ECED2200 – Digital Circuits

Introduction, Gates, Number Systems

Colin O'Flynn - CC BY-SA

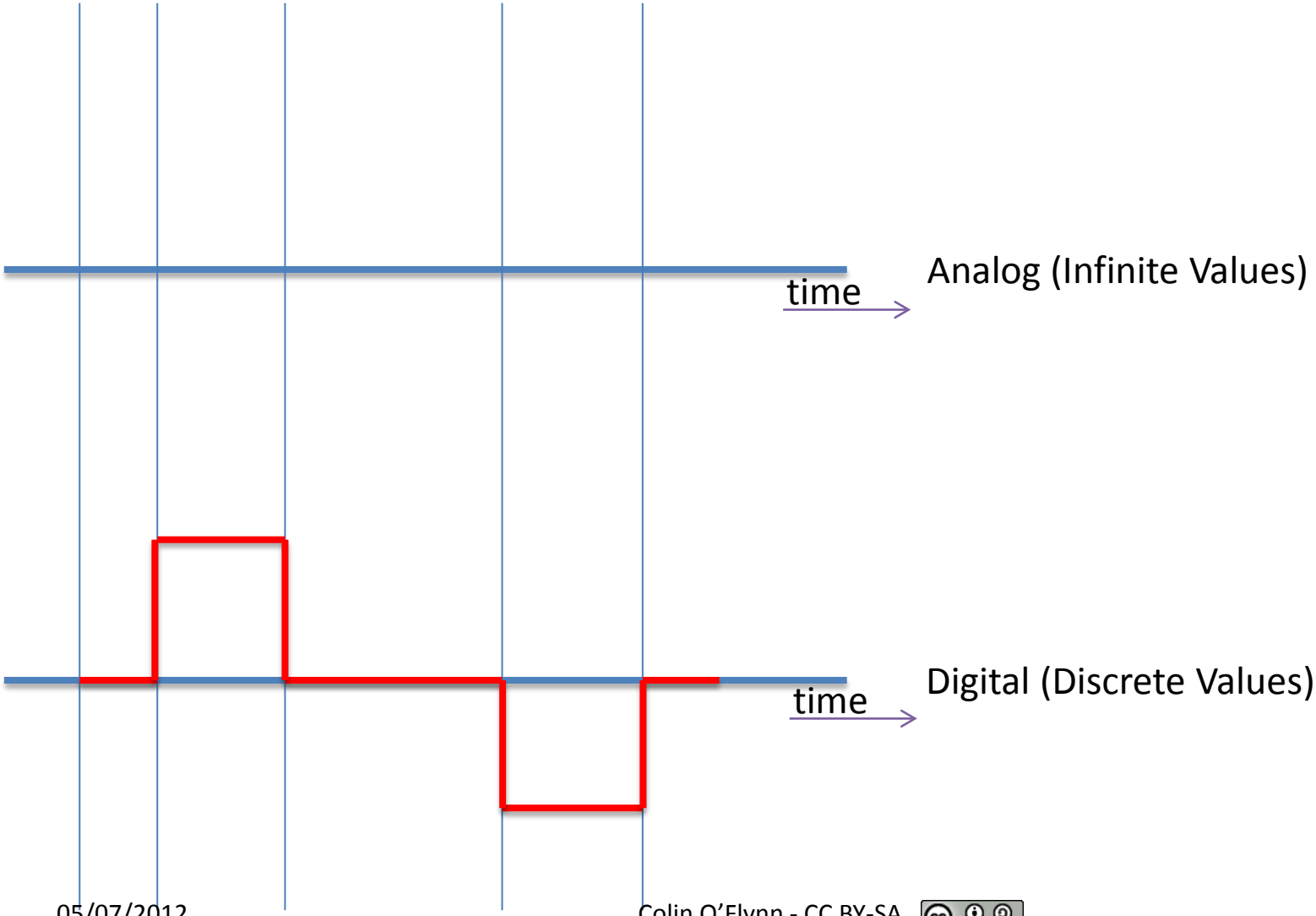
General Notes

- See updates to these slides: www.newae.com/teaching
- These slides licensed under 'Creative Commons Attribution-ShareAlike 3.0 Unported License'
- These slides are not the complete course – they are extended in-class
- You will find the following references useful, see www.newae.com/teaching for more information/links:
  - The book "Bebop to the Boolean Boogie" which is available to Dalhousie Students
  - Course notes (covers almost everything we will discuss in class)
  - Various websites such as e.g.: www.play-hookey.com
  - The book "Contemporary Logic Design", which was used in previous iterations of the class and you may have already

INTRODUCTION, BINARY, AND GATES

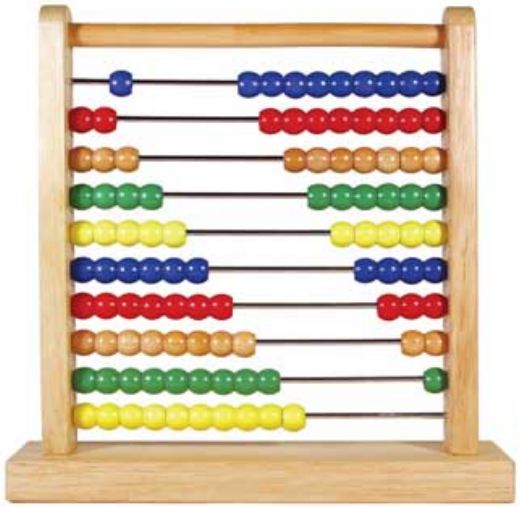
Colin O'Flynn - CC BY-SA

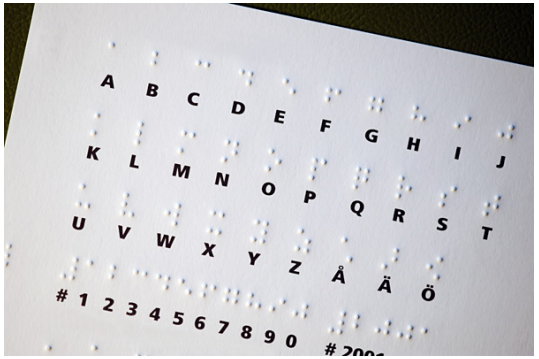
Analog vs. Digital



Colin O'Flynn - CC BY-SA

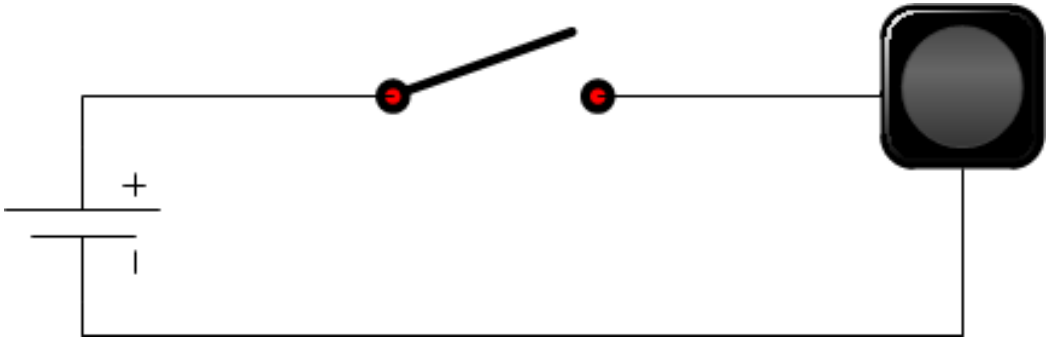
Digital Systems



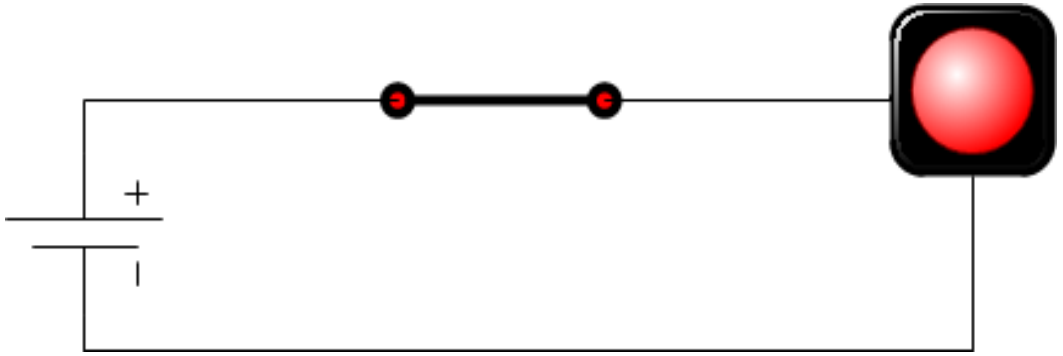


Colin O'Flynn - CC BY-SA

Binary



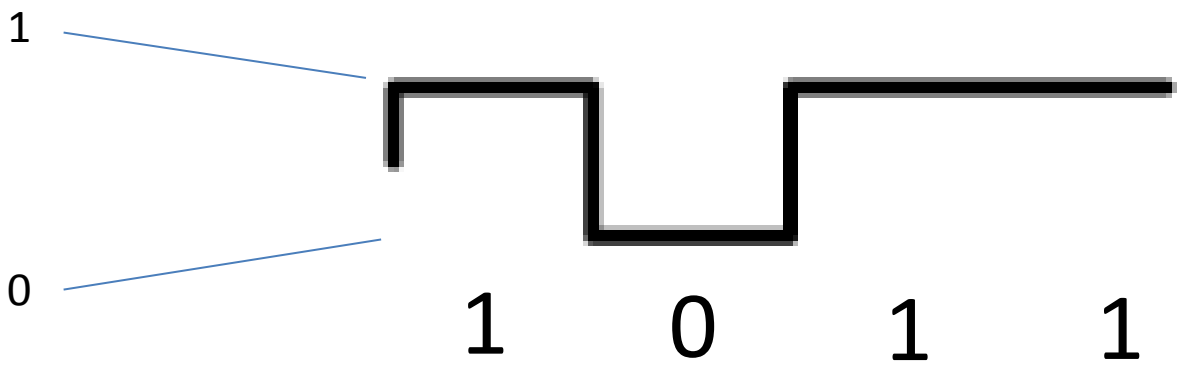
Off
False
0
Low



On
True
1
High

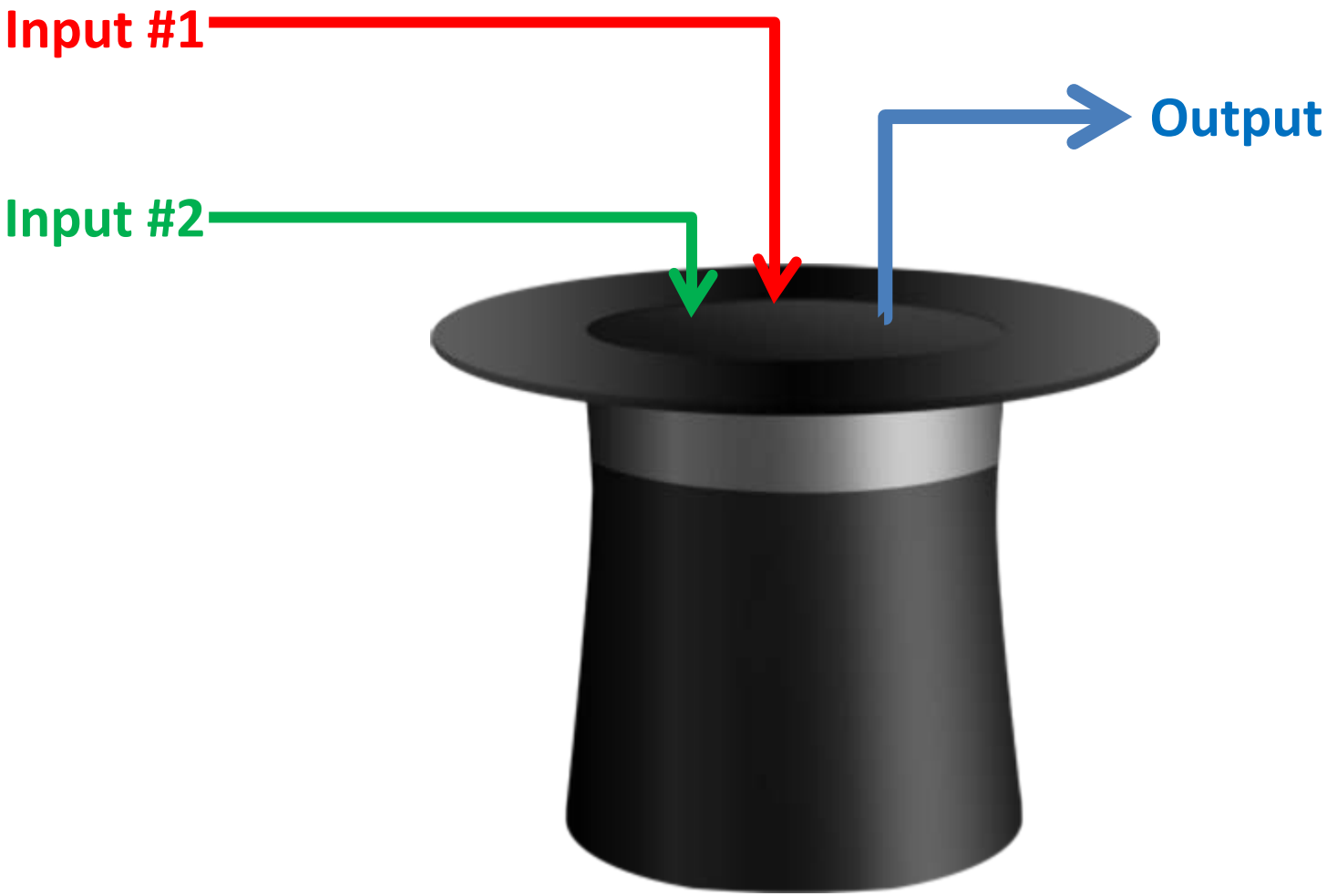
Colin O'Flynn - CC BY-SA

Binary – Waveforms in Time



Colin O'Flynn - CC BY-SA

GATES



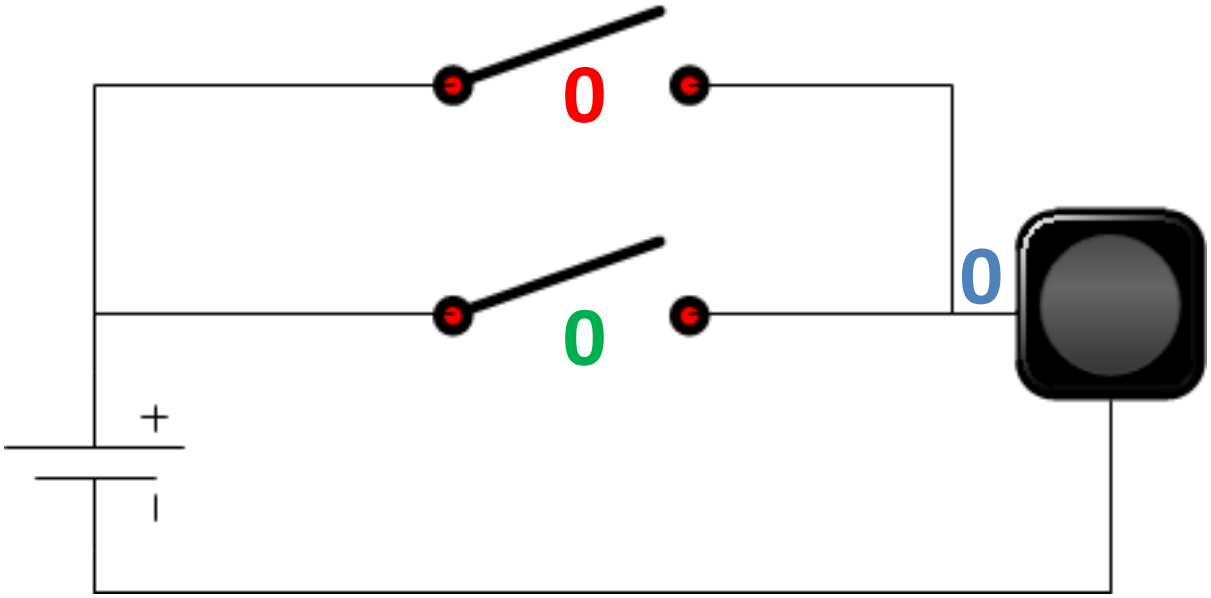
Colin O'Flynn - CC BY-SA

OR Gate – Truth Table

Input #1	Input #2	Output
0	0	0
0	1	1
1	0	1
1	1	1

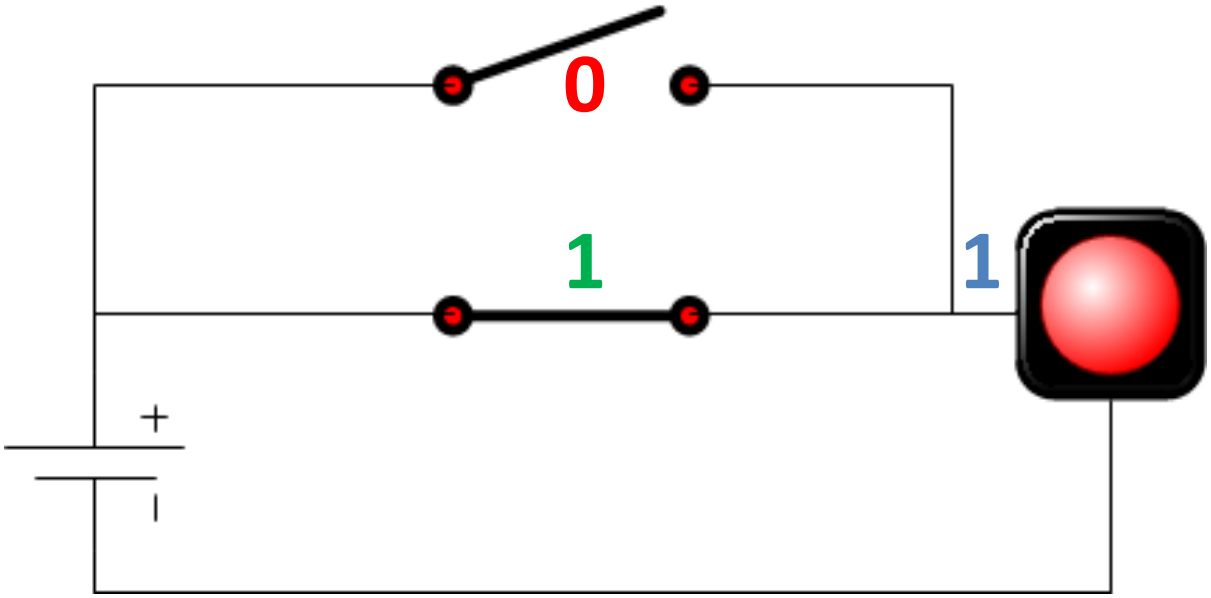
Colin O'Flynn - CC BY-SA

OR Gate



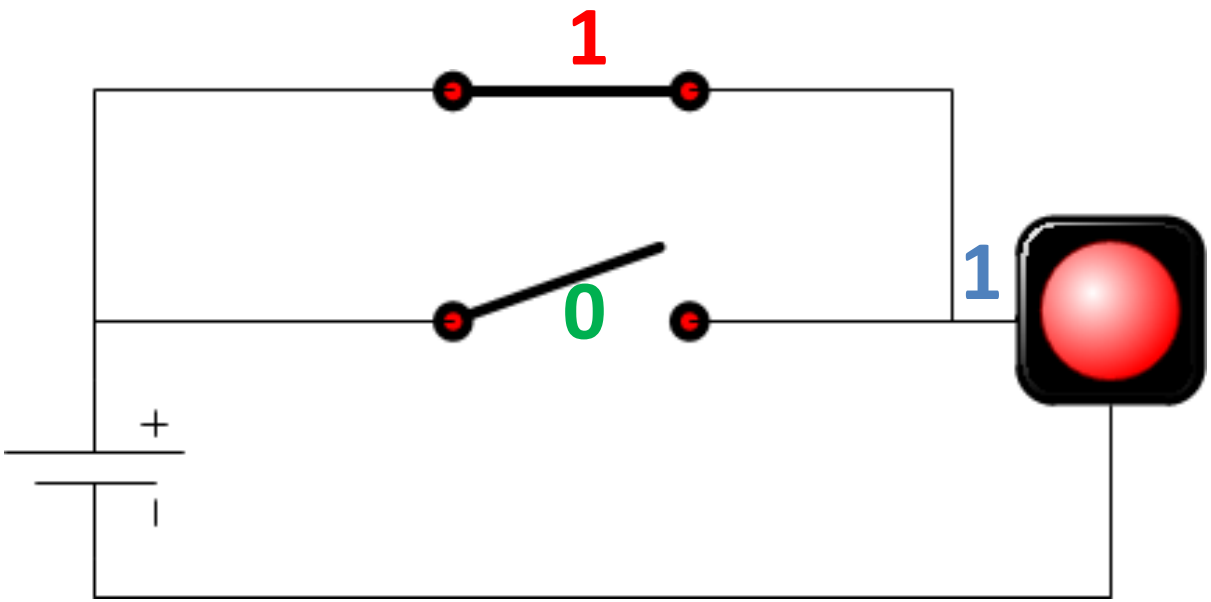
Colin O'Flynn - CC BY-SA

OR Gate



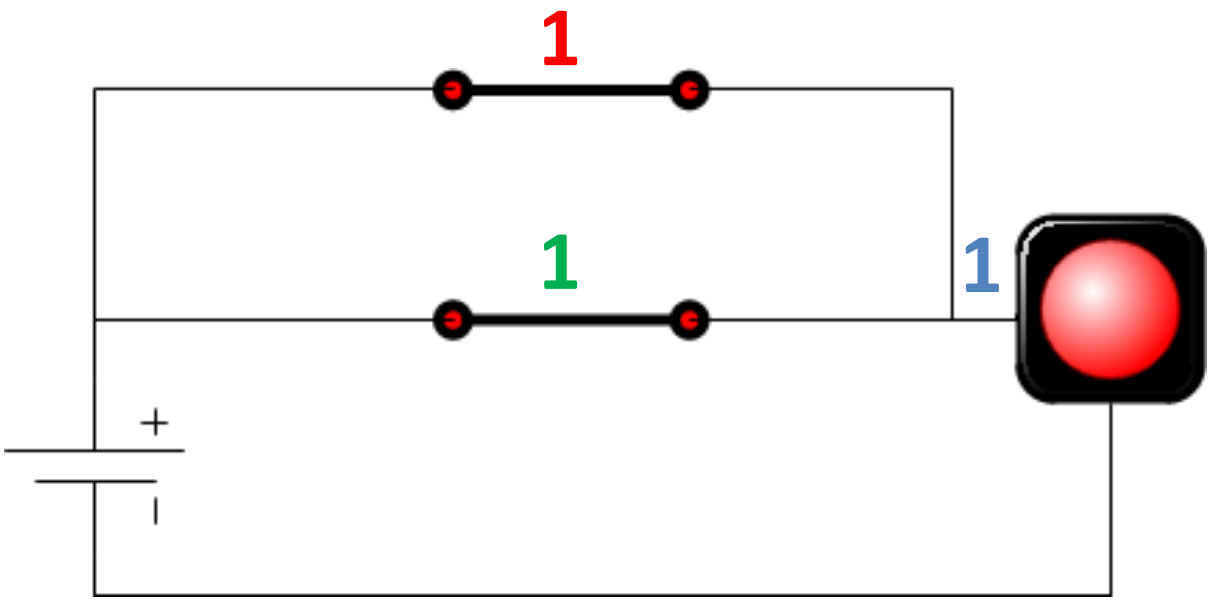
Colin O'Flynn - CC BY-SA

OR Gate



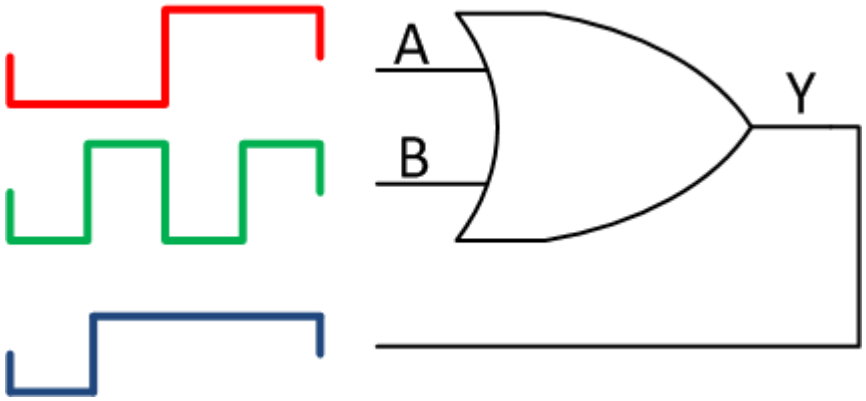
Colin O'Flynn - CC BY-SA

OR Gate



Colin O'Flynn - CC BY-SA

OR Gate

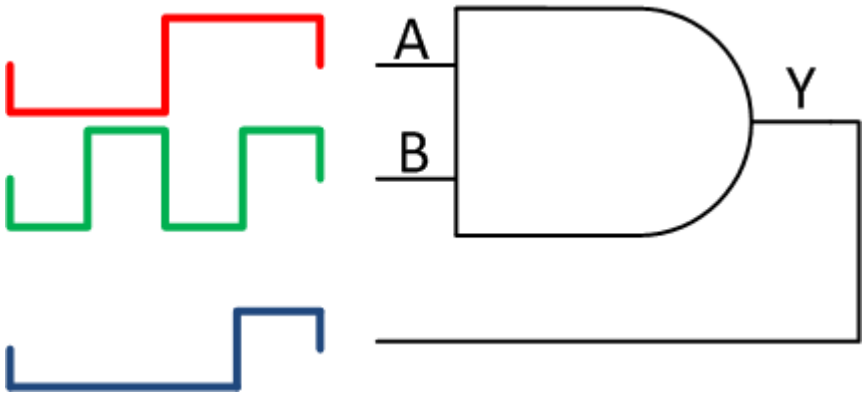


$$Y=A+B$$

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

Colin O'Flynn - CC BY-SA

AND Gate

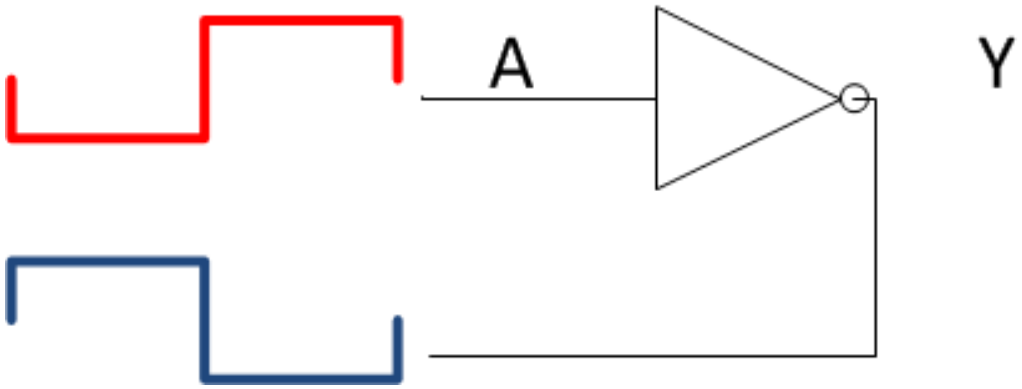


$$Y = A \bullet B$$

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

Colin O'Flynn - CC BY-SA

NOT Gate

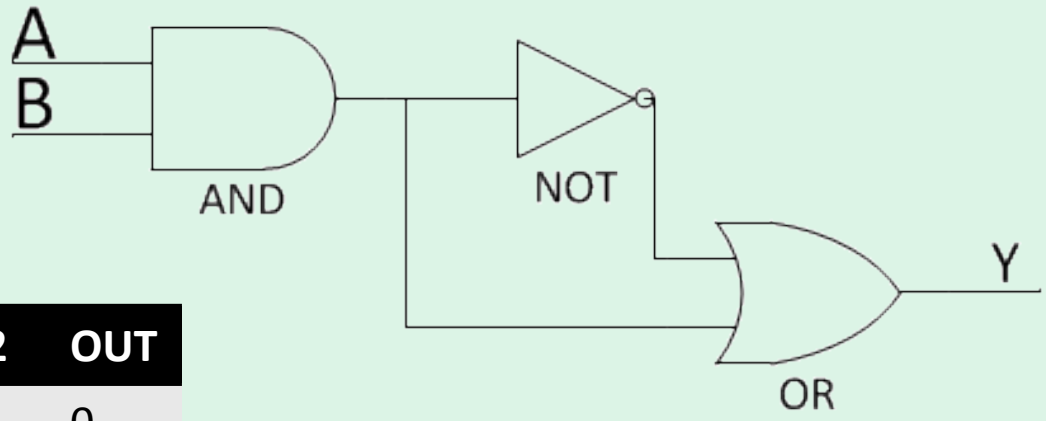


$$Y = \overline{A}$$

A	Y
0	1
1	0

Colin O'Flynn - CC BY-SA

Example!



AND

IN1	IN2	OUT
0	0	0
0	1	0
1	0	0
1	1	1

NOT

IN1	OUT
0	1
1	0

OR

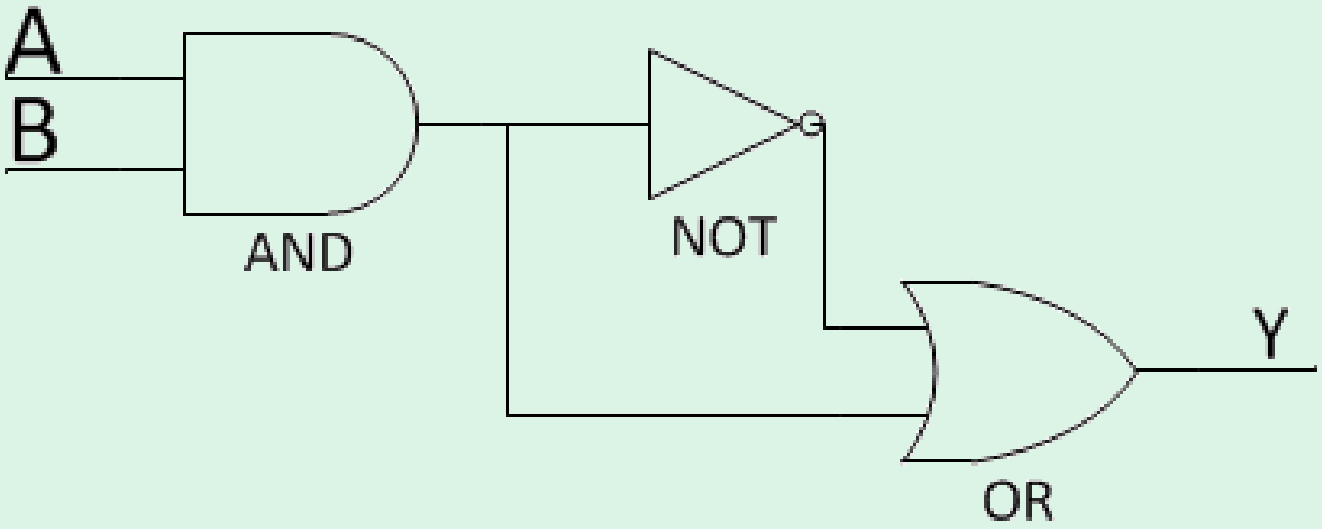
IN1	IN2	OUT
0	0	0
0	1	1
1	0	1
1	1	1

Fill in this table:

A	B	Y
0	0	
0	1	
1	0	
1	1	

Colin O'Flynn – CC BY-SA

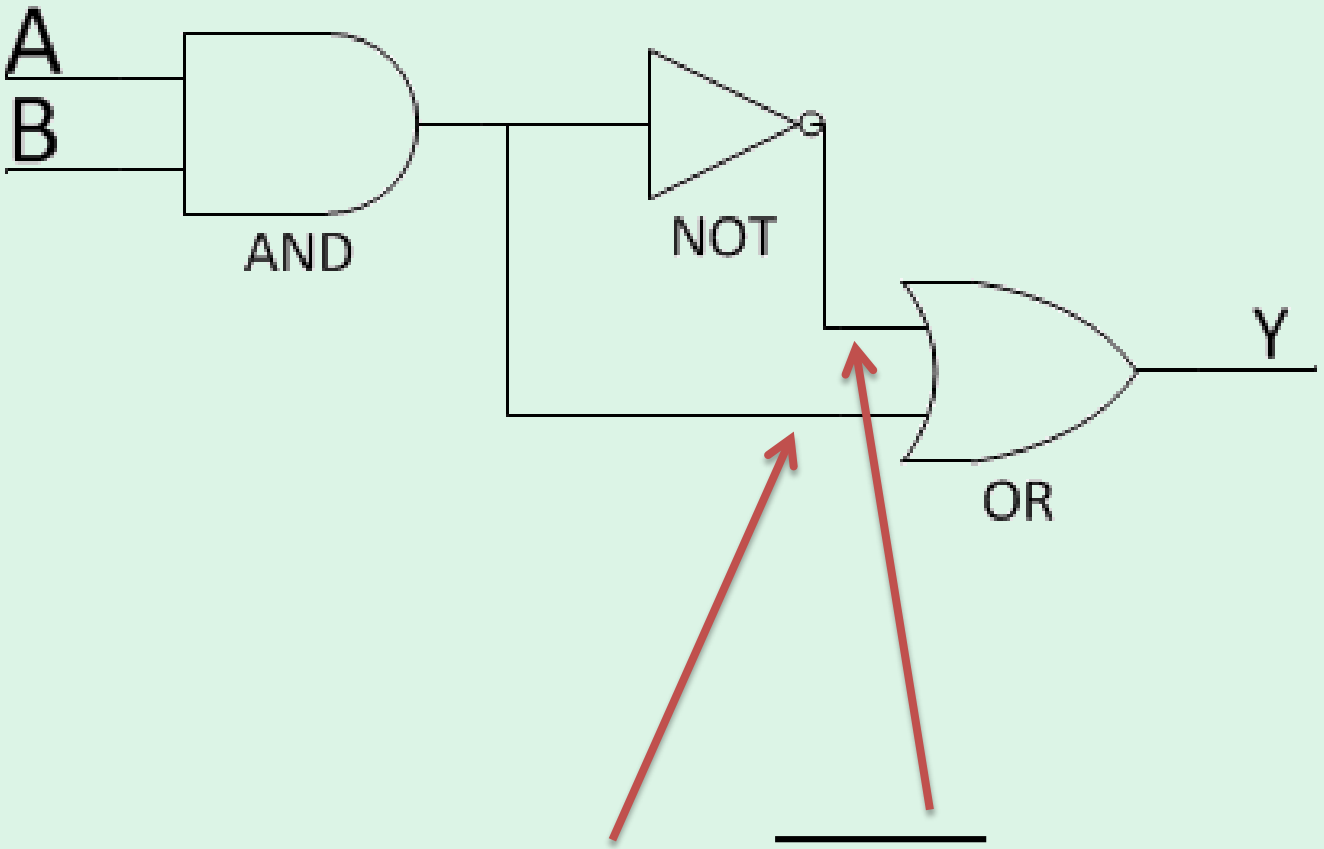
Example!



What is the Boolean function of the above schematic?

Colin O'Flynn - CC BY-SA

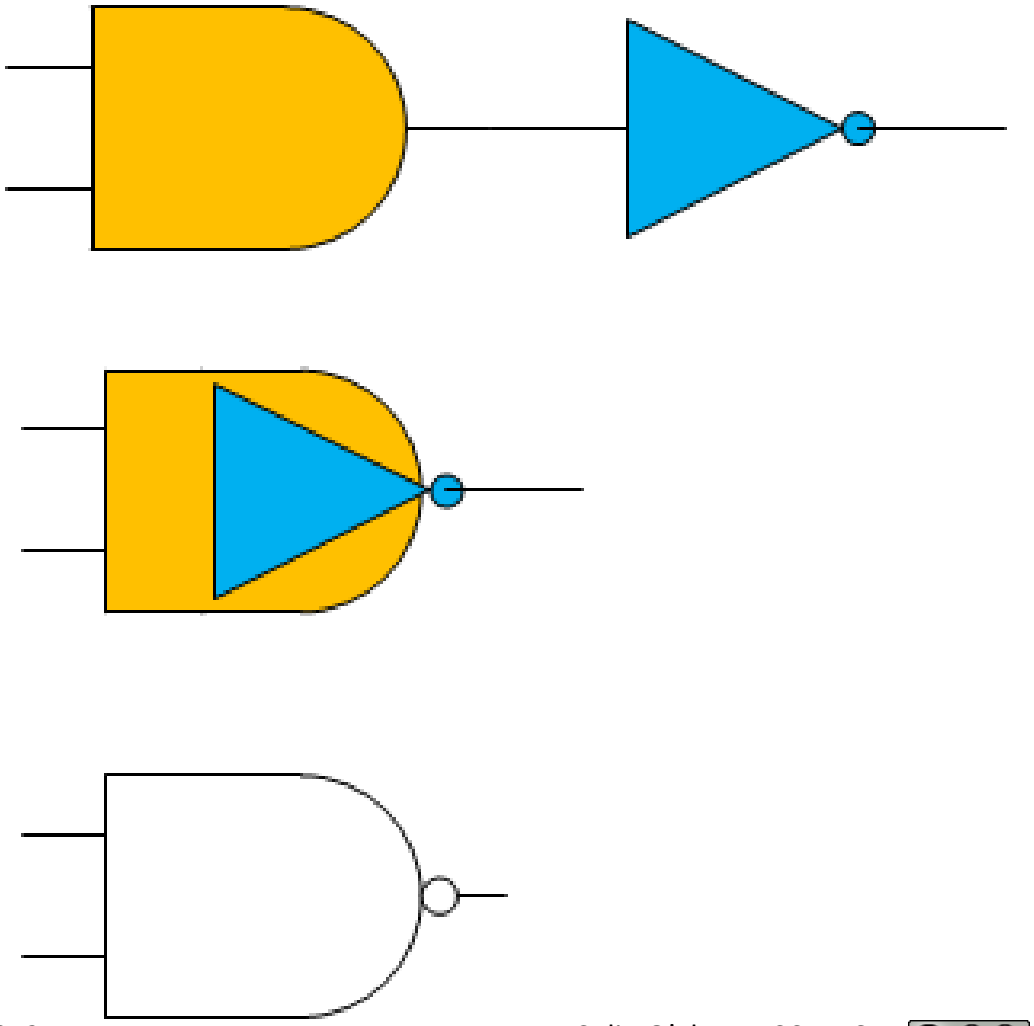
Example!



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

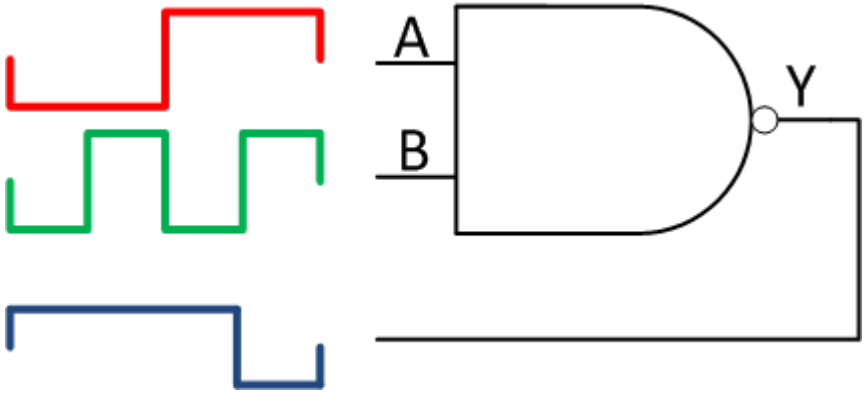
Colin O'Flynn - CC BY-SA

Little Circles



Colin O'Flynn - CC BY-SA

Not AND = NAND Gate

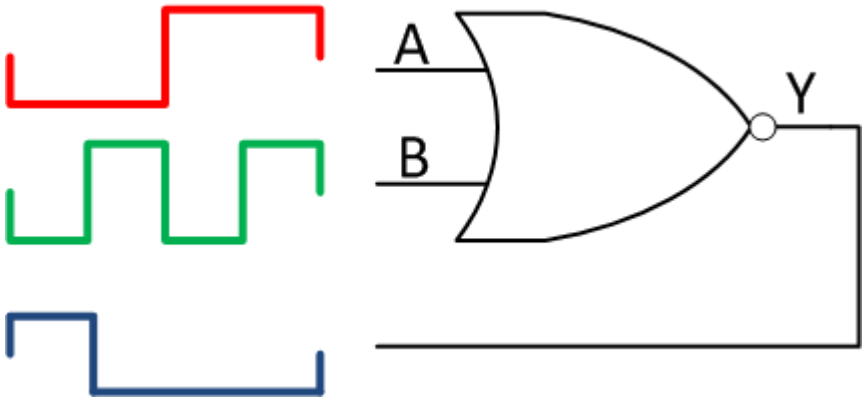


$$Y = \overline{A \bullet B}$$

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

Colin O'Flynn - CC BY-SA

Not OR = NOR Gate

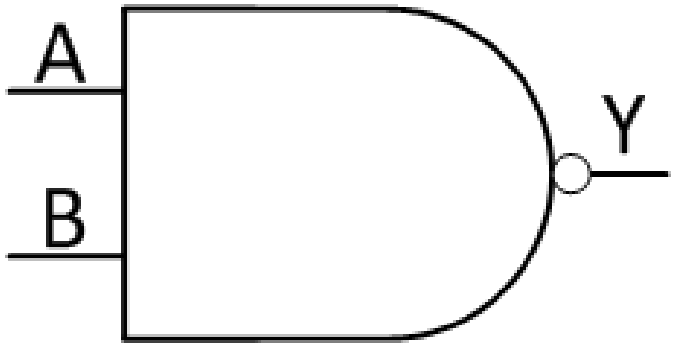


$$Y = \overline{A + B}$$

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

Colin O'Flynn - CC BY-SA

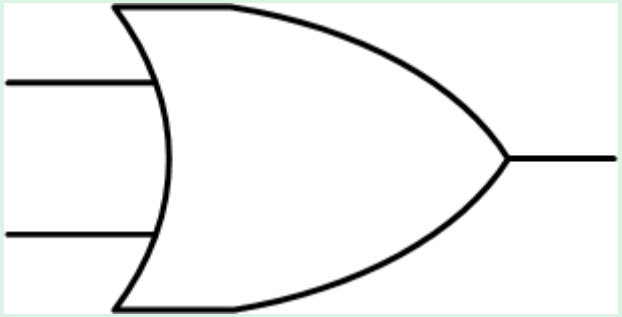
SUPERGATES



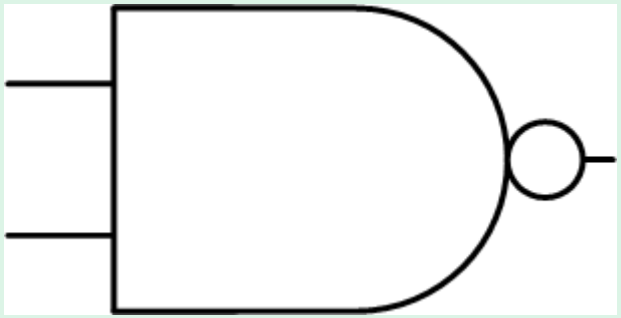
All basic logic operations can be formed with NAND gates (or NOR gates).

Colin O'Flynn - CC BY-SA

Example!



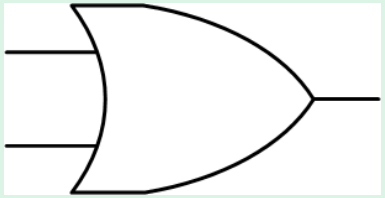
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



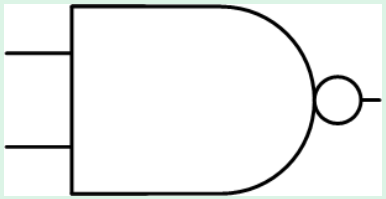
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Colin O'Flynn - CC-BY-SA

Example!



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

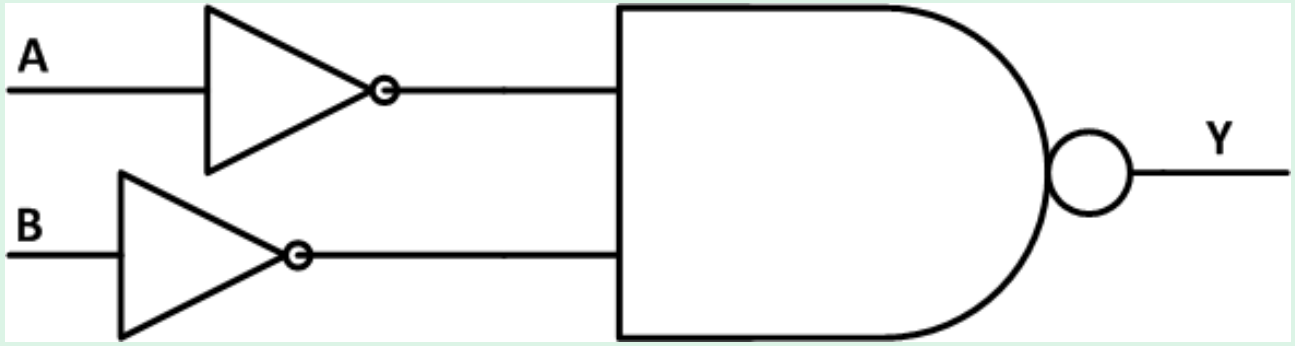


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

If we inverted each of the A & B inputs to the NAND gate, note we get the same truth table as the OR gate!

Colin O'Flynn - CC BY-SA

Example!

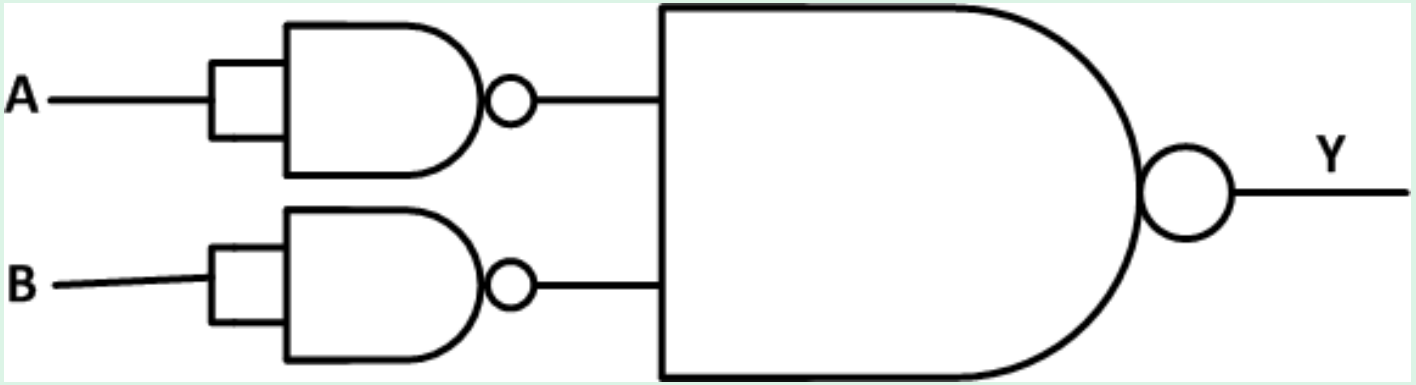


$$Y = \overline{\overline{A} \bullet \overline{B}} = A + B$$

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

Colin O'Flynn - CC BY-SA

Example!

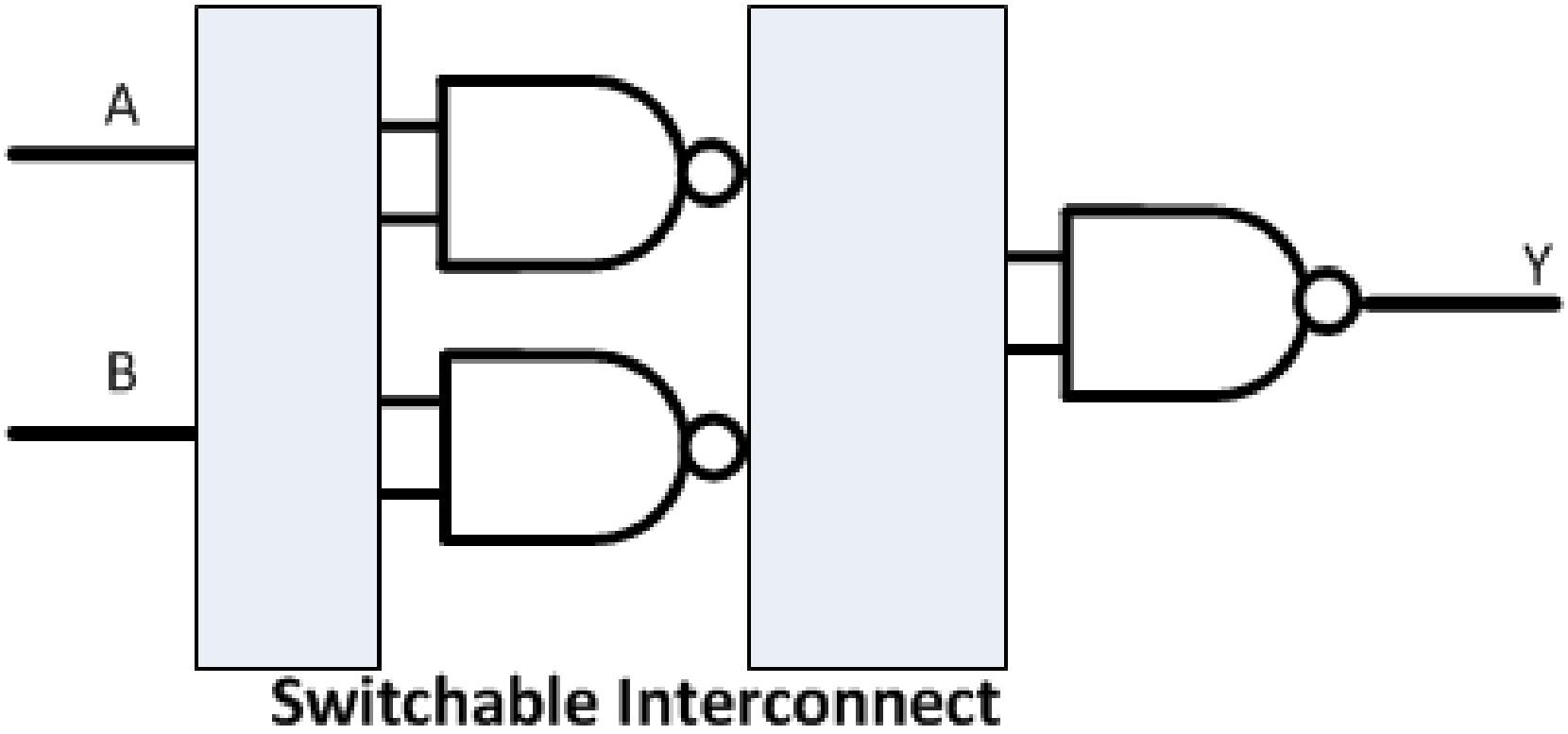


$$Y = \overline{\overline{A} \bullet \overline{B}} = A + B$$

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

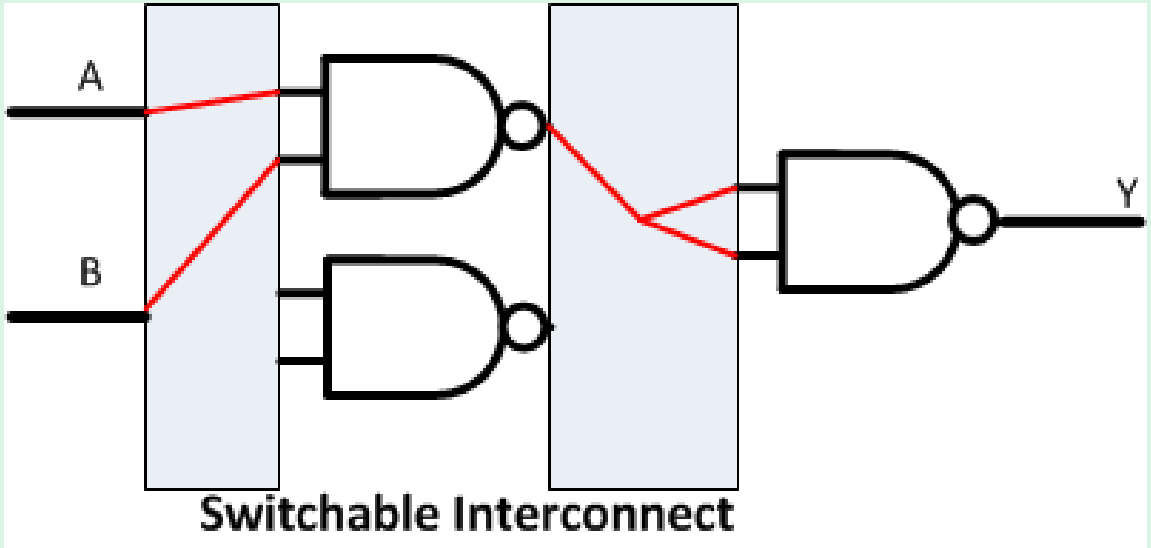
Colin O'Flynn - CC BY-SA

Why Do you Care?



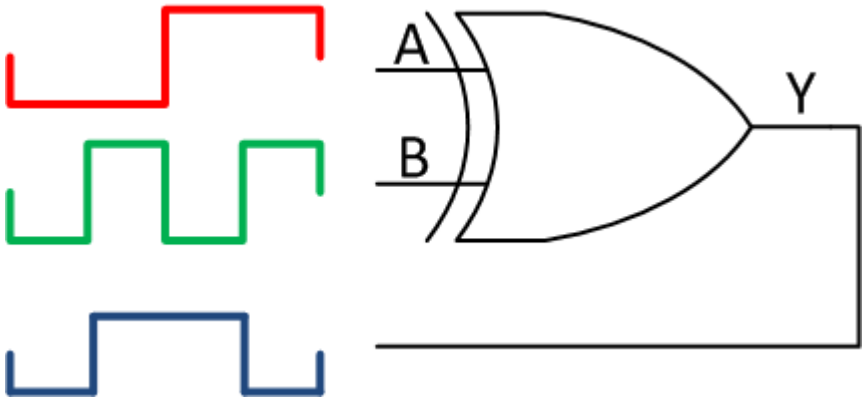
Colin O'Flynn - CC BY-SA

Example – AND Gate



Colin O'Flynn - CC BY-SA

Exclusive OR (X-OR) Gate

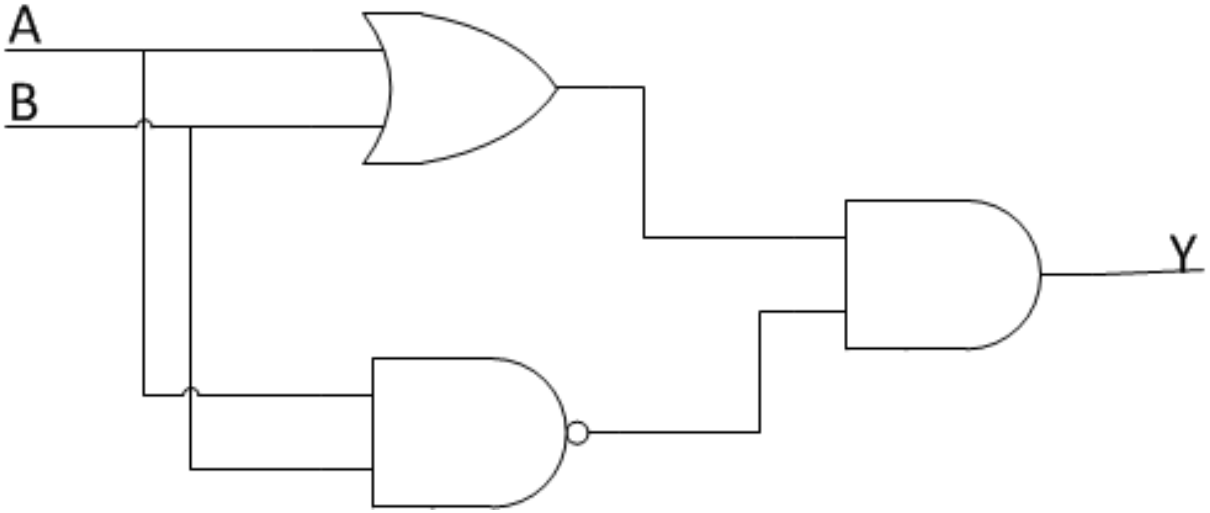


$$Y = A \oplus B$$

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

Colin O'Flynn - CC BY-SA

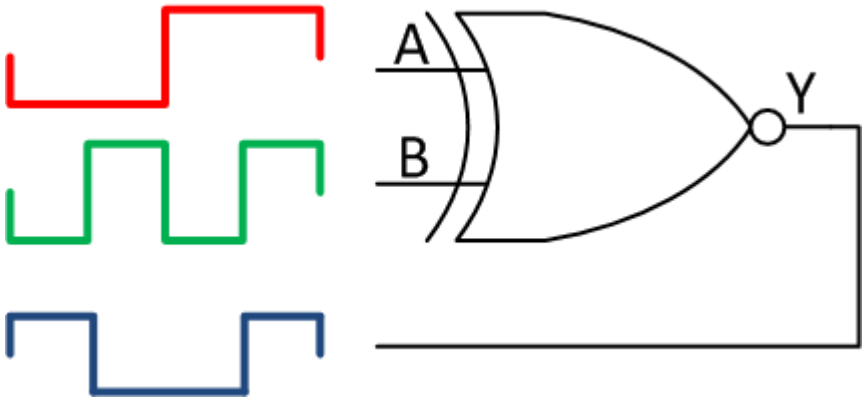
XOR Gate Implementation



$$A \oplus B = (A+B) \bullet (\overline{A \bullet B}) = \overline{A} \bullet B + A \bullet \overline{B}$$

Colin O'Flynn - CC BY-SA

Exclusive NOR (X-NOR) Gate



$$Y = A \odot B$$

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

Colin O'Flynn - CC BY-SA

Section Summary – 1/3

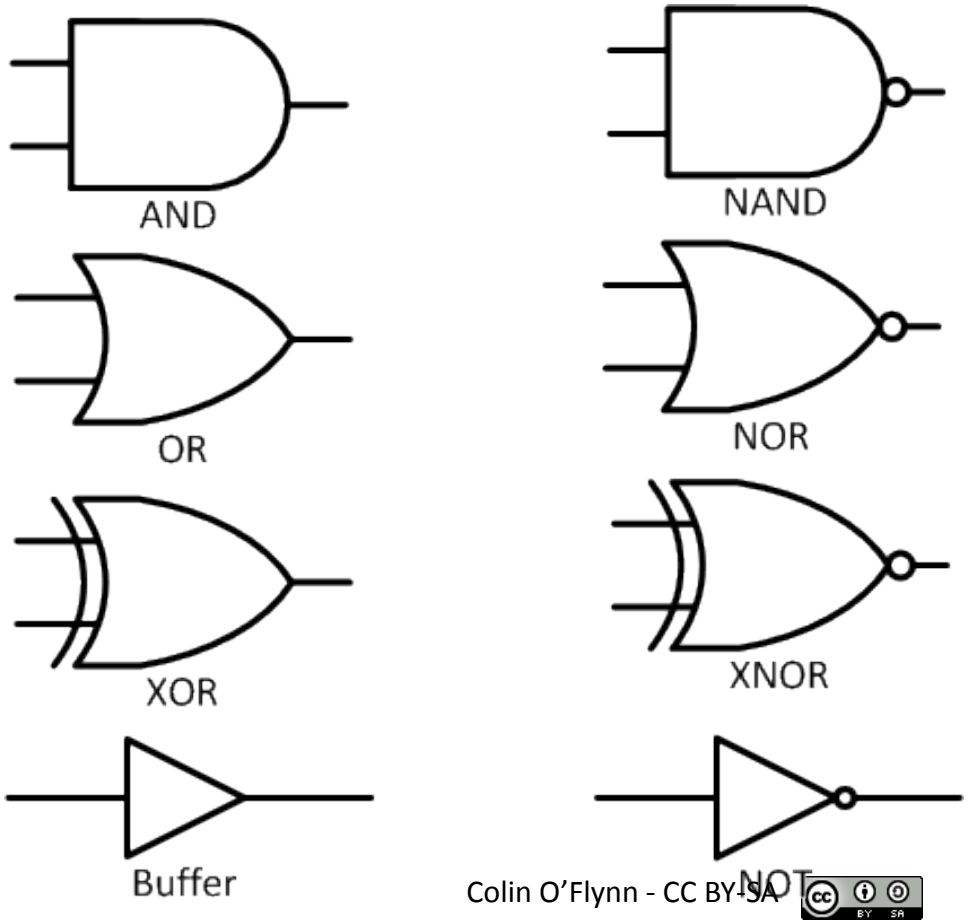
Gates have inputs & outputs. Inputs are binary (Boolean) variables with two possible values ‘1’ (True) or ‘0’ (False).

The ‘Truth Table’ is a table showing every possible input & the resulting output. Different gates have different truth tables.

Colin O’Flynn - CC BY-SA

Section Summary – 2/3

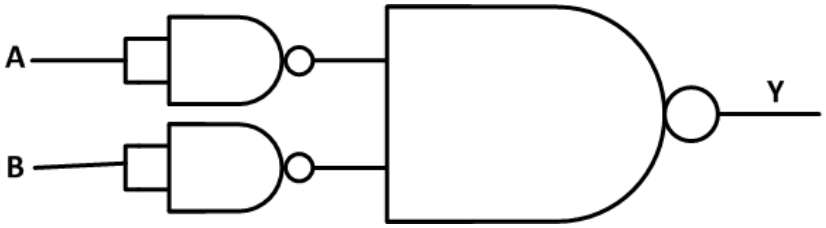
The following are the basic gates:



Colin O'Flynn - CC BY-SA

Section Summary – 3/3

Based on the truth tables, one can make any of the gates with NAND (or NOR) gates. E.g. making an OR gate with NAND gates:



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

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

Colin O'Flynn - CC BY-SA

References / Reading

- ECED2200 Notes, “Digital Circuits” section
- Bebop to the Boolean Boogie, Chapter 5
- CLD, Chapter 1

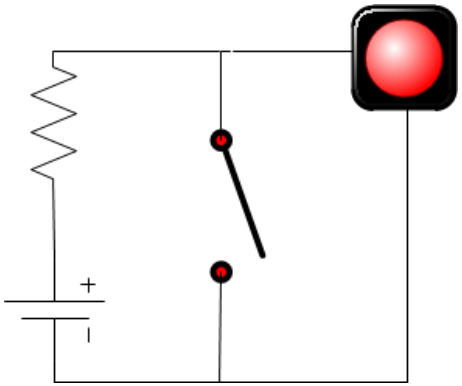
Colin O’Flynn - CC BY-SA

PHYSICAL GATE IMPLEMENTATION

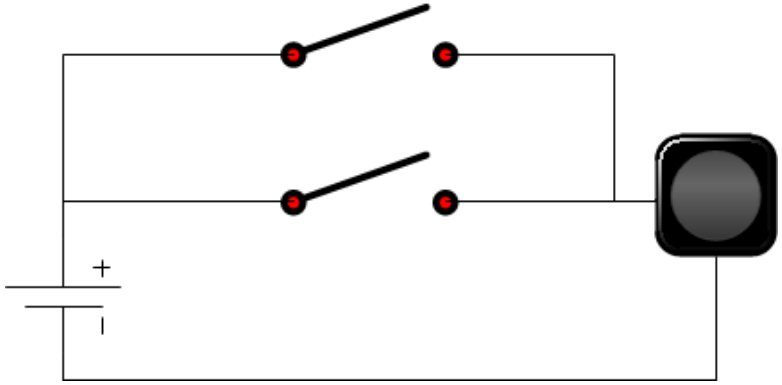
Colin O'Flynn - CC BY-SA

Switch Logic

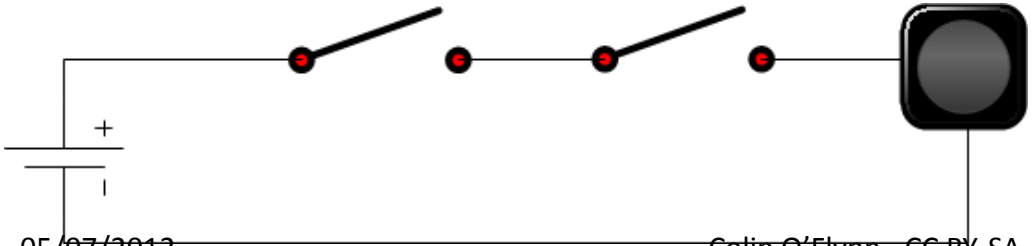
NOT Gate



OR Gate

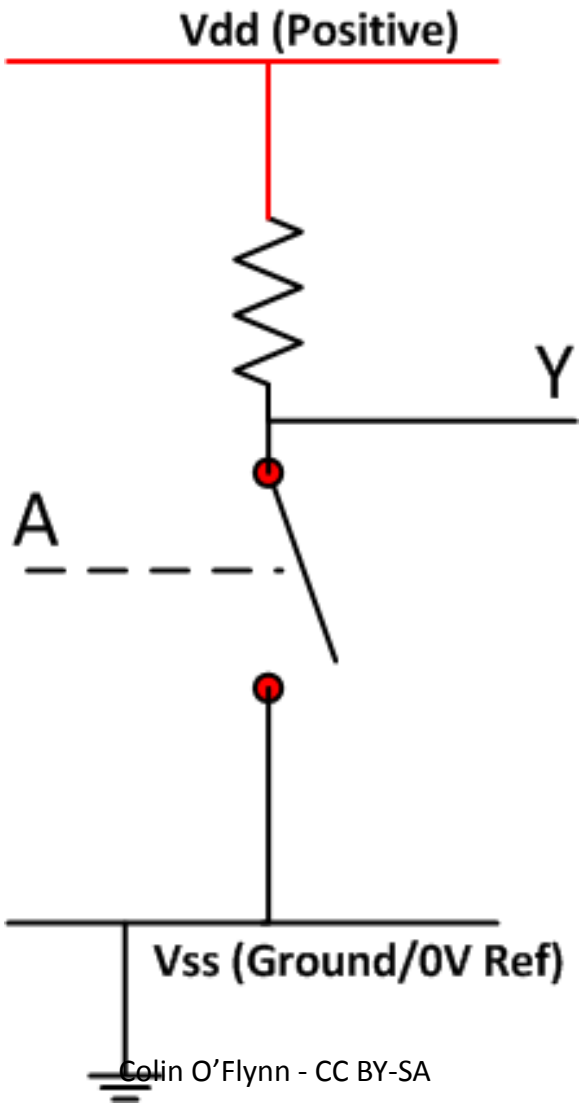


AND Gate



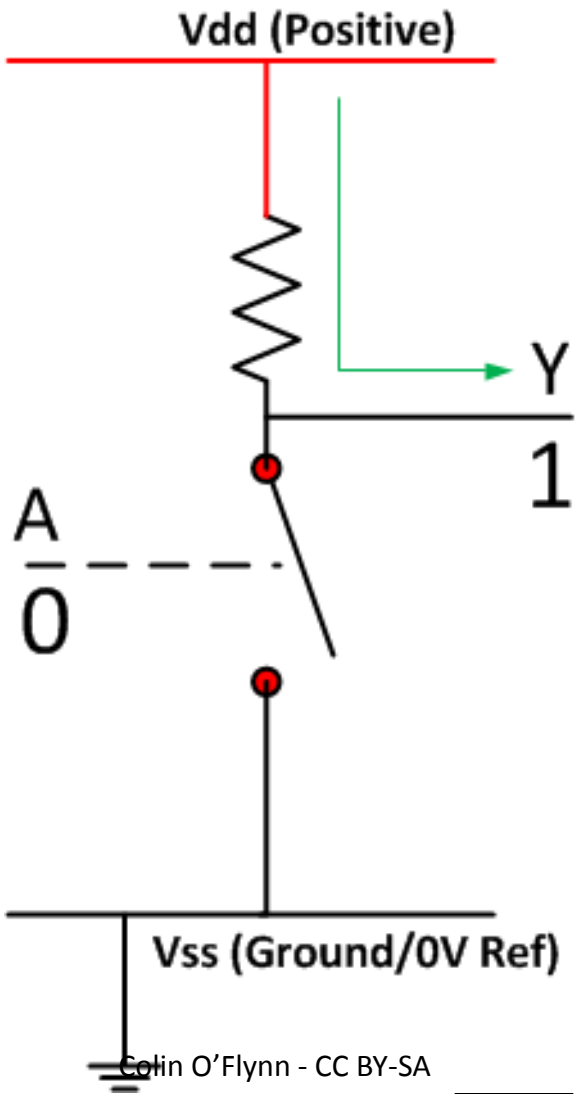
Colin O'Flynn - CC BY-SA

Switch NOT



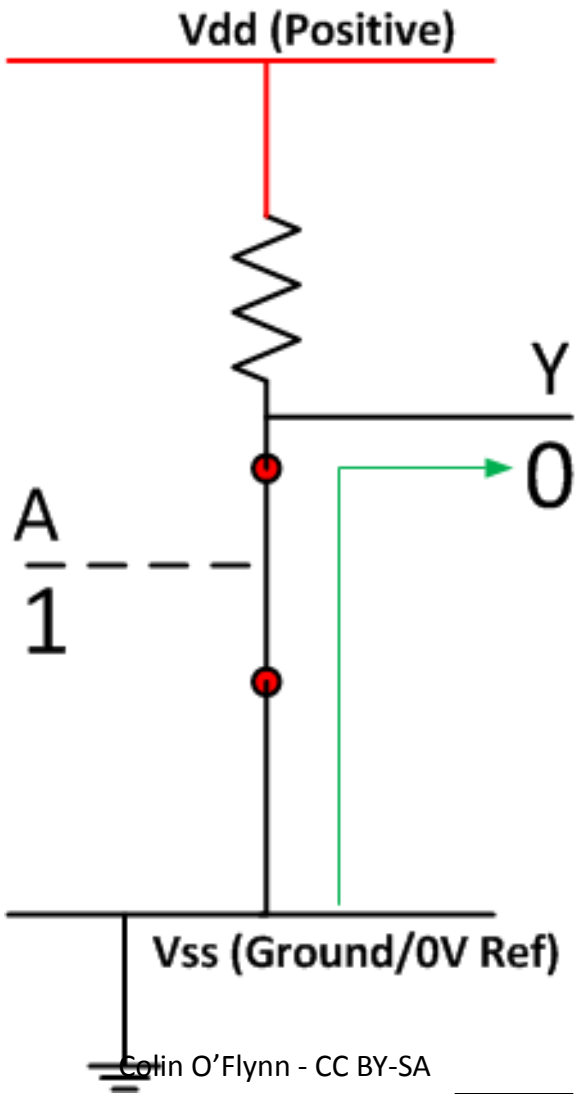
Colin O'Flynn - CC BY-SA

Switch NOT



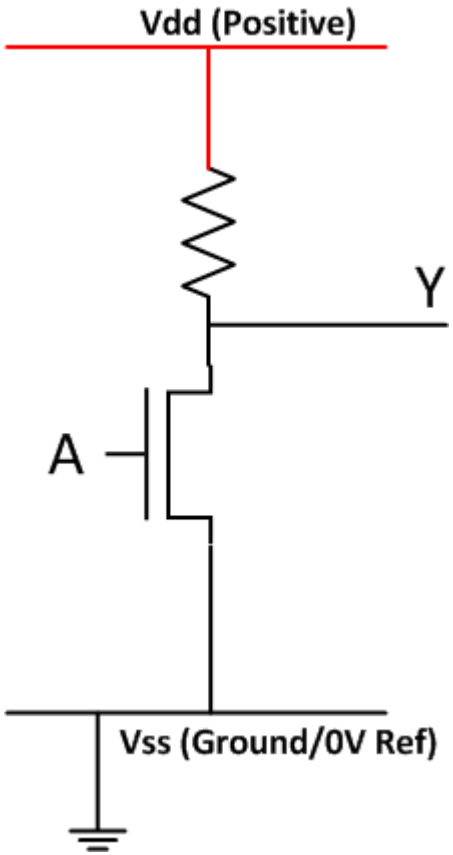
Colin O'Flynn - CC BY-SA

Switch NOT



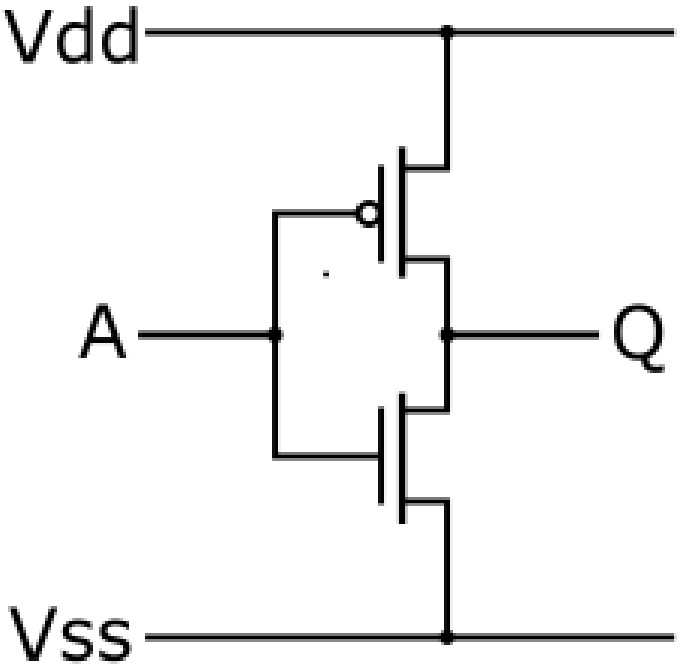
Colin O'Flynn - CC BY-SA

Field Effect Transistor Switch



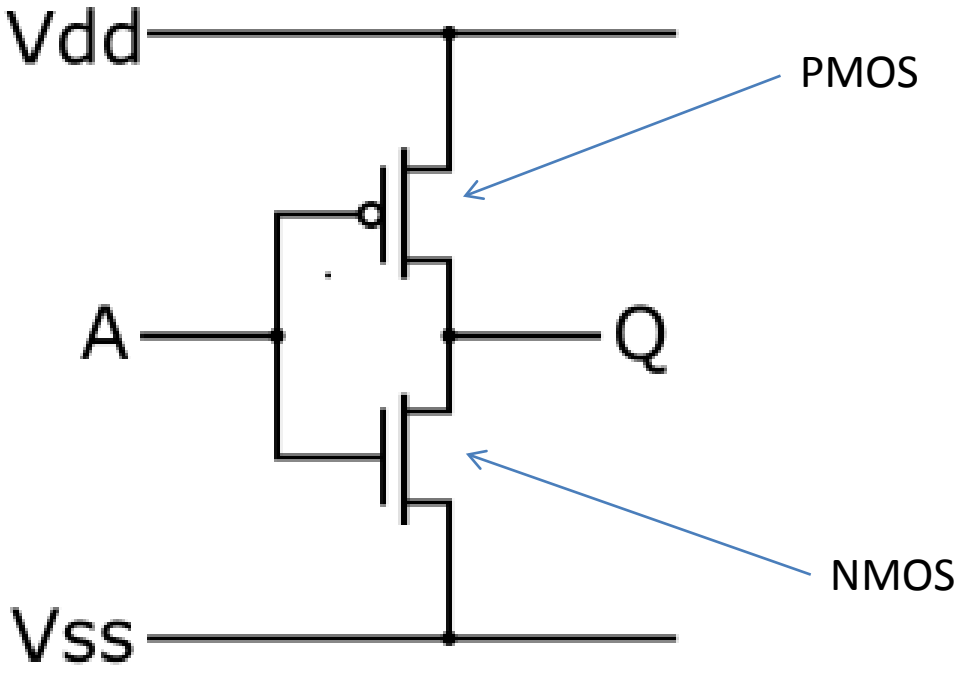
Colin O'Flynn - CC BY-SA

FET Logic Gates – Inverter (NOT)



Colin O'Flynn - CC BY-SA

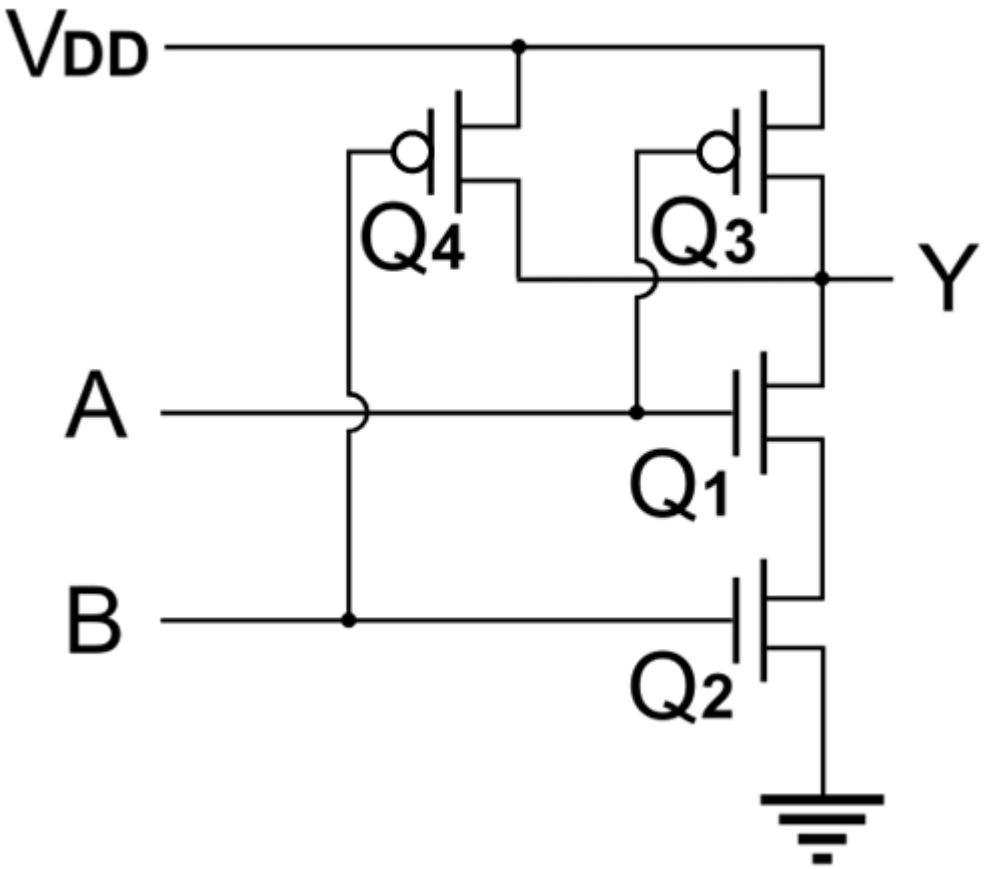
FET Logic Gates – Inverter (NOT)



CMOS = Complementary MOS (e.g.: uses both positive & negative MOS)

Colin O'Flynn - CC BY-SA

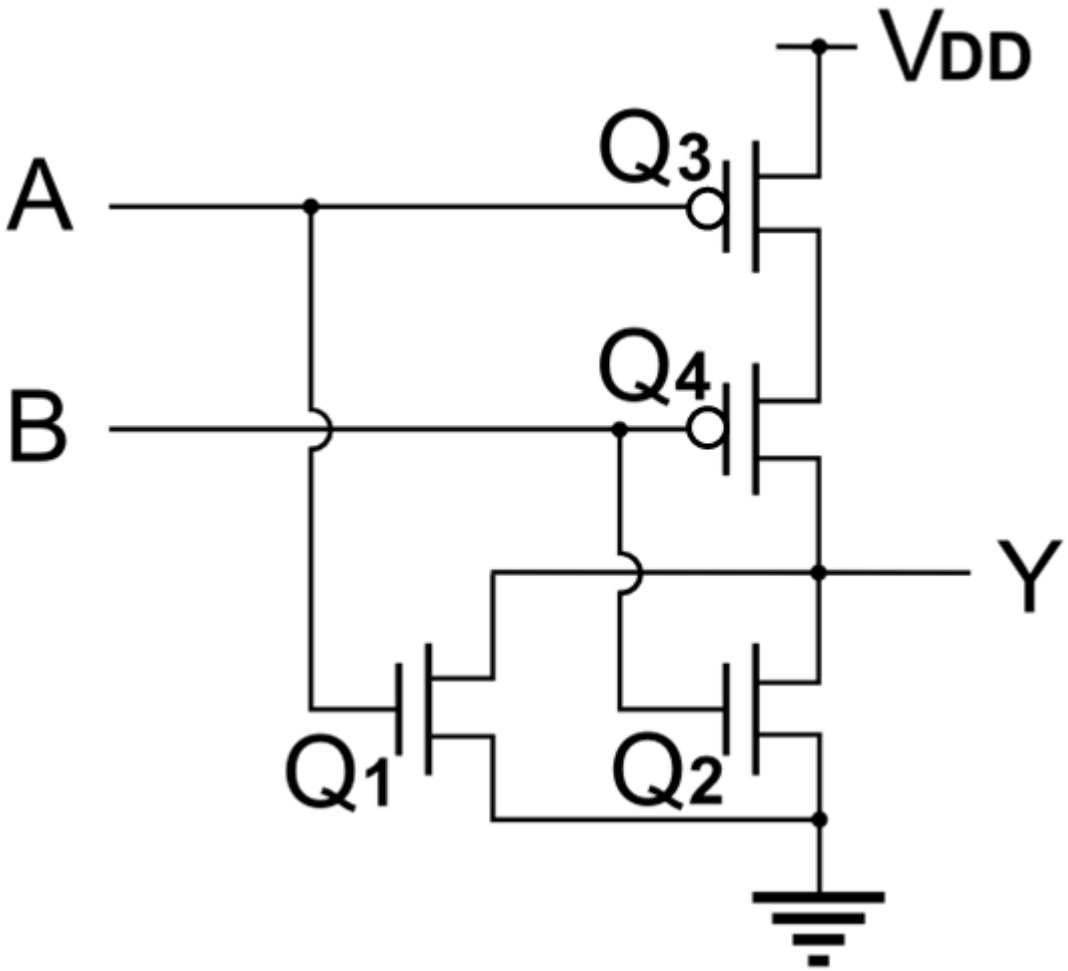
FET Logic Gates - NAND



Source: http://commons.wikimedia.org/wiki/File:NAND_gate_(CMOS_circuit).PNG

Colin O'Flynn - CC BY-SA

FET Logic Gates - NOR



http://commons.wikimedia.org/wiki/File:NOR_gate_%28CMOS_circuit%29.PNG

Colin O'Flynn - CC BY-SA

Logic Families (Types)

1. Diode Logic (DL)
2. Resistor-Transistor Logic (RTL)
3. Diode-Transistor Logic (DTL)
4. Transistor-Transistor Logic (TTL)
5. Metal-Oxide Semiconductor (MOS)
6. **Complementary MOS (CMOS)**
7. Emitter-Coupled Logic (ECL)
8. BiCMOS

Colin O'Flynn - CC BY-SA

Characteristics of Logic Types

- Fan-in
- Fan-out
- Speed
- Noise Margin
- Power
- Size

Colin O'Flynn - CC BY-SA

Section Summary

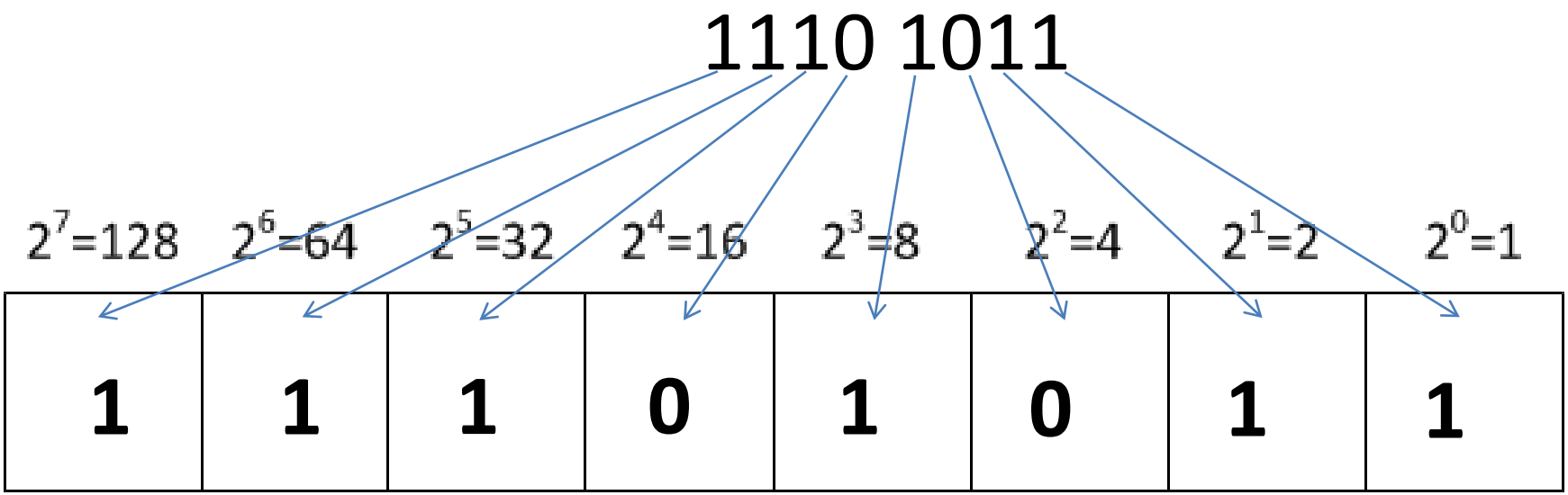
- Bebop to the Boolean Boogie Chapter 6
- ECED2200 Notes “Electric Switches + Logic Classifications”

Colin O’Flynn - CC BY-SA

NUMBER SYSTEMS

Colin O'Flynn - CC BY-SA

Binary to Decimal

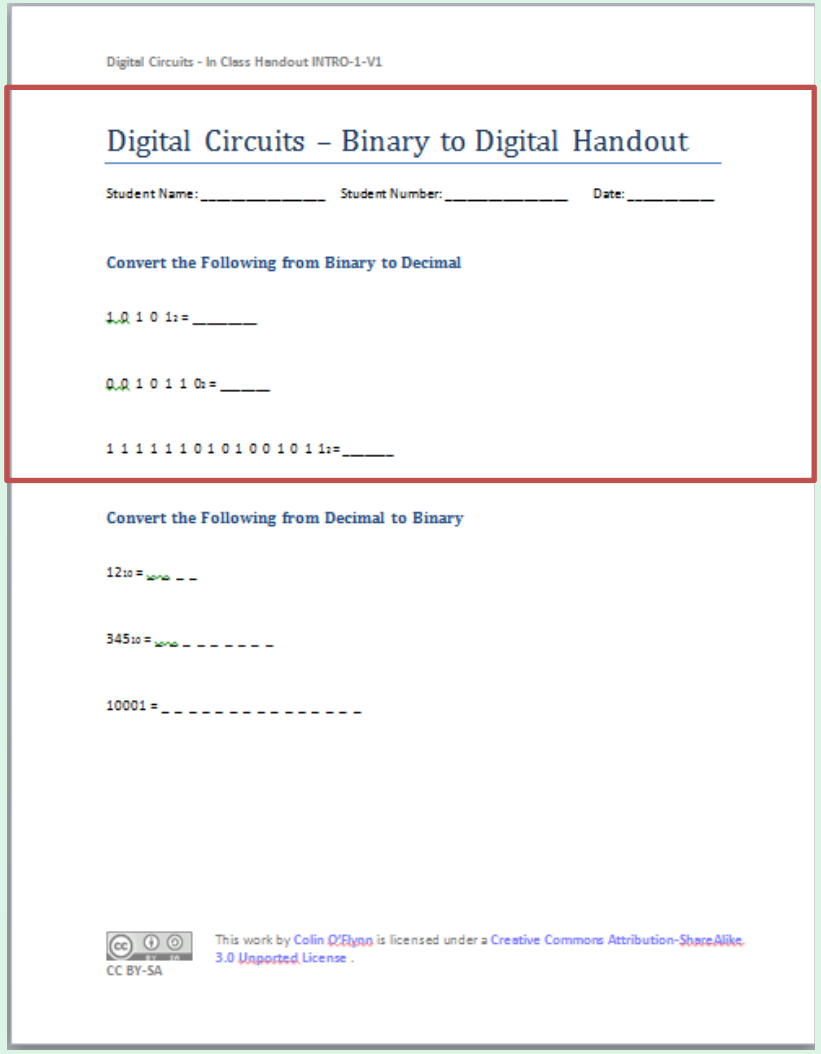


128 + 64 + 32 + + 8 + + 2 + 1

= 235 Decimal

Colin O'Flynn - CC BY-SA

Let's Do a Hand-out: INTRO-1 (Top)



Digital Circuits - In Class Handout INTRO-1-V1

Digital Circuits – Binary to Digital Handout

Student Name: _____ Student Number: _____ Date: _____

Convert the Following from Binary to Decimal

$1.0\ 1\ 0\ 1_2$ = _____

$0.0\ 1\ 0\ 1\ 1\ 0_2$ = _____

$1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1_2$= _____

Convert the Following from Decimal to Binary

12_{10} = _ _ _ _

345_{10} = _ _ _ _ _ _ _ _

10001 = _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

This work by Colin O'Flynn is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License .

CC BY-SA

Decimal to Binary

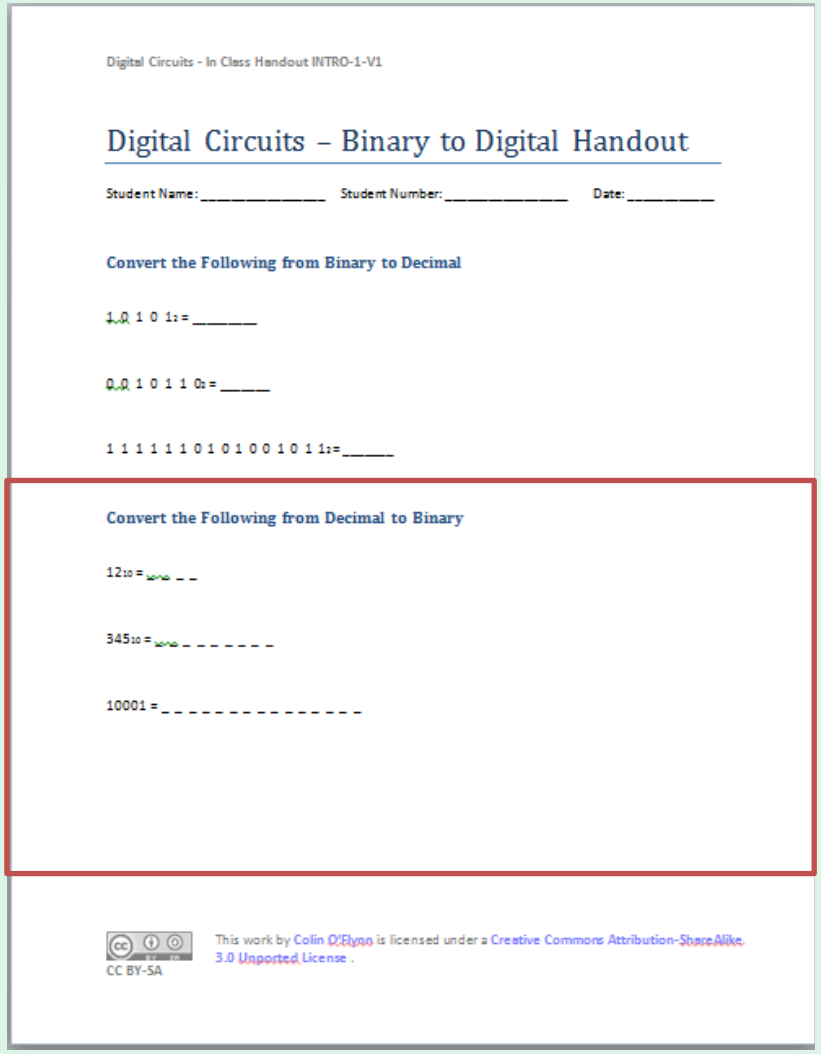
216 Decimal

216 – 128 = 88 (1 in 2^7 position)
88 – 64 = 24 (1 in 2^6 position)
24 – 32 < 0, so 0 in 2^5 position
24 – 16 = 8 (1 in 2^4 position)
8 – 8 = 0 (1 in 2^3 position)
0 in remaining positions

2^7=128	2^6=64	2^5=32	2^4=16	2^3=8	2^2=4	2^1=2	2^0=1
1	1	0	1	1	0	0	0

Colin O'Flynn - CC BY-SA

Let's Do a Hand-out: INTRO-1 (Bottom)



Digital Circuits - In Class Handout INTRO-1-V1

Digital Circuits – Binary to Digital Handout

Student Name: _____ Student Number: _____ Date: _____

Convert the Following from Binary to Decimal

10101_2 = _____

00101101_2 = _____

11111101010010110_2 = _____

Convert the Following from Decimal to Binary

12_{10} = _ _ _ _

345_{10} = _ _ _ _ _ _ _ _

10001 = _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

CC BY-SA

This work by Colin O'Flynn is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License .

Number Notation

There are 10 kinds of people in the word – those that understand binary, and those that don't.

Colin O'Flynn - CC BY-SA

Number Notation

Ambiguity unacceptable - we are engineers not comics.

$10_2 = 10_B = 2_{10}$ (2 decimal)

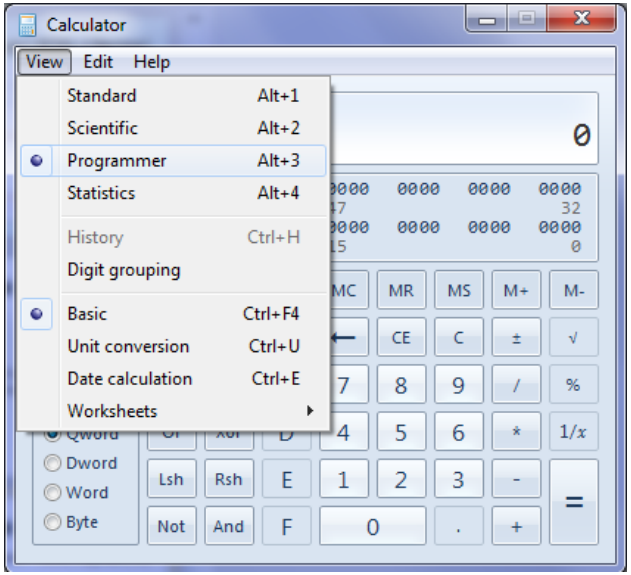
$10_{10} = 10_D = 1010_2$ (1010 binary)

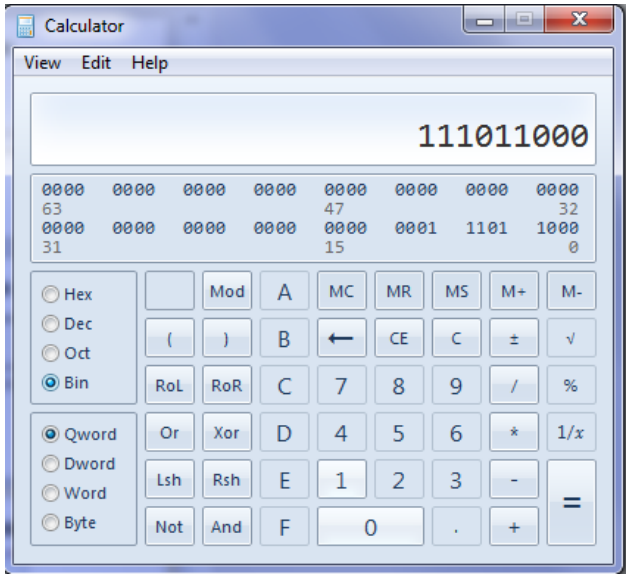
e.g.: Conversion:

$472_{10} = 111011000$

Colin O'Flynn - CC BY-SA

How to Check your Conversions





Windows Calculator (Windows 7 version shown)

How to Check your Conversions

Stand Alone Calculator

Colin O'Flynn - CC BY-SA

Other Number Systems: Hex & Octal

Colin O'Flynn - CC BY-SA

Cheat Sheet

Decimal	Binary	Hexadecimal (0x)	Octal (0)
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

Colin O'Flynn - CC BY-SA

Advantages of Hex (and Octal)

$1010\ 1000\ 1101\ 1111\ 1011\ 0010_2$

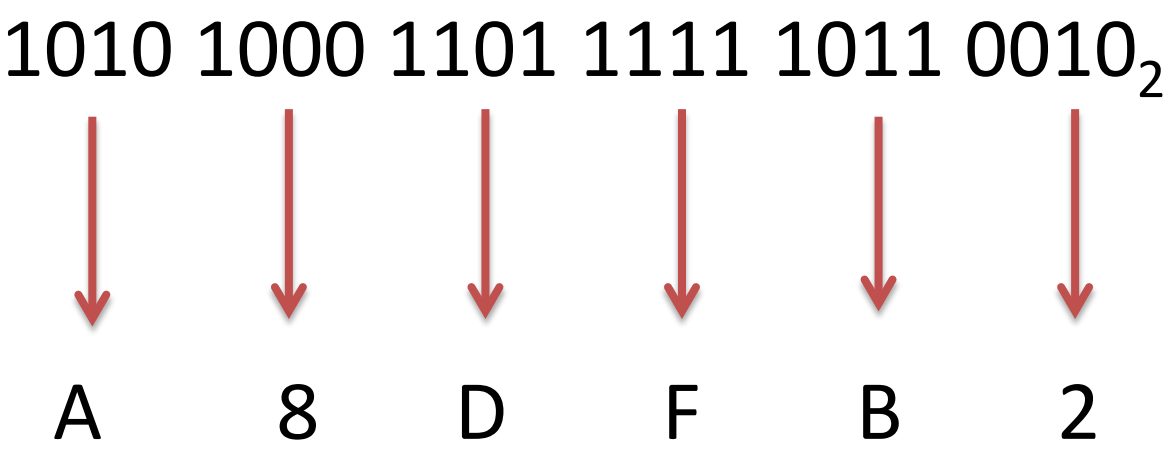
This is only 24 bits – but long/hard to write…

Equivalent to 11067314_{10}

Easier to write, but conversion is error-prone, plus we are normally lazy…

Colin O'Flynn - CC BY-SA

Advantages of Hex

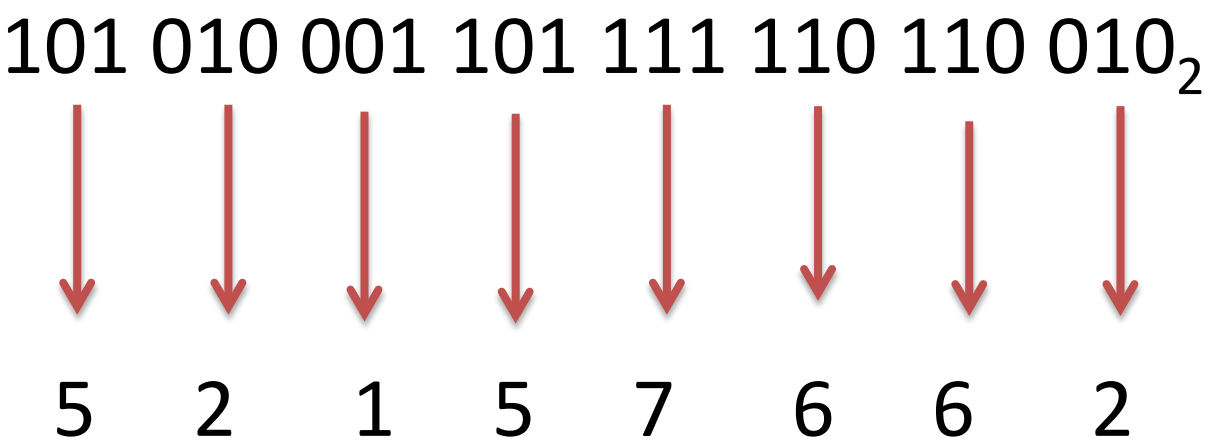


= 0xA8DFB2

Much easier to write, easy to convert!

Colin O'Flynn - CC BY-SA

Advantages of Octal



= 052157662 in Octal

Easy to write (longer than hex though), still easy to convert!

Colin O'Flynn - CC BY-SA

Binary Coded Decimal

Colin O'Flynn - CC BY-SA

Section Summary

- Bebop to the Boolean Boogie: Chapter 7
- ECED2200 Notes: “Number Systems”

Colin O’Flynn - CC BY-SA

ADDING & SUBTRACTING IN BINARY

Colin O'Flynn - CC BY-SA

Adding in Binary

Colin O'Flynn - CC BY-SA

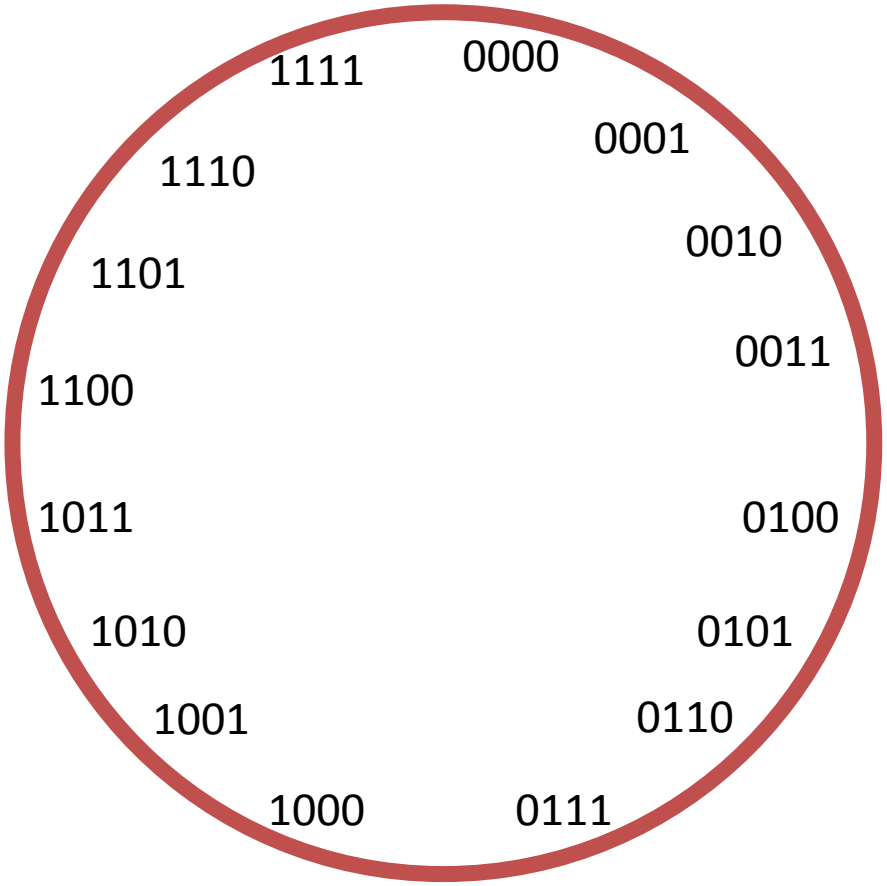
Subtracting in Binary

Colin O'Flynn - CC BY-SA

Positive & Negative Numbers

Colin O'Flynn - CC BY-SA

Number Wheel



Colin O'Flynn - CC BY-SA

Disadvantage of Sign Magnitude

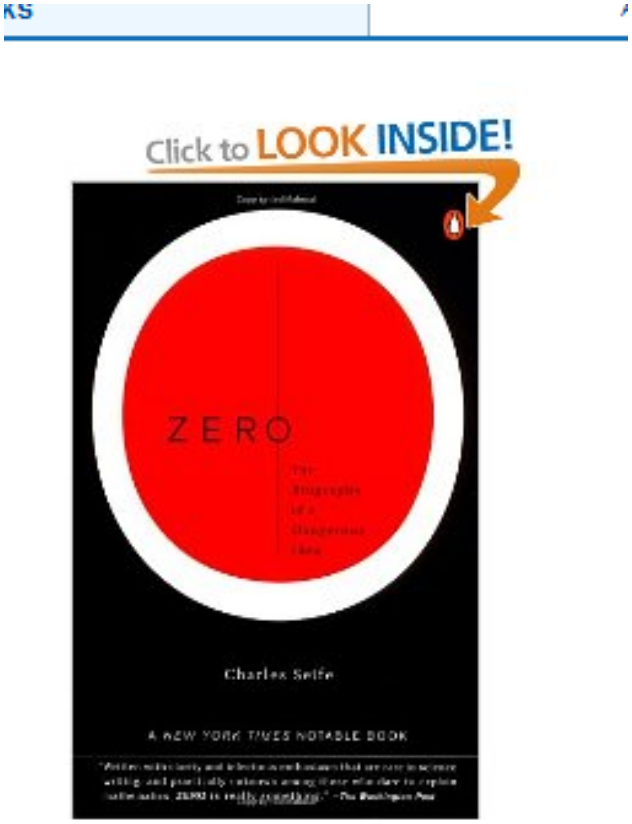
- Two Zeros
- Positive & Negative numbers require different processing or else addition is wrong:

  (-5) + (+1) = 1101 + 0001 = 1110 = -6

  (-3) + (-2) = 1011 + 1010 = 0101 = +5

Colin O'Flynn - CC BY-SA

Two Zeros? Where did Zero even come from?



Colin O'Flynn - CC BY-SA

Rules of Brahmagupta (628 AD):

- The sum of zero and a negative number is negative.
- The sum of zero and a positive number is positive.
- The sum of zero and zero is zero.
- The sum of a positive and a negative is their difference; or, if their absolute values are equal, zero.

Colin O'Flynn - CC BY-SA

Complementary Numbers

Colin O'Flynn - CC BY-SA

Diminished Radix Complement

Colin O'Flynn - CC BY-SA

Radix Complement

Colin O'Flynn - CC BY-SA

2's Complement

1 1 0 1 0 0 0 1 0 0

Colin O'Flynn - CC BY-SA

Instructions

1. Form 1's complement by inverting all bits

2. Add 1 to 1's complement to get 2's complement

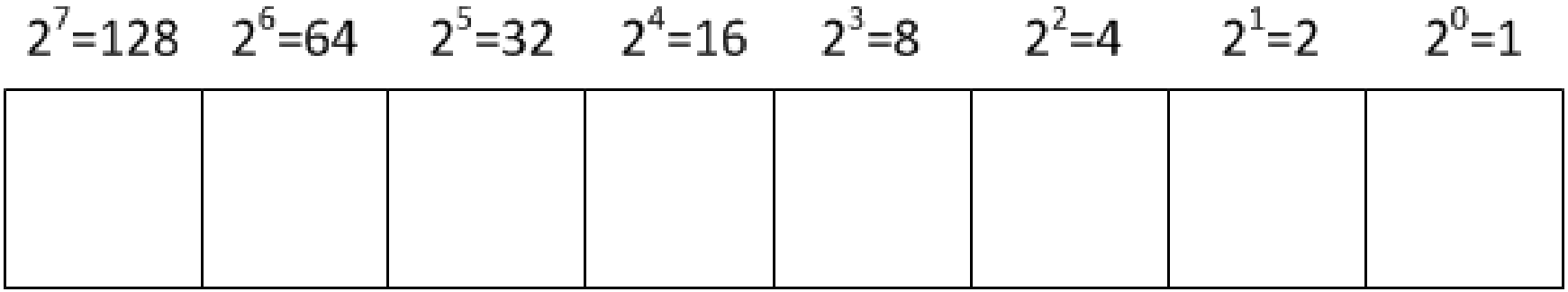
Colin O'Flynn - CC BY-SA

2's Complement

1 1 0 1 0 0 0 1 0 0

Colin O'Flynn - CC BY-SA

Bit Notations



Colin O'Flynn - CC BY-SA

Instructions:

1. Starting from Least Significant Bit (LSB) working towards Most Significant Bit (MSB), copy number one bit at a time.

2. Continue copying until you reach the first ‘1’. Copy this ‘1’ bit as-is

3. For remaining bits invert them (0->1, 1->0)

Colin O’Flynn - CC BY-SA

Some Examples

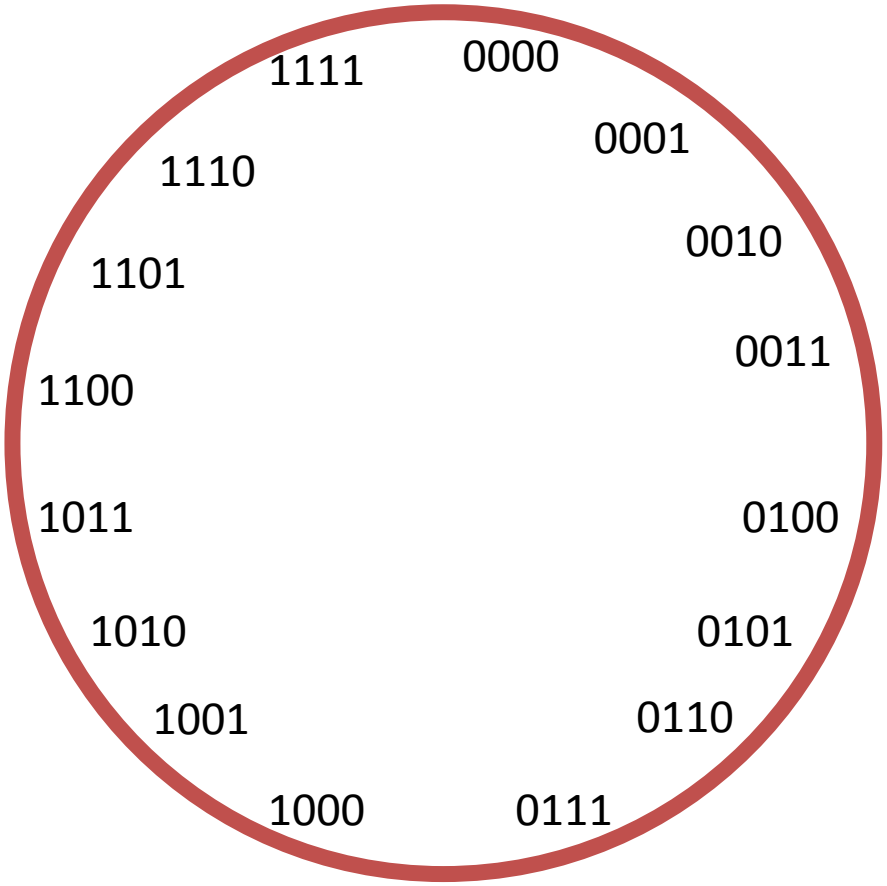
1 0 1 0 0 1 0 1 0 1 1 0

0 0 1 1 1 1 1 1 0 0 0 1

Colin O'Flynn - CC BY-SA

Colin O’Flynn - CC BY-SA

Number Wheel



Colin O'Flynn - CC BY-SA

Section Summary

- Bebop to the Boolean Boogie: Chapter 8
- Contemporary Logic Design:
- ECED Notes: “Number Systems”

Colin O’Flynn - CC BY-SA