Consider The Following Three-level NOR Circuit: (a) Find All Hazards In This Circuit. (b) Redesign The

Consider The Following Three-level NOR Circuit: (a) Find All Hazards In This Circuit. (b) Redesign The

---

Introduction

Digital logic circuits are fundamental to the operation of electronic devices, enabling complex computations and data processing. Among various logic gates, the NOR gate is particularly significant due to its functional completeness, meaning any logical function can be implemented solely with NOR gates. A three-level NOR circuit typically involves a combination of NOR gates arranged in multiple stages to perform a specific logical operation.

However, such multi-level combinational circuits are prone to hazards—undesirable transient conditions that cause momentary incorrect outputs. These hazards can be categorized primarily into static and dynamic hazards, which can compromise the reliability of digital systems. Therefore, understanding, identifying, and eliminating hazards are critical steps in digital circuit design.

This article delves into a specific three-level NOR circuit, focusing on identifying all potential hazards and proposing a redesign to mitigate or eliminate them. We will systematically analyze the circuit, classify the hazards, and then explore methods for hazard-free design.

Understanding the Given Three-level NOR Circuit

Typical Structure of a Three-level NOR Circuit

A three-level NOR circuit generally involves:

- First Level: Multiple NOR gates receiving input variables or their complements.
- Second Level: NOR gates combining outputs from the first level.
- Third Level: Final NOR gate producing the overall output.

The exact configuration depends on the specific logic function being implemented. For

illustration, consider a generic three-input scenario where the circuit computes a function F based on inputs A, B, and C.

Suppose the circuit is designed as follows:

- First-level NOR gates:
- NOR1: inputs A and B
- NOR2: inputs B and C

- Second-level NOR gate:
- NOR3: inputs from NOR1 and NOR2 outputs

- Final output:
- NOR4: inputs from NOR3 and possibly other signals (if any)

This arrangement forms a three-level hierarchy with the logic function realized through NOR operations.

Part A: Identifying All Hazards in the Circuit

Hazards are unintended fluctuations in the output caused by different propagation delays within the circuit. They can be classified mainly as:

- Static Hazards: Occur when the output should remain stable but momentarily switch due to input changes.
- Dynamic Hazards: Occur when the output changes multiple times before settling to the correct value during input transitions.

Methodology for Hazard Detection

To identify hazards, we follow these steps:

1. Determine the Boolean expression of the circuit.
2. Establish the conditions under which the output should change.
3. Analyze input transitions that can cause hazards.
4. Check for static and dynamic hazards by examining different input change scenarios.

Step 1: Derive the Boolean Expression

Assuming the first-level NOR gates:

- NOR1: outputs $Q_1 = \overline{A + B}$
- NOR2: outputs $Q_2 = \overline{B + C}$

Second-level NOR gate:

- NOR3: outputs $Q_3 = \overline{Q_1 + Q_2}$

Final output:

- $F = \overline{Q_3}$

Expressed altogether:

$$
Q_1 = \overline{A + B}
$$
$$
Q_2 = \overline{B + C}
$$
$$
Q_3 = \overline{Q_1 + Q_2}
$$

Therefore,

$$
F = \overline{Q_3} = \overline{\overline{Q_1 + Q_2}} = Q_1 + Q_2
$$

Substituting back:

$$
F = \overline{A + B} + \overline{B + C}
$$

This Boolean expression is key to analyzing hazards.

Step 2: Identify Input Conditions for Output Changes

The function:

$$
F = \overline{A + B} + \overline{B + C}
$$

can be simplified or analyzed for specific input transitions where hazards may appear.

Step 3: Analyze Potential Hazard Conditions

To identify hazards, consider transitions where only one input changes, and observe if the output remains stable or experiences a momentary glitch.

Case 1: Static Hazards

- Suppose the inputs change from a state where \(F=1 \) to another where \(F=1 \) again, but due to delays, the output temporarily drops to 0.
- For instance, consider the input sequence:

A	B	C	\(F\)	Comment
0	0	1	1	Initial state
0	1	1	0	Transition B from 0 to 1
0	1	1	1	Final state after transition

Here, during the transition at B, the output may momentarily drop, indicating a static hazard.

Case 2: Dynamic Hazards

- When multiple inputs change simultaneously, the output may oscillate before stabilizing.
- For example, changing A from 0 to 1 and C from 1 to 0 simultaneously could cause multiple glitches.

Step 4: Practical Hazard Detection Techniques

- K-Map Analysis: Using Karnaugh maps to identify regions where the output should be stable but might experience hazards.
- Timing Simulation: Circuit simulation with propagation delays to observe glitches.
- Boolean Difference Method: Calculating differences in the Boolean function with respect to input changes to locate potential hazards.

Part B: Redesigning the Circuit to Eliminate Hazards

Once hazards are identified, the goal is to modify the circuit to eliminate or mitigate them,

ensuring reliable operation.

Strategies for Hazard-Free Design

- Adding Redundant Logic: Incorporate additional gates to cover potential hazard conditions.
- Using Consensus Terms: Simplify Boolean expressions to include terms that prevent hazards.
- Implementing Hazard-Free Logic Forms: Use specific logic design techniques like the Quine-McCluskey method.

Step 1: Simplify the Boolean Expression

From earlier:

$$
F = \overline{A + B} + \overline{B + C}
$$

Applying Boolean identities:

$$
F = (\overline{A + B}) + (\overline{B + C}) = (\overline{A} \cdot \overline{B}) + (\overline{B} \cdot \overline{C})
$$

Factor out $\overline{B}$:

$$
F = \overline{B} (\overline{A} + \overline{C})
$$

This simplified form reveals that the function is true when $B=0$ and either $A=0$ or $C=0$.

Step 2: Implement Hazard-Free Logic

To prevent hazards, include consensus terms that cover all critical transitions. The minimal expression:

$$
\[

$$
F = \overline{B} (\overline{A} + \overline{C})
$$

can be implemented with:

- One NOR gate for $\overline{A}$ (NOT A)
- One NOR gate for $\overline{C}$ (NOT C)
- One AND gate to combine $\overline{B}$ and $(\overline{A} + \overline{C})$

Implementation steps:

1. Generate $\overline{A}$ and $\overline{C}$ using inverters.
2. OR $\overline{A}$ and $\overline{C}$.
3. Generate $\overline{B}$ using an inverter.
4. AND $\overline{B}$ with the OR output.

This approach ensures that the output remains stable during input transitions, eliminating static hazards.

Step 3: Use of Hazard-Free Logic Forms

The disjoint sum-of-products (DSOP) form can be employed to design hazard-free circuits by ensuring no overlapping minterms that could cause hazards. Alternatively, techniques like covering all minterms explicitly or adding redundant terms can help.

Step 4: Additional Techniques

- Delay Matching: Ensure that signals arrive simultaneously at the gates to prevent glitches.
- Use of Gate-Level Redundancy: Incorporate additional gates that produce the same logical function but help cover potential hazard scenarios.
- Synchronization: Employ flip-flops or latches to stabilize outputs during rapid input changes.

Conclusion

Designing hazard-free digital circuits requires careful analysis and strategic modifications. In the context of a three-level NOR circuit, hazards can arise from various sources, including static and dynamic effects caused by propagation delays and input transitions. By deriving the Boolean expression, analyzing potential transition scenarios, and simplifying the logic,

designers can identify the specific conditions leading to hazards.

Redesigning the circuit using hazard-free logic techniques, such as incorporating consensus terms, adding redundant logic, and ensuring

Frequently Asked Questions

What types of hazards should be identified in a three-level NOR circuit, and how do they affect circuit operation?

In a three-level NOR circuit, both static and dynamic hazards can occur. Static hazards cause unwanted switching when inputs change but outputs should remain stable, while dynamic hazards cause multiple unwanted transitions. Identifying these hazards is crucial to ensure reliable circuit operation.

How can static hazards be detected in a three-level NOR circuit, and what design strategies can eliminate them?

Static hazards can be detected by analyzing input transitions that do not change the output but might cause unwanted transitions due to differing propagation delays. To eliminate them, techniques such as adding redundant logic or ensuring that all input paths are balanced can be employed.

What are the common methods to redesign a three-level NOR circuit to eliminate hazards?

Redesign methods include adding redundant logic to cover critical input transitions, restructuring the circuit to minimize delay differences, or using hazard-free logic synthesis techniques. These approaches help ensure stable output without glitches.

Can you explain the process of hazard analysis in the context of a three-level NOR circuit?

Hazard analysis involves examining the circuit's truth table and input transitions to identify potential static or dynamic hazards. This includes analyzing the propagation delays along different paths and determining where unintended switching might occur during input changes.

What specific steps are involved in redesigning a three-level NOR circuit to consider hazard elimination?

The steps include: 1) Analyzing the existing circuit for hazards, 2) Identifying critical input

transitions that cause hazards, 3) Adding redundant logic or adjusting gate configurations to cover these transitions, and 4) Verifying the updated circuit for hazard-free operation through simulation or formal methods.

Are there automated tools or software that can assist in hazard detection and redesign of three-level NOR circuits?

Yes, several electronic design automation (EDA) tools and software, such as Logic Friday, Synopsys PrimeTime, and others, can perform hazard detection and suggest circuit modifications. These tools analyze the circuit's timing and logic to identify hazards and aid in hazard-free redesign.

Additional Resources

Consider The Following Three-level NOR Circuit: (a) Find All Hazards In This Circuit. (b) Redesign The

In the realm of digital electronics, ensuring the stability and reliability of logic circuits is paramount. As digital systems become increasingly complex, the presence of hazards—undesirable switching transients—poses a significant challenge. These hazards can lead to erroneous outputs, compromising system integrity. Today, we delve into a specific three-level NOR circuit, exploring how to identify all potential hazards within it and subsequently redesign the circuit to mitigate these issues. This comprehensive analysis aims to equip engineers and students alike with the insights needed to optimize their digital designs for safety and performance.

Understanding the Three-Level NOR Circuit

Before identifying hazards, it’s essential to understand the structure of the circuit in question. A three-level NOR circuit typically involves multiple NOR gates arranged in a cascade, with the outputs of some gates feeding into others, forming a multi-tiered logic network.

Basic Structure

- Level 1: The input stage, where primary signals are fed into NOR gates.
- Level 2: Intermediate NOR gates that process outputs from Level 1.
- Level 3: The final NOR gate(s) producing the circuit's output.

This configuration is common in combinational logic circuits designed to implement specific Boolean functions. The cascading structure allows complex logic functions to be realized using simple NOR gates, which are functionally complete.

Assumed Circuit Configuration

For our analysis, consider a simplified version of a three-level NOR circuit with the following characteristics:

- Inputs: A, B, C, D
- Level 1: Two NOR gates:
- NOR1: inputs A and B
- NOR2: inputs C and D
- Level 2: One NOR gate:
- NOR3: inputs from NOR1 and NOR2 outputs
- Level 3: Final NOR gate:
- NOR4: input from NOR3, producing the output Y

This setup is illustrative but captures the essential features pertinent to hazard analysis.

Part A: Identifying All Hazards in the Circuit

What Are Hazards?

Hazards in digital logic are unintended fluctuations or glitches in the output that occur during transitions of input signals. They are typically classified into:

- Static hazards: Occur when the output is supposed to remain stable but temporarily changes.
- Dynamic hazards: Occur during transitions when the output changes more than once before settling.

Understanding and identifying these hazards is critical in designing fault-free digital systems.

Analyzing the Circuit for Hazards

To find all hazards, we examine the circuit's behavior during input transitions, particularly focusing on changes that can cause glitches at the output.

Step 1: Identify the Boolean Function

Given the configuration:

- NOR1 = ¬(A + B)
- NOR2 = ¬(C + D)
- NOR3 = ¬(NOR1 + NOR2) = ¬(¬(A + B) + ¬(C + D))
- Y = ¬(NOR3)

Simplifying NOR3:

NOR3 = ¬(¬(A + B) + ¬(C + D))
Using De Morgan's theorem:

NOR3 = (A + B) · (C + D)

Therefore, the overall function:

$Y = \neg(NOR3) = \neg[(A + B) \cdot (C + D)]$

This is a product of sums form, and the circuit implements the function:

$Y = \neg(A + B) + \neg(C + D)$

which simplifies to:

$Y = (\neg A \cdot \neg B) + (\neg C \cdot \neg D)$

This helps in understanding how input transitions affect the output.

Step 2: Determine Input Transition Scenarios

Hazards often occur during specific input changes, such as:

- When only one input changes from 0 to 1 or vice versa.
- When multiple inputs change simultaneously.

Key scenarios to analyze include:

- Transition of A or B from 0 to 1 or 1 to 0
- Transition of C or D similarly

For instance, consider the case where:

- A and B are initially 0, C and D are 1
- A changes from 0 to 1

Initial State:

- A=0, B=0, C=1, D=1

Final State:

- A=1, B=0, C=1, D=1

Calculating initial and final outputs:

- Initial:

- $(\neg A \cdot \neg B) = (1 \cdot 1) = 1$

- $(\neg C \cdot \neg D) = (0 \cdot 0) = 0$

- $Y = 1 + 0 = 1$

- After A changes to 1:

- $(\neg A \cdot \neg B) = (0 \cdot 1) = 0$

- $(\neg C \cdot \neg D)$ remains 0

- $Y = 0 + 0 = 0$

The output switches from 1 to 0 smoothly, indicating no hazard here. But issues may arise during other transitions where the intermediate signals cause glitches.

Step 3: Identifying Static Hazards

Static hazards occur when the output should remain constant but momentarily changes during input transitions. For example:

- When changing inputs from (A=1, B=1, C=0, D=0) to (A=0, B=1, C=0, D=0), the output should stay at 0 but may glitch to 1 temporarily.

This can happen when the logic signals propagate at different speeds, causing momentary mismatches.

Method to Detect Static Hazards:

- Identify when the output function depends on multiple inputs changing simultaneously.
- Check if any path delays could cause a brief inconsistency.

In our circuit, static hazards are likely if the variables are not synchronized or if the circuit isn't properly designed with hazard mitigation in mind.

Step 4: Identifying Dynamic Hazards

Dynamic hazards are more subtle, occurring when an input transition causes multiple transitions in the output before settling.

For example:

- Changing C from 0 to 1 while other inputs are held steady can cause a momentary glitch due to the different paths and delays in the circuit.

Summary of Hazards

Based on the analysis, the hazards in this three-level NOR circuit include:

- Static hazards during transitions where multiple inputs change, especially in the product of sums form.
- Dynamic hazards when multiple inputs change simultaneously, leading to multiple output transitions before settling.

Practical implication: Without proper design techniques, these hazards can cause transient errors, especially in high-speed or sensitive digital systems.

Part B: Redesigning the Circuit to Eliminate Hazards

Having identified potential hazards, the next step involves redesigning the circuit to eliminate or significantly reduce them, ensuring stable and reliable operation.

Strategies for Hazard Reduction

Several techniques are used in digital design to mitigate hazards:

1. Adding Redundant Logic: Incorporating additional gates or connections to cover the hazard scenarios.
2. Using Hazard-Free Logic Forms: Transforming the Boolean functions into forms less susceptible to hazards, such as the sum-of-products or product-of-sums with specific redundancies.
3. Implementing Delay-Equalization: Ensuring signals propagate through the circuit at similar speeds.
4. Employing Edge-Triggered Elements: Using flip-flops or registers to synchronize signals, reducing dynamic hazards.

Redesign Approach for the Given Circuit

Given the Boolean function:

$Y = (\neg A \cdot \neg B) + (\neg C \cdot \neg D)$

Objective: Modify the circuit to prevent glitches during input transitions.

Step 1: Use a Hazard-Free Logic Expression

The current expression is in sum-of-products form, which is generally hazard-prone during certain transitions. To make it hazard-free, we can apply consensus theorem and include redundant terms where necessary.

Redundant Term Addition:

- Add a term that covers the transition scenarios where hazards are likely:

$Y = (\neg A \cdot \neg B) + (\neg C \cdot \neg D) + (\neg A \cdot \neg B \cdot \neg C \cdot \neg D)$

This redundant term ensures that during transitions, the output remains stable because the logic covers all potential hazard conditions.

Step 2: Implement Using Programmable Logic Devices

Modern digital design often employs programmable logic devices that facilitate hazard-free implementation:

- Use of Programmable Array Logic (PAL) or Complex Programmable Logic Devices (CPLDs): These devices can be programmed to implement the hazard-free logic expression, ensuring stable outputs.
- Inserting Buffer Stages: Place buffers at critical points to synchronize signals, minimizing

delay differences.

Step 3: Synchronize Inputs

Implement input synchronization:

- Use flip-flops or registers to latch inputs, preventing glitches caused by asynchronous transitions.
- This approach converts asynchronous input changes into synchronized signals, reducing dynamic hazards.

Step 4: Equalize Path Delays

- Adjust physical layout or buffer sizes to equalize signal propagation delays.
- Ensure that all signals reach their destination gates at roughly the same time.

Final Redesigned Circuit Features

- Redundant Logic Terms: To prevent static hazards.
- Input Synchronization: Using flip-flops for inputs.
- Balanced Path Delays: To mitigate dynamic hazards.
- Hazard-Free Logic Implementation: Using minimal redundant logic to maintain efficiency.

Benefits of the Redesigned Circuit

- Enhanced Stability: Eliminates glitches that could cause erroneous outputs.
- Increased Reliability: Suitable for high-speed digital systems.
-

Consider The Following Three Level Nor Circuit A Find All Hazards In This Circuit B Redesign The

Consider The Following Three Level Nor Circuit A Find All Hazards In This Circuit B Redesign The

Related Articles

- Correct The Four Pronoun-related Errors In This Paragraph: Fast-forwarding To Modern Times, The Heyday
- Coastal Climate Change Question 25 Of 25 1 Point True Or False: Changes In Sea Surface Temperature Can
- Describe Your Professional, Academic, Or Life Experience And Skills You Have That Enhance Your Ability

Back to Home