Khalifa University of Science, Technology and Research

Electronic Engineering Department

Microprocessor Systems Laboratory ELCE333

Experiment No. 1

Pre-Lab Report

MICROCONTROLLER ASSEMBLY PROGRAM DEVELOPMENT

Lab Partners:

Dana Bazazeh, 100038654

Moza Al Ali 100038604

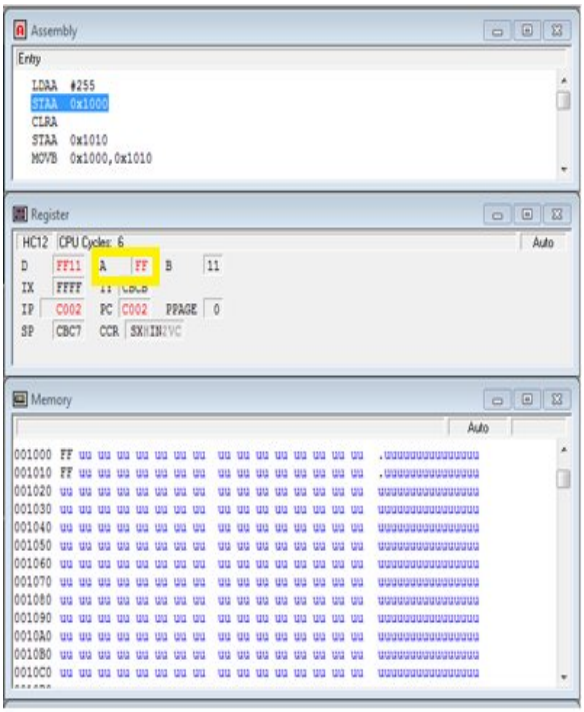
Date Report Submitted: 28/01/2015

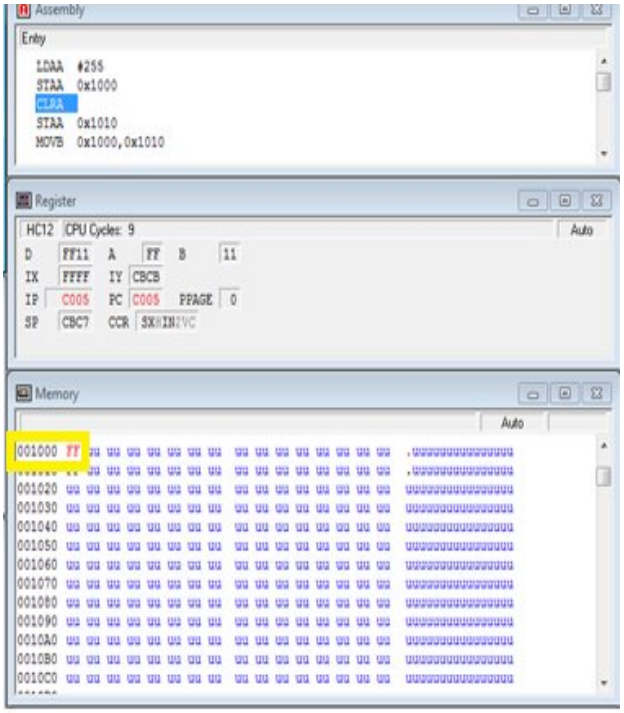
Lab Instructor: Dr.Mohamed Al Zaabi/ Dr.Mahmoud Khonji

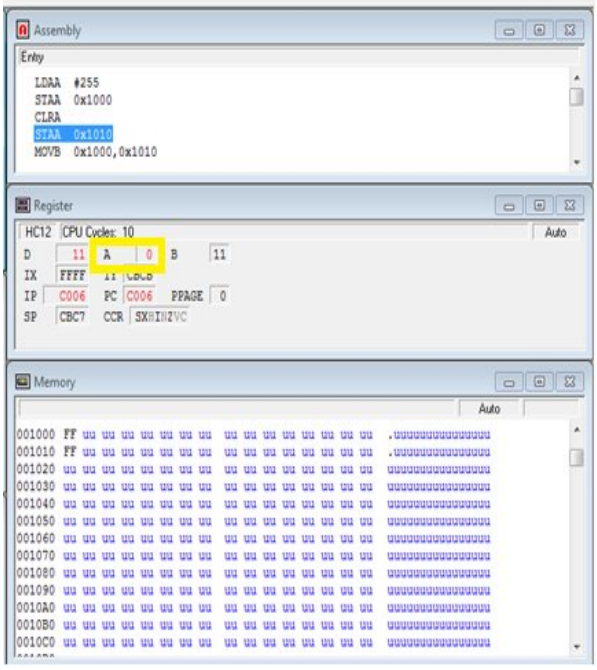
Pre-Lab Tasks

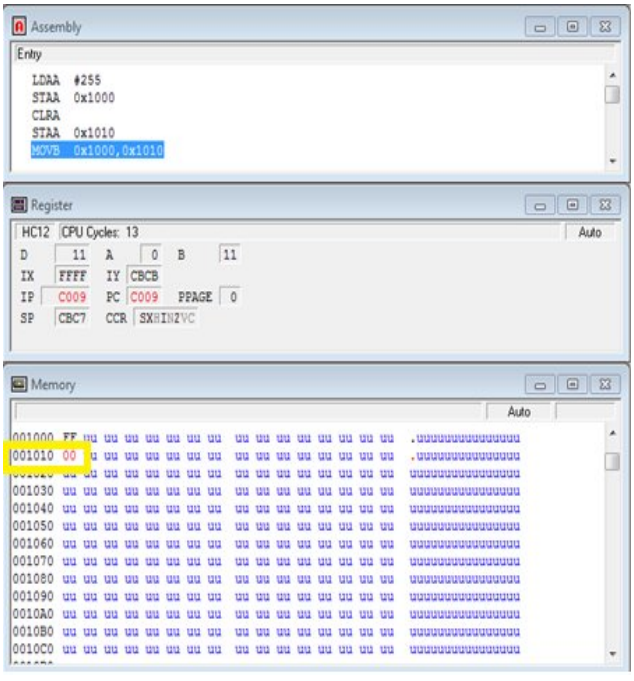
Task 2

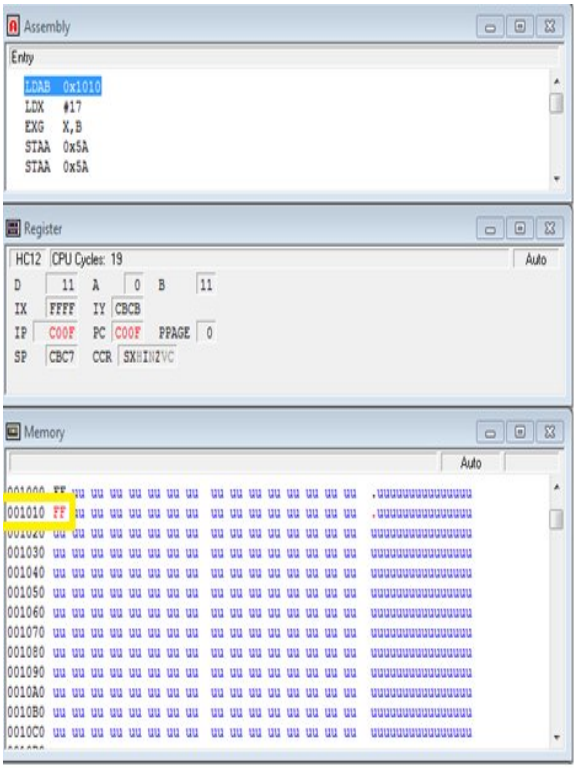
In this task, we are asked to simulate a program line by line using the Code Warrior IDE software in chip simulation mode and note down the changes that occur to the contents of the memory and registers after each instruction is executed. Below are the snaps showing the changes.

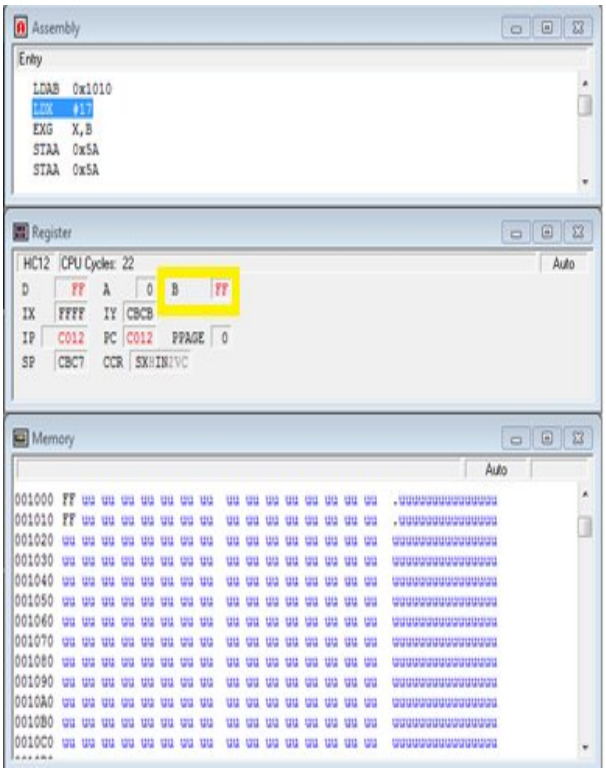


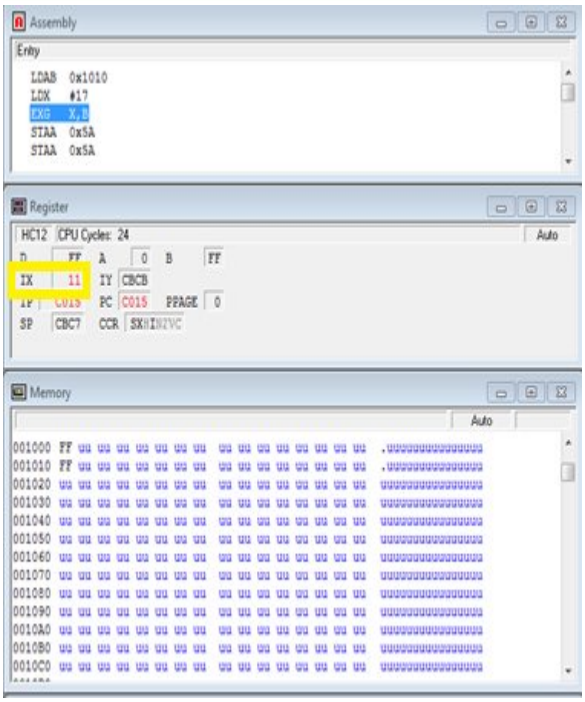







Assembly
Entry
LDAB 0x1010
LDX #17
EXG X,B
STAA 0x5A
STAA 0x5A
Register
HC12 CPU Cycles: 19
Auto
D 11 A 0 B 11
IX FFFF IY CBCB
IP C00F PC C00F PPAGE 0
SP CBC7 CCR SXHINZVC
Memory
Auto
001000 FF uu uu uu uu uu uu uu uu uu uu uu uu uu uu .uuuuuuuuuuuuuu
001010 FF uu uu uu uu uu uu uu uu uu uu uu uu uu uu .uuuuuuuuuuuuuu
001020 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001030 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001040 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001050 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001060 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001070 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001080 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001090 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010A0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010B0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010C0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu


Assembly
Entry
LDAB 0x1010
LDX #17
EXG X,B
STAA 0x5A
STAA 0x5A
Register
HC12 CPU Cycles: 22
Auto
D FF A 0 B FF
IX FFFF IY CBCB
IP C012 PC C012 PPAGE 0
SP CBC7 CCR SXHINZVC
Memory
Auto
001000 FF uu uu uu uu uu uu uu uu uu uu uu uu uu uu .uuuuuuuuuuuuuu
001010 FF uu uu uu uu uu uu uu uu uu uu uu uu uu uu .uuuuuuuuuuuuuu
001020 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001030 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001040 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001050 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001060 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001070 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001080 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001090 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010A0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010B0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010C0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu


Assembly
Entry
LDAB 0x1010
LDX #17
EXG X,B
STAA 0x5A
STAA 0x5A
Register
HC12 CPU Cycles: 24
Auto
D FF A 0 B FF
IX 11 IY CBCB
IP C015 PC C015 PPAGE 0
SP CBC7 CCR SXHINZVC
Memory
Auto
001000 FF uu uu uu uu uu uu uu uu uu uu uu uu uu uu .uuuuuuuuuuuuuu
001010 FF uu uu uu uu uu uu uu uu uu uu uu uu uu uu .uuuuuuuuuuuuuu
001020 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001030 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001040 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001050 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001060 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001070 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001080 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001090 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010A0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010B0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010C0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu

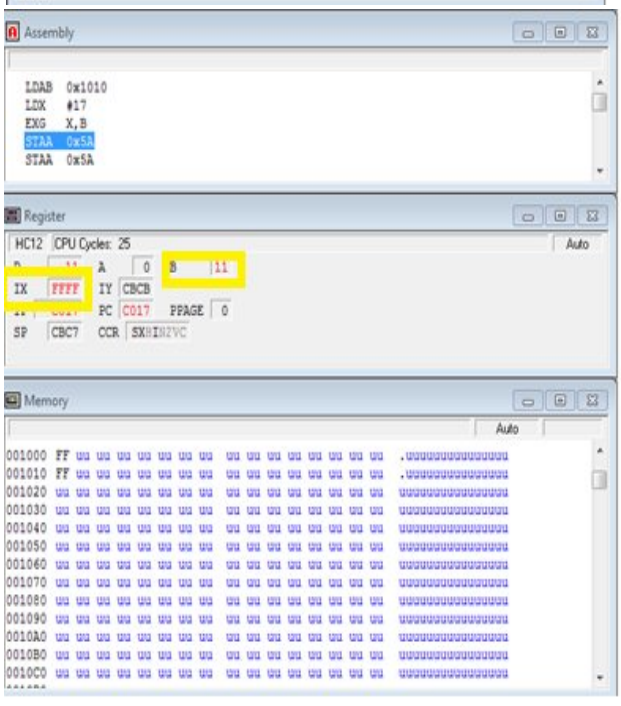

Assembly
LDAB 0x1010
LDX #17
EXG X,B
STAA 0x5A
STAA 0x5A
Register
HC12 CPU Cycles: 25
Auto
A 0 B 11
IX FFFF IY CBCB
PC C017 PPAGE 0
SP CBC7 CCR SXHINZVC
Memory
Auto
001000 FF uu uu uu uu uu uu uu uu uu uu uu uu uu uu .uuuuuuuuuuuuuu
001010 FF uu uu uu uu uu uu uu uu uu uu uu uu uu uu .uuuuuuuuuuuuuu
001020 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001030 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001040 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001050 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001060 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001070 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001080 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
001090 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010A0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010B0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu
0010C0 uu uu uu uu uu uu uu uu uu uu uu uu uu uu uu uuuuuuuuuuuuuu

Table 1 Program Step by Step Execution

Instruction	x1000	x1010	Accumulator A	Accumulator B	Index Register X
LDAA #$FF	**FF**		**FF**		
STAA $1000	**FF**		**FF**		
CLRA	**FF**	**FF**	**0**		
STAA $1010	**FF**	**00**	**0**		
MOVB $1000,$1010	**FF**	**FF**	**0**	**11**	
LDAB $1010	**FF**	**FF**	**0**	**FF**	
LDX #$0011	**FF**	**FF**	**0**	**FF**	**11**
EXG X,B	**FF**	**FF**	**0**	**11**	**FF**

Task 3

Identify the addressing mode of the following instructions

a) LDAA #$FF

Immediate Addressing

Operation: load the immediate hexadecimal number FF into 8 bit accumulator a.

b) CLRA

Inherent Addressing

Operation: clears accumulator a by setting its content to 0.

c) LDAB $1000

Extended addressing

Operation: load the contents of long address 1000 into accumulator b.

Pre-Lab Questions

1.How much RAM does your microcontroller have? How much EEPROM and how much Flash? Does it have any other kind of memory?

RAM: 12Kbyte

EEPROM: 4Kbyte

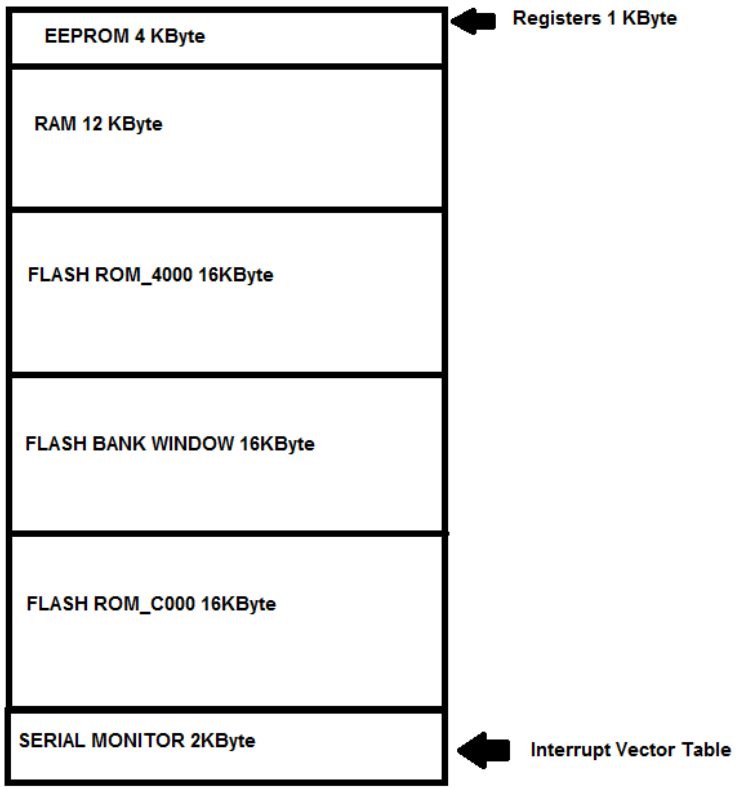
Registers: within the first 1 Kbyte of EEPROM

FLASH ROOM_4000 : 16 Kbyte

FLASH BANK WINDOW: 16Kbyte

HCS12 SERIAL MONITOR(VECTORS) : 2Kbyte

2.Draw the memory map of your microcontroller.



3. What registers does your microcontroller have?

Accumulators A, B, and D: A & B accumulators represent the upper and lower content of accumulator D which means A & B are each 8 bit and D is 16 bits. These are usually used to hold data resulting from ALU operations.

Index Registers X and Y: Used for indexed addressing modes usually as a base register.

Program Counter: The program counter shows the address of the next instruction to be executed and sometimes used as a base register

Condition Code Register: Register that is used during operations such as ALU operations that can be set or reset. They consist of: carry C, overflow V, Zero Z, Sign N and Half carry H, Interrup I XIRQ_L interrupt bit X and disable bit S.