NAME : SYED TAIMOOR SHAH

ID : 14832

SUBJECT : COMPUTER ARCHITECTURE

DEPARTMENT : BS CS 4TH

ASSIGNMENT NO 2

SUBMITTED TO : SIR AMIN

Date : 15/05/2020

Question No 1

(A) Discuss different desktop applications that requires the great power of contemporary microprossor-based systems?

Ans Different desktop applications that requires the great power of contemporary microprocessor based system are:

- Image processing
- Three-dimensional rendering
- Speech recognition
- Video conferencing
- Multimedia authoring

(2)

* Voice and video annotation of files
* Simulation modelling

(B) Discuss the techniques used in contemporary processors to increase speed?

Ans The techniques used in contemporary processors to increase speed are following:

(i) Pipelining:

Pipeling in enables a processor to work simultaneously on multiple instructions at same time

(ii) Branch prediction:

Branch

prediction potentially increases the amount of work available for the processor to execute.

(iii) **Superscale execution:**

This is the ability to issue more than one instruction in every processor clock cycle. In effect, multiple parallel pipelines are used.

(iv) **Data flow analysis:**

The processor analyzes which instructions are dependent on each other's result, or data, to create an

optimized schedule of instructions.

(v) **Speculative execution:**

Thus enables the processor to keep its execution engines as busy as posible by executing instructions that are likely to be needed.

(C) **Discuss the problems created due to increase in clock speed and logic density of the processor**

**Ans** The problems created due to increase in clock speed and logic density of the processor are:

**Power:**

As the density of and the clock speed increases, the power density increses too and heat is dissipated.

**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 (throughout) lag processor speeds.

(D) Discuss the speedup of a program using multiple processors compared to a single processor using Amdahl's Law?

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

Speed up = Time to execute program on a single processor / Time to execute program on N parallel processors

$= T(1-f) + Tf/T(1-f) + Tf/N = 1/(1-f) + f/N$

(E) Discuss the multicore, MIC, and GPGPU in detail.

Ans Multicore:

The use of multiple processors on the same chip provides the potential to increase performance without increasing the clock rate.

Strategy is to use simpler processors on the chip rather than one more ~~complp~~ complex processor.

With two processors larger caches are justified

**MIC:**

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

**GPUs:**

Core designed to perform parallel operations on graphic 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 a vector processor for a variety of applications that requires repetitive computations.

(A) Question No 2

Ans:

Effective CPI:

$$CPI = (1 \times 46000) + (2 \times 33000) + (2 \times 16000) + (2 \times 9000) / 100$$

$$CPI = 162000 / 100$$

$$CPI = 1620$$

MIPS Rate:

$$MIPS\ rate = 60\ MHz / 1620 \times 10^6$$

$$MIPS\ rate = 60 \times 10^6 / 1620 \times 10^6$$

$$= 60 / 1620$$

$$MIPS\ rate = 0.037$$

Execution Time:

$$T = Ic / (MIPS \times 10^6)$$

$$T = 104000 / (0.037 \times 10^6)$$

$= 104000 / 37 \times 10^3$

$T = 2811 \times 10^{-3}$

$T = 2.811$ sec

(B)

Ans

For Machine A:

$CPI = (1 \times 8 + 3 \times 4 + 4 \times 2 + 3 \times 4) \times 10^6 / (8+4+2+4) \times 10^6$

$CPI = 40 \times 10^6 / 18 \times 10^6$

$CPI = 2.22$

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

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

MIPS rate $= 90$

(11)

$T = Ic / (MIPS \times 10^6)$

$T = 18 \times 10_6 / 90 \times 10_0^6$

$T = 0.2$ sec

For Machine B:

$CPI = (1\times10 + 2\times8 + 4\times2 + 3\times4)\times10^6 / (10+8+2+4)\times10^6$

$CPI = 46/24$

$CPI = 1.92$

MIPs rate $= 200\ MH_2 / 1.92\times10^6$

MIPs rate $= 200\times10^6 / 1.92\times10^6$

MIPS rate $= 104$

$T = Ic / (MIPS \times 10^6)$

(12)

$T = 24 \times 10^6 / 104 \times 10^6$

$T = 0.23$ sec

(C)

Ans: (a) The MIPS rate can be computed as the follow:

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

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

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

$18 \times 1 \times 10^6 / 1 \times 12 \times 10^6$

$= 18/12$

$I_c = 1.5$

(b) Regarding to the VAX/780, the CPI = $(5MHz)/(1\times10^6) = 5\times10^6/1\times10^6$
= 5

Regarding to the IBM RS/6000, the CPI = $(25MHz)/(18\times10^6)$
$= 25\times10^6/18\times10^6$
$= 25/18 =$ 1.4 Ans

(D) Ans(a) Since we have the same instruction mix, which means the additional instructions for each task could be allocated appropriately b/w the instruction type. Therefore, 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 % |

(14)

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

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

(b) MIPS = $400/2.64 = 152$

There is a corresponding drop in the MIPS rate.

(c) The speed up factor equals to the ratio of the execution times. The execution time is called as the following:

$T = Ic / (MIPS \times 10^6)$

For one processor, $T_1 =$

$T_1 = (2 \times 10^6) / (178 \times 10^6) = 11 ms$

For 8 processors, each processor

executes 1/8 of the 2 million instruction plus 25000

$$T8 = 2 * x\ 10^6 / 8 + 0.025 \times 10^6 / 152 \times 10^6$$

$T_8 = 1.8 ms$

Therefore

Speed up = Time to execuiten program on a single prossors / Time to execute program ~~of~~ on N paralled processors.

Speed up = 11/1.8

= 6.11

(d) By depending on the information given, it is not obvious how to quantify this effect in Amdahles equation. Therefore, if it is supposed that

(16)

the fraction code, which is parallelizable, is $f=1$, then Amdahl's law decreases to Speedup = $N=8$. Therefore, the actual speedup is only 75% of the theoretical speedup.