

Roll No.:

Name:

Department of Computer Science and Engineering
Computer Organization and Architecture (CS31007)

IIT Kharagpur
Class Test 1

Autumn 2022-23
(Duration: 1 Hour)

1. Consider two different implementations, Machines M1 and M2, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. M1 has a clock rate of 2.4 GHz and M2 has a clock rate of 3 GHz. The average number of cycles for each instruction class and their frequencies for a typical program are as follows:

Class of Instruction	Machine M1 (Cycles per Instruction Class)	Machine M2 (Cycles per Instruction Class)	Frequency of each Instruction Class
A	1	2	50%
B	3	3	30%
C	5	4	20%

What is the Average Cycles per Instruction (CPI) for M1 and average MIPS rating for M2?

[3 Marks]

Answer [You MUST briefly show your calculations]:

$$\text{CPI_M1} = 1*50/100 + 3*30/100 + 5*20/100 = 2.4$$

$$\text{Similarly, CPI_M2} = 2*50/100 + 3*30/100 + 4*20/100 = 2.7$$

$$\text{MIPS_M2} = (3*10^9)/(2.7*10^6) = 1111.11$$

2. Suppose that we can improve the floating point instruction performance of machine by a factor of 15 (the same floating point instructions run 15 times faster on this new machine). At least what percentage of the execution time must be due to floating point operations for achieving a Speedup of 4? [3 Marks]

Answer [You MUST briefly show your calculations]:

Let n be the total time and k be the required fraction. Then, floating point time is $k*n$ and the rest is $(1-k)*n$.

As per the problem, $n/4 = kn/15 + (1-k)n$

i.e., $1/4 = k/15 + (1-k)$. Solving for k, we get **45/56 = 80.36%**

3. Computer A has an overall CPI of 2.0 and can be run at a clock rate of 4GHz. Computer B has a CPI of 2.5 and can be run at a clock rate of 2.5Ghz. We have a particular program we wish to run. When compiled for computer A, this program has exactly 50,000 instructions. How many instructions would the program need to have when compiled for Computer B, in order for the two computers to have exactly the same execution time for this program? [4 Marks]

Answer [You MUST briefly show your calculations]:

Let x be the required number of instructions. As per the problem,

$50000*2/4 = x*2.5/2.5$. Solving for x, we get **25000**

4. Consider the following MIPS assembly language code segment below (Note that it is a stored program computer). How many memory accesses will be required for executing this code segment? [2 Marks]

```
lw      $v1, 0($a0)
addi   $v0, $v0, 1
sw      $v1, 0($a1)
addi   $a0, $a0, 1
```

Answer with brief justification: **6**. 4 for fetching the instructions and 1 each for lw and sw.

5. Consider the following MIPS assembly language code segment:

```

Loop: sll    $t1, $s3,2
      add   $t1, $t1, $s6
      lw    $t0, 0($t1)
      bne   $t0, $s5, Exit
      addi  $s6, $s3, 2
      addi  $t0, $s6, 1
      j     Loop

```

Exit:

Assume that the corresponding machine code is placed starting at memory location 12004 as shown below. Fill in the contents of the cells **A-J**. Write your answer (in decimal) inside the box next to the corresponding letter. **[5 Marks]**

12004	0	A 0	19	9	2	0
12008	0	9	22	9	0	32
12012	35	B 9	8	C 0		
12016	5	8	21	D 3		
12020	8	E 19	22	F 2		
12024	G 8	H 22	I 8	1		
12028	2	J 3001				
12032	(Code on Exit. Ignore the content)					

6. Consider the following MIPS code segment. Next to each line after the # symbol, write the type of addressing mode being used. The various addressing modes are: Register Addressing (R), Immediate Addressing (I), PC-Relative Addressing (P), Base Addressing (B), Pseudodirect Addressing (D). Simply write the corresponding letter clearly. **[5 Marks]**

fact:

```

addi $sp, $sp, -8      # I
sw   $ra, 4($sp)       # B
sw   $a0, 0($sp)       # B
slti $t0, $a0, 1        # I
beq  $t0, $zero, L1     # P
addi $v0, $zero, 1       # I
addi $sp, $sp, 8        # I
jr   $ra                 # R
L1: addi $a0, $a0, -1    # I
jal  fact                # D
lw    $a0, 0($sp)       # B
lw    $ra, 4($sp)       # B
addi $sp, $sp, 8        # I
mul  $v0, $a0, $v0       # R
jr   $ra                 # R

```