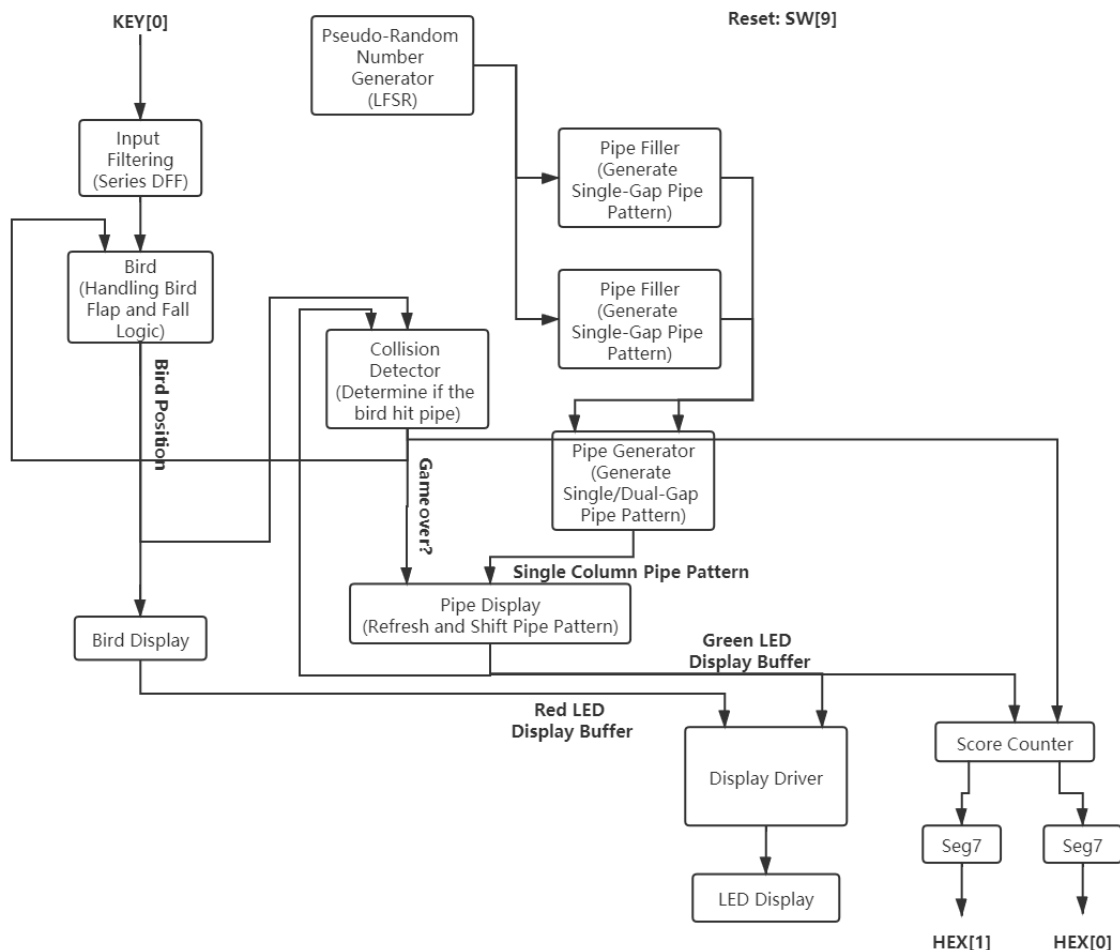


# EE271 Lab 8

1

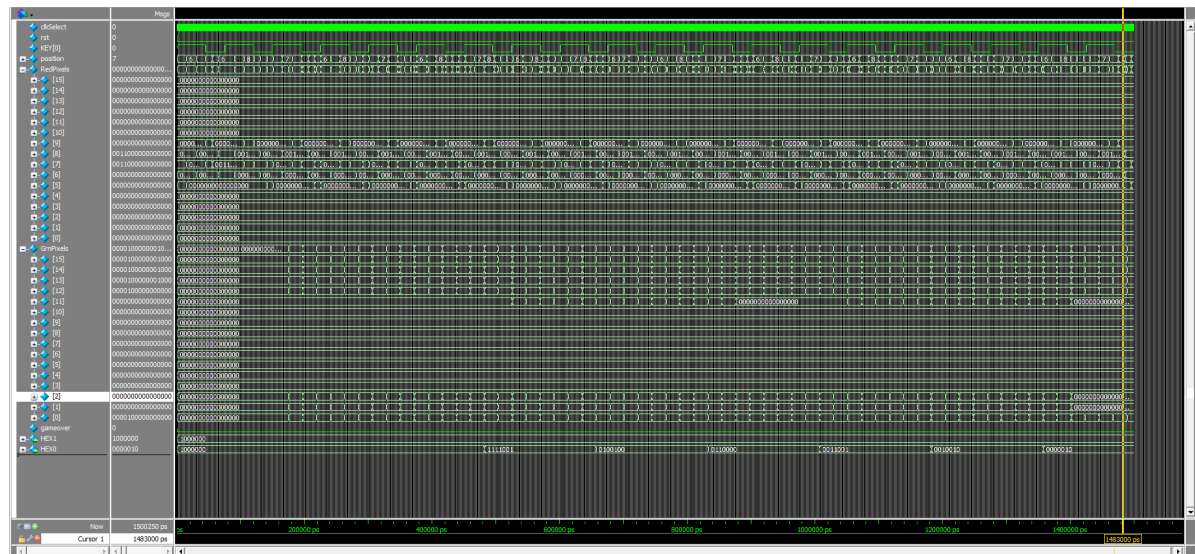
[https://drive.google.com/file/d/15SbrLGMBh2sdtAg6\\_5SGAXaNNVvc\\_gR/view?usp=sharing](https://drive.google.com/file/d/15SbrLGMBh2sdtAg6_5SGAXaNNVvc_gR/view?usp=sharing)

2



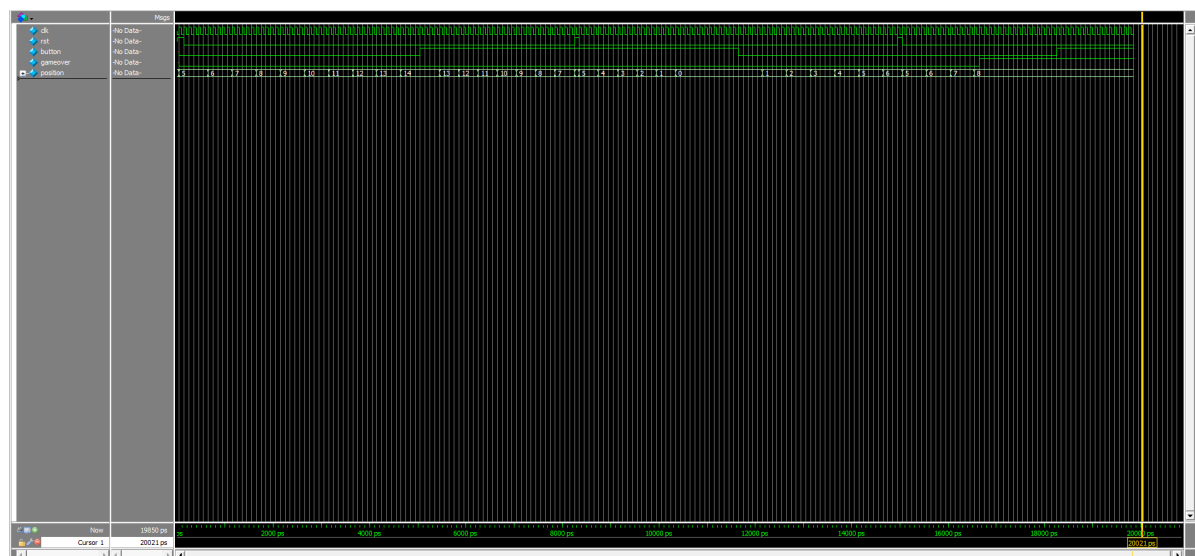
3

To generate random pipe pattern, `pipe_generator` uses an 8-bit LFSR to generate random numbers as input parameter to 2 `pipe_filler` that actually do the job of filling arrays with pipe patterns. `pipe_filler` can generate single-column one-gap pipe patterns whose gap width and position can be specified using two 2-bit number input. The gap is generated using a for loop. In addition, the minimal width and the base position (when offset is 00) of the gap, can be specified by two parameters. The two 4-bit halves of the 8-bit random number will serve as the input of the two `pipe_filler`. The output of two `pipe_filler` will be bitwise-Anded and used as the output column of `pipe_generator`, effectively creating an pseudo-random single/dual gap pipe pattern generator. The number of cycles between each pattern outputted can be adjusted by `SHIFT_THRESHOLD`.



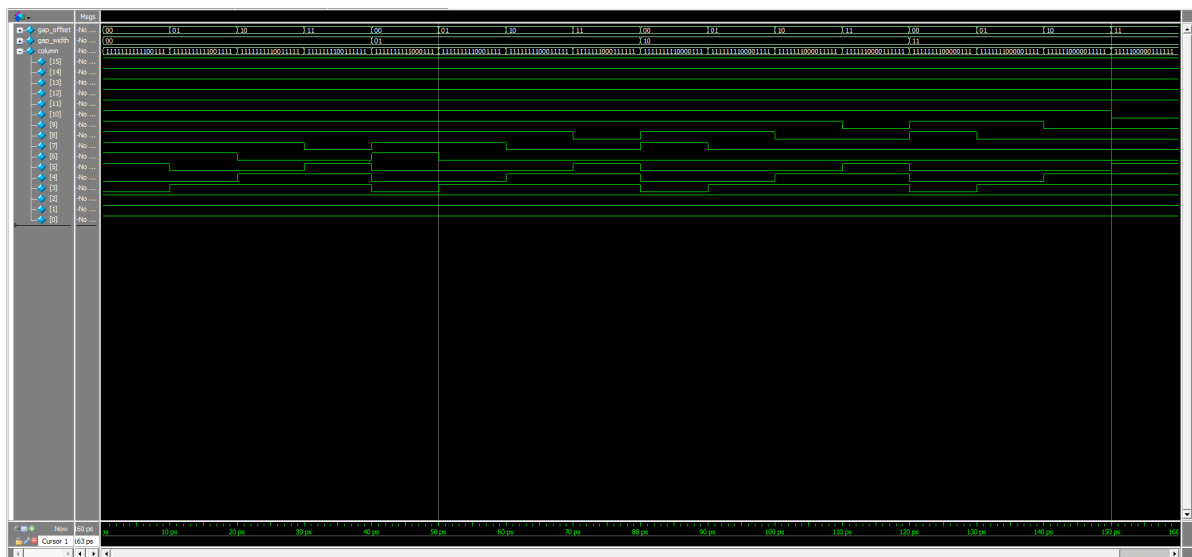
Basically toggling KEY[0] for 20 times so that the bird stays in the middle and scores.

bird.sv:



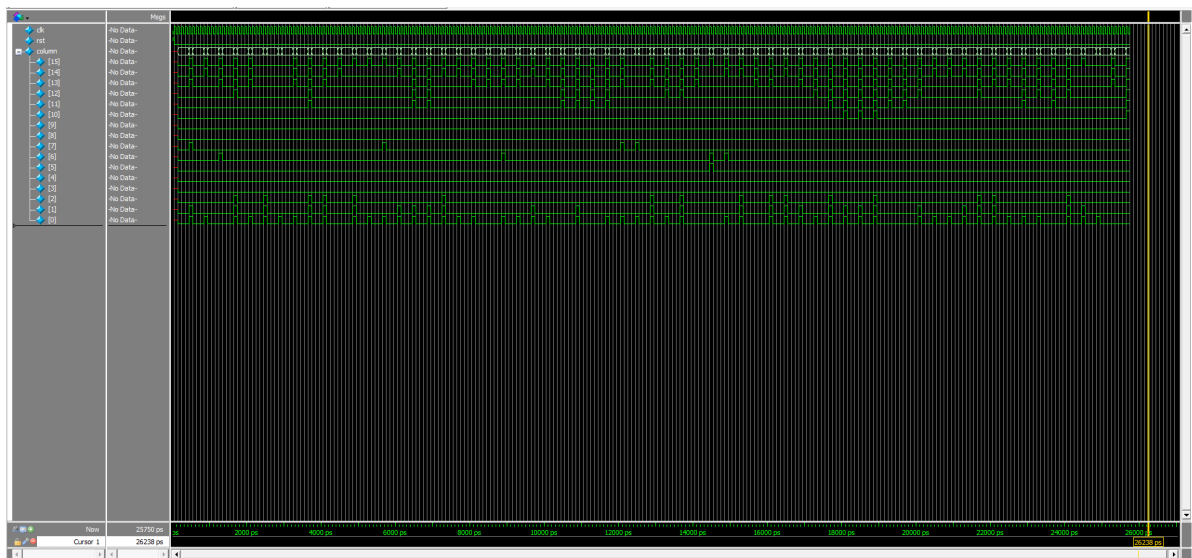
Set button 0, leave the bird to fall to the bottom, then set button to 1, hold, reset, the bird keeps rising to the top. Then set button to 0, the bird falls. At the same time, reset, it keeps falling. Set gameover to 1, the bird stops moving, regardless of the state of the button. `FLY_THRESHOLD` =3, `GRAV_THRESHOLD` =4

pipe\_filler.sv:



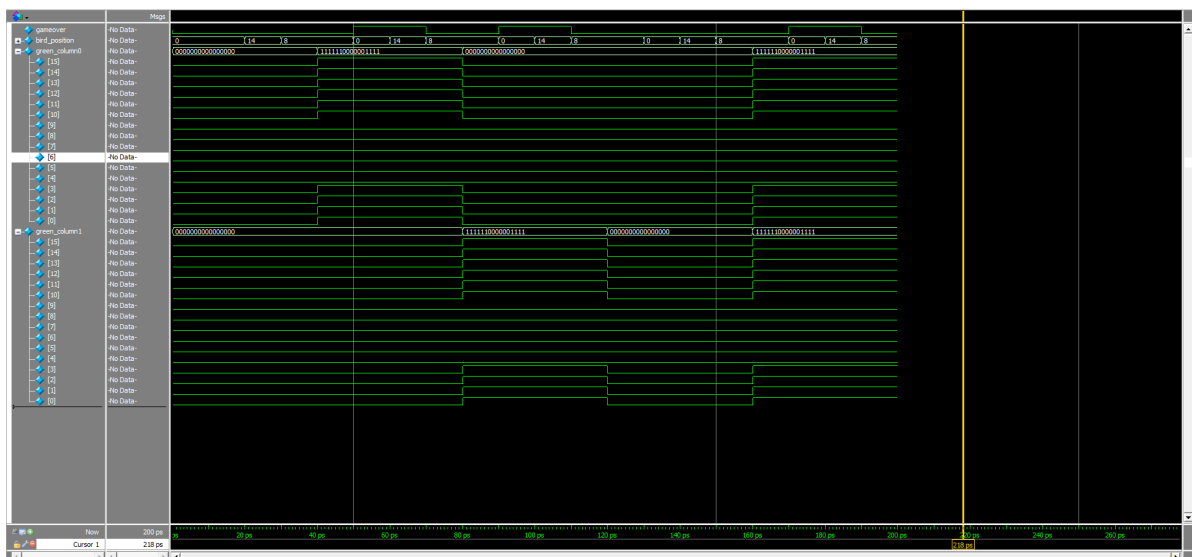
Enumerate all combinations of `gap_width` and `gap_offset` with `GAP_BASE`, or baseline offset, =3 and `GAP_BASE_WIDTH`, or min actual gap width, =2.

`pipe_generator.sv`:



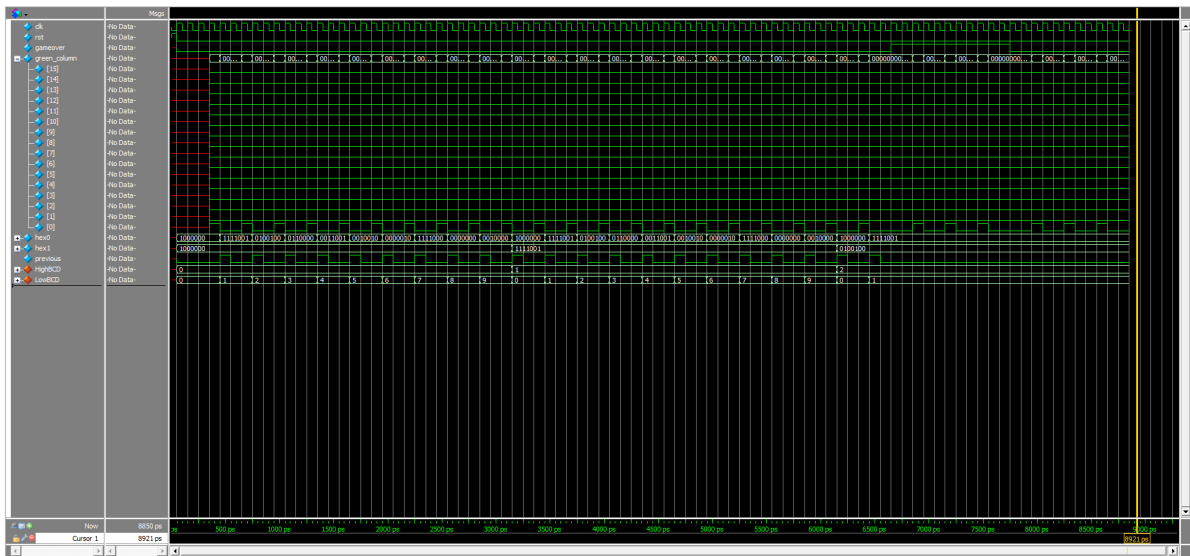
Reset and leave it run for 256 cycles. `SHIFT_THRESHOLD`, or generation interval, =3

`collision_detector.sv`



Set column 0 and 1 to `16'b11111110000001111` or 0, and try different bird position like top, bottom, and middle.

`score_counter.sv`



Continuously input scorable column pattern and try in different gameover conditions: never been 1, 1, post-1

`random.sv`: It's an 8-bit version `random` from last lab

5

12hr