EE40
Lecture 14
Josh Hug

7/26/2010

Logisticals

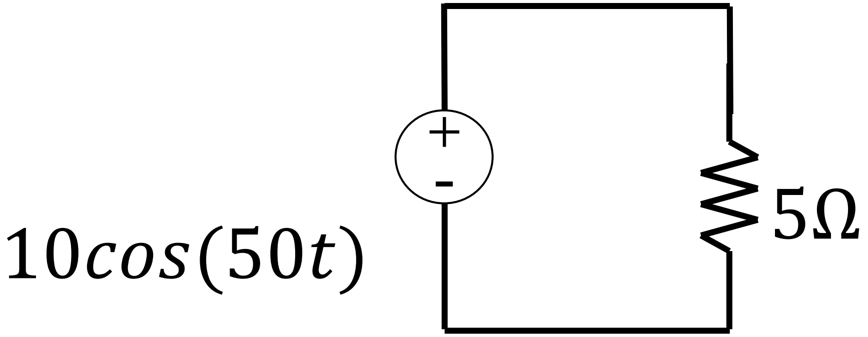
- Midterm Wednesday
  - Study guide online
  - Study room on Monday
    - Cory 531, 2:00
      - Cooper, Tony, and I will be there 3:00-5:10
  - Study room on Tuesday
    - Cory 521, 2:30 and on
- Completed homeworks that have not been picked up have been moved into the lab cabinet
- If you have custom Project 2 parts, I've emailed you with details about how to pick them up

Lab

- Lab will be open on Tuesday if you want to work on Project 2 or the Booster Lab or something else
  - Not required to start Project 2 tomorrow
- No lab on Wednesday (won’t be open)

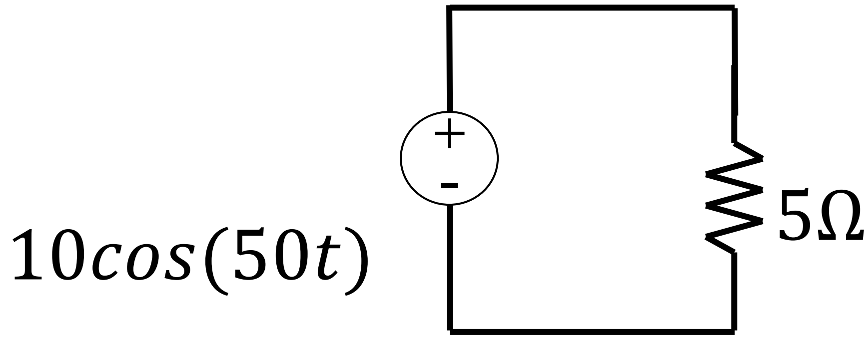
Power in AC Circuits

- One last thing to discuss for Unit 2 is power in AC circuits
- Let's start by considering the power dissipated in a resistor:

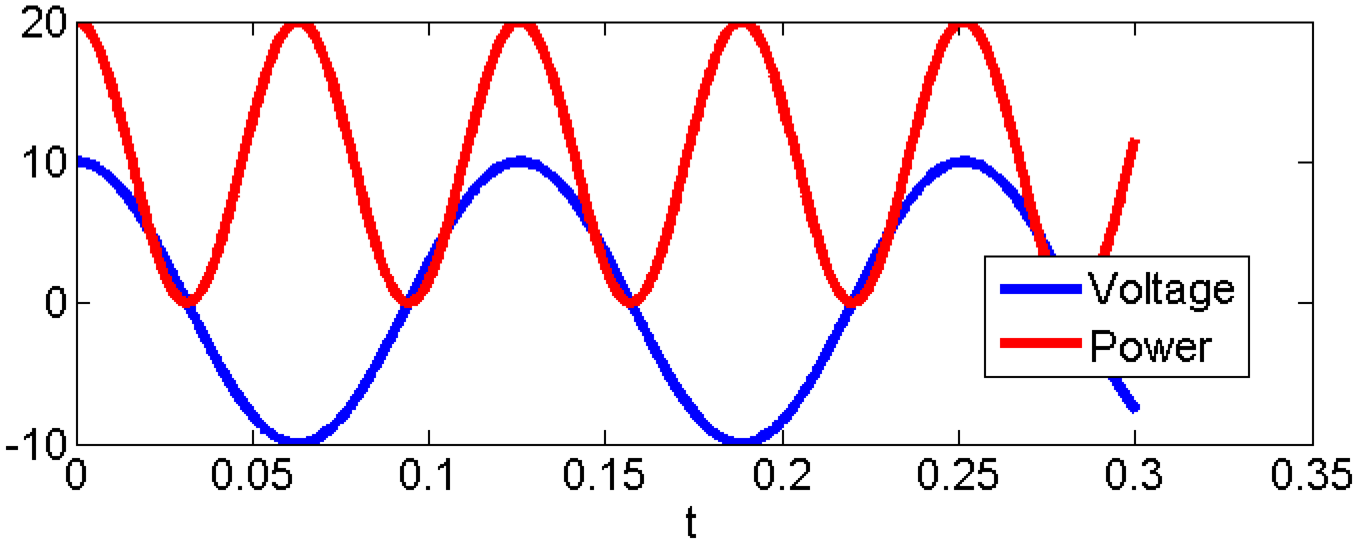


$$P(t) = v(t)i(t) = 10\cos(50t) \times \frac{10}{5}\cos(50t)$$
$$= 20\cos^2(50t)$$

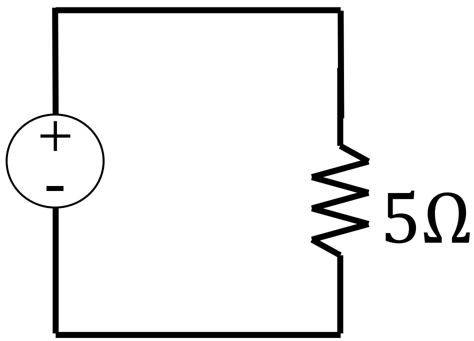
Or graphically



$$P(t) = v(t)i(t) = 10\cos(50t) \times \frac{10}{5}\cos(50t)$$
$$= 20\cos^2(50t)$$



Average Power

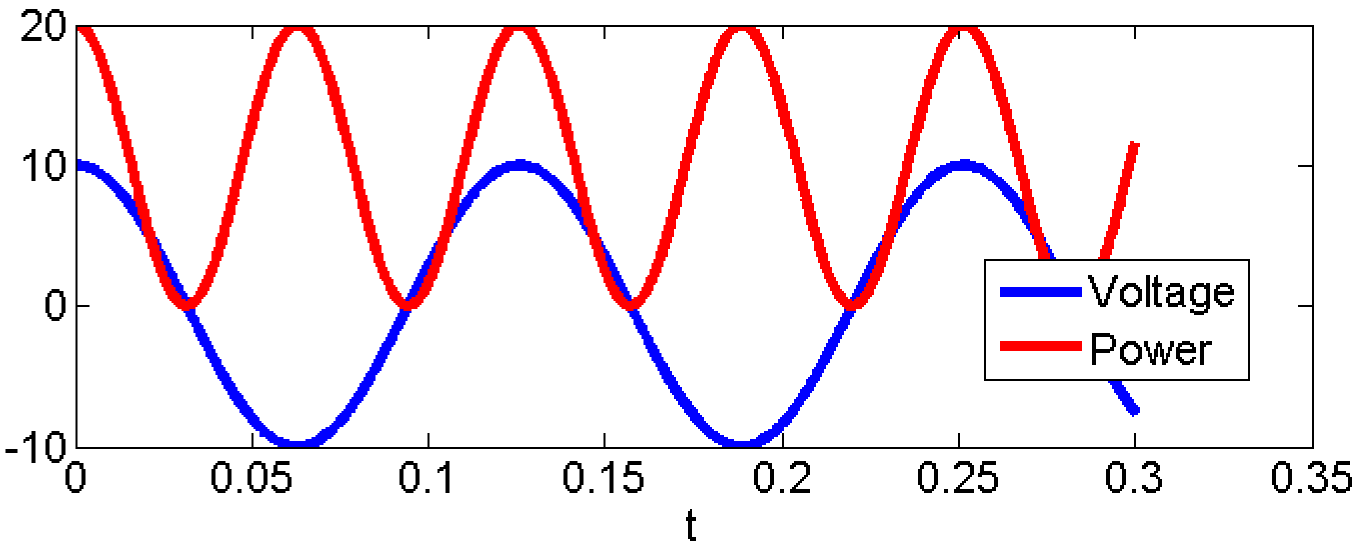


$10cos(50t)$

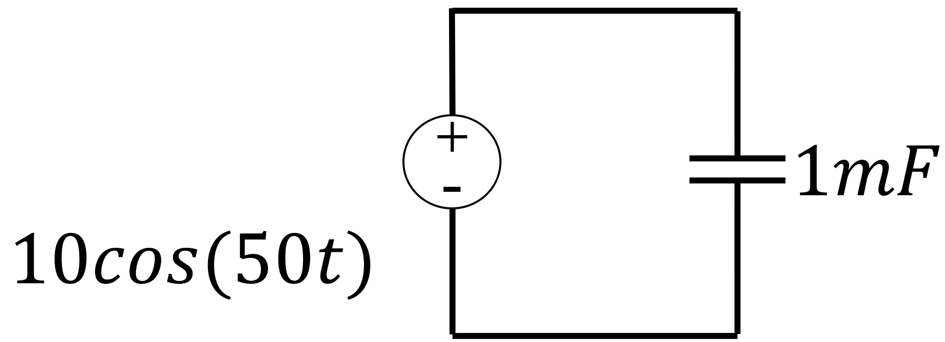
Peak Power: 20W

Min Power: 0W

Avg Power: 10W



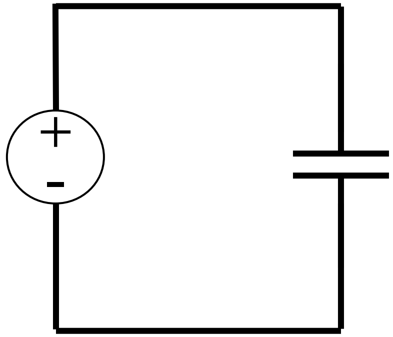
Capacitor example



Find p(t)

- $i(t) = 10^{-3} \times (-500\sin(50t)) = -0.5\sin(50t)$
- $p(t) = -5\sin(50t)\cos(50t)$

Graphically



$10cos(50t)$

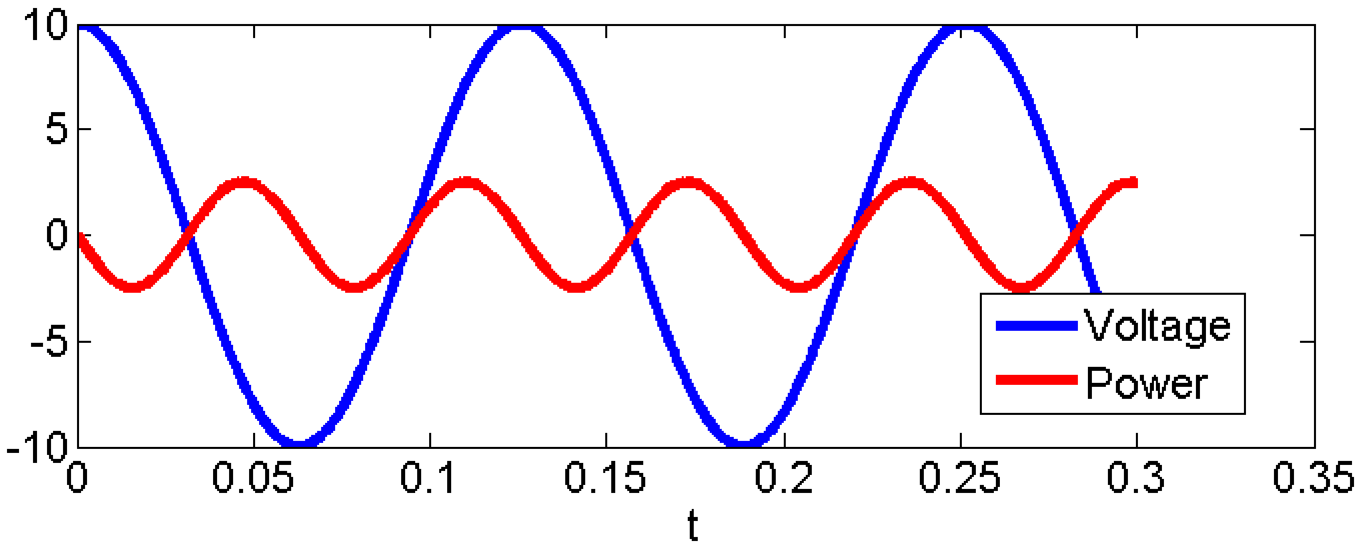
$1mF$

Peak Power: $5/2W$

Min Power: $-5/2W$

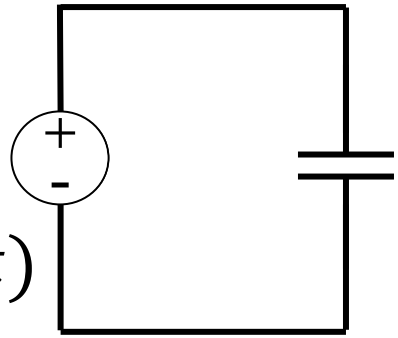
Avg Power: 0W

$$p(t) = -5\sin(50t)\cos(50t)$$



Is there some easier way of calculating power?

- Like maybe with… phasors?



$1mF$

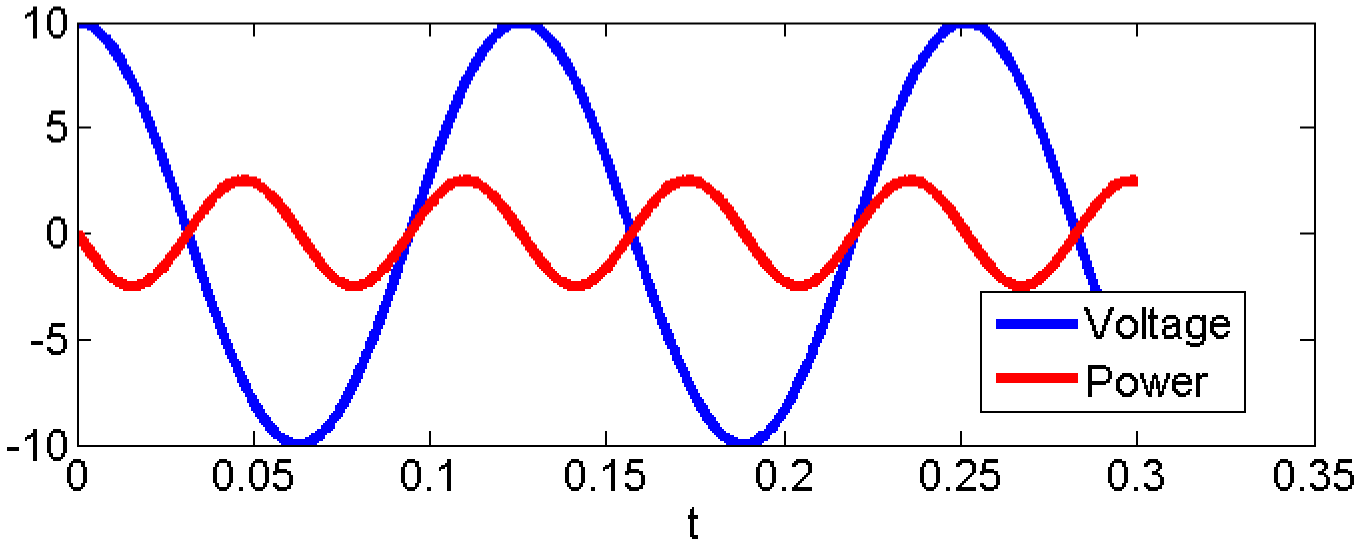
$10cos(50t)$

$$i(t) = 0.5\cos\left(50t + \frac{\pi}{2}\right)$$

$$p(t) = 5\cos(50t)\cos(50t + \frac{\pi}{2})$$

- Phasors are:
  - $\hat{V} = 10\angle 0$
  - $\hat{I} = 0.5\angle\frac{\pi}{2}$
- How about $\hat{P} = \hat{V}\hat{I}$?

Is there some easier way of measuring power?

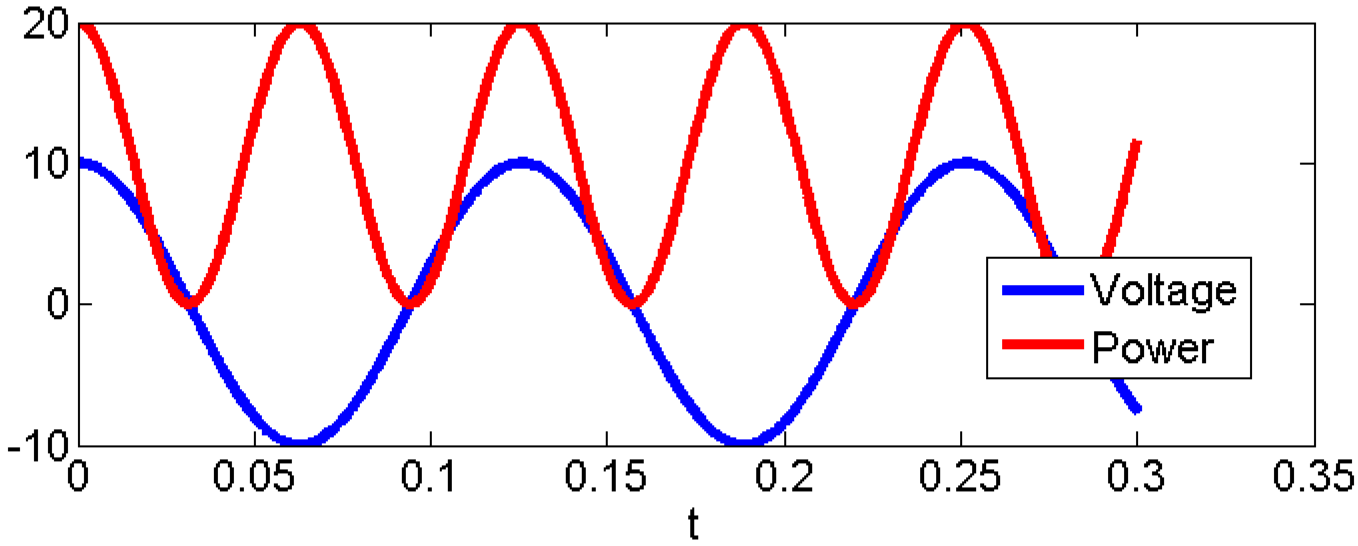


$$p(t) = 5\cos(50t)\cos(50t + \frac{\pi}{2})$$

- Phasors are:
  - $\hat{V} = 10\angle 0$
  - $\hat{I} = 0.5\angle\frac{\pi}{2}$
- Does $\hat{P} = \hat{V}\hat{I}$?
  - $\hat{V}\hat{I} = 5\angle\frac{\pi}{2}$

A. Yes, $\hat{V}\hat{I}$ matches p(t)
B. No, wrong magnitude
C. No, wrong phase
D. No, wrong frequency

It gets worse



- For the resistor, there is no phasor which represents the power (never goes negative)

Average Power

- Tracking the time function of power with some sort of phasor-like quantity is annoying
  - Frequency changes
  - Sometimes have an offset (e.g. with resistor)
- Often, the thing we care about is the average power, useful for e.g.
  - Battery drain
  - Heat dissipation
- Useful to define a measure of “average” other than the handwavy thing we did before
- Average power given periodic power is:

$$\bar{p} = \int_0^T \frac{1}{T} p(t) dt$$

T is time for 1 period

Power in terms of phasors

- We've seen that we cannot use phasors to find an expression for p(t)
- Average power given periodic power is:

$$\bar{p} = \int_0^T \frac{1}{T} p(t) dt$$

T is time for 1 period

- We'll use this definition of average power to derive an expression for average power in terms of phasors

Average Power

- $\overline{p(t)} = \overline{v(t)i(t)}$
- Note: $\bar{a}\bar{b} \neq \overline{ab}$
  - e.g. $a = 5\cos(t)\,, b = 4\cos(t)$
    - Average of each cosine is zero
    - Average of their product is 10
- Our goal will be to get the average power $\overline{p(t)}$ from phasors $\hat{V}$ and $\hat{I}$
- We'll utilize $\overline{Re[a]Re[b]} = \frac{1}{2}Re[ab^*]$
  - * denotes complex conjugate
  - See extra slides for proof of this identity

Power from Phasors

- $\overline{Re[a]Re[b]} = \frac{1}{2}Re[ab^*]$
- $\overline{p(t)} = \overline{v(t)i(t)}$
- $v(t) = Re\left[V_o e^{j(\omega t+\theta)}\right]$
- $i(t) = Re\left[I_o e^{j(\omega t+\phi)}\right]$
- $\overline{p(t)} = \overline{Re\left[Re[V_o e^{j(\omega t+\theta)}]\right] Re\left[Re[I_o e^{j(\omega t+\phi)}]\right]}$

$$= \frac{1}{2}Re\left[V_o e^{j(\omega t+\theta)} I_o e^{-j(\omega t+\phi)}\right]$$

$$= \frac{1}{2}Re[V_o I_o e^{j(\theta-\phi)}]$$

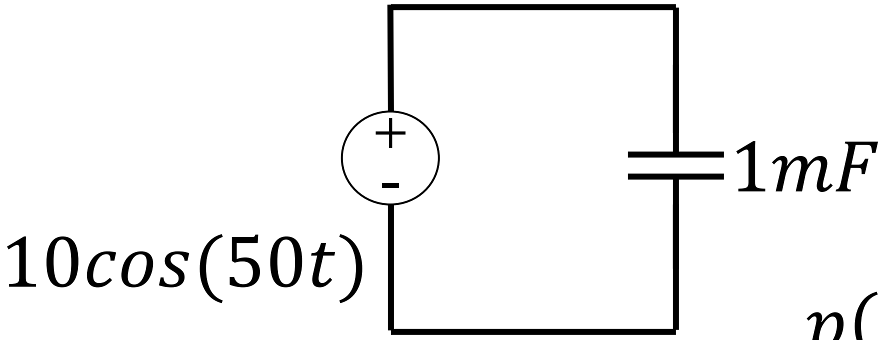
$$= \frac{1}{2}Re[\hat{V}\hat{I}^*]$$

Power from Phasors

- Thus, given a voltage phasor $\hat{V}$ and a current phasor $\hat{I}$, the average power absorbed is

$$p(t) = \frac{1}{2} Re\left[\hat{V}\hat{I}^*\right]$$

Capacitor Example



$$i(t) = -0.5\cos\left(50t + \frac{\pi}{2}\right)$$

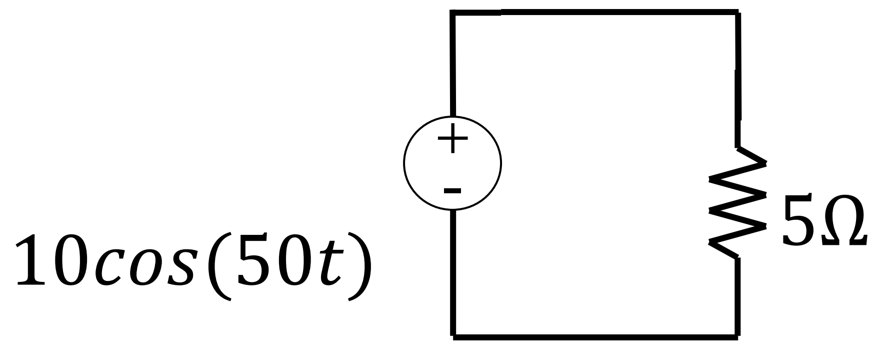
$$p(t) = 5\cos(50t)\cos(50t + \frac{\pi}{2})$$

- Phasors are:
  - $\hat{V}(t) = 10\angle 0$
  - $\hat{I}(t) = 0.5\angle \frac{\pi}{2}$
- $\overline{p(t)} = \frac{1}{2} Re\left[10\angle 0 \times 0.5\angle -\frac{\pi}{2}\right]$

$$= \frac{1}{2} Re\left[5\angle -\frac{\pi}{2}\right] = 0$$

Resistor Example

$$p(t) = \frac{1}{2} Re\left[\hat{V}\hat{I}^*\right]$$

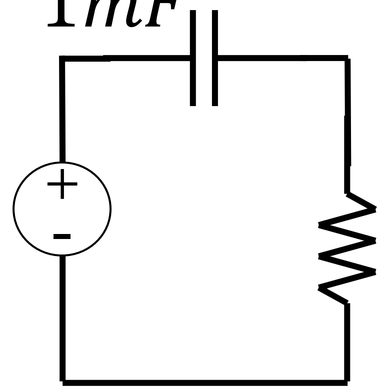


- $\hat{V} = 10\angle 0$
- $\hat{I} = 2\angle 0$
- $\overline{\left(p(t)\right)} = \frac{1}{2} Re[20] = 10$

Find avg power across resistor

A. 0 Watts
B. 10 Watts
C. 20 Watts

Resistor Example



Find avg power from source

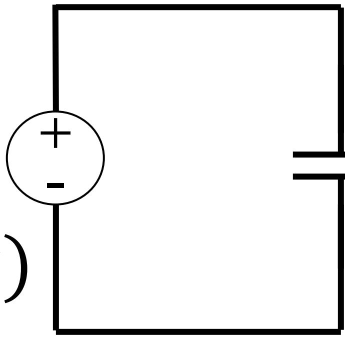
- $Z_{eq} = 5 - 20j$
- $\hat{V} = 10\angle 0$
- $\hat{I} = \hat{V}/Z_{eq}$
- $\overline{(p(t))} = \frac{1}{2}Re[\hat{V}\hat{I}^*] = \frac{1}{2}Re\left[\frac{\hat{V}\hat{V}^*}{Z_{eq}^*}\right] = \frac{1}{2}Re\left[\frac{100}{5+20j}\right]$

$$= \frac{1}{2}Re\left[\frac{100}{20.6155\angle 1.3258}\right] = \frac{1}{2}Re[1.17 - 4.7j] = 0.58W$$

Reactive Power

- So if power dissipated is $\frac{1}{2}Re\left[\hat{V}\hat{I}^*\right]$, then what is $\frac{1}{2}Im\left[\hat{V}\hat{I}^*\right]$?
- Imaginary part is called "reactive power"
- Physical intuition is that it's power that you put into an element with memory, but which the element eventually gives back

Capacitor Reactive Power Example



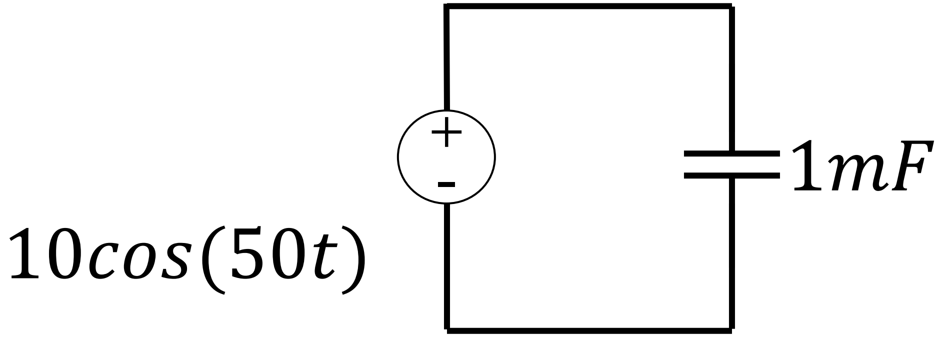
$$i(t) = -0.5\cos\left(50t + \frac{\pi}{2}\right)$$

$$p(t) = 5\cos(50t)\cos(50t + \frac{\pi}{2})$$

- Phasors are:
  - $\hat{V}(t) = 10\angle 0$
  - $\hat{I}(t) = 0.5\angle\frac{\pi}{2}$
- $\overline{p(t)} = \frac{1}{2} Im\left[10\angle 0 \times 0.5\angle-\frac{\pi}{2}\right]$

$$= \frac{1}{2} Im\left[5\angle-\frac{\pi}{2}\right] = -\frac{5}{2}W$$

Graphically

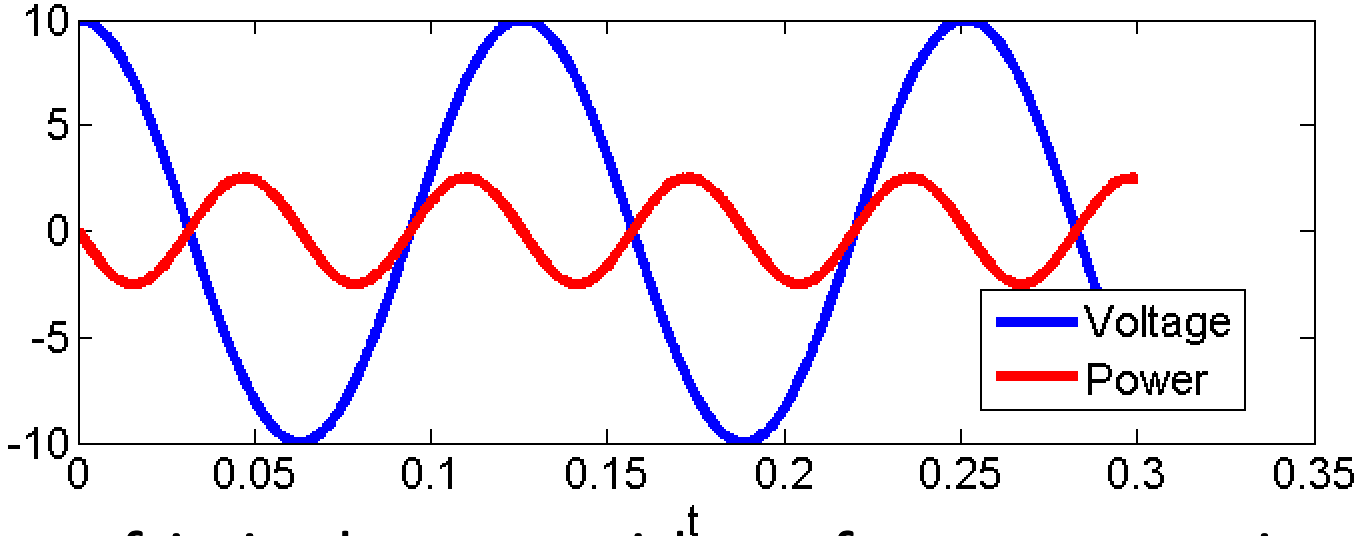


Peak Power: $5/2W$

Min Power: $-5/2W$

Avg Power: 0W

Avg Reactive Power: -5/2W



Like a frictionless car with perfect regenerative brakes, starting and stopping again and again and again

Note on Reactive Power

- “Providing” reactive power and “consuming” reactive power are physically the same thing
- Usually we say capacitors “provide” reactive power, which comes from our definition, whereas inductors “consume” reactive power

$$p_{reactive} = \frac{1}{2} Im\left[\hat{V}\hat{I}^*\right]?$$

- As you’ll see on HW7, capacitors and inductors can be chosen to get rid of reactive power

And that rounds out Unit 2

- We’ve covered all that needs to be covered on capacitors and inductors, so it’s time to (continue) moving on to the next big thing

Back to Unit 3 – Integrated Circuits

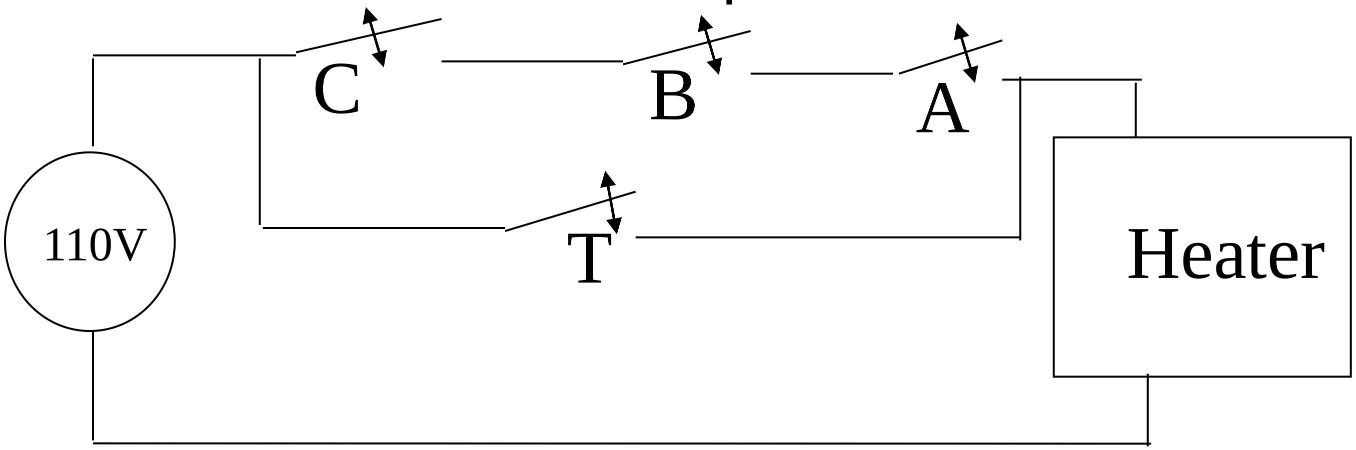
- Last Friday, we started talking about integrated circuits
- Analog integrated circuits
  - Behave mostly like our discrete circuits in lab, can reuse old analysis
- Digital integrated circuits
  - We haven't discussed discrete digital circuits, so in order to understand digital ICs, we will first have to do a bunch of new definitions

Digital Representations of Logical Functions

- Digital signals offer an easy way to perform logical functions, using Boolean algebra
- Example: Hot tub controller with the following algorithm
  - Turn on heating element if
    - A: Temperature is less than desired (T < Tset)
    - **and** B: The motor is on
    - **and** C: The hot tub key is turned to "on"
      - **OR**
    - T: Test heater button is pressed

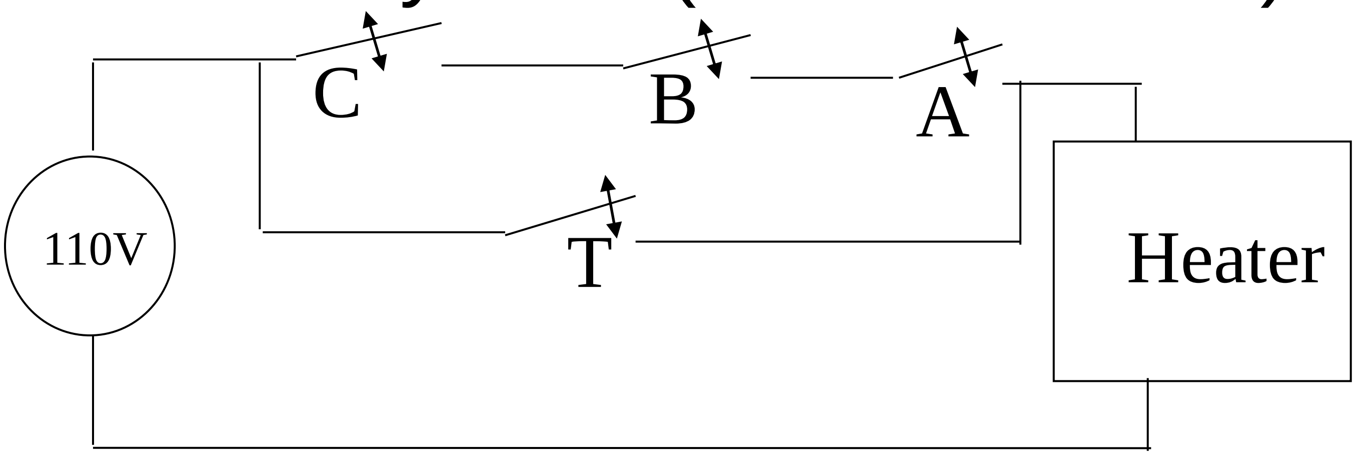
Hot Tub Controller Example

- Example: Hot tub controller with the following algorithm
  - Turn on heating element if
    - A: Temperature is less than desired (T < Tset)
    - **and** B: The motor is on
    - **and** C: The hot tub key is turned to "on"
      - **OR**
    - T: Test heater button is pressed



Hot Tub Controller Example

- Example: Hot tub controller with the following algorithm
  - A: Temperature is less than desired (T < Tset)
  - B: The motor is on
  - C: The hot tub key is turned to "on"
  - T: Test heater button is pressed
- Or more briefly: ON=(**A** and **B** and **C**) or **T**



Boolean Algebra and Truth Tables

- We'll next formalize some useful mathematical expressions for dealing with logical functions
- These will be useful in understanding the function of digital circuits

Boolean Logic Functions

- Example: ON=(**A** and **B** and **C**) or **T**
- Boolean logic functions are like algebraic equations
  - Domain of variables is 0 and 1
  - Operations are "AND", "OR", and "NOT"
- In contrast to our usual algebra on real numbers
  - Domain of variables is the real numbers
  - Operations are addition, multiplication, exponentiation, etc

Examples

- In normal algebra, we can have
  - 3+5=8
  - A+B=C
- In Boolean algebra, we'll have
  - 1 and 0=0
  - A and B=C

Have you seen boolean algebra before?

- A. Yes
- B. No

Formal Definitions

- “not” is a unary operator (takes 1 argument)
- Returns 1 if its argument is 0, and 0 if its argument is 1, e.g.
  - not 0=1
- There exist many shorthand ways of writing the not operation e.g.

$$\bar{0} = 1$$

$$0' = 1$$

$$\neg 0 = 1$$

- I will use bar notation for consistency with the book.

Formal Definitions

- "and" is a binary operator [takes 2 arguments] which returns 1 if both if its arguments are 1, and 0 otherwise
- Many ways to write "A and B" in shorthand:

$$AB$$

$$A \cdot B$$

$$A \wedge B$$

- As a table, if $Z = AB$, then:

A	B	Z
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

Formal Definitions

- “or” is a binary operator [takes 2 arguments] which returns 0 if either of its arguments are 1, and 0 otherwise
- Common ways to write “A and B” in shorthand:

$$A + B$$

$$A \vee B$$

- As a table, if $Z = A + B$, then:

A	B	Z
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Boolean Algebra and Truth Tables

- Just as in normal algebra, boolean algebra operations can be applied recursively, giving rise to complex boolean functions
- Z=AB+C
- Any boolean function can be represented by one of these tables, called a **truth table**

A	B	C	Z
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Boolean Algebra

- Originally developed by George Boole as a way to write logical propositions as equations
- Now, a very handy tool for specification and simplification of logical systems

Simplification Example

- Z: Shine the bat signal
- C: Crime in progress
- B: Want to meet Batman
- T: Test bat signal
- $Z = C + \bar{C}B + \bar{C}\bar{B}T$

- Simpler expression:
  - $Z = C + B + T$

C	B	T	Z
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Logic Simplification

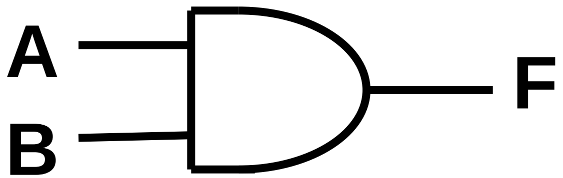
- In CS61C and optionally CS150, you will learn a more thorough systematic way to simplify logic expression
- All digital arithmetic can be expressed in terms of logical functions
- Logic simplification is crucial to making such functions efficient
- You will also learn how to make logical adders, multipliers, and all the other good stuff inside of CPUs

Quick Arithmetic-as-Logic Example

- Assuming we have boolean input variables A_1, A_2, B_2, B_2 and boolean output variables Z_1, Z_2
- Let's say that each variable represent one digit of a binary number, we have 16 possibilities

Logic Gates

- Logic gates are the schematic equivalent of our boolean logic functions
- Example, the AND gate:

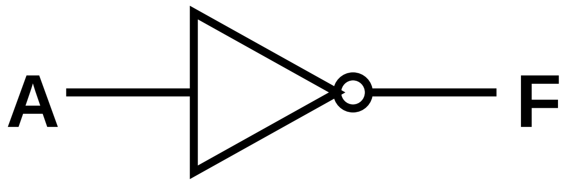
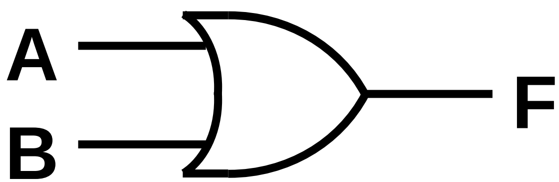
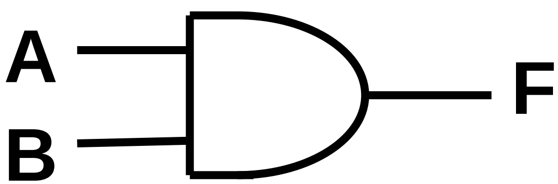


$F = A \cdot B$

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

- If we're thinking about real circuits, this is a device where the output voltage is high if and only if both of the input voltages are high

Logic Functions, Symbols, & Notation

NAME	SYMBOL	NOTATION	TRUTH TABLE
"NOT"		$F = \overline{A}$	see below
"OR"		F = A+B	see below
"AND"		F = A•B	see below

"NOT" truth table:

A	F
0	1
1	0

"OR" truth table:

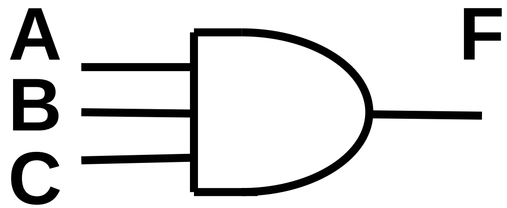
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

"AND" truth table:

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

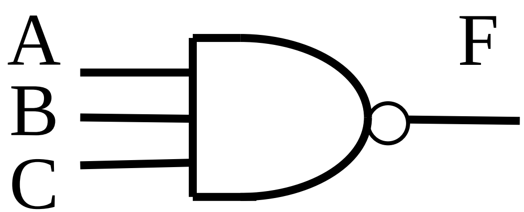
Multi Input Gates

- AND and OR gates can also have many inputs, e.g.



F = ABC

- Can also define new gates which are composites of basic boolean operations, for example NAND:



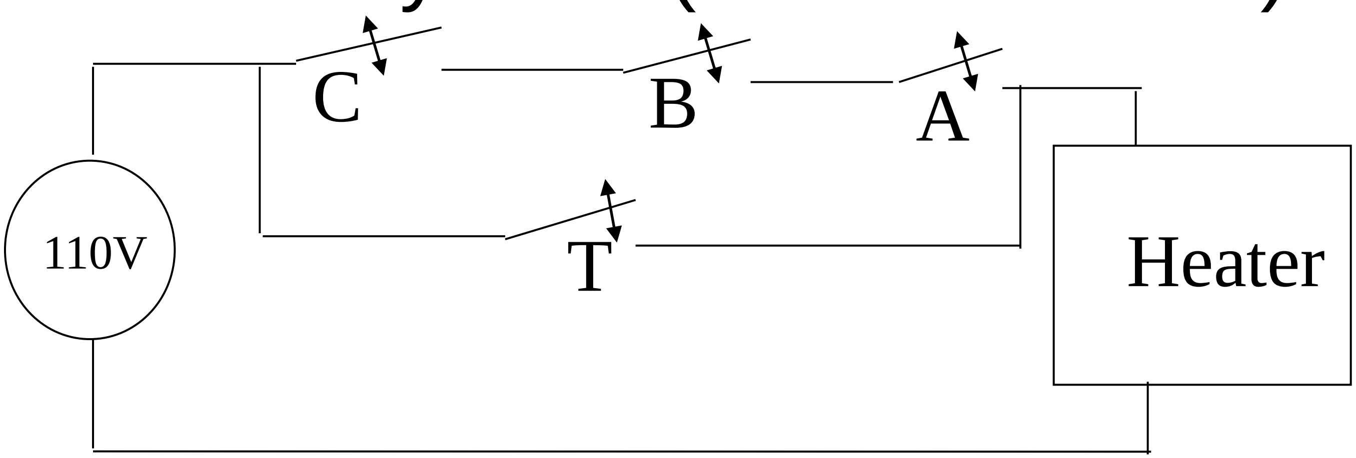
$F = \overline{ABC}$

Logic Gates

- Can think of logic gates as a technology independent way of representing logical circuits
- The exact voltages that we'll get will depend on what types of components we use to implement our gates
- Useful when designing logical systems
  - Better to think in terms of logical operations instead of circuit elements and all the accompanying messy math

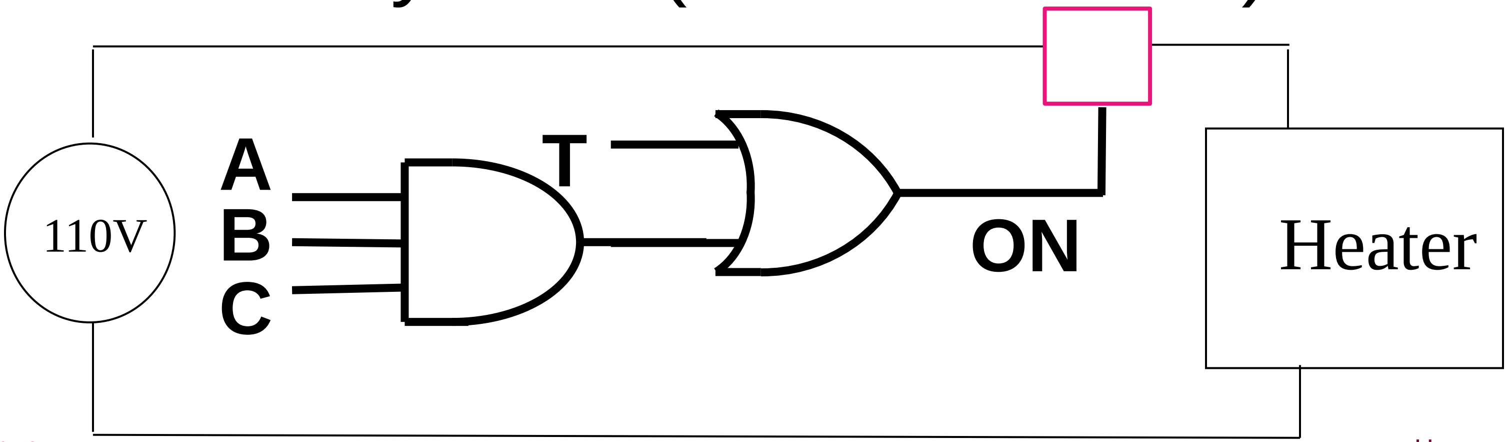
Hot Tub Controller Example

- Example: Hot tub controller with the following algorithm
  - A: Temperature is less than desired (T < Tset)
  - B: The motor is on
  - C: The hot tub key is turned to "on"
  - T: Test heater button is pressed
- Or more briefly: ON=(**A** and **B** and **C**) or **T**



Hot Tub Controller Example

- Example: Hot tub controller with the following algorithm
  - A: Temperature is less than desired (T < Tset)
  - B: The motor is on
  - C: The hot tub key is turned to “on”
  - T: Test heater button is pressed
- Or more briefly: ON=(A and B and C) or T



How does this all relate to circuits?

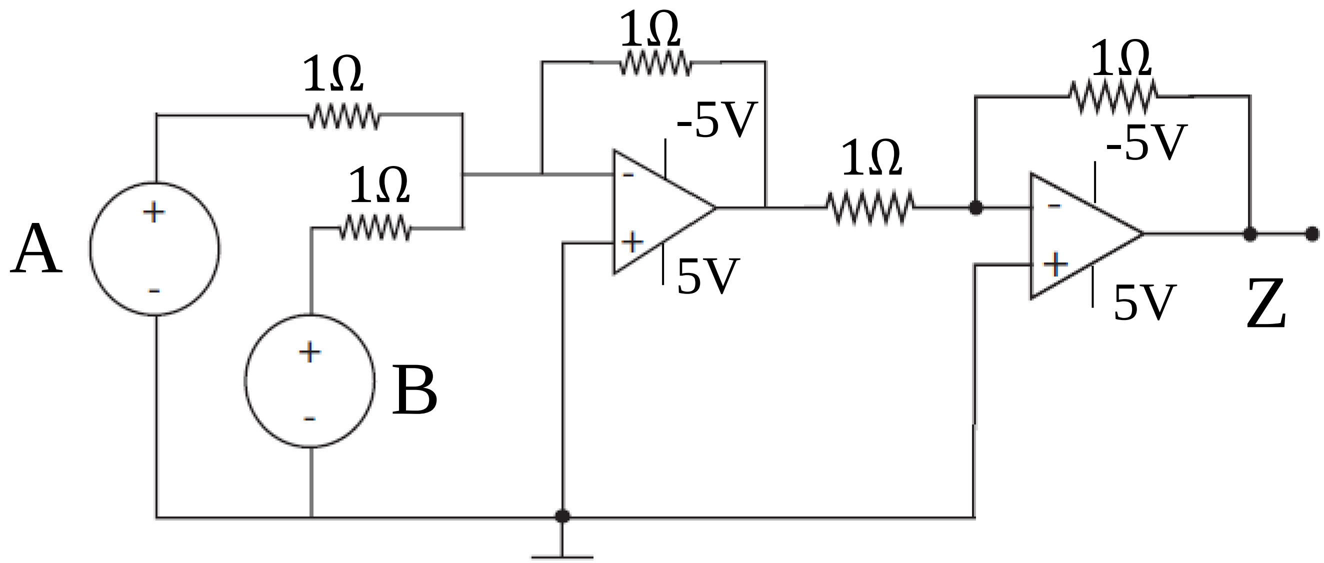
- A digital circuit is simply any circuit where every voltage in the circuit is one of two values
  - V_{low} (typically ground) will represent boolean 0
  - V_{high} (in modern CPUs, approximately 1V, though you can set this on your computer) will represent boolean 1
- In truth, of course, values will vary continuously, but entire design is conceptualized as simply 1s and 0s

The "Static Discipline"

- We can think of the whole circuit as obeying a contract to always provide output voltages V_{low} and V_{high} at all outputs as long as the inputs follow these same rules
- Up to the circuit designer to ensure this specification is met
- In truth, voltages may be a little lower or higher than these contractual values
- However, as long as the output values are close enough, the deviations are unimportant

Many Possible Ways to Realize Logic Gates

- There are many ways to build logic gates, for example, we can build gates with op-amps



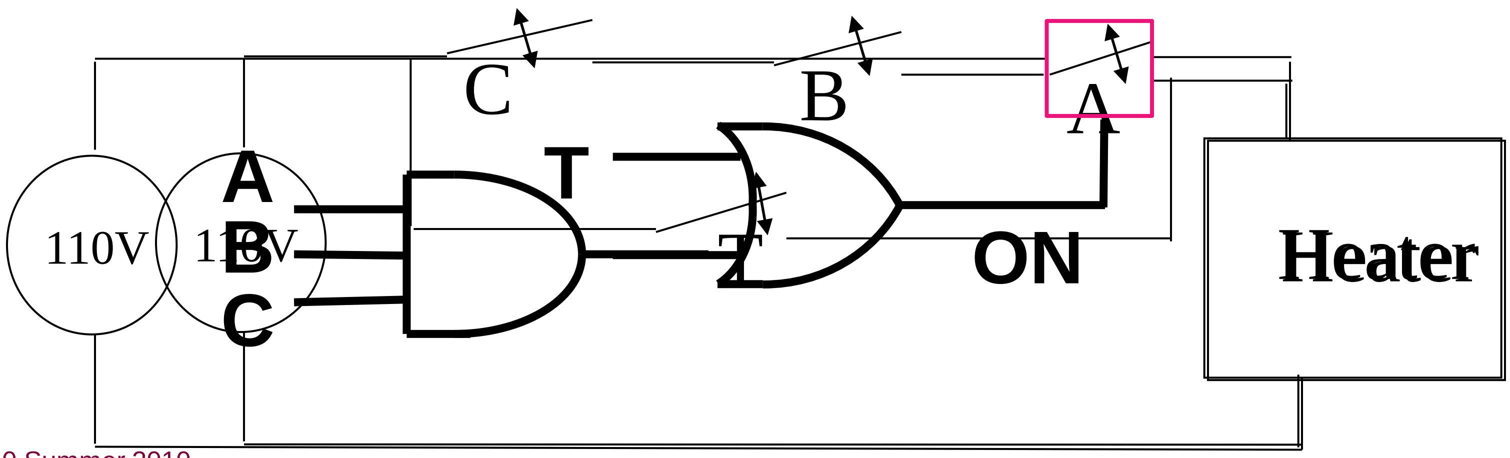
$$Z = f(A, B)$$

- Far from optimal
  - 5 resistors
  - Dozens of transistors

- Is this a(n):
  A. AND gate
  B. OR gate
  C. NOT gate
  D. Something else

Switches as Gates

- Example: Hot tub controller
- ON=(**A** and **B** and **C**) or **T**
- Switches are the most natural implementation for logic gates

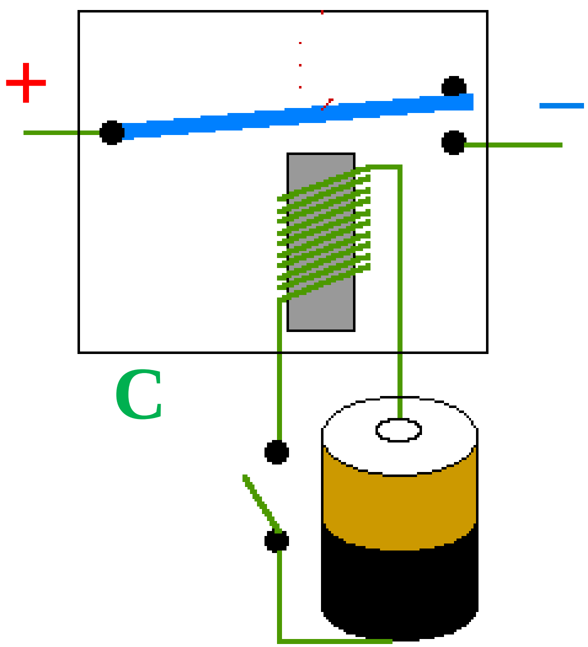


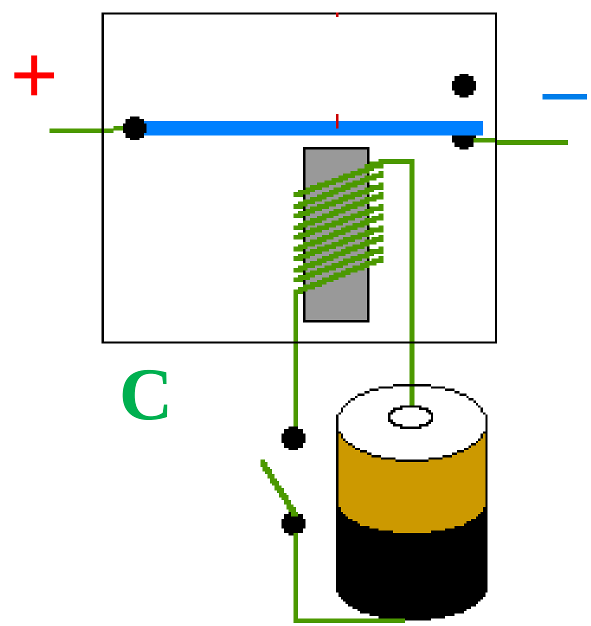
Relays, Tubes, and Transistors as Switches

- Electromechanical relays are ways to make a controllable switch:
  - Zuse's Z3 computer (1941) was entirely electromechanical
- Later vacuum tubes adopted:
  - Colossus (1943) – 1500 tubes
  - ENIAC (1946) – 17,468 tubes
- Then transistors:
  - IBM 608 was first commercially available (1957), 3000 transistors

Electromechanical Relay

- Inductor generates a magnetic field that physically pulls a switch down
- When current stops flowing through inductor, a spring resets the switch to the off position





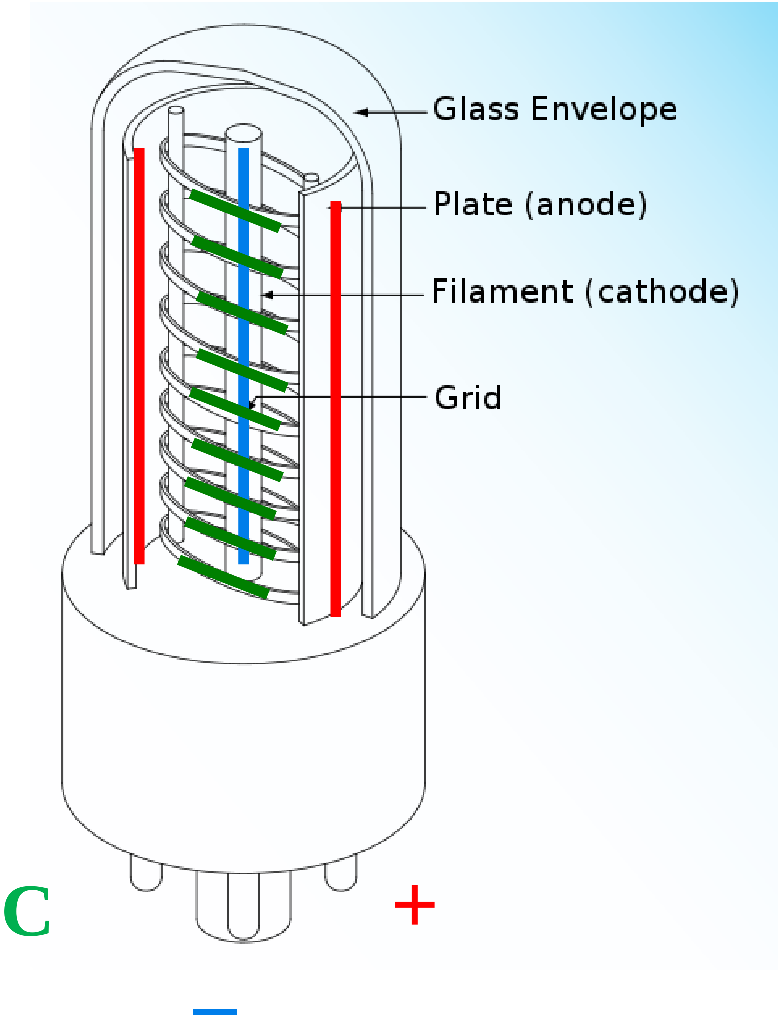
- Three Terminals:
  - **+ : Plus**
  - **– : Minus**
  - **C : Control**

Electromechnical Relay Summary

- "Switchiness" due to physically manipulation of a metal connector using a magnetic field
- Very large
- Moving parts
- No longer widely used in computational systems as logic gates
  - Occasional use in failsafe systems

Vacuum Tube

- Inside the glass, there is a hard vacuum
  - Current cannot flow
- If you apply a current to the minus terminal (filament), it gets hot
- This creates a gas of electrons that can travel to the positively charged plate from the hot filament
- When control port is used, grid becomes charged
  - Acts to increase or decrease ability of current to flow from – to +



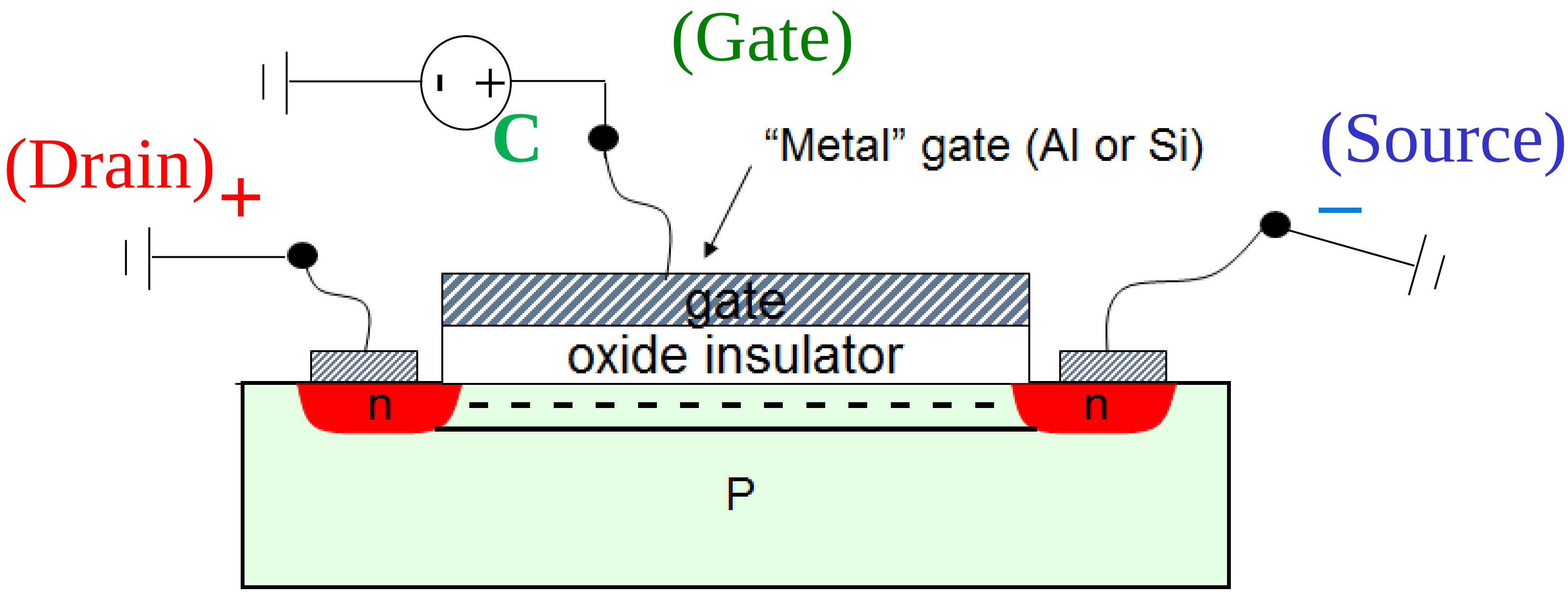
(Wikipedia)

Vacuum Tube Demo

Vacuum Tube Summary

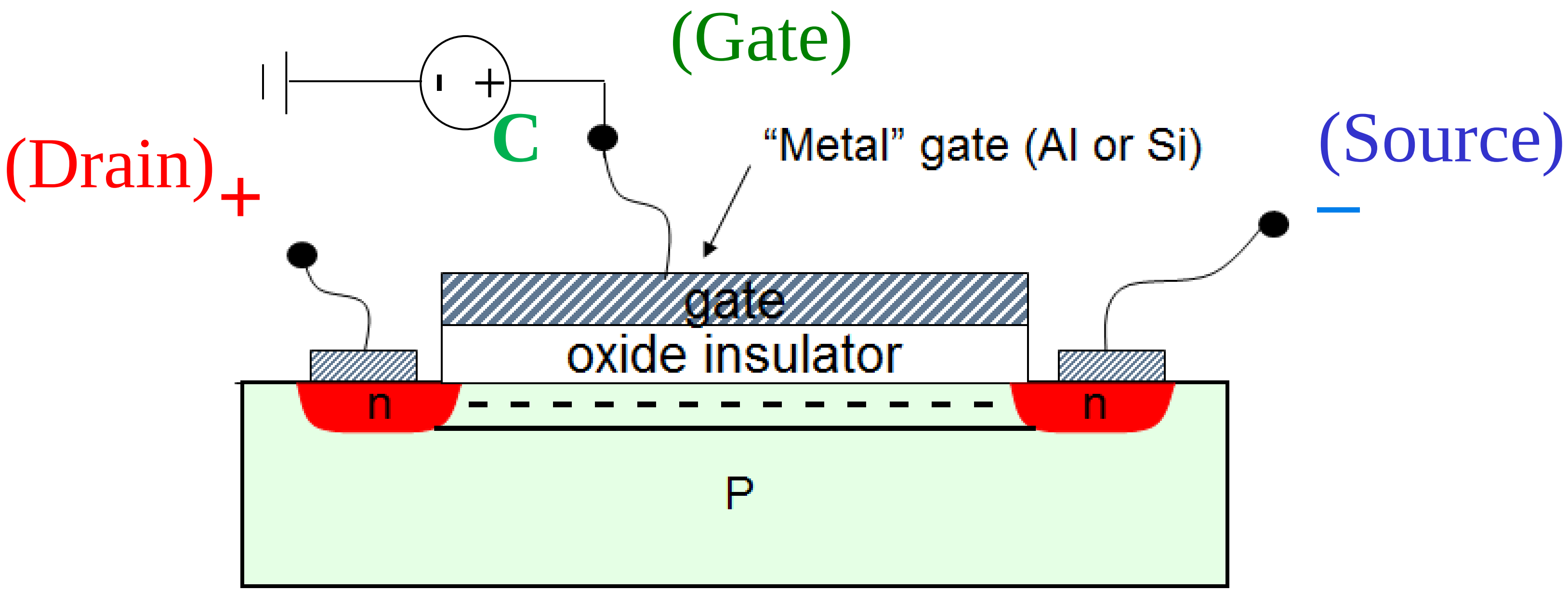
- “Switchiness” is due to a charged cage which can block the flow of free electrons from a central electron emitter and a receiving plate
- No moving parts
- Inherently power inefficient due to requirement for hot filament to release electrons
- No longer used in computational systems
- Still used in:
  - CRTs
  - Very high power applications
  - Audio amplification (due to nicer saturation behavior relative to transistors)

Field Effect Transistor



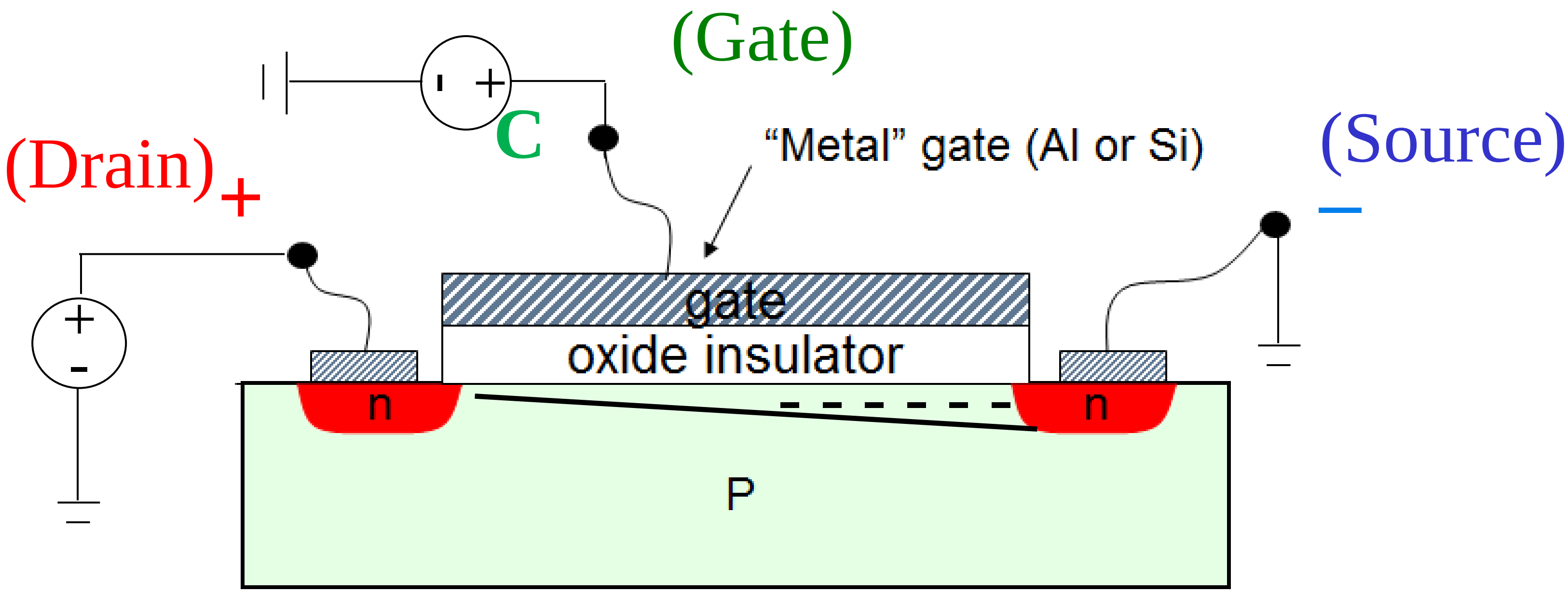
- P is (effectively) a high resistance block of material, so current can barely flow from **+** to **–**
- The n region is a reservoir of extra electrons (we will discuss the role of the n region later)
- When **C** is "on", i.e. V_c is relatively positive, then electrons from inside the P region collect at bottom of insulator, forming a "channel"

Field Effect Transistor



- When the channel is present, then effective resistance of P region dramatically decreases
- Thus:
  - When **C** is "off", switch is open
  - When **C** is "on", switch is closed

Field Effect Transistor



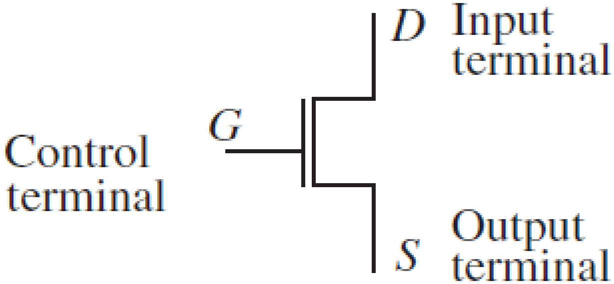
- If we apply a positive voltage to the plus side
  - Current begins to flow from + to –
  - Channel on the + side is weakened
- If we applied a different positive voltage to both sides?

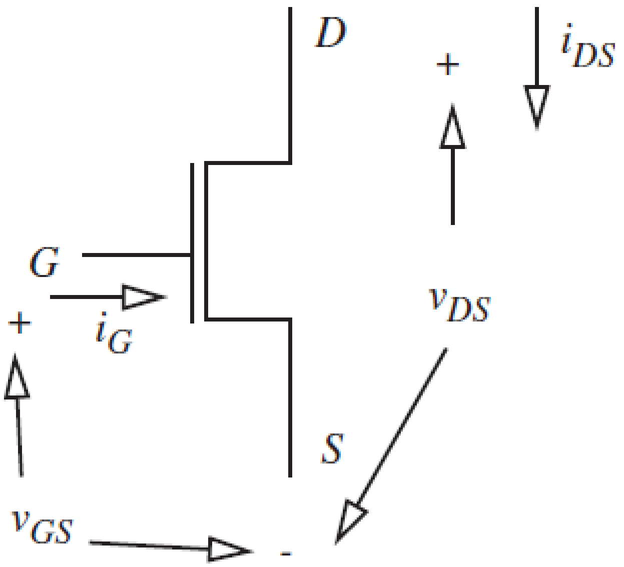
Field Effect Transistor Summary

- "Switchiness" is due to a controlling voltage which induces a channel of free electrons
- Extremely easy to make in unbelievable numbers
- Ubiquitous in all computational technology everywhere

MOSFET Model

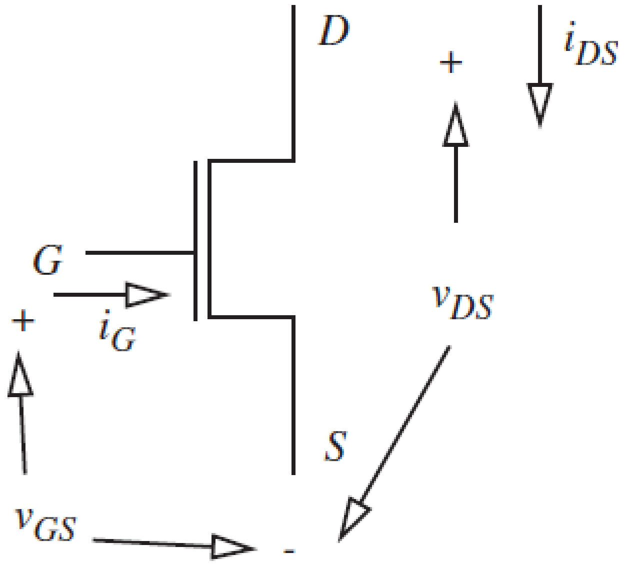
- Schematically, we represent the MOSFET as a three terminal device
- Can represent all the voltages and currents between terminals as shown to the right

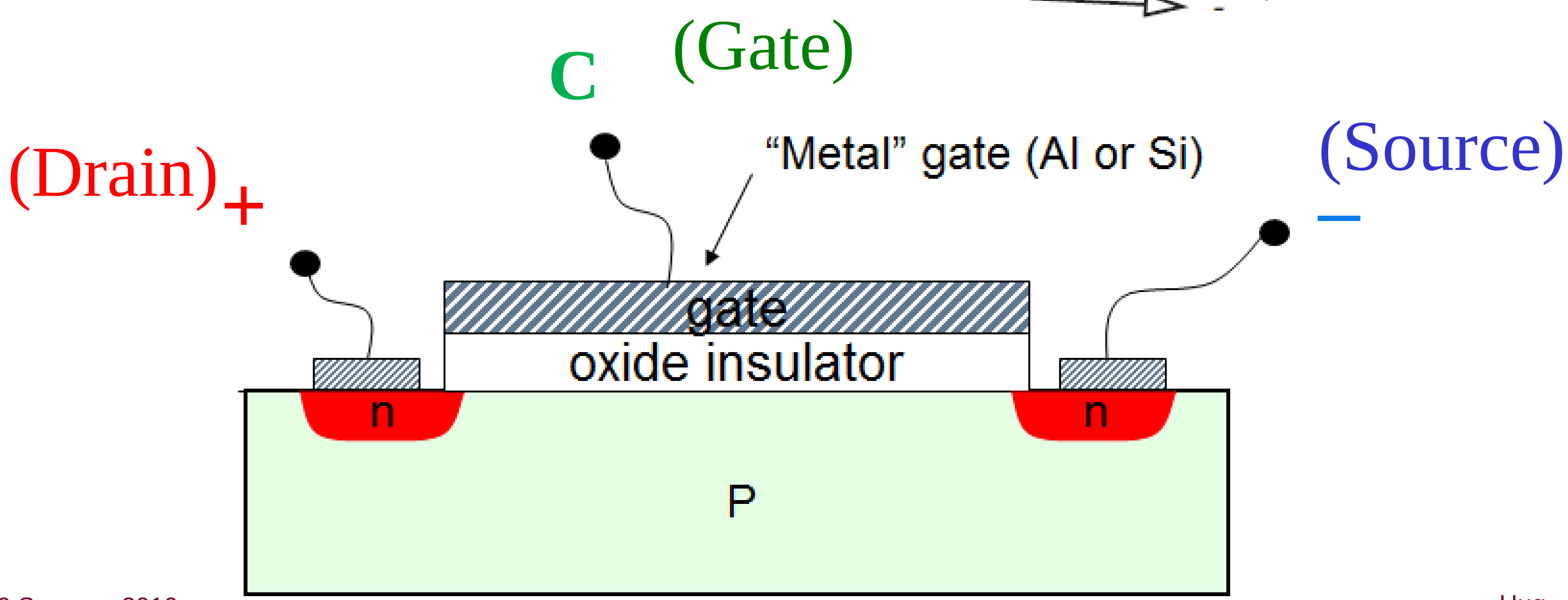




MOSFET Model

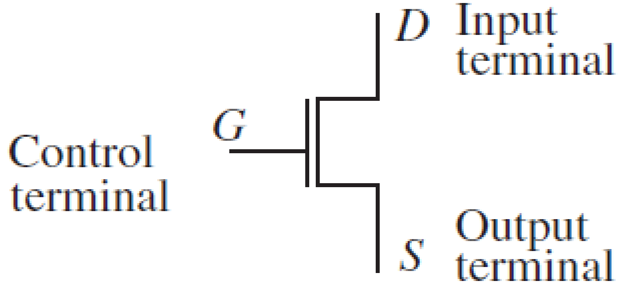
- What do you expect i_G to be?

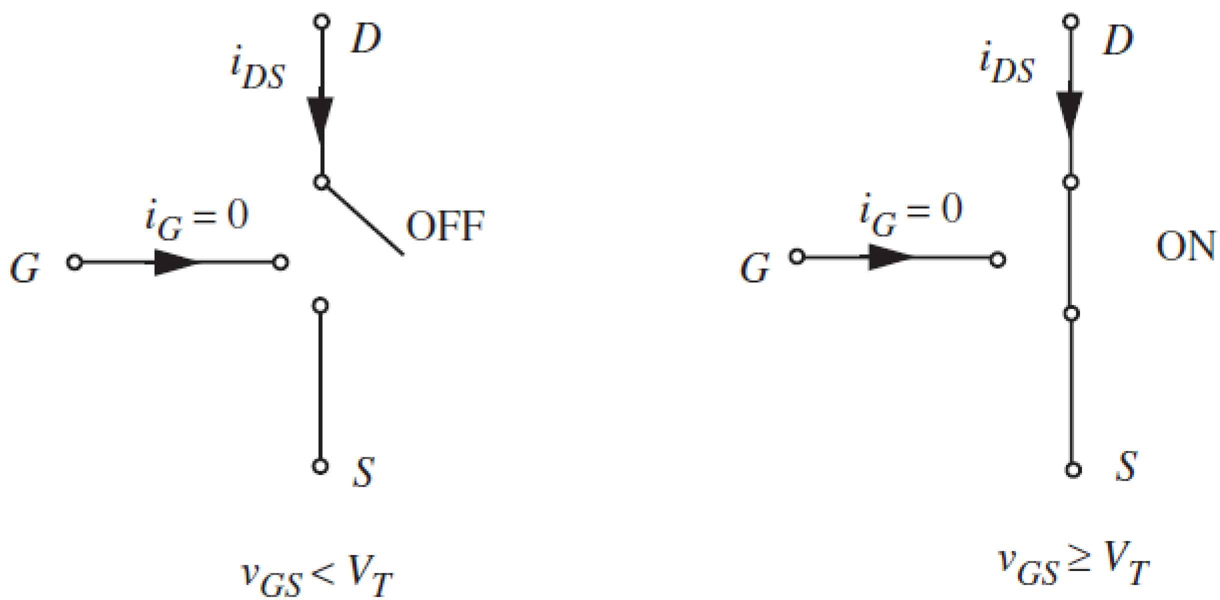




S Model of the MOSFET

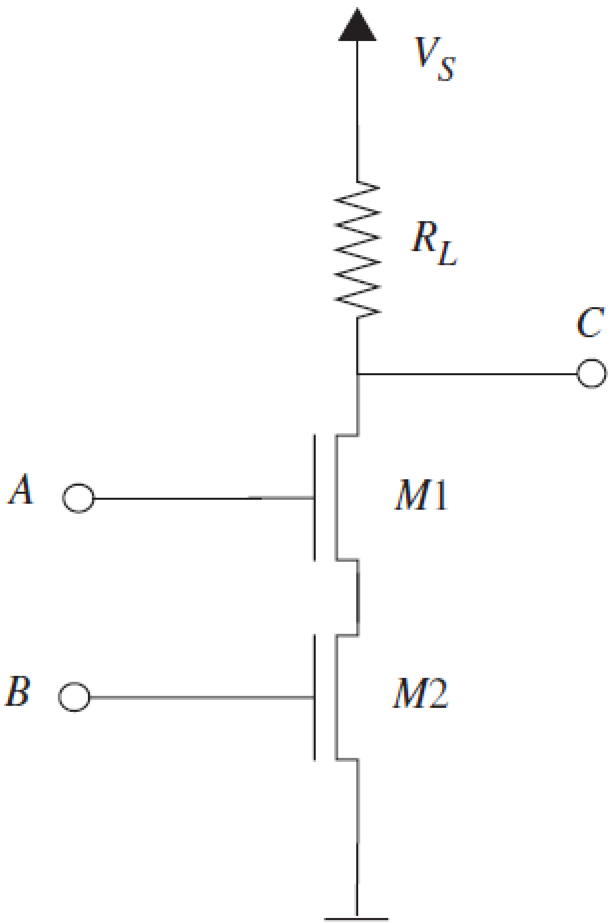
- The simplest model basically says that the MOSFET is:
  - Open for $V_{GS} < V_T$
  - Closed for $V_{GS} > V_T$





Building a NAND gate using MOSFETs

- Consider the circuit to the right where V_S
- On the board, we'll show that $C = \overline{AB}$
- Demonstration also on page 294 of the book



That's it for today

- Next time, we'll discuss:
  - Building arbitrarily complex logic functions
  - Sequential logic
  - The resistive model of a MOSFET
- Until then, study