EXTENDED SEVEN-SEGMENT DECODER/USING GIT

Lab01

Otieno Maurice (Scribe)

Daniel Hayes

31/10/2023

Introduction:

The NEXY'S A7 100T printed circuit board provided in the lab has a seven-segment display. This display consists of eight digits driven by a common anode signal and each digit has seven segments made of LED lamps connected in a particular manner. This arrangement of LEDs allows us to display information depending on the signals the display receives. Each digit can display a range of characters from 0 TO F and characters such as P and C. Each digit is also supplied with a decimal point to its right end. Displaying information on all or some of the LEDs utilizes a time-multiplexed approach in which the anode signal to only one of the digits is enabled at a given time. The display on the NEXY'SA7 100T board uses active low rather than active high signals to drive this display. The board allows us to interface designs written in an HDL such as system Verilog or VHDL with the display by specifying the constraints written in tool command language.

In this lab we focused on driving only one digit of the seven-segment display. The CAD software used is for simulation and optimization was Vivado. We modelled the desired behavior in system Verilog and wrote a test bench to check that the described behavior is consistent with the desired behavior. We then synthesized the design, implemented it, routed the PCB and generated bitstream. We programmed the board with the bitstream and made observations.

We also learnt how to use the git version control system to store our code.

The code for all our designs can be found at
https://github.com/otienomaurice1/ece212_alex_maurice.git

Design:

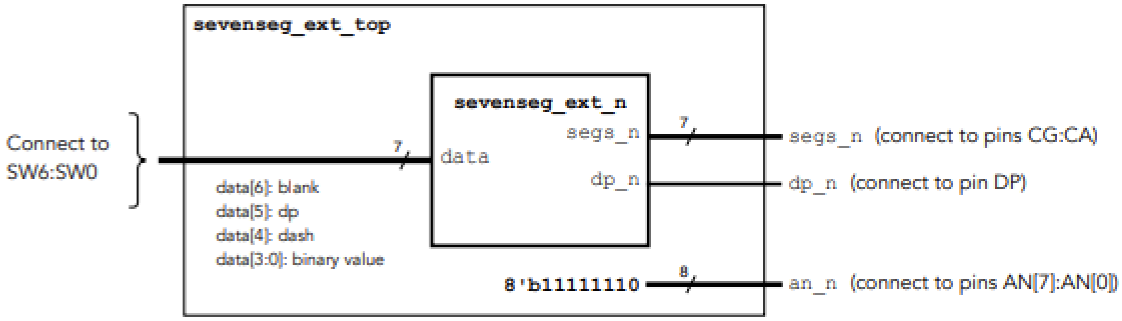
1. Design specifications:

- The design would interface with the last digit of the 8-digit seven-segment display for input and SWITCHES 0 TO 6 for output on the nexys100t printed circuit board.
- All elements of the design will be combinational
- The digit should have the ability to:
  a.) Turn off both the display and the decimal point
  b.) Display a minus sign on the display
  c.) Display any of the digits from 0 to 9
  d.) Display the characters A, C, F and P.

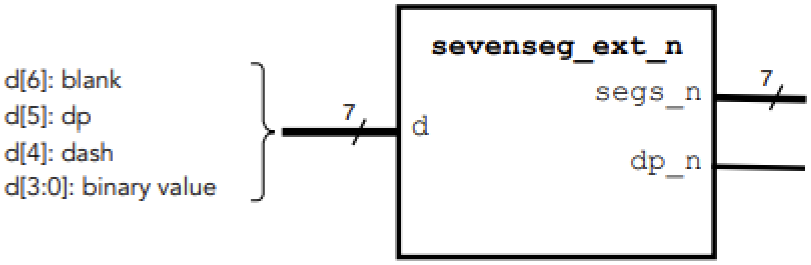
2. Design implementation

- The overall design of the extended seven-segment controller utilized a top module named sevenseg_ext_top. This module consists of a seven-bit input named data, a single-bit output named dp_n, a seven-bit output named segs_n and an eight-bit output named segs_n. The outputs represent the seven-segment signals, the anode signal and the decimal point signal respectively.

  This module generated the anode signal and also contains an instance of the sub module named sevenseg_ext_top. Since we are driving only the last digit, we could make the output as simple as an_n = 8'b11111110.
- The diagram for the top module is shown below



- The sevenseg_ext module contains the design to drive the seven segments. It consists of a seven-bit input d. The function of each bit is described below.
  1. d [6] (blank) – when asserted, all output segments and the decimal point are turned off (i.e., to logic high).
  2. d [5] (dp) – when asserted (and blank is not asserted), assert output dp_n low to display the decimal point.
  3. d [4] (dash) – when asserted (and blank is not asserted), assert output segs_n display a dash/minus sign.
  4. d [3:0] (data) – when blank and dash are not asserted, assert output segs_n to display a seven-segment digit corresponding to the value of data for values 0-9. When the value of data is 10, 11,12 and 15, the segs_n should display A, C, F and P respectively.
- We used multiplexers (if statement in Verilog) to test for the conditions when any of the first three bit are asserted and a decode to decode (case statement in Verilog) the outputs for segs_n.



3. Design realization:

- The design was realized by specifying the constraints for the switches and seven-segment display in an XDC file. The design was synthesized and bitstream generated for programming the board.

Results:

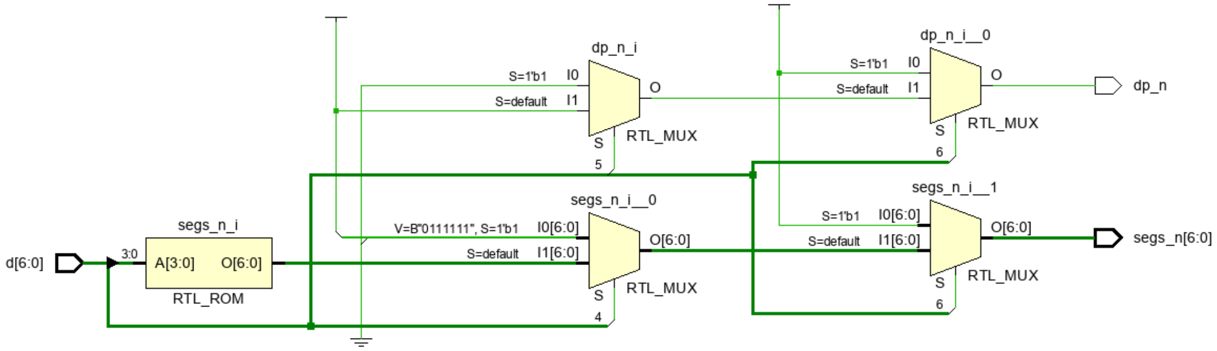
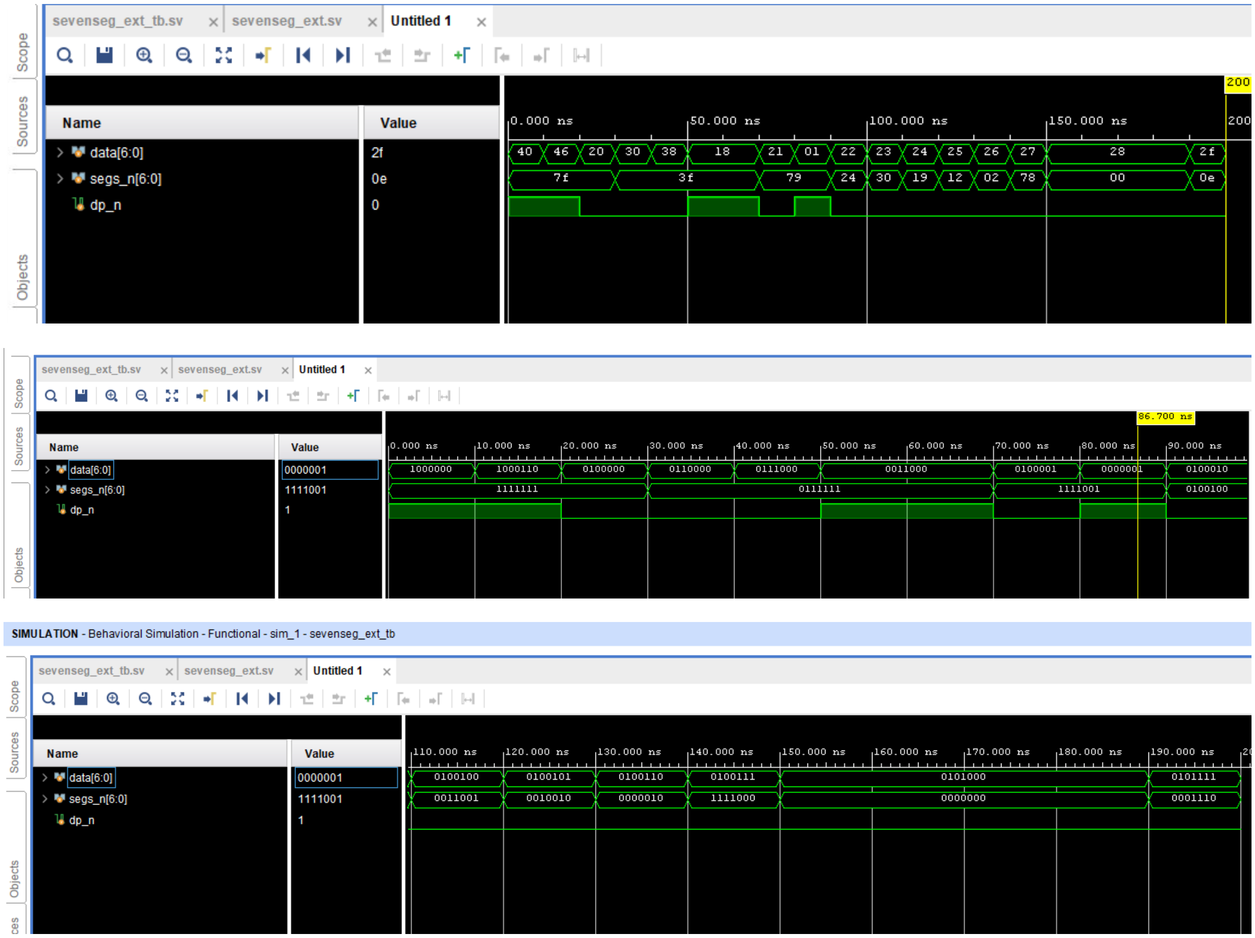


Figure 1schematic diagram for sevenseg_ext



Name	Value
data[6:0]	2f
segs_n[6:0]	0e
dp_n	0

Name	Value
data[6:0]	0000001
segs_n[6:0]	1111001
dp_n	1

SIMULATION - Behavioral Simulation - Functional - sim_1 - sevenseg_ext_tb

Name	Value
data[6:0]	0000001
segs_n[6:0]	1111001
dp_n	1

Figure 2simulation scope for sevenseg_ext_tb

The simulation scope diagrams show that the design is functioning as intended. For example when the first bit of the input is 1 , then all the bits of segs_n should be actively high i.e 11111111.

Conclusion:

The design was pretty simple to implement. However, because we had come from a long break it took some time to familiarize with system Verilog again.

Time spent on lab

4 hrs

N/B this lab was done by Hayes and Maurice in the first week before we switched partners