KHALIFA UNIVERSITY

Module Name: Microprocessor Systems Laboratory
Module Code: ELCE333

Laboratory Experiment No. 2

Expriment Tite: Development & Testing of HCS12 Programs Using Branching and Loops
Pre-lab Report

Group Members

Name: Afra Bin Fares ,ID#:100033139

Name: Alya Humaid AlAlili ,ID#100037087

Name:Anoud Alshamsi ,ID#100035514

Instructors

Mohammed Ali Saif Al Zaabi

Mahmoud Khonji

Spring 2015

1. Introduction:

In this prelab, branch instructions are performed to test logical conditions where it modifies the address pointer depending on the result of the branch condition, where it either goes through the branch again or continue to the next instruction.

Whereas a loop is an instruction that is repeated until a condition is reached. an infinite loop is a loop without an exit, it continues on looping until the OS halts it by recognizing an error.

1.1 Aim:

This prefab aim is to be more familiar in the design, development and testing of HCS12 assembly programs with conditional branching .

1.2 Objectives:

1. Apply branch instruction to execute branching.

2. learn the difference between the branching types

2. Download, run, and test code on a Dragon Plus Trainer board.

2.Pre-lab design and results

2.1 Pre-Lab Tasks

In the pre-lab it is required to write a set of instructions to compare the contents of Accumulator A and Accumulator B. this program should assign the highest value in the memory location $1000.

i. Load Acc A = #$10 and Acc B = #$20. Use BGT or BLE branch instructions.

```
; Include derivative-specific definitions
    INCLUDE 'derivative.inc'
    XDEF Entry
MAX   EQU $1000
    ORG $4000
Entry:
    CLRA
    CLRB
    LDAA #$10 ; ($10)--> A
    LDAB #$20 ; ($10)--> A
    CBA       ; compare A to B
    BGT  YES  ; branch if A is greater than B
    STAB MAX  ; set value of B to maximum
    BRA Exit  ; exit the program
YES   STAA MAX
Exit  BRA Exit
```

ii. Load Acc A = #$93 and Acc B = #$56. Use BMI or BPL branch instructions.

```
; Include derivative-specific definitions
    INCLUDE 'derivative.inc'
    XDEF Entry
MAX   EQU $1000
  ORG $4000
Entry:
    CLRA
    CLRB
    LDAA #$93 ; ($93)--> A
    LDAB #$56 ; ($56)--> B
    CBA       ; compare A to B
    BMI  YES  ; branch if minus
    STAA MAX  ; set value of A to maximum
    BRA Exit  ; exit the program
YES   STAB MAX
Exit  BRA Exit
```

iii. Load Acc A = #$85 and Acc B = #$92. Use BCC or BCS branch instructions.

```
; Include derivative-specific definitions
    INCLUDE 'derivative.inc'
    XDEF Entry
MAX   EQU $1000

    ORG $4000
Entry:
    CLRA
    CLRB
    LDAA #$85 ; ($85)--> A
    LDAB #$92 ; ($92)--> B
    CBA       ; compare A to B
    BCC  YES  ; branch if caryy clear
    STAB MAX  ; set value of b to maximum
    BRA Exit  ; exit the program
YES   STAA MAX
Exit  BRA Exit
```

2.2 Pre-Lab Questions:

1. What is the value of the operand register (i.e., IX, IY, and Acc A) after the execution of each of the following individual instructions? Assume LOC is the label on memory location $1000 and the contents of memory location $1000 is $15 and $1001 is $24.

Code:

```
; Include derivative-specific definitions
    INCLUDE 'derivative.inc'

    XDEF Entry

LOC1   EQU $1000
LOC2   EQU $1001

    ORG $4000

Entry:
    CLRA
    CLRB
    LDAA #$15 ; ($15)--> A
    LDAB #$24 ; ($24)--> B
    STAA LOC1
    STAB LOC2
    LDX # LOC1
    LDX LOC1
    LDY #(LOC1-1)
    LDAA (LOC1+1)
Here  JMP Here
```

Instructions	IX	IY	Acc A
LDX # LOC1	1000	FFF	15
LDX LOC1	1524	FFF	15
LDY #(LOC1-1)	1524	FFF	15
LDAA (LOC1+1)	1524	FFF	24