# CMSC 313
# COMPUTER ORGANIZATION
# &
# ASSEMBLY LANGUAGE
# PROGRAMMING

LECTURE 19, SPRING 2013

# TOPICS TODAY

- **Introduction to Digital Logic**
- **Semiconductors, Transistors & Gates**

# INTRODUCTION
# TO
# DIGITAL LOGIC

# Chapter 3 Objectives

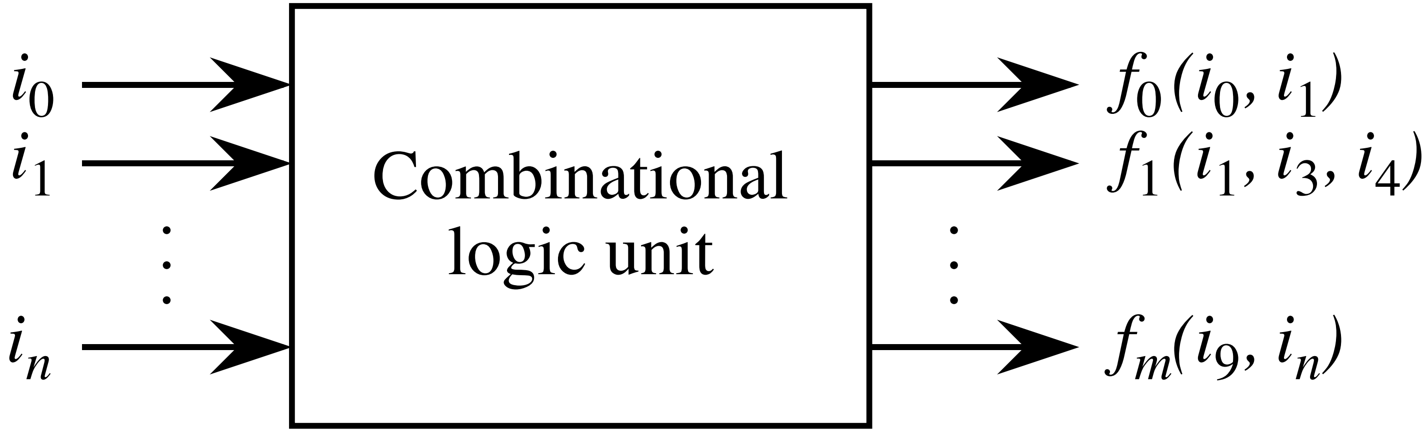
- Understand the relationship between Boolean logic and digital computer circuits.
- Learn how to design simple logic circuits.
- Understand how digital circuits work together to form complex computer systems.

# Some Definitions

- ***Combinational logic:*** **a digital logic circuit in which logical decisions are made based only on combinations of the inputs. *e.g.* an adder.**
- ***Sequential logic:*** **a circuit in which decisions are made based on combinations of the current inputs as well as the past history of inputs. *e.g.* a memory unit.**
- ***Finite state machine:*** **a circuit which has an internal state, and whose outputs are functions of both current inputs and its internal state. *e.g.* a vending machine controller.**

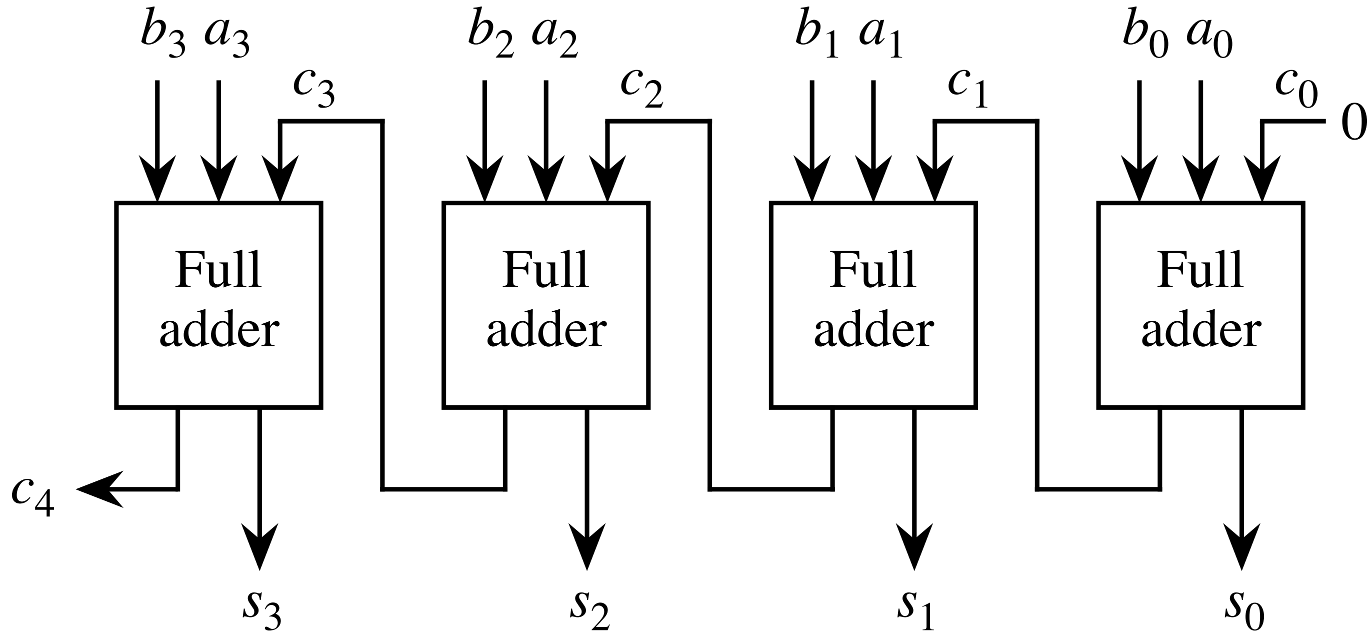
# The Combinational Logic Unit

- Translates a set of inputs into a set of outputs according to one or more mapping functions.
- Inputs and outputs for a CLU normally have two distinct (binary) values: high and low, 1 and 0, 0 and 1, or 5 V and 0 V for example.
- The outputs of a CLU are strictly functions of the inputs, and the outputs are updated immediately after the inputs change. A set of inputs $i_0 – i_n$ are presented to the CLU, which produces a set of outputs according to mapping functions $f_0 – f_m$.



# Ripple Carry Adder

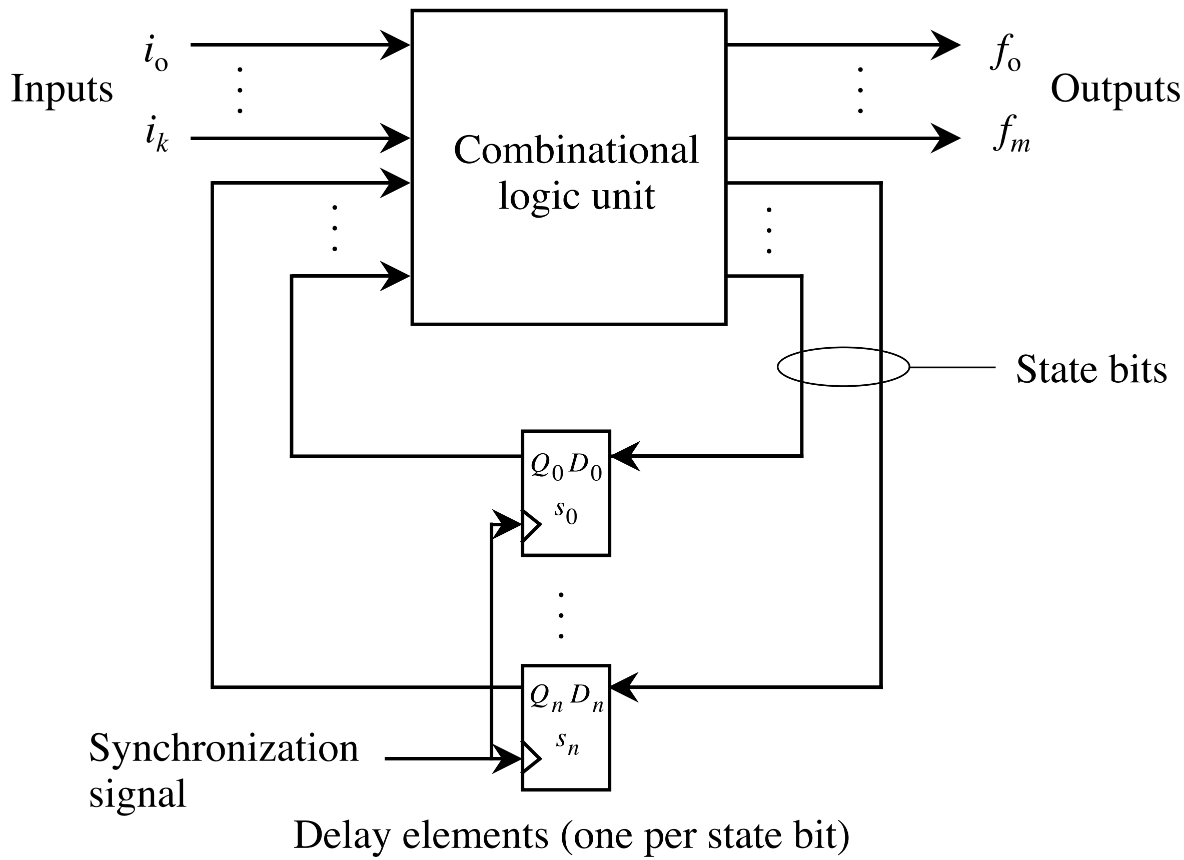
- Two binary numbers *A* and *B* are added from right to left, creating a sum and a carry at the outputs of each full adder for each bit position.



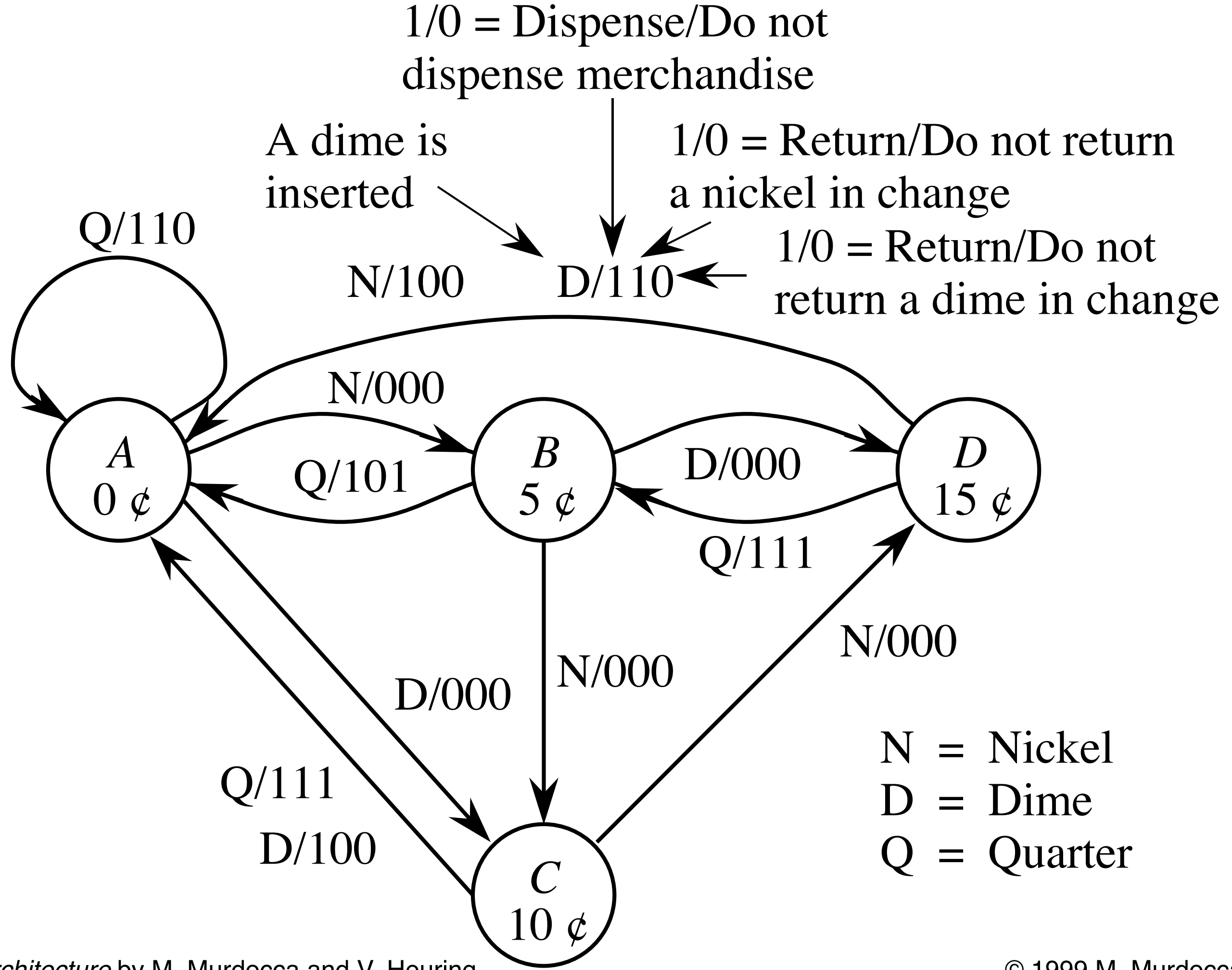
# Classical Model of a Finite State Machine

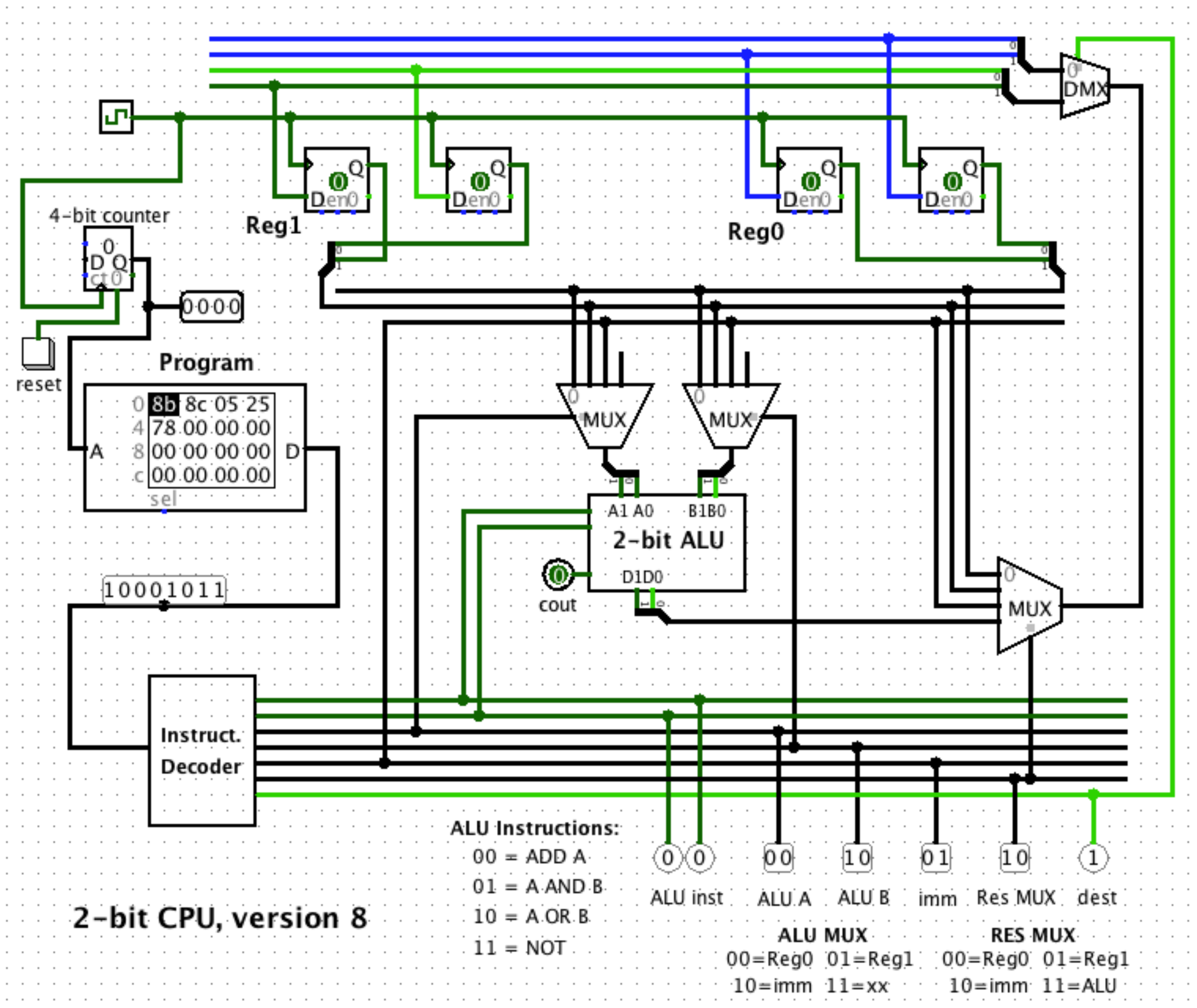
- **An FSM is composed of a combinational logic unit and delay elements (called *flip-flops*) in a feedback path, which maintains state information.**



# Vending Machine State Transition Diagram





DMX
MUX
Reg1
Reg0
4-bit counter
reset
0000
Program
0 8b 8c 05 25
4 78 00 00 00
8 00 00 00 00
c 00 00 00 00
sel
10001011
2-bit ALU
A1 A0
B1B0
D1D0
cout
Instruct. Decoder
ALU Instructions:
00 = ADD A
01 = A AND B
10 = A OR B
11 = NOT
ALU inst
ALU A
ALU B
imm
Res MUX
dest
ALU MUX
00=Reg0 01=Reg1
10=imm 11=xx
RES MUX
00=Reg0 01=Reg1
10=imm 11=ALU
2-bit CPU, version 8

# 3.2 Boolean Algebra

- Boolean algebra is a mathematical system for the manipulation of variables that can have one of two values.
  - In formal logic, these values are "true" and "false."
  - In digital systems, these values are "on" and "off," 1 and 0, or "high" and "low."
- Boolean expressions are created by performing operations on Boolean variables.
  - Common Boolean operators include AND, OR, and NOT.

# 3.2 Boolean Algebra

- A Boolean operator can be completely described using a truth table.
- The truth table for the Boolean operators AND and OR are shown at the right.
- The AND operator is also known as a Boolean product. The OR operator is the Boolean sum.

X AND Y

| X | Y | XY |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

X OR Y

| X | Y | X+Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# 3.2 Boolean Algebra

- The truth table for the Boolean NOT operator is shown at the right.
- The NOT operation is most often designated by an overbar. It is sometimes indicated by a prime mark ( ‘ ) or an “elbow” (¬).

NOT x

| $x$ | $\overline{x}$ |
| --- | --- |
| 0 | 1 |
| 1 | 0 |

# 3.2 Boolean Algebra

- A Boolean function has:
  - At least one Boolean variable,
  - At least one Boolean operator, and
  - At least one input from the set {0,1}.
- It produces an output that is also a member of the set {0,1}.

**Now you know why the binary numbering system is so handy in digital systems**.

# 3.2 Boolean Algebra

- The truth table for the Boolean function:

  $$F(x,y,z) = x\bar{z}+y$$

  is shown at the right.
- To make evaluation of the Boolean function easier, the truth table contains extra (shaded) columns to hold evaluations of subparts of the function.

$F(x,y,z) = x\bar{z}+y$

| x | y | z | $\bar{z}$ | $x\bar{z}$ | $x\bar{z}+y$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

# 3.2 Boolean Algebra

- As with common arithmetic, Boolean operations have rules of precedence.
- The NOT operator has highest priority, followed by AND and then OR.
- This is how we chose the (shaded) function subparts in our table.

$F(x,y,z) = x\bar{z}+y$

| x | y | z | $\bar{z}$ | $x\bar{z}$ | $x\bar{z}+y$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

# 3.2 Boolean Algebra

- Digital computers contain circuits that implement Boolean functions.
- The simpler that we can make a Boolean function, the smaller the circuit that will result.
  - Simpler circuits are cheaper to build, consume less power, and run faster than complex circuits.
- With this in mind, we always want to reduce our Boolean functions to their simplest form.
- There are a number of Boolean identities that help us to do this.

# 3.2 Boolean Algebra

- Most Boolean identities have an AND (product) form as well as an OR (sum) form. We give our identities using both forms. Our first group is rather intuitive:

| Identity Name | AND Form | OR Form |
|---|---|---|
| Identity Law | $1x = x$ | $0 + x = x$ |
| Null Law | $0x = 0$ | $1 + x = 1$ |
| Idempotent Law | $xx = x$ | $x + x = x$ |
| Inverse Law | $x\overline{x} = 0$ | $x + \overline{x} = 1$ |

# 3.2 Boolean Algebra

- Our second group of Boolean identities should be familiar to you from your study of algebra:

| Identity Name | AND Form | OR Form |
|---|---|---|
| Commutative Law | xy = yx | x+y = y+x |
| Associative Law | (xy)z = x(yz) | (x+y)+z = x + (y+z) |
| Distributive Law | x+yz = (x+y)(x+z) | x(y+z) = xy+xz |

# 3.2 Boolean Algebra

- Our last group of Boolean identities are perhaps the most useful.
- If you have studied set theory or formal logic, these laws are also familiar to you.

| Identity Name | AND Form | OR Form |
|---|---|---|
| Absorption Law | $x(x+y) = x$ | $x + xy = x$ |
| DeMorgan's Law | $\overline{(xy)} = \overline{x} + \overline{y}$ | $\overline{(x+y)} = \overline{x}\,\overline{y}$ |
| Double Complement Law | $\overline{(\overline{x})} = x$ | |

# 3.2 Boolean Algebra

- We can use Boolean identities to simplify:

$$F(X,Y,Z) = (X+Y)(X+\overline{Y})(\overline{X\overline{Z}})$$

as follows:

| | |
|---|---|
| $(X + Y)(X + \overline{Y})(\overline{X\overline{Z}})$ | |
| $(X + Y)(X + \overline{Y})(\overline{X} + Z)$ | DeMorgan's Law<br>Double complement Law |
| $(XX + X\overline{Y} + YX + Y\overline{Y})(\overline{X} + Z)$ | Distributive Law |
| $((X + Y\overline{Y}) + X(Y + \overline{Y}))(\overline{X} + Z)$ | Commutative and Distributive Laws |
| $((X + 0) + X(1))(\overline{X} + Z)$ | Inverse Law |
| $X(\overline{X} + Z)$ | Idempotent and Identity Laws |
| $X\overline{X} + XZ$ | Distributive Law |
| $0 + XZ$ | Inverse Law |
| $XZ$ | Identity Law |

# 3.2 Boolean Algebra

- Sometimes it is more economical to build a circuit using the complement of a function (and complementing its result) than it is to implement the function directly.
- DeMorgan's law provides an easy way of finding the complement of a Boolean function.
- Recall DeMorgan's law states:

$$\overline{(xy)} = \overline{x} + \overline{y} \quad \text{and} \quad \overline{(x+y)} = \overline{x}\,\overline{y}$$

# 3.2 Boolean Algebra

- DeMorgan's law can be extended to any number of variables.
- Replace each variable by its complement and change all ANDs to ORs and all ORs to ANDs.
- Thus, we find the the complement of:

$$F(X,Y,Z) = (XY) + (\overline{X}Y) + (X\overline{Z})$$

is:

$$\overline{F}(X,Y,Z) = \overline{(XY) + (\overline{X}Z) + (Y\overline{Z})}$$

$$= \overline{(XY)}\,\overline{(\overline{X}Z)}\,\overline{(Y\overline{Z})}$$

$$= (\overline{X}+\overline{Y})(X+\overline{Z})(\overline{Y}+Z)$$

# 3.2 Boolean Algebra

- Through our exercises in simplifying Boolean expressions, we see that there are numerous ways of stating the same Boolean expression.
  – These "synonymous" forms are *logically equivalent*.
  – Logically equivalent expressions have identical truth tables.
- In order to eliminate as much confusion as possible, designers express Boolean functions in *standardized* or *canonical* form.

# 3.2 Boolean Algebra

- There are two canonical forms for Boolean expressions: sum-of-products and product-of-sums.
  - Recall the Boolean product is the AND operation and the Boolean sum is the OR operation.
- In the sum-of-products form, ANDed variables are ORed together.
  - For example: $F(x,y,z) = xy + xz + yz$
- In the product-of-sums form, ORed variables are ANDed together:
  - For example: $F(x,y,z) = (x+y)(x+z)(y+z)$

# 3.2 Boolean Algebra

- It is easy to convert a function to sum-of-products form using its truth table.
- We are interested in the values of the variables that make the function true (=1).
- Using the truth table, we list the values of the variables that result in a true function value.
- Each group of variables is then ORed together.

$F(x,y,z) = x\bar{z}+y$

| x | y | z | $x\bar{z}+y$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

# 3.2 Boolean Algebra

- The sum-of-products form for our function is:

$$F(x,y,z) = (\bar{x}y\bar{z}) + (\bar{x}yz) + (x\bar{y}\bar{z}) + (xy\bar{z}) + (xyz)$$

**We note that this function is not in simplest terms. Our aim is only to rewrite our function in canonical sum-of-products form.**

$F(x,y,z) = x\bar{z}+y$

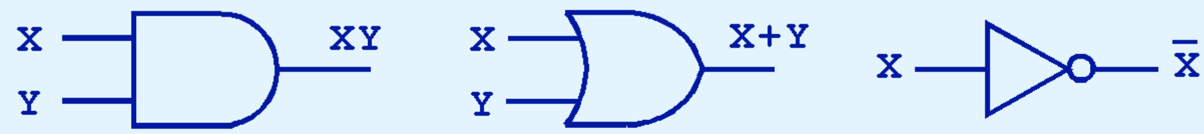
| x | y | z | $x\bar{z}+y$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

# 3.3 Logic Gates

- We have looked at Boolean functions in abstract terms.
- In this section, we see that Boolean functions are implemented in digital computer circuits called gates.
- A gate is an electronic device that produces a result based on two or more input values.
  - In reality, gates consist of one to six transistors, but digital designers think of them as a single unit.
  - Integrated circuits contain collections of gates suited to a particular purpose.

# 3.3 Logic Gates

- The three simplest gates are the AND, OR, and NOT gates.




X AND Y

| X | Y | XY |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

X OR Y

| X | Y | X+Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

NOT X

| X | $\overline{X}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

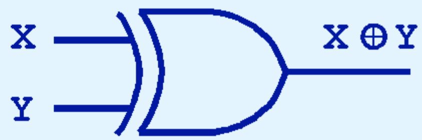
- They correspond directly to their respective Boolean operations, as you can see by their truth tables.

# 3.3 Logic Gates

- Another very useful gate is the exclusive OR (XOR) gate.
- The output of the XOR operation is true only when the values of the inputs differ.

X XOR Y

| X | Y | X ⊕ Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



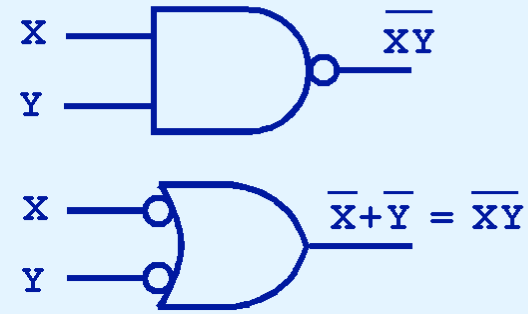

**Note the special symbol ⊕ for the XOR operation.**

# 3.3 Logic Gates

- NAND and NOR are two very important gates. Their symbols and truth tables are shown at the right.

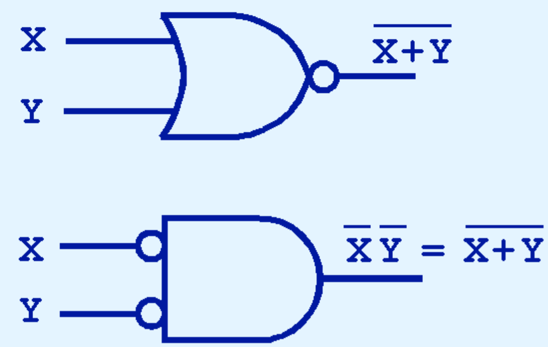
X NAND Y

| X | Y | X NAND Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



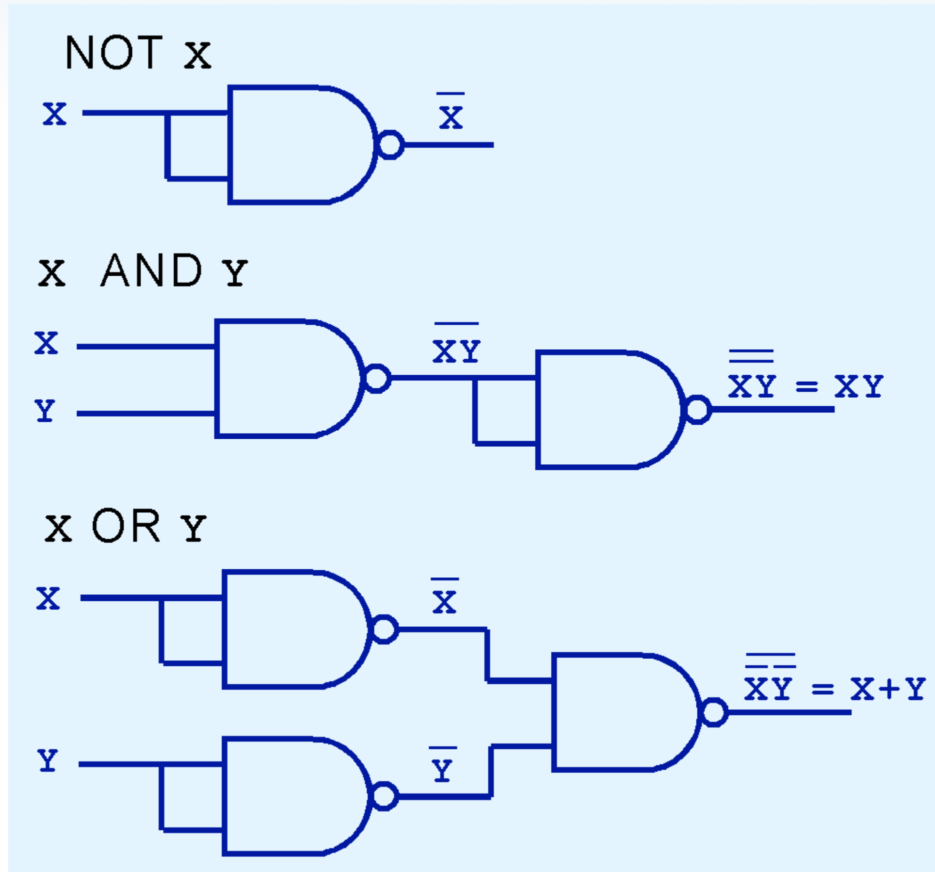

X NOR Y

| X | Y | X NOR Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



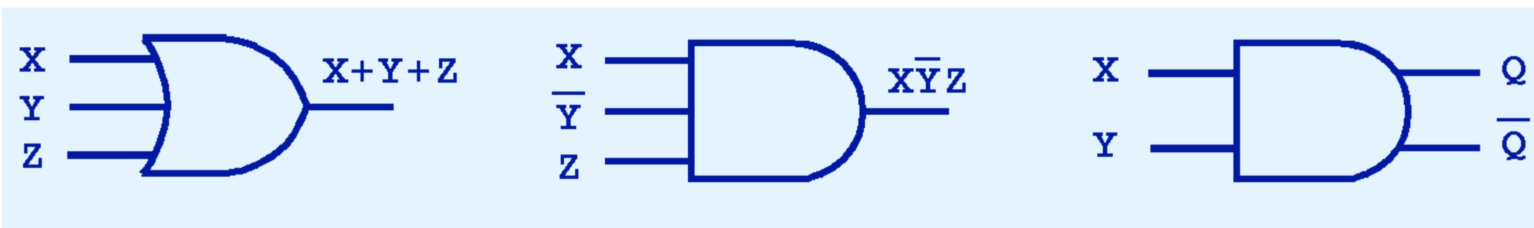
# 3.3 Logic Gates

- NAND and NOR are known as *universal gates* because they are inexpensive to manufacture and any Boolean function can be constructed using only NAND or only NOR gates.



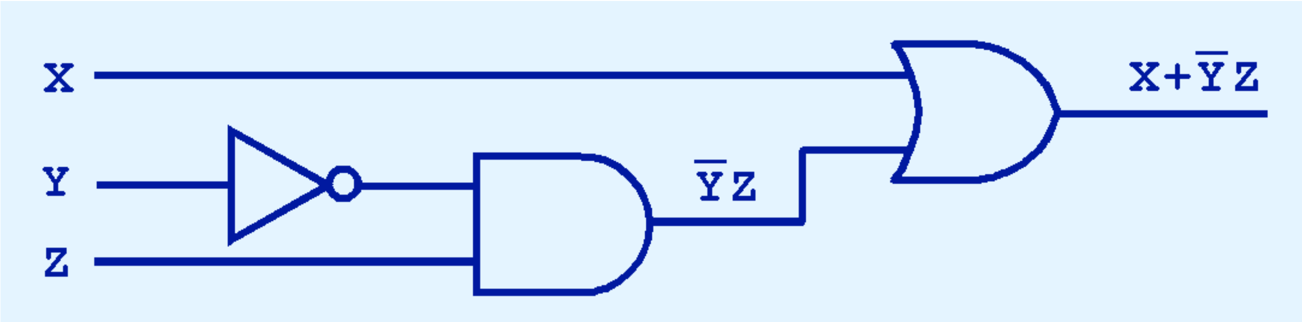
# 3.3 Logic Gates

- Gates can have multiple inputs and more than one output.
  - A second output can be provided for the complement of the operation.
  - We'll see more of this later.



# 3.3 Logic Gates

- The main thing to remember is that combinations of gates implement Boolean functions.
- The circuit below implements the Boolean function: $F(X,Y,Z) = X+\overline{Y}Z$




**We simplify our Boolean expressions so that we can create simpler circuits.**

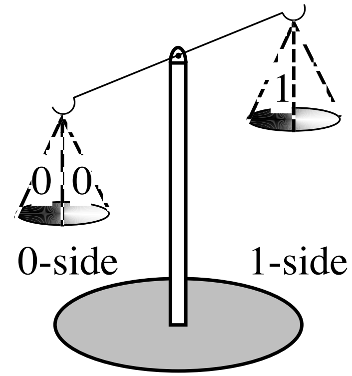
# Sum-of-Products Form: The Majority Function

- The SOP form for the 3-input majority function is:

  $M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = m3 + m5 + m6 + m7 = \Sigma\,(3, 5, 6, 7)$.

- Each of the $2^n$ terms are called *minterms*, ranging from 0 to $2^n - 1$.
- Note relationship between minterm number and boolean value.

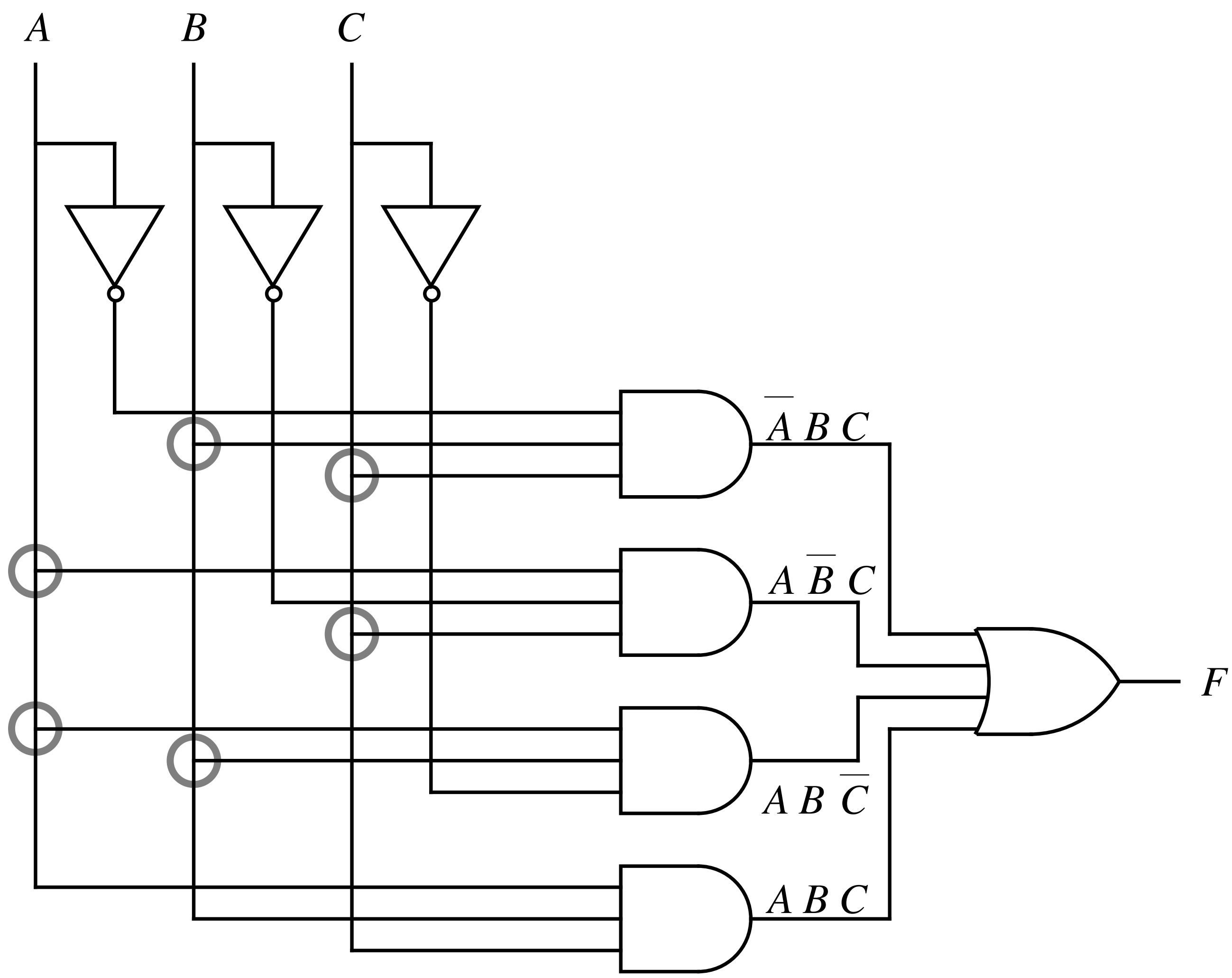
| *Minterm Index* | *A* | *B* | *C* | *F* |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |




A balance tips to the left or right depending on whether there are more 0's or 1's.

# AND-OR Implementation of Majority

- **Gate count is 8, gate input count is 19.**



# Sum of Products (a.k.a. disjunctive normal form)

- OR (i.e., sum) together rows with output 1
- AND (i.e., product) of variables represents each row
  e.g., in row 3 when $x_1 = 0$ AND $x_2 = 1$ AND $x_3 = 1$
  or when $\overline{x_1} \cdot x_2 \cdot x_3 = 1$
- MAJ3$(x_1, x_2, x_3) = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3 = \sum m(3, 5, 6, 7)$

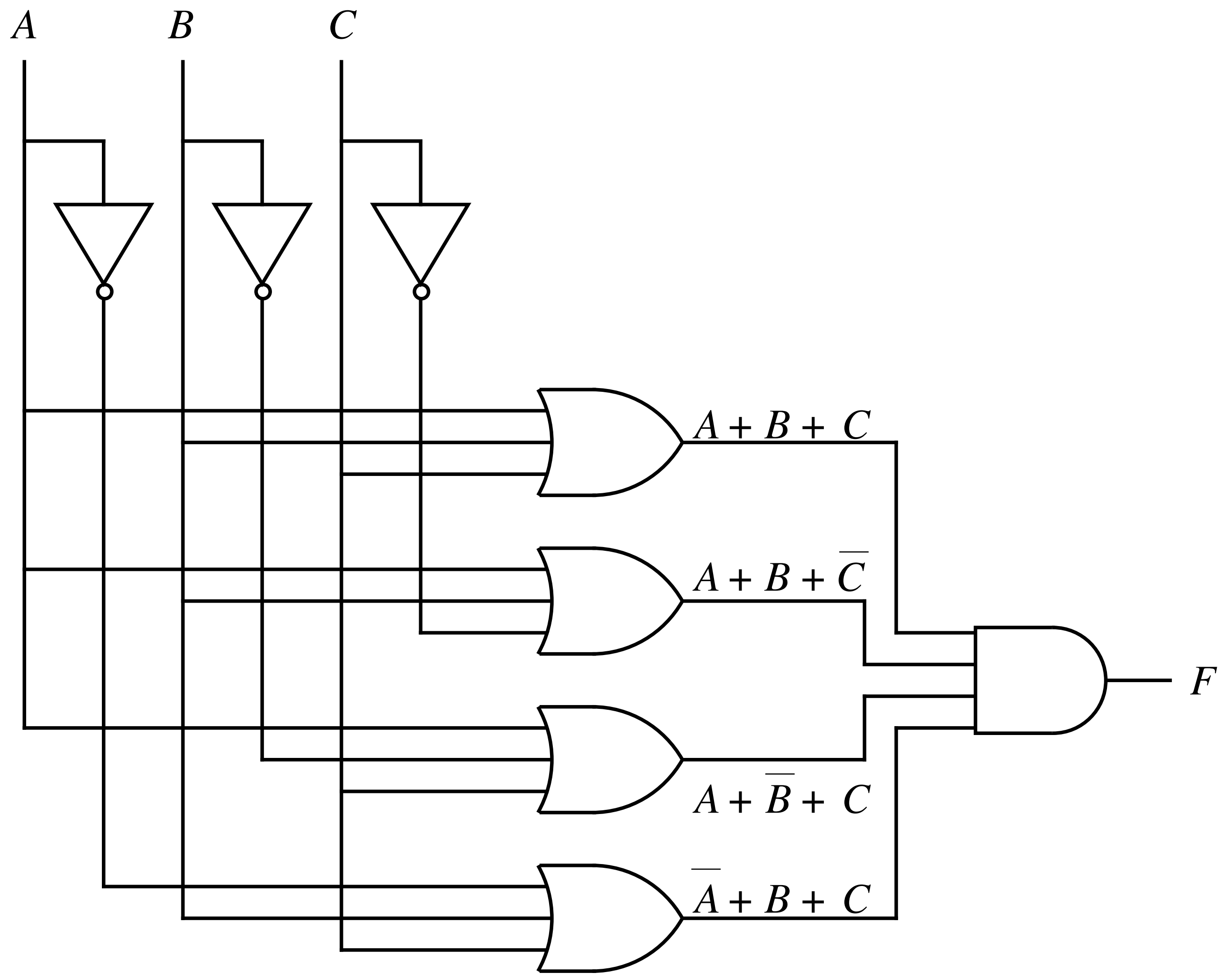
| | $x_1$ | $x_2$ | $x_3$ | MAJ3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

# Product of Sums (a.k.a. conjunctive normal form)

- AND (i.e., product) of rows with output 0
- OR (i.e., sum) of variables represents negation of each row
  e.g., NOT in row 2 when $x_1 = 1$ OR $x_2 = 0$ OR $x_3 = 1$
  or when $x_1 + \overline{x_2} + x_3 = 1$
- MAJ3$(x_1, x_2, x_3) = (x_1{+}x_2{+}x_3)(x_1{+}x_2{+}\overline{x_3})(x_1{+}\overline{x_2}{+}x_3)(\overline{x_1}{+}x_2{+}x_3)$
  $= \prod M(0, 1, 2, 4)$

| | $x_1$ | $x_2$ | $x_3$ | MAJ3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

# OR-AND Implementation of Majority



# Equivalences

- Every Boolean function can be written as a truth table
- Every truth table can be written as a Boolean formula (SOP or POS)
- Every Boolean formula can be converted into a combinational circuit
- Every combinational circuit is a Boolean function
- Later you might learn other equivalencies:

  finite automata $\equiv$ regular expressions

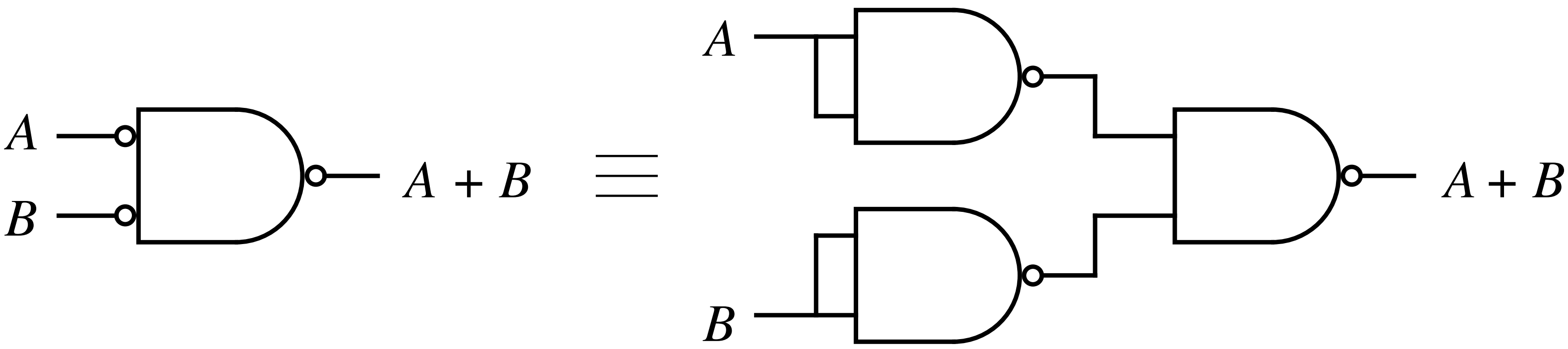
  computable functions $\equiv$ programs

## Universality

- Every Boolean function can be written as a Boolean formula using AND, OR and NOT operators.
- Every Boolean function can be implemented as a combinational circuit using AND, OR and NOT gates.
- Since AND, OR and NOT gates can be constructed from NAND gates, NAND gates are universal.

# All-NAND Implementation of OR

- NAND alone implements all other Boolean logic gates.

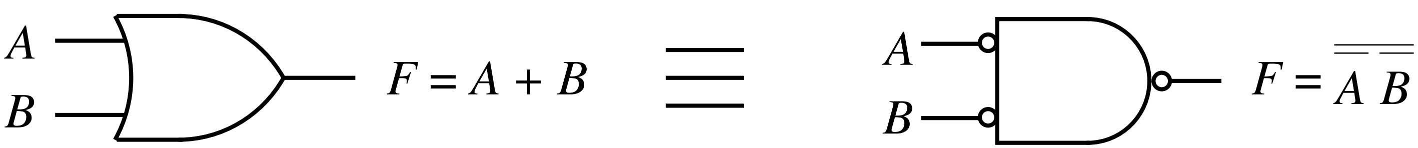


# DeMorgan’s Theorem

| $A$ $B$ | $\overline{A\,B} = \overline{A} + \overline{B}$ | | $\overline{A + B} = \overline{A}\,\overline{B}$ | |
|---|---|---|---|---|
| 0 0 | 1 | 1 | 1 | 1 |
| 0 1 | 1 | 1 | 0 | 0 |
| 1 0 | 1 | 1 | 0 | 0 |
| 1 1 | 0 | 0 | 0 | 0 |

DeMorgan’s theorem: $A + B = \overline{\overline{A + B}} = \overline{\overline{A}\,\overline{B}}$

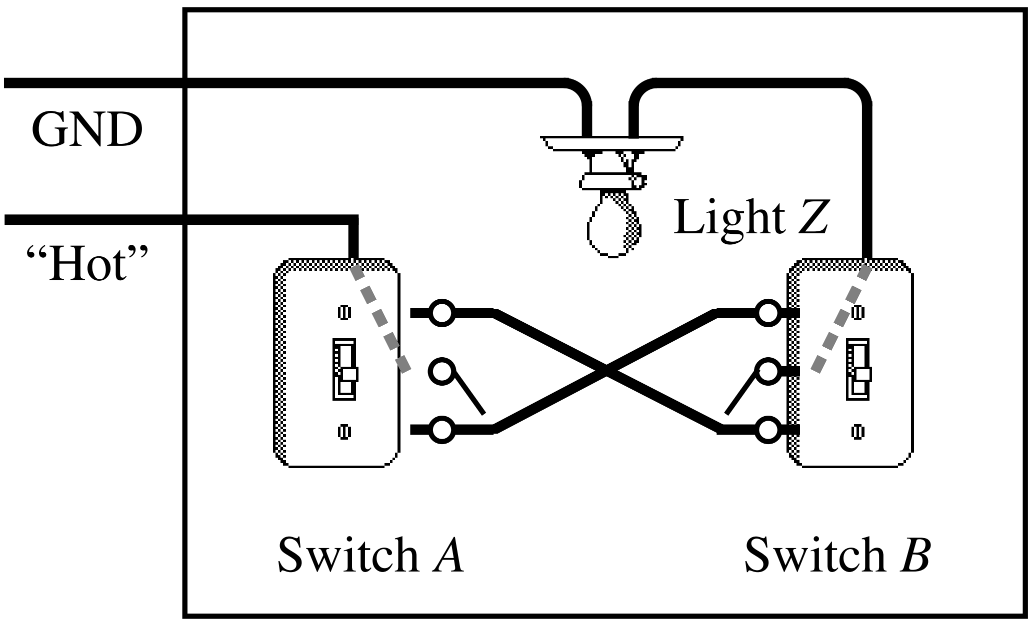


# SEMICONDUCTORS,
# TRANSISTORS
# &
# GATES

# How do we make gates???

# A Truth Table

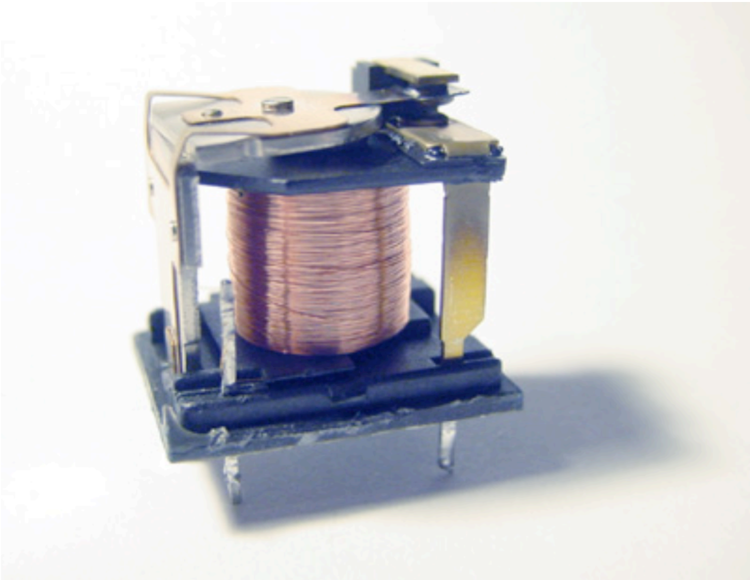
- **Developed in 1854 by George Boole.**
- **Further developed by Claude Shannon (Bell Labs).**
- **Outputs are computed for all possible input combinations (how many input combinations are there?)**
- **Consider a room with two light switches. How must they work?**

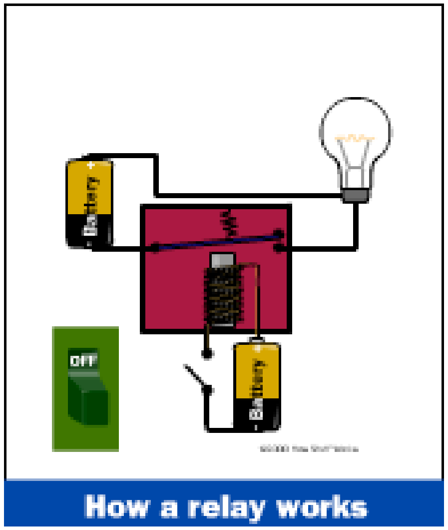



| Inputs | | Output |
|---|---|---|
| *A* | *B* | *Z* |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Electrically Operated Switch

- **Example: a relay**






source: http://www.howstuffworks.com/relay.htm

# Semiconductors

- Electrical properties of silicon
- Doping: adding impurities to silicon
- Diodes and the P-N junction
- Field-effect transistors

*Los Alamos National Laboratory's Chemistry Division Presents*

# *Periodic Table of the Elements*

| Period \ Group | 1 IA 1A | 2 IIA 2A | 3 IIIB 3B | 4 IVB 4B | 5 VB 5B | 6 VIB 6B | 7 VIIB 7B | 8 | 9 VIII | 10 | 11 IB 1B | 12 IIB 2B | 13 IIIA 3A | 14 IVA 4A | 15 VA 5A | 16 VIA 6A | 17 VIIA 7A | 18 vIIIA 8A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 H 1.008 | | | | | | | | | | | | | | | | | 2 He 4.003 |
| 2 | 3 Li 6.941 | 4 Be 9.012 | | | | | | | | | | | 5 B 10.81 | 6 C 12.01 | 7 N 14.01 | 8 O 16.00 | 9 F 19.00 | 10 Ne 20.18 |
| 3 | 11 Na 22.99 | 12 Mg 24.31 | | | | | | | | | | | 13 Al 26.98 | 14 Si 28.09 | 15 P 30.97 | 16 S 32.07 | 17 Cl 35.45 | 18 Ar 39.95 |
| 4 | 19 K 39.10 | 20 Ca 40.08 | 21 Sc 44.96 | 22 Ti 47.88 | 23 V 50.94 | 24 Cr 52.00 | 25 Mn 54.94 | 26 Fe 55.85 | 27 Co 58.47 | 28 Ni 58.69 | 29 Cu 63.55 | 30 Zn 65.39 | 31 Ga 69.72 | 32 Ge 72.59 | 33 As 74.92 | 34 Se 78.96 | 35 Br 79.90 | 36 Kr 83.80 |
| 5 | 37 Rb 85.47 | 38 Sr 87.62 | 39 Y 88.91 | 40 Zr 91.22 | 41 Nb 92.91 | 42 Mo 95.94 | 43 Tc (98) | 44 Ru 101.1 | 45 Rh 102.9 | 46 Pd 106.4 | 47 Ag 107.9 | 48 Cd 112.4 | 49 In 114.8 | 50 Sn 118.7 | 51 Sb 121.8 | 52 Te 127.6 | 53 I 126.9 | 54 Xe 131.3 |
| 6 | 55 Cs 132.9 | 56 Ba 137.3 | 57 La* 138.9 | 72 Hf 178.5 | 73 Ta 180.9 | 74 W 183.9 | 75 Re 186.2 | 76 Os 190.2 | 77 Ir 190.2 | 78 Pt 195.1 | 79 Au 197.0 | 80 Hg 200.5 | 81 Tl 204.4 | 82 Pb 207.2 | 83 Bi 209.0 | 84 Po (210) | 85 At (210) | 86 Rn (222) |
| 7 | 87 Fr (223) | 88 Ra (226) | 89 Ac~ (227) | 104 Rf (257) | 105 Db (260) | 106 Sg (263) | 107 Bh (262) | 108 Hs (265) | 109 Mt (266) | 110 --- () | 111 --- () | 112 --- () | | 114 --- () | | 116 --- () | | 118 --- () |

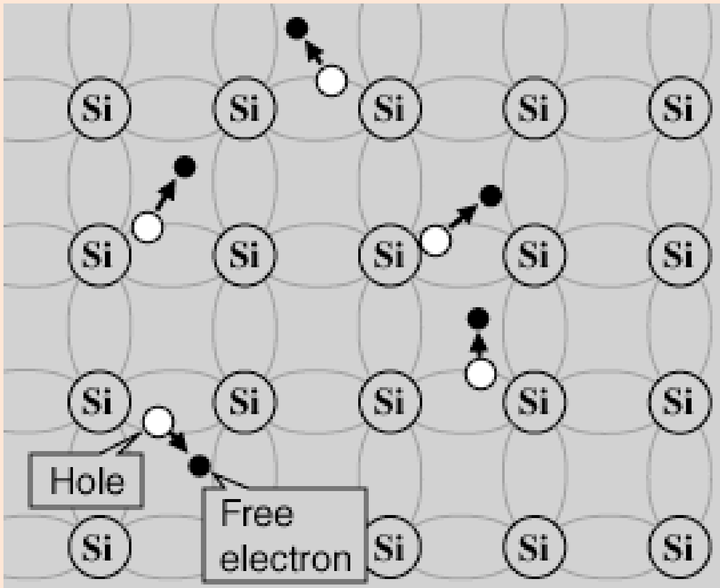
| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lanthanide Series* | 58 Ce 140.1 | 59 Pr 140.9 | 60 Nd 144.2 | 61 Pm (147) | 62 Sm 150.4 | 63 Eu 152.0 | 64 Gd 157.3 | 65 Tb 158.9 | 66 Dy 162.5 | 67 Ho 164.9 | 68 Er 167.3 | 69 Tm 168.9 | 70 Yb 173.0 | 71 Lu 175.0 |
| Actinide Series~ | 90 Th 232.0 | 91 Pa (231) | 92 U (238) | 93 Np (237) | 94 Pu (242) | 95 Am (243) | 96 Cm (247) | 97 Bk (247) | 98 Cf (249) | 99 Es (254) | 100 Fm (253) | 101 Md (256) | 102 No (254) | 103 Lr (257) |

# Intrinsic Semiconductor

A silicon crystal is different from an insulator because at any temperature above absolute zero temperature, there is a finite probability that an electron in the lattice will be knocked loose from its position, leaving behind an electron deficiency called a "hole".

If a voltage is applied, then both the electron and the hole can contribute to a small current flow.

The conductivity of a semiconductor can be modeled in terms of the band theory of solids. The band model of a semiconductor suggests that at ordinary temperatures there is a finite possibility that electrons can reach the conduction band and contribute to electrical conduction.



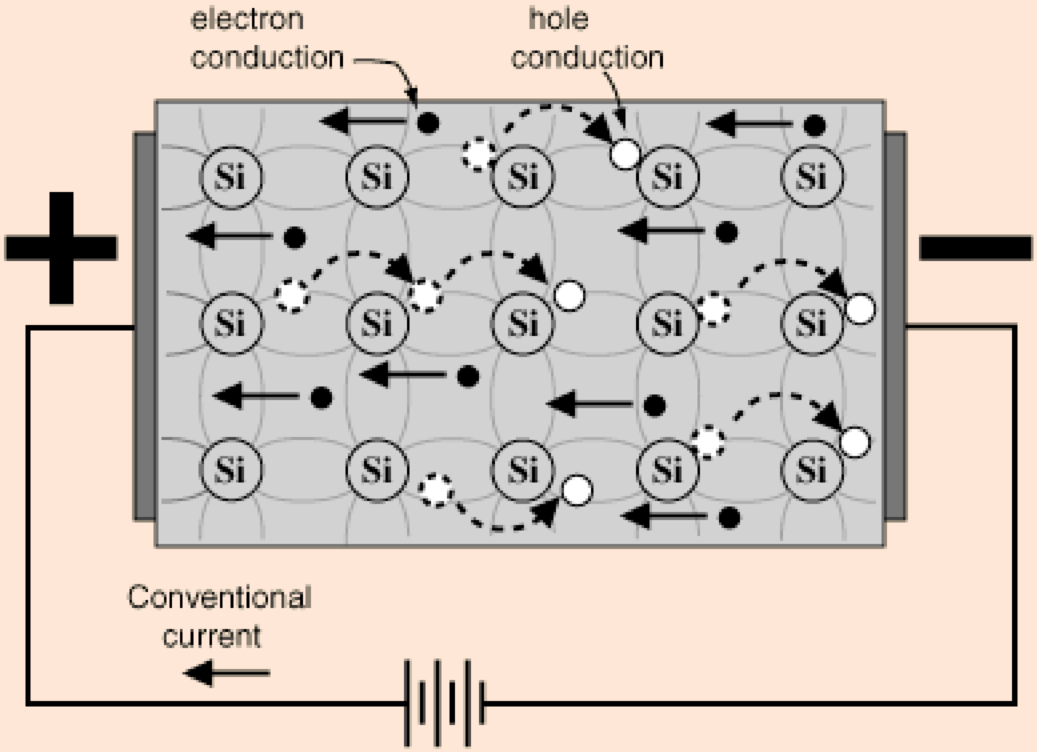

The term intrinsic here distinguishes between the properties of pure "intrinsic" silicon and the dramatically different properties of doped n-type or p-type semiconductors.

Index

Semiconductor concepts

# Semiconductor Current

Both electrons and holes contribute to current flow in an intrinsic semiconductor.




Further discussion

Index

Semiconductor concepts

HyperPhysics***** Condensed Matter

R Nave

Go Back

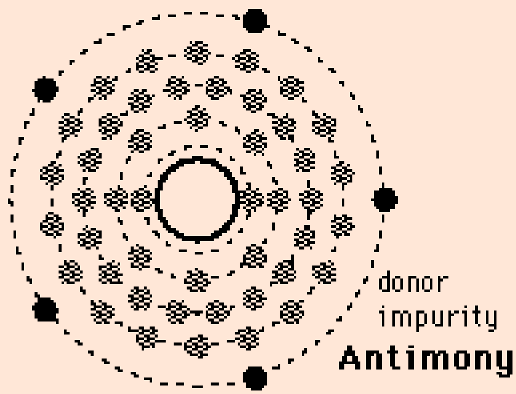
# The Doping of Semiconductors

The addition of a small percentage of foreign atoms in the regular crystal lattice of silicon or germanium produces dramatic changes in their electrical properties, producing n-type and p-type semiconductors.

Pentavalent impurities

(5 valence electrons) produce n-type semiconductors by contributing extra electrons.

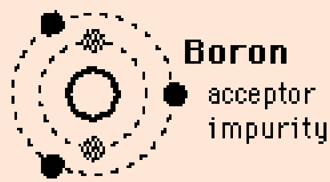
Antimony
Arsenic
Phosphorous




Trivalent impurities

(3 valence electrons) produce p-type semiconductors by producing a "hole " or electron deficiency.

Boron
Aluminum
Gallium

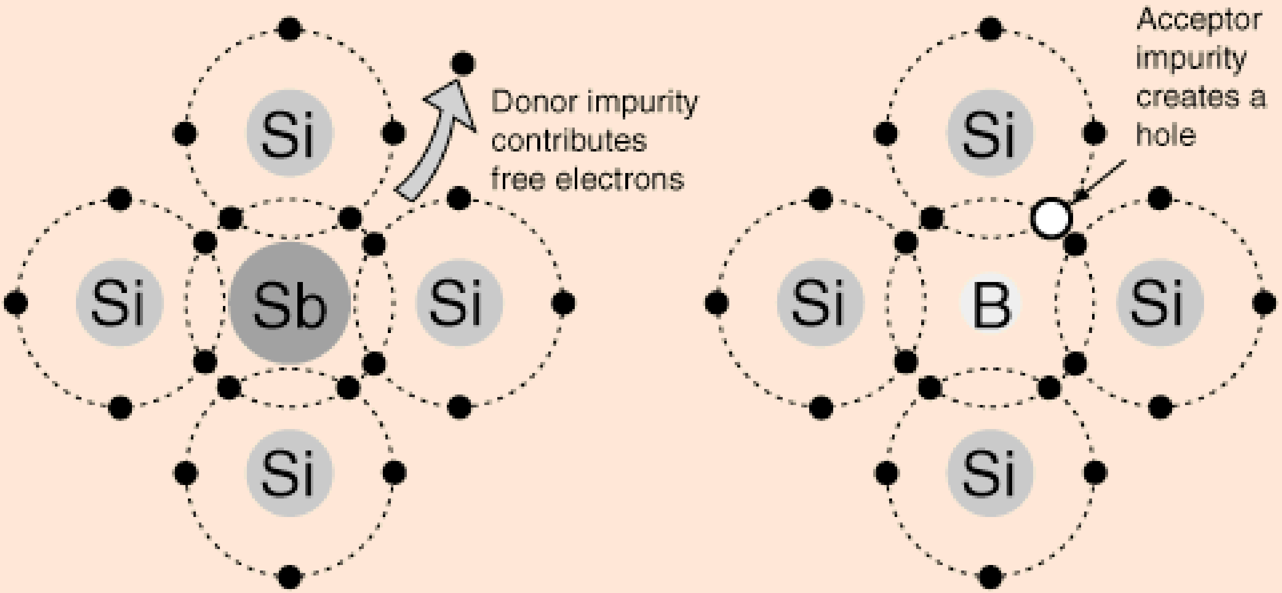



Index

Semiconductor concepts

Doped Semiconductors

# P- and N- Type Semiconductors




Click on either for further information.

Index

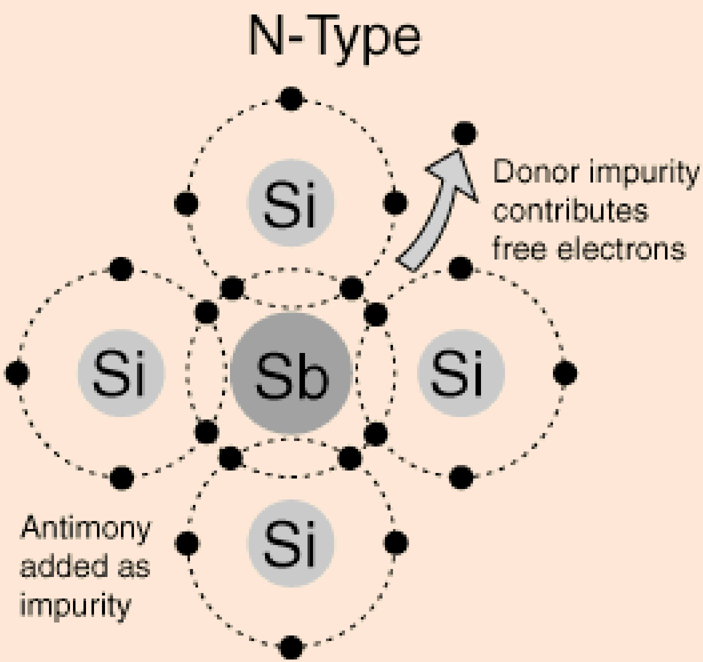
Semiconductor concepts

HyperPhysics***** Condensed Matter

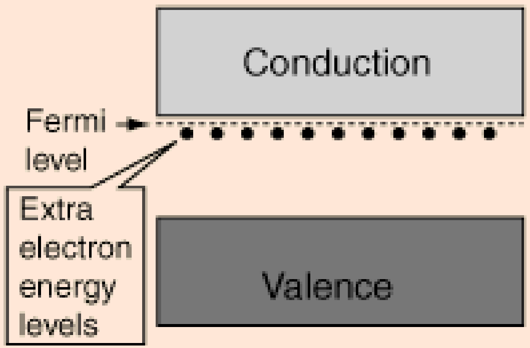
*R Nave*

Go Back

# N-Type Semiconductor




The addition of pentavalent impurities such as antimony, arsenic or phosphorous contributes free electrons, greatly increasing the conductivity of the intrinsic semiconductor. Phosphorous may be added by diffusion of phosphine gas (PH3).




Index

Semiconductor concepts

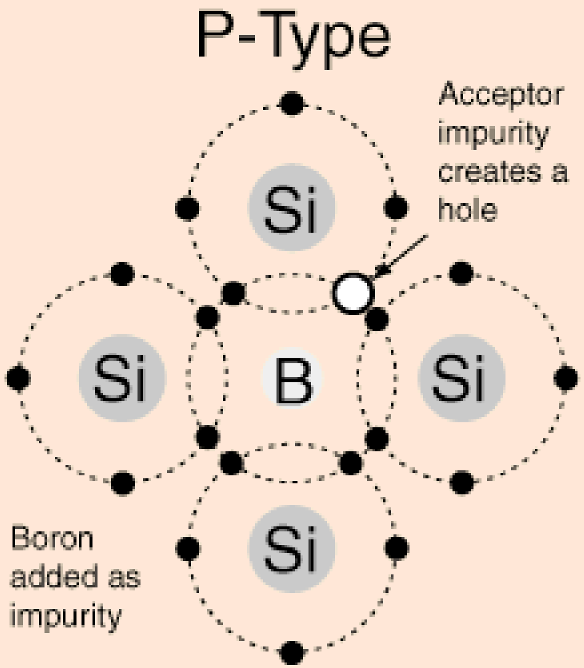
HyperPhysics***** Condensed Matter

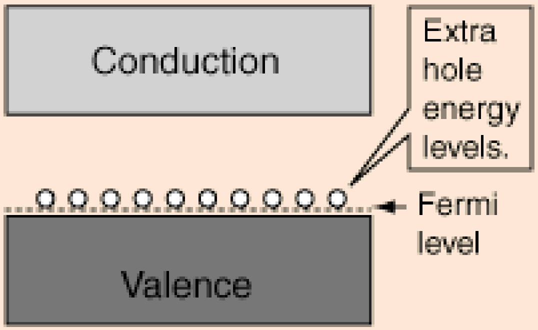
*R Nave*

Go Back

# P-Type Semiconductor

The addition of trivalent impurities such as boron, aluminum or gallium to an intrinsic semiconductor creates deficiencies of valence electrons,called "holes". It is typical to use $B_2H_6$ diborane gas to diffuse boron into the silicon material.




Index

Semiconductor concepts

HyperPhysics***** Condensed Matter

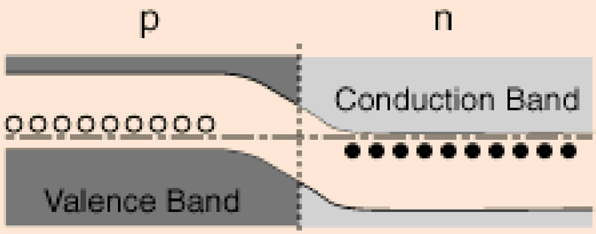
Go Back

*R Nave*

# P-N Junction

One of the crucial keys to solid state electronics is the nature of the P-N junction. When p-type and n-type materials are placed in contact with each other, the junction behaves very differently than either type of material alone. Specifically, current will flow readily in one direction (forward biased) but not in the other (reverse biased), creating the basic diode. This non-reversing behavior arises from the nature of the charge transport process in the two types of materials.

p-n junction



Energy bands at equilibrium

The open circles on the left side of the junction above represent "holes" or deficiencies of electrons in the lattice which can act like positive charge carriers. The solid circles on the right of the junction represent the available electrons from the n-type dopant. Near the junction, electrons diffuse across to combine with holes, creating a "depetion region". The energy level sketch above right is a way to visualize the equilibrium condition of the P-N junction. The upward direction in the diagram represents increasing electron energy.

Electron and hole conduction

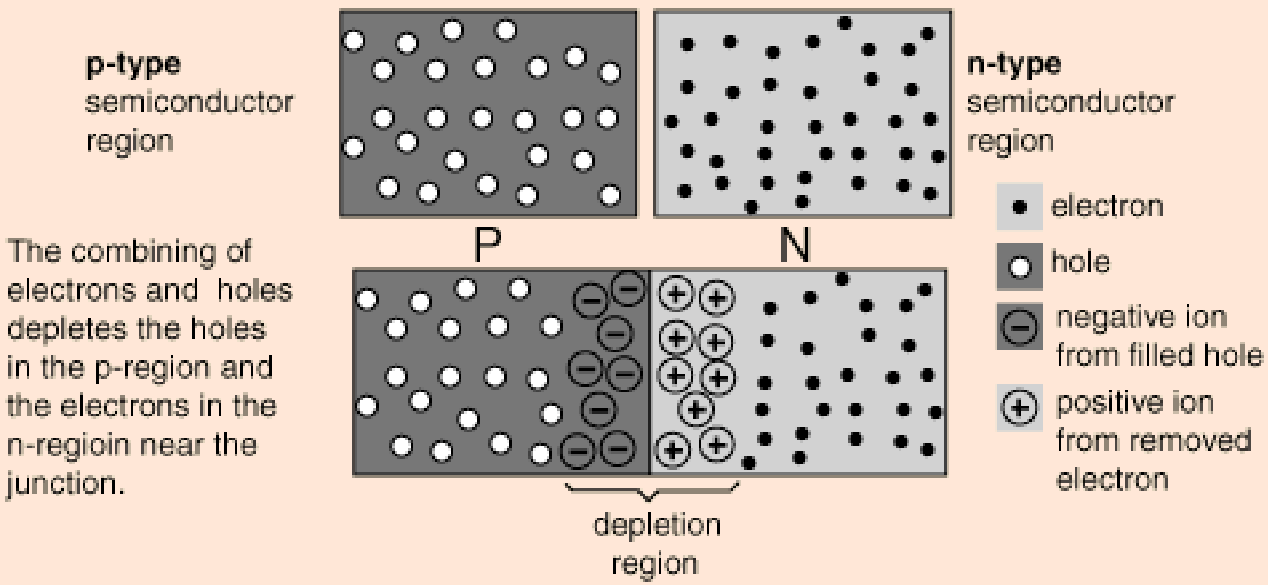
Index

Semiconductor concepts

Semiconductors for electronics

# Depletion Region

When a p-n junction is formed, some of the free electrons in the n-region diffuse across the junction and combine with holes to form negative ions. In so doing they leave behind positive ions at the donor impurity sites.



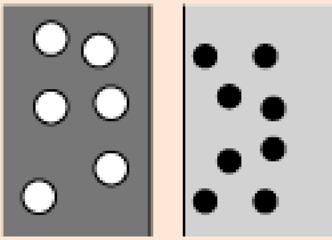

Show more detail of depletion region.

Index

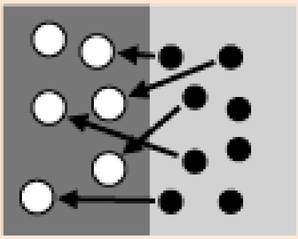
Semiconductor concepts

Semiconductors for electronics

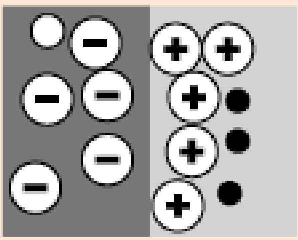
# Depletion Region Details



In the p-type region there are holes from the acceptor impurities and in the n-type region there are extra electrons.



When a p-n junction is formed, some of the electrons from the n-region which have reached the conduction band are free to diffuse across the junction and combine with holes.



Filling a hole makes a negative ion and leaves behind a positive ion on the n-side. A space charge builds up, creating a depletion region which inhibits any further electron transfer unless it is helped by putting a forward bias on the junction.

● Electron ○ Hole ⊖ Negative ion from filling of p-type vacancy. ⊕ Positive ion from removal of electron from n-type impurity.

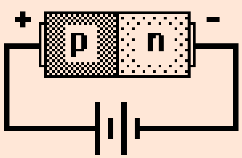
Show effects of biasing.

Index

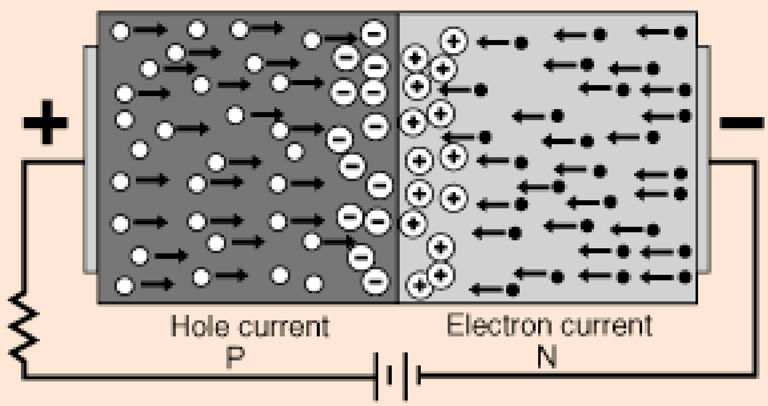
Semiconductor concepts

Semiconductors for electronics

# Forward Biased P-N Junction




Forward biasing the p-n junction drives holes to the junction from the p-type material and electrons to the junction from the n-type material. At the junction the electrons and holes combine so that a continuous current can be maintained.




Show energy bands. | Compare to reverse bias.


Index

Semiconductor concepts

Semiconductors for electronics

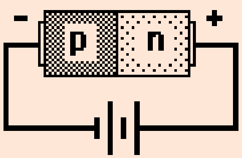


HyperPhysics***** Condensed Matter

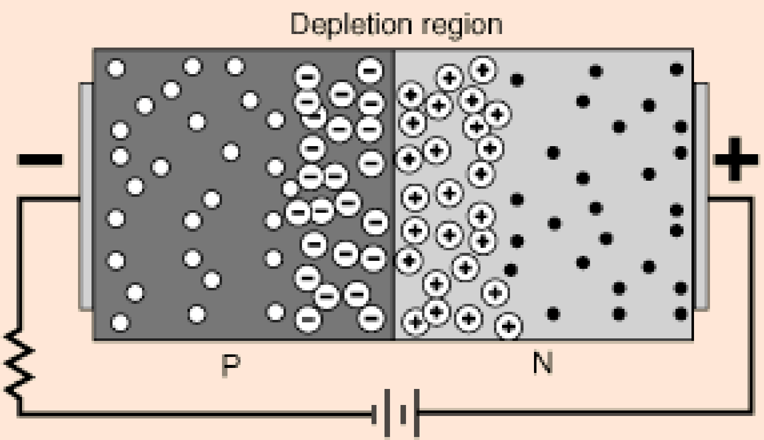

*R Nave*


Go Back

# Reverse Biased P-N Junction




The application of a reverse voltage to the p-n junction will cause a transient current to flow as both electrons and holes are pulled away from the junction. When the potential formed by the widened depletion layer equals the applied voltage, the current will cease except for the small thermal current.




Show energy bands. | Compare to forward bias.

Index

Semiconductor concepts

Semiconductors for electronics

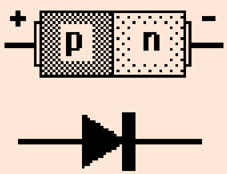
HyperPhysics***** Condensed Matter

*R Nave*

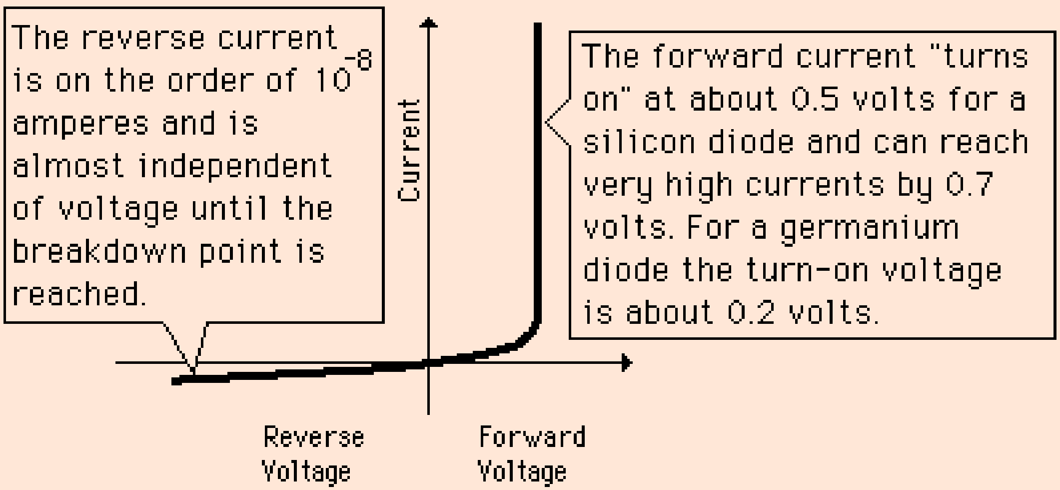
Go Back

Diodes

# The P-N Junction Diode




The nature of the p-n junction is that it will conduct current in the forward direction but not in the reverse direction. It is therefore a basic tool for rectification in the building of DC power supplies.





Diode varieties | PIN diode | Step-recovery diode | Diode applications

Index

Semiconductor concepts

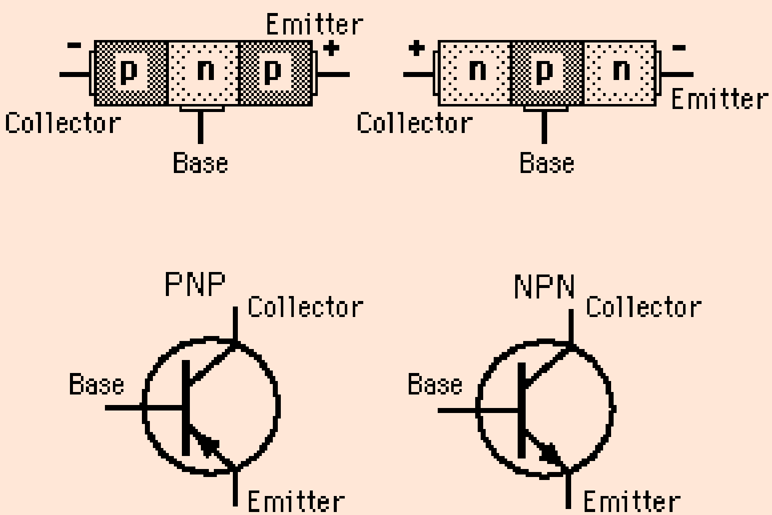
Semiconductors for electronics

HyperPhysics***** Condensed Matter

Go Back


*R Nave*

# The Junction Transistor




A bipolar junction transistor consists of three regions of doped semiconductors. A small current in the center or base region can be used to control a larger current flowing between the end regions (emitter and collector). The device can be characterized as a current amplifier, having many applications for amplification and switching.

| | |
|---|---|
| Constraints on operation | Transistor operating conditions |
| Varieties of Transistors | Details about conduction in transistors |
| Determining collector current | Details about base-emitter junction |

Index

Semiconductor concepts

Semiconductors for electronics

Electronics concepts

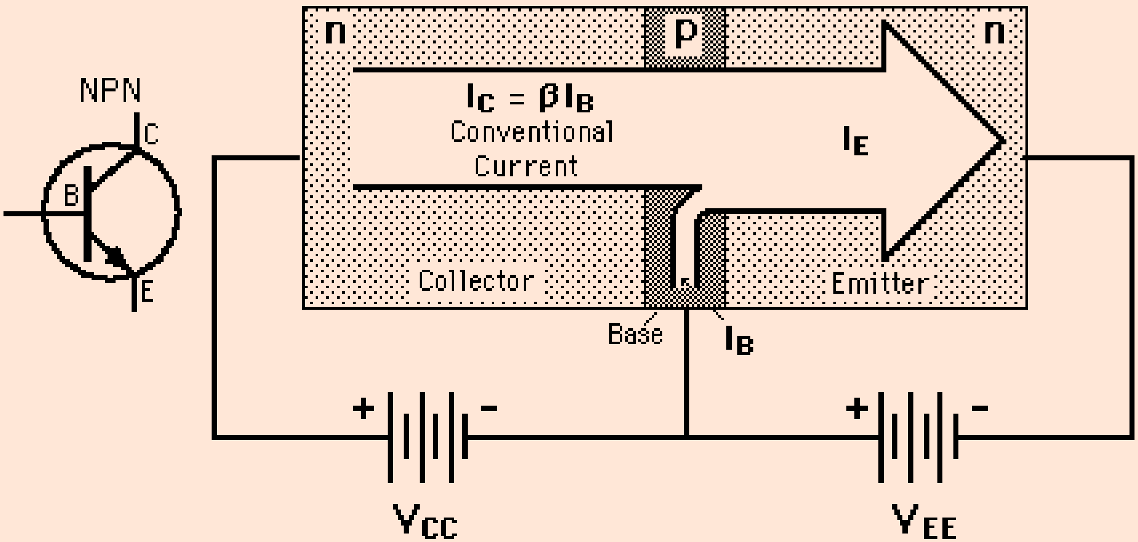
HyperPhysics***** Condensed Matter

*R Nave*

Go Back

# Transistor as Current Amplifier

The larger collector current $I_C$ is proportional to the base current $I_B$ according to the relationship $I_C = \beta I_B$ , or more precisely it is proportional to the base-emitter voltage $V_{BE}$ . The smaller base current controls the larger collector current, achieving current amplification.




Constraints on operation | Comments on structure | To pnp version

Index

Semiconductor concepts

Semiconductors for electronics

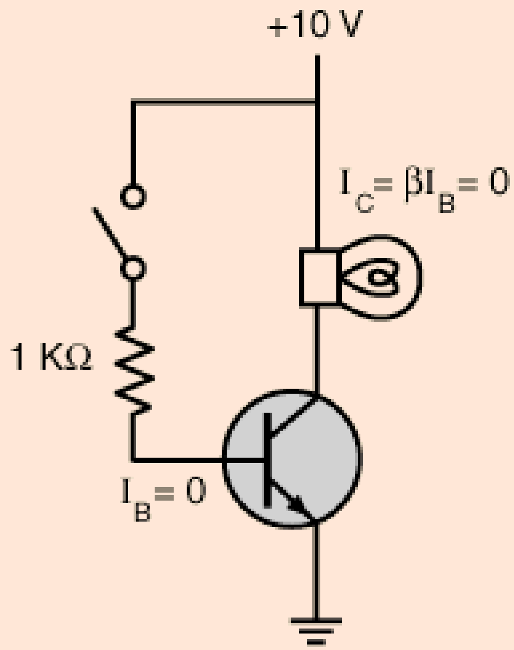
Electronics concepts

Reference
Diefenderfer / Holton
p156

# Transistor Switch Example

The switch is open.

Close the switch




There is no current to the base, so the transistor is in the cut off condition with no collector current. All the voltage drop is accoss the transistor.

Transistor operation for switch conditions

Transistor Switches

Index

Electronics concepts

Digital Electronics

Reference Horowitz & Hill p52

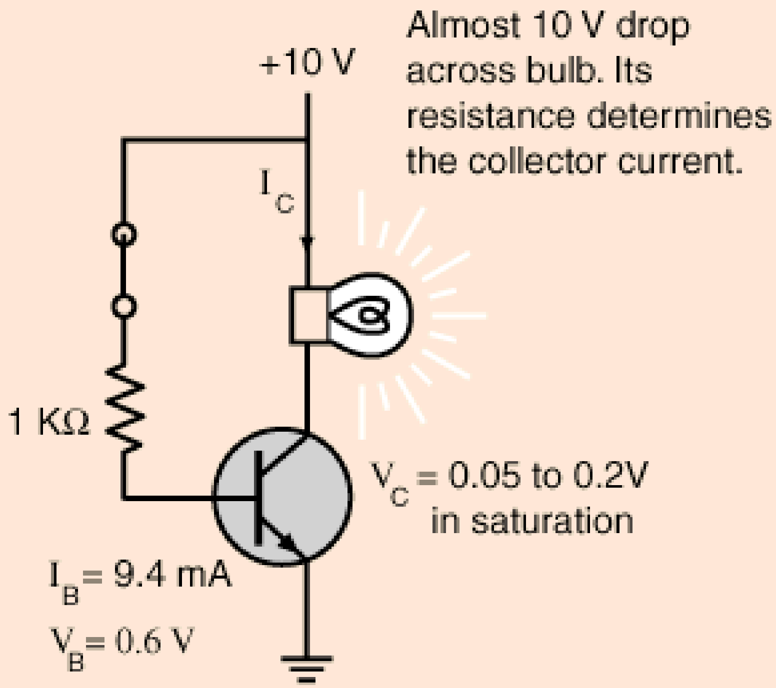
# Transistor Switch Example

The switch is closed.

Open the switch

The base resistor is chosen small enough so that the base current drives the transistor into saturation.

In this example the mechanical switch is used to produce the base current to close the transistor switch to show the principles. In practice, any voltage on the base sufficient to drive the transistor to saturation will close the switch and light the bulb.




Transistor operation for switch conditions

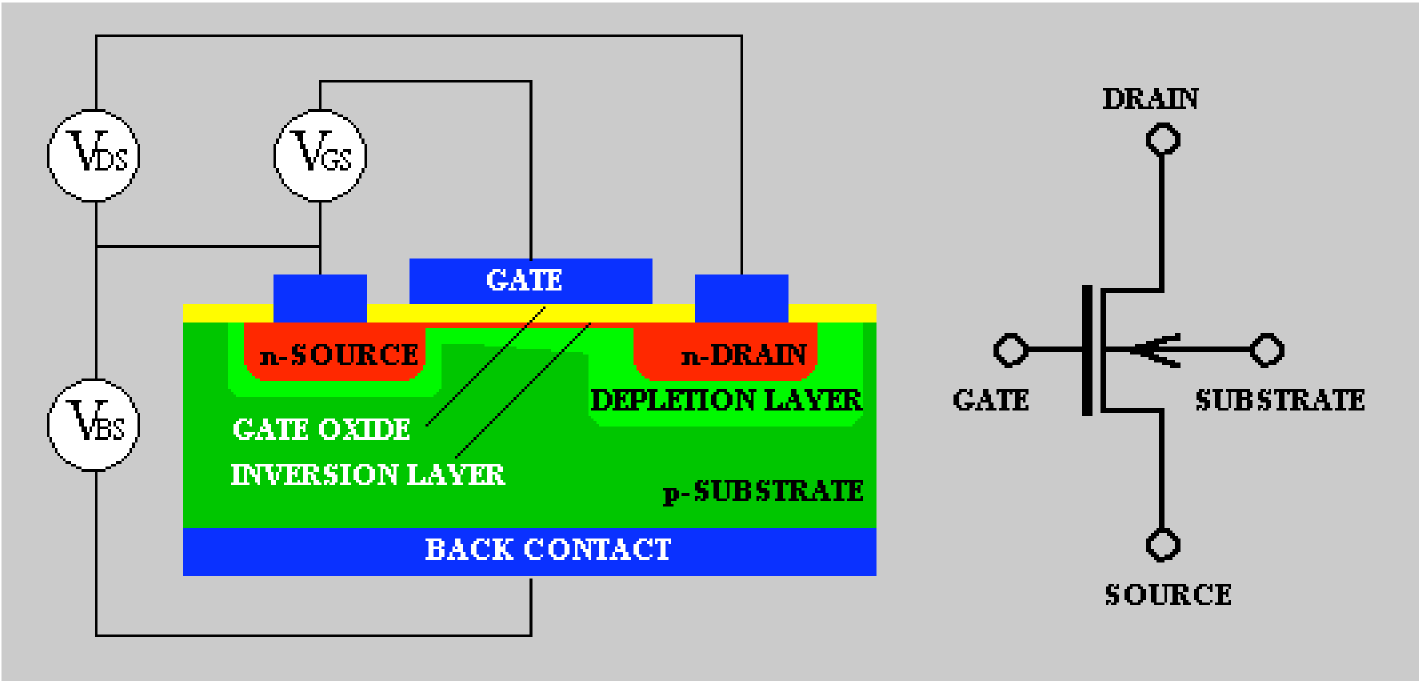
Transistor Switches


Index

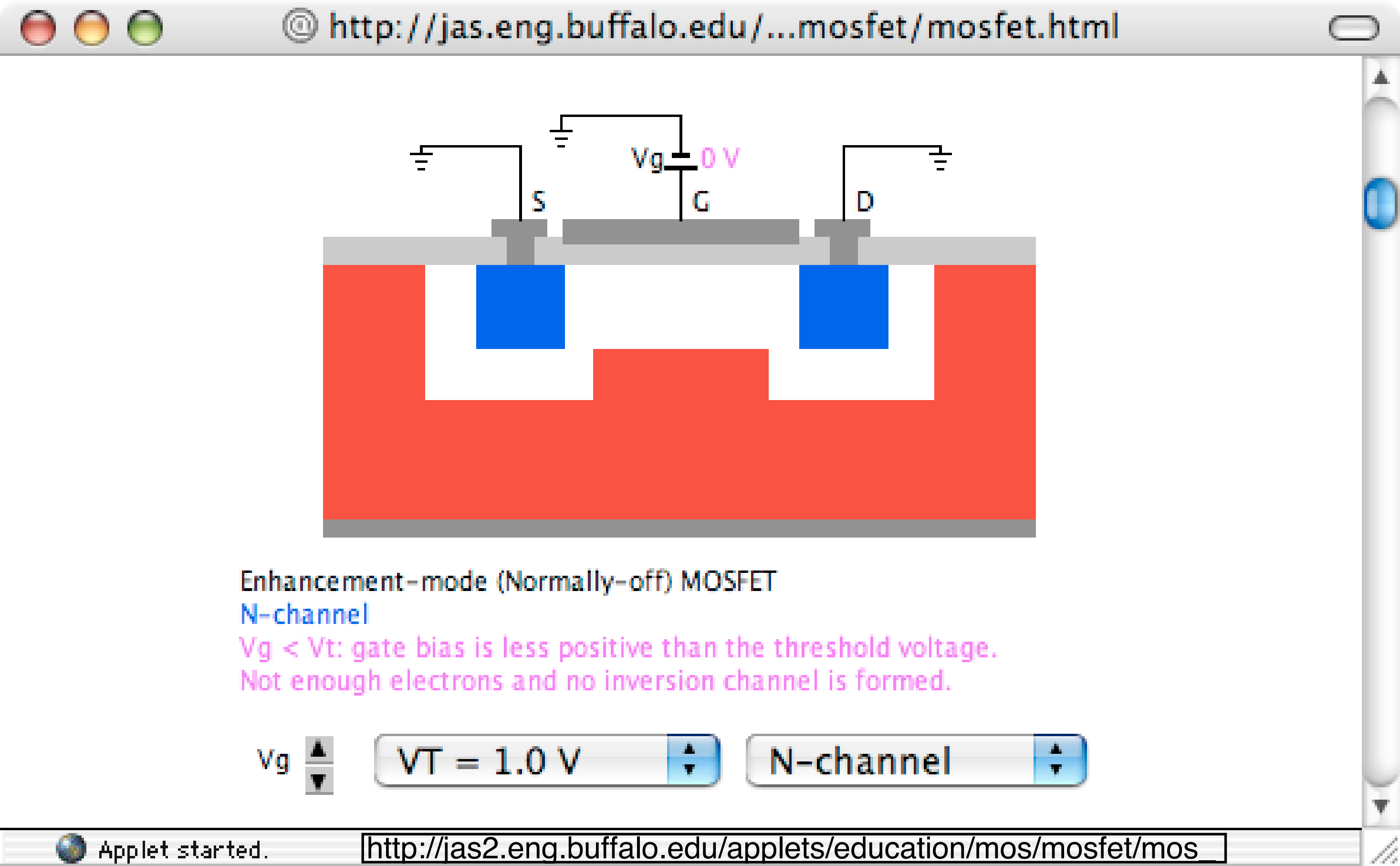
Electronics concepts

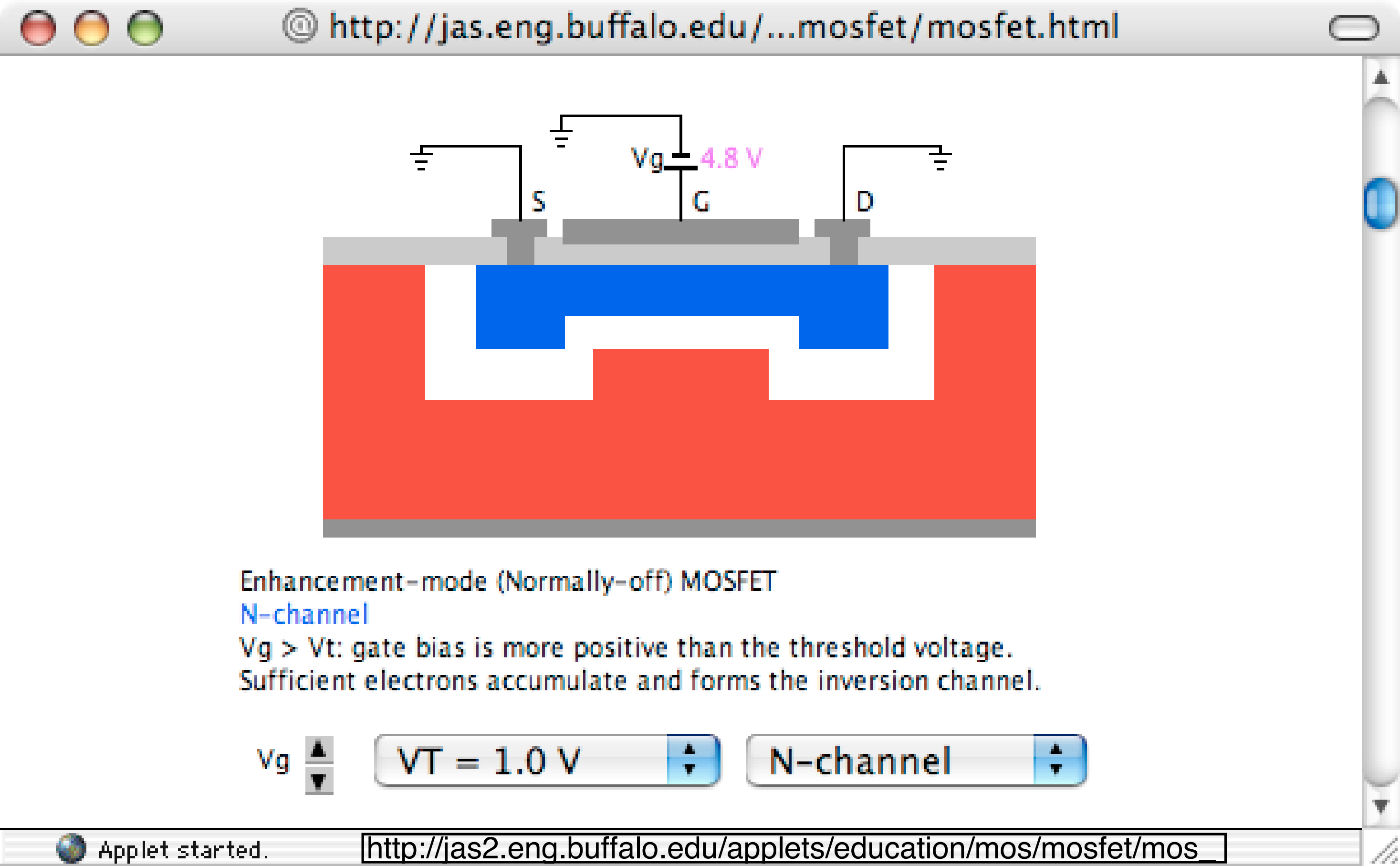
Digital Electronics


Reference Horowitz & Hill p52



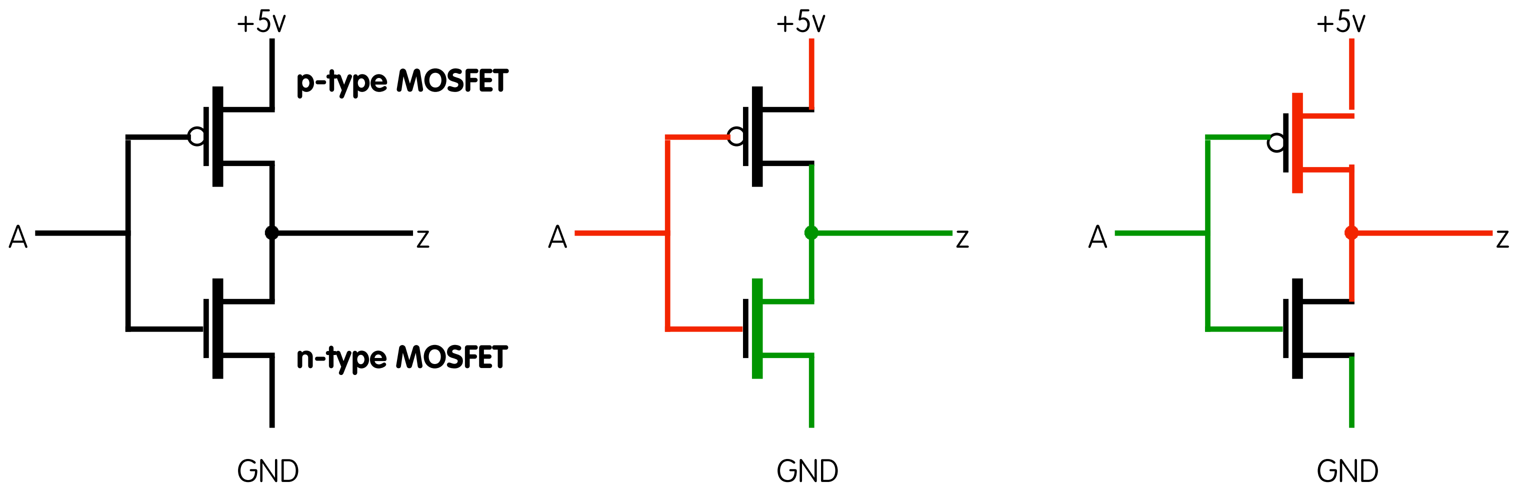

from: http://ece-www.colorado.edu/~bart/book/mosintro.htm

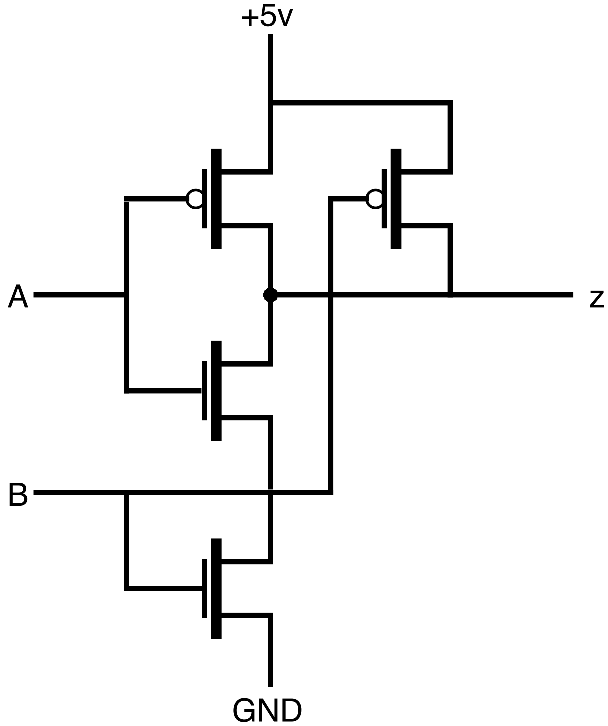

http://jas.eng.buffalo.edu/...mosfet/mosfet.html
Vg 0 V
S
G
D
Enhancement-mode (Normally-off) MOSFET
N-channel
Vg < Vt: gate bias is less positive than the threshold voltage.
Not enough electrons and no inversion channel is formed.
Vg
VT = 1.0 V
N-channel
Applet started.
http://jas2.eng.buffalo.edu/applets/education/mos/mosfet/mos_

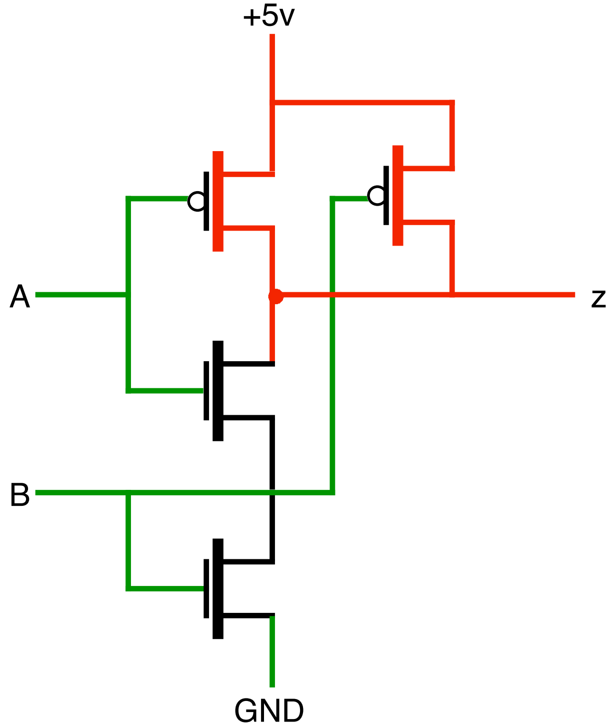

http://jas.eng.buffalo.edu/...mosfet/mosfet.html
Vg
4.8 V
S
G
D
Enhancement-mode (Normally-off) MOSFET
N-channel
Vg > Vt: gate bias is more positive than the threshold voltage.
Sufficient electrons accumulate and forms the inversion channel.
Vg
VT = 1.0 V
N-channel
Applet started.
http://jas2.eng.buffalo.edu/applets/education/mos/mosfet/mos_

# An Inverter using MOSFET

- CMOS = complementary metal oxide semiconductor
- P-type transistor conducts when gate is low
- N-type transistor conducts when gate is high

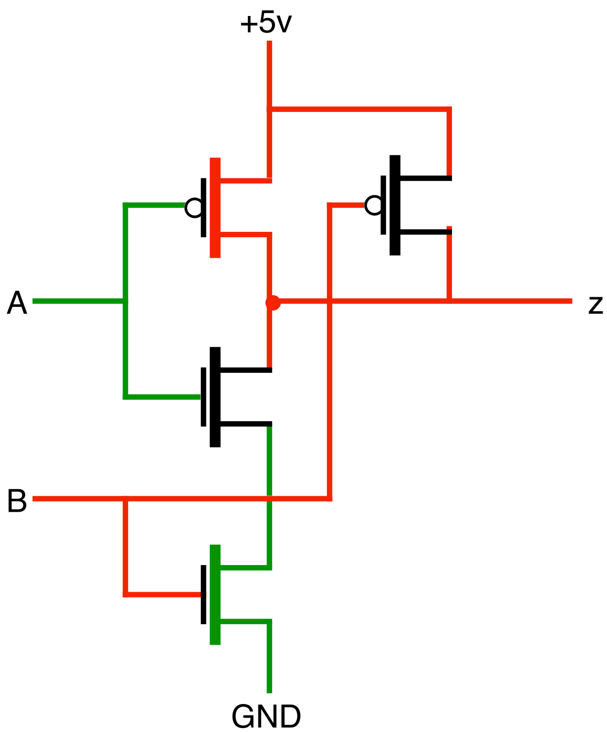


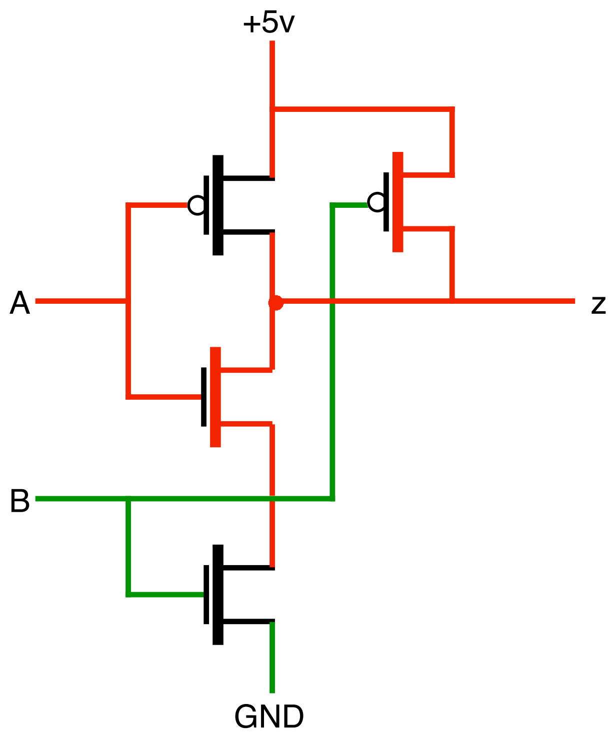


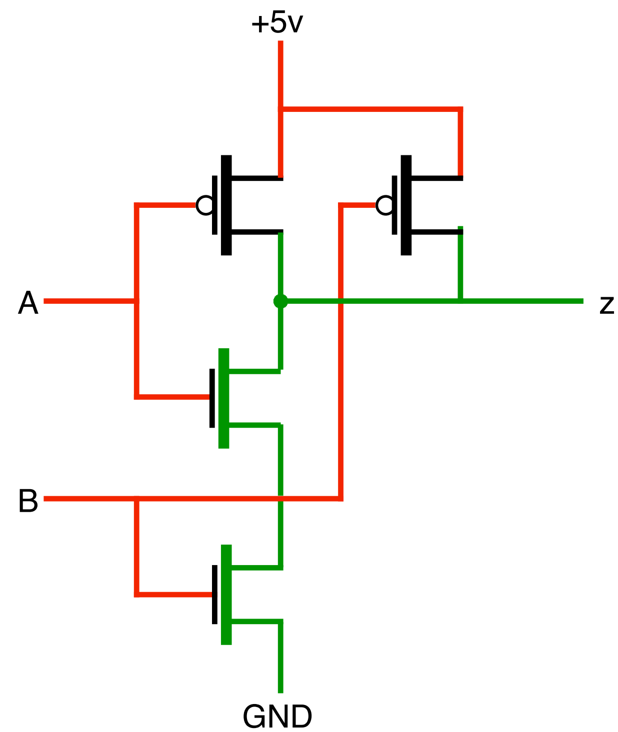



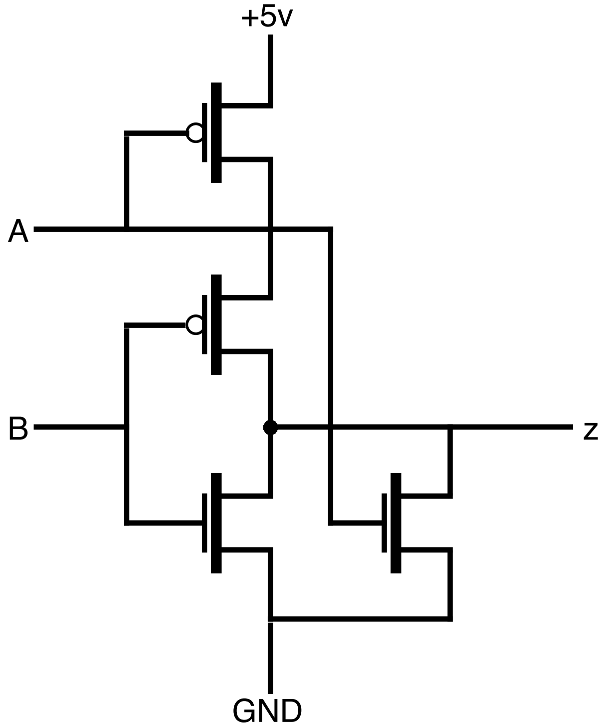

## NAND GATE

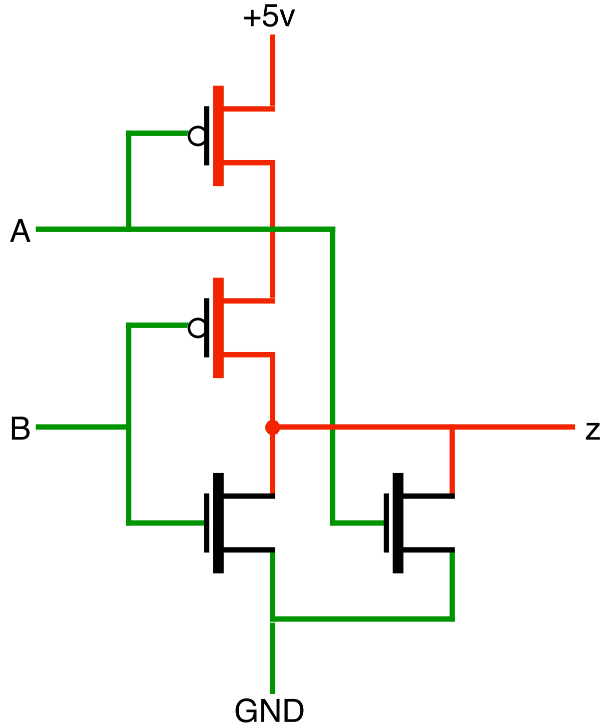
| A | B | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



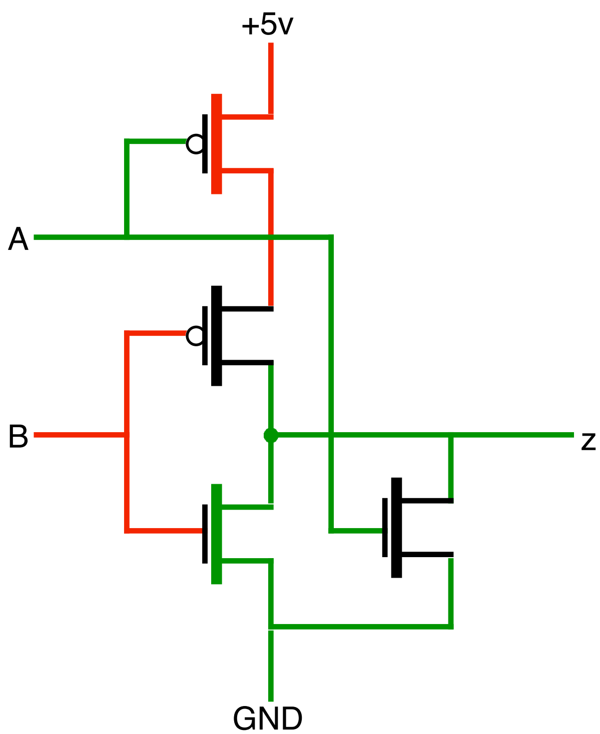



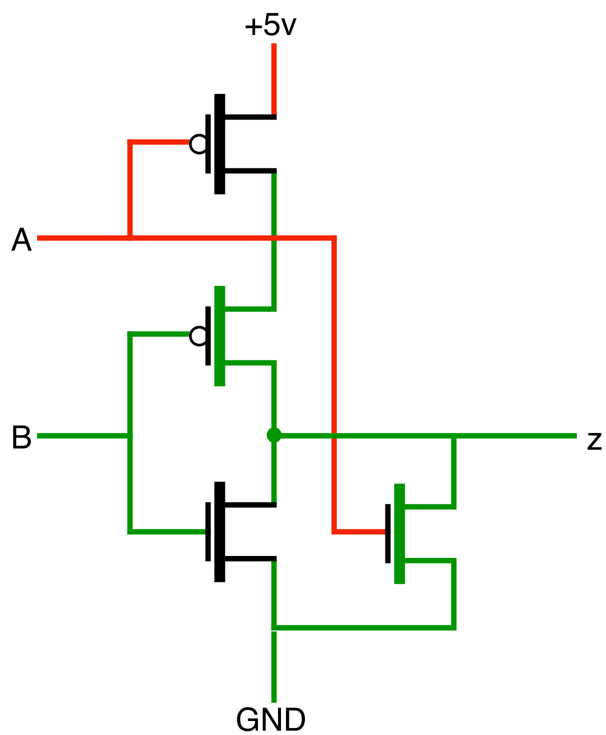


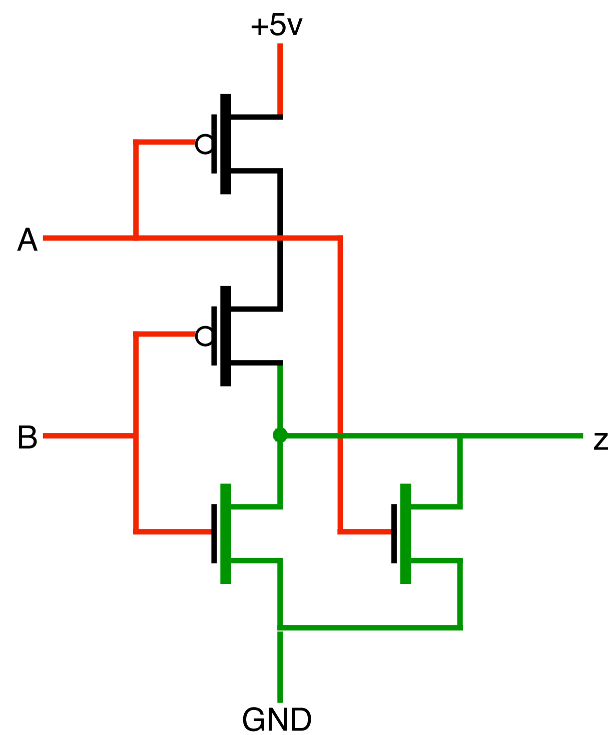




## NOR GATE

| A | B | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



# CMOS Logic vs Bipolar Logic

- MOSFET transistors are easier to miniaturize
- CMOS logic has lower current drain
- CMOS logic is easier to manufacture

# References

- Materials on semiconductors, PN junction and transistors taken from the HyperPhysics web site:

  <http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html>