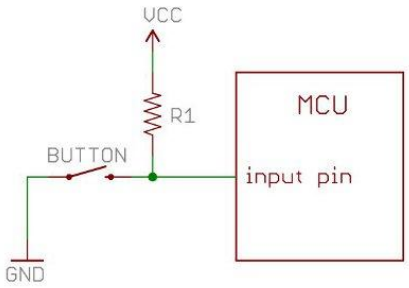
Supplement to:
"FLOATING INPUTS and RESISTORS FOR PULL-UP, PULL-DOWN, CURRENT LIMITING, and VOLTAGE DIVIDERS"
EGR/CS 333 Digital Design and Interfacing
J.Wundelich, Ph.D.

"Pull-up Resistors" adapted from https://learn.sparkfun.com/tutorials/pull-up-resistors
Color Code chart for sizing resistors from http://www.smpspowersupply.com/resistor/resistor_color_code.gif
"Introduction to Digital Circuits" adapted from: http://www.electronics-tutorials.ws/logic/logic_1.html

---

PULL-UP RESISTOR

For a MCU (**Microcontroller Unit**) with one pin configured as an input. If there is nothing connected to the pin and your program reads the state of the pin, will it be high (pulled to VCC) or low (pulled to ground)? It is difficult to tell. This phenomena is referred to as *floating*. To prevent this unknown state, a pull-up or pull-down resistor will ensure the pin is in either a high or low state, while also using a low amount of current.

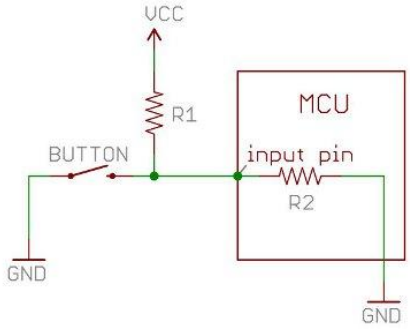


With a pull-up resistor, the input pin will read a high state when the button is not pressed. A small amount of current is flowing between VCC and the input pin (not to ground), thus the input pin reads close to VCC. When the button is pressed, it connects the input pin directly to ground. The current flows through the resistor to ground, thus the input pin reads a low state. If the resistor wasn't there, your button would connect VCC to ground, which is very bad and is known as a short.

So what value resistor should you choose?

The short and easy answer is that you want a resistor value on the order of 10kΩ for the pull-up.

A low resistor value is called a STRONG PULL-UP (more current flows), a high resistor value is called a WEAK PULL-UP (less current flows).



The value of the pull-up resistor needs to be chosen to satisfy two conditions:

1. *When button **pressed***, input pin is **pulled low**. The value of resistor R1 controls how much current you want to flow from VCC, through the button, and then to ground.
2. *When button **not pressed***, the input pin is **pulled high**. The value of the pull-up resistor controls the voltage on the input pin.

For condition 1, you don't want the resistor's value too low. The lower the resistance, the more power will be used when the button is hit. You generally want a large resistor value (10kΩ), but you don't want it too large as to conflict with condition 2. A 4MΩ resistor might work as a pull-up, but its resistance is so large (or weak) that it may not do its job 100% of the time.

The general rule for condition 2 is to use a pull-up resistor (R1) that is an order of magnitude (1/10th) less than the input impedance (R2) of the input pin. An input pin on a microcontroller has an impedance that can vary from 100k-1MΩ. For this discussion, impedance is just a fancy way of saying resistance and is represented by R2 in the picture above. So, when the button is not pressed, a very small amount of current flows from VCC through R1 and into the input pin. The pull-up resistor R1 and input pin impedance R2 divides the voltage, and this voltage needs to be high enough for the input pin to read a high state.

For example, if you use a 1MΩ resistor for the pull-up R1 and the input pin's impedance R2 is on the order of 1MΩ (forming a voltage divider), the voltage on the input pin is going to be around half of VCC, and the microcontroller might not register the pin being in a high state. On a 5V system, what does the MCU read on the input pin if the voltage is 2.5V? Is it a high or a low? The MCU doesn't know and you might read either a high or a low. A resistance of 10k to 100kΩ for R1 should avoid most problems.

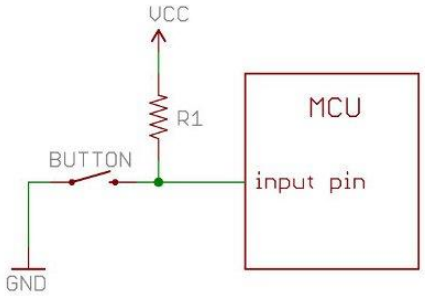
Since pull-up resistors are so commonly needed, many MCUs, like the ATmega328 microcontroller on the **Arduino** platform, have internal pull-ups that can be enabled and disabled. To enable internal pull-ups on an Arduino, you can use the following line of code in your setup() function:

```
pinMode(5, INPUT_PULLUP); // Enable internal pull-up resistor on pin 5
```

Another thing to point out is that **the larger the resistance for the pull-up, the slower the pin is to respond to voltage changes**. This is because the system that feeds the input pin is essentially a capacitor coupled with the pull-up resistor, thus forming a RC filter, and RC filters take some time to charge and discharge. If you have a really fast changing signal (like USB), a high value pull-up resistor can limit the speed at which the pin can reliably change state. **This is why you will often see 1k to 4.7KΩ resistors on USB signal lines**.

All of these factors play into the decision on what value pull-up resistor to use.

Calculating a Pull-up Resistor Value



Let's say you want to limit the current to approximately 1mA when the button is pressed in the circuit above, where Vcc = 5V. What resistor value should you use? Use Ohm's Law:

$$V = I \cdot R$$

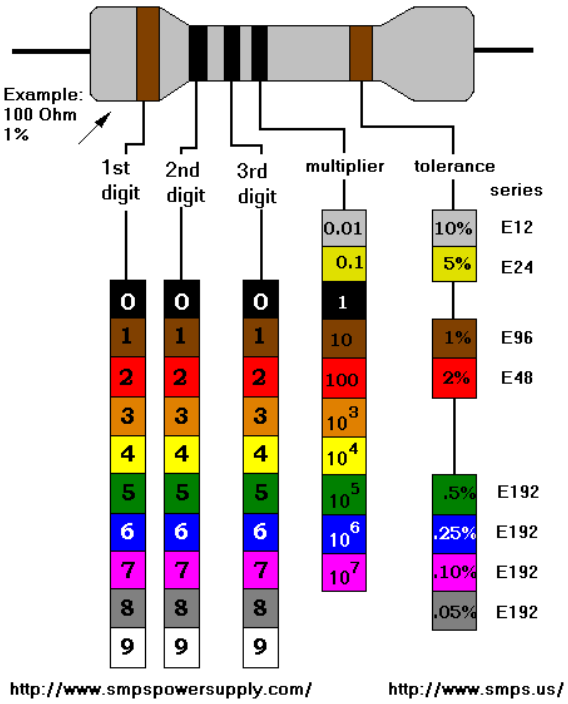
Referring to the schematic above, Ohm's Law now is:

$$V_{CC} = (\text{current through } R1) \cdot R1$$

Rearrange the above equation with some simple algebra to solve for the resistor:

$$R1 = \frac{V_{CC}}{\text{current through } R1} = \frac{5\text{V}}{0.001\text{A}} = 5\text{k}\,Ohms$$

Remember to convert all of your units into volts, amps and Ohms before calculating (e.g. 1mA = 0.001 Amps). The solution is to use a 5kΩ resistor.



But what is INTERNAL RESISTANCE of SSI (Small Scale Integration) Gates like what we put on Breadboards in Circuit trainers?

ANSWER: It Varies depending on type chips; See:

Introduction to Digital Logic Gates

Standard digital logic gates are available in two basic families or forms, **TTL** which stands for *Transistor-Transistor Logic* such as the 7400 series, and **CMOS** which stands for *Complementary Metal-Oxide-Silicon* which is the 4000 series of chips. This notation of TTL or CMOS refers to the logic technology used to manufacture the integrated circuit, (IC)

Digital Logic Gate (Chip Package)

TTL logic IC's use NPN and PNP type Bipolar Junction Transistors while **CMOS** logic IC's use complementary MOSFET or JFET type Field Effect Transistors for both their input and output circuitry.

As well as TTL and CMOS technology, Digital Logic Gates can also be made by connecting together diodes, transistors and resistors to produce **RTL**, Resistor-Transistor logic gates, **DTL**, Diode-Transistor logic gates or **ECL**, Emitter-Coupled logic gates but these are less common now compared to the popular **CMOS** family.

Classification of Integrated Circuits

- Small Scale Integration or (**SSI**) – Contain up to 10 transistors or a few gates within a single package such as AND, OR, NOT gates.
- Medium Scale Integration or (**MSI**) – between 10 and 100 transistors or tens of gates within a single package and perform digital operations such as adders, decoders, counters, flip-flops and multiplexers.
- Large Scale Integration or (**LSI**) – between 100 and 1,000 transistors or hundreds of gates and perform specific digital operations such as I/O chips, memory, arithmetic and logic units.
- Very-Large Scale Integration or (**VLSI**) – between 1,000 and 10,000 transistors or thousands of gates and perform computational operations such as processors, large memory arrays and programmable logic devices. The term VLSI is also often used for all circuits larger than this; however:
    - *Super-Large Scale Integration or (SLSI) – between 10,000 and 100,000 transistors within a single package and perform computational operations such as microprocessor chips, micro-controllers, basic PICs and calculators.*
    - *Ultra-Large Scale Integration or (ULSI) – more than 1 million transistors – the big boys that are used in computers CPUs, GPUs, video processors, micro-controllers, FPGAs and complex PICs.*

The **Digital Logic Gate** is the basic building block from which all digital electronic circuits and microprocessor based systems are constructed from. Basic digital logic gates perform logical operations of AND, OR and NOT on binary numbers.

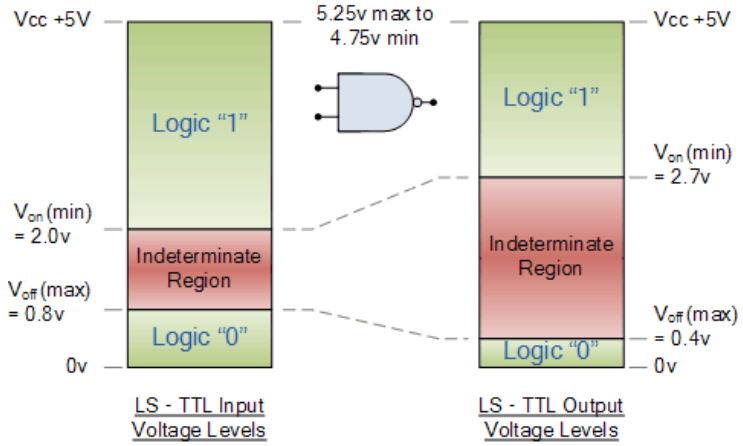
In digital logic design only two voltage levels or states are allowed and these states are generally referred to as Logic "1" and Logic "0", High and Low, or True and False. These two states are represented in Boolean Algebra and standard truth tables by the binary digits of "1" and "0" respectively.

A good example of a digital state is a simple light switch as it is either "ON" or "OFF" but not both at the same time. Then we can summarise the relationship between these various digital states as being:

Boolean Algebra	Boolean Logic	Voltage State
Logic "1"	True (T)	High (H)
Logic "0"	False (F)	Low (L)

In standard TTL (transistor-transistor logic) IC's there is a pre-defined voltage range for the input and output voltage levels which define exactly what is a logic "1" level and what is a logic "0" level and these are shown below.

TTL Input & Output Voltage Levels

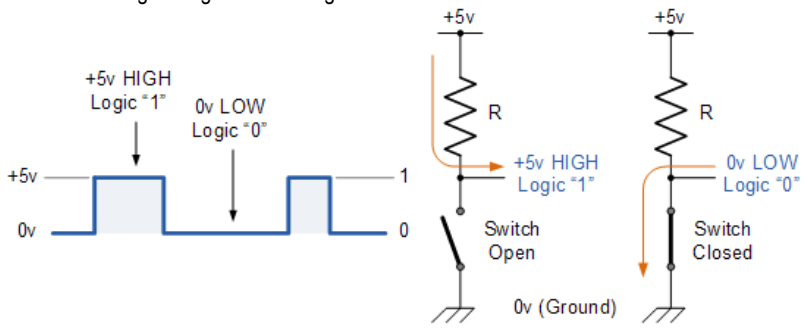


There are a large variety of logic gate types in both the bipolar 7400 and the CMOS 4000 families of digital logic gates such as 74Lxx, 74LSxx, 74ALSxx, 74HCxx, 74HCTxx, 74ACTxx etc, with each one having its own distinct advantages and disadvantages compared to the other. The exact switching voltage required to produce either a logic "0" or a logic "1" depends upon the specific logic group or family.

However, when using a standard +5 volt supply any TTL voltage input between 2.0v and 5v is considered to be a logic "1" or "HIGH" while any voltage input below 0.8v is recognised as a logic "0" or "LOW". The voltage region in between these two voltage levels either as an input or as an output is called the *Indeterminate Region* and operating within this region may cause the logic gate to produce a false output.

The CMOS 4000 logic family uses different levels of voltages compared to the TTL types as they are designed using field effect transistors, or FET's. In CMOS technology a **logic "1" level operates between 3.0 and 18 volts and a logic "0" level is below 1.5 volts**.

Then from the above observations, we can define the ideal **Digital Logic Gate** as one that has a "LOW" level logic "0" of 0 volts (ground) and a "HIGH" level logic "1" of +5 volts and this can be demonstrated as Ideal Digital Logic Gate Voltage Levels

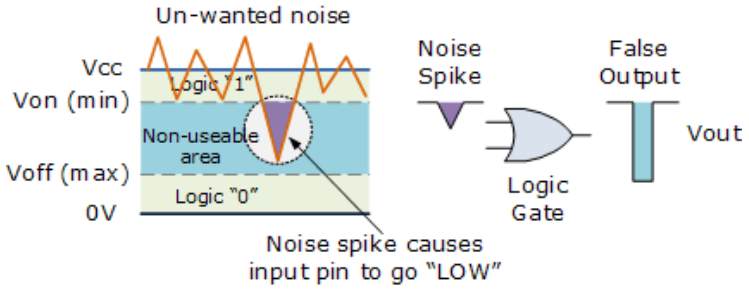


Where the opening or closing of the switch produces either a logic level "1" or a logic level "0" with the resistor R being known as a "pull-up" resistor.

Digital Logic Noise

However, between these defined HIGH and LOW values lies what is generally called a "no-man's land" (the blue area's above) and if we apply a signal voltage of a value within this no-man's land area we do not know whether the logic gate will respond to it as a level "0" or as a level "1", and the output will become unpredictable. **Noise** is the name given to a random and unwanted voltage that is induced into electronic circuits by external interference, such as from nearby switches, power supply fluctuations or from wires and other conductors that pick-up stray electromagnetic radiation. Then in order for a logic gate not to be influence by noise in must have a certain amount of noise margin or noise immunity.

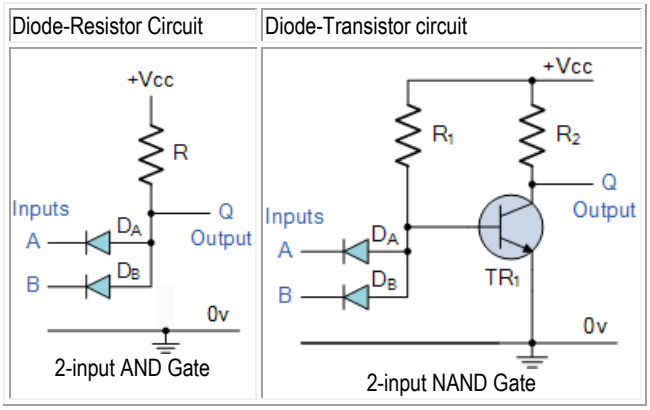
Digital Logic Gate Noise Immunity



In the example above, the noise signal is superimposed onto the Vcc supply voltage and as long as it stays above the minimum level (Von-min) the input an corresponding output of the logic gate are unaffected. But when the noise level becomes large enough and a noise spike causes the HIGH voltage level to drop below this minimum level, the logic gate may interpret this spike as a LOW level input and switch the output accordingly producing a false output switching. Then in order for the logic gate not to be affected by noise it must be able to tolerate a certain amount of unwanted noise on its input without changing the state of its output.

Digital Logic Gates

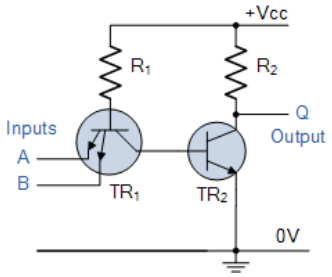
Digital logic gates can be made by combining transistors, diodes and resistors with a simple example of a Diode-Resistor Logic (DRL) AND gate and a Diode-Transistor Logic (DTL) NAND gate given below.



The 2-input Diode-Resistor AND gate can be converted into a NAND gate by the addition of a single transistor inverting (NOT) stage. Using discrete components such as diodes, resistors and transistors to make digital logic gate circuits are not used in practical commercially available logic IC's as these circuits suffer from propagation delay or gate delay and also power loss due to the pull-up resistors. Another disadvantage of diode-resistor logic is that there is no "Fan-out" facility which is the ability of a single output to drive many inputs of the next stages. Also this type of design does not turn fully "OFF" as a Logic "0" produces an output voltage of 0.6v (diode voltage drop), so the following TTL and CMOS circuit designs are used instead.

TTL Logic Gates

The simple Diode-Resistor AND gate above uses separate diodes for its inputs, one for each input. As a transistor is made up off two diode circuits connected together representing an NPN or a PNP device, the input diodes of the DTL circuit can be replaced by one single NPN transistor with multiple emitter inputs as shown.



2-input NAND Gate

As the NAND gate contains a single stage inverting NPN transistor circuit (TR_2) an output logic level "1" at Q is only present when both the emitters of TR_1 are connected to logic level "0" or ground allowing base current to pass through the PN junctions of the emitter and not the collector. The multiple emitters of TR_1 are connected as inputs thus producing a NAND gate function.

In standard TTL logic gates, the transistors operate either completely in the "cut off" region, or else completely in the saturated region, Transistor as a Switch

The "74" Sub-families of Integrated Circuits

With improvements in the circuit design to take account of propagation delays, current consumption, fan-in and fan-out requirements etc, this type of TTL bipolar transistor technology forms the basis of the prefixed "74" family of digital logic IC's, such as the "7400" Quad 2-input AND gate, or the "7402" Quad 2-input OR gate, etc.

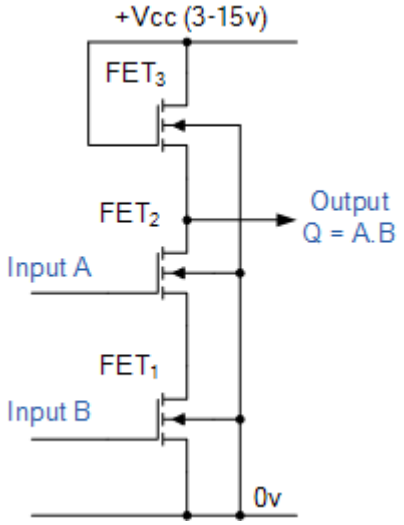
Sub-families of the 74xx series IC's are available relating to the different technologies used to fabricate the gates and they are denoted by the letters in between the 74 designation and the device number. There are a number of TTL sub-families available that provide a wide range of switching speeds and power consumption such as the 74**L**00 or 74**ALS**00 AND gate, were the "L" stands for "Low-power TTL" and the "ALS" stands for "Advanced Low-power Schottky TTL" and these are listed below. • 74xx or 74Nxx: Standard TTL – These devices are the original TTL family of logic gates introduced in the early 70's. They have a propagation delay of about 10ns and a power consumption of about 10mW.

- 74**L**xx: Low Power TTL – Power consumption was improved over standard types by **INCREASING THE NUMBER OF INTERNAL RESISTANCES** but at the cost of a reduction in switching speed.
- 74**H**xx: High Speed TTL – Switching speed improved by **REDUCING THE NUMBER OF INTERNAL RESISTANCES**. This also increased power consumption.
- 74**S**xx: Schottky TTL – **Schottky** technology is used to improve input impedance, switching speed and power consumption (2mW) compared to the 74Lxx and 74Hxx types.
- 74**LS**xx: Low Power Schottky TTL – Same as 74Sxx types but with **INCREASED INTERNAL RESISTANCES** to improve power consumption.
- 74ASxx: Advanced Schottky TTL – Improved design over 74Sxx Schottky types optimised to increase switching speed at the expense of power consumption of about 22mW.
- 74ALSxx: Advanced Low Power Schottky TTL – Lower power consumption of about 1mW and higher switching speed of about 4nS compared to 74LSxx types.
- 74HCxx: High Speed CMOS – CMOS technology and transistors to reduce power consumption of less than 1uA with CMOS compatible inputs.
- 74HCTxx: High Speed CMOS – CMOS technology and transistors to reduce power consumption of less than 1uA but has increased propagation delay of about 16nS due to the TTL compatible inputs.

CMOS Digital Logic Gate

One of the main disadvantages with the **TTL** digital logic gate series is that the logic gates **are based on bipolar transistor logic technology and as transistors are current operated devices, they consume large amounts of power** from a fixed +5 volt power supply.

Also, TTL bipolar transistor gates have a limited operating speed when switching from an "OFF" state to an "ON" state and vice-versa called the "gate" or "propagation delay". To overcome these limitations complementary MOS called "**CMOS**" logic gates using "Field Effect Transistors" or FET's were developed. As these gates use both P-channel and N-channel MOSFET's as their input device, **at quiescent conditions with no switching, the power consumption of CMOS gates is almost zero,** (1 to 2uA) making them ideal for use in low-power battery circuits and with switching speeds upwards of 100MHz for use in high frequency timing and computer circuits.



2-input NAND Gate

This CMOS gate example contains 3 N-channel MOSFET's, one for each input FET_1 and FET_2 and one for the output FET_3. When both the inputs A and B are at logic level "0", FET_1 and FET_2 are both switched "OFF" giving an output logic "1" from the source of FET_3.
When one or both of the inputs are at logic level "1" current flows through the corresponding FET giving an output state at Q equivalent to logic "0", thus producing a NAND gate function.

Improvements in the circuit design with regards to switching speed, low power consumption and improved propagation delays has resulted in the standard CMOS 4000 "CD" family of logic IC's being developed that complement the TTL range.

As with the standard TTL digital logic gates, all the major digital logic gates and devices are available in the CMOS package such as the CD4011, a Quad 2-input NAND gate, or the CD4001, a Quad 2-input NOR gate along with all their sub-families.

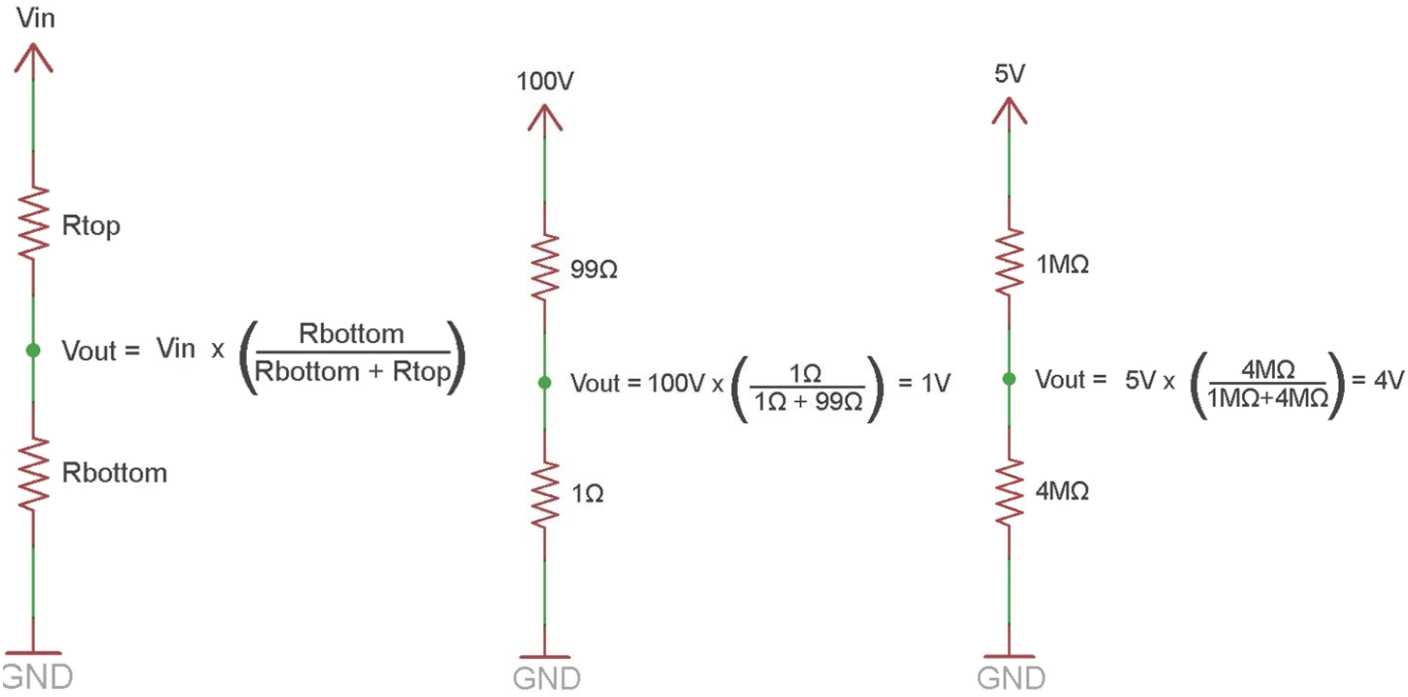
Like TTL logic, complementary MOS (CMOS) circuits take advantage of the fact that both N-channel and P-channel devices can be fabricated together on the same substrate material to form various logic functions.

One of the main disadvantages **with the CMOS range of IC's compared to their equivalent TTL types is that they are easily damaged by static electricity** so extra care must be taken when handling these devices.

Also **TTL logic gates operate on single +5V voltages** for both their input and output levels, **CMOS** digital logic gates **operate on a single supply voltage of between +3 and +18 volts.**

CHANGING VOLTAGES

In EGR/CS333, the preferred method for controlling a device that operates with a different voltage than the device that is originating the signals is to use relays, However, to control a device that operates with a lesser voltage than the device that is originating the signals, you may use resistors to create Voltage Divider circuits, however if you do this, use relatively large resistors so you don't burn up the resisters due a large power dissipation in the resistors; And this method is not good for driving loads of more than 10mA – so use relays instead to drive motors! SEE: https://www.youtube.com/watch?v=XxLKfAZrhbM



So, if you replaced the bottom resistor in the circuit above on the right (i.e., the one with Vin = 5V for TTL circuits) with a switch, which almost always has zero resistance when closed, and infinite resistance when open, the upper resister is then a pull-up resistor,

*and Vout = Vin * (0 / (0 + Rtop)) = 0 volts (i.e.,Logic-0) when the switch is closed*
*and Vout = Vin * (infinity / (infinity + Rtop)) = ~ Vin (i.e.,Logic-1) when the switch is open*

And **<u>use relatively big resistors to keep resistors from burning up</u>** since:
Power dissipated = (Vin)^2 / (Rtop + Rbottom)

Also, this is not a regulated power supply, and <u>shouldn't be used for supplying more than ~10mA</u>

For example, the 100mA load on the right results in a Vout = 0.51V which <u>won't drive the motor</u> !