

Lab 1 - Loading and Running Programs

Instructor: Fred Etcheverry

Read Chapters 1 and 2, *MC 68HC11 An Introduction Software and Hardware Interfacing* by Han-Way Huang. All *CME11E9-EVBU Development Board Instruction Manual*. Download form www.axman.com.

Equipment: PC computer with Windows 3.1 or higher.

CME-11E9-EVBU Development Board Instruction Manual. Purchased from Axiom Manufacturing Inc. www.axman.com.

Software: AxIDE v. 2. comes with Development Board or can be downloaded.

Connect the Development Board to a PC according to Instruction Manual. **Install** AxIDE under Windows.

Before starting this lab the student must be familiar with *number systems*. Computers are constructed of transistors that function as switches. That is, these transistors are either “on” represented by a 1 or “off” represented by a 0. The number system using just 1s and 0s is the *binary* system. Thus computers use the binary system. Since binary numbers are much longer than their decimal equivalents and can be easily misread, we usually gather them into groups of four and express them as hexadecimal numbers. Usually, we just call them “hex.” A complete discussion of number systems is covered in Appendix A in the Text, *MC 68HC11 An Introduction Software and Hardware Interfacing* by Han-Way Huang.

The Axiom terminal emulation and cross assembler software (AxIDE) is an improvement over the one described by Huang (p.76 — 82). This software is shipped with the EVBU. The latest version can be downloaded from www.axman.com. Read the CME11E9-EVBU Development Board Instruction Manual. This can be downloaded from the same site. Go to SUPPORT. Under the heading MC68HC11 go to CME-11E9-EVBU click on DOCUMENTATION.

To invoke the software, click on the AxIDE icon. The AxIDE window should appear with a flashing cursor in the left upper corner. Make sure that you have the latest version of AxIDE; click HELP>ABOUT. The latest version at this writing is Version 2.0. Press the reset button on the EVBU Board. The BUFFALO monitor that resides in the 68HC11 should respond:

BUFFALO 3.4 (ext) - Bit User Fast Friendly Aid to Logical Operation

Press the Enter key and the monitor will respond with the > prompt indicating that it is ready to take commands. If the board responded with garbage, it maybe that the port settings are incorrect. Click

the red check mark or File>options to view the port setting. Click on the Port tab. The setting should be:

Baud Rate	9600
Parity	None
Data Bits	8
Stop Bits	1

Look at the Memory Map in the Instruction Manual (p.12). Notice that memory locations B600 to B7FF are dedicated to **HC11 Internal EEPROM in U1 Program or Data**. We will soon be uploading programs to this area. Let's use BUFFALO commands to manipulate a few memory locations. You will find a list of these commands in Huang (p.86) and in the Axiom Development Board Manual (p.24). Notice that the command MD will "dump memory to terminal." This means that the contents of memory will be read and sent to the monitor. Read Huang's description of the MD command (p.86 - 87). At the BUFFALO prompt type:

>MD B600 [press the enter key]

BUFFALO will display the following:



Each memory location contains eight bits (one byte) which may be either 1 or 0. Four bits are call a "nibble" and may be represented by a single hex digit. Thus two hex digits can represent the contents of each memory location.

The content FF is hex equivalent for binary 11111111. That is, each memory location contains a 1 indicating that it is erased. If the contents are not FF then use the command **BULK** to erase the EEPROM and then recheck by running the MD command.

Let's use the **MM** (memory modify) command to change a few locations. At the BUFFALO prompt type **MM B601**. BUFFALO should respond with B601 FF. The FF shows the presents contents of B601. Type AA and press the Enter key. Use the **MD** command to recheck the memory and observe that B601 now contains AA.

1. Use BUFFALO commands to change location B660 to 48. Describe the procedure.

Notice that the letter H appears to the right of the block of FFs on the same line as B660. Look in Appendix A (p.484) at Table A.6 ASCII code chart. ASCII is a code used by computers to transmit text characters. Notice in this table that hex “48” represents the letter “H.”

The MD command displays a hex memory map on the left side of the PC’s screen. This is the Hex Field. On the right side of the screen, the printable ASCII memory contents are displayed in the ASCII field. This field is blank if none of the memory contains printable ASCII codes. Thus when the contents of a memory location represent an ASCII letter, number, or punctuation it appears in the ASCII field. This is useful when the programmer is working with messages such as, “FOOD IS BURNING!” We will use this feature in future labs.

2. Use the BUFFALO MM command and the ASCII table to write a message that will appear in the ASCII field such as. “FOOD IS BURNING!”
3. Record steps.

4. Write the ASCII codes.

[NOTE: You will need to insert the ASCII code for space **SP**.]

Later we will learn to type messages in the assembler. Using the MM command would be tedious if we were writing long messages. Occasionally, we will use the MM to make minor adjustments to programs. Usually, we will write the program in assembly language and let the Assembler write instructions in machine code that we will then transmit to the Development Board via the COM Port and cable. Let’s demonstrate this with the following sample program. Review Chapter 2 of the Text to understand the instructions in this program.

***This is a program to add two numbers.**

START	ORG	\$B600	;starting address
	LDAA	#03	;initialize A to 3
	STAA	\$0210	;store in user RAM
	ADDA	#02	;add 2
	STAA	\$0211	;store in user RAM
	END		

Write this program in MS-DOS. Go to Start>Programs>MS-DOS Prompt. DOS will prompt C :\ Windows. After this prompt, type **CD** and press Enter. DOS will prompt: C:\. Type **MKDIR 68HC11** and press the Enter key. The last command MKDIR 68HC11 will create a directory (folder) called 68HC11. Go to this directory by typing **CD 68HC11** after the DOS prompt and pressing Enter. The command **CD 68HC11** changes directory to the one that you just created. DOS will prompt C:

\68HC11. Type EDIT LAB1.ASM and press Enter. The DOS Edit Screen will appear. Type in the above program. Make sure that it is lined up against the left side. That is, there is no space between the left side and, in this case, the word "START."

Save this file [File>Save] and exit the Edit Screen. At the DOS prompt type "EXIT." This should bring you back to the AxIDE window. Click on **Build. Browse** to find the directory **68HC11** and **LAB1.ASM** file. Enable **Listing Options; Output Cross Reference Table** and **Enable Cycle Counting** by clicking a check in their boxes. Click **OK**.

If there are any syntax errors, they will be listed by line number. Were you careful to put an asterisk before the first line and not to leave any spaces before **START**? When the assembly file is syntax correct, the assembler should produce a listing LAB1.LST as follows.

```
0001                                     *This is a program to add two numbers.0002 0200
START  ORG      $0200      ;STARTING ADDRESS
0003 0200 86 03          [ 2 ]          LDAA  #03      ;INITIALIZE A TO 3
0004 0202 b7 02 10      [ 4 ]          STAA  $0210   ;STORE IN USER RAM
0005 0205 8b 02          [ 2 ]          ADDA  #02      ;ADD 2
0006 0207 b7 02 11      [ 4 ]          STAA  $0211   ;STORE IN USER RAM
0007                                     END
```

In this listing, the original **source program** is displayed on the right. The numbers on the far right are the line numbers. Notice that after line 0003 the address 0200 is shown. This is the starting address of the program. The word START is just a label. The assembler directive ORG \$0200, in this case, directs the assembler to locate the beginning of the program at \$0200. The address 0200 appears again after line 0003 and then the number 86 and 03. This indicates that the content of location 0200 is 86 while that of 0201 is 03. These numbers, 86 and 03, are the first two bytes of machine code; they tell the microprocessor to load the accumulator A immediately with 03 (LDAA #03). Notice that hex address and constants in the **source** that was assembled must be preceded by a dollar sign, if they are to be interpreted by the assembler as hex. Those in machine language or the **object** file are not preceded by a dollar sign since the BUFFALO monitor only understands hex. The assembler created another file, LAB1.S19. Return to DOS. At the C:\68HC prompt, type EDIT LAB1.S19. The Edit screen will show S10D02008603 B702108B02B7021147S9030000FC . This is the stream of data that you will soon send to the processor serially through the COM Port and cable. The first character, S10D tell the processor that data to be loaded in memory is coming. The next four digits 0200 are the starting address. The next digits up to the S are the machine code or object that you see in the above listing. Exit this file and DOS. Normally, you will not have to look at this file.

To up load this program, at the BUFFALO prompt type LOAD T {enter}, that is, press the Enter key. Click on **Upload. Browse** to find the file **LAB1.S19** and then click **OK**. BUFFALO will respond after the file is loaded with "done." To run this program, type G 0200 at the BUFFALO prompt. To check that the program ran, press the reset button and at the BUFFALO prompt type MD 0200 {enter}.

5. Record the contents of:

[\$0210] _____ , and [\$0211] _____ . Are these values correct?

Notice that the constants [the two numbers to be added] 03 and 02 in the source file are not preceded

by a dollar sign. Would the dollar sign in this case make a difference?

6. Explain.

7. Use the MM instruction to change the two numbers being added to \$14 and \$43.

8. Run the program and record the contents of:

[\$0210] _____ , and [\$0211] _____ . Are these values correct? [Remember these numbers are in hex.]

9. Modify the above program to subtract one number from another. Write the program and check the results.

10. Write a program to add \$4322 to \$2417. Store the results in \$0210 to \$0213 [See p.46 Multiprecision Addition]