ECE 3300L Lab 1 - Switch ↔ LED Interface

By: Raj Gokidi and Priyanka Ravinder
California State Polytechnic University, Pomona

June 18th, 2025

Introduction

This lab involves creating a simple interface between switches and LEDs on the Nexys A7-100T FPGA board. The goal is to build a Verilog module that reads the 16 user switches and directly controls the 16 LEDs on the board. Each switch will turn its corresponding LED on or off creating a mapping between the inputs and outputs. This lab teaches the basics of FPGA development such as writing Verilog code, setting up pin constraints, and programming the board.

Verilog Code

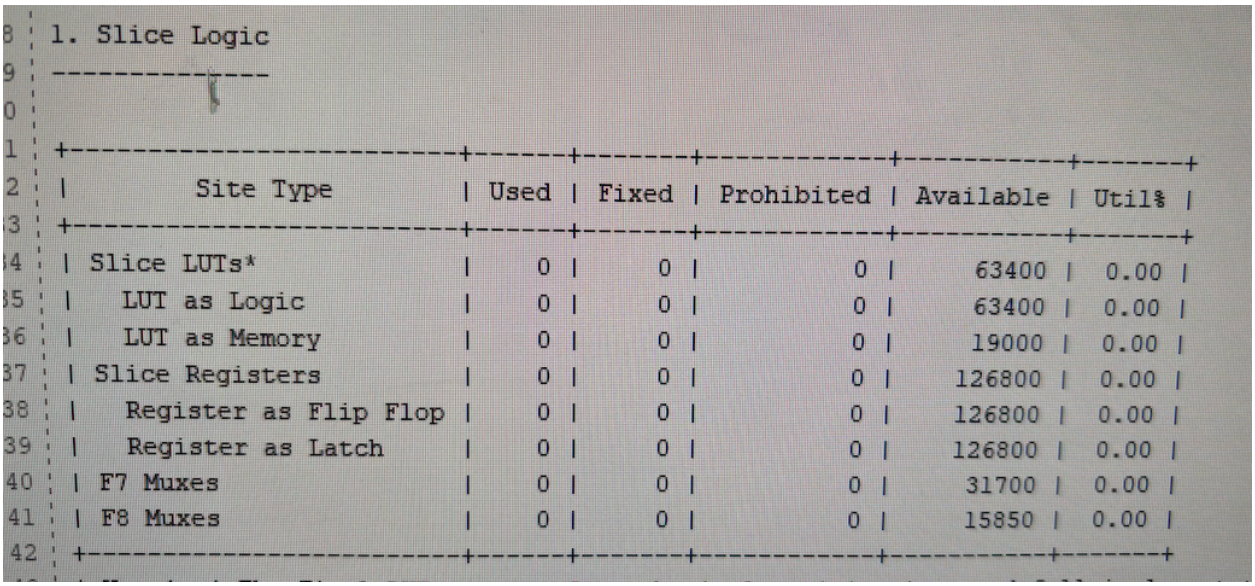
```
module switch_led_interface(
    input wire [15:0] sw,
    output wire [15:0] led
   );
   assign led = sw;
endmodule
```

Constraint Code

```
## Switches
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports {sw[0]}]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports {sw[1]}]
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports {sw[2]}]
set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports {sw[3]}]
set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports {sw[4]}]
set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports {sw[5]}]
set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports {sw[6]}]
set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports {sw[7]}]
set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS33 } [get_ports {sw[8]}]
set_property -dict { PACKAGE_PIN U8    IOSTANDARD LVCMOS33 } [get_ports {sw[9]}]
set_property -dict { PACKAGE_PIN R16   IOSTANDARD LVCMOS33 } [get_ports {sw[10]}]
set_property -dict { PACKAGE_PIN T13   IOSTANDARD LVCMOS33 } [get_ports {sw[11]}]
set_property -dict { PACKAGE_PIN H6    IOSTANDARD LVCMOS33 } [get_ports {sw[12]}]
set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports {sw[13]}]
set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports {sw[14]}]
set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports {sw[15]}]
```

LEDs

```
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports {led[0]}]
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports {led[1]}]
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports {led[2]}]
set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports {led[3]}]
set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports {led[4]}]
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {led[5]}]
set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports {led[6]}]
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {led[7]}]
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {led[8]}]
set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports {led[9]}]
set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports {led[10]}]
set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports {led[11]}]
set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports {led[12]}]
set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports {led[13]}]
set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports {led[14]}]
set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports {led[15]}]
```



1. Slice Logic

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	0	0	0	63400	0.00
LUT as Logic	0	0	0	63400	0.00
LUT as Memory	0	0	0	19000	0.00
Slice Registers	0	0	0	126800	0.00
Register as Flip Flop	0	0	0	126800	0.00
Register as Latch	0	0	0	126800	0.00
F7 Muxes	0	0	0	31700	0.00
F8 Muxes	0	0	0	15850	0.00

According to the utilization report we can see that we used 0 LUTs and 0 FFs since our Verilog code was simply wiring the switches to the LEDs. This does not require any combinational logic hence the 0 LUTs and no sequences hence the 0 FFs.

Reflection

This lab provided a good introduction to FPGA development by teaching the basic workflow of writing Verilog code, creating constraint files, and programming hardware. The simple switch-to-LED mapping helped me understand how digital signals flow from inputs to outputs in an FPGA without getting overwhelmed by complex logic. Working with the XDC file taught me how important it is to properly map logical port names to physical pins on the board.

Youtube link: https://youtu.be/IX6tRP1XMj8