COL215
Assignment-9

SOURAV(2019CS10404)
HARSH TANWAR(2019CS10432)

May 2022

1 Introduction

In lab we Design a FIFO buffer with switches as inputs and 7-segment displays as outputs. Also display FIFO status of full and empty on LEDs. For simulation and implementation of our VHDL code, we use vivado software.

2 Steps Performed in this Assignment

1. We create a component for BRAM and then import in the main file.
2. We create an entity for our inputs and outputs we define a clock and a vector for display the output in the seven segment and also define a input vector for taking the buttons as input and take two push down buttons as input for the data and showing the data.
3. We define a proccess on clock that basically increase the counter on the rising edge and after a specific value just make the counter as 0 and also in this process we use a debounce clk and after a specific counter value we take the negation of the debounce clk.
4. after that we make a process over the debounce clk that basically tells us about the working of the two push down button what which button is pressed and what we need to do in that situation.
5. From Assignment 5, we use the 4 digit-seven segment display code.
6. after that in new process we basically increase the refresh clk counter.
7. in the prcess we are doing the calculation for which anode we need to glow at which condition.
8. In another process clk on rising edge we make cases for each state i.e idle,write fifo read fifo when will go to the next step and at which step

and counter which we need to show the output on the seven segment display according the given input by the keyboard i.e the receiver part of the code.

9. In another process clk on rising edge we make cases for each state i.e queue is full and empty and glow the led corresponding to that the width of read abd write is 8 bit and deoth of the fifo buffer is 10.
10. We run simulations on our code and check the outputs.
11. Then we write the constraints file for our code.
12. After this we Synthesize and implement our project and generate a bitstream.
13. Finally, we download the bitstream in the FPGA board and check outputs..

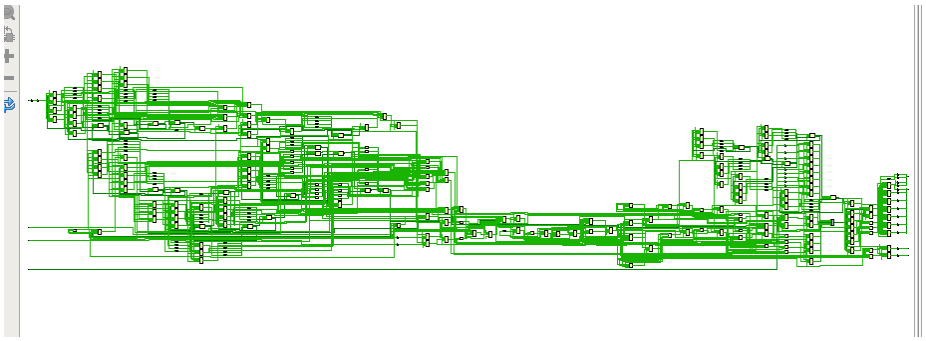
3 Resources Utilization

Name	Slice LUTs (20800)	Slice Registers (41600)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
a9	116	96	2	24	1
bram_fifo (BRAM)	7	2	2	0	0

Primitives

Ref Name	Used	Functional Category
FDRE	96	Flop & Latch
LUT1	53	LUT
LUT2	25	LUT
LUT3	21	LUT
LUT6	17	LUT
CARRY4	17	CarryLogic
LUT4	14	LUT
OBUF	13	IO
IBUF	11	IO
LUT5	4	LUT
RAMB36E1	2	Block Memory
BUFG	1	Clock

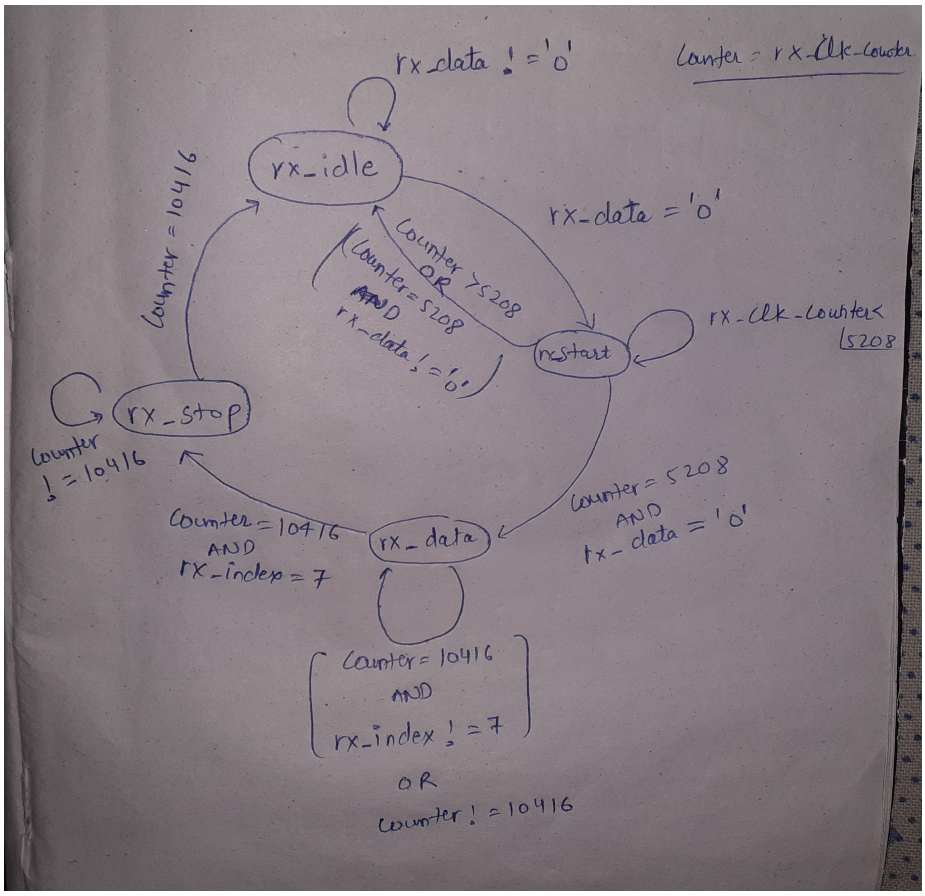
4 Digital Circuit of the Code



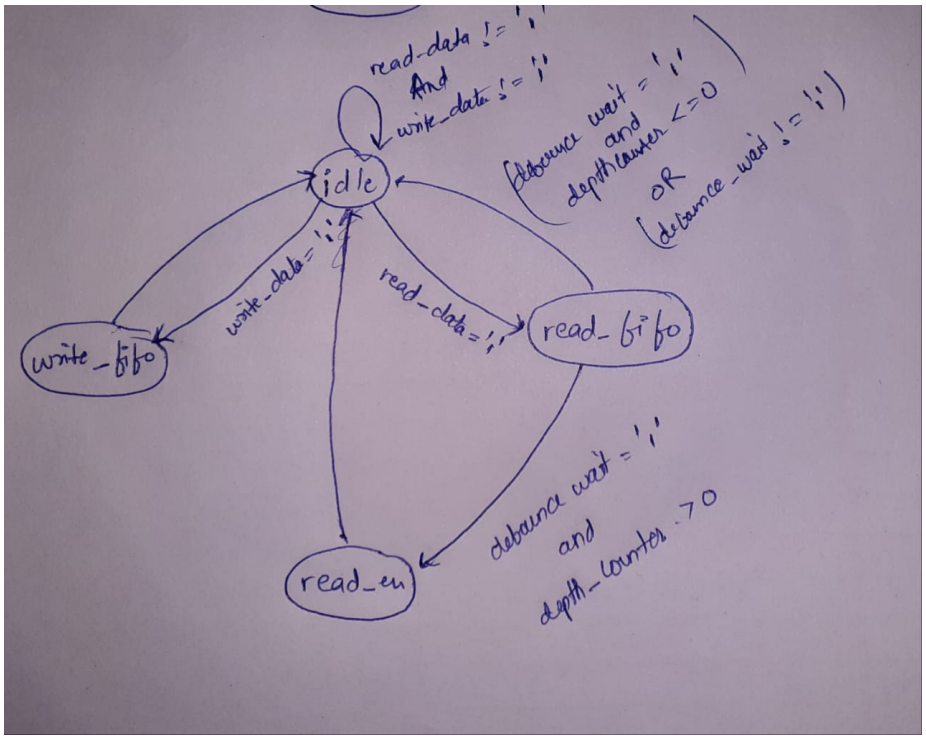
5 Expressions Derivation Truth Table

B3	B2	B1	B0	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0
1	0	1	0	0	0	0	1	0	0	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	0	0	0	0	1	0
1	1	1	0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0

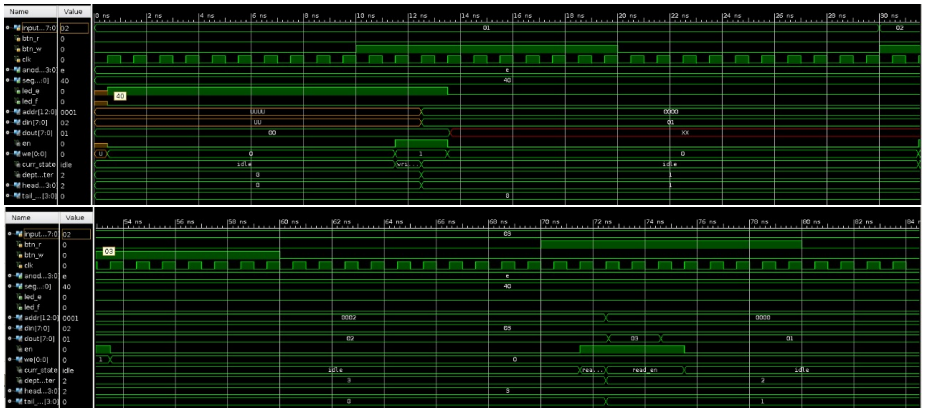
6 FSM Diagram reciever



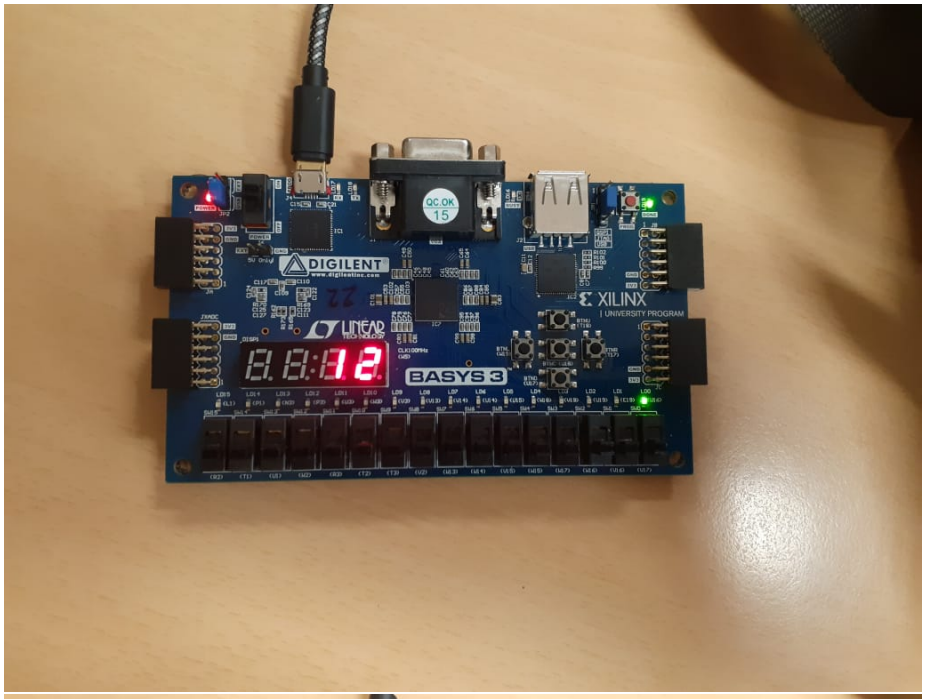
7 FSM Diagram

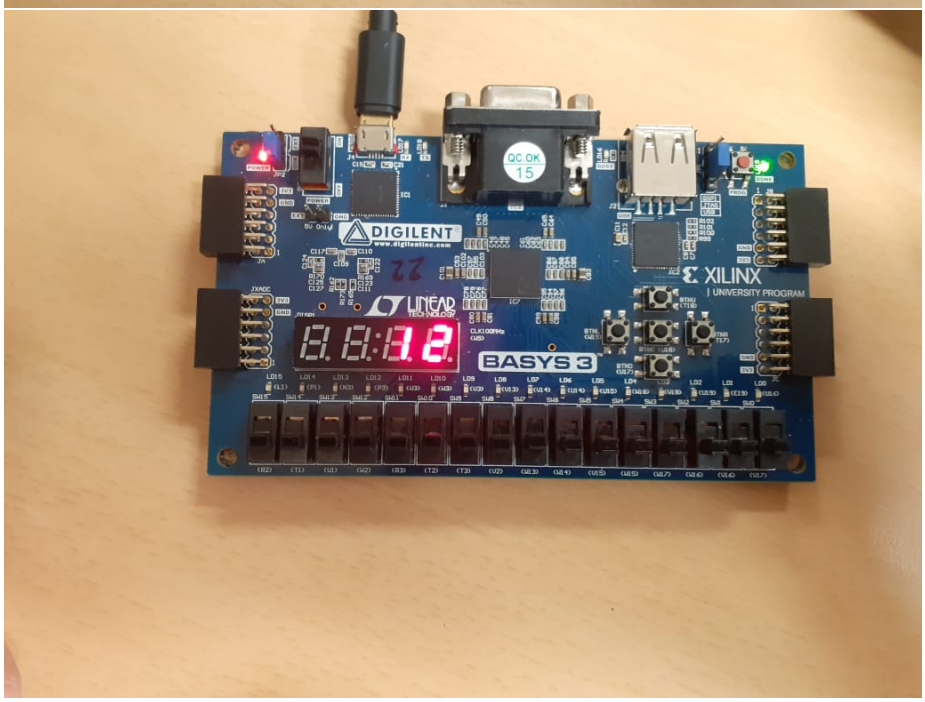


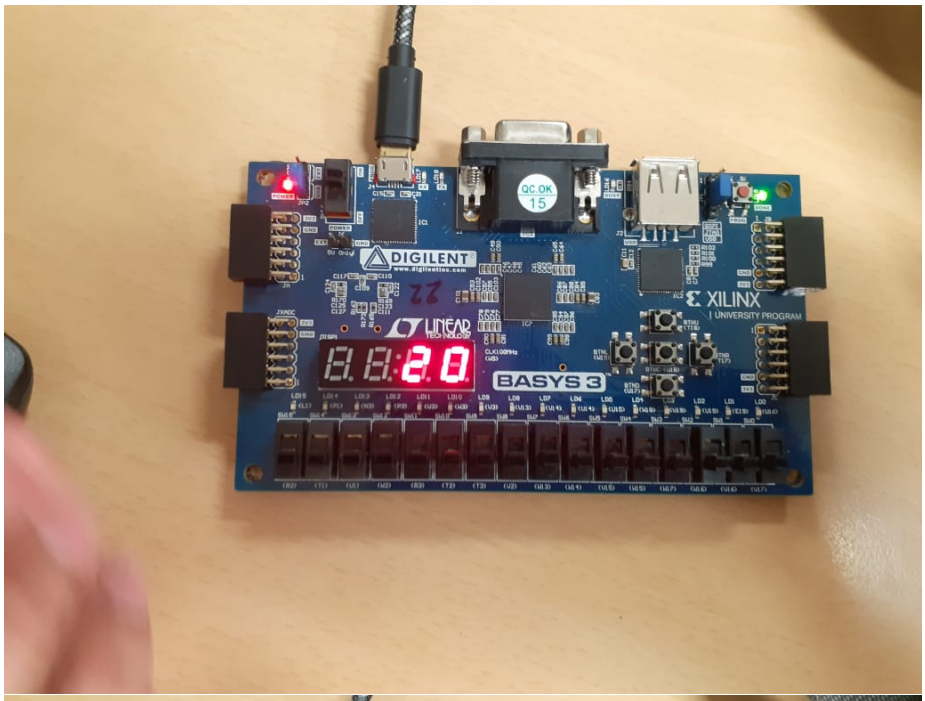
8 Simulation of the Code

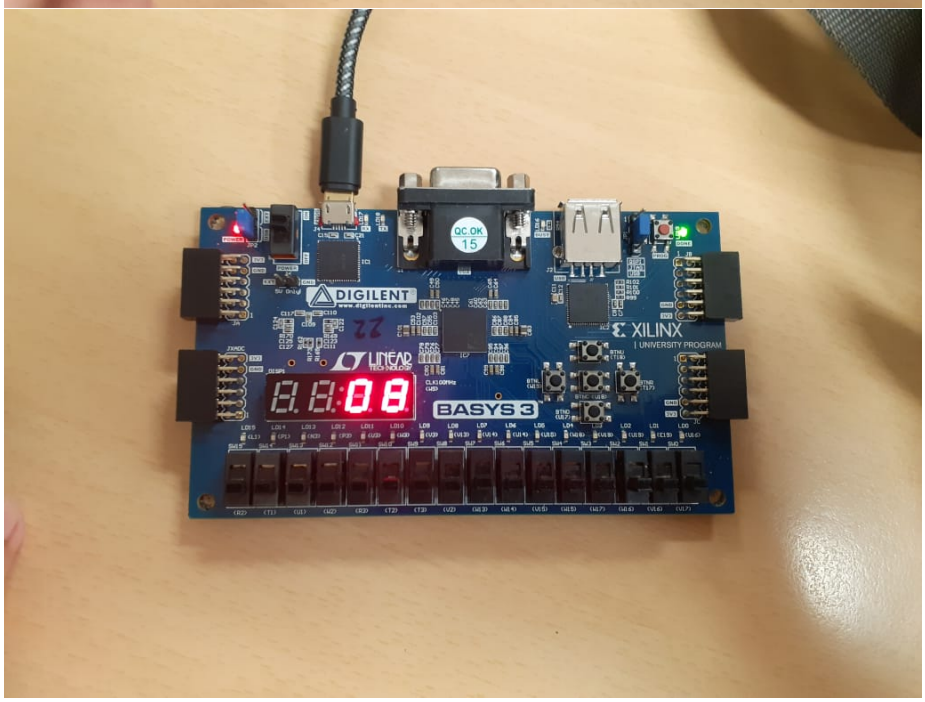


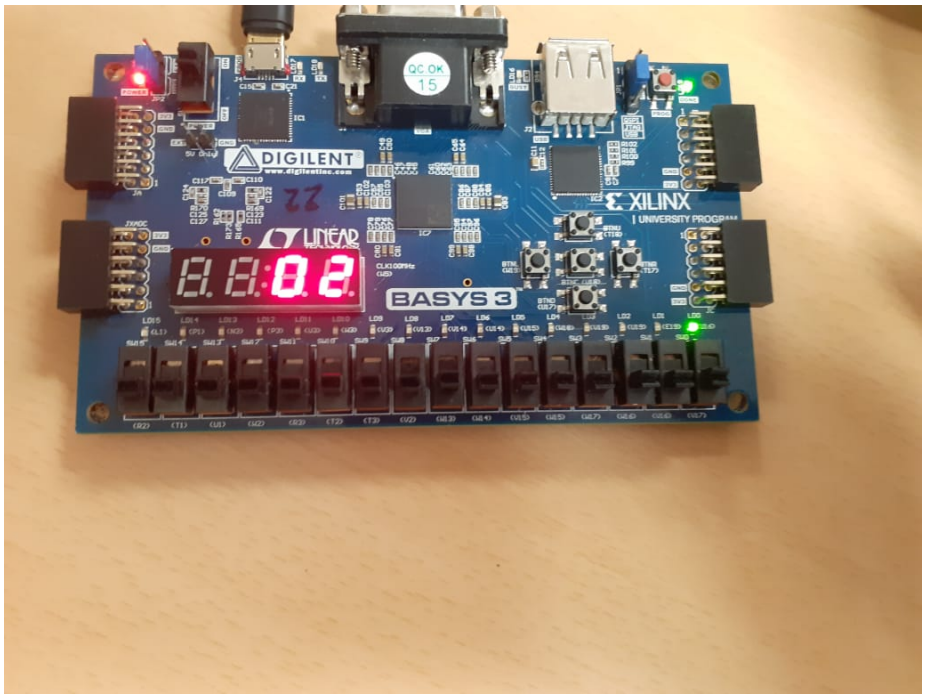
9 FPGA OUTPUT






DIGILENT
XILINX
UNIVERSITY PROGRAM
BASYS 3


DIGILENT
XILINX
UNIVERSITY PROGRAM
BASYS 3


DIGILENT
XILINX
BASYS 3