Assignment 2C (40 marks) – Lab Week Eight

Due: End of your week Nine's lab period: Week of 18 – 23 Mar 2018

PURPOSE OF LAB:

The purpose of this lab is to gain more experience with both the assembler and simulator that will be extensively used in this course. Additionally, you will have the opportunity to confirm your understanding of the HCS12 Instruction Set and BCD Arithmetic. The figure at the right lists some of the resources you will likely use to complete this lab exercise, all of which are found on Blackboard in the Resources folder.

Course Text - 68HCS12Text

HCS12-9S12 Instruction Set Reference

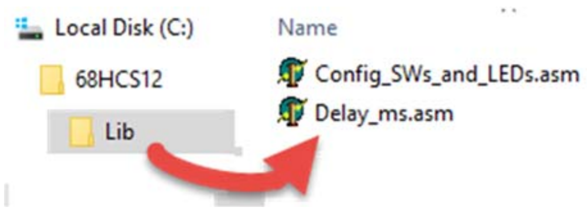
HCS12 Assembly Language Reference Manual

Prior to commencing each of the tasks, I highly recommend that you read the "Hand-In" Sheet, which contains the marking guide for each of the tasks.

TASK One – (7 marks) – A Simple Hardware Programming Exercise

The purpose of the supplied assembly language is to flash an LED on and off every 250 ms. We will use the simulator to observe the correct program run.

a. Download the compressed Library Files package (***Library_Files.zip***). Create a new folder called **Lib** in your **C:\68HCS12** folder (this was the folder where you originally installed the assembly language software). Unzip ***Library_Files.zip*** into **C:\68HCS12\Lib**. The figure at the right illustrates what you should have after unzipping the files. **Ensure that you did not created a sub-folder inside the Lib folder when you unzipped the files**.



b. Download the corrupted ***Flash_PB0.asm*** code listing into your Week Eight source directory (e.g. CST8216\Lab8) and correct the assembly language code source code listing so that the labels, opcodes, operands and comments are in the correct columns. You will also have to add the two missing lines of code that will include the two library files found in **C:\68HCS12\Lib** (read those files for instructions on how to use them – *ensure that their "include" statements are placed just before the "end" statement in your code*). **Note that if you receive the following error, then you incorrectly installed the software package and that you must de-install the software and then reinstall it into the correct path: Fatal error -- Can't open #include file C:\68HCS12\registers.inc**

c. Once you have the previous steps complete, assemble Flash_PB0.asm ensuring that there are no errors or warnings.

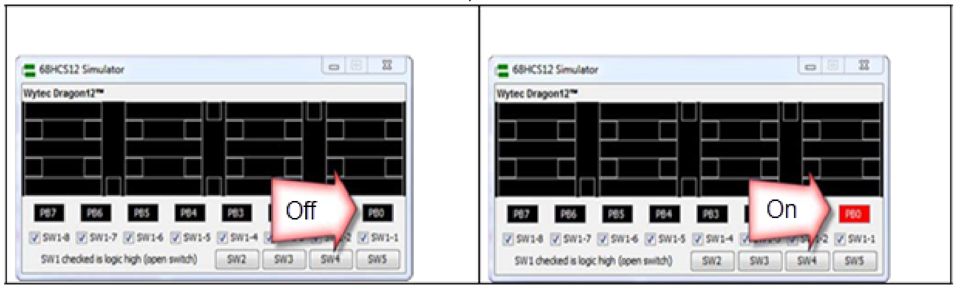
d. Load the .s19 file into the Simulator - Dragon12 & Student Mode and take the following action to run it:

I. Ensure Tracing is unchecked; otherwise, the LED will flash on and off only after several minutes of program run:

II. On the main menu, click on View → Parallel Ports;

III. For your convenience, I have included a short video presentation on the expected behaviour of this program. Observe that the simulator and parallel ports are identical to the ones in that video. If they are not, then you are using the wrong simulator and you will have to select the correct one before proceeding any further

IV. Next, click the "Go" button on the simulator and observe that the Port B LED (PB0) on the simulator Parallel Port flashes on and off at a rate of about 250 ms, or ¼ of a second:



Your solution should very closely match it. Once the program run is correct, demonstrate it, printing out the Lab Week Eight Hand—In Sheet BEFORE doing so. Note that your timing **may** be slightly different from that illustrated in the video.

Task Two – Those were the Memories (10 Marks)

To complete this task, analyze the given code and complete the hardware-based memory maps **as if you were using the actual hardware**, *not just the simulator* and complete the Pre-Execution and Post-Execution Memory Maps.

Pay particular attention to the **base** of the numbers in the source code and how they must be stored in memory.

Hint: Perform this task MANUALLY, and THEN check your answers using AsmIDE and the *Dragon12 & Student Mode Simulator.* Use your lab period to ask questions if you don't understand the results.

PLACE YOUR ANSWERS ON THE HAND-IN SHEET.

Given the following source code listing, complete the Pre-Execution Memory Map and Post-Execution Memory Map tables below.

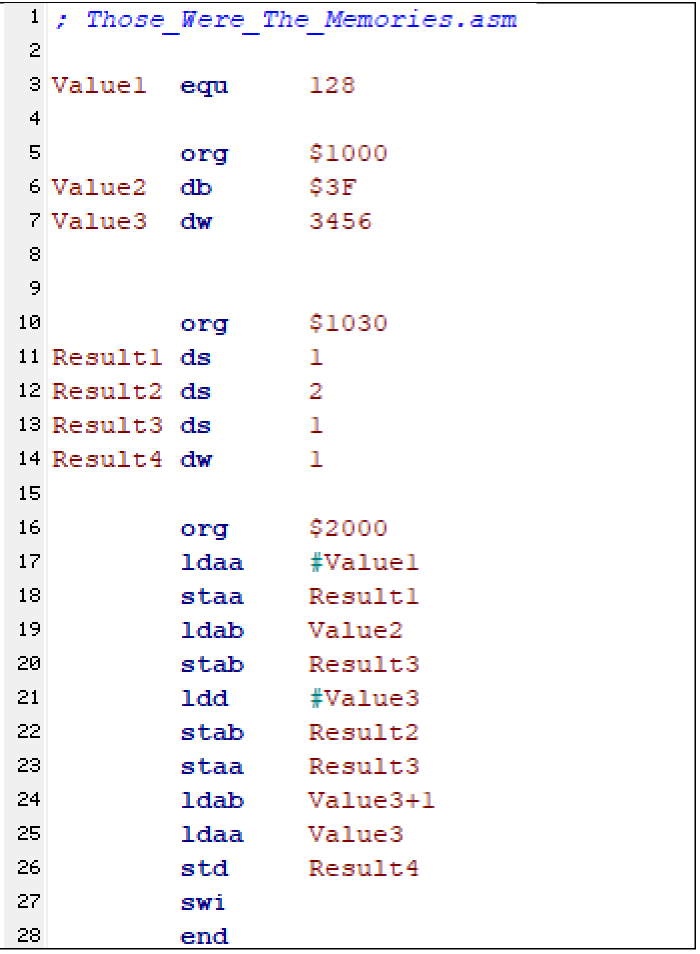
- **Ensure all values are expressed in *Hexadecimal***
- Where memory contents are unknown by the hardware, fill in the entry with two dashes – e.g. " - - ".
- **Ensure all memory addresses contain a value or " - - "**

Pre-Execution Memory Map

	← 8 bits →
$1000	
$1001	
$1002	
$1003	

Post-Execution Memory Map

	← 8 bits →
$1030	
$1031	
$1032	
$1033	
$1034	
$1035	



```
; Those_Were_The_Memories.asm

Value1  equ     128

        org     $1000
Value2  db      $3F
Value3  dw      3456


        org     $1030
Result1 ds      1
Result2 ds      2
Result3 ds      1
Result4 dw      1

        org     $2000
        ldaa    #Value1
        staa    Result1
        ldab    Value2
        stab    Result3
        ldd     #Value3
        stab    Result2
        staa    Result3
        ldab    Value3+1
        ldaa    Value3
        std     Result4
        swi
        end
```

Task Three –BCD Arithmetic (8 Marks)

Complete the material on page 1 of the Hand-In Sheet.

Task Four – Write Some Code Snippets (15 Marks)

Complete the material on page 2 of the Hand-In Sheet.

Assignment 2C (40 marks) – Lab Week Eight **Page 1 of 2**

Due: End of your week Nine's lab period: Week of 18 – 23 Mar 2018

Name: ______________________________ ***Indicate Your Lab Period*** ***/ 40 marks***

Student Number: _________________ ____

TASK One – (7 marks) – A Simple Hardware Programming Exercise

Code Inspection (3 marks):

a. The filename for the code should be "as provided" in this assignment and *your Student Information* should be evident in the program header.
b. The code should be correctly aligned with all Labels, Opcodes, Operands and Comments in their correct columns as discussed in a previous lecture.
c. The program code should contain the missing two lines of code in the correct location in the source code.
d. The two supplied library files must be in the correct folder on your system.

Program Demo (4 marks): The program must run in the simulator without error *on the first demonstration attempt.*

Professor's Initials _______________ /7 (In-class Code Inspection + Demo)

Task Two – Those were the Memories (10 marks)

Complete the following hardware-based memory maps according to the instructions contained within this assignment:

Pre-Execution Memory Map

	← 8 bits →
$1000	
$1001	
$1002	
$1003	

Post-Execution Memory Map

	← 8 bits →
$1030	
$1031	
$1032	
$1033	
$1034	
$1035	

Assessed Post-Lab _____ /10

Task Three –BCD Arithmetic (8 Marks)

(8 marks) To confirm your understanding of BCD Arithmetic methods, represent the following base 10 numbers in BCD and then correctly perform BCD addition. ***Show all steps (including all annotations to the right of the additions as per the BCD Arithmetic video) for full credit***.

a. Add 86 + 21	b. Add 572+ 299

Assessed Post-Lab _____ /8

Assignment 2C (40 marks) – Lab Week Eight *Page 2 of 2*

Due: End of your week Nine's lab period – Week of 4 – 18 Dec 2017

Task Four – Write Some Code Snippets (15 Marks)

Course Text - 68HCS12Text

HCS12-9S12 Instruction Set Reference

HCS12 Assembly Language Reference Manual

Using the resources available on Blackboard in the Resources folder, lecture material, and your class notes write a single line of code, using the HCS12 Instruction Set, that performs the following:

Hints:

I. Since The HCS12-9S12 Instruction Set Reference will be provided to you on Term Test Two and the Final Exam, it may be a good idea to look through that document first.
II. You would also gain better experience with the instruction set if you actually coded these one-line snippet of code to see how they work.

Required Operation	Line of Code
a. Loads Accumulator A with the value of 8	
b. Loads Accumulator B with the contents of memory address $1005	
c. Increments Accumulator B	
d. Decrements Accumulator A	
e. Compares the values in Accumulators A and B	
f. Exchanges Register Y and X's values	
g. Adds Accumulator A and B	
h. Compares Accumulator B with the contents of memory address $1A00	
i. Points X to memory address designated by the label Max_Value	
j. Loads Y with the contents of memory designated by the label Car_Value	
k. Logically shifts Accumulator B to the Right	
l. Subtracts B from A	
m. Transfers A to B	
n. Stores D at a memory location designated by the label Temp	
o. Loads D with the value $1234	

Assessed Post-Lab _____ /15