

3. Logic gate and circuits

3.1 Logic switch

3.2 Logic gates

3.3 Logic circuit

3.1 Logic switch

3.1.1 Introduction

- Digital systems are concerned with digital signals.
- Digital signals are most commonly represented by **Binary Signals**.
- **Binary Signals** can be used individually to represent a single binary quantity or the state of a single switch.

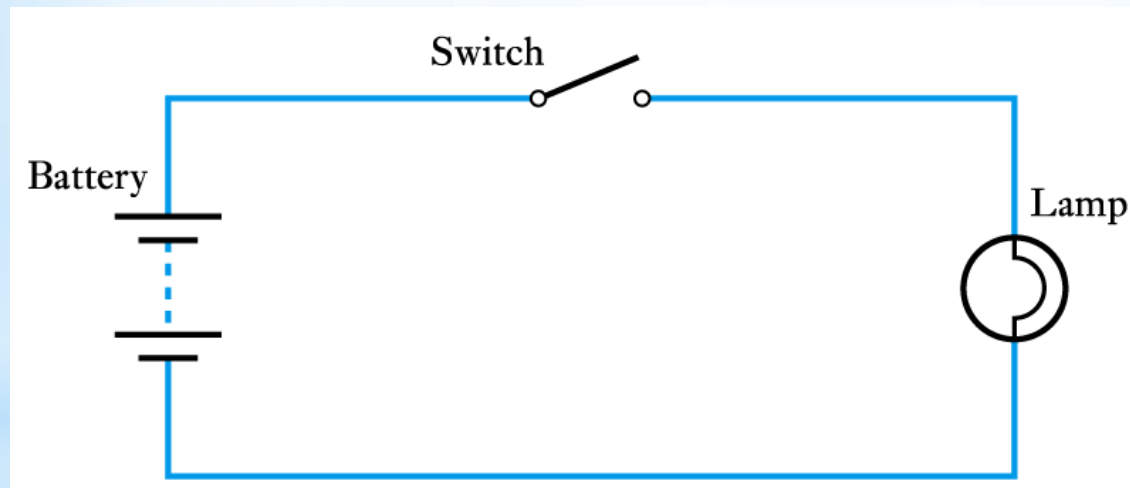
OR

- Used in combination to represent more complex quantities or state of a logical relationship.

3.1.2 Binary Quantities and Variables

- A **binary quantity** is one that can take only 2 states

Example: state of a switch



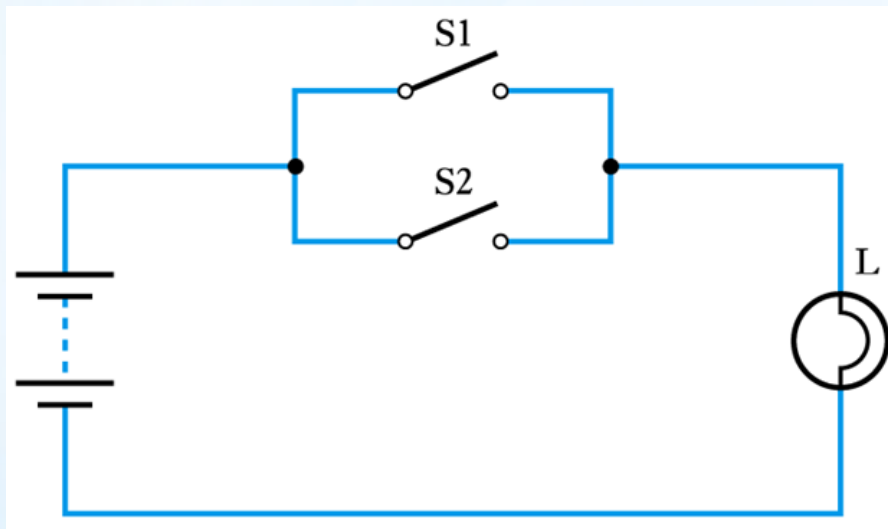
A simple binary arrangement

S	L
OPEN	OFF
CLOSED	ON

S	L
0	0
1	1

A truth table

3.1.3 A binary arrangement with two switches in parallel



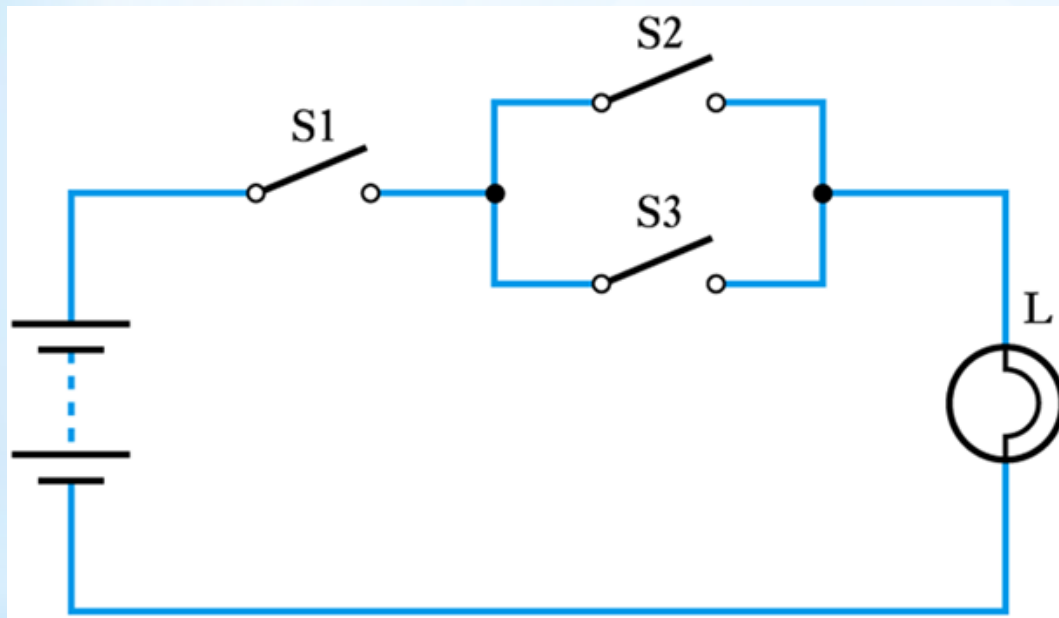
Circuit

S1	S2	L
0	0	0
0	1	1
1	0	1
1	1	1

Truth table

Logic expression: $L = S1 \text{ OR } S2$

3.1.4 A series/parallel arrangement



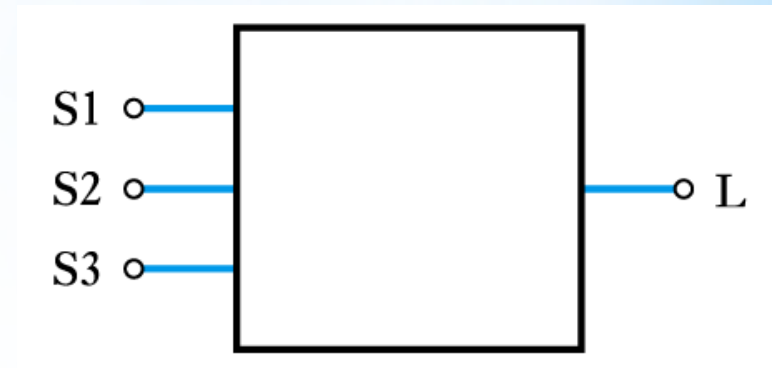
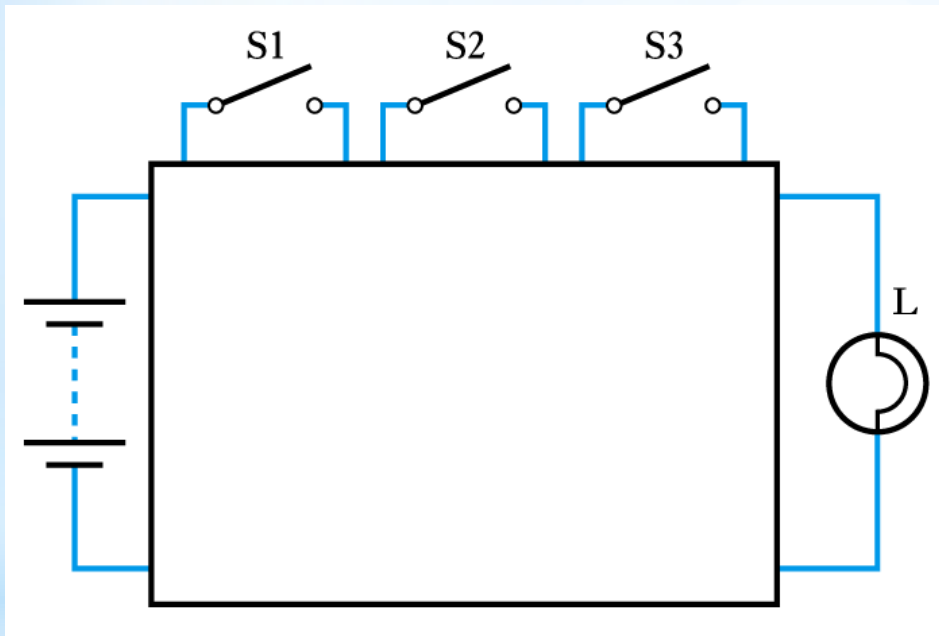
Circuit

S1	S2	S3	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Truth table

Logic expression: $L = S1 \text{ AND } (S2 \text{ OR } S3)$

3.1.5 Representing an unknown network



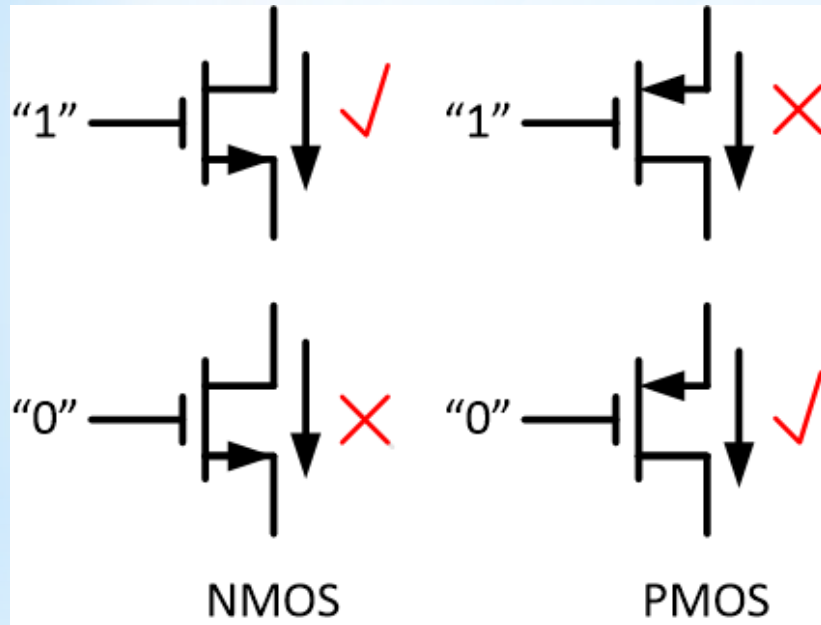
3.1.6 What is a logic switch?

- **Ideal switches** means a switch is completely ON or completely OFF.
- No such device implemented by semiconductor.
- Association between switch state and logic voltage:

Switch on: **logic “1”** represented by a high voltage

Switch off: **logic “0”** represented by a low voltage

■ Consider a voltage controlled switch



Logic “1” – a high voltage

Logic “0” – low voltage

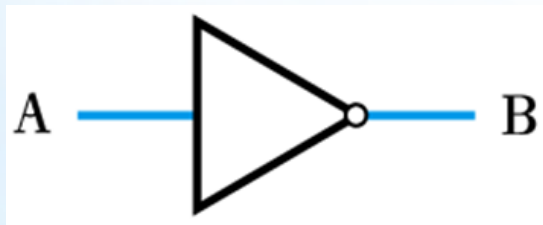
Current can only flow from high voltage to low voltage node.

Current flow $\Rightarrow$ switch close = Logic “1” or “HIGH”

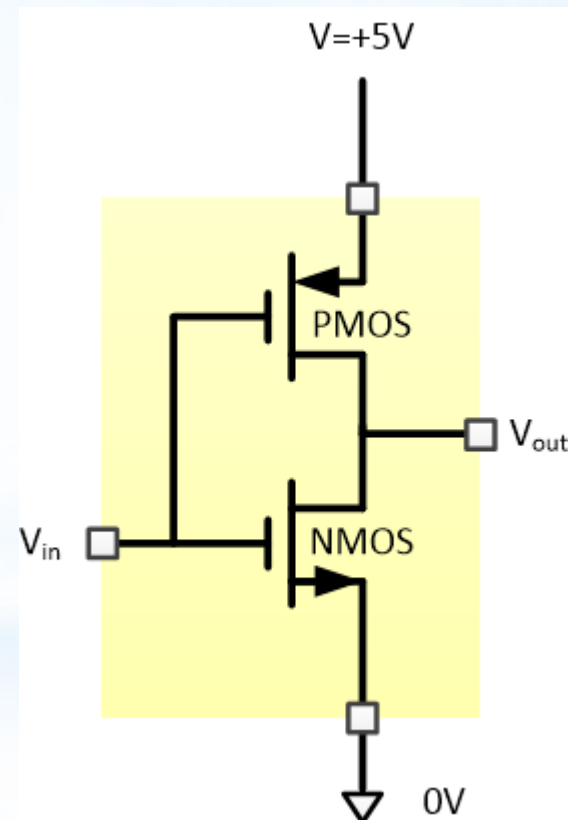
No current flow $\Rightarrow$ switch open = Logic “0” or “LOW”

3.1.7 Implementing logic gate with logic switches

- An **inverter** can be implemented with a PMOS and a NMOS as follow:

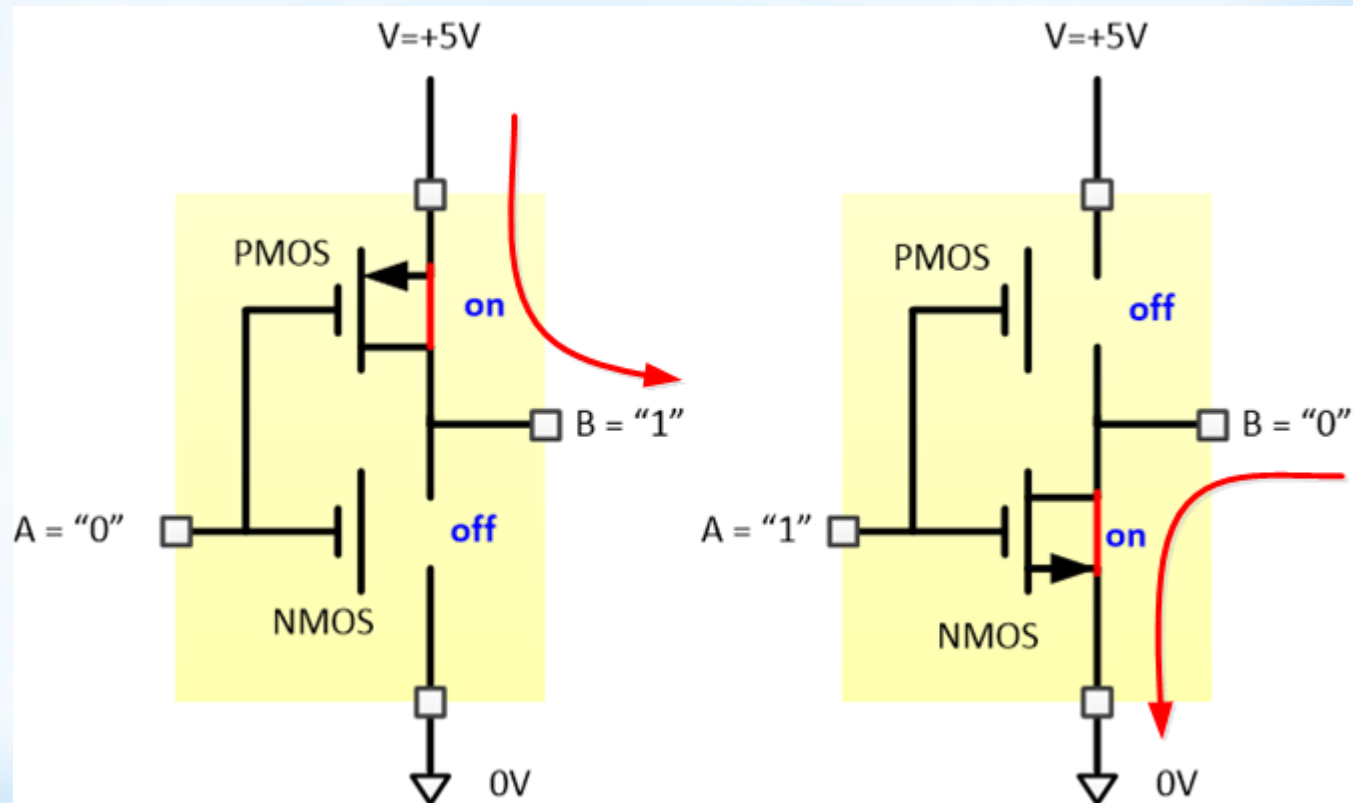


Symbol



Circuit

Inverter
operation:



Truth table:

A (Input)	B (Output)
0	1
1	0

3.2 Logic gates

3.2.1 AND gate



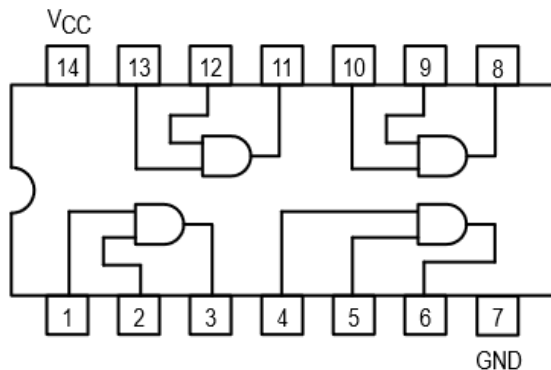
(a) Circuit symbol

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

(b) Truth table

$$C = A \cdot B$$

(c) Boolean expression



74LS08



3.2.2 OR gate



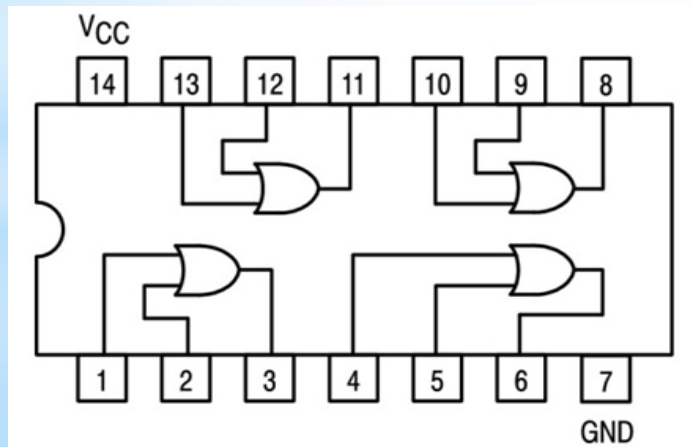
(a) Circuit symbol

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

(b) Truth table

$$C = A + B$$

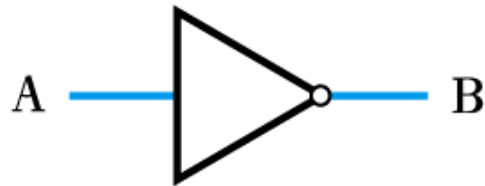
(c) Boolean expression



74LS32



3.2.3 Inverter (or NOT gate)



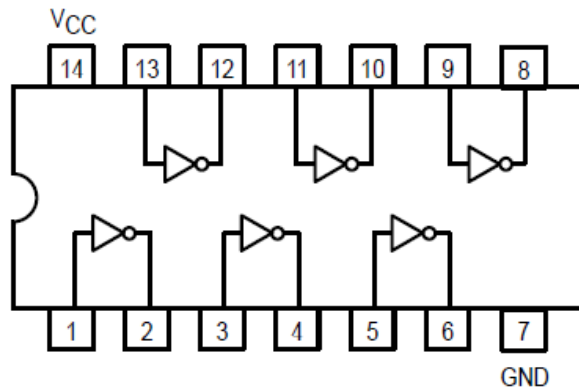
(a) Circuit symbol

A	B
0	1
1	0

(b) Truth table

$$B = \bar{A}$$

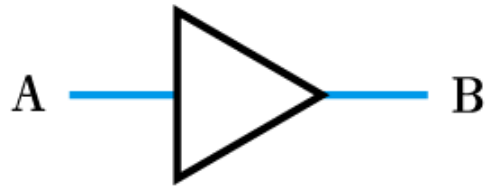
(c) Boolean expression



74LS04



3.2.4 Non-inverter (or buffer gate - for increasing o/p current)



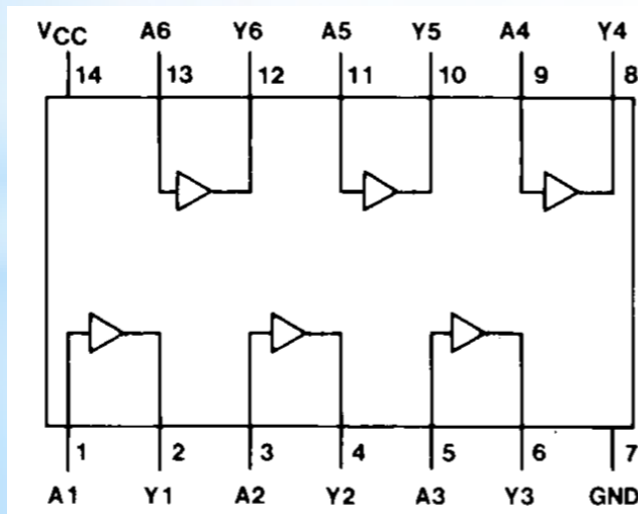
(a) Circuit symbol

A	B
0	0
1	1

(b) Truth table

$$B = A$$

(c) Boolean expression



74AS34

3.2.5 NAND gate



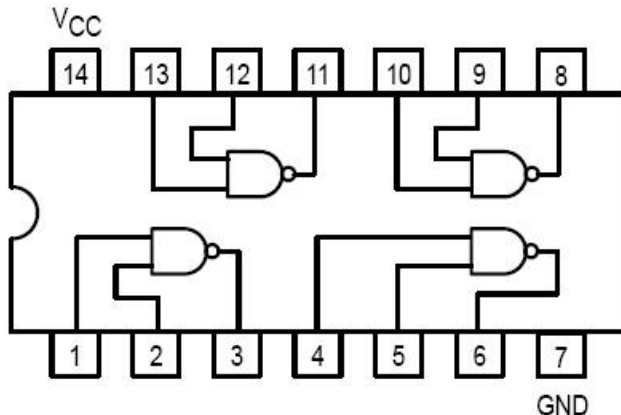
(a) Circuit symbol

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

(b) Truth table

$$C = \overline{A \cdot B}$$

(c) Boolean expression



74LS00



3.2.6 NOR gate



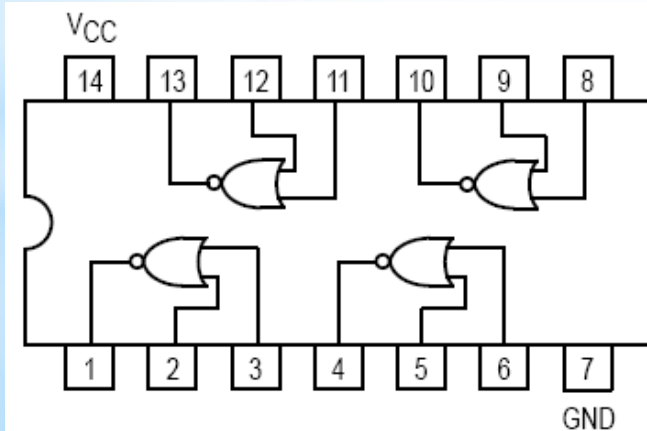
(a) Circuit symbol

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

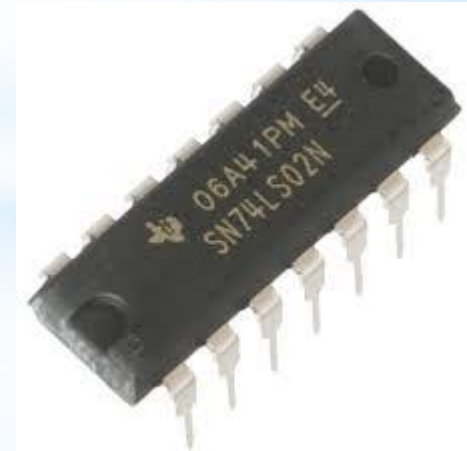
(b) Truth table

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

(c) Boolean expression



74LS02



3.2.7 XOR gate



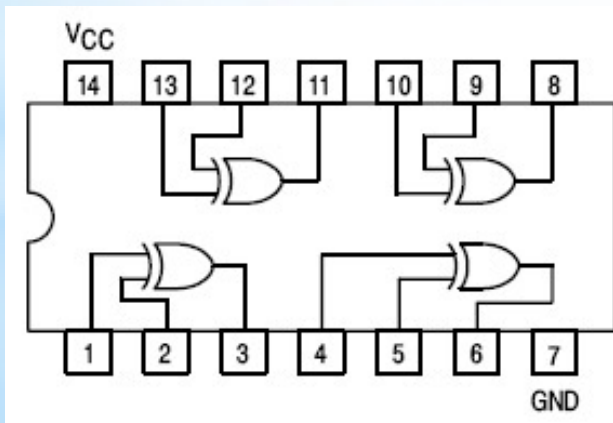
(a) Circuit symbol

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

(b) Truth table

$$C = A \oplus B$$

(c) Boolean expression



74LS86



3.2.8 XNOR gate



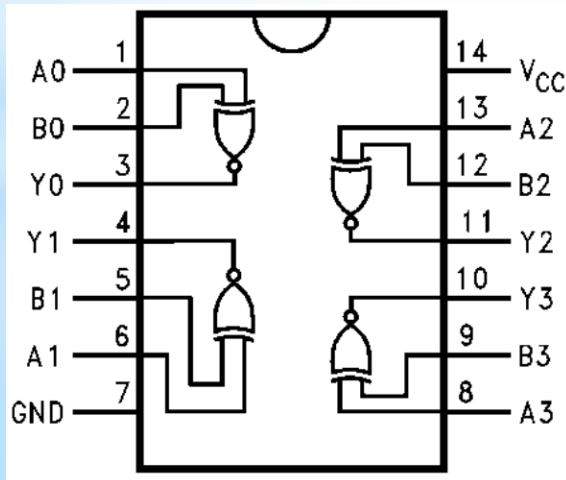
(a) Circuit symbol

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

(b) Truth table

$$C = \overline{A \oplus B}$$

(c) Boolean expression

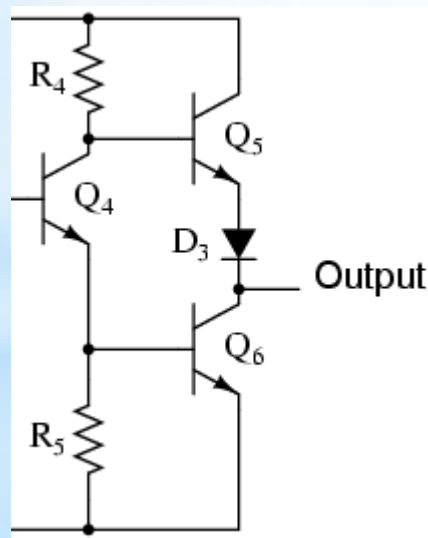


GD74HCT266

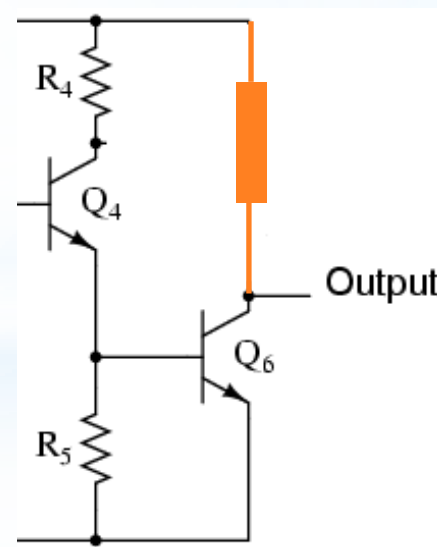


3.2.9 TTL logic gate with open-collector output

- Some TTL logic gates are with different kind of output circuit – *open collector output*.
- For *open collector output*, logic “1” cannot be produced directly, external pull-up resistor is required.



Totem-pole
output



Open-
collector
output

Examples:

	Totem-pole o/p	Open-collector o/p
AND gate	74LS08	74LS09
OR gate	74LS32	74LS33
Inverter	74LS04	74LS05
Non-inverter	74AS34	
NAND gate	74LS00	74LS01
NOR gate	74LS02	74LS03
XOR gate	74LS86	74LS136
XNOR gate	CD74HCT266	74LS266

Choose TTL logic gates with care!!!!

Get some ideas from the following site when making decision:

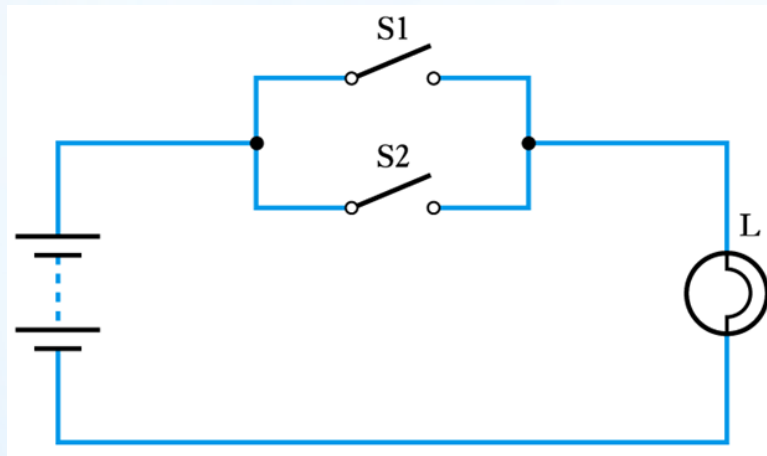
https://en.wikipedia.org/wiki/List_of_7400_series_integrated_circuits

3.2.10 IC (Integrated Circuit)

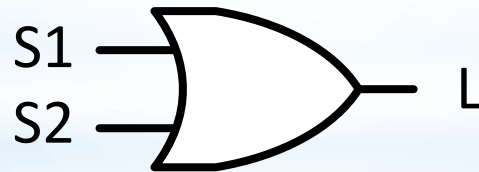
- A collection of one or more gates fabricated on a single silicon chip is called an integrated circuit (IC).
- ICs were classified by size:
 - SSI – small scale integration - 1~20 gates
 - MSI – medium scale integration - 20~200 gates
 - LSI – large scale integration - 200~200,000 gates
 - VLSI – very large scale integration - over 1M transistors
- Core i-7 CPU – from 1.2 billion to 2.6 billion transistors depending on number of core and architecture.

3.3 Logic circuit

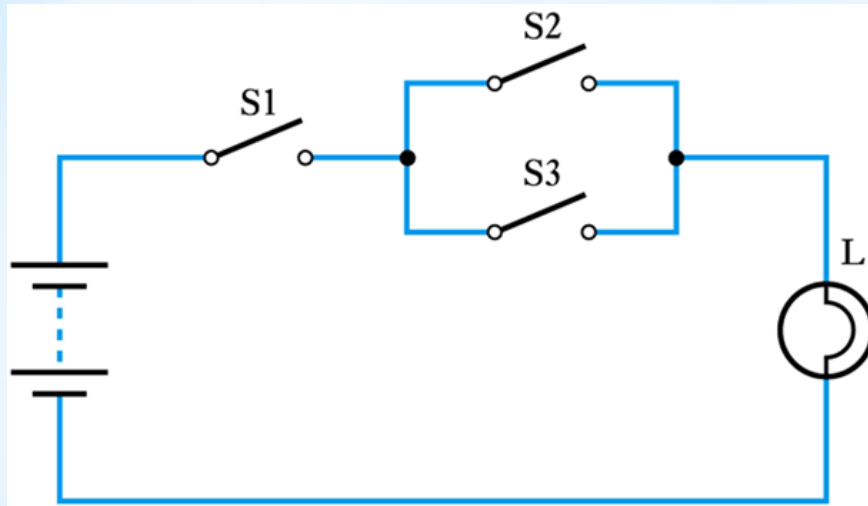
3.3.1 Using the examples from section 3.1



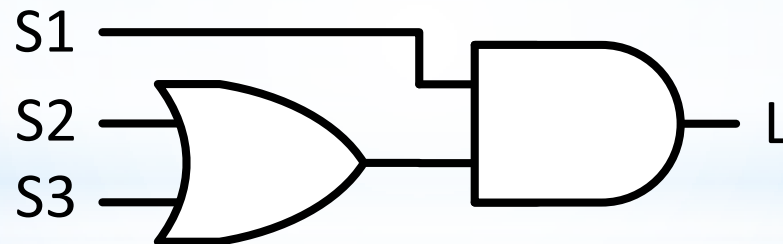
S1	S2	L
0	0	0
0	1	1
1	0	1
1	1	1



$$L = S1 \text{ OR } S2 \Rightarrow L = S1 + S2 \text{ (Boolean expression)}$$

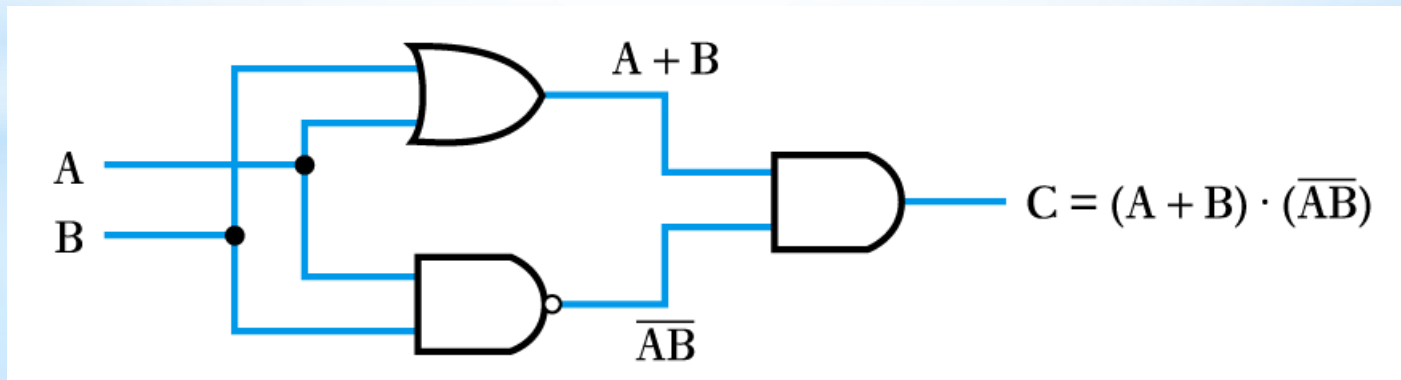
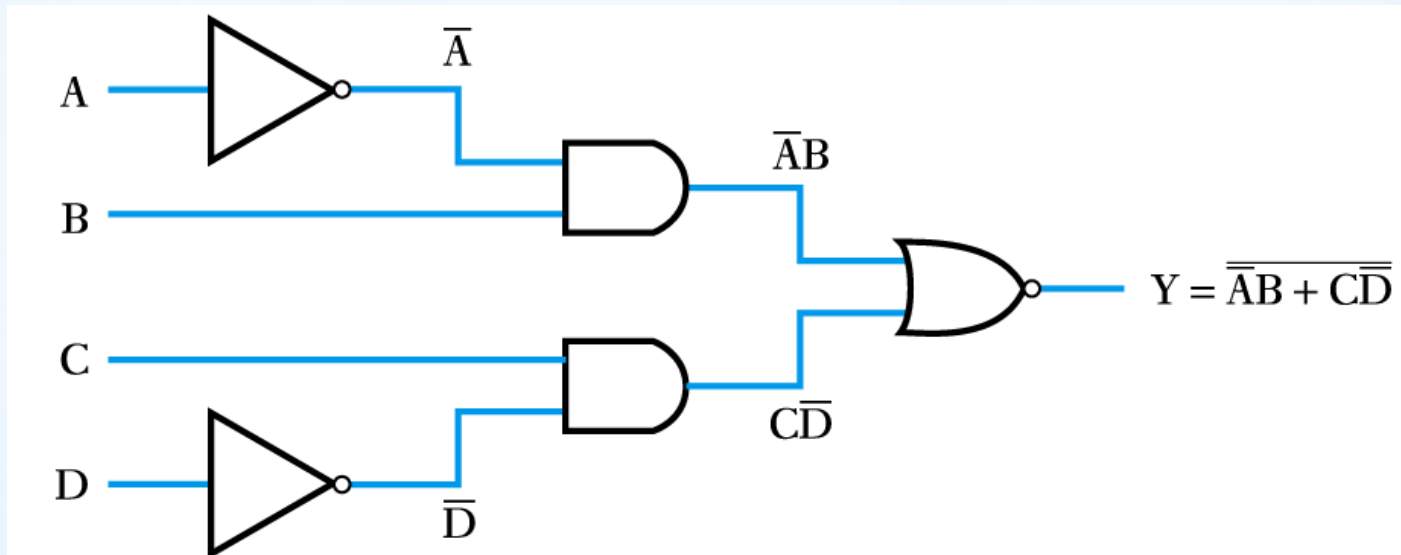


S1	S2	S3	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

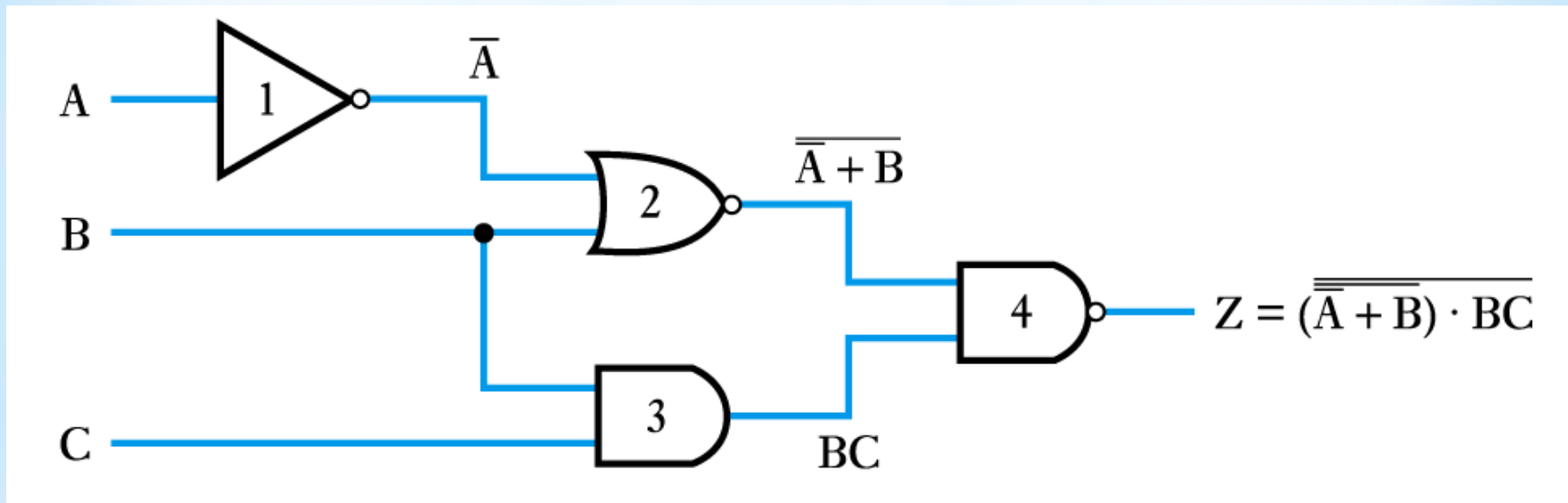


$$L = S1 \text{ AND } (S2 \text{ OR } S3) \Rightarrow L = S1 \cdot (S2 + S3)$$

More examples on using logic to implement Boolean function:



More examples:

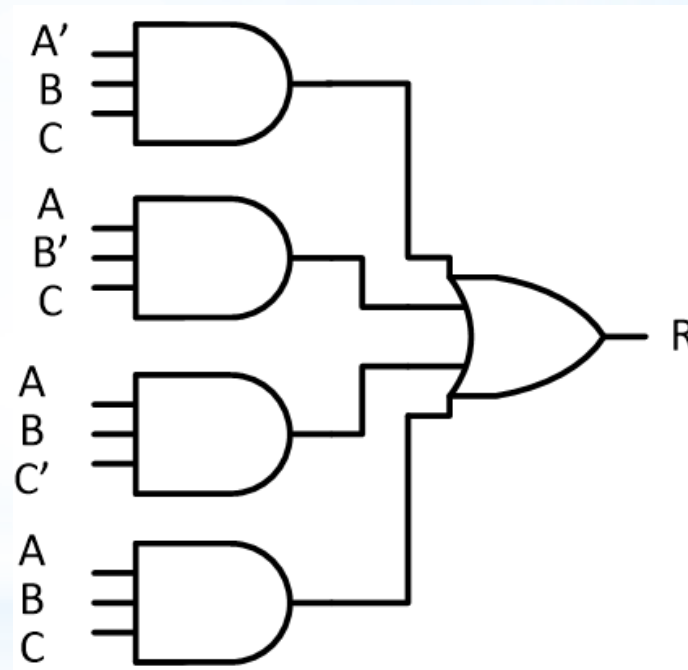


Logic gates are not limited for designing circuit that performs logical operation only, it can be used to design circuit for arithmetic operations !!!

3.3.2 Design a voting system - majority win (Logic operation)

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

$$R = A'BC + AB'C + ABC' + ABC$$



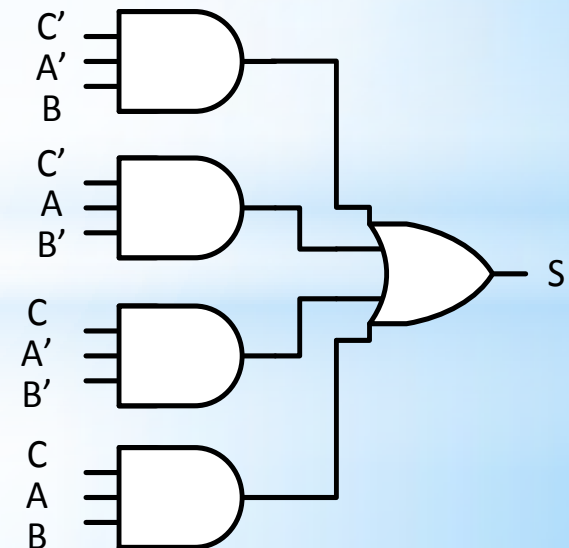
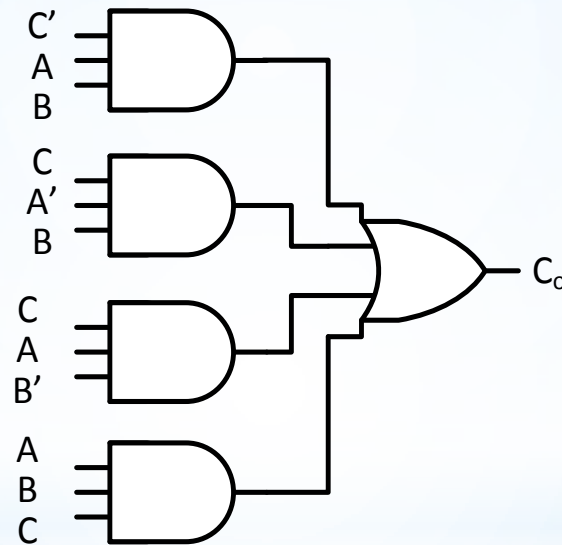
Note that: $A' \equiv \bar{A}$

3.3.3 Design a 1-bit full-adder (Arithmetic operation)

carry out $C_o = A'BC + AB'C + ABC' + ABC$

sum $S = A'B'C + A'BC' + AB'C' + ABC$

C_i A B	C_o S
0 0 0	0 0
0 0 1	0 1
0 1 0	0 1
0 1 1	1 0
1 0 0	0 1
1 0 1	1 0
1 1 0	1 0
1 1 1	1 1

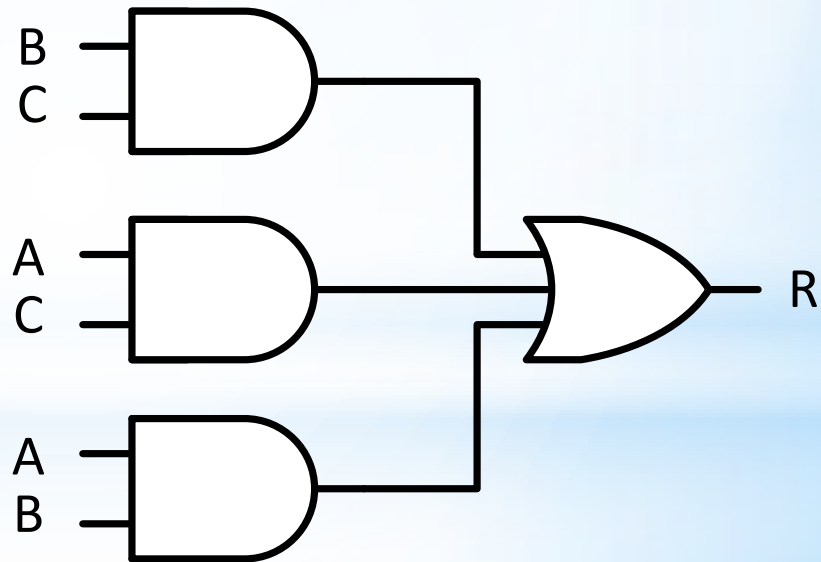


3.3.4 Simplifying the logic circuit

Again, the voting system in 3.3.2

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

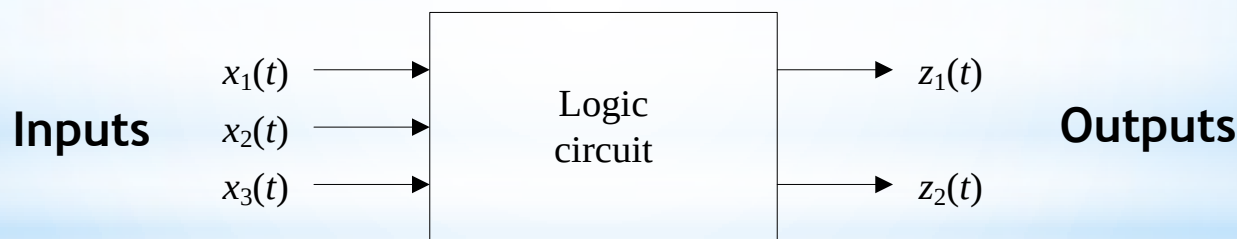
$$\begin{aligned} R &= A'BC + AB'C + ABC' + ABC \\ &= BC + AC + AB \end{aligned}$$



3.3.5 Combinational and sequential circuits

■ Combinational circuit

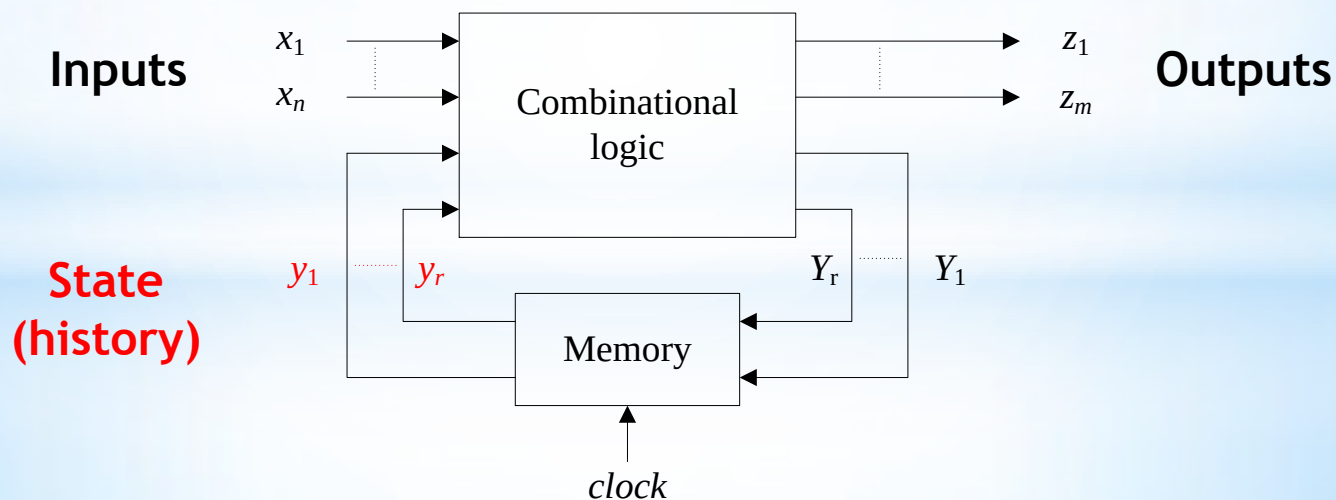
- The output of a combinational circuit changes as inputs change subject to logic device delay
- The voting system and adder are classified as combination circuit



3.3.5 Combinational and sequential circuits (cont'd)

■ Sequential circuit

- Sequential circuit requires a clock to synchronize the operation of all devices
- The output of a sequential circuit depends on both the inputs and circuit state (history - device with memory is needed)





3.3.6 More Techniques for logic circuit design

Design logic circuit is fun !!!

- Knowledge enable us to achieve the goals.
- Boolean function
- Simplification techniques
- MIS components
- Device with memory
- State machine design concept
- Sequential circuit design

See you all in EE2000 (Logic Circuit Design)