SVCE BENGALURU

SRI VENKATESHWARA COLLEGE OF ENGINEERING

— Affiliated to VTU, Approved by AICTE, Recognised by UGC u/s 2(f) & 12(B)—

"Digital Design & Computer Organisation"

REPORT ON:

"MINI PROJECT ON 4-bit Ripple up counter"

SUBJECT CODE: BSCK302

Department of Computer Science

Sri Venkateshwara College of Engineering

Vidyanagar, Bengaluru-562157

2023-2024

BY:

NAME	USN
ADITYA KULKARNI	**1VE22CS004**
AMAR BIRADAR	**1VE22CS010**
DARSHAN B N	**1VE22CS038**
DHANAVANTHESH S	**1VE22CS043**

Abstract

This study presents the design and analysis of a 4-bit ripple up counter circuit. Counters are fundamental building blocks in digital electronics, finding applications in various fields such as digital signal processing, communications, and computing. The ripple up counter is a basic sequential logic circuit capable of incrementing its count value by one with each clock pulse. In this work, we discuss the theory behind ripple counters, including the operation principles and design considerations. We then propose a specific implementation of a 4-bit ripple up counter using flip-flops and logic gates. The design is simulated using digital design tools to verify its functionality and performance metrics such as propagation delay and power consumption. Additionally, we explore potential applications and extensions of the ripple up counter design. Overall, this study contributes to the understanding and implementation of sequential circuits, laying a foundation for further research and practical applications in digital system design.

Introduction

Counters are fundamental components in digital electronics, widely used in various applications such as frequency dividers, digital clocks, and sequential circuits. Among different types of counters, the ripple up counter is one of the simplest and most commonly employed designs.

The 4-bit ripple up counter is a sequential logic circuit capable of incrementing its count value by one with each clock pulse. It consists of four flip-flops connected in cascade, where the output of each flip-flop serves as the clock input for the next flip-flop. As the name suggests, the count ripples through the flip-flops, propagating from the least significant bit (LSB) to the most significant bit (MSB) upon each clock edge.

In this introduction, we provide an overview of the importance of counters in digital systems and the significance of the 4-bit ripple up counter specifically. We discuss its basic operation, key characteristics, and applications. Additionally, we highlight the objectives of this study, which include understanding the theory behind ripple counters, designing and simulating a 4-bit ripple up counter, and exploring potential applications and extensions of the design.

The 4-bit ripple up counter consists of four flip-flops arranged in series, where the output of each flip-flop serves as the clock input for the subsequent flip-flop. This cascading effect results in a binary count that increments by one with each clock cycle, hence the term "ripple up." The least significant bit (LSB) toggles with every clock pulse, causing a ripple effect that propagates through the counter, eventually incrementing the most significant bit (MSB) when the LSB transitions from high to low.

Objectives

1.Understanding Flip-Flop Characteristics: Gain insights into the fundamental characteristics and behaviors of T and D flip-flops, including their truth tables, timing diagrams, and functionalities in sequential circuits.

2.Study of Ripple Counters: Explore the theory behind ripple counters, including their architecture, operation principles, and advantages compared to other counter designs.

3.Design Implementation: Develop a comprehensive design plan for constructing a 4-bit ripple up counter using T and D flip-flops. This involves determining the arrangement of flip-flops, establishing clocking strategies, and selecting appropriate logic gates for feedback and input connections.

- **Logic Gate Selection:** Choose the appropriate logic gates to implement the feedback and input connections, as well as any additional control logic required for the counter. This may involve using AND, OR, XOR, and NOT gates to generate the necessary clock and control signals based on the current state of the counter.
- **Output Encoding:** Determine the desired output encoding scheme for the counter, such as binary or binary-coded decimal (BCD), and ensure that the logic gates and connections are configured to produce the correct output sequence based on the count values.
- **Simulation and Testing:** Simulate the designed counter circuit using digital design tools to verify its functionality and performance characteristics. Conduct thorough testing to ensure that the counter operates correctly under various conditions and count sequences.

4.Circuit Simulation and Verification: Utilize digital design software or hardware simulation tools to simulate the behavior of the designed counter circuit. Verify its functionality, including proper counting sequence, timing characteristics, and output stability under different operating conditions.

5.Exploration of Practical Applications: Investigate potential applications of the designed counter circuit in real-world scenarios, such as frequency division, digital clock generation, event counting, and sequence generation. Analyze the suitability of the counter design for different application requirements.

6.Documentation and Reporting: Document the entire design process, including the rationale behind design decisions, circuit schematics, simulation results, and performance analysis findings. Prepare a comprehensive report or presentation summarizing the *objectives, Aim, procedure,circuit diagram,Verilog code,system requirements, results, and conclusions of the project.*

Aim:

The aim is to implement and simulate a 4-bit ripple up counter using T and D flip-flops in Verilog, demonstrating proficiency in sequential circuit design principles and providing a practical application for further analysis in ModelSim.

The counter should increment its binary output from 0000 (0) to 1111 (15) in a specific pattern and then reset back to 0000 upon each rising edge of the clock signal.

System Requirement:

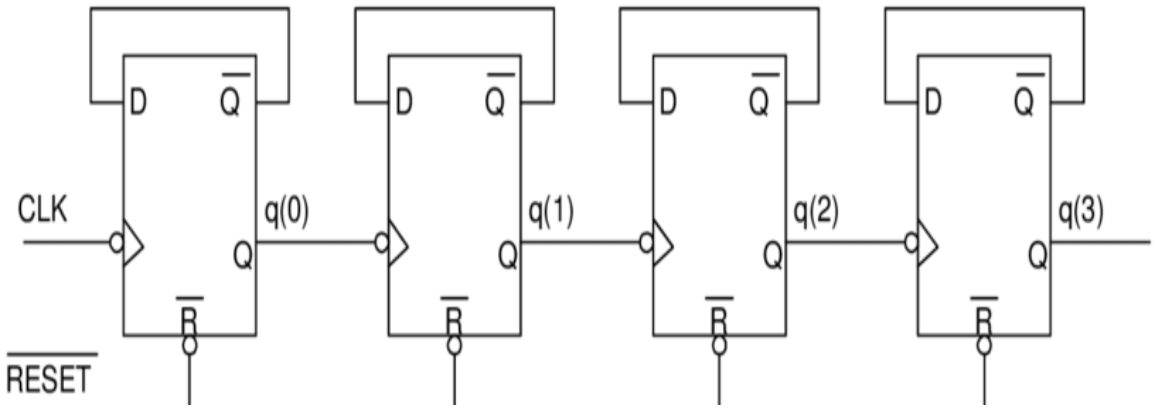
- System installed with model sim package.

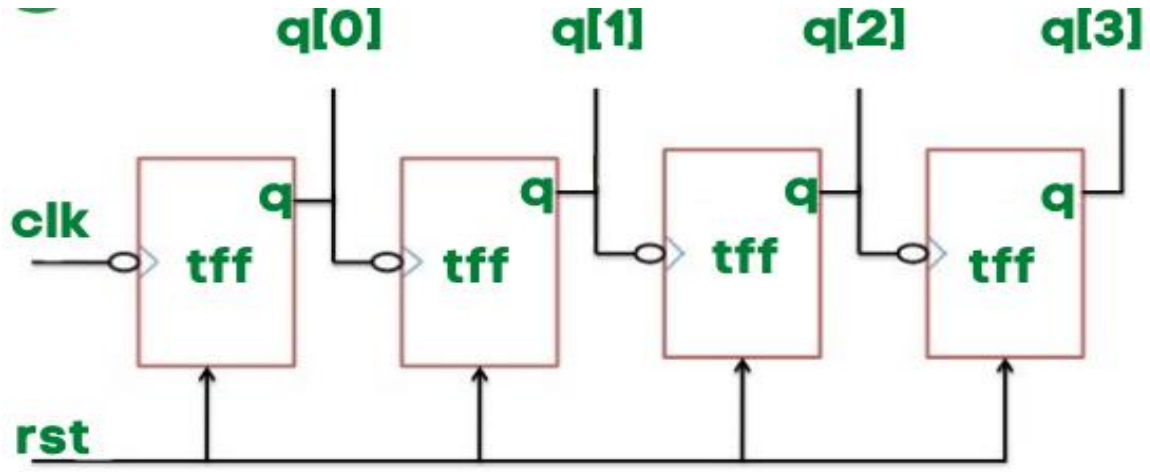
Procedure:

Procedure for designing a 4-bit ripple up counter using T and D flip-flops:

- **Step 1:** Write the Verilog code for the 4-bit ripple up counter using T and D flip-flops in ModelSim.
- **Step 2:** Draw the circuit diagram showing the arrangement of T and D flip-flops and their interconnections to implement the ripple up counter functionality.
- **Step 3:** Write Verilog code for each module (T flip-flop, D flip-flop, and the main ripple up counter module) in ModelSim.
- **Step 4:** Compile the Verilog code to check for syntax errors and ensure proper module instantiation.
- **Step 5:** Simulate the design in ModelSim by providing clock and reset signals and observing the output waveforms to verify the functionality of the ripple up counter.
- **Step 6:** Verify the timing diagrams to ensure that the counter increments correctly with each clock pulse and resets properly when the reset signal is asserted

Circuit diagram:





Implementation:

Verilog code:-

```
module ripple_carry_counter(q, clk, reset);
output [3:0] q;
input clk, reset;
T_FF tff0(q[0], clk, reset);
T_FF tff1(q[1], q[0], reset);
T_FF tff2(q[2], q[1], reset);
T_FF tff3(q[3], q[2], reset);
endmodule

module T_FF(q, clk, reset);
output q;
input clk, reset;
wire d;
D_FF dff0(q, d, clk, reset);
not n1(d, q);
endmodule

module D_FF(q, d, clk, reset);
```

output q;

input d, clk, reset;

reg q;

always @(posedge reset or negedge clk)

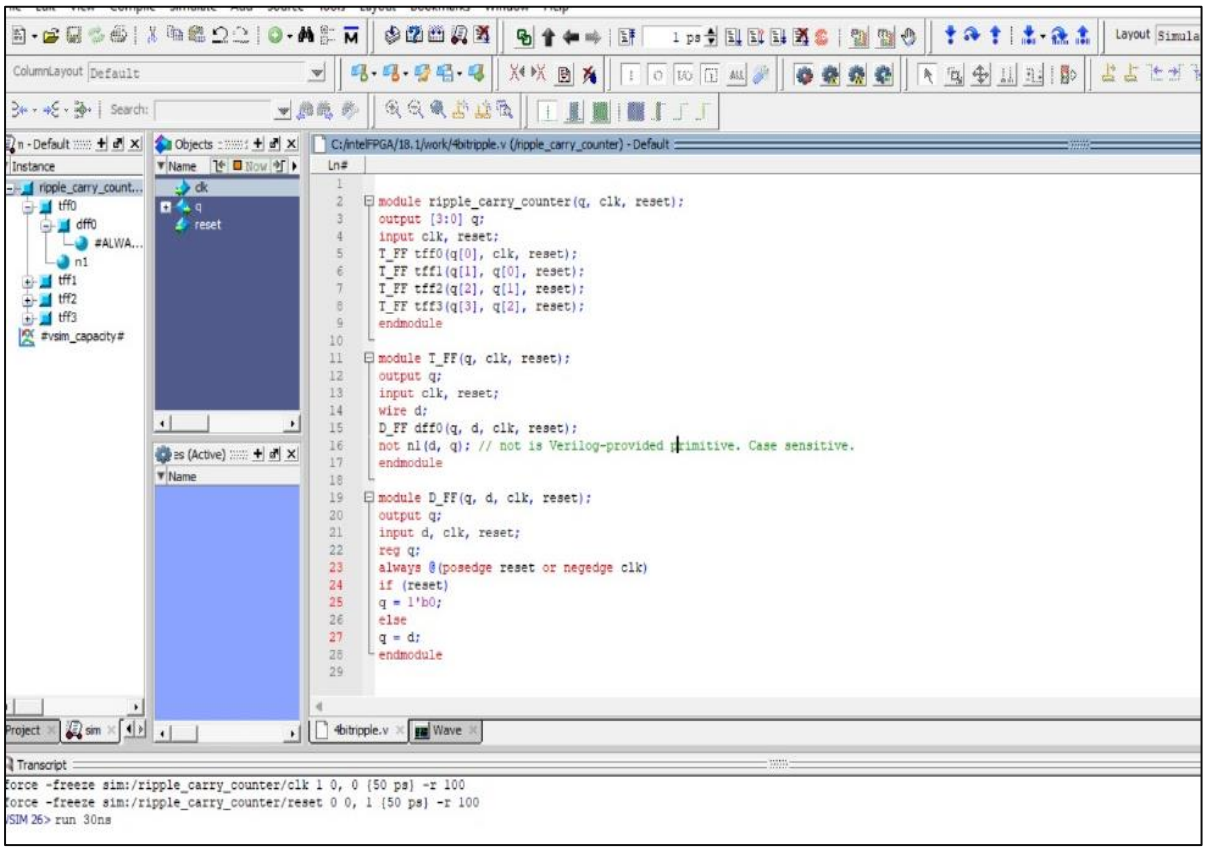
if (reset)

q = 1'b0;

else

q = d;

endmodule



Compile:

- Open ModelSim and load your project.
- Navigate to the Compile menu and select Compile All or Compile Design.
- After compilation is complete, you should see a transcript window displaying any errors or warnings encountered during compilation. You can take a screenshot of this window for documentation purposes.

Simulate:

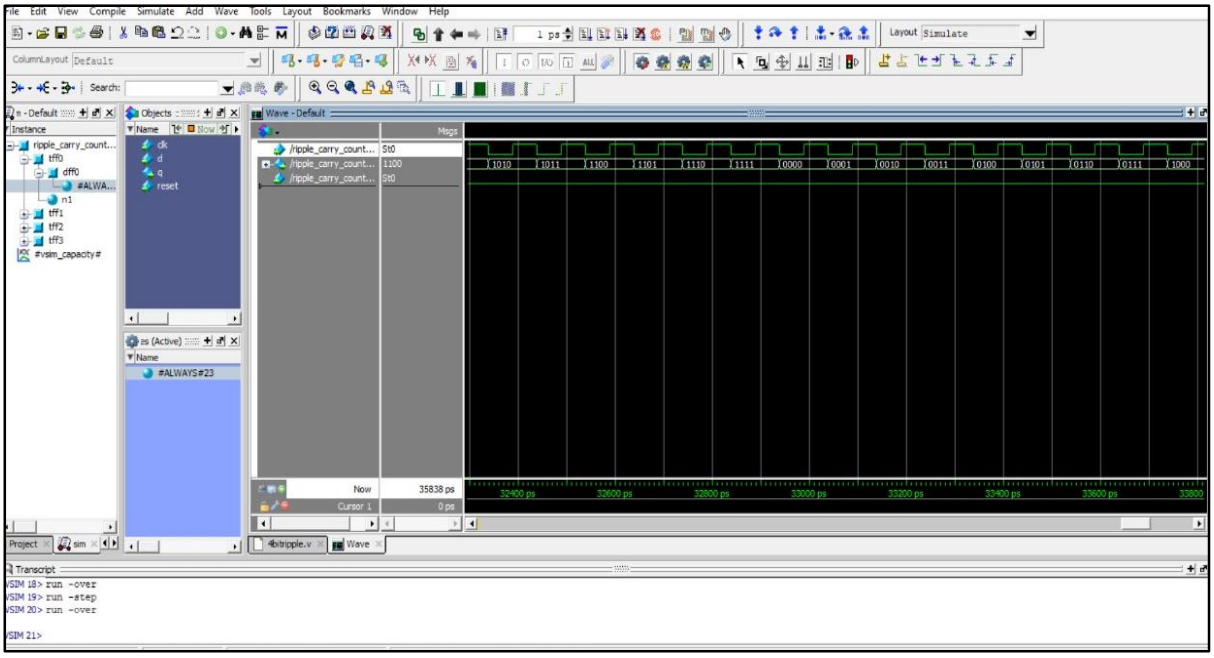
- Once your design is compiled successfully, navigate to the Simulate menu and select Start Simulation.
- Choose the appropriate simulation library and testbench/module.
- In the simulation window, add signals to the waveform viewer that you want to monitor during simulation.
- Run the simulation for the desired duration or number of clock cycles.
- After simulation is complete, you can take a screenshot of the waveform viewer displaying the simulated waveforms.

Display Waveform:

- During simulation, you can view the waveform by navigating to the Wave menu and selecting Add Wave.
- In the Add Wave window, select the signals you want to display in the waveform viewer.
- Click OK to add the selected signals to the waveform viewer.
- You can then run the simulation and observe the waveform behavior in the waveform viewer. Below is the **screenshot** of the output.

Result:-

- 4 Bit ripple up counter is realized and simulated using Verilog Modelsim.



Conclusion

In conclusion, the successful implementation and simulation of a 4-bit ripple up counter using T and D flip-flops in Verilog within ModelSim have been achieved. Through meticulous design and adherence to sequential circuit principles, the counter effectively increments its binary output from 0000 to 1111 in a consistent pattern upon each rising edge of the clock signal, resetting back to 0000 when the reset signal is asserted.

This project has demonstrated a proficient understanding of sequential circuit design principles and Verilog modeling techniques. By following the outlined procedure and meticulously designing each module, including the T and D flip-flops and the main ripple up counter module, we have realized a functional counter circuit capable of accurately counting in binary from 0 to 15.

The comprehensive simulation in ModelSim has allowed for thorough verification of the counter's functionality, including waveform analysis to ensure proper timing and behavior of the counter output. The waveform simulation results validate the expected operation of the counter, confirming its correctness and reliability.

Overall, this project serves as a practical application of theoretical knowledge in digital design, showcasing the effectiveness of Verilog and ModelSim in implementing and simulating sequential circuits. The successful realization of the 4-bit ripple up counter underscores its importance and versatility in digital system design, laying a solid foundation for further exploration and application in future projects and endeavors.