Practice Problems
Modules 1 and 2

1. Consider a processor that takes the following execution time for different types of instructions in process A:

Instruction Type	Time taken	Number in process A
Integer arithmetic	**1**	**10**
Float	**3**	**4**
Control	**3**	**8**
Procedure call	**4**	**2**

i.How much performance improvement will be achieved if float instructions are made faster by 2 times?
ii.How much improvement in execution time of procedure call is needed to achieve an overall improvement of 20% for process A?
iii.How much improvement in execution time of control instructions is needed to achieve 1.5 times speedup in overall execution time of process A?
Solution:

Set 1

(i)

	CC	count	Time (cc)
int	1	10	10
float	3	4	12
control	3	8	24
proc	4	2	8
			54

$S_{float} = 2$

$f_{float} = 12/54$

$$S_{overall} = \frac{1}{1-0.22+0.22/2}$$

$$\boxed{S = 1.12}$$

Alternate

$ExecTime_{A\,old} = 54$

If float is sped up by 2

float Time $= 1.5$ cc

$\therefore ExecTime_{A\,new} = 10+6+24+8$

$= 48$

Improvement $= 54/48 = \boxed{1.125}$

(ii) $Exec\ A_{old} = 54$

20% imp = 80% of 54

= 43.2 cc

speedup = 54/43.2 = 1.25

$f_{proc} = 8/54 = 0.148$

$S_{proc} = ?$

$$\frac{1}{(1-0.148) + \frac{0.146}{S}} = 1.25$$

Cannot satisfy

Improvement not possible

Set 1

(iii) $f_{control} = 24/54$

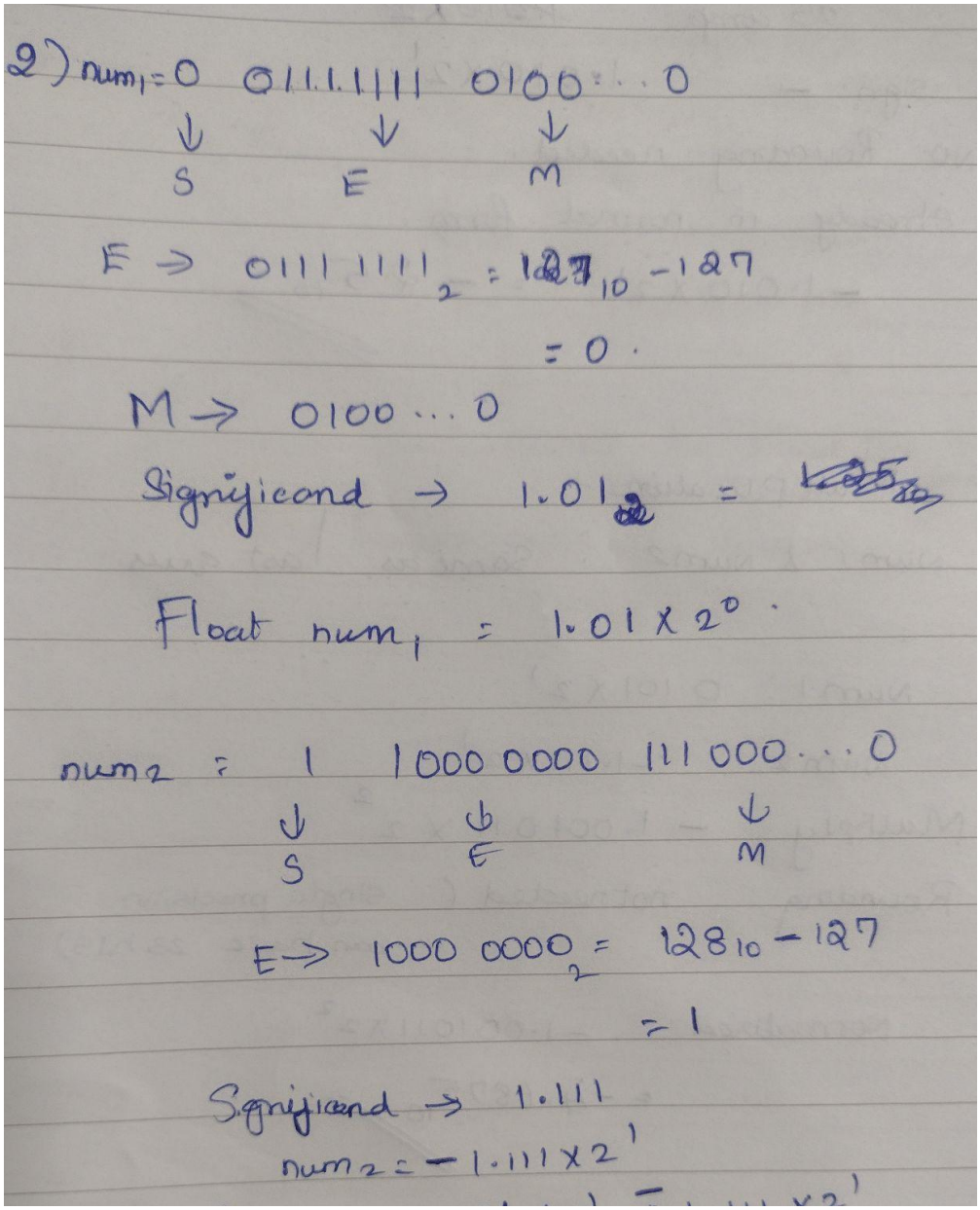
$S_{control} = ?$

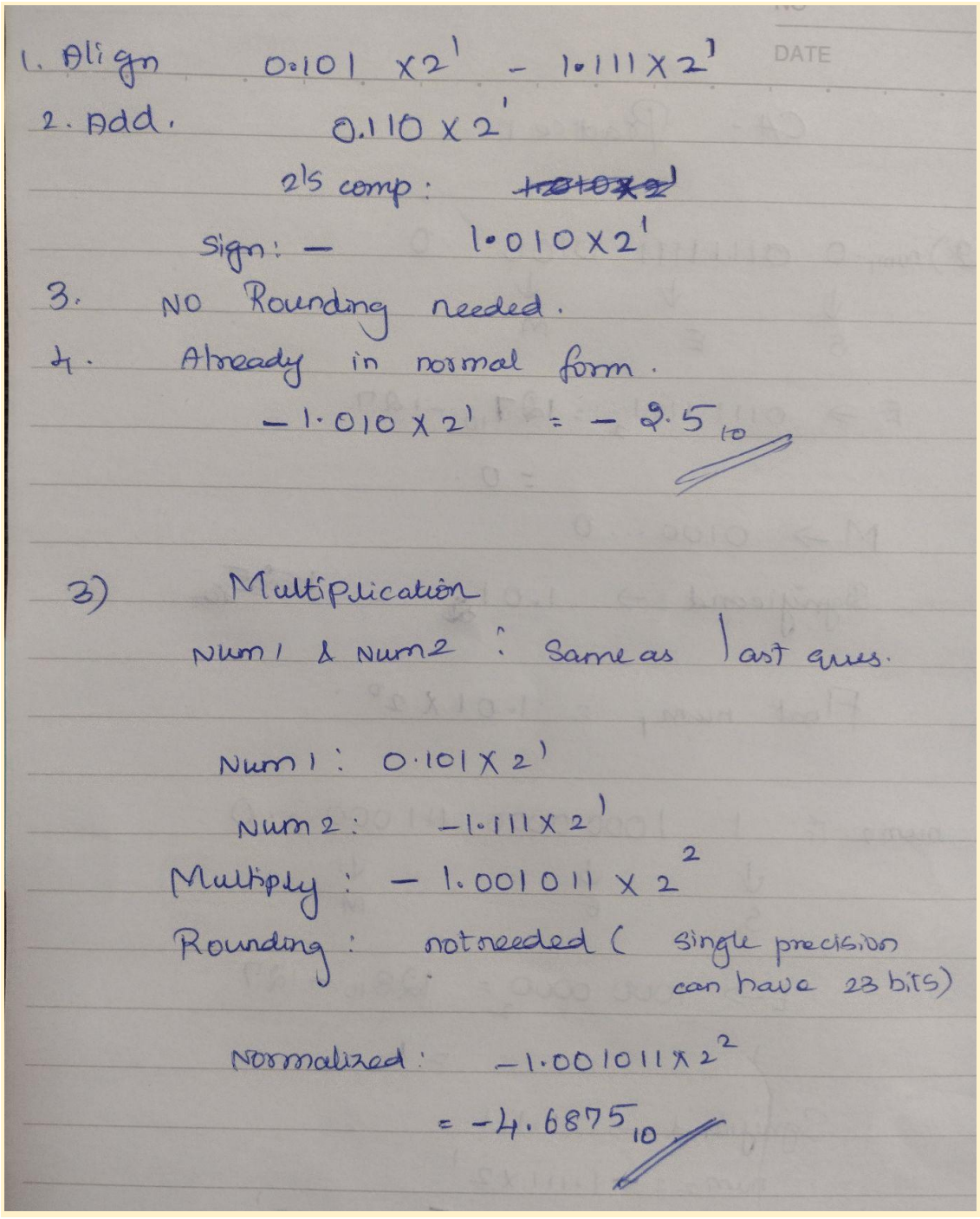
$S_{overall} = 1.5$

$$\frac{1}{(1-0.44) + \frac{0.44}{S}} = 1.5$$

S = 4.4

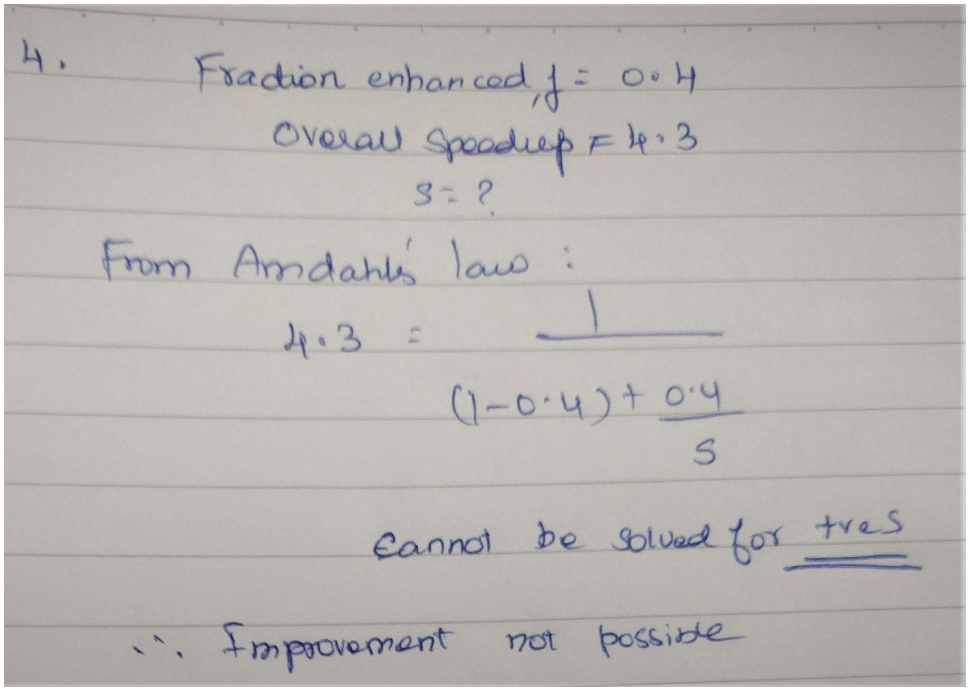
2. Perform IEEE binary floating point addition for the following two values. Convert the result to decimal floating point. Show step-by-step process.
   00111111101000000000000000000000
   11000000011100000000000000000000





3. Perform IEEE binary floating point multiplication for the following two values. Convert the result to a decimal floating point. Show step-by-step process.
   00111111101000000000000000000000
   11000000011100000000000000000000

4. Let a program have 40 percent of its code enhanced to yield a overall speedup 4.3 times faster. What is the fractional speedup enhanced?



5. What do you mean by word size in a CPU? If the word size in a CPU is 32 bits, what can be the maximum address range for the CPU?
6. What is the primary difference between a big-endian and a small-endian machine, in terms of organizing the data?
7. Complete the following tables:
   a.

Hexadecimal	Binary	a<<3	Hexadecimal
E5			
46			

   b.

Hexadecimal	Binary	a>>2	Hexadecimal
B4			
49			

   c.

Hexadecimal	Binary	a>>2 - arithmetic	Hexadecimal

AD			
D4			

8. Complete the following table:

Hexadecimal	Binary	$B2U_4$ (in vector form)	$B2T_4$ (in vector form)
0xA	1010	$2^3 + 2^1 = 10$	$-2^3+2^1 = -6$
0xB			
0x8			
0x3			
0xF			

9. In an 8-bit machine, what is the maximum range (TMax) of representation possible for signed integers? If we know the value of TMax, how can we directly compute the minimum range of signed integers (TMin) and maximum range of unsigned integers (UMax)?
10. Represent the following in single and double precision format:

(a) 1259.125 (b) -114.625

Solutions 6 to 10

Solutions

(5) CPU - Word Size :- The amount of data a CPU's internal data registers can hold and process at a time.

CPU word size is 32 bits, ~~Address range~~

Maximum address range will be $0 - 2^{32-1}$

(6) In a big-endian machine, the least significant byte has the highest address, whereas in a little endian machine the least significant byte has lowest address

For example, 0x45921362

	0x100	0x101	0x102	0x103	0x104	0x105	0x106
Big Endian			45	92	13	62	
Little Endian			62	13	92	45	

7. (a)

Hex	Binary	a << 3	Decimal
E5	11100101	00101000	28
46	01000110	00110000	30

(b)

Hex	Binary	a >> 2	Decimal
B4	10110100	00101101	2D
49	01001001	00010010	12

(c)

Hex	Binary	a >> 2 arithmetic	Decimal
AD	10101101	11101011	EB
D9	11010100	11110101	F5

TNPL
DATE / /

(8)

Hexadecimal	Binary	B2U4($\vec{x}$)	B2T4($\vec{x}$)
0xA	1010	$2^3+2^1=10$	$-2^3+2^1=-6$
0xB	1011	$2^3+2^1+2^0=11$	$-2^3+2^1+2^0=-5$
0x8	1000	$2^3=8$	$-2^3=-8$
0x3	0011	$2^1+2^0=3$	$2^1+2^0=3$
0xF	1111	$2^3+2^2+2^1+2^0=15$	$-2^3+2^2+2^1+2^0=-1$

(9) Refer Slide "Numeric Ranges"

$TMax = 2^{w-1}-1$

Here $w=8$

$TMax = 2^{8-1} = 128-1 = 127$

$|TMin| = TMax+1 = 127+1 = 128$

$UMax = 2*TMax+1 = 2\times127+1 = 255$

(10) (a) There are two parts = 1259 . 125

First we convert these number to binary.

$(1259)_{10} = (10011101011)_2$

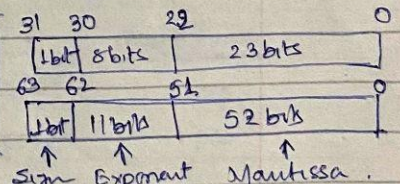
$(0.125)_{10} = (001)_2$

$= 10011101011.001$

Now, we will normalize ~~these~~ this:

We know for

Single Precission	$(1.N)\times2^{E-127}$
Double Precission	$(1.N)\times2^{E-1023}$



Therefore,

$(10011101011.001)_2$

$= (1.0011101011\times2^{10})$ $1.0011101011001\times2^{10}$

For single Precission $(1.N)\times2^{E-127}$

In this case, $E-127=10 \Rightarrow E=137=(10001001)_2$

In single precission,

0	10001001	00111010110010...0

Similar Approach can be

TNPL
DATE / /

(b) -114.625

We will follow the same procedure -

$114.625 = 01110010.101$

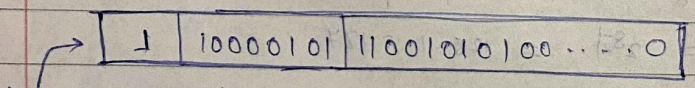
$\Rightarrow 1.1100101 01 \times 2^6$

Exponent - bias $= 6$

Exponent $- 127 = 6$

Exponent $= 6 + 127 = 133 = 10000101$

Single Precission



indicate negative.

Similar Approache can be followed for doub precission