Lecture - 1 Digital circuit

- Number systems
- Logic gates
- Logic family
- Combinational & sequential circuit

→ Number system :-

1) Binary No. system
base - 2 or B

digits - 0, 1 ex. $(101)_B$ or $(101)_2$

2) octal No. system

base - 8 or O
digits - 0, 1, 2 ... 7
ex. $(345)_8$

3) Decimal
base - 10 or D
digits - 0, 1, ..., 9
ex. $(189)_{10}$

4) Hexadecimal No. system

base - 16 or H

digits - 0 . — 9

symbol A B C D E F

10 11 12 13 14 15

5) Base - 4

Digits 0, 1, 2, 3

ex. $(12)_4$

0, 1, 2, 3

10, 11, 12, 13

20, 21, 22, 23

In Hexadecimal number system

0, 1, 2 --- 9, A, B, C, D, E, F, 10, 11, 12,
19, 1A, 1B — 1F, 20, 21, 22, --- 29,
2A, 2B, --- 2F

Base - 4 0, 1, 2, 3

Base - 10 0 — 9

Base - 8 0 — 7

Note: largest number in any number system is 1 less than its base.

Conversion:-

(1) Decimal to Binary:

Dividend

$(25)_{10} = (\quad)_2$

Quotient

Remainder

Divisor

2	15	
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

$(25)_{10} = (11001)_2$

$(0.625)_{10} = (\quad)_2$

(2)

$0.625 \times 2 = 1.25 \rightarrow 1$

$0.25 \times 2 = 0.5 \rightarrow 0$

$0.5 \times 2 = 1.0 \rightarrow 1$

$(0.625)_{10} = (101)_2$

$(15.625)_{10} = (11001.101)_2$

* octal to binary :-

$(73)_8 = (\quad)_2$

* Decimal - octal.

$(73)_{10} - (\quad)_8$

8	73	
8	9	1
8	1	1
	0	1

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

* $(73)_{10} - (\quad)_{16}$

16	73	
16	4	9
	0	4

$(73)_{10} = (49)_{16} = (111)_8$

Note :- If base is greter, than number will be smaller, If base is lesser then number will be bigger.

Binary to Decimal.

MSB $\rightarrow$ LSB

$(1001)_2 = (\quad)_{10}$

$(1\ 0\ 0\ 1)_2$

3 2 1 0 = n

$= 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3$

$= 1 + 0 + 0 + 8$

$(1001)_2 = (9)_{10}$

$\rightarrow$ binary to decimal:

$(10.101)_2 = (\quad)_{10}$

$(1\ 0\ .\ 1\ 0\ 1)_2$

1 0 = n -1 -2 -3

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

$= 0 + 2 + 0.5 + 0 + 0.125$

$(10.101)_2 = (2.625)_{10}$

* Binary to octal:-

$(1\ 1\ 0\ 1\ 0\ 1\ 0)_2 = (\quad)_8$

$$\underbrace{001}_{1}\ \underbrace{101}_{5}\ \underbrace{010}_{2} = (152)_8$$

Make group of 3 bit for octal (bcoz using 3 bit you can represent any number in octal)

Binary to hexadecimal

$(10101011)_2 = (\quad)_{16}$

Make group of 4 bit for hexadecimal (bcoz using 4 bit you can represent any number in hexadecimal number system

$$\underbrace{1010}_{A}\ \underbrace{1011}_{B} = (AB)_{16}$$

Convert $(4F)_{16} = (\ldots)_8$

$$4 \quad F$$

$$= (0100 \quad 1111)_2$$

$$= \underbrace{001}_{1}\ \underbrace{001}_{1}\ \underbrace{111}_{7} = (117)_8$$

$(6A)_{16} = (\quad)_{10}$

$(6 \quad A)_{16}$

$= A \times 16^0 + 6 \times 16^1$

$= 10 + 96 = (106)_{10}$

* Gate Question

Q The octal equivalent of $(AB.CD)_{16}$

$(A \quad B \,.\, C \quad D)_{16}$

$(1010 \quad 1011 \,.\, 1100 \quad 1101)_2$

$\underbrace{010}_{2} \; \underbrace{101}_{5} \; \underbrace{011}_{3} \,.\, \underbrace{110}_{6} \; \underbrace{011}_{3} \; \underbrace{010}_{2}$

$(253.632)_8$

Q The binary representation of decimal number (1.375) is

$(1)_{10} = (\ )_2$

$$\begin{array}{c|c|c} 2 & 1 & \\ \hline & 0 & 1 \end{array}$$

$= (1)_2$

$(0.375)_{10} = (\ )_2$

$0.375 \times 2 = 0.75 \rightarrow 0$

$0.75 \times 2 = 1.5 \rightarrow 1$

$0.5 \times 2 = 1.0 \rightarrow 1$

$(0.375)_{10} = (.011)_2$

✓ $(1.375)_{10} = (1.011)_2$

Note // Me
double verifying

$(1.011)_2 = (\ )_{10}$

$(1.011)_2 = 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$

0 (1) -1 -2 -3

$= 1 + 0 + 0.25 + 0.125$

$= 1 + 0.375 = (1.375)_{10}$

01 (23)

01 (22)

8 E

swap

01 (8) di

Arithmatic operation :

ADDITION

$$
\begin{array}{r}
(23)_{10} \\
+(55)_{10} \\
\hline
78
\end{array}
$$

answer

it $8 < \underset{\text{base}}{10}$ = Valid decimal number

No Carry.

$$
\begin{array}{r}
1\ 1\ \ \\
68 \\
+\ 46 \\
\hline
114
\end{array}
\quad 14
$$

$14 > \underset{\text{base}}{10}$ It's not

single digit valid decimal Number so it written as

$14 = \underset{\text{Base}}{10} \times \underset{\text{Carry}}{1} + \underset{\text{Result}}{4}$

Base × Carry + Result

$11 = 10 \times 1 + 1$

addition in Binary

$$
\begin{array}{r}
(1\ 0\ 1)_2 \\
+\ (1\ 0\ 0)_2 \\
\hline
(1\ 0\ 0\ 1)_2
\end{array}
$$

2 < base (2) ✗

2 is not valid binary ~~Num~~ digit

$$2 = \underbrace{2}_{\text{Base}} \times \underbrace{1}_{\text{carry}} + \underbrace{0}_{\text{Result}}$$

* Octal

$$
\begin{array}{r}
1 \\
6\ 7 \\
+\ \ \ 2 \\
\hline
7\ 1
\end{array}
$$

$7 + 2 = 9$ ✗ valid

$9 = 8 \times 1 + 1$

* Hexadecimal

$$
\begin{array}{r}
1 \\
2\ F\ 1 \\
+\ A\ 3\ A \\
\hline
D\ 2\ B
\end{array}
$$

$F + 3 = 15 + 3 = 18$ ✗ valid

$18 = 16 \times 1 + 2$

Ex: ADD $(63)_7 + (56)_8$

different → [illegible] direct add them,

So we have to convert both into same number system then only we can add them.

$(56)_8 = (\quad)_{10} = (\quad)_7$

$(56)_8$

$6 \times 8^0 + 5 \times 8^1$

$= 6 + 40$

$= (46)_{10}$

7	46	4
7	6	4
	0	6

$= (64)_7$

$$\begin{array}{r} {}^{1}\,{}^{1}\quad \\ (64)_7 \\ +\ (63)_7 \\ \hline 160 \end{array}$$

$7 = 7 \times 1 + 0$

$13 = 7 \times 1 + 6$

$\boxed{\text{ans}} = (160)_7$

Subtraction:-

$$\begin{array}{r} 5\ 6 \\ -\ 2\ 3 \\ \hline 3\ 3 \end{array}$$

$\xrightarrow{}$ Base
10+2

$$\begin{array}{r} 5\ \cancel{2} \\ -\ 2\ 3 \\ \hline 2\ 9 \end{array}$$

$\xrightarrow{}$ Base
10+0

$$\begin{array}{r} 6\ 0 \\ -\ 4\ 5 \\ \hline 1\ 5 \end{array}$$

Binary

$\xrightarrow{}$ Base
2+0

$$\begin{array}{r} 1\ 1\ \cancel{0} \\ -\ 1\ 0\ 1 \\ \hline 0\ 0\ 1 \end{array}$$

Octal:-

8+2

$$\begin{array}{r} 6\ \cancel{2} \\ -\ 4\ 5 \\ \hline 1\ 5 \end{array}$$

8

$$\begin{array}{r} 5\ \cancel{0} \\ -\ 2\ 5 \\ \hline 3\ 3 \end{array}$$

Hexadecimal:-

16+A

$$\begin{array}{r} 2\ A \\ 1\ F \\ 1\ \\ \hline 0\ B \end{array}$$

Lecture: 12

Q $(15)_7 - (26)_7 + (53)_7$ Wrong explaination

* Multiplication no system

$$\begin{array}{r} 2\,1\,1 \\ *\ \ 6 \\ \hline 6\,6 \end{array}$$

$$\begin{array}{r} 5 \\ *\ 36 \\ \hline 30 \\ +\ 150 \\ \hline 180 \end{array}$$

* Binary:-

$$\begin{array}{r} 101 \\ *\ \ 11 \\ \hline 101 \\ 1010 \\ \hline 1111 \end{array}$$

* octal :-

$$\begin{array}{r} 74 \\ *\ 21 \\ \hline 74 \\ 170\times \\ \hline 1774 \end{array}$$

$8 = 8 \times 1 + 0$

$15 = 8 \times 1 + 7$

* Hexadecimal :-

AB   16 × 1 + 6

* 22 FFF   16 × 1 + 5

1 5 6

1 5 6 X

1 6 B 6

\# Complements :-

base's Complement   (base-1)'s Complement

ex

binary   2's   1's

Decimal   10's   9's

How to find complement

(1) (base-1)'s complement

subtracting each digit from the largest digit in the number system.

(base-1)'s complement.

ex (405)$_8$ ⇒ 7's complement

$$\begin{array}{r} 777 \\ 405 \\ \hline 372 \end{array}$$

2) base's complement:-

by simply adding 1 to the result of (base-1)'s complement

(405)$_8$ = 8's complement

= 7's complement + 1

$$= \begin{array}{r} 372 \\ +\ \ 1 \\ \hline 373 \end{array}$$

* Calculate 3's complement for (301)$_4$

(base-1)'s complement

~~4's complement~~

3's complement

$$\begin{array}{r} 333 \\ 301 \\ \hline 032 \end{array}$$

~~3's complement: 032 +1 33~~

* Calculate 2's complement for

$(1001)_2$

1	0	0	1
↓	↓	↓	↓
0	1	1	0

1's complement

1111
1001
0110

∴

0110
+ 1
2's complement 0111

direct 1's complement & 2's complement

1001 ⇒ Reverse all bit
1's complement → 0110

1001 ⇒ start from LSB
LSB
Look for first "1"
keep that "1" as it is, after first "1" reverse All bit

2's complement

first

1 0 1 0
↓ ↓ ↓ ↓
0 1 1 0

ex

	1	0	0	1	(1001)
	↓	↓	↓	↓	
2's Complement →	0	1	1	1	

Number Representation

- Magnitude
  - Unsigned Magnitude
  - Signed Magnitude
- Complement
  - 1's Complement
  - 2's Complement

Unsigned Magnitude	Signed Magnitude	1's Complement	2's Complement
→ we can represent only +ve number	→ Both +ve & -ve	→ Both +ve & -ve	→ Both +ve & -ve
→ +6 = 110	→ +6 ~~0110~~ = '0'110 ↓ +ve	→ +6 '0'110	→ +6 = '0' 110

representation of positive Number are same in all case

Unsigned Magnitude

-6

X

Signed Magnitude

→ +6

'1'110

+6 in 6bit

'0'00110

-6 in 6bit

~~111110~~

'1'00110

1's Complement

→ -6

'0'110 ⇒ +6

↓ 1's complement

1001 → -6

-ve

↓ 1's complement

0110

→ 6 is Magnitude

This is representation of -6 in 1's complement form

+6 in 6 bit

'0'00110

-6 in 6 bit

~~111001~~

000110 ⇒ +6

↓ 1's complement

111001 ⇒ +6

or

Copy MSB of -6

Representation of 1's complement

11'1'001

2's complement

→ -6

'0'110 ⇒ +6

↓ 2's complement

1010 ⇒ -6

-ve

↓ 2's complement

0110

→ 6 is Magnitude

This is representation of -6 in 2's complement

+6 in 6 bit

-6 in 6 bit

000110 → +6

↓ 2's complement

111010 → -6

or

Copy MSB of -6

Representation of 2's complement

11'1'010

Singed Magnitude	Decimal
0 0 0 0	+0
0 0 0 1	+1
0 1 1 0	+6
0 1 1 1	+7
1 0 0 0	−0
1 0 0 1	−1
1 1 1 0	−6
1 1 1 1	−7

disadvantages:-

→ 2 different representation for 0

→ Range

-7 to $+7$ 4 bit

$-(2^{4-1}-1)$ to $+(2^{4-1}-1)$

-7 to $+7$

n bit $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

1's Complement	Decimal
0 0 0 0	+0
0 0 0 1	+1
0 1 1 0	+6
0 1 1 1	+7
1 0 0 0	−7
1 0 0 1	−6
1 1 1 0	−1
1 1 1 1	−0

disadvantage:
→ 2 different representation for 0
→ Range 0 1 0

-7 to $+7$ 4 bit

$-(2^{4-1}-1)$ to $+(2^{4-1}-1)$

-7 to $+7$

n bit $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

2's complement	Decimal
0 0 0 0	+0
0 0 0 1	+1
0 1 1 0	+6
0 1 1 1	+7
1 0 0 0	−8
1 0 0 1	−7
1 1 1 0	−2
1 1 1 1	−1

-8 to $+7$ 4 bit

$-(2^{4-1})$ to $+(2^{4-1}-1)$

n bit $-(2^{n-1})$ to $+(2^{n-1}-1)$

* Arithmatic operation using Complement

e) $5 - 4 = 1$

1's complement

$+5 \Rightarrow 0101$
$+4 \Rightarrow 0100$
$\downarrow$ 1's
$-4 \Rightarrow 1011$

$$
\begin{array}{r}
0101 \quad 5 \\
+\ 1011 \quad 4 \\
\hline
\boxed{1}\ 0000 \\
\quad 1 \\
\boxed{0001} \quad +1
\end{array}
$$

This is in 1's complement form

2's complement

e) $5 - 4$

$+5 = 0101$
$+4 = 0100$
$\downarrow$ 2's
$-4 = 1100$

$$
\begin{array}{r}
0101 \quad 5 \\
+\ 1100 \quad 4 \\
\hline
\boxed{1}\ \boxed{0001} \quad +1
\end{array}
$$

This result is in 2's complement form

discard it

Gate: $x = 01110$ & $y = 11001$ are 2 5 bit binary number represented in 2's complement form. The sum of x & y represented in 2's complement format using 6 bit is

(a) 100111 (b) 001000
(c) 000111 (d) 101001

X = 01110 = +14
Y = 11001 = -7
↓ 2's
00111 = +7

14
- 7
+ 7

in 6 bit 2's complement form
1'000111

Lecture:-3

Codes
→ Weighted Code
→ NON-weighted Code

weights
8 4 2 1
1 0 1 0

→ BCD

→ Excess -3
→ gray Code

BCD is always 4 bit

BCD

Decimal	BCD B_3	B_2	B_1	B_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 ⋮ 15	Invalid BCD Code			

to represent greater number

$(23)_{10} \rightarrow (0010 \ 0011)_{BCD}$

Excess -3 code

Excess - 3 = BCD + 3

Decimal	B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

it's also called self complementing code.

→ gray code :- (Cyclic code or unit distance code)

	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

BCD Addition

①

23	0010	0011
+16	0001	0110
39	0011	1001
	↓	↓
	3	9

②

66 ⇒	0110	0110
+55	0101	0101
121	1011	1011
	↓	↓
	11	11

Both Invalid BCD Code

→ to make it valid add $\boxed{0110 = +6}$ to each Invalid BCD Number

	1011	1011
	0110	0110
0001	0010	0001
↓	↓	↓
1	2	1

Convert binary to gray code

e

$\oplus \quad \oplus \quad \oplus$

1 0 1 1

↓

1 1 1 0

$(1011)_2 = (1110)_G$

e

$\oplus \quad \oplus \quad \oplus$

1 1 1 1

↓

1 0 0 0

$(1111)_2 = (1000)_G$

First MSB as it is
then do X-OR operation of 1st
and 2nd bit (from MSB side)
result will be your 2nd bit
of gray code .. same process

a

$\oplus \quad \oplus \quad \oplus$

0 1 0 0

↓

0 1 1 0

$(0100)_2 = (0110)_G$