Journal of Electrotechnology, Electrical Engineering and Management (2025)
Clausius Scientific Press, Canada
DOI: 10.23977/jeeem.2025.080117
ISSN 2560-6697 Vol. 8 Num. 1

An Investigation of Race Hazard Elimination in Digital Counter Circuits

Mengzhe Liang

Moorestown Friends School, 110 E Main St, Moorestown, NJ 08057, USA
pro.mengjia12105@gmail.com

Keywords: Race Condition; Logic Hazard; Modulo-13 Counter; Capacitor Filtering; 74ALS175M; Multisim

Abstract: Race hazards, which can lead to premature asynchronous resets or erroneous state transitions, occur when concurrent signals with different propagation delays arrive at the nodes of a combinational logic circuit. This paper investigates a modulo-13 counter centered on the SN74LS161N IC. A critical race hazard scenario is constructed in Multisim by intentionally introducing signal delays, and two common suppression strategies are evaluated: (1) connecting a parallel capacitor at the reset node to form an RC low-pass filter for analog pulse smoothing; and (2) using a 74ALS175M flip-flop to edge-register the outputs before the reset logic in a structured timing approach. Simulation results demonstrate that the RC filter significantly attenuates narrow pulses at low frequencies but introduces reset latency. In contrast, the registering method transforms sub-cycle instability into clean edge-sampling, trading a one-clock-cycle delay for enhanced robustness and scalability. This study provides a practical reference for the engineering implementation of race hazard elimination in counter circuits.

1. Introduction

In the design of modern digital circuits, even nanosecond-level timing deviations can lead to unpredictable changes in the logical state of a system, and this sensitivity is particularly pronounced in sequential circuits such as counters, state machines, or register chains. One of the most destructive timing-related issues is race hazard, which occurs when two or more signals act on a single logic node almost simultaneously, and the final output depends on the chronological order in which the signals arrive. This phenomenon is typically influenced by factors such as gate delay, uneven wiring, capacitive loading, or chip manufacturing variations [1].

In ideal simulation environments or textbook circuits, race hazards are usually simplified or ignored. However, in practical high-speed systems—such as embedded controllers, counter chains, or real-time processing modules—even the instability of a single output bit can cause incorrect state transitions in the entire system, leading to false resets, counting interruptions, or even logical collapse [2]. Such problems are highly concealed and difficult to predict during the initial design phase; their severity often only becomes apparent after the circuit is put into actual operation. Therefore, proactively identifying and eliminating potential hazardous paths during the design phase is a crucial task for ensuring the stability and robustness of digital systems. This study selects a typical

synchronous circuit—a modulo-13 counter based on the SN74LS161N as the analysis and testing platform. A simulation model is constructed in the Multisim environment, and signal delays are intentionally introduced to create race hazards. Two common hazard elimination strategies are then implemented separately to observe the circuit behavior and changes in its waveforms.

The two hazard elimination techniques discussed in this paper include: capacitive filtering based on RC characteristics, and D flip-flop output buffering for edge alignment. By structurally comparing their effectiveness in hazard suppression, timing stability, and logical clarity, this paper aims to provide a reliable reference framework for engineering practice and conduct preliminary verification for their application in more complex systems in the future.

2. Principles and Methods

2.1. Race Hazards in Digital Circuits

A race hazard in a digital circuit is a condition where two or more signals change concurrently along the same logic path and, due to their different propagation delays, cause an unpredictable transition in the output state. This phenomenon is particularly common in circuits where multiple logic conditions jointly control a state transition or a reset signal, and it can lead to severe consequences in high-speed clocked systems [3].

Generally, race hazards can be classified into two types: non-critical hazards, which only manifest as transient glitches that do not affect the circuit's final state, and critical hazards, which can cause erroneous state machine transitions, premature or delayed resets, or even logic deadlocks [2]. The root cause lies in the non-uniform propagation delays of combinational logic gates. For instance, some output paths may pass through multiple gate levels or longer interconnects, failing to achieve logical stability before the system clock edge arrives, thus causing an erroneous decision in the subsequent logic stage.

In the reset logic of the modulo-13 counter, the reset signal is typically generated by combining the three outputs (e.g., QA, QB, and QD) corresponding to the count "1101" through an AND-gate combination. If one of these signals (e.g., QB) experiences an additional delay while the other two have already reached a high level, the reset signal can be prematurely pulled low, triggering a false reset. This premature reset phenomenon, caused by a timing mismatch, is a typical example of a critical race hazard.

Because race hazards are highly latent and not easily exposed during functional simulation, they must be proactively prevented and eliminated through careful structural circuit design or signal path control measures.

2.2. Principles of Elimination Methods

1) Principle of Capacitor Filtering

Capacitor filtering is a technique that utilizes an RC low-pass filter to smooth the edges of digital signals, primarily for suppressing high-frequency glitches or transient pulses. In reset logic, if a narrow pulse is generated at a combinational logic output due to a hazard, this pulse can be "absorbed" or smoothed by the capacitor, thus preventing it from affecting subsequent flip-flops or state machines [4].

In this study, by connecting a 100nF capacitor between the reset signal output and ground, the rise and fall times of the logic transition are effectively extended. This prevents short hazard pulses of insufficient width from crossing the logic decision threshold, thereby achieving physical suppression of the race hazard. However, this method also introduces a certain response delay, and its use in high-speed circuits requires a careful trade-off.

Mathematically, an RC filter can be treated as a first-order low-pass network with a time constant of:

$$\tau = R_s \cdot C \; |H(j\omega)| = \frac{1}{\sqrt{1 + (\omega R_s C)^2}}$$

When the input pulse width $t_p \ll \tau$, the output amplitude is approximately:

$$V_{out} \approx V_{in} * \left(1 - e^{\left\{-\frac{t_p}{\tau}\right\}}\right) \approx \frac{t_p}{\tau} V_{in}$$

2) D-Type Flip-Flop Edge-Capturing Mechanism

A D-type flip-flop is a sequential logic element triggered by a clock edge and possesses excellent glitch immunity. Unlike the physical smoothing of capacitor filtering, a D-type flip-flop samples the input signal through a structured timing mechanism, updating its state only on the rising (or falling) edge of the clock. This avoids incorrect sampling caused by input instability within the clock cycle.

In the circuit used in this study, an additional D-type flip-flop is used to buffer the counter's outputs, delaying the stable output value by one clock cycle before it is passed to the reset logic. By doing so, even if transient transitions caused by hazards exist in the original outputs, the D-type flip-flop can mask these glitches and only output the value that was stable before the clock edge, thereby enhancing the robustness of the entire logic chain [5].

$$t_{setup} \leq t_{clk} - t_{signal} \quad , \quad t_{hold} \geq t_{signal} - t_{clk}$$

From a timing theory perspective, for a flip-flop to sample correctly, the input signal must satisfy the setup time requirements relative to the clock edge. The data must be stable for the duration of the setup time before the clock edge and for the duration of the hold time after the clock edge.

3. Simulation Platform and Circuit under Test

In this study, a modulo-13 counter was used as the research object for investigating race hazards, with Multisim serving as the simulation platform. To construct the modulo-13 counter, the classic SN74LS161N chip model from Texas Instruments (TI) was selected. This chip is a 4-bit synchronous binary counter featuring key interfaces including a clock input (CLK), a clear terminal (CLR), an enable terminal (EN), and four output terminals (QA ~ QD). It adopts a rising-edge triggering mechanism, where all state changes are synchronized with the system clock [6].

The key to implementing modulo-13 operation lies in accurately detecting the moment when the count value reaches 1101 (decimal 13) and immediately resetting the counter. For this purpose, a 3-input NAND gate was employed to perform combined detection on the counter outputs {QA, QB, QD}. When all three outputs are at a high level simultaneously, the NAND gate produces a short low pulse, which directly drives the active-low CLR terminal of the 74LS161 to complete the reset operation.

Since the three outputs QA, QB, and QD do not change "instantly" at the same time—instead, their changes depend on the propagation of delays through their respective internal gates—if the delay of any one bit is slightly longer (e.g., QB passing through an additional logic path), the CLR terminal may be triggered prematurely. This gives rise to a race hazard.

On the Multisim platform, the basic circuit was constructed with a clock frequency set to 1 kHz. A 4-channel virtual oscilloscope was used to monitor the clock signal, CLR signal, QD output, and the output of the NAND gate in the reset logic, respectively. Preliminary observations (Figure 1) showed that without any delay control measures, the counter could operate stably up to the count

value 1100 (decimal 12). However, once an artificial delay was introduced into the QB path via NOT gates (U8, U9), a brief low-level pulse of the reset signal could be observed before the expected time point. This caused the counter to unexpectedly skip the count value 13 and reset directly. This behavior indicates that the race hazard has been triggered, providing an observable and reproducible baseline sample for the subsequent testing of elimination strategies.

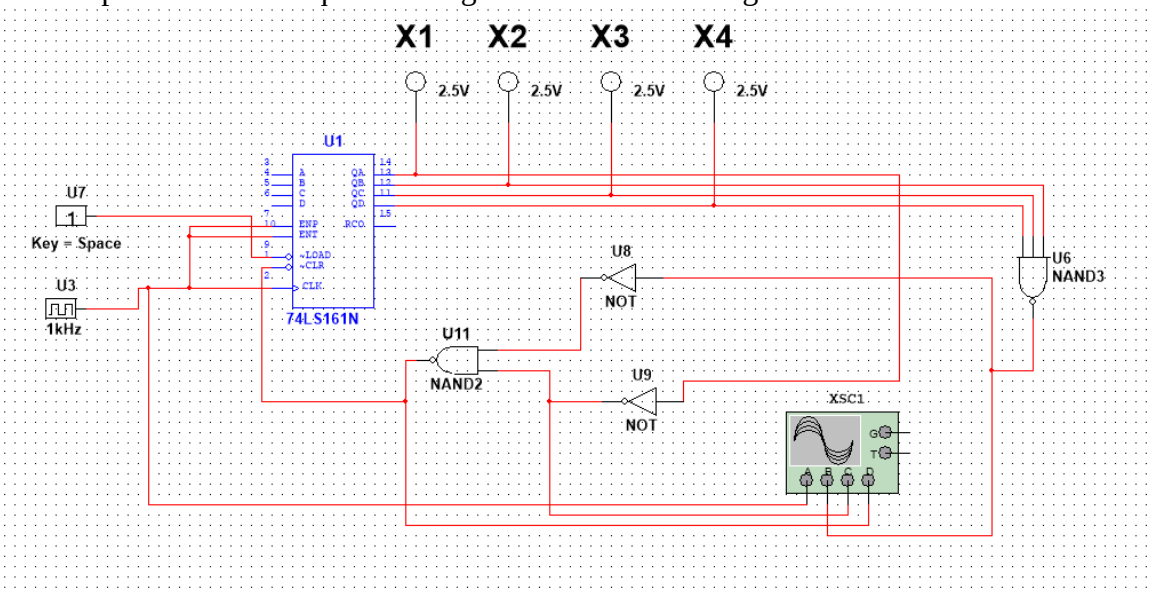

Figure 1 Schematic Diagram of the Baseline Circuit (Without Suppression) Baseline circuit description: U1 = SN74LS161A; U6 = 3-input NAND; Additional delay is introduced into the QB path via U8 and U9; XSC1 (oscilloscope) is connected to RESET/QA/QB_delay/QD. VCC = 5 V, fCLK = 1 kHz.

Although the concept of race hazards is well-defined in theory, creating a clear and observable hazard phenomenon in a practical circuit is not a trivial task. Most standard logic ICs, such as the 74LS series, have internal gate delays, but these are often insufficient to trigger obvious hazardous behavior under normal simulation conditions. Therefore, to ensure the verifiability of the subsequent hazard elimination techniques, this study adopted a strategy of "proactively designing a flaw" by intentionally introducing signal delay differences into the original modulo-13 counter circuit to construct a reproducible critical hazard.

The key to this modification lies in the three-input logic gate of the reset logic. As previously described, the three inputs to this gate are connected to the counter outputs QA, QB, and QD, respectively. We introduced a delay path for one of the inputs, QB, by using a pair of inverters (NOT gates), causing this signal to arrive at the logic gate slightly later than the other two signals. This minute timing skew becomes the entry point for the hazard to occur. When the counter transitions from 12 (1100) to 13 (1101), because QA and QD have already transitioned to 1 while QB is still at a low level, the reset logic can "misinterpret" this as the target state, prematurely outputting a high level and triggering the counter's clear function

This behavior is particularly intuitive in the Multisim simulation. By adding an oscilloscope channel between the logic gate's output and the CLR pin, we clearly observed the reset signal being briefly pulled low before the count reached 13, causing the counter to jump directly from 12 back to 0, skipping the 13th state that should have existed. This "premature reset" behavior is not a functional error but is precisely the critical hazard produced by the combination of gate delay differences.
Furthermore, to enhance the observational comparison of the hazard phenomenon, we set up two versions of the test circuit: one with the original structure where all inputs are connected directly, and

another with the delay structure where QB passes through two series-connected NOT gates before entering the logic gate. Running both under the same simulation conditions (1kHz clock frequency and 5V), a comparison of their waveforms reveals a clear premature logic transition and glitch phenomenon. This comparison forms the control variable baseline for verifying the various hazard elimination techniques that follow.

Through the circuit modifications in this section, we not only successfully constructed an experimental environment with a realistic hazard problem but also demonstrated that the risk of systemic logic instability is hidden within seemingly trivial gate-level differences. This methodology of "intentionally designing instability" also provides an effective experimental platform for educational, testing, and verification-based research.(Figure 2)

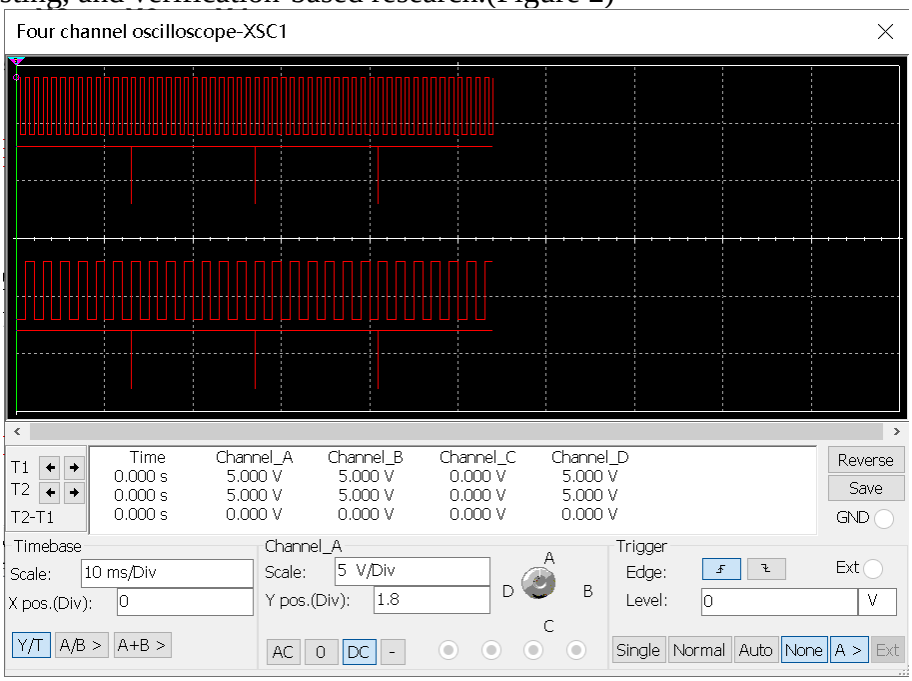

Figure 2 Baseline Oscilloscope Waveform (Narrow low pulse at RESET node) - Baseline Waveform: At the 12→13 transition, a narrow low pulse appears on RESET (premature clear), and the 13th state is skipped. XSC1: 5 V/div, 10 ms/div.

4. Simulation Results

In the preceding experiments, a race hazard phenomenon was successfully induced in the modulo-13 counter by intentionally introducing input delays. To further analyze and verify the effectiveness and applicability of different elimination techniques in a practical circuit, this section will detail the implementation of two strategies. They target different mechanisms of hazard occurrence, making improvements at the structural, timing, or synchronization levels to block the propagation of the hazard signal at its source or along its path.

The first method is capacitor filtering. This strategy is the most straightforward in terms of circuit structure, involving the connection of a small-value capacitor (typically 100nF) in parallel between the logic gate's output (the reset signal) and ground to form an RC low-pass filter. Its working principle is as follows: if a very narrow pulse exists in the reset signal, the capacitor will "buffer" it during its charge or discharge cycle, smoothing its waveform and preventing it from reaching the triggering threshold of the SN74LS161N. This filter can effectively suppress isolated glitches caused by delay mismatches, making it suitable for control nodes driven by combinational logic outputs (Figure 3).

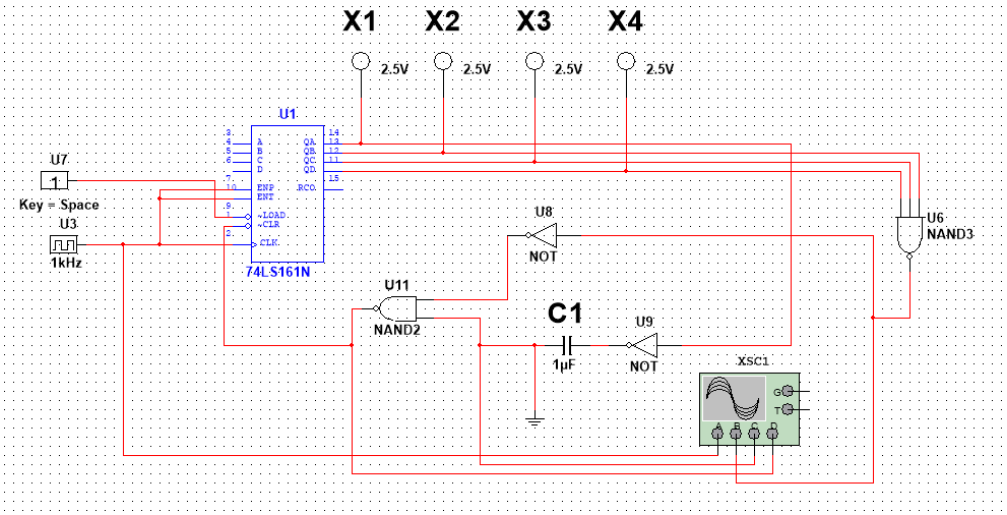

Figure 3 Capacitor Method Circuit Schematic (Example with C=1 µF) - RC Suppression Scheme: A C1=1 µF capacitor is connected in parallel from RESET to ground, forming a first-order low-pass filter with the upstream equivalent source impedance.

In the Multisim simulation, the reset signal waveform becomes significantly smoother after adding the filter, and it no longer erroneously triggers the CLR pin when the count is less than 13, greatly mitigating the hazard phenomenon. However, the side effects of RC filtering also become apparent: when the count reaches 13, the reset signal response shows a slight delay, causing the clear operation to lag behind the ideal state by one clock cycle. This delay is acceptable in low-speed systems but must be carefully evaluated in high-speed logic (Figure 4).

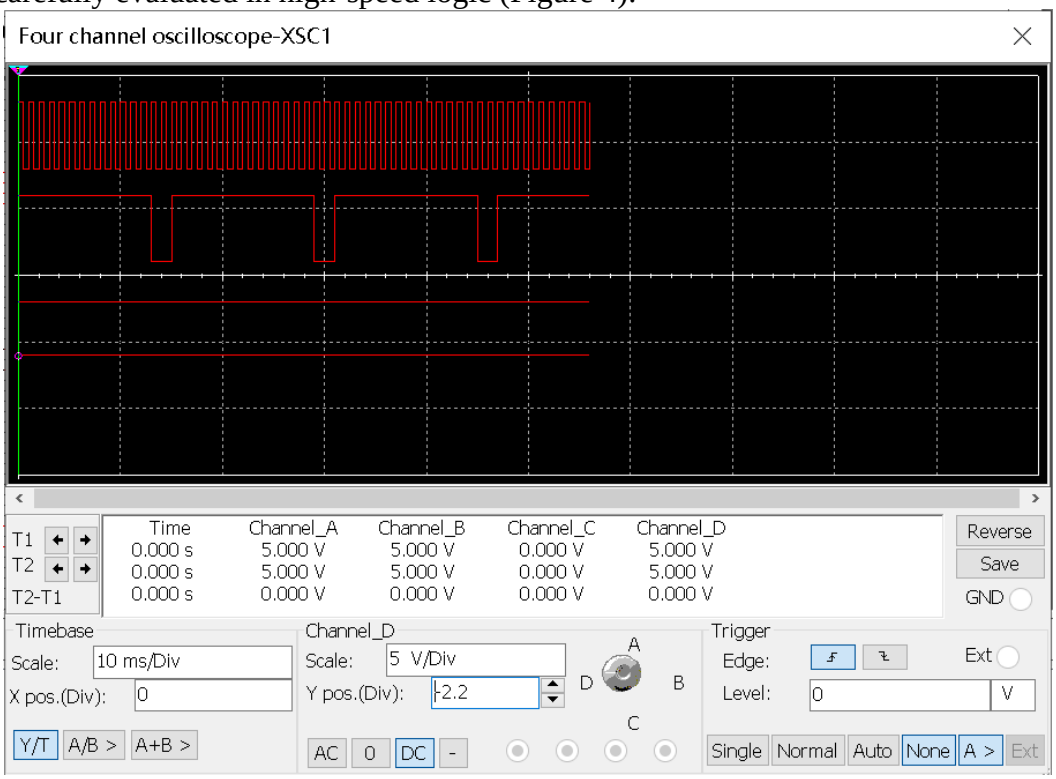

Figure 4 Capacitor Method Oscilloscope Waveform (Glitch smoothed, reset edge delayed) - RC Waveform: The glitch is significantly smoothed/eliminated; the effective reset edge shows a visible phase lag. XSC1: 5 V/div, 10 ms/div.

The second method is D-type flip-flop output buffering. This strategy does not act directly on the reset logic but instead introduces a layer of edge-triggered buffering between the counter's outputs and the reset decision circuit to enhance the timing stability of the output signals. Specifically, the outputs QA, QB, and QD are each connected to a D-type flip-flop (e.g., a 74LS74), which share the main system clock as their triggering edge (Figure 5). The resulting output signals will only update on the rising edge of the clock and will remain stable for the entire clock period. This method effectively masks the unstable propagation from the combinational logic, preventing logic misinterpretations caused by signals transitioning mid-cycle. A comparison in Multisim shows that with this buffering, the CLR pin remains stable even with delays in the input paths, and only outputs a low level to perform the reset when the "1101" state is truly reached. The trade-off is a small increase in gate-level delay and a one-clock-cycle data latency, but it is considered a highly reliable solution with an acceptable trade-off in most synchronous systems (Figure 6, Figure 7).

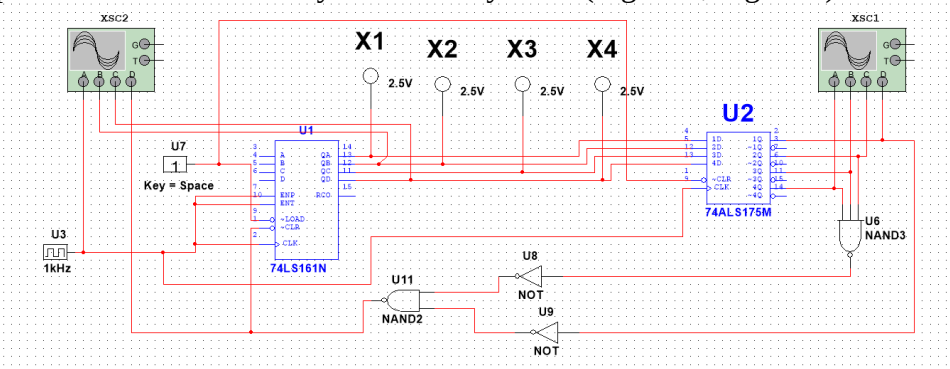

Figure 5 74ALS175M Edge-Registering Schematic - Edge-Registering Scheme: U2=SN74ALS175A samples {QA,QB,QD} uniformly on the clock edge before sending them to U6.

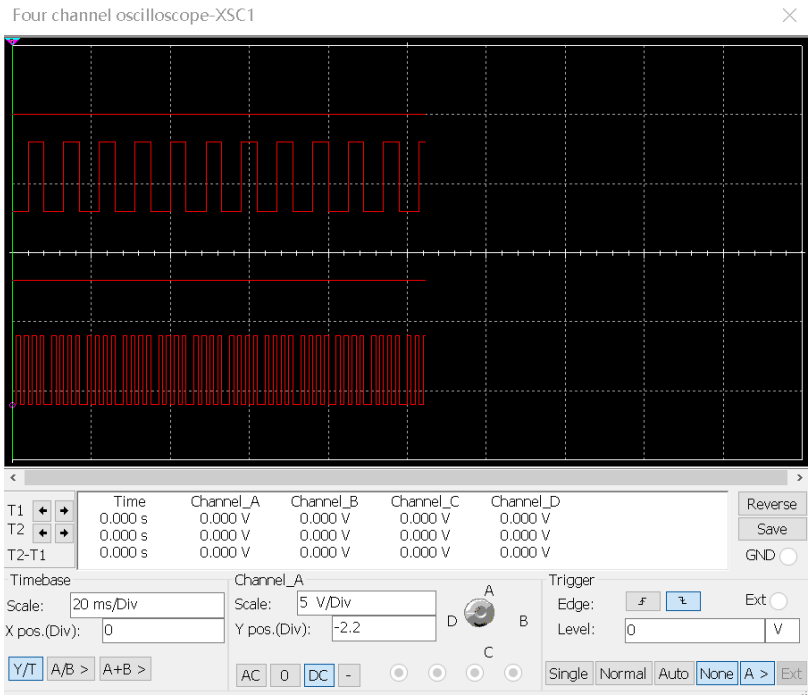

Figure 6 Waveform After 74ALS175M Registering (XSC1: Clean output after register) - Post-Registering (XSC1): The output updates only on the sampling edge, with no sub-cycle low pulses on RESET; functionality is correct, with the cost of a one-cycle delay (≈1 ms). 20 ms/div.

For ease of comparison, the observed results for each scheme are summarized in Table 1:

Table 1 Table of Observed Results for Each Scheme

Method	Glitch Width (μs)	Reset Delay	Reliability
Baseline	~200	0	Unstable
RC (100 nF)	Partially smoothed (~50)	~0.2 ms phase lag	Medium
RC (1 μF)	Fully smoothed	~0.8 ms lag	High
74ALS175M Registering	Eliminated	1-cycle delay (1 ms)	Very High (Stable)

As seen in the table, the effectiveness of the capacitor method depends mainly on the time constant, whereas the registering method remains robust regardless of component values.

These two hazard elimination techniques each have distinct characteristics: capacitor filtering is physically simple and level-sensitive, suitable for low-speed control nodes; D-type flip-flop buffering is ideal for stabilizing outputs in synchronous systems. In practical design, these two methods are not mutually exclusive and are often deployed in combination to achieve greater robustness and fault tolerance.

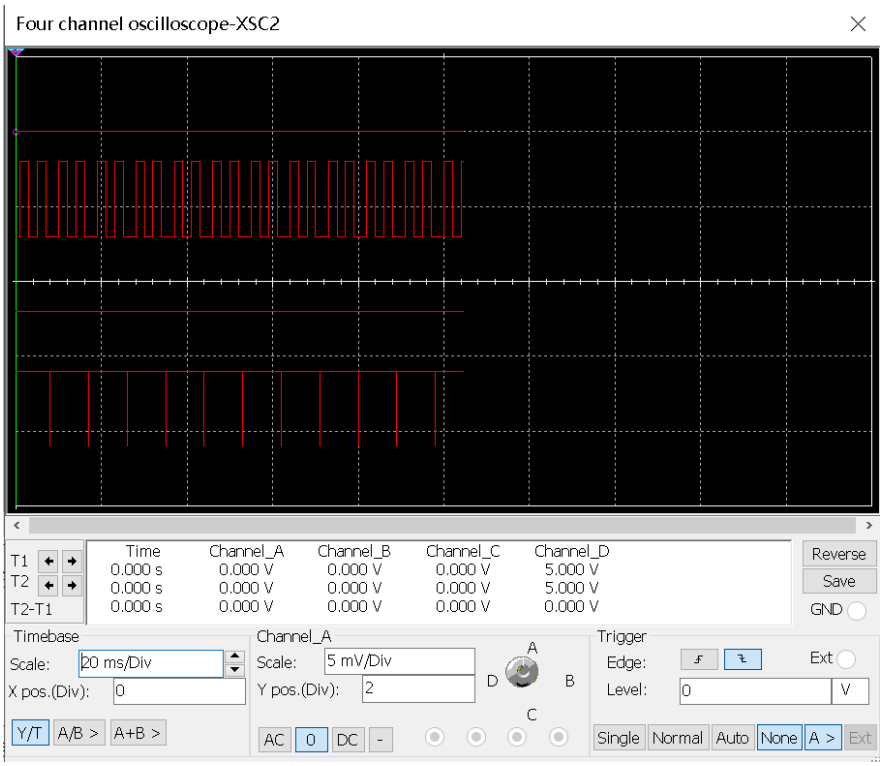

Figure 7 Waveform Before 74ALS175M Registering (XSC2: Glitch still visible before register) - Pre-Registering (XSC2): A short glitch is still visible on the input side, indicating that the register blocked it from passing by keeping it outside the sampling window. 5 mV/div (illustrative), 20 ms/div.[7]

5. Discussion

The Multisim simulation results highlight the existence of the race condition and the effectiveness of two common techniques in reducing or eliminating it. Without any mitigation measures, the modulo-13 counter circuit is susceptible to timing issues, particularly around the reset logic. The reset signal can trigger prematurely or exhibit glitches, as shown in the oscilloscope outputs. This is a

classic example of a race condition caused by unbalanced propagation delays in combinational logic.

Of the techniques tested, capacitor filtering offers a simple and low-cost method for glitch suppression. When a capacitor is added to the reset line, minor voltage spikes are absorbed before they can propagate to the counter's reset input. This method is particularly useful in low-speed or moderately noisy circuits where the goal is to eliminate transient, unintentional voltage changes. The drawback, however, is that the capacitor also slows down the signal's transition speed. In circuits where precise timing is required or where the reset must occur quickly, this delay could introduce new problems. Therefore, capacitor filtering is best used in conjunction with other methods.

The output cascading with flip-flops, specifically the scheme using edge-triggered D-type flip-flops, proved to be highly effective. These flip-flops allow the circuit to isolate the internal counting logic from the output transitions, ensuring that changes only occur on a defined clock edge. Consequently, even if the internal logic is briefly unstable due to signal overlap, the output remains consistent. This approach is suitable for any medium- or high-speed design where stability is critical and the output data needs to be sampled or transmitted further downstream.

Ultimately, no single method is suitable for all scenarios. The designer must choose based on the specific needs of the circuit. For a low-speed circuit with occasional glitches, a simple capacitor may suffice. For a system with strict timing and frequent state changes, flip-flops should be used. In many cases, the best results come from combining multiple techniques for layered protection.

6. Conclusion

Race conditions are a latent but serious threat to the stability of digital circuits, especially in sequential systems where timing and signal order are critical. Through simulations of a modulo-13 counter in Multisim, this paper tested two techniques for eliminating race conditions: capacitor filtering and edge-triggered flip-flops. Each method enhances reliability in different ways, suggesting that using them in combination can yield the most robust results. Although experimental aspects like capacitor filtering still require further development, the theoretical analysis and preliminary tests support a clear conclusion: with careful control of timing and signal stabilization strategies, race conditions can be effectively mitigated, enabling designers to build more robust and predictable digital systems.

The experimental platform of this study can also be extended to FPGA or ASIC verification to further investigate the impact of non-ideal factors such as routing parasitics and power supply noise. Future work includes conducting physical experiments under higher frequency conditions and incorporating formal verification methods from EDA tools to establish a more systematic hazard detection and elimination workflow. This not only provides a clear case study for educational experiments but also offers a reference pathway for high-reliability design in engineering practice.

References

[1] J. F. Wakerly, Digital Design: Principles and Practices, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2006.

[2] M. M. Mano and M. D. Ciletti, Digital Design, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2013.

[3] C. H. Roth and L. L. Kinney, Fundamentals of Logic Design, 7th ed. Boston, MA, USA: Cengage Learning, 2014.

[4] Analog Devices, Inc., "How To Perform Accurate Measurement and Reduction of Glitch Impulse Energy in DACs," Technical Article, Sep. 2021. [Online]. Available: https://www.analog.com/en/resources/technical-articles/how-to-perform-accurate-measurement-and-reduction-of-glitch-impulse-energy-in-dacs.html

[5] Texas Instruments, Inc., "Metastable Response in 5-V Logic Circuits," Application Note SDYA006, 1997. [Online]. Available: https://www.ti.com/lit/pdf/sdya006

[6] Texas Instruments, Inc., "SN74LS161A Synchronous 4-Bit Binary Counters," Datasheet, Rev. 1998. [Online]. Available: https://www.ti.com/lit/gpn/SN74LS161A

[7] Texas Instruments, Inc., "SN74ALS175 Quadruple D-Type Positive-Edge-Triggered Flip-Flops With Clear," Datasheet, Rev. 1995. [Online]. Available: https://www.ti.com/product/SN74ALS175