ACA LAB-5

Name: Chaniyara Prince
Roll No: CE003

Sessional 1 Solution:

Name: Chaniyara Prince K.
Roll No: CE003

LAB-5

1. Convert the decimal no. -94.625 to its IEEE-754 double precision representation.

→ $94 = 1011110$

0.625×2 1
0.250×2 0
0.500×2 1

$94.625 = 1011110.101$

$= 1.011110101 \times 2^6$

exponent $= 6 + 1025 = 1029$
$= (10000000101)_2$

$$\underbrace{1}_{1}\ \underbrace{10000000101}_{11}\ \underbrace{01111010100.\,--}_{52}$$

$= \boxed{C057A80000000000H}$

2. Consider the following data items defined as double word (DD) $a = 2.5670$, ~~b=~~ $b = 3.50$, $c = 4.50$, $d = 4.0$, $x = 3.50$ & $y = 2.50$ Analyze the given 8087 program code snippets and report the values of the stack register after each instruction.

Stack

i) fld b — ST(0): 3.5

fmul b — ST(0): 12.25

fld d — ST(0): 4
ST(1): 12.25

fld a — ST(0): 2.5669
ST(1): 4
ST(2): 12.25

fmul — ST(0): 10.267
ST(1): 12.25

fmul c — ST(0): 46.2059
ST(1): 12.25

fsubr — ST(0): 33.9559

fsqrt — ST(0): 5.8271

ii) fld x — ST(0): 3.5

fld y — ST(0): 2.5
ST(1): 3.5

fyl2x — ST(0): 4.6267

fld d ST(0) : 4
ST(1) : 4.6267

fsub ST(0) : 0.6267

f2xm1 ST(0) : 0.5440

fld1 ST(0) : 1
ST(1) : 0.5440

fadd ST(0) : 1.5440

fld1 ST(0) : 1
ST(1) : 1.5440

fxch ST(0) : 1.5440
ST(1) : 1

fscale ST(0) : 3.088
ST(1) : 1

3. ideal speed Up = 5
unconditional brances = 4 %
Conditional branches = 15 % (85 % taken
Branch taken delay = 3 cycles
not taken delay = 2 cycles

a. $\text{Speed Up} = \dfrac{5}{1 + 0.09 \times 3 + 0.15(0.85 \times 3 + 0.15 \times 2)}$

$= \dfrac{5}{1.6975}$

$= \underline{2.9455}$

$\%\ \text{loss of speed Up} = \dfrac{5 - 2.9455}{5} \times 100$

$= \boxed{41.2\%}$

b. taken delay = 1
not taken delay = 0

$\text{Speed Up} = \dfrac{5}{1 + 0.09 \times 1 + 0.15(0.85 \times 1)}$

$= \dfrac{5}{1.2175}$

$= \underline{4.10}$

$\%\ \text{loss of Speed Up} = \dfrac{5 - 4.10}{5} \times 100$

$= \boxed{18\%}$

Gain over previous case = 41.2 − 18

$= \boxed{23.2\%}$

c. BTB prediction accuracy = 91%
0.96 probability of branch entry is present in BTB.

→ delay for Uncon. branch found in BTB
= 0
not found in BTB = 0.04 × 3
= 0.12

→ Con. Branch found in BTB = 0
delay for cond. branch = 0.85 × 3 + 0.15 × 2
= 2.85
not found in BTB = 0.04 × 2.85
= 0.114

misprediction = 0.96 × 0.09 × 2.85
= 0.24624

$$\text{Speed Up} = \frac{5}{1 + 0.09 \times 0.12 + 0.15(0.114 + 0.24624)}$$

$$= \frac{5}{1.06483} = \boxed{4.6955}$$

% loss of SpeedUp = $\frac{5 - 4.6955}{5} \times 100$

$= \boxed{6.09\%}$

4. Time Space diagram for the assignment of instructions to diff. processing elements give total clock cycles taken to complete the instruction stream over 5 elements of vector A using Array processor & Vector processor.

LD VR ← A[4:0]
ADD VR ← VR, 1
MUL VR ← VR, 2
SUB VR ← VR, 3
DIV VR ← VR, 4
ST A[4:0] ← VR

→ Array Processor :-

Clock Cycle	PE	PE	PE	PE	PE
1	LD0	LD1	LD2	LD3	LD4
2	ADD0	ADD1	ADD2	ADD3	ADD4
3	MUL0	MUL1	MUL2	MUL3	MUL4
4	SUB0	SUB1	SUB2	SUB3	SUB4
5	DIV0	DIV1	DIV2	DIV3	DIV4
6	ST0	ST1	ST2	ST3	ST4

Total Clock cycles = 6

→ Vector Processor

0	LD0					
1	LD1	ADD0				
2	LD2	ADD1	MUL0			
3	LD3	ADD2	MUL1	SUB0		
4	LD4	ADD3	MUL2	SUB1	DIV0	
5		ADD4	MUL3	SUB2	DIV1	ST0
6			MUL4	SUB3	DIV2	ST1
7				SUB4	DIV3	ST2
8					DIV4	ST3
9						ST4

Total Clock - Cycles = 10

openMP Tutorial & Exercises:

a. Exercise 1A : Hello World Program

```
#include<stdio.h>
#include<omp.h>
int main()
{

   #pragma omp parallel
   {
     int ID = 0;
   printf(" hello(%d) ", ID);
   printf("world(%d) \n", ID);
   }
}
```

Output:

```
hello(0) world(0)
```

b. Exercise 1B : Write a multithreaded program that prints “Hello World”

```
#include<stdio.h>
#include <omp.h>
int main()
{
        #pragma omp parallel
        {
        int ID = omp_get_thread_num();
        printf("hello(%d) ", ID);
        printf("world(%d) \n", ID);
        }
}
```

Output:

```
hello(0) world(0)
hello(3) world(3)
hello(1) world(1)
hello(2) world(2)
```