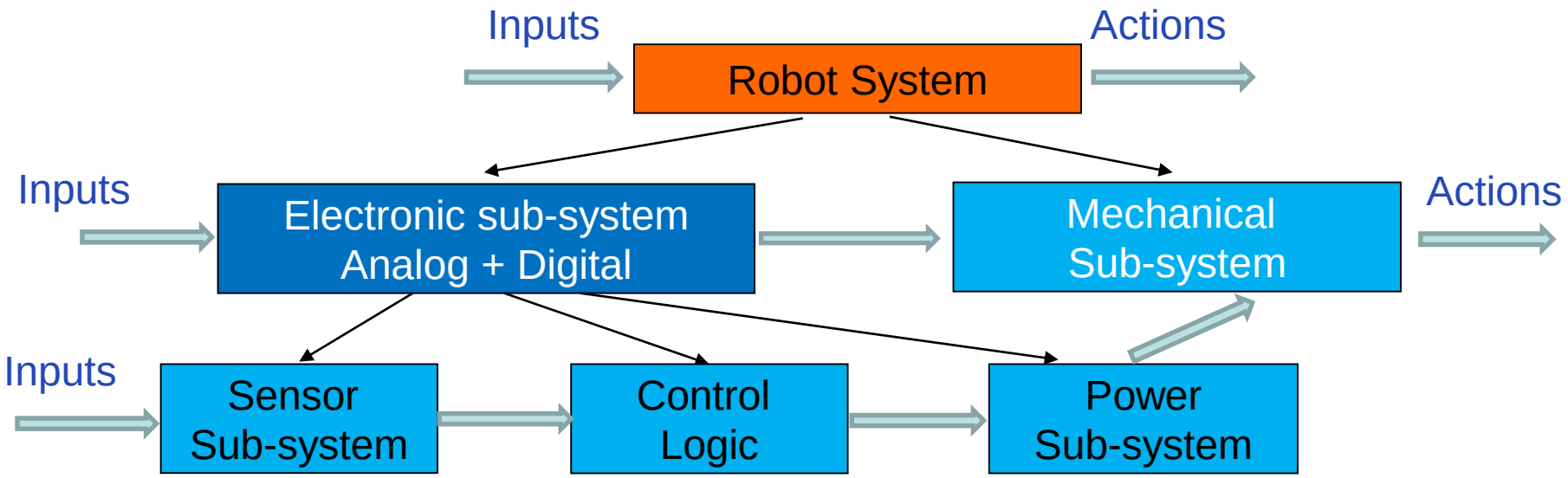
ELEC 1100: Introduction to Electro-Robot Design

Lecture 04: Logic + K-map + Midterm Review

CHEN Wei, Dept. of ECE, HKUST

ELEC1100 ROADMAP



Sensor Basics:
Wk2: Sensor Basic

Combinational/Sequential Logic:
Wk3: Robot Brain: Logic Gate & Logic Operation

Basic electronics:
Wk1: Basic Electronics & DC Sources

Motor Power Supply:
Wk1: Pulse Signal & PWM Control
Wk2: Transistor and H-Bridge

BINARY NUMBER

- Decimal number system – base 10, each digital is coming from the set {0,1,2,3,4,5,6,7,8,9}
- Binary number system – base 2, each digital is coming from the set {0,1}

Multiplication and addition of binary number						
+	0	1		*	0	1
0	0	1		0	0	0
1	1	0		1	0	1

Base 10	Base 2
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010

BINARY NUMBER OPERATION

❖ Addition

Decimal	Binary
7	0111
+ 5	+ 0101
12	1100

❖ Multiplication

Decimal	Binary
7	0111
× 5	× 0101
35	0111
	0111
	100011

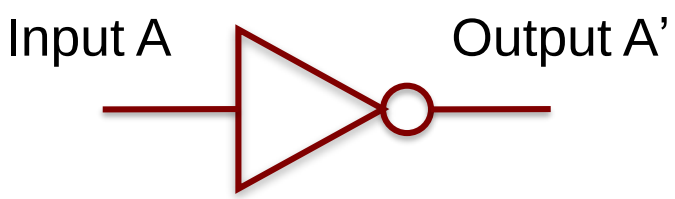
- ❖ Binary digit: 0 and 1 can be represented by logic (True or False)
  - ➢ 0 is equivalent to False
  - ➢ 1 is equivalent to True
- ❖ Use Boolean Algebra (which operates on {T,F}) to manipulate the binary digit operation

LOGIC GATES

- Two values for logic: True (“T”) and False (“F”)
- A logic input can be combined with another logic input in different ways to form a new logic output. We call this combination of the inputs as logic gates.
- There are seven fundamental logic gates:
  - Inverter (Not) – 1 input, 1 output
  - AND – 2 or more inputs, 1 output
  - NAND – 2 or more inputs, 1 output
  - OR – 2 or more inputs, 1 output
  - NOR – 2 or more inputs, 1 output
  - XOR – 2 or more inputs, 1 output
  - XNOR – 2 or more inputs, 1 output

TRUTH TABLE

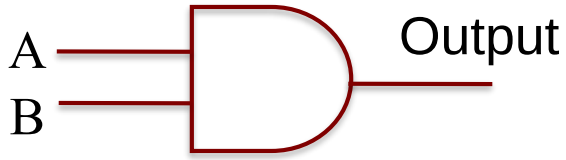
- A tabular summary for all the possible outputs of a logic gate, given all the possible input values
- For a logic gate that has n inputs, how many possible input combinations do we have?
- Sometimes we use 0 to represent F and 1 to represent T
- Truth table of an inverter



Input A	Output A'
0	1
1	0

AND/OR GATES

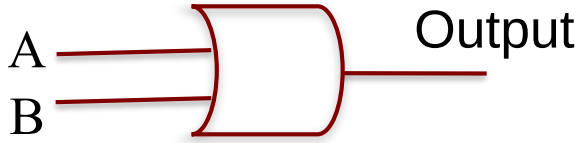
❖ AND gate: Output = A•B



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

e.g. if tmr is sunny and I finish ELEC 1100 hw, I'll go hiking.

❖ OR gate: Output = A+B

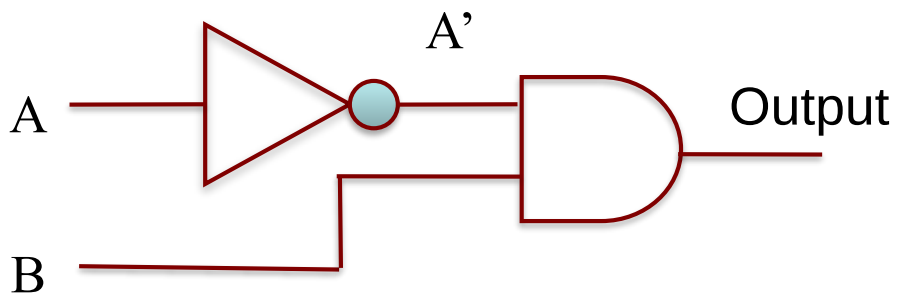


A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

e.g. if it rains tmr or I cannot finish ELEC 1100 hw, I'll stay home.

EXAMPLE OF CIRCUIT WITH AND GATE

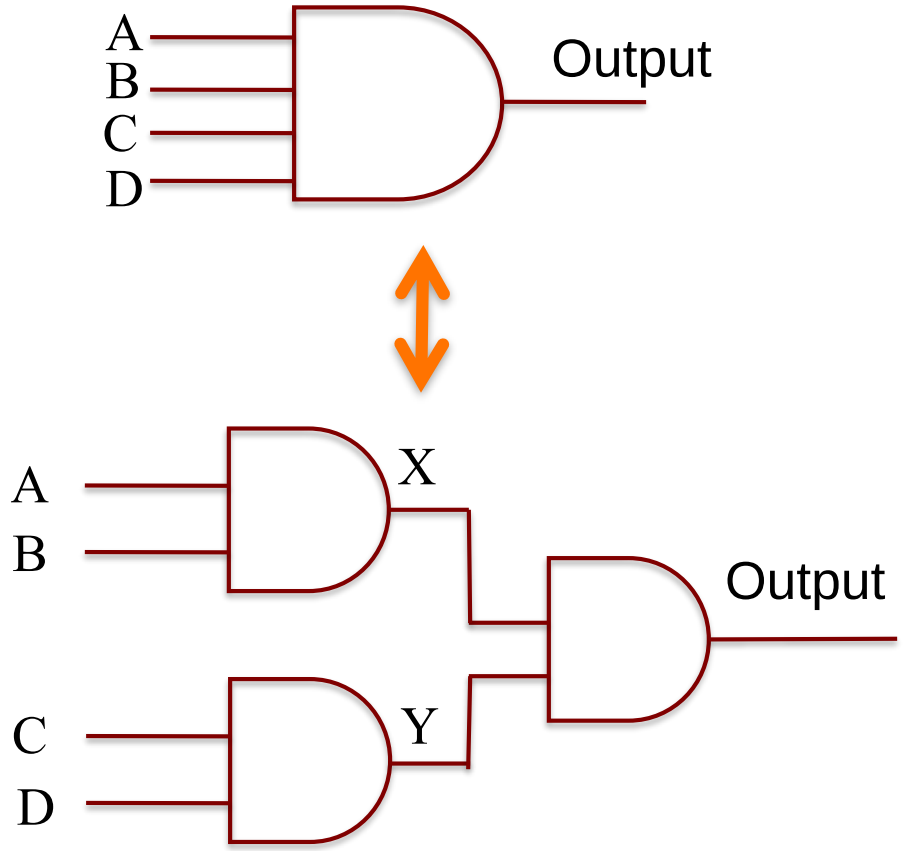
❖ e.g. Output = A'•B



Q: what's the truth table?

A	B	A'	Output
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0

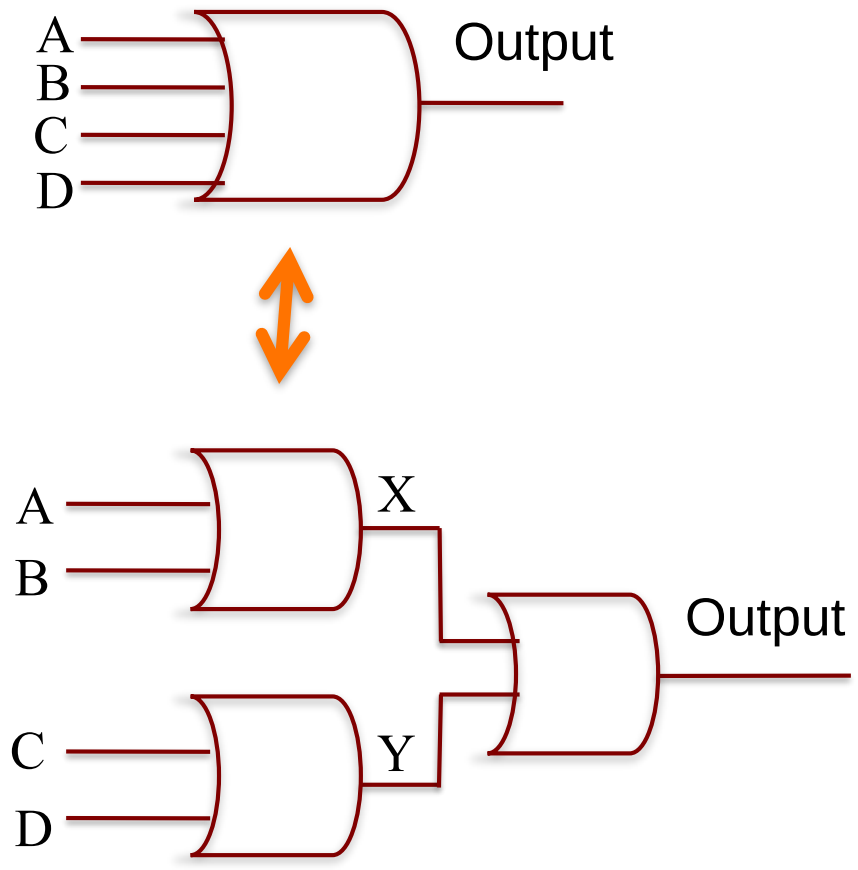
MORE EXAMPLE (AND GATE)



4-input AND gate

A	B	C	D	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

MORE EXAMPLE (OR GATE)

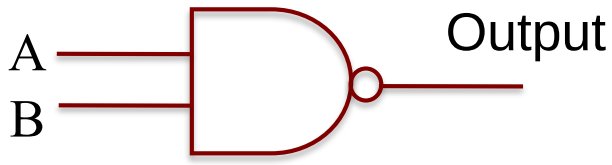


4-input OR gate

A	B	C	D	Output
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
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

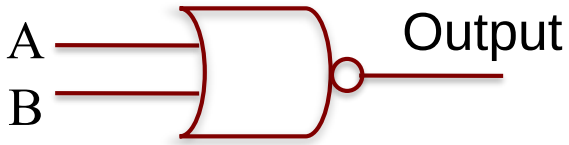
NAND/NOR GATES

❖ NAND gate: Output = $\overline{A \bullet B}$



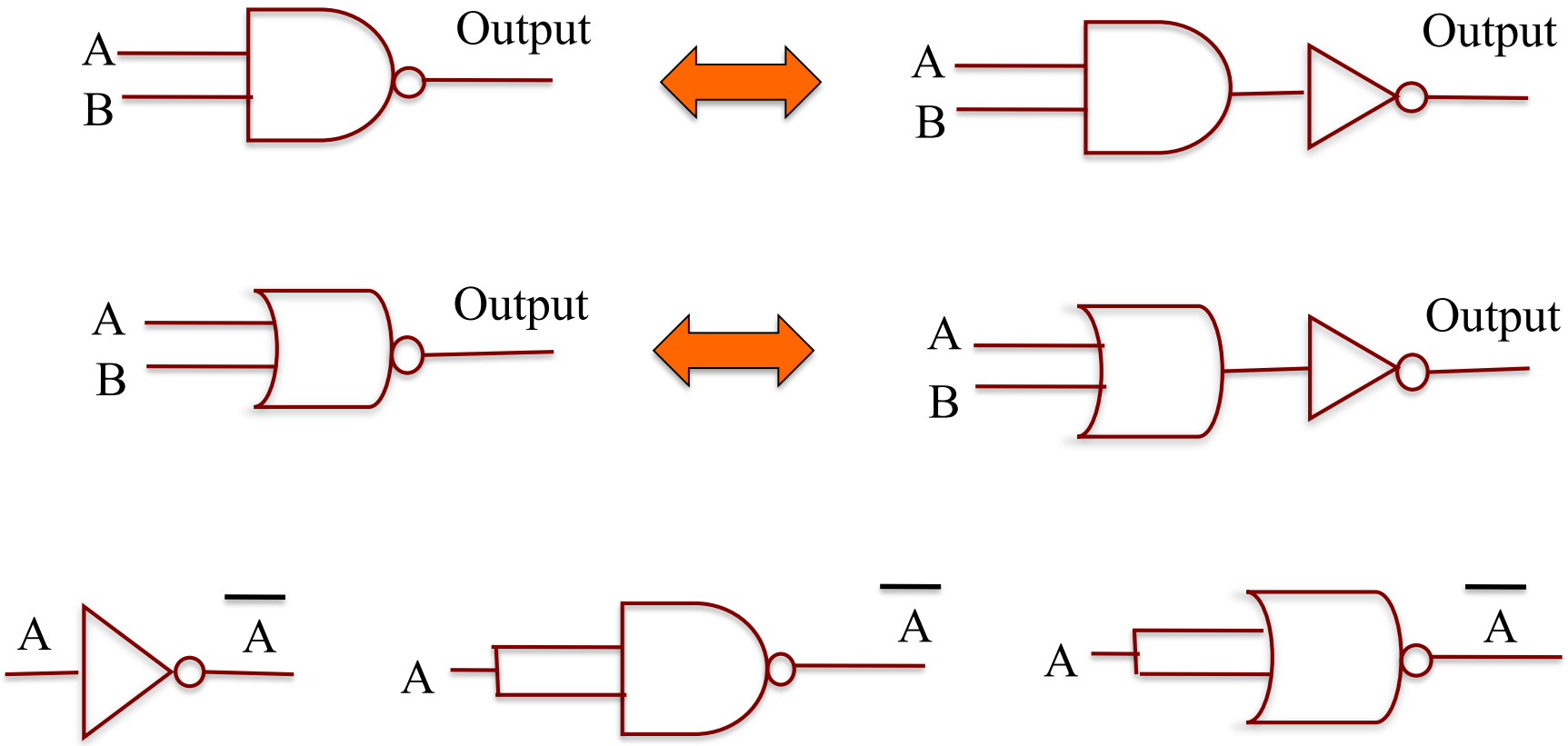
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

❖ NOR gate: Output = $\overline{A+B}$



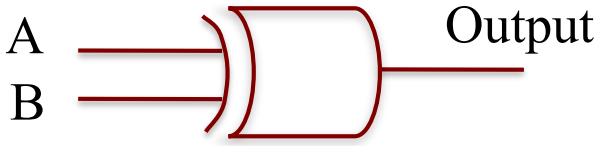
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

SOME EQUIVALENT GATES



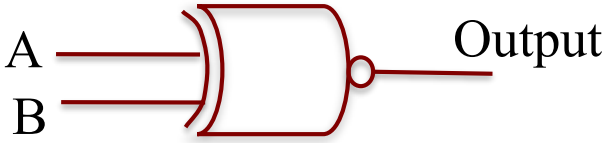
XOR/XNOR GATES

❖ XOR gate: Output = $A \oplus B$



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

❖ XNOR gate: Output = $\overline{A \oplus B}$



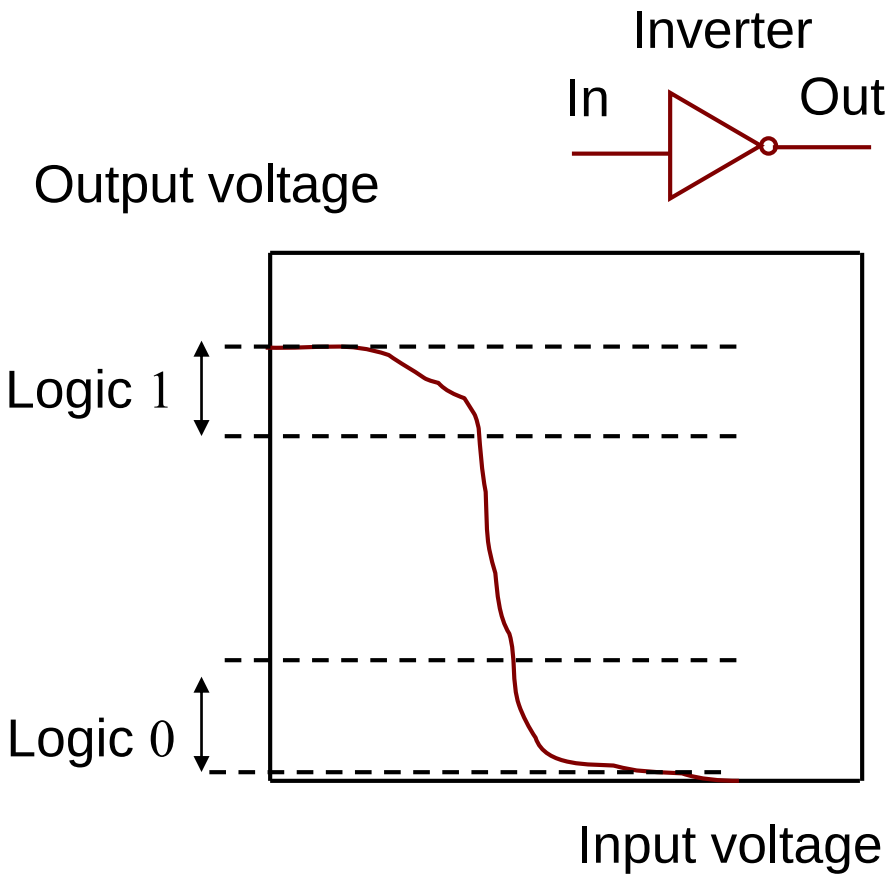
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

LAW OF BOOLEAN ALBEGRA

- $0+X = X$
- $1+X = 1$
- $X'+X = 1$
- $0 \bullet X = 0$
- $1 \bullet X = X$
- $X \bullet X = X$
- $X \bullet X' = 0$
- $(X')' = X$
- $X+Y = Y+X$
- $X \bullet Y = Y \bullet X$
- $X+(Y+Z) = (X+Y)+Z$ (Associativity)
- $X \bullet (Y \bullet Z) = (X \bullet Y) \bullet Z$ (Associativity)
- $X \bullet (Y+Z) = X \bullet Y + X \bullet Z$ (Distributivity)
- $X+X \bullet Z = X$
- $X \bullet (X+Y) = X$
- $(X+Y) \bullet (X+Z) = X+Y \bullet Z$
- $X'+XY = X'+Y$
- $(XY)'+(YZ)'+(XZ)' = (XY)'+(XZ)'$
- $(X+Y)' = X' \bullet Y'$
- $(XY)' = X'+Y'$ (DeMorgan's Law)

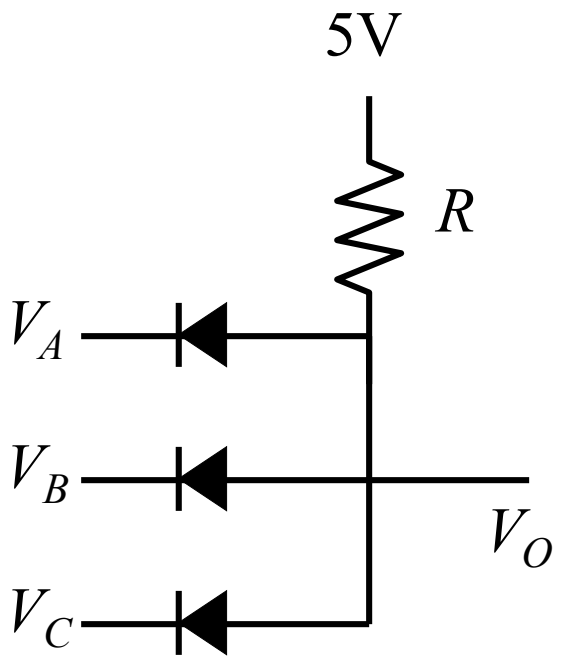
VOLTAGE AND LOGIC VALUE

- How do we represent binary digital signals?
- We can use a range of voltage values to represent logic 0 or 1
- Sometimes we just use high voltage to represent 1 and low voltage to represent 0



LOGIC GATE CONSTRUCTION WITH DIODES [1]

- AND gate



Input/Output Table

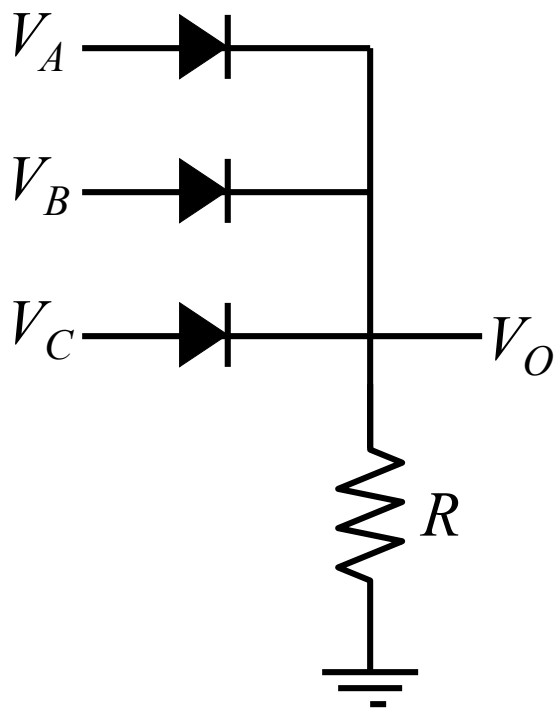
V_A	V_B	V_C	V_O
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$1 = 5\text{V}$

$0 = 0\text{V}$

LOGIC GATE CONSTRUCTION WITH DIODES [2]

- OR gate



Input/Output Table

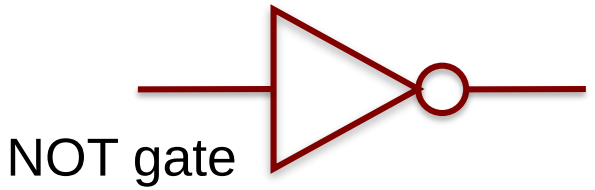
V_A	V_B	V_C	V_O
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

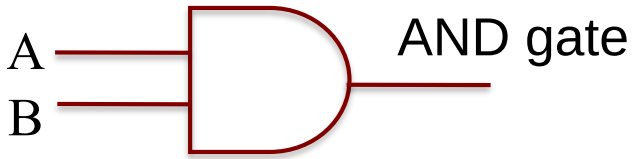
$1 = 5\text{V}$

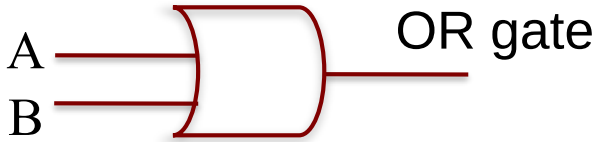
$0 = 0\text{V}$

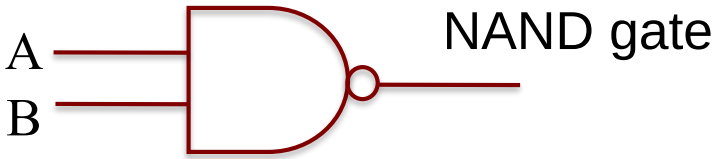
SUMMARY

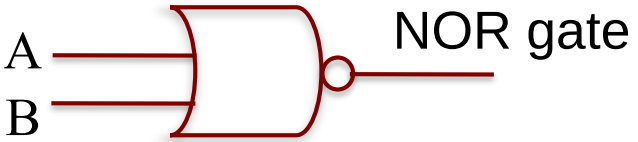
- Binary digit: 0 and 1 can be represented by logic (True or False)
  - 0 is equivalent to False
  - 1 is equivalent to True

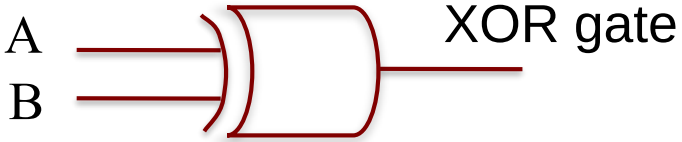


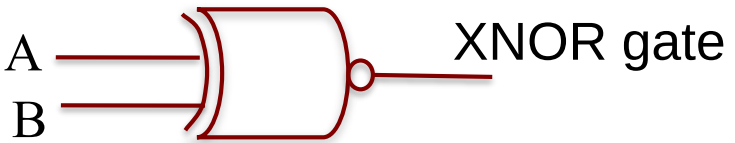






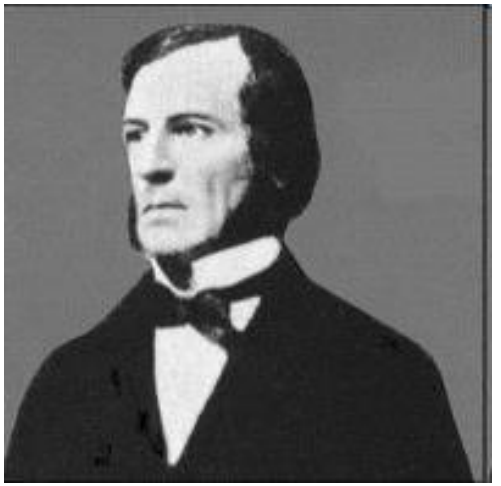






SOME HISTORY [1]

- Boolean algebra was invented by George Boole (1815-1864).
- Boolean Logic is the basis of modern digital computer logic. Boole is regarded as one of the founders of the field of computer science.

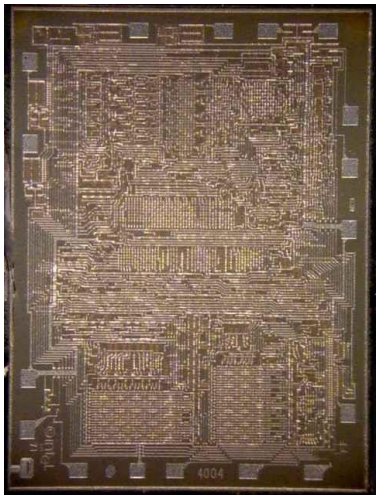


SOME HISTORY [2]

- Claude Shannon notices similarities between Boolean algebra and electronic telephone switches.
- Shannon's 1937 MIT Master's Thesis introduces the world to binary digital electronics.
- It took 30 years to combine Boolean algebra with electronics.
- IBM's first commercially available scientific computer, 1953 (16,000 adds/sec)

SOME HISTORY [3]

- First really pocketable calculator
  - Bowmar 901B, 1971
  - Four functions, 8-digit LEDS
  - Costs 3 weeks average wage in America
- First Microprocessor: Intel 4004, 1971
  - 1K RAM chip and the 4004, a 4-bit microprocessor
  - Clock frequency 740 kHz
  - Two years later comes the 8008, an 8-bit microprocessor
- Pentium 4 @ 3GHz, 2005: 12×10^9 adds/sec



COMBINATIONAL LOGIC: STANDARD FORM

- All Boolean equations can be written in standard forms
- Sum of Products (SOP) – OR'ing (sum) many AND (product) terms

  e.g. $X = A \bullet B + B \bullet C \bullet D + \bar{E} \bullet \bar{F}$

- Product of Sums (POS) – AND'ing (product) many OR (sum) terms

  e.g. $\bar{X} = (\bar{A}+\bar{B}) \bullet (\bar{B}+\bar{C}+\bar{D}) \bullet (E+F)$

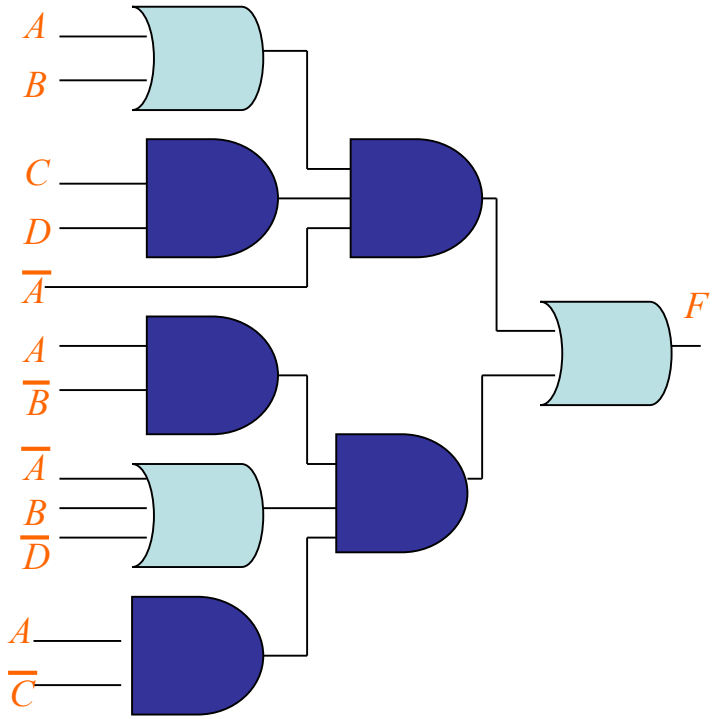
- How do you change from one form to another?

COMBINATIONAL LOGIC AND TRUTH TABLE [1]

- Build the truth table of the following circuit

$$F = (A+B)(C \bullet D)(\bar{A}) + (A \bullet \bar{B})(\bar{A}+B+\bar{D})(A \bullet \bar{C})$$

- Before you build the truth table, can you simplify F ?



COMBINATIONAL LOGIC AND TRUTH TABLE [2]

$$F = (A+B)(C \bullet D)(\bar{A}) + (A \bullet \bar{B})(\bar{A}+B+\bar{D})(A \bullet \bar{C})$$

$$= \bar{A}BCD + A\bar{B}\bar{C}\bar{D}$$

A	B	C	D	$\bar{A}BCD+A\bar{B}\bar{C}\bar{D}$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

BUILDING CIRCUIT FROM TRUTH TABLE

A	*B*	*C*	*F*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

- We can implement using SOP from each '1' at the output term
- Add it up together means OR'ing all these terms together

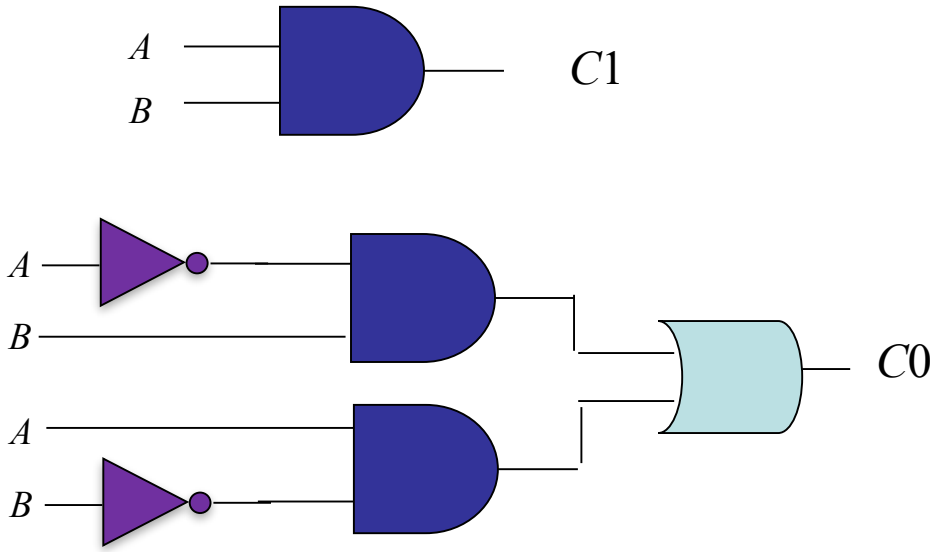
$$F = \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

- We need 5 AND gates, 3 NOT gates and 1 OR gate,
- Can we use fewer gates?
- Is there any systematic way to build a circuit from a truth table?

EXAMPLE: HALF ADDER

- The half adder is a circuit that adds two "1-bit numbers" and result of the addition is a "2-bit number"

Input		Output $C=A+B$	
A	B	$C1$	$C0$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



KARNAUGH MAP (K-MAP)

- K-map can help to convert any Boolean function or truth table into an equivalent SOP form with fewest possible product terms
- Two-variable K-map

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

	$\overline{A}$	A
$\overline{B}$	1	1
B	1	0

K-map

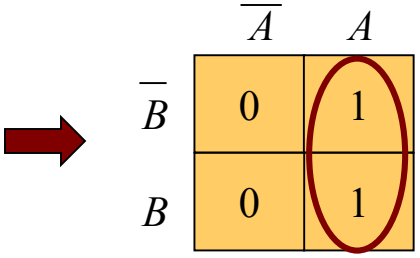
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

	$\overline{A}$	A
$\overline{B}$	1	0
B	0	0

LOGIC MINIMIZATION USING K-MAP

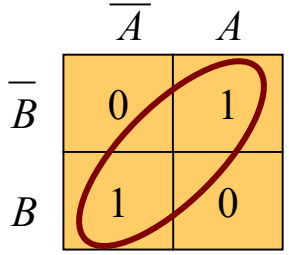
- Begin with isolated cells that no simplification is possible.
- Find all cells that are adjacent to only one other cell, forming two-cell subcubes
- Find cells that form four-cell subcubes, eight-cell subcubes, etc.
- Collect the smallest number of maximal subcubes

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



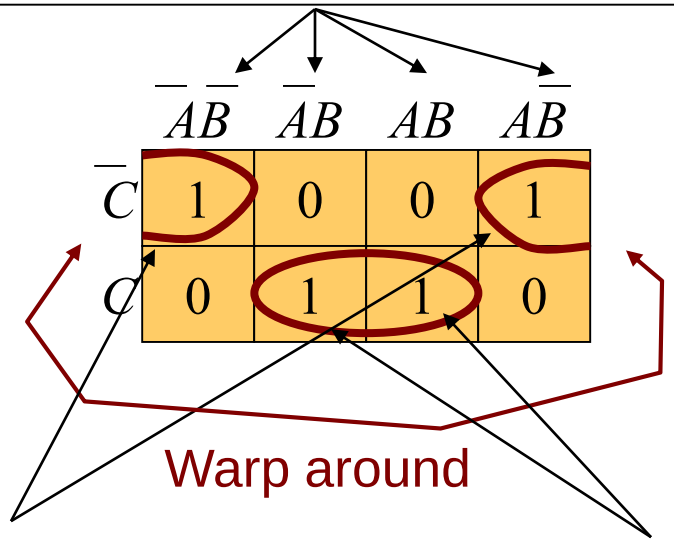
$F = A\overline{B} + AB$ → $F = A$

Can you do this?



K-MAP WITH 3 VARIABLES

Adjacent label should have one and only one variable difference



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

Before circling:

$F = \overline{A}\overline{B}\overline{C}+A\overline{B}\overline{C}$

After circling:

$F = \overline{B}\overline{C}$

Before circling:

$F = \overline{A}BC+ABC$

After circling:

$F = BC$

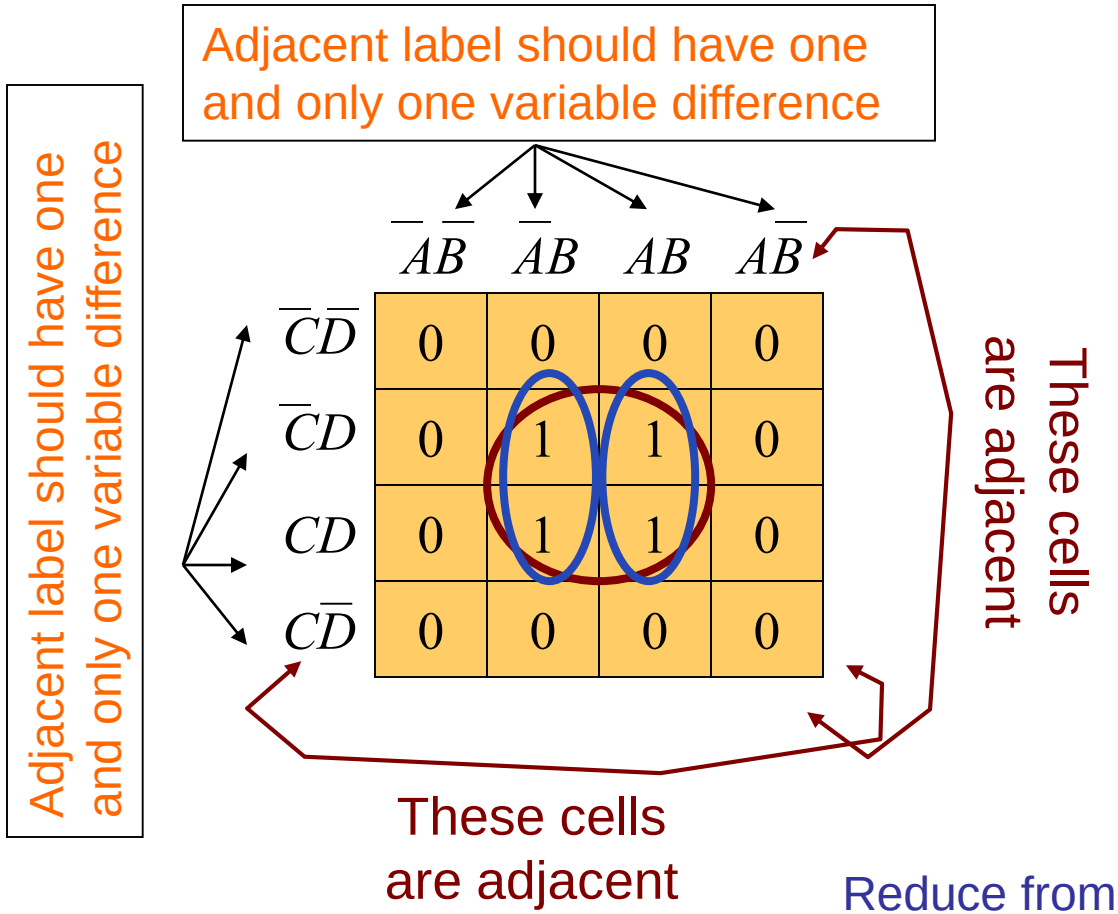
Before circling:

$F = \overline{A}\overline{B}\overline{C}+A\overline{B}\overline{C}+\overline{A}BC+ABC$

After circling:

$F = \overline{B}\overline{C}+BC$

K-MAP WITH 4 VARIABLES



Before circling:

$$F = \bar{A}B\bar{C}D + \bar{A}BCD + AB\bar{C}D + ABCD$$

First circling:

$$F = \bar{A}BD + ABD$$

Second circling:

$$F = BD$$

Reduce from 4 4-input-AND gates and 1 4-input-OR gate to 1 2-input AND gate

EVEN MORE EXAMPLE

- Find and simplify F
- What is the gate implementation?

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

SUMMARY

❖ Logic minimization using K-map

- ➢ Begin with isolated cells
- ➢ These must be used as they are and no simplification is possible
- ➢ Find all cells that are adjacent to only one other cell, forming two-cell subcubes
- ➢ Find cells that form four-cell subcubes, eight-cell subcubes, etc.
- ➢ Collect the smallest number of maximal subcubes

	$\overline{A}\,\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}\,\overline{D}$	0	0	0	0
$\overline{C}D$	0	1	1	0
CD	0	1	1	0
$C\overline{D}$	0	0	0	0

WRITTEN MIDTERM

- Date: July 11 (Wed)
- Time: 10:10am-11:50am (100 minutes)
- Exam Venue: Rm6591

**Seating plan will be posted outside of the exam venue on exam day.

- Coverage: Lectures 01-04

LAB MIDTERM

- Date: July 13 (Fri)
- Time: 15:00am-15:50am (50 minutes)
- Exam Venue: Rm2134
- **Single-person exam**

- Coverage: Labs 01-05

LAB AND WRITTEN MIDTERM FORMAT

- Lab Exam (15%):
  - 3 Questions in total.
  - Closed-book. No electronic devices allowed. Bring your own calculator.
  - No need to memorize formulae or circuit connections as we will provide you.
  - You need to know how to use the voltage supply, function generator and DSO.
  - Primarily tests on circuit building skills. Test is based on labs.

- Written Midterm (20%):
  - 5 Questions in total.
  - Closed-book. No electronic devices allowed. Bring your own calculator.
  - No need to memorize formulae or circuit connections as we will provide you.
  - Similar to examples in Lecture & Tutorial notes. Mainly tests on key concepts.

WRITTEN MIDTERM: PAST PAPER (1)

❖ Short questions:

- What is the difference between a Zener diode and a diode?

  Zenor diode allows current to flow when reverse voltage is larger than a certain value.

- Suppose the resistance of a motor is 50Ω and a PWM signal with a duty cycle of 60% is applied to the motor. If the voltage high V_H is 5V and the voltage low V_L is 0V, determine the power consumed by the motor.

$$P = \frac{5^2}{50} \times 60\% = 0.3W$$

WRITTEN MIDTERM: PAST PAPER (2)

❖ Diodes and Regulators:

In the voltage regulator shown below, the Zener diode has a breakdown voltage of V_{bd} = -6.8V.

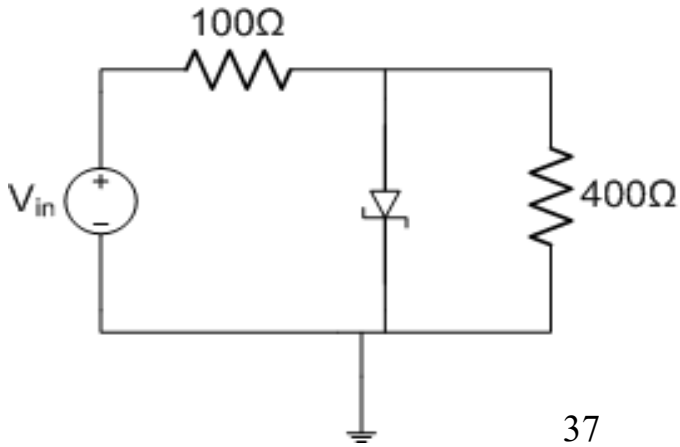
➢ What is the mistake in the circuit?

Not reverse biasd.

➢ Supposed the mistake is fixed, determine the minimum V_{in} that the Zener diode regulates the voltage.

$$V_{in} \times \frac{400}{100 + 400} = 6.8$$

$$V_{in} = 8.5V$$



WRITTEN MIDTERM: PAST PAPER (3)

- PWM:

To design a PWM signal of frequency 10kHz and high time 81.25μs to drive a DC motor:

- (a) Determine Q3-Q0.
- (b) Calculate R_b using timer equations.
- (c) Complete the schematic.

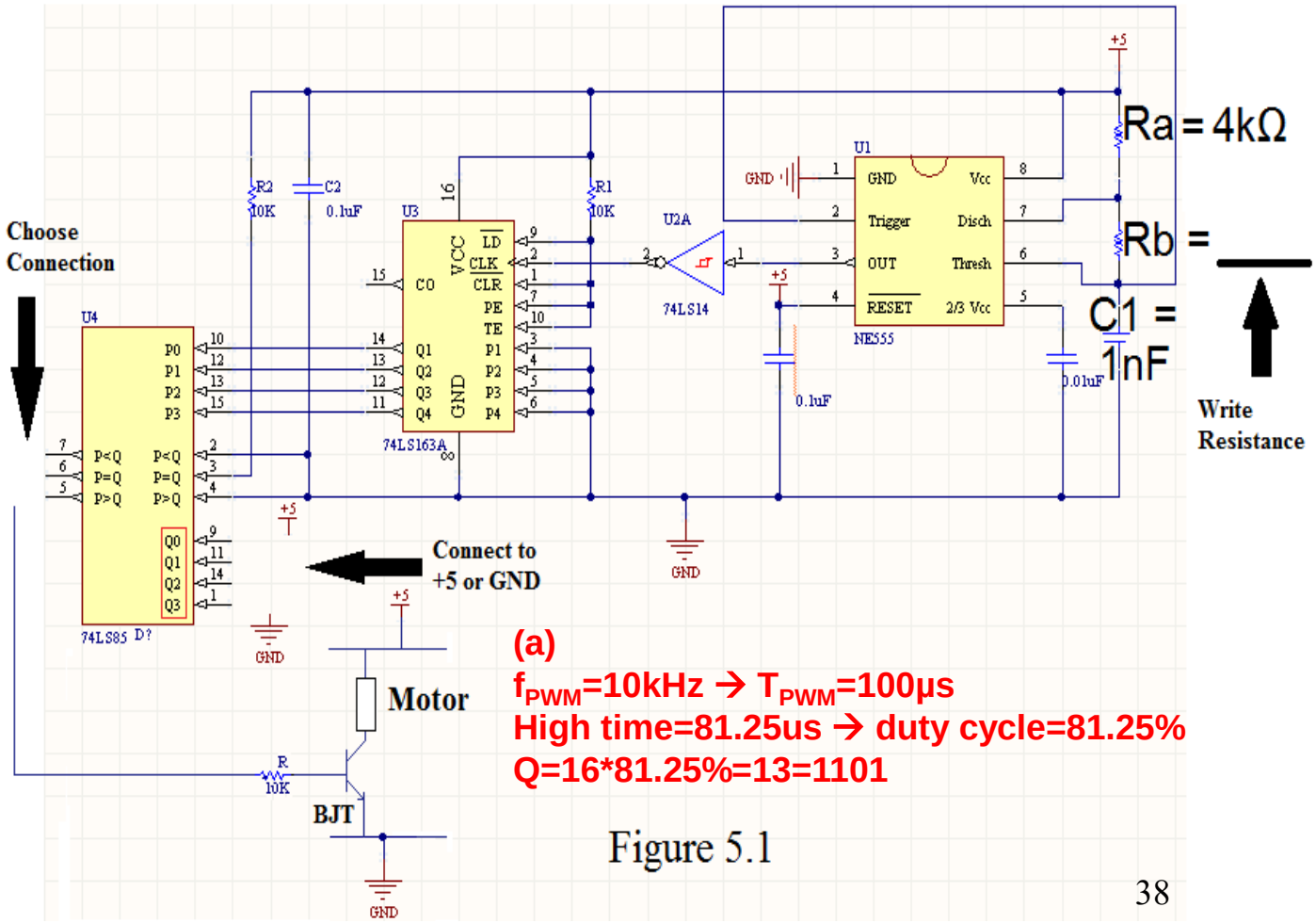


Figure 5.1

WRITTEN MIDTERM: PAST PAPER (3)

PWM:

To design a PWM signal of frequency 10kHz and high time 81.25μs to drive a DC motor:

- (a) Determine Q3-Q0.
- (b) Calculate R_b using timer equations.
- (c) Complete the schematic.

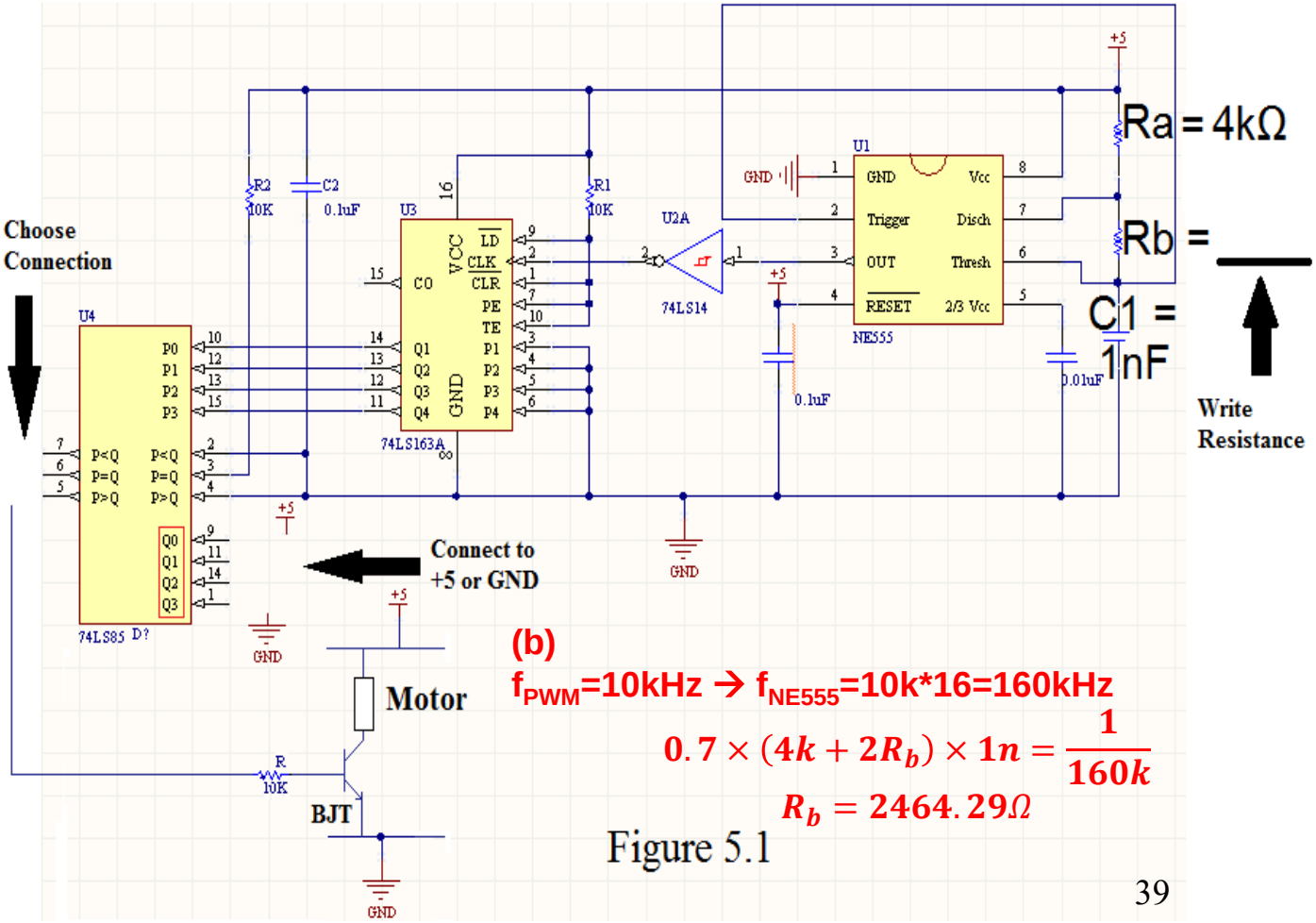


Figure 5.1

WRITTEN MIDTERM: PAST PAPER (3)

- PWM:

  To design a PWM signal of frequency 10kHz and high time 81.25μs to drive a DC motor:

  - (a) Determine Q3-Q0.
  - (b) Calculate R_b using timer equations.
  - (c) Complete the schematic.

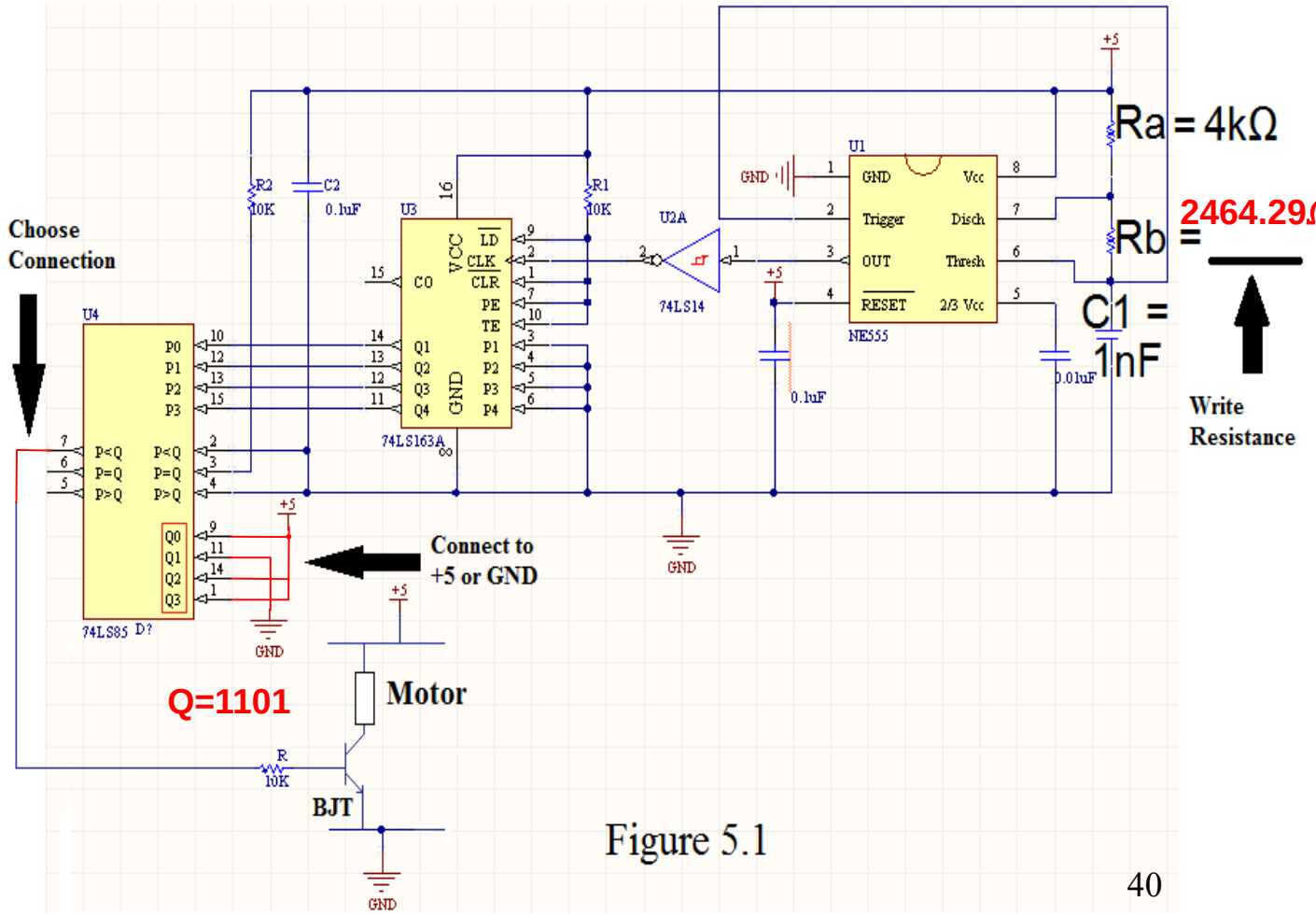


Figure 5.1

WRITTEN MIDTERM: PAST PAPER (4)

❖ Logic and K-map:

We want to design a device to check whether an integer (from 0 to 9) is a multiple of 2 or 3. For example, the device returns true for integers 0, 2 and 3 but false for integers 1, 5 and 7. The decimal integer is input to the device as a 4-bit binary number (ABCD).

- Write down all the integers from 0 to 9 which are multiples of 2 or 3.

  0, 2, 3, 4, 6, 8, 9

- Finish the truth table for outputs. (Remarks: "don't care" conditions are allowed.)

- Use K-map to find out the ***simplest*** output expression in terms of the binary input ABCD.

A	B	C	D	Output
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	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	

WRITTEN MIDTERM: PAST PAPER (4)

❖ Logic and K-map:

We want to design a device to check whether an integer (from 0 to 9) is a multiple of 2 or 3. For example, the device returns true for integers 0, 2 and 3 but false for integers 1, 5 and 7. The decimal integer is input to the device as a 4-bit binary number (ABCD).

- Write down all the integers from 0 to 9 which are multiples of 2 or 3.

  0, 2, 3, 4, 6, 8, 9

- Finish the truth table for outputs. (Remarks: "don't care" conditions are allowed.)

- Use K-map to find out the *simplest* output expression in terms of the binary input ABCD.

A	B	C	D	Output
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

WRITTEN MIDTERM: PAST PAPER (4)

❖ Logic and K-map:

We want to design a device to check whether an integer (from 0 to 9) is a multiple of 2 or 3. For example, the device returns true for integers 0, 2 and 3 but false for integers 1, 5 and 7. The decimal integer is input to the device as a 4-bit binary number (ABCD).

- Write down all the integers from 0 to 9 which are multiples of 2 or 3.

  0, 2, 3, 4, 6, 8, 9

- Finish the truth table for outputs. (Remarks: "don't care" conditions are allowed.)

A	B	C	D	Output
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

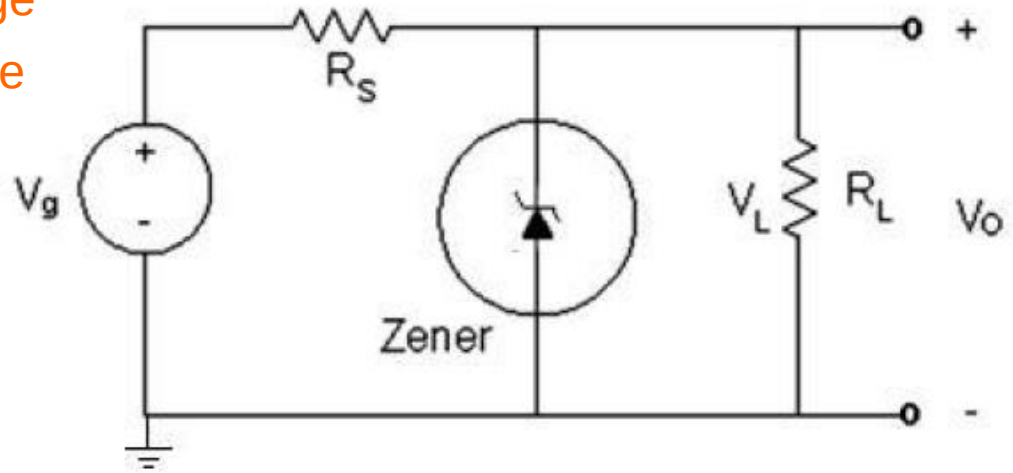
- Use K-map to find out the *simplest* output expression in terms of the binary input ABCD.

		AB			
		00	01	11	10
CD	00	1	1	X	1
	01	0	0	X	1
	11	1	0	X	X
	10	1	1	X	X

Output = D'+A+B'C

LAB MIDTERM: PAST PAPER (1)

- Zener diode:
  - Construct a voltage regulator circuit using a Zener diode.
  - Set the function generator for a triangular wave from -10V to 10V at 500Hz.
  - (a) Display one complete cycle of the input and output voltage waveforms.
  - (b) From the waveform, determine:
    - i. The diode breakdown voltage
    - ii. The diode forward voltage
    - iii. The output voltage range



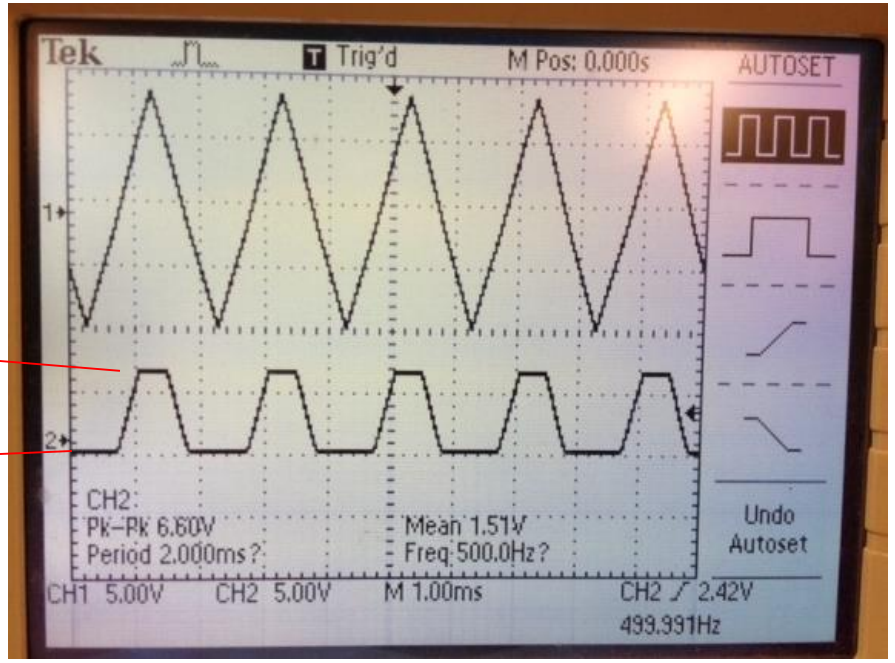
LAB MIDTERM: PAST PAPER (1)

- Zener diode:
  - Construct a voltage regulator circuit using a Zener diode.
  - Set the function generator for a triangular wave from -10V to 10V at 500Hz.
  - (a) Display one complete cycle of the input and output voltage waveforms.
  - (b) From the waveform, determine:
    - i. The diode breakdown voltage
    - ii. The diode forward voltage
    - iii. The output voltage range

(b)(i) V_{bd} = -5.5V ~ -6V

(b)(ii) V_{on} = 0.7V

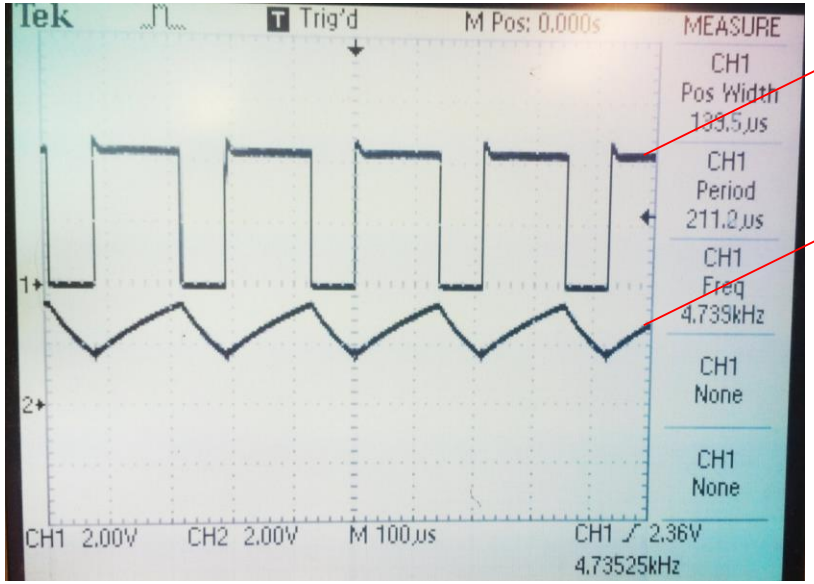
(b)(iii) Range = -0.7V ~ 5.5V/6V



LAB MIDTERM: PAST PAPER (2)

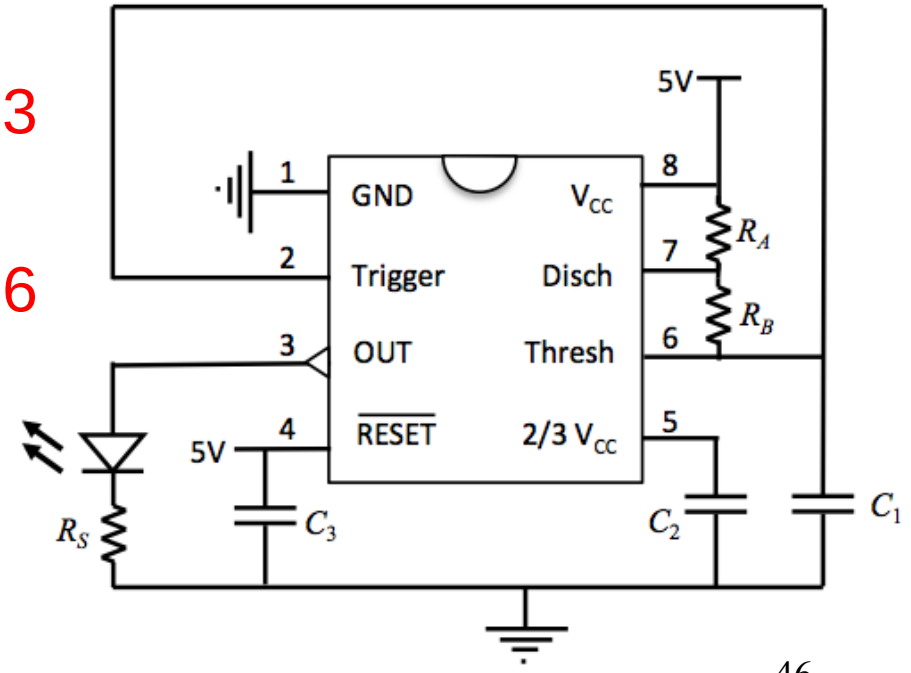
- NE555 timer:
  - Construct the circuit shown in the figure. $R_A = R_B = 1k\Omega$, $C_1 = C_2 = C_3 = 0.1\mu F$.
  - Sketch the waveform at Pin 3 and Pin 6 of the NE555.

$$T_H = 0.7 \times (1k + 1k) \times 0.1\mu = 140\mu s$$
$$T = 0.7 \times (1k + 2k) \times 0.1\mu = 210\mu s$$



Pin 3

Pin 6



NEXT LECTURE

- Sequential Logic
- Computer Architecture
- Embedded System
- MCU & Arduino

Questions ?!