**BUG 1:**

**Source Day 1/Two\_Wire\_SPI**

A screenshot of a social media post

Description automatically generatedA screen shot of a video game

Description automatically generated

Looking at pictures of bug 1 when the bottom half of getting errors generating bitstream error is “[Synth 8-27] else clause after check for clock not supported”

A screen shot of a video game

Description automatically generated

A screenshot of a social media post

Description automatically generated

The code was fixed by removing the elseif clause and replacing it with an if. This didn’t break simulation and can now generate bitstream 😊

**BUG 2:**

**Source Day 1/LED Control**

A screenshot of a social media post

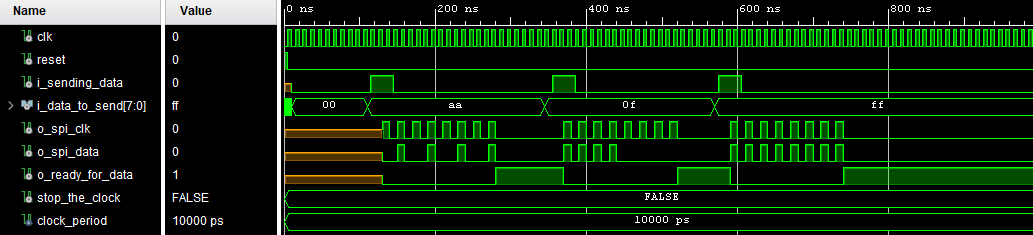
Description automatically generated

This code is suppose to use the two\_wire\_spi source to send some pixel data. It should be sending “00000000” for the first 4 transmissions but this doesn’t work. The outputs are just U I suspect this is due to the actual two\_wire\_spi code. You can see here we are waiting for ready\_for\_data to be high before sending anything. I think that the code is not initializing the o\_ready\_for\_data from the two\_wire\_spi source to be 1 initially. The testbench is shown below for led\_control.

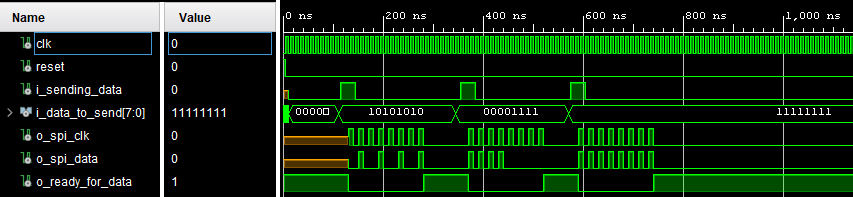
A screenshot of a computer

Description automatically generated

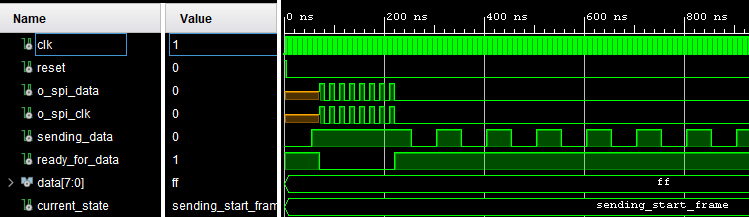
Below is the testbench of the two\_wire\_spi source which verifies my suspicions.



Below is the test bench for two\_wire\_spi fixed



Retesting the led\_control reveals a successful testbench now I just need to complete the code



**BUG 3:**

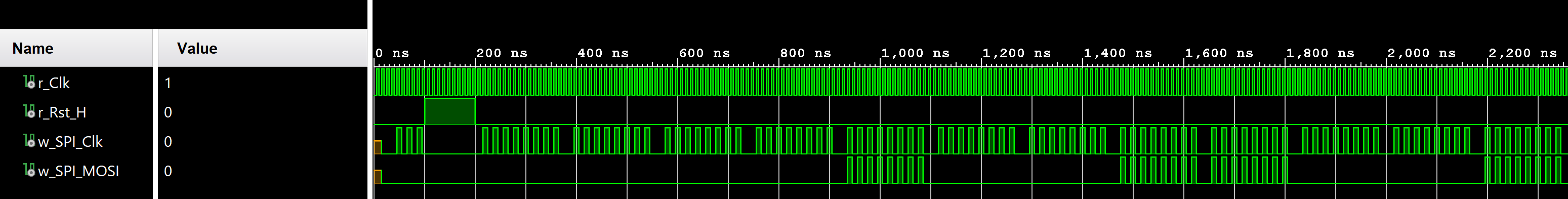
**Source Day 1/Two Wire Spi**

??

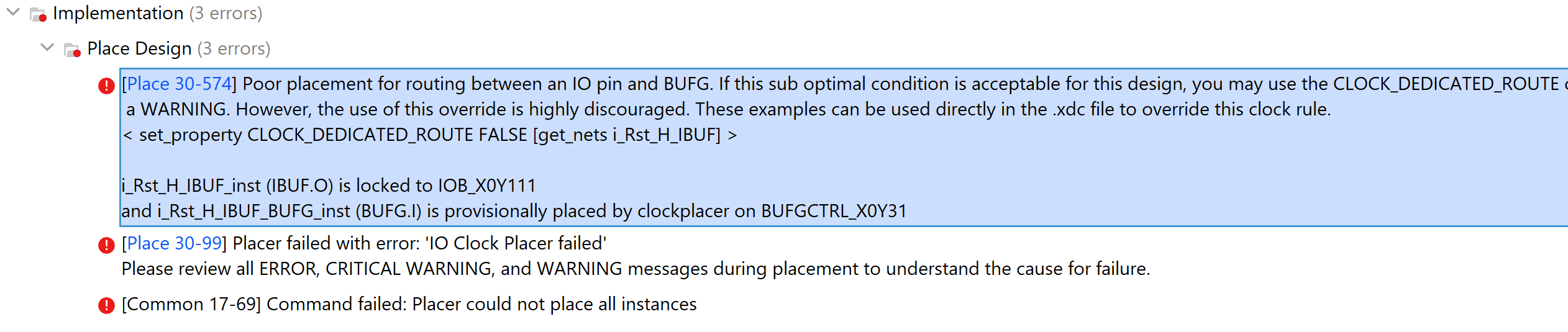
**Bug 4:**

**Source Day 2/Addresable\_Strips**

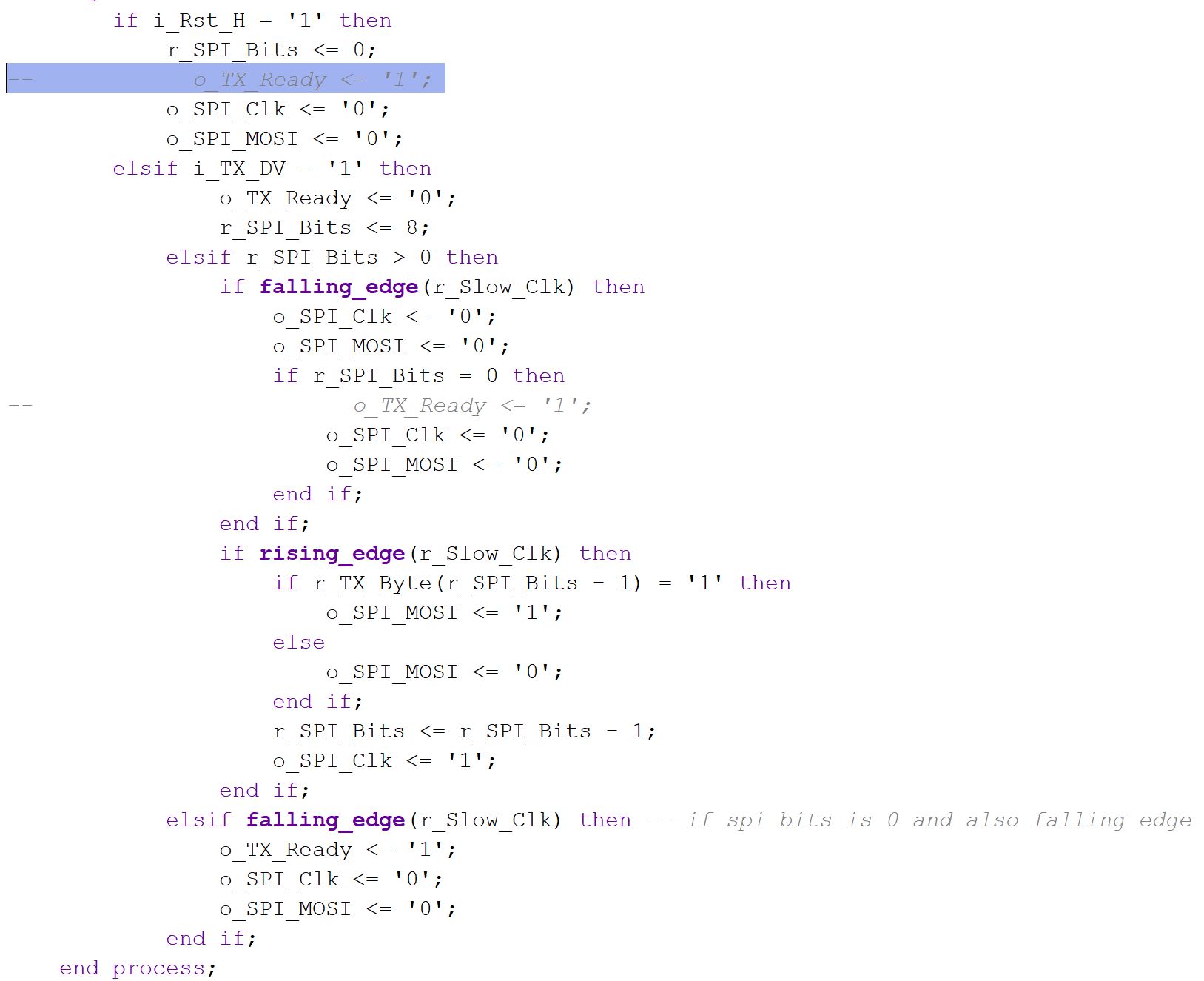
I decided to completely rewrite all the code I made and although I lost some time I managed to do everything eventually simulation below for just the start frame and the first two leds.



However even though I have no errors in the code and the simulation nworks fine now I can’t generate bitstream because of this error….



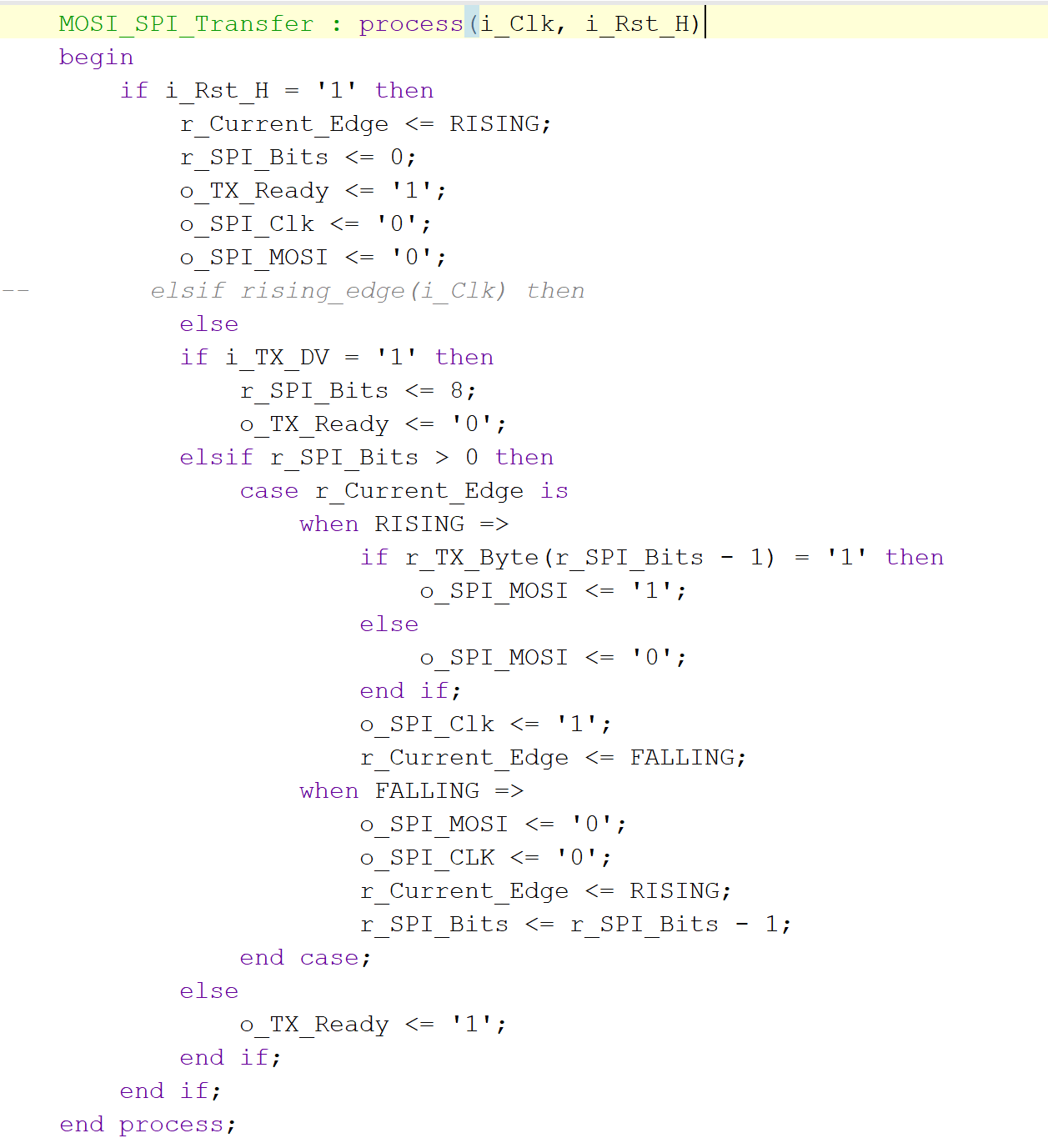
After like 2 hours of debugging I found the issue……. The highlighted line when uncommented caused the error. I have no idea why this isn’t a clock.

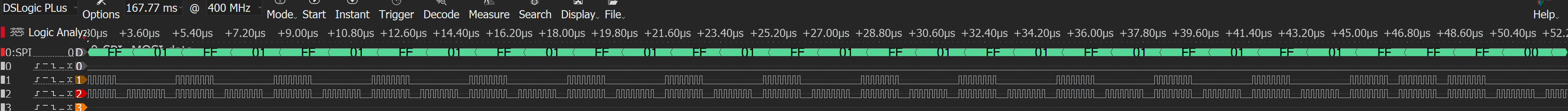


**Bug 5: Day 3**

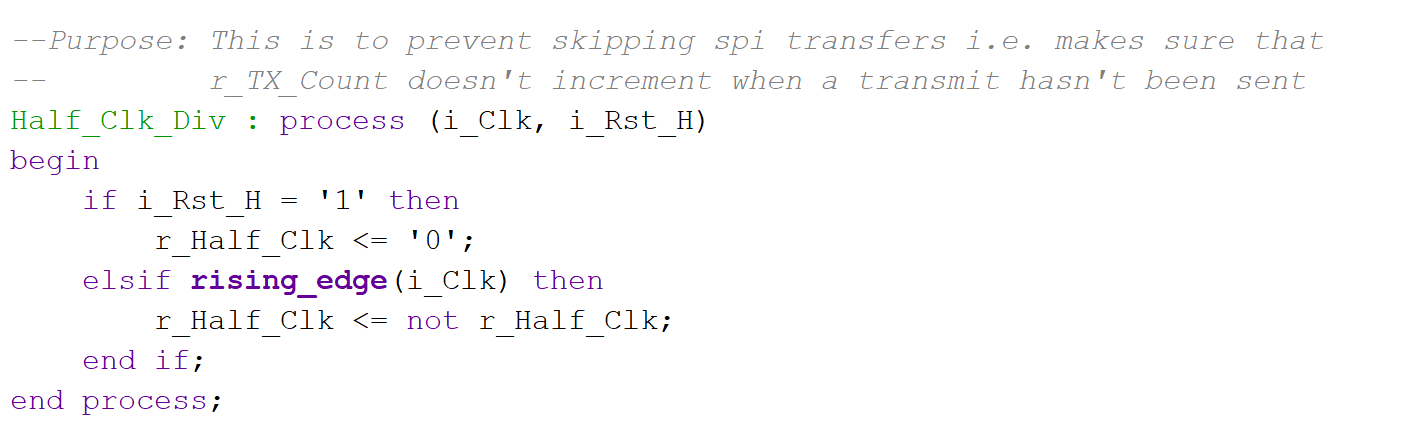
No pictures for this but even though my simulation is working every time I upload my code nothing is coming up on the FPGA when I probe with my logic analyser. I have confirmed that it’s not the issue with the logic analyser as I have tested other codes on the same pins and they worked. I even drove the spi pins with a clock directly and that worked but when I try to display the spi output it just stays low. I am pretty sure the spi code needs to be improved so I guess I’m going to rewrite it for the third time.

I fixed the problem with no output showing I suspect its to do with the fact that I was doing stuff on rising and falling edges before. Now I only do stuff on rising edges



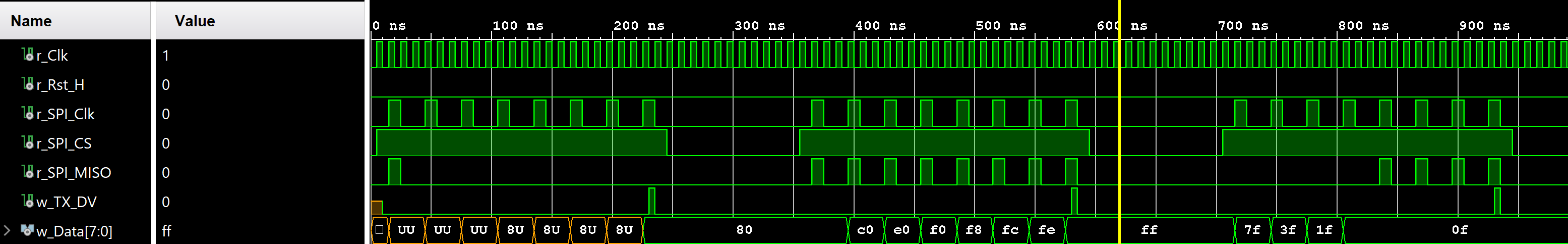


Success!! I just have an issue where the o\_tx\_ready which signifies that the spi bus is no longer busy is hanging for too long which accepts multiple spi transactions while actually only proessing one.

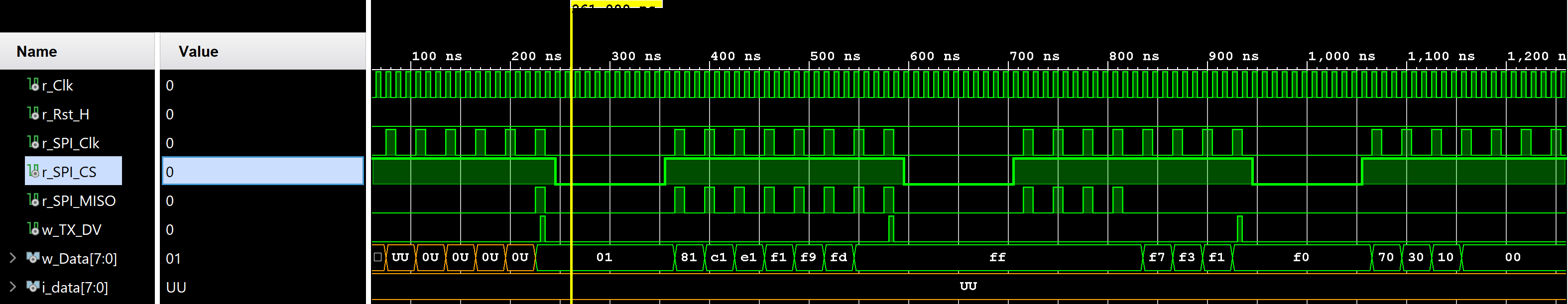


The error above where the spi transactions were being skipped was fixed by halfing the clock for the LED strip control this will mean that between transactions the code will be waiting 2 times longer before sending another spi byte but atleast it fixes the issue and it isn’t too big of a problem

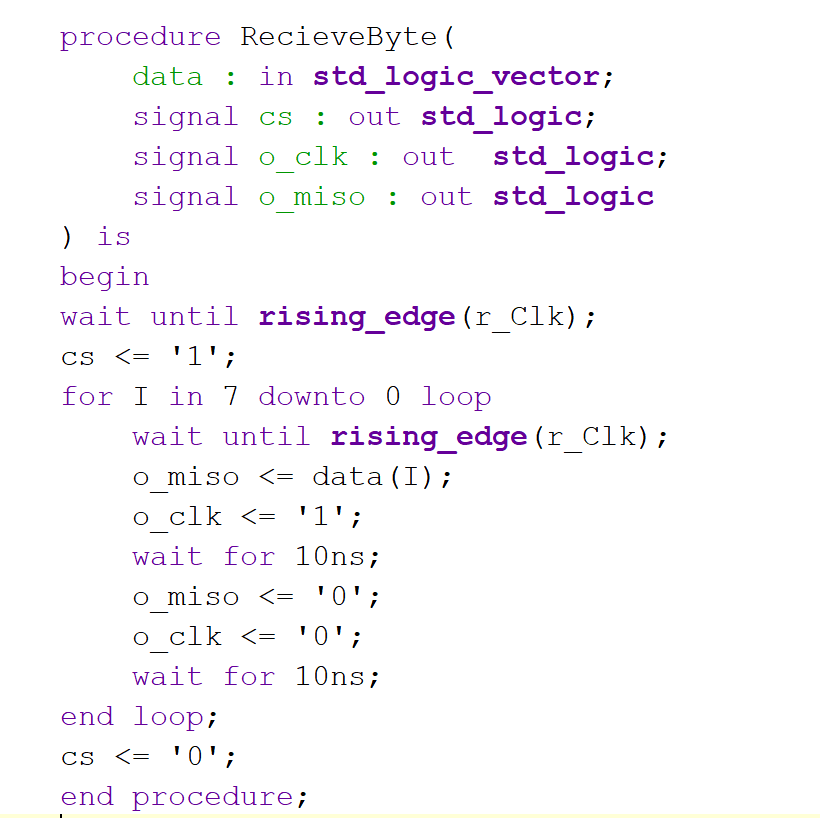
Bug 6 (Spi Slave):



In the above test bench I made some code to communicate between the esp32 and fpga. This communication will be SPI but the problem is when I am sending x”01” like the first transmission it sends the one first which means it’s sending the LSB first wheras the other spi master code for the fpga needs the MSB first.

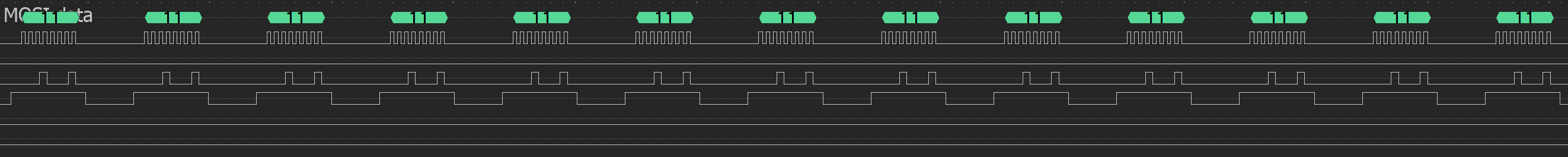


In the above test bench I fixed the msb first thing. The problem wasn’t the code it was the testbench



To fix I changed the for loop to go from 0 to 7 rather than 7 to 0

BUG 7



Again even though the test bench is fully functional there is no output on the spi outputs (bottom two waves)

After relooking the testbench I noticed that the spi write was only happening after the second spi read. After seeing this I then only sent one spi read and saw that the spi write wasn’t happening at all. I then suspected the dv wasn’t working properly so I scoped it and noticed that after a spi read instead of pulsing the dv it instead just turned it on shown in the test bench below. Although this shouldn’t be the reason that no spi output is showing on the logic analyser it should just be

