CH8: Logic Gates and Transistors

Chapter Goals

- Logic gates
  - Combinational circuits
  - Sequential circuits – register, memory
- Transistor
  - Basic transistor operation
  - Logic gates using transistor
- Physical manufacturing

Today

- Abstraction of logic gates
- Notation
- Simple logic gates
- Combining logic gates
- Creating our own circuits

Logic gate abstraction

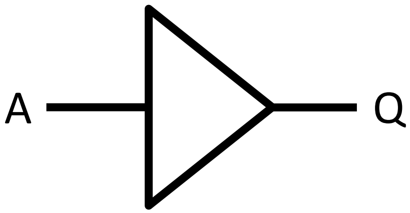
- Simple type of circuit block
- All inputs are one bit
- Typically every logic gate has one output

- Symbol, truth-table, text-notation

- Boolean equations and schematics are equivalent representations of the same thing

Basic Logic Gates

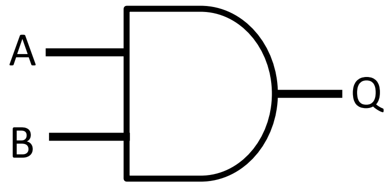
NOT GATE



A	Q
0	1
1	0

Text notation: ~

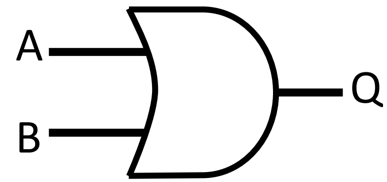
AND GATE



A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Text notation: ·

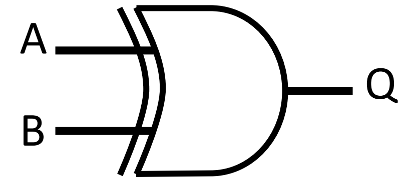
OR GATE



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Text notation: +

XOR GATE

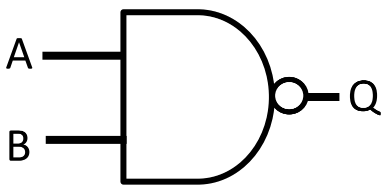


A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Text notation: ⊕

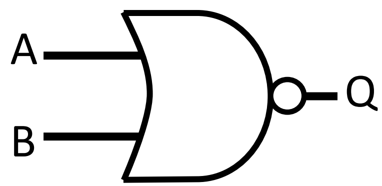
NAND, NOR, XNOR gate

NAND GATE



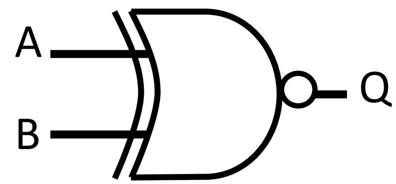
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

NOR GATE



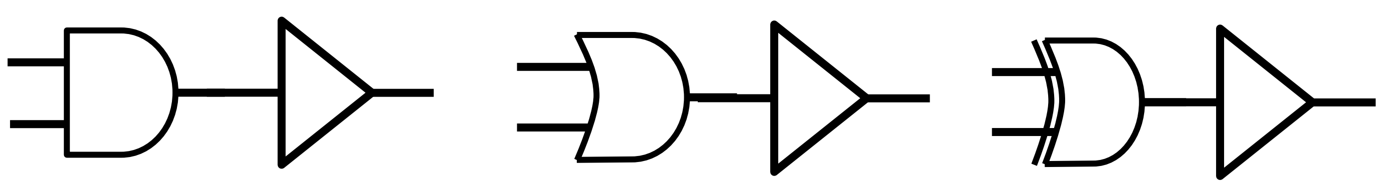
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

XNOR GATE



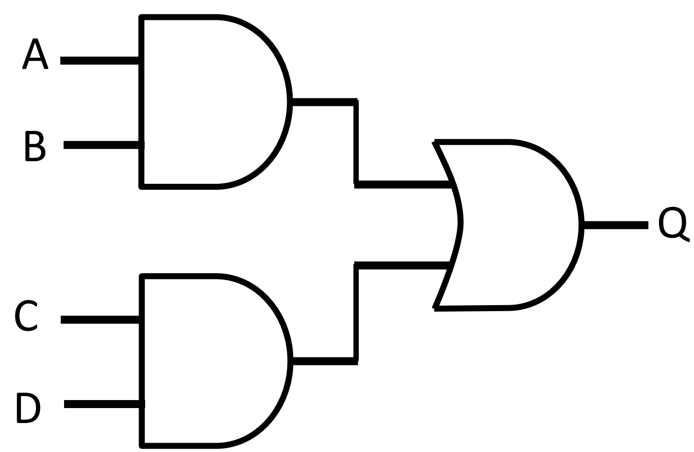
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Using other gates



Combining logic gates

- Output of logic gate can be connected to any number of other inputs of other logic gates



- What does this circuit do?

What does a circuit do?

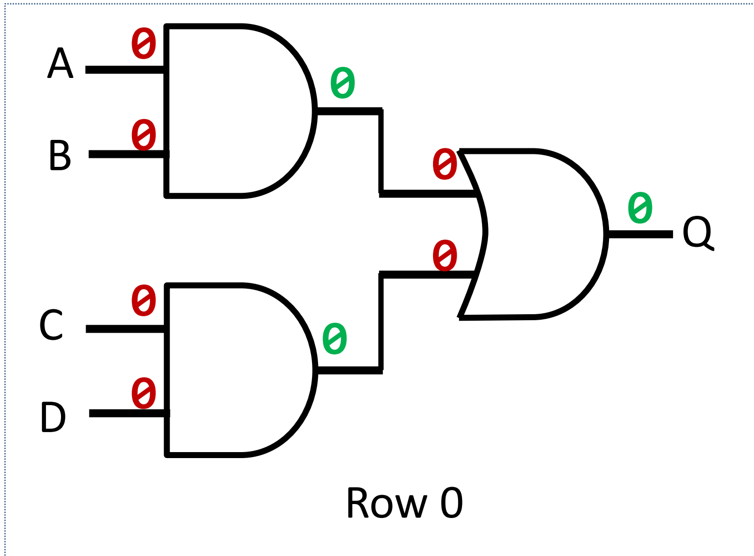
- Write down the truth table for all possible inputs

OR

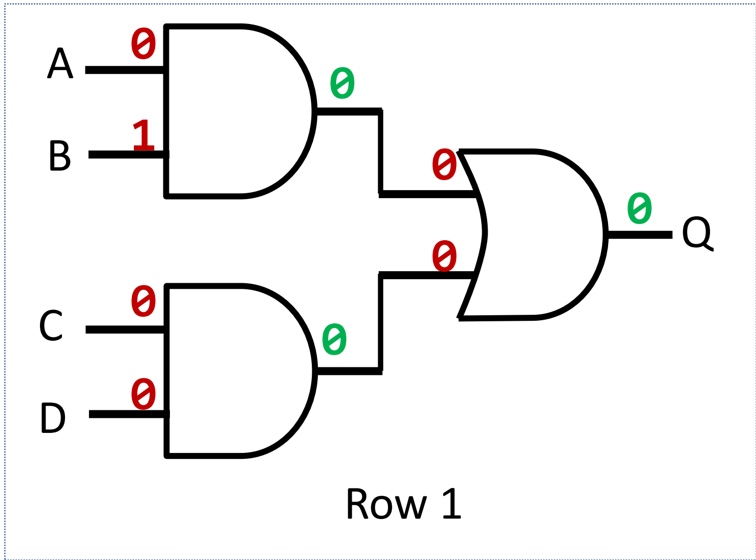
- Write the Boolean equation for it

OR

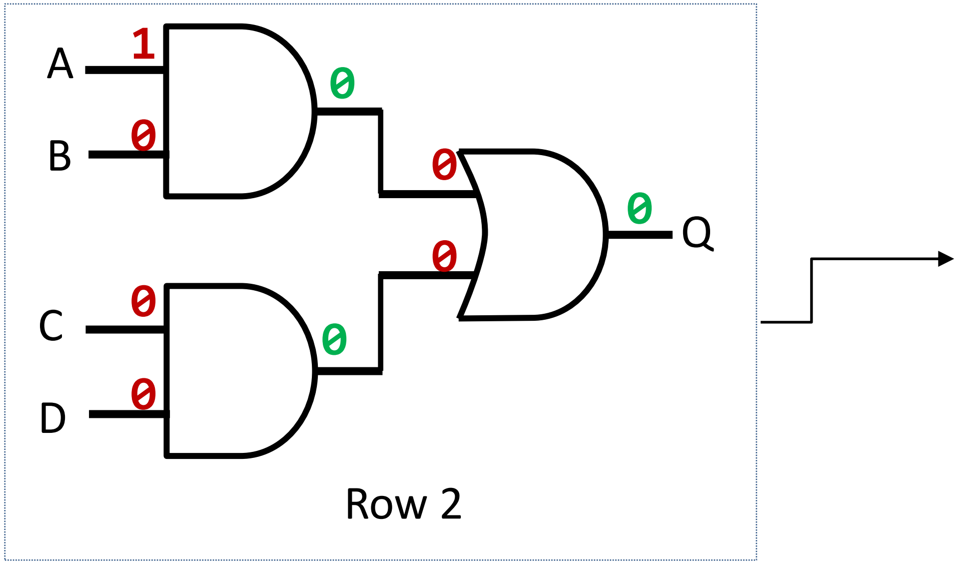
- Write the “minimized” logic equation



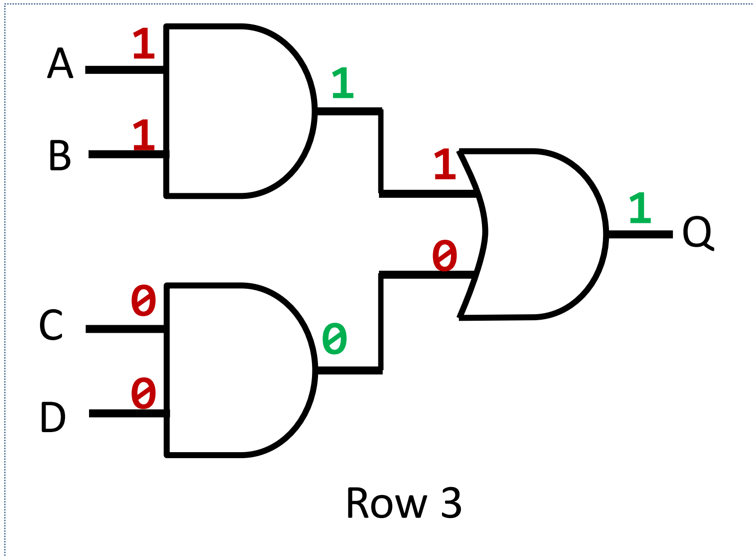
C	D	A	B	Q
0	0	0	0	0



C	D	A	B	Q
0	0	0	0	0
0	0	0	1	0



C	D	A	B	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0



C	D	A	B	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1

Creating circuit given truth table

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Creating circuit given truth table

A	B	Q
0	**0**	**1**
0	1	1
1	0	1
1	1	0

NOT(A)   AND NOT (b)

Creating circuit given truth table

A	B	Q
0	0	1
0	**1**	**1**
1	0	1
1	1	0

NOT(A)   AND (b)

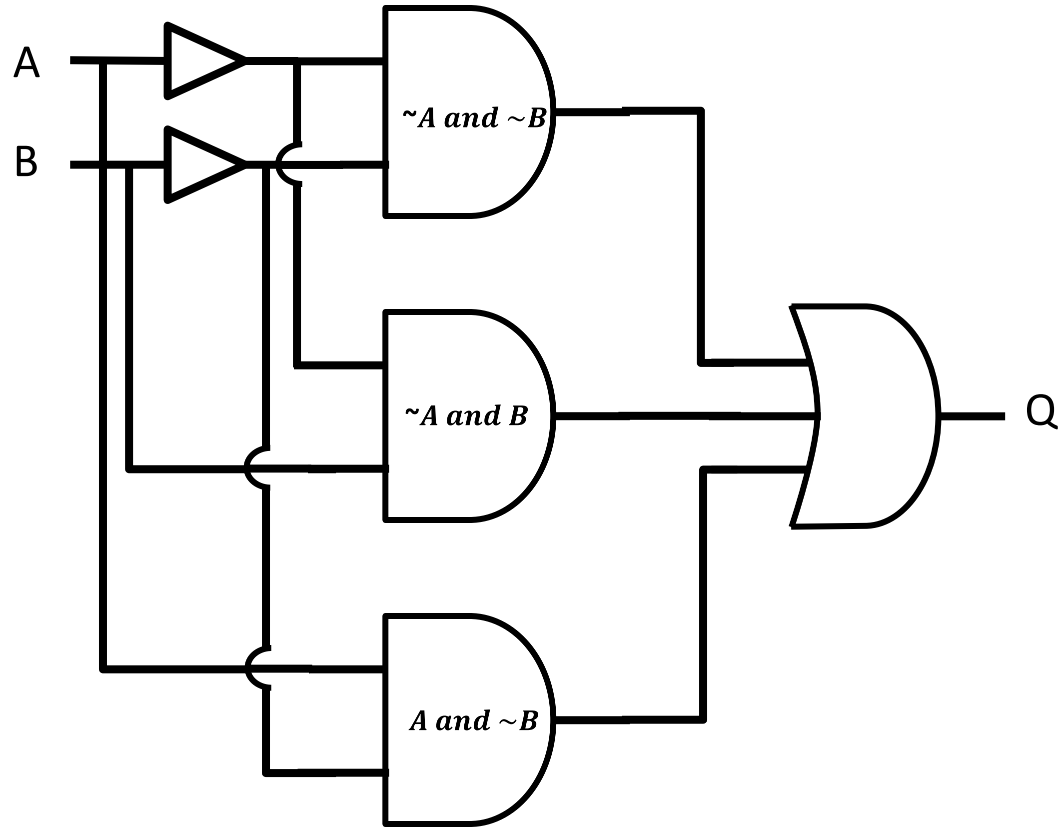
Creating circuit given truth table

A	B	Q
0	0	1
0	1	1
1	**0**	**1**
1	1	0

A AND NOT (b)

Creating Circuit given Truth Table

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



Algorithm

- For each row which has 1
  1. Find columns which are 0, negate that variable
  2. Find columns which are 1, keep that variable
  3. AND all the terms from (1) and (2) above for that row
- OR the terms across all the rows which are 1

We can build circuit blocks now

We can build circuit blocks now

State number	State name	PC_sel	PC_we	INST_we	REG_sel	REG_we	OFF_we	OFF_sel	REG2_we	RF_sel	RF_we	VAL1_we	VAL2_we	VAL2_sel	ADDR_we	A_sel	B_sel
00000	INST = PM[PC]		0	1		0	0		0		0	0	0		0		
00001	REG = 1,INST[7:4]		0	0	0	1	0		0		0	0	0		0		
00010	REG = INST[8:4]		0	0	1	1	0		0		0	0	0		0		
00011	OFF = Z?SEXT16(INST[9:3]):0		0	0		0	1	1	0		0	0	0		0		
00100	OFF = SEXT16(INST[11:0])		0	0		0	1	0	0		0	0	0		0		
00101	VAL = INST[11:8],INST[3:0]		0	0		0	0		0		0	0	0		0		
00110	VAL = RF[REG]		0	0		0	0		0	0	0	0	0		0		
00111	VAL1 = RF[REG]		0	0		0	0		0	0	0	1	0		0		
01000	VAL2 = RF[REG2]		0	0		0	0		0	1	0	0	1	0	0		
01001	ADDR = X		0	0		0	0		0		0	0	0		1		
01010	REG2 = INST[9],INST[3:0]		0	0		0	0		1		0	0	0		0		
01011	VAL = VAL1 + VAL2		0	0		0	0		0		0	0	0		0	1	0
01100	VAL = VAL1 - VAL2		0	0		0	0		0		0	0	0		0	0	0
01101	Update SREG		0	0		0	0		0		0	0	0		0		
01110	VAL = OFF+PC		0	0		0	0		0		0	0	0		0	0	1
	VAL2 =																

We can build circuit blocks now

State number	State name	PC_sel
00000	INST = PM[PC]	
00001	REG = 1,INST[7:4]	
00010	REG = INST[8:4]	
00011	OFF = Z?SEXT16(INST[9:3]):0	
00100	OFF = SEXT16(INST[11:0])	
00101	VAL = INST[11:8],INST[3:0]	
00110	VAL = RF[REG]	
00111	VAL1 = RF[REG]	
01000	VAL2 = RF[REG2]	
01001	ADDR = X	

We can build circuit blocks now

$S_4S_3S_2S_1S_0$	**PC_sel**
00000	
00001	
00010	
00011	
00100	
00101	
00110	
00111	
01000	
01001	
01010	
01011	
01100	
01101	
01110	
01111	
10000	
10001	
10010	
10011	**1**
10100	0

PC_sel = $S_0S_1 \sim S_2 \sim S_3S_4$

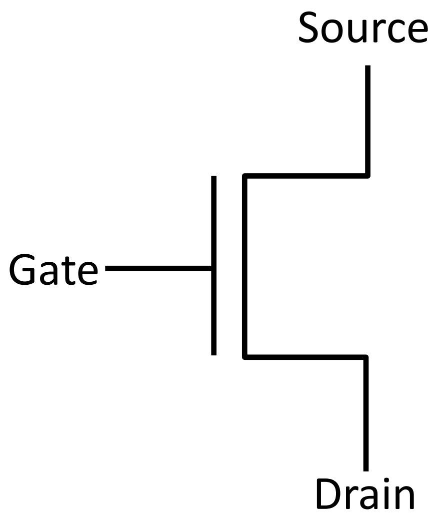
We can build circuit blocks now

$S_4S_3S_2S_1S_0$	PC_we
00000	
00001	
00010	
00011	
00100	
00101	
00110	
00111	
01000	
01001	
01010	
01011	
01100	
01101	
01110	
01111	
10000	
10001	
10010	
10011	**1**
10100	**1**

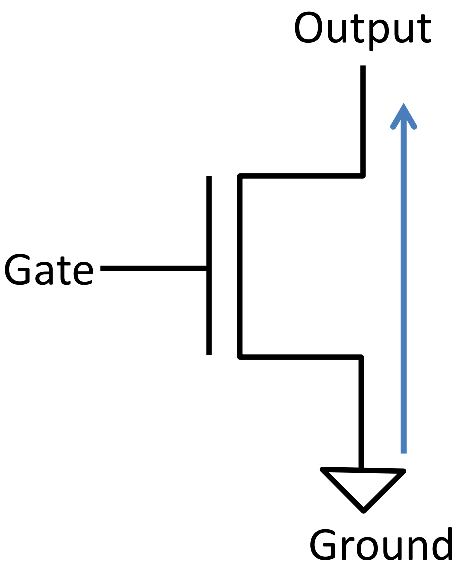
PC_we = $S_0S_1 \sim S_2 \sim S_3S_4 \; OR \; \sim S_0 \sim S_1 \, S_2 \sim S_3S_4$

What is a gate physically?

Transistor

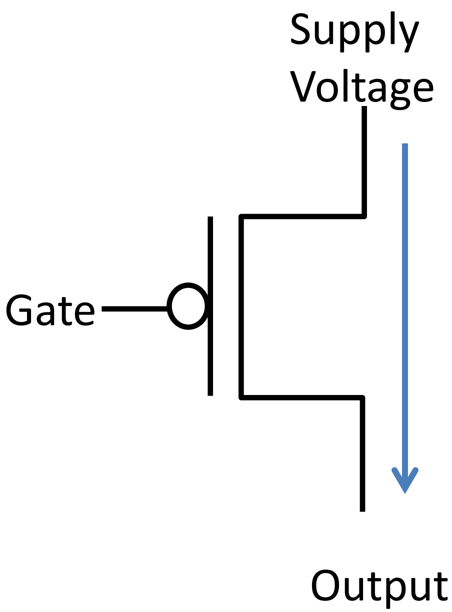


Transistor terminals



N-type transistor

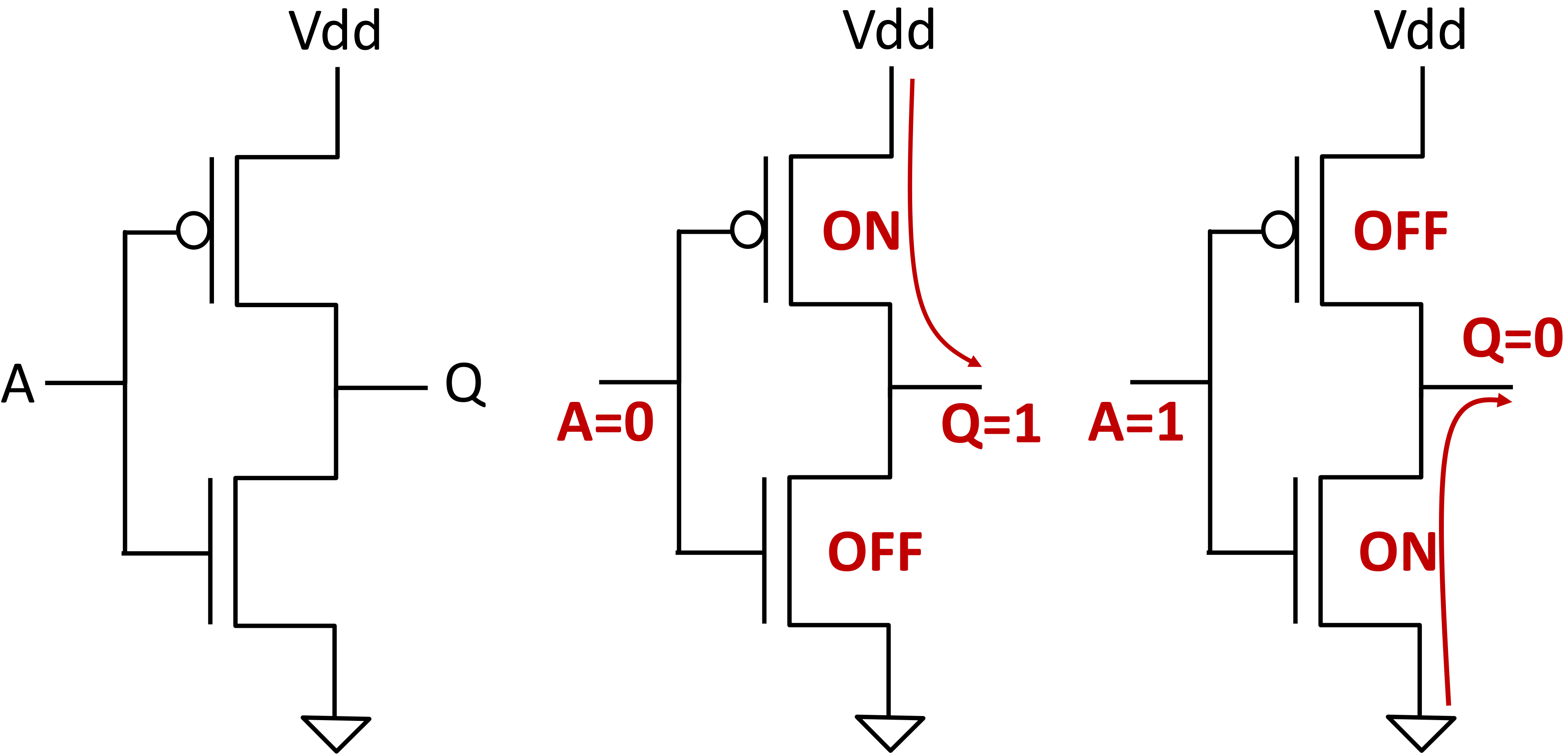
Gate	Behavior
1	Closed Output=0
0	Open Output=Z



P-type transistor

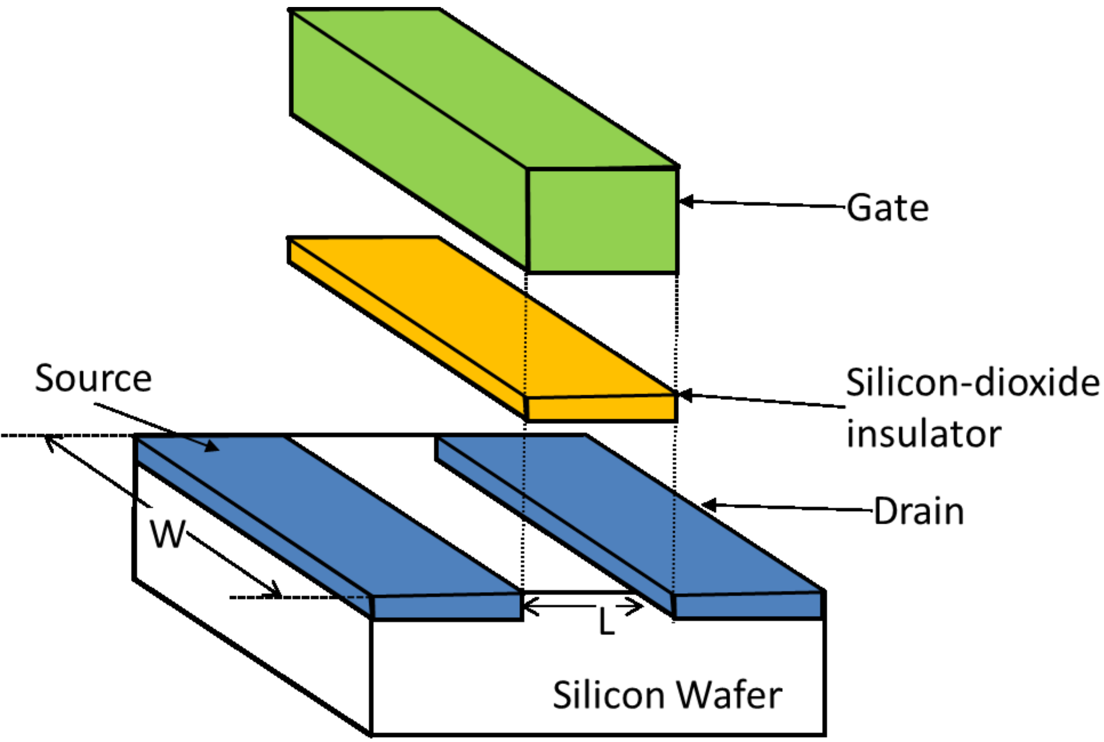
Gate	Behavior
0	Closed Output=1
1	Open Output=Z

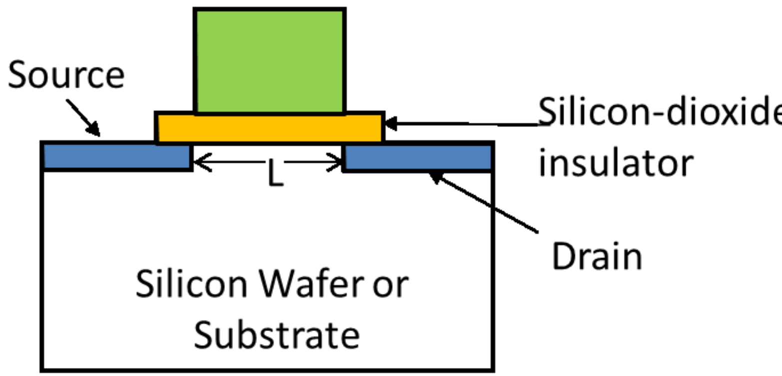
NOT Gate with Transistors



What is it really?

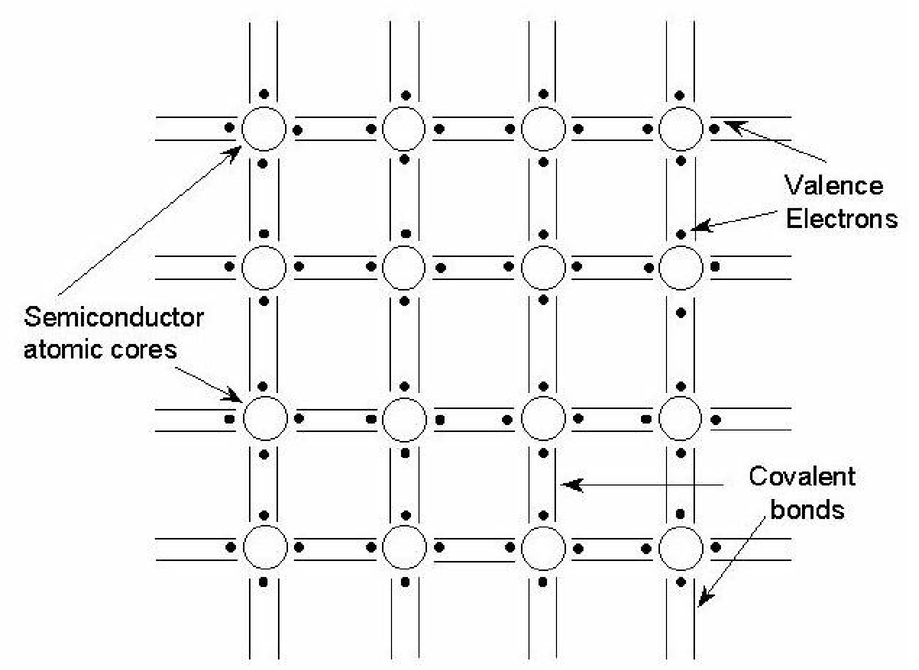
Transistor physical diagram





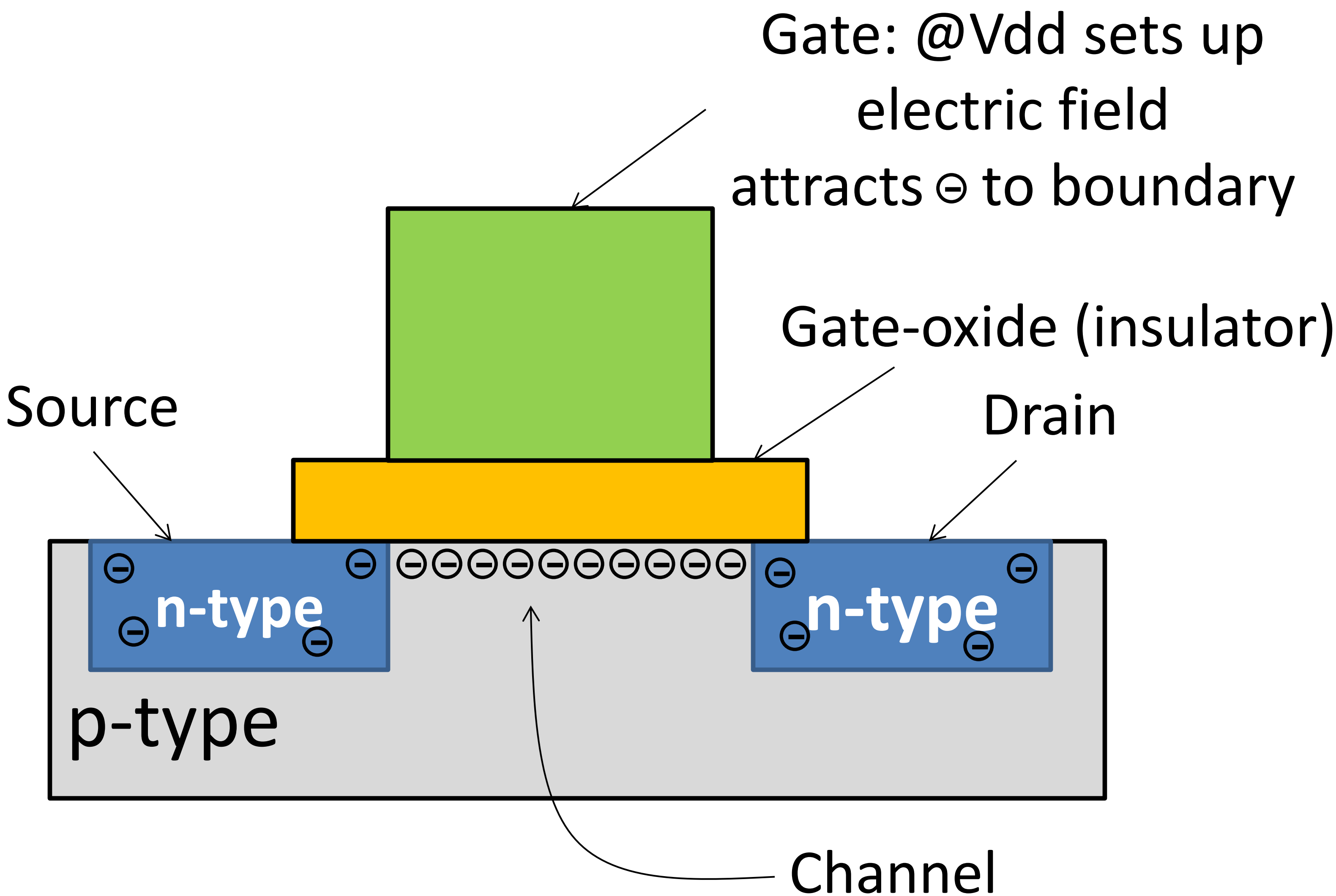
Side view

- Silicon by itself is semiconductor
  - Has 4 valence electrons and forms a co-valent bond with 4 neighbors
  - Hard to break and does not conduct ☹



- Silicon by itself is insulator
  - Has 4 valence electrons and forms a co-valent bond with 4 neighbors
  - Had to break and does not conduct ☹
- Silicon + Arsenic makes it filled with electrons and conducting
- Silicon + Boron makes it filled with “holes” and conducting

Transistor Operation



Chapter Goals

- Logic gates
  - Combinational circuits
  - Sequential circuits – register, memory
- Transistor
  - Basic transistor operation
  - Logic gates using transistor
- Physical manufacturing

CS 252

Lecture 28; 2015 Nov 18th; Transcribed Lecture notes

Announcements

- Chapter 8 is released, but a few figures still need to be updated
- Chapter 8 is not in exam 3
- Homework 7 due on Friday, November 20th
- Exam 3 on Monday, November 23rd

Outline

- Chapter 8 Logic Gates
- Chapter 8 Transistors

Questions

- When a register outputs something and the register is not updated on the next instruction,
    - does the register output the same thing on the next clock cycle? Yes.

Chapter Goals

- Logic gates
- Transistors
- Physical manufacturing (maybe some chemistry)

Logic Gates

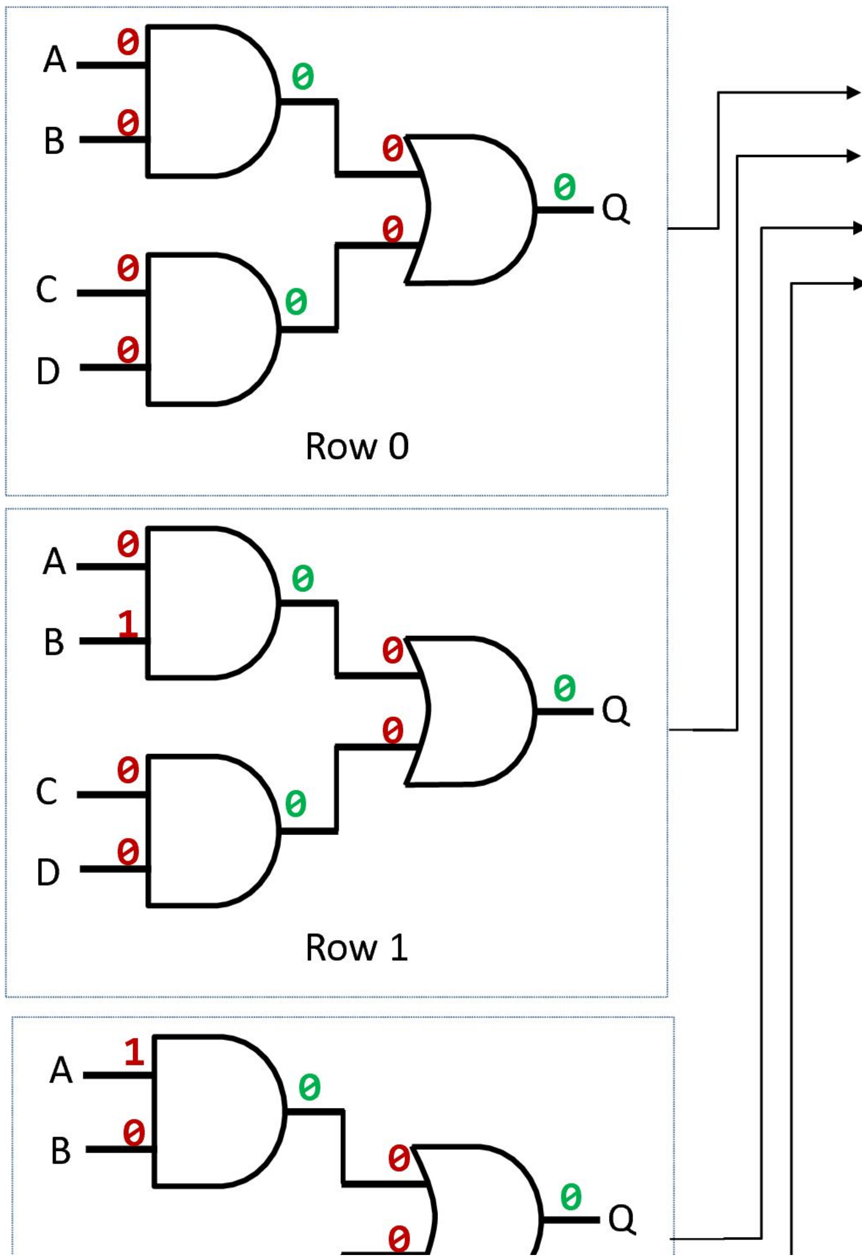
- definition
    - a circuit block
    - all inputs are one bit
    - every logic gate has one output
    - boolean equations and schematics are equivalent representations
- basic logic gates (see truth tables for full description)
    - NOT gate (~) is set if the input is low
    - AND gate ($\cdot$) is set if both inputs are high
    - OR gate (+) is set if either inputs are high
    - XOR gate ($\oplus$) is set if exactly 1 input is high
    - NAND gate is set if either inputs are low
    - NOR gate is set if both inputs are low
    - XNOR gate is set if the inputs are the same
- combining logic gates
    - output of logic gates can be connected to inputs of other logic gates
    - determining what combined gates do

use truth table

write boolean equation

write minimized loic equation

example 1: gates to truth table



C	D	A	B	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

example 2: truth table to boolean equation

A	**B**	**Q**	**max term**
0	0	1	~A * ~B
0	1	1	~A * B
1	0	1	A * ~B
1	1	0	0

Q = (~A * ~B) + (~A * B) + (A * ~B) + 0

We can build circuit blocks now

Let's look at the large truth table at the end of chapter 7.

- Every output has to be independent on all other outputs
- We can use the state number as the input and control signals as output
    - Example 1: PC_sel is high at state 10011
        - PC_sel = $S_4 * \sim S_3 * \sim S_2 * S_1 * S_0$
    - Example 2: PC_we is high at state 10011 and 10100
        - PC_we = $(S_4 * \sim S_3 * \sim S_2 * S_1 * S_0) + (S_4 * \sim S_3 * S_2 * \sim S_1 * \sim S_0)$

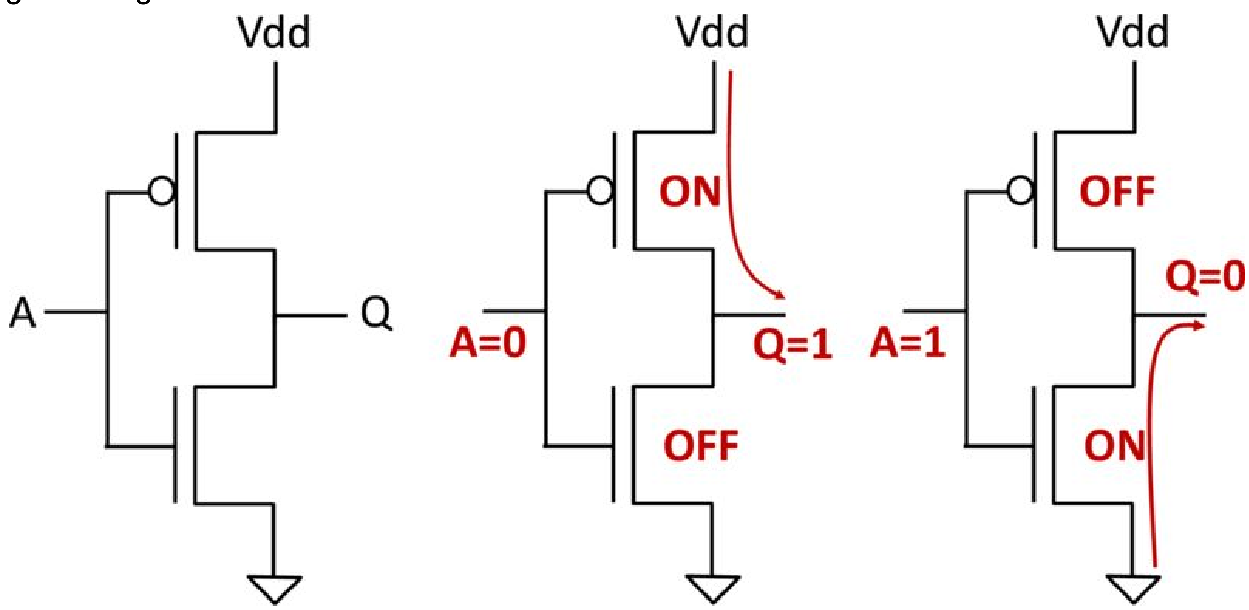
What is a logic gate physically?

Transistors

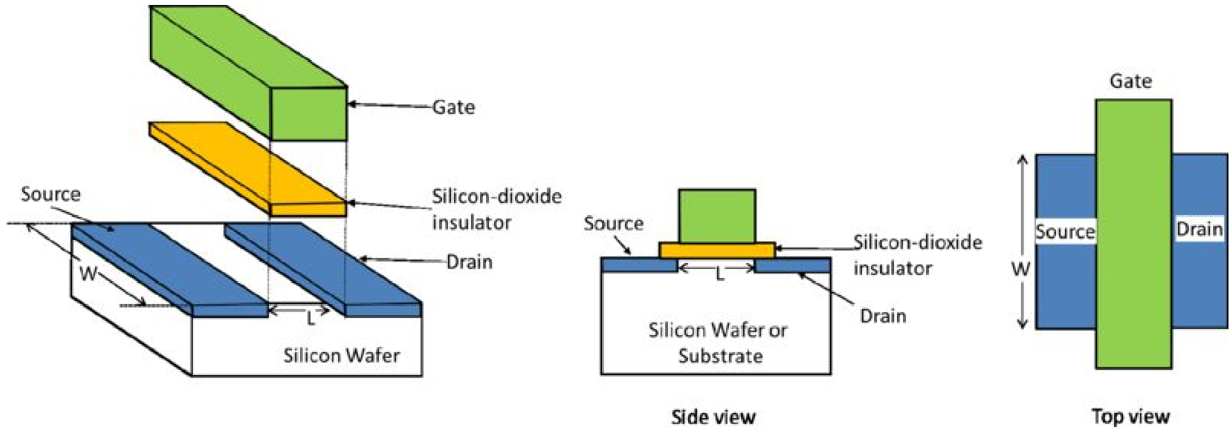
Definition

- Transistors has a source, gate, and drain
- N-type transistor
    - If the gate is 1, then the switch is closed/connected
    - If the gate is 0, then the output is undefined
    - One terminal of P-type transistor is connected to ground
- P-type transistor
    - If the gate is 0, then the switch is closed/connected
    - If the gate is 1, then the output is undefined
    - One terminal of P-type transistor is connected to high voltage

Building a NOT gate with transistors



Make up



- Transistor size is the length
- The width is usually set to be twice the length
- Every 2 years, manufacturers continue to find a way to decrease the length and width
- Moore's Law: the amount of transistors on a given silicon area double every 2 years.
- The semiconductor industry is in a race to follow Moore's law
- Smaller transistors also means lower power consumption

Chemistry

- Important to understand how transistors work. Will NOT be covered on exams.
- Silicon is an insulator (4 valence electrons and does not conduct)
- Silicon + Arsenic makes it filled with electrons and conducting
- Silicon + Boron
- Hopping valence bands to conduct electricity

Transistor creation and operation

- Start with Silicon. Lightly dope it with Boron to make a p-type material
- Carve out 2 n-type regions doped with Arsenic
- Apply voltage to set up an electric field so that the p-type substrate attach electrons
- There is now a path for electrons to flow between the n-type bands
- Electron mobility is the measure of how long the electrons take to move between n-type
    - wells
- Can use electron mobility to determine exact timing results

Transistor history

- Moore's law makes the length shorter, which allows the transistor to switch faster
- Problem if length is too short: Avalanche effect of current to flow from gate to p-type
    - this causes leakage power, which is inefficient use of standby power
- Gate-oxide is switch to high K material to set up high capacitive field
    - Silicon gate had to change to metal gate to make the connection