Due on: November 18, 2021

Introduction to Microcomputers

Lab3

The goal of this lab is to emulate the looping constructs using PIC16F877A instructions.

Assignment

Fibonacci numbers are recursively defined as follows:

Fib(0) = 0;
Fib(1) = 1;
Fib(N) = Fib(N-1) + Fib(N-2) for N>=2

Iteratively, the following C code can be used to compute the N^{th} Fibonacci number:

```
uint8_t fib0 = 0;
uint8_t fib1 = 1;
uint8_t fib;
uint8_t i=2;
for (i=2; i<=N; i++){
   fib = fib0 + fib1;
   fib0 = fib1;
   fib1 = fib;
} //end-for
// When we come here, fib contains the Nth Fibonacci number
```

Here are the Fibonacci numbers from 2 to 13:

N	Fib(N) [Decimal]	Fib(N) [Hex]	Fib(N) [Binary]
2	1	0x01	00000001
3	2	0x02	00000010
4	3	0x03	00000011
5	5	0x05	00000101
6	8	0x08	00001000
7	13	0x0D	00001101
8	21	0x15	00010101
9	34	0x22	00100010
10	55	0x37	00110111
11	89	0x59	01011001
12	144	0x90	10010000
13	233	0xE9	11101001

Instead of computing just the Nth Fibonacci number and displaying the result on the LEDs connected to PORTD, you will display each and every Fibonacci number from 2 to 13 within the while loop on the LEDs, and wait for the user to press Button3 (RB3 on PICSIM) to move on to the next iteration of the loop. Also make a 250ms delay before checking if the button is pressed and moving on to the next iteration of the loop. Essentially, you will be implementing the following C code:

```
BANKSEL TRISB      ; Select the Bank where TRISB is located (Bank 1)
TRISB = 0xFF       ; Make all pins of PORTB as input pins
TRISD = 0x00       ; Make all pins of PORTD as output pins

BANKSEL PORTD      ; Select the Bank where PORTB is located (Bank 0)
CLRF    PORTD      ; Turn off all LEDs

uint8_t fib0 = 0;
uint8_t fib1 = 1;
uint8_t fib;
uint8_t i=2;
for (i=2; i<=N; i++){
   fib = fib0 + fib1;
   fib0 = fib1;
   fib1 = fib;

   PORTD = fib;                ; Display the current Fibonacci number on the LEDs
   DelayMs(250)                ; Wait for 250ms
   while (PORTB3 == 1) ;       ; Wait for Button3 (RB3) to be pressed
} //end-for
while (1);                     ; Infinite loop
```