Spring 2021
EEE212-03 Microprocessors
Project 1
12.11.2021 08:30 - 19.11.2021 12:20

In this lab, you are going to use 8051 board to simulate your code written using MCU 8051 IDE Software. You will practice how to utilize arithmetic instructions in complex problems and how to utilize the LCD screen to provide output for your code. This lab assignment has two parts, where the completion of the easier first part will provide hints on how to complete the harder second part. Please read the notes and the assignment requirements carefully since they are pretty important in terms of evaluation.

Important Notes:

- After you have completed your lab, you need to get a check from one of the lab assistants (not tutors). The check consists of explanation of the code and a small demonstration. Demonstration will be performed using **8051 board**, so we will not accept simulations via Proteus.
- This is an individual lab. You can cooperate but you have to write your **OWN** code. Any kind of plagiarism will not be tolerated. Codes will be compared manually by assistants and by Turnitin software after the lab.
- The deadline is strict. Submit your code before the deadline.
- You can get a check after the deadline if the queue for the check is long, so do not worry. If such a case occurs, you will get your check based on your latest submission to the Moodle. Therefore, do not try to change your code after you have submitted your code.
- Your board needs to be working properly by this time. Your grade will be mostly (almost all of it) based on your demo. Therefore, fix your board issues if you have some problems.

Part 1: Divisibility (25 pts)

In this section of the lab, you will have the MCU decide if a value is divisible by a second value. More importantly, you will familiarize yourself better with programming the LCD display.
Consecutively take two decimal inputs $x, y \in [0, 99] \cap \mathbb{Z}$ from the keypad and display them on the first line of the LCD, separated by the '|' operator and followed by a space character and question mark character after y. Then, press the D button to have the 8051 MCU decide whether x divides y, and display the result (as `TRUE` or `FALSE`) on the second line. In addition to the LCD display, have the P2.5 output port cleared if $x \nmid y$, or have it set if $x|y$.

Notes:

- While taking input, have each digit be displayed right when the corresponding digit is pressed on the keypad. The '|' character can either be displayed when the second button is pressed or when the third button is pressed. The question mark character (and the space character before it) must be displayed right after the fourth button is pressed.
- If one of the inputs is intended to be smaller than 10, use a leading zero. In other words, press 0 before entering your single-digit inputs.
- Verify the output taken from P2.5 via an oscilloscope.
- Your code should run indefinitely without reset. In other words, while the LCD is displaying the result of a previous query, the MCU should be able to accept new input from the keypad and have the LCD display and P2.5 update accordingly.
- The MCU does not need to interact with P2.5 while taking a new input.

Hints:

- Try to update the CY flag according to the success of the divisibility test, and attempt to have P2.5 monitor the CY flag whenever necessary. Such an approach will especially be useful in the next part.
- Do not forget about the possibility of special cases while devising your algorithm.
- Do not overthink in this part.

Part 2: Sample Application (75 pts)

In this section of the lab, you will have the MCU compute the number of positive integers smaller than N that are relatively prime (i.e. have a greatest common divisor of 1) with N where $N \in (0, 255] \cap \mathbb{N}$. A formula to calculate this value, represented by "$\phi(N)$", is given in Equation 1

$$\phi(N) = N\left(\frac{p_1 - 1}{p_1}\right)\left(\frac{p_2 - 1}{p_2}\right)\left(\frac{p_3 - 1}{p_3}\right)...\left(\frac{p_n - 1}{p_n}\right) \quad (1)$$

where n is the number of unique primes that divide N, and p_i ($\forall i \in \{1, 2, ..., n\}$) are the unique prime numbers that divide N.

Have the string '`PHI(`' be displayed on the first line on LCD before taking your input. Take a decimal number with at most three digits as your keypad input (N), and print it on the first line, following the aforementioned string. Then, press the D button on the keypad to close the bracket and compute the result of $\phi(N)$ using the formula given in Equation 1. Have the computed result be displayed on the second line of the LCD, preceded by an '`=`' character.

Notes:

- Your code should run indefinitely without reset. In other words, while the LCD is displaying the result of a previous query, the MCU should be able to accept new input from the keypad and have the LCD display accordingly.
- While taking input, have each digit be displayed right when the corresponding digit is pressed on the keypad. The closing paranthesis of '`PHI(___)`' should be displayed only when the D button is pressed.
- The primes smaller than 256 are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163 ,167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, and 251.
- During giving a check, you will not be asked to display any result for inputs outside the provided domain of $(0, 255] \cap \mathbb{N}$.
- For those who are curious, $\phi(N)$ is formally called *Euler's totient function*.

Hints:

- Even though the mathematical formula given in Equation 1 does technically work for it, treat $N = 1$ as a special case.
- $N \neq 0$
- Try to do all necessary divisions *before* any necessary multiplication.
- If you want to dispose of a value at the top of the stack without popping it onto its original location (possibly to push another value in its place), you can use `DEC SP` to do so.
- Try to completely avoid using the `SUBB` instruction. Use `DEC` instead.
- Try to make liberal use of `XCH A, B`.
- Try using different conditional jumps for different situations. It is entirely reasonable to never use the `DJNZ` instruction during the computation of $\phi(N)$.