Victor Yuan

October 20 2016

Lab 8

Introduction:

In this lab a Set-Rest latch was examined. A set reset latch has two inputs S and R, S is the input to one nor gate B while R is the input for a different nor gate A the output from B is P, which is also the other input into nor gate A the output from A is Q, which is also the other input into nor gate B.

Team Member Responsibilities:

This Lab was done alone.

Materials:

ELENCO Analog kit, assorted wires, Verilog simulation program through Putty, TTL Logic chips.

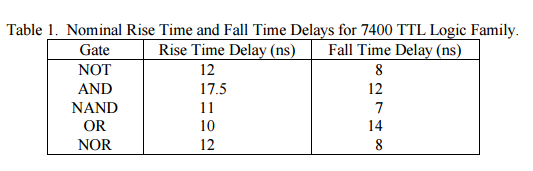
Procedure:

Part I:

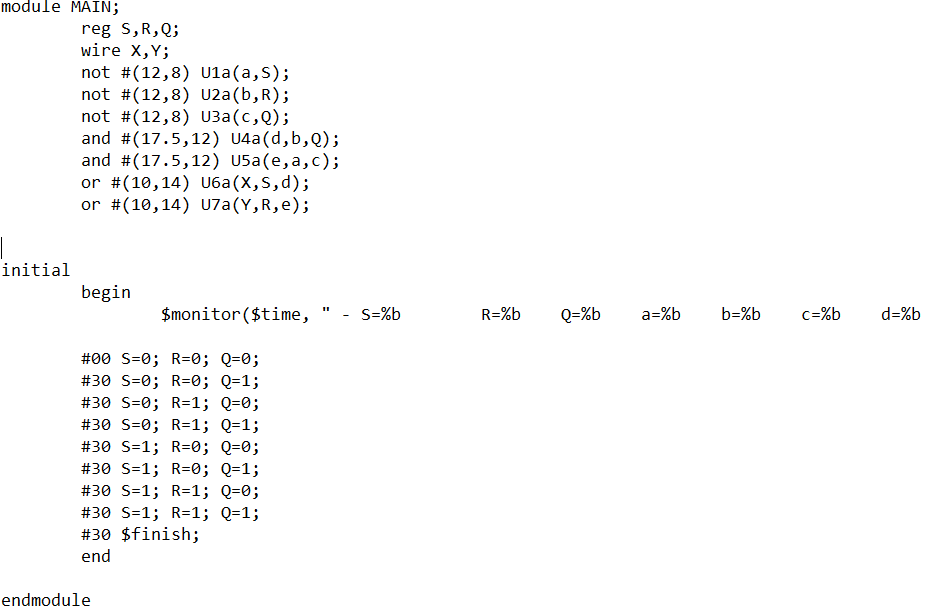
First derive a truth table form figure 1 for all values of R and S and initial Q for the outputs of Q+ and P+, the next state.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q | S | R | Q+ | P+ |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | x | x |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | x | x |

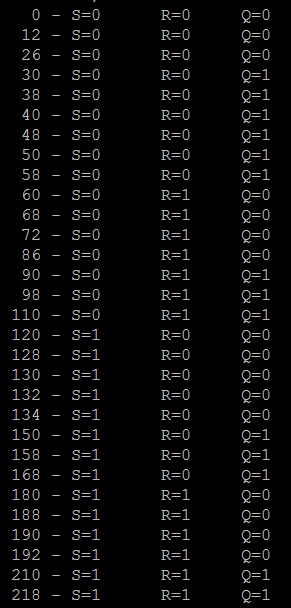
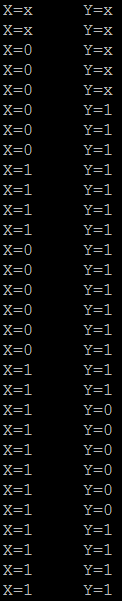
Then simply the above into a Boolean expression that can be simulated in Verilog. The code for the simulation should include the correct Delay times for each gate the delay times for teach gate is listed in the table below



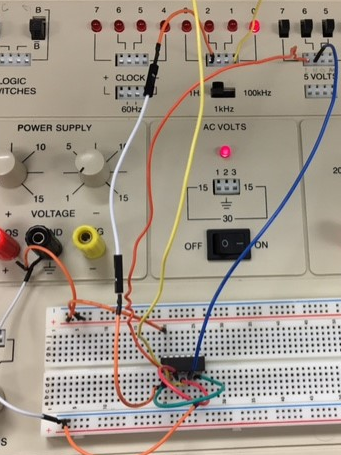
The finale code for the simulation should look like the following.



The output should look like the following:

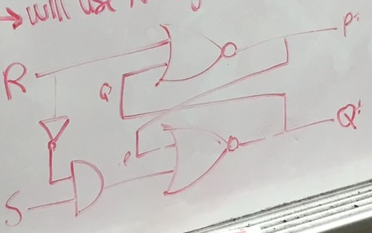
 

After the simulation is checked with the turth table derived implment figure 1.



Part 2:

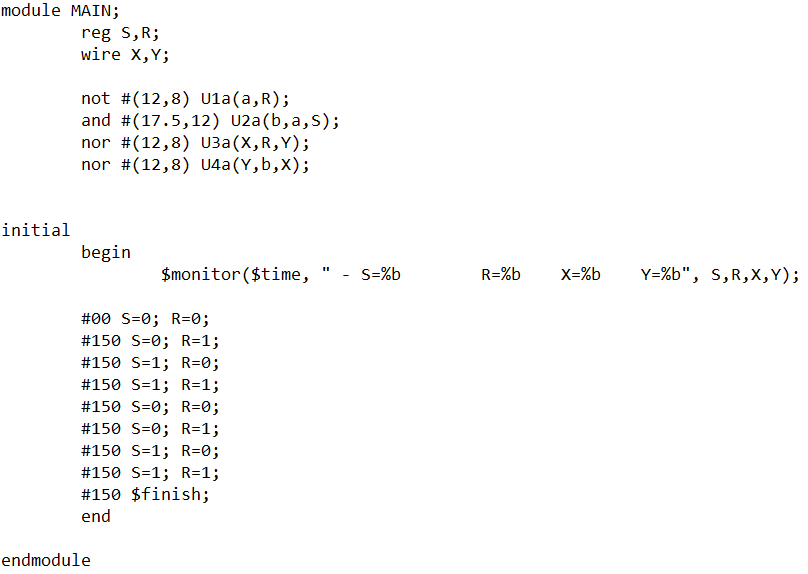
A truth table is derived from the following circuit.



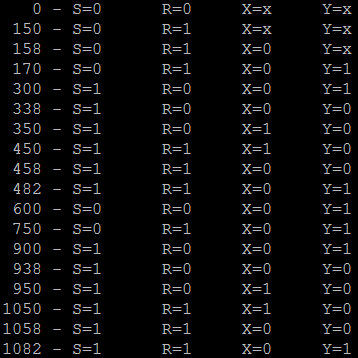
Truth table for the above circuit (derived).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q | S | R | Q+ | P+ |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |

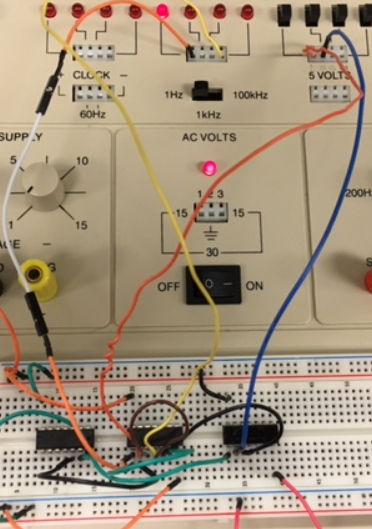
Then simulate the circuit in Verilog as is with proper delays.



The output should match the derived truth table

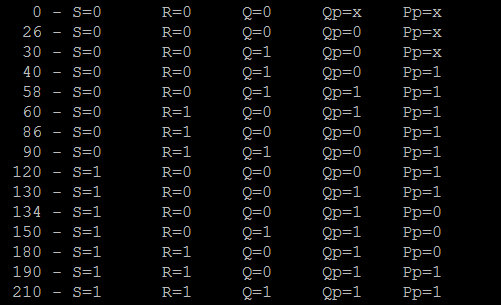


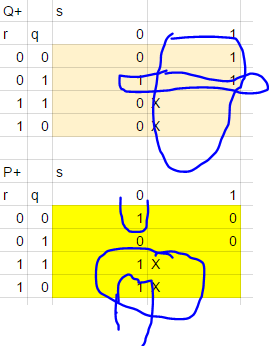
After comparing the simulation with the truth table implement it into hardware.



Questions:

I. Experiment 1:

1. Using Verilog, derive the truth-table for a SR latch from your simulations, assuming that Q, S and R are the inputs and Q+ and P+ are the outputs.
   1. 
2. Using the truth-table from part I.2 above, derive Karnaugh maps for Q+ and P+ for the SR latch. Derive a minimal SOP realization for Q+ and P+.



Q+ : S+R’Q

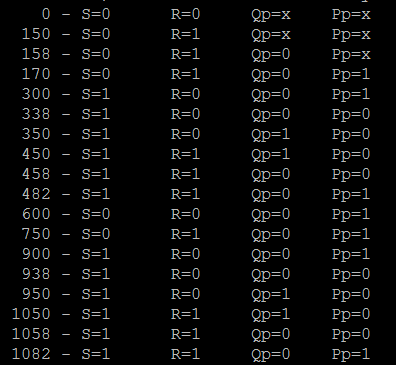
P+ : R+S’Q’

1. Under what conditions will P+ always be the complement of Q+, that is, P+ = (Q+)’?

P+ is the complement of Q+ except for when R and S are both logical 1.

II. Experiment 2:

1. Using Verilog, derive the truth-table for a reset dominate SR latch from your simulations using the 7400 TTL Logic Family, assuming that Q, S and R are the inputs and Q+ and P+ are the outputs.



1. Using the truth-table from part II.1 above, derive Karnaugh maps for Q+ and P+ for the SR latch. Derive a minimal SOP realization for Q+ and P+.

Q+: S+R’Q

P+: R+S’Q’

Under what conditions will P+ always be the complement of Q+, that is, P+ = (Q+)’?

They will always be the complement of each other

Conclusions:

1. How do the truth-tables for the next-state of a SR latch compare as derived from the theory derived in class and your Verilog simulations? Are there any differences?

They matched only the simulations are extra repeated data points.

2. Did your modification to the logic for the reset dominate SR latch result in P+ = (Q +)’ always being true?

Yes the modification resulted P+ to always be the compliment of Q+.