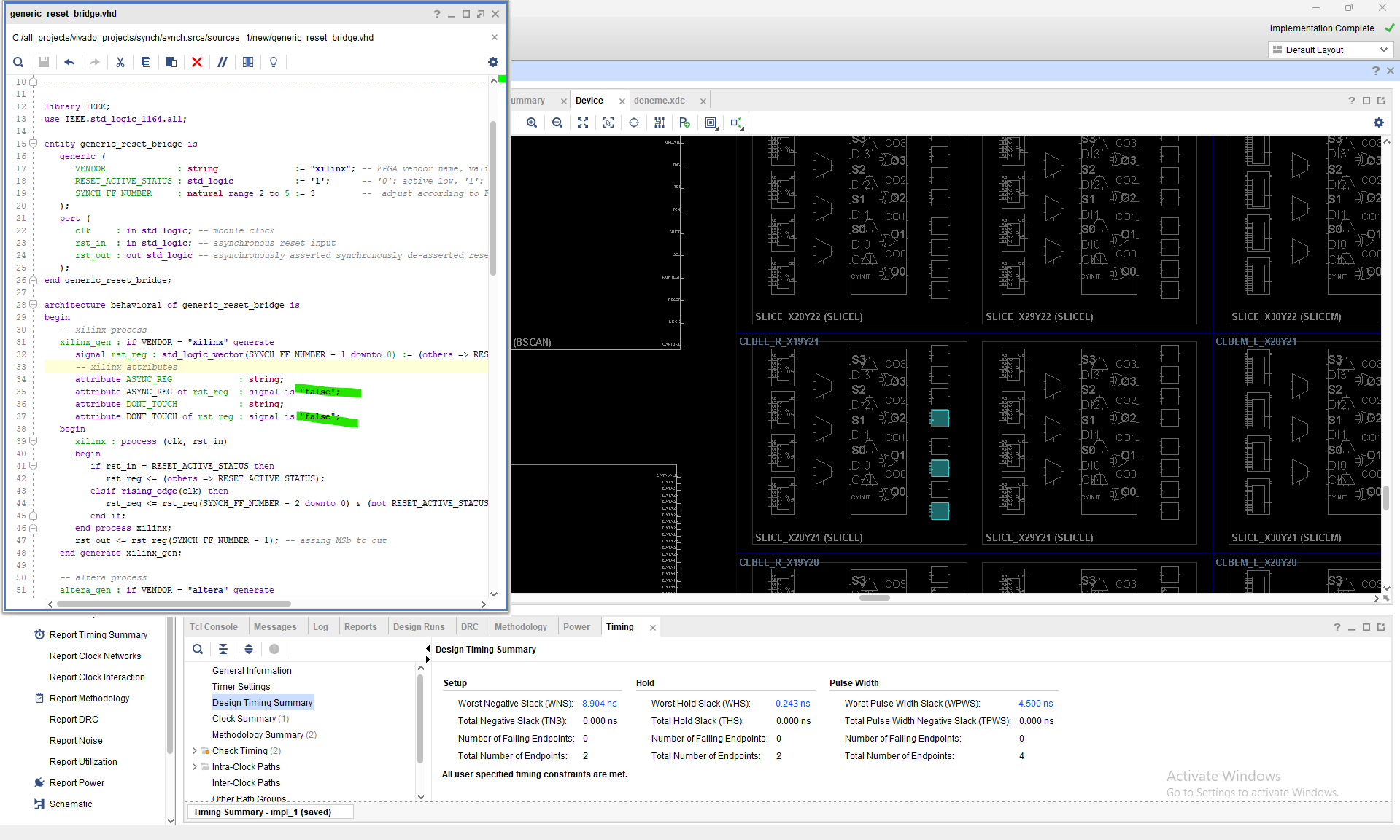
**Attibutes and False Paths of Synchronizers**

**Effects of attributes (Xilinx):**

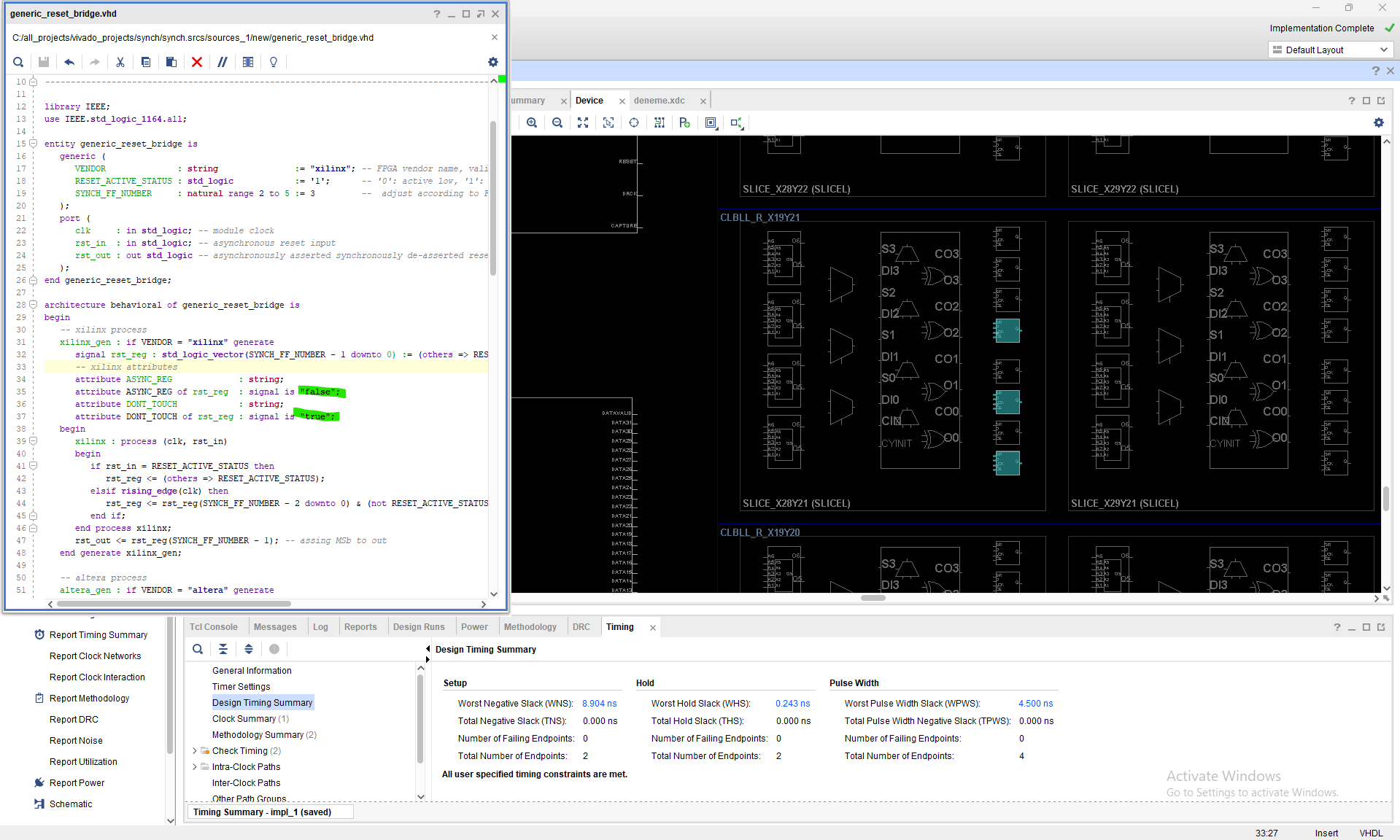
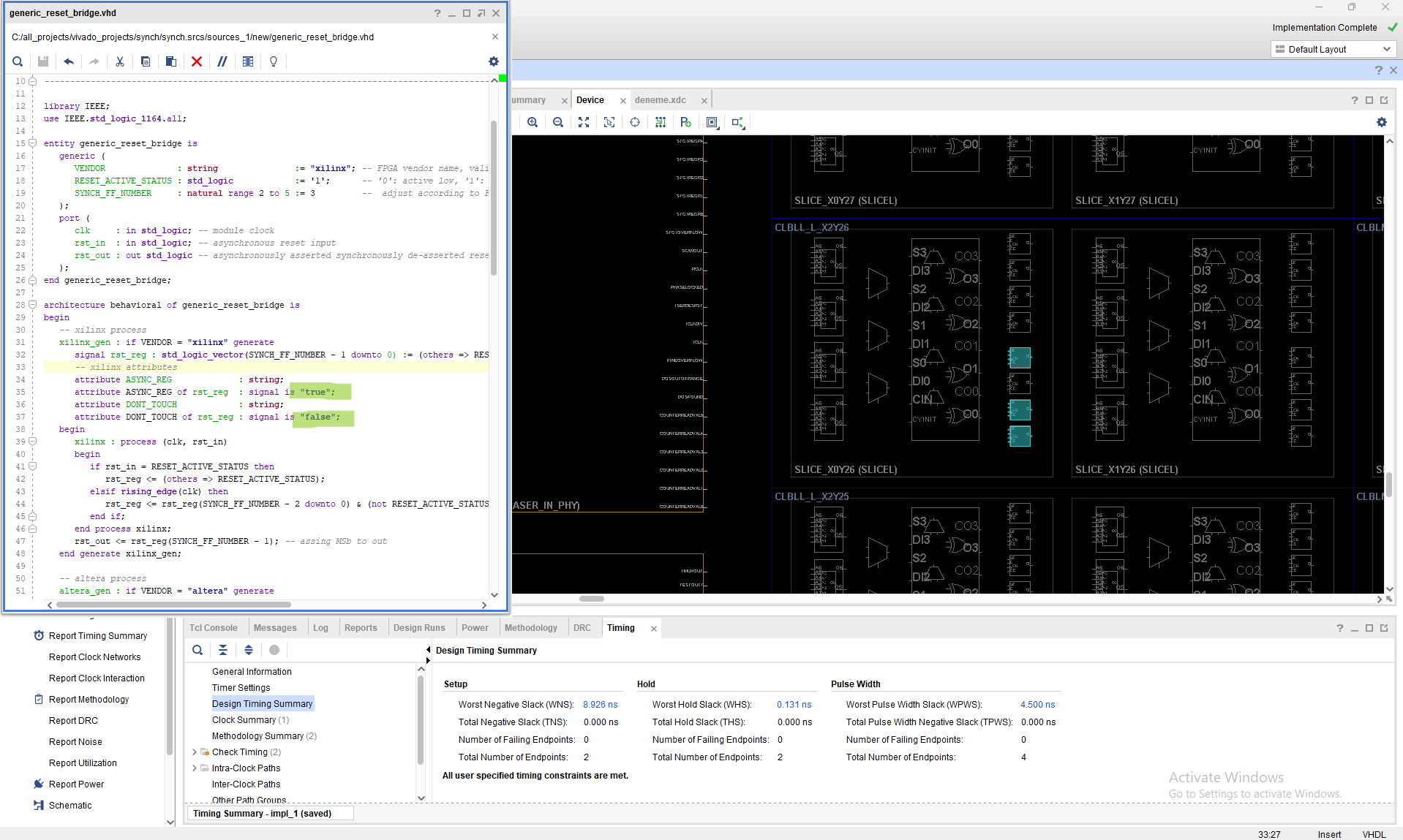
I added “async\_reg” and “dont\_touch” attributes for Xilinx FPGAs. In brief, “dont\_touch” does not optimize the defined signal and “async\_reg” tries to place defined registers (Flip Flops - FFs) close to each other to increase MTBF (Mean Time Between Failures). The effects of these attributes are shown in the below figures.

**Note**: These attributes are tested on the “generic\_reset\_bridge.vhd” HDL code.

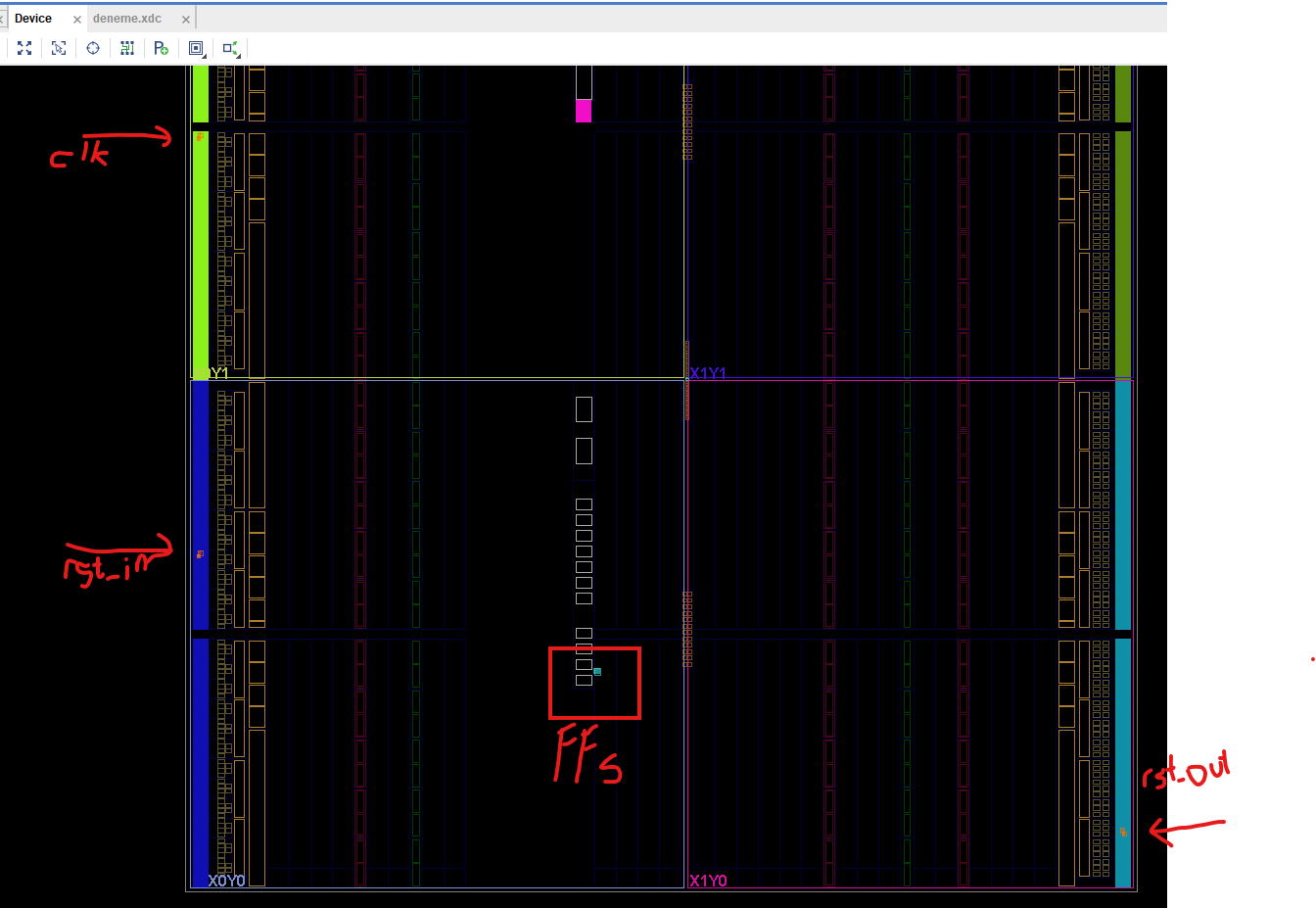
Firstly, the design was implemented when the both attributes are false and the timing and placement in the device are like below figure.

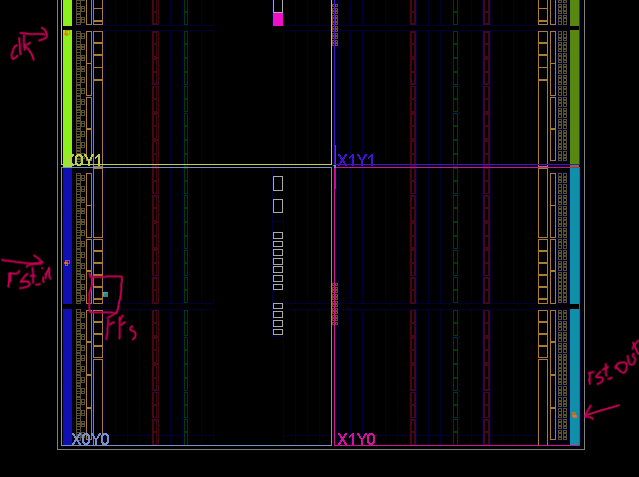


“dont\_touch” attribute was activated and the result was not changed because the design was implemented before with no optimization and it is still same. Also the design is quite simple.

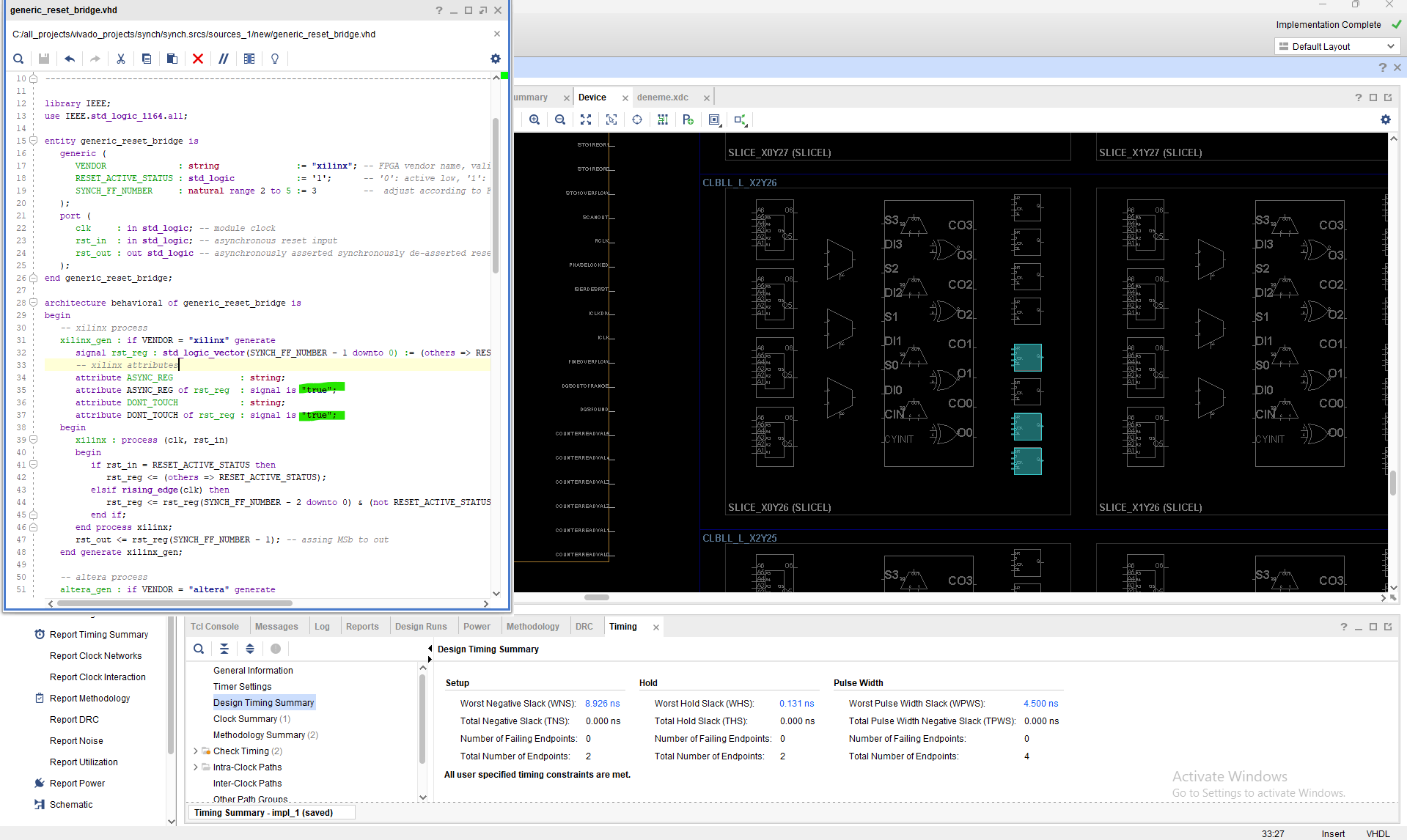
But when I activated the “async\_reg” attribute, the defined registers are placed contiguously. Also the timing was also improved because the net delay decreased and slice that has FFs in it was moved next to input reset package pin of FPGA (to see that effect, “rst\_in” and “rst\_out” ports deliberately chosen to be remote).

While ”async\_reg” is passive.



When ”async\_reg” is activated.

The correct and final trial was tested on both attributes was activated and the result is same like before because “async\_reg” is still active and “dont\_touch” attribute did not effect the design because the before design was not optimized.



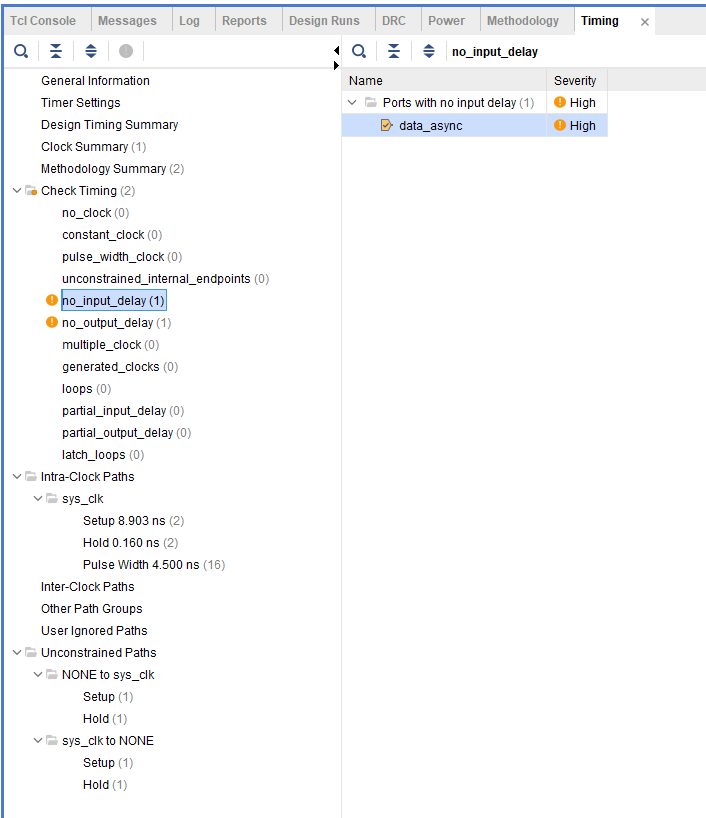
To sum up, “async\_reg” attribute works like said in the Xilinx document and to avoid the optmization, “dont\_touch” attribute is necessary especially in the big or repeated designs.

**Effects of false path constraint (Xilinx):**

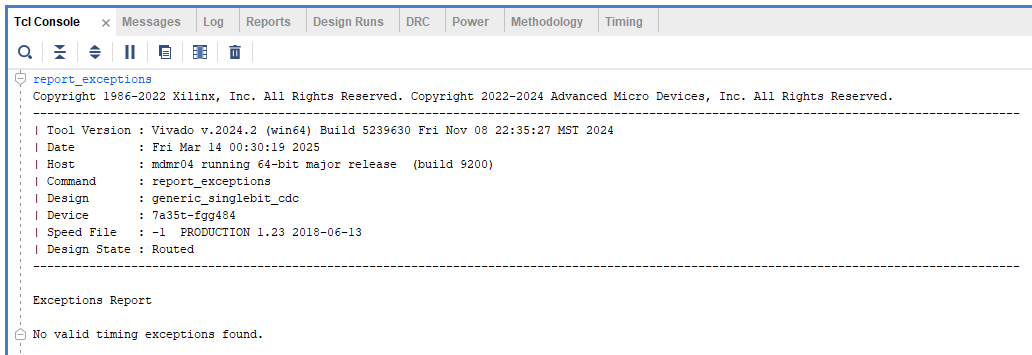
FPGA IDE tools like Vivado analyze the timing of the design to see that it meets defined constraints or not meets. Static Timing Analyze (STA) is done for all design unless a constraint is defined to not analyze the specified path for timing. These constraints are known as false path constraints. False path constraint must be given correctly and to only the path that does not need to be analyzed for timing like synchronizers input data that come from another clock domain.

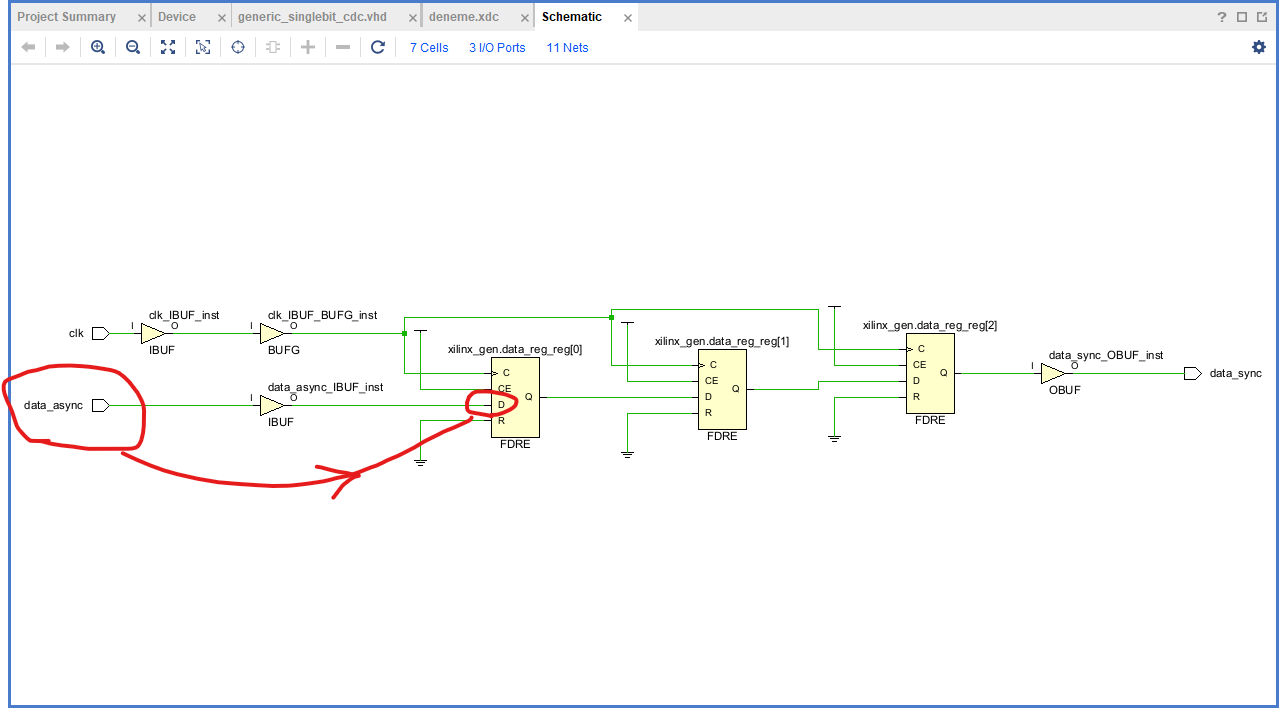
**Note**: That false path is tested on the “generic\_singlebit\_cdc.vhd” HDL code.

I defined asynchronous data input to a FPGA pin. Normally, Vivado gives error when there is no input timing constraint, see below figure.

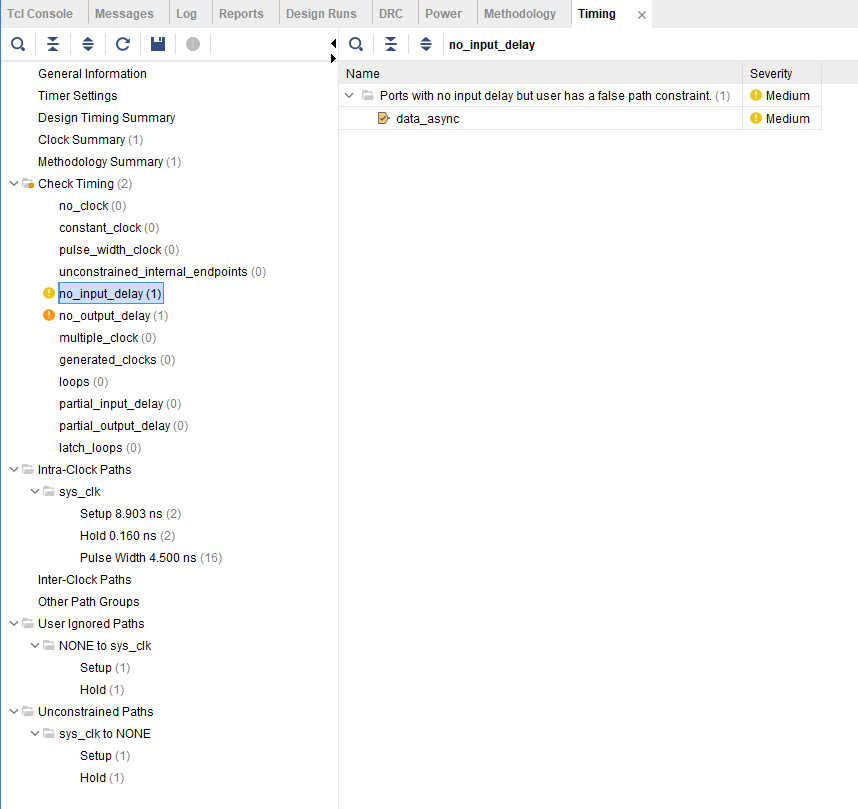


Also, the tcl command says that there are no timing exceptions.

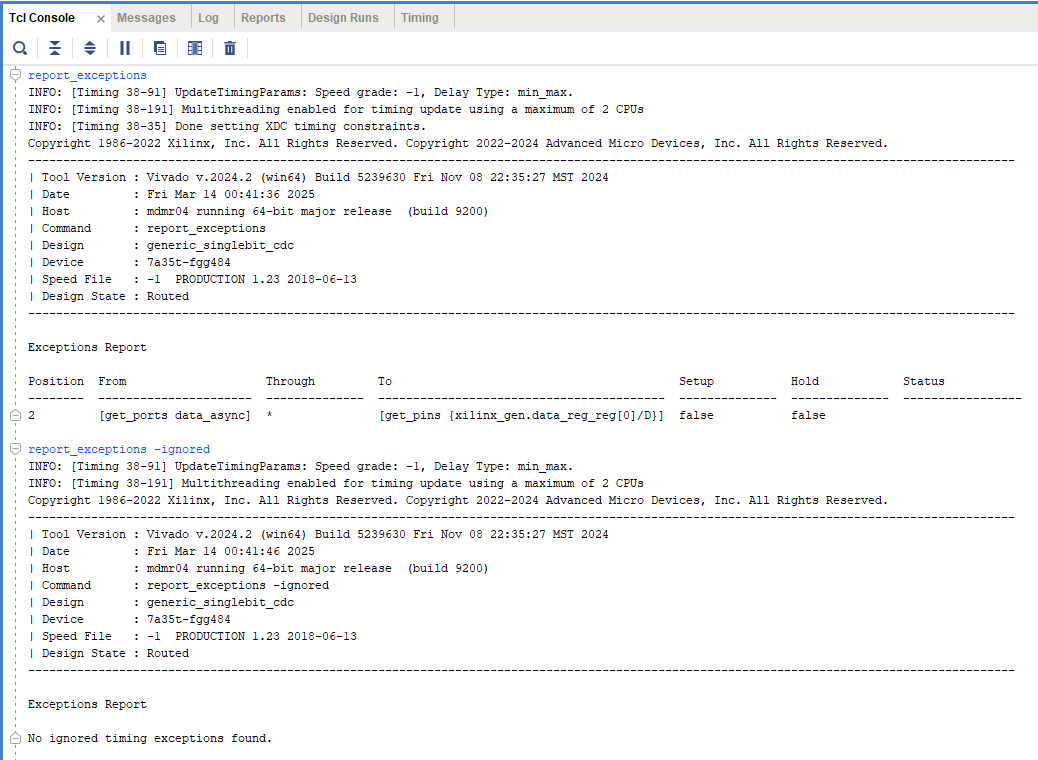
I defined the false path to the only below path because that data path comes from unknown clock domain and only that path should be aborted.



After that, I saw that the false path is given correctly.



Also, the timing exception Tcl (Tool command language) command says same thing, and the false path also was not ignored.



In conclusion, there is no need to analyze the timing of asynchronous data path that come to synchronizer and the false path constraint given to that path works correctly.