**Objective:**

This lab you will learn to write your own programs to control the 8 LEDs on our trainer board. In your pre-lab work you use the skills and logic from previous lab assignments, to design, write and execute a sequential logic program to display specific light patterns on the 8-LEDs. For this lab assignment you design and write another sequential logic program and then your will re-design and re-write your sequential logic program such that the code is more efficient using list declaration and indexing addressing.

**Instructions:**

**Part I:**

**Bit Patterns and Sequential Programming**

Ok, now that you have an outline, for the pre-lab work, of how to design and write a sequential program to implement bit-pattern logic, next you will try it on your own. 1. The table below displays a bit pattern where the \* indicates 1 and blanks indicates 0. This bit pattern, when displayed in proper sequence to the 8 LEDs on our training board would mimic the front scanner light behavior of the fictional, robotic, 1982 Pontiac Firebird Trans Am T-top automobile called KITT. (From the Popular 80’s crime fighting TV series called **Knight Rider,** This crime fighting car was controlled by artificial intelligence controller called KITT.). [View front bumper light scanner](https://www.youtube.com/watch?v=54O_1mOab4Y)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |  | Hex Value |
|  |  |  |  |  |  | \* | \* | = | 03 |
|  |  |  |  |  | \* | \* | \* | = | 07 |
|  |  |  |  | \* | \* | \* |  | = | 0E |
|  |  |  | \* | \* | \* |  |  | = | 1C |
|  |  | \* | \* | \* |  |  |  | = | 38 |
|  | \* | \* | \* |  |  |  |  | = | 70 |
| \* | \* | \* |  |  |  |  |  | = | E0 |
| \* | \* |  |  |  |  |  |  | = | C0 |
| \* | \* | \* |  |  |  |  |  | = | E0 |
|  | \* | \* | \* |  |  |  |  | = | 70 |
|  |  | \* | \* | \* |  |  |  | = | 38 |
|  |  |  | \* | \* | \* |  |  | = | 1C |
|  |  |  |  | \* | \* | \* |  | = | 0E |
|  |  |  |  |  | \* | \* | \* | = | 07 |

1. In the table above, for each row, fill in the hex values.

1. Using the Program Development Cycle design worksheet, design a program that displays the “KITT” bit pattern hex values to the 8 LEDs on the training board. Program logic must use sequential logic for each value. Sequential Logic meaning no indexing, just one value at a time. Use the outline as exampled in the Pre-lab. You can find the Development Cycle design worksheet posted with this assignment or in the Canvas | Resources module or posted with this lab assignment.

* 1. Complete steps 1- 3 and then have your design worksheet checked by the instructor or TA. Check Point

* 1. After you have been checked for your design sheets continue to complete steps 4-6. Code, run, debug your program. Step through each instruction and ensure your program lights the LEDs in proper order.
  2. Get a screen capture of your code (make sure the capture shows your commented header) and include the capture in your word document report.

**Part II. Indexed Addressing**

1. Next you will make your code more efficient by replacing the sequential logic with index addressing logic.

**Read but Do not start coding**

*When you do start to code Part II: You will comment out Part I of your program*. **Program Description:** Design a program that reads the sequence data (provided in Part 1 of the lab) out of a list (array of data) and then writes each value to the 8 bits of the LEDs on our trainer board. This program should behave exactly as the Part I program, just more efficient code using a list and index addressing. Place the data in a list (array) as exampled in Lecture Notes, and use indexed addressing to read the data from the list (array).

* 1. Using the design worksheets: Complete steps 1- 3 and then have your design worksheet checked by the instructor or TA. Check Point Design sheets no program code yet!
  2. ONLY After you have been checked for your design sheets continue to complete steps 4-6 of the design worksheet process. o **When you test your code: Step through** your program and note the LED’s sequence. o Test your program by stepping through the code. Do not test by running the program…you must test by single stepping. o Once your program performs as expected have an Instructor or TA view your program. Check Point program code!
  3. Get a screen capture of your code (make sure the capture shows your commented header) and include the capture in your word document report.
  4. **Question**: How would you change the array to display the new Ford Mustang tail light sequence? (look up 2019 [Ford Mustang sequential tail lights)](https://www.hagerty.com/articles-videos/articles/2019/01/24/first-car-with-sequential-taillights) o Map the bit sequence in the following table as exampled in Pre-lab and Part I. Include this table in your report. Add more rows if needed.

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

**Questions**:

* 1. Describe the sequence of the LED’s bit pattern in Part I.
  2. Did the LED’s display the sequence as expected in Part I?
  3. Does Part I program end after all data in the array is processed or does it repeat?
  4. Explain what happens if you run your Part I program as opposed to stepping through it. Explain why this happens.

**Write: your report for this Canvas assignment:**

* Word document no more than a few pages, well formatted with: Your name, lab section #, lab assignment #, and date, (save your .docx as YourLastName\_Lab##). (where ## is the lab number)
* Include sections:
  + - Objective (in your own words), o Procedure (one short concise paragraph),
    - Answers to questions (clearly identify the question and your answer. *Ensure your answers are indicated in highlight, red, or some means to indicate the answer from the instructions*),
    - Conclusion: Summarize what you have learned and observed. Include discussion on indexed addressing. Do not be vague on this reflection as you must convey to me you understand what you have done and can provide, in detail, examples for your statements. Don’t just say I did this and then I did this….etc.. Use critical thinking when writing your thoughts and observations and provide specific examples. o Make sure to view lab assignment for additional report criteria instructions.

**Submit:** to this Canvas assignment by due date.

1. Your completed report with all completed sections and with
2. Your main.asm file completed, fully commented program (just the main.asm file) o Be sure your code is well organized, includes a commented header, and logical comments and each Part clearly noted.