IQRA NATIONAL UNIVERSITY

INU

DEFINING POSSIBILITIES, CREATING OPPORTUNITIES

Name : Muhammad Musa

Department : BS(CS)

Semester : 4th

ID # : 15366

Sessional Assignment No : 2nd

Subject : Computer Architecture

Submitted To : Muhammad Amin Sir

Dated : 17th May 2020

M-Amin Sir. BS (CS)

Name : Muhammad Musa
ID # : 15366
Subject : Computer Architecture
Deptt : BS (CS)
Assignment No : 02

Q1:-

Ans:(A) Different desktop applications that require the great power of contemporary microprocessor based systems are:

- Image processing
- Three-dimensional rendering
- Speech recognition
- Videoconferencing
- Multimedia authoring
- Voice and video annotation of files
- Stimulation modelling.

Ans:(B) The techniques used in contemporary processors to increase speed are;

- **Pipelining:-** Pipelining enables a processor to work simultaneously on multiple instructions by performing a different phase for each of the multiple instructions at the same time.
- **Branch Prediction:-** Branch prediction potentially increases the amount of work available for the processor to execute.

- **Superscalar Execution:-** This is the ability to issue more than one instruction in every processor clock cycle. In affect, multiple parallel pipelines are used.
- **Data flow analysis:-** The processor analyzes which instructions are dependent on each other's results, or data, to create an optimized schedule of instructions.
- **Speculative Execution:-** This enables the processor to keep its execution engines as busy as possible by executing instructions that are likely to be needed.

Ans.(C) The problems created due to increase in clock speed and logic density of the processor are:

→ **Power:-** As the density of logic and the clock speed on a chip increase, so the power density increases and also dissipated the heat.

→ **RC delay:-** The speed at which electrons can flow on a chip between transistors is limited by the resistance and capacitance of the metal wires connecting them; specifically, delay increases as the RC product increases.

→ Memory Latency:- Memory access speed (latency) and transfer speed (throughput) lag processor speeds.

Ans: (D) The speedup using a parallel processor with N processors that fully exploits the parallel portion of the program is as follows:

$$\text{Speedup} = \frac{\text{Time to execute Program on a processor}}{\text{Time to execute program on N parallel Processors}}$$

$$\text{Speedup} = \frac{T(1-f) + Tf}{T(1-f) + \frac{Tf}{N}}$$

$$\text{Speedup} = \frac{1}{(1-f) + f/N}$$

Ans: (E) Multicore:-

- The use of multiple processor on the same chip provides the potential to increase performance without increasing the clock rate.
- Strategy is to use two simpler processors on the chip rather than

one more complex processor.

- With two processors larger caches are justified.
- As caches became larger it made performance sense to create two and then three levels of cache on a chip.

MIC:-

- Leap in performance as well as the challenges in developing software to exploit such a large number of cores.
- The multicore and MIC strategy involves a homogeneous collection of general purpose processors on a single chip.

GPUs:-

- Core designed to perform parallel operations on graphics data.
- Traditionally found on a plug-in graphics card, it is used to encode and render 2D and 3D graphics as well as process video.
- Used as vector processors for a variety of applications that require repetitive computations.

Q2:-

Ans: (A) **Effective CPI:-**

$CPI = (1*46000) + (2*33000) + (2*16000)$

$+ (2 * 9000)/100$

$CPI = 162000/100$

$CPI = 1620$

MIPS Rate:-

$\text{MIPS Rate} = 60\,MHz / 1620 * 10^6$

$\text{MIPS rate} = 60 \times \cancel{10^6}\,Hz \,|\, 1620 * \cancel{10^6}$

$\text{MIPS rate} = 0.037$

Execution Time:-

$T = I_c / (MIPS * 10^6)$

$T = 104000 / (0.037 * 10^6)$

$T = 2811 * 10^{-3}$

$T = 2.811\ sec$

Ans: (B) For Machine A.:-

$$CPI = \frac{(1*8 + 3*4 + 4*2 + 3*4) * 10^6}{(8+4+2+4) * 10^6}$$

$$CPI = \frac{40}{18}$$

$CPI = 2.22$

$$MIPS\ rate = 200MHz / 2.22 * 10^6$$

$$MIPS\ rate = 200 * 10^6 / 2.22 * 10^6$$

$$\boxed{MIPS\ rate = 90}$$

$$T = I_c / (MIPS * 10^6)$$

$$T = 18 * 10^6 / 90 * 10^6$$

$$\boxed{T = 0.2\ sec}$$

For Machine B:-

$$CPI = \frac{(1*10 + 2*8 + 4*2 + 3*4) * 10^6}{(10 + 8 + 2 + 4) * 10^6}$$

$$CPI = 46/24$$

$$\boxed{CPI = 1.92}$$

$$MIPS\ rate = 200MHz / 1.92 * 10^6$$

$$MIPS\ rate = 200 * 10^6 / 1.92 * 10^6$$

$$\boxed{MIPS\ rate = 104}$$

$$T = I_c / (MIPS * 10^6)$$

$$T = 24 * 10^6 / 104 * 10^6$$

$$\boxed{T = 0.23\ sec}$$

Ans: (C) a. The MIPS rate could be computed as following:

MIPS rate = $I_c / T * 10^6$

I_c = MIPS rate * T * 10^6

Now by computing the ratio of the instruction count of the IBM RS/6000 to the VAX 11/780 which is:

$18 * 1x * 10^6 / 1 * 12x * 10^6$

= 18/12

= 1.5

b. • Regarding to the VAX 11/780, the

CPI = 5MHz / 1 * 10^6

CPI = $5 \times 10^6 / 1 * 10^6$

CPI = 5

• Regarding to the IBM RS/6000, the

CPI = 25MHz / 18 * 10^6

CPI = $25 \times 10^6 / 18 * 10^6$

CPI = 1.4

Ans:(D) a. Since we have the same instruction mix, that means the additional instructions for each task could be allocated appropriately between the instruction types. Therefore, the following table be gotten:

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 %

Average CPI = $(1 \times 0.6) + (2 \times 0.18) + (4 \times 0.12) + (12 \times 0.1) = 2.64$.

Therefore, the CPI has been increased since the time for memory access is also increased.

b. MIPS = $400/2.64 = 152$.

There is a corresponding drop in the MIPS rate.

c. The speedup factor equals to the ratio of the execution times. The execution time is calculated as the following:

$$T = I_c / (MIPS * 10^6)$$

For one processor, $T_1 = 2 * 10^6 / 178 * 10^6$

$$T_1 = 11ms$$

For the 8 processors, each processor executes 1/8 of the 2million instructions plus the 25,000

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

$$T_8 = 1.8ms$$

Therefore we have

Speedup = 11/1.8

Speedup = 6.11

d. By depending on the information given, it is not obvious how to quantify this effect in Amdahl's equation. Therefore, if it is supposed that the fraction of code, which is parallelizable, is f = 1, then Amdahl's law decreases to Speedup = N = 8. Therefore, the actual speedup is only about 75% of the theoretikel speedup.

{The END}