Reg.No: | | | | | | | |

SNS College of Technology, Coimbatore-35.
(Autonomous)
B.E/B.Tech- Internal Assessment -II
Academic Year 2023-2024(ODD)
Third Semester
Computer Science and Engineering

B

19ITT202 & Computer Organization and Architecture
[Common to CSE & IT]

Time: 1.5 Hours **Maximum Marks: 50**

Answer Key

PART - A (5x 2 = 10 Marks) — CO — Blooms

1. How does Computer architecture impact the speed of addition and Multiplication? CO2 Und

The performance improvement in arithmetic operations like addition, multiplication and division will increase the overall computational speed of the machine.

2. Why floating-point number is more difficult to represent and process than integer? CO2 Ana

This is a side effect of how the CPU represents floating point data. For this reason, you may experience some loss of precision, and some floating-point operations may produce unexpected result

3. List the algorithm for non-restoring division CO2 Rem

- Compare the divisor with the accumulated remainder.
- If the remainder is greater than or equal to the divisor, subtract the divisor from the remainder and set a "borrow" flag to 0.
- If the remainder is less than the divisor, set the borrow flag to 1.

4. What are the steps required for a pipelined processor to process the instruction? CO3 Rem

Fetching the instruction from memory.

Decoding the obtained instruction.

Calculating the effective address.

Fetching the operands from the given memory.

Execution of the instruction

5. How addressing modes affect the instruction pipelining? CO3 Und

Performance degradation in an instruction pipeline could be caused by address dependency, in which the operand address cannot be determined without the information required by the addressing mode

PART – B (13+13+14=40 Marks)

6. (a) Examine the two Signed numbers of 23 and -9 by using the Booth's multiplication algorithm 13 CO2 App

Here, M = 23 = (010111) and Q = -9 = (110111)

Q_n **Q_{n+1}**	**M = 0 1 0 1 1 1** **M' + 1 = 1 0 1 0 0 1**	**AC**	**Q**	**Q_{n+1}** **SC**
	Initially	000000	110111	0 6
1 0	Subtract M	101001		
		101001		
	Perform Arithmetic right shift operation	110100	111011	1 5
1 1	Perform Arithmetic right shift operation	111010	011101	1 4
1 1	Perform Arithmetic right shift operation	111101	001110	1 3
0 1	Addition (A + M)	010111		
		010100		
	Perform Arithmetic right shift operation	001010	000111	0 2
1 0	Subtract M	101001		
		110011		
	Perform Arithmetic right shift operation	111001	100011	1 1
1 1	Perform Arithmetic right shift operation	**111100**	**110001**	**1** 0

Q_{n+1} = 1, it means the output is negative.

Hence, 23 * -9 = 2's complement of 111100110001 => **(00001100111)**

(or)

(b) Perform Division Restoring Algorithm for the following inputs: 13 CO2 App

Dividend = 11 Divisor = 3

Solution:

Dividend = 11

Divisor = 3

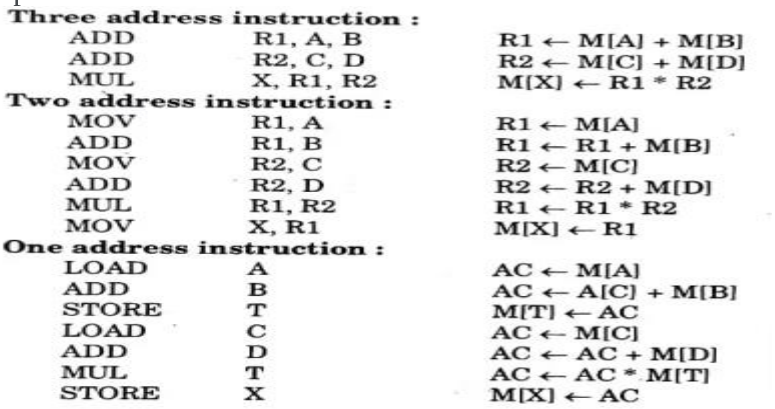
First the registers are initialized with corresponding values (Q = Dividend, M = Divisor, A = 0, n = number of bits in dividend)

n	M	A	Q	Operation
4	00011	00000	1011	initialize
4	00011	00001	011_	shift left AQ
	00011	11110	011_	A=A-M
	00011	00001	0110	Q[0]=0 And restore A
3	00011	00010	110_	shift left AQ
	00011	11111	110_	A=A-M
	00011	00010	1100	Q[0]=0 And restore A
2	00011	00101	100_	shift left AQ
	00011	00010	100_	A=A-M
	00011	00010	1001	Q[0]=1
1	00011	00101	001_	shift left AQ
	00011	00010	001_	A=A-M
	00011	00010	0011	Q[0]=1

register Q contain the quotient 3and register A contain remainder 2

7. (a) Analyze the purpose of the various elements of an instruction with the help of a sample instruction format. 13 CO3 Ana

1. Instruction is a command to the processor to perform a given task on specified data.



```
Three address instruction :
     ADD          R1, A, B          R1 ← M[A] + M[B]
     ADD          R2, C, D          R2 ← M[C] + M[D]
     MUL          X, R1, R2         M[X] ← R1 * R2
Two address instruction :
     MOV          R1, A             R1 ← M[A]
     ADD          R1, B             R1 ← R1 + M[B]
     MOV          R2, C             R2 ← M[C]
     ADD          R2, D             R2 ← R2 + M[D]
     MUL          R1, R2            R1 ← R1 * R2
     MOV          X, R1             M[X] ← R1
One address instruction :
     LOAD         A                 AC ← M[A]
     ADD          B                 AC ← A[C] + M[B]
     STORE        T                 M[T] ← AC
     LOAD         C                 AC ← M[C]
     ADD          D                 AC ← AC + M[D]
     MUL          T                 AC ← AC * M[T]
     STORE        X                 M[X] ← AC
```

Instruction cycle:

- 1. Instruction cycle is a complete process of instruction execution.

The instruction cycle is divided into three sub cycles:

- **Fetch cycle:** To fetch an opcode from a memory location following steps are performed:
- i. The address of the memory region where the opcode is stored is put on the address bus by the programme counter.

Decode cycle:

- i. The first thing to go into the Data Register (DR), or data/address buffer in the case of the Intel 8085, is the opcode that was just received from memory. After then, it is directed to the Instruction Register (IR).

Execute cycle:

- i. In this cycle, function of the instruction is performed.
- ii. If the instruction involves arithmetic or logic, ALU is utilized.

(or)

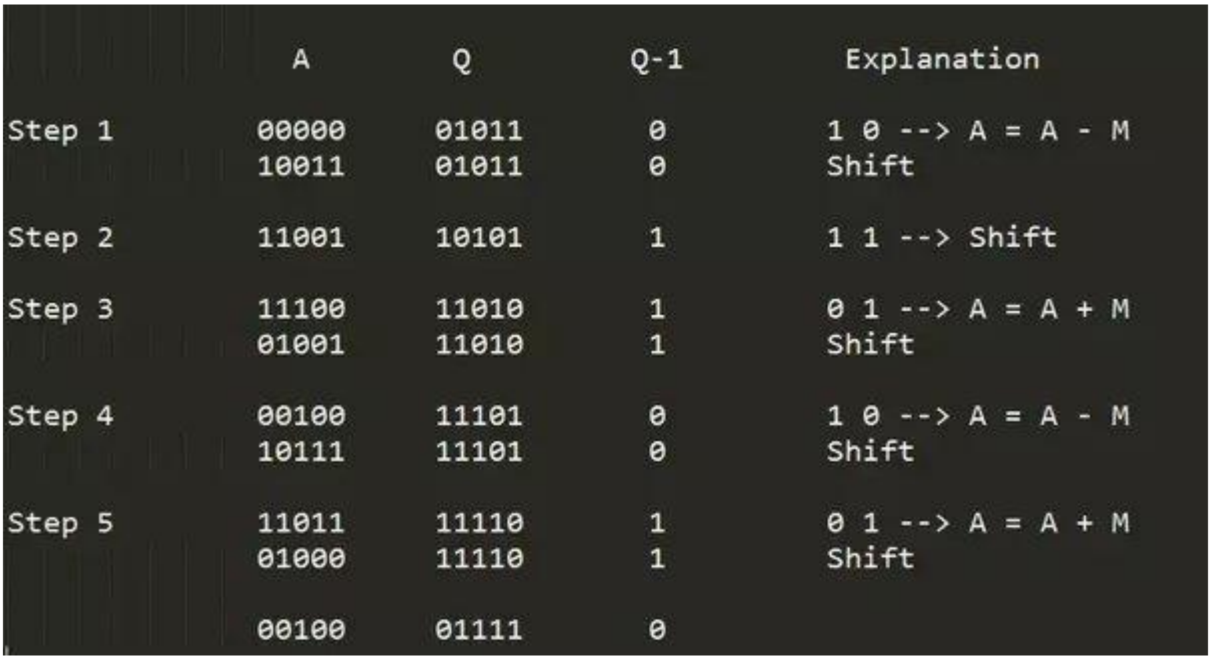
(b) Write the steps in fetching a word from memory. Differentiate between branch instructions and call subroutine instruction. 13 CO3 Und

Step 1: The CPU has to perform opcode fetch cycle and operand fetch cycle.
Step 2: The CPU receives the operation code necessary to retrieve a word from memory during the opcode fetch cycle.
Step 3: The operand fetch cycle is then started by the CPU.
Step 4: The address of the memory location where the data is saved is specified by the opcode.
Step5: The Address Register (AR), which is coupled to the address lines of the memory bus, receives the address of the required bit of information from the CPU. The address is therefore sent to the memory.
Step 6: When a read operation is required, the CPU triggers the memory's read signal.
Step 7: As a.result, memory copies data from the addressed register on the data bus.
Step 8: The CPU then loads the data into the designated register after reading it from the data register.
Step 9: For this memory transfer, the signal Memory Functions Completed (MFC) is also employed as a control signal.
Step10: MFC is set to 1 by memory to signify that the specified location's contents have been read and are now accessible via the data bus.

8. (a) Formulate binary Multiplication of Positive integer Numbers with Register Configuration diagram. 14 CO2 App

Multiplicand (M=13) : 1101 &

Multiplier (Q=11) : 1011



	A	Q	Q-1	Explanation
Step 1	00000	01011	0	1 0 --> A = A - M
	10011	01011	0	Shift
Step 2	11001	10101	1	1 1 --> Shift
Step 3	11100	11010	1	0 1 --> A = A + M
	01001	11010	1	Shift
Step 4	00100	11101	0	1 0 --> A = A - M
	10111	11101	0	Shift
Step 5	11011	11110	1	0 1 --> A = A + M
	01000	11110	1	Shift
	00100	01111	0	

(or)

(b) Evaluate the arithmetic statement X = (A +B)*(C + D) using a general register 1 CO3 App computer with three address, two address and one address instruction format a program to evaluate the expression. 4

Zero Address Instruction		
TOS: Top of the Stack		
PUSH	A	TOS ← A
PUSH	B	TOS ← B
ADD		TOS ← (A + B)
PUSH	C	TOS ← C
PUSH	D	TOS ← D
ADD		TOS ← (C + D)
MUL		TOS ← (C + D) ∗ (A + B)
POP	X	M [X] ← TOS
One Address Instruction		
LOAD	A	AC ← M [A]
ADD	B	AC ← A [C] + M [B]
STORE	T	M [T] ← AC
LOAD	C	AC ← M [C]
ADD	D	AC ← AC + M [D]
MUL	T	AC ← AC ∗ M [T]
STORE	X	M [X] ← AC
Two Address Instructions		
MOV	R1, A	R1 ← M [A]
ADD	R1, B	R1 ← R1 + M [B]
MOV	R2, C	R2 ← M [C]
ADD	R2, D	R2 ← R2 + M [D]
MUL	R1, R2	R1 ← R1∗R2
MOV	X, R1	M [X] ← R1
Three Address Instruction		
ADD	R1, A, B	R1 ← M [A] + M [B]
ADD	R2, C, D	R2 ← M [C] + M [D]
MUL	X, R1, R2	M [X] ← R1 ∗ R2

****

(Note: Und-Understand Rem-Remember Ana-Analyze App-Apply Cre- Create)

Prepared by Verified by HoD