|  |
| --- |
| **EE-2003: Computer Organization & Assembly Language** |
| Date: 4th April, 2024 |
| **Course Instructor(s)** |
| Mehreen Javaid, Ahmad Raza, Shahbaz Hassan |
| **Sessional-II Exam** |
| **Total Time: 1 Hour** |
| **Total Marks: 50** |
|  |

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Student Name Roll No. Course Section Student Signature |

**Do not write anything on the question paper except the information required above.**

**Instructions:**

1. Read the question carefully, understand the question, and then attempt your answers in the provided answer booklet.
2. Verify that you have **3** printed pages of the question paper including this page. There are **Four (4)** questions.
3. Calculator sharing is strictly prohibited.
4. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking

**Question 1 [10 Marks]**

Carefully dry run the given program. Show complete traces of runtime stack for **both** fill and clear phases by writing the actual offset and values as provided. Stack segment starts at 1000h. Write the final values of register at the end of program and also where asked in between the code in the form of comments. Suppose data segment starts at **000Fh** offset. ***(No Need to Write Code Just Fill the Stack on Provided Answer Sheet)***.

|  |  |
| --- | --- |
|  | .model small |
|  | .data |
|  | ary db 25, 35, 52, 49, 53 |
|  | .code |
|  | mov ax, @data |
|  | mov ds, ax |
|  | jmp start |
|  | **MySub PROC uses dx cx** |
|  | push bp |
|  | mov bp,sp |
|  | sub sp, 4 |
|  | mov word ptr [bp-4], 59 |
|  | mov word PTR [bp-2], 57 |
|  | mov dX, [bp-2] ;dX= |
|  | mov dx, [bp+4] ;dx= |
|  | mov bx, [bp+10] ;bx= |
|  | mov dx, [bx] ;dx= |
|  | mov sp, bp |
|  | pop bp |
|  | ret 2 |
|  | MySub ENDP |
|  | start: |
|  | **main proc** |
|  | mov si, offset ary |
|  | mov word ptr [si], 45h |
|  | add si, 2 |
|  | mov dx, [si-2] ;dx= |
|  | mov dx, [si] ;dx= |
|  | mov di, offset ary |
|  | Add di,3 |
|  | push di |
|  | mov cx, 49 |
|  | push cx |
|  | call MySub |
|  | inc cx |
|  | push cx |
|  | push dx |
|  | call AddTwo |
|  | mov dx, ax ;dx= |
|  | main endp |
|  | mov ah, 4ch |
|  | int 21h |
|  | End main |
|  | **AddTwo Proc** |
|  | push bp |
|  | mov bp, sp |
|  | mov ax, [bp+6] ;AX= |
|  | mov si, [bp+8] |
|  | add ax,[si] ;AX= |
|  | pop bp |
|  | ret |
|  | AddTwo endp |

**Question 2 [5 + 5 + 5 = 15 Marks]**

**[a].** Write an assembly code **to multiply [Ax \* 42]** using Shift and rotate instructions. The value of **Accumulator register (Ax = 04).**

mov ax,04

mov bx,ax

mov cx,ax

shl ax,5

shl bx,3

shl cx,1

add ax,bx

add ax,cx

**[b].** Write down the **values of each register** at each step. (0.5 mark for each correct value)

Mov ax, -96 ;ax= FFA0 1111111110100000

Mov cx, 2 ;cx= 0002 0000000000000010

Shl al, 2 ;ax= FF80 1111111110000000

Shr ah, 3 ;ax= 1F80 0001111110000000

Mov dx, ax ;dx= 1F80 0001111110000000

Sal dx, cl ;dx= 7E00 0111111000000000

Sar Ax, 1 ;ax= 0FC0 0000111111000000

rcr dl, 1 ;dl= 00 00000000

Shl dx, 1 ;dh= FC 11111100

Rol Ax, 1 ;ax= 1F80 0001111110000000

**[c].** Perform the **1-bit logical left shift operation** on following **Qword Number**. You are required to write an **assembly language code**.

.data

**Var1 dq 45F37C4A556DE13h**

**.code**

**mov si,offset var1 ;1 marks**

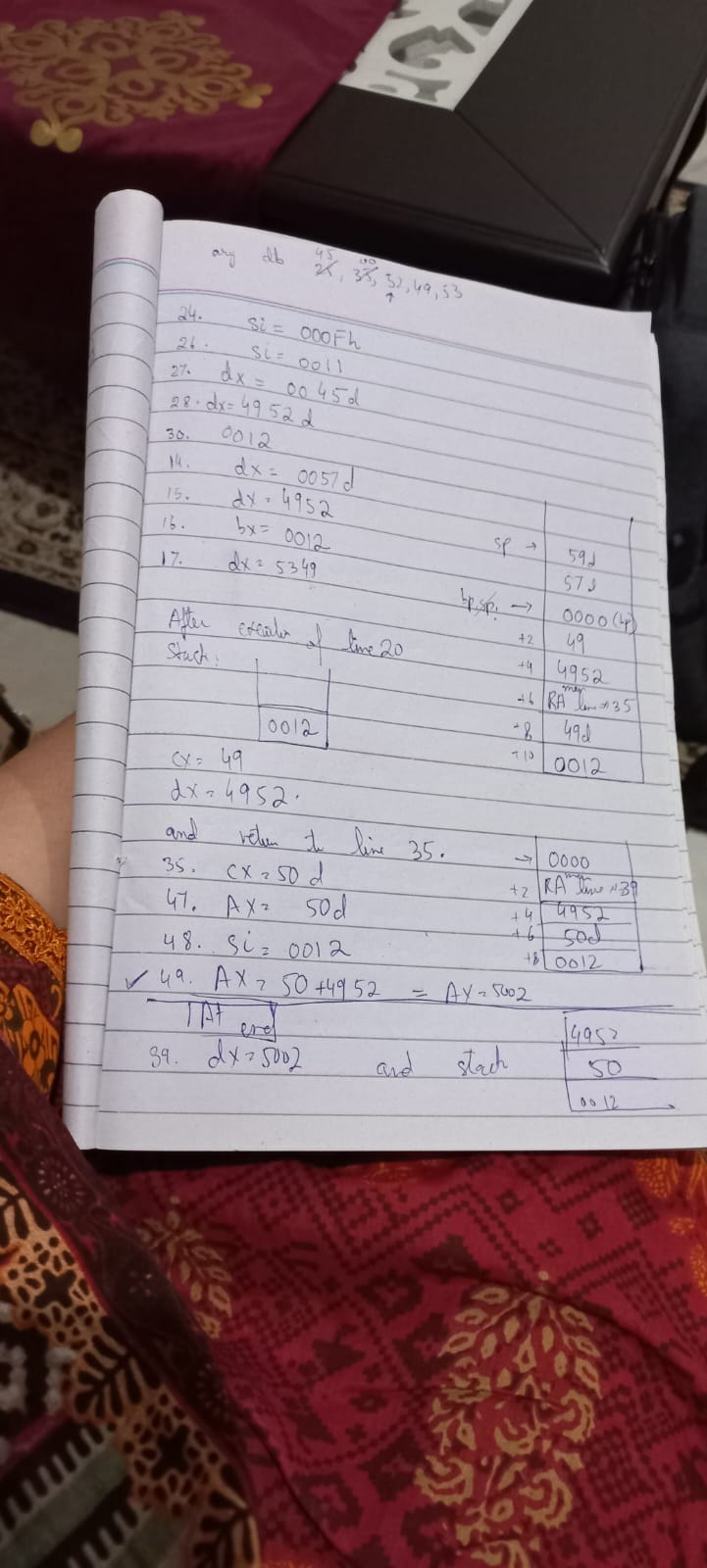
**shl word ptr [si],1 ;1 marks**

**rcl word ptr [si+2],1 ;1 marks**

**rcl word ptr [si+4],1 ;1 marks**

**rcl word ptr [si+6],1 ;1 marks**

**Question 1 solution**

****

**Question 3 [5 Marks]**

**Write the following code on provided answer sheet and clearly illustrate which jumps will be taken or not taken in front of each line of code. Also,** update the flag registers value after execution of the code.

mov al, 50

cmp al, -30

ja L1 ;NOT TAKEN

L2: jnl L3 ;TAKEN

L1: jg L2 ;NOT TAKEN

L3: jnge L4 ;NOT TAKEN

L4: jbe LM ;TAKEN

LM: mov bl, -15

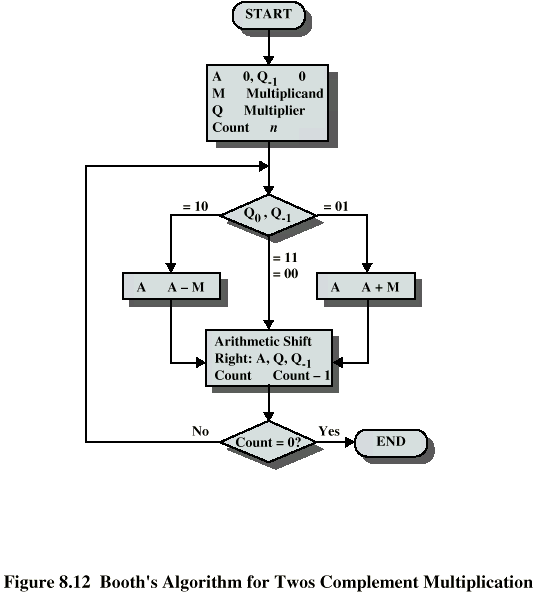
;FLAGS

CF: 1 ZF: 0 SF: 0

OF: 0 PF: 1 AF: 0

**Question 4 [10 Marks]**

Use Booth’s Multiplication Algorithm to **multiply 8 (take it as Q)** by **-28 (take it as M).** Show all steps considering your computer’s **data width equal to 6-bits.**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operation | A | Q | Q-1 | M |
| n=6 | 000000 | 001000 | 0 | 100100 |
| SAR=5 | 000000 | 000100 | 0 | 100100 |
| SAR=4 | 000000 | 000010 | 0 | 100100 |
| SAR=3 | 000000 | 000001 | 0 | 100100 |
| A=A-M  SAR=2 | 011100  001110 | 000001  000000 | 0  1 | 100100  100100 |
| A=A-M  SAR=1 | 110010  111001 | 000000  000000 | 1  0 | 100100  100100 |
| SAR=0 | 111100 | 100000 | 0 | 100100 |

Answer = 111100100000

2’s complement = 000011100000

Decimal = -28x8 = -224