HARDWARE IMPLEMENTATION ON FPGA

A PROJECT REPORT

submitted by

Gundra Samuel Rohan
(622150)

Department of Electronics & Communication Engineering
National Institute of Technology, Andhra Pradesh
Tadepalligudem, West Godavari
534101
April 2024

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr. Puli Kishore Kumar Sir for their outstanding instruction and mentorship during the course of Digital System Design which created a strong base in learning the Hardware Implementation.

I would also like to extend my appreciation to Koushik Kalyani for their invaluable guidance, support, and encouragement in the community of FPGA Speaks which helped me in gaining hands on experience on FPGA.

ABSTRACT

This report includes a brief description of the FPGA used (Zybo Z-7010), RTL coding and the generation of bitstream for a design. A Github link for the Demonstration video is given at the end of the report.

Design of the component which is to be implemented on the FPGA starts with programming a design source in Xilinx Vivado using Verilog HDL. A test bench can be programmed in a simulation source to verify the functionality of the design source. Simulation graph and Schematic can be viewed for the design source as well. Selection of I/O ports can be done after opening the schematic. This varies with different FPGA boards. This RTL analysis is followed by Synthesis. Synthesis is a straightforward mapping of your design to a library of digital logic gates available in your FPGA. Synthesised Schematic is different from the RTL analysis schematic. Synthesis schematics shows the internal LUTs in the design source. Implementation comes after the synthesis. Implementation tool will take the netlist as input and does optimization, placement and routing. Implementation Schematic shows the CLBs in the FGPA that is being implemented. Generation of bitstream can be done after the implementation.

CONTENTS

ACKNOWLEDGMENT **i**

ABSTRACT **ii**

Chapter 1. FPGA **1**

1.1 ZYBO Z-7010 1

Chapter 2. VERILOG DESIGN **2**

Chapter 3. SIMULATION **3**

Chapter 4. RTL SCHEMATIC **4**

Chapter 5. SYNTHESIZED SCHEMATIC **6**

Chapter 6. IMPLEMENTATION **7**

Chapter 7. GENERATION OF BITSTREAM **8**

Chapter 8. DEMONSTRATIONS **9**

Chapter 1

FPGA

FPGAs or **Field Programmable Gate Arrays** are a class of ICs designed to be configured by the user after manufacturing. These are different from ASICs which are custom designed for specific applications. FPGAs contain an array of CLBs interconnected via programmable switches. These CLBs can be configured to implement complex digital logic functions, including combinational and sequential circuits, arithmetic units, memory elements, and specialized processing units. CLBs contain smaller components, including flip-flops, look-up tables (LUTs), and multiplexers. The programmable switches allows designers to establish connections between logic blocks to create custom logic circuits tailored to their application requirements. Designers can use HDLs such as Verilog or VHDL to program the FPGA, which is then synthesized into a configuration file called **Bitstream** that programs the FPGA.

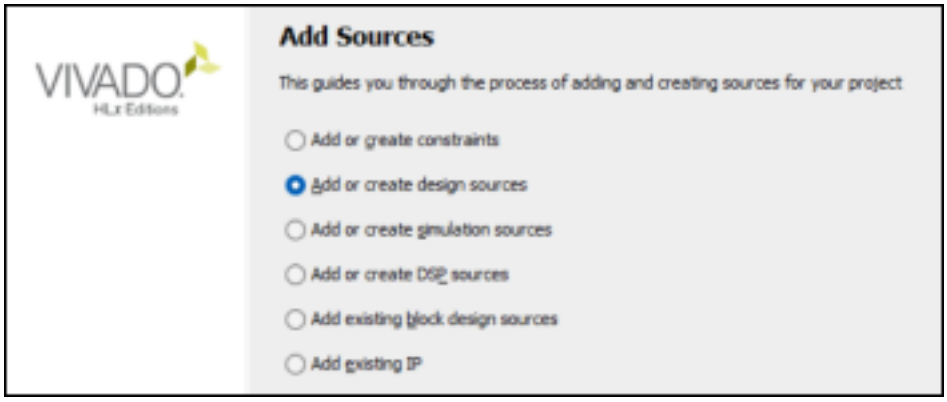
1.1 ZYBO Z-7010

The Zybo Z-7010 is a development board designed by Digilent, featuring the Xilinx Zynq-7000 SoC. The board has a variety of I/O interfaces, including HDMI, VGA, audio, Ethernet, USB, Pmod and GPIO. Along with these the Zybo Z-7010 features on-board peripherals such as switches, buttons, LEDs, and a microSD card slot(used for rebooting). Zybo can be powered by USB and external power source which can be either power jack or a 5v DC source. This provides power to the board for programming and basic functionality.

Chapter 2

VERILOG DESIGN

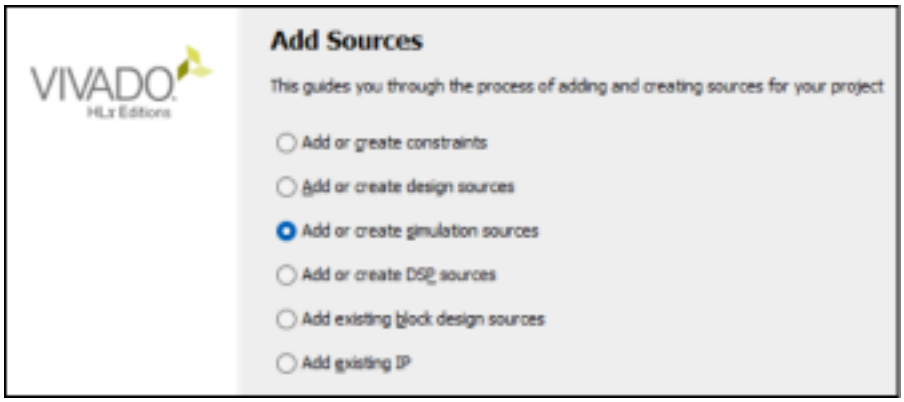
After creating a new project in Vivado with Zybo z-7010 as the selected component, we're brought to the project page. On the left we have the Flow Navigator which has the various operations required in the Hardware Implementation. From the Flow Navigator, we go to add sources to get a design source for the verilog design. The design in programmed in verilog in this design source.



Chapter 3

SIMULATION

To simulate a design source, we need a simulation for it. We can add this simulation source from the add sources section. This simulation source is usually called the **Test Bench** for its respective design source. Test cases of inputs are given in this simulation source to verify the design source.

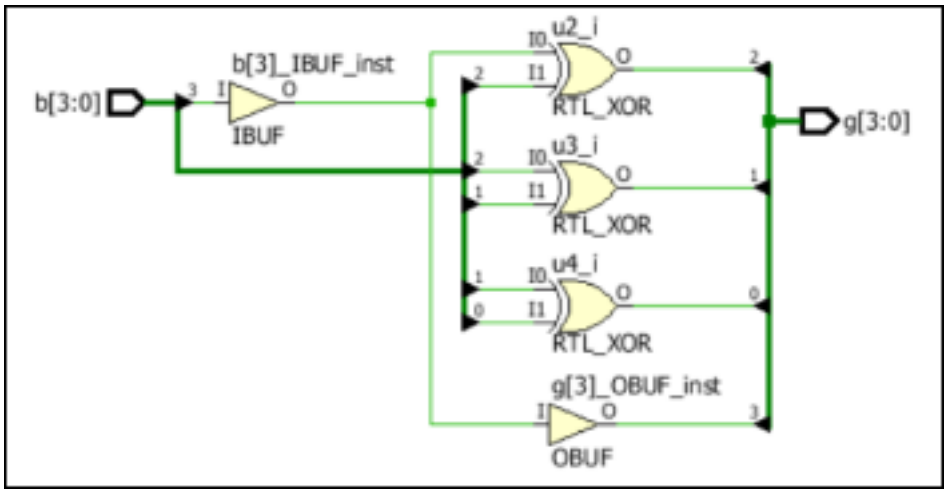


Chapter 4

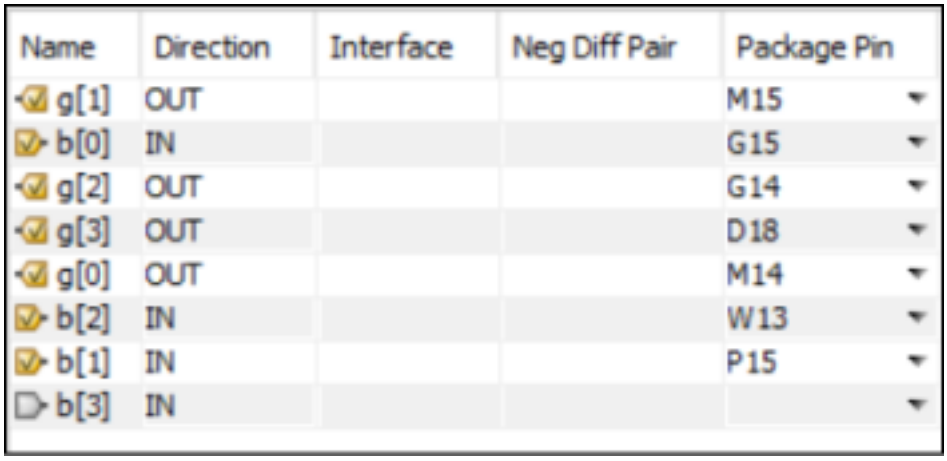
RTL SCHEMATIC

Once we're sure that our design is functional, we can view what are the logic gates inside it, how are they connected to the inputs and outputs. Under **RTL Analysis**, we have the Elaborated design section which has the Schematic view.

Below figure is a schematic of Binary to Gray code converter. Once we open the RTL Schematic, we'll find **I/O Ports** section where we can allocate/select a package pin on the FPGA for a certain Input/Output. Inputs are usually assigned switches and Outputs are assigned LEDs. After selecting this we need a save this configuration under a **Constraint**.



This figure shows the Package pin allocation for the inputs and outputs. Going through the data sheet of the FPGA is absolutely necessary before assign the pins.



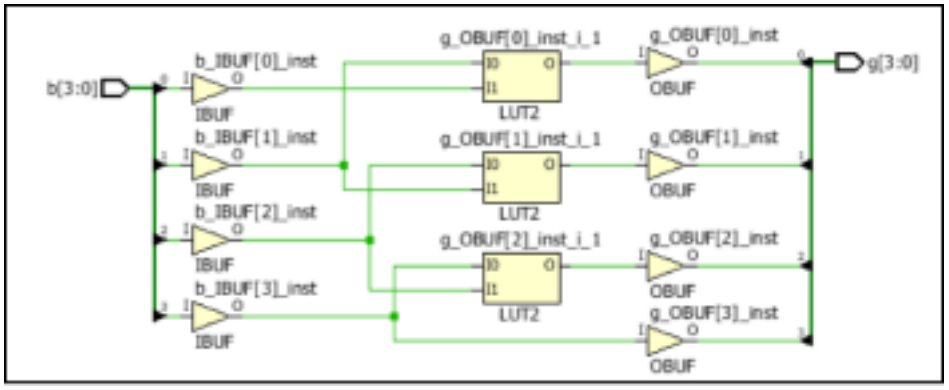
Name	Direction	Interface	Neg Diff Pair	Package Pin
g[1]	OUT			M15
b[0]	IN			G15
g[2]	OUT			G14
g[3]	OUT			D18
g[0]	OUT			M14
b[2]	IN			W13
b[1]	IN			P15
b[3]	IN			

Chapter 5

SYNTHESIZED SCHEMATIC

Under Synthesis section, we can view the synthesized schematic of the design source. The synthesized schematic shows the internal connections of the LUTs and buffers within the FPGA.

Below figure is a synthesized schematic of a Binary to Gray code converter.



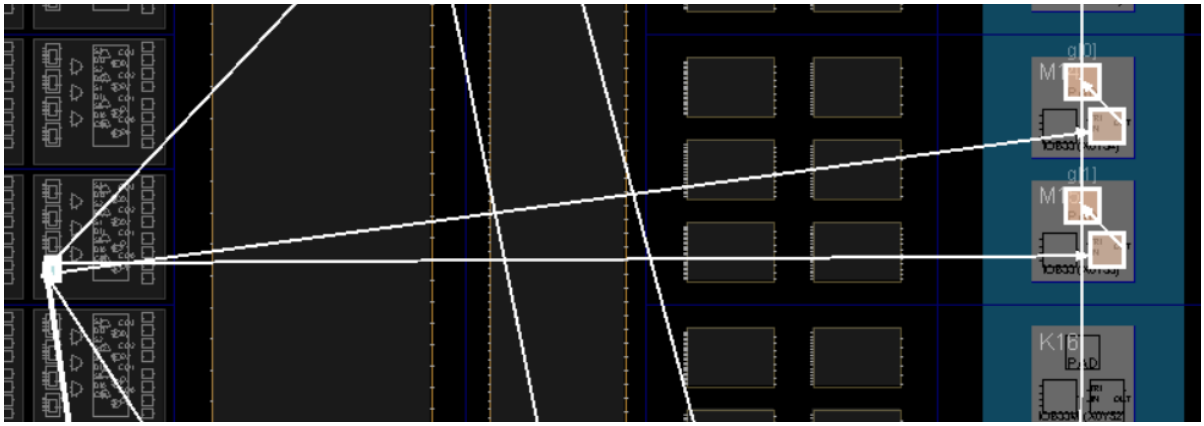
Under Synthesis, we also have a view of the device and the position of allocated leaf cells.

Chapter 6

IMPLEMENTATION

Under Implementation section, we can view the FPGA device and how the LUTs are connected to the input and output ports.

The below picture shows how the LUT (Look up table) maps with the input and output ports of the FPGA.



Chapter 7

GENERATION OF BITSTREAM

From the Flow Navigator, at the bottom we have the generate bitstream which generates the bitstream required for the hardware implementation. After the bitstream generation, FPGA needs to be connected to the PC via a USB cable which powers the FPGA and also programs the FPGA.

Chapter 8

DEMONSTRATIONS

Demonstration link is given below

Link to the Demonstration