National University of Computer and Emerging Sciences 

**Laboratory Manual**

*for*

**Computer Organization and Assembly Language Programming**

| Course Instructor | Aleena Ahmad |
| --- | --- |
| Lab Instructor | Sana Ejaz |
| Semester | Fall 2024 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

Page 1

**OBJECTIVES:**

∙ How to perform bit operations.

∙ How to swap alternate bits.

∙ How to create a basic stack function.

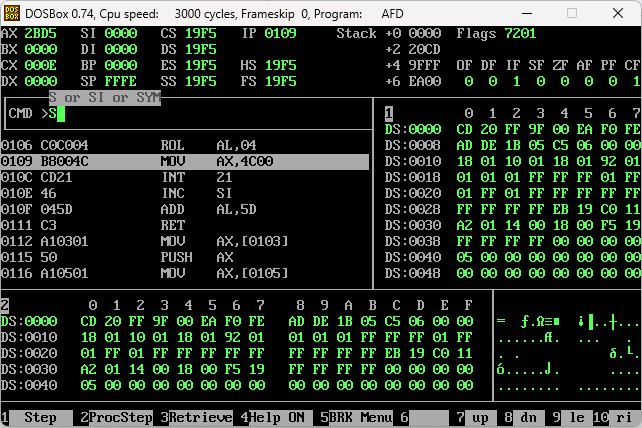
**Instructions:**

| 1. Run and debug the programs, ensuring that they behave as expected.  2. Document your observations and note any issues encountered during implementation in a Word document.  3. Submit work in a single Word file with screenshots. No asm, lst , or com. (Do not submit a zip folder) |
| --- |

**Task 1: Write a program to swap the nibbles (4-bits = 1 nibble) in each byte of the AX register.**

**Sample:**

| AX before Swap | **1011 0010 0101 1101** | **0xB25D** |
| --- | --- | --- |
| AX after Swap | **0010 1011 1101 0101** | **0x2BD5** |

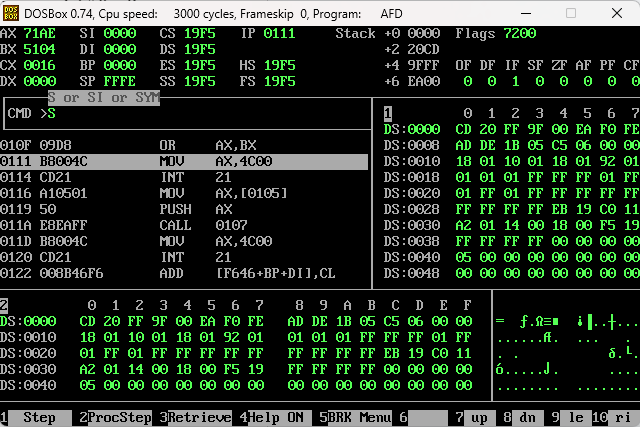


| [org 0x100]  MOV AX, 0xB25D  ROL AH, 4  ROL AL, 4  MOV AX, 0x4C00  INT 0x21 |
| --- |

**Task 2: Write a program to swap every pair of bits in the AX register i.e. swap bit # 0 with bit # 1, bit # 2 with bit # 3 and so on.**

**Sample:**

| AX before Swap | **10 11 00 10 01 01 11 01** |
| --- | --- |
| AX after Swap | **01 11 00 01 10 10 11 10** |



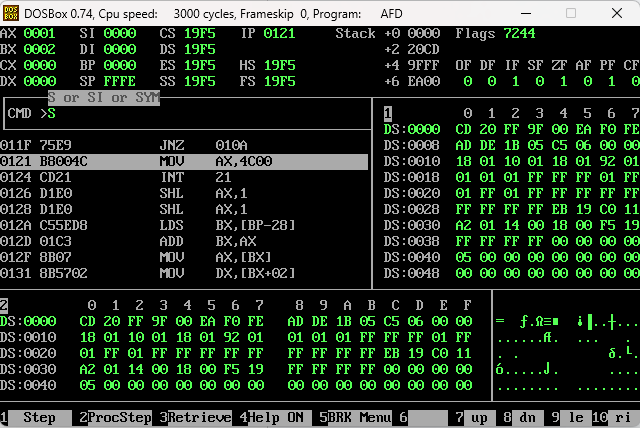
| [org 0x0100]  MOV AX, 0xB25D ; 1011 0010 0101 1101  MOV BX, 0xAAAA ; 1010 1010 1010 1010  AND BX, AX ; 1010 0010 0000 1000  AND AX, 0x5555 ; 0001 0000 0101 0101  SHL AX, 1 ; 0010 0000 1010 1010  SHR BX, 1 ; 0101 0001 0000 0100  OR AX, BX ; 0111 0001 1010 1110  MOV AX, 0x4C00  INT 0x21 |
| --- |

**Task 3: AX contains a non-zero number. Count the number of ones in it and store the result back in AX. Repeat the process on the result (AX) until AX contains one. Calculate in BX the number of iterations it took to make AX one. For example BX should contain 2 in the following case:**

AX = 1100 0101 1010 0011 (input – 8 ones)

AX = 0000 0000 0000 1000 (after first iteration – 1 one)

AX = 0000 0000 0000 0001 (after second iteration – 1 one) STOP



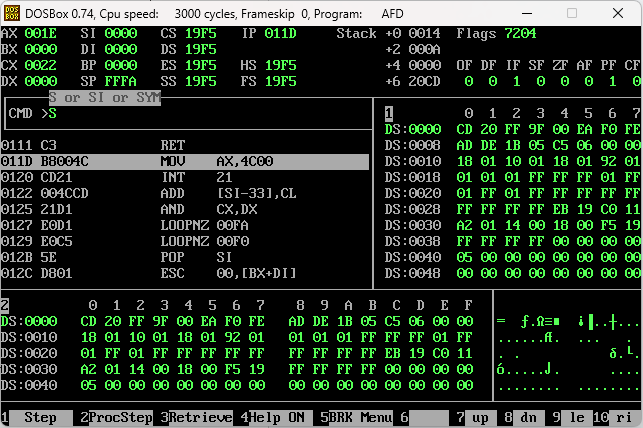
| [org 0x0100]  MOV AX, 0xC5A3  MOV BX, 0  MOV CX, 0  ADD CX, 0  checkbit:  ADC CX, 0  SHR AX, 1  JNZ checkbit  ADC CX, 0  INC BX  MOV AX, CX  MOV CX, 0    CMP AX, 1  JNZ checkbit  MOV AX, 0x4C00  INT 0x21 |
| --- |

**Homework (Submit before 12:00a.m)**

**Bonus 2 marks if done in class after completing tasks 1-3**

**Task 4: Write a program that defines a basic stack function to calculate the sum of two numbers. The function should accept two integers as parameters, passed through the stack OR use hardcoded values in registers. The function should return the sum in AX. Use stack pointer registers, subroutines and push/pop to perform addition.**

| [org 0x0100]  JMP start  num1: dw 10  num2: dw 20  ad:  PUSH BP  MOV BP, SP  MOV AX, [BP + 6]  ADD AX, [BP + 4]  POP BP  RET  start:  MOV AX, [num1]  PUSH AX  MOV AX, [num2]  PUSH AX  CALL ad  MOV AX, 0x4C00  INT 0x21 |
| --- |

****