HindPhotostat

Hind Photostat & Book Store

Best Quality Classroom Topper Hand Written Notes to Crack GATE, IES, PSU's & Other Government Competitive/ Entrance Exams

MADE EASY
ELECTRONICS ENGINEERING
C.O
By-Sagar Sir

- Theory
- Explanation
- Derivation
- Example
- Shortcuts
- Previous Years Question With Solution

Visit us:-www.hindphotostat.com

Courier Facility All Over India
(DTDC & INDIA POST)
Mob-9311989030

HindPhotostat

GET IT ON Google Play

MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

ESE , GATE, PSU BEST QUALITY TOPPER HAND WRITTEN NOTES

MINIMUM PRICE AVAILABLE @ OUR WEBSITE

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE

PUBLICATIONS BOOKS -

MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK

RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON...OTHERS

HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE

HIND PHOTOSTAT AND BOOK CENTER F230, Lado Sarai New Delhi-110030 Phone: 9311 989 030 9560 163 471	Shop No: 46 100 Futa M.G. Rd Near Made Easy Ghitorni, New Delhi-30 Phone:	F518 Near Kali Maa Mandir Lado Sarai New Delhi-110030 Phone:	Shop No.7/8 Saidulajab Market Neb Sarai More, Saket, New Delhi-30

Website: www.hindPhotostat.com

Contact Us: 9311 989 030

Courier Facility All Over India

FREEMIND
Date 29/12/21
Page 141

COMPUTER FUNDAMENTALS

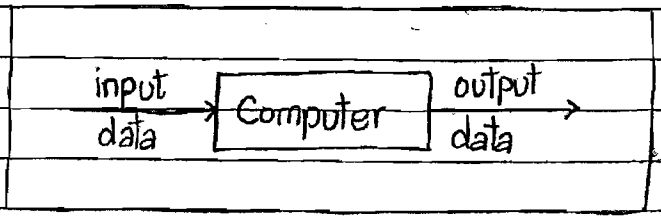
■ SYLLABUS

1. Data Representation
2. Computer Architecture
3. Computer Organization
4. Operating System
5. Networking
6. Programming Elements

■ KEYWORDS

1. Computer

It is a computational machine used to process data under the control of a program applicatinn initiated by the user.



2. Program

Program { Instruction
Data

3. Instruction

It is a binary code designed inside the processor to perform some task.

Binary code - Bind with - Operation

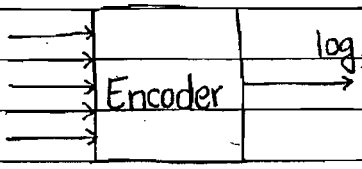
Eg. CPU-X supports 8 different operations, then
opcode size = $\log_2 8$ bits
{binary code} = 3-bit

Say, 000 → multiplication
001 → subtraction
101 → additions & so-on

These opcodes are decided by the designed

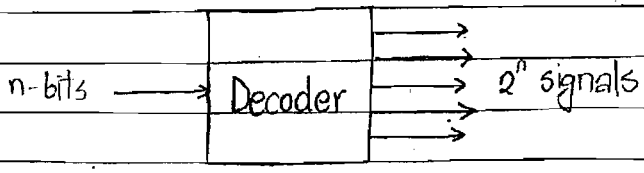
4. Encoding

In this process, N signals are represented using $\log_2 N$ bits



5. Decoding

In this process, an n-bit decoder produces 2^n o/p signals.



6. Data

Data is a binary code which is associated with a value based on the data format.

Binary code - Bind with - value

Eg. $(101)_2$
- unsigned → 5
- sign mag. → -1
- 2 comp. → -3
- 1's comp. → -2
- Floating pt. → fraction

DATA REPRESENTATION

In the computer system, data is always represented in binary. Different formats are used in the Computer system to defined the data, described as follows:

DATA FORMATS

- Fixed Point
  - Magnitude
    - Unsigned → only +ve data → $(0 \text{ to } 2^n - 1)$
    - Sign Magnitude → both +ve & -ve data → $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$
  - Complement
    - 1's Comp. → +ve & -ve data → $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$
    - 2's Comp. → +ve & -ve data → $-(2^{n-1})$ to $+(2^{n-1}-1)$
- Floating Point
  - → single precision (32-bit)
  - → double precision (64-bit)

□ Fixed Point Data

4-Bit Code	Unsigned	Sign Magnitude	1's Comp.	2's Comp.
0000	0	(+0)	(+0)	+0
0001	1	+1	+1	+1
0010	2	+2	+2	+2
0011	3	+3	+3	+3
0100	4	+4	+4	+4
0101	5	+5	+5	+5
0110	6	+6	+6	+6
0111	7	+7	+7	+7
1000	8	(-0)	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3
1110	14	-6	-1	-2
1111	15	-7	(-0)	-1

- Consider a hypothetical system with a word length of 12 bit. What is the range of a 2's complement data possible in this system?

→ n = 12 → range : -2^{11} to $2^{11}-1$

: -2048 to +2047

• What is the range of a signed data?

→ Signed ≠ Sign Magnitude.

Signed data { Sign Magnitude
1's complement
2's complement

when a particular format isn't specified, the default format i.e. 2's complement is taken.

∴ range : −128 to +127

• What is the range of a data possible to process on a 8-bit CPU?

→ Since the data type itself isn't specified, so we consider default type i.e. unsigned data.

∴ range : 0 to 255

* Sign Extension

This concept is used in the signed data, to preserve the sign of a data when the data is stored in a large storage space.
Sign Extension means replication of MSB of data

E.g. −7 : 1001 → store in a 8-bit space

+9 ← 00001001 (0's padding / zero's padding) ✗

11111001 (MSB replication / sign extension) —2's comp→ −7 ✓

Note: 0's padding is used in Unsigned data & Sign extension is used in Signed data

□ Bit Overflow

Eg. 4 bit data

```
  1111   +15
+ 1111   +15
(1) 1110  +30
 ↳ bit overflow {carry}
```

→ Carry Flag holds the range exceeding condition of an unsigned data, represented by Cy

n-bit + n-bit = (n+1)-bit
↳ carry flag is this additional bit

if Cy=1, then carry is present i.e. range has been exceeded

if Cy=0, no carry is present.

Eg. 4-bit data

```
 1000    8
 1001   +9
(1)0001  17
↳Cy=1 ⇒ out of range
```

□ Multiplication

→ In the multiplication process, two steps are performed.
i) generation of partial products
ii) summation of partial products

→ Partial products are generated based on multiplier bits i.e. when multiplier bit is 1, multiplicand itself comes out as partial product else partial product is 0.

→ After the generation of p.p., provide the summation to produce the final product.

```
       multiplicand   multiplier
E.g.       1111   *   1111
       10 1
           1111  }
       011         } partial
          1111    } products
      011
         1111
     010
        1111
      11100001  → final product = 225
      (8-bit)
```


Note: (n bit) * (n bit) = 2n bits
→ register "pair" is used to store final product

· Consider the following multiplication

$$(10w1z)_2 * (15)_{10} = (y0101\,1001)_2$$

what are the values of w, y & z ?

→ $(10w1z)_2 * (1111)_2 = (y01011001)_2$

$$\begin{array}{l} \;\;\;\;\;\;\;\;\;\;\;\;\;\;\;\;\; 10w1z \\ \;\;\;\;\;\;\;\;\;\;\;\;\;\; 10w1z \\ \;\;\;\;\;\;\;\;\;\; 10w1z \\ \;\;\;\;\;\; 10w1z \\ 1010110 0z \end{array}$$

$1+w+1+z = 0$

$1+w+1+1 = 0$

$\Rightarrow \underline{w=1} \rightarrow cy = 2$

$\underline{y=1}$ ↙ ↳ $\underline{z=1}$, else $(4w+z+18)*15 = (256y+89)$

$\Rightarrow (w,y,z) \equiv (1,1,1)$

□ Division

→ In the division procedure, bits are scanned from MSB to LSB in bit-wise sequence.

→ After scanning, dividend value is compared with divisor value.

→ If Dividend ≥ Divisor, then put '1' in quotient & subtract the divisor from the dividend & scan the next bit.

→ If Dividend < Divisor, then put '0' in quotient & scan the next bit

→ Continue the process till all the bits are serviced.

Eg. 00000111 ÷ 0010