CSE 112:
Computer Organization
Duration : 120 Minutes
Date: June 12, 2023

End Semester Exam Rubric

IIIT-Delhi, Winter 2023
Sujay Deb
Tammam Tillo

Guidelines

If you found any ambiguity in any of the questions or there appears to be a lack of information, then write an assumption on the answer sheet to explain your side interpretation of the problem and solve accordingly.

Problem I: Microprocessor Pipeline Hazards and Execution Time: **8 + 1 + 5 Points**

The instructions supported by a virtual ISA are mentioned in the table below. The ISA has 16 general purpose registers: R0 to R15

Mnemonic	Instruction Format	Operation Explanation
ADD	ADD R1 R2 R3	Performs R1 = R2 + R3
SUB	SUB R1 R2 R3	Performs R1 = R2 - R3
MOV #Imm	MOV R1 **#Imm**	Performs R1 = Imm
MOV	MOV R1 R2	Performs R1 = R2
Branch if equal	BEQ R1 R2 **addr**	Branch to "**addr**" if R1 = R2
Branch unconditional	B **label**	Branch to Label unconditionally

Consider a five-stage pipeline with the stages, Fetch (F), Decode (D), Execute (E), Memory Access (M) and Writeback (W). There is **no-bypassing** between the stages. The description of the stages is given as follows:

Fetch - Fetches the instructions from instruction memory during the second half of the stage.
Decode - Decodes the instructions and type of operations, read in the decode happens in the second half of the stage.
Execute - Executes the instruction in the ALU during the first half of the stage.And the PC update is done during the second half of the stage.
Memory access - Perform loads and stores. All load operations are performed during the second-half of the stage and all store operations are performed during the first-half of the stage. if the instruction is not load and store, then this stage has no effect.
Writeback - Updates the register value during the first half of the stage.

- - - - - - - - - - - - - - - - Assembly Code Begins - - - - - - - - - - - - - -

Instruction No.	Instructions
1	SUB R11, R12, R10
2	ADD R12, R13, R9
3 **loop:**	ADD R13, R13, R10
4	MOV R11, #10
5	BEQ R11, R13, **loop**
6	HALT

- - - - - - - - - - - - - - - - Assembly Code Ends - - - - - - - - - - - - - -

A Five-stage Microprocessor following the above-mentioned ISA is executing the above-mentioned assembly program.
The initials value of R9=1, R10=1, R11=5, R12 =9, R13=9

Please evaluate the following.

a. Draw the pipeline diagram for the assembly code given above..
b. Calculate the CPI.
c. List out all data hazards (RAW, WAR, WAW).

Solution:

Evaluation Note : We are having three possible solutions of this problem. So, please evaluate properly.

a. Pipeline Diagram

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	SUB R11, R12, R10	F	D	X	M	W														
2	ADD R12, R13, R9		F	D	X	M	W													
3 **loop:**	ADD R13, R13, R10			F	D	X	M	W												
4	MOV R11, #10				F	D	X	M	W											
5	BEQ R11 R13, **loop**					F	D	D	D	X	M	W								
6	HALT						F	F	F	D	-	-	-							
7	ADD R13, R13, R10									F	D	X	M	W						
8	MOV R11, #10										F	D	X	M	W					
9	BEQ R11, R13, **loop**											F	D	D	D	X	M	W		
10	HALT												F	F	F	D	X	M	W	
11																				

CPI = (18 Cycles) / (10 Instructions) = 1.8

b. Pipeline Diagram

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	SUB R11, R12, R10	F	D	X	M	W														
2	ADD R12, R13, R9		F	D	X	M	W													
3 **loop:**	ADD R13, R13, R10			F	D	X	M	W												
4	MOV R11, #10				F	D	X	M	W											
5	BEQ R11 R13, **loop**					F	D	D	D	X	M	W								
6	HALT						F	F	F	D	X	-	-							
7	**Random_Instruction**									F	D	-								
8	ADD R13, R13, R10										F	D	X	M	W					
9	MOV R11, #10											F	D	X	M	W				
10	BEQ R11, R13, **loop**												F	D	D	D	X	M	W	
11	HALT													F	F	F	D	X	M	W
12																				

CPI = (19 Cycles) / (11 Instructions) = 1.7

CPI = (19 Cycles) / (10 Instructions) = 1.9

c. HAZARDS

Evaluation Note: *There are six hazards possible. But a student needs to find only 5-hazards. One mark for each hazard.*

Hazard occurring in instruction no.	Hazard Occurred in respect to instruction no.	Hazard occurred in register	Type of Hazard
2	1	R12	WAR
3	2	R113	WAR
4	1	R11	WAW
5	1	R11	RAW
5	4	R11	RAW
5	3	R13	RAW

Problem II: Average Memory Access Time: **3 Points**

A cache has an additional penalty of 120ns whenever there is a miss. The time taken to access data from cache is 10 ns. If the average memory access time has an upper limit of 13 ns. What is the minimum hit rate of the cache?

Solution:

AMAT = (Hit_rate x Hit_time) + Miss_rate x (Hit_time + Miss_penalty).

Let h be the hit ratio of the cache.

Then 13 ns $\geq$ h x 10 + (1-h) x (10 +120)

=> 13 $\geq$ 10h +130 - 130h

=> -117 $\geq$ -120h

=> 120h $\geq$ 117

=> h $\geq$ 117/120

=> h $\geq$ 0.975

The minimum hit rate of the cache is 97.5%.

Problem III: Assembly Program Execution: **10 Points**

The instructions supported by a virtual ISA are mentioned in the table below.

Mnemonic	Instruction Format	Operation Explanation
ADD	ADD R1 R2 R3	Performs R1 = R2 + R3
SUB	SUB R1 #Imm	Performs R1 = R1 - Imm
ADD	ADD R1 #Imm	Performs R1 = R1 + Imm
MOV #Imm	MOV R1 #Imm	Performs R1 = Imm
MOV	MOV R1 R2	Performs R1 = R2
CMP	CMP R1 R2	Set the zero flag if R1 = R2.
JNE	JNE **location**	Jump to "**location"** if zero flag is not set.
OUT	OUT *	Prints the output at stdio.
OUT	OUT \n	Jump to a new line.
HALT	HALT	Stops the program execution.

Execute the below-mentioned program and produce the appropriate pattern.

Serial No. (Instruction Address)	Instructions
1	MOV R3, #5
2	OUT *
3	OUT **\n**
4	MOV R2, #1
5	MOV R1, R2
6 **do_again:**	OUT R1
7	OUT *
8	SUB R1, #1
9	CMP R1, #0
10	JNE **do_again**
11	OUT **\n**
12	ADD R2, #1
13	MOV R1, R2
14	CMP R2, R3
15	JNE **do_again**
16	HALT

Solution:

- - - - - - - - - - - - - - - - - Output Printed Pattern begins- - - - - - - - -

```
*
1 *
2 * 1 *
3 * 2 * 1 *
4 * 3 * 2 * 1 *
```

| 2 Points
| 2 Points
| 3 Points
| 3 Points

- - - - - - - - - - - - - - - - - Output Printed Pattern ends- - - - - - - - - - -

Problem IV: Number Conversion, Cache Addressing: **2 x 3 Points**

Evaluate the following operations.

a) Conversion of a $(127)_{10} = (1002)_x$. Find the value of x.
b) Represent the decimal floating point number (10.75) into a binary floating point number having 3-mantissa bits and 5-exponent bits and no sign bit. Please note that the bias is included in the exponent while storing in binary.
c) Find the location of tag-bits, index-bits, offset-bits in the 32-bit memory address. The data access is 32-bits only. The cache is directly mapped and each cache line can store 4-words. The size of the cache is 16KByte.

Solution:

a) Here,

$(127)_{10} = (1002)_x$
=> $127 = 1x^3 + 0x^2 + 0x^1 + 2x^0$
=> $127-2 = 1x^3$
=> $x = 5$

b) Here, 10.75 in binary is represented in binary as 1010.11
The bias number for the representation is $2^{5-1}-1$ i.e. 15
Then, $1010.11 = 1.01011 \times 10^3$
Exponent = 3 therefore 3+15 = 18 =10010
The floating point representation is, 10010010

Exponent(10010)	Mantissa(010)

c) word size = 32 bit = 4 Byte
block/line size =4 x 4 Byte = 16 Byte
No of Sets $= \frac{16kByte}{16Byte} = 1024 = 2^{10}$
No of bits for set index = 10
Tag bit = 32 - set index - block offset
= 32 -10 -4 = 18

Tag (18 bits)	Index (10 bits)	Block offset (4 bits)

Problem V: Microprocessor Pipeline and Frequency of operation: **3 + 2 Points**

A Microprocessor is operating at 6GHz and has a three-stage pipeline and the stage delays are T1, T2, T3. Delays follow the relation T1 = 6 x (T2) / 7 = 3 x (T3). Assume that the microprocessor has a maximum Instruction Per Cycle(IPC) of 1.

a) What is the delay of each stage in Nanoseconds.
b) If the pipeline having the highest latency is split into two stages having equal delay, what will be the new frequency of the Microprocessor in GHz.

Solution:

a)
Let 't' be the common multiple of each ratio.
T1 = t.
T2 = 7t/6.
T3 = t /3.
Pipeline cycle time = Maximum delay due to any stage + Delay due to register stage
= Max {t, 7t/6, t/3} + 0
= 7t/6.
Frequency of operation = 1 / Pipeline cycle time
= 1 / (7t / 6)
= 6 / 7t.
6 GHz = 6 / 7t.
t = 1/7 ns.
Delayof stage-1 = 1/7 ns.
Delay of stage-2 = 1/6 ns.
Delay of stage-3 = 1/21 ns.

b) The stage with the longest Delay=> stage-2 is now divided into two-further stages.
Now, eventually we have four stages.
Delayof stages = {1/7, 1/12, 1/12, 1/21}ns.
Pipeline cycle time = Maximum delay due to any stage + Delay due to register stage
= Max {1/7, 1/12, 1/12, 1/21}+ 0
= 1/7 ns.
New frequency of operation = 1 / pipeline_cycle_time.
= 1 / 7ns.
= 7 GHz.

Problem VI: Cache Allocation and Eviction: **8 + 2 Points**

A directly mapped cache capable of storing 8 address locations is attached to a microprocessor. Each cache line can store only one microprocessor word. Now, assume that the whole cache is invalidated at any instant of time(there is no data inside the cache corresponding to any address) and then the below-mentioned assembly program is executed.

Task is to fill the below-mentioned table after execution of each instruction. And, do calculate the total number of hits and misses that occur during execution.

Instruction	Hit/Miss	Evicted Address	Address of Newly stored data	Address list of the available data in Cache
LD R1, #4000	Miss	NA	4000	{4000}
LD R1, #4001				
LD R1, #4002				
LD R1, #4009				
ST R1, #4002				
LD R1, #4008				
LD R1, #4007				
ST R1, #4007				
LD R1, #400A				

Solution:

Instruction	Hit/Miss	Evicted Address	Address of Newly stored data	Address list of the available data in Cache
LD R1, #4000	Miss	NA	4000	{4000}
LD R1, #4001	Miss	NA	4001	{4000,4001}
LD R1, #4002	Miss	NA	4002	{4000,4001,4002}
LD R1, #4009	Miss	4001	4009	{4000,4009,4002}
ST R1, #4002	Hit	NA	NA	{4000,40094002}
LD R1, #4008	Miss	4000	4008	{4008,4009,4002}
LD R1, #4007	Miss	NA	4007	{4008,4009,4002,4007}
ST R1, #4007	Hit	NA	NA	{4008,4009,4002,4007}
LD R1, #400A	Miss	4002	400A	{4008,4009,400A,4007}

Total Misses = 7.

Total Hit = 2.

Problem VII: Microprocessor Microarchitecture and Assembly Encoding: **4 + 4 + 4 Points**

A program following the below-mentioned algorithm is implemented using four-diff types of microprocessor architectures. Below is the corresponding assembly program for that.

- - - - - - - - - - - - - - - - Algorithm begins - - - - - - - - - - - - - -

C = A x B;
D = C x A;
E = D + B;

- - - - - - - - - - - - - - - - Algorithm ends - - - - - - - - - - - - - - -

Note: *There is an abundance of registers available for both Register-Memory and Register-Register Architecture. And the variable (A, B, …. G) are memory addresses for the corresponding variable.*

Serial No	Stack Based	Accumulator-Memory Based	Register-Memory	Register-Register
1	Push A	Load A	Load R1, A	Load R1, A
2	Push B	Mul B	Mul R2, R1, B	Load R2, B
3	Mul	Store C	Store R2, C	Mul R3, R1, R2
4	Pop C	Mul A	Mul R3, R2, A	Store R3, C
5	Push A	Store D	Store R3, D	Mul R4, R3, R1
6	Mul	Add B	Add R4, R3, B	Store R4, D
7	Pop D	Store E	Store R4, E	Add R5, R4, R2
8	Push B			Store R5, E
9	Add			
10	Pop E			

Now, the problem is to **choose any three architectures** and write the assembly code for the algorithm **below-mentioned**.

- - - - - - - - - - - - - - - - Algorithm begins - - - - - - - - - - - - - -

C = A x B;
F = E + D;
G = F x B;

- - - - - - - - - - - - - - - - Algorithm ends - - - - - - - - - - - - - - -

Solution:

Serial No	Stack Based	Accumulator-Memory Based	Register-Memory	Register-Register
1	Push A	Load A	Load R1, A	Load R1, A
2	Push B	Mul B	Mul R2, R1, B	Load R2, B
3	Mul	Store C	Store R2, C	Mul R3, R1, R2
4	Pop C	Load E	Load R3, E	Store R3, C
5	Push D	Add D	Add R4, R3, D	Load R4, D
6	Push E	Store F	Store R4, F	Load R5, E
7	Add	Mul B	Mul R5, R4, B	Add R6, R4, R5
8	Pop F	Store G	Store R5, G	Store R6, F
9	Push B			Mul R7, R6, R2
10	Mul			Store R7, G
11	Pop G			