INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Mid-Autumn Semester Examination 2023-24

Date of Examination: **Session: (FN/AN)** **Duration: 2 Hrs** **Full Marks:60**

Subject No. : CS31007 **Subject : Computer Organization and Architecture**

Department: Computer Science and Engineering

Answer all the questions

All parts of a question must be answered together

1. Answer the following questions. **[5+5+(1+2)+2=15]**

   a) In a processor, the instruction type, frequency of usage, and clock cycles required to execute are summarized as follows:

Instruction Type	Usage	Clock Cycles
LOAD & STORE	30%	3
Integer ALU Instruction	40%	2
Branch	20%	3
Floating-point add/subtract	10%	6

In a modification to the processor that also includes enhancements to the instruction set, the following changes are made: (i) the floating-point operations take 4 clock cycles, (ii) 20% of the integer ALU instructions are eliminated (that is, number of ALU instructions reduces by 20%), (iii) 30% of the ***remaining*** integer ALU instructions take 3 clock cycles while rest remain unchanged.

Compare the relative performances of the two versions of the processor.

Instructions	**Before**	**After modification**
		(20% reduction in overall instructions also)
Load/Store	0.3 (3 cycles)	0.3/0.92
ALU Instruction	0.4 (2 cycles)	0.32/0.92 —> 0.3 (3 cycles)
		—> 0.7 (2 cycles)
Branch	0.2 (3 cycles)	0.2/0.92
Floating Point	0.1 (6 cycles)	0.1/0.92 (4 cycles now)

CPI_before = 0.3*3 + 0.4 *2 + 0.2*3 + 0.1*6 = 2.9

CPI_after = (0.3*3 + 0.32 *3 + 0.32 * 2 + 0.2*3 + 0.1*4) / 0.92 = 2.86

Thus, the second processor is faster that the first one [5 marks]

b) Consider a stack-based computer architecture that has four unary operations **ADD**, **SUB**, **MUL** and **DIV**, each of which pops out two operands from the processor stack, operates on them, and pushes the result back into the stack. In addition, there is an instruction **PUSH X**, which pushes the contents of memory location **X** into the stack, and an instruction **POP X**, which pops an operand from the stack and stores it in memory location **X**.

Write a program in assembly language of the machine to evaluate the following expression:

RES = (A * (B – C)) + (D / (E + F))

Clearly state any assumptions you make.

```
PUSH A
PUSH B
PUSH C
SUB
MUL
PUSH D
PUSH E
PUSH F
ADD
DIV
ADD
POP RES
```

[5 marks]

c) State the main difference between the von-Neumann and Harvard architectures. How can the Harvard architecture prove to be beneficial when instructions are executed in a pipeline?

Von-Neumann architecture

Instructions and data are stored in the same memory module.

- More flexible and easier to implement.
- Suitable for most of the general purpose processors.
- General disadvantage:
- The processor-memory bus acts as the bottleneck.
- All instructions and data are moved back and forth through the pipe.

Harvard Architecture

Instructions and data are stored in the same memory module.

- More flexible and easier to implement.
- Suitable for most of the general purpose processors.
- General disadvantage:
- The processor-memory bus acts as the bottleneck.
- All instructions and data are moved back and forth through the pipe.

[2 marks]

In clock cycle 4 of a pipeline execution, an instruction tries to fetch an instruction (IF), while another instruction finishing up in the fourth stage of the pipeline, may be trying to access data (MEM).
In von-Neumann architecture, one of these two operations will have to wait resulting in pipeline slowdown.
In Harvard architecture, the operations can go on without any speed penalty as the instruction and data memories are separate. [1 mark]

d) Explain the difference between ***indirect addressing*** and ***register indirect addressing*** with the help of illustrative examples.

Indirect Addressing

The instruction contains a field that holds the memory address, which in turn holds the memory address of the operand.

Example:

```
ADD   R1,(20A6H)        // R1 = R1 + (Mem[20A6])
```

Register Indirect Addressing

The instruction specifies a register, and the register holds the memory address where the operand is stored.

Example:

```
ADD   R1,(R5)           // PC = R1 + Mem[R5]
```

[1+1 = 2 marks]

2. Answer the following questions. **[4+4+3+4=15]**

a) For the MIPS32 instruction set architecture, assume that the opcodes for the instructions **`SUB`** and **`LOAD`** are **`010011`** (with the value of the **`func`** field as **`111011`**) and **`100101`** respectively. Show the complete 32-bit encoding of the following instructions:

i) **`SUB    R26, R11, R5`** // R26 = R11 – R5

ii) **`LOAD   R3, -25(R10)`** // R3 = Mem[R10-25]

(i) SUB R26, R11, R5

R type: Opcode, rs, rt, rd, shamt, func

010011 01011 00101 11010 00000 111011

(ii) LOAD R3, -25 (R10)

I type: Opcode, rs, rt, 16 bit Immediate with sign extension

100101 01010 00011 1111111111100111 [2+2 = 4 marks]

b) Consider the following switch/case code segment in C, where the switch variable **option** can assume one of the three values in [0,2]:

```
switch (option) {
   case 0:  x = x + 5;  break;
   case 1:  x = x - 3;  break;
   case 2:  y = 8 * y;  break;
}
```

Write down the equivalent code segment in MIPS32 assembly language.

```
beq  $s1,0,case_0
beq  $s1,1,case_1
beq  $s1,2,case_2
j default_case

case_0:      # x = x+5
     addi $s0, $s0, 5
     j    done

case_1:      # x = x-3
     li   $t4,3
     sub  $s0,$s0,$t4
     j    done

case_2:
     li   $t0,8
     mult $s2,$t0
     mflo $s2
     j    done
done: ......
```

[4 marks]

c) Show how you can do the following in MIPS32 assembly language? [3 marks]

i) Load the 32-bit value **2FFF ADAD** (in hexadecimal) in register **R5**.

```
lui   $t0,0x2FFF
ori   $t0,$t0,0xADAD
```

ii) Branch to a label **Loop1** if **R1 < R2**, and to a label **Loop2** otherwise.

```
blt   $t1,$t2,loop1
b     loop2
```

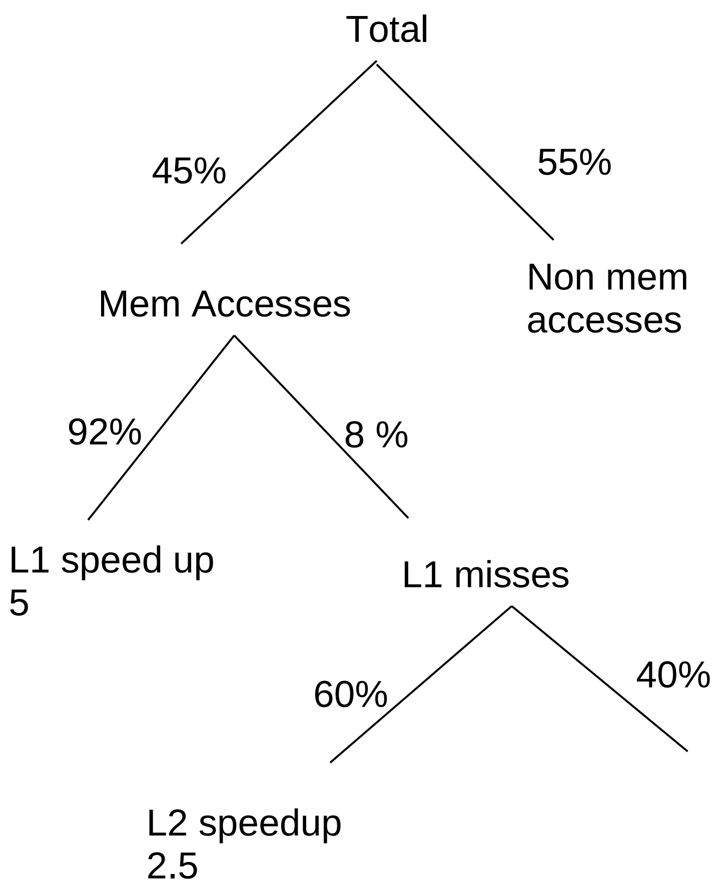
iii) Move the contents of register **R6** to **R8**.

```
addi $t2, $t1, $zero
```

d) Consider a memory system with two levels of cache memory, L1 and L2. Without the cache, memory operations consume 45% of the total execution time. It is experimentally found that the L1-cache speeds up 92% of all memory operations by a factor of 5, while the L2-cache speeds up 60% of the ***remaining*** memory operations by a factor of 2.5. Calculate the overall speedup achieved by using the cache memories ***using Amdahl's law***.

By Amdahl's Law,

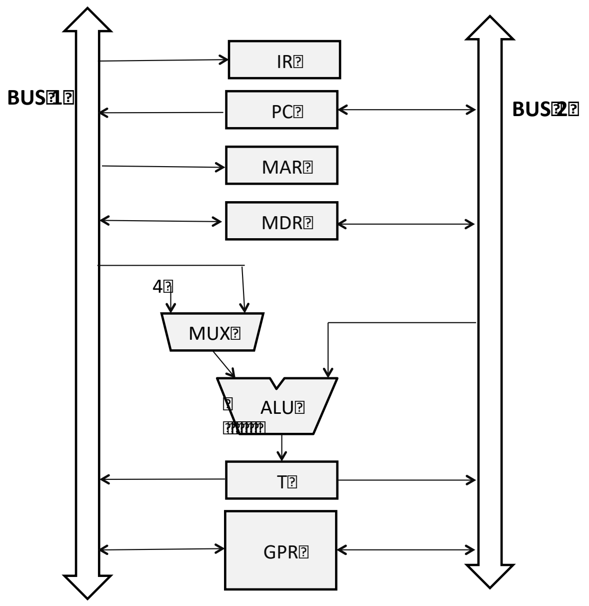
$$\text{Overall Speed up: } \frac{1}{(1-0.45) + 0.45 * (0.92/5 + 0.08 * (0.6/2.5 + 0.4))} = \frac{1}{0.65584} = 1.5247.$$



[4 marks]

3. Answer the following questions. **[(3+3)+(4+2)+3=15]**

a) Consider the two-bus processor architecture as shown in the diagram. In addition to the standard registers IR, PC, MAR and MDR, there is a register bank containing eight general-purpose registers R0, R1, …, R7, and a temporary register T. All the registers are 32-bits in size. MAR and MDR are connected to the memory system, which is not shown in the diagram.



Show the control signals along with the time steps for executing the following instructions:

i) **ADD R3,X** // R3 = R3 + Mem[X]

ii) **LOAD R5,X** // R5 = Mem[X]

Make relevant assumptions as needed. If there is any inconsistency in the architecture diagram, correct the same. Clearly state the assumptions you make.

The control signals must be specified with respect to the various time steps (T1, T2, etc.). No marks are given if micro-operations are shown instead of control signals.

If there is any inconsistency in the above diagram, it must be clearly specified. For instance, the arrow from Bus1 to IR must be bidirectional.

WMFC signal for memory read operations must be shown.

[1.5 + 1.5 marks]

b) With the help of a schematic diagram, briefly explain the concept of microprogrammed control unit design. Briefly explain how branches in microinstruction execution are handled.

This is a descriptive question, where the block diagram of microprogrammed control unit has to be given, along with brief explanation. [2.5 marks for block diagram, 1.5 marks for explanation]

Specifically, it must be explained how branches within microinstructions are handled. Typically, we use the "next address field" within each control word. [2 marks]

c) In the design of the control unit for a processor, there are 150 control signals that can be divided into three groups **G1**, **G2** and **G3**. **G1** contains 35 control signals that can be activated either individually or in parallel. **G2** contains 75 control signals that are mutually exclusive (that is, at most one of them can be active at a time). Similarly, **G3** contains 40 control signals that are also mutually exclusive. Suggest a suitable encoding for the control signals that require the smallest number of bits without sacrificing concurrency.

G1 must use horizontal encoding, which requires 35 control signals.

G2 must use vertical encoding – for 75 control signals, we require 7 bits.

G3 must use vertical encoding – for 40 control signals, we require 6 bits.

Hence, total number of bits required = 35 + 7 + 6 = 48 bits

[Total 3 marks, if G1 is answered wrongly but G2 & G3 are correct, 1.5 marks is deducted]

4. Answer the following questions. **[7+4+4=15]**

a) Draw the schematic diagram to design a 256K x 16 memory system using 64K x 8 memory modules. Clearly show how the different address blocks are mapped to the individual memory modules.

Number of memory modules required = (256K x 16) / (64K x 8) = 8

Each 64K x 8 module will have 16 address lines and 8 data lines.

For the overall 256K x 16 memory system, there will be 18 address lines and 16 data lines.

The address bits A16 and A17 will be fed to a decoder, the outputs of which will be selecting one of four module rows (each module row will have two 64K x 8 modules).

The schematic diagram must clearly show the decoder connections to generate the chip select signals, how the remaining 16 address lines are connected to the individual modules, and how the data lines are connected to the modules (D15-D8 for the first column of 4 modules, and D7-D0 for the second column of 4 modules). [5 marks for schematic diagram]

High order address lines (A_{31} and A_{30}) select one of the memory modules.

a) When is M0-1 and M0-0 selected?
   - Address is: 0 0 x x x x x x x x x x x x x x x x
   - Range of addresses is: 0x00000 to 0x0FFFF

b) When is M1-1 and M1-0 selected?
   - Address is: 0 1 x x x x x x x x x x x x x x x x
   - Range of addresses is: 0x10000 to 0x1FFFFF

c) When is M2-1 and M2-0 selected?
   - Address is: 1 0 x x x x x x x x x x x x x x x x
   - Range of addresses is: 0x20000 to 0x2FFFFF

d) When is M3-1 and M3-0 selected?
   - Address is: 1 1 x x x x x x x x x x x x x x x x
   - Range of addresses is: 0x30000 to 03xFFFFF [2 marks]

b) Consider a computer system with two-level cache. Suppose that in 2000 memory references there are 100 misses in L1 cache and 20 misses in L2 cache. If the miss penalty of L2 is 100 clock cycles, hit time of L1 is 2 clock cycle, and hit time of L2 is 20 clock cycles, what will be the average memory access time in number of clock cycles?

For L1 cache, out of 2000 references, there are 100 misses, and hence 2000-100 = 1900 hits.

Hit ratio $H_{L1} = 1900/2000 = 0.95$

For L2 cache, total number of accesses is 100 (i.e. number of misses in L1). Out of these, 20 are misses, and hence 100-20 = 80 are hits.

Hit ratio $H_{L2} = 80/100 = 0.80$

Thus, average memory access time

$$\begin{aligned} T_{avg} &= H_{L1} * 2 + (1 - H_{L1}) * [H_{L2} * 20 + (1 - H_{L2}) * 100] \\ &= 0.95 * 2 + 0.05 [0.80 * 20 + 0.20 * 100] \\ &= 1.90 + 0.05 (16 + 20) \\ &= 3.7 \text{ clock cycles} \end{aligned}$$

[4 marks, partial marks awarded for wrong calculations]

c) What is memory interleaving? How does it help to improve the processor-memory bandwidth?

The main points that need to be mentioned is that in memory interleaving, consecutive addresses are mapped to consecutive memory modules. Thus, if there are four interleaved modules, four consecutive bytes can be accessed in a single memory cycle. This improves the processor-memory bandwidth.

[4 marks]