# EE 212 Spring 24/25 Lab 2

Due Date: Monday, February 24th at 13:30

Please read the notes and the assignment requirements carefully since they are essential for evaluation.

## Assignment Requirements

- Your submission will be checked using the Proteus simulation software during lab hours. Thus, ensure your code works before the lab session.
- Lab assignment has two parts. Therefore, you should upload two different files, one for each part. Please upload your files in '.txt' format.
- The deadline is strict. Submit your code before the deadline. **You cannot change your uploaded codes during lab hours. You will show your demos based on your uploaded codes**.
- This is an individual assignment. You can cooperate but must submit your **OWN** code. plagiarism will **NOT** be tolerated. After the lab, the codes will be compared manually by assistants and by TURNITIN software.
- Use the keypad and LCD codes provided in Lab 1, and feel free to add them to your code. If you do not include these codes at the end of your files, the LCD and keypad will not function. Additionally, review the LCD and keypad functions provided in Tutorial 1 to utilize them effectively for receiving input from the keypad and displaying results on the LCD.

## 1 Base Conversion Program (50 pts)

In this part of the assignment, you will write a program that converts a given decimal (base-10) number into another base (2, 4, or 8).

Your program should take **two inputs** from the keypad in the the Proteus simulator:

1. The **number to be converted**, which must be an integer between 1 and 255 (inclusive).

2. The **target base**, which must be either 2, 4, or 8.

Your program should then output the representation of the given number in the selected base on LCD in the Proteus simulator. Result can be printed after inputs are taken and LCD is cleared. The output format should be as follows:

```
NUMBER = ...
BASE = ...
RESULT = ...
```

Moreover, your results may include trailing (or leading) zeros. Specifically, when converting a decimal number within the given range to binary, the binary representation may be up to 8 digits long. Similarly, a number converted to base 4 may be up to 4 digits long. Including trailing zeros is optional and will not result in any point deductions.

## 1.1 Examples

- NUMBER = 25
  BASE = 2
  RESULT = 11001 or RESULT = 0001 1001

- NUMBER = 9
  BASE = 4
  RESULT = 201 or RESULT = 0201

- NUMBER = 100
  BASE = 4
  RESULT = 1210

- NUMBER = 255
  BASE = 8
  RESULT = 377

## 1.2 Hints

- You need to perform a decimal-to-hexadecimal conversion when storing the decimal input number in the 8051 registers.

- You need to consider ASCII values when printing characters for LCD. For example, if you need to print the number `3`, you must convert it to its ASCII value, which is `33` in hexadecimal.

## 1.3 Grading of Part 1 (50 pts)

- Show the correct result properly for base 8 (8 pts)

- Show the correct result properly for base 4 (12 pts)

- Show the correct result properly for base 2 (25 pts)
- Answer interview questions properly (5 pts).

# 2 Calculation of $n$th Fibonacci Number (50 pts)

In this part, you will write a program for the 8051 microcontroller that calculates the $n$th Fibonacci number. The user will input the $n$ from the keypad in the Proteus simulator and the program will then calculate and output the $n$th Fibonacci numbers to an LCD display in the Proteus simulator. $n$ is in the range of 0 to 24, i.e. $n \in \{0, 1, \ldots, 24\}$ in its decimal format. The end point $n = 24$ is chosen so that operations could be done in two registers. The output format should be as follows:

```
n = ...
Fib(n) = "RESULT"
```

### 2.0.1 Fibonacci Sequence Calculation

The Fibonacci sequence is defined as:

$$F(0) = 0, \quad F(1) = 1, \quad F(n) = F(n-1) + F(n-2) \quad for \quad n \geq 2$$

The program will calculate the $n$th Fibonacci number recursively from the above equation and lookup tables should not be used.

## 2.1 Examples

- n = 4
  Fib(n) = 3
- n = 10
  Fib(n) = 55
- n = 20
  Fib(n) = 6765
- n = 0
  Fib(n) = 0
- n = 1
  Fib(n) = 1

## 2.2 Implementation

- You need to do addition operation in two different registers.
- You need to store the result in multiple registers.

- If you opt to show your results in hex format, that would be fine with little point deduction (look at the grading part).
- Do not use lookup tables!

### 2.3 Grading of Part 1 (50 pts)

- Show the correct result in its decimal form properly up to $n = 13$ (10 pts).
- Show the correct result in its decimal form up to properly for $n \in \{14, 15, \ldots, 24\}$ (30 pts). You may opt to show the results in its hexadecimal format in this case you get 15 pts from this part.
- Answer interview questions properly (10 pts).

### 2.4 Fibonnaci Numbers

| $n$ | Fibonacci($n$) | Hexadecimal |
|---|---|---|
| 0 | 0 | 0h |
| 1 | 1 | 1h |
| 2 | 1 | 1h |
| 3 | 2 | 2h |
| 4 | 3 | 3h |
| 5 | 5 | 5h |
| 6 | 8 | 8h |
| 7 | 13 | Dh |
| 8 | 21 | 15h |
| 9 | 34 | 22h |
| 10 | 55 | 37h |
| 11 | 89 | 59h |
| 12 | 144 | 90h |
| 13 | 233 | E9h |
| 14 | 377 | 179h |
| 15 | 610 | 262h |
| 16 | 987 | 3DBh |
| 17 | 1597 | 63Dh |
| 18 | 2584 | A18h |
| 19 | 4181 | 1055h |
| 20 | 6765 | 1A65h |
| 21 | 10946 | 2AD2h |
| 22 | 17711 | 452Fh |
| 23 | 28657 | 6FD1h |
| 24 | 46368 | B520h |

Table 1: Fibonacci numbers up to 24 with hexadecimal equivalents