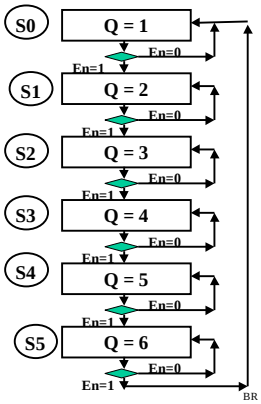
Dice Game (Chapter 22)

- **The dice game in Chapter 22 is a good example of a Finite State Machine controlling a Datapath.**
  - **The combined FSM/Datapath implement a Dice Game.**
- **Dice Game Rules:**
  - **After 1st roll of dice, player wins if roll if 7 or 11. Player loses if roll is 2, 3 or 12. Anything else is saved as the players "POINT".**
  - **On subsequent rolls, player loses if roll is a 7. Player wins if the dice roll is equal to their "POINT". You must keep rolling until you win or lose.**

What do we need: A Dice Roll

- **Some way to simulate a dice roll. Dice have values of 1,2,3,4,5,6.**
- **We could generate a "random" number between 1-6.**
  - **Could use a psuedo random sequence generator in the form of a shift register for this.**
  - **Can be difficult to test if using psuedo random sequences**
- **An easier way is to use a counter whose count sequence is 1,2,3,4,5,6,1,2, etc.**
  - **Even though count sequence is NON-RANDOM, a high speed clock and a user pushing buttons to stop/start the roll will make the dice roll look random!!!**

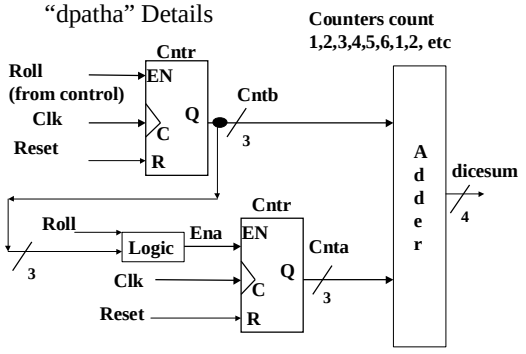


1-6 Counter

Need two counters, one for each dice.

Need an ADDER to sum the two 3-bit dice outputs to produce a four bit sum that we can test!

"dpatha" Details

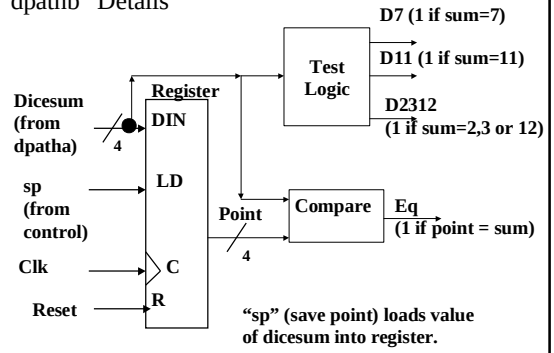


Ena = 1 when Cntb = 6 and Roll = 1

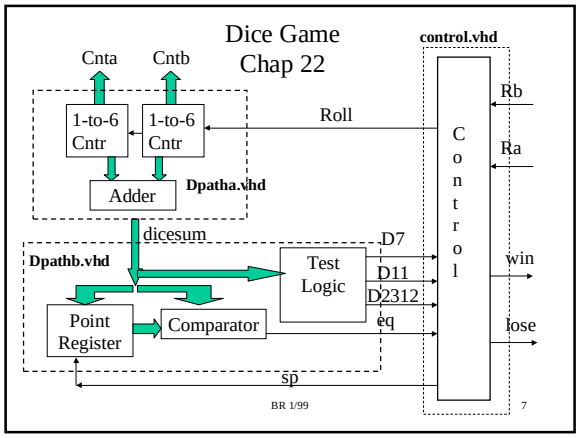
What else do we need???

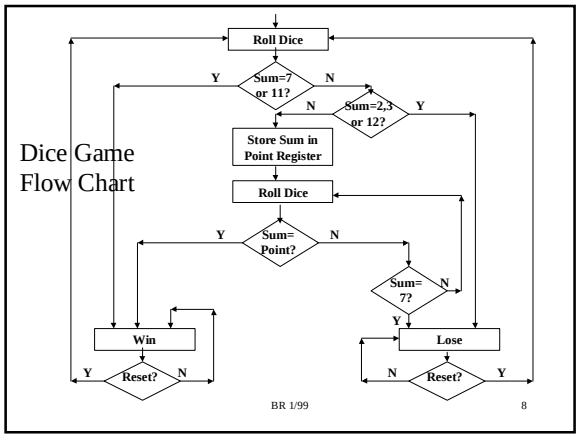
- We need to save the "POINT" value from the first roll somewhere.
  - Use a Register!
- We need to compare our POINT value and the current dice roll to several values
  - Compare Current dice roll to POINT value: Use a comparator
  - Compare Current dice roll to values of 7, 11, 2, 3, 12: Use comparator logic
- Need inputs to start the dice roll, stop the dice roll.

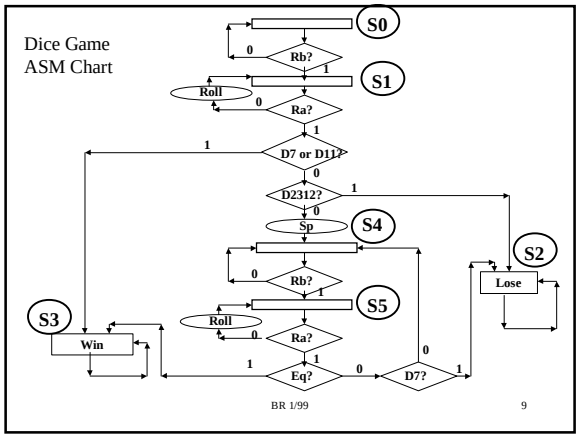
"dpathb" Details



"sp" (save point) loads value of dicesum into register.


Dice Game
Chap 22
control.vhd
Cnta
Cntb
1-to-6
Cntr
1-to-6
Cntr
Roll
Rb
Ra
Adder
Dpatha.vhd
dicesum
D7
Dpathb.vhd
Test
Logic
win
D11
D2312
Point
Register
Comparator
eq
lose
sp
BR 1/99
7


Dice Game
Flow Chart
Roll Dice
Sum=7
or 11?
Sum=2,3
or 12?
Store Sum in
Point Register
Roll Dice
Sum=
Point?
Sum=
7?
Win
Lose
Reset?
Reset?
BR 1/99
8

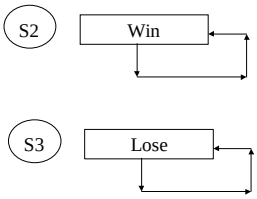

Dice Game
ASM Chart
S0
Rb?
S1
Roll
Ra?
D7 or D11?
D2312?
Sp
S4
S2
Lose
Rb?
S5
S3
Win
Roll
Ra?
Eq?
D7?
BR 1/99
9

Dice Game Implementation

- Dice game implemented in three 22V10 PLDs
  - control.vhd (Finite State Machine)
  - dpatha.vhd (two 1-6 counters, adder)
  - dpathb.vhd (point register, comparator, test logic)
- The Control FSM is implemented using one-hot encoding.
  - Outputs q0, q1, q5 are states S0, S1, S5. Output "win" is state S2, output "lose" is state S3. Outputs q0,q1,q5 are available just for debugging purposes.

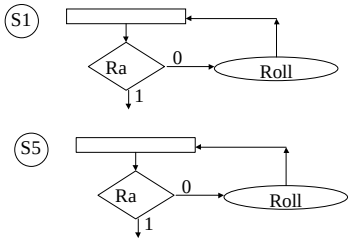
Finite State Machine Changes

Asynchronous Reset now use to exit states S2, S3. Resets back to State S0.



Finite State Machine Changes (cont).

Because we don't have debounced switches, added extra "roll" input called "Ra". Changed States S1, S5 to use Ra, not Rb.



This means that to roll the dice, you use flip switch Rb up, then down. This starts the dice rolling. To stop the dice, Flip Ra switch up then down.