

[Question 1]

A program's run time is determined by the product of instructions per program, cycles per instruction, and clock frequency. Assume the following instruction mix for a MIPS-like instruction set and compute the overall CPI. **Show your work.**

Instruction Type	% of program	Cycles needed
Store and Load	40%	2
Branch	10%	4
Integer Add	30%	1
Shift	10%	1
Integer Multiply	10%	10

[Question 2]

Consider three different processors P1, P2, and P3 executing the same instruction set.

Processor	Clock Speed in GHz	CPI
P1	3.0	1.5
P2	2.5	1.0
P3	4.0	2.2

Which processor has the highest performance expressed in instructions per second? Remember, 1GHz = 10^9 Hz. **Show your work.**

[Question 3]

- a. Provide the assembly language instruction for the following R type instruction which is written in binary: [Use the MIPS Green Sheet]

0000 0010 0001 0000 1000 1000 0010 0000

- b. Provide the binary of the following R type Instruction [Use the MIPS Green Sheet]

sub \$v1,\$v1,\$v0

[Question 4]

For the following C statement, what is the corresponding MIPS assembly code? Assume that the variables *f*, *g*, *h*, and *i*, are given and are assigned to registers \$s2, \$s3, \$s4, and \$s5 respectively. Use a minimal number of MIPS assembly instructions.

$f = (g + h) - (5 + i);$

[Question 7]

Given the following instruction,

```
    beq $s0, $s1, L1
```

Replace it by using other instruction(s) that offers a much greater branching distance.

[Question 8]

Main routine M1 calls a procedure P1 with the instruction `jal P1`. The return address in the main routine when it returns from the procedure P1 is labelled as RM. The stack pointer has an initial value of `0x0700C`.

Procedure P1 will use save registers `$s0` and `$s1`, and hence need to save these on the stack.

Procedure P1 calls another procedure P2, with a return address (in P1) labeled RP1.

Note, RM (which is stored in `$ra`) needs to be saved on the stack and restored eventually to `$ra` to return to the Main routine (since there is a call to another procedure).

Procedure P2 uses save register `$s4`, (and hence needs to be saved on the stack). Show the stack pointer value right after `$s4` is saved. Also, show the stack contents (updated in the entire sequence).