INSTITUTE OF INFORMATION TECHNOLOGY

JAHANGIRNAGAR UNIVERSITY

Assignment : 01

Submission Date : 10/10/2021

Course Tittle : Computer Architecture

Course Code : ICT - 2207

Submitted To

Md Biplob Hosen

Lecture

IIT – JU

Submitted By

Md. Shakil Hossain

Roll – 2023

2^{nd} year 2^{st} Semester

IIT – JU

Roll — 2023

Exercise : 2.10

$$CPI = \frac{\sum CPI \times I_i}{I_c}$$

$$= \frac{45000 \times 1 + 32000 \times 2 + 15000 \times 2 + 8000 \times 2}{100000}$$

$$= 1.55$$

Ans

$$\text{MIPS rate} = \frac{I_c}{T \times 10^6}$$

$$= \frac{f}{CPI * 10^6}$$

$$= \frac{40 \times 10^6}{1.55 \times 10^6}$$

$$= 25.806 \text{ Ans}$$

$$\text{Execution time} = I_c \times CPI \times I_i$$

$$= 100000 \times 1.55 \times \frac{1}{40 \times 10^6}$$

$$= 0.00387 \text{ s}$$

$$= 3.87 \text{ ns.}$$

Roll – 2023

Exercise 2.11

a. $$CPI_A = \frac{\Sigma CPI \times I_i}{I_c}$$

$$= \frac{(8\times1 + 4\times3 + 2\times4 + 4\times3)\times10^6}{(8+4+2+4)\times10^6}$$

$$= 2.22$$

$$MIPS_A = \frac{f}{CPI_A \times 10^6}$$

$$= \frac{200\times10^6}{2.22\times10^6}$$

$$= 90$$

$$CPU_A = \frac{I_c \times CPI_A}{f}$$

$$= \frac{18\times10^6 \times 2.2}{200\times10^6}$$

$$= 0.2\ s$$

$$CPI_B = \frac{\Sigma CPI_i \times I_i}{I_c}$$

$$= \frac{(10\times1 + 8\times2 + 2\times4 + 4\times3)\times10^6}{(10+8+2+4)\times10^6}$$

$$= 1.92$$

$$MIPS_B = \frac{f}{CPI_B \times 10^6}$$

$$= \frac{200 \times 10^6}{1.92 \times 10^6}$$

$$= 104$$

$$cpu_B = \frac{I_c \times CPI_B}{f}$$

$$= \frac{24 \times 10^6 \times 1.92}{200 \times 10^6}$$

$$= 0.23 \text{ s}$$

6. Machine B has a higher MIPS than machine A. it requires a longer cpu time to execute the same set of benchmark Programs.

Roll-2023

Exercise: 2.12

a. We can express the MIPs rate as:

$$(\text{MIPS rate})/10 = \frac{I_c}{T}$$

$$\therefore I_c = T \times (\text{MIPS rate}/10^6)$$

The ratio of the instruction count of the RS/60000 to the VAX is

$$[x \times 18]/[12x \times 1] = 1.5$$

b. For the Vax

$$CPI = 5\ MHz/1 MIPS$$
$$= 5$$

For the Rs/6000

$$CPI = 25/18$$
$$= 1.39$$ Ans.

Roll—2023

Exercise: 2.13

$$MIPS = I_c / T \times 10^6$$

$$= \frac{100}{T}$$

	Computer A	Computer B	Computer C
Program 1	100	10	5
Program 2	0.1	1	5
Program 3	0.2	0.1	2
Program 4	1	0.125	1

	Arithmetic mean	Rank	Harmonic mean	Rank
Computer A	25.325	1	0.25	2
Computer B	2.8	3	0.21	3
Computer C	3.26	2	2.1	1

Roll—2023

Exercise : 2.14

a Normalized to R

Benchmark	Processor		
	R	M	Z
E	1.00	1.71	3.11
F	1.00	1.49	1.19
H	1.00	0.43	0.49
I	1.00	1.11	0.60
K	1.00	2.10	2.09
Arithmetic mean	1.00	1.31	1.50

b Normalized to M

Benchmark	Processor		
	R	M	Z
E	0.59	1.00	1.82
F	0.84	1.00	1.00
H	2.32	1.00	1.13
I	0.90	1.00	0.54
K	0.48	1.00	1.00
Arithmetic mean	1.01	1.00	1.10

c. Recall that the larger the ratio the higher the speed. Based on (a) R is the slowest machine by a significant amount. Based on (b) M is the slowest machine by a modest amount.

d. Normalized to R.

Benchmark	Processor		
	R	M	Z
E	1.00	1.71	3.11
F	1.00	1.19	1.19
H	1.00	0.43	0.49
I	1.00	1.11	0.60
K	1.00	2.10	2.09
Geometric mean	1.00	1.15	1.18

Normalized to M

Benchmark	Processor		
	R	M	Z
E	0.59	1.00	1.82
F	0.84	1.00	1.00
H	2.32	1.00	1.13
I	0.90	1.00	0.54
K	0.48	1.00	1.00
Gremetic mean	0.87	1.00	1.02

Using the geometric mean R is the slowest no matter which machine is used for normalization.

Roll - 2023

Exercise : 2.15

a. Normalized to X.

Benchmark	Processor		
	X	Y	Z
1	1	2.0	0.5
2	1	0.5	2.0
Arithmetic mean	1	1.25	1.25
Geometric mean	1	1	1

Normalized to Y

Benchmark	Processor		
	X	Y	Z
1	0.5	1	0.25
2	2.0	1	4.0
Arithmetic mean	1.25	1	2.125
Geometric mean	1	1	1

Machine Y is twice as fast as machine X for benchmark 1 but half as fast for benchmark 2. similarly machine Z is half as X for benchmark 1 but twice as fast for benchmark 2. Intuitively these three machines have equivalent Performance.

However if we normalize to x and compute the arithmetic mean of the speed metric we find that Y and Z are 25% faster than X. Now if we normalize to Y and compute the arithmetic mean of the speed metric we find that X is 25% faster than Y and Z is more than twice as fast as Y. Clearly the arithmetic mean is worthless is this content.

b. When the geometric mean is used the three machines are shown to have equal performance when normalized to X and also equal performance when normalized to Y. These results are much more in line with our intuition.

Exercise: 2.16

a. Assuming the same instruction mix means that the additional instructions for each task shold be allocated proportionally among the instruction type. So we have the following table.

Instruction Type	CPI	Instruction Mix
Arithmetic and logic	1	60%
Load/Store with cache hit	2	18%
Branch	4	12%
Memory reference with Cache miss	12	10%

$$CPI = 0.6 + (2 \times 0.18) + (4 \times 0.12) + (12 \times 0.1)$$
$$= 2.64$$

The CPI has increased due to the increased time for memory access.

b. $$MIPS = \frac{400}{2.64}$$
$$= 152$$

There is a corresponding drop in the MIPS rate.

c. The speedup factor is the ratio of the execution times. We calculate the execution time as $T = I_c / (MIPS \times 10^6)$ For single-Prosson case $T_1 = (2 \times 10^6) / (178 \times 10^6)$

$= 11$ ms. With 8 Processor each Processor executes 1/8 of the 2 million instruction plus the 25,000 overhead instructions.

$$T_8 = \frac{\frac{2 \times 10^6}{8} + 0.025 \times 10^6}{152 \times 10^6}$$

$$= 1.8 \text{ ms}$$

Therefore we have

$$\text{Speedup} = \frac{\text{time to execute Program on a single Processor}}{\text{time to execute Program on N Parallel Processor}}$$

$$= \frac{11}{1.8}$$

$$= 6.11 \quad \text{Ans.}$$

Exercise: 2.17

a. $$\text{Speedup} = \frac{\text{time to acess in main memory}}{\text{time to access in cache}}$$

$$= \frac{T_2}{T_1}$$

b. The average access time can be computed as $T = H \times T_1 + (1-H) \times T_2$

$$\text{Speedup} = \frac{\text{Execution time before enhancement}}{\text{Execution time after enhancement}}$$

$$= \frac{T_2}{T_1} = \frac{T_2}{H \times T_1 + (1-H) \times T_2} = \frac{1}{(1-H) + H\frac{T_1}{T_2}}$$

c. $T = H \times T_1 + (1-H) \times (T_1 + T_2)$

$= T_1 + (1-H) \times T_2$

$$\text{Speedup} = \frac{\text{Execution time before enchancement}}{\text{Execution time after enhancement}}$$

$$= \frac{T_2}{T_1} = \frac{T_2}{T_1 + (1-H)T_2} = \frac{1}{(1-H) + \frac{T_1}{T_2}}$$

In this case the denominator is langer. So that the speedup is less.

THE END