COL215 – Digital Logic and System Design
Department of Computer Science & Engineering, IIT Delhi
Semester I, 2025-26
Lab Assignment 3

7-Segment Display

1 Introduction

In this assignment, we will capture and display a 4-digit decimal number on the 7-Segment Display devices of the Basys3 board. This will be done in 2 parts:

1. Capture and store the four digits using the input slider switches on the Basys3 board.
2. Display the four digits on the 7-Segment LED display devices of the board.

You are expected to reuse the Verilog module you developed for assignment 2 in this assignment.

2 Problem Description

You can reuse your knowledge of displaying a single digit from Assignment 2.

2.1 Capturing the four digits

Since only one set of input switches are available (SW0..SW9) to capture a single digit, we need to perform 4 different actions, in sequence, to store 4 different digits in our 4 variables. The four switches SW10..SW13 can be used to indicate which digit is being stored currently. See table below.

Slider switch turned ON	**Function**
SW13	Capture digit 3
SW12	Capture digit 2
SW11	Capture digit 1
SW10	Capture digit 0
SW9–SW0	Value '9'–'0' is entered for digit

A Verilog process construct of the following nature can be used to store the switch positions corresponding to a decimal digit 'sw[9:0]' in variable 'dataout0', on an enable condition, 'sw10', on the rising edge of a 'clk' signal (change from 0 to 1). 'clk' represents a *clock*, a periodic signal that stays high for some time and low for some time, with an associated frequency. Such a clock signal, with a fixed frequency, is already available on the Basys3 board and should be used in this assignment.

```
always @(posedge clk) begin
    if (sw10) begin
        dataout0[9:0] <= sw[9:0];   // Sample slider switches' positions when sw10 is high
    end
    // If sw10 is OFF, dataout0 holds its value
end
```

Note that with the above code, we repeatedly store the same value at several consecutive clock edges as long as the 'enable' signal, sw10 in the code above, is ON, but that is OK. We will retain the last value that was sampled. After sampling the value, sw10 should be turned OFF.

Repeat the above process for capturing all the 4 digits to be displayed.

2.2 Driving the displays

2.2.1 Seven segment display

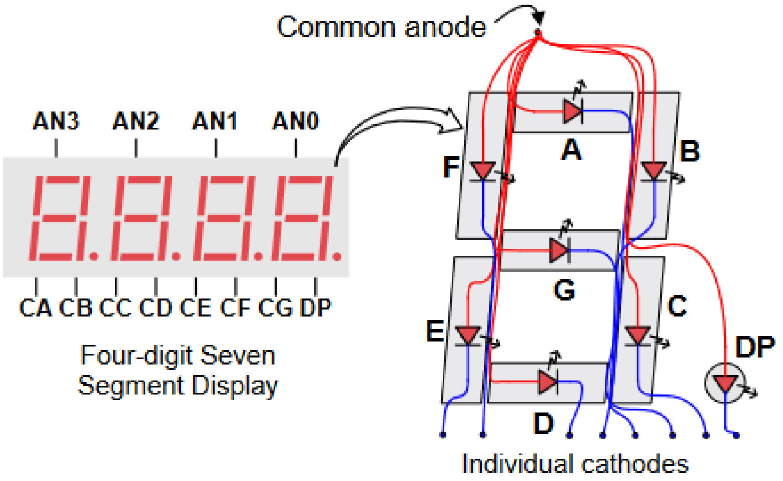

Figure 1: Pin details for 7 seven display on Basys 3 board

We have four 7-Segment Display Units (7SD-3, 7SD-2, 7SD-1, and 7SD-0) on Basys3 (illustrated in Figure 1) controlled by:

- four anode pins AN3, AN2, AN1, and AN0,
- seven cathode pins (CA, CB, CC, CD, CE, CF, and CG), and
- a decimal point pin (DP).

Since there is only one set of cathodes, we can display only one decimal digit on one of the 7SDs at any time, by selecting the cathode values and the appropriate anode.

Figure 1 also shows a zoomed-out view of one of the 7SDs. To display a decimal digit on one 7SD (say 7SD-2), we drive the anode AN2 LOW (this makes it ACTIVE), and set the values of the seven cathode pins (CA,...,CG) and DP appropriately (reuse the logic you developed in Assignment 2).

2.2.2 Displaying digits on all four 7SDs

To display a decimal digits on all four 7SDs, we need to periodically cycle through the displays on the individual 7SDs in sequence. Figure 2 shows the possible timing sequence of anode and cathode changes. The recommended 'refresh period' (repeating the cycle) is between 1-16 ms (frequency 60Hz-1kHz). (Why should any limit be there on the period, high or low?)

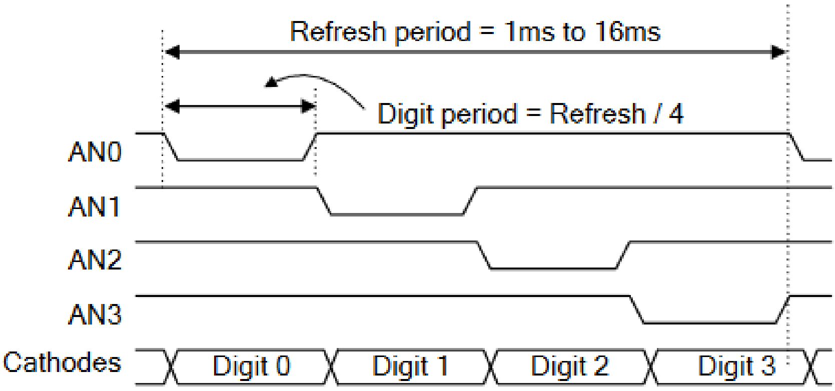

Figure 2: Timing details for 4 seven segment displays

A possible block diagram for the design could look like the one shown in Figure 3. You need to ensure that the refresh rate is appropriately controlled, i.e., the timing control circuit outputs remain stable for sufficient time for each 7SD. A modulo counter can be used to drive the 4 anodes each for sufficient time. An example configuration is shown below for driving each anode for 1ms.

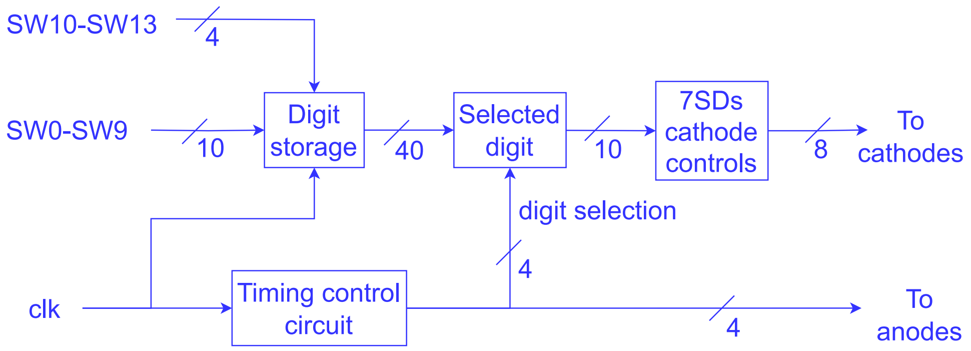

Figure 3: Block diagram for driving four 7 seven segment displays

NOTE: On Basys 3 board, anode/cathode are ACTIVE LOW pins, i.e., 0 = ACTIVE, 1 = INACTIVE. For details, refer to the Basys 3 reference manual. Section 8.1 Seven-Segment Display.

```
always @(posedge clk) begin // Increment cycle counter every clock
    counter <= counter + 1;
    // After 100,000 cycles (1ms), move to next signal
    if (counter == 17'd99999) begin
        counter <= 0;
        active_anode <= (active_anode==2'd3)? 2'd0 : active_anode + 1;
        // anyways active_anode can wrap over if it is 2-bit wide
    end
end

// Activate the anodes based on counter value
assign an0 = ~(active_anode == 2'd0);
assign an1 = ~(active_anode == 2'd1);
assign an2 = ~(active_anode == 2'd2);
assign an3 = ~(active_anode == 2'd3);
```

For simulation, you would need to force active_anode to zero for 1ms and counter to 0 for 10ns, corresponding to 1ms digit period.

3 Submission and Demo Instructions

1. Demo should be given in the assigned lab slot itself.
2. You are required to submit the following on Gradescope:
   - 40 points: Verilog files for all the designed modules and testbench.
   - 10 points: Warning-clean constraint file (.xdc), Bit file.
   - 30 points: Questionnaire corresponding to the assignment.
   - 20 points: A short report (2-3 pages) outlining simulation snapshots, synthesis report (particularly resource counts: Flip-flops, LUTs, BRAMs, and DSPs) and generated schematics. Explain your design decisions. Having only snippets in the report (and no description) will result in penalty.

We advise you to be ready with your design before the lab session, and during the session, perform validation by downloading it into the FPGA board.

4 Resources references

- IEEE document: https://ieeexplore.ieee.org/document/1620780
- Basys 3 board reference manual: https://digilent.com/reference/_media/basys3:basys3_rm.pdf
- Online Verilog simulator: https://www.edaplayground.com