1

Name	Manahil Khan
ID	14525
Assigment no	2nd
Subject	Computer architecture
Submitted to	Sir M. AMIN
program	BS(CS)
Semester	4th

2

Q NO # 1

Part (A)

A:- Discuss different desktop application that require the great power of contemporary microprocessor-based system.

Ans) Different desktop application that require the great power of contemporary microprocessor based system are:

* image processing
* Three-dimensional rendering
* Speech recognination
* video conferencing
* Multimedia authoring
* voice & video annotation of files
* Simulation modeling.

Part (B)

B:- Discuss the technique used in contemporary processor to increase speed?

The technique used in contemporary processor to increase speed are following.

* **pipeling:-**
  pipeling enables a processor to work simultaneusly on multiple instruction by performing a different phase for each of the multiple instructions at the same time.
* **Branch prediction:-**
  Branch prediction potentially increase the amount of work avaiable for process to execute

3

* Superscalar execution:-
This is the ability to issue more than one instruction in every processor clock cycle. In effects multiple parallel pipline are used.

* Data flow analysis:-
The processor analyzes which instruction are dependent on each other's results or data, to create an optomized 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.

Part (E)

E:- Discuss the problem created due to increase in clock speed & logic density of processor.

problem created due to increase in clock speed & logic density of the processor are.

* power
As the density of logic & the clock speed on a chip increase, so the power density increases & also dissipated the heat.

* RC delay:-
The speed at which electron can flow on a chip between transistors

is limited by the resistance & capacitence of the metal wires connecting them specifically, delay increases as the RC product increases.

* Memory latency:-
  Memory access speed (latency) & transfer speed (throughput) lag processor speeds.

---

part (G)

Q:- Discuss the speedup of a program using multiple processors compared to a single processor using Amdahl's law?

The speedup using the parallel processor with N processor 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.

---

part (H)

H: Discuss the multicore MIC & GPGPU in detail.

→ Multicore:-
  The use of multi processors on the same chip provides the potential to increase performance without increasing the clock rate.

Strategy is to use two simpler processor on the chip rather than one more complex processor.

* With two processor larger cache are justified.
* As cache became larger it made performance sense to create two & then three levels of cache on a chip.

* MIC

Leap in performance as well as the challenges in developing software to exploit such a larger number of cores.

* The multicore & MIC strategy involve a homogenous collection of general purpose processor on a single chip.

GPUS:-

* Core design to perform parallel operation on graphics data.
* Traditionally found a plug in graphics card. its used to encode & render 2D & 3D graphics as well as process video.
* Used a vector processor for a variety of application that require computation.

6

Solve each of the following:-

Part (A)

A:- A benchmark program is run on a 60MHz processor. The executed program consist of 104,000 instruction execution, with the instruction mix & clock cycle count given below. Determine the effective CPI MIPS rate & execution time for this program.

Instruction type	instruction count	cycle per instruction
integer arthimatic	46,000	1
Data transfer	33,000	2
Floating point	16,000	2
Control transfer	9000	2

* Effective CPI:-

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

$CPI = 162000/100$

$CPI = 1620$

* MIPS rate

MIPS rate = $60MHz/1620*10^6$

MIPS rate = $60*10^6/1620*10^6$

MIPS rate = $60/1620$

MIPS 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

ANS

7

Part (b)

b:- Consider two different machines with two different instruction set both of which have a clock rate of 200MHz. The following measurement are recorded on two machines runing a given set of benchmark programs

instruction type	instruction count (million)	cycle per instruction
Machine A Arthimatic logic	8	1
load & store	4	3
Branch	2	4
Others	4	3
Machine A		
Arthimatic logic	10	1
load & store	8	2
Branch	2	3
Others	4	4

Determine the effective CPI MIPS rate & execution time for each machine

Ans For Machine A:-

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

8

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

CPI = 2.22

MIPS rate = 200M = / 2.22 * 10^6

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

MIPS rate = 90

T = IC / (MIPS * 10^6)

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

T = 0.2 sec

For machine B

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

CPI = 46 / 24

CPI = 1.92

MIPS rate = 200MHz / 1.92 * 10^6

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

MIPS rate = 104

T = IC / (MIPS * 10^6)

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

T = 0.23 sec

part (C)

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

9

processor	Clock frequency (MHz)	performance (MIPS)	CPU times (seconds)
VAX 11/780	5	1	12n
IBM RS/6000	25	18	n

The final column show 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 runing on the two machines?

MIPS rate could be computed as the following.

MIPS rate = IC/T * 106

IC = MIPS rate * T * 106

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

= 18 * 1 * 106 / 1 * 12x * 106

= 18/12

= 1.5

b) what are the CPI value for the two machines.

10

Regarding to the vax 11/780, the
CPI = (5MHz) (1 * 106) = 5 * 106 / 1 *
106
= 5/1 = 5

Regarding to the IBM RS/6000 the
CPI = (25MHz) / (18 * 106) = 25 * 106 /
18 * 106
= 25/10
= 1.4

part (D)

D:- Consider the example in Section 2.5 for the calculation of average CPI & MIPS rate, which yieded the result of CPI = 2.24 & MIPS rate = 178. Now assume that the program can be executed in eight parallel tasks or threads with roughly equal numbers of instructions executed in each task. Execution is on an 8-core system with each core system (processor) having the same performance as the single processor orginally used. Coordination & synchronization between the parts add an extra 25,000 instruction execution 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 cycles due to contention for memory.

a) Determine the average CPI.
b) Determine the corresponding MIPS rate.
c) Calculate the Speedup factor.
d) Compare the actual Speedup factor with the theoratical speedup factor determined by Amdahl's law.

ANS a) Since we have the same instruction mix, that means the additional instruction for each task could be allocated appropriately between

the instruction types. Therefore the following table be gotten

instruction type	CPI	instruction Mix
Arithmatic & logic	1	60%
local/store with cache hit	2	18%
Branch	4	12%
Memory reference with cache miss	12	10%

The average CPI = (1x0.6) + (2x0.18) + (4x0.12) + (12x0.1) = 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 equals to the ratio of the execution times. The execution time is calculated as the following.

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

For the one processor, $T_1 = (2 * 10^6) / (178 * 10^6) = 11$ ms.

13

FOR the 8 processor, each processor executes 1/8 of the 2 million instruction plus the 25,000.

$T8 = 2*106 \div 8 + 0.025 * 106 / 152 * 106$

$T8 = 1.8ms$

Therefor we have

Speedup = Time to execute program on a single processor / time to execute program on N parallel processor.

Speedup = 11 / 1.8

Speedup = 6.11

* By depending on the information given. its not obvious how to quantity this effect in Amdahl's evuation. Therefor its supposed that the fraction of code which is parallable is f=1 then Amdahl's decreases to Speedup = N = 8. Therefore the actual speedup is only about 75% of the theoritical speedup.