

Assignment 2B (30 marks) – Lab Week Seven**Due: End of your week Eight's lab period: Week of 26 – 30 Oct 2015****REVISED**

2:55 pm, Oct 23, 2015

Second Assembly Program – Assemble and Trace

This portion of the lab exercise has you create and trace a simple program using 68HCS12 Assembly Language as the target language using *AsmIDE* and *Simulator – Dragon12 & Student Mode*.


PURPOSE OF LAB:

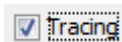
The purpose of this lab is to gain more experience with both the assembler and simulator that will be extensively used in this course. Additionally, you will have the opportunity to confirm your understanding of Mathematical Principles taught during in-class lectures.

Resources:

The following material is available on Blackboard in the Resources folder to assist you in answering questions contained in this lab exercise. The Help feature in *AsmIDE* may also be of assistance to you.

[Course Text - 68HCS12Text](#)[HCS12-9S12 Instruction Set Reference](#)[HCS12 Assembly Language Reference Manual](#)**Task One – Assemble; Trace; Answer (30 Marks)**

- Load the source code for Math_Operations.asm into *AsmIDE*. You may find that you will have to align Labels, Opcodes, Operands and Comments into their correct columns in order to correctly assemble the program. This alignment was explained in a previous lab exercise and discussed in a lecture period.
- Assemble the code, ensure that there are no errors or warnings present.
- Start *Simulator – Dragon12 & Student Mode* and load Math_Operations.s19
- As you  through the program, **trace the Task One – Part One Questions on the A2B Lab Week Seven Hand-In sheet**.
- Once you have completed the above steps, prepare the simulator to rerun the program by using **File → Reset**. Now, click on



then select **View → Console Log** and Step through the program again. What you will observe is a complete trace of your program, which is very handy in debugging Assembly Language programs!

- Now, **answer the Task One – Part Two Questions on the A2B Lab Week Seven Hand-In sheet**.

```

1 ; Math_Operations.asm
2 ;
3 ; Author:          D. Haley
4 ; Student Number:  nnn-nnn-nnn
5 ; Date:            4 Oct 2015
6 ;
7 ; Purpose:         To Gain Experience with Assembly Language instructions
8 ;
9 ; Program Constants
10 STACK equ $2000
11
12 org $1000 ; Data starts at $1000
13 MyArray db $0C,$81
14 Result ds 2
15
16 org $2000 ; Program Code starts at $2000
17 Start lds #STACK ; Setup the stack
18
19 ldaa #$25 ; Note: This code is purposely NOT documented
20 ldab #25
21 adda MyArray
22 staa Result
23 stab Result+1
24 ldaa Result+1
25 ldab Result
26 incb
27 ldaa #%10101010
28 std Result
29 swi
30 end

```

Assignment 2B (30 marks) – Lab Week Seven Hand-In Sheet (Due: End of your week Eight's lab period).**Name:** _____**Circle Your Lab Period/Time****Student Number:** _____

Tue: 10 – 12 Wed: 1 – 3 Wed 3 – 5 Fri 2 – 4

Task One – Part One Questions (18 marks)–Place your answers in the allocated space.

Record the values of the registers and memory locations **after** executing each listed instruction for **Math_Operations.asm**. Watch how each instruction changes the memory locations and registers that are recorded in the table.

Ensure all hexadecimal values are preceded with a dollar (\$) sign and are in CAPITAL LETTERS – e.g. \$4F (otherwise, marks may be deducted).

To assist you, I have included the entries for the first line of interest in the supplied source code listing, which is line 19 **Idaa #\$25**. Note that memory locations \$1000 and \$1001 were filled with their values before line 19 executes.

Line, Instruction	PC	A	B	D	\$1000	\$1001	\$1002	\$1003
19 Idaa #\$25	\$2005	\$25	\$00	\$2500	\$0C	\$81	\$00	\$00
20 Idab #25								
21 adda MyArray								
22 staa Result								
23 stab Result+1								
24 Idaa Result+1								
25 Idab Result								
26 incb								
27 Idaa #%10101010								
28 std Result								

Task One – Part Two Questions (12 marks) – Place your answers in the allocated space.

Now, based upon the values in the above table, answer the following questions.

Ensure all hexadecimal values are preceded with a dollar (\$) sign and are in CAPITAL LETTERS – e.g. \$4F (otherwise, marks may be deducted).

Question	Answer
a. How did \$1000 and \$1001 initially have their memory addresses filled with the indicated values at line 19? To answer this question, replicate the line of code that accomplished that.	
b. After the execution of line 20, what value is loaded into Accumulator B?	
c. What is the address associated with the label “MyArray”?	
d. After the execution of line 21, the value in Accumulator A changed. What arithmetic operation took place?	
e. What is the address associated with the label “Result+1”?	
f. What high-level programming function do lines 22 to 25 perform? (5 marks – think about this one)	
g. What address mode is used in line 27? To answer this question, look up Load Accumulator A (LDAA) in the HCS12 Assembly Language Reference Manual. This instruction uses a Source Form of opr8i	
h. What address mode is used in line 28? To answer this question, look up Store Double Accumulator (STD). This instruction uses a Source Form of opr16a	