Assignment -1

Gouthami Channakeshavamurthy
A20445435

1.2

a. Assembly lines in automobile manufacturing

- *The principle of an* ***assembly line*** *is that each worker is assigned one very specific task, which he or she simply repeats, and then the process moves to the next worker who does his or her task, until the task is completed and the product is made. It is a way to mass produce goods quickly and efficiently.*
- *This approach is used in automobile manufacturing industry, where in the industrial system/ robot completes its assigned activity and moves on to the next system/robot for consecutive tasks. Hence, Assembly lines in automobile manufacturing matches with term "****Performance via Pipeline****".*

b. Suspension bridge cables

- *On a suspension bridge, smaller cables called suspenders run vertically from the bridge deck up to the main supporting cables. The suspenders transfer the bridge deck's compression forces to the towers via the main supporting cables, which create graceful arcs between the towers and down to the anchorages on each end.*
- *Since each cable hold a part of the total bridge weight, it is similar to multiple processes performing work at the same time; the given statement resembles "****Performance via Parallelism****".*

c. Aircraft and marine navigation systems that incorporate wind information

- *During flight, one of the main considerations that will affect an aircraft is the motion of the wind. Referred to as wind effect, the speed and direction of the wind will alter the progress of any aircraft in flight. Although an aircraft has its own means of propulsion, the pilot must compensate for the wind speed and direction, in order for an aircraft to maintain the desired course.*
- *The aircraft and marine navigation systems incorporate wind information, it uses wind speed prediction for better route generation. Hence, the statement resembles the term "****Performance via Prediction****"*

d. Express elevators in buildings

- *An express elevator does not serve all floors. For example, it moves between the ground floor and a sky-lobby or a terrace, or it moves from the ground floor or a sky-lobby to a range of floors, skipping floors in between. Since it is not stopping at each floor and making the common stops faster, the statement resembles to the term "****Make the Common Case Fast****".*

e. Library reserve desk

- *Hierarchy of memories is implemented on the basis of a very common idea i.e., the library reserve desk. A library reserve desk contains the things that are requested most frequently by the students. The course instructors or college staff individuals put these books or papers. Similarly the cache memory, which is part of hierarchy of memories incorporates the same idea and resembles the term "****Hierarchy of Memories****"*

f. Increasing the gate area on a CMOS transistor to decrease its switching time

- *The idea to increase the gate area on a CMOS transistor is to decrease its switching time. Hence the increase in the gate area is redundant to increase in switching time. So, the given statement resembles to the term "**Dependability via Redundancy**"*

g. Adding electromagnetic aircraft catapults (which are electrically-powered as opposed to current steam-powered models), allowed by the increased power generation offered by the new reactor technology

- *The idea of adding electromagnetic aircraft catapults was seen possible because of the recent technological advancements, that increased power generation offered by the new reactor technology.*
- *Moore's law predicts the increase in the performance in technology and performance over time. These technological advancements resembles the "**Design for Moore's Law**".*

h. Building self-driving cars whose control systems partially rely on existing sensor systems already installed into the base vehicle, such as lane departure systems and smart cruise control systems

- *Abstraction means to represent a design with intricate/complex details in a high level design i.e., to hide the internal working design and use larger representational building blocks. Hence the given statement is an example of abstraction and it resembles **"Use Abstraction to Simplify Design".***

———————————————————————————————————————

1.6

- *Consider the four classes of instruction as A, B, C and D. The clock rate and CPI for each Implementation is given below:*

	Clock Rate	CPI of Class A	CPI of Class B	CPI of Class C	CPI of Class D
P1	2.5×10^9 Hz	1	2	3	3
P2	3×10^9 Hz	2	2	2	2
Instruction Count		10% = 0.1	20% = 0.2	50% = 0.5	20% = 0.2

$$\text{CPU time} = \frac{\sum (IxCPI)}{ClockRate}$$

Given, dynamic instruction count I = 1 x 10^6

- -CPU time for processor P1:

$$CPU\ time = \frac{\sum (1x10^6[(0.1x1) + (0.2x2) + (0.5x3) + (0.2x3)]}{2.5x10^9}$$

CPU time = 1.04 x 10^{-3} = 1.04 milliseconds. ————————————————(1)

- -CPU time for processor P2:

CPU time = $$\frac{\sum(1x10^6[(0.1x2)+(0.2x2)+(0.5x2)+(0.2x2)]}{3x10^9}$$

CPU time = 6.6667 x 10^{-4} = 0.67 milliseconds. ————————————(2)

The performance is inversely proportional to CPU time, the processor taking the least time performs better. From equations (1) and (2), we can say that processor P2 performs faster than P1.

a. What is the global CPI for each implementation?

- *Global CPI For Processor P1,*

Given: Clock rate = 2.5 x 10^9 ; number of instruction = 1 x 10^6

From equation (1),

The CPI time for Processor P1 = 1.04 x 10^{-3} = 1.04 milliseconds.

Global CPI = $$\frac{CPIxClockRate}{numberofinstructions}$$

Global CPI = $$\frac{(1.04x10^3)x(2.5x10^9)}{1x10^6} = 2.6$$ ———————— (3)

Therefore, the global CPI for Processor P1 = 2.6.

- *Global CPI for Processor P2,*

Given: Clock rate = 3 x 10^9 ; number of instruction = 1 x 10^6

From equation (2),

The CPI time for Processor P1 = 0.67 x 10^{-3} = 0.67 milliseconds.

Global CPI = $$\frac{CPIxClockRate}{numberofinstructions}$$

Global CPI = $$\frac{(0.67x10^3)x(3x10^9)}{1x10^6} = 2.01$$ ———————— (4)

Therefore, the global CPI for Processor P1 = 2.01.

b. Find the clock cycles required in both cases.

Number of Clock Cycles = Global CPI x number of instructions

For Processor P1: from equation (3),

Number of Clock Cycles = (2.6) x (1 x 10^6) = ***2.6x10⁶***

For Processor P2: from equation (4),

Number of Clock Cycles = (2.01) x (1 x 10^6) = ***2.01x10^6***

——

1.9

1.9.1

- *Given: Clock Rate = 2 GHz = 2 x 10^9 Hz and,*

	arithmetic	*load/store*	*branch*
CPI (Clock cycles per instruction)	1	12	5
Number of instructions per processor	2.56 x 10^9	1.28 x 10^9	256 x 10^6

We know,

$$\text{Execution time for Processor} = \frac{CPUclockCycle}{ClockRate}$$

Where Clock cycle can be derived from,

CPU Clock Cycles = ($CPI_{arithmetic}$ x No. Arithmetic instructions) + ($CPI_{load/store}$ x No. load/store instructions) + (CPI_{branch} x No. branch instructions)

- *Execution time for 1 processor,*

CPU Clock Cycles = (1 x 2.56 x 10^9) + (12 x 1.28 x 10^9) + (5 x 256 x 10^6)

CPU Clock Cycles = 1.92 x 10^{10} cycles

Compute the CPU execution time using the formula,

$$\text{Execution time for Processor} = \frac{CPUclockCycle}{ClockRate}$$

Therefore,

$$\text{CPU execution time } = \frac{19.2x10^9 cycles}{2x10^9 cycles/second} = \textbf{9.6 seconds}. \text{ ——— (for 1 processor)}$$

- *Execution time for 2 processor,*

When the number of processor is more than 1, then the number of instruction is divided by ***0.7 x p ,***

(Where p is he number of processors) for arithmetic and load/store types of instructions.

$$\text{CPU Clock Cycles } = (\frac{1x2.56x10^9}{0.7x2}) + (\frac{12x1.28x10^9}{0.7x2}) + (5x256x10^6)$$

CPU Clock Cycles = 1.408 x 10^{10} cycles

Compute the CPU execution time using the formula,

$$\textit{Execution time for Processor} = \frac{CPUclockCycle}{ClockRate}$$

Therefore,

$$\textit{CPU execution time} = \frac{14.08x10^9 cycles}{2x10^9 cycles/second} = \textbf{7.04 seconds}. \text{———} \textit{(for 2 processors)}$$

- <u>*Execution time for 4 processor,*</u>

When the number of processor is more than 1, then the number of instruction is divided by ***0.7 x p ,***

(Where p is he number of processors) for arithmetic and load/store types of instructions.

$$\textit{CPU Clock Cycles} = (\frac{1x2.56x10^9}{0.7x4}) + (\frac{12x1.28x10^9}{0.7x4}) + (5x256x10^6)$$

CPU Clock Cycles = 7.680 x 10^9 cycles

Compute the CPU execution time using the formula,

$$\textit{Execution time for Processor} = \frac{CPUclockCycle}{ClockRate}$$

Therefore,

$$\textit{CPU execution time} = \frac{7.680x10^9 cycles}{2x10^9 cycles/second} = \textbf{3.84 seconds} \quad \text{————-} \textit{(for 4 processor)}$$

- <u>*Execution time for 8 processor,*</u>

When the number of processor is more than 1, then the number of instruction is divided by ***0.7 x p ,***

(Where p is he number of processors) for arithmetic and load/store types of instructions.

$$\textit{CPU Clock Cycles} = (\frac{1x2.56x10^9}{0.7x8}) + (\frac{12x1.28x10^9}{0.7x8}) + (5x256x10^6)$$

CPU Clock Cycles = 4.480 x 10^9 cycles

Compute the CPU execution time using the formula,

$$\textit{Execution time for Processor} = \frac{CPUclockCycle}{ClockRate}$$

Therefore,

$$\text{CPU execution time } = \frac{4.480x10^9 cycles}{2x10^9 cycles/second} = \textbf{2.24 seconds}. \text{ --- (for 8 processors)}$$

- *Relative speedup of 2 processors over 1 processor,*

$$\frac{SpeedUp_2}{SpeedUp_1} = \frac{ExecutionTime_1}{ExecutionTime_2} = \frac{9.6}{7.04} = 1.36$$

Therefore, 2 processors are ***1.36*** *times faster than 1 processor.*

- *Relative speedup of 4 processors over 1 processor,*

$$\frac{SpeedUp_4}{SpeedUp_1} = \frac{ExecutionTime_1}{ExecutionTime_4} = \frac{9.6}{3.84} = 2.5$$

Therefore, 4 processors are ***2.5*** *times faster than 1 processor.*

- *Relative speedup of 8 processors over 1 processor,*

$$\frac{SpeedUp_8}{SpeedUp_1} = \frac{ExecutionTime_1}{ExecutionTime_8} = \frac{9.6}{2.24} = 4.48$$

Therefore, 8 processors are ***4.48*** *times faster than 1 processor.*

"From the above results we can say, As the number of processors increases, the total execution time gradually decreases. Thus, it can be said that more processors are more efficient".

1.9.2

- *Given: Clock Rate = 2 GHz = 2 x 10^9 Hz and,*

	arithmetic	*load/store*	*branch*
CPI (Clock cycles per instruction)	*2 (doubled)*	*12*	*5*
Number of instructions per processor	*2.56 x 10^9*	*1.28 x 10^9*	*256 x 10^6*

We know,

$$\text{Execution time for Processor} = \frac{CPUclockCycle}{ClockRate}$$

Where Clock cycle can be derived from,

CPU Clock Cycles = ($CPI_{arithmetic}$ x No. Arithmetic instructions) + ($CPI_{load/store}$ x No. load/store instructions) + (CPI_{branch} x No. branch instructions)

- *Execution time for 1 processor,*

CPU Clock Cycles = $(2x\ 2.56\ x\ 10^{9}) + (12\ x\ 1.28\ x\ 10^{9}) + (5\ x\ 256\ x\ 10^{6})$

CPU Clock Cycles = $1.6645\ x\ 10^{10}$ *cycles*

Compute the CPU execution time using the formula,

$$Execution\ time\ for\ Processor = \frac{CPUclockCycle}{ClockRate}$$

Therefore,

$$CPU\ execution\ time\ = \frac{16.64x10^{9}cycles}{2x10^{9}cycles/second} = \mathbf{8.32\ seconds}. \ ---\ (for\ 1\ processor)$$

- *Execution time for 2 processor,*

When the number of processor is more than 1, then the number of instruction is divided by ***0.7 x p ,***

(Where p is he number of processors) for arithmetic and load/store types of instructions.

$$CPU\ Clock\ Cycles = \ (\frac{2x2.56x10^{9}}{0.7x2}) + (\frac{12x1.28x10^{9}}{0.7x2}) + (5x256x10^{6})$$

CPU Clock Cycles = $1.591\ x\ 10^{10}$ *cycles*

Compute the CPU execution time using the formula,

$$Execution\ time\ for\ Processor = \frac{CPUclockCycle}{ClockRate}$$

Therefore,

$$CPU\ execution\ time\ = \frac{15.908x10^{9}cycles}{2x10^{9}cycles/second} = \mathbf{7.954\ seconds}. \ ---\ (for\ 2\ processors)$$

- *Execution time for 4 processor,*

When the number of processor is more than 1, then the number of instruction is divided by ***0.7 x p ,***

(Where p is he number of processors) for arithmetic and load/store types of instructions.

$$CPU\ Clock\ Cycles = \ (\frac{2x2.56x10^{9}}{0.7x4}) + (\frac{12x1.28x10^{9}}{0.7x4}) + (5x256x10^{6})$$

CPU Clock Cycles = $8.594x\ 10^{9}$ *cycles*

Compute the CPU execution time using the formula,

$$Execution\ time\ for\ Processor = \frac{CPUclockCycle}{ClockRate}$$

Therefore,

$$\text{CPU execution time } = \frac{8.594x10^9 cycles}{2x10^9 cycles/second} = \textbf{4.29 seconds} \quad \text{------(for 4 processor)}$$

- *Execution time for 8 processor,*

When the number of processor is more than 1, then the number of instruction is divided by ***0.7 x p ,***

(Where p is he number of processors) for arithmetic and load/store types of instructions.

$$\text{CPU Clock Cycles} = \left(\frac{2x2.56x10^9}{0.7x8}\right) + \left(\frac{12x1.28x10^9}{0.7x8}\right) + (5x256x10^6)$$

CPU Clock Cycles = 4.937 x 10^9 *cycles*

Compute the CPU execution time using the formula,

$$\text{Execution time for Processor} = \frac{CPUclockCycle}{ClockRate}$$

Therefore,

$$\text{CPU execution time } = \frac{4.480x10^9 cycles}{2x10^9 cycles/second} = \textbf{2.468 seconds}. \text{ --- (for 8 processors)}$$

- *Relative Slowdown of 2 processors over 1 processor,*

$$\frac{Slowdown_2}{Slowdown_1} = \frac{ExecutionTime_1}{ExecutionTime_2} = \frac{8.32}{7.954} = \textbf{1.046}$$

Therefore, 2 processors are ***1.36*** *times faster than 1 processor.*

- *Relative speedup of 4 processors over 1 processor,*

$$\frac{SlowDown_4}{SlowDown_1} = \frac{ExecutionTime_1}{ExecutionTime_4} = \frac{8.32}{4.29} = \textbf{1.939}$$

Therefore, 4 processors are ***2.5*** *times faster than 1 processor.*

- *Relative speedup of 8 processors over 1 processor,*

$$\frac{SlowDown_8}{SlowDown_1} = \frac{ExecutionTime_1}{ExecutionTime_8} = \frac{8.32}{2.468} = \textbf{3.37}$$

1.9.3

$$(2x\ 2.56\ x\ 10^9) + (CPI_{load/store-new}\ x\ 1.28\ x\ 10^9) + (5\ x\ 256\ x\ 10^6) =$$
$$\left(\frac{1x2.56x10^9}{0.7x4}\right) + \left(\frac{12x1.28x10^9}{0.7x4}\right) + (5x256x10^6)$$

$$CPI_{load/store-new} = \frac{3.84x10^9}{1.28x10^9} = \mathbf{3}$$

*Therefore, the CPI of load/store instructions has to be reduced by **3** in order for a single processor to match the performance of four processors using the same CPI values*

———————————————————————————————————————

1.13

1.13.1

Given, the computer running instruction with CPU time is as follows

FP instructions	*L/S instructions*	*branch instructions*	*Time t_{total}*
70	*85*	*40*	*250 s*

Total reduced time of FP operations = 20%

When the CPU time is reduced by 20% he number of Floating point instruction can be calculated as follows,

<u>Calculate the FP instructions CPU time:</u>

The reduced time for the FP instructions is the difference of total time taken for reduced floating point instructions from the total time for Floating point instructions.

$$t_{FP-red} = t_{FP} - (t_{FP}\ x\ 0.2)$$

$$t_{FP-red} = 70 - (70\ x\ 0.2) = \mathbf{56\ seconds.}$$

Total time after reducing Floating point instructions is,

$$t_{tot-FP-red} = t_{FP} + t_{INT} + t_{L/S} + t_{branch}$$

$$t_{tot-FP-red} = 56 + 85 + 55 + 40 = \mathbf{236\ seconds.\ i.e.,}$$

<u>Calculate the total reduced time:</u>

The total reduced time after reducing FP instructions is the difference of the total time taken after reduction from the total time taken before reduction.

$$t_{tot-red} = t_{tot} - t_{tot-FP-red}$$

$$t_{tot-red} = 250 - 236 = \mathbf{14\ seconds.}$$

The percentage of total time reduced when the total reduced for FP instruction is calculated as follows,

$$t_{per} = (\ t_{tot-red}\ /\ t_{tot}\ )\ x\ 100$$

$$t_{per} = (\ 14 \ / 250) \times 100 = 5.6\%$$

Hence the total time reduced for 20% of FP instructions is ***5.6%***

1.13.2

$$200 = X \times \text{Execution Time}_{INT} + \text{Execution Time}_{Rest}$$
$$200 = X \times (250 - 70 - 85 - 40) + (70 + 85 + 40)$$
$$9\% \approx X$$
$$91\% \text{ reduced}$$

Therefore,

INT operation time has to be reduced by 91%, if the total time is reduced by 20%

1.13.3

$$200 = X \times \text{Execution Time}_{Branch} + \text{Execution Time}_{Rest}$$
$$200 = X \times 40 + (250 - 40)$$
$$-10 \neq X \times 40$$

"No, the total time cannot be reduced by 20% by only reducing the time for branch instructions".

—————————————————————————————————————

1.14

1.14.1

- *Given,*

The instructions (FP, INT, L/S and branch) for a computer program with CPI and clock rate are as follows:

Instructions	***Execution***	***CPI***	***Clock Rate***
FP	50×10^6	*1*	2×10^9
INT	110×10^6	*1*	2×10^9
L/S	80×10^6	*4*	2×10^9
Branch	16×10^6	*2*	2×10^9

To calculate the execution time,

$$\text{CPU time} = \sum \left(\frac{ClockCycle)}{ClockRate}\right)$$

$$\text{Clock cycles} = (CPI_{FP} \times \text{no. of FP instructions}) + (CPI_{INT} \times \text{no. of INT instructions}) + (CPI_{L/S} \times \text{no. of L/S instructions}) + (CPI_{Branch} \times \text{no. of Branch instructions})$$

$$\text{Clock Cycles} = (50 \times 10^6 \times 1) + (110 \times 10^6 \times 1) + (80 \times 10^6 \times 4) + (16 \times 10^6 \times 2) = \mathbf{512 \times 10^6 \ seconds}$$

CPU time = $\sum(\frac{ClockCycle)}{ClockRate}) = \frac{512x10^6)}{2x10^9}$ ***= 256 10⁻³ seconds***

For improved no. of FP's:

Reduce the no. of clock cycles to half in order to improve the CPI of FP instructions,

$\frac{Clockcycles}{2}$ *= (CPI_{FP-improved} x no. of FP instructions) + (CPI_{INT} x no. of INT instructions) + (CPI_{L/S} x no. of L/S instructions) + (CPI_{Branch} x no. of Branch instructions)*

$$CPI_{FP\text{-}improved} = \frac{\frac{Clockcycles}{2} - [CPI_I x I_I) + (CPI_L x I_L) + (CPI_B x I_B)]}{no.ofFPinstructions}$$

$$CPI_{FP\text{-}improved} = \frac{\frac{512x10^6}{2} - [110x10^6x1) + (80x10^6x1) + (16x10^6x2)]}{50x10^6} = \mathbf{-\ 4.12 < 0}$$

Thus the CPI of FP instructions cannot be improved because the value might be negative, when the programs run 2 times faster.

1.14.2

$$Execution\ Time = CPU\ time = \frac{ClockCycle}{ClockRate}$$

$$128\ ms = \frac{(50 \times 10^6 \times 1) + (110 \times 10^6 \times 1) + (80 \times 10^6 \times X) + (16 \times 10^6 \times 2)}{2x10^9}$$

$$X = 0.8$$

Therefore,

*improve the CPI of L/S instructions by **0.8**, if the program has to run two times faster.*

1.14.3

$$Execution\ Time = CPU\ time = \frac{ClockCycle}{ClockRate}$$

$$Execution\ Time = \frac{(50 \times 10^6 \times 0.6) + (110 \times 10^6 \times 0.6) + (80 \times 10^6 \times 2.8) + (16 \times 10^6 \times 1.4)}{2x10^9} = 171.2ms$$

Therefore,

the execution time of the program improved by 171.2 milliseconds, if the CPI of INT and FP instructions is reduced by 40% and the CPI of L/S and Branch is reduced by 30%

————————X————————————————X————————————————X————————