COL215P

LAB ASSIGNMENT 10 REPORT

SHREYA GUPTA, GAURAV KUMAR

Introduction

In this assignment we work on Vivado to learn how to create a FIFO buffer by first instantiating a memory component.

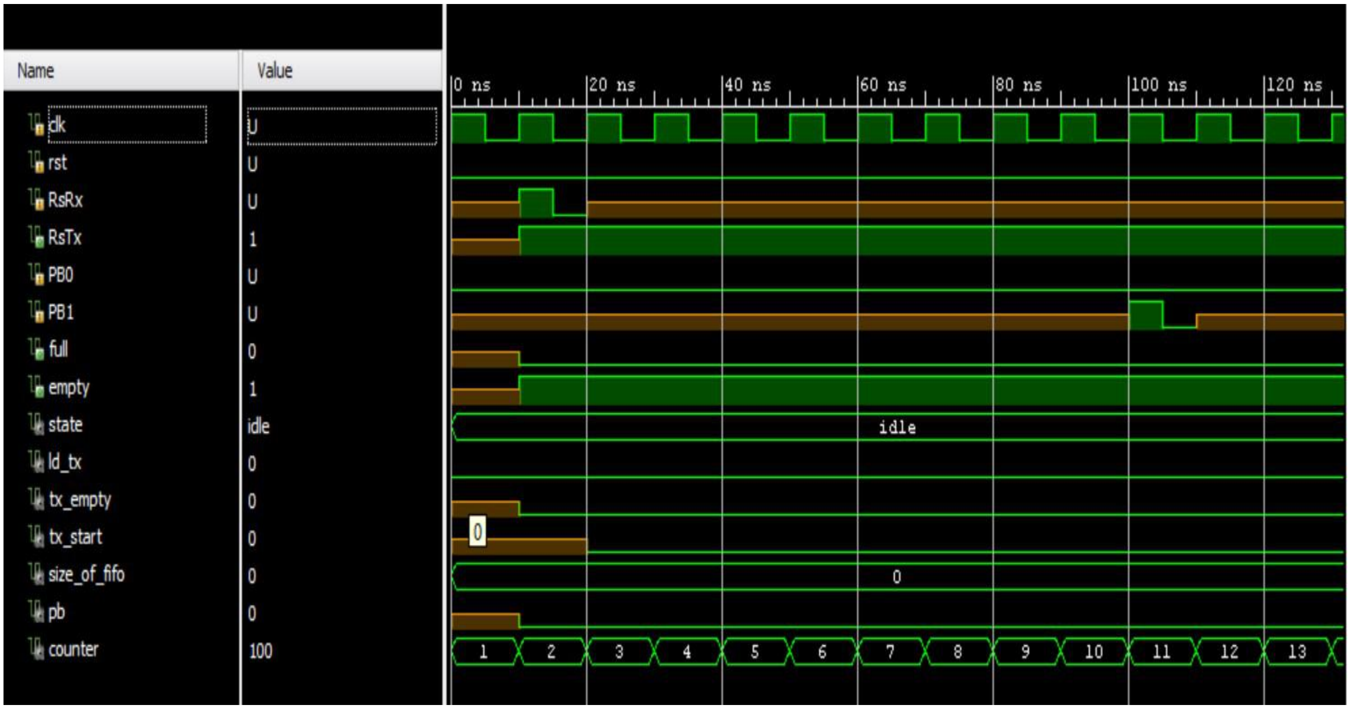
First we implement memory using BRAM which we create using vivado’s built-in block design and then design a FIFO with push button switches for WRITE and READ. WRITE pushes an input from the switches to the FIFO buffer whereas READ pops the head of the FIFO buffer onto a register that is displayed on the 7- segment displays. We maintain two pointers for Head and Tail of the queue and using them, we display the FIFO status of full and empty.

Stage wise view of work done in lab:

1. We first created a new project on Vivado named project_9 and added a new VHDL source file named project_9_code.vhd (main file with other modules port mapped), BRAM_wrapper.vhd (created from our block design of dual ported BRAM), and an XDC constraint file project_9_cons.xdc.
2. We then added our VHDL code for FIFO using port mapped BRAM in the main vhdl file.
3. Then we added the required constraints for anode (4 bit, decided by clock), 16 switches for input, 7-segment display leds (7 distinct cathodes for determining lighting of the led connected), a clock (as defined in the master constraint file provided), 2 switched(for read and write) and 2 LEDs (for empty and full detection).
4. Then we synthesized, implemented and generated bitstream and then programmed our BASYS3 board with the generated bit file.

5. Upon force simulating the following simulation was procured:

SIMULATIONS

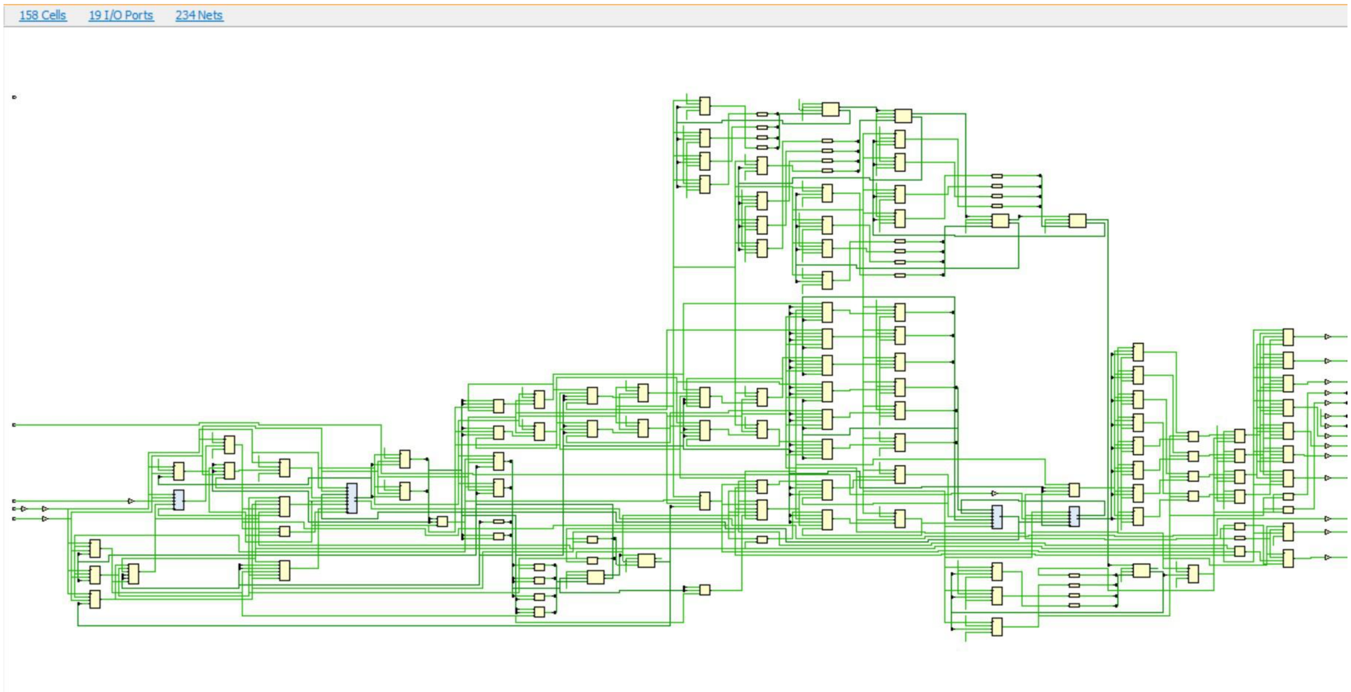


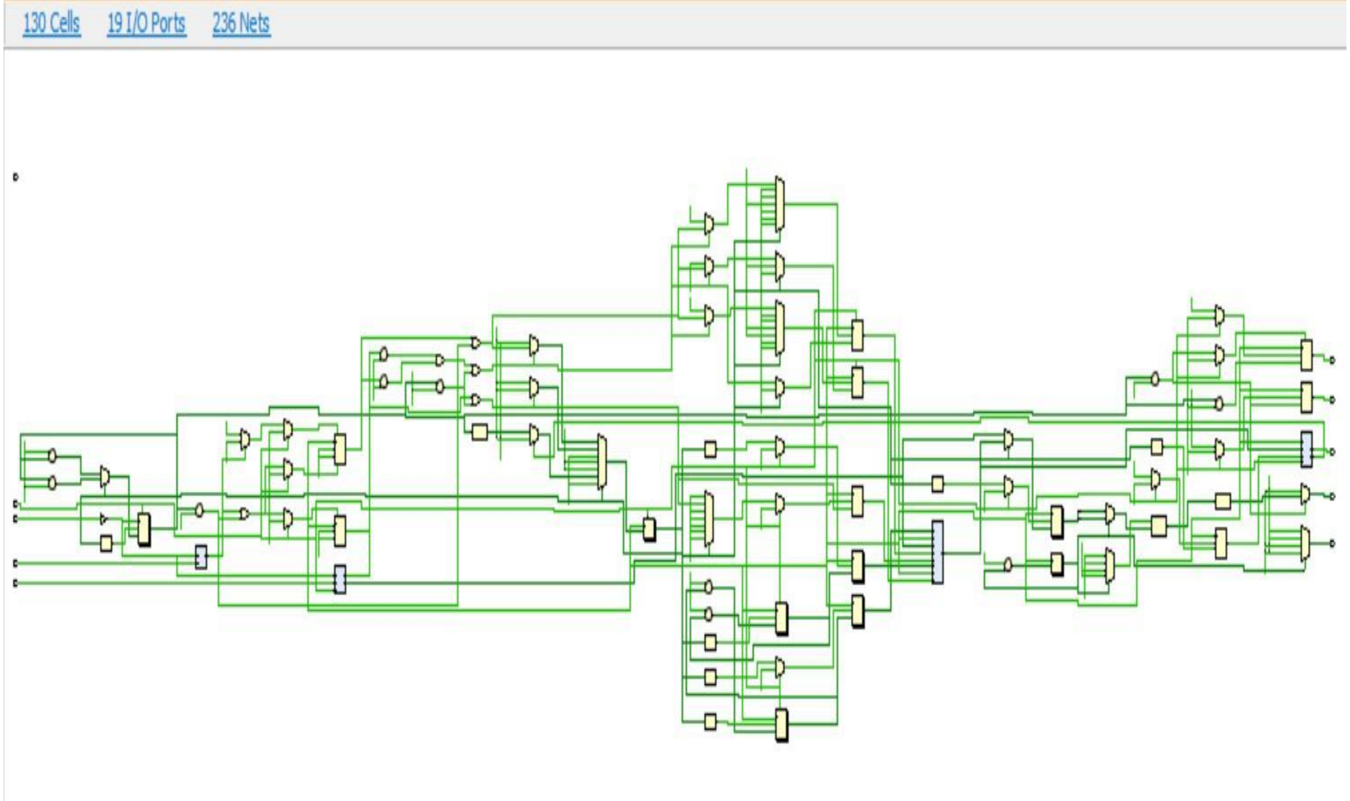
Stage wise view of work done in lab:

6. We first created a new project on Vivado named project_9 and added a new VHDL source file named project_9_code.vhd (main file with other modules port mapped), BRAM_wrapper.vhd (created from our block design of dual ported BRAM), and an XDC constraint file project_9_cons.xdc.
7. We then added our VHDL code for FIFO using port mapped BRAM in the main vhdl file.
8. Then we added the required constraints for anode (4 bit, decided by clock), 16 switches for input, 7-segment display leds (7 distinct cathodes for determining lighting of the led connected), a clock (as defined in the master constraint file provided), 2 switched(for read and write) and 2 LEDs (for empty and full detection).
9. Then we synthesized, implemented and generated bitstream and then programmed our BASYS3 board with the generated bit file.

Observation

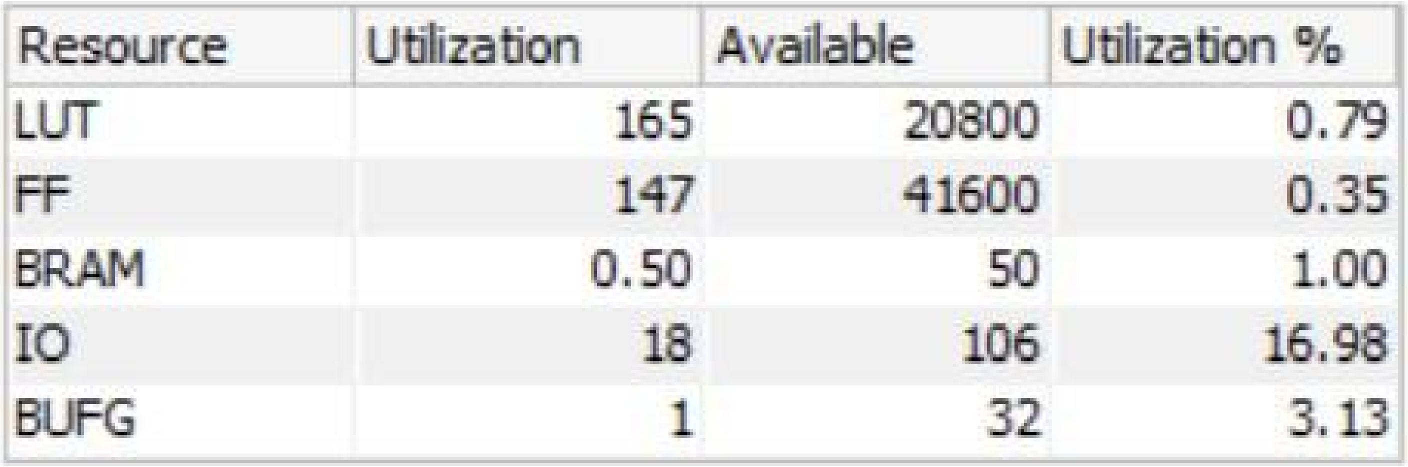
The observed design schematic:





The observed synthesis schematic (LUT):

Some resources observed:



Resource	Utilization	Available	Utilization %
LUT	165	20800	0.79
FF	147	41600	0.35
BRAM	0.50	50	1.00
IO	18	106	16.98
BUFG	1	32	3.13