Comparison of Various Reset Designs in Simulation Behavior

This report is a reminder study note for me. It explains the trigger mechanisms of different reset designs. The report compares the functionality of designs listed below in Xilinx Vivado's Behavioral Simulation. The comparison is made exclusively for simulation purposes, any considerations beyond simulation are ignored (e.g. Reset Tree). Verilog HDL is used.

- Synchronous Active High Reset
- Synchronous Active Low Reset
- Asynchronous Active High Reset
- Asynchronous Active Low Reset

Top module contains four different reset modules that use different reset approaches listed above. Testbench has four commented out initial-type procedural blocks representing scenarios for each reset design, respectively. 100MHz system clock is used.

Normally, 24-bits hexadecimal value "c0ffee" should be seen at output "result" of respective module. When the impact of the reset takes place, the result "0" is expected at output.

The last part of the report focuses on a pitfall about design of asynchronous resets. An additional non-synthesizable fifth module is created for this purpose. The report compares two similar Asynchronous Active Low Reset designs with different sensitivity lists. The same testbench scenario for Asynchronous Active Low Reset is used for this comparison.

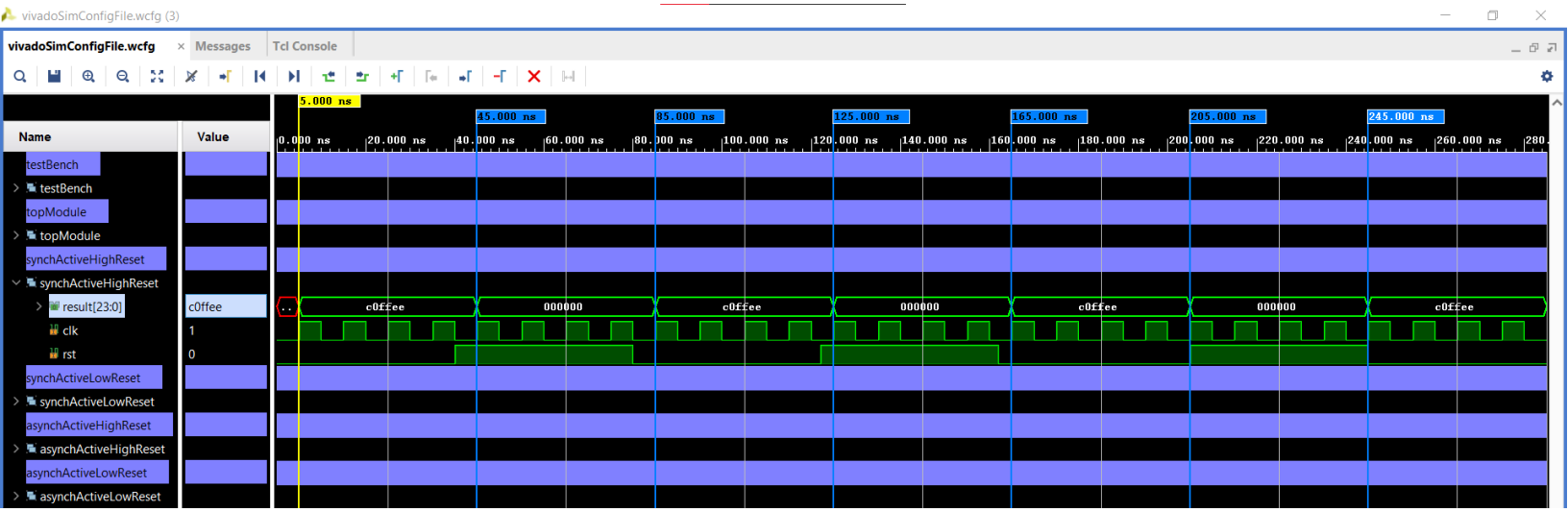
- Muhammed Emin Özdemir,
February 2024

Synchronous Active High Reset

Basically, if the reset signal has a logic high value when the clock signal rises, the impact of the reset takes place (see 45th nanosecond). However, one should notice that if the reset gets a logic high value before the rising edge of the clock, the impact waits until the rising edge of the clock and then takes place (see 125th nanosecond).

Additionally, if the reset gets a logic high value at the same moment the clock rises, the effect immediately takes place (see 205th nanosecond). Likewise, if the reset signal has a logic low value, the reset effect waits for the rising edge of the clock signal to disappear (see 85th, 165th, and 245th nanoseconds).

Therefore, this reset design is synchronized to the positive edge of the clock and gets triggered at the logic high values of the reset signal.

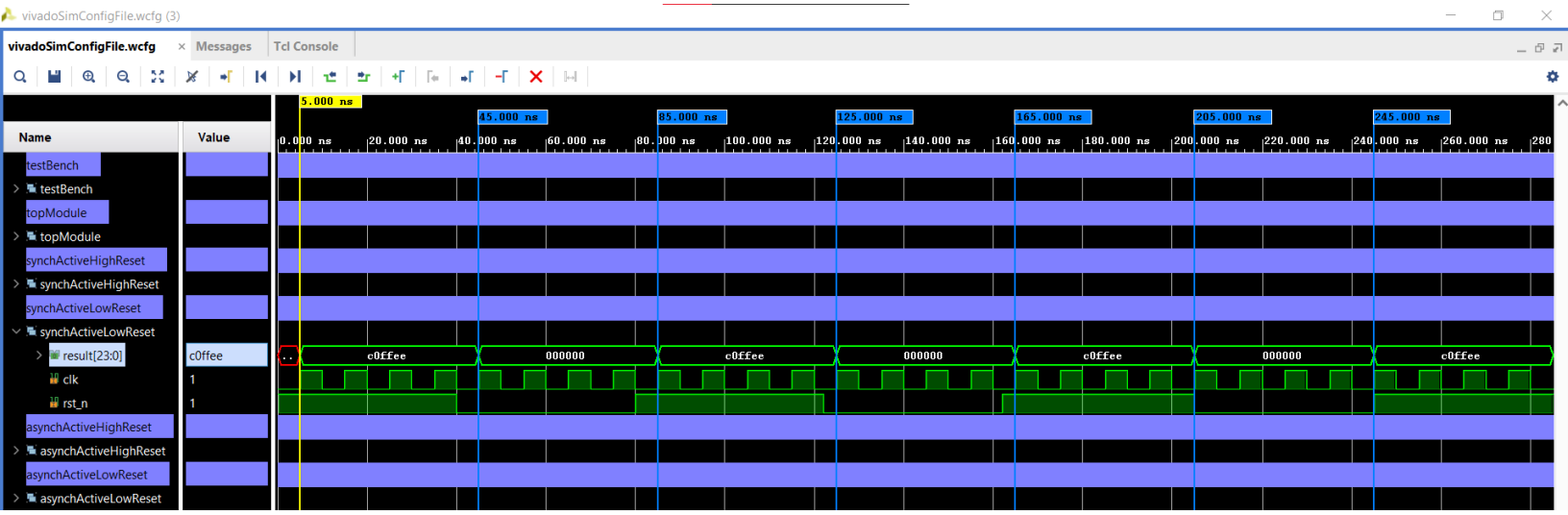


Synchronous Active Low Reset

In essence, if the reset signal holds a logic low value during the rise of clock signal, the reset impact occurs instantly (see 45th nanosecond). However, if the reset signal transitions to a logic low value before the rising edge of the clock, the impact is delayed until the rising edge of the clock (see 125th nanosecond).

Furthermore, if the reset signal gets a logic low value at the exact moment the clock rises, the effect immediately takes place (see 205th nanosecond). Similarly, the reset effect waits for the rising edge of the clock signal to disappear when the reset signal holds a logic high value, (see 85th, 165th, and 245th nanoseconds).

Hence, this reset design is synchronized to the positive edge of the clock and is triggered by logic low values of the reset signal.



Asynchronous Active High Reset

In this type of reset design, the reset impact occurs in two different circumstances.

Firstly, even if there isn't a rising clock signal, the existence of a rise in the reset signal causes the impact to instantly take effect without waiting for a rising edge in the clock signal (see 40th and 122nd nanoseconds).

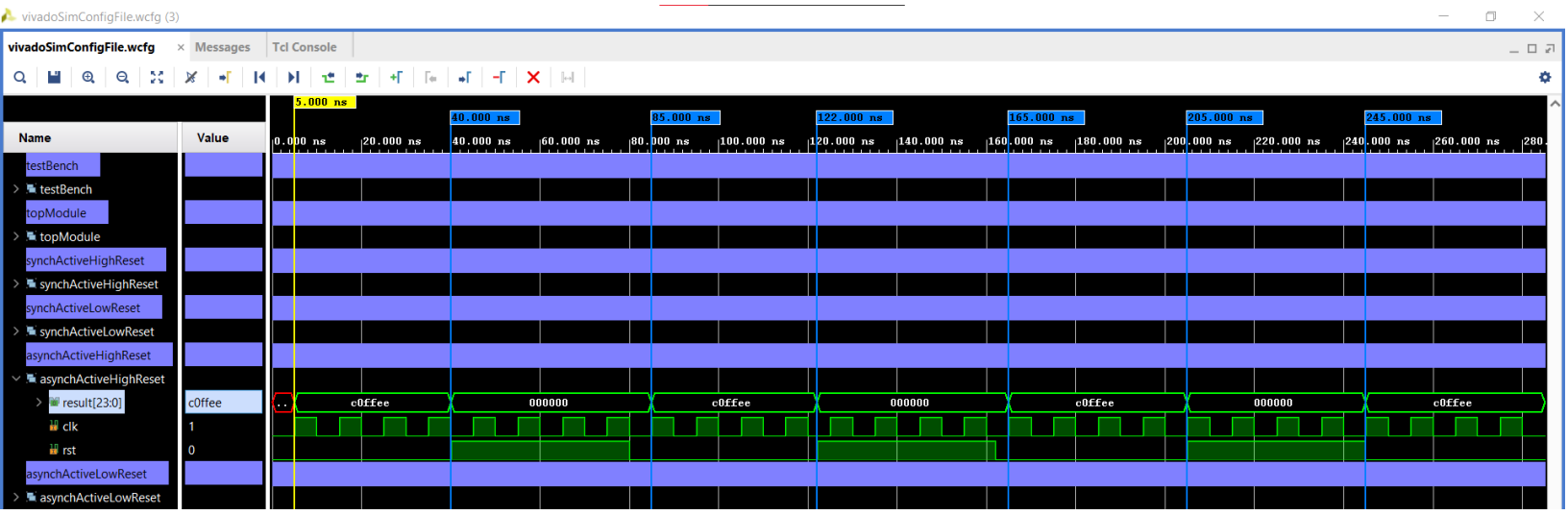
Secondly, the reset effect takes place at the rising edge of the clock signal with a logic high value in the reset signal at that very moment. In this case, what triggers the reset mechanism is a rise in the clock. The logic high value in the reset signal is just a condition that is met when the clock triggers the mechanism with a rising edge.

In detail, the impact of the reset signal, which is the assignment of “0s” to the “result” output of the module, occurs and re-occurs at every rising edge of the clock as long as the reset signal has a logic high value. This is what happens at the 45th, 55th, 65th, and 75th nanoseconds in the simulation.

When both the clock and the reset signal transition to a logic high value at the same moment, the reset mechanism is triggered, and the condition for reset is met. Therefore, the reset effect immediately takes place (see 205th nanosecond). It is not clear which signal triggers the reset mechanism in this case.

Additionally, I have read that this circumstance may lead to metastability problems in real logic circuitry. However, these considerations are out of this report’s scope. This example only shows the functionality of asynchronous active high reset design when both the clock and the reset signal have a rising edge at the same time on Xilinx Vivado’s Behavioral Simulation.

Even though the reset design is asynchronous to the clock, what removes the reset effect on the module output occurs at the positive edge of the clock. Even if the reset signal has a logic low value, the reset impact does not disappear instantly but remains until the clock has a rising edge (see 85th, 165th, and 245th nanoseconds).



Asynchronous Active Low Reset

Just like the previous design, the reset impact takes place in two distinct cases with this reset approach.

First, even if clock signal does not rise at that moment, a fall in the reset signal immediately causes the reset impact to take effect without waiting for a rising clock edge (see 40th and 122nd nanoseconds).

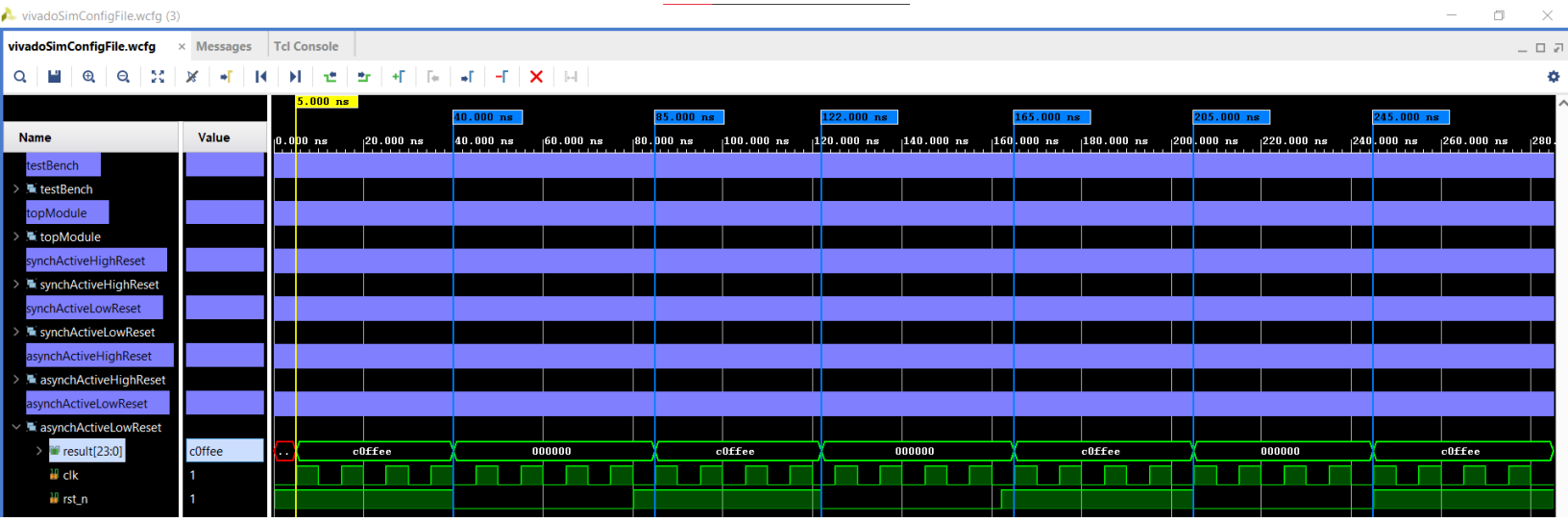
Second, the reset effect takes place at the rising edge of clock signal under the condition that the reset signal has a logic low value at that same moment. Here, what triggers the reset mechanism is a rise in the clock. The logic low value of the reset signal is simply a condition that is met when the clock triggers the mechanism with a transition to logic high value.

In-depth, the reset effect which is the assignment of “0s” to the “result” output of the module occurs and re-occurs at every rising edge of the clock as long as the reset signal is at logic low state. This situation can be seen at the 45th, 55th, 65th, and 75th nanoseconds of simulation.

When the clock transitions to a logic high value at the same moment the reset signal gets a logic low value, the reset mechanism is triggered and the condition for reset is met. Hence, the reset impact immediately takes effect (see 205th nanosecond). It’s not clear whether the clock or the reset signal triggers the reset mechanism in this case.

Despite these considerations beyond the scope of this report, this situation may lead to metastability issues in real digital IC applications. This instance solely examines the functionality of the asynchronous active low reset design when clock signal has rising edge at the same time the reset signal gets a logic low value on Xilinx Vivado’s Behavioral Simulation.

Although this reset design is asynchronous to clock, what takes the reset impact away from the module output happens at the rising edge of the clock. Even if the reset signal transitions to a logic high state at some moment, the reset impact does not disappear simultaneously with that same moment but remains unchanged till the clock has a rising edge (see 85th, 165th and, 245th nanoseconds).



The Non-Synthesizable Reset Design

This reset design is identical to the Asynchronous Active Low Reset design, except for the module's sensitivity list. The always-type procedural block in the module has "posedge clk or rst_n" instead of "posedge clk or negedge rst_n". This type of design works in simulation but causes errors in synthesis.

```
module asynchActiveLowReset
    (
    output reg [23:0] result,

    input clk,
          rst_n
    );

    always@(posedge clk or negedge rst_n)      // asynchronous
        begin
        if(!rst_n)                             // active low reset
            begin
            result <= 'b0;
            end
        else
            begin
            result <= 24'hc0ffee;
            end
        end
endmodule
```

```
module nonSynthesizableReset
    (
    output reg [23:0] result,

    input clk,
          rst_n
    );

    always@(posedge clk or rst_n)   // asynchronous
        begin                       // "rst_n" is used instead of "negedge rst_n"
        if(!rst_n)                  // active low reset
            begin
            result <= 'b0;
            end
        else
            begin
            result <= 24'hc0ffee;
            end
        end
endmodule
```

Here, the reset mechanism gets triggered at both rising and falling edges of the reset signal. Therefore, the timing of when the reset impact takes effect or disappears changes. In contrast to the behavior in the previous design, the reset impact immediately disappears when the reset signal transitions to a logic high value as seen at 80th and 162nd nanoseconds for comparison. Additionally, since the contents of the sensitivity list not only trigger the reset mechanism but also activate the normal function of the module, the "c0ffee" value is assigned to the module output instantly when the reset signal rises at 0ns.