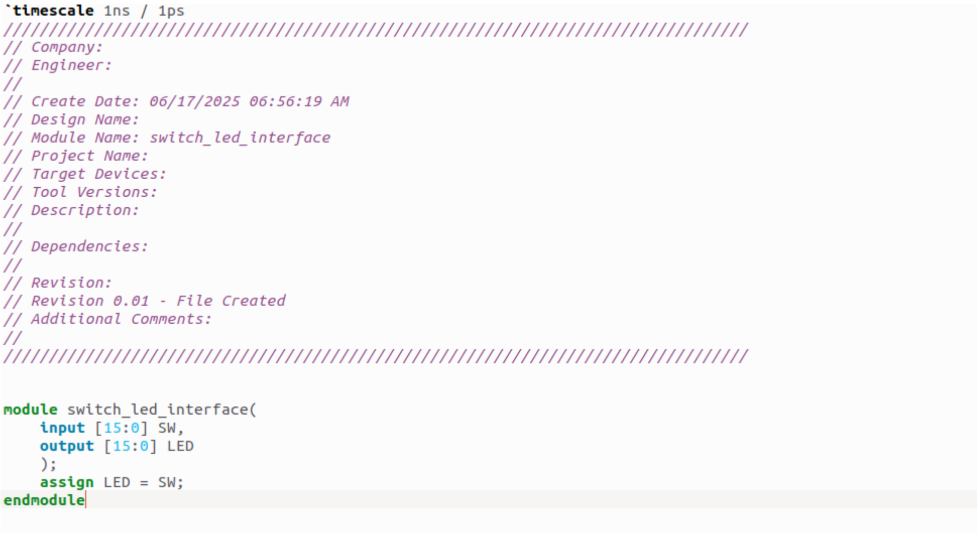
ECE3300 Lab 1 Group R

Introduction

Lab 1 is a basic introduction to using the Digilent Nexys A7-100T that reads 16 switch inputs to corresponding LEDs. We will learn how to use HDL I/O mapping, constraint files, synthesis, and FPGA programming.

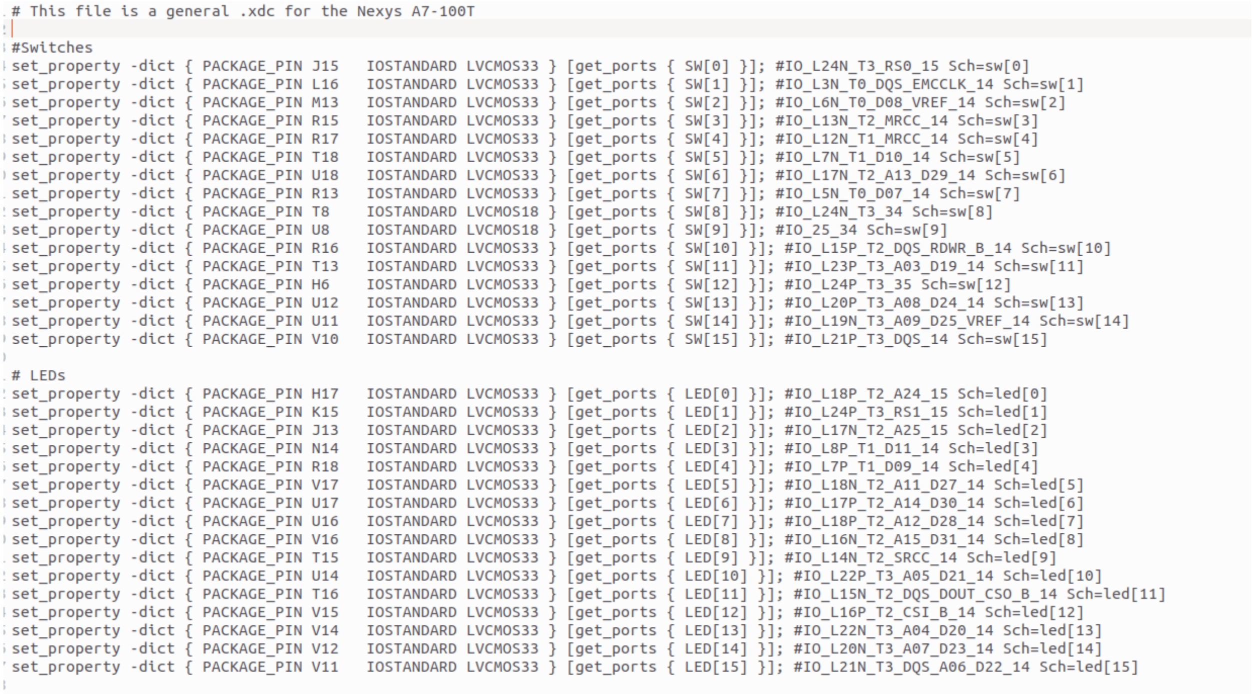
Verilog Code Snippet



```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 06/17/2025 06:56:19 AM
// Design Name:
// Module Name: switch_led_interface
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module switch_led_interface(
    input [15:0] SW,
    output [15:0] LED
    );
    assign LED = SW;
endmodule
```

We have declared an input bus named SW and an output bus named LED.

XDC Snippet

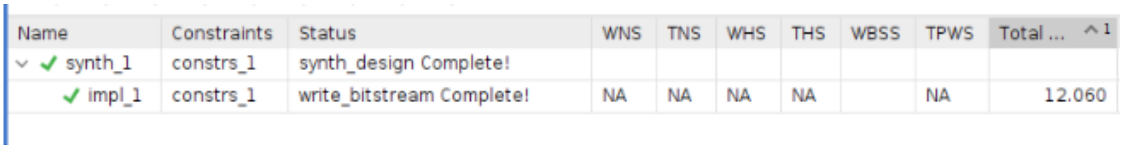


```
# This file is a general .xdc for the Nexys A7-100T

#Switches
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { SW[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { SW[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { SW[2] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { SW[3] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { SW[4] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { SW[5] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports { SW[6] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports { SW[7] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS18 } [get_ports { SW[8] }]; #IO_L24N_T3_34 Sch=sw[8]
set_property -dict { PACKAGE_PIN U8    IOSTANDARD LVCMOS18 } [get_ports { SW[9] }]; #IO_25_34 Sch=sw[9]
set_property -dict { PACKAGE_PIN R16   IOSTANDARD LVCMOS33 } [get_ports { SW[10] }]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
set_property -dict { PACKAGE_PIN T13   IOSTANDARD LVCMOS33 } [get_ports { SW[11] }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
set_property -dict { PACKAGE_PIN H6    IOSTANDARD LVCMOS33 } [get_ports { SW[12] }]; #IO_L24P_T3_35 Sch=sw[12]
set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports { SW[13] }]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports { SW[14] }]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports { SW[15] }]; #IO_L21P_T3_DQS_14 Sch=sw[15]

# LEDs
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { LED[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { LED[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { LED[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports { LED[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports { LED[4] }]; #IO_L7P_T1_D09_14 Sch=led[4]
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { LED[5] }]; #IO_L18N_T2_A11_D27_14 Sch=led[5]
set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports { LED[6] }]; #IO_L17P_T2_A14_D30_14 Sch=led[6]
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { LED[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7]
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { LED[8] }]; #IO_L16N_T2_A15_D31_14 Sch=led[8]
set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports { LED[9] }]; #IO_L14N_T2_SRCC_14 Sch=led[9]
set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports { LED[10] }]; #IO_L22P_T3_A05_D21_14 Sch=led[10]
set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports { LED[11] }]; #IO_L15N_T2_DQS_DOUT_CSO_B_14 Sch=led[11]
set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports { LED[12] }]; #IO_L16P_T2_CSI_B_14 Sch=led[12]
set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports { LED[13] }]; #IO_L22N_T3_A04_D20_14 Sch=led[13]
set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports { LED[14] }]; #IO_L20N_T3_A07_D23_14 Sch=led[14]
set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports { LED[15] }]; #IO_L21N_T3_DQS_A06_D22_14 Sch=led[15]
```

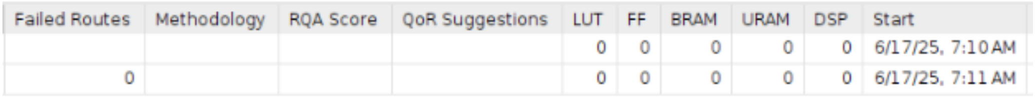
In the above screen shot out .XDC file We're telling the FPGA: ‘Connect switch 0 to pin J15 and expect 3.3V logic.’ This ensures our hardware and software align correctly

set_property -dict : sets properties for an object in case a port
{ PACKAGE_PIN J15: Assigns the port SW[0] to physical pin j15 on the FPGA package
IOSTANDARD LVCMOS33 }: Specifies the I/O standard LVCMOS33 means its using 3.3V Cmos Logic Levels.
[get_ports { SW[0] }]: Refers to the top level input and output port named SW[0] which is the switch on the board.

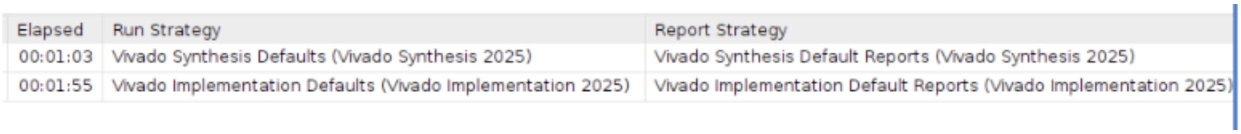
Synthesis screenshots



Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total ... ^1
✓ synth_1	constrs_1	synth_design Complete!							
✓ impl_1	constrs_1	write_bitstream Complete!	NA	NA	NA	NA		NA	12.060



Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start
				0	0	0	0	0	6/17/25, 7:10 AM
0				0	0	0	0	0	6/17/25, 7:11 AM



Elapsed	Run Strategy	Report Strategy
00:01:03	Vivado Synthesis Defaults (Vivado Synthesis 2025)	Vivado Synthesis Default Reports (Vivado Synthesis 2025)
00:01:55	Vivado Implementation Defaults (Vivado Implementation 2025)	Vivado Implementation Default Reports (Vivado Implementation 2025)

Our synthesis screenshot consists of two rows, synthesis 1 and implementation 1 which are shown completed. The columns WNS, TNS, WHS, THS, WBSS and TPWS do not apply to this lab because timing stats are not required. We have 0 failed routes in the second screenshot and our LUT, FF, BRAM, URAM, DSP are 0 because we required 0 logic and memory.

Group video link

https://youtu.be/t5BkGERtoOQ

Reflection/Conclusion

This lab was simply familiarizing ourselves with Vivado. And understanding how a constraint file works. Overall we successfully defined an input bus (`SW`) and an output bus (`LED`) and mapped them to the physical hardware using the `.XDC` constraints file. Specifically, we assigned `SW[0]` to pin **J15** with a **3.3V LVCMOS** logic standard, ensuring proper communication between our hardware and design. The synthesis and implementation steps completed successfully, with no errors or failed routes. Since this lab did not require complex logic or memory usage, the utilization of LUTs, flip-flops, BRAM, URAM, and DSPs remained at zero.

Timing statistics such as WNS, TNS, WHS, THS, WBSS, and TPWS are not applicable in this case. Overall, our basic setup confirms correct mapping and functional integration between the switch input and the FPGA environment.