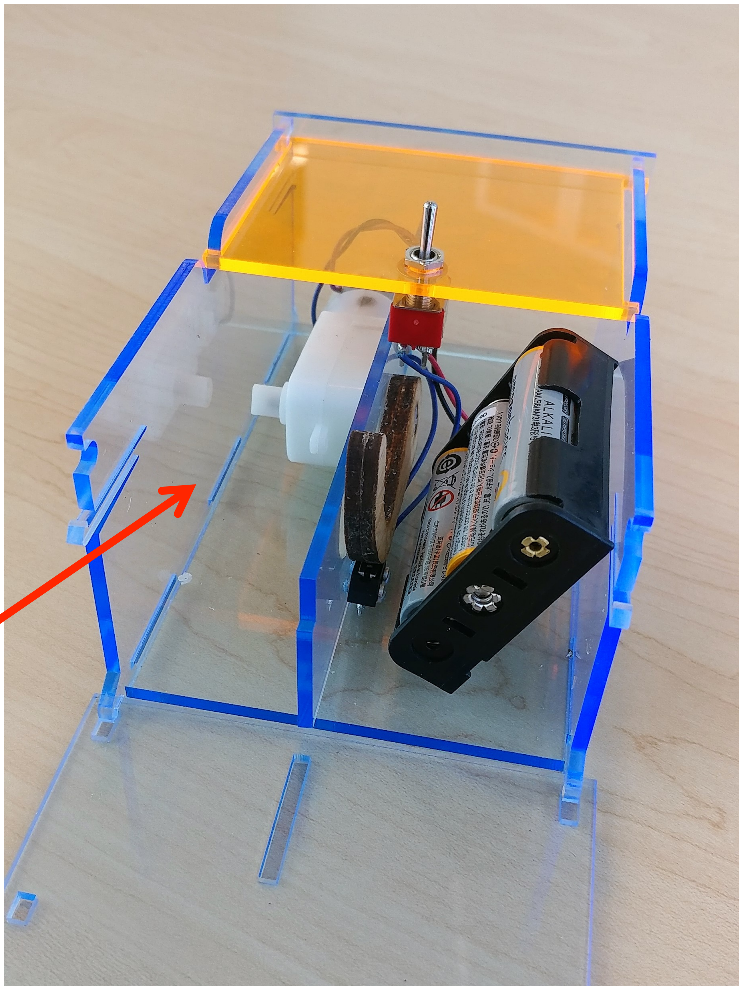
E40M
Useless Box, Boolean Logic

Useless Box Lab Project #2a

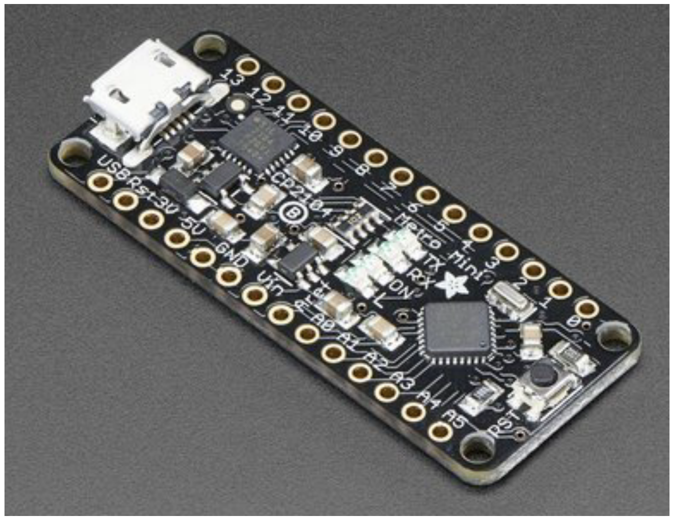
- Motor
- Battery pack
- Two switches
  - The one you switch
  - A limit switch

The first version of the box you will build (Lab 2a) uses mechanical switches to determine the “state” of the box e.g. https://www.youtube.com/watch?v=aqAUmgE3WyM

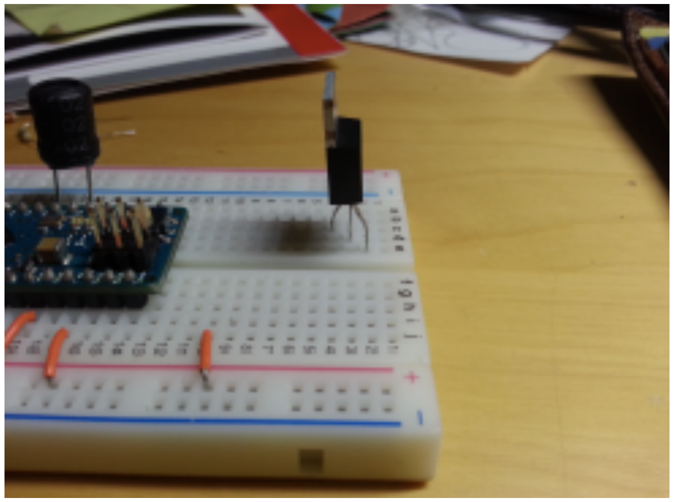


Useless Box Lab Project #2b

- Concepts
  - Finite State Machines
  - Digital Logic
  - Binary numbers
  - MOS Transistors
  - CMOS Gates

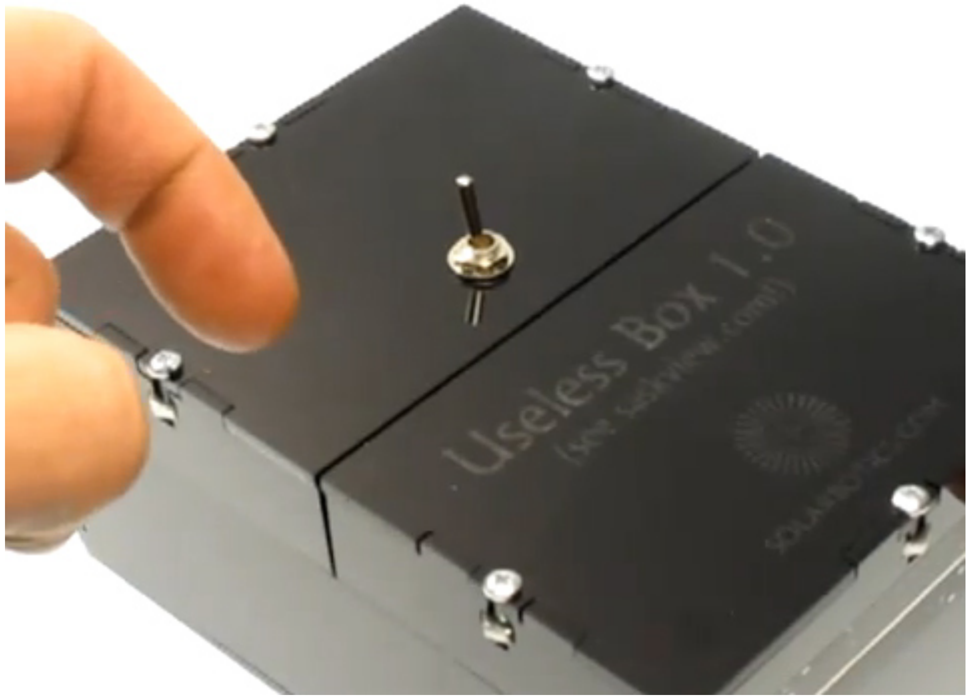


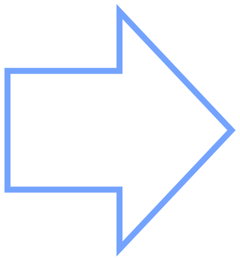
Adding a computer (Arduino) (Lab 2b) makes the box much more interesting e.g. https://www.youtube.com/watch?v=-PqcCjFaf3I



Useless Box Lab Project #2

The concepts we'll discuss will help you to understand how modern digital systems work.





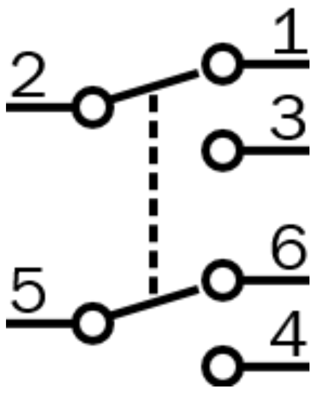
Readings For This Material

- Chapter 4 in the reader up to MOS transistors
- For more details
  - A&L 5.1 Digital Signals

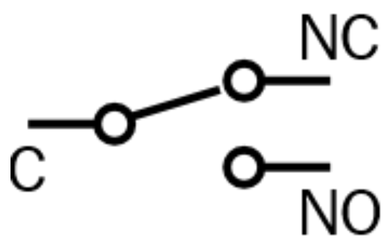
    (goes in much more detail than we need)

Useless Box Operation

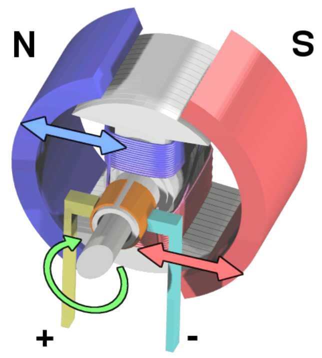
- The simple version of the Useless Box uses switches, batteries and a motor.



DPDT Switch



Momentary Switch

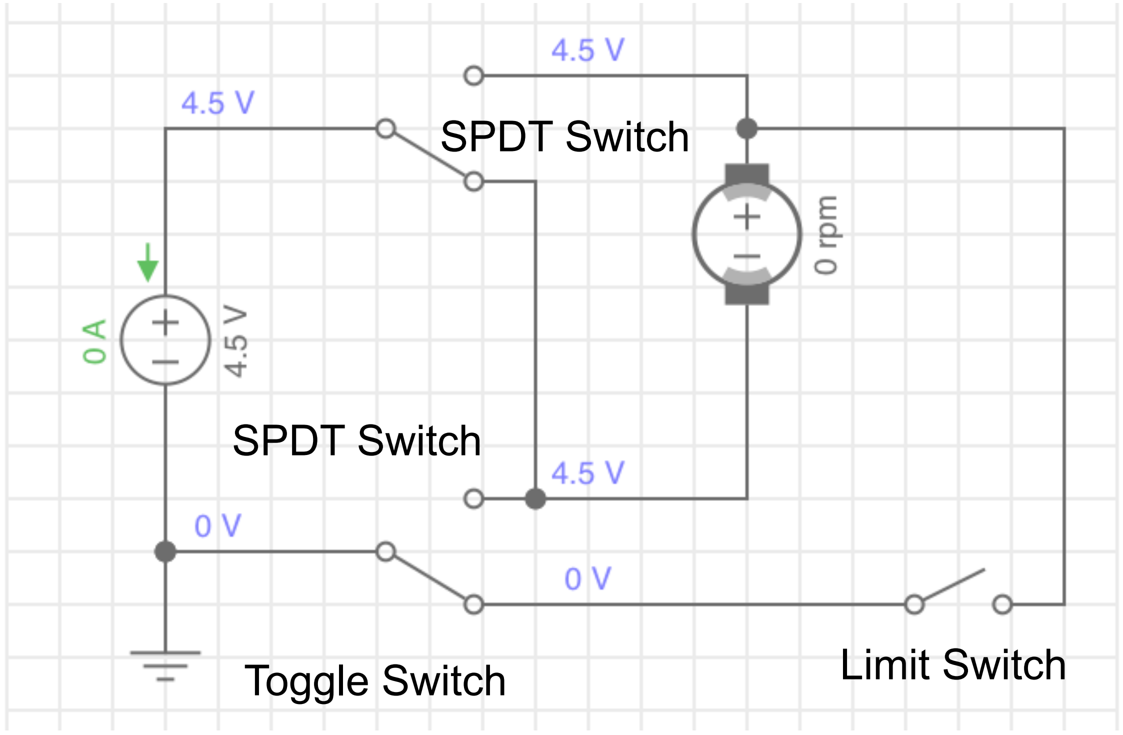


- As discussed in last Friday's lecture, in order to figure out how to wire these components together, we can use an "action diagram" to illustrate what we want the box to do.

- Last Friday's Prelab lecture also introduced how we actually wire the components in a circuit.

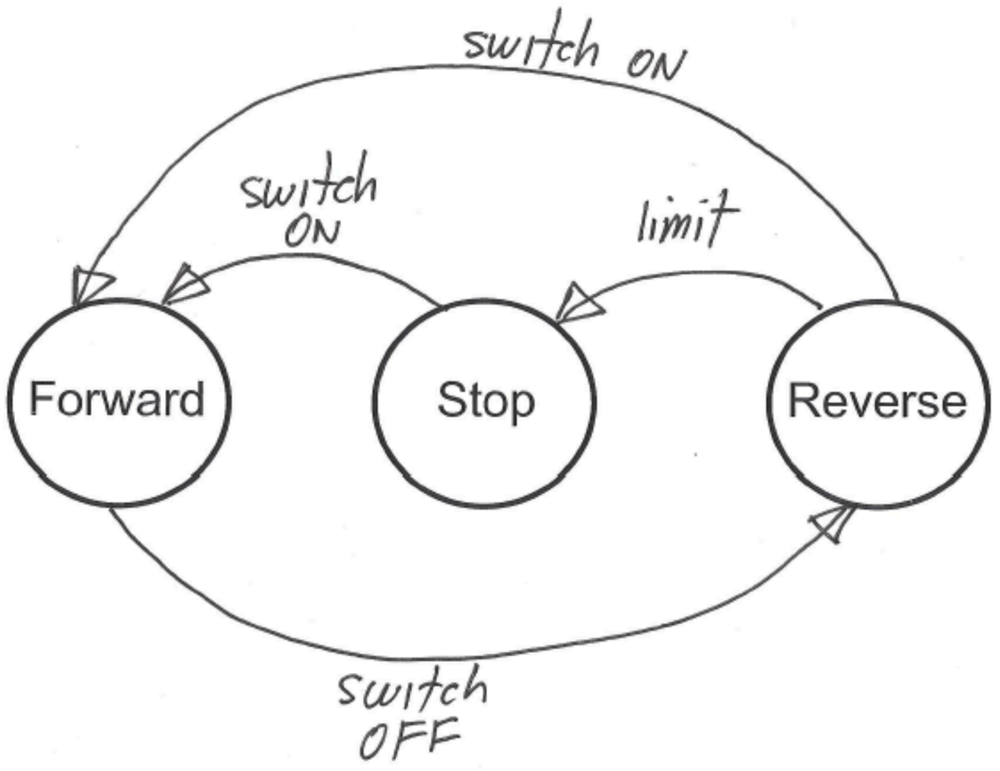
EveryCircuit Implementation of Useless Box 2a

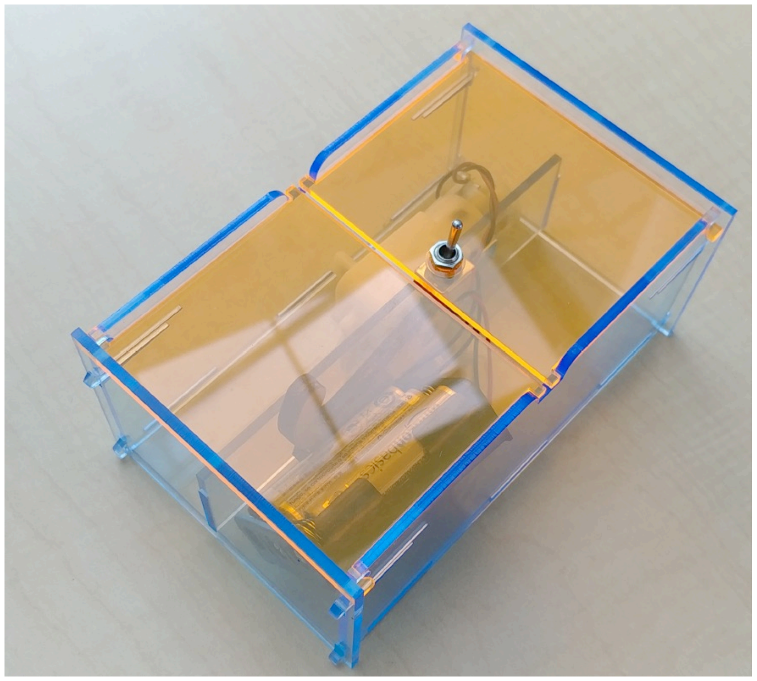
- EveryCircuit can be used to test your circuit design and see what your circuit actually does.
- EveryCircuit doesn't have DPDT or momentary switch components so you will need to use SPDT switches and manually switch them.



Action Diagram - Finite State Machine

- The motor in the box can be in three different states
- Forward, Stop, Reverse.
- How does it know when to change state?





Useless Box Operation – Boolean Logic

- The motor could be in one of three states:
  - Forward, reverse, off
  - State determined by the voltage on the motor terminals

State	M+	M-
Forward	4.5V	0V
Reverse	0V	4.5V
Off	0V (4.5V)	0v (4.5v)

- This voltage is set by the position of two switches:
  - Switch1
    - On or not on
  - Switch2
    - Limit or not limit

Boolean Variables

- The voltages on the wires in this circuit have two values
  - At least two stable values
    - 4.5V and Gnd

- The switches also seem to have two values (positions)
  - On, off; at limit and not at limit

- What does this remind you of?
  - A Boolean variable?

- Boolean Logic is a form of algebra in which all variables are reduced to True and False (1 and 0 in a binary numbering system).

Electrical Boolean Signal

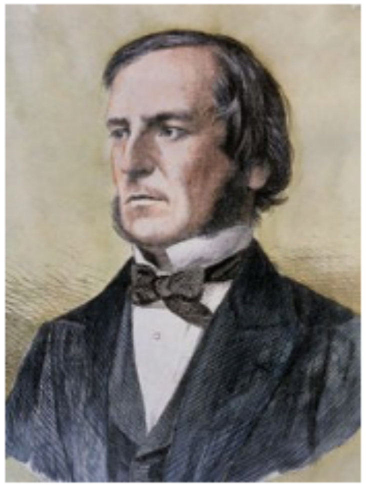
- Still is just a voltage on a node
  - And to find the voltage you use nodal analysis
    - Or some short cut

- But the voltages of the node settle to only two values
  - True (1) is a high value near the supply (4.5V)
  - False (0) is a low value near the reference (Gnd)

- Each node carries one bit of information

Who was George Boole?

A contemporary of Charles Babbage, whom he briefly met, Boole is these days credited as being the "forefather of the information age". An Englishman by birth, in 1849 he became the first professor of mathematics in Ireland's new Queen's College (now University College) Cork.



George Boole
November 2, 1815 - December 8, 1864

Boole's thinking has become the practical foundation of digital circuit design and the theoretical grounding of the digital age.

Useless Box Operation

- Think about the situation in logical values

State	M+	M-
Forward	true	false
Reverse	false	true
Off	false true	false true

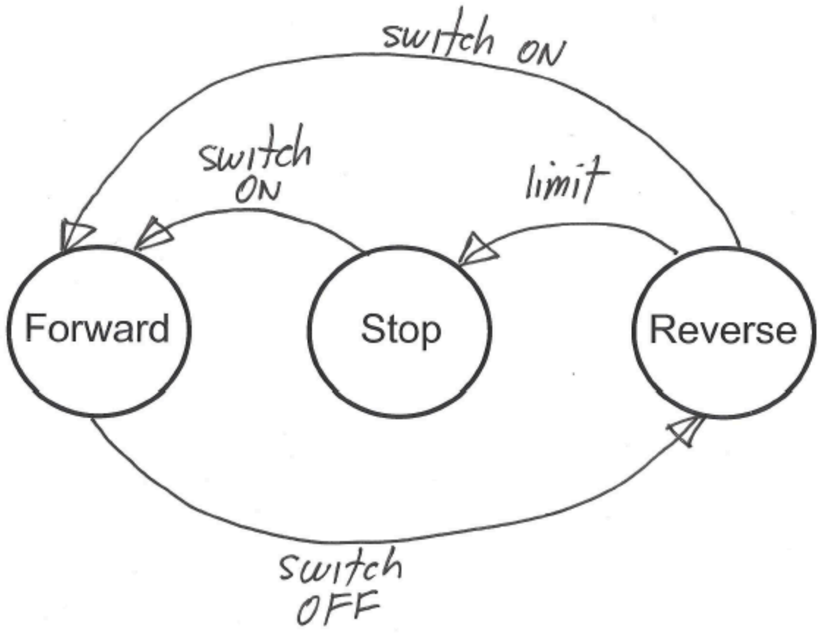
$\equiv$

State	M+	M-
Forward	4.5V	0V
Reverse	0V	4.5V
Off	0V (4.5V)	0v (4.5v)

- These outputs are a function of two switches:
  - OnSwitch
    - True, false
  - LimitSwitch
    - True, false

Useless Box Program

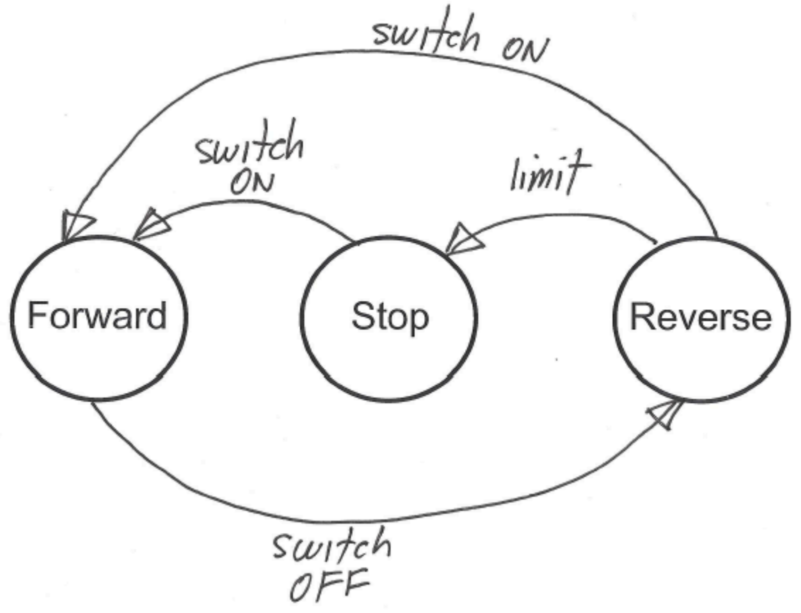
```
If (SwitchOn){
          Motor = Forward;
}
else {
          if (Limit){
                    Motor:= Stop;
          }
          else {
                    Motor = Reverse;
          }
```



- Computer programs use Boolean logic

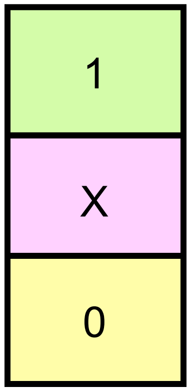
Useless Box Boolean Expression

- Operators:
  - (A && B) **AND** – Both have to be true
  - (A || B) **OR** – True if either is true
  - !(A) **NOT** – True if A is false
- What are the Boolean expressions for this Finite State Machine?
  - Forward
  - Reverse



Digital Logic

- In most programming languages
  - True = 1; False = 0
- So to build a circuit that can represent a bit {0,1}
  - Need something that can drive its output to either:
    - The power supply voltage (which we call Vdd)
    - Or the reference level (which we call ground, or gnd)

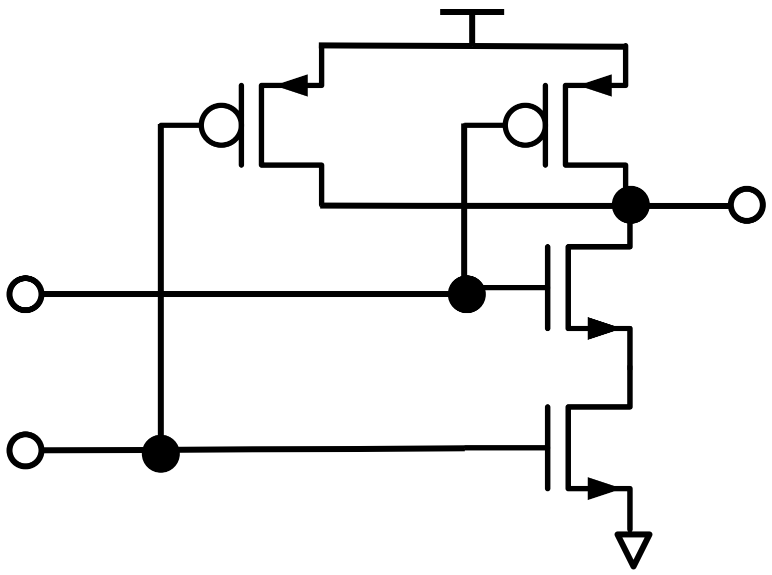


- In the useless box we built the logic from switches
  - And the first computers used mechanical switches and relays too
- But that is so yesterday …

Modern Digital Logic - CMOS

- In the next set of lecture notes you'll learn about CMOS logic gates that perform digital logic operations.
- Your Arduino has tens of thousands of these gates.

- CMOS "NAND" Gate



Truth Tables & Logic Gates

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

(A && B) AND

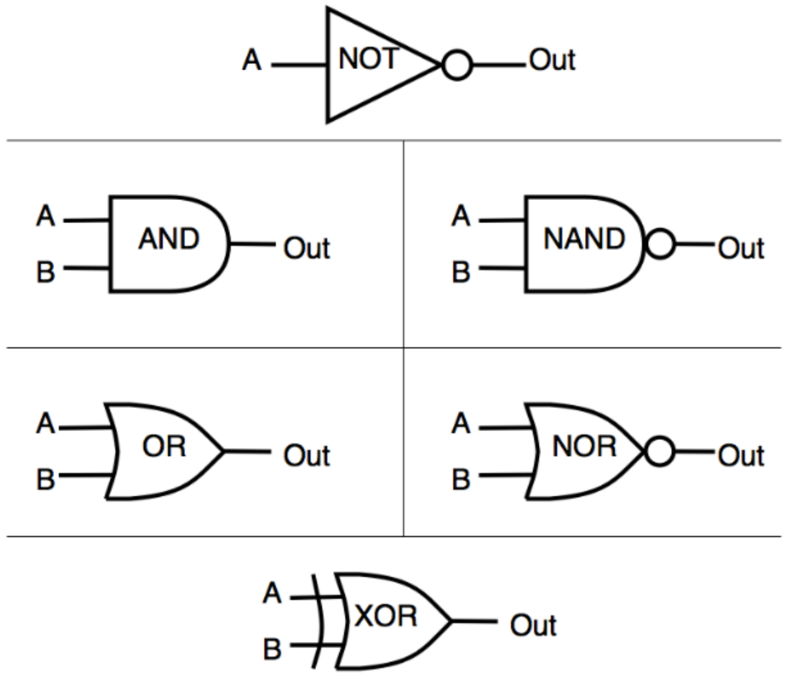
A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

(A || B) OR

A	NOT
0	1
1	0

!(A) NOT

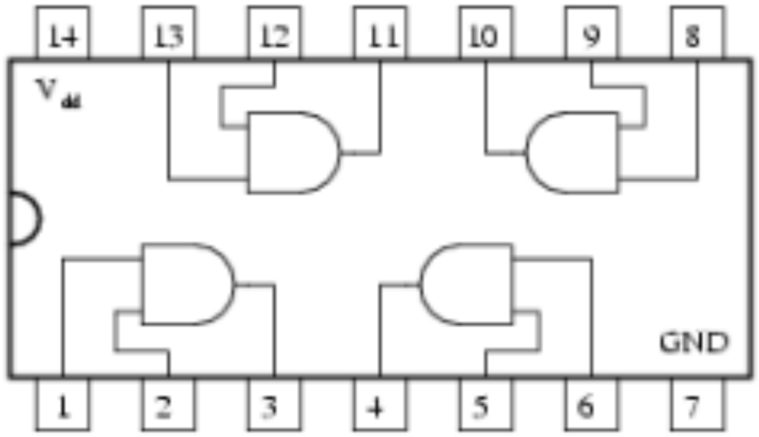
Logic Gate Symbols

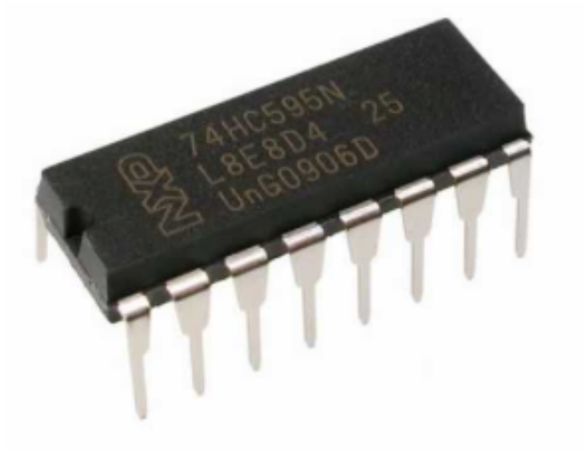


V_{dd} and Gnd

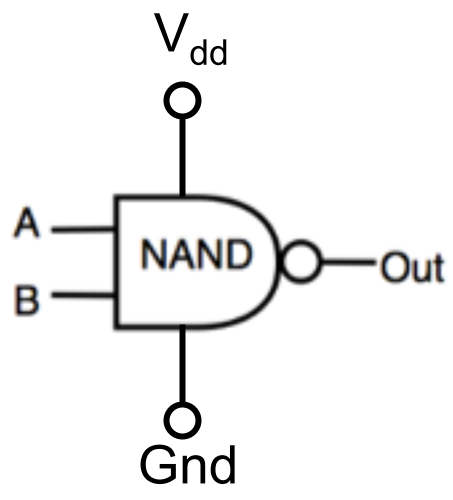
- For many circuits
  - The bottom supply is chosen as the reference
    - So it is called Gnd
  - And many devices connect to the same power supply
    - This is often called V_{dd} (or V_{cc})

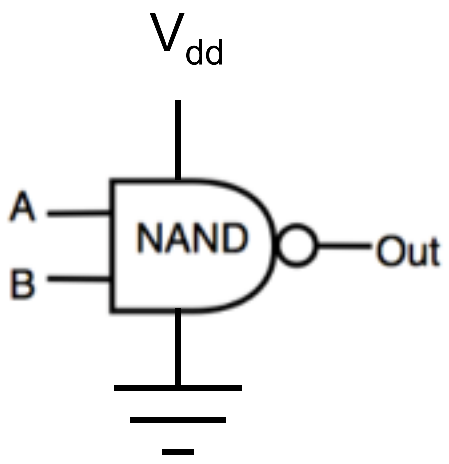
- We'll see specific examples in the next set of notes when we discuss CMOS transistors and logic gates.

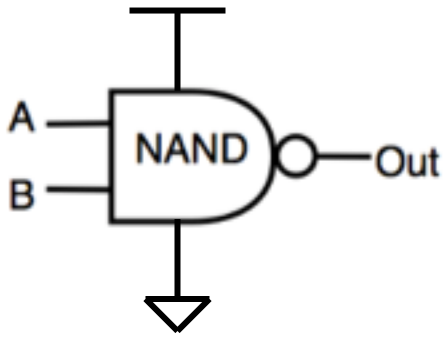




Symbols For V_{dd} and Ground







Learning Objectives

- Understand how to describe a simple system as a finite state machine

- How to represent a Boolean signal in an electrical circuit
  - V_{dd} = True; Gnd =False
  - Understand the function of AND, OR, NOT operations