Lab Assignment Report 8 Part 3

Course: COL215 – Digital Logic & System Design

Department of Computer Science & Engineering, IIT Delhi
Semester III, 2025-26

Students: Rohit Shakya (2024CS10410)

Aryan Patel (2024CS50489)

Introduction:

The task is to generate a rival car (different color) in a random position at the top of the road and move it vertically downward with constant speed. The rival car starts at the top of the road and vertically moves down the image. If no collision occurs with main car, then the rival car is re-generated again from the top.

Design Decisions:

- **8-Bit Pseudo random generator:**
  As mentioned in the pdf we have created the Linear Feedback Shift Register (LFSR). The parameter (seed) is bitwise XOR of 8 LSBs of two kerberos IDs and it came out to be 11110011. The left and right limits as 44 and 104 in decimal. The new LSB of output out is calculated as out[2]^out[4]^out[5]^out[7]. Then in the always block we are assigning the next_out as {7 LSB of out (previous value) , feedback}.
  If next_out > 2*right then next_out = left + next_out – 2*right.
  Else If next_out > right + left then next_out = next_out – right.
  Else If next_out > right then next_out = next_out – right + left;
  Else if next_out < left then next_out = next_out + left;
  These condition will keep the next_out under the bound and similar thing done when reset is on.

- **Moving rival car down Logic:**

  We defined rival_x and rival_y as the bottom left corner of the car. Then we have defined a register scroll of 22 bytes and a localparamter MOVE_COUNT_MAX of d'1680000 which is equivalent to 16 ms. When scroll reaches to MOVE_COUNT_MAX then it increment the rival_y by 1 if rival_y is in bound otherwise setting it to OFFSET_BG_Y.

- **Collision Detection:**

  To detect collision we have to check the conditions in both the directions x and y.

  For x axis:

  If(car_x > rival_x) then rival car bottom right corner should be more than car_x for collision which is checked by car_x < rival_x + main_car_width.

  Else if(car_x < rival_x) then car right top right corner should be more than rival_x for collsion which is checked by car_x + main_car_width > rival_x.

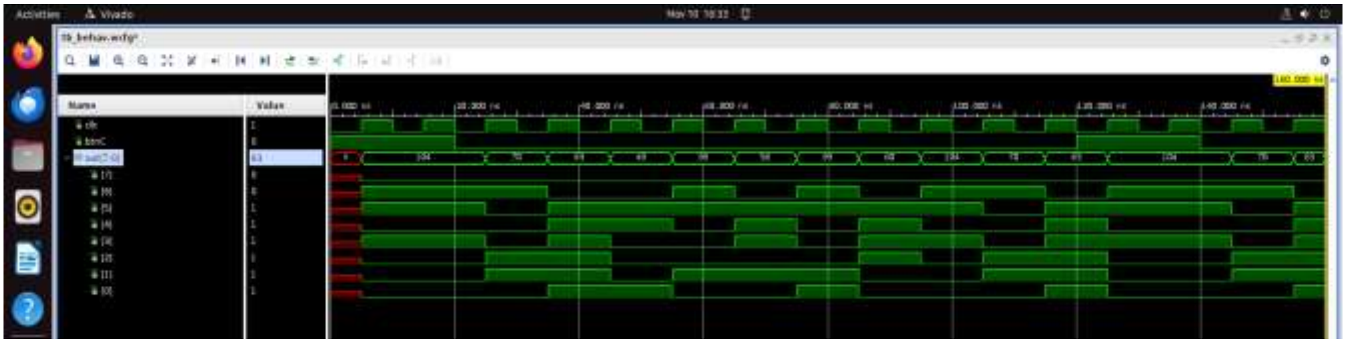
  For y axis:

  If (rival_y > car_y) then there will be collision if (rival_y < car_y + main_car_height) i.e our car collide from front of the rival car.

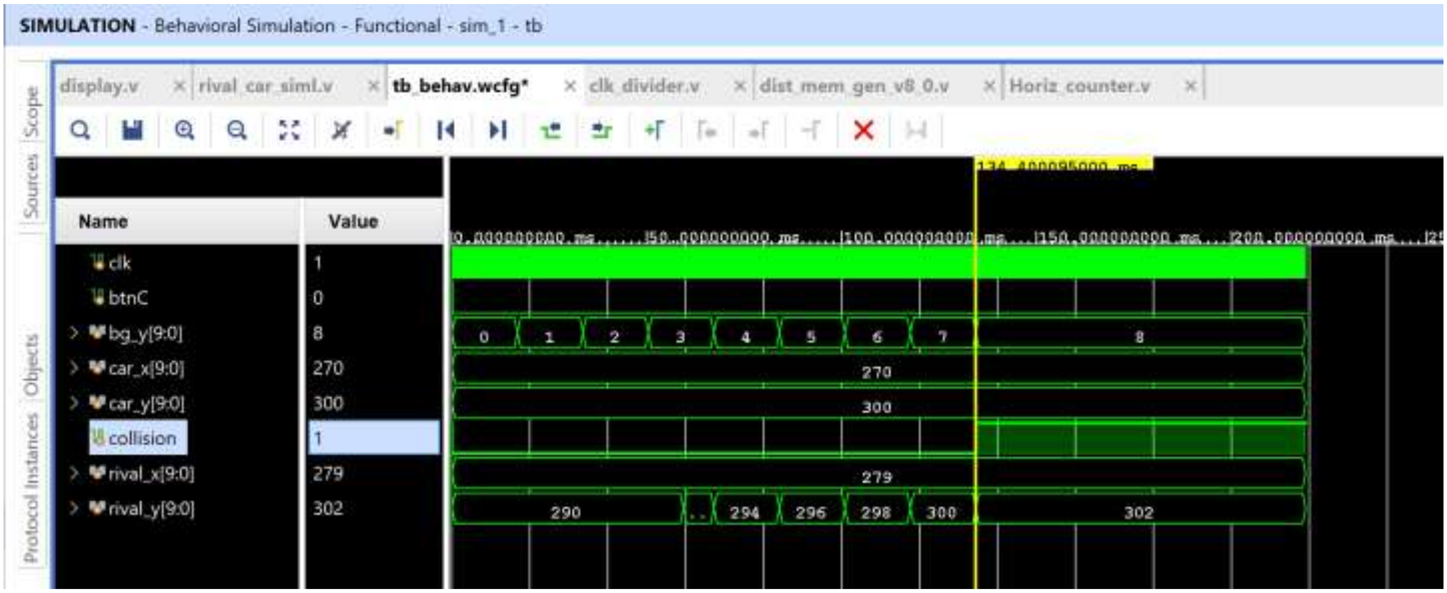
Else if(rival_y > car_y + main_car_height) then there will be a collision if (rival_y < car_y + 2*main_car_height) i.e our car back body collide with rival_car back body.

- **Simulation Snapshots:**

LFSR: Out is the output value of the lfsr module that generate a 8 bit pseudo random number.

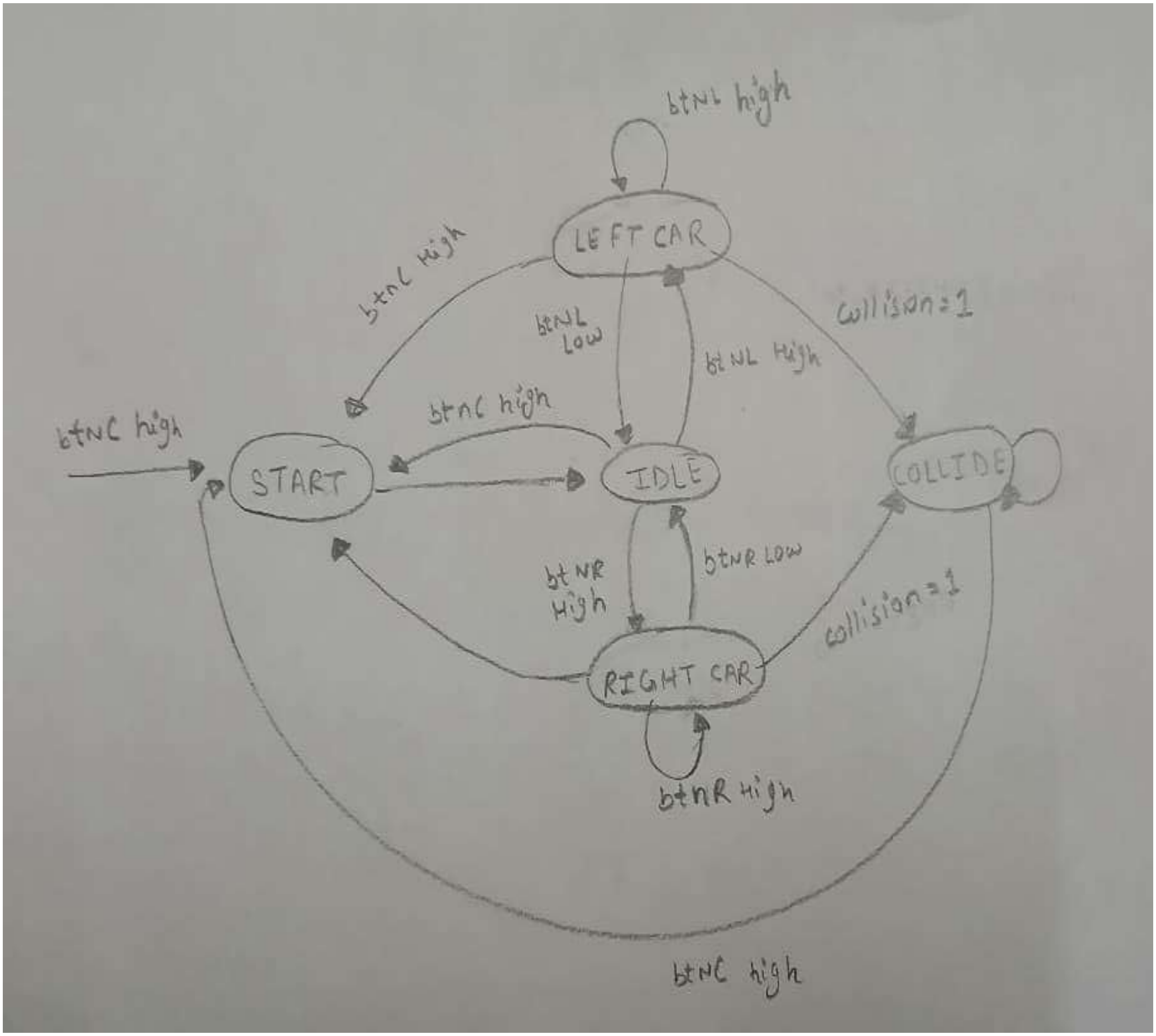


Collision of rival car with our car: We have start the game and not moving the car. The rival car started at 279 in x and forced constant rival y to 290. At rival y = 302 we see a collision between two. Since rival_x is greater than car_x and car_x + main_car_width is greater than rival_x so they will collide in x axis.

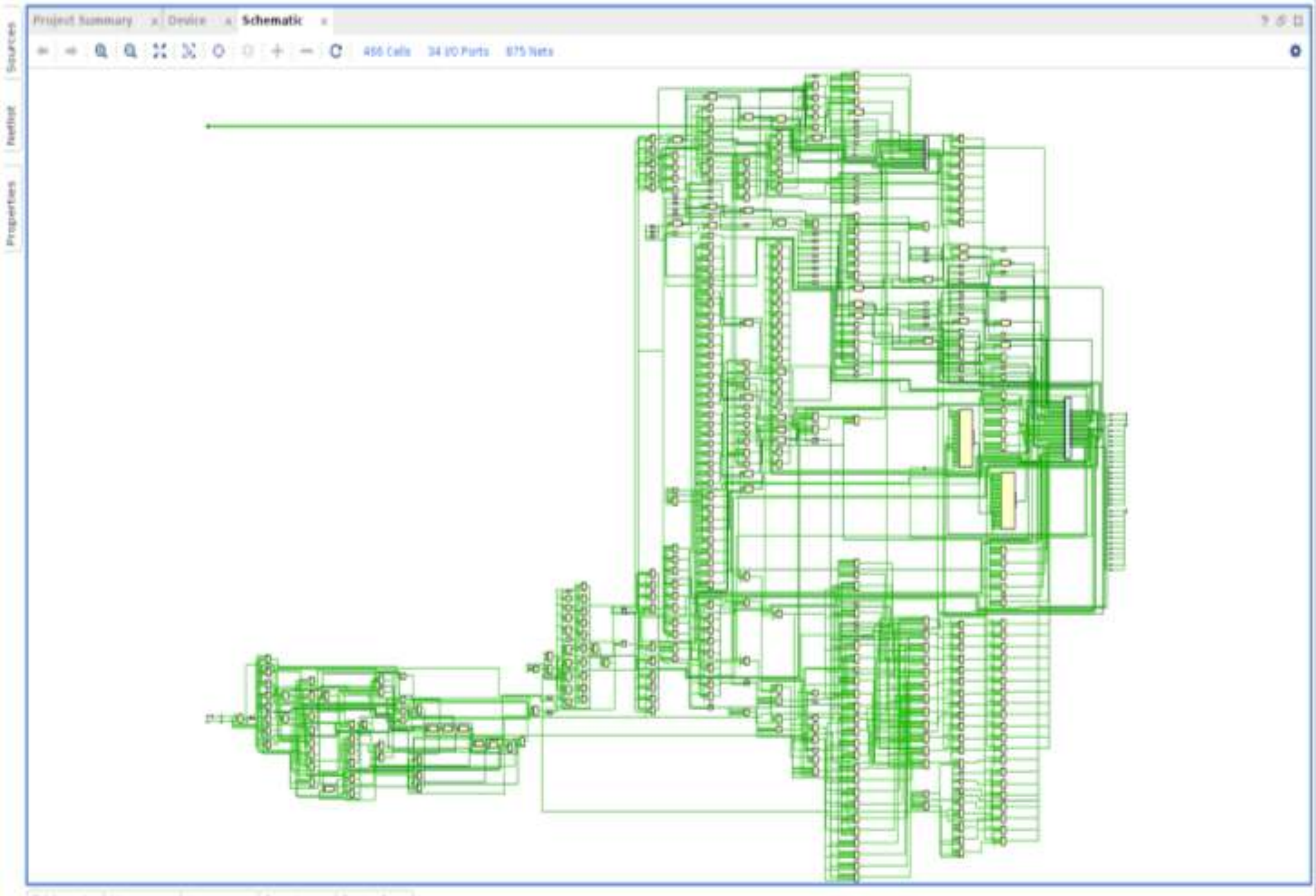


After collision all the parameters became constant. Since bg_y is constant the background stopped moving.

FSM:


btNL high
LEFT CAR
btnc high
btNL LOW
btNL High
collision=1
btnc high
btNC high
START
IDLE
COLLIDE
btNR High
btNR LOW
collision=1
RIGHT CAR
btNR High
btNC high

Generated Schematics:



Synthesis Report:

Resource	Used	Available	Utilization (%)
LUTs	960	20,800	4.55%
Flip-Flops (FFs)	211	41,600	0.51%
BRAMs	0	50	0.00%
DSPs	2	90	0.00%