**Q.1** Consider the following instructions:

I1: MOVE R1, #80

I2: MOVE R2, #40

I3: CMP R1, R2

I4: HLT

After the execution, what will be status of Z and N flags?

(a) Z=1; N=1

(b) Z=1; N=0

(c ) Z=0; N=1

(d) Z=1; N=1

**Q.2.** Consider the following expression used to execute on ACCUMULATOR CPU and STACK CPU

X= (A+B\*C) / (D-E\*F+G\*H)

where A, B, C, D, E, F, G, H and X are the memory addresses. How many number of instructions are required to execute the expression on the respective CPUs?

(a) STACK CPU = 15, ACC CPU= 16

(b) STACK CPU = 16, ACC CPU= 15

(c) STACK CPU = 17, ACC CPU= 16

(d) STACK CPU = 16, ACC CPU= 14

**Q.3.** Let A be an array of 10 integers. The base address of A, 0x00000000, is stored in R1. The values of R4 and R2 are 10 and 21, respectively. The following C language program segment

for(i=0; i<10; i++)

A[i] = 21+ A[i];

can be implemented in assembly code as

I1: loop: LOAD R3, 0(R1)

I2 ADD R3, R2, R3

I3 ---------?

I4 ----------?

I5 BNE R1, R4, loop

What would the possible instructions I3 and I4?

(a) I3: LOAD R3, 0(R1), I4 : ADD R1, R1, @4

(b) I3: STORE R3, 0(R1), I4 : ADD R1, R1, #4

(c) I3: LOAD R3, 0(R1), I4 : ADDI R1, R2, 4

(d) I3: STORE R4, 0(R1), I4 : ADD R1, R2, @4

**Q4.** Consider a hypothetical CPU that supports 32-bits instruction. Suppose that the current value of PC is 100. Assume that the instruction I2 tells the CPU to call a subroutine A which starts at the address 200 and finishes at 208 as depicted below.

|  |  |
| --- | --- |
| Instruction (Memory) | Address |
| I1 | 100 |
| I2: CAL A | 104 |
| I3 | 108 |
|  |  |
| SUBROUTINE A: J1 | 200 |
| J2 | 204 |
| J3 (RET) | 208 |
|  |  |
|  |  |

How the value of PC and Link Register should change throughout this process?

(a) PC: 100-104-108-200-204-208 : LR: 108

(b) PC: 100-104-108-200-204-108 : LR: 200

(c ) PC: 100-104-200-204-208-108 : LR: 108

(d) PC: 100-104-108-200-204-208 : LR: 200

**Q.5.** Consider an instruction ADD @500, (200), total memory references to execute the instruction?

(a) 4

(b) 5

(c) 6

(d) 2

**Q.6.** A 4Byte long PC relative instruction is located in the memory with starting address 283050 (decimal). A “-8” signed displacement is present in the address field of the instruction, what is the branch / target /effective address?

(a) 283046

(b) 283042

(c) 283058

(d) 283062

**Q.7.** Consider the following instructions:

MOV R1, A

ADD R1, B

what is the content of R1 register after execution of the program segment?

(a) B

(b) A+B

(c) A

(d) 0

**Q.8.** Consider a computer that has a byte-addressable memory organized in 32-bit words. In this organization, registers R1 and R2 currently hold the decimal numbers 1000 and 3000 respectively before the addressing mode is used to access a memory operand. What is the effective address (EA) for the instruction LOAD R3, 25(R1, R2) ?

(a) 3025

(b) 1025

(c) 4000

(d) 4025

**Q.9.** Consider the following program which calls a subroutine S1. To call a subroutine the MSF (Mark Stack Frame) is required to store the return address of the calling program.

I1: MOV #0, R01

I2: S0: ADD #1, R01

I3:

I4: CAL S1

I5:

I6: CMP #5, R01

I7: JNE S0

I8: HLT

I9:

I10: S1:

I11:

I12: OUT 24, 0

I13: RET

What would be the correct position of MSF in the given program?

(a) I3

(b) I5

(c) I9

(d) I11

**Q.10.** Which of the following instruction stores a decimal number 10 in memory location whose address is currently stored in register R5?

(a) STW #10, #R5

(b) STW @10, R5

(c) STB @10, R5,

(d) STB #10, @R5

**Q11.** Consider a string of ASCII-encoded characters stored in memory starting at address START. The string ends with the NULL character. A partial possible RISC-style program is given to determine the length of the string and store it in location FINAL.

I1: Move R1, #START

I2: Clear R2

I3: Move R3, #NULL

I4: LDB R4, (R1)

I5: ? R4, R3 FINISH

I6: Add R1, R1, #1

I7: Add R2, R2, #1

I8: Branch LOOP

FINISH I9: Store R2, FINAL

In instruction I8, where the Label LOOP will be targeted and what would be the possible opcode in Instruction I5?

(a) I3 & BEQ

(b) I4 & BEQ

(c) I4 & BGT

(d) I3 & BNQ