Introduction to Digital Systems Lab4

Student: Li Peng
Student ID:101047123
Due: 2017 Nov 12th
Submit: 2017 Nov 12th

Introduction

During Lab4, we tried to give a logical for input BallX and BallY and it will show on 14(X)×15(Y) LED Matrix. In addition, we also have to make special component to meet special require for Rest or Sticky.

Part1.

Over here, we build BallX to meet it would show on the display. As we know, BallX is the range of 1 to 14, so we need something to let x in the range. We find we can me two comparators, one of it let input x greater than 1 and the other and let input smaller than 14, because we want to in this range we have use an NAND Gate (Do not use AND when the input go out of the range it would never back in) to put them together. Right now, we got on output which is control input, we want it goes in the clock when output equal 0 does addition (shift to right) and when output equal to 1 one does subtraction (shift to left). We need D flip flop, previous output become control in for D flip flop, QB output just connect to Data input to does clock divide by 2 circuit. Right now, we still need an Add to decide to BallX does addition or subtraction. Unfortunately, D flip flop only give a one bite output, we cannot direct connect to our Add (because we are doing 4 bites). Therefore, there must have some between them, and Mux must be an easier way to decide 1 or 0. Well, sad news is we are not allowed use it. So we use 4 bites constant “1” and separate as four one bite by splitter, we also need 4 XOR gate. We ADD/SUB logic over. This would give four one bite output, and we use splitter to transfer to one four bite output. Finally, we can connect to ADD. Obviously, the output is our Next BallX.

Part2.

This part is almost same as BallX, we only have to change the label to Y.

Part3.

First of all, this require our points move at half speed at 4 to 12. Which means we need the stop one time in D flip flop when it applies in the range. Thus, we control enable input, because when enable equal to 0 it would stop, and enable equal to 1 it will keep going. It problem is label of Keep Going do not know either stop or keep going. In other word, we must give a logical for Keep Going. During this time, we have to consider BallX and BallY and same times, so we need

4 comparators make sure X and Y in the range 4 to 12, and using 3 AND gates connect them together which shows they occur at same times. By the way, for the part of stuck is easy, the reason is the light would up if and only if in the range, so just use tunnel of stuck connect output in the range. We got one of output right now, if we want the circuit do a half speed, there must do something like a switch light up at 12, light down at 12, light up at 13 and light down at 13. The number represent if and only if number in the range, else it will do light up at 1, light down at 2 and light up at 3. We already build range part, so to consider whether light up or down it can represent as 1 or 0, which getting close to our D flip flop. As PDF mention, Ticker represents the period which can gives 1 or 0. It could the control input for our D flip flop, and D flip flop the output is our need. Now we put previous output (in the range) and let them happen at same with an NAND, because we want to make sure the circuit still work out of the range (just not stop).

Pat4.

We want pick reset button would back point (1,7) and we only allow to change it in step 3 and step4. Simply, in step3 x equal to 1 when reset, and in step4 y equal to 7 when reset. I would consider x first. Obviously, reset only response 1 or 0. 1 means reset, and 0 means reset not work. it is kind of a simply logical of Mux, reset is control input, one of the data input is regular x input and the other input is we set as x equal to 1, when reset equal to 1 regular x close and constant 1 in. Even if we are not allowed direct use Mux, we can make an Mux. 1 bite Mux is made by 2 AND gates and 1 OR gate, because we consider 4 bits value, we need 4 group of it. (hint: we also use splitter to split 4 bites as four one bit for regular x and constant 1). And now, we have to consider Y, it almost same as X, we only have to change constant 1 of 4 bits to constant 7 of 4 bits, that is it.