Explain how you might use this interactive typing in combination with a Tcl file in a real simulation and debugging activity.

You could immediately add new forces to test different inputs and run for various amounts of time to quickly see the result rather than restarting the entire simulation.

Write up an explanation of how this matches or does not match the waveform you drew by hand above. Does the simulation match your understanding? What is the transition ordering of the output signals in response to an input change? Did you draw the waveform correctly by hand? If not, how did you do it wrong?

This somewhat matches the waveform that I drew. I didn’t really account for the exact delays which viviado takes into account. The simulation does match my understanding of how the waveform should be shown. I didn’t quite draw the waveform correctly at the end because I was unsure as to which output would be determined first. I thought that if both inputs were zero, then the outputs would revert to what they were at the beginning when both inputs were zero.

Answer the following questions while considering only the waveform from time 0 to 160ns: (1) From this simulation, what do you deduce the tCLK-Q time is for this flip flop? (2) Do you get the same delay for when Q rises vs. when Q falls? You should not. Explain why this is. (3) So, if there are different values, which one would you use and why? (4) How does this compare to the tCLK-Q you would calculate from the book and its analysis? Are they same? If not, does your simulation have a bug? Explain any discrepancies. If you want full credit for this, you need to answer all the questions, not just some.

1. 9ns
2. No, there is a delay when one of the nor gates executes and has to transfer it’s input into the other nor gate. This happens in the slave and master nor gate crosses. As a result, the delay is not exactly the same for when the final outputs are given resulting in different lengths of the rises and falls. A lack of delay could also speed up when a gate executes.
3. If there is a nor gate that requires a value for another gate, it will take the value that is ready to be input and wait for the other. For example, the crossing nor gates have the longest delay. The outputs of each nor gates won’t change until it receives both of its new inputs and is ready to be executed. That’s why there is a difference between the rise and fall times of q. Despite different values, you should just use the one that corresponds with the longest path and delay to stay consistent.
4. 9ns/7ns, They are sometimes the same. When the outputs of gates are crossed over to be the input of another, it usually adds more time to the circuit. When qbar falls, there seems to be a lack of delay causing it to be 2ns faster than the rise delay. The falling edge of q is faster than the rising edge. This causes the simulation and the calculated time of tCLK-q to sometimes differ.

Answer the following questions: (1) From this simulation, what would you deduce the tSETUP time is for this flip flop? (2) Is this different than what you computed above? If so, explain in detail why. (3) Is the tSETUP time you see in this simulation different for when D rises vs. when D falls? It should be. Thus, what would be the proper value to use as a general tSETUP for this flip flop so you can design with it and create working circuits? Why?

1. 7ns on the first flip flop and 8ns on the next.
2. Yes, the longest path that d can travel has one not gate added while the other path doesn’t . This difference causes a 1ns difference in the tSETUP flip flop which changes depending on whether d is on the rising or falling edge.
3. Yes, You should use the setup time(8ns) that is given in the simulation to account for any extra delays in the circuit to maintain stability. The critical path is the longest path that the circuit will travel through. This is the setup time used for the flip flops used to make working circuits.