Simulink Project: Sequential Logic

Total Points: 100
Software: Simulink with Simulink Copilot

Overview

In this assignment, you will use Simulink to construct, simulate, and test sequential logic circuits. Beginning with an SR latch, you will progressively develop a D latch, D flip-flop, and 2-bit parallel-input, parallel-output register.

Each circuit will be developed into a reusable subsystem that serves as a building block for the next circuit.

You may use Simulink Copilot to assist with circuit development, troubleshoot simulation errors, and investigate unexpected behavior. Your goal is to produce functioning circuits and demonstrate an understanding of their operation.

Learning Outcomes

Upon completing this assignment, you should be able to:

1. Construct and simulate sequential logic circuits using logic gates and reusable subsystems.
2. Identify and resolve simulation errors and unexpected circuit behavior.
3. Analyze the relationship between data inputs, enable/clock signals, and outputs.
4. Develop a 2-bit register using D flip-flops with a shared clock.
5. Use Simulink Copilot to support troubleshooting and verify the effectiveness of suggested solutions.
6. Document circuit operation, simulation results, and troubleshooting decisions using Simulink annotations.

General Requirements

- Complete all four tasks in a single Simulink model.
- Use Pulse Generator blocks to test each circuit and Scope blocks to observe input and output signals.
- Create a reusable subsystem after successfully testing each circuit.
- Include annotations within your model documenting the findings requested in each task.
- Use Simulink Copilot as needed to troubleshoot errors or unexpected behavior.
- All four circuits must operate as expected before submission.

When creating subsystems, ensure that your annotations are preserved. You may select the annotations along with the circuit blocks or copy them into the subsystem after it has been created.

Part 1 – SR Latch (20 points)

Construct an SR latch using NOR gates in Simulink. Use Pulse Generator blocks to simulate the operation of the circuit and observe the outputs.

If you encounter simulation errors, use Simulink Copilot to investigate the problem and identify a solution. Continue troubleshooting until the circuit operates correctly.
Once the circuit is functioning, create an SR latch subsystem with appropriate input and output ports.

Documentation
Include an annotation addressing:

1. Did you encounter any errors when simulating the SR latch?
2. What caused the error, and what modification resolved it?

Part 2 – D Latch (20 points)
Using your SR latch subsystem from Part 1, construct a D latch with Data (D) and Enable (EN) inputs.
Use Pulse Generator blocks to test the circuit and observe how the output responds to changes in the data input and enable signal. Troubleshoot any errors and verify that the circuit operates correctly before converting it into a reusable D latch subsystem.

Documentation
Include an annotation addressing:

1. Did the D latch operate correctly, or were additional modifications necessary?
2. How does the output behave when Enable is active?
3. How does the output behave when Enable is inactive?

Part 3 – D Flip-Flop (30 points)
Using two D latch subsystems, construct a D flip-flop. The first D latch will serve as the Input Latch and the second as the Output Latch. Configure the enable signals so the two latches operate during complementary phases of the clock.

Use Pulse Generator blocks to test the circuit. Investigate and troubleshoot any unexpected behavior until the flip-flop operates correctly. Once the circuit is functioning, create a reusable D flip-flop subsystem with Data (D), Clock (CLK), and Output (Q) ports.

Documentation
Include an annotation addressing:

1. Did you observe any unexpected behavior during the simulation?
2. What modification did you make to resolve the unexpected behavior?
3. Why did this modification improve the simulation?

Part 4 – 2-Bit Parallel Register (30 points)
Using two D flip-flop subsystems, construct a 2-bit parallel-input, parallel-output register. The register should have two independent data inputs (D0 and D1), two outputs (Q0 and Q1), and a shared clock (CLK).

Use Pulse Generator blocks to test the register. Observe the relationships between the data inputs, clock signal, and outputs. Ensure that the register operates correctly before converting it into a reusable subsystem.

Documentation
Include an annotation addressing:

1. How are the clock inputs of the two flip-flops connected?
2. When do the register outputs change relative to the clock signal?
3. What happens to the outputs between clock transitions?
4. What do you observe when a data input changes at the same time as a rising clock edge?

Simulink Copilot Documentation
Include an annotation in your model summarizing how Simulink Copilot contributed to your work.

Identify any significant simulation errors or unexpected behavior for which you used Copilot, describe the recommendations it provided, and explain how you verified whether those recommendations resolved the problem.

You do not need to document every prompt or interaction.