



BAŞKENT UNIVERSITY

ENGINEERING FACULTY

**ELECTRICAL-ELECTRONICS
ENGINEERING DEPARTMENT**

EEM 322 – MICROPROCESSORS LAB

EXPERIMENT NO. 03:

ARITHMETIC AND LOGIC OPERATIONS

SAMET BAYAT

22293730

1. Set the value of the AX register to 0056₁₆ and the value of the BX register to 024C₁₆. Have the contents of the AX and BX registers changed with a single command line. Take a screenshot of the results and put it in your report.

```
===== Happy gaming! =====
C: is mounted as local directory /Users/sametbayat/Desktop/Dev/322/
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0100  NV UP EI PL NZ NA PO NC
0745:0100 0000          ADD     [BX+SI],AL          DS:0000=CD
-
-A 100
0745:0100 MOV AX, 56
0745:0103 MOV BX, 24C
0745:0106 XCHG AX,BX
0745:0108
-
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0100  NV UP EI PL NZ NA PO NC
0745:0100 B85600          MOV     AX,0056
-
-T
AX=0056 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0103  NV UP EI PL NZ NA PO NC
0745:0103 BB4C02          MOV     BX,024C
-
-T
AX=0056 BX=024C CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0106  NV UP EI PL NZ NA PO NC
0745:0106 87C3          XCHG     AX,BX
-
-T
AX=024C BX=0056 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0108  NV UP EI PL NZ NA PO NC
0745:0108 3C6C          CMP      AL,6C
-
-
```

The XCHG (Exchange) instruction is used to exchange the contents of two operands. It takes two operands, the first operand is a register or a memory location, and the second operand is also a register or a memory location.

SINTAX: XCHG DESTINATION, SOURCE

2. Load the value 0098_{16} into the BX register. Write and run the command that takes the two's complement of the BX register. Take a screenshot of the results and put it in your report.

```
-A 108
0745:0108 MOV BX, 98
0745:010B NEG BX
0745:010D
-
-R
AX=024C BX=0056 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0108  NV UP EI PL NZ NA PO NC
0745:0108 BB9800          MOV     BX,0098
-
-T
AX=024C BX=0098 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=010B  NV UP EI PL NZ NA PO NC
0745:010B F7DB          NEG     BX
-
-T
AX=024C BX=FF68 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=010D  NV UP EI NG NZ AC PO CY
0745:010D 0000          ADD     [BX+SI],AL          DS:FF68=00
-
-
```

The NEG (Negate) instruction is used to obtain the two's complement of the specified operand. It takes one operand, which can be a register or a memory location.

During execution, the NEG instruction performs the following steps:

- Negates the specified operand by subtracting it from zero.
- Sets the flags according to the result of the operation.

The result of the NEG instruction is stored in the same location as the operand.

SINTAX: NEG OPRERAND

3. Load the value 0098_{16} into the BX register. Type and run the command that takes the two's complement of BX according to the clue below. Take a screenshot of the results and put it in your report.

Hint: As you may remember from the EEM211 Numerical Logic course, taking the 2's complement of a number is found by taking the 1's complement of the same number and adding 1 to the 1's complement value.

```
-
-A 10D
0745:010D MOV BX, 98
0745:0110 NOT BX
0745:0112 ADD BX, 1
0745:0115
-
-R
AX=024C BX=FF68 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=010D  NV UP EI NG NZ AC PO CY
0745:010D BB9800      MOV     BX,0098
-
-T
AX=024C BX=0098 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0110  NV UP EI NG NZ AC PO CY
0745:0110 F7D3      NOT     BX
-
-T
AX=024C BX=FF67 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0112  NV UP EI NG NZ AC PO CY
0745:0112 83C301    ADD     BX,+01
-
-T
AX=024C BX=FF68 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0115  NV UP EI NG NZ NA PO NC
0745:0115 0000      ADD     [BX+SI],AL      DS:FF68=00
-
```

Here, we performed the same operation on Q3 except, as mentioned on hint, we did it in 2 steps:

- 1st complemen
- BX ++

After all steps we see that carry and auxiliary flag became 0.

4. Set the value of the AX register to 0147₁₆ and the value of the BX register to 0057₁₆. Set the value of the Carry Flag to 0. Run the ADC AX, BX code and observe the result. Take a screenshot of the results and put it in your report.

Hint: You need to set the CF value to 0 before running the ADC command. For this, you need to use the flag changing method we did in Experiment 2 and enter the value required for CF=0 in the DEBUG INTRO file.

```
-A 115
0745:0115 MOV AX, 147
0745:0118 MOV BX, 57
0745:011B CLC
0745:011C ADC AX, BX
0745:011E
-
-R
AX=024C BX=FF68 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0115  NV UP EI NG NZ NA PO NC
0745:0115 B84701      MOV     AX,0147
-
-T
AX=0147 BX=FF68 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0118  NV UP EI NG NZ NA PO NC
0745:0118 BB5700      MOV     BX,0057
-
-T
AX=0147 BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=011B  NV UP EI NG NZ NA PO NC
0745:011B F8          CLC
-
-T
AX=0147 BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=011C  NV UP EI NG NZ NA PO NC
0745:011C 11D8        ADC     AX,BX
-
-T
AX=019E BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=011E  NV UP EI PL NZ NA PO NC
0745:011E 3407        XOR     AL,07
-
```

NG and PL are two of the condition codes in the flags, which are used to indicate the sign of the result of an arithmetic or logic operation.

NG (Negative) flag (MSB) of the result is 1.

PL (Positive or Zero) (MSB) of the result is 0.

ADC (Add with Carry) is an instruction that performs an addition operation between two operands with an additional carry input.

SINTAX: ADC DESTINATION, SOURCE

First, we clear carry flag, after operation carry flag not effected and MSB became 1.

5. Set the value of the AX register to 0147₁₆ and the value of the BX register to 0057₁₆. Set the value of the Carry Flag to 1. Run the ADC AX, BX code and observe the result. Take a screenshot of the results and put it in your report.

Hint: You need to set the CF value to 1 before running the ADC command. For this, you need to use the flag changing method we did in Experiment 2 and enter the value required for CF=1 in the DEBUG INTRO file

```
-A 11E
0745:011E MOV AX, 147
0745:0121 MOV BX, 57
0745:0124 STC
0745:0125 ADC AX,BX
0745:0127

-R
AX=019E BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=011E  NV UP EI PL NZ NA PO NC
0745:011E B84701      MOV     AX,0147

-T

AX=0147 BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0121  NV UP EI PL NZ NA PO NC
0745:0121 BB5700      MOV     BX,0057

-T

AX=0147 BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0124  NV UP EI PL NZ NA PO NC
0745:0124 F9          STC

-T

AX=0147 BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0125  NV UP EI PL NZ NA PO CY
0745:0125 11DB      ADC     AX,BX

-T

AX=019F BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0127  NV UP EI PL NZ NA PE NC
0745:0127 0000      ADD     [BX+SI],AL          DS:0057=00
```

First, we set carry flag as 1, which can be observed in second to last block (CY). After running all statements, we get a result that +1 of previous one. The reason of that difference is carry flag. Also there is change in PO-PE.

PO (Pop) and PE (Pop Extop) are two instructions in the 8086 processor that are used to retrieve data from the stack.

6. Use the MOV command ONLY as you did in the first question. Use the CX register as the buffer register. Take a screenshot of the results and put it in your report.

```
-A 127
0745:0127 MOV AX, 56
0745:012A MOV BX, 24C
0745:012D MOV CX, AX
0745:012F MOV AX, BX
0745:0131 MOV BX, CX
0745:0133

-R
AX=019F BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0127  NU UP EI PL NZ NA PE NC
0745:0127 B85600          MOV     AX,0056

-T
AX=0056 BX=0057 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=012A  NU UP EI PL NZ NA PE NC
0745:012A BB4C02          MOV     BX,024C

-T
AX=0056 BX=024C CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=012D  NU UP EI PL NZ NA PE NC
0745:012D 89C1           MOV     CX,AX

-T
AX=0056 BX=024C CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=012F  NU UP EI PL NZ NA PE NC
0745:012F 89D8           MOV     AX,BX

-T
AX=024C BX=024C CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0131  NU UP EI PL NZ NA PE NC
0745:0131 89CB           MOV     BX,CX

-T
AX=024C BX=0056 CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0133  NU UP EI PL NZ NA PE NC
0745:0133 0CC6           OR      AL,C6
```

This code first stores the contents of the AX register in the CX register, then moves the contents of the BX register to the AX register, and finally moves the original contents of the AX register (now stored in CX) to the BX register. This effectively exchanges the contents of the AX and BX registers with using only MOV instruction.

7. Set the value of the AX register to 00A3₁₆ and the value of the BX register to 002C₁₆. Perform the AX=AX-BX operation. Examine AX, BX, Carry Flag, Zero Flag and Overflow Flag. Take a screenshot of the results and put it in your report.

```
-A 133
0745:0133 MOV AX, A3
0745:0136 MOV BX, 2C
0745:0139 SUB AX, BX
0745:013B

-R
AX=024C BX=0056 CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0133 NV UP EI PL NZ NA PE NC
0745:0133 B8A300 MOV AX,00A3

-T
AX=00A3 BX=0056 CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0136 NV UP EI PL NZ NA PE NC
0745:0136 BB2C00 MOV BX,002C

-T
AX=00A3 BX=002C CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0139 NV UP EI PL NZ NA PE NC
0745:0139 29D8 SUB AX,BX

-T
AX=0077 BX=002C CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=013B NV UP EI PL NZ AC PE NC
0745:013B 0000 ADD [BX+SI],AL DS:002C=00

-
```

As we guess, if we subtract smaller number from greater number (which are positive in our circumstance) MSB must be 0 (PL). Because of the result was not effected any sign operation we got our result in AX easily in HEX :
(A3 - 2C = 77)

In addition to this, auxiliary flag became 1 (there was carry transferred from d3 to d4). BX, overflow, carry and zero flag kept their values.

8. Set the value of the AX register to 00A3₁₆ and the value of the BX register to 002C₁₆. Perform the BX=BX-AX operation. Examine AX, BX, Carry flag, Zero Flag and Overflow Flag. Take a screenshot of the results and put it in your report.

```

-A 13B
0745:013B MOV AX, A3
0745:013E MOV BX, 2C
0745:0141 SUB BX, AX
0745:0143
-
-R
AX=0077 BX=002C CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=013B  NV UP EI PL NZ AC PE NC
0745:013B B8A300      MOV     AX,00A3
-
-T
AX=00A3 BX=002C CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=013E  NV UP EI PL NZ AC PE NC
0745:013E B2C00      MOV     BX,002C
-
-T
AX=00A3 BX=002C CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0141  NV UP EI PL NZ AC PE NC
0745:0141 29C3      SUB     BX,AX
-
-T
AX=00A3 BX=FF89 CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0143  NV UP EI NG NZ NA PO CY
0745:0143 0000      ADD     [BX+SI],AL      DS:FF89=00
-
-

```

Here we subtract smaller number from greater number. Therefore, PL-positive flag became 0. Auxiliary flag became 0 and carry occurred (CY). AX kept its value. The MOV instructions would load the values A3 (163 in decimal) into the AX register and 2C (44 in decimal) into the BX register, and the SUB instruction would subtract the value in AX (163) from the value in BX (44), storing the result in BX.

BX = BX - AX

BX = 44 - 163 (in decimal)

**BX = -119 (in decimal), or 0xFF89 in hexadecimal
(using two's complement representation)**

9. Set the value of the AX register to $82CD_{16}$ and the value of the BX register to $82CD_{16}$. Write the command that performs $AX=AX-BX$ and observe the values of AX, BX, Carry flag, Zero Flag and Overflow Flag. Take a screenshot of the results and put it in your report.

```
-A 143
0745:0143 MOV AX, 82CD
0745:0146 MOV BX, 82CD
0745:0149 SUB AX, BX
0745:014B

-R
AX=00A3 BX=FF89 CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0143 NV UP EI NG NZ NA PO CY
0745:0143 B8CD82 MOV AX,82CD

-T
AX=82CD BX=FF89 CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0146 NV UP EI NG NZ NA PO CY
0745:0146 BBCD82 MOV BX,82CD

-T
AX=82CD BX=82CD CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0149 NV UP EI NG NZ NA PO CY
0745:0149 29DB SUB AX,BX

-T
AX=0000 BX=82CD CX=0056 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=014B NV UP EI PL ZR NA PE NC
0745:014B 0000 ADD [BX+SI],AL DS:82CD=42
```

Here we subtract two identical number from each other. In that case, result became 0 and zero flag became 1. No overflow, carry occurred.

10. Set the value of the AX register to 82CD₁₆ and the value of the BX register to 82CD₁₆. Write the command that compares AX and BX and observe the values of AX, BX, ZF Carry flag, Zero Flag and Overflow Flag. Take a screenshot of the results and put it in your report.

(Because of crash I had to refresh the session. To observe the wanted flags I also changed CF as 1 (CY).)

```

Happy gaming!
C: is mounted as local directory /Users/samethbayat/Desktop/Dev/322/
-
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0100 NU UP EI PL NZ NA PO NC
0745:0100 0000 ADD [BX+SI],AL DS:0000=CD
-
-A 100
0745:0100 STC
0745:0101
-
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0100 NU UP EI PL NZ NA PO NC
0745:0100 F9 STC
-
-T
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0101 NU UP EI PL NZ NA PO CY
0745:0101 0000 ADD [BX+SI],AL DS:0000=CD
-

```

```

0745:0101 0000 ADD [BX+SI],AL DS:0000=CD
-
-A 101
0745:0101 MOV AX, 82CD
0745:0104 MOV BX, 82CD
0745:0107 CMP AX, BX
0745:0109
-
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0101 NU UP EI PL NZ NA PO CY
0745:0101 B8CD82 MOV AX,82CD
-
-T
AX=82CD BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0104 NU UP EI PL NZ NA PO CY
0745:0104 B8CD82 MOV BX,82CD
-
-T
AX=82CD BX=82CD CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0107 NU UP EI PL NZ NA PO CY
0745:0107 39D8 CMP AX,BX
-
-T
AX=82CD BX=82CD CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0745 ES=0745 SS=0745 CS=0745 IP=0109 NU UP EI PL ZR NA PE NC
0745:0109 6C DB 6C
-

```

Again, we set AX and BX as same numbers and with compare command, we investigate whether the two numbers in these registers are equal or not. The last block shows that the zero flag became 1 and carry flag became 0. That means these two numbers are identical. In addition to this, no overflow observed and values of AX-BX registers are not changed, because the "compare" method deals with flags, it just effects the value of flags.