Lecture - 4 | 19th Aug. 2019

Last Lecture

- Basic digital gates
- Combinational logic blocks
  - Multiplexer
  - DeMultiplexer
  - Decoder
  - Encoder
- Multiway- Multibit logic
- Hardware descriptor Language (BHDL)
- Test bench using (TSC) & Simulation

Todays Lecture :-

- Objective - Designing an Arithmetic & Logic Unit
- Data representation
- Basic arithmetic operations
  - Addition
  - Substraction
- Combinational Logic for arithmetic operation
  - Half adder
  - Full adder
  - n-bit adder & substractor
- Arithmetic & Logic Unit -

Data representation :-

The number system: ~~has~~

- Integer
- Fractional numbers

Integer :-
- Decimal number System (base - 10)
- Octal number System (base - 8)
- Hexadecimal System (base - 16)

A number can be represented as:

$$\overset{4}{a_4}\overset{3}{a_3}\overset{2}{a_2}\overset{1}{a_1}\overset{0}{a_0} = \cancel{+10^2 \times a_2 + 10^1 \times a_1 + 10^0 \times a_0}$$

$$= 10^4 \times a_4 + 10^3 \times a_3 + 10^2 \times a_2 + 10^1 \times a_1 + 10^0 \times a_0$$

if $a_i \in \{0,1,2,3,4,5,6,7,8,9\}$

call it as decimal system where base is $\underline{10}$

$a_0 = 2$
$a_1 = 1$
$a_2 = 4$
$a_3 = 0$
$a_4 = 8$

$a_4\ a_3\ a_2\ a_1\ a_0$

$$\boxed{8\ 0\ 4\ 1\ 2}$$

$$= \boxed{10^4 \times 8 + 10^3 \times 0 + 10^2 \times 4 + 10^1 \times 1 + 10^0 \times 2}$$

$$= b^4 \times a_4 + b^3 \times a_3 + b^2 \times a_2 + b^1 \times a_1 + b^0 \times a_0$$

$$= \sum_{i=n}^{0} b^i a_i \quad \rightarrow \text{for } (n+1) \text{ digit no.}$$

For binary System

$$\boxed{b = 2 \text{ and } a_i \in \{0,1\}}$$

+ve & -ve ~~integer~~ number in binary system:-

Example:- 3 bit number system

$a_2\, a_1\, a_0$	Mag	2's comp
0 0 0	0	0
0 0 1	1	+1
0 1 0	2	+2
0 1 1	3	+3
1 0 0	4	-4
1 0 1	5	-3
1 1 0	6	-2
1 1 1	7	-1
1 $\boxed{000}$	8	

$\boxed{(+a)+(-a)=0}$

$1+7=8$

$2+6=8$

$3+5=8$

$4+4=8$

General form =

$\boxed{2^n - a = -a}$

in 2's complement

How to generat 2's complement number:-

$$\begin{array}{r} 1000 \\ -\ 010 \\ \hline \boxed{110} \end{array} \text{ — } -2$$

Easy way

One's complement +1

$$\begin{array}{r} 010 \\ \rightarrow 101 \\ 1 \\ \hline 110 \end{array} \text{ — } -2$$

$$\begin{array}{r} 001 \\ 1\text{'s} \rightarrow 110 \\ +1 \rightarrow 1 \\ \hline 111 \end{array} \text{ — } -1$$

Overflow & Largest possible number :-

Example 3-bit → $\boxed{-4 \text{ to } 3}$

Example 4-bit → $\boxed{-8 \text{ to } 7}$

General n-bit → $\boxed{-2^{(n-1)} \text{ to } 2^{(n-1)} - 1}$

Arithmetic operation:-

- Addition (+)
- Subtraction (−)

Example (adding two bits)

Combinational cir-

Half adder

Example (Adding three bits)

Combinational cir-

- Full adder
- Using XOR gates
- Full adder using half adder

Example - n-bit number

$\boxed{+5 + (+2) = +7}$ → required 4 bit size

$$
\begin{array}{rcl}
 & 0101 & \text{— } +5 \\
+ & 0010 & \text{— } +2 \\
 & 000 & \\
\hline
 & 0111 & \rightarrow +7
\end{array}
$$

Designing a circuit:-