

COL215

Assignment-10

SOURAV(2019CS10404)
HARSH TANWAR(2019CS10432)

May 2022

1 Introduction

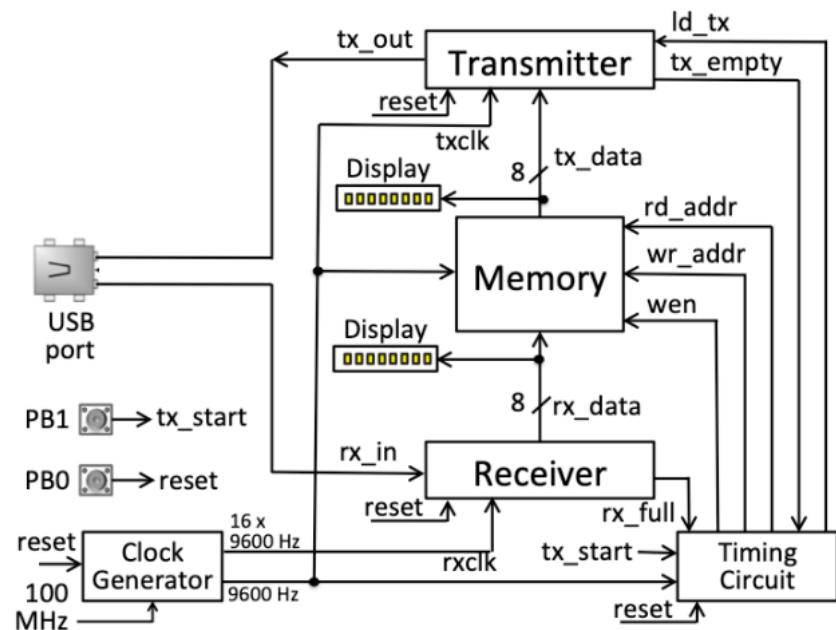
In lab we Design a Interface the serial receiver and transmitter designed in the previous assignments to a memory module for downloading (PC to BASYS-3) and uploading files (BASYS-3 to PC) For simulation and implementation of our VHDL code, we use vivado software.

2 Steps Performed in this Assignment

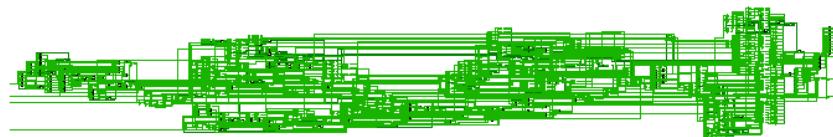
1. We create a component for BRAM and then import in the main file.
2. We create an entity for our inputs and outputs we define a clock and a vector for display the output in the seven segment and also take two push down buttons as input for the data and showing the data and also take the input for the glow the led when the queue is full and empty.
3. We define a process 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 process we are doing the calculation for which anode we need to glow at which condition.
8. we define a process over the ridging edge and the data we give in our rx in fill in the rxdata that basically tells us the state.

9. In another process clk on rising edge we make cases for each state i.e idle,start,data shift,stop 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.
10. In another process clk on rising edge we make cases for each state i.e idle,start,datashift,stop 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 transmitter part of the code.
11. 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.
12. 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 and write is 8 bit and depth of the fifo buffer is 10.
13. We run simulations on our code and check the outputs.
14. Then we write the constraints file for our code.
15. After this we Synthesize and implement our project and generate a bit-stream.
16. Finally, we download the bitstream in the FPGA board and check outputs using the gtkterm by pressing the key on the keyboard.

3 Block Diagram



4 Digital Circuit of the Code



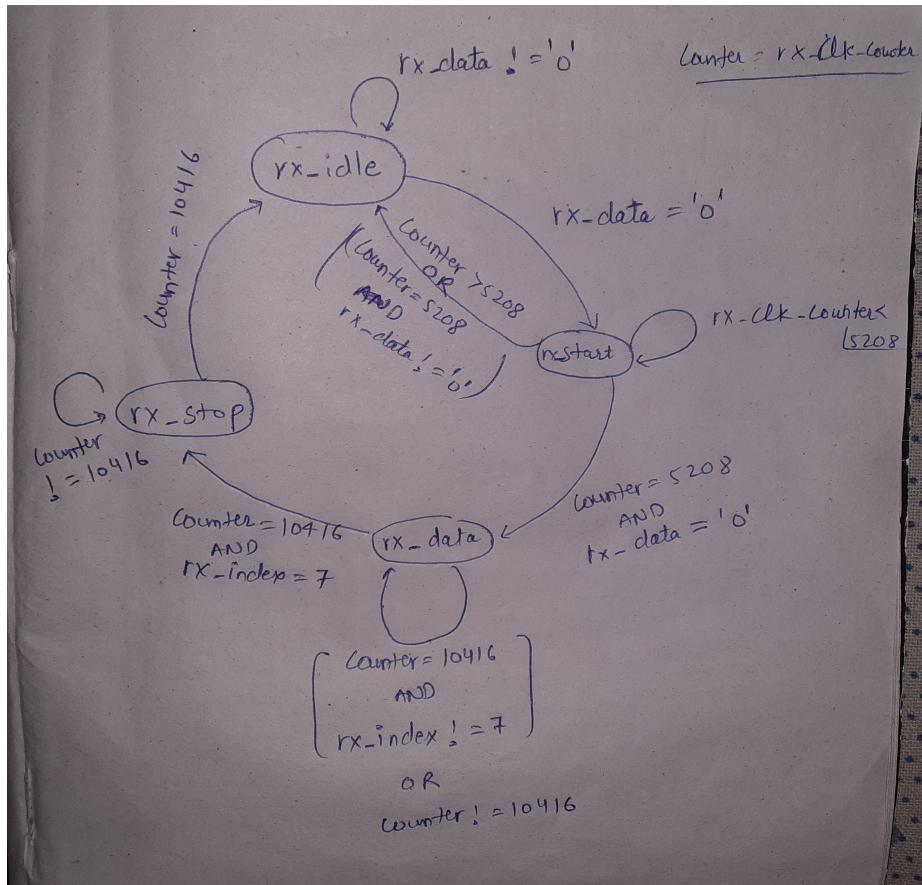
5 Resources Utilization

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
9:N a10 ↳ bram_fifo (BRAM)		296	193	2	18	1
Primitives						
Ref Name	Used	Functional Category				
FDRE	193	Flop & Latch				
LUT1	130	LUT				
LUT6	84	LUT				
CARRY4	40	CarryLogic				
LUT2	34	LUT				
LUT3	33	LUT				
LUT4	29	LUT				
OBUF	14	IO				
LUT5	13	LUT				
IBUF	4	IO				
RAMB36E1	2	Block Memory				
BUFG	1	Clock				

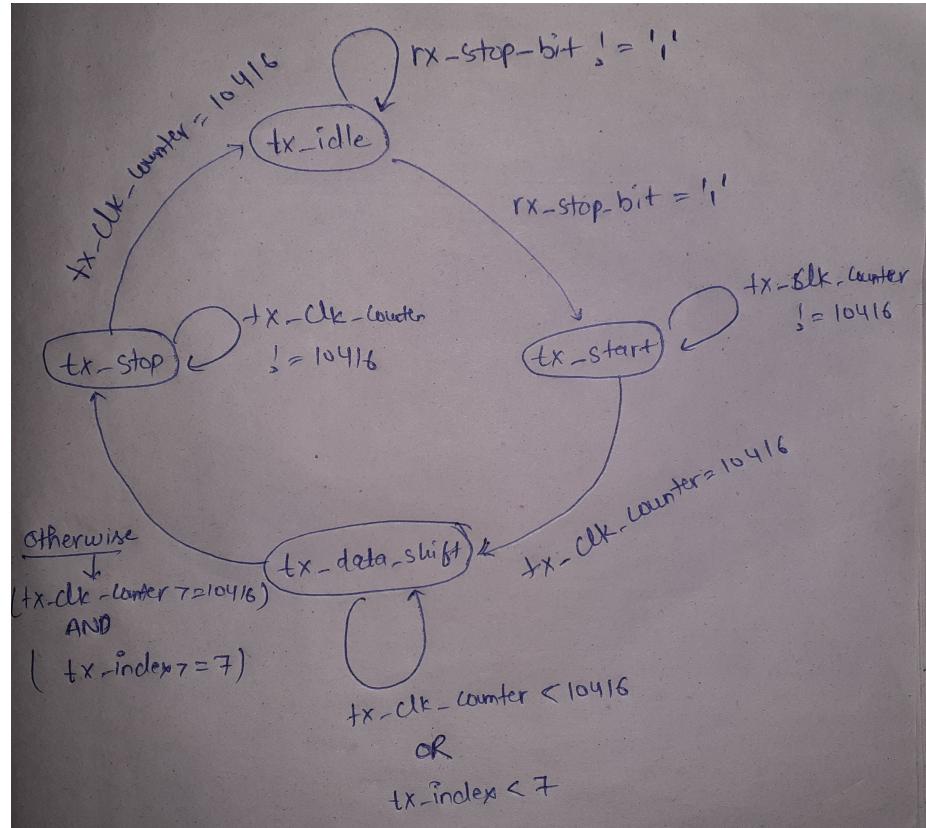
6 Expressions Derivation Truth Table

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

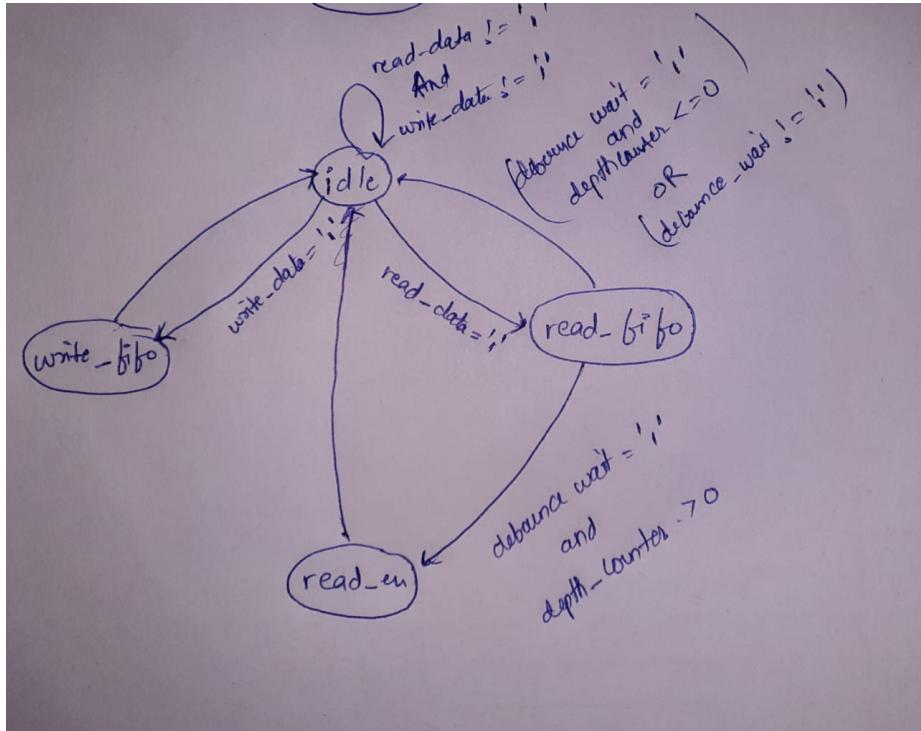
7 FSM Diagram reciever



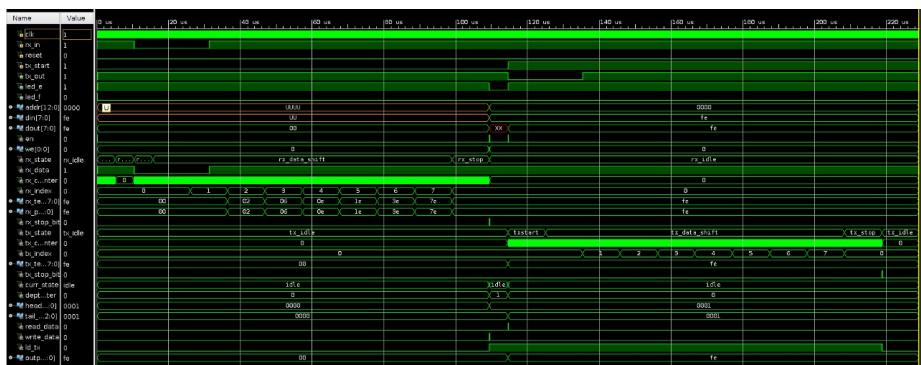
8 FSM Diagram transmitter



9 FSM Diagram



10 Simulation of the Code



11 FPGA OUTPUT

