# Sequential Logic Design with Verilog

## Introduction

In this lab you will display your BCIT ID, four digits at a time, on the 4-digit, 7-segment LED display.

Your circuit should display the last four digits of *your* BCIT ID when keypad button [1] is pressed and the first four digits when it is not. For example, if your BCIT ID is **A00123456** the display should show and 3456 when [1] is pressed and 0012 otherwise.

You will use the same components as in the previous lab, connected the same way.

## Multiplexed LED Displays

Review the schematic of the LED display in the previous lab. Note that the four digits on the LED display share the same segment (anode) connections. The seven segments on each digit have a common cathode connection. You can therefore only display one digit at a time. But cycling through the digits faster than the eye can perceive[1] makes it appear that all digits are displayed simultaneously.

## Design

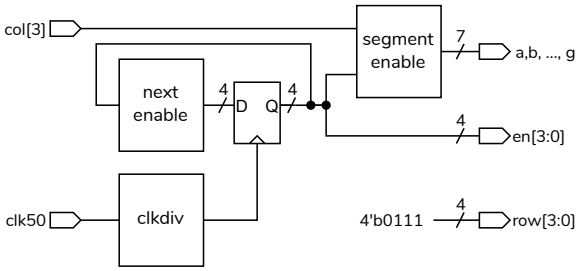
One possible solution is described below. It consists of two parts.

One part enables the digits one after the other, using a register whose value continuously cycles through the required values for the **en** output. For example, if you have a clock signal **clk** (see below):

```
logic [...] en ;    // active-low digit enable
 ...
always @(posedge clk)
   en <= ... ;      // next value of en
                    // (depends on current value)
```

A second part must set the segments (**a** through **g**) to the correct values. These will depend on the enabled digit and whether the key [1] is pressed or not (which you can determine by the **col** value). This part will be similar to the previous lab.

[1] The flicker fusion threshold.




## Component Connections

The CPLD board, keypad and LED display should be connected as in the previous lab.

## Procedure

Follow the general procedure in the Software Installation and Use document on the course website to create a project, compile it and configure the CPLD.

## Import Pin Assignments

You can import the pin assignments from the previous lab if you are not changing them (this is recommended). Select **Assignments** / **Import Assignments...** and select **lab1.qsf** from your previous lab's project folder as the file. Click on **Categories...**, un-check everything except **Pin and Location Assignments** and click on **OK**. Check that the pin assignments are correct. You may need to add the weak pull-up resistor assignments on the **col** input.[2]

## Clock

The board has a 50 MHz clock signal connected to pin 12 of the CPLD. This signal was not used in the previous lab although it was listed in the sample pin assignments and named **clk50**. Add this pin assignment if you don't have it already.

The LED circuits cannot respond fast enough if you use the 50 MHz clock directly so you will need to use a supplied "clock divider" module to create a 200 Hz clock from the 50 MHz clock. To do this, copy

[2] These don't seem to be imported.

the **clkdiv.sv** file from the course website to your project folder and add it to your project (**Project > Add/remove Files in Project**..). This module has one input, the 50 MHz clock and one output, a 200 Hz clock. You can instantiate it into your **lab2.sv** file with the following two lines:

```
logic clk ;
clkdiv c0 ( clk50, clk ) ;
```

You can then use the **clk** signal in your register descriptions:

```
always_ff @(posedge clk) en <= ...
```

## Lab Report

Submit the following to the appropriate Assignment folder on the course website:

1. A PDF document containing:
    - A listing of your Verilog code.
    - A screen capture of your compilation report.
2. If you were not able to demonstrate your solution to the lab instructor during your scheduled lab period, submit a video showing the keypad and the LED display as you push the 1 key.

Follow the *Report and Video Guidelines* and the *Coding Guidelines* documents on the course website.