Hindbookcenter

GET IT ON Google Play

ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE

COURIERS SERVICE AVAILABLE

MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

ESE , GATE, PSUs 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

Shop No.7/8 Saidulajab Market Neb Sarai More, Saket, New Delhi-30	Shop No: 46 100 Futa M.G. Rd Near Made Easy Ghitorni, New Delhi-30	F518 Near Kali Maa Mandir Lado Sarai New Delhi-110030	Shop No.7/8 Saidulajab Market Neb Sarai More, Saket, New Delhi-30

Website: www.hindbookcenter.com

Contact Us: 9711475393

16 August 2

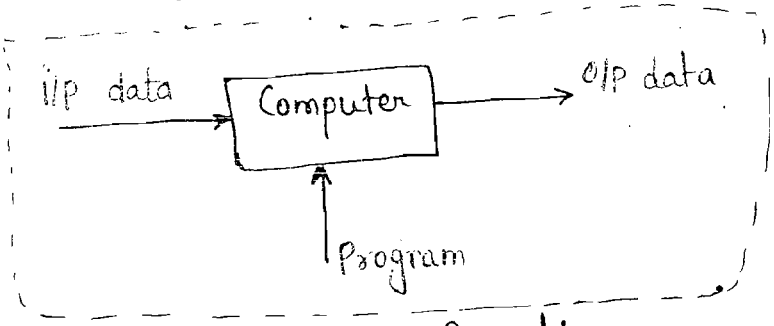
Contents

Chapter-1

Data Representation

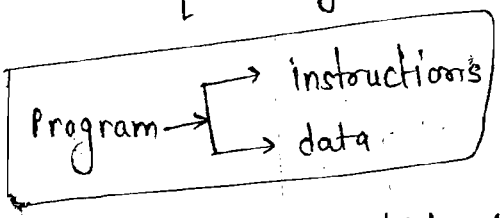
keywords

(1) Computer ⇒ it is a computational machine used to process the data under the control of a application program which is initiated by the user(or) programmer therefore computer system functionality is program execution.



Program Execution

(2) Program ⇒ Program is a sequence of instructions along with a data.



3) Instructions ⇒ It is a binary code which is designed inside the procesor to perform some task.

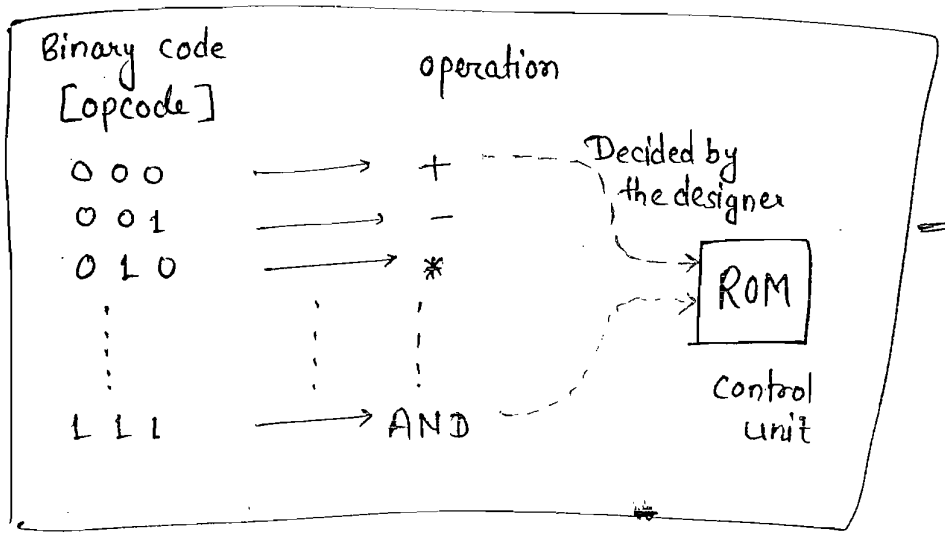
Binary code – Bind with – operation

Example:- Suppose CPU-X supports '8' bit different operation then

opcode size = $\log_2 8$ bit

$= \log_2 2^3$ bit

$= 3$ bit



⇒ Prepare the Instruction manual

(4) Design time ⇒

this flow diagram related to designer view (top to bottom)

designer view

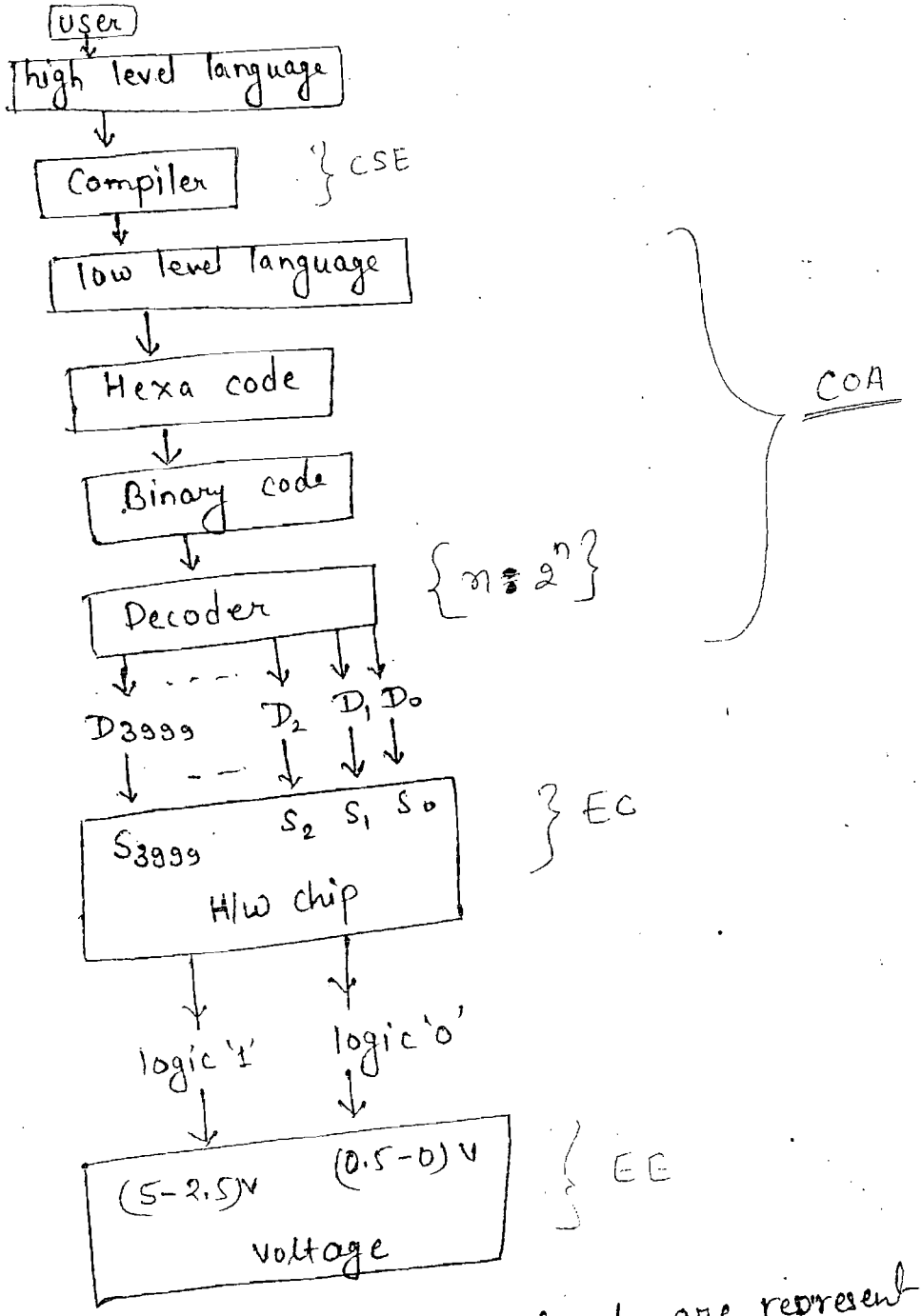
EE { Voltage (5 – 2.5)V (0.5 – 0) → logic '1' / logic '0'

EC { hardware chip $S_3 S_2 S_1 S_0$ → encoder { N: $\log_2 N$ bit } → Binary code → Hexa code → low-level language → Assembly code

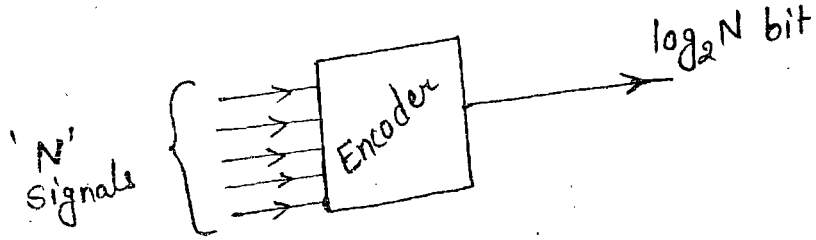
COA

CS { Compiler (SIS of software) → high level language → C code → User

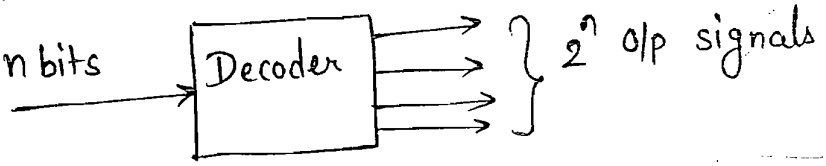
User view



5) Encoding :- In this process, 'N' signals are represented with $\log_2 N$ bit format.



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



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

Binary code	—	Bind with	—	Value

eg $(101)_2$: 5, -1, -2, -3, fraction

$(1\times2^2)+(0\times2^1)+(1\times2^0)$

$4+0+1$

5 — unsigned format (by default)

101 : -1 — sign magnitude format

101 : -2 — 1's complement format

101 → 010 (2)

101 : -3

101 → 010

+ 1

011 (3)

Floating Point format

2's complimed format

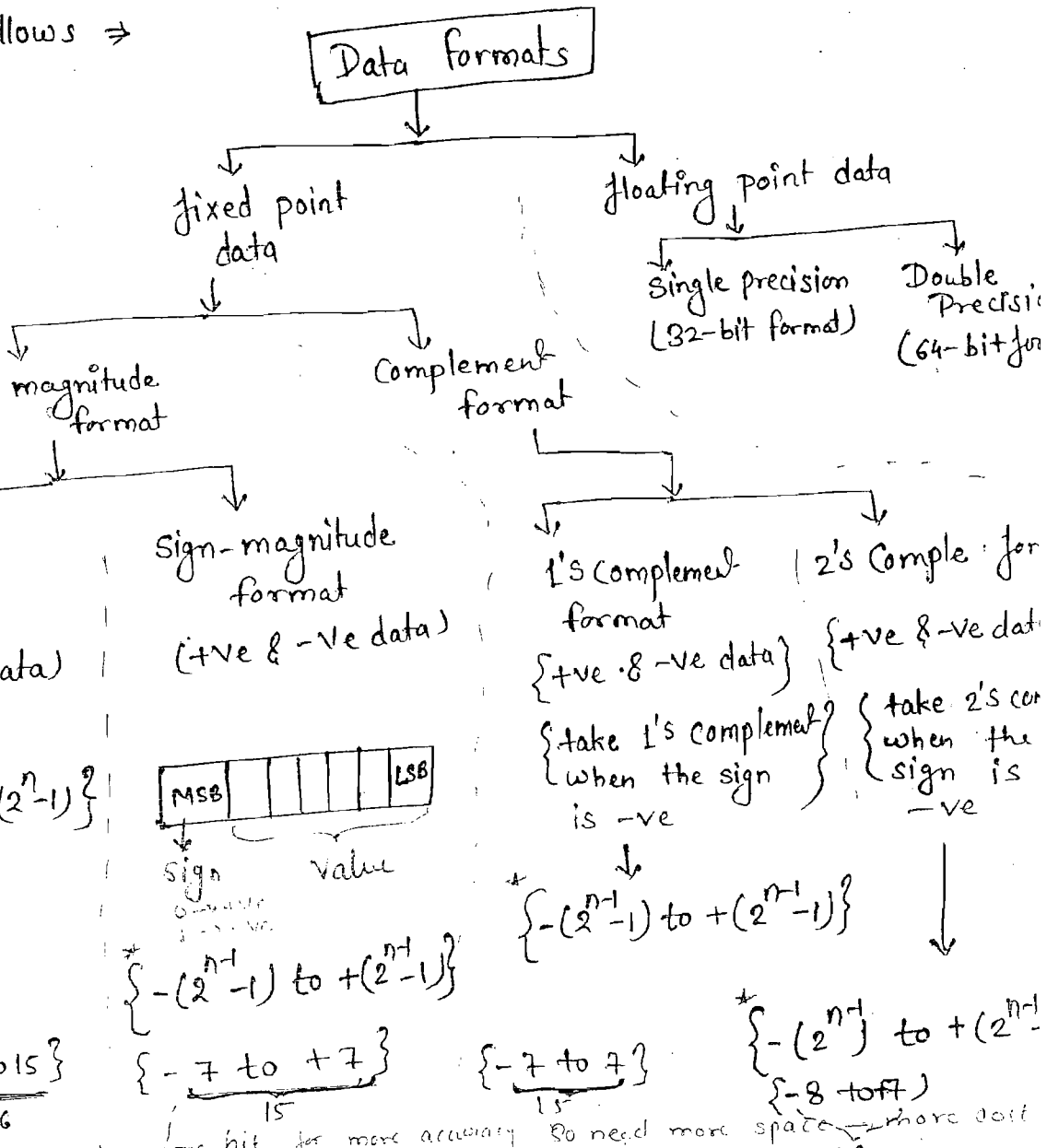
[Computer] — [user]

{Base=2} {Base=10}

$(101)_2$ → ? Value → {5} data

$(101)_2$ → ? operation ↑ decided by designer

Different data formates used in the computer system is as follows ⇒



* fixed point data gives more bit for more accuracy so need more space → more cost

* floating point → less bit for less space → less cost ↓ so if 8 bit processor → 0 to 2^8-1
word length ↑ so data ↑ so accuracy ↑ and 16 bit proces → 0 to $2^{16}-1$
(16 bit processor) > (accuracy

* redundant → duplication

→ ½ code is wasted here due to redundel- (duplication of 0) so can't use this format in computer

Binary +ve data → 1's compl → 1's compl = +ve value

fixed-point data

n bit CPU :- word size = n bit
{Data size}

eg → 4 bit CPU :- Data size = 4 bit

4 bit binary	unsigned Data	sign-magnitude Data	1's complement Data	2's complement Data
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

Same as sign-mag for +ve

for 0000 → +{ }
0111
Sign Value

1000 → -{ }
1111
Sign Value

1000
-7
1000
0111
7

1001 → -6
1001 → 0110 = 6

1000 → -8
1000 → 0111
+1
1000
8

1001 → -7
1001 → 0110
+1
0111 = 7

Conclusion

Data type → data format

fixed point data
- unsigned data (only +ve data) → unsigned format
- signed Data (+ve & −ve data)
  - sign-magnitude format
  - 1's compl. format
  - 2's compli format

} Not in use (contain 2 possible codes)

So **bydefault signed data is 2's compl data**

Note

2's complement of "N" & "−N"

eg 2's compl of '5' & '−5'

4 bit code

5 : 0101
↓ 2's compli
1010
+ 1
−5 ← 1011

7 : 0111
↓ 2's comp
1000
+ 1
−7 ← 1001

unsigned data
↓
7 : 0111
↓ stored in 8 bit register

0	0	0	0	0	1	1	1	: 7

0's padding

signed data
↓
−7 : 1001
↓ stored in 8 bit register

0	0	0	0	1	0	0	1	: 9

0's padding

sign extention

1	1	1	1	1	0	0	1	: −7

11111001 → 00000
+
00001

Q) Consider the following binary code 10001010
what is its decimal equivalent when the data is in

(1) Unsigned
(2) sign - magnitude
(3) 1's compl
(4) . 2's comple format

Sol^n binary code → $2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$
1 0 0 0 1 0 1 0

$8 \times 8 \times 2$
64×2
128

(i) unsigned : $= (1\times2^7)+(1\times2^3)+(1\times2^1)$

$= 128 + 8 + 2$

$= 138$

(ii) Sign - magnitude = 1 000 1010
8+2
(10)
↓
−10

iii) 1's complement of 1000 1010 : (−117)

1's 1000 1010
0111 0101
$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$ → $2^6+2^5+2^4+2^2+1 = 117$

iv) 2's complement of 1000 1010 : (−118)

1000 1010
0111 0101
+ 1
0111 0110
$2^6+2^5+2^4+2^2+2^1 = 118$

Q) which of the following codes are redundent codes in the cpu using sign - magnitude & 1's comp format.

(i) 00 & AF
(ii) 00 & 80
(iii) 00 & FF
(iv) 00 & CF

Sol^n

sign magnitude:
0000 0000 : +0
1000 0000 : −0

1's complimeny:
0000 0000 : +0
1111 1111 : −0