Name	Adnan Khan
Id	14837
Semester	4th
Subject	Computer Architecture
Assignment	2

Q1 Discuss different desktop applications that require the great power of contemporary microprocessor based systems?

Ans Different desktop applications that require the great power of contemporary microprocessor based system are.

* Image processing
* Three-dimensional rendering
* Speech recognition
* Video conferencing
* Multimedia authoring
* Voice and video annotation of files.
* Simulation modeling.

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

Ans The techniques used in contemporary processors to increase speeds are following.

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 process to execute.

Superscalar execution:-

This is the ability to issue more than one instruction in every processor clock cycle. In effect, 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 instruction that are likely to be needed.

C. Discuss the problems created due to increase in clock speed and logic density of the processor?

Ans 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 electron 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.

d. Discuss the speedup of a program using multiple processors compares to a single processor using Amdah l's law?

Ans The speedup using parallel processor with N processors that fully exploits the parallel portion of the Program as follows.

Speedup = 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 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:-

- Logic in a performance as well as the 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 graphics data.
- Traditionaly 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 repetive computations.

A A benchmark program is run on a 60 MHz processor. The execteprogram consists of 104,000 instruction executions, with the instructionmin and clock cycle count given below. Determine the effective CPI, MIPs rate, and execution # time for this program.

Instruction type	Instruction Count	Cycles per Instructior
Integer arithmetic	46000	1
Data type	33,000	2
floating point	16,000	2
Control tranfer	9000	2

Effective CPI:-

CPI = (1 * 46000) + (2 * 33000) + (2 * 16000) + (2 * 9000)/100

CPI = 162000/100

CPI = 1620

MIPs rate:-

MIPs rate = 60 MHz/1620 * 106

MIPs rate = 60 * 106/1620 * 106

MIPs rate = 60/1620

MIP rate = 0.037

Execution time:

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

$T = 104000 / (0.37 * 10^6)$

$T = 104000 / 37 * 10^3$

$T = 2811 * 10^3$

$T = 2.811$ sec

b Consider two different machines, with two different instruction sets, both of which have a clock rate of 200 MHz. The following measurements are recorded on the two machines running a given set of benchmark programs.

Instruction type	Instruction count	Cycles per instruction
Machine A		
Arithmetic & Logic	8	1
Load & Store	4	3
Branch	2	4
others	4	3
Machine B		
Arithmetic & Logic	10	1
Load & Store	8	2
Branch	2	4
Others	4	3

Determine the effective CPI, MIPS rate, and execution time for each machine

For Machine A:

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

$CPI = 40 * 10^6 / 18 * 10^6$

CPI = 2.22

MIPs rate = 200Mz / 2.22 * 106

MIPs rate = 200 * 106 / 2.22 * 106

MIPs rate = 90

T = IC / (MIPs * 106)

T = 18 * 106 / 90 * 106

T = 0.2 sec

For Machine B.

CPI = (1 * 10 + 2 * 8 + 4 * 2 + 3 * 4) * 106 / (10 + 8 + 2 + 4) * 106

CPI = 46 / 24

CPI = 1.92

MIPs rate = 200 MHz / 1.92 * 106

MIPs rate = 200 * 106 / 1.92 * 106

MIPs rate = 104

T = IC / (MIPs * 106)

T = 24 * 106 / 104 * 106

T = 0.23 sec.

C Early examples of CISC and RISC design are the VAX 11/780 and the IBM RS/6000 respectively. Using a typical benchmark program, the following machine characteristics result:

Processor	Clock frequency (MHz)	Performance	CPU time (seconds)
VAX 11/780	5	1	12n
IBM RS/6000	25	18	n

The final colum shows that the vax required 12 times longer than the IBM measured in CPU time.

a What is the relative size of the instruction count of the machine code for this benchmark program running on the two machines?

Ans The MIPs rate could be computed as the following.

$$\text{MIPs rate} = IC / T * 10^6$$

$$IC = \text{MIPs rate} * T * 10^6$$

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 What are the CPI values for the two machines?

Regarding to the VAX 11/780, the CPI = (5 MHz) $(1 * 10^6) = 5 * 10^6 / 1 * 10^6$

$$= 5/1 = 5$$

Regarding to the IBM Rs/6000, the
CPI = (25MHz) / (18 * 106) - 25 * 106 /
18 * 106

= 25/10 = 1.4.

Q Consider the example in section 2.5 for the calculation of average CPI and MIPs rate which yeilded the result of CPI, 2.24 and MIPs rate 178. Now assume that the program can be executed in eight parallel tasks or threads with roughly equal number of instructions executed in each task. Execution is on an 8-core system with each core (processor) having the same performance as the single processor originally used. coordination and synchronization between the parts adds an extra 25,000 instruction executions to each task Assume the same instruction mix as in the example for each task, but increase the CPI for memory reference with cache miss to 12 cycle due to contention for memory.

a) Determine the average CPI.
b) Determine the corresponding MIPs rate.
c) Calculate the speed factor

d. Compare the actual speedup factor with the theoretical speedup factor, determined by Amdahl's law.

Ans 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 in MIX
Arithmetic & Logic	1	60 %
Load/store with Cache Hit	2	18 %
Branch	4	12 %
Memory reference with Cache Mics	12	10 %

The average CPI = (1 x 0.6) + (2 x 0.18) + (4 x 0.12) + (12 x 0.1) = 2.64. Therefore the CPI has been increased since the time for money access is also increased.

b MIPs = 400/2.64 = 152. There is a corresponding drop in the MIPs rate

c The speedup factors eavals to the ratio of the execution times. The

execution time is calculated at the following.

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

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

$= 11ms$.

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

$$T_8 = 2 * 10^6 \div 8 + 0.025 * 10^6 / 152 * 10^6$$

$$T_8 = 1.8 ms$$

Therefore we have

speedup = Time to execute program on single processor / Time to execute program on N parallel processor

$$Speedup = 11 / 1.8$$

$$Speedup = 6.11$$

By depending on the information given. It is not obvious how to quantify this effect in Amdahl's equation. Therefore it is supposed that the fraction of code which is parraller-able is $f = 1$, then Amdahl's law decreases to speedup = N = 8. Therefore the actual speedup is only about 75% of theoratical speed up.