Lab 05 – Worksheet

Name: Faaz Shahab, Mohib Ul Mehdi, Basim Hussain	ID: fs09911, mm09730, mh09732	Section: T2

Pre-Lab

Task 1

The given Verilog code is not synthesizable because the loop bound uses the variable n, which is not a constant. In synthesis, the hardware structure must be fixed at compile time, and the synthesizer must know exactly how many operations or components to generate. Since n can change dynamically, the required hardware size would also change, which is not possible on an FPGA. Therefore, this loop cannot be mapped to real hardware.
To make it synthesizable, the loop limit must be constant so that the synthesizer can create fixed hardware such as parallel adders.

Task 2

The given code is not synthesizable because it uses #10 delay to toggle the clock. Delay statements are only valid in simulation and cannot be implemented in real FPGA hardware. In actual hardware, clock signals are provided by external oscillators or FPGA clocking resources rather than arbitrary delays.
A synthesizable approach is to use the real clock signal and toggle logic on its rising edge. This ensures that the design operates synchronously and can be implemented using flip-flops in hardware.

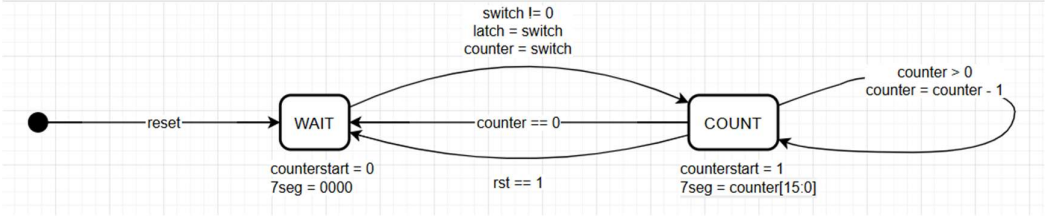
Task 3

This code describes a D flip-flop with an enable signal. On every rising edge of the clock, if the enable signal (en) is high, the output q takes the value of input d. If the enable signal is low, the output retains its previous value.
Therefore, the hardware inferred from this code is a clocked register that updates its value only when the enable signal is active. This is sequential logic because the output depends on both the clock and previous state.

Lab

Task 1

Diagram:



FSM States
WAIT (Input Waiting State): system monitors switch input while counter is disabled.
COUNT (Countdown State): system decrements the counter and displays the value on the 4 seven-segment displays.
State Transitions
Reset → WAIT
WAIT → COUNT when switch ≠ 0
COUNT → COUNT while counter > 0
COUNT → WAIT when counter = 0
Reset Behavior
When rst = 1, the system immediately returns to WAIT and clears the counter and display.
Switch Latching
On the transition from WAIT to COUNT, the switch value is latched and loaded into the counter.
Counter Control
Counter disabled in WAIT (counterstart = 0)
Counter enabled in COUNT (counterstart = 1) and decrements each clock cycle.
4 Seven-Segment Display Behavior
In WAIT: display shows 0000.
In COUNT: display shows the current counter value (counter[15:0]) across all four digits.

Task 2

Required Verilog Files attached in the folder uploaded in Github

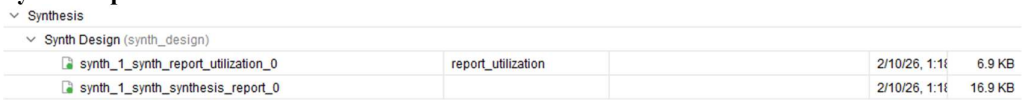
Synthesis proof:

Simulation:

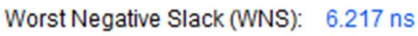
Task 3

Required Verilog Files attached in the folder uploaded in Github

Synthesis proof:



WNS:



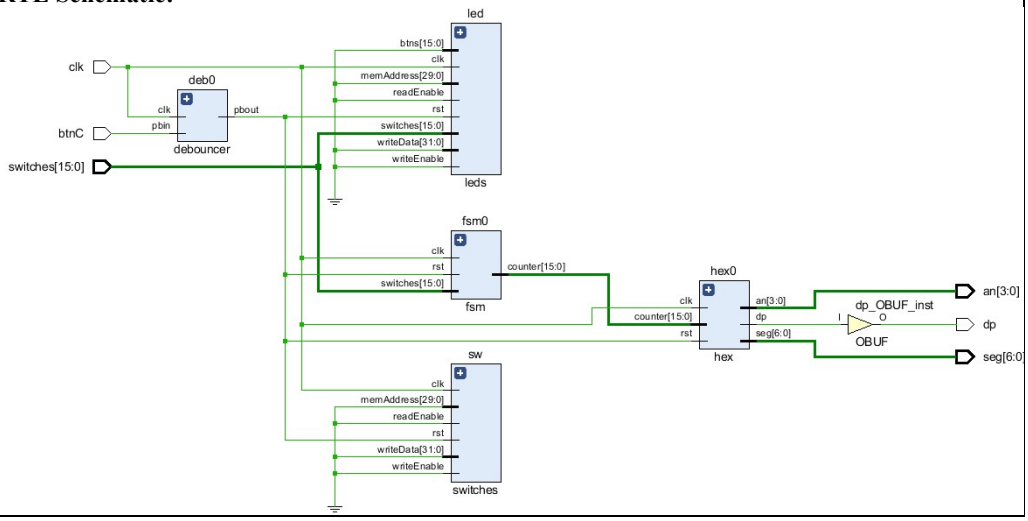
Interpretation:
The slack value is positive, therefore timing constraints are satisfied and the design operates correctly at 100 MHz.

Task 4

Required Verilog Files attached in the folder uploaded in Github

No need for synthesis proof since shown on FPGA.

RTL Schematic:



Assessment Rubric
Lab 05: Implementing Nested Procedures and Sorting

	Task No.	LR 2 Code	LR 5 Results	LR 11 (Design)
In-Lab	Pre-Lab	-	/15	-
	Task 1	-	-	/10
	Task 2	/20	/20	-
	Task 3	-	/15	-
Total Points		**/20**	**/50**	**/10**
CLO Mapped		CLO 1		

Affective Domain Rubric		Points	CLO Mapped
AR7	Report Submission & Git Upload	/10 & /10	CLO 1

CLO	Total Points	Points Obtained
1	100	
Total	**100**	

For description of different levels of the mapped rubrics, please refer to the Lab Evaluation Assessment Rubrics and Affective Domain Assessment Rubrics provided here.

Lab Evaluation Assessment Rubric

#	Assessment Elements	Level 1: Unsatisfactory	Level 2: Developing	Level 3: Good	Level 4: Exemplary
LR2	**Program/Code/ Simulation Model/ Network Model**	Program/code/simulation model/network model doesnot implement the requiredfunctionality and has several errors. The student is not able to utilize even the basic tools of the software.	Program/code/simulation model/network model has some errors and doesnot produce completely accurate results. Student has limited command on the basic tools of the software.	Program/code/simulation model/network model gives correct output but not efficiently implemented or implemented by computationally complex routine.	Program/code/simulation /network model is efficiently implemented and gives correct output. Student has full commandon the basic tools of the software.
LR5	**Results & Plots**	Figures/ graphs / tables are not developed or are poorlyconstructed with erroneousresults. Titles, captions, units are not mentioned. Data is presented in anobscure manner.	Figures, graphs and tables are drawn but contain errors. Titles, captions, units are notaccurate. Data presentation is not tooclear.	All figures, graphs, tables are correctly drawn but contain minorerrors or some of the details are missing.	Figures / graphs / tables are correctly drawn and appropriate titles/captionsand proper units are mentioned. Data presentation is systematic.
AR9	**Report Content/Code Comments**	No summary provided. The number/amount of tasks completed below the level of satisfaction and/or submitted late	Couldn't provide good summary of in-lab tasks. Some major tasks were completed but not explained well. Submission on time. Some major plots andfigures provided	Good summary of In-lab tasks. All major tasks completed except few minor ones. The work is supported by some decent explanations, Submission on time, All necessary plots, and figures provided	Outstanding Summary of In-Lab tasks. All task completed and explained well, submitted on time, good presentation of plotsand figure with proper label, titles and captions