1. Combinational logic detected before a synchronizer

20 January 2025 18:10

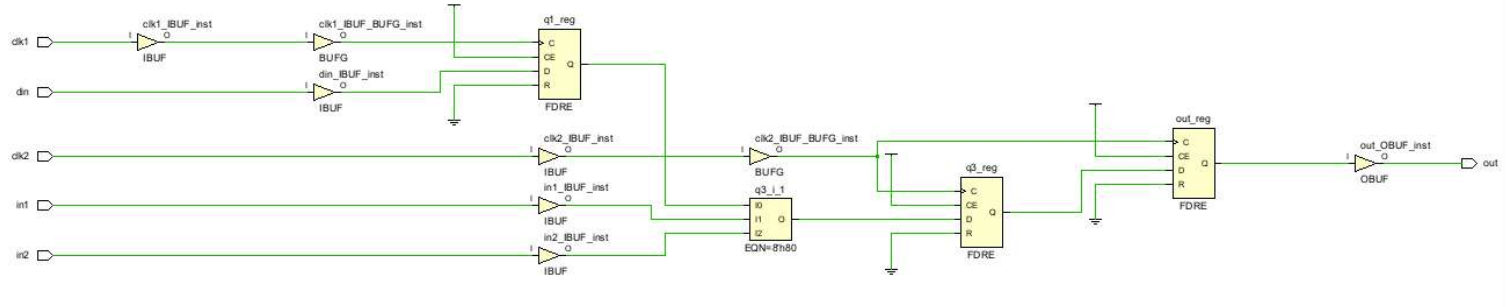
Verilog Code

```
`timescale 1ns / 1ps
module Comb_logic_sync(
   input clk1,
   input clk2,
   input in1,
   input in2,
   input din,
   output reg out
);
   reg q1;
   always@(posedge clk1) begin
      q1 <= din;
   end
   wire q2;
   assign q2 = q1 && in1 && in2;
   reg q3;
   always@(posedge clk2) begin
      q3 <= q2;
      out<= q3;
   end
endmodule
```

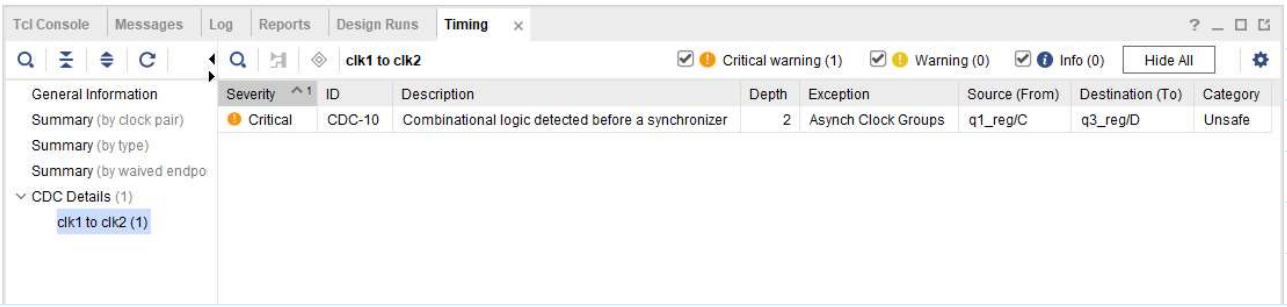
XDC File

```
set_property ASYNC_REG true [get_cells q3_reg]
set_property ASYNC_REG true [get_cells out_reg]

create_clock -period 10.000 -name clk1 -waveform {0.000 5.000} [get_ports clk1]
create_clock -period 20.000 -name clk2 -waveform {0.000 10.000} [get_ports clk2]

set_input_delay -clock [get_clocks clk1] -min -add_delay 2.000 [get_ports din]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports din]

set_input_delay -clock [get_clocks clk1] -min -add_delay 2.000 [get_ports in1]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports in1]

set_input_delay -clock [get_clocks clk1] -min -add_delay 2.000 [get_ports in2]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports in2]

set_output_delay -clock [get_clocks clk1] -min -add_delay 0.600 [get_ports out]
set_output_delay -clock [get_clocks clk1] -max -add_delay 2.300 [get_ports out]

set_clock_groups -asynchronous -group [get_clocks clk1] -group [get_clocks clk2]
```

Schematic



CDC Result



The problem in the above circuit is here is an Combinational block before the synchronizer. This is not

Allowed. To overcome it, we need to use a flipflop after the combinational logic.

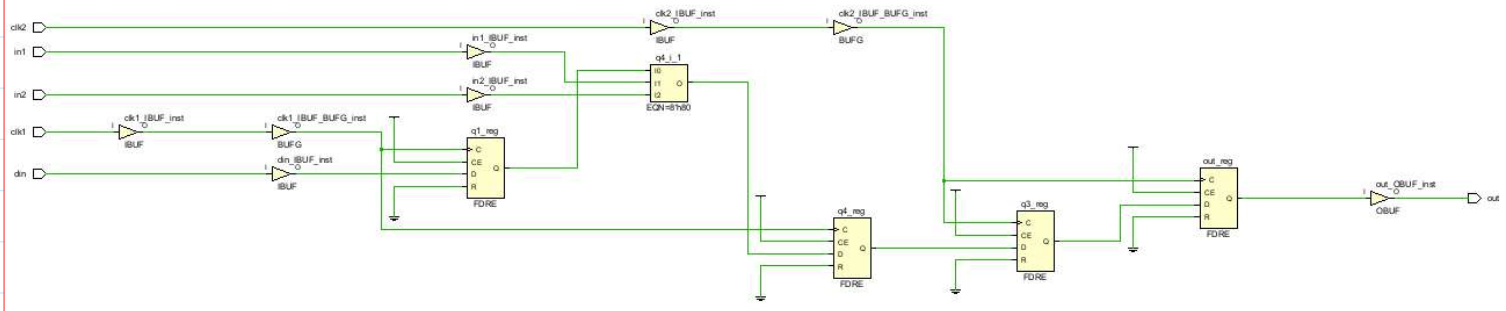
Verilog Code

```
`timescale 1ns / 1ps
module Comb_logic_sync(
   input clk1,
   input clk2,
   input in1,
   input in2,
   input din,
   output reg out
);
   reg q1,q4;
   wire q2;
   always@(posedge clk1) begin
      q1 <= din;
      q4 <= q1 && in1 && in2;
   end
   reg q3;
   always@(posedge clk2) begin
      q3 <= q4;
      out<= q3;
   end
endmodule
```

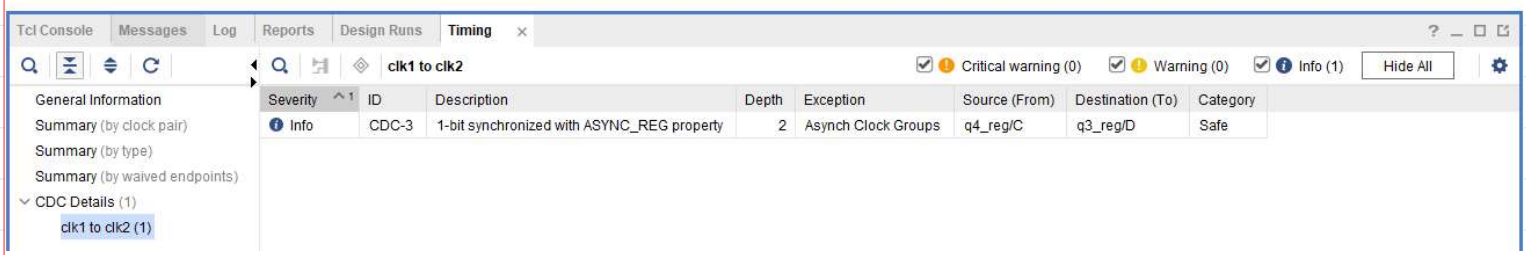
XDC

```
set_property ASYNC_REG true [get_cells q3_reg]
set_property ASYNC_REG true [get_cells out_reg]

create_clock -period 10.000 -name clk1 -waveform {0.000 5.000} [get_ports clk1]
create_clock -period 20.000 -name clk2 -waveform {0.000 10.000} [get_ports clk2]

set_input_delay -clock [get_clocks clk1] -min -add_delay 2.000 [get_ports din]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports din]

set_input_delay -clock [get_clocks clk1] -min -add_delay 2.000 [get_ports in1]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports in1]

set_input_delay -clock [get_clocks clk1] -min -add_delay 2.000 [get_ports in2]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports in2]

set_output_delay -clock [get_clocks clk2] -min -add_delay 0.600 [get_ports out]
set_output_delay -clock [get_clocks clk2] -max -add_delay 2.300 [get_ports out]

set_clock_groups -asynchronous -group [get_clocks clk1] -group [get_clocks clk2]
```

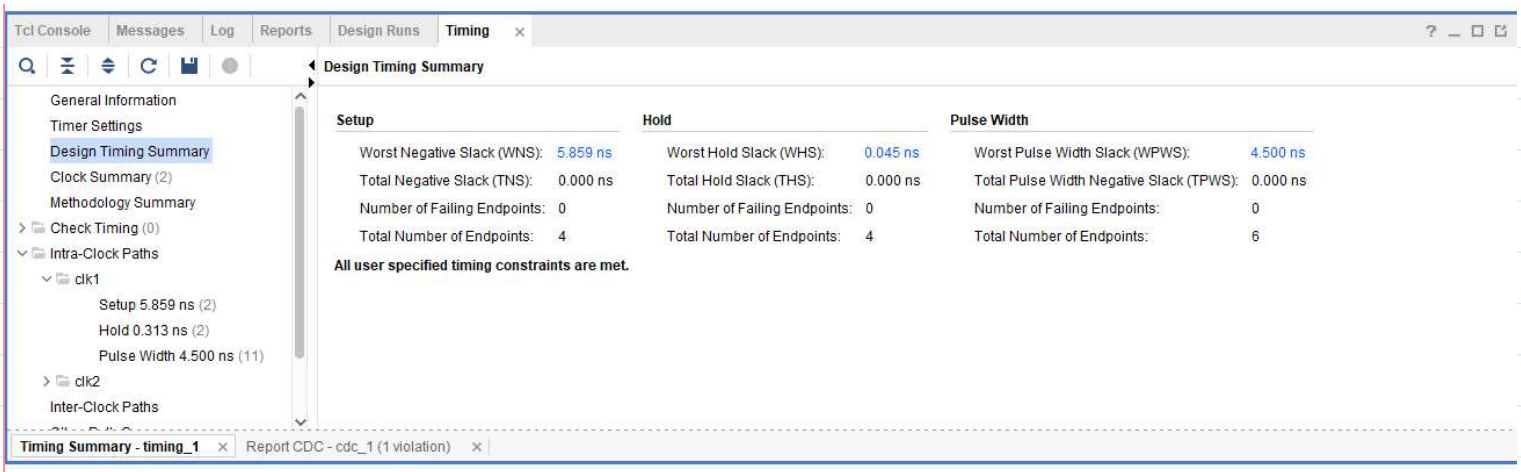
Schematic



CDC



STA



Problem Solved.