Shane Bolding

[Shb7@students.uwf.edu](mailto:Shb7@students.uwf.edu)

EEL4744L: Microprocessor Lab

Lab 4: Writing and Testing A Simple Program

10-3-19

**Objectives:**

The object of this lab is as the title of the lab suggests. We are writing and testing a simple program that’s going to test to see if each number inside an array is either: positive, negative, even or odd. With this information it is then going to increment a counter that represents the number I.e. positive, negative, even, odd. For example, if the array was filled with the numbers 5 and -6 it should increment the odd and positive counter for 5 and it should also increment the even and negative counter for -6.

**Introduction:**

To do this lab we will be using a new bit of syntax, the BRACLR operation. This operation will branch if the memory that it checks has bits in the location that is provided. For example, if the line is BRCLR N, $01, exit the statement will jump to location of exit if the value if N has a bit in the ones spot. Using this syntax we can produce all the if else statements needed to separate negative numbers from positive numbers and odd numbers from even numbers.

**Procedure:**

We need to write a program that counts out of a single array the number of even, odd, positive, and negative numbers. The program below does so successfully.

\*Program SHB7

\*Checks Array of N bytes to see if entries are +, -, even, or odd

\*Declares flag counters

ORG $00

Ne RMB 1

P RMB 1

E RMB 1

O RMB 1

\*Declares Array

ORG $100

N equ 5

array fcb N

\*Start program

ORG $B600

ldx #array

Odd brclr 0,X,$01,Even

inc O

bra Pos

Even inc E

Pos brclr 0,X,%10000000,Neg

inc P

bra chkend

Neg inc Ne

chkend cpx #array+N-1

bhs exit

inx

bra Odd

\*End Program

exit swi

Programming the code into the board and modifying the initialization of the array to have preset values we can test the array. For example, the code below.

\*Program SHB7

\*Checks Array of N bytes to see if entries are +, -, even, or odd

\*Declares flag counters

ORG $00

Ne RMB 1

P RMB 1

E RMB 1

O RMB 1

\*Declares Array

ORG $100

N equ 8

array fcb $12, $44, $31, $89, $51, $F4, $D6, $91

\*Start program

ORG $B600

ldx #array

Odd brclr 0,X,$01,Even

inc O

bra Pos

Even inc E

Pos brclr 0,X,%10000000,Neg

inc P

bra chkend

Neg inc Ne

chkend cpx #array+N-1

bhs exit

inx

bra Odd

\*End Program

exit swi

As you can see the array is preset. The lab is based around this array and the instructor gave us preset arrays to test. The three sets and their tests are shown below.

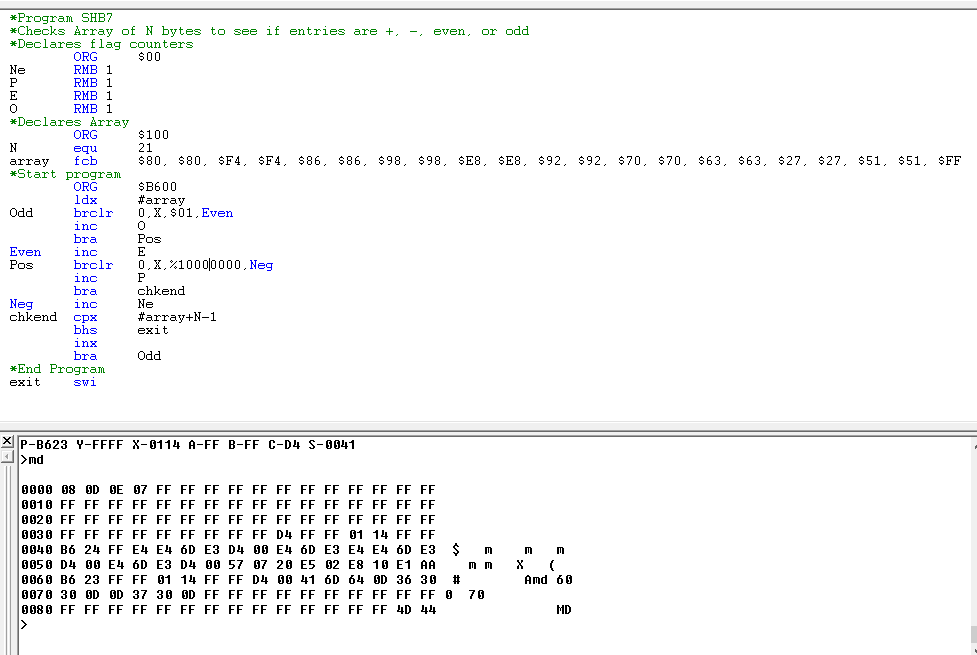


Figure 1 The code and results above show that array N has 8 Negative #, 13 positive #, 14 even #, and 7 odd #

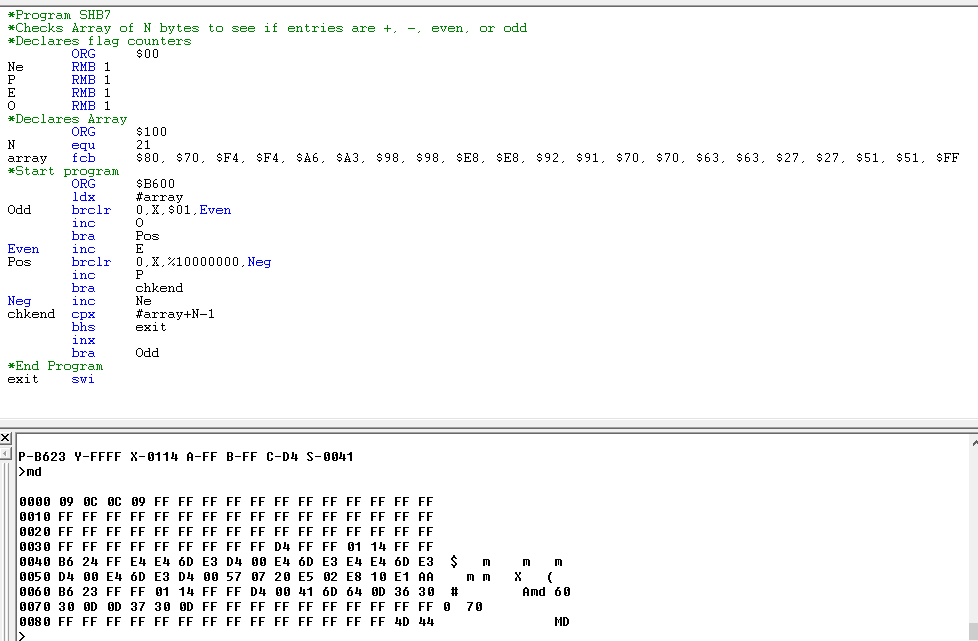


Figure 2 The code and results above show that array N has 9 Negative #, 12 positive #, 12 even #, and 9 odd #

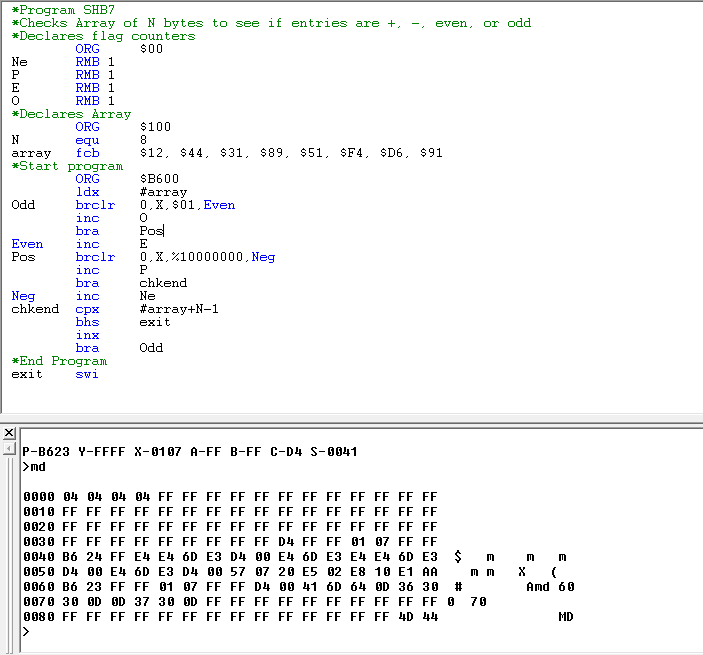


Figure 3 The code and results above show that array N has 4 Negative #, 4 positive #, 4 even #, and 4 odd #

As you can see these pictures show the count of the even, odd, positive, and negative numbers accurately without flaw.

**Conclusion:**

This lab gave us great practice with more advance branched conditions like branch if clear and branch if set. We used these commands to identify a number’s MSB and LSB successfully and demonstrated the success as well. The Lab went better this week because of now knowing the Baud rate issue from the last weeks lab. All in all this was a great learning experience to step up out syntax in the assembly language.