

14:332:437 Digital Systems Design

Fall 2007

Problem Set 5

Electrical and Computer Engineering Department

School of Engineering

Rutgers University

Prof. Michael L. Bushnell

Assigned: October 24, 2007

Due: November 2, 2007

STUDENTS ARE EXPECTED TO WORK INDEPENDENTLY ON BOTH HOMEWORK AND EXAMINATIONS. NO COLLABORATION IS PERMITTED. YOUR WORK MUST BE YOUR OWN.

1. Please design a fault-tolerant electronic pool table that will set time limits for each player's shot, and also keep track of the score. The rules of this pool game are that one player takes solid balls, and the other takes stripe balls. Also, once a player gets all of their balls into pockets in the pool table, they can then sink the 8 ball in a pocket to win the game. However, if a player sinks the 8 ball before all of his other balls, he/she immediately loses the game. Also, if a player sinks the queue ball, then one point is taken off his/her score, and one of the balls that he previously got in is put back on the table. In addition, in this game, each player is allowed only 25 seconds to make his/her shot. Your machine must restart a 25 second counter at the start of a person's turn, and sound the buzzer when 25 seconds have elapsed, at which point the other player's turn starts. Sound the buzzer for only 10 seconds. All of the balls have RFID tags built in, so the pool table can sense what type of ball has been sunk into a pocket.

Signal	Type	Meaning
<i>/reset</i>	input	Active-low signal that initializes the pool table
<i>clk</i>	input	System clock – positive edge triggered, operates at 60 Hz
<i>eightballin</i>	input	If 1, then the 8 ball just went into a pocket
<i>solidballin</i>	input	If 1, the a solid ball just went into a pocket (not including the 8 ball)
<i>stripeballin</i>	input	If 1, the a stripe ball just went into a pocket
<i>queueballin</i>	input	If 1, then the queue ball just went into a pocket
<i>player1score</i>	output	Ranges from 0 to 8 (it cannot go negative)
<i>player2score</i>	output	Ranges from 0 to 8 (it cannot go negative)
<i>soundbuzzer</i>	output	If 1, a buzzer sounds

You should make this pool table fault-tolerant, using triple modular redundancy. For all output signals, you need a voter circuit, which latches its three input signals. However,

assume that the voter will be built out of special logic gates that are fault tolerant, so you do NOT need to triplicate the voters.