MICROPROCESSORS & INTERFACING (ENSC332)

LAB 2: ASSEMBLY LANGUAGE PROGRAMMING: BRANCH, CALL, AND TIME DELAY LOOPS

OBJECTIVES

To create assembly language programs to demonstrate branch, call, and time delay loops

Required Components:

- CodeWarrior software
- Dragon12 Trainer board for sending outputs to LEDs.

PRELAB ASSIGNMENT

Read chapter 3 of textbook and prepare programs for activities below.

NOTE: In the lab activities below you may send your outputs to LEDs connected to PORT B of the HCS12 MCU on the Dragon12 Trainer instead of storing the result in a memory location. Each port B line on the trainer is connected to a LED. In order to turn on port B LEDs, the PJ1 (pin 21 of MC9S12DG256) must be programmed as output and set for logic zero as follows:

```
BSET  DDRJ, %00000010     ;set PJ1 as output
BCLR  PTJ, %00000010    ;clear PJ1
```

Also, PORTB has to be programmed as an output as follows:

```
MOVB  #$FF, DDRB        ; make port B an output port
BSET  DDRT, %00100000   ; make port T bit 5 output  (connected to speaker)
```

ACTIVITY 1

Write and assemble a program to clear register A, then add 3 to A fifteen times, and store the result in RAM location of your choice (e.g., $3000). Using the simulator, single-step through the program and examine A and memory location where you store the result.

ACTIVITY 2

Write and assemble a subroutine that can handle more than 16 million iterations. Then using the simulator test the program for a small number of iterations (e.g., 8 iterations).

Perform single stepping and observe how the stack pointer and contents of the stack change as you go through the program. Specify the content of Program Counter before subroutine call and after RTS is executed.

ACTIVITY 3

Write a program to count up twice from $0 to $A and send the count to accumulator B. Use the subroutine for each count and a short time delay loop in between the calls. Implement the delay routine using the program developed in Activity 2 above.