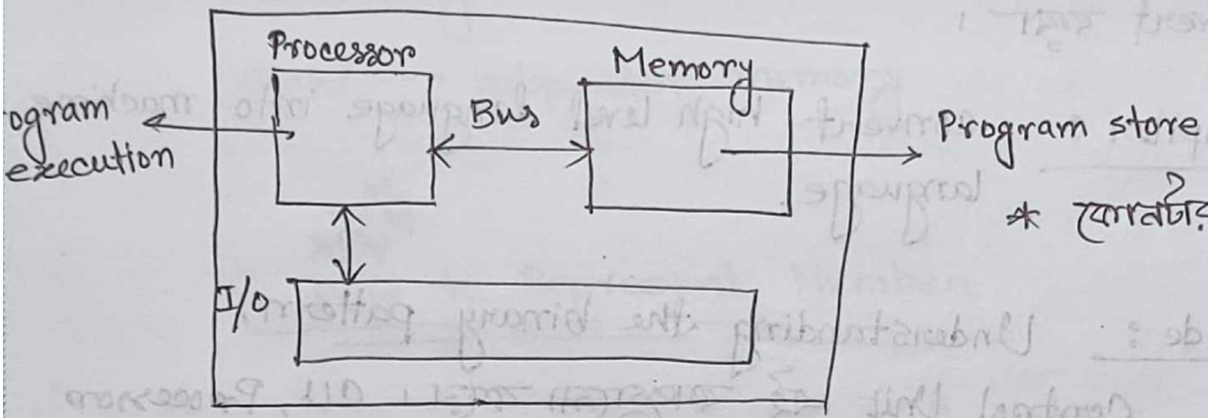
Subject.......
Date: 12.03.2022 Time: Sat
L-1

Computer → compute or executing program

→ processor's job



* কোনটার কাজ কী?

Input & output Device

Keyboard, mouse

Monitor, Printer

Memory

Program → a set of instruction

Bus → connect করে রাখে

Processor এর ভিতরে আছে:

→ Control Unit → Fetch decode

→ Execution Unit → Execution

Memory থেকে Bus এর মাধ্যমে Fetch করে decode করে program executed হয়।

* Processor এর কাজ কী? → Fetch, Decode, Execution

* What is Decode? → Convert এর পর আমার instruction টাকে understand করাটাই decode.

→ Machine Learning মানেই 0 and 1.

Compiler এর কাজ program কে machine Language এ convert করা।

Compiler : convert high level language into machine language.

Decode : Understanding the binary pattern. Control Unit এই কাজগুলো করে। CU, Processor এর অংশ।

Assembler :
Converting to assembly Language into machine language.

Opt Code:

$$\text{Program} \xrightarrow{\text{Compiler}} \text{Memory} \longrightarrow \underbrace{\underbrace{\text{fetch} \to \text{Decode}}_{\text{CU}} \to \text{Execution}}_{\text{Processor}}$$

ALU : Arithmatic Logic Unit:
execution এ help করে
Arithmatic operation

Subject........ Date:........ Time:........

Memory : stores the program

→ primary → RAM , ROM
→ Secondary → Resister, FF

Cache → independent memory

How to Represent Number

→ Signed Number (can be + & -)
→ Unsigned Number (always +)

How to represent negative number?
→ 2's compliment এ use করা হয়।

কেন 2's complement use করি অন্য method use না করে?
3টা method of number representation →

① Sign Magnitude :

3 bit = 2^3 = 8 combinations

Half positive Half negative numbers

+3 → 011	−1 → 101
+2 → 010	−2 → 110
+1 → 001	−3 → 111
0 → 000	

— — —

sign magnitude

Scanned with CamScanner

Subject............................
Date:.............................Time:

① Sign Magnitude
② 1's compliment
③ 2's compliment

sign magnitude এ 100 represent করা যায় না cz it denotes negative zero which is impossible. So, it's a drawback.

1's compliment :

0 1 1
0 1 0
0 0 1
0 0 0
1 1 0
1 0 1
1 0 0

এক্ষেত্রে, 111 missing.

এক্ষেত্রে, negative number এ sign position বাদে magnitude এর জায়গায় 0 থাকলে 1 হবে, 1 থাকলে 0।

SO, 111 ⟶ 1 0 0 = −0

যেটা negative zero denote করে which is impossible.

2's compliment :

(1's compliment + add 1) → negative number হলে
positive number always unchanged.

+3	→ 011
+2	→ 010
+1	→ 001
0	→ 000
−1	→ 111
−2	→ 110
−3	→ 101

$-3 \rightarrow 011$

$$\begin{array}{r} 100 \\ +1 \\ \hline 101 \end{array}$$

* short cut way for 2's compliment :

From the right hand side, copy the number as it is, till you get first 1, then invert everything.

→ first 1 টা বসবে, এর পর থেকে যা যা থাকবে তা invert হবে।

এখানেও,
100 missing.

After checking,

100
negative, so এটার আবার 2's compliment করব

100 = 4 → এখন 100, 4 কে represent করে।

$$\begin{array}{r} 011 \\ +1 \\ \hline 100 \end{array} = 4 \text{ (Binary)}$$

2's compliment এ কোন negative zero নাই, আমরা এক্ষেত্রে unique combination পাই তাই আমরা 2's compliment use করি।

Scanned with CamScanner

Binary Number :

16 8 4 2 1

2^4 2^3 2^2 2^1 2^0

3 bit → $2^3(8)$ → (-4, -1) (0, 3)

4 bit → $2^4(16)$ → (-8, -1) (0, 7)

8 bit → $2^8(128)$ → (-128, -1) (0, 127)

5 → +5 → 0101

5 → -5 → 1011

(3bit) 101 → এটা 2's compliment এ নিয়ে then বলব যে এটা কী।

1 → sign (negative)

$$\begin{array}{r} 101 \Rightarrow 010 \\ +1 \\ \hline 011 = +3 \end{array}$$

অর্থাৎ, 101 হল (-3)।

0000000001 → positive number এ finite leading zero থাকতে পারে।

in a same way, 1111111101 → negative number এ finite leading 1 থাকতে পারে।

Subject: Mon

Date: 14.03.2022 Time: Lec-2

3 bit / 4 bit

↓ ↓

$2^3 = 8$ $2^4 = 16$

(-4, -1) (0, 3) (-8, -1) (0, 7)

5 → +5 = 0101
5 → -5 = 1011

* 7 → +7 = 0111
7 → -7 = 1001

9 → +9 = 00001001
9 → -9 = 11110111

এখানে 4টা না দিয়ে একটা দিলেই হবে।

67 → +67 = 01000011
67 → -67 = 10111101

128	64	32	16	8	4	2	1
	1	0	0	0	0	1	1

85 → +85 = 01010101
85 → -85 = 10101011

Subject.......
Date:.......Time:.......

Floating Point Number

56.7
78.2
42.5

fixed position number

0.0158
1.275
23.5

floating point number → Decimal point fixed না।

* Any group of number where the position of the number is not fixed is called floating point numer

Computer এ আমরা কিভাবে দশমিক represent করব এটা বুঝাতে floating point লাগে।

73.2
649
.847
5.63

Convert the floating point into fixed point
└→ Normalization Process

fixed point এ যোগ বিয়োগ করা easy.

$73.2 \rightarrow 7.32 \times 10^{1}$

$649 \rightarrow 6.49 \times 10^{2}$

$.847 \rightarrow 8.47 \times 10^{-1}$

$5.63 \rightarrow 5.63 \times 10^{0}$

একটা list দিয়ে রাখবে এগুলোর Normalization করো।

Floating Point Numbers :

73.2	7.32×10^1
649	6.49×10^2
.847	8.47×10^{-1}
5.63	5.63×10^0

point কখনো pc তে store করা যায় না। এটা register এ অন্যভাবে stored হয়।

Fixed point এর ক্ষেত্রে 1st এ point ignore করে then last এ বসায় ~~scale~~ করে।

But floating point এর ক্ষেত্রে point store ও করা যায় না। আর ignore করলে থাকবে না। তাই floating point কে fixed point এ convert করতে হয়। একে Normalization বলে।

Normalization :

Rule হল just দশমিকের left side এ only one non-zero number থাকবে।

There should be only one non-zero digit to the left of the point.

Exponent → position of the point represent করে

Mantissa → number এর digit represent করে

5.63×10^0 $\quad \boxed{E \mid M}$ E = 0, M = 563

$0101.001 = \cancel{0.101} \; (-1)^0 \times 1.01001 \times 2^2$

$11111.01 = (-1)^0 \, 1.111101 \times 2^4$

$0.00010 = \cancel{0.0010 \times 10^2} \; (-1)^0 \times 1.01 \times 2^{-3}$

$-10.01 \Rightarrow = (-1)^1 \, 1.001 \times 2^1$

এক্ষেত্রে base 2. দশমিক left এ কত ঘর সরবে। left এ (+) right এ (−)

Binary normalization এ always দশমিকের আগে 1 থাকে। তাই এটা extra করে store করি না।

E	M

E → এক্ষেত্রে base 2

M → এক্ষেত্রে Mantissa just দশমিকের পরের digit গুলো হবে।
1.M format

Negative value গুলোর জন্য different problem:

cz PC কে hyphen set করা যাবে না।

এক্ষেত্রে,

$(-1)^s$ থাকে $s=0$ positive
$s=1$ negative

S.E.M (digit of the number without 1 &e.)

$$\boxed{(-1)^s \times 1.M \times 2^E}$$

→ Normalize form for the binary pattern.

ex: $M=1101$
$E=3$.

$1.1101 \times 2^3 = 1110.1$

Floating point এর দুইটা structure.

*** IEEE-754-32 bit format (single precision)

1	8	23 bit
S	E	M

*** IEEE-754-64 format (Double precision)

1	11	52 bit
S	E	M

Subject.........
Date:........../Time.....

Prob যদি exponent এ minus আসে?

$$\begin{array}{r} 3\times10^{3} \\ (+)\ 2\times10^{3} \\ \hline 5\times10^{3} \end{array} \qquad \begin{array}{r} 3\times10^{3} \\ 2\times10^{4} \end{array}$$

extra কাজ করা লাগবে exponent এর জন্য যদি neg হ

এই prb solve করবে।

Biased exponent :

এখানেও সময় লাগে But কম।

~~সময় কম লাগে~~ কেন?

↳ ez এখানে একবারই change হ

then ঐ change টাই সব জা

used হয়।

$$\boxed{\begin{array}{c} \text{Bised Exponent} = \text{True Exponent} + \text{Bias} \\ BE = TE + Bias \end{array}}$$

Bias হিসেবে কত নিব।

Bias যদি 200 হয় negative side এ তাহলে আমরা positive number কম নিতে হবে।

For 32 bit case : 8 bit

Single precision

$2^8 = 256$

200 + 56

↙ neg ↘ pos যা কম।

তাই 127 Bias নেই

Bias selection → support equal number of positive & negative

64-bit :

Double Precision

11 bit exeponent

$2^{11} =$

1023 Bias

Q: Why we use bias?

Q: Bias 127 হয় কেন?

* **Q:** Convert $(14.125)_D \rightarrow$ single precision

→

Step:
1. convert Binary
2. Normalization
3. Take the bias
4. Convert Bias exponent into binary
5. Substitute into required format

1110.001

1.110001×2^3

$0.125 \times 2 = 0.25 \quad 0$
$0.25 \times 2 = 0.5 \quad 0$
$0.5 \times 2 = 1 \quad 1$

$BE = 3 + 127 = 130$

$(130)_D = (10000010)_B$

S	E	M (23 bit)
0	10000010	11000 1000....

এখানে 23 পর্যন্ত নিব এবং দশমিকের পরে দেখে হবে জিরো বসাবো

Subject: 21.03.2022
Date: Mon Time: Lec-4

Floating Point Number:

S	E	M
1	2	563

$\Rightarrow -5.63e^2 = -5.63 \times 10^2$

$\Rightarrow -563$

S	E	M
1	1	001

$\Rightarrow$ Normalize form $\Rightarrow (-1)^1 \times 1.001 \times 2^1$

$\Rightarrow -10.01$

exponent এ neg sign দিলে extra work করতে হয় প্রতি calculation এ. এটা solve করতে Bias করা হয়।

Q. Bias কেন করা হয়?

→ To make the calculation faster. কেন? exponent এ minus sign দিলে addition করার সময় প্রতিবার compare করতে হয়। এক্ষেত্রে extra steps লাগে। আর addition এ extra step মানে multiplication, divide etc সব ক্ষেত্রেই 10 গুণ or extra steps লাগে।

Q: Biasing এর number কত ধরা হয়?

Q: Single precision 127 কেন?

BE = 127 & BE = 1023 (Double precision)

Exp: 8, $2^8 = 256 \longrightarrow \frac{255}{2} = 127.5$

So that it can support equal number of positive and negative.

Subject............ Date:............ Time:............

Q: 0.00101 represent the number in double precision.

$0.00101 = 1.01 \times 2^{-3} = (-1)^0 \times \boxed{1.01 \times 2^{-3}}$

S	E	M
~~0~~	~~3~~	~~01~~

2 expo কে এভাবে ~~রাখি না~~ দেখেই Biasing করি।

right side এ ~~দশমিক নিলে~~ minus হবে। left এ নিলে plus হবে।

Bias exponent = $1023 - 3 = 1020$

$(1020)_D$ = ~~10000000010~~
1111111100

S	E	M
0	~~0000000000~~ 0 1111111100	0100000 000 → 0

Mantisa এর ডানে zero দিব cz mantissa, decimal point এর পরের number represent করে, তাই এর right এ zero বসালে value change হবে না।

Exponent এর ক্ষেত্রে left এ zero বসাতে হবে, bit fill up করার জন্য।

0101.001 → single precision

$\Rightarrow (-1)^0 \times 1.01001 \times 2^2$ ⇒ Normalized form

Mantissa always single precision এ 23 bit দিয়ে represented হয়।

BE = 127 + 2 = 129

S	E	M
0	10000001	01001000 → 0

Subject..

Date:..........................Time:........................

সামনে always 0 ই দিব cz Bias টা কখনো neg ধরা হয় না।
তাই 1 হবে না।

Book : William Stalling → W

Q: convert $(14.125) \rightarrow$ single precision

$\rightarrow (1110.001) = (-1)^0 \times 1.110001 \times 2^3$

BE = 130

0	10000010	110001 → 0

Q: (-14.125)

$\rightarrow (-1)^1 \times 1.110001 \times 2^3$

S	E	M
1	10000010	110001 → 0

Q: $(2A3B)_H$

2 → 0010, A → 1010, 3 → 0011, B → 1011

$\Rightarrow (0010101000111011) = (1.0101000111011)$

$= (-1)^0 \times 0101000111011 \times 2^{13}$

BE = 13 + 127 = 140

1 bit	8 bit	23 bit
0	10001100	0101000111011 → 0

Mon
Subject: Lec-12
Date: 23.05.22 Time:

III Algorithm for adding floating Point Number

$010110 \longrightarrow 1.\underbrace{01}_{M} \times 2^{2}$ E

S	M	E

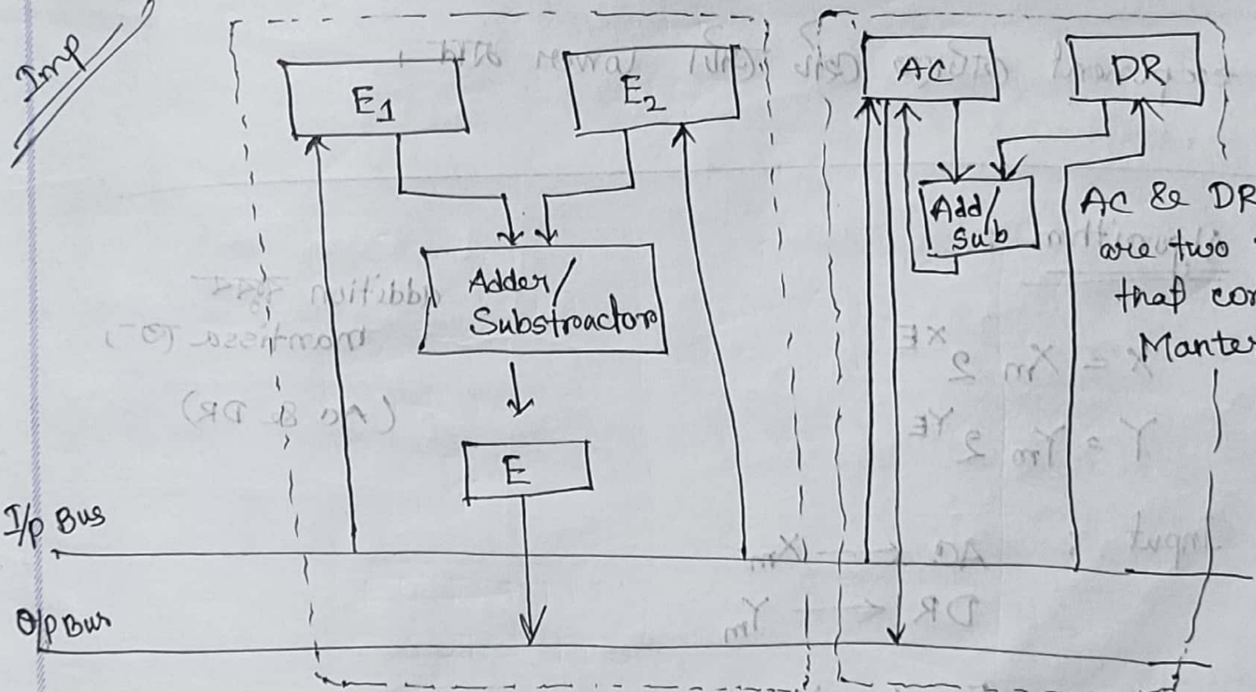
S	M	E

101
010
111

$(-1)^S \times 1.M \times 2^E$

4×10^2
42×10^4
8.2×10

Imp



AC & DR are two register that contain Mantessa

Subject........
Date:........ Time........

Algorithm : Decimal

4×10^3	4×10^4	4×10^3
2×10^4	2×10^4	20×10^3
	2.4×10^4	24×10^3

Normalized form (better)

* Always get the lower one high.

ALU → Exponent unit
ALU → Mantissa unit

exponent যেটাতে ছোট সেটা lower মান।

Algorithm :

$X = X_m \, 2^{XE}$

$Y = Y_m \, 2^{YE}$

addition করবে mantissa তে, (AC & DR)

Input :

$AC \leftarrow X_m$

$DR \leftarrow Y_m$

$E_1 \leftarrow XE$

$E_2 \leftarrow YE$

$AC.\text{overflow} \leftarrow 0$

$\text{Error} \leftarrow 0$

Compare : $E \leftarrow E_1 - E_2$

Equalize : If $(E < 0)$ then

{ $AC \leftarrow RS(AC)$

$E \leftarrow E + 1$

go to equalize

}

if $(E > 0)$ then

{ $DR \leftarrow RS(DR)$

$E \leftarrow E - 1$

go to equalize

}

RS = Right shift

↳ We store the position of the point.

4×10^3

2×10^5

4×10^5

$E = 2$

$E = -1$

Add :

$AC \leftarrow AC + DR$

$E \leftarrow \max(E_1, E_2)$

Overflow : If (AC overflow = 1) then

{ if ( ) then goto Error

$AC \leftarrow RS(AC)$

$E \leftarrow E + 1$

go to End

}

Defination of AC overflow :

It is a flag that will automatically set by ALU as 1 if there are more than one digits to the left of the point

Input
comp
equalize
Add

overflow
normalize
Zero
underflow

E can only become E+1, if there is room for E to become E+1.

Normalize : (If AC normalized) then
go to End

Zero : If (AC=0), then E ← 1

Error flag

Result = 0

Normalize form

1.01

Underflow :

if ($E > E_{min}$) then

AC ← L.S (AC)

E ← E − 1

go to Normalise

{
}

E = Error flag

We can't represent 0 in normalized form as in this form there should be at least one point to the left of the point.

→ decimal

Error : Error ← 1

End : End of process

0.2×10^3

2.0×10^4

number left shift

——— 6 ———

(Imp Topic)

Subject.......................

Date:.................... Time:..................

Q: 0.00062×10

5.4×10^5

floating point number 2 add = ? & print in every steps.

$X = 6.2 \times 10^3$

$Y = 5.4 \times 10^5$

I/P : $AC \rightarrow 6.2$ $\quad E_1 \rightarrow 3$

$DR \rightarrow 5.4$ $\quad E_2 \rightarrow 5$

AC overflow $\leftarrow 0$

Compare : $-2 \leftarrow 3-5$

Equalize : $(-2 < 0)$

$0.62 \leftarrow RS(6.2)$

$E = -1$

$0.062 \leftarrow R_s(0.62)$

Sweet Bird

Wed
Subject: Lec - 13
Date: 25.05.22

CT - 1

Microprocessor এর কাছে
2's C
floating point (Representation, ...) Normalization
floating point Algorithm
(Lec ⇒ 1-4, 12)

---

s:

Q: What are the error conditions?

Q: Write the O/P of the algo fo the addition of two floating points & Justify your answer. → [more marks]

Q: Write the only necessary parts of the algo for adding the given ~~flat~~ floating point number & explain your ans.

Q: Discuss the overflow/ _ / _ happen with example.
এমন দুইটা point বের করা যা overflow/zero etc দিবে।

I/P : _ _
_ _
print ( )

compare :
_ _ _
_ _ _
print ( )

equalize :
_ _
_ _

overflow এর জন্য
Emax লাগবে।

কিছু না দেওয়া
থাকলে ধরে নিব।

zero কে floating point এ represent করা যাচ্ছে না তাই, zero আসলে error flag on হবে। অর্থাৎ E=1 হবে।

#

Ans:

O/p:

AB
CD

4×10^2
2×10^3

where the print function is?

Now Justify
→ calculation দেখাতে হবে॥

Example – 1 :

4×10^{2} E_1
2×10^{3} E_2

$E = E_1 - E_2 = 2 - 3 = -1$

print i/p
" compare
" equalize
" Add
End

$E < 0$ {
}

i/p
Compare
equalize
Add
End

Subject..........................
Date:..........................

i/p :
print("i/p")

compare :
print(" compare")

Equalize :
if(E<0) {
print("E1") }
go to Equalize

(E>0)
{ print (" E_2")
go to equalize }

Answer :

i/p
compare
E1

আগের lec গুলো execution unit এর কথা ছিল।

COA CU Instruction Pipelining

one of the reason why processor works fast.

COA → Computer Organization Architechture

CU → Control Unit (Processor এর একটা unit)

→ Fetch & Decode

Q: Why processor becomes so fast?

→ Increase data bus size (Now 64 bit is used)

→ Increase clock frequency / Trigger

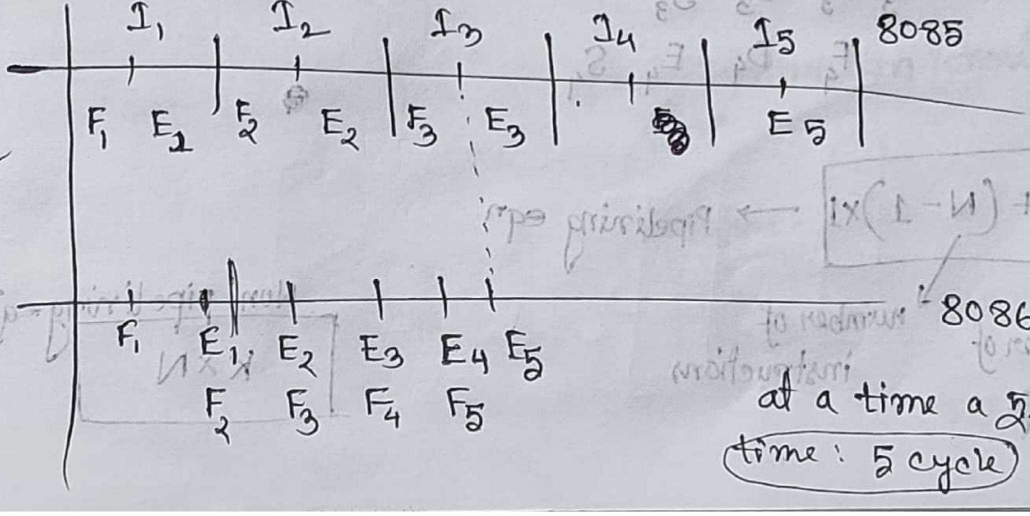
→ ~~Pipeling~~ Pipelining

8085 → Non pipelined

8086 → Pipelined

2 stage Pipelining :

figure



Decode খুবই কম time নেয় so নিচ্ছি না।

time: $5 \times 2 = 10$ cycle

at a time a দুইটা কাজ হচ্ছে।

time: 5 cycle