|  |
| --- |
| ; variable/data section  MY\_ZEROPAGE: SECTION SHORT ; Insert here your data definition  Counter: DS.B 1  TIMER\_STATUS: EQU $30  TIMER\_MAXI\_COUNT\_UPPER: EQU $33  TIMER\_MAXI\_COUNT\_LOWER: EQU $34  ; code section  MyCode: SECTION  main:  Entry:  LDHX #\_\_SEG\_END\_SSTACK  TXS  CLI  CLRA  STA $03  LDA #$FF  STA $01  STA $43  mainLoop:  CLRA  LDA $00  NSA  STA $40    CLRA  LDA #$02  PSHA  LDA #$80  PSHA  JSR delay |

Lab 5

# Purpose

The purpose of this lab is to concentrate on learning how to write modular assembly code.

# Objective

This lab gives you:

* More experience with the 6808 development board, environment, and lifecycle
* Assembly programming exposure to timing delays and sub routines

# Procedure

1. Open up the CodeWarrior IDE and copy the code from figure. In.
2. Then, using your delay code from last week, copy past code into your new codebase and run It on the 6808.
3. Observe the change in your lab book. How does the introduction of this 5 second delay affect the program’s ability for you to enter a bit pattern on the buttons and for it to appear on the leds.

|  |
| --- |
| JMP mainLoop  delay:  CLRA  LDA #%00010111  STA TIMER\_STATUS  LDA 3,SP  STA TIMER\_MAXI\_COUNT\_LOWER    LDA 4, SP  STA TIMER\_MAXI\_COUNT\_UPPER  checkLoop:  LDA TIMER\_STATUS  AND #$80  BEQ checkLoop  RTS |

Result: Once the delay code had been added, there is a 5 second delay in between button readings so you have to hold down the buttons a long time for there to be any effect. Once the leds have come on however, you can let go and they will stay on for whatever is left of the 5 seconds.

1. Next, rewrite your code so that it counts up from 0 to 8 in increments of one and display it in binary counting order on the leds provided by the 6808 development board. Also, between each increment, include a 5 second delay so that you can see the output. Show Jane when you’re complete

Result: See figure.