# Comp 3350: Computer Organization & Assembly Language

# HW # 8. Theme: Integer Arithmetic

*All main questions carry equal weight.*

*Points will be awarded to only those answers which have work clearly shown*

1. In the following code sequence, show the value of AL after each shift or rotate instruction has executed. This question is to be done by hand, not by running a program.

mov cl, 1

mov al, 12h ; al= 00010010b

rol al, cl ; al= 00100100b = 24h

mov al, 34h ; al= 00110100b = 34h

mov cl, 2

ror al, Cl ; al= 00001101b = 0Dh

stc

mov al, 56h ; al= 01010110b = 56h

mov cl, 3

rcl al, cl ; al= 10110101b = B5h

stc

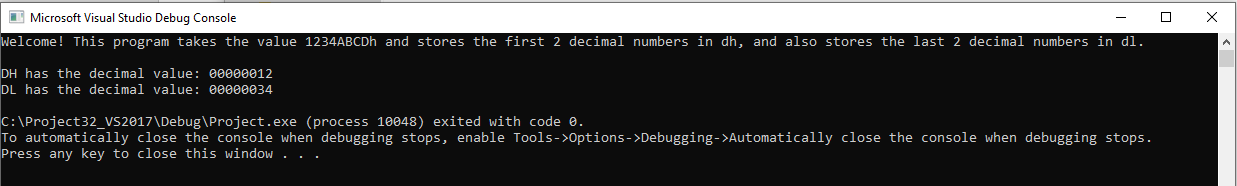
mov al, 78h ; al= 01111000b = 78h

mov cl, 1

rcr al, cl ; al= 10111100 = BCh

1. (a) Write a program which calculates EAX\**2810* using binary multiplication. (Only typewritten code required)

(b) Consider the following unsigned value: 1234ABCD. Let this value be stored in register EAX. Write a program that will extract the decimal digits from this value using shifts and logical instructions. Place the first two **decimal numeric** digits in DH and the other two into DL. Submit a screenshot of **the console output** of the program and the **asm/lst file**.



1. (a) What will be the contents of AX and DX after the following operation? What may happen if you do not set dx to 0 in the beginning? You must work this problem by hand, not by a program run.

mov dx, 0

mov ax, 1111h

mov cx, 2222h

mul cx ; AX= 8642, DX = 0246

**Explanation:** When multiplying a 16 bit, the higher order gets stored in DX. If DX was not set to 0 before the operation occurred, then the value that gets stored into DX could be incorrect.

1. When does an IDIV instruction cause an overflow? Provide an example.

**Answer:** An overflow is caused by a quotient produced by the instruction that is too large (or too small if working with signed integers) for the destination operand. When an overflow happens, the program abruptly stops and an error code is provided.

**Example:** mov ax, 700

mov bl, 2

idiv bl ; 350 cannot be stored in AL – overflow error

1. What will be the values of DX:AX after the following instructions execute? What might be the use of such a sequence of instructions in a 16-bit computer?

mov ax, 0h

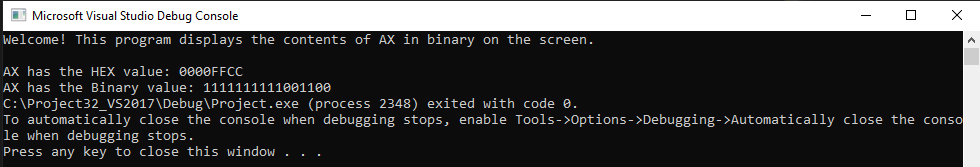
mov dx, 0h

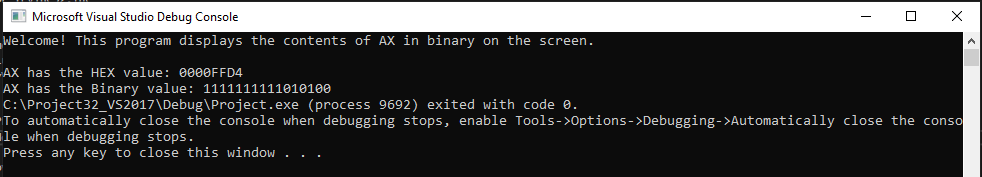
sub ax, 2h ; AX = FFFEh

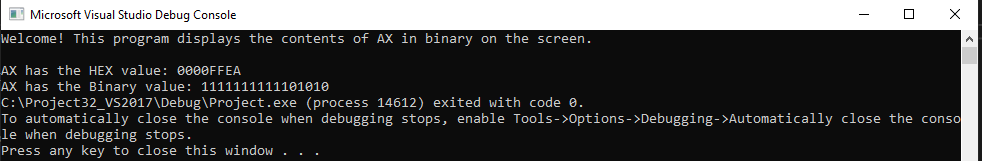
sbb dx, 0 ; DX = FFFFh

**Explanation:** These instructions, along with the use of DX and AX, can emulate 32-bit integers and perform 32-bit operations, as 16-bit computers cannot naturally use any 32-bit operations/registers/integers.

1. Write a program which will have the register ‘AX’ as its input. It would display the contents of AX in binary on the screen. You should use shifts to achieve the function. Demonstrate by using several values of AX.



**

**

1. Write a program that performs C = A + B using extended addition. See textbook pg. 270-271.

Use the following:

Apple QWORD 1111222233334444h

Berry QWORD 13572468ABCD0000h

Cherry QWORD ?

You may only use16-bit registers to perform the addition, e.g. AX, BX etc.

Submit the asm/list file and a screenshot of your code printing the contents of all the arrays after the run.

