# NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI
## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
## CYCLE TEST -I

**Subject Code/ Name: CSPC34/ Computer Organization** **Date:21 / 09/ 2022**
**Marks: 20** **Time: 11:00AM–12:00 PM**

106121002

**Answer all the Questions**

1. List out and discuss briefly the eight great ideas invented by computer architects. (3)
2. Translate the following C code to MIPS. Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively. Assume that the elements of the arrays A and B are 4-byte words: (4)

```
while (f==g) A[f] = B[f] + h
```

3. For the register values shown above, what is the value of $t2 for the following sequence of instructions? Assume $t0 as holding the value 0xABCDEFAA (2)

```
srl $t2, $t0, 4
andi $t2, $t2, 0xEEEF
```

4. Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class A, B, C, and D). P1 with a clock rate of 3.5 GHz and CPIs of 1, 2, <*last digit of your roll_no*>, and 3, and P2 with a clock rate of 3 GHz and CPIs of 1, <*second largest digit of your full roll_no.*>, 3, and 2.

   Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 20% class A, 30% class B, 40% class C, and 10% class D, which implementation is faster? (4)

   i. What is the global CPI for each implementation?

   ii. Find the clock cycles required in both cases.

5. What are Pseudoinstructions? Why are they used? Give two examples. (2)
6. Provide the type, assembly language instruction, and binary representation of the instruction described by the MIPS fields: `op=0x43, rs=4, rt = 3, constant = 0x55`. Explain your answer. (2)
7. Distinguish between server computers and super computers. (2)
8. What are the different kinds of branch instructions in the MIPS ISA? (1)

0x43 → 0x[illegible]

00

~~01~~00 0011

0x3

**op(31:26)**

| 28–26<br>31–29 | 0(000) | 1(001) | 2(010) | 3(011) | 4(100) | 5(101) | 6(110) | 7(111) |
|---|---|---|---|---|---|---|---|---|
| 0(000) | R-format | Bltz/gez | jump | jump & link | branch eq | branch ne | blez | bgtz |
| 1(001) | add immediate | addiu | set less than imm. | set less than imm. unsigned | andi | ori | xori | load upper immediate |
| 2(010) | TLB | FlPt |  |  |  |  |  |  |
| 3(011) |  |  |  |  |  |  |  |  |
| 4(100) | load byte | load half | lwl | load word | load byte unsigned | load half unsigned | lwr |  |
| 5(101) | store byte | store half | swl | store word |  |  | swr |  |
| 6(110) | load linked word | lwc1 |  |  |  |  |  |  |
| 7(111) | store cond. word | swc1 |  |  |  |  |  |  |

**op(31:26)=010000 (TLB), rs(25:21)**

| 23–21<br>25–24 | 0(000) | 1(001) | 2(010) | 3(011) | 4(100) | 5(101) | 6(110) | 7(111) |
|---|---|---|---|---|---|---|---|---|
| 0(00) | mfc0 |  | cfc0 |  | mtc0 |  | ctc0 |  |
| 1(01) |  |  |  |  |  |  |  |  |
| 2(10) |  |  |  |  |  |  |  |  |
| 3(11) |  |  |  |  |  |  |  |  |

**op(31:26)=000000 (R-format), funct(5:0)**

| 2–0<br>5–3 | 0(000) | 1(001) | 2(010) | 3(011) | 4(100) | 5(101) | 6(110) | 7(111) |
|---|---|---|---|---|---|---|---|---|
| 0(000) | shift left logical |  | shift right logical | sra | sllv |  | srlv | srav |
| 1(001) | jump register | jalr |  |  | syscall | break |  |  |
| 2(010) | mfhi | mthi | mflo | mtlo |  |  |  |  |
| 3(011) | mult | multu | div | divu |  |  |  |  |
| 4(100) | add | addu | subtract | subu | and | or | xor | not or (nor) |
| 5(101) |  |  | set l.t. | set l.t. unsigned |  |  |  |  |
| 6(110) |  |  |  |  |  |  |  |  |
| 7(111) |  |  |  |  |  |  |  |  |