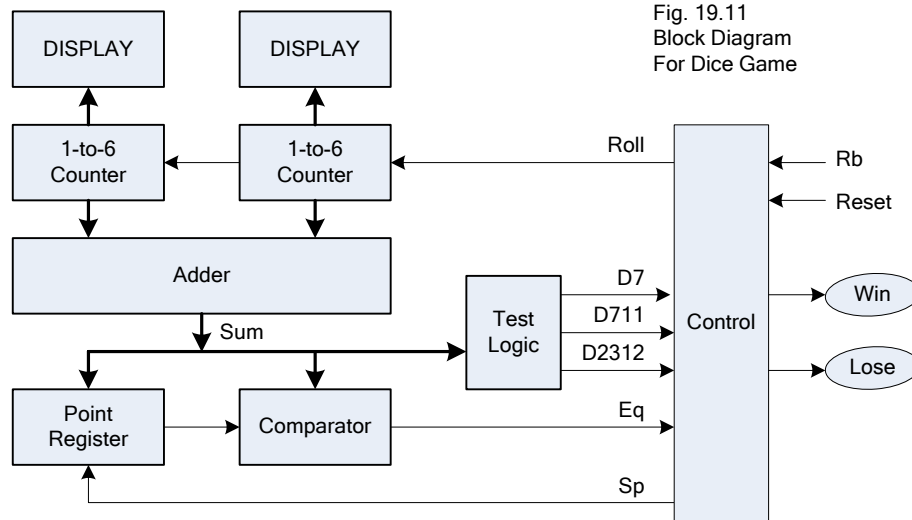


Final Project

Grading will be based on:

- Correctness of the design [20 %]
- Effectiveness of testing [20 %]
- Coding style [20%]
- Project Demonstration [10%]
- Project Report [30%]

Design, test and implement on a Xilinx SPARTAN 3E FPGA board a digital system for a dice game. Fig. 19.11 shows the block diagram for the dice game. The block diagram omits to show the Clk signal and its distribution.



The counters are used to simulate the roll of the dice. Each counter counts in the sequence 1,2,3,4,5,6,1,2,3,.....
Thus, after the "roll" of the dice, the sum of the values in the two counters will be in the range 2 through 12.

The rules of the game are as follows:

1. After the first roll of the dice, the player wins if the sum is 7 or 11. He loses if the sum is 2, 3, or 12. Otherwise, the sum which he obtained on the first roll is referred to as his *point*, and he must roll the dice again.
2. On the second or subsequent roll of the dice, he wins if the sum equals his point, and he loses if the sum is 7. Otherwise, he must roll again until he finally wins or loses.

The inputs to the dice game come from two push buttons, *Rb* (roll button) and *Reset*. *Reset* is used to initiate a new game. When the roll button is pushed, the dice counters count at a high speed, so the values cannot be read on the display. When the roll button is released, the values in the two counters are displayed and the game can proceed. Because the button is released at a random time, this simulates a random roll of the dice. If the *Win* light or *Lose* light is not on, the player must push the roll button again. We will assume that the push buttons are properly debounced and that the changes in *Rb* are properly synchronized with the clock. Methods for debouncing and synchronization were discussed previously.

Figure 19-12 shows a flowchart for the dice game. After rolling the dice, the sum is tested. If it is 7 or 11, the player wins; if it is 2, 3, or 12, he loses. Otherwise, the sum is saved in the point register, and the player rolls again. If the new sum equals the point, he wins; if it is 7, he loses. Otherwise, he rolls again. After winning or losing, he must push *Reset* to begin a new game.

The components for the dice game shown in the block diagram (Figure 19-11) include an adder which adds the two counter outputs, a register to store the point, test logic to determine conditions for win or lose, and a control circuit. The input signals to the control circuit are defined as follows:

$D_7 = 1$ if the sum of the dice is 7

$D_{711} = 1$ if the sum of the dice is 7 or 11

$D_{2312} = 1$ if the sum of the dice is 2, 3, or 12

$Eq = 1$ if the sum of the dice equals the number stored in the point register

$Rb = 1$ when the roll button is pressed

$Reset = 1$ when the reset button is pressed

The outputs from the control circuit are defined as follows:

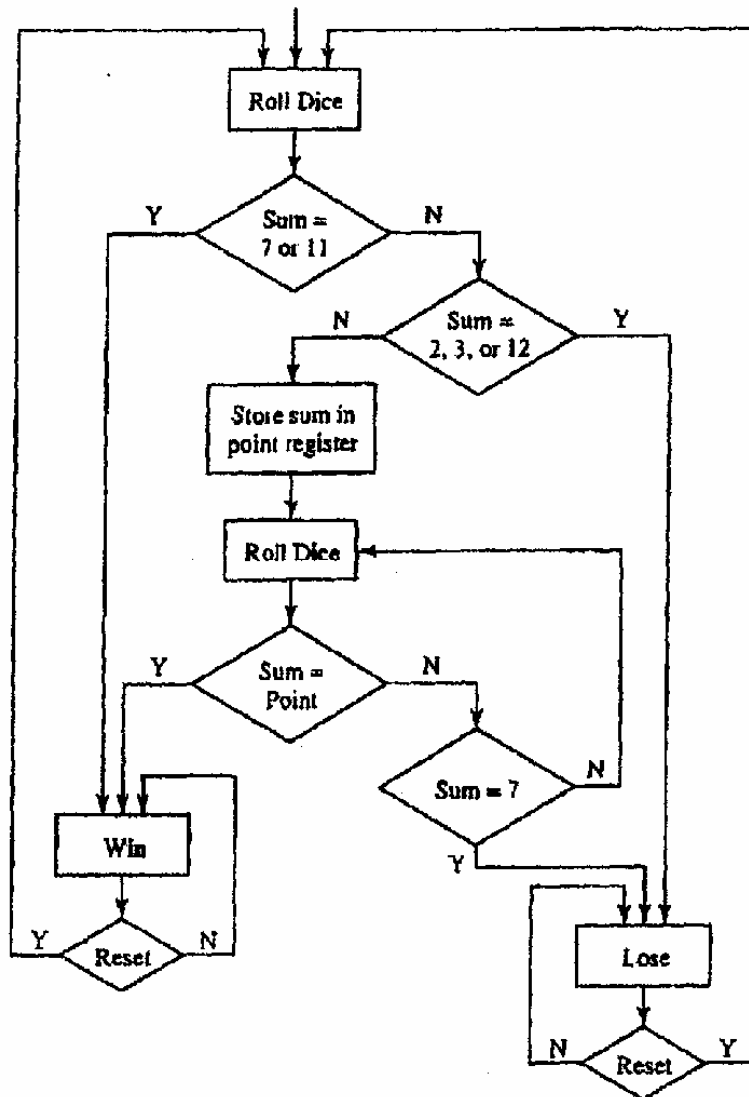
$Roll = 1$ enables the dice counters

$Sp = 1$ causes the sum to be stored in the point register

$Win = 1$ turns on the win light

$Lose = 1$ turns on the lose light

FIGURE 19-12
Flowchart for
Dice Game



We can now convert the flowchart for the dice game to an SM chart for the control circuit using the defined control signals. Figure 19-13 shows the resulting SM chart. The control circuit waits in state S_0 until the roll button is pressed ($Rb = 1$). Then, it goes to state S_1 , and the roll counters are enabled as long as $Rb = 1$. As soon as the roll button is released ($Rb = 0$), D_{711} is tested. If the sum is 7 or 11, the circuit goes to state S_2 and turns on the Win light; otherwise, D_{2312} is tested. If the sum is 2, 3, or 12, it goes to state S_3 and turns on the Lose light; otherwise, the signal Sp becomes 1, and the sum is stored in the point register. It then enters S_4 and waits for the player to "roll the dice" again. In S_5 , after the roll button is released, if $Eq = 1$, the sum equals the point and state S_2 is entered to indicate a win. If $D_7 = 1$, the sum is 7 and S_3 is entered to indicate a loss. Otherwise, the control returns to S_4 so that the player can roll again. When in S_2 or S_3 , the game is reset to S_0 when the Reset button is pressed.

