Indian Institute of Information Technology - Vadodara

EL - 101
Digital Logic Design

Jignesh S. Bhatt

Lecture # 3
Autumn 2014

Today’s class

- Number systems – part 2
- Binary storage
- Binary logic gates – part 1

Decimal-to-Binary conversion

Example: $(13.375)_{10} = (?)_2$

- The integer part = 13

Divisor	Dividend	Remainder
2	13	—
2	6	1
2	3	0
2	1	1
—	0	1

The binary equivalent of $(13)_{10}$ is therefore $(1101)_2$

..continued

- The fractional part = .375

$0.375 \times 2 = 0.75$ with a carry of 0

$0.75 \times 2 = 0.5$ with a carry of 1

$0.5 \times 2 = 0$ with a carry of 1

The binary equivalent of $(0.375)_{10} = (.011)_2$

- Solution: $(13.375)_{10} = (1101.011)_2$

Decimal-to-Octal conversion

Example: $(73.75)_{10} = (?)_8$

- The integer part = 73

Divisor	Dividend	Remainder
8	73	—
8	9	1
8	1	1
—	**0**	**1**

$$(73)_{10} = (111)_8$$

….continued

- The fractional part = 0.75

  $0.75 \times 8 = 0$ with a carry of 6

  $(0.75)_{10} = (.6)_8$

- Solution: $(73.75)_{10} = (111.6)_8$

Decimal-to-Hexadecimal conversion

Example: $(82.25)_{10} = (?)_{16}$

- The integer part = 82

Divisor	Dividend	Remainder
16	82	—
16	5	2
—	0	5

$$(82)_{10} = (52)_{16}$$

…continued

- The fractional part = 0.25

  $0.25 \times 16 = 0$ with a carry of 4

- Solution: $(82.25)_{10} = (52.4)_{16}$

Octal-to-Binary conversion

- Replace each octal digit with 3-bit binary equivalent.
- Why? base of the octal is the third power of the base of the binary, i.e., $8 = 2^3$.

Example: $(374.26)_8 = (?)_2$

3 7 4 2 6

$(011 \;\; 111 \;\; 100 \,.\, 010 \;\; 110)_2$

$\Rightarrow (11111100.01011)_2$

Binary-to-Octal conversion

Example: $(1110100.0100111)_2 = (?)_8$

$(1 \quad 110 \quad 100 \,.\, 010 \quad 011 \quad 1)_2$

$= (001 \quad 110 \quad 100 \,.\, 010 \quad 011 \quad 100)_2$

$= (164.234)_8$

Hexadecimal-to-Binary conversion

- Replace each hex digit with 4-bit binary equivalent.
- Why? base of the hexadecimal is the fourth power of the base of the binary, i.e., $16 = 2^4$.

- Example: $(17E.F6)_{16}$

  $= (0001 \quad 0111 \quad 1110 \; . \; 1111 \quad 0110)_2$

  $= (000101111110.11110110)_2$

  $= (101111110.1111011)_2$

Binary-to-Hex conversion

- Example:

$(1011001110.011011101)_2$

$= (10\ \ 1100\ \ 1110\ .\ 0110\ \ 1110\ \ 1)_2$

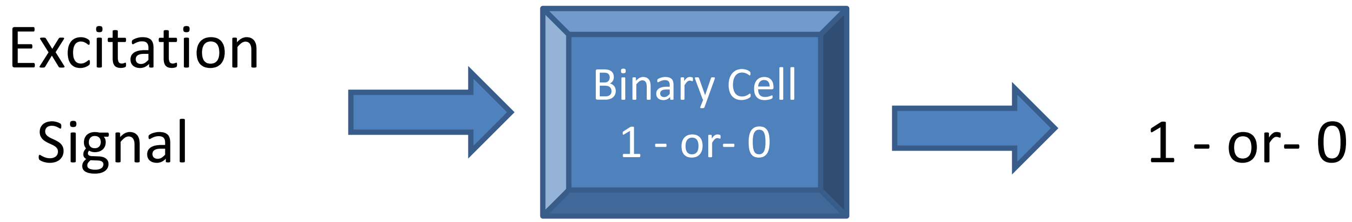
$= (0010\ \ 1100\ \ 1110\ .\ 0110\ \ 1110\ \ 1000)_2$

$= (2CE.6E8)_{16}$

Binary data storage

- Physical medium in computers to store (save) the binary information.

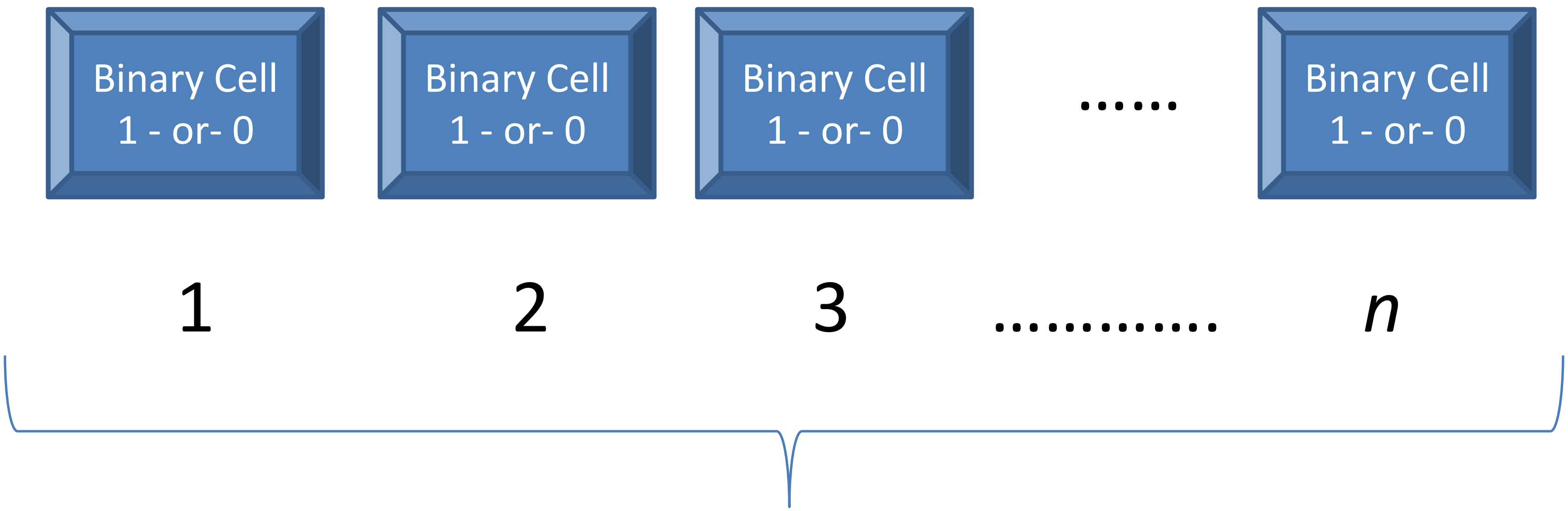
- “Binary Cell”: Holds one of the two coefficients.
  i.e., either “0” or ”1” at a time.



Two stable state device

Binary data storage

- Register : A group of the binary cells.



A *n* - bit register

Remarks:

1. A register with “n” cells can be in one of 2^n possible states.

   Example: 16-bit register => 2^{16} possible states.

2. The information could be interpret in octal, hexadecimal or any other coding system which is machine compatible.

   i.e., depend upon the type of data,

   Example: Keyboard data => ASCII coded data.

Positive and Negative Logic

- Binary: Logic “0” state and Logic “1” state.

- Digital systems use two different voltage levels to represent the two states.

$\Rightarrow$ Example: Two voltage levels 0 volts and +5 volts system,

- Positive logic system: “0” state => 0 volts

  “1” state => +5 volts

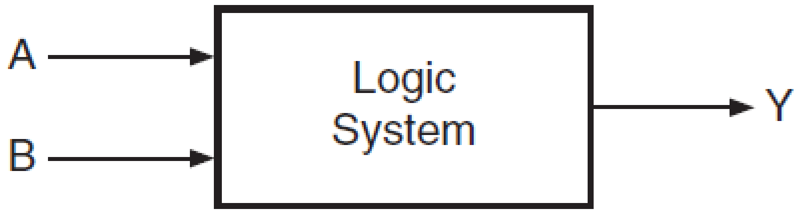
- Negative logic system: “0” state => +5 volts

  “1” state => 0 volts

Binary Logic Gates

- Transistor as a switch.
- The most basic building blocks of any digital system including computers.
- The logic gates and their combinations are used to represent the logic expression to build a digital module.

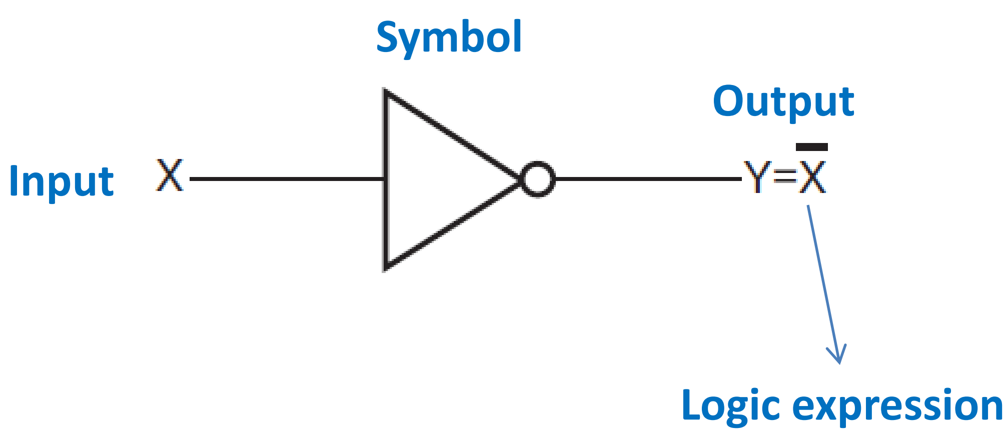
Binary Logic Gates



An exemplary "Truth-Table"

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

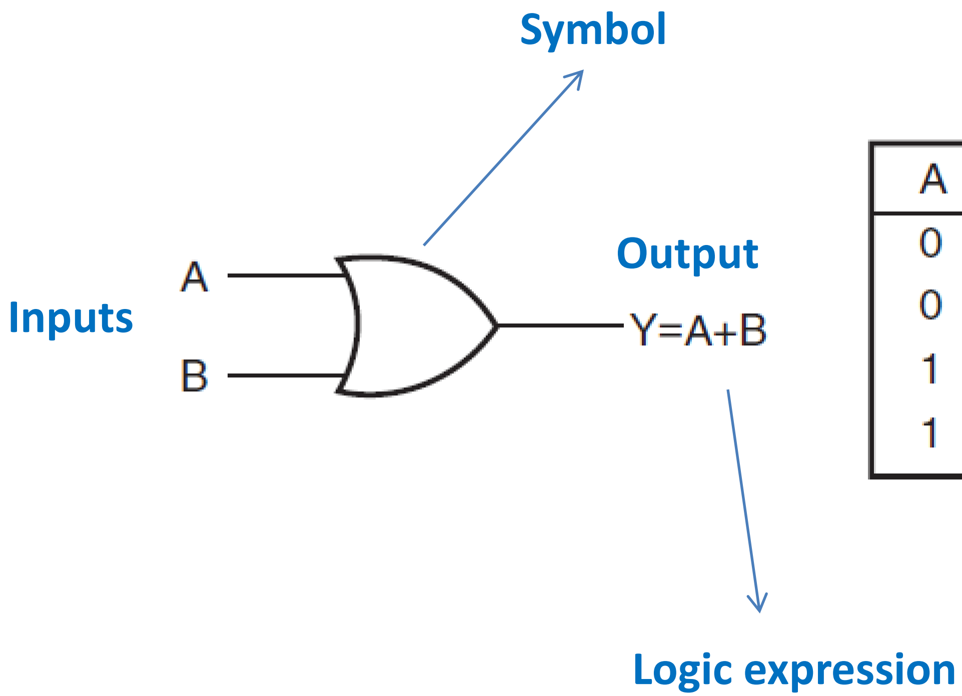
NOT Gate



Truth-table

X	Y
0	1
1	0

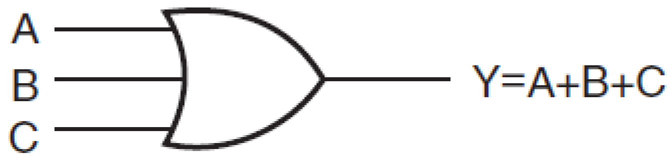
OR Gate

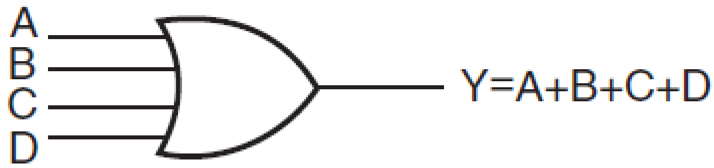


Truth-table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

3 input OR gate and 4 input OR gate





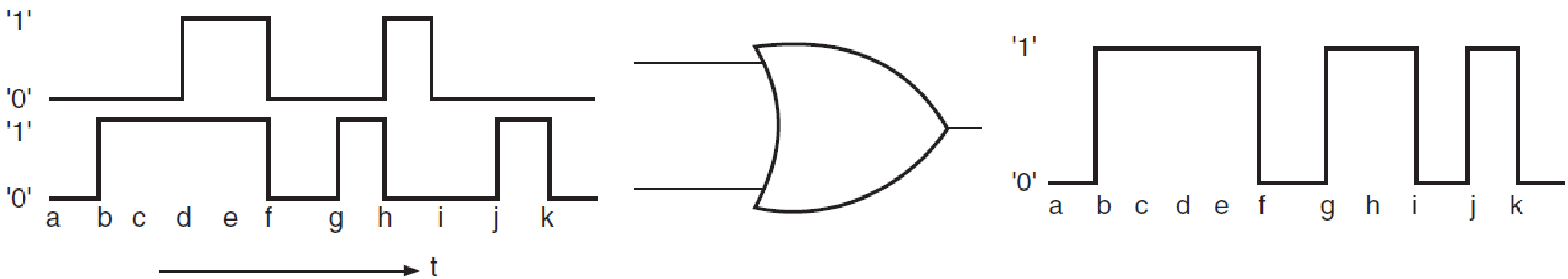
Truth-table of 3 input OR gate

A	B	C	Y
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

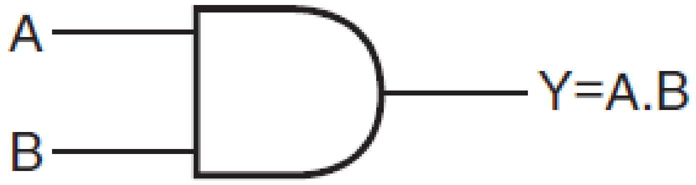
Exercise

- Build a 4-input OR gate logic circuit using only 2-input OR gates.

Drawing output waveform given a pulsed input to an OR gate

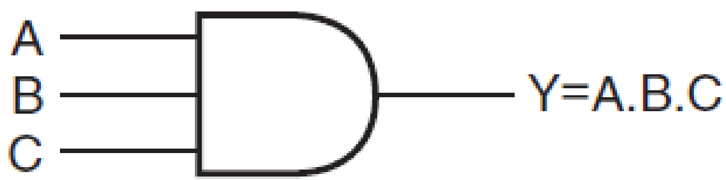


AND Gate

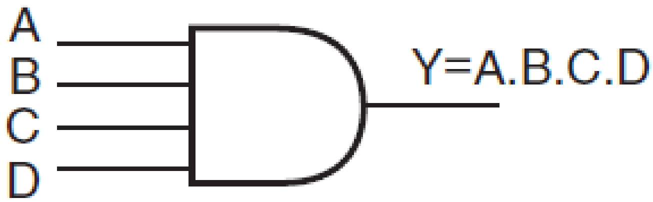


A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

3-input AND gate



4-input AND gate

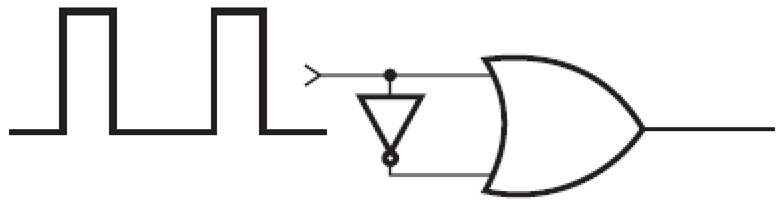


A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

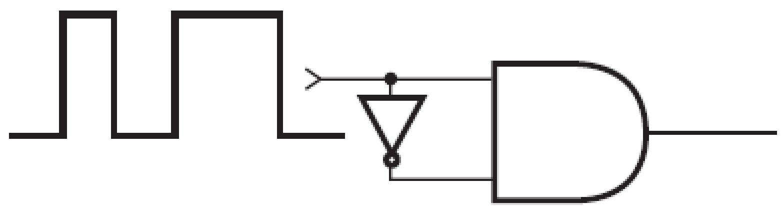
Exercise

- Build a 4-input AND gate logic circuit using only 2-input AND gates.

Exercise



Output waveform ?



Output waveform ?