/

CS3853: Computer Architecture
Midterm 1
Spring 2017
Total Marks: 100
Time: 75 min

Name: Solution
Banner Id:

1. [10 pt] Indicate **True** or **False** for the following statements:
   - Speed up of system A over system B is ET_B/ET_A T
   - CPI is proportional to performance. F
   - If 25% of a program cannot be parallelized, then the maximum speed up that can be achieved through parallelization is 5. F
   - Higher MIPS rating always implies lower execution time. F
   - Pipeline improves throughput not latency. T
   - Throughput can be used as a performance metric. T
   - Structural hazard is related to branch instructions. F
   - If a branch is actually taken, then the static *predict not taken* policy incurs stalls. T
   - In the 5 stage MIPS pipeline, register access (i.e., read or write) takes an entire cycle. F
   - Forwarding and bypassing are synonymous. T

2. [16 pt] Amdahl's law:

Suppose you can apply two enhancements - enh_1 and enh_2. enh_1 can be applied to 40% of the program whereas enh_2 can be applied to 30% of the program. The speed ups of enh_1 and enh_2 are 4 and 2 respectively.

(a) [4 pt] Write down the equation of Amdahl's law that applies two enhancements together.

(b) [4 pt] What is the overall speed up if you use only enh_1?

(c) [4 pt] What is the overall speed up if you use only enh_2?

(d) [4 pt] What is the overall speed up if you use both enhancements together?

a) $$S = \frac{1}{1-f_1-f_2+\frac{f_1}{S_{enh_1}}+\frac{f_2}{S_{enh_2}}}$$

b) $$S_1 = \frac{1}{1-f_1+\frac{f_1}{S_{enh1}}}$$

$$= \frac{1}{1-0.4+\frac{0.4}{4}} = \frac{1}{0.7} = 1.43$$

c) $$S_2 = \frac{1}{1-f_2+\frac{f_2}{S_{enh2}}}$$

$$= \frac{1}{1-0.3+\frac{0.3}{2}} = \frac{1}{0.85} = 1.18$$

d) $$S = \frac{1}{1-0.4-0.3+\frac{0.4}{4}+\frac{0.3}{2}}$$

$$= \frac{1}{0.55}$$

$$= 1.82$$

3. [16 pt] CPI calculation:

Suppose your program has 30% load, 20% store, 40% ALU, and 10% branch instructions. The CPIs of those instructions are 1.2, 1.5, 1.05, and 1.9 respectively.

(a) [4 pt] What is the overall CPI?

(b) [4 pt] If we use a faster memory, CPIs of load and store are reduced by 10%. What is the new overall CPI?

(c) [4 pt] If the program has a total of 5 billion instructions and the clock cycle time is 1 ns, what are the original and new execution time?

(d) [4 pt] What is the speed up of the new machine?

a) $CPI_0 = 0{\cdot}3 \times 1{\cdot}2 + 0{\cdot}2 \times 1{\cdot}5 + 0{\cdot}4 \times 1{\cdot}05 + 0{\cdot}1 \times 1{\cdot}9$

$= 1{\cdot}27$

b) $CPI_{load} = 0{\cdot}9 \times 1{\cdot}2 = 1{\cdot}08$

$CPI_{store} = 0{\cdot}9 \times 1{\cdot}5 = 1{\cdot}35$

$CPI_{new} = 0{\cdot}3 \times 1{\cdot}08 + 0{\cdot}2 \times 1{\cdot}35 + 0{\cdot}4 \times 1{\cdot}05 + 0{\cdot}1 \times 1{\cdot}9$

$= 1{\cdot}204$

c) Orig exec time $= 5 \times 10^9 \times 1{\cdot}27 \times 1 \times 10^{-9}$ S

$= 6{\cdot}35$S

New exec time $= 1{\cdot}204 \times 5 \times 10^9 \times 10^{-9}$S

$= 6{\cdot}02$S

d) ~~I~~ Speed up $= \frac{6{\cdot}35}{6{\cdot}02} = 1{\cdot}05$

4. [38 pt] Pipeline & CPI:

Let us consider a classic 5 stage MIPS pipeline where a **branch is resolved in the ID stage** and is handled by **static predict not taken** policy. Consider the following code fragment:

```
L1: LD R1, 0(R10)
SD R1, 4(R2)
ADD R3, R1, R2
ADD R3, R3, R5
BEQZ R3, L1
ADD R3, R2, R4
```

(a) [12 pt] Assume that the pipeline has no forwarding support. Show the pipeline timing diagram of the same code.

Ins	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
L1: LD R1, 0(R10)	F	D	E	M	W																
SD R1, 4(R2)		S	S	F	D	E	M	W													
ADD R3, R1, R2					F	D	E	M	W												
ADD R3, R3, R5						S	S	F	D	E	M	W									
BEQZ R3, L1									S	S	F	D	E	M	W						
ADD R3, R2, R4												F	D	E	M	W					

(b) [12 pt] Assume that the pipeline has full forwarding support. Show the pipeline timing diagram of the same code.

Ins	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
L1: LD R1, 0(R10)	F	D	E	M	W																
SD R1, 4(R2)		F	D	E	M	W															
ADD R3, R1, R2			F	D	E	M	W														
ADD R3, R3, R5				F	D	E	M	W													
BEQZ R3, L1					F	S	D	E	M	W											
ADD R3, R2, R4							F	D	E	M	W										

(c) [14 pt] Assume that our 5 stage MIPS pipeline has full forwarding. Also assume that that branch condition is evaluated in the ID stage, target is calculated in the EX stage and a branch is handled using static **freeze** policy. 10% loads have a dependence with the immediately following ALU instruction and 5% loads have a dependence with the immediately following branch instruction. There are no other load dependencies.

i. [3 pt] What is the number of stalls when a load is immediately followed by a dependent ALU instruction? Show your work.

ii. [3 pt] What is the number of stalls when a load is immediately followed by a dependent branch instruction? Show your work.

iii. [4 pt] What is the average number of stalls caused by a load instruction? Show your work.

iv. [4 pt] What is the CPI of a load instruction? Show your work.

i) LD R1, 0(R2) F D E M W
ADD R3, R1, R2 F D S E M W

# of stall = 1

ii) LD R1, 0(R2) F D E M W
BEQZ R1, L1 F S S D E M W

# of stall = 2

iii) Avg stall = $1 \times 0.1 + 2 \times 0.05$
$= 0.2$

iv) CPI $= 1 + 0.2 = 1.2$

5. [20 pt] Short Questions. Answer precisely.

(a) [4 pt] What is normalized execution time? How can you summarize the normalized execution times of n programs?

(b) [4 pt] If speed up of a program due to some hardware optimization is 3, what is the percentage improvement of performance?

250%

(c) [4 pt] What are the different types of dependences? Give an example of each type.

RAW
WAR
WAW

(d) [4 pt] What are the different ways to eliminate structural hazards?

i) Resource duplication
ii) Stall

(e) [4 pt] In our MIPS 5 stage pipeline, can we have WAR hazard? Explain your answer.

No, read always happens early in the pipeline. before write.