6. Synchronizer's

17 January 2025 17:50

A synchronizer is a device that samples an asynchronous signal and outputs a version of the signal that has transitions synchronized to a local or sample clock

Case 1: Normal Verilog Code where data is between two clocks

Verilog Code

```
module cdc1(
   input wire clk1,
   input wire clk2,
   input wire d,
   output reg out
);
   reg q;
   always@(posedge clk1) begin
         q <= d;
      end
   always@(posedge clk2) begin
         out  <= q;
   end
endmodule
```

XDC File

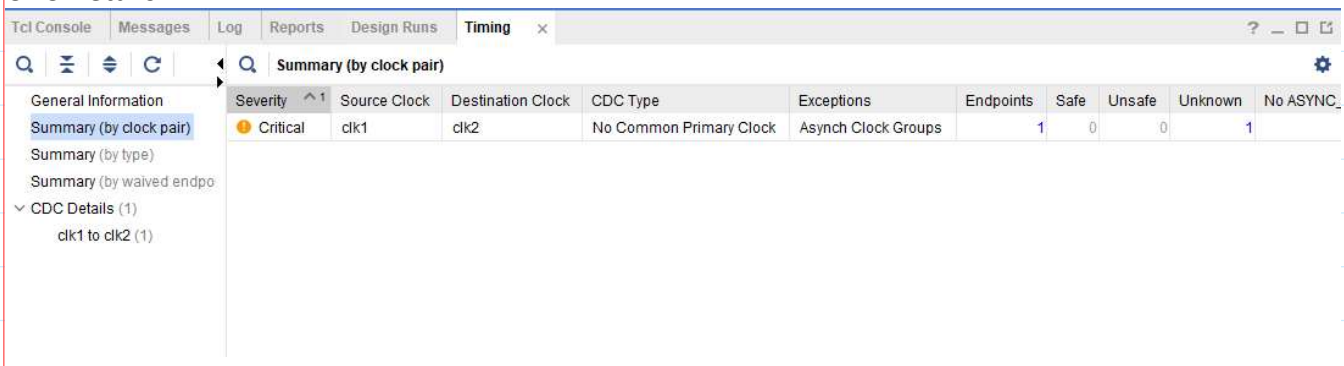
```
create_clock -period 10.000 -name clk1 -waveform {0.000 5.000} [get_ports clk1]
create_clock -period 20.000 -name clk2 -waveform {0.000 10.000} [get_ports clk2]

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

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]
```

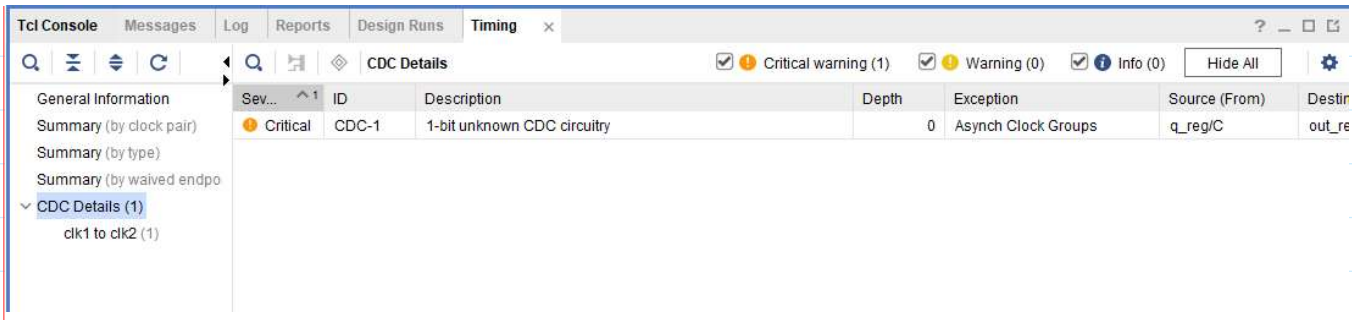
STA Details

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.895 ns	Worst Hold Slack (WHS):	0.879 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	2	Total Number of Endpoints:	2	Total Number of Endpoints:	4

All user specified timing constraints are met.

CDC Details





The major error in the above code is it is unable to understand that is 1-bit Unknown CDC Circuitry

Let us overcome the above issue by adding an synchronizer.

Case 2: Using an Synchronizer

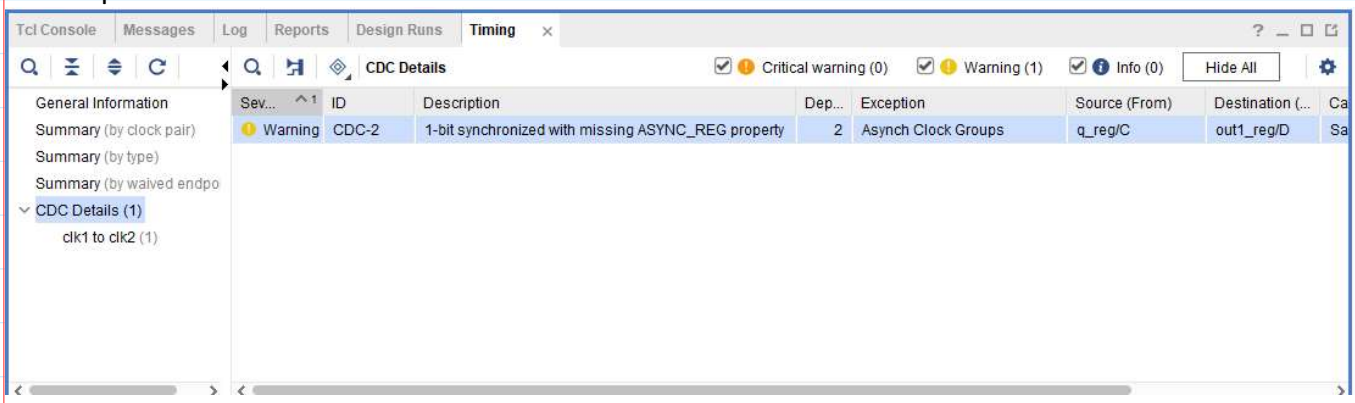
Verilog Code

```
`timescale 1ns / 1ps
module cdc1(
   input wire clk1,
   input wire clk2,
   input wire d,
   output reg out
);
   reg q;
   always@(posedge clk1) begin
        q <= d;
     end
   reg out1;
   always@(posedge clk2) begin
        out1  <= q;
        out   <= out1;
   end
endmodule
```

XDC File

```
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 d]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports d]

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]
```

CDC Reports



STA Reports

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.895 ns	Worst Hold Slack (WHS):	0.152 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	3	Total Number of Endpoints:	3	Total Number of Endpoints:	5

All user specified timing constraints are met.

The problem in the above code is it is missing the ASYNC_REG Property. To overcome that issue, let us modify the above code.

Case 3: To overcome the problem of missing ASYNC_REG.
Why to use Missing ASYNC_REG :

Synchronizer registers must have their ASYNC_REG property set to TRUE in order to preserve the cells through any logic optimization during synthesis and implementation, and to optimize their placement for the best mean time between failure (MTBF) statistics. (UG912)

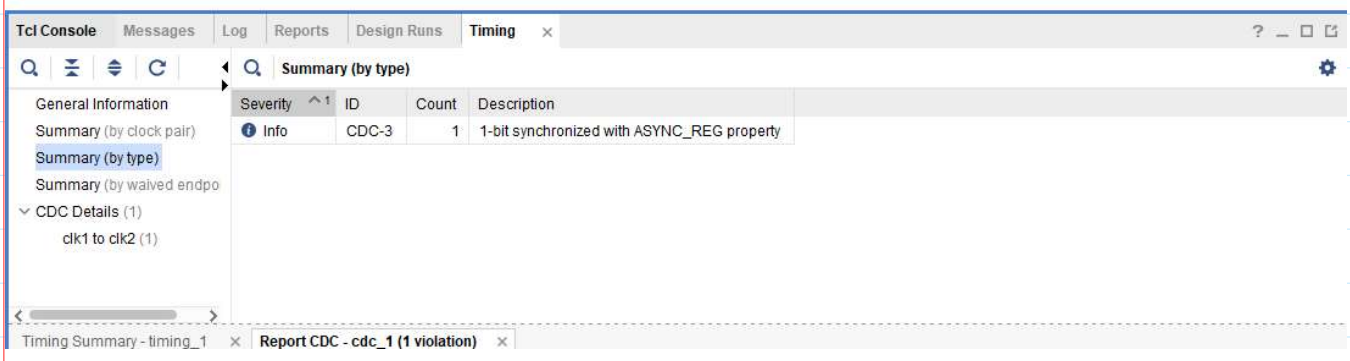
Verilog Code

```
`timescale 1ns / 1ps
module cdc1(
   input wire clk1,
   input wire clk2,
   input wire d,
   output reg out
);
   reg q;
   always@(posedge clk1) begin
         q <= d;
      end
   reg out1;
   always@(posedge clk2) begin
         out1  <= q;
         out   <= out1;
   end
endmodule
```

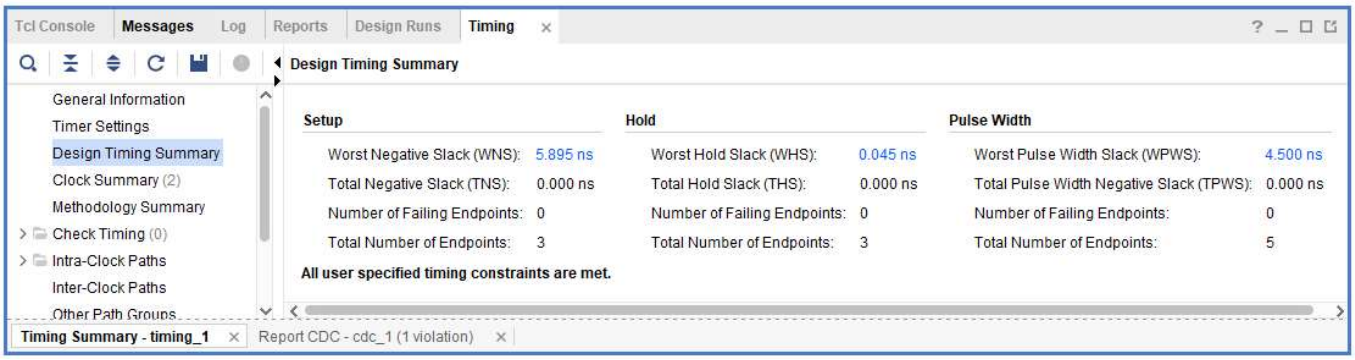
XDC File

```
set_property ASYNC_REG true [get_cells out1_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 d]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports d]

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]
```

CDC Reports



STA Reports



Case 4 Muti Bit Synchronizers

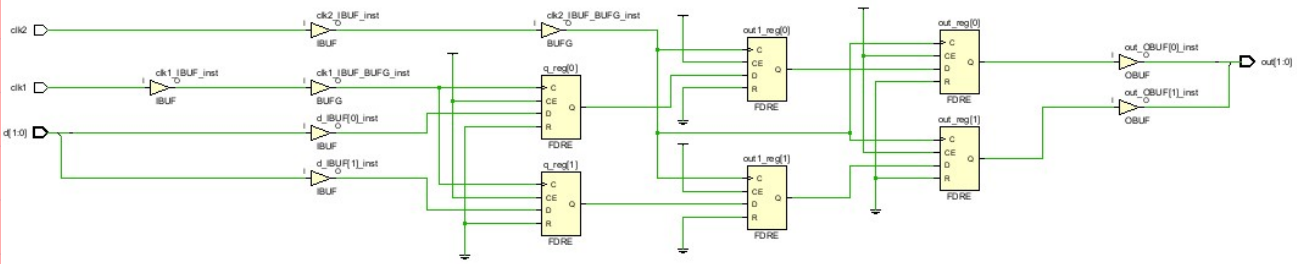
Verilog Code

```
`timescale 1ns / 1ps
module cdc1(
   input wire clk1,
   input wire clk2,
   input wire [1:0]d,
   output reg [1:0]out
);
   reg [1:0]q;
   always@(posedge clk1) begin
         q <= d;
      end
   reg [1:0]out1;
   always@(posedge clk2) begin
         out1  <= q;
         out   <= out1;
   end
endmodule
```

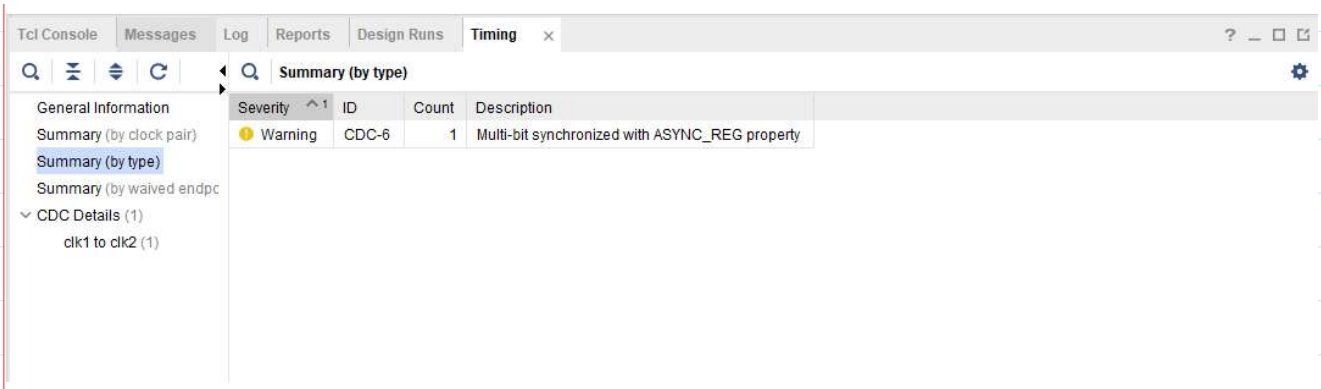
XDC File

```
set_property ASYNC_REG true [get_cells out1_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 d[*]]
set_input_delay -clock [get_clocks clk1] -max -add_delay 4.000 [get_ports d[*]]

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]
```

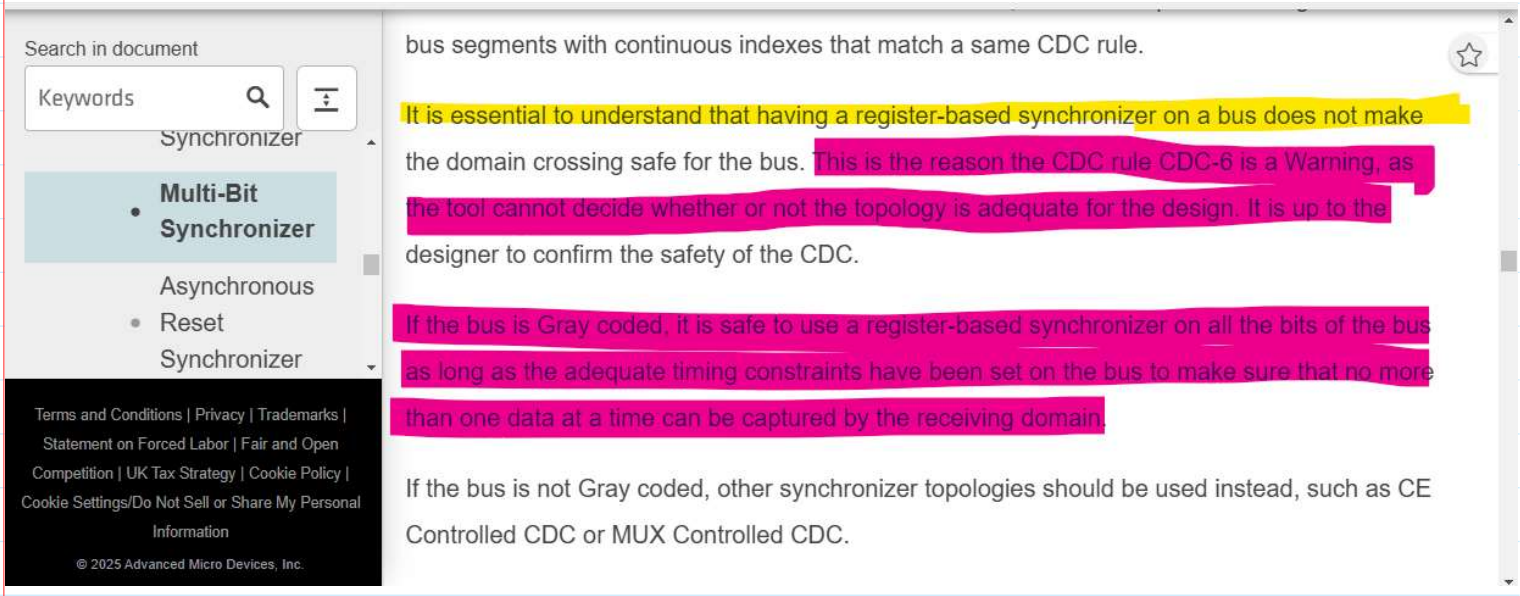
Schematic



CDC Report



The reason for the above error is



Drawback : We cannot synchronize a pulse, from quick clock domain to moderate clock domain.
(Input clock_a = 10ns, clock_b = 20ns)

Therefore a pulse cannot be detected
To overcome this problem we go for Toggle Synchronizer.