Logic Design (Part 1)
Transistors & Gates (Chapter 3)

1

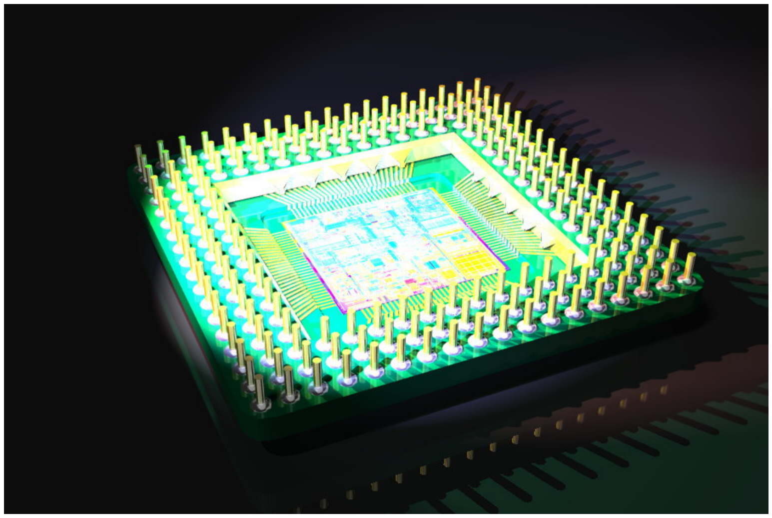
Recap…where are we:

- Data representation
  - 2C representation of integers
    - Convert decimal to 2C and 2C to decimal
    - 2C rep of decimal N : if positive, same as unsigned
      if negative, flip bits and add
- ASCII
- Arithmetic operations
- Boolean logic:
  - Truth tables
  - Logical connectives: AND, OR, NOT, XOR, ….
  - Derive values of boolean expression by filling truth table
- How is data represented in C

2

2

Agenda next 3 weeks: Inside a microprocessor



3

3

Recall: what are Computers meant to do ?

- We will be solving problems that are describable in English (or Greek or French or Hindi or Chinese or ...) and *using a box filled with electrons* and magnetism to accomplish the task.
  - This is accomplished using a system of well defined (sometimes) transformations that have been developed over the last 50+ years.
- At the lowest level, computers use 0's and 1's (binary) to represent data

- We first take a quick look at technology that gets electrons to run around

4

4

Problem Transformation- levels of abstraction

Natural Language
Algorithm
Program
Machine Architecture
Micro-architecture
Logic Circuits
Devices

Our current focus:
The building blocks:
electronic devices

5

5

Recall:
Why use Binary and How to represent data in a computer?

- At the lowest level, a computer has electronic "plumbing"
  - Operates by controlling the flow of electrons
    - Electrons flowing on the wire when voltage exists

- Easy to recognize two conditions: 0 or 1
  1. presence of a voltage – call this state "1"
  2. absence of a voltage – call this state "0"

  More complex to base state on value of voltage, but can be done

- Think of the two states 0,1 as states of a switch
  - Change from 0 to 1 means throwing switch to turn on the light
  - Presence of voltage on the wire means value of bit = 1 else 0

6

6

Physics review from Labs

- Electricity corresponds to the flow of negatively charged particles called electrons.
  - Particles of opposite sign, (+ve and –ve), attract each other
  - Particles of the same sign repel each other.
- A voltage difference between 2 points captures the amount of work it would take to move charge from one point to another
- analogous to an elevation difference in a waterfall
- Current is the flow of electrons

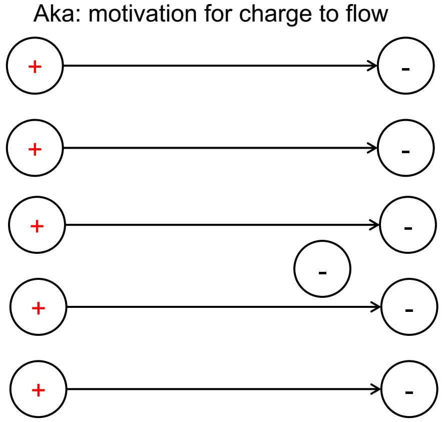
- **Ohm's Law V = IR**

7

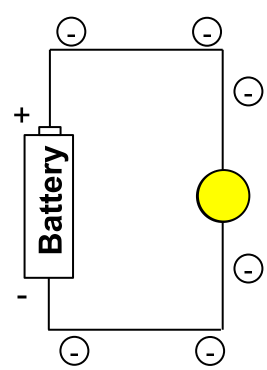
7

Voltage/Current and Electric Field

E-field produces "potential difference"
Aka: motivation for charge to flow



Battery provide voltage
Aka: potential difference



Direction of charge carrier (e-)

Direction of current

Direction of current

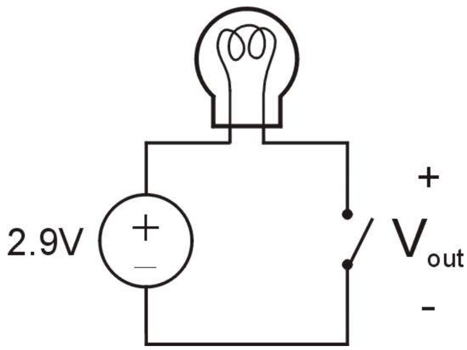
Ohm's Law: V = IR

8

8

Simple Switch Circuit

The light bulb has a resistance value



- Switch open:
  - No current through circuit
    - Resistance=infinity
  - Light is off
  - V_{out} is +2.9V
- Switch closed:
  - Short circuit across switch
  - Current flows
  - Light is on
  - V_{out} is 0V

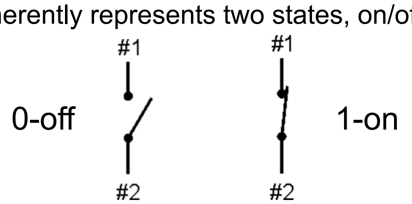
Key Takeaway:
Switch-based circuits can easily represent two states:
on/off, open/closed, voltage/no voltage, 0/1!!

9

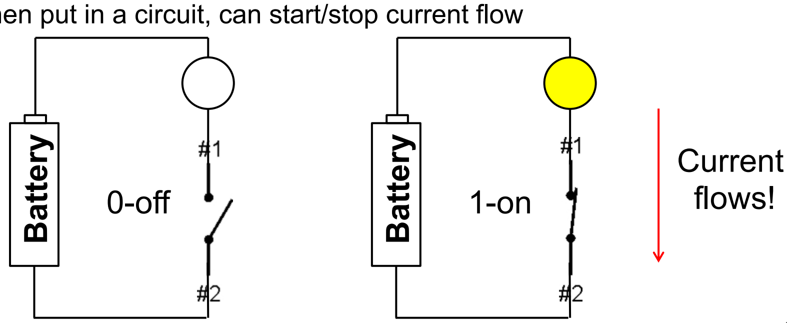
9

Switches to logic

- A switch inherently represents two states, on/off



- When put in a circuit, can start/stop current flow



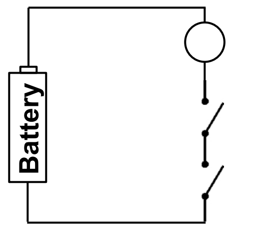
10

10

Switches to logic

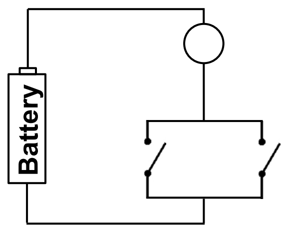
- Putting multiple switches together, and we get basic logic structures

Switches are in series (AND)



Both switches must be “on” for bulb to light up (AND)

Switches are in Parallel (OR)



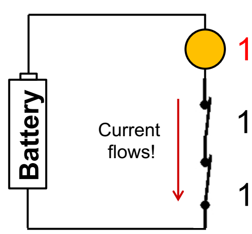
Only 1 switch Must be “on” for Bulb to light up (OR)

11

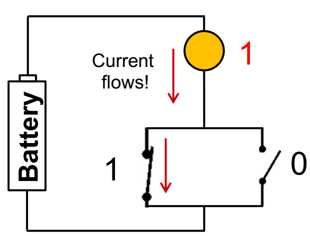
11

Switches to logic

- Putting multiple switches together, and we get basic logic structures



Both switches must be “on” for bulb to light up (AND)



Only 1 switch Must be “on” for Bulb to light up (OR)

12

12

Digital Circuits:
It's all about switching...

- Tubes
- Transistors
- CMOS FET

Computers use transistors as switches to manipulate bits

Before transistors: tubes, electro-mechanical relays (pre 1950s)

Mechanical adders (punch cards, gears) as far back as mid-1600s

13

13

Vacuum Tubes

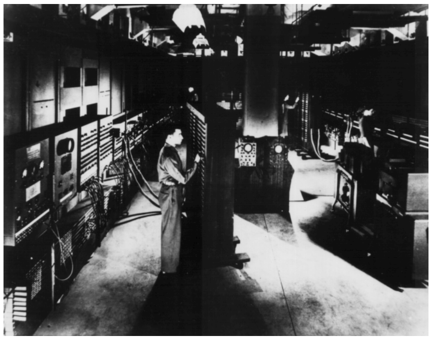
- Also known as valves because they control the flow of electrons
  - Flow from Cathode to Anode
- First computer built using vacuum tubes

14

14

Historical Perspective

- ENIAC built in World War II the first general purpose computer
  - Used for computing artillery firing tables
  - 80 feet long by 8.5 feet high and several feet wide
  - Each of the twenty 10 digit registers was 2 feet long
  - Used 19,000 vacuum tubes
  - Performed 1900 additions per second



Historical Fact: Who are the “top secret rosies”?

15

15

Transistors: Building block of computers

- Also viewed in digital circuits as a “switch”
  - Transistors used in analog circuits: Stereos, Image proc., etc.
- Brought about a big change
  - Size, Speed, Precision
  - Moore’s law: they get smaller and faster
    - Can put more and more onto a single chip
- Microprocessors contain millions of transistors
  - Intel 80286 (1982): 200,000
  - Intel i860 (1989): 1 million
  - Intel Pentium 4 (2000): 48 million
  - Intel Core Duo 2 (2006): 291 million
  - Intel 8-core Xeon Nehalem-EX (2010): 2.3 billion
  - Intrl Core 9 (2019): >5 billion
  - GPUs: nVIDIA GA100 (2020): 54 billion
    - Some flash memory chips contain over trillion

16

16

Basics of Digital Circuit Design

- How to build a switch ?
  - Transistors
- How to build basic logic functions – gates using transistors ?
  - Build simple gates (AND, NOT, OR, …) using transistors
- How to build more complex combinational logic using gates
  - Build Adders, multiplexer, decoder, storage devices using simple gates (AND,NOT, OR..)
- Build a whole computer using complex logic devices
  - Assemble all the pieces together into an 'orchestra' – this is the CPU !

- Important: power of abstraction (and layers)
  - Once you know how to build a gate using transistors, you don't have to think transistors any more!
  - Once you have a collection of gates on a single chip, you don't have to think about individual gates.
  - etc. etc.

17

17

What is a transistor?

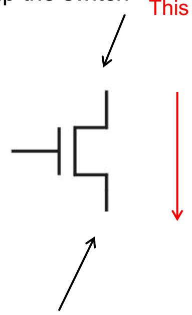
- A transistor is an electrical device that allows us to control the flow of current in a circuit
  - A transistor can act like an electronic "switch" in a circuit
  - A transistor can also function as an "amplifier" of voltage or current
- Over the decades, engineers have developed several electronic "switches" in circuits:
  - mechanical relays, vacuum tubes
  - diodes, transistors
  - MEMS devices, photonic, biological
- Switch-like behavior is important, because it can give rise to logic
  - In a CPU, we use transistors as switches, to implement logic gates
- **Voltage controlled switch**
  - the switch is closed or open depending on input voltage

18

18

Transistor as electronic switch

- In the previous example with switches, someone must manually "flip" the switches to control the "input" to our gates
- In a computer we need to generate signal to flip the switch
  - Transistor offers us this capability
  - We use voltage, to remotely flip the switch
- A transistor has 3 terminals:



19

19

How does a transistor work – Semiconductor basics

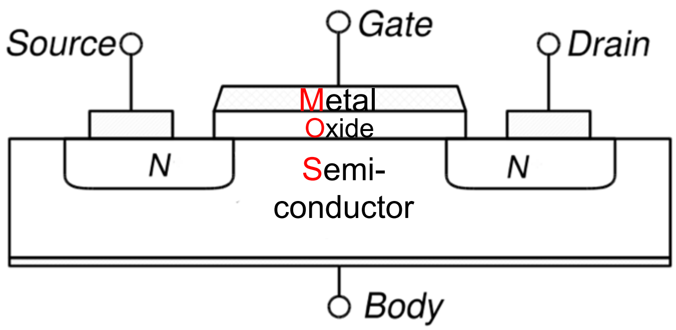
- Most materials are either insulators or conductors
  - They don't "change" their properties
- Semiconductors: between insulator and conductor
- Semiconductors: Based on voltage applied to "gate" it is either insulator or a conductor
  - Electric field creates a circuit
  - Changes the device from an insulator to a conductor
- Overview: two types of semiconductor materials
  - N-type: extra electrons can be used to carry a current
  - P-type: extra 'holes' into which electrons can flow
- how does it work ?? For more details read Appendix slides (at the end)

20

20

the MOSFET (your 1st Transistor!)

- MOSFET : Metal Oxide Semiconductor Field Effect Transistor
- Picture shows a cross section of such a device.
- Materials: metal, oxide, semiconductor

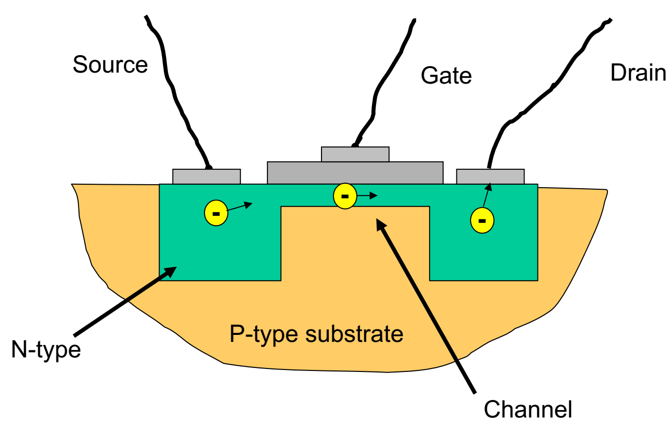


21

21

MOSFET (Metal Oxide SemiConductor)

Notice it has 3 terminals:
Source, Drain, Gate
Voltage applied to Gate determines switch behavior

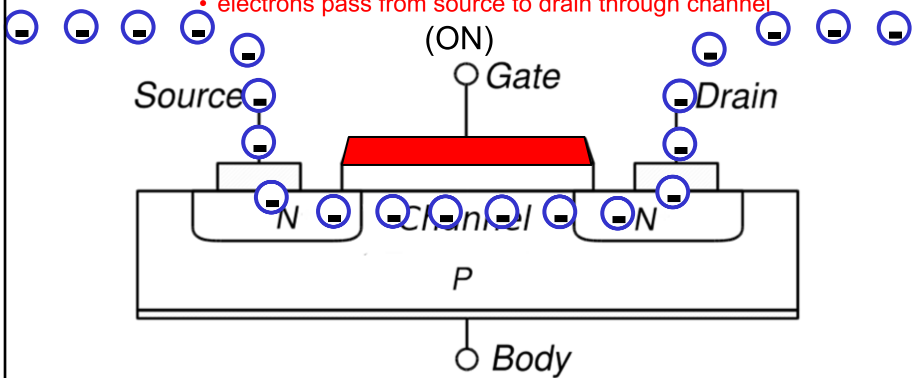


22

22

How we want it to work…

- Goal: Pass current through this device (from drain to source)
  - BUT we want to control that current (using the gate terminal)
    - If GATE is ON
      - electrons pass from source to drain through channel

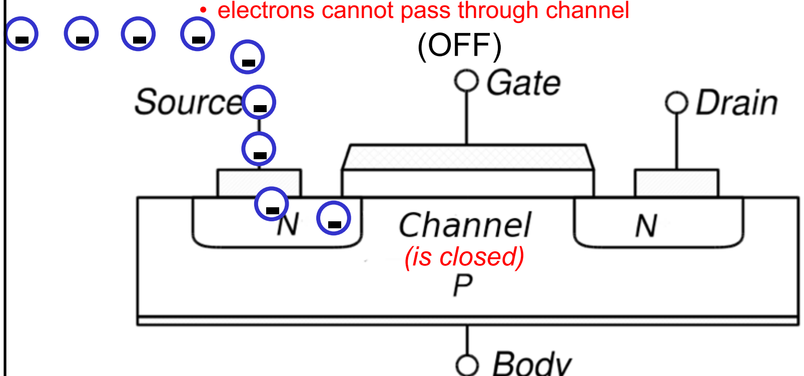


23

23

How we want it to work…

- Goal: Pass current through this device (from drain to source)
  - BUT we want to control that current (using the gate terminal)
    - If GATE is OFF
      - electrons cannot pass through channel

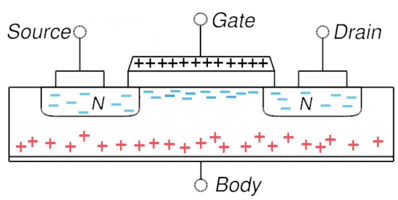


24

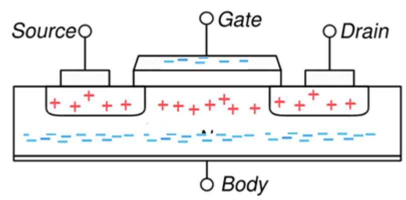
24

Two types of MOSFETs: nMOSFET and pMOSFET

- nMOSFET (nMOS): channel carries negative charges (electrons)
  - GATE MUST BE (+) to be ON
- pMOSFET (pMOS): channel carries positive charges (holes)
  - GATE MUST BE (-) to be ON



n-type



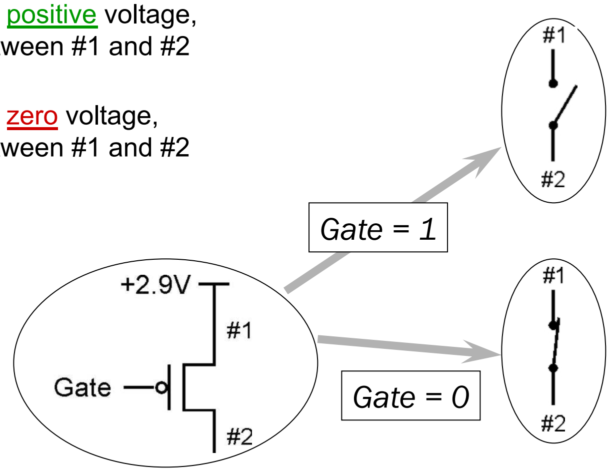
p-type

25

25

Abstraction: Simplified view of p-type MOS Transistor

- p-type
  - when Gate has positive voltage, open circuit between #1 and #2 (switch open)
  - when Gate has zero voltage, short circuit between #1 and #2 (switch closed)



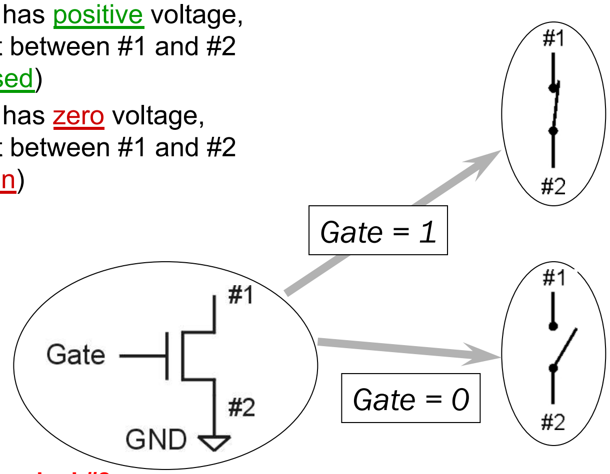
Important: For p-type, Terminal #1 must be connected to Voltage Source.

26

26

Abstraction: Simplified view of n-type MOS Transistor

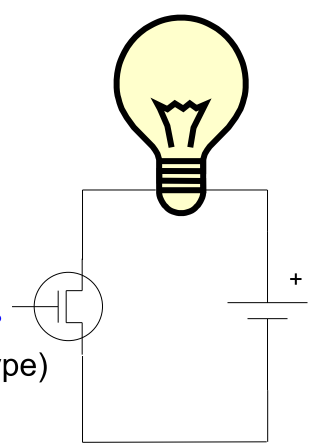
- n-type *complementary* to p-type
  - when Gate has positive voltage, short circuit between #1 and #2 (switch closed)
  - when Gate has zero voltage, open circuit between #1 and #2 (switch open)



mportant: For n-type, Terminal #2
nust be connected to Ground (0V).

27

27

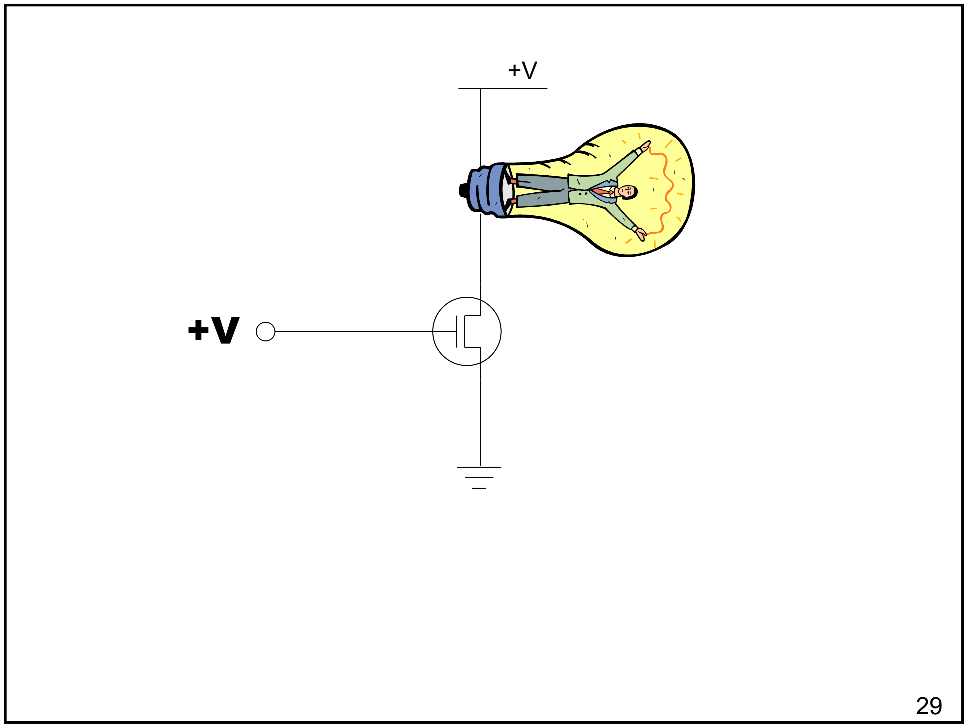


To turn on the light
What voltage do we apply here?

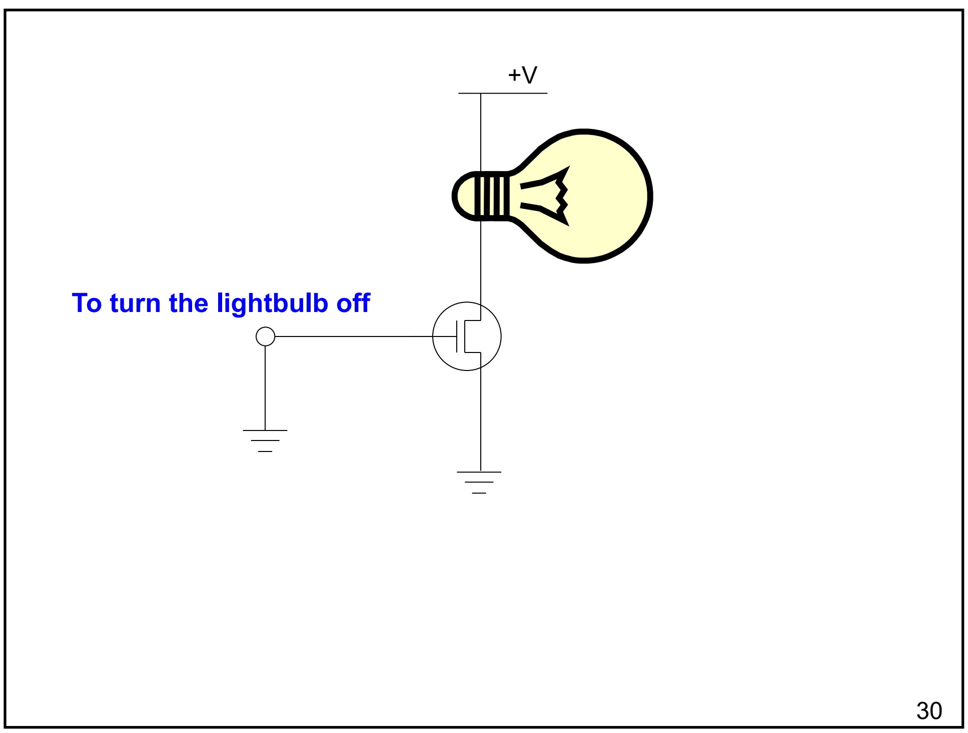
(N type)

28

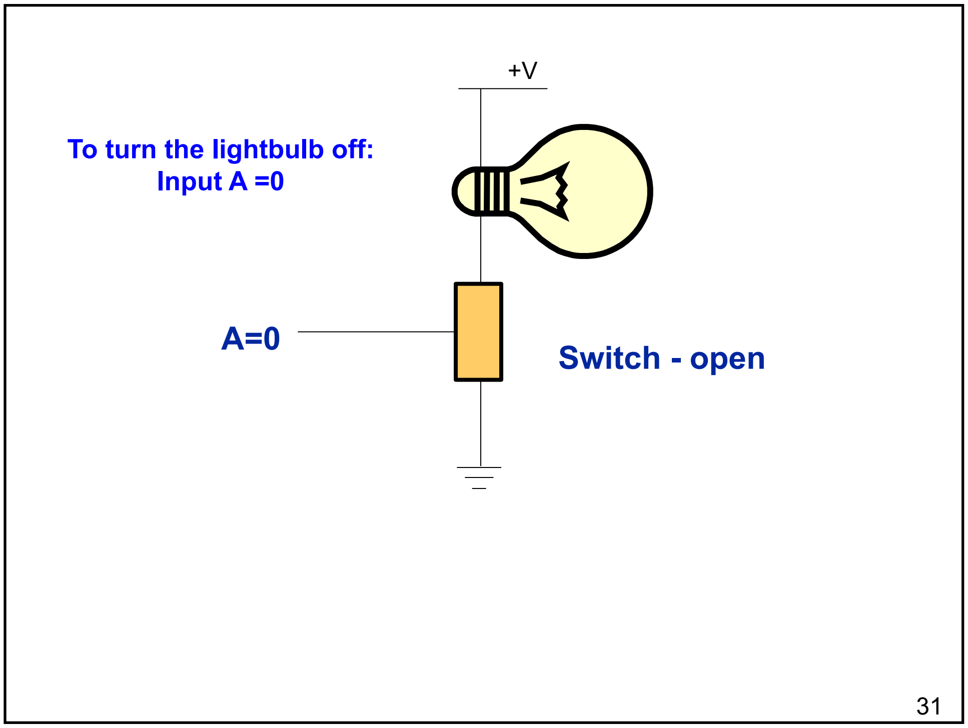
28



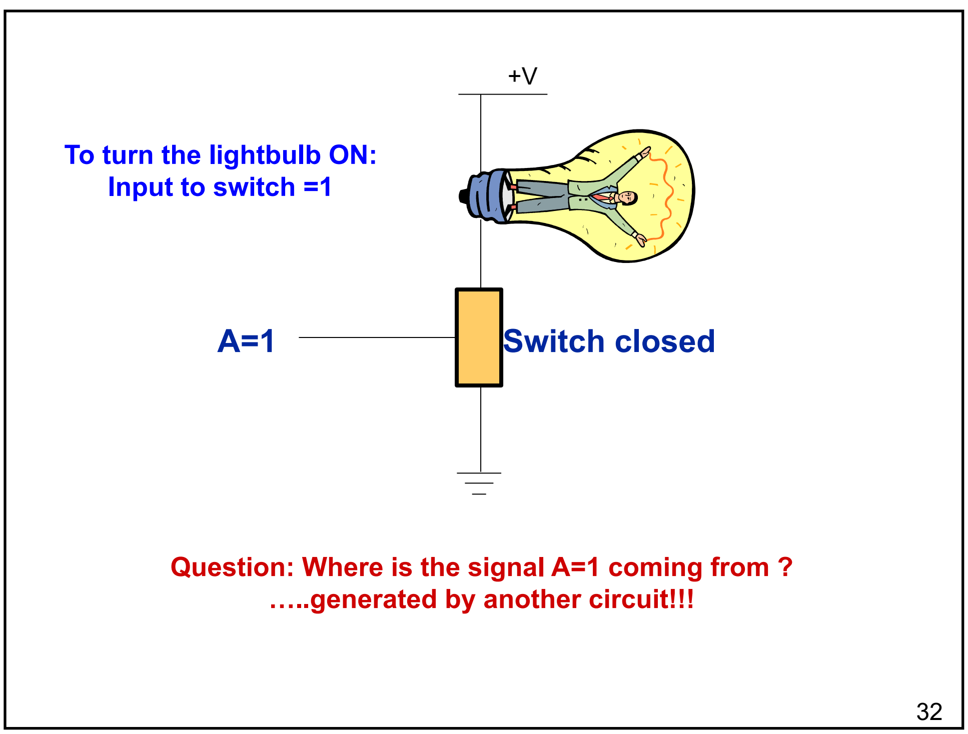
29



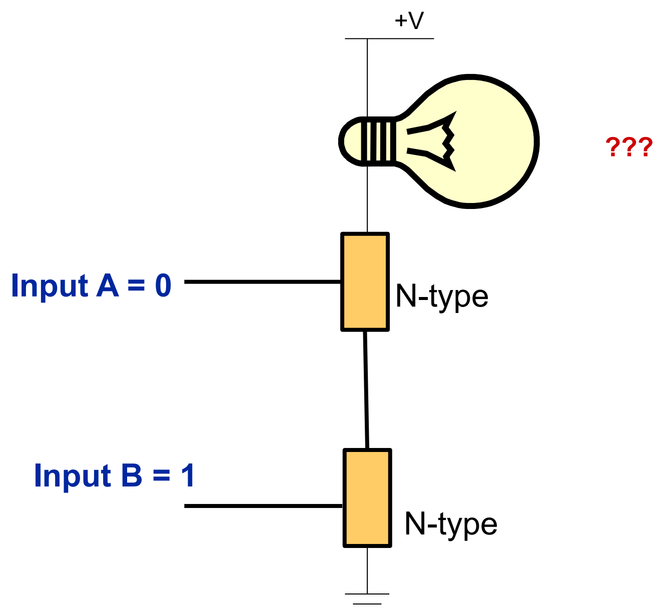
30


+V
To turn the lightbulb off:
Input A =0
A=0
Switch - open
31

31


+V
To turn the lightbulb ON:
Input to switch =1
A=1
Switch closed
Question: Where is the signal A=1 coming from ?
…..generated by another circuit!!!
32

32



33

33

Light bulbs and computer hardware –what the &@?#&#&!

- Let’s look back at what we’ve learnt
  - Numbers can be represented as 0s and 1s
    - 1 is presence of voltage on line, 0 is no voltage on line
  - Arithmetic operations on these numbers
  - Logical operations on these numbers
- Starting point: how to implement the basic logic operators using transistors/switches ?
  - NOT, AND, OR
- Next: how to implement arithmetic operations and other functions
  - Combinational circuits; example: adder

34

34

Logical Operations

- NOT, AND, OR, NAND, NOR, XOR
- These are binary functions
  - Input is binary, output is binary
- Boolean function – operates on boolean variables
  - Boolean function can be expressed using truth table
  - Eg: addition can be represented as a boolean function
- Recall from Discrete 1 - CS 1311: can implement any boolean function using AND, OR, NOT, etc.
  - In fact, can implement any bool function using just NAND
- Start by building these logical operator "gates" using transistors

35

35

Ok….start building logic gates

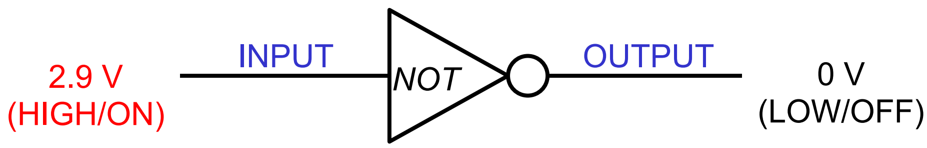
- Use Complementary MOS (CMOS) circuits
  - Using N type and P type transistors
- Use switch behavior to implement logic functions/operators
- 'signal' is a 1 or 0 and nothing else
- Output value will be voltage measured at some point in the "circuit"
  - Need to determine where to designate the output point (i.e., where to measure)
  - *This output point must (at all times) have a path (connection) to Voltage source (1) or to ground (0)*
    - The path is selected based on the value to transistor gates
- Inputs will be applied to the transistor gate
  - A line in the circuit always tied to 1 (voltage source) and one always tied to 0 (ground)
- Start by looking at the truth table for the logic function

36

36

So now what? How to go from "switch" to logic?

- Our first logic device will be an <u>inverter</u>: the NOT gate



- Logical Behavior: "inverts" the incoming signal:
  - Input: LOW-> output: HIGH
  - Input: HIGH->output: LOW

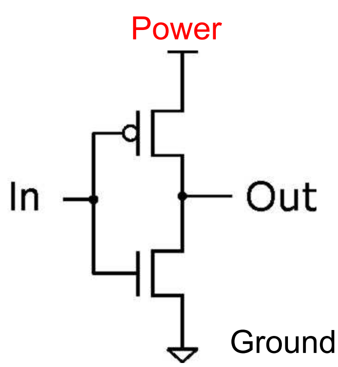
INPUT	OUTPUT
LOW (0)	HIGH (1)
HIGH (1)	LOW (0)

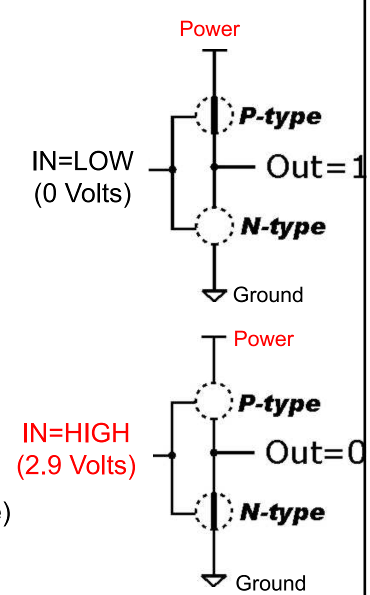
Truth Table
All possible
Combinations
Of inputs

37

37

How do we configure transistors to make inverter?



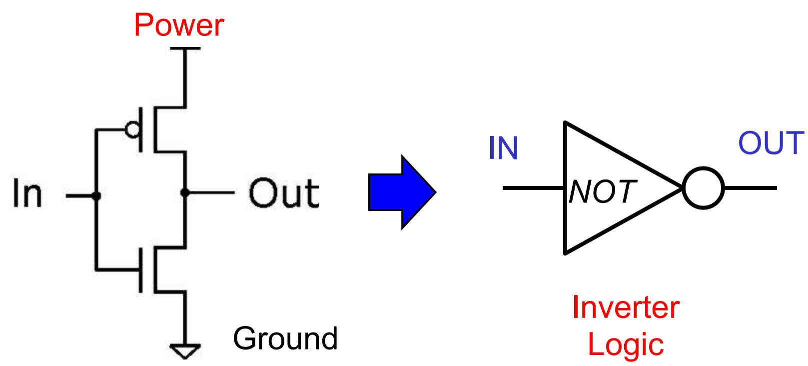


We take advantage of opposing nature!
-If pMOS turns on when GATE=0 (Ground)
and if nMOS turns on when GATE=1 (Voltage)
-then *if we put them together & connect their gates, we get inverting behavior!*

38

38

This configuration is called: CMOS



Inverter
Logic
Gate

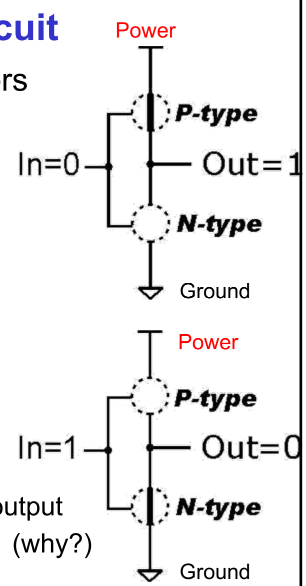
CMOS = Complimentary MOS Inverter

We have “jumped up” 1 level of abstraction
--From transistors to “gate”
--Technology inside the gate (CMOS here) isn’t as crucial as its behavior
  --could be: transistors, vacuum tubes, biological device, etc.

39

Things to notice about a CMOS Circuit

- Uses both n-type and p-type MOS transistors
  - p-type
    - Attached to POWER (high voltage)
    - Pulls output voltage UP when input is zero
    - Call PMOS devices “pull up” devices
  - n-type
    - Attached to GROUND (low voltage)
    - Pulls output voltage DOWN when input is one
    - Call NMOS devices “pull down” devices



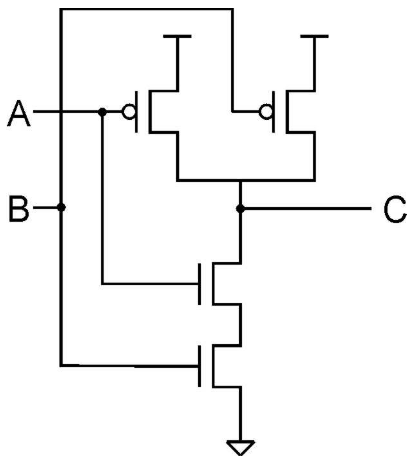
- For all inputs, this configuration makes certain that output
- connected to GROUND or to POWER, but not both! (why?)

40

40

Circuit ?

Note: Output is being measured at some location in the circuit.
make sure that point can only be 0 or 1



Note: Parallel structure on top, serial on bottom.

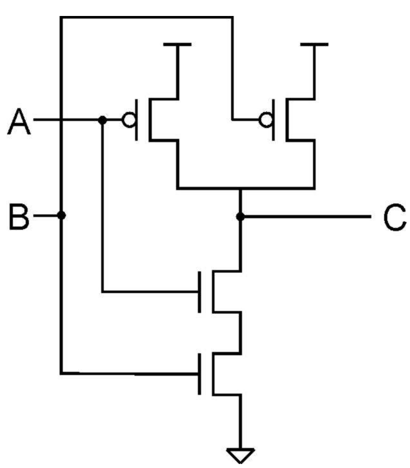
Truth table ?

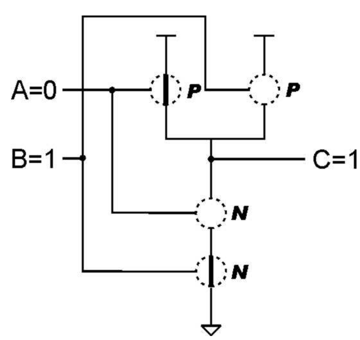
A	B	C
0	0	
0	1	
1	0	
1	1	

41

41

Example





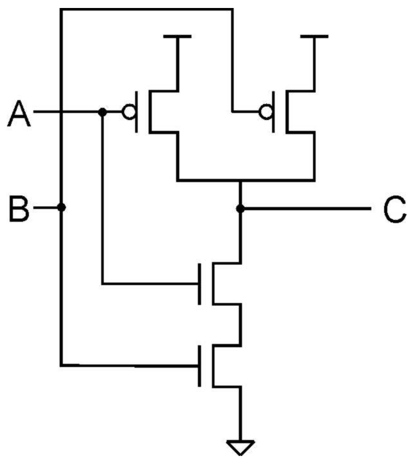
A	B	C
0	0	?
0	1	1
1	0	?
1	1	?

Note: Parallel structure on top, serial on bottom.

42

42

NAND Gate (AND-NOT)



Truth Table

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Note: Parallel structure on top, serial on bottom.

43

43

The “Logic” Behind CMOS Gate Implementation

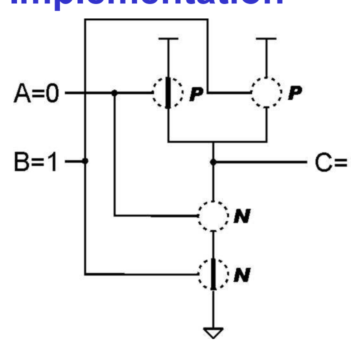
Transistors in series implement “AND”
- Current flows only if both are “ON”

Transistors in parallel implement “OR”
- Current flows if either is “ON”

CMOS is naturally inverting

Result: n-network implements function

NAND example
- n-network transistors in series gives AND
- Natural inversion gives NAND



A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

44

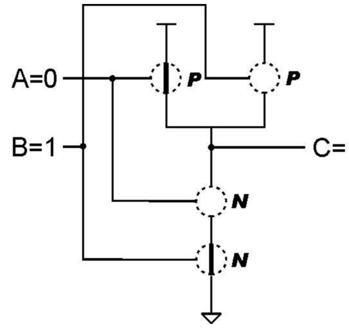
44

The "Logic" Behind CMOS Gate Implementation

P-network is complement of n-network
- Series n-network → parallel p-network
- Parallel n-network → series p-network

NAND example
- p-network transistors in parallel

- Designing in CMOS:
  - We always design the n-network (aka – the pull-down network) first
  - Then, complement it and you've figured out the p-network (aka – the pull-up network)

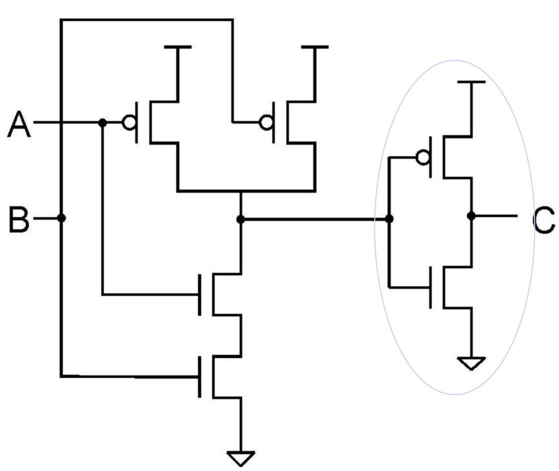


A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

45

45

AND Gate: Combining two circuits- NAND, NOT



A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

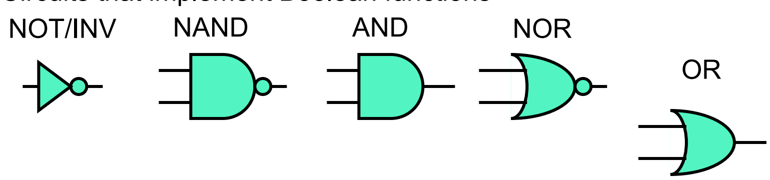
Add inverter to NAND.

46

46

Basic Logic Gates

- From Now On… Gates
  - Covered transistors mostly so that you know they exist
  - Note: "Logic Gate" not related to "Gate" of transistors
- Logic gates ~ Propositional logic operators
  - Propositional logic formula = Boolean logic circuit !
- Will study implementation in terms of gates
  - Circuits that implement Boolean functions



- More complicated gates from transistors possible
  - XOR, Multiple-input AND-OR-Invert (AOI) gates

47

47

Truth Table for common 2 input gates

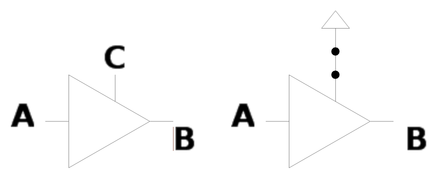
A	B	AND	OR	NAND	NOR	XOR
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

48

48

Another "gate": Tri-State Buffer

- Acts as a basic switch – a valve that is open or closed
- If C=0 then no connection from A to B
- If C=1 then A connected to B
- Why use this ?
  - Access to Bus – only signals with "valve closed" are sent to bus
  - Boost current to circuit – as resistance builds up in long paths, the signal gets weaker

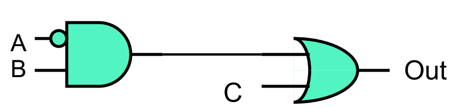


49

49

Example: Your first combinational circuit

- Combinational logic circuits ~ propositional logic statements
- Use gates to implement the logic operators ('functions')
  - No necessity to show the circuit using transistors since each gate corresponds to an implementation using transistors
- Output = ((NOT A) AND B) OR C
- Need one AND gate and one OR gate (and one NOT gate/invertor)

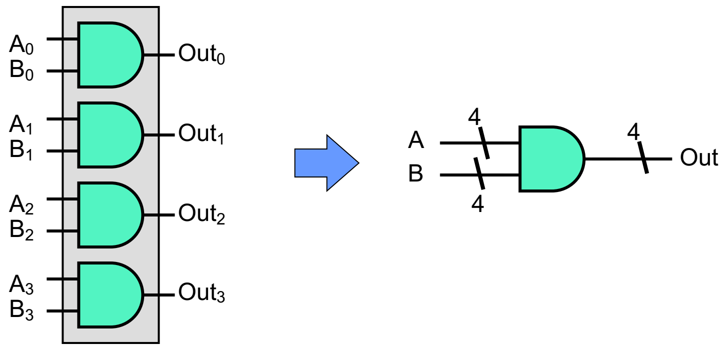


50

50

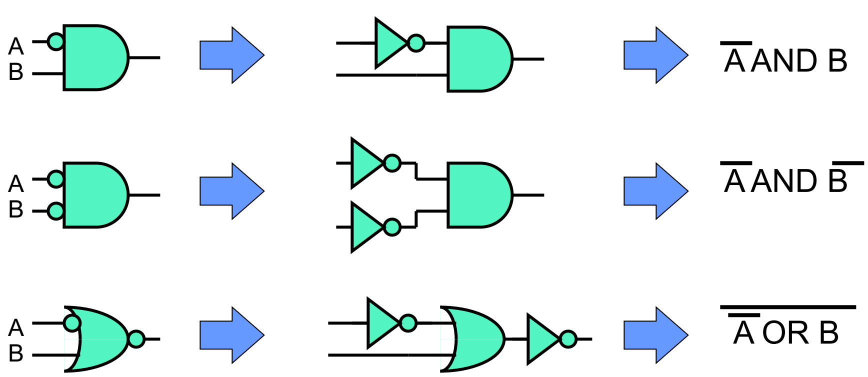
Visual Shorthand for Multi-bit Gates

- Use a cross-hatch mark to group wires
  - Example: calculate the AND of a pair of 4-bit numbers
  - A_3 is “high-order” or “most-significant” bit
  - If “A” is 1000, then $A_3 = 1$, $A_2 = 0$, $A_1 = 0$, $A_0 = 0$



Shorthand for Inverting Signals

- Invert a signal by adding either
  - a ○ before/after a gate
  - a “bar” over letter



Building Combinational logic circuits

- Integrated circuits (chips) package multiple gates into a single chip
  - Using a single gate requires connecting inputs to the appropriate pins on the chip and taking the output from a specific pin
  - A "datasheet" for each chip specifies how the pins are connected
- Labs this week: Using 7400 series chips to design logic circuits

53

53

Reading

- Chapter 3
- Lecture notes posted on webpage
- and Notes linked from webpage
- Start using Cedar Logic (Windows) or Logisim (Mac)
  - Go over the Set1.cdl examples
    - Download and save the file, open in Cedar Logic/Logisim
- Review boolean algebra concepts from CS1311
  - Summary notes on Bool.Alg. Posted on my lectures webpage

54

54

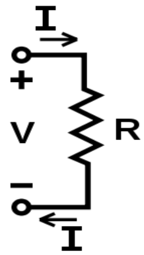
Appendix: Additional Reading/Slides

55

55

More Physics: Conductors, Insulators, Semiconductors

- Materials like metals are termed *conductors* because they allow the free flow of electrons
- Materials like rubber are termed *insulators* because they impede flow of electrons
  - Resistors are devices that will conduct some current if you encourage the electrons with a potential difference
- Semiconductors are poor *conductors* and poor *insulators*, hence "semi." They can be used for either or both properties



Ohm's Law: V = IR

56

56

How does a transistor work – Semiconductor basics

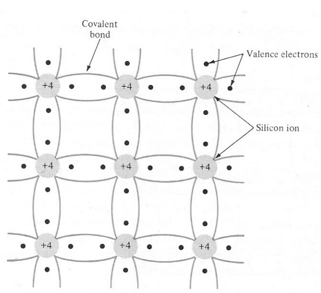
- Most materials are either insulators or conductors
  - They don’t “change” their properties
- Semiconductors: between insulator and conductor
- Semiconductors: Based on voltage applied to “gate” it is either insulator or a conductor
  - Electric field creates a circuit
  - Changes the device from an insulator to a conductor
- how does it work ??

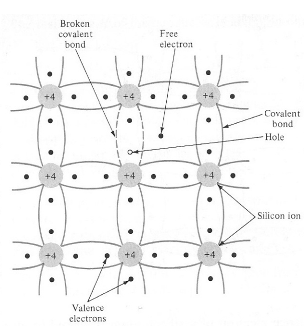
57

57

How does a transistor work?

- Begin at the beginning (*what is it made of ?*)
  - Currently transistors are etched on Silicon
    - Atomic symbol: Si – atomic number 14
  - In its crystalline state, silicon atoms form covalent bonds wi neighbors using their 4 outer electrons
  - At room temperature, Silicon is a semiconductor



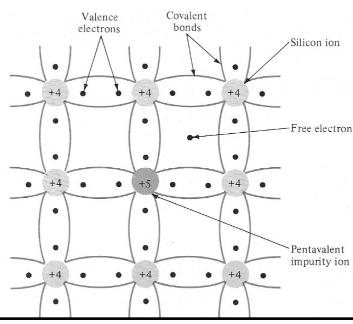


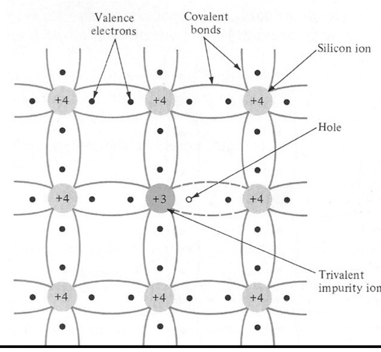
58

58

Doping – not what you think

- We can improve the conduction of Silicon by doping it with other elements.
  - N-type regions are formed by adding small amounts of elements that have more than 4 electrons in their outer shell and, these extra electrons can serve as charge carriers.
  - P-type materials are formed by adding elements that have 3 electrons in their outer valence shell.These atoms create spaces in the lattice of covalent bonds into which electrons can flow.

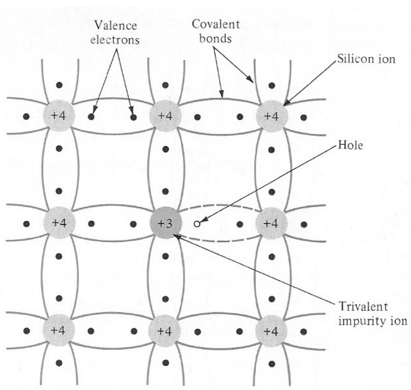


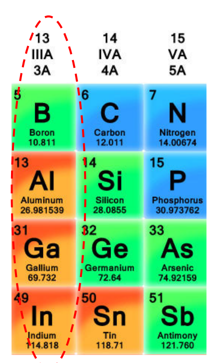


59

P-type doping

- P-type materials are formed by adding elements that have 3 electrons in their outer valence shell.
- These atoms create spaces in the lattice of covalent bonds into which electrons can flow.
- P-type dopants : boron (B), gallium (Ga), indium (In)





60

60

Bottom Line

- N-type materials are good semiconductors because they have extra electrons which are negatively charged and can be used to carry a current.
- P-type materials are good semiconductors because they have extra spaces into which electrons can move. These ‘holes’ can be thought of as positive charge carriers.

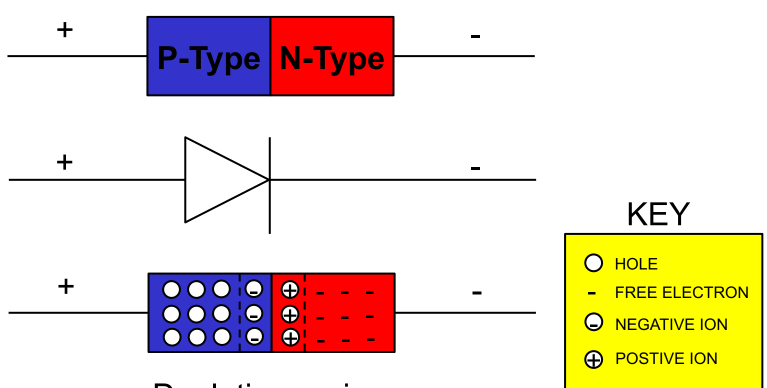
- Now we are ready to describe our building block: MOSFET transistor

61

61

A Diode (a pn-junction) – recall LED from lab

- A union of P-type and N-type materials
- Functions as a one-way “valve” in an electric circuit
- Only allows current to flow in one direction



Depletion region

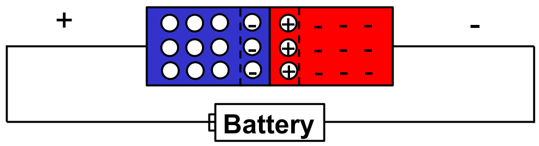
- Depletion region is an E-field that impedes the flow of current

62

62

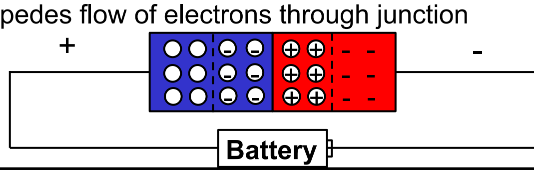
A Diode (a pn-junction)

- Forward bias:
  - Depletion region gets smaller
    - Allows current to flow from + to -
  - Allows flow of electrons through junction



- Reverse bias (reverse the battery):
  - Depletion region gets bigger
    - impedes flow of current from + to -
  - Impedes flow of electrons through junction

A diode is
Like a 1-way valve
Only lets current
In 1 direction in a
circuit

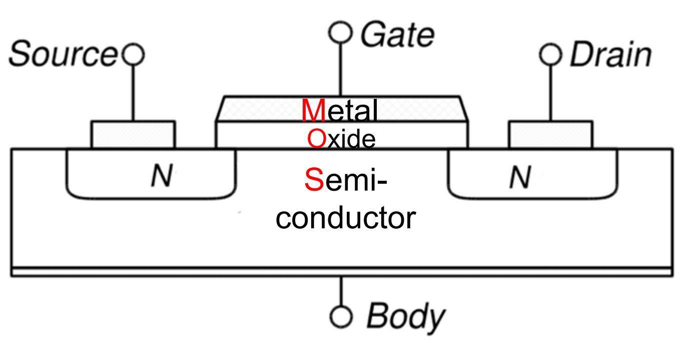


63

63

the MOSFET (your 1st Transistor!)

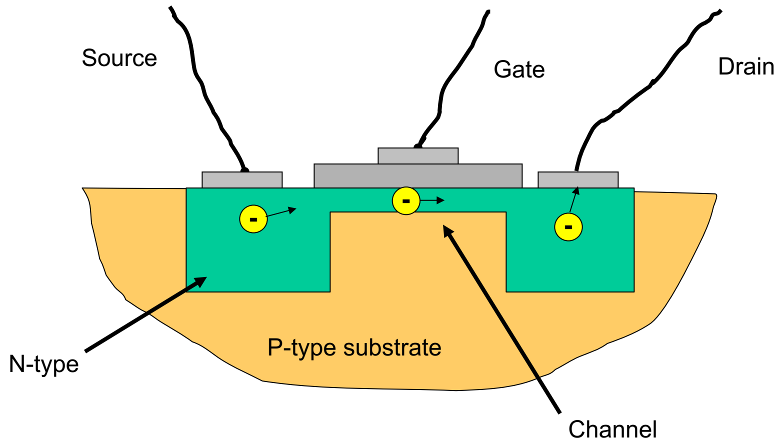
- MOSFET : Metal Oxide Semiconductor Field Effect Transistor
- Picture shows a cross section of such a device.
- Notice it has 4 electrical terminals: Source/Drain/Gate/Body



64

64

MOS FET (Metal Oxide SemiConductor)

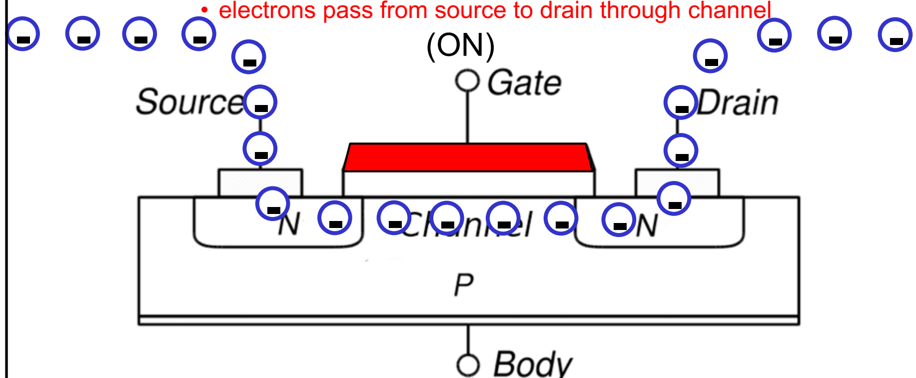


65

65

How we want it to work…

- Goal: Pass current through this device (from drain to source)
  - BUT we want to control that current (using the gate terminal)
    - If GATE is ON
      - electrons pass from source to drain through channel

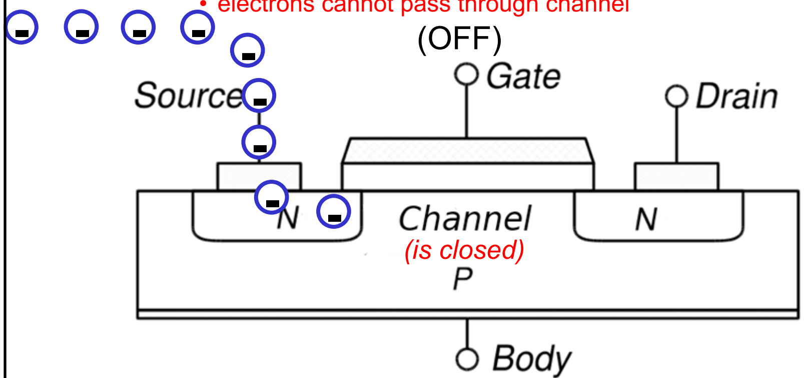


66

66

How we want it to work…

- Goal: Pass current through this device (from drain to source)
  - BUT we want to control that current (using the gate terminal)
    - If GATE is OFF
      - electrons cannot pass through channel

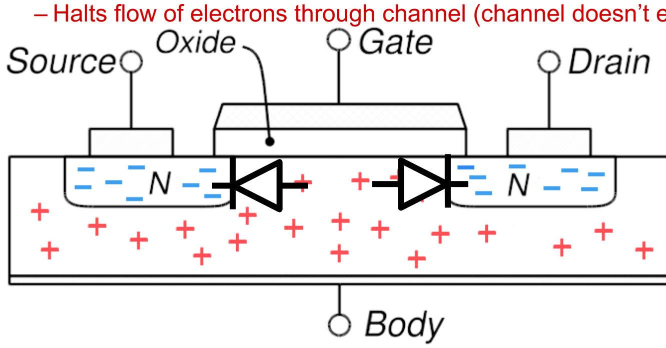


67

67

How we achieve this behavior…

- At "rest" we have (closed state)
  - 2 n-type spots (source/drain)
  - 1 p-type spot (channel region)
  - 2 back-to-back diodes!
    - Halts flow of electrons through channel (channel doesn't exist!)

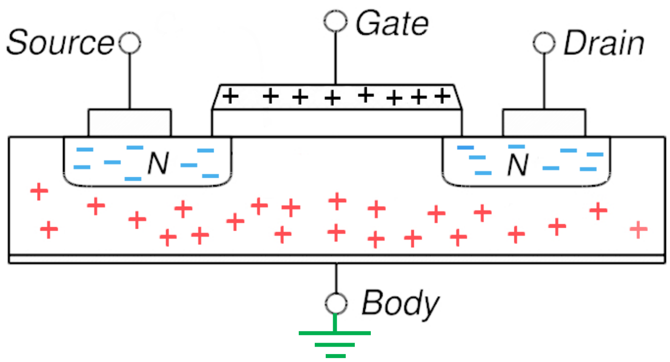


68

68

How we achieve this behavior…

- If we wish to turn device on:
  - We apply a “positive” voltage to GATE with respect to BODY
    - This positive voltage “repels” holes from under the gate
    - “depletes” the future channel region of all its holes

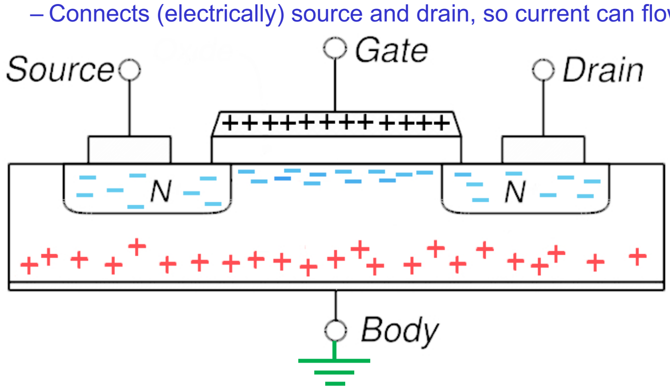


69

69

How we achieve this behavior…

- If we go further:
  - Apply a “very positive” voltage to the gate
    - Begins to attract electrons (from source & drain)
    - The channel region has been “inverted”
    - Connects (electrically) source and drain, so current can flow!

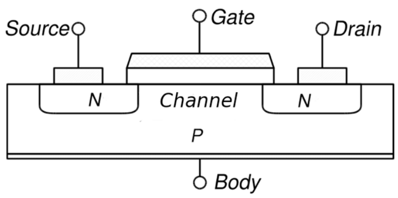


70

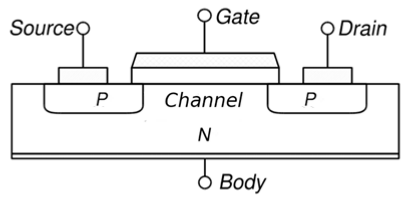
70

Two types of MOSFETs: nMOSFET and pMOSFET

- nMOSFET (nMOS): channel carries negative charges (electrons)
- pMOSFET (pMOS): channel carries positive charges (holes)



nMOSFET



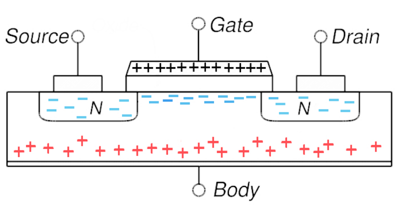
pMOSFET

71

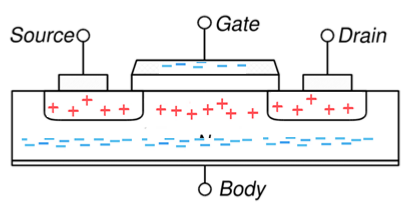
71

Two types of MOSFETs: nMOSFET and pMOSFET

- nMOSFET (nMOS): channel carries negative charges (electrons)
  - GATE MUST BE (+) to be ON
- pMOSFET (pMOS): channel carries positive charges (holes)
  - GATE MUST BE (-) to be ON



nMOSFET



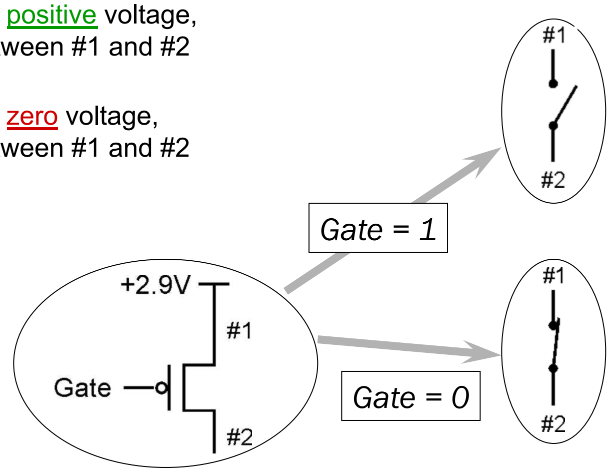
pMOSFET

72

72

Simplified view: p-type MOS Transistor

- p-type
  - when Gate has positive voltage, open circuit between #1 and #2 (switch open)
  - when Gate has zero voltage, short circuit between #1 and #2 (switch closed)



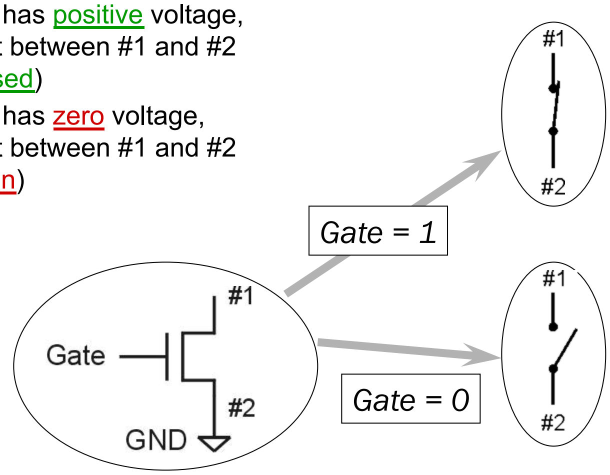
Terminal #1 must be connected to +2.9V.

73

73

Simplified view: n-type MOS Transistor

- n-type *complementary* to p-type
  - when Gate has positive voltage, short circuit between #1 and #2 (switch closed)
  - when Gate has zero voltage, open circuit between #1 and #2 (switch open)



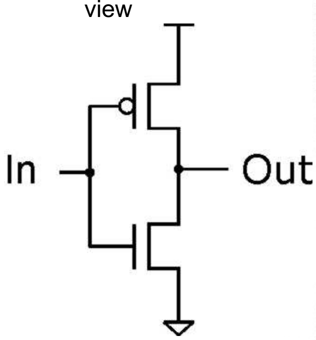
Terminal #2 must be connected to GND (0V).

74

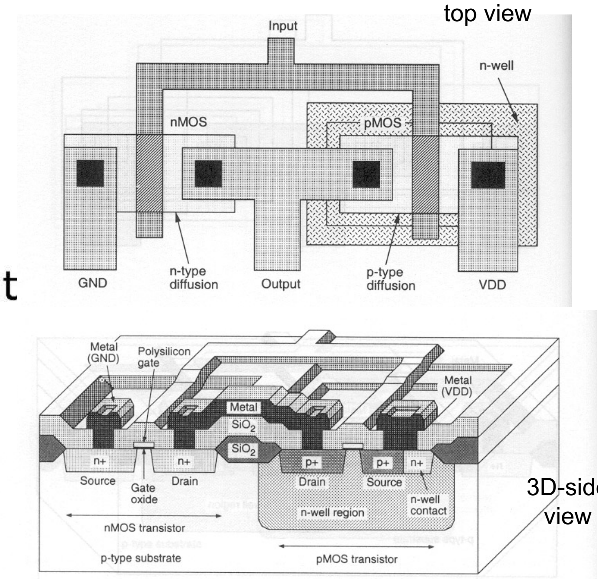
74

3-D View of CMOS Inverter in Silicon

schematic view



top view



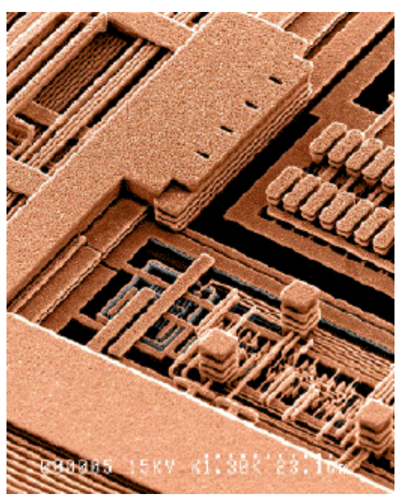
3D-side view

Note: we can make pMOS and nMOS transistor on the same piece of silicon

. 5

75

3-D of larger CMOS circuits



This is an SEM photo

shows all the metal Interconnections On an IC

pMOS/nMOS are at the very bottom

76

76

Some observations about CMOS – why does your laptop get hot ?

- Note that when the circuit is fully ON or fully OFF there is no path from the high voltage to the low voltage so no current flows

- However, when the output is in the process of switching from one logic level to another, there can be overlap of the two switches being on
  - this causes a momentary short
    - – (current goes from pwr-to-gnd)
  - Longer the short, more current you burn (more power wasted)!

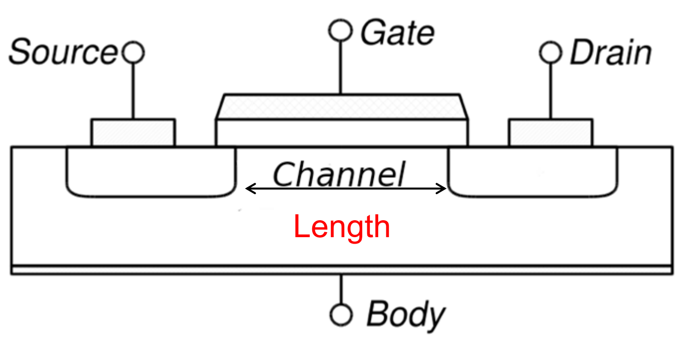
- When current flows, device gets hot
  - The faster you switch the circuit, the more current flows, the more heat is generated, the hotter your laptop gets.
  - This has proven to be an important barrier to speeding up CMOS circuitry
    - led to wide use of Multi-Core processors….

77

77

Speed of MOSFET

- Dependent on many factors, 1 crucial factor: Length of Channel
  - Why? Electron takes less time to travel across smaller distance!
    - – Currently, 11nm in length!
  - Smaller the length, faster the 'speed'

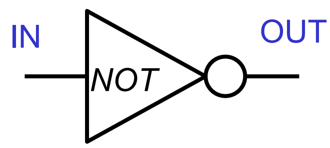


78

78

Gate Delays …. How fast is your computer

- With any logic circuit there will be a short delay between the time you change one of the inputs and the time the output settles to its final value.
- This time is referred to as the gate delay.
- For modern circuitry, these gate delays are on the order of nano seconds (10^{-9} seconds) or pico seconds (10^{-12} seconds) .
- Nonetheless, these delays ultimately limit the rate at which you can compute – limiting the number of operations you can perform per second.



Inverter Logic Gate

79

79

The “Logic” Behind CMOS Gate Implementation

Transistors in series implement “AND”

- Current flows only if both are “ON”

Transistors in parallel implement “OR”

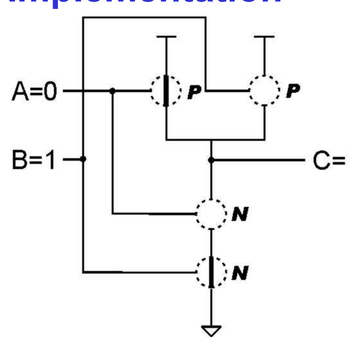
- Current flows if either is “ON”

CMOS is naturally inverting

Result: n-network implements function

NAND example

- n-network transistors in series gives AND
- Natural inversion gives NAND



A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

80

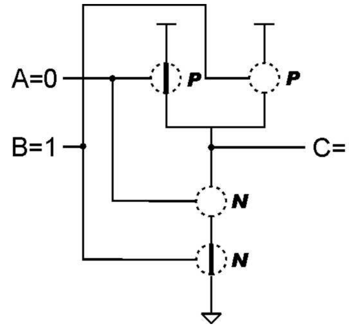
80

The "Logic" Behind CMOS Gate Implementation

P-network is complement of n-network
- Series n-network → parallel p-network
- Parallel n-network → series p-network

NAND example
- p-network transistors in parallel

- Designing in CMOS:
  - We always design the n-network (aka – the pull-down network) first
  - Then, complement it and you've figured out the p-network (aka – the pull-up network)



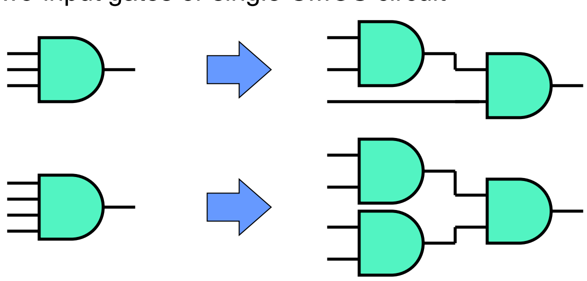
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

81

81

More than 2 Inputs? Arbitrary Functions?

- AND/OR can take any number of inputs
  - AND = 1 if all inputs are 1
  - OR = 1 if any input is 1 (0 if all inputs are 0)
- Implementation
  - Multiple two-input gates or single CMOS circuit

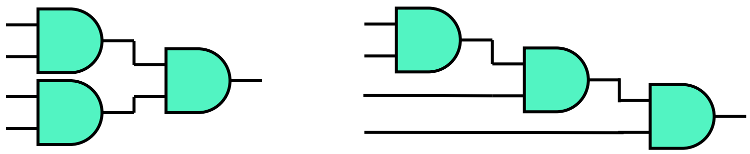


- Can implement arbitrary boolean functions as a gate
  - More complex n- and p- networks

82

82

Gate Delays



- Which is the better implementation of 4-input AND?
  - One on the left
  - Why? It's faster, 2 "gate delays" instead of 3
- Gate delays: longest path (in gates) through a circuit
  - Grossly over-simplified, ignores gate differences, wires
  - Good enough for our purposes

83

83