Lab 9 – Craps Dice Game Controller

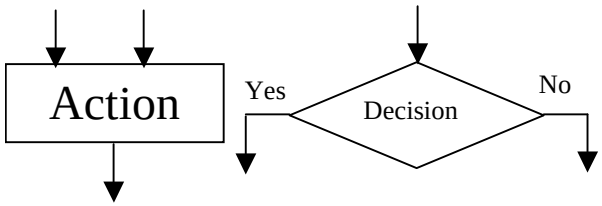
The rules of the games called craps are:

> *After rolling the dice (two six-sided die), the sum is tested. If the sum is 7 or 11, the player wins; if it is 2, 3, or 12, the player loses. Otherwise, the sum is either 4, 5, 6, 8, 9, or 10 and is saved as the point and the player must roll again. If the new sum equals the point, the player wins; if it is 7, the player loses. Otherwise, the player must roll again. Play continues until the player wins or loses.*

Review your design for the dataflow portion of the design. In the dataflow portion we answered what must be done. For the controller portion we must answer in what sequence and when must things be done. When the controller portion is successfully married to the dataflow portion the design is complete.

ASM Chart of Craps Dice game

Now we will develop a Algorithmic State Machine chart by completing Figure 1. We will use the following notation:



The actions you must consider are – Roll dice (twice), Store sum in point register, Win and Lose. The decision you might consider are – Sum=7 or 11, Sum=2, 3 or 12, Sum=7, Sum=Point, and Reset. To help you with your flowchart the first step is partially completed along with two of the terminus steps in Figure 1. You may have multiple decision blocks between actions.

The ASM chart enables us to clarify the sequence of events and what decisions must be made. Next we will translate this flowchart in to a state diagram for a Moore type sequential finite state machine. That means that the outputs are associated with a state or bubble and only conditions are associated with the transition arrows.

State Diagram of Craps Dice Game

Construct a state diagram by completing Figure 2. The Moore style design will contain eight states (S0 to S7). Consult the dataflow block diagram and we see that Roll and Sp are inputs to the dataflow logic. Therefore, these must be outputs for the controller. The state of Win and Lose must be signaled with output signals.

The dataflow logic produces the signals D7, D711, D2312, and Eq. These are inputs to the controller. Two additional inputs are Rb for Roll button and Reset. Therefore transition arrows may contain the symbols D7, D711, D2312, Eq, Rb and Reset.

Craps Dice Game Controller VHDL

The next step in the process is to convert the Moore type state diagram into a VHDL module. There will be an ENTITY with the required inputs and outputs. Note the state signal State must be BUFFER mode. The other signals are IN mode or OUT mode.

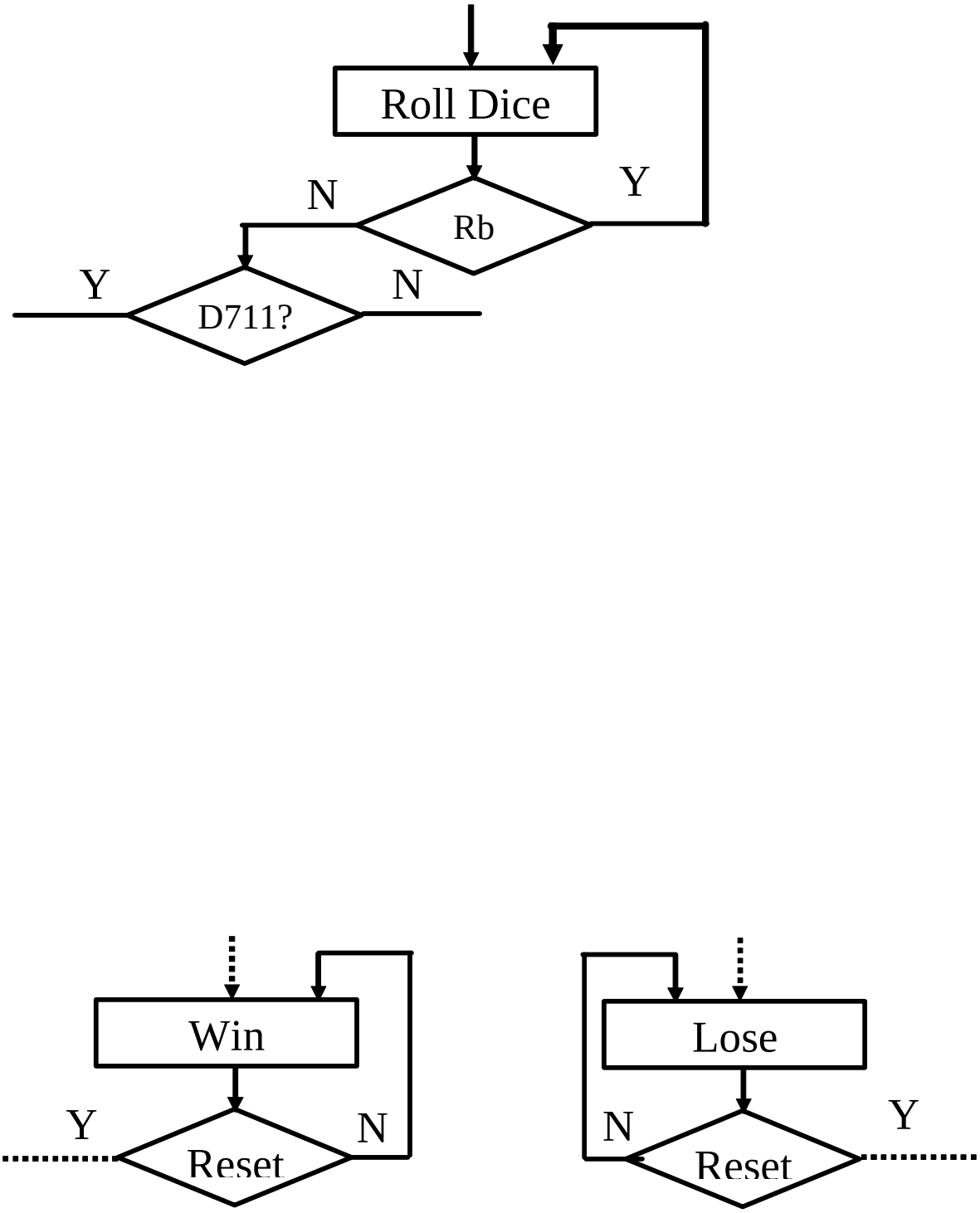


Figure 1 Algorithmic State Machine Chart of Craps Dice Game

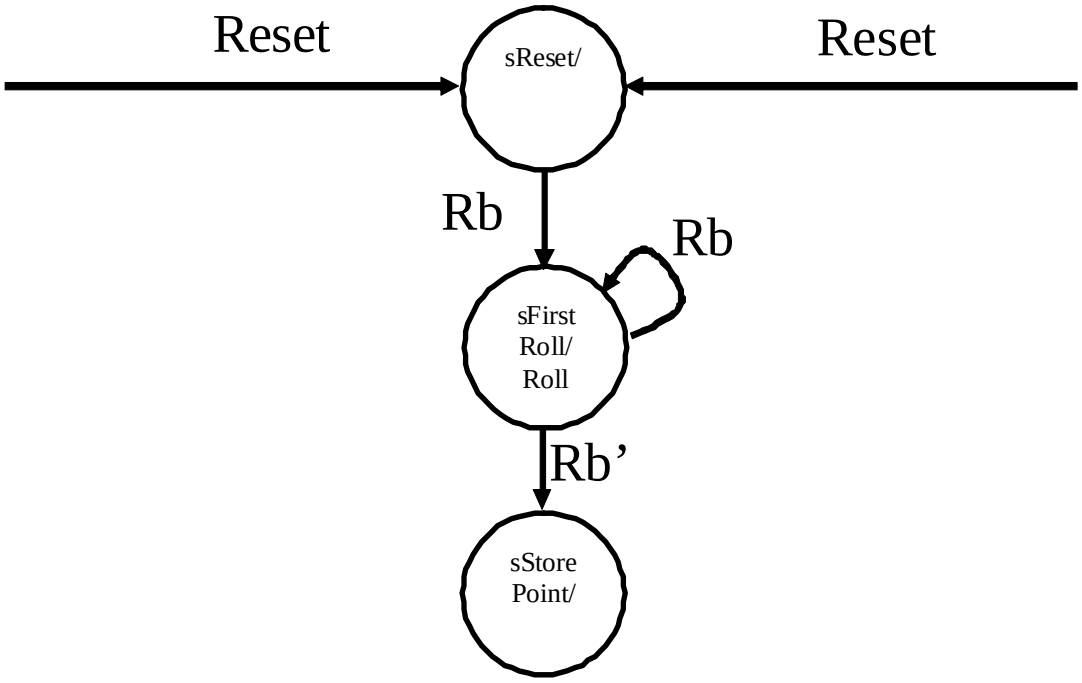


Figure 2 Craps Dice Game Controller State Diagram

```
ENTITY Controller IS
   PORT
   (
         Rb,Reset,Clk            : IN BIT;
         D7,D711,D2312,Eq  : IN BIT;
         Roll,Sp                       : OUT BIT;
         Win,Lose                : OUT BIT;
         State                   : BUFFER BIT_VECTOR(2 DOWNTO 0));
END Controller;
```

The architecture will have three blocks within it. One for deciding on the next state, one for producing the outputs based on the state, and one that represents the memory of the finite state machine. NextState will be a signal that is internal to the controller and is thus declared within the ARCHITECTURE statement.

```
ARCHITECTURE Behavioral OF Controller IS
TYPE STATE_TYPE IS  (sReset,sFirstRoll,sStorePoint, sPlayerWins,
            sPlayerLoses,sWaitNextRoll,sNextRoll,sEndNextRoll);
SIGNAL State : STATE_TYPE;
BEGIN

--Place behavioral description here

END Behavioral;
```

Use a CASE statement to move the controller to the next state. As an example when the Rb is pressed the controller moves to the next state, otherwise it remain in the “000” state.

```
CASE State IS
      WHEN sReset =>                      --Reset State
            IF Rb = '1' THEN
                  state <= sFirstRoll
            ELSE
                  state <= sReset
            END IF;

-- place the rest of the cases here

END CASE;
```

The Moore outputs may be specified as dataflow type equations.

```
--output combinational logic
Sp <= ‘1’  WHEN State = sStorePoint Else ‘0’;
Roll <= ;--insert function here
Win <= ;--insert function here
Lose <= ;--insert function here
```

Perform analysis and synthesis until all errors and warning are removed. Then make assignments that are consistent with those for the dataflow part of the design performed in the previous laboratory exercise. Perform a full compile and simulation of the controller. After you have inspected the simulations and are convinced that it is correct, program the EPM7182SLC84-15 on the PLDT-2. Exercise the control by manipulation of switches and observe the LEDs.

Second Project

Using the dataflow project as a beginning, add the controller to the schematic. Interconnect the dataflow and controller. Analyses and Synthesize, assign pins, full compile, and program the device. No synthesis

is required. Verify that the hardware is working properly by playing and logging each game. Capture the following information in the log:

Roll Number	Point	Sum	Win/Lose	Note

After determining that the game is working properly, demonstrate your completed to design to the laboratory assistant or laboratory instructor. If they determine that it is working properly, they will provide their signature.

_____ _____
Signature Date

Report Format

This laboratory exercise requires a formal report. See report format handout for detail on the content of a formal reports.

This page intentionally left blank.

Lab 9 – Craps Dice Game Controller

Group Member # 1_____
Please Print

Group Member #2 _____
Please Print

Group Member #3_____
Please Print

Lab Period: _____
Monday ... Friday

Lab Station: _____
1 through 12

Check List:

√	Description	Score
	Purpose	
	Discussion to include at a minimum the following:	
	Flowchart	
	Moore State Diagram	
	Controller VHDL	
	Controller Simulation	
	All Dataflow VHDL and BDF	
	Dataflow with Controller BDF	
	Log of 10 Games using Dataflow with Controller	
	Signature Sheet	
	Conclusion	
	Raw data, if any	
	Laboratory procedure handout	
	Total	

The work presented is the sole work of the stated group members. Representing the work of others as your own is plagiarism and is punishable by failure of the course for the slightest infraction.