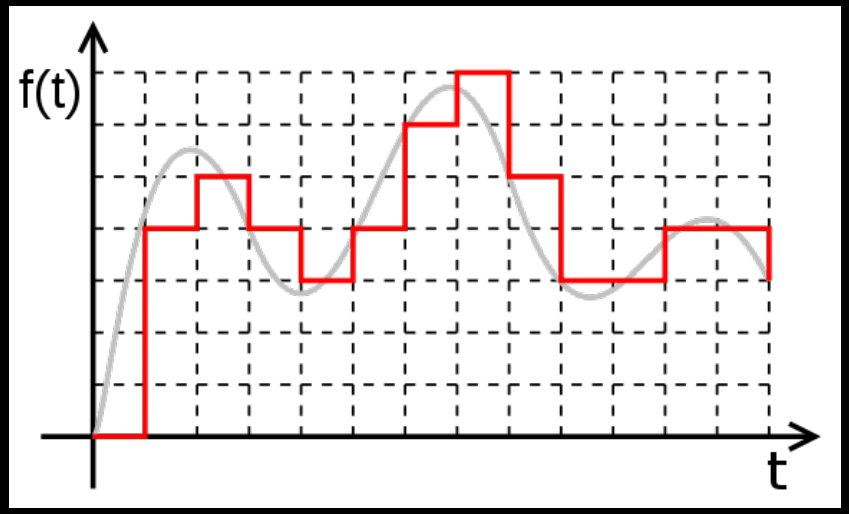
- Finish sampling & quantization
  - How to create digital data: bits, bytes, etc;
- How do digital systems (e.g. computers, microcontrollers, timers, etc) process these data to accomplish useful tasks
- Two classes of digital logic: combinational & sequential
  - Combinational: output depends only on the current input values; Example logic gates: AND, OR, NOT, XOR, NOR, etc.
  - Sequential: outputs depend not only on current inputs but also on the circuit's past. Sequential logic circuits have memory!
- Using sequential & combinational logic gates we can realize many complex digital systems, including computers, microcontrollers (Arduino), etc.
  - We can also build useful devices and systems: e.g. alarms, light controls, etc
- We will also talk about Boolean algebra and how to design logic circuits!

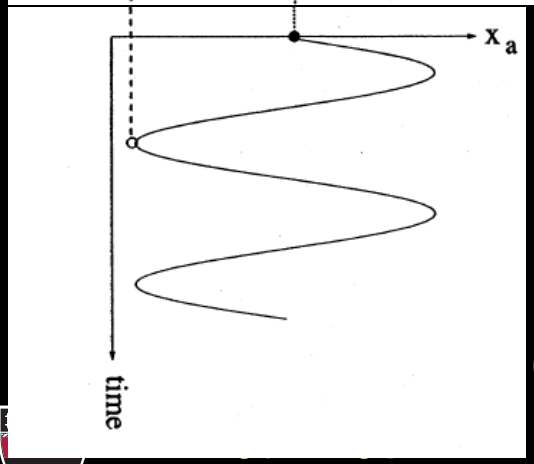
Quantizer

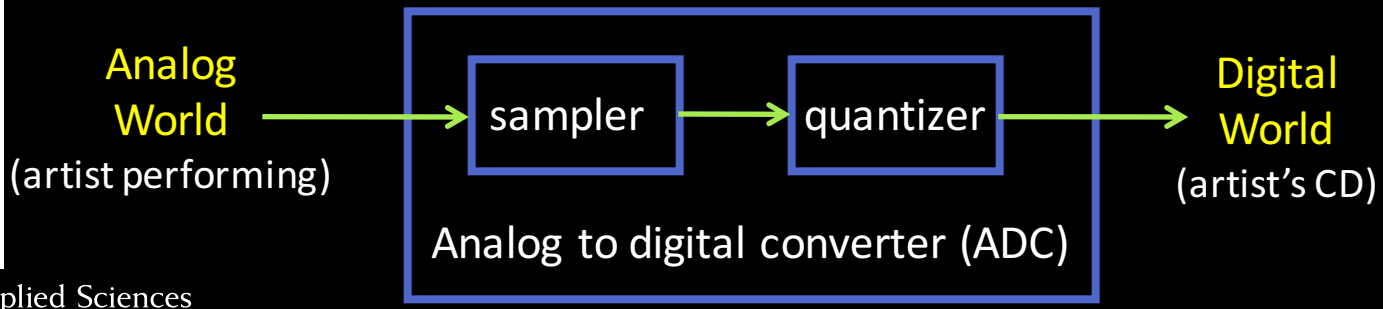
Why do we have to quantize in the first place?

- A sample of an analog signal can have any value. However, digital signals, represented with ***b*** bits can take only 2^b values so we need to round "analog" numbers to set of discrete values









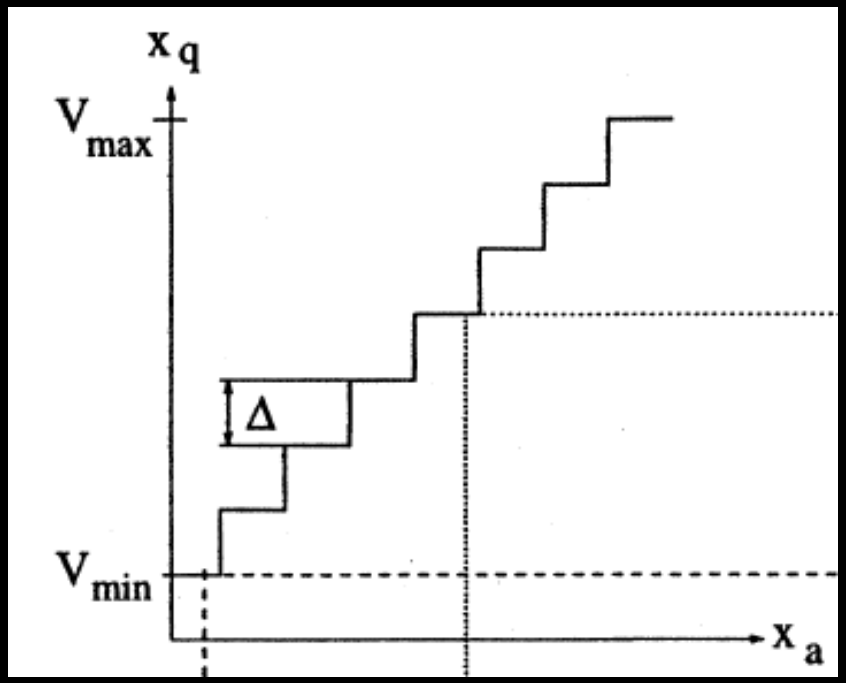
Quantizer Example

Let's say that we have a signal with $V_{min} = 0\ V$, $V_{max} = 7\ V$, and want to quantize with 3 bits.

- There are either possible binary values as $N_{steps} = 2^3 = 8$.
- Step Size = 1V

Step Size $\Delta = \frac{V_{max} - V_{min}}{N_{steps} - 1}$

Analog Value	Binary Number
0 V	000
1 V	001
2 V	010
3 V	011
4 V	100
5 V	101
6 V	110
7 V	111



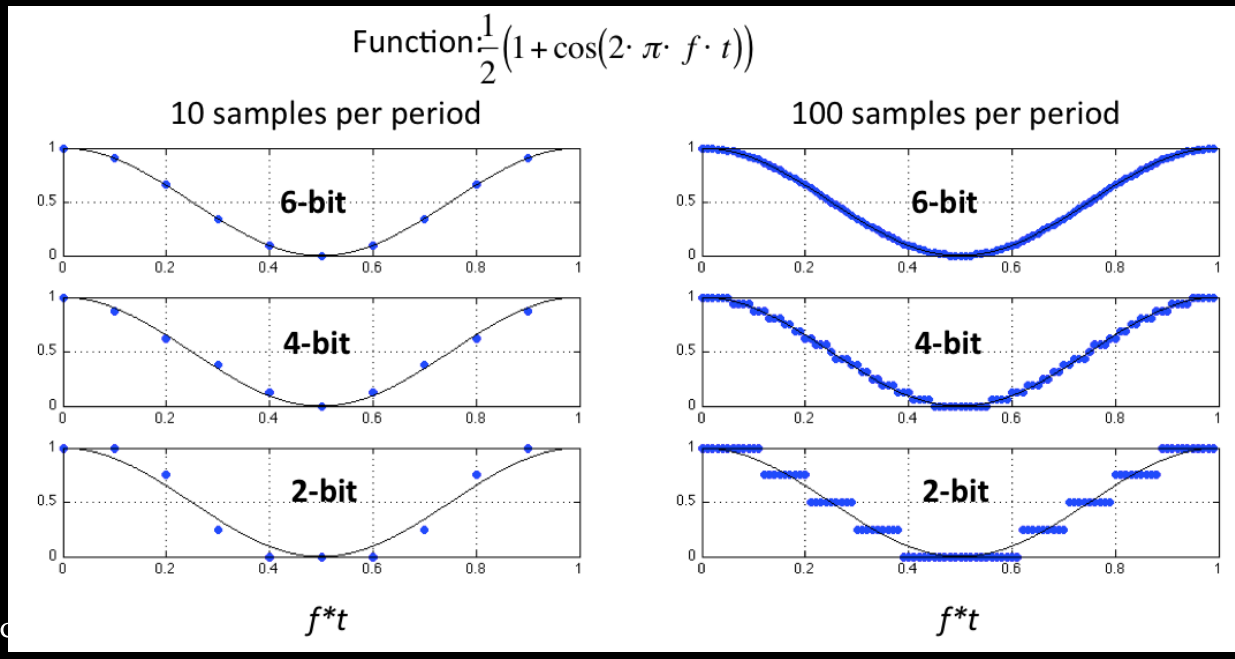
What about 4.3V input?

- It will be rounded up to 5V with error! (in this case, error is 0.7V)

ADC: Sampling & Quantization

- **Sampler:** Repeatedly **measures (samples) the amplitude** (volume) of an incoming music signal (an electrical voltage)
- **Quantizer:** rounds the sample amplitude to the nearest "allowed" level. Number of level is determined by number of bits used in quantizer. It also converts the quantized levels into digital data, stream of 0s and 1s.
  - 4 bits ⇔ 16 levels: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111
- **Higher sample rates produce better quality recordings but also bigger file sizes** which demand greater space on storage devices (such as hard drives), and faster processors (CPUs) to manipulate.
- **Sample length in bits determines the accuracy of each sample measurement.** Better accuracy means less distortion which results in better sound!

Sampled & Quantized signal
Discrete both in time and amplitude, & converted to stream of binary digits (bits)



Quantizer Errors

64k levels (16 bits)

64k to 4 level and back (16 ➔ 2 bit)

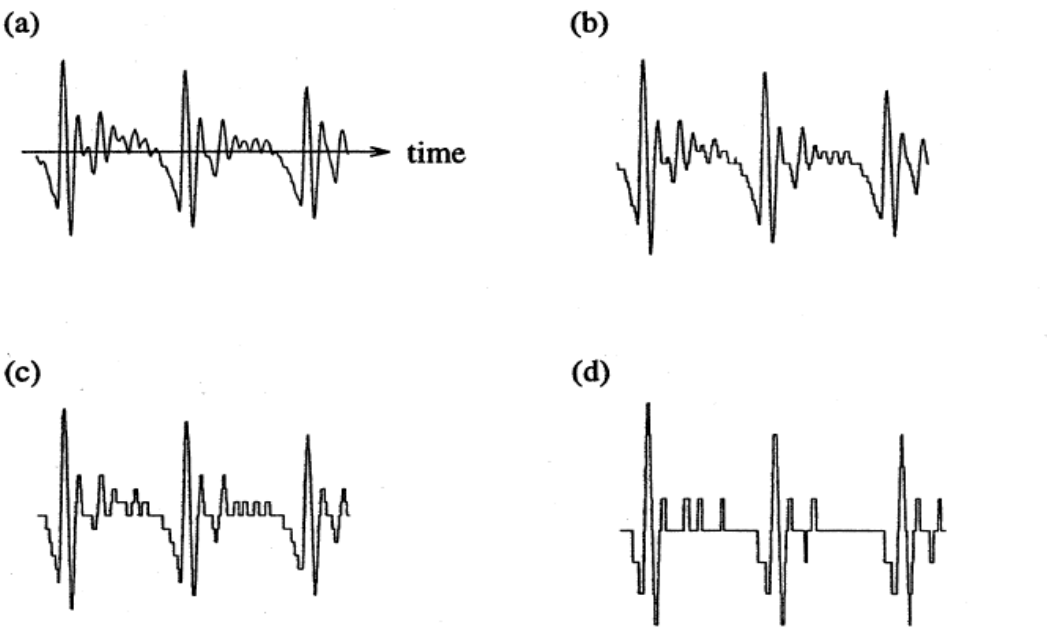


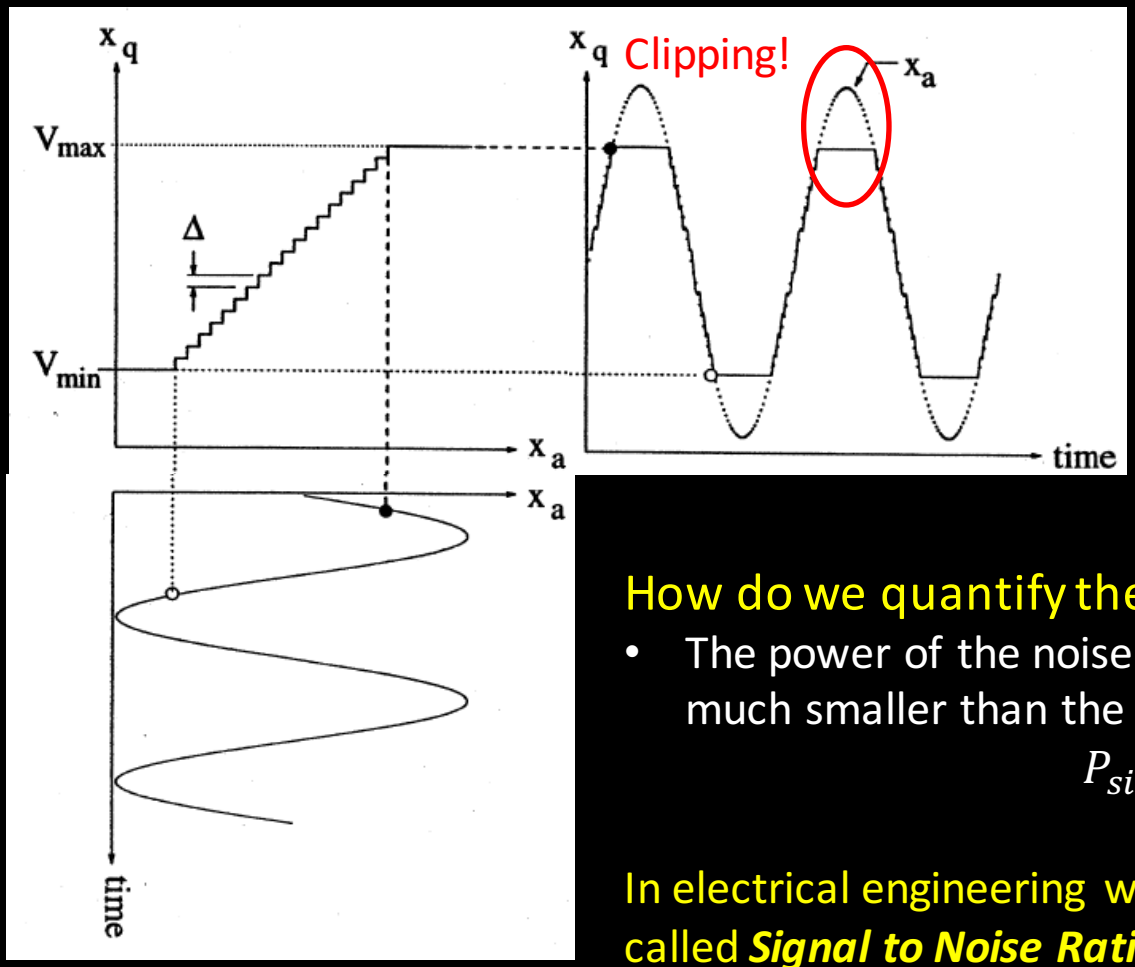
Figure 3.21 Quantization of the /ah/ speech waveform. (a) 16 bits (65,536 levels). (b) 5 bits (32 levels). (c) 4 bits (16 levels) (d) 3 bits (8 levels).

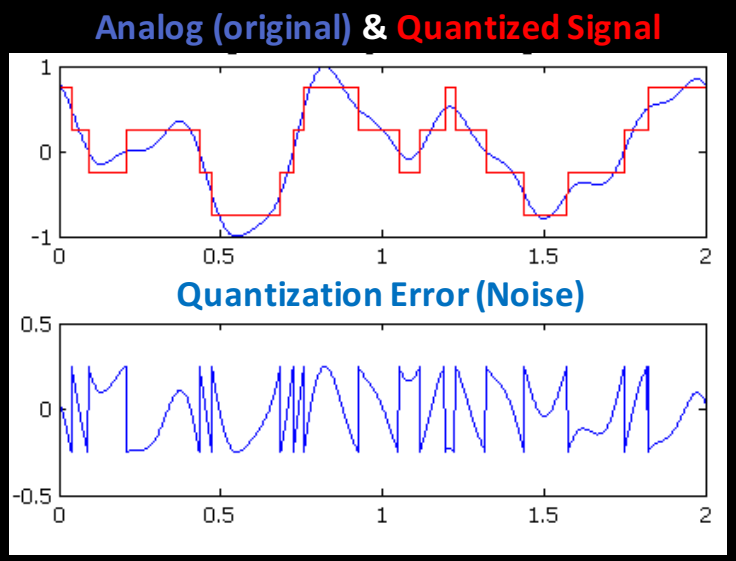
Two main types of quantizer errors:

1. **Approximation errors** – caused by the finite step size Δ. Δ is related to number of bits ***b*** in quantizer. Therefore, ***b*** has to bee chosen properly to minimize the errors.
2. **Clipping errors** – caused by inadequate quantizer range limits. The problem arises when quantizing variable-amplitude signals, such as music, which contains soft and loud sections.

http://www.coe.montana.edu/ee/rmaher/aes_cd/1.2.htm

Quantizer Errors





How do we quantify the performance of quantizer?

- The power of the noise due to quantization needs to be much smaller than the power of the signal

$$P_{signal} > P_{noise}$$

In electrical engineering we often define the ***figure of merit*** called ***Signal to Noise Ratio (SNR)***, defined as:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

Signal to Noise Ratio: SNR

Without going into much detail, it can be shown that $P_{noise} \approx \Delta^2$, where Δ is the height of the quantization step we saw earlier:

$$\Delta = \frac{V_{max} - V_{min}}{N_{steps} - 1}$$

Remember that # of steps $N_{steps} \approx 2^b$, & b is number of bits used in the quantizer (4, 8, 16, …)

V_{max} and V_{min} define the range within which analog signal is expected.

Also without going into much detail, it can be shown that

$$P_{signal} \approx (V_{max} - V_{min})^2$$

- This sort of makes sense as the larger the "swing" of analog signal the more power it has

Now, taking into account the multiplicative scaling factors (beyond the scope of this discussion) we can derive the equation for signal to noise ratio to be:

$$SNR = \frac{12\left(\frac{V_{max}-V_{min}}{8}\right)^2}{\left(\frac{V_{max}-V_{min}}{2^b-1}\right)^2} \approx \frac{3}{16}4^b, \text{ assuming } 2^b >> 1$$

(true for 4-bit and better quantizers)

Signal to Noise Ratio: SNR

SNR is typically represented in log-scale, using decibels (dB)

According to definition:

$SNR \approx \frac{3}{16} 4^b$, assuming $2^b >> 1$

$$SNR_{dB} = 10 \log_{10}(SNR)$$

Intuition: $SNR_{dB} = 10$ means that signal is 10 times more powerful than noise.
$SNR_{dB} = 20$, means that signal is 100 times more powerful than noise.
$SNR_{dB} = 100$, means that signal is 10,000,000,000 more powerful than noise. And that's a pretty good value!

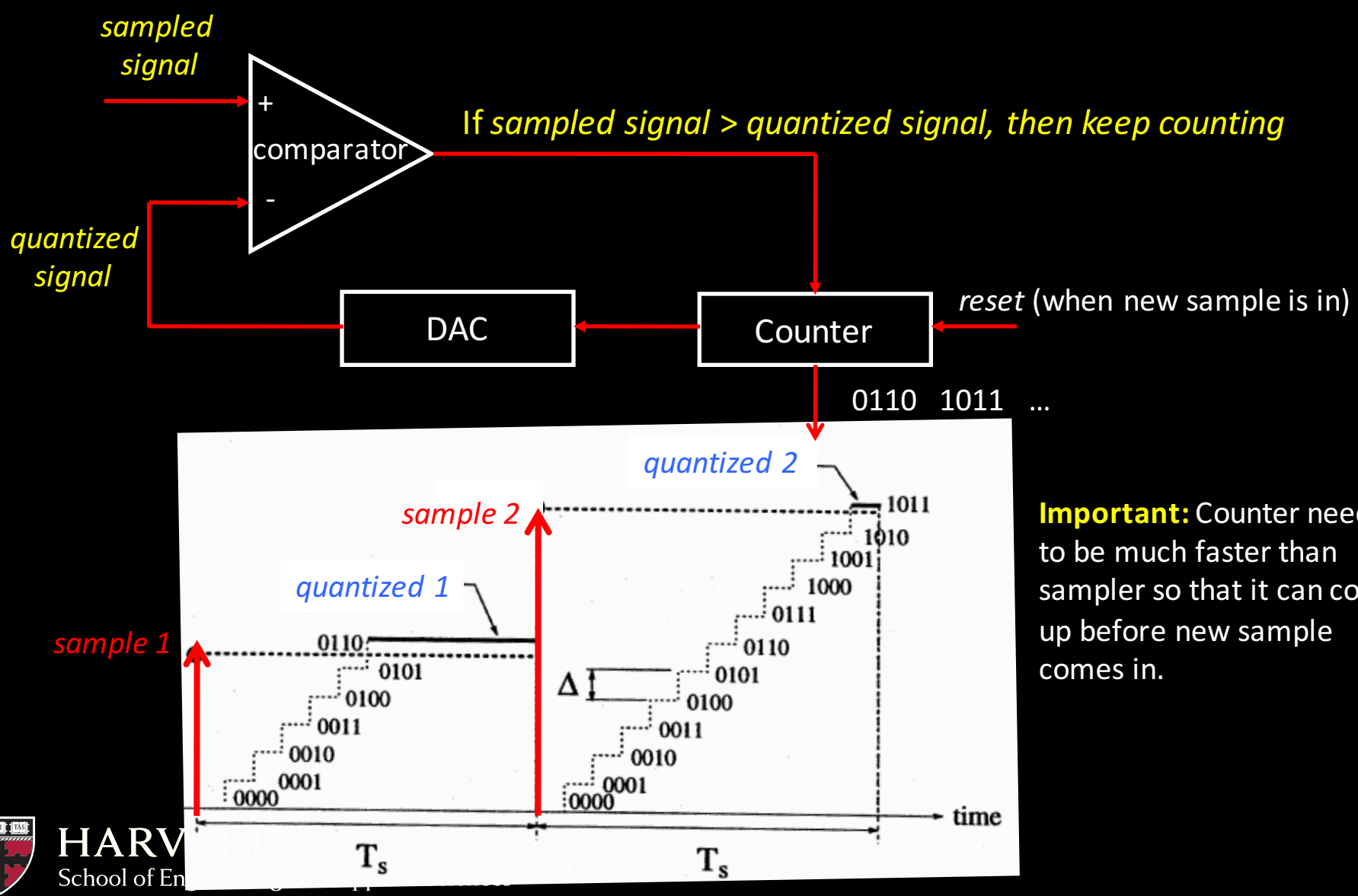
Now by converting our equation to dB scale we find:

$$SNR_{db} \approx 10 \log_{10}\left(\frac{3}{16} 4^b\right),$$
$$SNR_{db} \approx 10 \log_{10}(4^b) + 10 \log_{10}\left(\frac{3}{16}\right),$$
$$SNR_{db} \approx b\, 10 \log_{10}(4) - 7.3 = 6b - 7.3$$

We finally find that $SNR_{db} \approx 6b - 7.3\ dB$, that is *every extra bit in quantizer improves signal to noise ratio by 6dB (nearly 4 times)!*

Example: 16-bit quantizer has $SNR_{db} \approx 90\ dB$
That is, signal is nearly nine orders of magnitude (1,000,000,000 times) larger than noise!

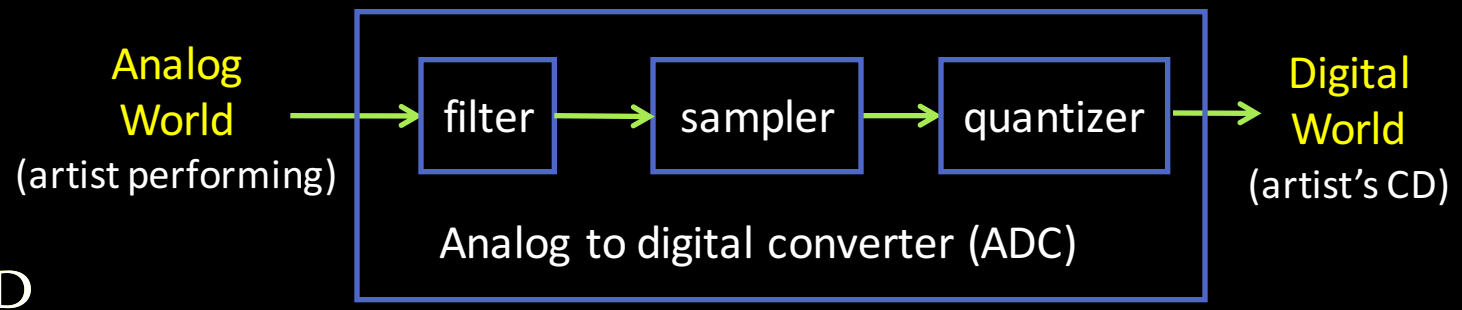
How does quantizer work?



Important: Counter needs to be much faster than sampler so that it can count up before new sample comes in.

Summary: Analog to Digital Converter Design

1. Low pass filter, with cut-off frequency f_{max}
   - Needed to make sure that signal is *bandwidth limited*
2. Sampling frequency needs to be large enough to obey Nyquist Criterion
   - $f_s > 2 \cdot f_{max}$
   - CD quality: $f_{sample} = 44.1\ kHz$
3. Quantizer needs to have enough bits to avoid quantization noise
   - CD quality: 16-bit quantizer ➔ $SNR = 90\ dB$
4. Quantizer needs to be fast & have enough dynamic range to be able to capture large and small amplitudes



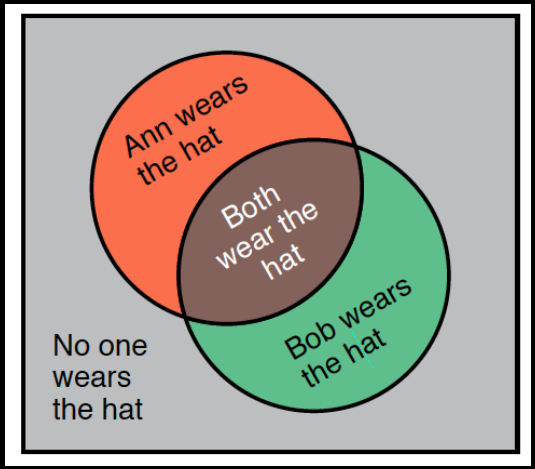
A bit of history: Boolean Algebra

George Boole, a British mathematician, born in 1815:

- Wanted to reduce the world to a series of true-or-false statements which could overlap in interesting ways. VERY useful for computers as we know them.
- He made a link between LOGIC & ALGEBRA

Example: suppose there is a couple, Ann & Bob, who own only one hat. Clearly, if you happen to run into Ann and Bob walking down the street, there are four possibilities: neither of them may be wearing a hat, Ann may be wearing it, Bob may be wearing it, or they are both wearing it (not likely though!).

- The diagram below (Venn diagram) illustrates this. All the states are possible except the one where the circles overlap. The situation can be also described by the **truth table.**



Ann is wearing the hat	Bob is wearing the hat	This combination can be
NO	NO	TRUE
NO	YES	TRUE
YES	NO	TRUE
YES	YES	FALSE

Logical Outcomes & Truth Tables

What does the "LOGIC" look like for Ann & Bob?

- Ann is not wearing a hat, $A = 0$; Ann is wearing a hat, $A = 1$
- Bob is not wearing a hat, $B = 0$; Bob is wearing a hat, $B = 1$
- A *Truth Table* represents the inputs & outputs for what is "true" or logically valid

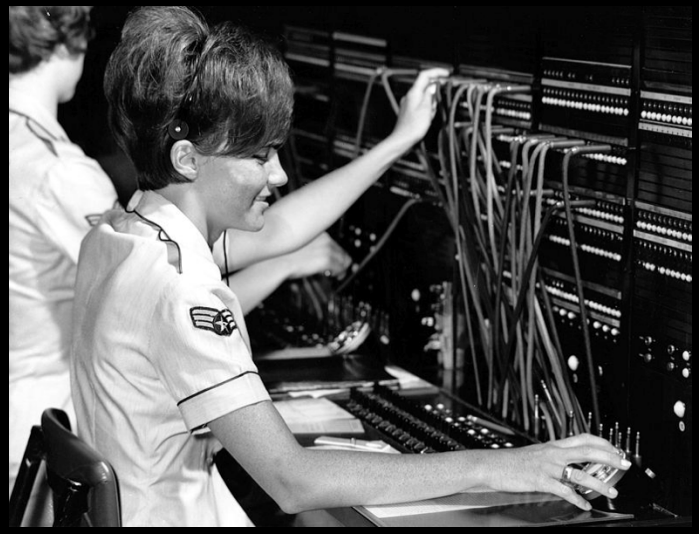
Ann is wearing the hat	Bob is wearing the hat	This combination can be
NO	NO	TRUE
NO	YES	TRUE
YES	NO	TRUE
YES	YES	FALSE

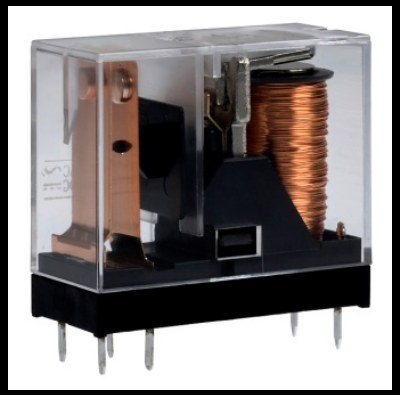
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

So what is the expression for C in terms of A & B?

Boolean and Shannon

- Boole published his treatise on logic in 1854, long before it could be applied to electrical or electronic devices. During his lifetime, his work seemed to have no practical applications at all.
- Claude Shannon encountered Boolean logic while studying at MIT in the 1930s. In 1938 he published a paper describing how Boolean algebra could be applied to circuits using relays (switches).
  - This had immediate practical applications, as telephone networks were growing rapidly, creating complicated switching problems.





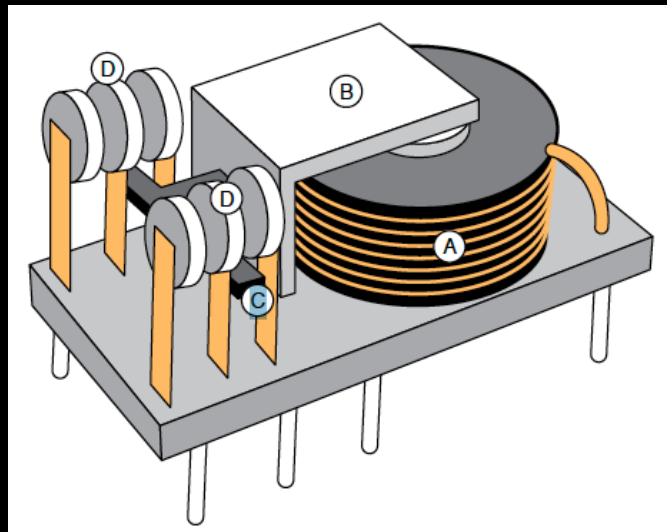
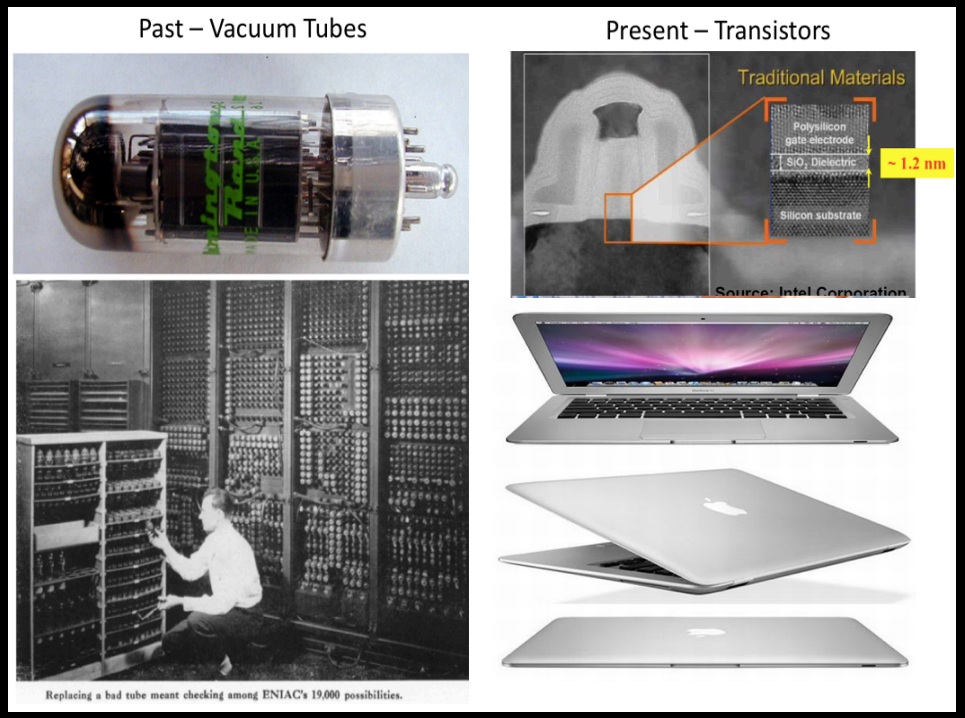


Figure 2-51. This is one way that the parts inside a relay can be arranged. The coil, A, generates a magnetic attraction pulling lever B downward. A plastic extension, C, pushes outward against flexible metal strips and moves the poles of the relay, D, between the contacts.

Shannon solves telephone switchboard problem

- A very simple telephone problem: Suppose two customers in a rural area share one telephone line. If one of them wants to use the line, or the other wants to use it, or neither of them wants to use it, there's no problem. But they cannot both use it at once (like sharing hats!).
- Now imagine that there are hundred of thousands of users and only hundreds of lines… becomes very complicated. Shannon realized that Boolean analysis could be used for this purpose!
  - **if you used an "on" condition to represent numeral 1 and an "off" condition to represent numeral 0, you could build a system of relays that could count and do arithmetic!**



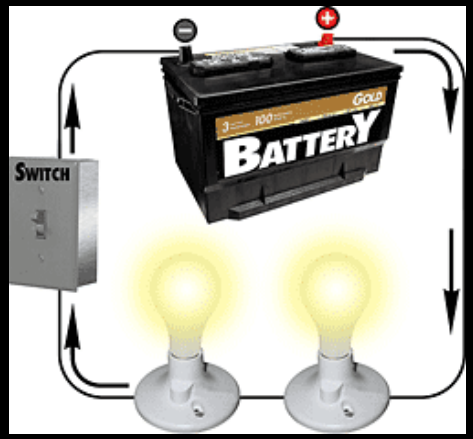
Logic Variable

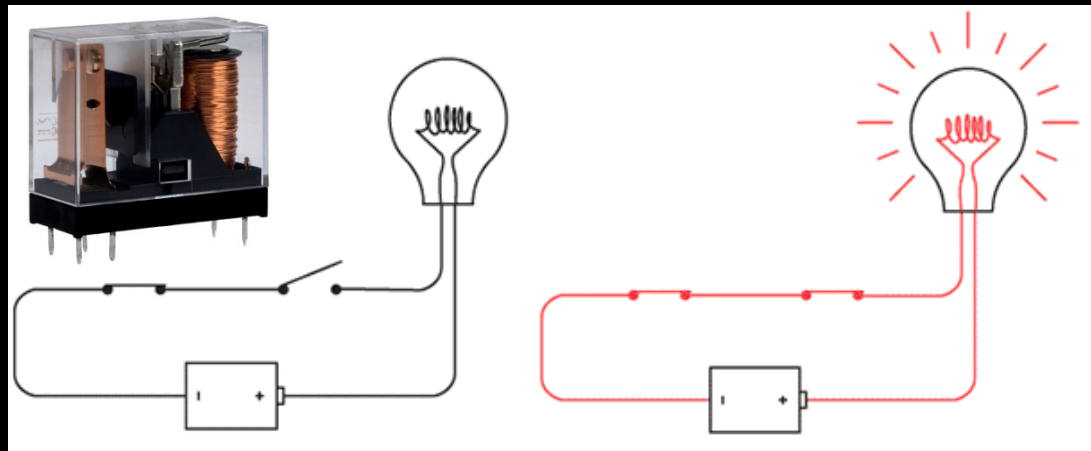
Can have only two values, e.g: TRUE or FALSE, 0 or 1, YES or NO, OPEN switch or CLOSED switch, high voltage (e.g. 5V) or low voltage (e.g. 0V)

- Convention: we assign
  - 1 to {YES, CLOSED, high voltage, 5V}
  - 0 to {NO, OPEN, low voltage, 0V}

The state of the switch determines the logic state of the variable: 1 or 0

- Digital integrated circuits (including computers) are essentially bunch of switches and memory elements. Therefore, the size of the switch determines the size of the computer



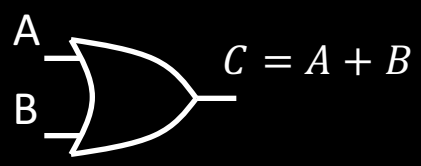


Logical OR

If the world had only 1's & 0's

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

Logic Symbol

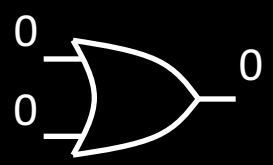


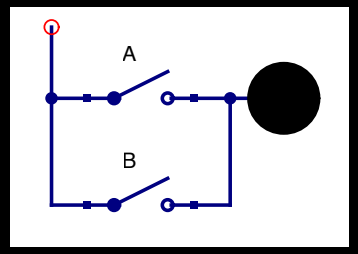
Truth Table

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

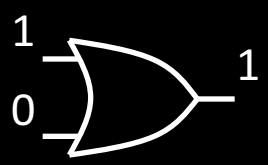
Logic operations are mapped to switches in PARALLEL

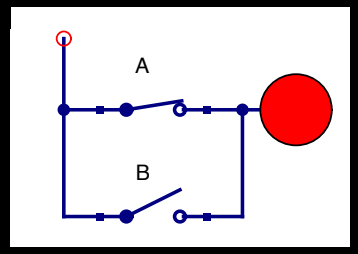
$0 + 0 = 0$



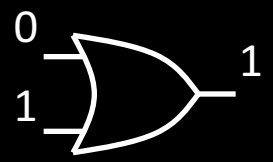


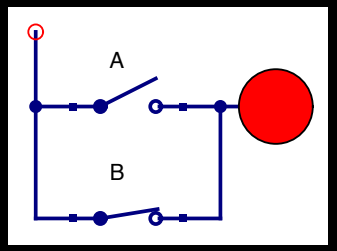
$1 + 0 = 1$



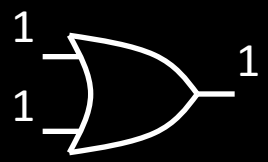


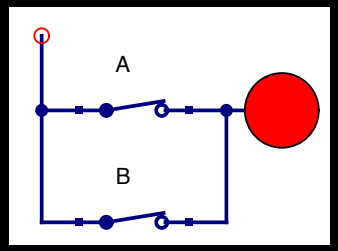
$0 + 1 = 1$





$1 + 1 = 1$



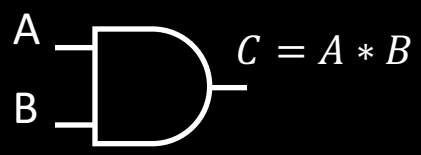


Logical AND

If the world had only 1's & 0's

$$0 * 0 = 0$$
$$0 * 1 = 0$$
$$1 * 0 = 0$$
$$1 * 1 = 1$$

Logic Symbol

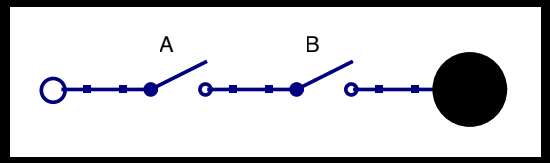


Truth Table

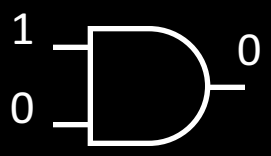
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

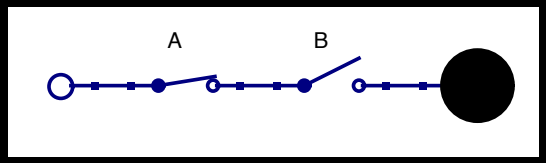
Logic operations are mapped to switches in SERIES

$$0 * 0 = 0$$

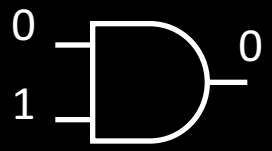


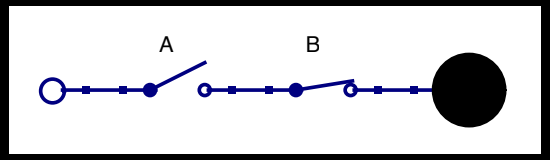
$$1 * 0 = 0$$



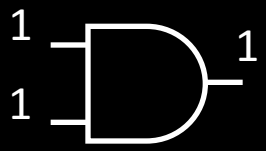


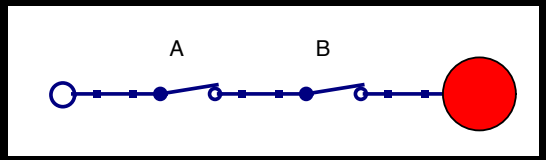
$$0 * 1 = 0$$





$$1 * 1 = 1$$





Logical NOT

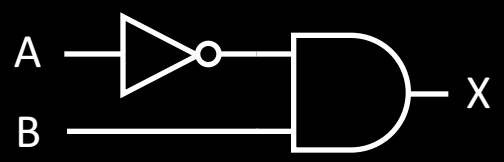
Logic Symbol

$C = \overline{A}$

Truth Table

A	C
0	1
1	0

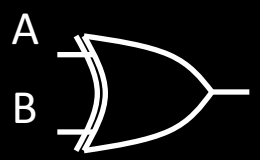
So how do we find X in terms of A & B?



$$X = \overline{A} * B = \overline{A}B$$

XOR (Exclusive OR)

Logic Symbol



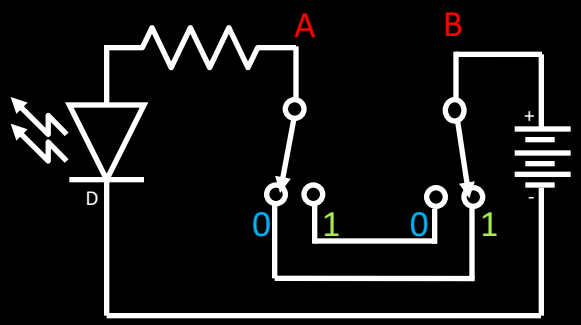
$$C = A \oplus B = \overline{A}B + A\overline{B}$$

Truth Table

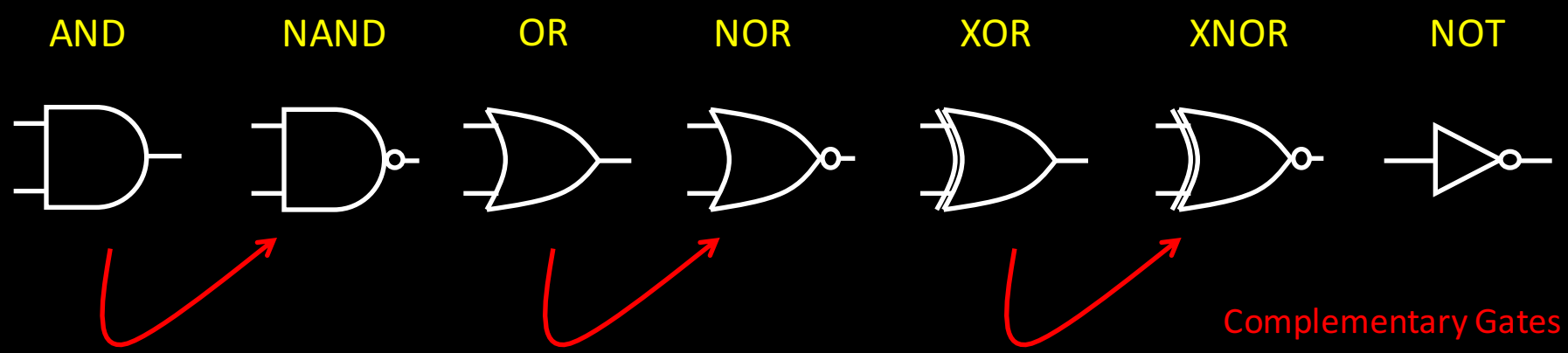
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Home Application of XOR:

Wiring light switches in your home



Logic Gates & Complements



A	B	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

A	B	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

A	B	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

A	NOT
0	1
1	0

The Rules of Boolean Algebra

Communitive Law

- $A + B = B + A$
- $AB = BA$

Associative Law

- $(A + B) + C = A + (B + C)$
- $(AB)C = A(BC)$

Distributive Law

- $A(B + C) = AB + AC$
- $A + (BC) = (A + B)(A + C)$

Identity Law

- $A + A = A$
- $AA = A$

Redundance Law

- $A + AB = A$
- $A(A + B) = A$

Zeros

- $0 + A = A$
- $0A = 0$

Ones

- $1 + A = 1$
- $1A = A$

Complements

- $A' + A = 1$
- $A'A = 0$
- $(A')' = A$

DeMorgan's Theorem

- $(A + B)' = A'B'$
- $(AB)' = A' + B'$

Other Useful Identities

- $AB + AB' = A$
- $(A + B)(A + B') = A$
- $A + A'B = A + B$
- $A(A' + B) = AB$

Complements can be represented by both $\overline{A}$ and A'

Truth Tables for Complex Circuits

- A truth table lists all possible combinations of the input variables and specifies the output for each input combination
- If a logic circuit has N inputs, how many possible combinations are there?
  - The most systematic way to arrange the table is to arrange inputs in numerical order starting with 0.
  - Once the truth table is known, it is EASY to figure out equation that describes the dependence of the outputs (they may be many of them) on the inputs.

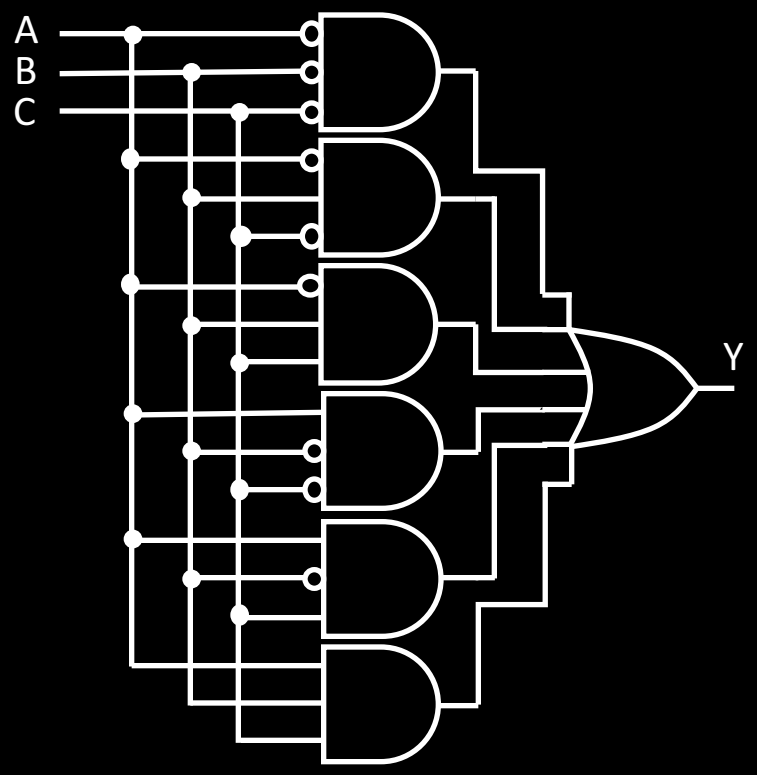
How do we find circuit that represents this table?

1. Find all combinations that result in output $Y = 1$
2. Since $Y = 1$ when ***any*** of these combinations is present, we join them using OR gate!

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$Y = A'B'C' + A'BC' + A'BC + AB'C' + AB'C + ABC$$

And the circuit that does this is



Cool! But is this the only/optimal solution? Can we use fewer gates?

- Well, our systematic approach does not always give the simplest solution. We can simply the answer using the rules we just learned

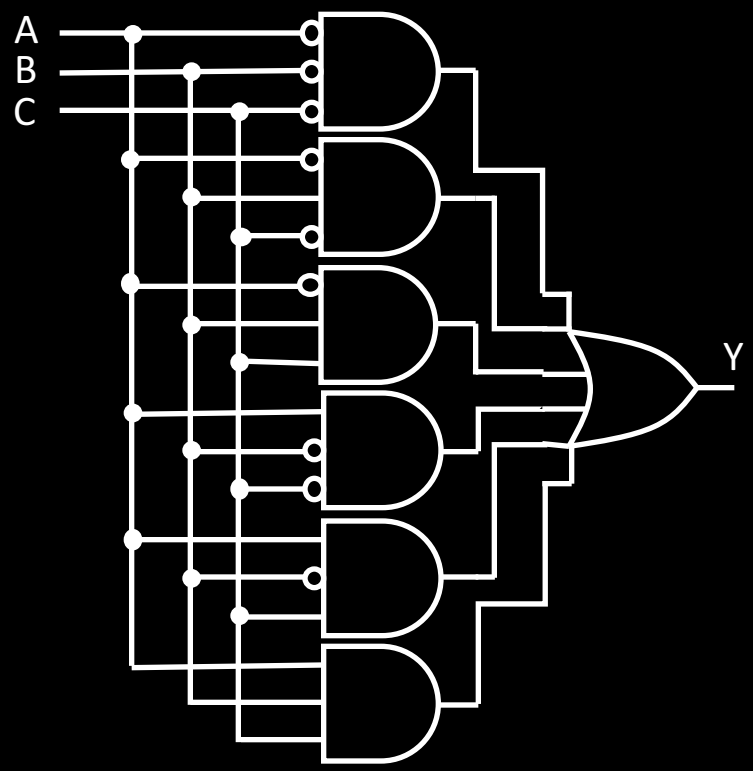
Example: Consider the complement

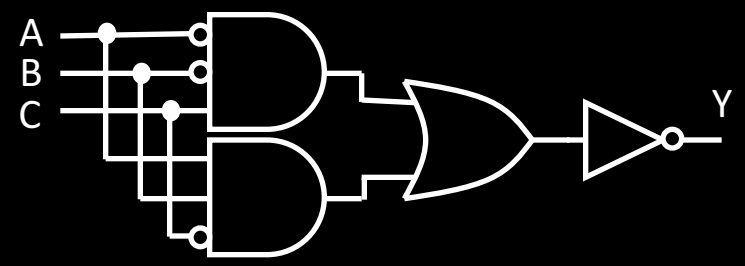
A	B	C	Y	Y'
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

$$Y' = A'B'C + ABC'$$

$$Y = A'B'C' + A'BC' + A'BC + AB'C' + AB'C + ABC$$

Simpler circuit





$$Y' = A'B'C + ABC'$$

In ES 50 we will use Boolean Algebra to simplify the logic expressions and obtain simpler circuits

- This, however, does not guarantee that we will find the optimal solution!

$$Y = A'B'C' + A'BC' + A'BC + AB'C' + AB'C + ABC$$

There is a systematic way to find the optimal solution: Karnaugh map (check wikipedia, as it is beyond this course)