IC150: Applied Digital Logic Design

August 17, 2021
10-11:30AM

Second Tierce Examination

Max Marks: 30

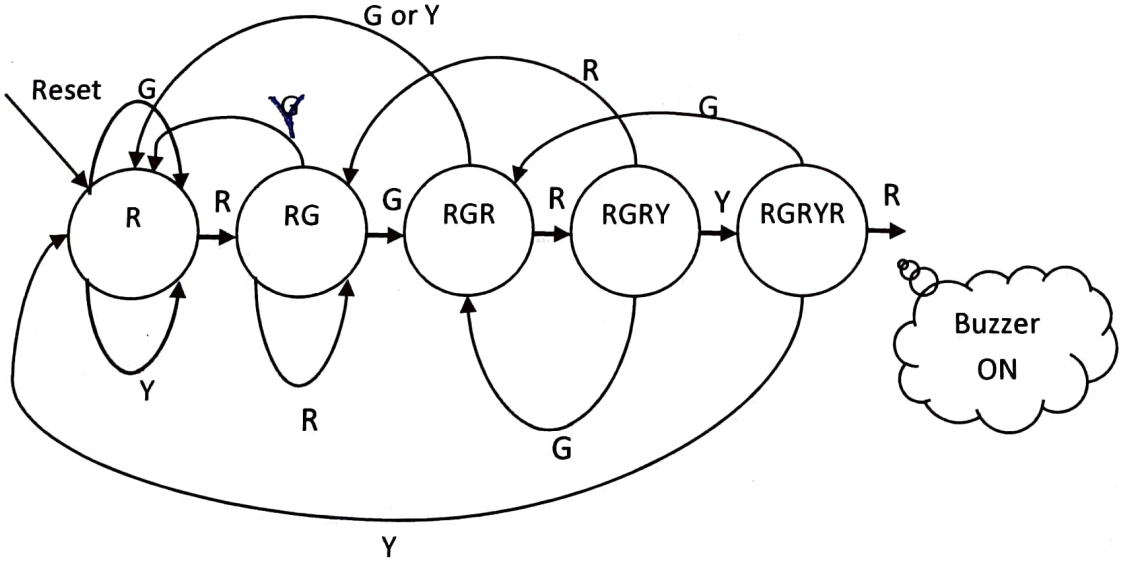
Notes:

1. Answer all questions. The exam is only conducted in the offline mode.
2. Write your name on the question paper as well as on the answer sheets.
3. Provide all intermediate steps of deriving your answers. Merely writing the final answer will not result in any marks to be awarded even if the answer is correct.
4. Mobile phones and calculators are not permitted. A person in possession of these will be treated as using unfair practices in the examination.

Q1 (8 Marks): In this question you are required to implement a game. In this game, there is one player who plays it against a machine implementing your hardware. The player is provided with a bin containing finite number of tennis balls in one of the three colours – Yellow, Red and Green.

The player is required to throw one ball at a time at a specific destination where a sensor is mounted. The sensor gives two signals, the event of ball being thrown is indicated by a positive edge of a signal on a wire called **`BallPresent`**. The game console seeks a sequence of Red – Green – Red – Yellow – Red. Each time this sequence is observed, a buzzer is turned on which is indicated by a signal **`Buzz`** being set to 1.

In order to detect this sequence, the following state machine is used where R state is used for "looking for Red Ball", RG state is used for "Red Ball found and looking for Green Ball" and so on. A transition is shown by the colour of ball found. For example, if RGRY state, if a Yellow ball is seen, state is transited to RGRYR – or "Red, Green, Red, Yellow balls found (in that sequence) and looking for another Red".



You need to implement the game in the following Verilog module. Use case statement to implement the state machine.

```
parameter R = 2'b00;
parameter G = 2'b01;
parameter Y = 2'b10;
module gamer (input BallPresent, input Reset,
              input [1:0] ballColor, output Buzz);
......
endmodule
```

Q2 (8, 6 marks): (a) Use the algorithm discussed in the class to find the square root of 11b'001_1101_1100. (note that an _ character is used in this which is permitted by Verilog to improve the readability.)

(b) Given a module D Latch, make a D Flip Flop. The two module interfaces are given below.

```
module DLatch (input D, input LE, output Q, output Qbar);
  always @(D, LE) begin
    if (LE == 1b'1) begin Q <=D; Qbar <= ~D end
  end
endmodule
module DFF (input D, input clk, output Q, output Qbar);
......
endmodule
```

Q3 (8 marks): You are given a **bufz** module with definition as follows.

```
module bufz (input A, input EN, output Y);
    assign Y = (EN == 1'b1)? A: 1'bz;
endmodule
```

Use this module to make a register with two *n*-bit outputs ($n = 8$) – outputs being named A and B, one *n*-bit input D and one 1-bit input LE. Additionally, it takes two one-bit signals ENA and ENB which cause the output of the register to appear on output A and output B respectively if the corresponding EN_ signals are given as 1. The outputs remain in high impedance mode if the corresponding EN_ signal is 0.

```
module reg8 (input [7:0] D, input LE, input ENA, input ENB,
             output [7:0] A, output [7:0] B);
.....
endmodule
```