Indian Institute of Technology Kharagpur

EXAMINATION ANSWER SCRIPT

Stamp/Signature of the Invigilator

EXAMINATION: Mid Semester	SEMESTER: Spring 2023-24

Roll Number											Section		Name	

Subject Number	C	S	3	1	7	0	2		Subject Name	Computer Architecture and Operating Systems

Department/Centre/School		Additional Sheets	

Important Instructions and Guidelines for Students

1. You must occupy your seat as per the Examination Schedule/Sitting Plan.
2. Do not keep mobile phones or any similar electronic gadgets with you even in the switched off mode.
3. Loose papers, class notes, books or any such materials must not be in your possession; even if they are irrelevant to the subject you are taking examination.
4. Data book, codes, graph papers, relevant standard tables/charts or any other materials are allowed only when instructed by the paper-setter.
5. Use of instrument box, pencil box and non-programmable calculator is allowed during the examination. However, the exchange of these items or any other papers (including question papers) is not permitted.
6. Write on both sides of the answer-script and do not tear off any page. **Use last page(s) of the answer-script for rough work.** Report to the invigilator if the answer-script has torn or distorted page(s).
7. It is your responsibility to ensure that you have signed the Attendance Sheet. Keep your Admit Card/Identity Card on the desk for checking by the invigilator.
8. You may leave the Examination Hall for wash room or for drinking water for a very short period. Record your absence from the Examination Hall in the register provided. Smoking and the consumption of any kind of beverages are strictly prohibited inside the Examination Hall.
9. Do not leave the Examination Hall without submitting your answer-script to the invigilator. **In any case, you are not allowed to take away the answer-script with you.** After the completion of the examination, do not leave your seat until the invigilators collect all the answer-scripts.
10. During the examination, either inside or outside the Examination Hall, gathering information from any kind of sources or exchanging information with others or any such attempt will be treated as **'unfair means'**. Don't adopt unfair means and also don't indulge in unseemly behavior.

Violation of any of the above instructions may lead to severe punishment.

Signature of the Student

To be Filled by the Examiner

Question Number	1	2	3	4	5	6	7	8	9	10	Total
Marks Obtained											

Marks Obtained (in words)	Signature of the Examiner	Signature of the Scrutineer

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Mid-Spring Semester Examination 2023-24

Date of Examination: 20-Feb-2024 Session: AN Duration: 2 hrs. Full Marks: 60
Subject No.: CS31702 Subject: Computer Architecture and Operating Systems
Department: Computer Science and Engineering
Specific charts, graph paper, log book etc., required: NONE
Special Instructions (if any): No clarification will be provided during the exam. State any assumptions made.
Answers must be written only in the boxes provided in the booklet.
Rough works may be done only in the specified spaces provided in the booklet.

ANSWER ALL THE QUESTIONS

Space for Rough Work

Q1. Answer the following. **[4+3+3 = 10 marks]**

a) Consider the following three instructions stored in consecutive memory locations in a byte addressable computer:

```
MOV   R1,R2          // R1 = R2
ADD   R3,LOCA        // R3 = R3 + Mem[LOCA]
SUB   R4,LOCB        // R4 = R4 - Mem[LOCB]
```

The first instruction is stored at memory location 2000 (in hexadecimal), the addresses LOCA and LOCB are 3004 (in hexadecimal) and 3100 (in hexadecimal) respectively. The contents of memory location LOCA and LOCB are 3E (in hexadecimal) and AE (in hexadecimal). Assume each instruction is of size 32 bits (i.e. 4 bytes). Answer the following, assuming that instruction fetch and decode are already over. **[1+1.5+1.5=4]**

i) What will be the content of program counter (PC) while the second instruction (`ADD R3,LOCA`) is getting executed?

ii) What will be the content of memory address register (MAR) while the second instruction (`ADD R3,LOCA`) is getting executed?

iii) What will be the content of memory data register (MDR) while the third instruction (`SUB R4,LOCB`) is getting executed?

Answer to Q1(a):

b) Consider a 0-address machine where the operands are stored in the stack, with instructions like `ADD`, `SUB`, `MUL`, `DIV`, `PUSH X`, and `POP X`, where 'X' refers to a memory location. Show the assembly language code to evaluate the following expression: **[3]**

```
A = (B * (C + D)) - (E * F)
```

Answer to Q1(b):

c) Consider the following code segment for a 1-address accumulator-based machine. Assume that the variables **X**, **Y** and **Z** in memory are initialized with the values 50, 20 and 15 respectively (all in decimal). What will be the final values of **X**, **Y** and **Z**? **[3]**

```
LOAD   X
ADD    Y
SUB    Z
STORE  X
LOAD   Y
SUB    X
STORE  Z
```

Answer to Q1(c):

X = Y = Z =

Q2. Answer the following. **[3+4+3 = 10 marks]**

a) In one sentence each, state how an operand stored in memory can be accessed using: (i) register indirect addressing, (ii) direct addressing, and (iii) indirect addressing. **[3]**

Answer to Q2(a):

(i) **Register indirect addressing:**

(ii) **Direct addressing:**

(iii) **Indirect addressing:**

b) Consider a processor with the following specifications. The processor uses 32-bit (4-byte) instruction format. It has 32 registers (R0, R1, ..., R31), each of 32-bits size. There are a total of 50 different instructions, which are categorized into two types – ***R-type*** and ***I-type***. In the R-type instruction, three register operands can be specified. In the I-type instruction, two register operands and an immediate data can be specified. Some example instructions are as follows:

```
ADD    R2,R5,R9          // R2 = R5 + R9       (R-type)
ADDI   R3,R5,#25         // R3 = R5 + 25       (I-type)
LOAD   R1,73(R4)         // R1 = Mem[R4+73]    (I-type)
```

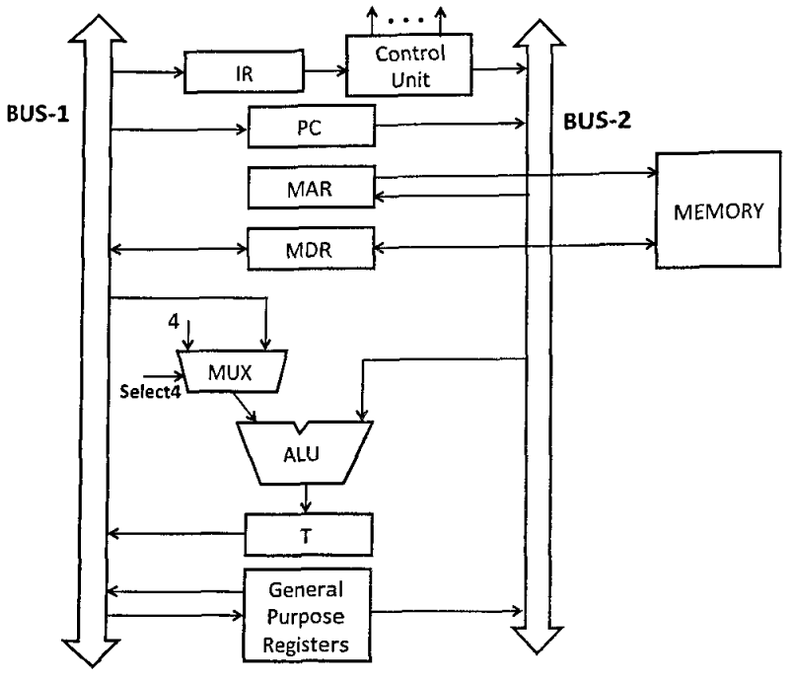
Show a possible instruction format for the R-type and I-type instructions, clearly mentioning the size and purpose of the different fields in the instruction. **[4]**

Answer to Q2(b):

c) In the design of the control unit for a processor, there are 118 control signals that can be divided into three groups **G1**, **G2** and **G3**. **G1** contains 40 control signals that can be activated either one (or multiple) at a time. **G2** contains 55 control signals that are mutually exclusive (that is, at most one of them can be active at a given time). Similarly, **G3** contains 23 control signals that are also mutually exclusive. For a micro-programmed control unit, suggest a suitable microinstruction word encoding that requires the smallest number of bits without sacrificing concurrency. Show a schematic diagram showing how the control signals are decoded. **[3]**

Answer to Q2(c):

Q3. Consider the two-bus processor architecture as shown in the diagram below. In addition to the standard registers IR, PC, MAR and MDR, there are eight general-purpose registers R0, R1, ..., R7 in a register bank, and a temporary register T. All the registers are 32-bits in size. MAR and MDR are connected to the memory system as shown. Two registers can be simultaneously read from the register bank, one through BUS-1 and the other through BUS-2. Follow the usual convention for naming the control signals; in addition, Ri_{out}^{B1} and Ri_{out}^{B2} respectively denote the control signal to read the content of register Ri to the bus BUS-1 and BUS-2 respectively. **[4+3+3 = 10 marks]**



a) Show the control signals along with the time steps for fetching and executing the following instructions: [4]

```
SUB     R1,R2       // R1 = R1 - R2
STORE   X,R3        // Mem[X] = R3
```

Answer to Q3(a):	

b) Draw the schematic diagram to generate the time-step signals T1, T2, ... using a step counter/decoder circuit, assuming that the maximum number of time steps in an instruction is 7. [3]

Question 3(b):

c) For hardwired control unit design, show the logic equations to generate the control signals MAR_{in} and T_{in} with respect to the two instructions given in part (a). **[3]**

Answer to Q3(c):

Q.4 (a) Consider a scenario where a CPU has to run 5 processes with different arrival and burst times. Calculate the **average waiting time** for the given scenario with the following scheduling algorithms: **[2.5 +2.5 = 5]**

i) Non-preemptive Shortest Job First (SJF)

ii) Round Robin (RR) Scheduling (time quantum = 5ms)

Process	P1	P2	P3	P4	P5
Arrival Time	0	0	4ms	6ms	8ms
Burst Time	8ms	2ms	40ms	8ms	6ms

Question 4(a):

Q4. (b) Consider three processes P1, P2, P3 with CPU burst times 2, 4, 8 units. All processes arrive at the time zero. Consider the preemptive longest remaining time first (LRTF) scheduling algorithm. In this algorithm, the process with the longest next remaining CPU time will be scheduled first. In LRTF, ties are broken by giving priority to the process with the lower process id. Compute the average turnaround time with the help of Gantt chart. Assume that the time units are integral. **[5]**

Question 4(b):

Q4. (c) A process-management module uses a prioritized round-robin (RR) scheduling policy (that is, the ready queue is implemented as a priority queue). New processes are assigned an initial quantum of length "q". Whenever a process uses its entire quantum without blocking, its new quantum is set to twice its current quantum (for its next turn). If a process blocks before its quantum expires, its new quantum is reset to "q". You assume that every process requires a finite total amount of CPU time.

Is starvation possible for each of the following cases? The scheduler gives higher priority to (i) processes that have larger quanta, and (ii) processes that have smaller quanta. Justify our answers in both the cases.

[1+2+2 = 5]

Question 4(c):

Q.5 (a) Consider the following C code snippet?

```
...
for (int i=0; i<n; i++) {
        fork();
        printf("\nTest1");
        if (fork()==0){
              printf("\nTest2");
        }
}
printf("\nTest3");
...
```

How many times "Test1", "Test2" and "Test3" will be printed? Justify. **[5]**

Question 5(a):

Q.5 (b) What are Singular, Orphan and Zombie processes? Give example scenarios. **[5]**

Question 5(b):

Q.5 (c) Consider the following C-program (Assume all function calls are successful).

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>

#define BUFSIZE 100

main(){
      int fd[2], i;
      char mybuf[BUFSIZE];

      pipe(fd);

      if (fork() > 0) {
            close(_____);
            for (i=0; i < 10; i++) {
                  printf("Parent Process..\n");
                  strcpy(mybuf, "Msg from Parent");
                  write(_____, mybuf, BUFSIZE);
                  printf("Parent writes: %s\n", mybuf);
                  sleep(1);
            }
      }
      else {
            close(_____);
            for (i=0; i < 10; i++) {
                  printf("\t\t\t Child  Process..\n");
                  read(_____, mybuf, BUFSIZE);
                  printf("\t\t\t Child reads: %s\n", mybuf);
                  sleep(2);
            }
      }
}
```

Fill-up the blank spaces with appropriate expressions. Justify your answer. **[5]**

Question 5(c):