ECE 3544: Digital Design I
Project 2: Modeling the Timing of a Device

Student Name: **Jonathan Lemarroy**

Honor Code Pledge: I have neither given nor received unauthorized assistance on this assignment.

Jonathan Lemarroy

Grading: The design project will be graded on a 100 point basis, as shown below:

Manner of Presentation (30 points)

_____ Completed cover sheet included with report (5 points)

_____ Organization: Clear, concise presentation of content; Use of appropriate, well-organized sections (15 points)

_____ Mechanics: Spelling and grammar (10 points)

Technical Merit (70 points)

_____ General discussion: *Did you describe the objectives in your own words? Did you discuss your conclusions and the lessons you learned from the assignment?* (5 points)

_____ Design discussion: *Did you discuss your design approach, and the design decisions that you made as a part of implementing your modules?* (10 points)

_____ Timing analysis discussion: *Did you determine the minimum clock period that allows correct operation of the system?* (5 points)

_____ Testing discussion: *What was your approach to formulating your test benches? How did you verify the correctness of the modules you designed?* (10 points)

_____ Supporting figures: *Waveforms showing the correct operation of the various modules, Waveforms demonstrating valid and invalid behavior of the system.* (20 points)

_____ Supporting files: *Do the modules pass any tests applied by the grading staff? Modules whose declarations do not conform to the requirements of the project specification cannot be tested, and will receive no credit.* (20 points)

_____ **Project Grade**

Purpose:

The objective of this assignment is to gain experience using Verilog to model an even/odd parity bit generator/checker. Also, to gain experience timing a circuit with Verilog and analyzing the timing of a multi-level circuit to see its limitations.

Project Description:

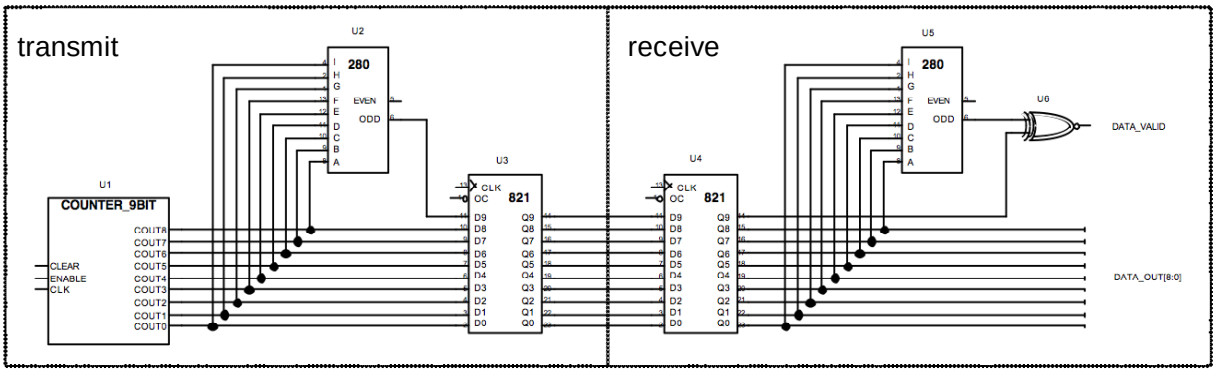
Study and analyze the clock and 4-bit counter modules provided and simulate the circuits using their respective test benches. Using the 4-bit counter as a basis, design and simulate a 9-bit counter.

Next, design and simulate a 9-bit even/odd parity-bit generator/checker whose output is an even-bit and an odd-bit. Using the HC280 datasheet, assign the typical propagation delay parameters assuming the device is operating at 4.5V and 25C.

Then design and simulate a 10-bit register whose values only update on the positive edge of a clock.

Using the 9-bit counter, parity-bit generator, and 10-bit register, create a data line transmitter and receiver that is constructed like **Figure 1.**

Figure 1: Transmitter and Receiver



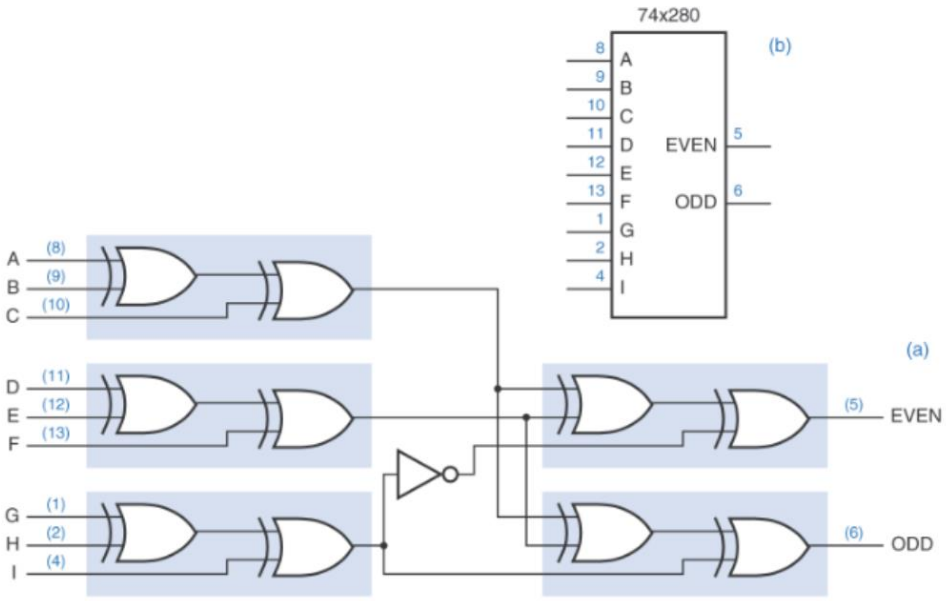
Lastly simulate and analyze the transmitter and receiver together when operating with two different clock periods. The first operating under a clock frequency that maintains the expected “valid” signal output. The second operating with a frequency too faster for the circuit.

Technical Design:

When designing the 9-bit counter the only things modified from the 4-bit version provided were the number of bits and what count value that causes the circuit to start over from 0 again. See **Appendix** for the 9-bit and 4-bit counter’s waveform samples.

The idea for the 9-bit parity-bit generator was taken from lecture, see **Figure 2**. See **Appendix** for the waveform sample. Its delays from the datasheet were found to be 20ns for even output and 23ns for odd.

Figure 2: 9-Bit Even/Odd Generator



The 10-bit register was the simplest of the circuits to make. It is created using an "always" block whose trigger is the positive edge of the clock, and where each of the regs are assigned in parallel by using non-blocking assignments. See **Appendix** for a sample waveform.

The transmitter and receiver were made by just wiring the circuit found in **Figure 1**. Because the 9-bit counter has counter increment propagation delay of 15ns and the HC280 odd bit has a propagation delay of 23ns. The chosen clock periods for normal and error prone operation modes were 20ns and 40ns, respectively. See **Appendix** for their sample waveforms.

Analysis & Conclusion:

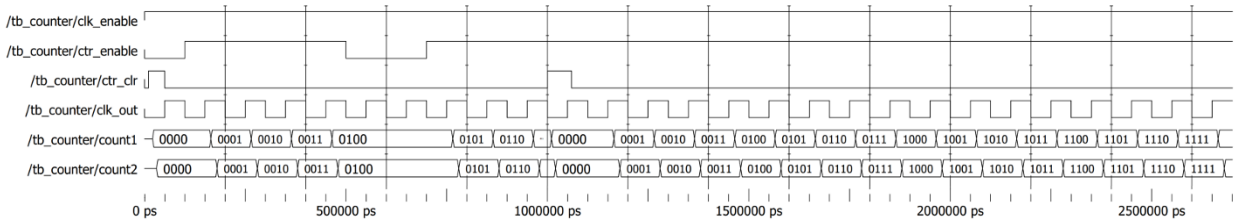
Although the waveform for the last part of this assignment shows the valid bit toggling, this is its normal function even with the longest period possible. This is due to the difference in delay caused by the receiver's parity bit checker and the 10-bit registers output, which is exactly 23ns. This can be fixed in simulation by adding a delay before the XNOR's input and the 10-bit register's MSB output, but when implementing the circuit delay cannot just be manually inserted into it.

Because the longest/critical path of this circuit is found in the transmitter's propagation, the minimal requirement for the clocks period must be greater than 38ns. But because these are using typical delay instead of maximum delay choosing a value very close to 38 would very likely cause actual problems in a physical circuit.

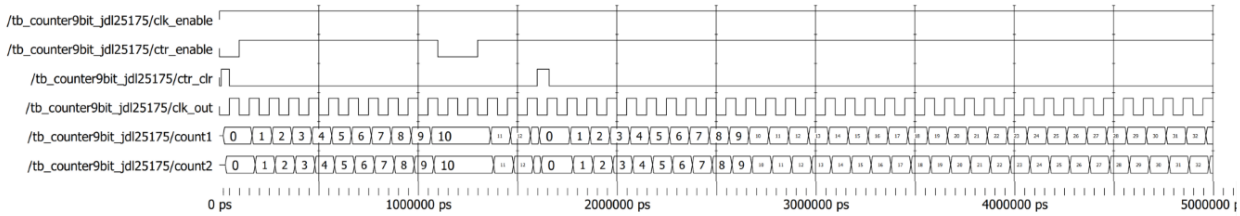
This project gave me experience with some of the difficulties that come into play when accounting for the timing differences of modules implemented in a multi-level circuit.

Appendix:

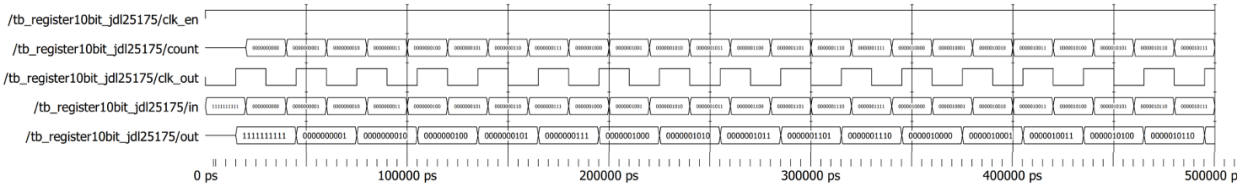
Waveform: 4-Bit Counter



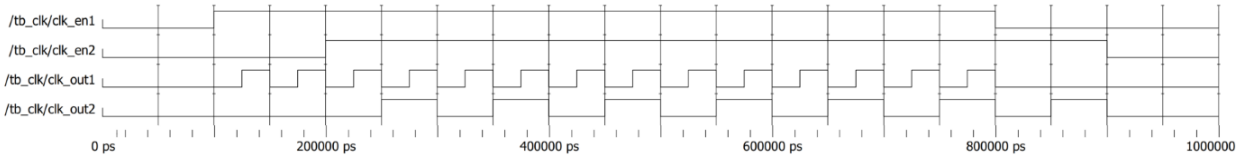
Waveform: 9-Bit Counter



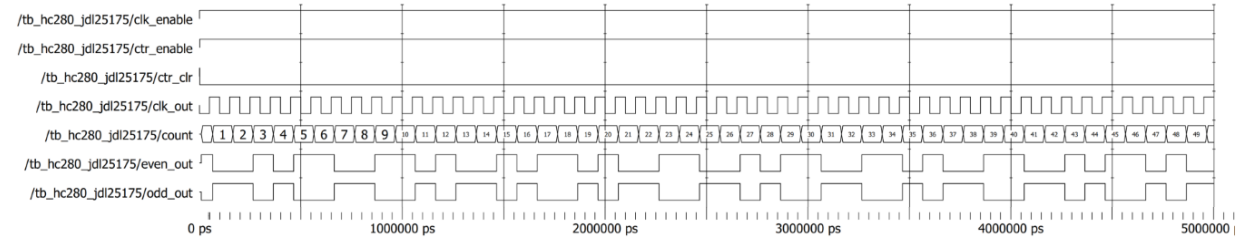
Waveform: 10-Bit Register



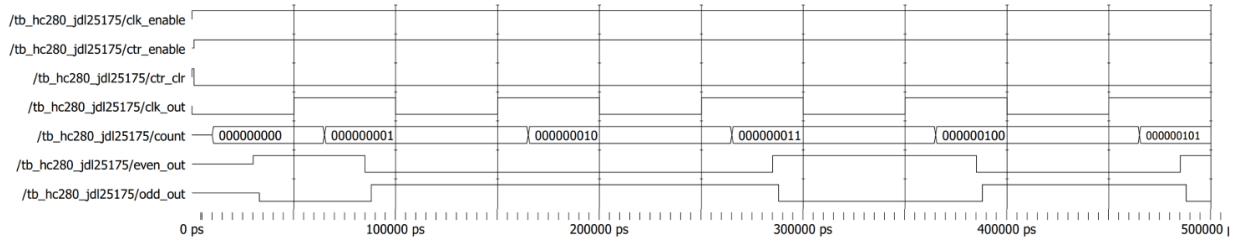
Waveform: Clock



Waveform: HC280 Without Delay



Waveform: HC280 With Delay



Waveform: Transmitter and Receiver with Both Clock Periods