CS 2110 - Lab 03

Digital Logic, Combinational Logic

Wednesday, May 25, 2022

Lab Assignment: Canvas Quiz

1. Go to Quizzes on Canvas
2. Select Lab 03, password: **CMOS**
3. Get 100% to get attendance!
   a) Unlimited attempts
   b) Collaboration is **allowed**!
   c) Ask your TAs for help :)

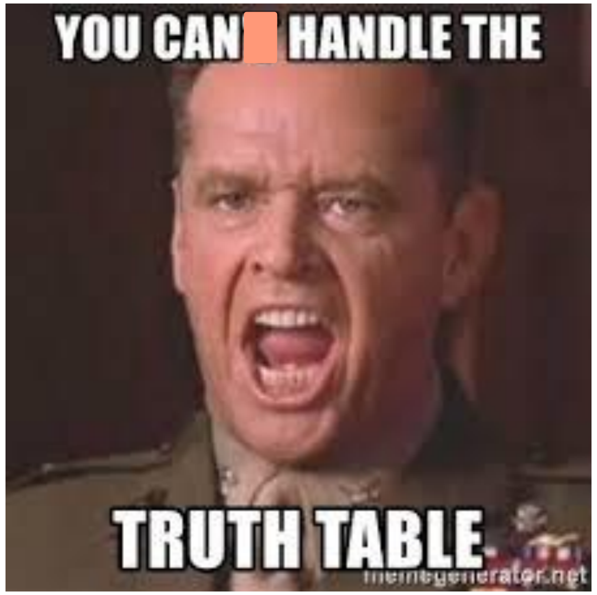
Homework 1

- Released!
- **Due Thursday, June 2nd at 11:59 PM** (standard 24 hr grace period)
- Files available on Canvas
- Submit on Gradescope
    - **Double check that your grade on Gradescope is your desired grade!**

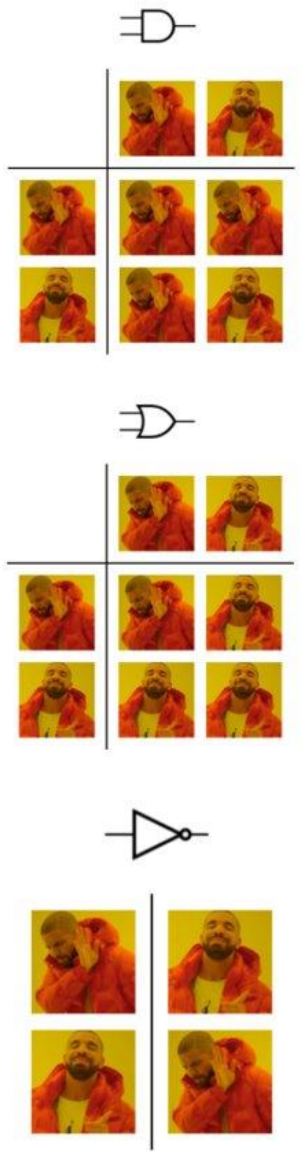
Homework 2

- Released this Friday, May 27th
- **Due Monday, June 6th at 11:59 PM**
- Files available on Canvas
- Submit on Gradescope (unlimited submissions)

Today's Topics

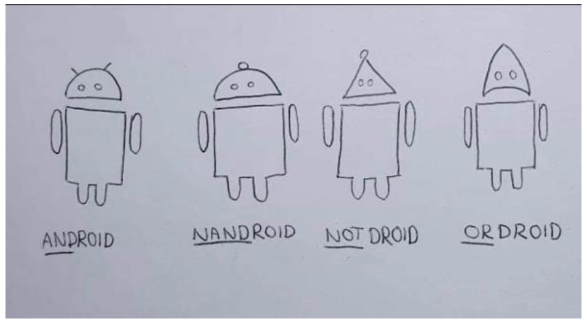


- Transistors and CMOS Design
- Logic Conversions
- Decoders and Multiplexers
- IEEE 754

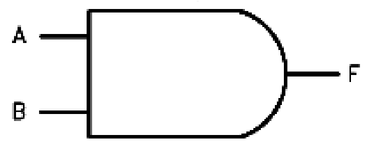

Exhaustive truth tables with Drake

Logical Operations

How can we implement these different operations in hardware?

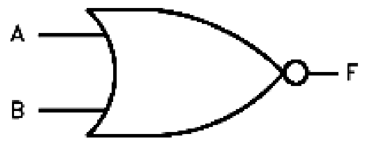


AND Gate



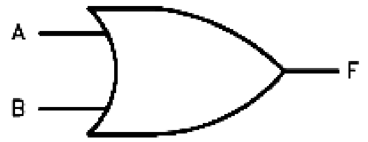
INPUT		OUTPUT
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

NOR Gate



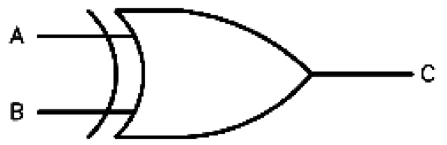
INPUT		OUTPUT
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

OR Gate



INPUT		OUTPUT
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

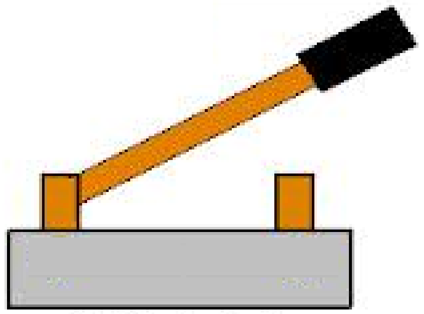
Exclusive OR Gate



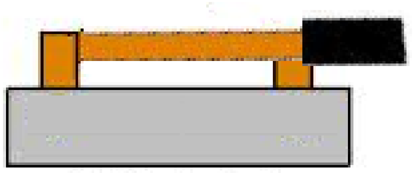
INPUT		OUTPUT
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Transistors

Transistors are digital “switches” and are the magical building blocks of all gates.



Open
No current can
flow through
switch



Closed
Current can flow
through the
switch

Beautiful game btw

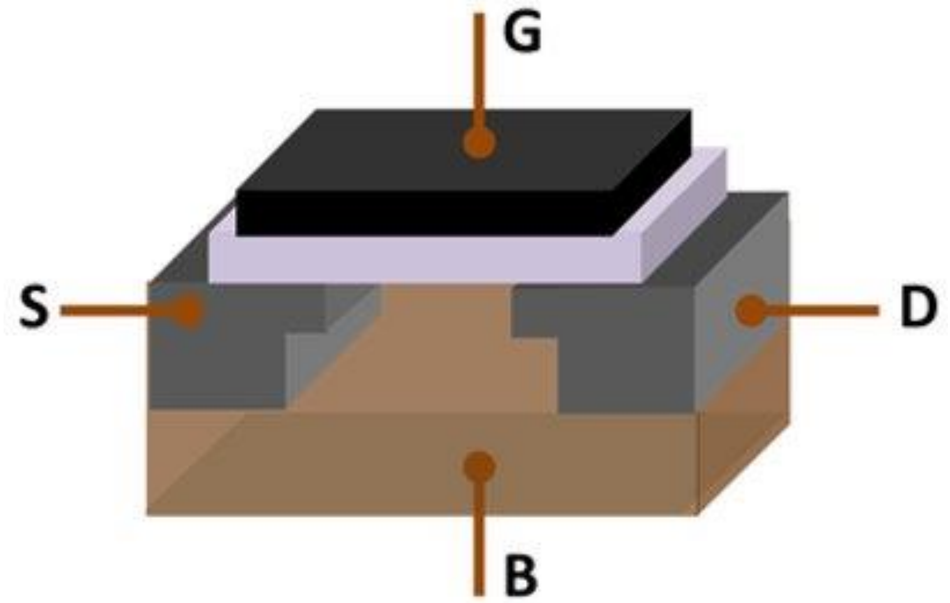
Transistors

Parts of a MOS transistor:

- Gate
- Source
- Drain

Types of MOS transistors:

- P-type
- N-type

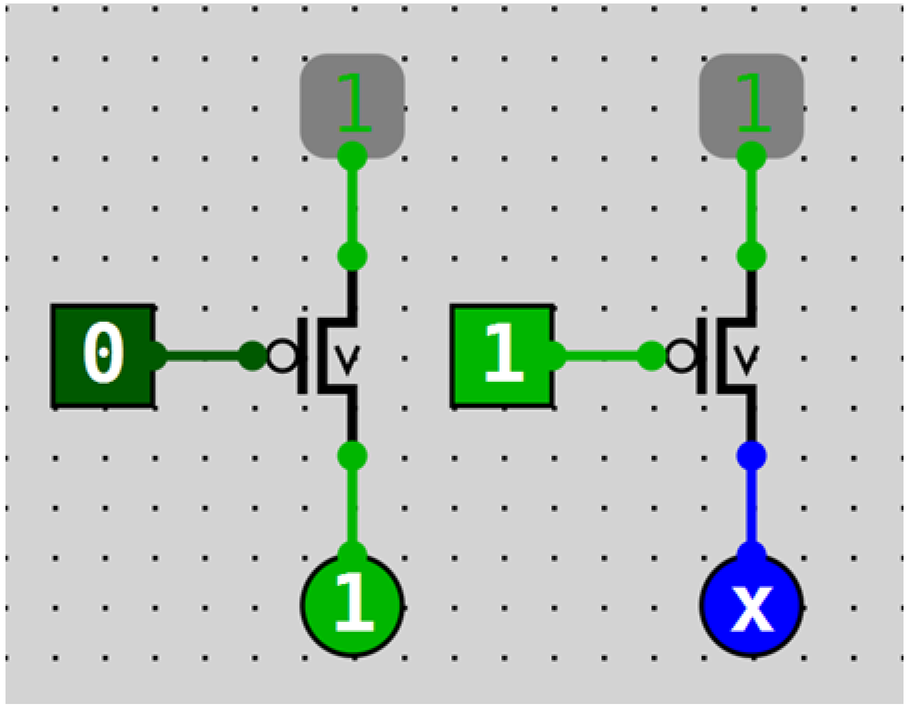


B = Body (irrelevant chemical magic!)

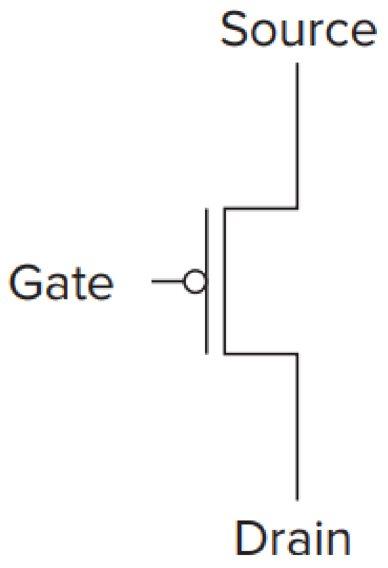
Complementary MOS ("CMOS") design: combining PMOS and NMOS transistors to make more complicated circuits

P-type Transistors

- Connected to Power
  - P-type transistors CANNOT propagate a strong "0" signal
- "Normally closed"

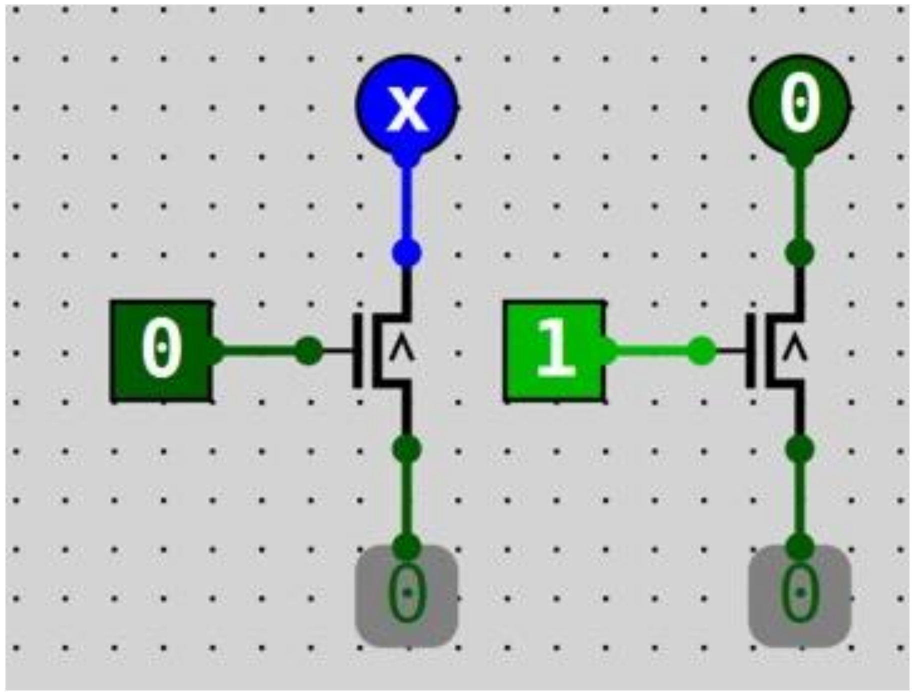


Closed Open

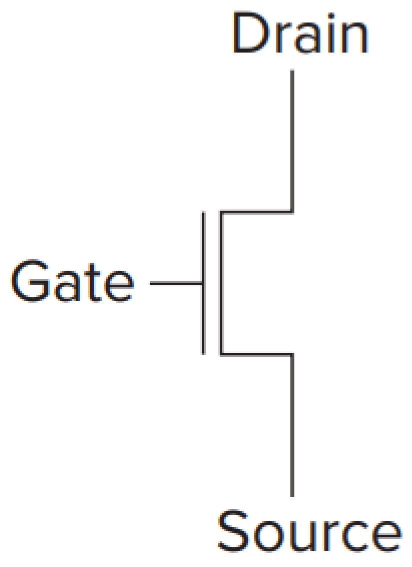


N-type Transistors

- Connected to Grou**N**d
    - N-type transistors CANNOT propagate a strong “1” signal
- “Normally open”

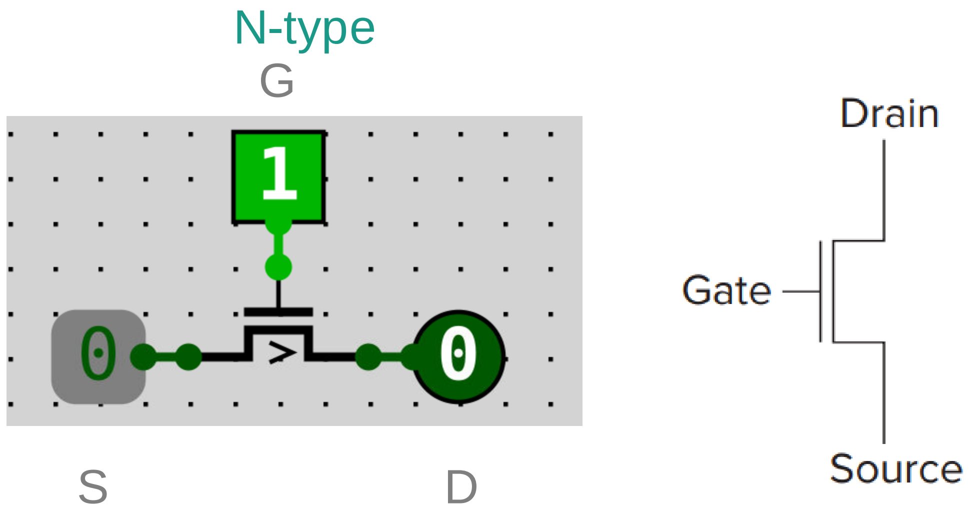


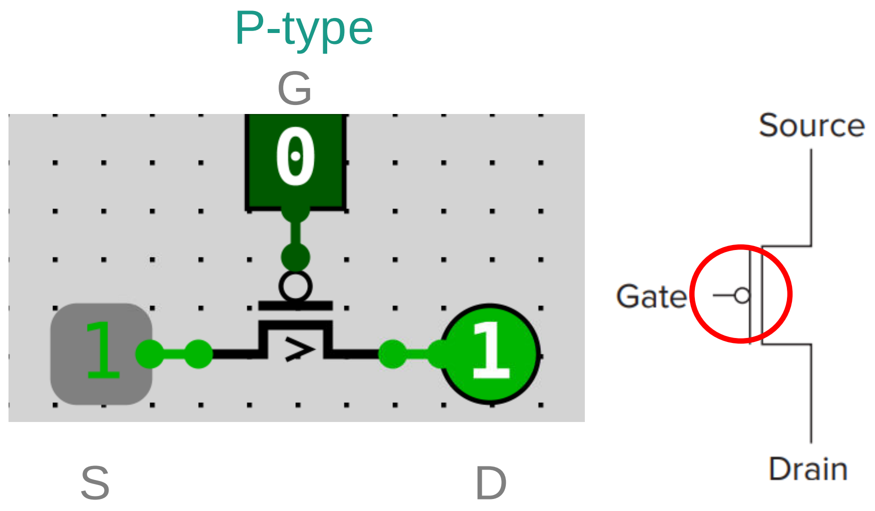
Open Closed



N-Type vs P-Type Comparison

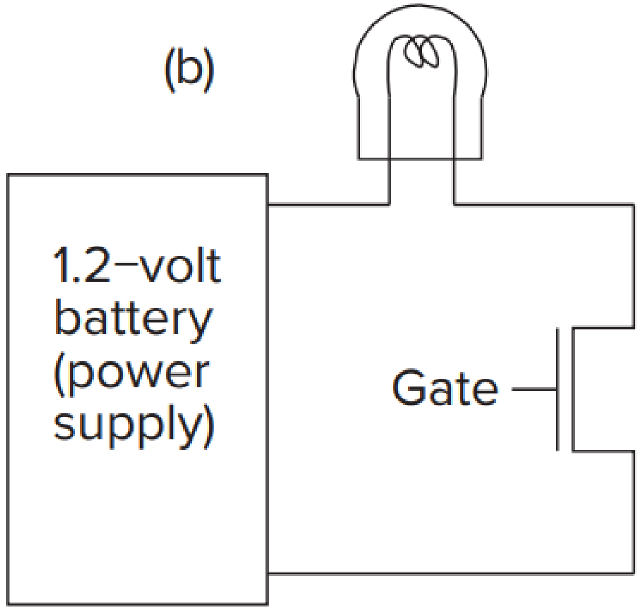
- They have the opposite effect; notice the bubble!
- Check the direction of the arrows – we always point from source (input) to drain (output)
- In CMOS design, connecting multiple P-type in series implies we will connect some *complementary* N-type transistors in parallel, and vice versa

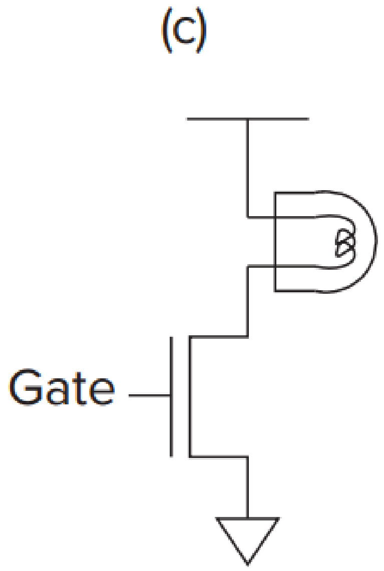




Transistors

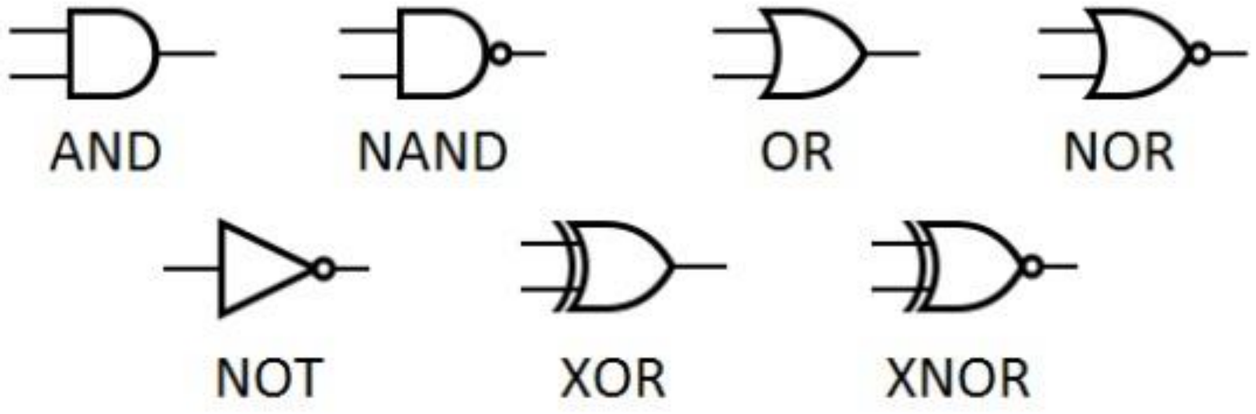
- What transistor is in this circuit?
- Figure c) is shorthand for b)





Logic Gates

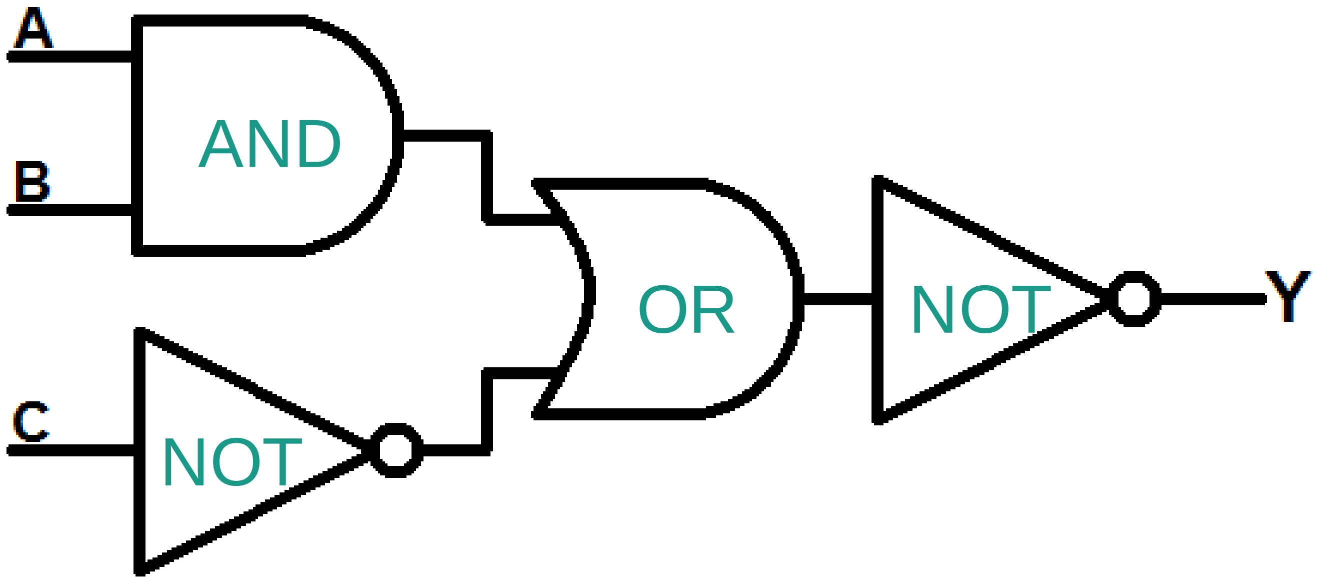
- Abstraction for representing groups of transistors
- Perform basic binary operations
- Basic building blocks of circuits
- Wires connect inputs and outputs



A basic circuit looks like...

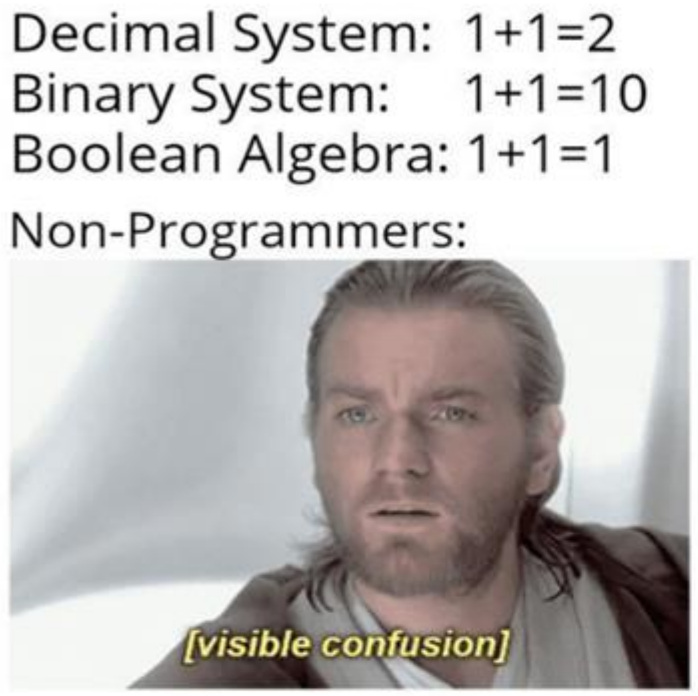
- An Input
- An Output
- Some binary operations

What would *Y* equal if *A*, *B*, and *C* are all 1?
This is on your attendance quiz!



Logic Conversion

- From Truth Tables
- From Boolean Expressions
- From Circuits



Converting a Truth Table to Sum of Products

A	B	C	Func(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

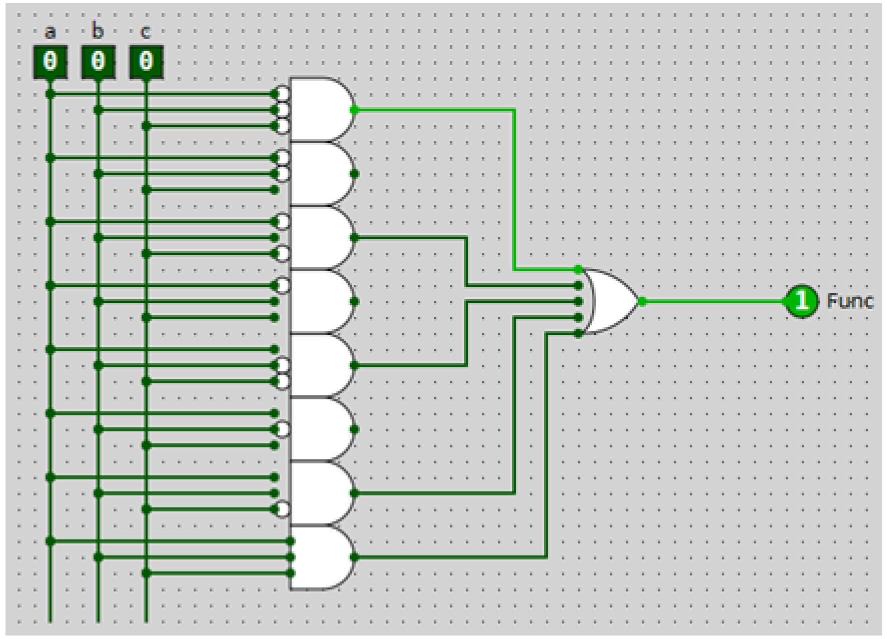
- Create a term for each case that results in 1
    - Each term is an AND
- OR all the terms together
- Func(A,B,C) = A’B’C’ + A’BC’ + AB’C’ + ABC’ + ABC

Converting Truth Table to Sum of Products Circuit

A	B	C	Func(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

We can build out a bunch of AND gates to represent every line in the table and use an OR gate to get the correct output.

Func(A,B,C) = A'B'C' + A'BC' + AB'C' + ABC' + ABC

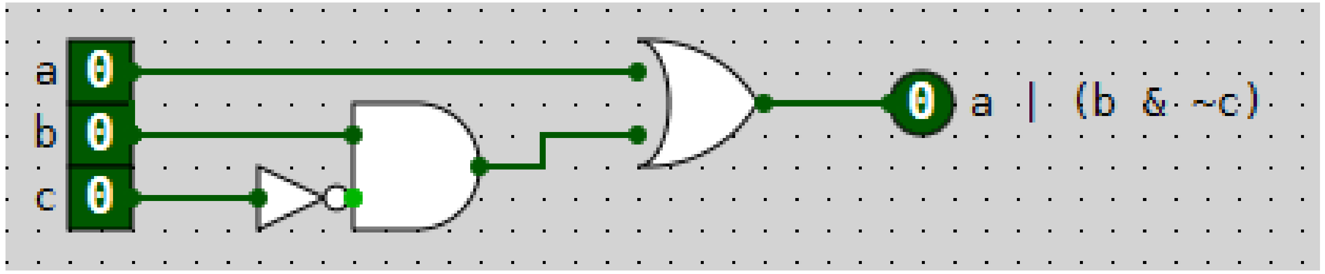


Converting Boolean Expression: A | (B & ~C)

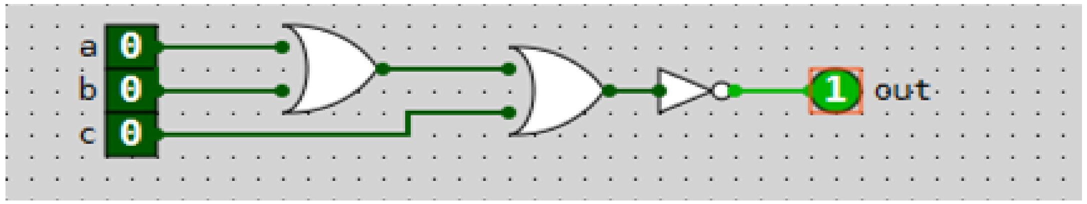
A	B	C	Func(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Fill out the first 3 columns of the table and plug the inputs into the expression.

To build the circuit, you can draw each of the gates from the expression and connect the pins



Converting from a Circuit



From a circuit, we can easily get the expression by looking at the gates:

~((A | B) | C)

This is on your attendance quiz!

We can use the same steps from the previous slide to build the truth table from the expression

A	B	C	Func(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

DeMorgan's Law

$$(\text{not A and not B}) = \text{not (A or B)}$$

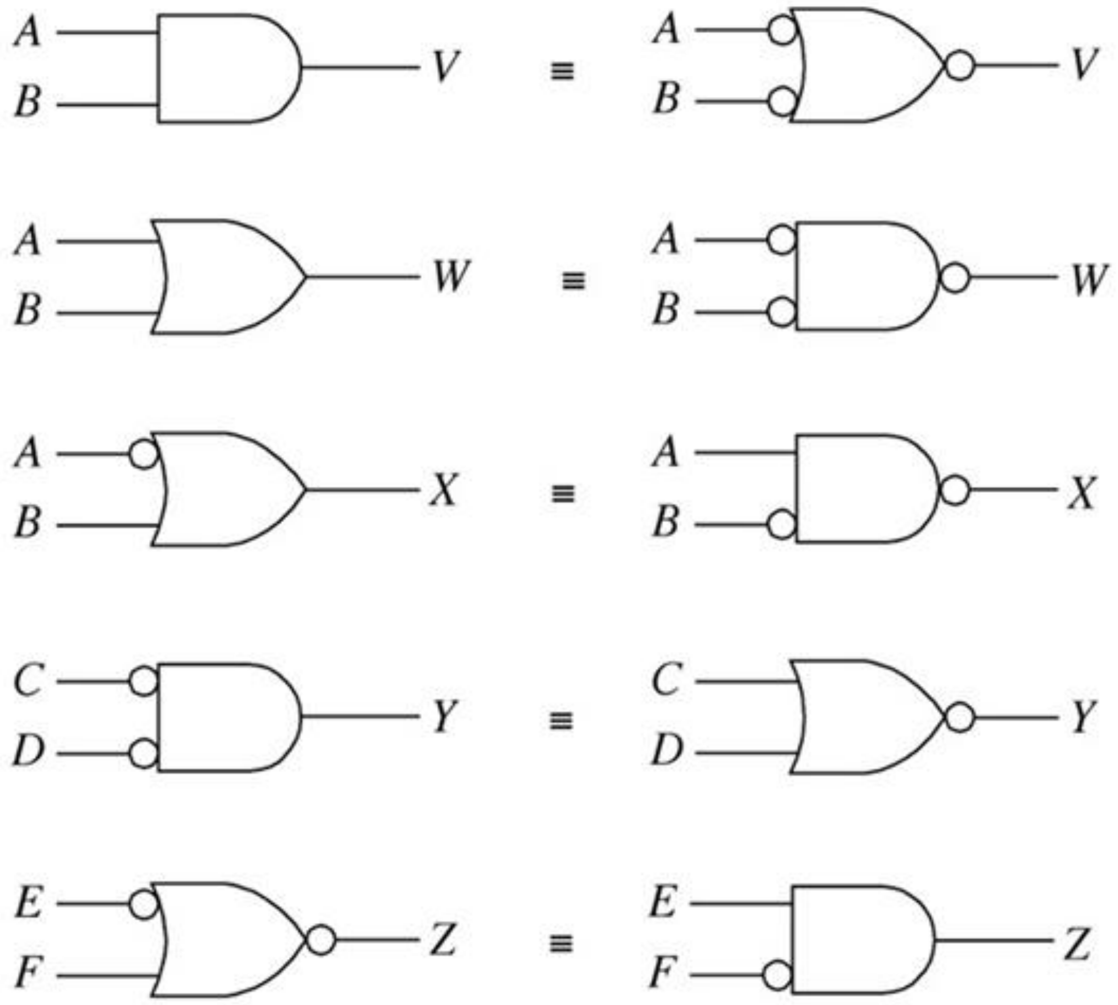
$$\overline{A} \cdot \overline{B} = \overline{A + B}$$

$$(\text{not A or not B}) = \text{not (A and B)}$$

$$\overline{A} + \overline{B} = \overline{A \cdot B}$$

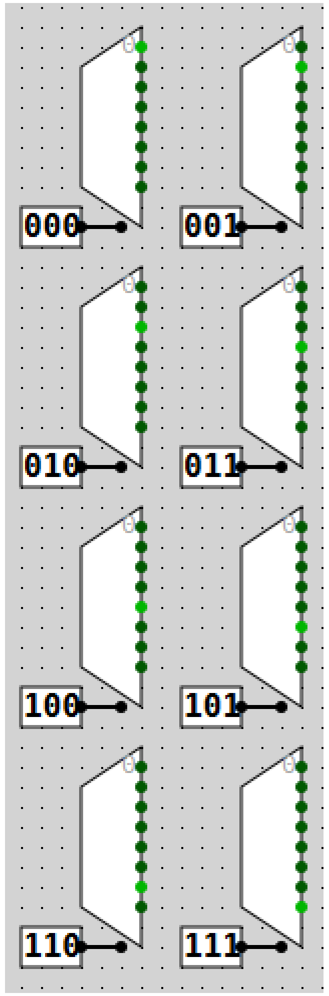
Conte Bubble Theorem

- Switch AND & OR
- Bubble → No Bubble
- No Bubble → Bubble
- They are equivalent!



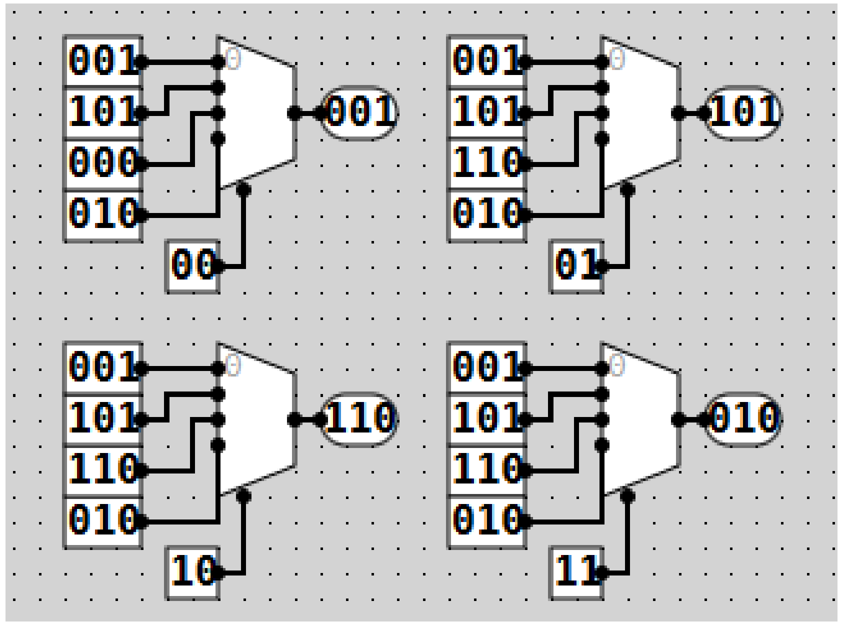
Decoders

- Sets exactly one output based on which of the input bits are set
- If there are *n* input bits then there are 2^n outputs. (Why?)



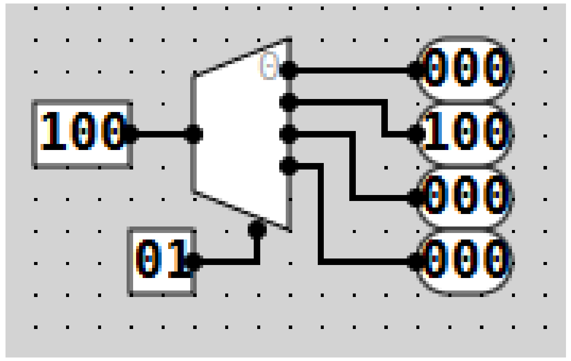
Multiplexer (mux)

- Selects between inputs using a selector
- If there are 2^n inputs, then there are n selector bits
- There is always just one output



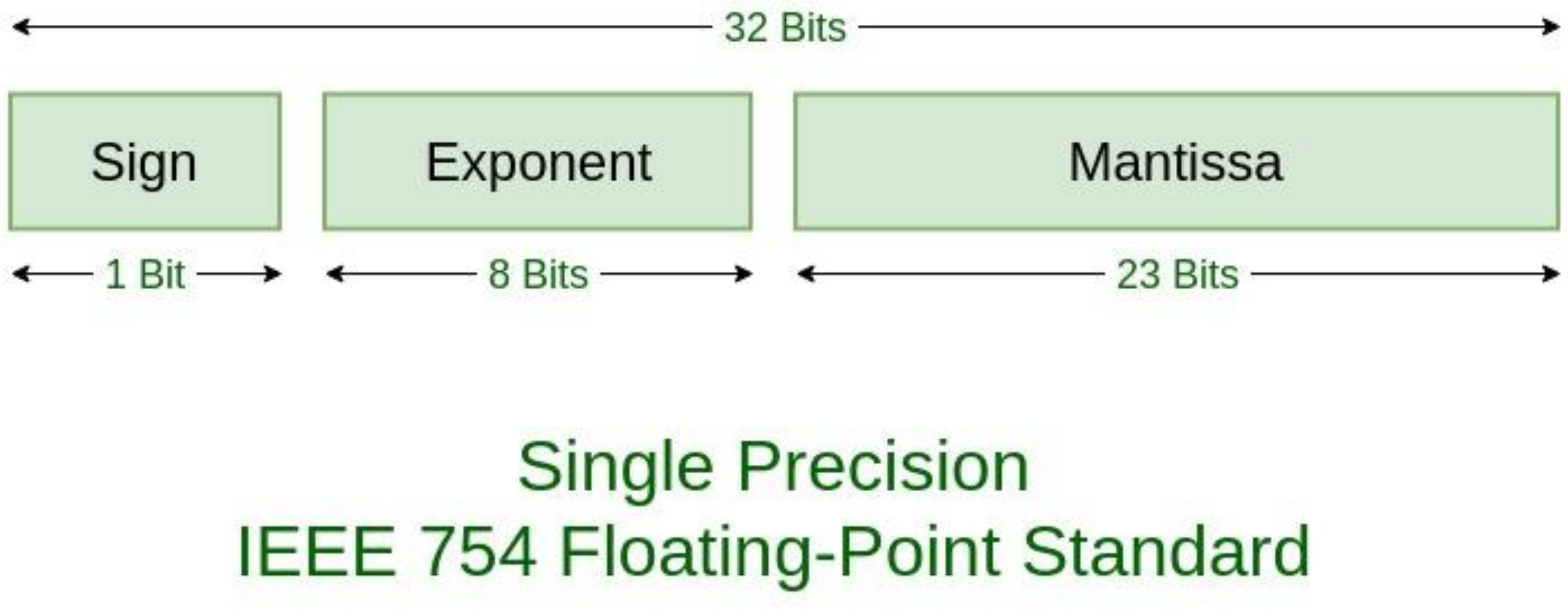
Demultiplexer (demux)

- Sends the input across exactly one of the output lines
- Other outputs remain zero
- If there are 2^n outputs, then there are *n* selector bits
- There is always just one input



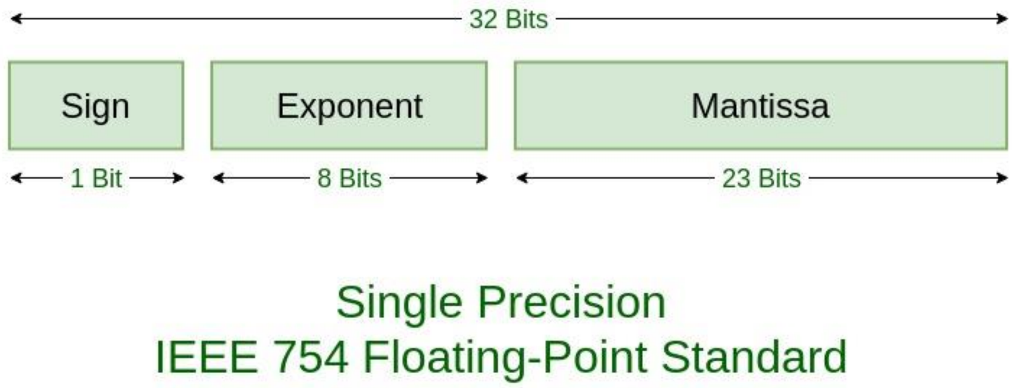
IEEE 754 Floating Point Numbers

Formula: (-1)^S * *1.M* * *2^(E-127)*



Exponent

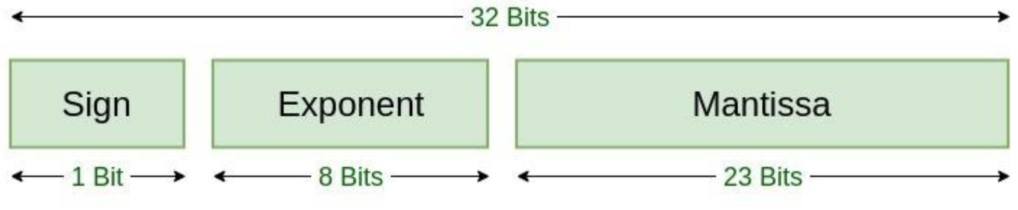
- Unsigned 8-bit integer, minus 127
- What is the range of the exponent?
  - Normal numbers have a range from -126 to +127.
  - Exponents -127 and +128 are reserved for special cases.



Mantissa

- 23-bit fractional part of a number
- In the form 1.M; the 1 is implied and not stored
- Example: 1.625 corresponds to:

  10100000000000000000000



Single Precision
IEEE 754 Floating-Point Standard

Conversion Practice

- What is the decimal representation of the following IEEE-754 floating point number?
- 11000000011000000000000000000000

How to compare?

- Treat like a signed, 32 bit whole number
- Go bitwise
- This is why the signed exponent is not defined using 2s complement—it allows us to compare bitwise, rather than having to do a 2’s complement comparison on the exponent portion

```
01101100010100001000000000101000
00101100010100001000000000101000
```

IEEE 754 Edge Cases

Formula: (-1)^S * *1.M * 2^(E-127)*

	E==0	0<E<255	E==255
M==0	**0**	**Powers of 2**	**infinity**
M!=0	**Non-normalized**	**Regular numbers**	**NaN**

Non-normalized formula: (-1)^S * *0.M * 2^(-126)*

note: 0.M and 2^(-126) instead of 1.M and 2^(-127)

Conversion Practice

- Interpret the following hexadecimal value as floating point using the IEEE-754 standard.
- x419B0000

Conversion Practice

- Convert the following **floating point** value to Binary and Hexadecimal using the **IEEE-754** standard.
- 13.25