

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.111 - Digital Systems Laboratory
Problem Set #4

Issued: Monday, March 5, 2001

Due: Wednesday, March 14, 2001 (IN CLASS)

Problem 4.1 - A Little Bit of Vegas

Having grown bored with 6.111's more mundane applications (such as stoplight controllers and diffusing bombs) you decide you would like to use your newfound knowledge for something a little more exciting and, perhaps, profitable. You've seen video poker machines before, and you like the idea, but poker was never your game. So you decide to build a variant that you do like: craps. You get your friends from 6.001, Ben and Alyssa, to help you. However, as a condition for his help Ben insists on using 4-sided dice, as an homage to the many D&D games he played during his formative years.

Craps is usually played with standard 6-sided dice, so this requirement means you'll have to change the rules of the game slightly. Craps with 4-sided dice is played as follows:

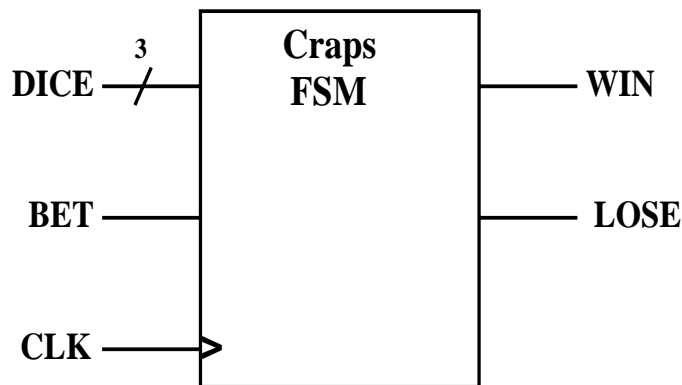
- The dice-roller, or "shooter", places a bet (known as a "pass" bet) and rolls two dice
- If the roll is a 5 or 7, he wins immediately
- If the roll is a 2 or 8, he loses immediately
- If he rolls a 3, 4, or 6, that number becomes his "point", and he rolls again:
 - If he rolls his point number, he wins
 - If he rolls 5, he loses
 - If he rolls any other number, he rolls again, until his point or 5 comes up

E.g. a set of rolls that is (4, 2, 8, 4) is a win, whereas (3, 7, 8, 4, 4, 5) is a loss.

To make things slightly more complicated, this machine will allow **one** "come" bet, in addition to the "pass" bet. A come bet is placed anytime after a shooter has established a point. For the come bet only, the next roll of the dice is treated as the first roll of the rule sequence above. A come bet can result in an instant win (5 or 7), instant loss (2 or 8), or establishment of a different point for the come bet (3, 4, or 6). The win/loss rules for a come bet are exactly the same as for a pass bet. Once a come point has been established, it stays until won or lost, regardless of what happens to the pass bet. Some examples:

- 4, <come bet placed> 6, 3, 6 <come bet won>, 5 <pass bet lost>
- 3, 4, <come bet placed> 5 <come bet won> <pass bet lost>
- 3, <come bet placed> 4, 5 <both bets lost>
- 6, 7, 8 <come bet placed> 8 <come bet lost>, 6 <pass bet won>
- 6, 4, <come bet placed> 6 <pass bet won>, 5 <come bet lost>

You, Ben, & Alyssa divvy up the work; your responsibility will be to build the FSM to determine if bets are won or lost. It looks like this:



The **DICE** input encodes the dice roll:

“000” = No Roll	“010” = 3 Rolled	“100” = 5 Rolled	“110” = 7 Rolled
“001” = 2 Rolled	“011” = 4 Rolled	“101” = 6 Rolled	“111” = 8 Rolled

The **BET** input equals ‘1’ when a bet is placed. The input is held high through the next valid roll, so you do not need to worry about latching it (you can wait until **DICE** != “000” to worry about it). This bet is a pass bet when there is no pass point, and a come bet otherwise. You do **not** need to worry about multiple come bets - only one is allowed.

WIN should be set to ‘1’ when either a pass or come bet is won.

LOSE should be set to ‘1’ when wither a pass bet or come bet is lost. In the case where both are lost simultaneously, **LOSE** should be held high for 2 clock cycles to indicate 2 losses.

WIN and **LOSE** should be asserted simultaneously if there is a win and a loss after a roll.

After a win or loss, your FSM should be ready to accept new bets, while continuing to play any pass or come bets still going.

Implement this FSM and compile it to a 374I. Submit:

1. State Diagram
2. Well-commented VHDL code
3. Pinout section of report file