Lab #1 Introduction to EVBU Board, TUTE, and Buffalo ECE 271

This lab will allow you to experiment with the commands used by Buffalo to control the operation of the EVBU. The commands you will use are **help**, **md**, **mm**, **rm**, and **t**. You will use these commands to examine and modify the memory contents of the EVBU and finally to cause the 68HC11 to execute some simple instructions. In steps **1 through 6** you are asked to fill in the answer sheet with your results and hand them in at the end of the lab period.

Step 1.

- Plug in all the connectors discussed in class and start the TUTE program by selecting Programs → TUTE on the PC.
- When TUTE opens, you must initialize the serial connection to the EVBU board by selecting Run→HC11→Buffalo→Connect→Comm 1. This should establish a connection between the PC and the EVBU.
- To test this connection, click on the Comm tab and then its white area . Finally, press the Reset button on the EVBU board (next to lit LED). If the connection works, the EVBU board will print a welcome message to the Comm pane that will start as

BUFFALO 3.42 – Bit User Fast Friendly Aid to Logical Operation

- To make using the "Comm" area easier to use you can expand that part of the screen by clicking and dragging the bottom edge of the blue area upward.
- Click in the "Comm" area and then hit the "ENTER" key. Several lines and a prompt (>) will be printed. Commands are given to the EVBU board by typing them after the prompt.
- Try entering **help** after the prompt. This will cause the EVBU to print a list of all the commands that it supports.
- Find the commands **md**, **mm**, **rm**, and **t** that you will be using today in the list printed to the screen. Can you make sense of their cryptic notation? This help listing will always be there if you forget a command. NOTE: These instructions are NOT case sensitive.

Step 2.

Use the **md E5A0** command to display the contents of memory starting at $E5A0. You will see two representations of the data. The one on the left is hex, the other is the ASCII equivalent of the hex digit. In the midst of the ASCII data you should recognize the welcome message printed to the screen whenever you press the reset button on the EVBU. What is the hexadecimal value for the capital letter B in BUFFALO that appears in the string, and what is its memory address? Record your answers on the answer sheet. [Note that BUFFALO does not use or allow a dollar sign ($) for hexadecimal; all numbers are assumed to be hexadecimal.]

Step 3.

Type **mm 0100 [Ret]** You should see 0100 FF. To the right of the FF type **41 [Ret]** and then see if the change took place by typing **mm 0100** again. Hit **[Ret]** to cancel modifying the location. Try using this technique to change the contents of location $E633 to $41. What happens and why?

Step 4.

- Enter the command **md 1003 1004** which will cause Buffalo to print only one line (16 bytes) of memory. (Note that the memory starts at $1000, NOT $1003.)
- Location $1003 is called port C. It is connected to the "DIP" switches just below the row of 8 LEDs. The 8 bits of location $1003 will be 1 or 0 depending on the state of the switches. The least significant 2 bits are also connected to the 2 buttons labeled PC0 and PC1 on the board. The buttons can be used when the corresponding bits of the DIP switches are in the UP position. The EVBU prints the contents of memory locations in hexadecimal so you'll need to convert them to binary to find out which bits are changing.
- By trial and error determine the relationship between the DIP switches, the 2 buttons and the 8 input bits of this port. Record your answers on the answer sheet.

Step 5.

- Memory location $1004 is called port B. It is an output port, and its bits control the 8 LEDS on your board.
- Experiment to determine which bits control which LED's by using the **mm** command to change the data stored at $1004. Is a zero ON or OFF? Note that with the **mm** command, entering a ***/*** (front slash) after the data you enter, allows you to keep changing the memory location without retyping the address. Hit Enter when done.

Step 6.

- In this step you will initialize some 68HC11 registers and then create and run a simple program.
- Use the **rm** command to display the contents of the registers of the 68HC11. The display will finish with P=FFFF and the cursor will be to the right of the FFFF.
- Entering a value causes Buffalo to change the program counter (P) to the value you entered. Change the program counter to 0100. Do NOT press Enter – this exits **rm**
- Pressing the space bar will cause Buffalo to cycle through the registers. When you get to A change it to $00, and then change accumulator B to $40.
- Now use the **mm** command to set memory location $0100 to $1B, $0101 to $20, and $0102 to $FD. Note that hitting the space key after entering data brings up the next memory location.
- $1B is an instruction (ABA) for the 68HC11 that will cause it to add the contents of the A register and the B register and place the sum back into the A register. The next two bytes ($20 and $FD) form a branch instruction that will cause the processor to go back and execute the instruction at $100 again. When you set the P (program counter) register in the 68HC11 to $100 you aimed the processor at the ABA instruction. The processor is all set to execute this instruction.
- Execute the instruction at $100 by entering the trace command, **t**. Examine the contents of the registers A, B, C, and P and record them in the Trace 1 row. Notice that the value of the P register is now $101. It has been incremented by 1 to point to the next instruction of the code.
- Hit **t** again to cause another trace. The program counter should now point to $100 so you can execute the ABA instruction again.
- Cycle through this program until the PC = 0100 and A = 0 (back to their starting values). Make a list of the values in A, B, C, and P for 8 traces starting from the first one.

Answer Sheet for Lab#1. Name_____

Step 2. The hexadecimal value for B in the string of ASCII characters that forms the welcome message is
$_____.

The address of the B mentioned above is $_____.

Step 3. What happened at location $0100?

What happened at location $E633?

Step 4. Answer the following questions about the relationship between the switches, buttons and the value read at PortC ($1003)

The leftmost switch (SW8) controls Bit #_____ of PortC, and the rightmost switch (SW1) controls Bit #_____ of PortC? Note that Bit # 0 is labeled as “1” on the switch itself, and Bit # 7 is labeled as “8”

Does “UP” create a “1” or “0” ? Answer: _____

When the right two switches are UP, what does pushing the button create a “1” or a “0” ? Answer: _____

Step 5. For the LEDs in PortB ($1004), does writing a “1” make the LED go ON or OFF? Answer _____

Step 6. A. Fill in the following table to show what you found for the contents of the registers <u>after</u> the indicated traces. Give values in hex. Note that C is Buffalo's shorthand for CCR, the Condition Code Register and that P is Buffalo's shorthand for PC the program counter.

Trace	A	B	C	P
1	___	___	___	_____
2	___	___	___	_____
3	___	___	___	_____
4	___	___	___	_____
5	___	___	___	_____
6	___	___	___	_____
7	___	___	___	_____
8	___	___	___	_____

B. Use your results from the table above to fill out the table below.
Note that Figure 2-1 on page 50 of your text (2nd or 3rd edition) lists the individual bits of the CCR. That information is listed again here for reference

MSB LSB
CCR bits = S X H I N Z V C

Values in these two columns are at the <u>start</u> of the trace

Values in these 6 columns are <u>after</u> the indicated trace. (The left-hand C refers to the value in the CCR while the right-hand C refers to the Carry bit.)

Trace	A	B		A	C = CCR	N	Z	V	C
1	$00	+ $40	=	___	____	__	__	__	__
3	___	+ $40	=	___	____	__	__	__	__
5	___	+ $40	=	___	____	__	__	__	__
7	___	+ $40	=	___	____	__	__	__	__

C. Briefly explain why the condition code bits behaved as they did for each of the entries in the above table.