Homework 1

Jennifer Du

10.22.2016

Problem 5: Target Board Familiarization

Demoed: yes

Keypad:

When I was looking at memory location 080H:
The values shown when pressing keys in the first row were: 7f7e, 7f7d, 7f7b, and 7f77. You can create values of 7f70 through 7f7e, with the first button as 1, 2nd as 2, 3rd as 4, 4th as 8 for last digits. (so if you don't press the first one but press all the rest, then you will get 7f71). Each key in order from left to right was essentially subtracting 2^n from the hex value 7f7f, if the keys were numbered 0, 1, 2, and 3 in that order. This row of keys sets the last digit of the value that we see in the debugger.

Values shown when pressing keys in the second row: 7e7f, 7d7f, 7b7f, and 777f. You can create 707f through 7e7f, with the first button as 1, 2 as 2, 3 as 4, and 4 as 8, with the same basic idea as it was in the first row. Pressing this row of keys sets the second digit.

When pressing keys in the third and fourth rows, the only value I got was 7f7f.

When I moved to locations 081H, 082H, and 083H, I saw that the "rules" of each row transferred down 1-3 rows from what it was like at location 080H.

Display:

Changing values of 00H gives you changes in the first digit. Setting the value equal to 1, 2, 4, 8, and subsequent values of 2, up to 128, will highlight the outline of the 0 (when value is 1, 2, 4, 8, 16, 32). A value of 64 will highlight the horizontal line in the middle. A value of 128 will light up the dot. 255 is the maximum value, and that lights up everything. Any value in between will light up whichever branches whose corresponding numbers add up to that value. Setting the value to 0 will turn it off.

To change which digit in the display you want to change, use 00H for the first digit, 01H for the second digit, and so on, until 07H, the last digit.

Problem 6: Basic Debugger Commands

For each of these interrupts, I looked at the vector interrupt table, and viewed the memory address stored there. This was the address of the first instruction to be executed when each of these interrupts happen. Then, I disassembled the code at those locations.

NMI (Non-Maskable Interrupt):

```
    Address of first instruction: f800:753c
    The first three instructions: PUSH  BP
                                  MOV   BP,SP
                                  PUSH  AX
```

```
        A breakpoint (INT 3 input):
            Address of first instruction: 2000:0004
            The first three instructions: CALL  e8ef
                                          CALL  e8f2
                                          CALL  e8f5

        A single-step interrupt:
            Address of first instruction: f800:75fe
            The first three instructions: NOP
                                          NOP
                                          JMP   SHORT 7510
```