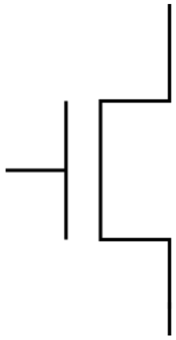
COMPUTER SYSTEMS AND ORGANIZATION

Part 1

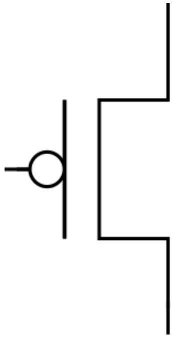
Daniel Graham

University of Virginia | ENGINEERING

REVIEW

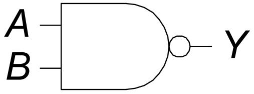


push to close



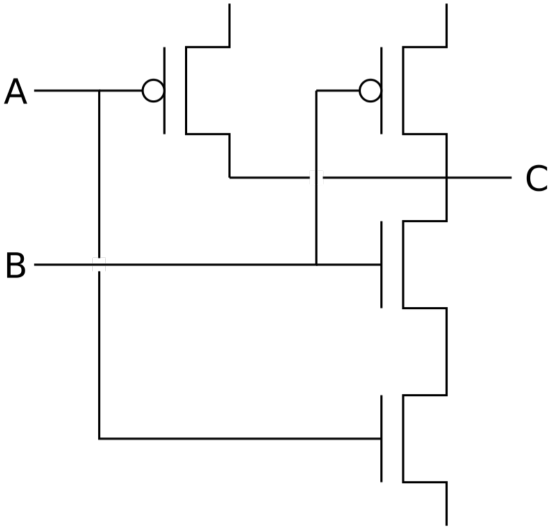
push to open

NAND



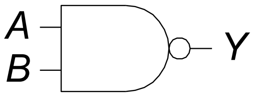
$Y = \overline{AB}$

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



Y

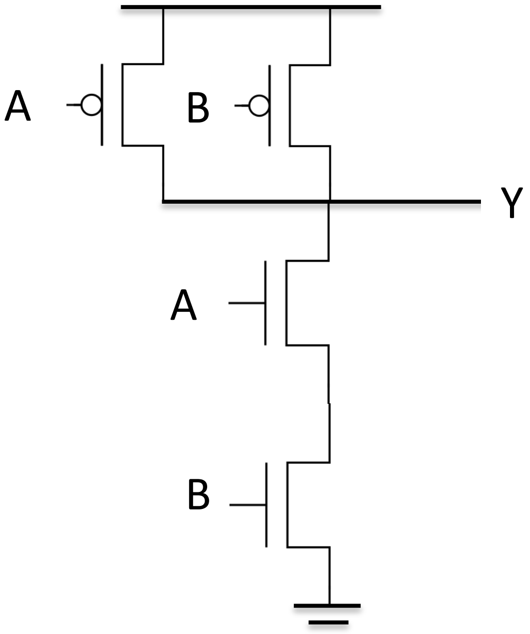
NAND



$Y = \overline{AB}$

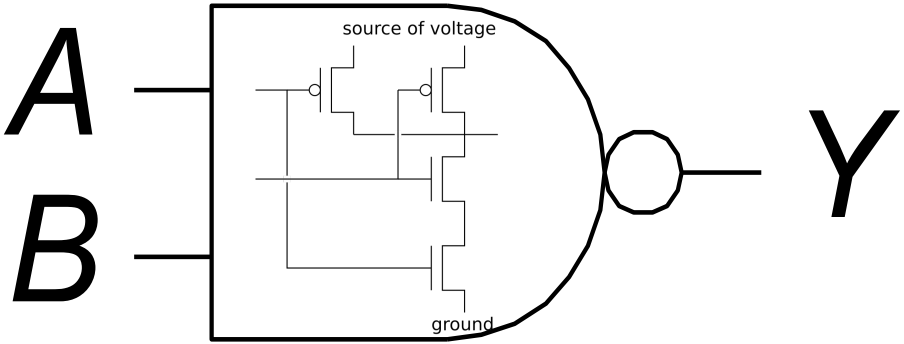
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Source of Voltage (VDD)



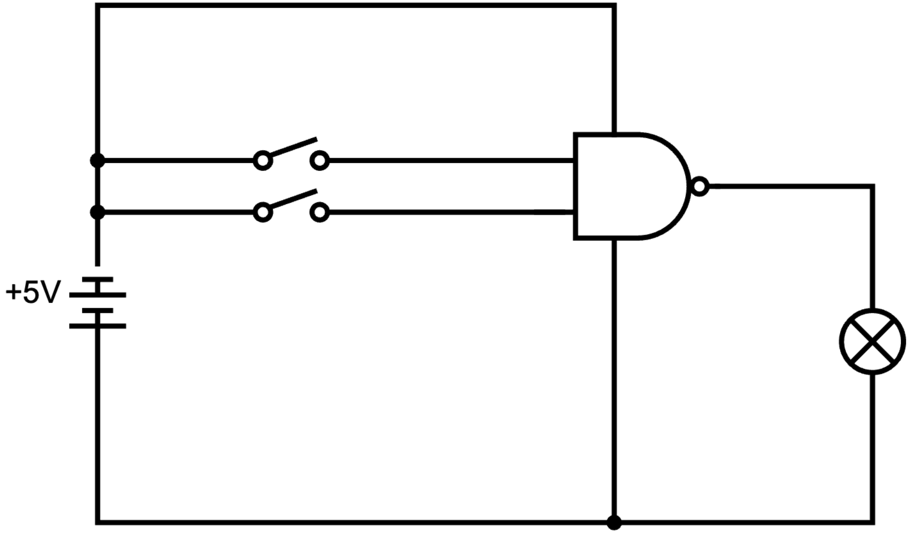
Ground

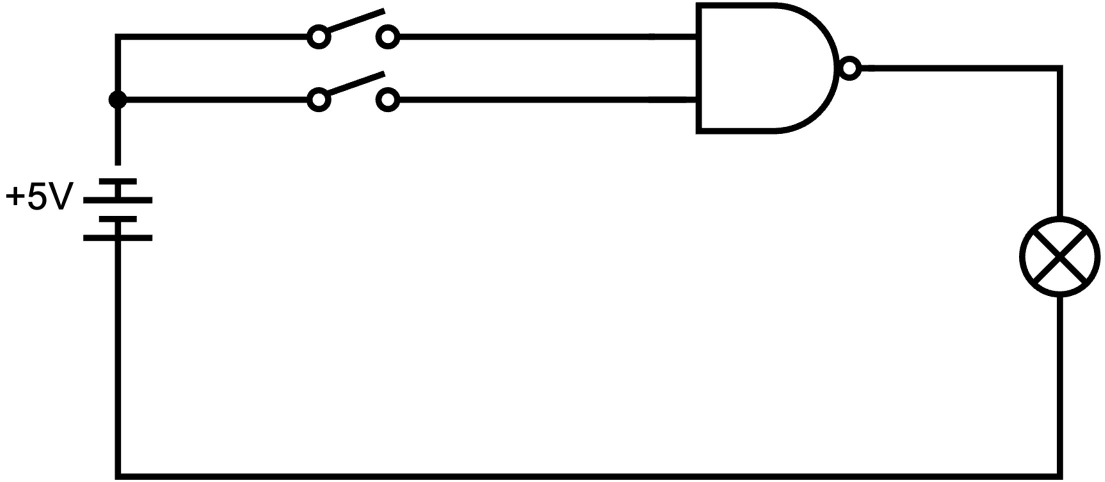
NAND



$$Y = \overline{AB}$$

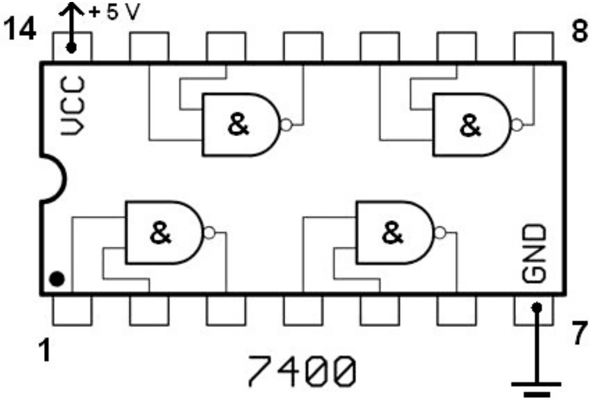
Circuit with a NAND Gate

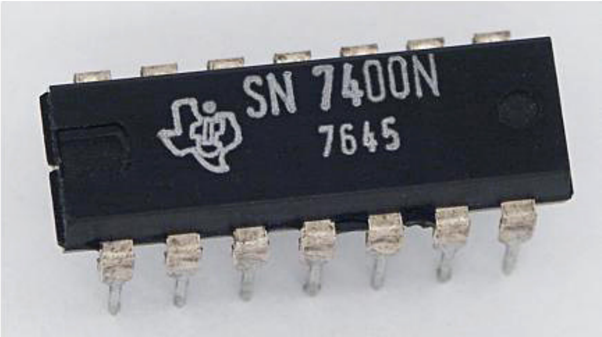


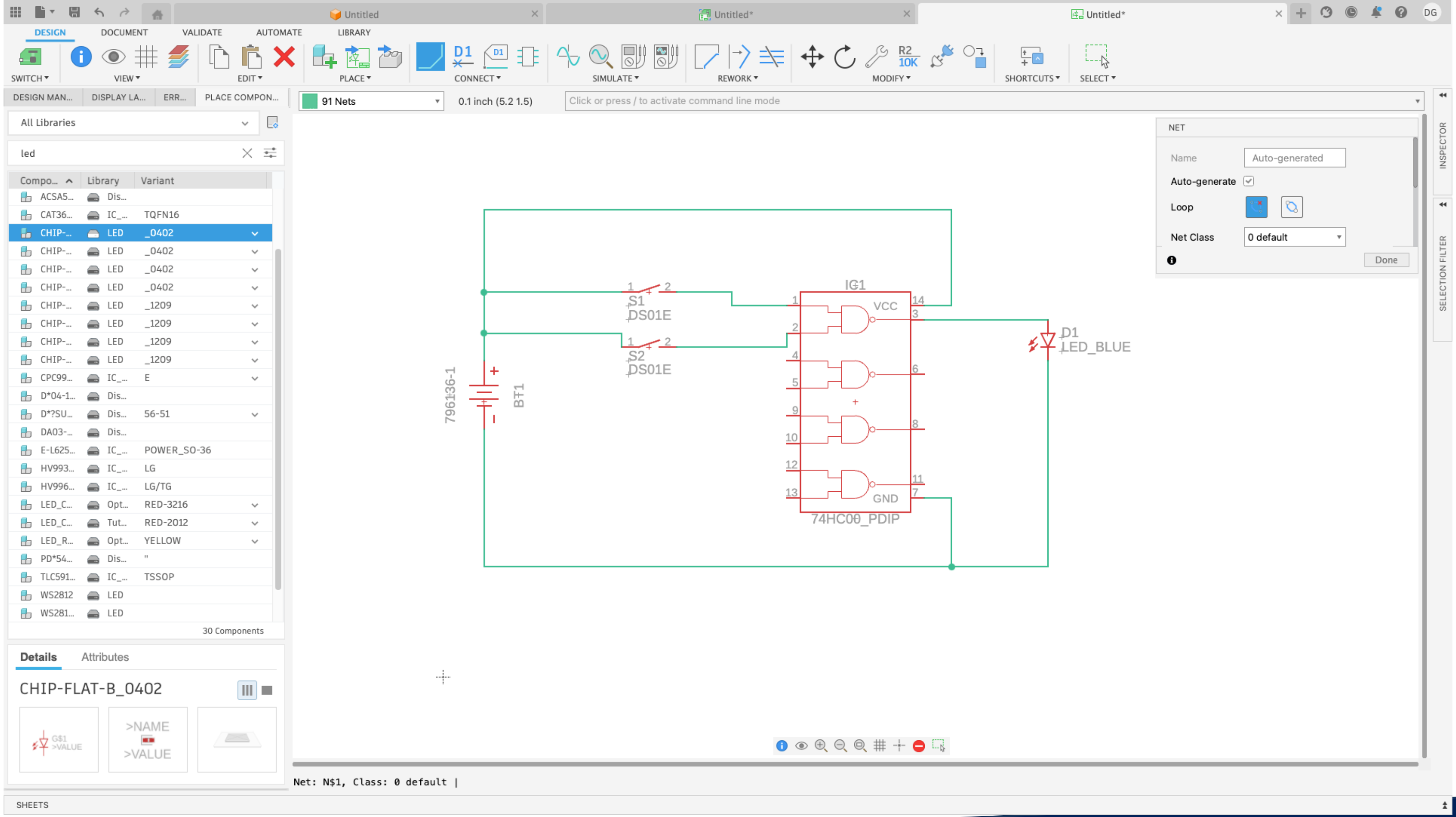


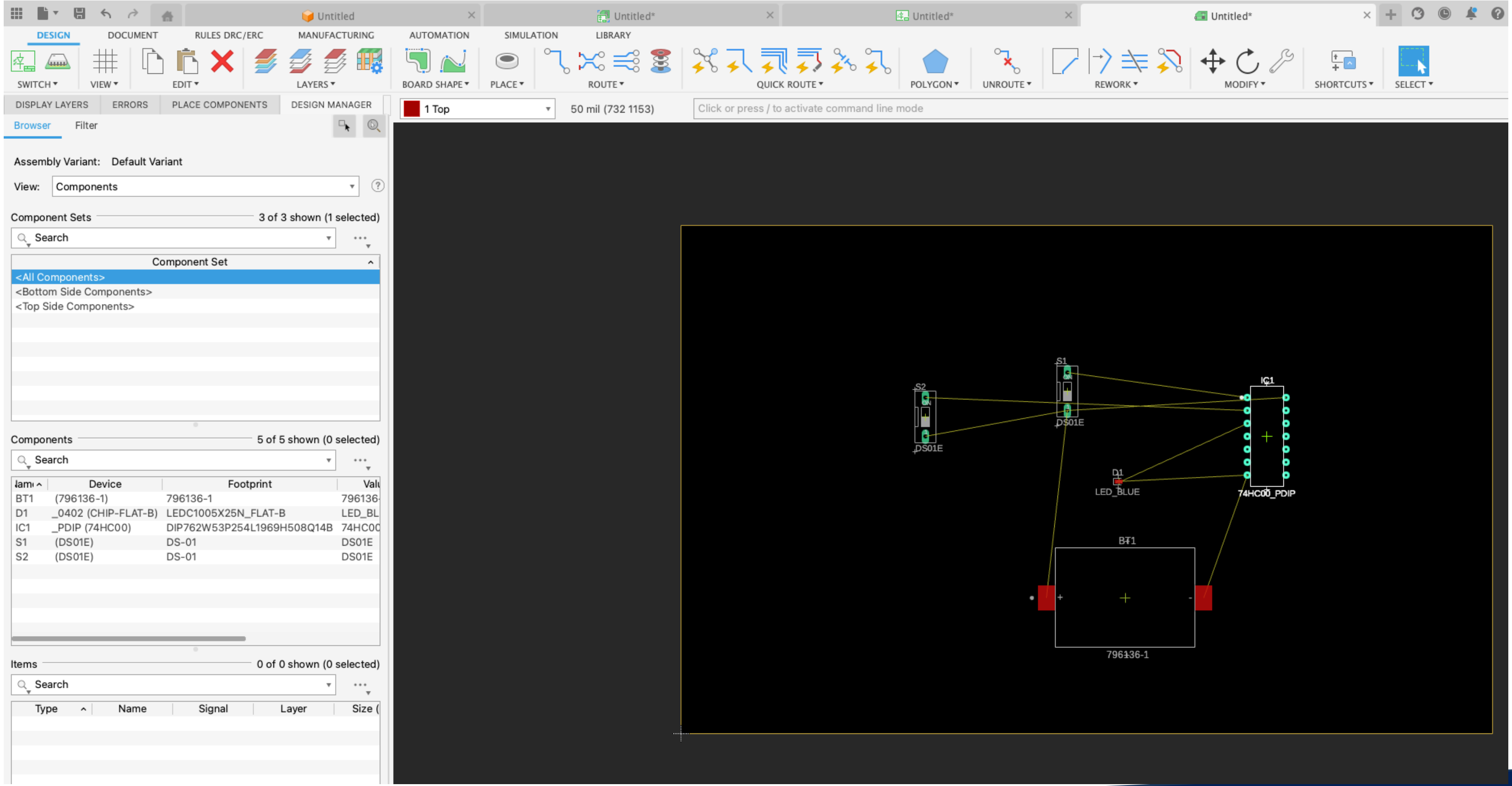
We don't normally draw the voltage and ground with gates. So you will normally see circuits that look like this.

CHIPS WITH NAND GATES

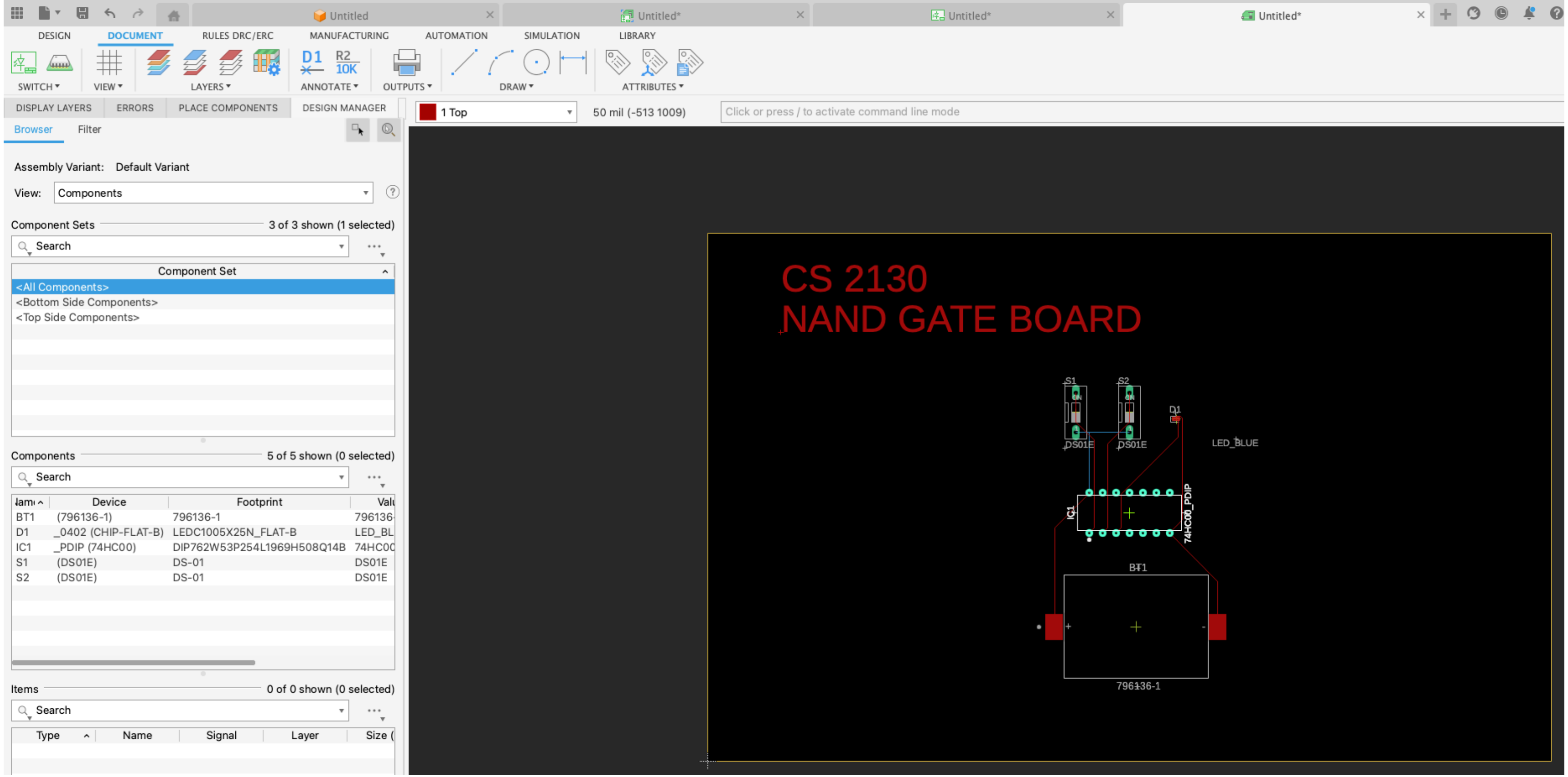


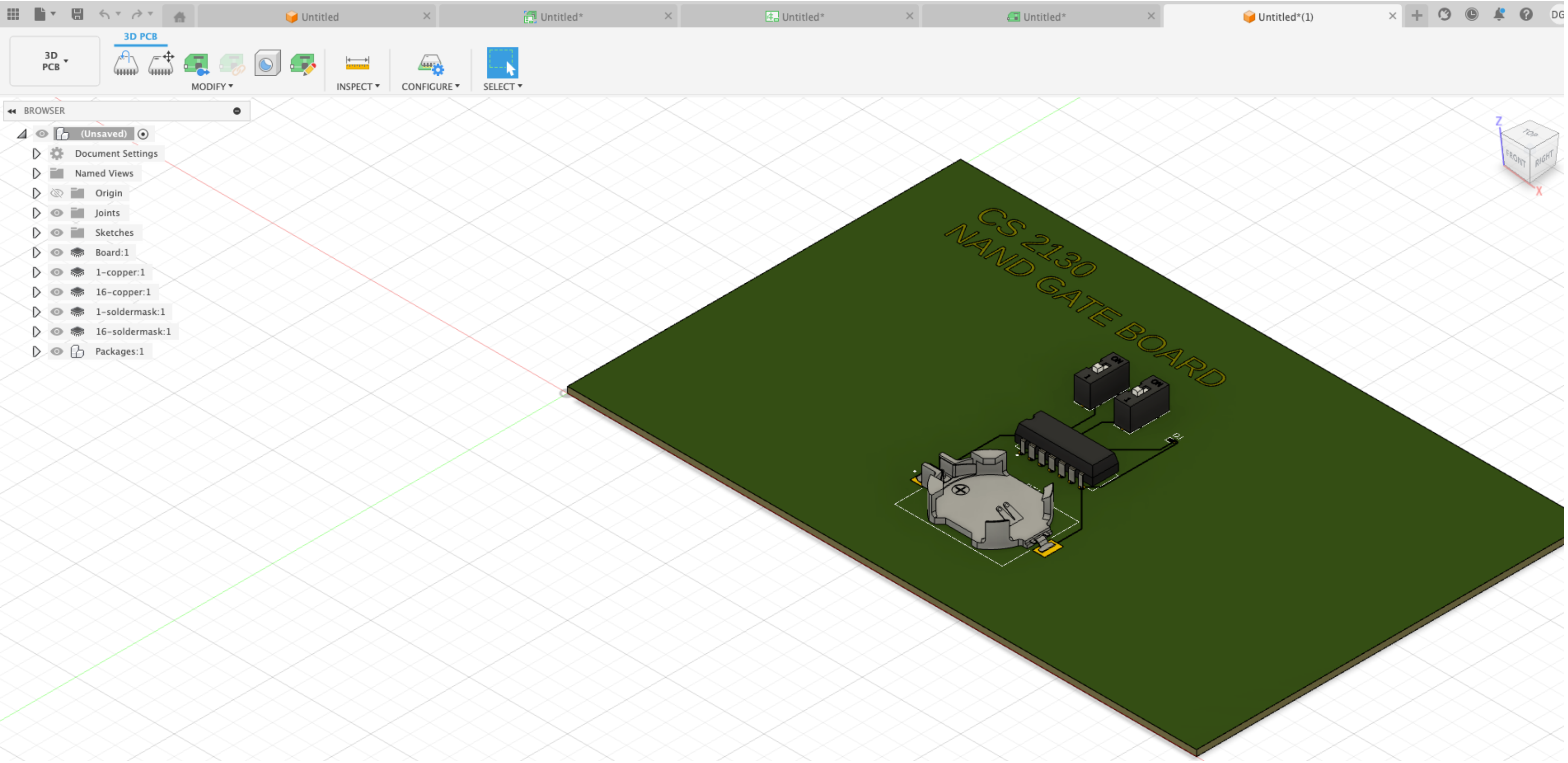



Untitled
Untitled*
Untitled*
DESIGN
DOCUMENT
VALIDATE
AUTOMATE
LIBRARY
SWITCH
VIEW
EDIT
PLACE
CONNECT
SIMULATE
REWORK
MODIFY
SHORTCUTS
SELECT
DESIGN MAN...
DISPLAY LA...
ERR...
PLACE COMPON...
91 Nets
0.1 inch (5.2 1.5)
Click or press / to activate command line mode
All Libraries
led
Compo...
Library
Variant
ACSA5... Dis...
CAT36... IC_... TQFN16
CHIP-... LED _0402
CHIP-... LED _0402
CHIP-... LED _0402
CHIP-... LED _0402
CHIP-... LED _1209
CHIP-... LED _1209
CHIP-... LED _1209
CHIP-... LED _1209
CPC99... IC_... E
D*04-1... Dis...
D*?SU... Dis... 56-51
DA03-... Dis...
E-L625... IC_... POWER_SO-36
HV993... IC_... LG
HV996... IC_... LG/TG
LED_C... Opt... RED-3216
LED_C... Tut... RED-2012
LED_R... Opt... YELLOW
PD*54... Dis... "
TLC591... IC_... TSSOP
WS2812 LED
WS281... LED
30 Components
Details
Attributes
CHIP-FLAT-B_0402
>NAME
>VALUE
NET
Name
Auto-generated
Auto-generate
Loop
Net Class
0 default
Done
INSPECTOR
SELECTION FILTER
IC1
VCC
GND
74HC00_PDIP
S1
DS01E
S2
DS01E
796136-1
BT1
D1
LED_BLUE
Net: N$1, Class: 0 default |
SHEETS

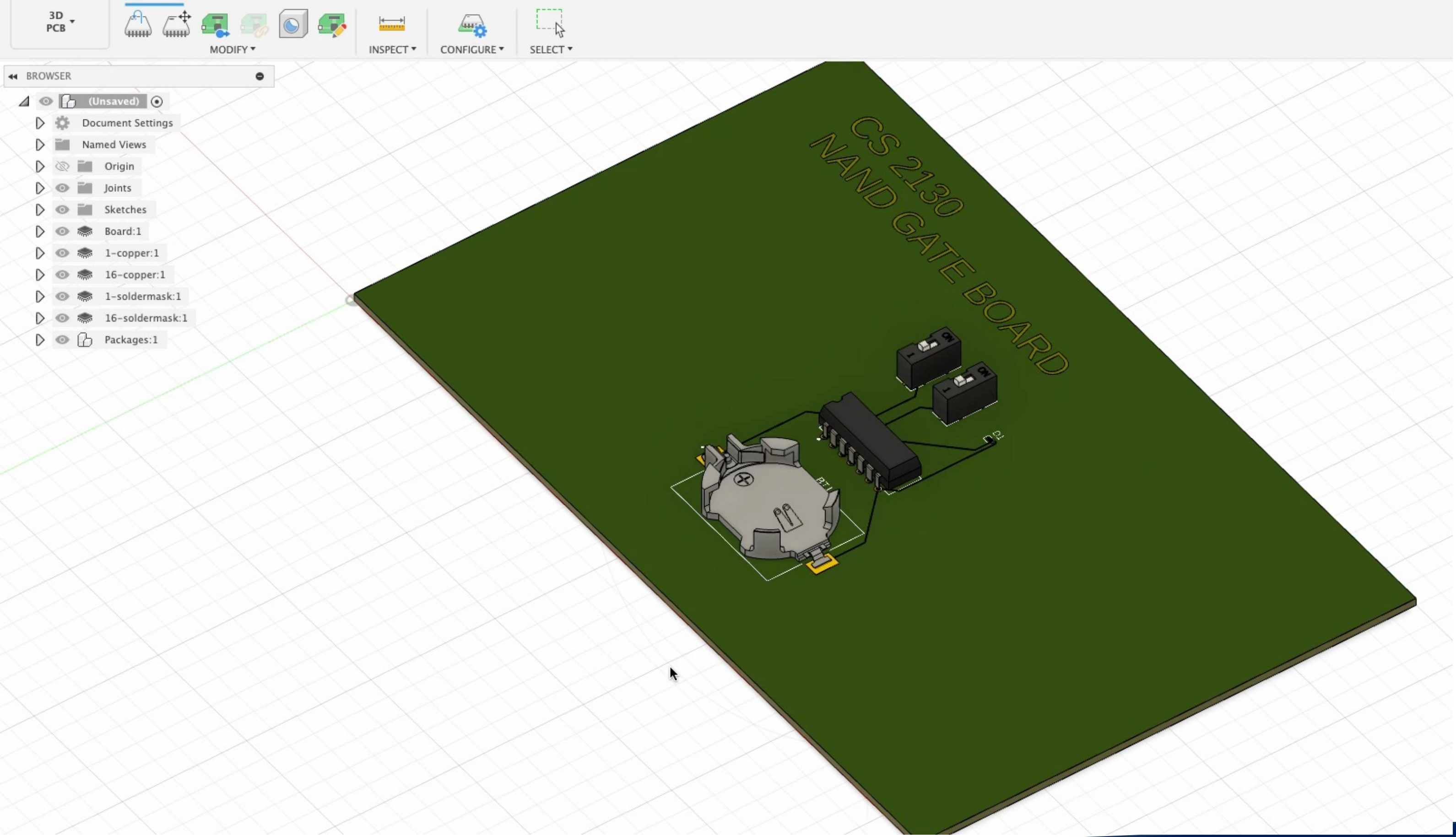


UNIVERSITY of VIRGINIA ENGINEERING


Untitled
Untitled*
Untitled*
Untitled*
DESIGN
DOCUMENT
RULES DRC/ERC
MANUFACTURING
AUTOMATION
SIMULATION
LIBRARY
SWITCH
VIEW
LAYERS
D1
R2
10K
ANNOTATE
OUTPUTS
DRAW
ATTRIBUTES
DISPLAY LAYERS
ERRORS
PLACE COMPONENTS
DESIGN MANAGER
1 Top
50 mil (-513 1009)
Click or press / to activate command line mode
Browser
Filter
Assembly Variant: Default Variant
View: Components
Component Sets
3 of 3 shown (1 selected)
Search
Component Set
<All Components>
<Bottom Side Components>
<Top Side Components>
Components
5 of 5 shown (0 selected)
Search
Device
Footprint
BT1 (796136-1) 796136-1 796136
D1 _0402 (CHIP-FLAT-B) LEDC1005X25N_FLAT-B LED_BL
IC1 _PDIP (74HC00) DIP762W53P254L1969H508Q14B 74HC0C
S1 (DS01E) DS-01 DS01E
S2 (DS01E) DS-01 DS01E
Items
0 of 0 shown (0 selected)
Search
Type
Name
Signal
Layer
Size
CS 2130
NAND GATE BOARD
S1
S2
D1
DS01E
DS01E
LED_BLUE
IC1
74HC00_PDIP
BT1
796136-1

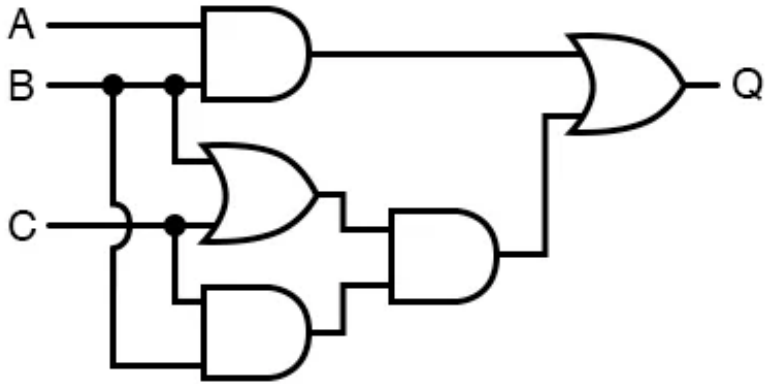


Untitled
Untitled*
Untitled*
Untitled*
Untitled*(1)
3D PCB
3D PCB
MODIFY
INSPECT
CONFIGURE
SELECT
BROWSER
(Unsaved)
Document Settings
Named Views
Origin
Joints
Sketches
Board:1
1-copper:1
16-copper:1
1-soldermask:1
16-soldermask:1
Packages:1
CS 2130
NAND GATE BOARD

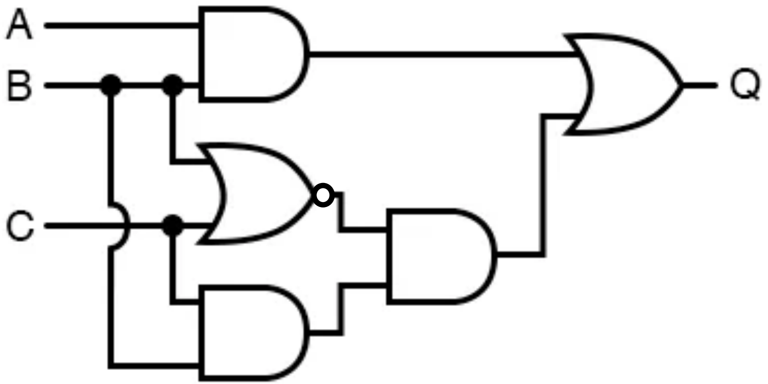

3D PCB
MODIFY
INSPECT
CONFIGURE
SELECT
BROWSER
(Unsaved)
Document Settings
Named Views
Origin
Joints
Sketches
Board:1
1-copper:1
16-copper:1
1-soldermask:1
16-soldermask:1
Packages:1
CS 2130
NAND GATE BOARD

WHAT IS THE OUTPUT OF THIS CIRCUIT?

A	*B*	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

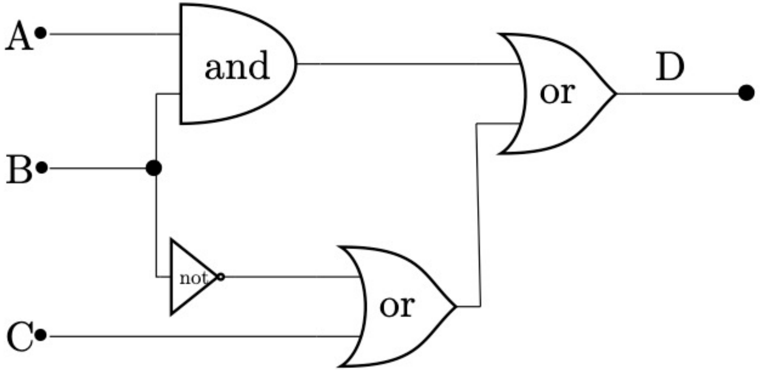


EXPRESS CIRCUIT AS AN EQUATION



Write the equation representing the circuit. Note I replaced the OR with a NOR.

EXAM QUESTION



SPRING 2022
Midterm 1

Fill in the following truth table for this circuit:

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

CREATIVE QUESTIONS

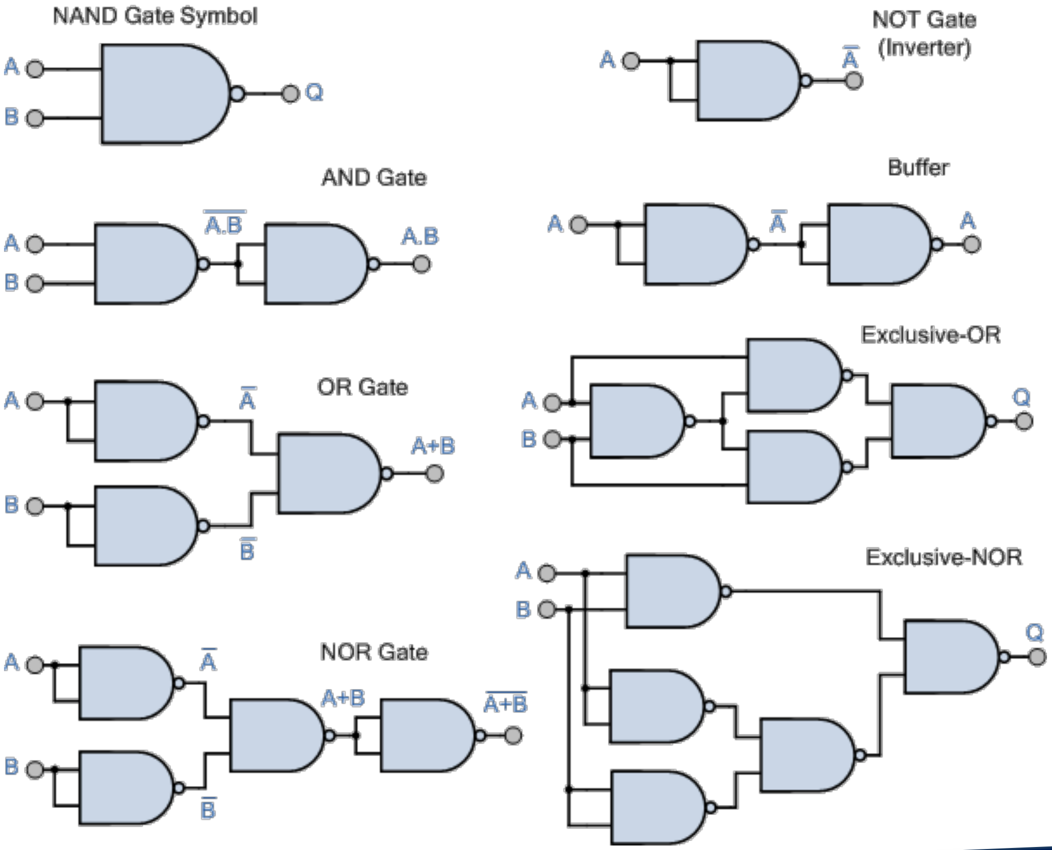
NAND GATES ARE TURNING COMPLETE

It is possible to implement every other gate by using a NAND. You can implement the complete RISC-V architecture using only NAND gates. What a beautiful building block right ☺

Hint: Start by asking NOT what a NAND gate can do for you but what you can do with a NAND gate.

Use a NAND gate to implement the following gates:

1. NOT
2. AND
3. OR
4. NOR
5. XOR


NAND Gate Symbol
A
B
Q
AND Gate
A.B
A.B
OR Gate
A
B
A+B
NOR Gate
A+B
A+B
NOT Gate
(Inverter)
A
Buffer
A
Exclusive-OR
Q
Exclusive-NOR
Q

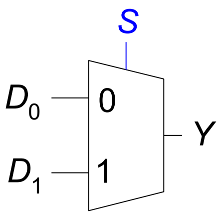
TODAY’S LECTURE

Contents

1. Building Component out of gates. (Muxes a case study)
2. Towards building a digital adding machine
3. How can we represent numbers? (What about decimals and negative numbers)

MUXES

Example: **2:1 Mux**



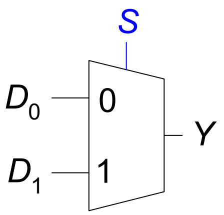
S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S	Y
0	D_0
1	D_1

- Selects between one of *N* inputs to connect to output
- **Select** input is **$\log_2 N$ bits** – control input

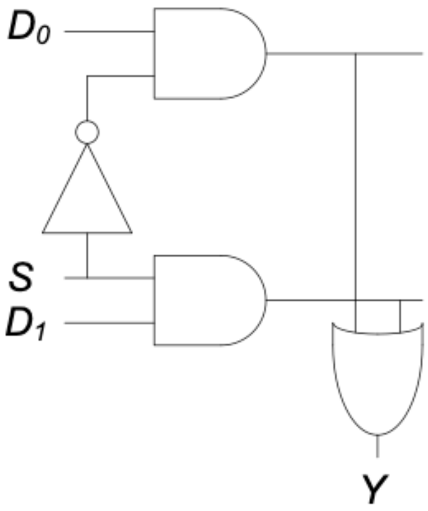
1 BIT MUX

$$Y = D_0\overline{S} + D_1 S$$



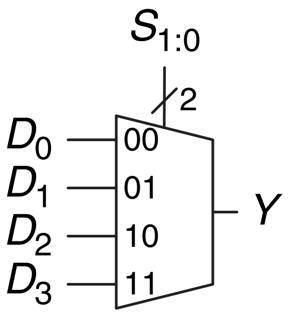
S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

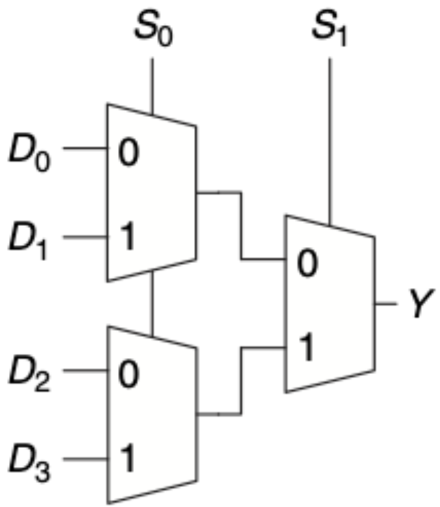
S	Y
0	D_0
1	D_1



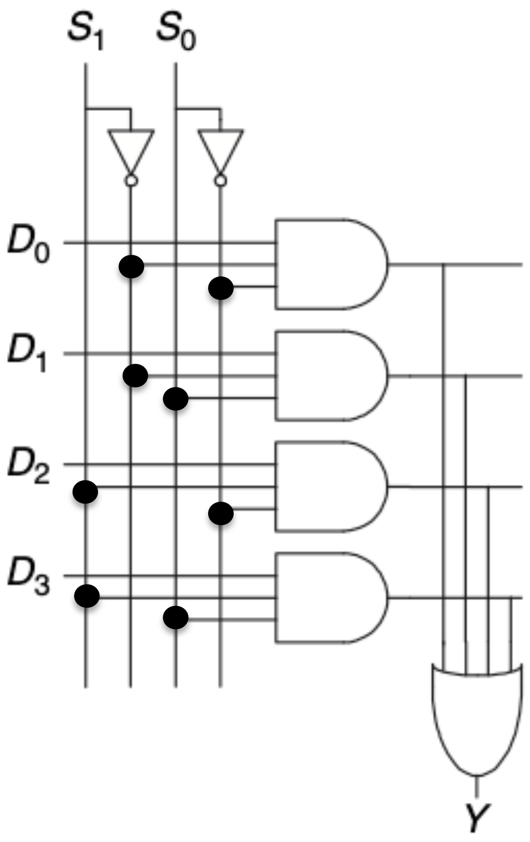
UNIVERSITY of VIRGINIA | ENGINEERING

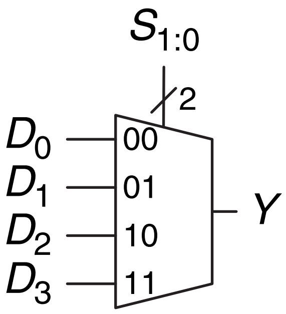
2 BIT MUX





2 BIT MUX



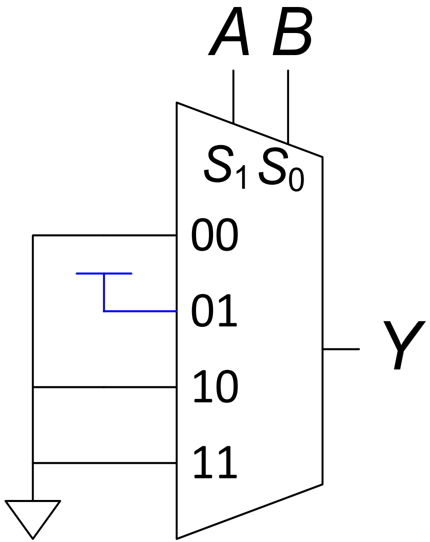


MUX AS A LOOK UP TABLE

Using mux as a **lookup table**

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

$$Y = \overline{A}B$$

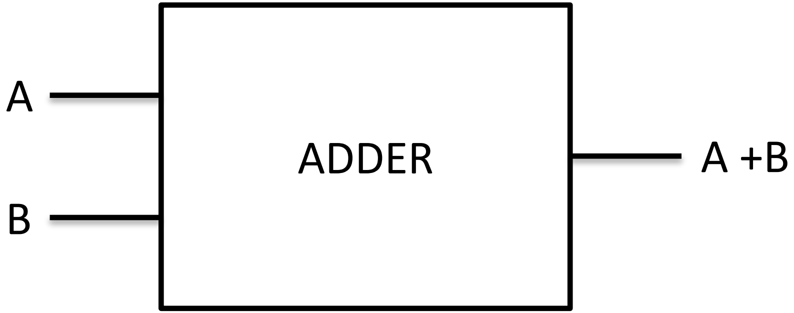


GREAT WE HAVE GATES

NOW LET’S BUILD SOMETHING.

HOW ABOUT A MACHINE THAT ADDS NUMBERS?

THE IDEA



THE CHALLENGE

Our gates only support 0 and 1s.

How can we represent other decimal numbers?

How can we present negative numbers?

What about fractions ☺?

DECIMAL

- Decimal numbers

1000's column
100's column
10's column
1's column

$$5374_{10} = 5 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

five thousands | three hundreds | seven tens | four ones

BINARY

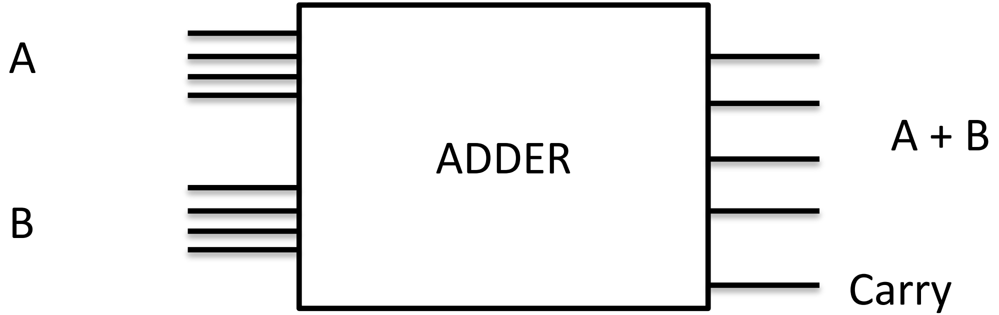
8's column
4's column
2's column
1's column

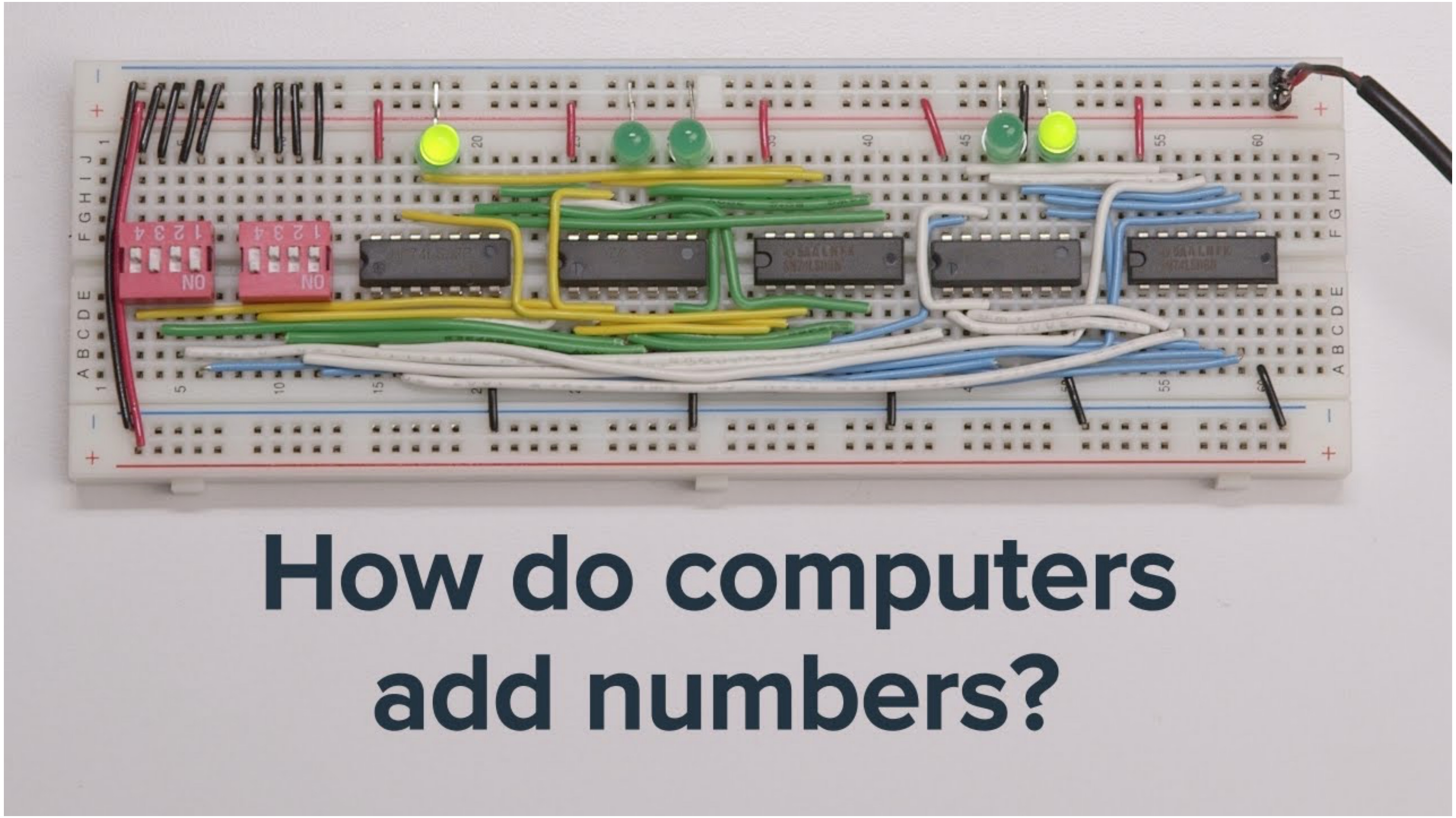
$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

one eight	one four	no two	one one

UNIVERSITY of VIRGINIA | ENGINEERING

4-BIT ADDER




How do computers
add numbers?

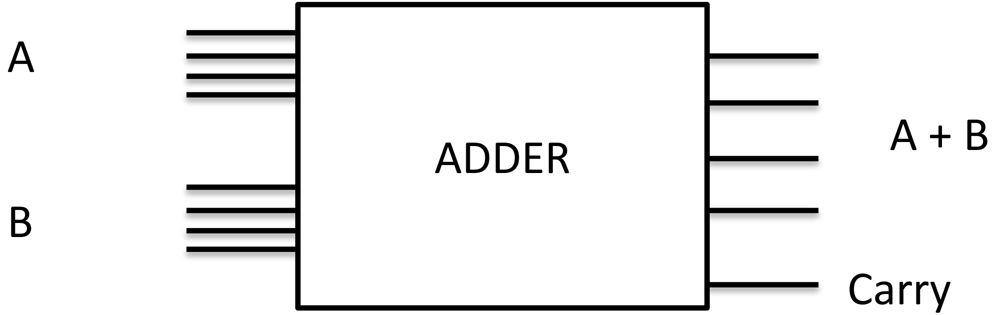
INPUTS AND OUTPUT OF OUR ADDER

What would the input be if wanted to add 5, and 9?

Notice we need to pick and order for the wires. More on this later ☺

Which output lights would we want to light

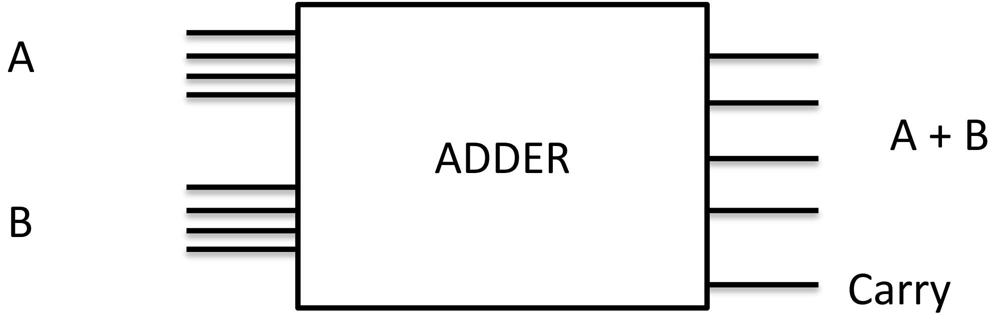
Up?



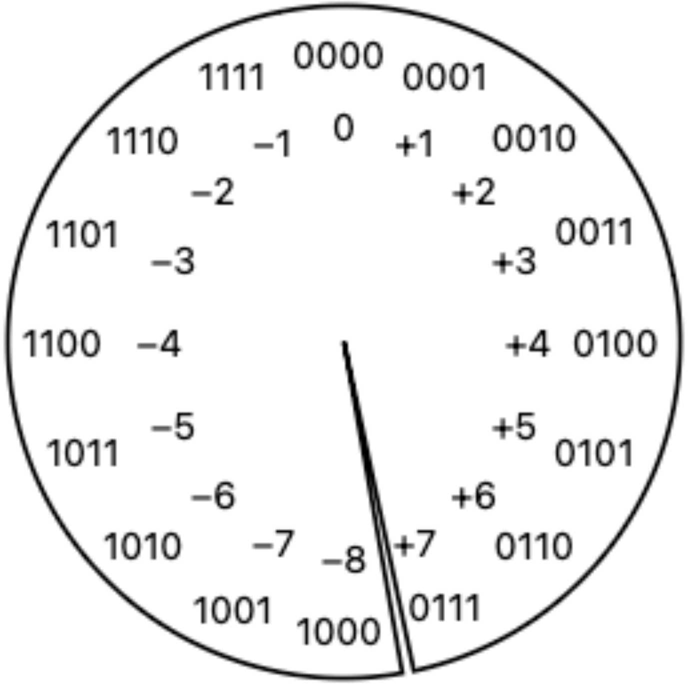
INPUTS AND OUTPUT OF OUR ADDER

What if we now added 7 and 9?

What would our inputs be and which lights do we expect to light up?



WHAT ABOUT NEGATIVE NUMBERS



Two’s complement picks a number (typically half of the maximum number we can write, rounded up) and decides that that number and all numbers bigger than it are negative

Two’s complement is nice because the three most common mathematical operators (addition, subtraction, and multiplication) work the same for signed and unsigned values. Division is messier, but division is always messy

EXAM REVIEW FALL 2018

The following assume 8-bit 2's-complement numbers. For each number, bit 0 is the low-order bit, bit 7 is the high-order bit.

Question 2 [2 pt]: (see above) Complete the following sum, showing your work (carry bits, etc)

```
  0 0 1 1 0 0 1 1
+ 0 1 1 0 0 1 1 0
-----------------
```

What is the result in base 10? Is it negative or positive? Would you get the same result in decimal if you had more bits ☺ ?

WRITING LONG BINARY IS NO FUN.

LET'S EXPRESS IT IN ANOTHER BASE TO MAKE EASIER. DEFINITELY CHOOSE SOMETHING LARGER THAN BASE 10

HEXADECIMAL

Hex Digit	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

UNIVERSITY of VIRGINIA | ENGINEERING