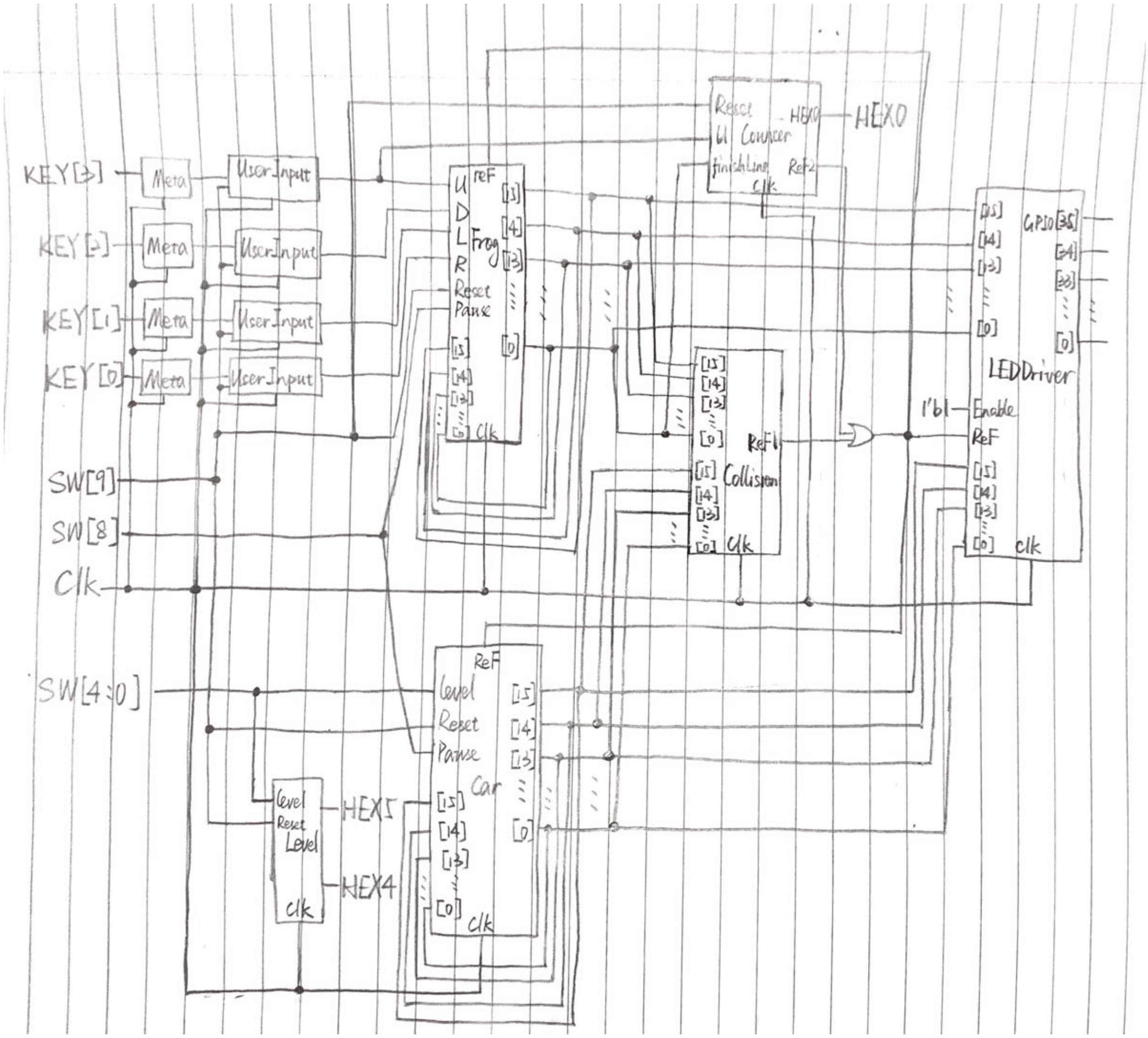
EE 271

Cynthia Li

Lab 8 Report

1. **Live demoed to Nick on 3/13/2021.**
2. **A block diagram of your entire system, and a detailed description of how it should operate (i.e. a User's Manual).**



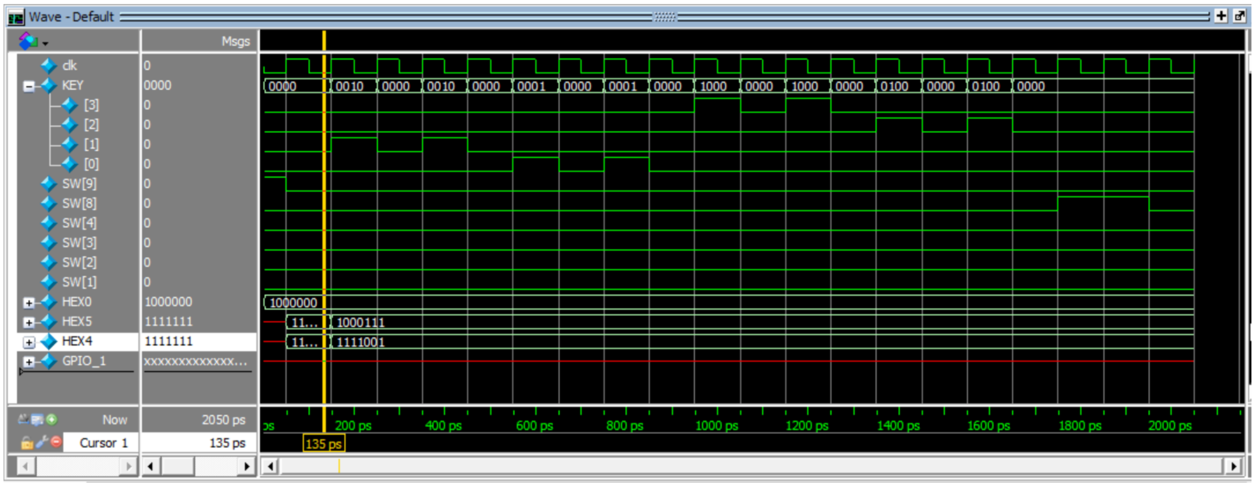
User's Manual:

- User input components and their functions:
  - Switch 9: to start the game or restart the game
  - Switch 8: to pause the game

  - Switch 4/3/2/1: to change difficulty level (change car’s speed) (switches correspond to level 4, 3, 2, 1, higher number indicate harder game)
  - Key 3/2/1/0: to move up/down/left/right
- Display components and their functions:
  - HEX5/4: to show users the current difficulty level
  - HEX0: to show users the current score
  - 16*16*2 LEDBoard: green dot is the frog, red pattern has two shapes, representing cars and motorcycles.
- How to play:
  - Push up and push down Switch 9 to start the game. We can see the frog (green dot) is on the bottom row of the board, the cars and motorcycles (red patterns) are moving left or right on the roads, the cars in different directions are separated by one row as a road separating strips. Set/change your difficulty level by pushing up Switch 2/3/4 (you may want to observe how fast the car goes and then determine which level you want to play), you will see that the difficulty level is displayed on HEX5 and HEX4.
  - Now you may use Key 3 to move the frog up, Key 2 to move it down, Key 1 to move it left, and Key 0 to move it right. Take some time to try moving around the frog, you will notice that the frog cannot cross the bottom boundary, it must cross the road, when it crosses the left boundary of the board, it will appear on the right, when it crosses the right boundary of the board, it will appear on the left. You may use Switch 8 to pause the game.
  - After you familiarize yourself with the Keys, you may use the Keys to lead the frog cross the road. Be careful to avoid the cars and motorcycles (red patterns), if you get squished by them, you will be forced back to the beginning; but once you cross the upper boundary of the board, you have successfully crossed the road, the scoreboard HEX0 will increment your score by 1. You may use Switch 9 to restart the game and follow the same procedure to train your frog.

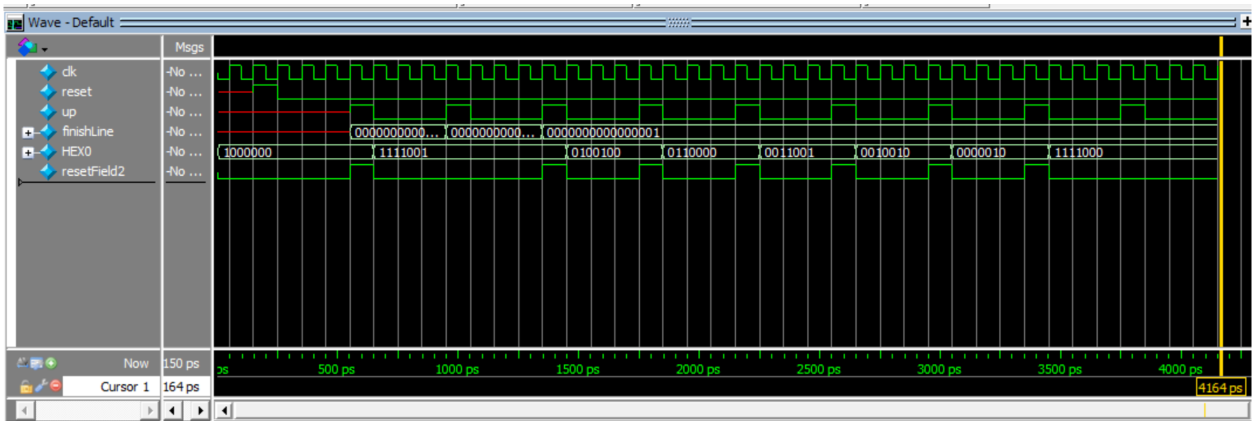
- ○ After you win 7 times, you have successfully trained your frog. You may continue to play the game, but the scoreboard will stop increasing your score.

3. **A screenshot of ModelSim simulation of your top-level entity with explanations.**



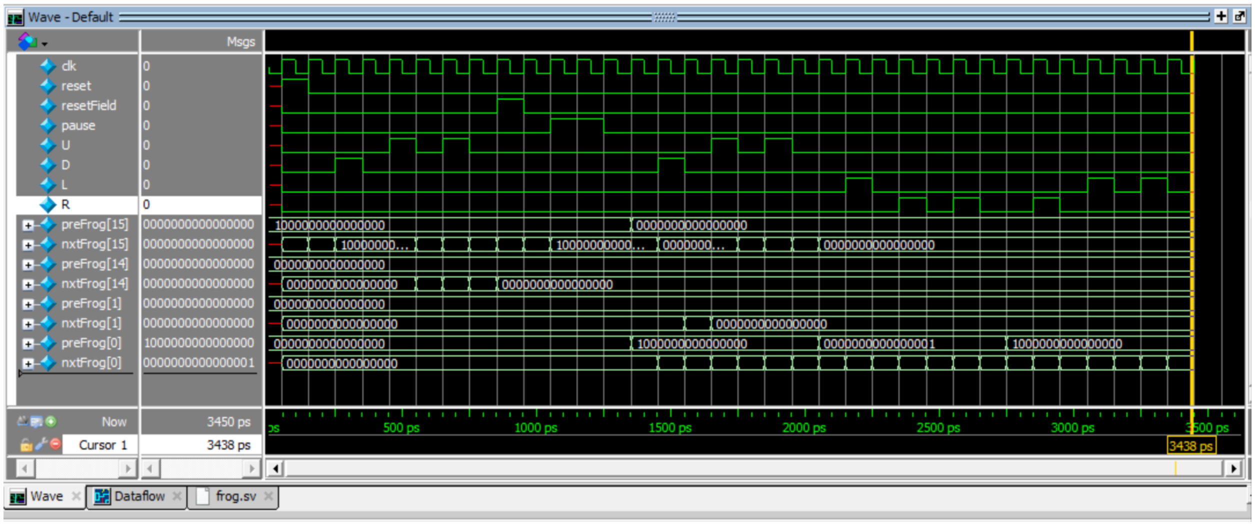
We can see that HEX0, HEX5, HEX4 all have correctly displayed, indicating the design is working. Also since we have made sure that each segment works properly, I've also previously tried to add frog and car in the top entity as output, seeing them simulate correctly and knowing that everything is wired correctly, I'm assuming that there is something with modelsim or with LEDdriver that doesn't like to produce GPIO_1 here (though the design does work on the board after we change the clock). Also, since GPIO_1's value is hard to analyze if my output LED matrix is correct, here it doesn't really matter to me.

4. **Screenshots of ModelSim simulation of your essential modules with description, along with other material you find helpful (e.g. state diagrams).**



Simulation of counter_testbench

Test if the score (HEX0) changes from 0 to 7 and if resetField2 turns to true when collision happens (up & finishLine != 16'b0000000000000000) and test if the score stays the same when no collision happens. Expect to see HEX0's display changes from 0 to 7, resetField should be triggered corresponding to the collision time.



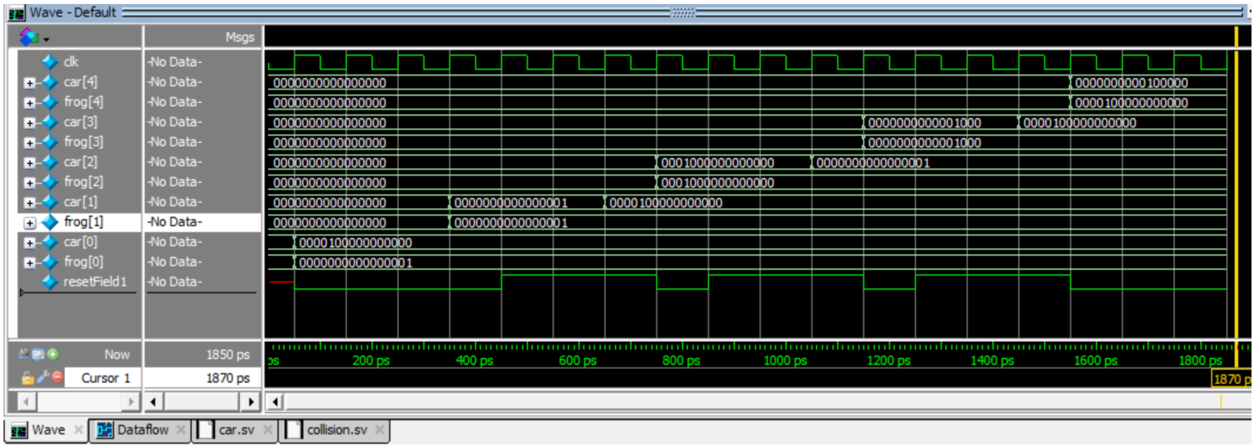
Wave - Default

	Msgs
clk	0
reset	0
resetField	0
pause	0
U	0
D	0
L	0
R	0
preFrog[15]	0000000000000000
nxtFrog[15]	0000000000000000
preFrog[14]	0000000000000000
nxtFrog[14]	0000000000000000
preFrog[1]	0000000000000000
nxtFrog[1]	0000000000000000
preFrog[0]	1000000000000000
nxtFrog[0]	0000000000000001

Now	3450 ps
Cursor 1	3438 ps

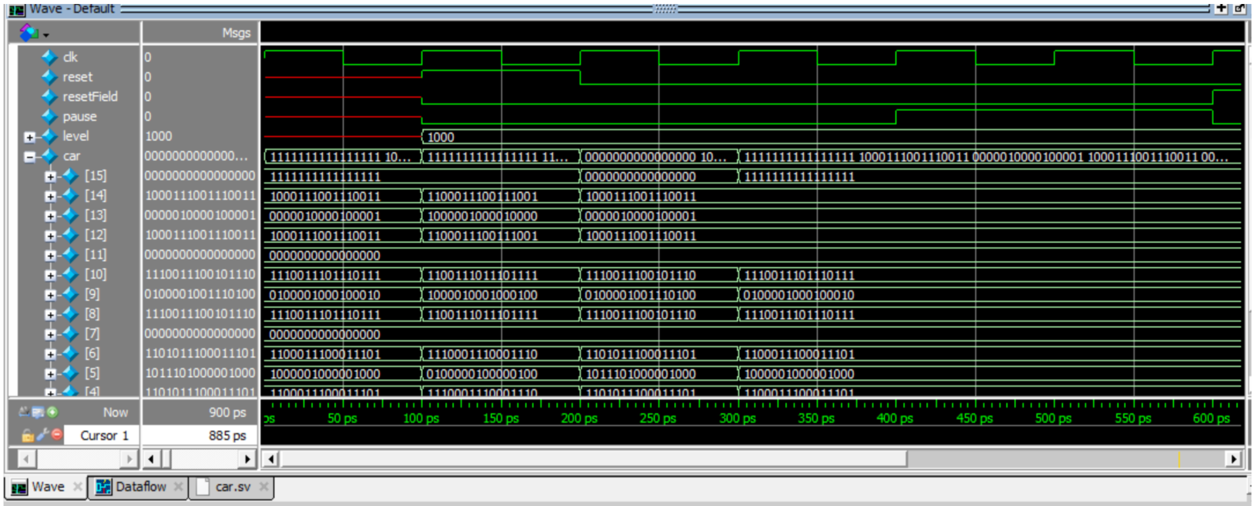
Simulation for Frog_testbench

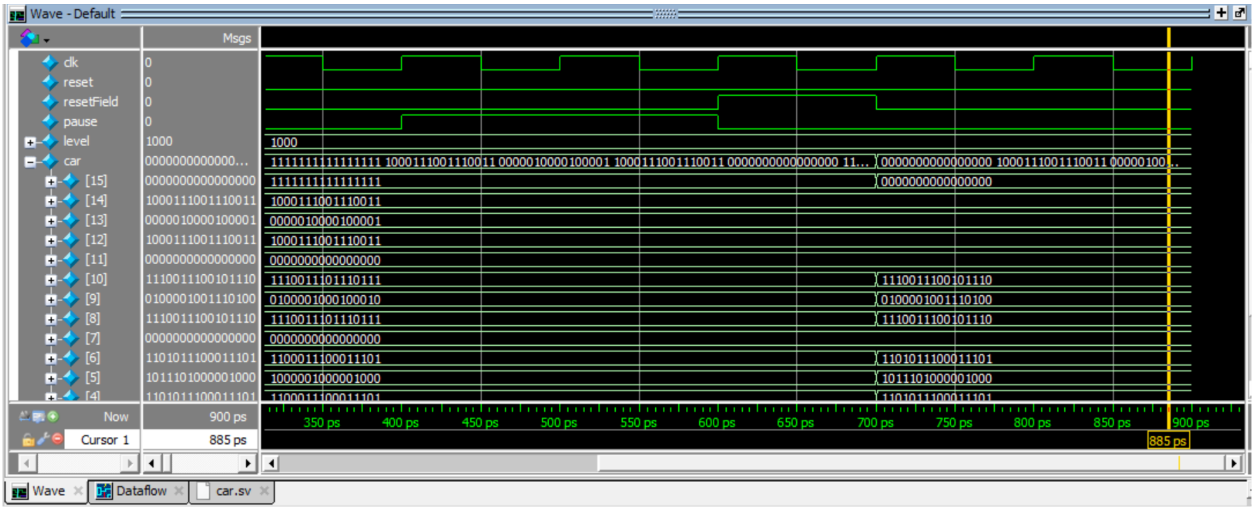
Test if reset, the frog will be turned on, if resetField changes the frog to default pattern, if the pause stops the frog, if the frog can move up, down, left and right, test the cross boundary cases for up, down, left and right. Expect reset and resetField change the frog to default pattern, pause should let frog stay in current pattern, normal up and down should move the frog up a row or down a row, normal left and right should shift the 1 in the 16'b value left or right a bit. When the frog crosses the bottom boundary of the board, the pattern should stay the same, when the frog crosses the upper boundary, the pattern should return to default pattern, when the frog crosses left boundary, it should appear on the right boundary in the row, and vice versa for crossing right boundary.



Simulation for collision_testbench

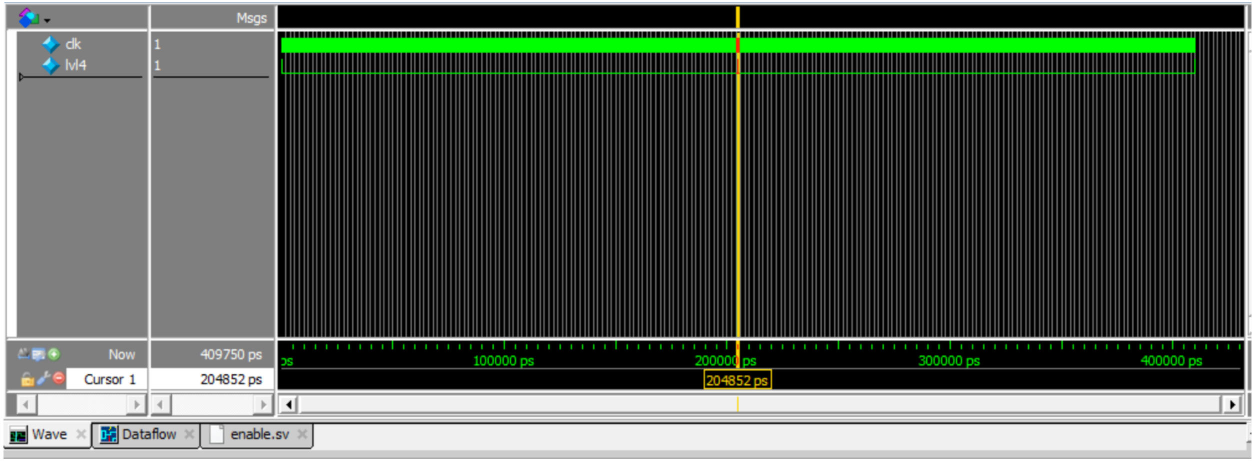
Test if when the frog and car are in same position (collide), resetField 1 will be turned to true. Expect the first case (no collision) to set resetField1 to 0, the second to fourth cases (collide) to set resetField1 to 1, the last case (no collision) to set resetField1 to 0.





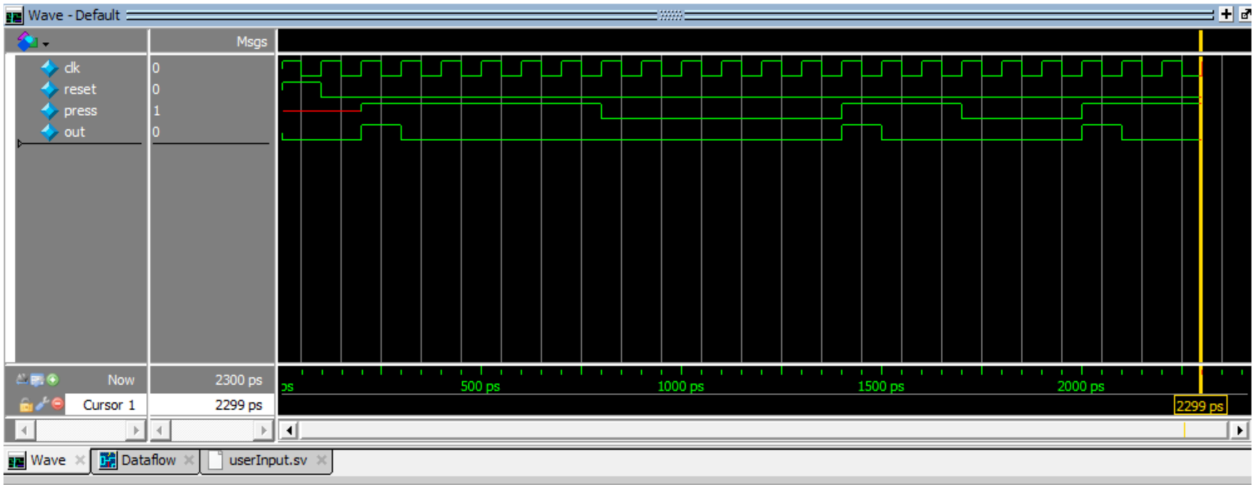
Simulation for Car_testbench

Test if reset, resetField change the pattern of the car to default, test if pause stops the car moving, test if the car moves. Expect to see, in first screenshot, at the very beginning, the car moves ([14:12] to the right, [10:8] to the left); after reset is turn to true, the pattern is turned back to default (obvious for car[15]), after pause, the pattern stop changes; after resetField is turned to true, the pattern is turned back to default.



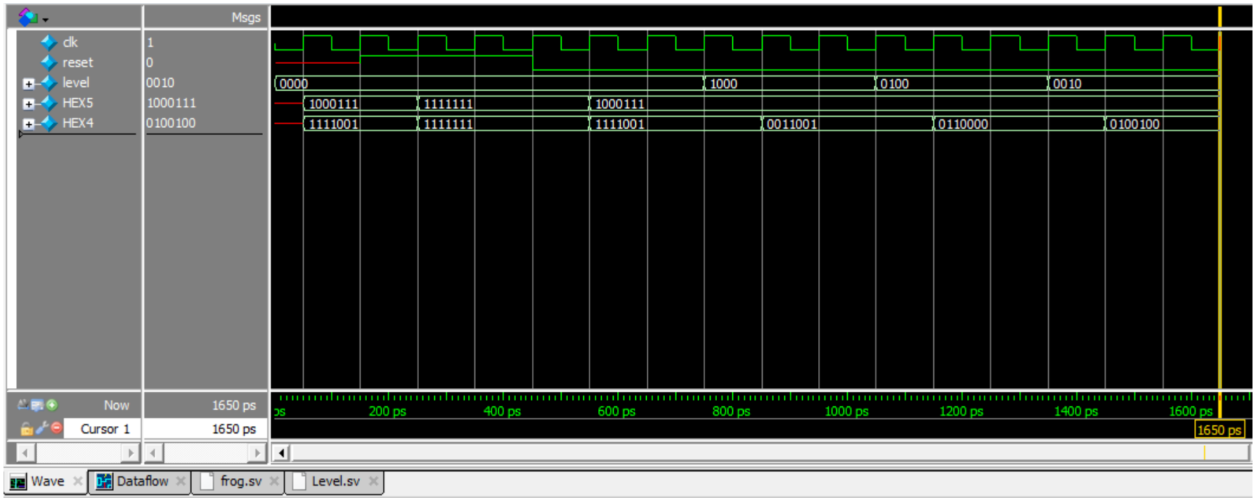
Simulation for Level4_testbench

Test if Level4 can be used as a slowed clk to trigger the car. If it does, Level3, Level2, Level1 should also work as their only variation is the "cycle." Expect to see lvl4 turn true every 2048 cycles.



Simulation for userInput_testbench

Test if out is true for only one period given a long pulse in input. Expect to see that when press is true for many consecutive cycles, out is only true for one cycle.



Simulation for Level_testbench

Test if reset turns off the level display(HEX5, HEX4), if the display is correct for each level. Expect to see reset turns off HEX5, HEX4, level=4’b1000 output HEX5 as “L”, HEX4 as “4,” level=4’b0100 output HEX5 as “L”, HEX 4 as “3,” level=4’b0010 output HEX5 as “L,” HEX4 as “2,” other level output HEX5 as “L”, HEX4 as “1.”

5. **Approximately how much time did you spend on this lab (including reading, planning, design, coding, debugging etc.)?**

   Reading: 2 hours

   Planning: 3 hours

Designing, coding, and debugging: ~37 hours

Finishing report: 4 hours

Total: 47 hours