EEE1024: Fundamentals of Electrical and Electronics Engineering

Dr. Sanchit Khatavkar

MODULE 3:

DIGITAL SYSTEMS

NUMBER SYSTEMS

DECIMAL	BINARY	HEXADECIMAL	OCTAL
BASE - 10	**BASE - 2**	**BASE - 16**	**BASE - 8**

DECIMAL 743.2 = $7 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 2 \times 10^{-1}$

BINARY 1101.1 = $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} = 13.5$

In Binary – a 3-bit word = ?? combinations → **$2^3 = 8$**

8 binary numbers = ?? Decimal integers

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

In Binary – a 4-bit word = ?? combinations → **$2^4 = 16$**

DECIMAL ⟷ BINARY CONVERSIONS

Decimal to Binary

- Repeatedly divide the decimal by 2, till quotient is zero
- **Remainders, read in reverse order, give the binary form**

Conversion of

343_{10} to binary

		Quotient	Remainder
343/2	=	171	1
171/2	=	85	1
85/2	=	42	1
42/2	=	21	0
21/2	=	10	1
10/2	=	5	0
5/2	=	2	1
2/2	=	1	0
1/2	=	0	1

101010111_2

Read binary equivalent in reverse order

Stop when quotient equals zero

$$343_{10} = 101010111_2$$

DECIMAL ⟺ BINARY CONVERSIONS

Decimal Fraction to Binary

- **Repeatedly multiply the fractional part by 2, and retain the whole parts of the result.**
- Stop till the desired precision is reached.

Conversion of

0.392_{10} to binary

$$
\begin{array}{lclcl}
2 \times 0.392 & = & 0 & + & 0.784 \\
2 \times 0.784 & = & 1 & + & 0.568 \\
2 \times 0.568 & = & 1 & + & 0.136 \\
2 \times 0.136 & = & 0 & + & 0.272 \\
2 \times 0.272 & = & 0 & + & 0.544 \\
2 \times 0.544 & = & 1 & + & 0.088
\end{array}
$$

0.011001_2 (approximate binary equivalent)

$$0.392_{10} \cong 0.011001_2$$

To convert a decimal which has a both a whole part and a fractional part,
Convert each part seperately and combine the two

343.392_{10} to binary

$$343_{10} = 101010111_2 \qquad 0.392_{10} \cong 0.011001_2$$

$$343.392_{10} \cong 101010111.011001_2$$

DECIMAL ⟷ BINARY CONVERSIONS

Binary to Decimal

➢ **Multiply by the power of 2 based on its place value**

Conversion of

10011.011 to Decimal

$$10011.011_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 19.375_{10}$$

HEXA-DECIMAL & OCTAL Number Systems

Used extensively in microprocessors

Hexadecimal	
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Octal	
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Binary
0
1

SYMBOLS Used ?

System	Symbols used
BINARY	2
OCTAL	8 (including "0") – All numbers
HEXA-DECIMAL	16 (including "0") – Numbers + Alphabets

OCTAL & HEX → DECIMAL CONVERSIONS

Octal to Decimal

➢**Multiply by the power of 8 based on its place value**

Conversion of

173.2 to Decimal

$$173.21_8 = 1 \times 8^2 + 7 \times 8^1 + 3 \times 8^0 + 2 \times 8^{-1} + 1 \times 8^{-2}$$
$$= 123.265625_{10}$$

Hexadecimal to Decimal

➢**Multiply by the power of 16 based on its place value**

Conversion of

1FA.2A to Decimal

$$1FA.2A_{16} = 1 \times 16^2 + 15 \times 16^1 + 10 \times 16^0 + 2 \times 16^{-1} + 10 \times 16^{-2}$$
$$= 506.1640625_{10}$$

OCTAL & HEX → BINARY CONVERSIONS

Octal to Binary

- **Replace every Octal digit by its binary**

Conversion of

317.2$_8$ to Binary

$$317.2_8 = 011\ 001\ 111.010_2$$

$$= 011001111.010_2$$

Hex to Binary

- **Replace every Hex digit by its binary**

Conversion of

F3A.2 to Binary

$$\text{F3A.2}_{16} = 1111\ 0011\ 1010.0010$$

$$= 111100111010.0010_2$$

BINARY → OCTAL & HEX CONVERSIONS

Binary to Octal

- **Form *3-bit* groups**
- **Start from binary point**
- **Append leading and trailing zeros**

Conversion of 11110110.1_2 to Octal

$$\begin{aligned} 11110110.1 &= 11\ 110\ \ 110.1 \\ &= 011\ 110\ \ 110.100 \\ &= 366.4_8 \end{aligned}$$

Binary to Hex

- **Form *4-bit* groups**
- **Start from binary point**
- **Append leading and trailing zeros**

Conversion of 11110110.1_2 to Hex

$$\begin{aligned} 11110110.1 &= 1111\ 0110.1 \\ &= 1111\ 0110.1000 \\ &= F6.8_{16} \end{aligned}$$

Decimal	Binary	Octal	Hexadecimal
0	0000	0 (000)	0 (0000)
1	0001	1 (001)	1 (0001)
2	0010	2 (010)	2 (0010)
3	0011	3 (011)	3 (0011)
4	0100	4 (100)	4 (0100)
5	0101	5 (101)	5 (0101)
6	0110	6 (110)	6 (0110)
7	0111	7 (111)	7 (0111)
8	1000	10	8 (1000)
9	1001	11	9 (1001)
10	1010	12	A (1010)
11	1011	13	B (1011)
12	1100	14	C (1100)
13	1101	15	D (1101)
14	1110	16	E (1110)
15	1111	17	F (1111)

ASSIGNMENT

Q1) Convert the following decimals to binary.
Stop after max 6 bit for fractional part

a) 23.75
b) 17.25
c) 4.3

Q2) Convert the following binary to decimals.

a) 1101.111_2
b) 100.001_2

Q2) Convert the following into binary, octal and hexadecimal forms

a) 97_{10}
b) 229_{10}

Q4) Convert the following numbers into binary form

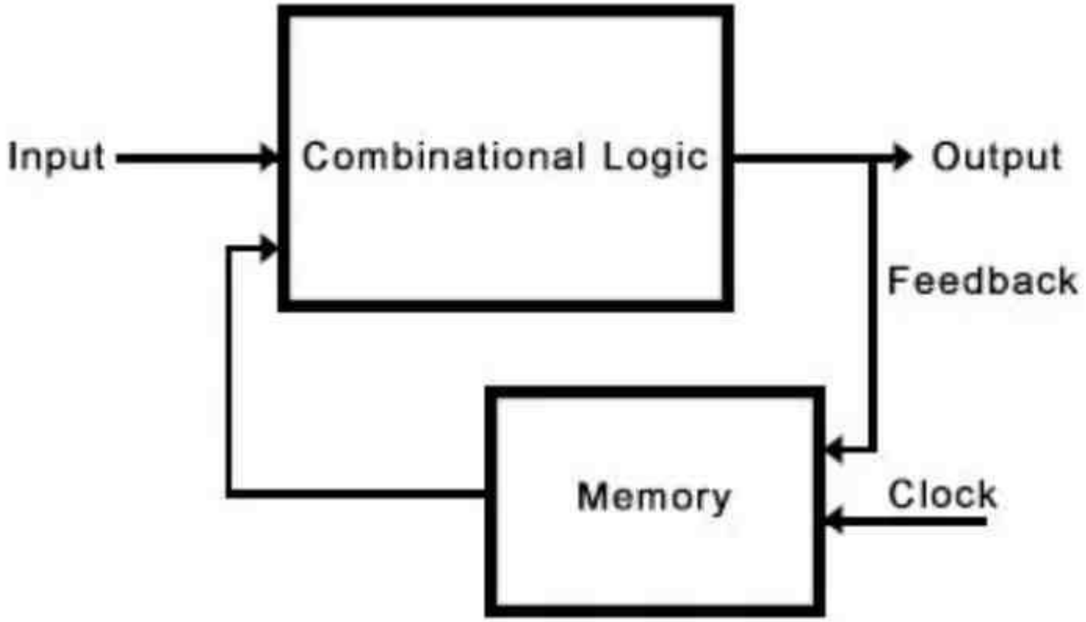
a) 72_8
b) $FA6_{16}$

ASSIGNMENT

LOGIC CIRCUITS

COMBINATIONAL

SEQUENTIAL



LOGIC GATES

AND gate

NAND gate

XOR gate

OR gate

NOR gate

NOT gate

AND Gate

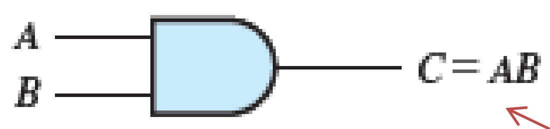
AND gate

- For two logic variables A and B, the “AND” operation represented as – “AB” and read as “A and B”
- Operation: Logical multiplication

Truth table – 2 input AND gate

A	B	$C = AB$
0	0	0
0	1	0
1	0	0
1	1	1

Symbol



Boolean Expression

A and B is 1
iff A is 1 *and* B is 1

OR Gate

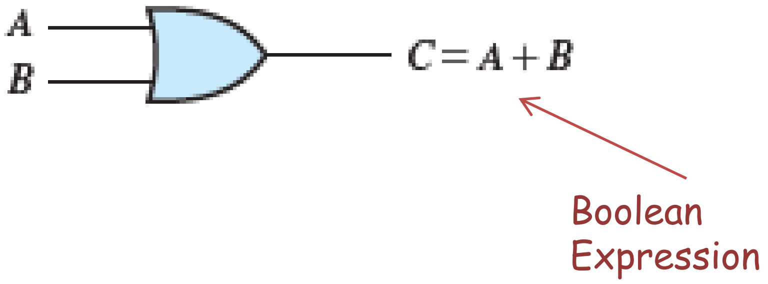
OR gate

- For two logic variables A and B, the "OR" operation represented as – "A+B" and read as "A or B"
- Operation: Logical addition

Truth table – 2 input OR gate

A	B	$C=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Symbol



A or B is 1

if A is 1 or B is 1 or both AB is 1

3-input Gates

3-input AND

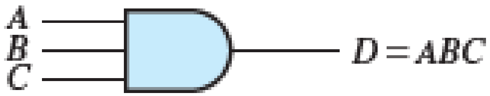
3-input OR

Truth tables

A	B	C	$D = ABC$
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

A	B	C	$D = A + B + C$
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

Symbols



NOT GATE

NOT gate

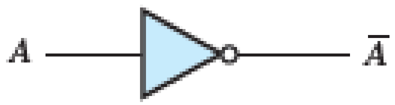
- For two logic variables A and B, the “OR” operation represented by placing bar on top of logical variable - $\overline{A}$ (or $\overline{B}$)

- Read as “Not A” or “A inverse”

Truth table - NOT gate

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

Symbol



NAND Gate

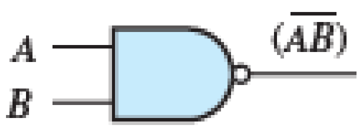
NAND
gate

- AND gate followed by an Inverter – NAND gate
- Symbol: same as AND, with a bubble

Truth table – 2 input NAND gate

Symbol

A	B	C= AB	C= $\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



A nand B is 1
If either A or B is 1 but both not 1

NOR Gate

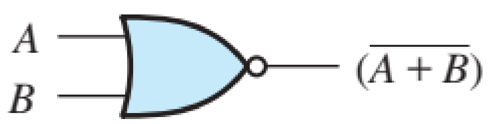
NOR
gate

- OR gate followed by an Inverter – NOR gate
- Symbol: same as OR, with a bubble

Truth table – 2 input NOR gate

A	B	C= *A*+*B*	C= $\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Symbol



A nor B is 1
iff A and B is 0

XOR Gate

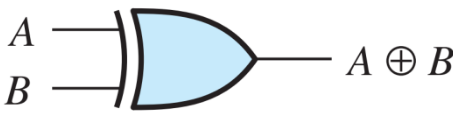
XOR
gate

- XOR gate for 2 logical variable represented by $A \oplus B$
- Defined as
$$0 \oplus 0 = 0$$
$$1 \oplus 0 = 1$$
$$0 \oplus 1 = 1$$
$$1 \oplus 1 = 0$$

Truth table – 2 input XOR gate

A	B	C= $A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Symbol



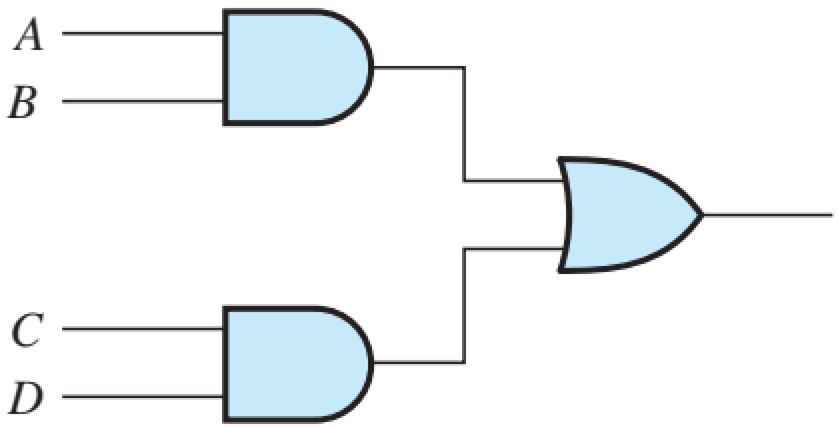
$$A \oplus B = A\overline{B} + \overline{A}B$$

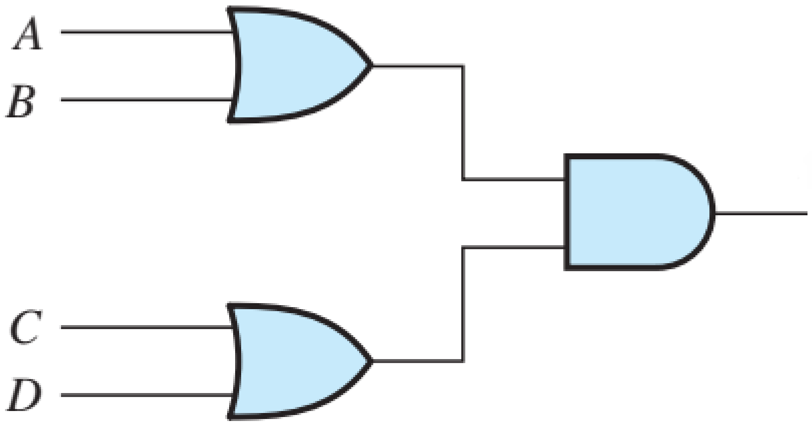
XOR operation: A XOR B is 1
iff only A or only B is 1

It is 0 for if both A and B are 1

EXAMPLES -1

Give the Boolean expression for the logic circuits shown



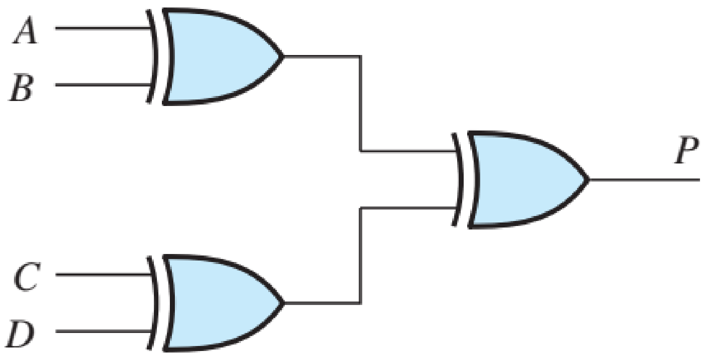


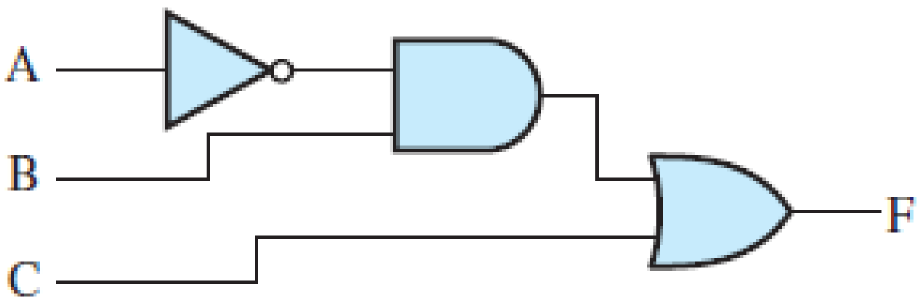
AB+CD

(A+B) (C+D)

EXAMPLES -2

Give the Boolean expression for the logic circuit shown



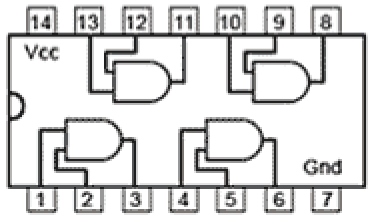


$$P = A \oplus B \oplus C \oplus D$$

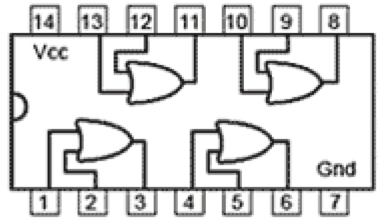
$$F = \overline{A}B + C$$

Logic IC's

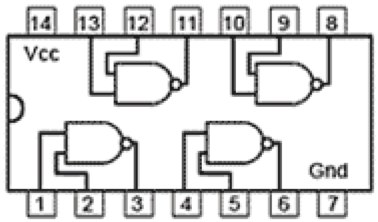
Logic Gates from 7400 series TTL IC family



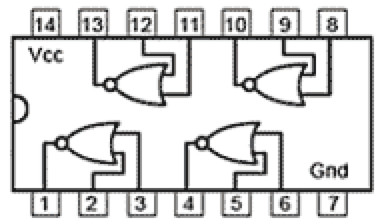
7408 Quad 2 input AND Gates



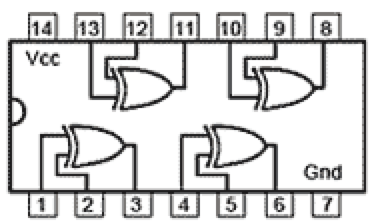
7432 Quad 2 input OR Gates



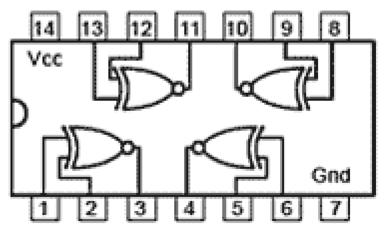
7400 Quad 2 input NAND Gates



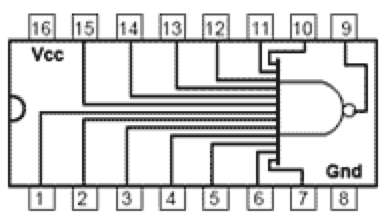
7402 Quad 2 input NOR Gates



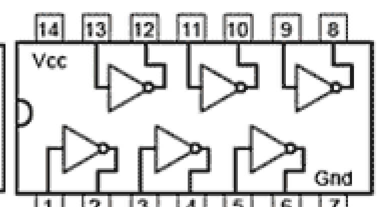
7486 Quad 2 input XOR Gates



747266 Quad 2 input XNOR Gates



74133 Single 13 input NAND Gate



7404 Hex NOT Gates (Inverters)

AND	7408
OR	7432
NOT	7404
NAND	7400
NOR	7402
XOR	7486

Acknowledgements

1. Allan R. Hambley, 'Electrical Engineering - Principles & Applications, Pearson Education, First Impression, 6/e, 2013
2. https://technobyte.org/sequential-combinational-logic-circuits-types/
3. https://learnabout-electronics.org/Digital/dig21.php