Midsem
CSE/ECE 511 Computer Architecture

INSTRUCTIONS:

Total Marks = 50
Time Duration = 60 mins solving + 10 mins uploading

1. **The duration of the exam is 60mins, and 10 mins for scanning and uploading the solutions. No further extension of time will be given regarding this. Any late submission will be awarded 0 marks.**
2. **The question paper will be uploaded in google classroom. Do not forget to turn it in. Solutions submitted by any other means (email etc.) won't be considered for evaluation.**
3. **Students are required to switch on their cameras and mute themselves. Make sure you are sitting in a well-lit room so that we are able to see your faces clearly. If you are not clearly visible, you will be awarded 0 marks.**
4. **The answers should be in your own handwriting and submission should be in PDF format only.**
5. **Write any assumption clearly, if any. Needless to say, only reasonable assumptions will be considered if any ambiguity is found in the question.**
6. **During the exam, if you have any queries, write them in the meet chatbox. It will be taken into notice by us. Don't unnecessarily unmute your mic for it creates a disturbance to others.**
7. **Calculators are NOT allowed during exam time. ONLY use pen and paper for writing the exam.**
8. **Students need to be present and visible for the whole exam duration (till the end of solution uploading time) even if they upload the solution before time.**
9. **NAMING CONVENTION - <Name>_<Roll number>_MidSem.pdf.**
   Example Abc_Def_2020123_MidSem.pdf
10. **Show your intermediate calculations and justifications in each question.**

Q1

For all the following questions, follow the given instructions: **[7.5 + 7.5 = 15 Marks]**

a) The pipeline has 5 stages: **F**etch, **D**ecode, e**X**ecute, **M**emory and **W**riteback;
b) Each stage requires one clock cycle;
c) All memory references hit in the cache, and assume that you have separate instruction and data memory;
d) L1 and L2 denote labels and are not part of instructions.
e) LWI refers to load immediate, where an immediate value is loaded to a register.
f) The branch instructions are resolved at the X stage.
g) In case of no bypassing for RAW hazards, an extra cycle needs to be spent to decode the updated register values after the write-back stage.

```
MOV R14 R15
LD  R4 0(R5)
LD  R2 4(R4)
LWI  R10 #0
BEQ R14 R15 L1
ADD R7 R8 R9
SUB R7 R3 R6
L1: BNEZ R10, L2
ADD R11 R12 R13
ADD R11 R1 R15
L2: ADD R1 R11 R3
```

a) Calculate how many clock cycles are required to execute the above segment on the simple pipeline without bypassing. Show pipeline timing diagram of the code segment.

b) Calculate how many clock cycles are required to execute the above segment on the simple pipeline with full bypassing. Show pipeline timing diagram of the code segment.

Q2

Consider two types of ISAs: **[15 Marks]**

i) FL-ISA: A fixed-length ISA whose every instruction is **32-bits** long.

ii) **VL-ISA**: A variable-length ISA that has two types of instructions: short (type S), and long (type L). Type S instructions are **16-bits** long, and type L instructions are **32-bits** long.

A single benchmark when compiled on the same ISA yields the following information.

ISA	No. of instructions	Type of instructions
FL-ISA	10,000	All are 32-bit long instructions
VL-ISA	10,000	Instructions are split in the ratio of 1:3 for type S and type L respectively.

For both the types of ISAs, microarchitecture includes two instruction memories whose configurations are mentioned below in the table. Note that, CPU can access only one memory at a time.

Instruction Memory	Address Space	Width of each location	Addressability	Usage
Imem A	32-bit	32-bit	Word (32 -bit) Addressable	Stores 32-bit long instructions only
Imem B	32-bit	16-bit	Half Word (16-bit) Addressable	Stores 16-bit long instructions only

For both the types of ISAs, the CPU implementations follow the steps mentioned in the table below in a loop. For each of these steps energy and latency values are as follows:

Steps	Energy(nJ/byte) 16- bit instructions	Energy(nJ/byte) 32-bit instructions	Latency (ms/byte)
i. The CPU sends the instruction address to memory.	1	1	1
ii. The memory replies with data representing the instruction.	3	1	1
iii. The CPU performs the computations.	0	0	0

Assuming everything else requires 0 energy and 0 time, answer the following questions

a. How many bytes are required to represent the program in FL-ISA? [2 Marks]

b. How many bytes are required to represent the program in VL-ISA? Mention how many bytes are required for type S and type L instructions. [2 Marks]

c. How much energy is required to fetch all instructions in FL-ISA? [2 Marks]

d. How much energy is required to fetch all instructions in VL-ISA? [2 Marks]

e. How much time does the FL-ISA based CPU take to fetch all the instructions of the benchmark? [2 Marks]

f. How much time does the VL-ISA based CPU take to fetch all the instructions of the benchmark? [2 Marks]

g. How much space is required in (Bytes) for storing the instructions of benchmark in both CPU implementations. Specify separately for Imem A and Imem B. [2 Marks]

h. Which ISA would you use for a mobile processor and server processor? Give reasons [1 Mark]

Q3

Through this question, we will try to find out the impact of associativity on the processor memory access time. Consider a processor with a cycle time of **0.4ns**. Assume 3 caches, **Direct mapped**, **2-way** set associative, and **4-way** set-associative caches. All caches have the same size of **8KiB**. Miss rates of the three caches are **6.8%, 4.9%,** and **4.4%** respectively. Assume the cache miss penalty to be the same for all types i.e **100ns**.

You might be aware that introducing associativity increases the hardware complexity in terms of MUXs. As a result, for associative caches(Any type) the cycle time is increased to **1.4ns .** Assume Hit-time of **1 cycle**.

a. Find the AMAT(Average memory access time) for all three configurations.
b. Deduce which configuration of cache is best for this particular situation.

[4.5 + 0.5 = 5 Marks]

Q4

Assume a full-associative, write allocate, and write back cache that uses Least recently used (LRU) as a block replacement policy. For ease, no index and tag bits have been given, instead the question provides the data in cache directly indicating the address it refers to in the main memory. For eg: Address 200 will indicate that this cache line contains the data which is present in the main memory at address 200. Also, dirty bit 1 indicates that the data here is dirty and 0 indicates that the data is not dirty. Direct memory access is not available. The cache can store only six (6) memory blocks at a time.

ST R1, [100] : Store content of register R1 into memory address 100.

LD R1, [100]: Load immediate value 100 into register R1. **[15 Marks]**

Consider the set of instructions given below.

1. LD R3, [100]
2. LD R1, [106]
3. ADD R4, R3, 0(R1)
4. ST R4, [105]
5. SUB R5, R2, R1
6. ST R5, [103]
7. LD R2, [104]
8. ADD R7, R1, R2
9. ST R7, [104]

Main Memory status		Register file status	
Address	Data	Registers	Value
100	55	R1	100
101	10	R2	103
102	18	R3	1
103	16	R4	7
104	69	R5	90
105	53	R6	45
106	21	R7	3

Cache Status			
	Corresponding to addresses of main memory blocks	Data	Dirty_bit
Cache block 1	103	15	0
Cache block 2	105	60	0
Cache block 3	102	18	0
Cache block 4	101	7	0
Cache block 5	-	-	-
Cache block 6	-	-	-

Find the following:

a) **For each instruction, find out if it is a cache hit or cache miss, and the type of miss. Write your answer in the following table format. You can put a dash wherever needed.**

Inst. no.	Destination register value	Miss/Hit	Type of miss

b) Write the status of the cache at the end of the execution of the given set of instructions. Write your answer in the following table format. You can put a dash wherever needed.

	Corresponding to addresses of main memory blocks	Data	Dirty_bit
Cache block 1			