LAB 1 – Introduction

ASSEMBLY PROGRAMMING WITH MINIIDE

The purpose of this lab is to introduce you to the layout and structure of Assembly Language programs and their format. You will write your own programs later on in the semester with similar structure. All code should be capitalized. Comments should not be capitalized.

From left to right, you will notice four columns. The first column contains label names. The second column contains instruction/assembler keywords. The third column contains data or operands. The fourth column is used for comments. Your programs should always be in this format.

INTRODUCTION TO MiniIDE

MiniIDE is a windows based program, which allows assembly programmer to assemble, debug, and download a program onto the 68HCS12 board. This lab familiarizes the student with all the steps involved in loading, running, and debugging assembly language programs using MiniIDE. The student will be required to remember all the steps to load programs in future labs.

Please use your U: drive for all your lab work.

READ ALL INSTRUCTIONS CAREFULLY!!

PROCEDURE:

I) USING MINIIDE

1. Start → All Programs → Engineering → ECE CAD Lab → MGTEK MiniIDE
2. Click on the **File → New.** This is where you will write your program.

Type the following program into the edit window

```
*****************************************************************
* LAB1.ASM                      Student Name*
* Introduction                  *
*****************************************************************

DATA    EQU     $2100   ;label Data equals hex 2100
NUM     EQU     $AC     ;label NUM equals hex AC
        ORG     $2100
NUM_ONE DC.B    $8F     ;define Constant Byte with label NUM_ONE &
                        ;value hex 8F
NUM_TWO RMB     1       ;reserve ONE Memory Byte
        ORG     $2000    ;start at memory location 2000
        CLR     NUM_TWO ;clear reserved memory location, $2101
        LDAA    #NUM    ;load the value of label NUM into Accumulator A
        LDX     #DATA   ;load register X with the value of label DATA
        LDAB    $0,X    ;load Accumulator B with content of memory
                        ;location pointed to by register X incremented
                        ;by 0 (2100)
        STAB    NUM_ONE ;store the content of Accumulator B into
                        ;NUM_ONE
        STAA    NUM_TWO ;store the content of Accumulator A into
                        ;NUM_TWO
        LDAA    #$80    ;load Accumulator A with value $80 (immediate
                        ;addressing mode
        ABA             ;add the content of A & B and store result in A
        SWI             ;Software Interrupt Command. Makes
                        ;microcontroller Stop executing till further
                        ;instructions
```

II) RUNNING A PROGRAM

1. First, make sure you can communicate with the 68HCS12 board. Go to **Terminal → Options…**. Here you should see a window that has the COM settings. Make sure the COM port that is selected is **COM1**. Also, make sure the HCS12 board is plugged in!

2. Hit the **RESET** button on the 68HCS12 board. This is located in the middle of the bottom edge of the board. You should get the following text in the Terminal Window:

```
D-Bug12 4.0.0b29
Copyright 1996 - 2005 Freescale Semiconductor
For Commands type "Help"

>
```

3. In the **Terminal Window**, at the prompt, type **LOAD→ Enter**. The terminal window commands are not case sensitive.

4. Go to **Build**--> **Build Current**

5. Click the **F8** button. A window will pop up. Select the LAB1.s19 file. Click **OPEN**.

Normally, you would run this program by typing **g 2000**. However, for this lab, we want to see what will happen as we execute the program step-by-step. To do so follow the following instructions:

- In the Terminal Window, type **PC 2000 → Enter.** This sets the PC (Program Counter) to the value 2000. Note that all values are hexadecimal values.
- Type **t → Enter**.

T stands for Trace. This command is used to run the program one line at a time on the screen. One may also use **t 8** to execute all 8 lines and stop. T will trace singly or through as many lines as specified.

Observe the content of Accumulators A & B, Registers X as well as the CCR (Condition Code Register) for each line. ***Record*** these values in Table 1.

A	B	X	N	Z	V	C

Table 1

III) D-BUG12 MONITOR COMMANDS

This section will provide a walk-through of important and useful D-BUG12 commands. After steps 3, 4, and 5 – copy the terminal window output and paste it into Word as shown by the TA for inclusion in the report.

1. Hit the **RESET** button on the 68HCS12 board. Make sure the terminal window is still activated. If not, follow the instructions in part 2, step 1.

2. In the **Terminal Window** type **bf 2000 2200 0** and hit **Enter**. This has block filled all memory locations from $2000 to $2200 with zeroes. To view this change, type **md 2000 2200 → Enter**. The memory contents from $2000 to $2200 are displayed on screen, and all should be filled with zeroes. **md** stands for "memory display".

Note: Each memory display line holds the 16 byte block associated with the beginning address. The $2200 block displays $2200 - $220F for example on one line. Using the command **md $220A** will display the entire 16 byte block associated with it; therefore it will display $2200 - $220F.

3. Now let's look at modifying memory contents from the terminal window.
Type **mm 2110 → Enter**.

Make the following addresses hold these values by typing in the value corresponding to the memory location and pressing enter (note that the 20xx will be displayed as a prompt):
2110 86
2111 9A
2112 25.
At **2113** type the following:
12 / → Enter.
You will see the results of changing $2113 to **12**.
Type **Enter → 55 ^ → Enter → 44 . → Enter**. Lastly, display the memory contents of $2110 using **md 2110** and record the final values below (assume these are hex. values):
2110 _____
2111 _____
2112 _____
2113 _____

4. To modify the registers, type the register name, space, and then the value. Examples below to type are:
A: 99
B: EB
X: 569A

Approved: Lab TA _____ Date _____

What to turn in during the next lab session

(in this order please with a cover sheet):

1. A print-out of LAB1.asm and Lab1.lst (*.lst in landscape). Use **Programmer's File Editor** available in the **Engineering** portion of the start menu. **DO NOT USE MINI_IDE TO PRINT THE LISTING IN LANDSCAPE! IT WILL TRUNCATE THE FILE.**
2. A print-out from the terminal screen (highlight, type ctrl+c, and paste into MS Word – use Courier New as the font) that includes the following:

- Results of entering **t**
- Results of Section III, Parts 3 and 4

Comment the print-out to indicate what each portion is displaying

3. Answers to the following questions about the code:
   - What was the original content of the memory location 2100?
   - What is the final content of memory locations 2100 & 2101? (use **md**)
   - When you added Accumulator A and B, what was the answer you got? What was the answer supposed to be? Why didn't you get that answer?