Department of Computer Science and Engineering
Motilal Nehru National Institute of Technology
Mid-Semester Exam, Computer Organization (CS1403)
BTech (IT, CSE) IV Semester
Time: 1.5 Hours, MM:20

Note: There are 5 questions. Attempt all.

1. **(2 marks)Warm up:** Consider Figure.1 for answering the following questions.

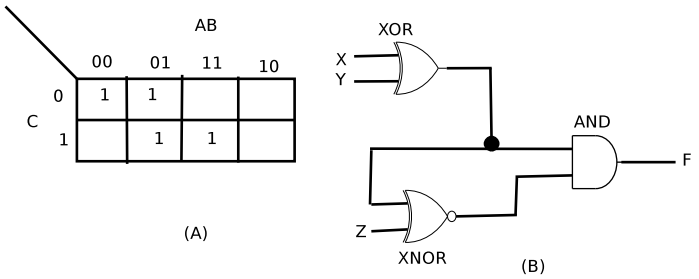

Figure 1: Figure for Question.1

   (a) Which logic circuit is a realization of the function F whose Karnaugh map is given in Figure.1(A)
   (b) What is the output F in the digital logic circuit shown in Figure.1(B).

2. **(3 marks)Processor Design, Single Cycle Implementation:** While answering this question you can take help of MIPS datapath given in Figure.2.

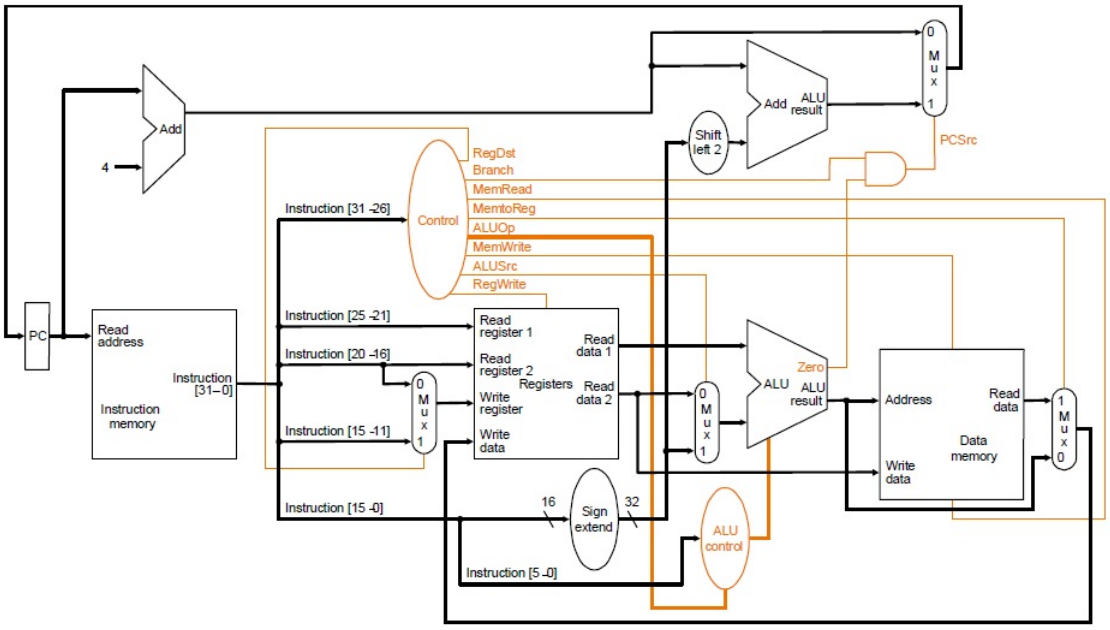

Figure 2: Single Cycle MIPS data-path

   (a) Explain briefly the execution of *sw* instruction using Figure.1 (Maximum 6 sentences).
   (b) Exaplain the role and purpose of different multiplexers in the figure (not more than two sentences for each multiplexer).

(c) How the control signals for multiplexers are generated(You are not supposed to design full control unit, just provide the idea how it is done).

3. **MIPS ISA:**

(a) **4 marks** Write the MIPS assembly instructions for following C language statement:

$$X[i] = A[B[i]] + C[B[i+4]];$$

Here, X, A, B and C are arrays stored in memory. You can assume that starting addresses of these arrays are contained in registers $s0, $s1, $s2 and $s3 respectively.

(b) **4 marks** Consider the C code given in Figure.3. Answer the following questions.

```
int main(void) {
 int i=0;   # i maps to $s0
 int j=1;   # j maps to $s1
 int k=2;   # k maps to $s2
 int l, m, n, o, p, q; # see below
 int x, y, z;          # see below
 l = i+j+k; # l maps to $s3
 m = i*j*k; # m maps to $s4
 n = l-m;   # n maps to $s5
 o = i+j;   # o maps to $s6
 p = m-n;   # p maps to $s7
 q = n+o-p; # q maps to $t1
 x = function_call_1(o,p,q);
 m = l+x+j;
 y = function_call_2(m,l,x);
 z = x+y; }
```

```
int function_call_1(x,y,z) {
      int a;    # a maps to $s1
      int b;
      a = x+y+z;
      b = function_call_3(a);
      return b;}

int function_call_2(x,y,z) {
      int a;    # a maps to $s1
      a = x-y-z;
      return a;}

int function_call_3(x) {
      return x + x; }
```

Figure 3: Code for Question.3

- Write the MIPS assembly for the line "return x+x;" in function_call_3.
- What stack operations (if any) take place inside of function_call_1?
- What MIPS instruction(s) would you expect to see before function_call_1 in main()?
- What register would the variable x map to in the line "x = function_call_1(o,p,q)" in main()?
- What stack operations (if any) must main() perform?

4. **(4 marks)Performance:** Assume that you have the following mix of instructions with average CPI given in Table.1. The clock rate for this machine is 1 GH. You want to improve the performance of this machine, and are considering redesigning your multiplier to reduce the average CPI of multiply instructions.If you make this change, the CPI of multiply instructions would drop to 6 (from 8). The percentage of ALU instructions that are multiply instructions is 23%. How much will performance improve by?

Table 1: Parameters for Question.4

	% of Mix	Average CPI
ALU	47%	6.7
Load	19%	7.9
Branch	20%	5
Store	14%	7.1

5. **(3 marks) Freebies:** Explain briefly the following (Maximum 6 sentences):

(a) Target address calculation in beq, j and jal instructions.

(b) Dealing with 32 bit constants and how to branch far away.