**Introduction**

In this lab we had to wire and program a 7 segment display to the 68HC11 microcontroller. After wiring we needed to find the binary values for each ASCII character based on what is needed to turn on the appropriate LEDs in the display. After that we had to write a program that reads the character from the monitor and display it taking into consideration many condition rules.

**Preliminary Work**

I wired the circuit by connecting the ports to the 74LS06 and that to the 7 segment display. The following wire list better depicts how it was done.

**Wire List**

U1, U2 = SN74LS06

U3 = 7 segment display

GND 🡪 U1-7, U2-7, U3-3

VCC 🡪 U1-14, U2-14

PortB0 🡪 U2-1

PortB1 🡪 U1-3

PortB2 🡪 U1-5

PortB3🡪 U1-13

PortB4 🡪 U2-1

PortB5 🡪 U2-3

PortB6 🡪 U2-5

PortB7 🡪 U2-13

U1-2 🡪 U3-1 (a)

U1-4 🡪 U3-2 (b)

U1-6 🡪 U3-5 (c)

U1-14 🡪 U3-6 (d)

U2-2 🡪 U3-8 (e)

U2-4 🡪 U3-10 (f)

U2-6 🡪 U3-9 (g)

U2-14 🡪 U3-7 (dp)

I tested the circuit by memory modifying port B (mm 1004, c0), but I was not able to get anything to display. I tried many different hex codes, but nothing worked. I double checked the circuit, but everything seemed to be in order. I continued with the homework anyhow.

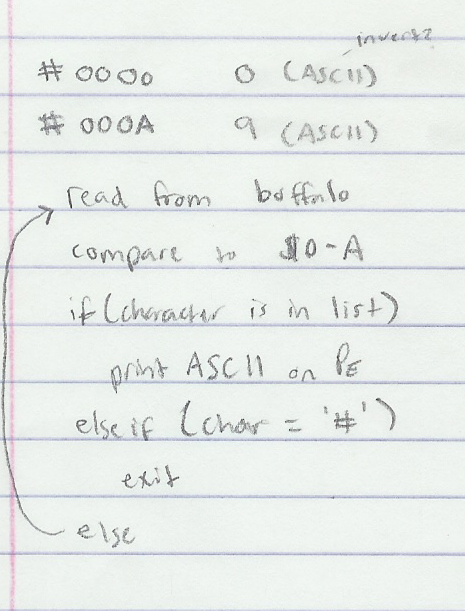
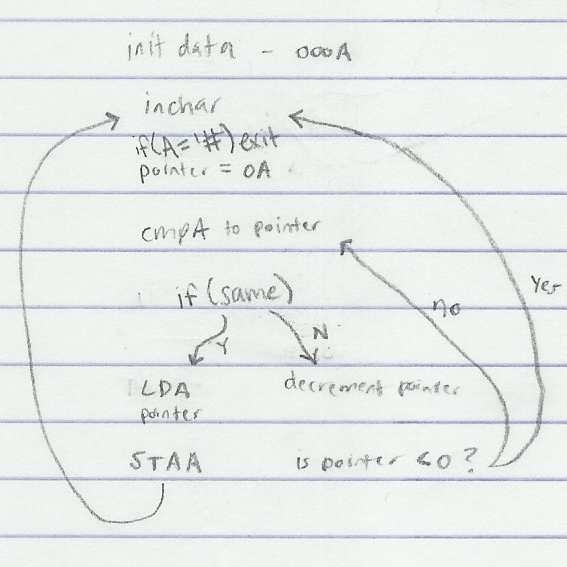
After the circuit was completed I designed the LED displays by using binary values and converted them by using provided converter program and an ASCII table. It’s important to note that the binary values are associated in the following order: Dgfedcba.

**Table 1. LED Display Conversions**

|  |  |  |  |
| --- | --- | --- | --- |
| **LED Display** | **ASCII equivalent** | **Port B(7:0) binary** | **Port B (Hex)** |
| **0** | $30 | 11000000 | $C0 |
| **1** | $31 | 11001111 | $CF |
| **2** | $32 | 10010010 | $92 |
| **3** | $33 | 10000110 | $86 |
| **4** | $34 | 10001101 | $8D |
| **5** | $35 | 10100100 | $A4 |
| **6** | $36 | 10100000 | $A0 |
| **7** | $37 | 11001110 | $CE |
| **8** | $38 | 10000000 | $80 |
| **9** | $39 | 10001100 | $8C |
| **0** | $30 | 11000000 | $C0 |
| **A** | $41 | 10001000 | $88 |
| **b** | $62 | 10100001 | $A1 |
| **C** | $43 | 11110000 | $F0 |
| **d** | $64 | 10000011 | $83 |
| **E** | $45 | 10110000 | $B0 |
| **F** | $46 | 10111000 | $B8 |
| **g** | $67 | 10000100 | $84 |
| **H** | $48 | 10001001 | $89 |
| **P** | $50 | 10011000 | $98 |

Finally I started to begin programming. A flow chart was created in class and translated into pseudo code as seen below.

**Flow Chart Pseudo code**



A lot of the programming was done in class. I just had to add a way to associate ASCII code with Port B hex code shown in table 1. Switch cases didn’t seem intuitive in assembly, so I manipulating another data source somehow would work. I did not finish the program.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 2.  Recording Single Step Results**   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | PC | Y | X | A | B | CCR | SP | Char | 7 | | 0100 |  |  |  |  |  |  | 0 |  | | 0102 |  |  |  |  |  |  | 1 |  | | 0102 |  |  |  |  |  |  | 2 |  | | 0103 |  |  |  |  |  |  | 3 |  | | 0107 |  |  |  |  |  |  | 4 |  | | 0109 |  |  |  |  |  |  | 5 |  | | 010B |  |  |  |  |  |  | 6 |  | | 010D |  |  |  |  |  |  | 7 |  | | 010F |  |  |  |  |  |  | 8 |  | |

**Conclusion**

I started working on this homework too late which left space for errors or time for questions.