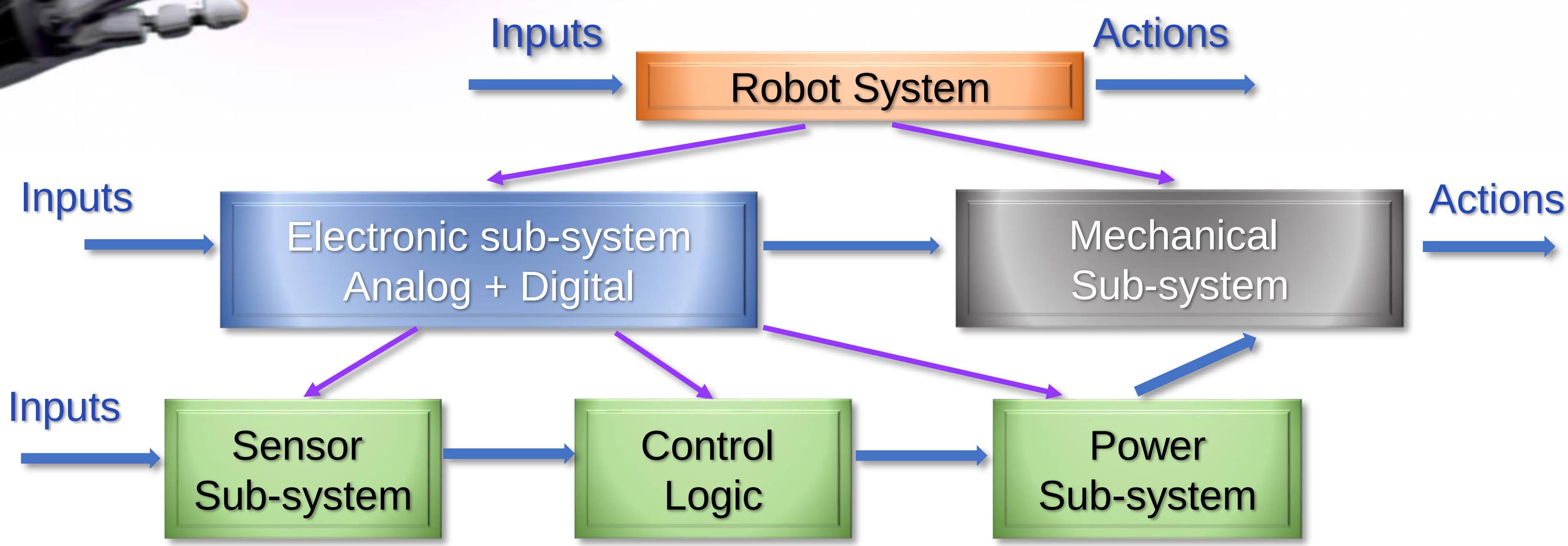
ELEC1100: Introduction to Electro-Robot Design

Lecture 12: Boolean Algebra and K-Map

SONG Shenghui and MURCH Ross, Dept. of ECE, HKUST

ELEC1100 ROADMAP



Sensor Basics:

Wk5: Sensor Basic – Sensor/Line/ADC

Combinational/Sequential Logic:

Wk6: Robot Brain: Logic Gate and Logic Operation

Basic electronics:

Wk1: Basic Electronics - Charge/Current/Voltage/Resistor

Wk2: Energy/Power and DC Sources

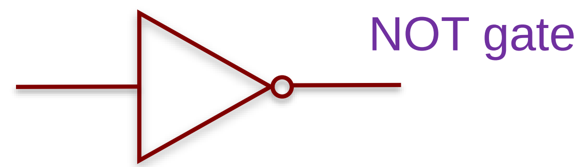
Motor Power Supply:

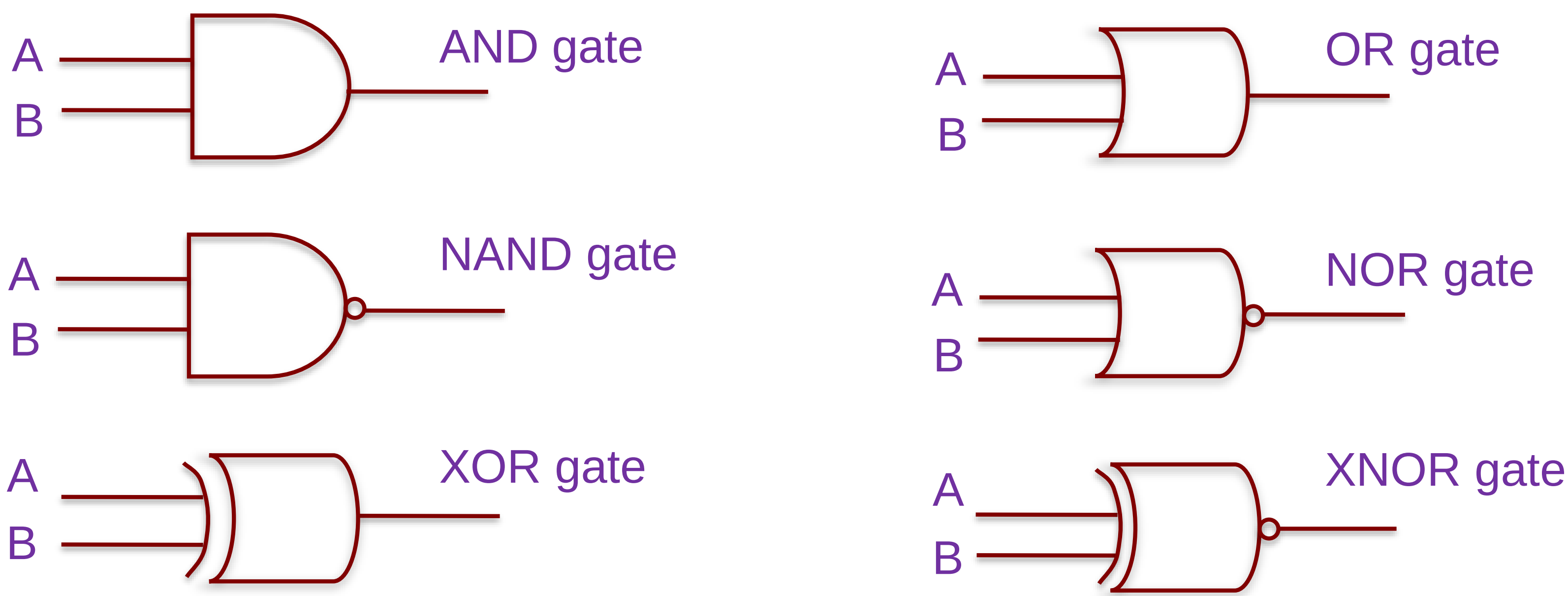
Wk3: Pulse Signal and PWM Control

Wk4: Transistor and H-Bridge

FROM LAST LECTURE

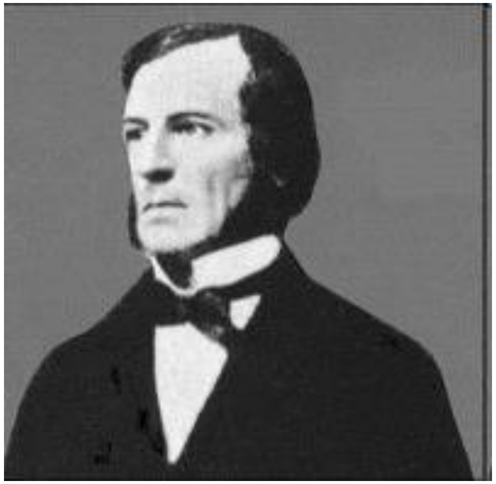
- Binary digit: 0 and 1 can be represented by logic (True or False)
  - 0 is equivalent to False
  - 1 is equivalent to True





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.



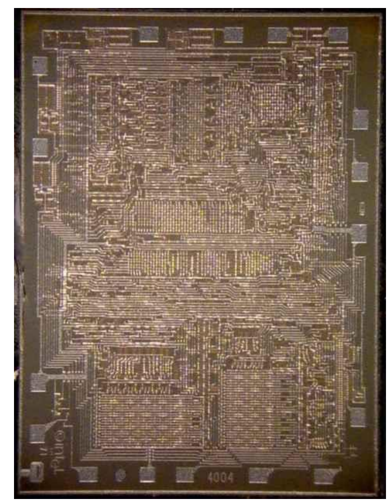
SOME HISTORY [2]

- 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)

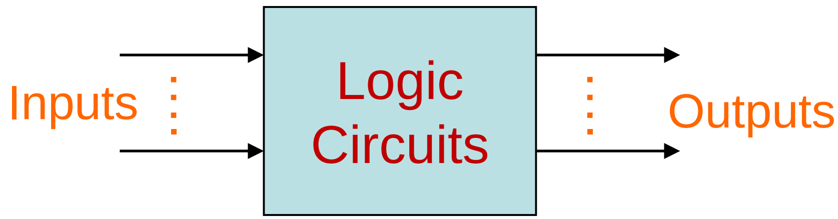
SOME HISTORY [3]

- 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

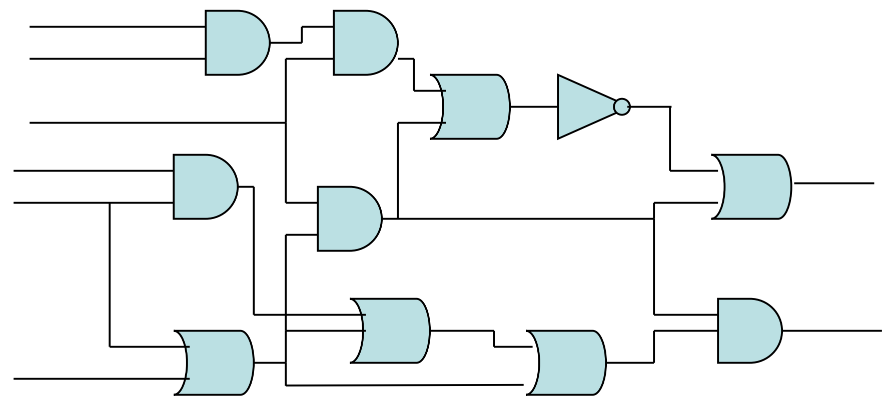


COMBINATIONAL CIRCUIT

- The outputs depend only on the current inputs of the circuits
- Output values are expressed by the truth table of the inputs

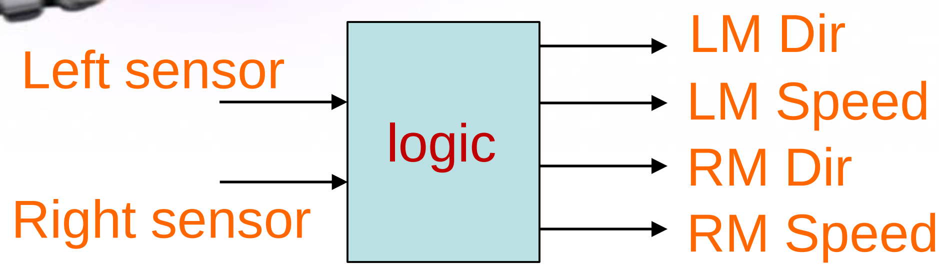


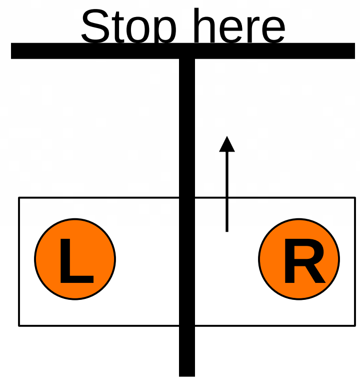
- Logic level of combinational circuit: maximum number of gates of all paths from inputs to outputs of the circuit



Logic level = 5

HOW IS IT RELATED TO YOUR ROBOT?





Sensor signal		**Actions**		**Left motor output (L293)**		**Right motor output (L293)**	
Left	Right	Left	Right	Dir	Speed	Dir	Speed
0	0	forward	forward	1	1	1	1
0	1	forward	reverse	1	1	0	1
1	0	reverse	forward	0	1	1	1
1	1	stop	stop	x	0	x	0

(1-detect the line) (1-Dir forward) (1-full speed)

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]

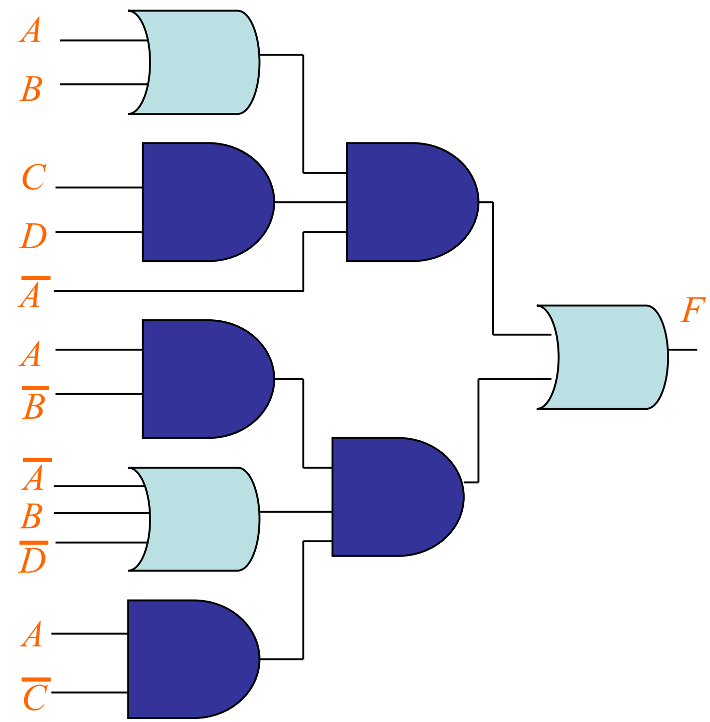
- Build the truth table of the following circuit

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

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

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

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



COMBINATIONAL LOGIC AND TRUTH TABLE [2]

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

$$= \overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D}$$

A	B	C	D	$\overline{A}BCD+A\overline{B}\,\overline{C}\,\overline{D}$
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?

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

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

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

K-map

	$\overline{A}$	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

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

$F = A\overline{B} + AB$ → $F = A$

Can you do this? →

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

K-MAP WITH 3 VARIABLES

Adjacent label should have one and only one variable difference

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}$	1	0	0	1
C	0	1	1	0

Warp around

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:

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

After circling:

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

Before circling:

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

After circling:

$$F = BC$$

Before circling:

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

After circling:

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

K-MAP WITH 4 VARIABLES

Adjacent label should have one and only one variable difference

Adjacent label should have one and only one variable difference

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

These cells are adjacent

These cells are adjacent

Before circling:

$$F = \overline{A}B\overline{C}D+\overline{A}BCD+AB\overline{C}D+ABCD$$
$$+A\overline{B}\,\overline{C}\,\overline{D}+A\overline{B}\,\overline{C}D+A\overline{B}CD+A\overline{B}C\overline{D}$$

First circling:

$$F = \overline{A}BD+ABD+A\overline{B}\,\overline{C}+A\overline{B}C$$

Second circling:

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

Reduce from eight 4-input-AND gates and one 8-input-OR gate to two 2-input-AND gate and one 2-input-OR gate

DON'T CARE (IT IS A WILD CARD)

- Some combination of inputs maybe impossible for a given situation
- For example: A – door locked, B – door open, F – you come in
- Truth Table:

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

- It is impossible for a door to be locked and open together, and we "don't care" about the output in this case
- You may assign it to either "0" or "1" to simplify the case

EXAMPLE IN USING DON'T CARE

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	x	1	0	0
$\bar{C}D$	1	1	1	0
CD	0	1	0	0
$C\bar{D}$	0	0	x	0

1 assignment
of don't care

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

$$F = \bar{A}B\bar{C} + \bar{A}BD + \bar{A}\bar{C}D + B\bar{C}D + ABC\bar{D}$$

another assignment
of don't care

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

$$F = \bar{A}\bar{C} + B\bar{C}D + \bar{A}BD$$

- By assigning different values to don't cares, you can simplify the function and the circuit implementation

LECTURE 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

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

NEXT LECTURE

- Combinational logic
- Sequential Logic with memory unit
- MCU and Arduino

QUESTIONS?