Week 5 | Lab Assignment | I/O Devices, Sequential Logic vs. Index Addressing

September 26, 2023

Timothy McNamara

**Objectives**

For this lab, students are expected to code the eight LEDs on the department provided trainer board utilizing the CodeWarrior software. The way we code to them is by writing not only to the address of the LEDs but also to the Data Directional Register. (DDRc is what we refer to it as) Students are to hard code for the first part of the Lab and then take advantage of index addressing and compare the two types of coding. Prior to the completion of the lab, there is a pre-lab in which students are first introduced to sequential logic coding and are to apply critical thinking to understand this week’s lab material.

**Procedure**

The lab has a simple flow to it. First, we come across a table that is bit mapping the eight LEDs. Using the asterisk ( \* ) symbol to dictate when the LEDs are shown to be on or off, we are able to come up with hex values that correspond with matching eight-bit binary values. So, we fill out the table with the hex values. We design a flow chart to mimic the process of going through each hex value to display those on the eight LEDs and then we use sequential logic to code them. We then go onto the second part of the lab. This time we bit map another sequence table that is mimicking a ford mustang 2019 taillights. Once we have it bit mapped, we find the hex values and create pseudo code all over again. However, instead of coding with sequential logic, we code using index addressing. We declare the size of an array and every single hex value in the bit mapped sequence table in the constants section. Then in the data section we added the output of the LEDs and the LEDs data directional register as well. We loaded all of the DDRc with ones so that every single LED is an output. We wrote out our code and created our loops and utilized the x register. Once we were done, we completed the lab.

**Analysis (Answers to Questions)**

**Part I**

3c.

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

**Part II**

1c.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer code

Description automatically generated

1d.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |  | Hex Value |
|  |  |  |  |  |  |  |  | = | 00 |
|  |  | \* |  |  | \* |  |  |  | 24 |
|  | \* | \* |  |  | \* | \* |  | = | 66 |
| \* | \* | \* |  |  | \* | \* | \* | = | E7 |
|  |  |  |  |  |  |  |  | = | 00 |
|  |  |  |  |  | \* |  |  | = | 04 |
|  |  |  |  |  | \* | \* |  | = | 06 |
|  |  |  |  |  | \* | \* | \* | = | 07 |
|  |  |  |  |  |  |  |  | = | 00 |
|  |  | \* |  |  |  |  |  | = | 20 |
|  | \* | \* |  |  |  |  |  | = | 60 |
| \* | \* | \* |  |  |  |  |  | = | E0 |

**Questions**

1. Describe the sequence of the LED’s bit pattern in Part I.

The sequence of LED’s showed just three bits sweeping across the row of LED’s. But at the beginning of the row and the end of the row it went down to two LED’s. Once it got down to two LED’s, it started to sweep back and go to the other end.

1. Did the LED’s display the sequence as expected in Part I?

They sure did.

1. Does Part I program end after all data in the array is processed or does it repeat?

The program repeats itself by re-inputting hex values to the LED memory location. It doesn’t loop back to the beginning and re-add FF to the DDRc.

1. Explain what happens if you run your Part I program as opposed to stepping through it. Explain why this happens.

The program will run too fast and not be able to properly display what the output of the LED’s is showing. The clock speed or frequency of the chip is so high that it would loop faster than the LED’s can output a value. We would need to add delays to the code in order for the speed to slow down enough to actually show values to the LED’s.

**Conclusion**

This was a fun lab. It was a little bit hard to set up the trainer board for me as it was my first time using it. But, once I got my drivers set up and was able to communicate with the board, it was a cinch. I was able to easily see how both parts of the lab interacted with the trainer board and the eight LEDs. Coding was a completely different jump from the first part to the second part. Index addressing is extremely easy and makes the time to write the program a breeze compared to hard coding the sequential logic. I was extremely happy learning index addressing and how we can make something simple if we were to hard code it. I am very happy with my results.