

Lab # 8

Introduction to the MicroStamp11 and 68HC11 Assembly Language Programming

A. Objectives

The objectives of this laboratory are introduce the student to:

- assembly language programming
- MGTEK MiniIDE assembler
- MicroLoad downloader
- MicroStamp11 68HC11-based microcontroller
- Building 7-segment display circuits and controlling them with a microcontroller programmed using assembly language

B. Materials

Breadboard	MicroLoad downloader
5V Power Supply	Common-anode 7-segment display
MGTEK MiniIDE assembler	Seven 220 or 330 ohm resistors

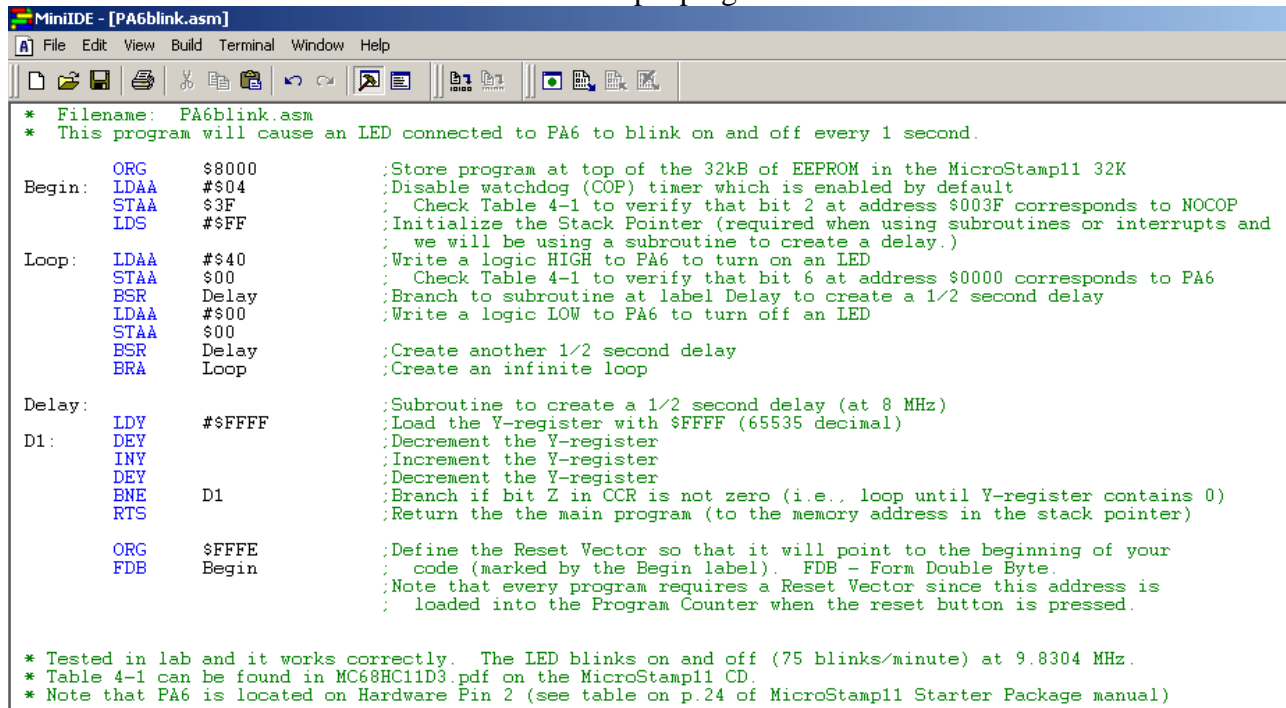
C. Reference

Refer to the following items (available on the instructor's web page):

- *Lab #8 Lecture Notes (PowerPoint)*
- *Example: Mini IDE Assembler and Wookie Simulator*
- *Classroom Lecture Notes on Assembly Language Programming (PowerPoint)*
- *Example: Mini IDE Assembler and Wookie Simulator (pp. 11-18)*

D. Introduction

Refer to the references above and to the sample program below.



```
MiniIDE - [PA6blink.asm]
A File Edit View Build Terminal Window Help
* Filename: PA6blink.asm
* This program will cause an LED connected to PA6 to blink on and off every 1 second.

Begin:   ORG     $8000           ;Store program at top of the 32kB of EEPROM in the MicroStamp11 32K
        LDAA   #$04           ;Disable watchdog (COP) timer which is enabled by default
        STAA  $3F             ; Check Table 4-1 to verify that bit 2 at address $003F corresponds to NOCOP
        LDS   #$FF            ;Initialize the Stack Pointer (required when using subroutines or interrupts and
                                ; we will be using a subroutine to create a delay.)
Loop:    LDAA   #$40           ;Write a logic HIGH to PA6 to turn on an LED
        STAA  $00             ; Check Table 4-1 to verify that bit 6 at address $0000 corresponds to PA6
        BSR   Delay          ;Branch to subroutine at label Delay to create a 1/2 second delay
        LDAA  #$00           ;Write a logic LOW to PA6 to turn off an LED
        STAA  $00
        BSR   Delay          ;Create another 1/2 second delay
        BRA   Loop           ;Create an infinite loop

Delay:   LDY    #$FFFF        ;Subroutine to create a 1/2 second delay (at 8 MHz)
D1:      DEY                    ;Load the Y-register with $FFFF (65535 decimal)
        DEY                    ;Decrement the Y-register
        INY                    ;Increment the Y-register
        DEY                    ;Decrement the Y-register
        BNE   D1              ;Branch if bit Z in CCR is not zero (i.e., loop until Y-register contains 0)
        RTS                   ;Return the the main program (to the memory address in the stack pointer)

        ORG     $FFFE        ;Define the Reset Vector so that it will point to the beginning of your
        FDB   Begin         ; code (marked by the Begin label). FDB - Form Double Byte.
                                ;Note that every program requires a Reset Vector since this address is
                                ; loaded into the Program Counter when the reset button is pressed.

* Tested in lab and it works correctly. The LED blinks on and off (75 blinks/minute) at 9.8304 MHz.
* Table 4-1 can be found in MC68HC11D3.pdf on the MicroStamp11 CD.
* Note that PA6 is located on Hardware Pin 2 (see table on p.24 of MicroStamp11 Starter Package manual)
```

E. Preliminary Work

1. **Program #1:** Program #1 is an assembly language program that will cause an LED connected to PA6 to blink ON and OFF once per second. This program has already been provided above. You can download it from the instructor's web site (PA6blink.asm) or enter it into MiniIDE or Notepad and save it as an asm file.
2. **Logic Diagram for Circuit #1:** Draw a logic diagram including the MicroStamp11 and an LED (and current-limiting resistor) connected to PA6. Program #1 will operate this circuit.
3. **Program #2:** Write an assembly language program that will display all of the digits in your EmplID (in order) on a common-anode 7-segment display and repeat the sequence of digits indefinitely. Each digit should appear for about 1 second.
4. **State Diagram for Program #2:** Draw the state diagram for Program #2 above.
5. **Logic Diagram for Circuit #2:** Draw a logic diagram including the MicroStamp11 and a common-anode 7-segment display (and the current-limiting resistors). Label the function and number for all pins used on the MicroStamp11 and the 7-segment display.
6. **Simulation for Program #2:** Use Wookie to simulate Program #2. In particular,
 - Refer to the handout: Example: Mini IDE Assembler and Wookie Simulator (pp. 11-18)
 - Open the Code View, MCU, and Memory Watch windows.
 - Add names and address to the Memory Watch window for PortA, PortD, and DDRD.
 - Add a breakpoint to stop the simulation each digit in your sequence has been reached and capture the screen (so if you used your 9-digit SSN, include 9 screen captures). On each screen, explain how the PortA and PortD values indicate the correct digit.

F. Laboratory Work

1. **Program/Circuit #1**
 - A) Use MiniIDE to assemble program #1. Be sure that there are no errors.
 - B) Build Circuit #1.
 - C) Use MicroLoad to download the S19 file into the MicroStamp11
 - D) Run Program #1. The LED should blink ON and OFF about 1 time per second.
 - E) Count and record the number of blinks in 60 seconds. If the program is off by more than 2 seconds, modify the delay subroutine in the asm file to correct the delay and record the modified count.
 - F) Demonstrate the proper operation of the circuit to the instructor.
 - G) For your report print the original asm file, the modified asm file, and the modified LST file.
 - H) Discuss the performance for Program/Circuit #1.
2. **Program/Circuit #2**
 - A) Use MiniIDE to assemble program #1. Be sure that there are no errors.
 - B) Build Circuit #2.
 - C) Use MicroLoad to download the S19 file into the MicroStamp11
 - D) Run Program #2. Verify that the correct sequence is displayed.
 - E) Record and discuss any corrections that you made to your program.
 - F) Demonstrate the proper operation of the circuit to the instructor.
 - G) For your report print the asm and LST files.
 - H) Draw a memory map for Circuit #2. Show the exact addresses where Program #2 is stored on the memory map (check the range of PC values in the LST file).
 - I) Discuss the performance for Program/Circuit #2.