

EE 308 – LAB 3

Assembly Language Programming

Introduction and Objectives

The purpose of this lab is to write a few assembly language programs and test them on the simulator and on your EVBU.

Pre-Lab

Make sure you have the programs clearly thought out and written down before you come to lab. You should put all your code starting at memory location 0x1000. You are encouraged to bring the programs in on disk.

The Lab

As in last week's lab you will write some programs in assembly language and run the programs on the HC12 simulator and the EVBU. To make sure your programs work you will sometimes use D-Bug 12 op codes as your input data. There is one problem when running programs on the simulator – D-Bug 12 code which is present in the EVBU is not on the simulator. We can fix this problem by loading D-Bug 12 into the simulator. To do this, copy the file `n:\ee308\dbug12.h12` into your directory, start the ZAP simulator, and load the file `dbug12.h12`

Write and run the following programs:

1. Write a program which writes a 0xffff to address 0x0002, and then increments the 16-bit number at address 0x0000 indefinitely. (Note: this is about a five-line program.)

Test your program both on the simulator and the EVBU. Trace through the program on the simulator and observe what happens to the data at address 0x0000.

Note: This should cause the pins labeled D0-15 on the HC12 to toggle. Use a logic probe to verify that this is happening.

2. Write a program to swap the last element of an array with the first element, the next-to-last element with the second element, etc. The array should have 0x20 eight-bit numbers and should start at 0x2000. (This is similar to Problem 5 from the homework due Feb. 10.)

Check that your program works both on the simulator and the EVBU. Use the following data for your test:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2000	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
2010	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F

3. Write a program which puts the the exclusive OR of the eight-bit numbers from memory locations 0x8000 through 0x8FFF into memory location 0x2003. (This operation is often

used to generate a check sum to verify data transmission. The sending computer generates and transmits the check sum along with the data. The receiving computer calculates the check sum for the received data and compares it with the check sum sent by the sending computer. If the two values do not match, then there was an error in the transmission. The last byte in a line of an S19 file is a check sum used for this purpose.)