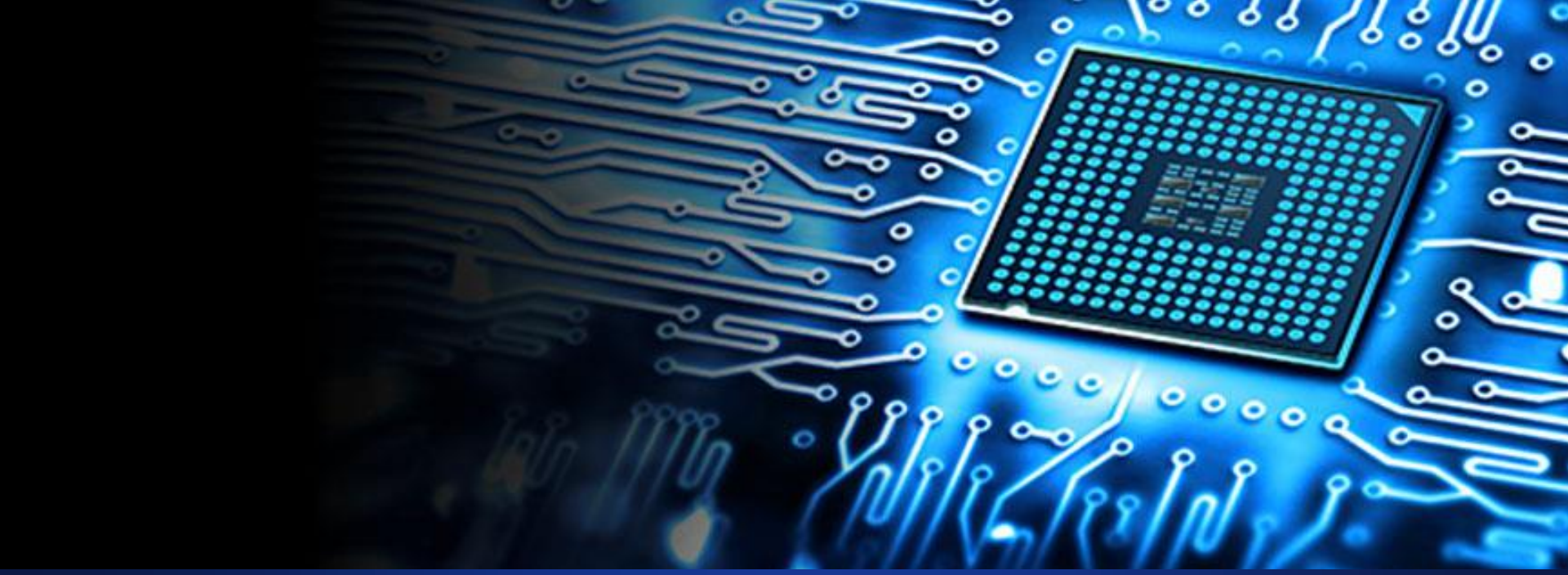


# Chapter 5
# SM Charts and Microprogramming

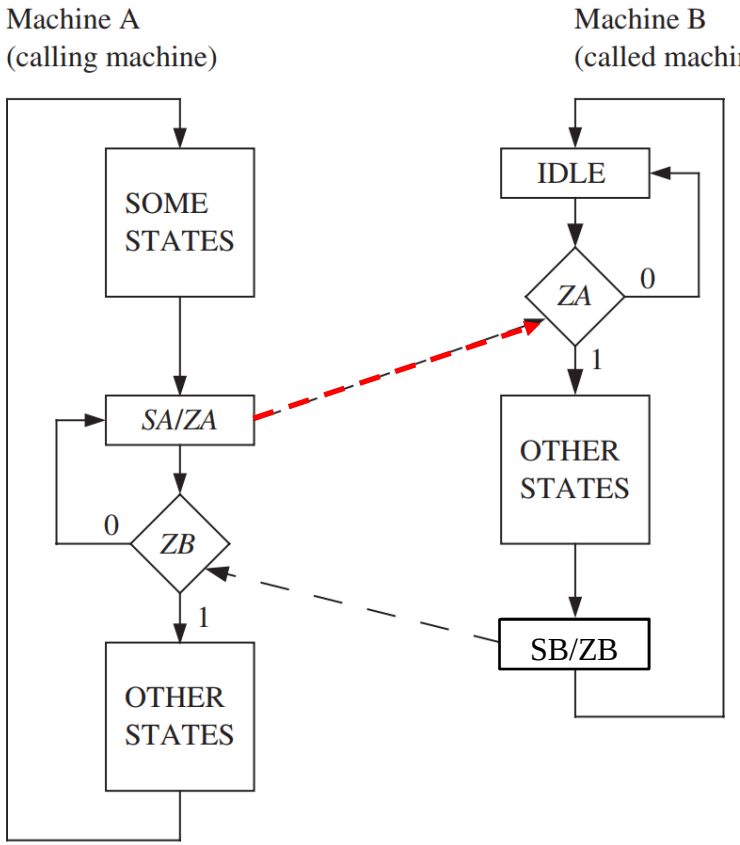
**Version: 2023/12/19**

| | Contents |
|---|---|
| 5.1 | State Machine Charts |
| 5.2 | Derivation of SM charts |
| 5.3 | Realization of SM charts |
| 5.4 | Implementation of the Dice Game |
| 5.5 | Microprogramming |
| 5.6 | Linked State Machines |

| | Contents |
|---|---|
| 5.1 | State Machine Charts |
| 5.2 | Derivation of SM charts |
| 5.3 | Realization of SM charts |
| 5.4 | Implementation of the Dice Game |
| 5.5 | Microprogramming |
| 5.6 | Linked State Machines |

# 5.6 Linked State Machines (链接状态机)

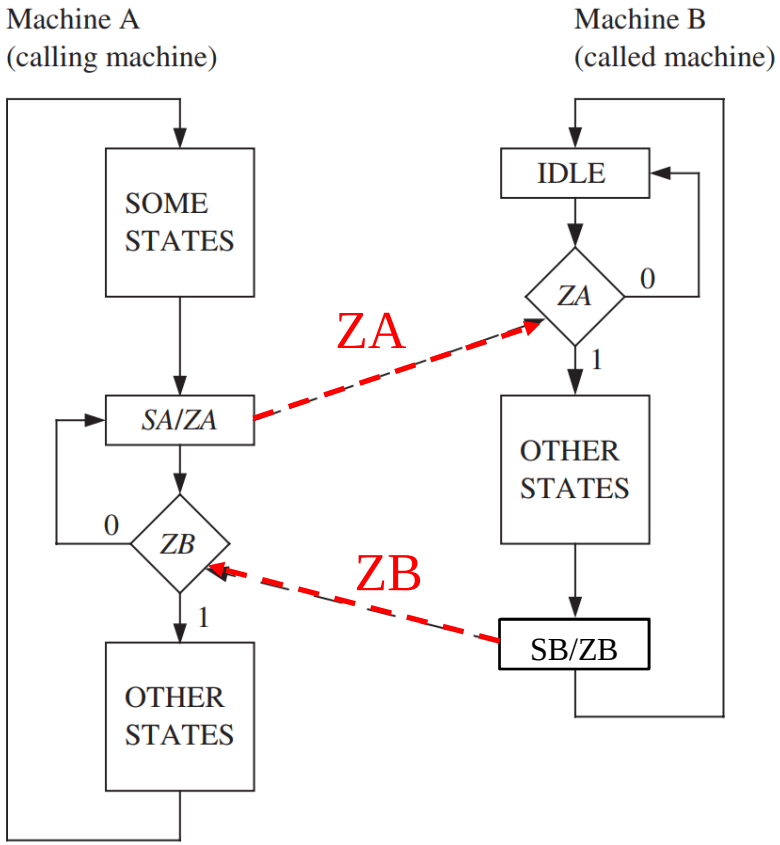
## SM Charts for Serially Linked State Machines




When a sequential machine becomes large and complex, it is desirable to divide the machine into several smaller machines that are linked together

- Each of the smaller machines is easier to design and implement
- Also, one of the submachines may be "called" in several different places by the main machine

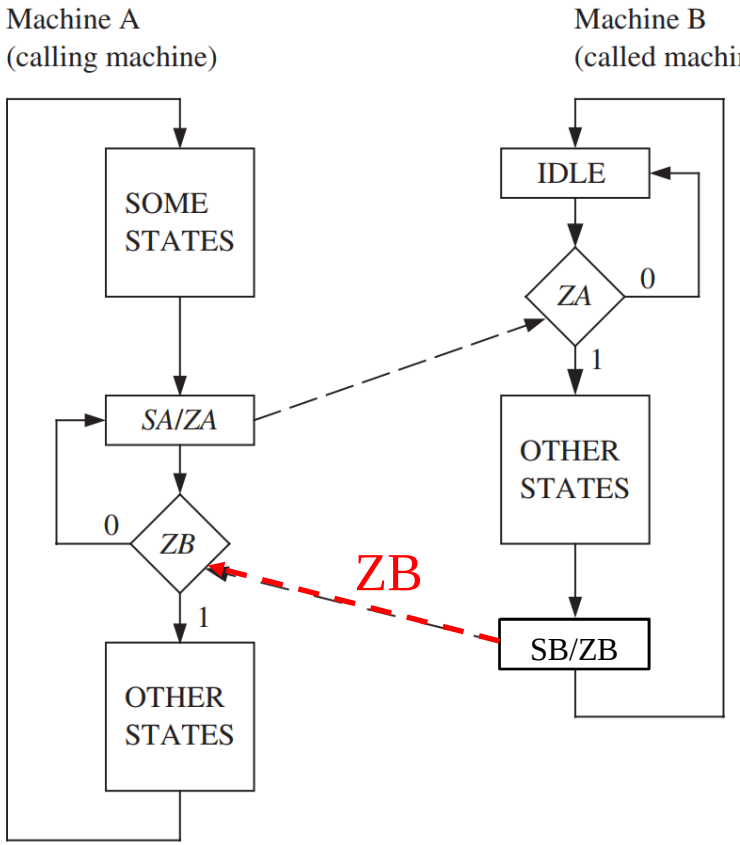
# SM Charts for Serially Linked State Machines




- The main machine (machine A) executes a sequence of “some states” until it is ready to call the submachine (machine B)
- When state SA is reached, the output signal **ZA** activates machine B

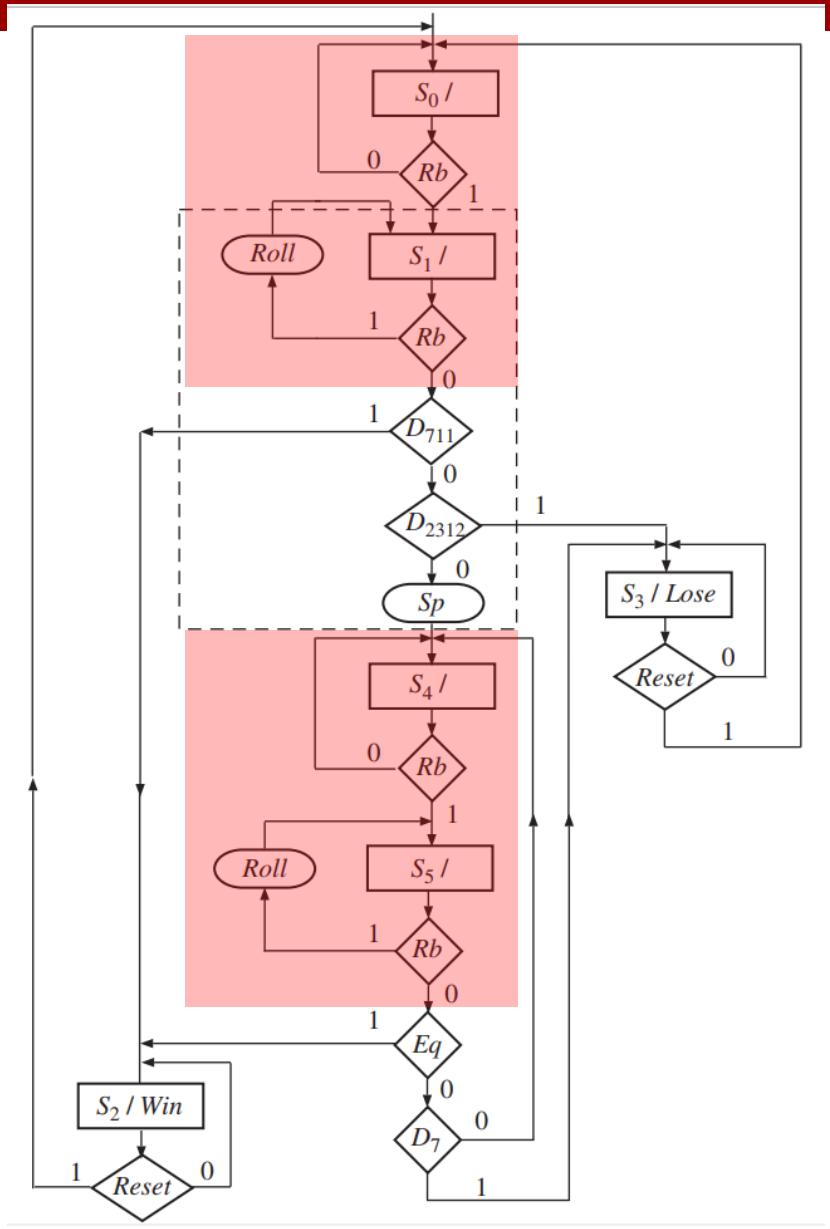
- Machine B then leaves its idle state and executes a sequence of “other states”
- When it is finished, it outputs **ZB** before returning to idle state

# SM Charts for Serially Linked State Machines




When machine A receives **ZB**, it continues to execute “other states”

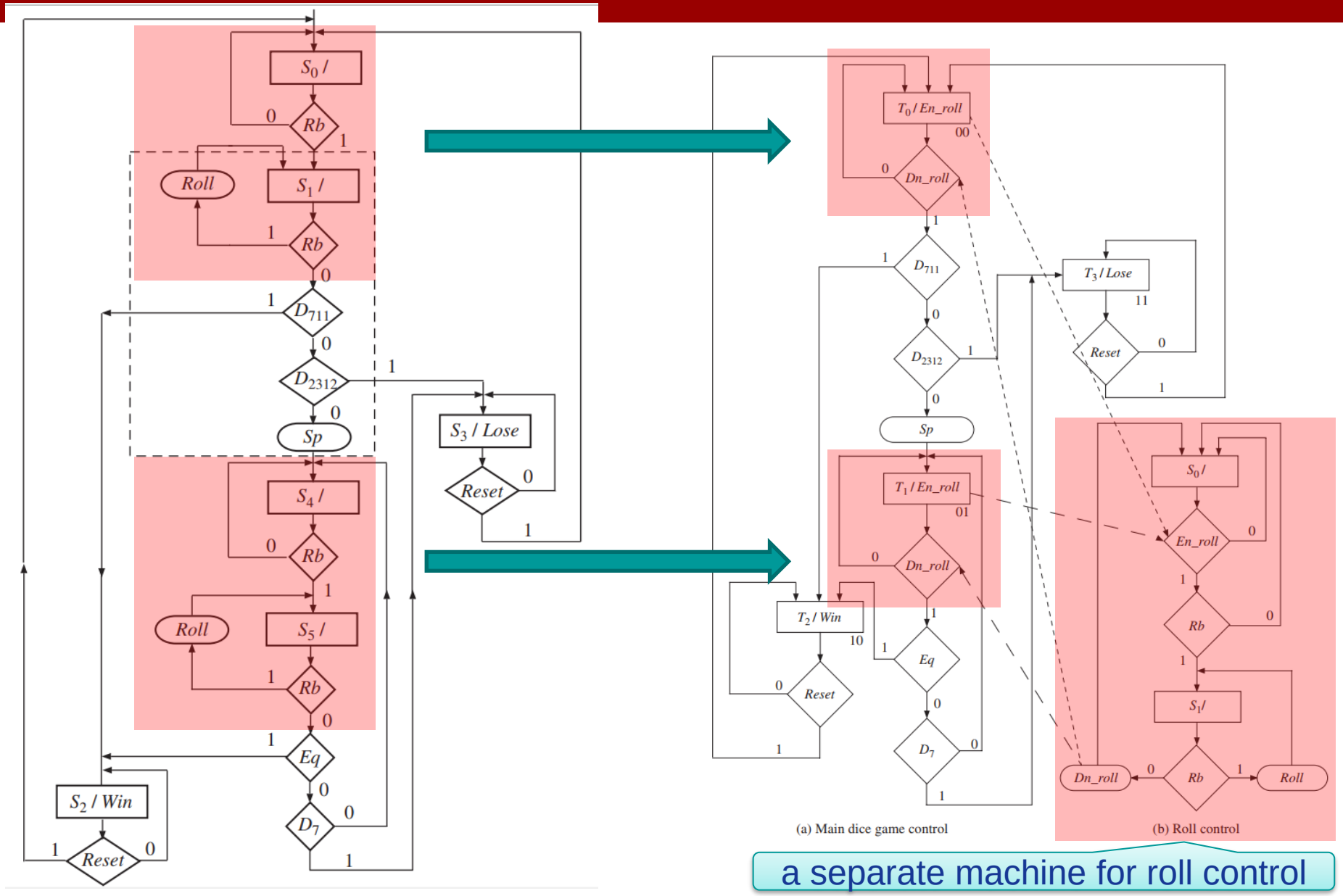
It assumes that the two machines have a common clock




## SM Chart for Dice Game

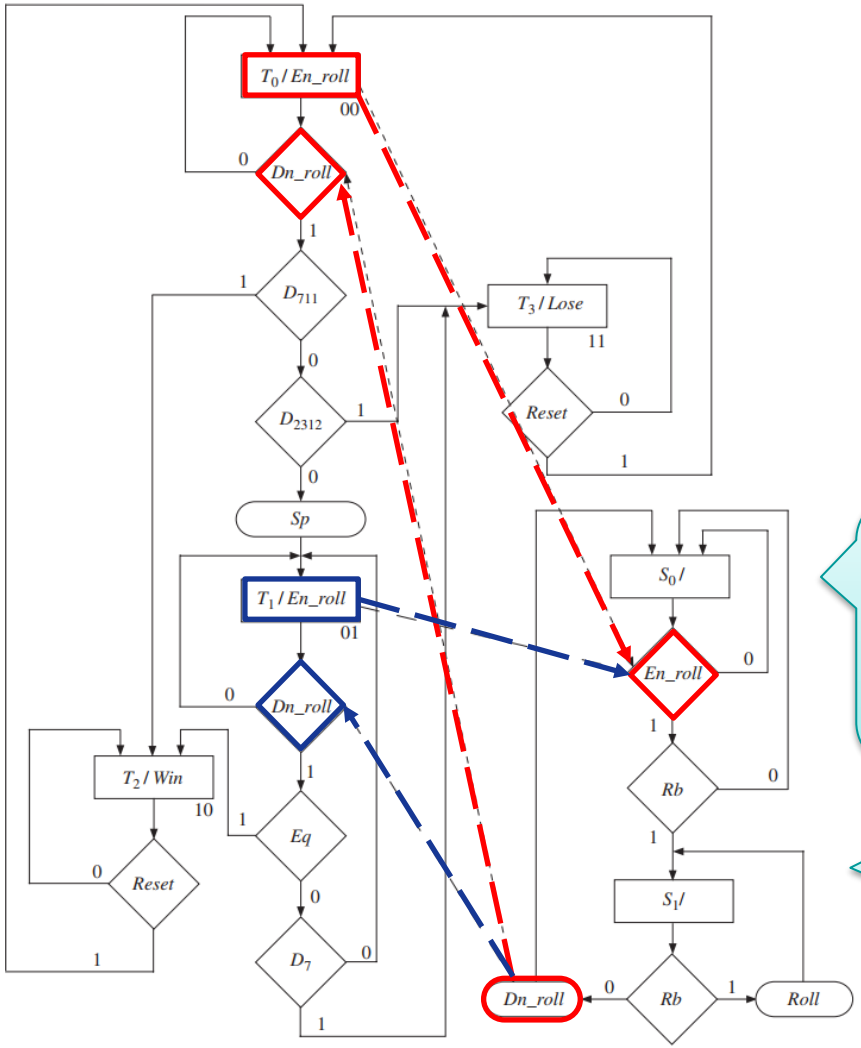
**Rb** is used to control the roll of the dice in S0 and S1 and in an identical way in S4 and S5

Since this function is repeated in two places, it is logical to use a separate machine for the roll control




a separate machine for roll control

# Linked SM Charts for Dice Game

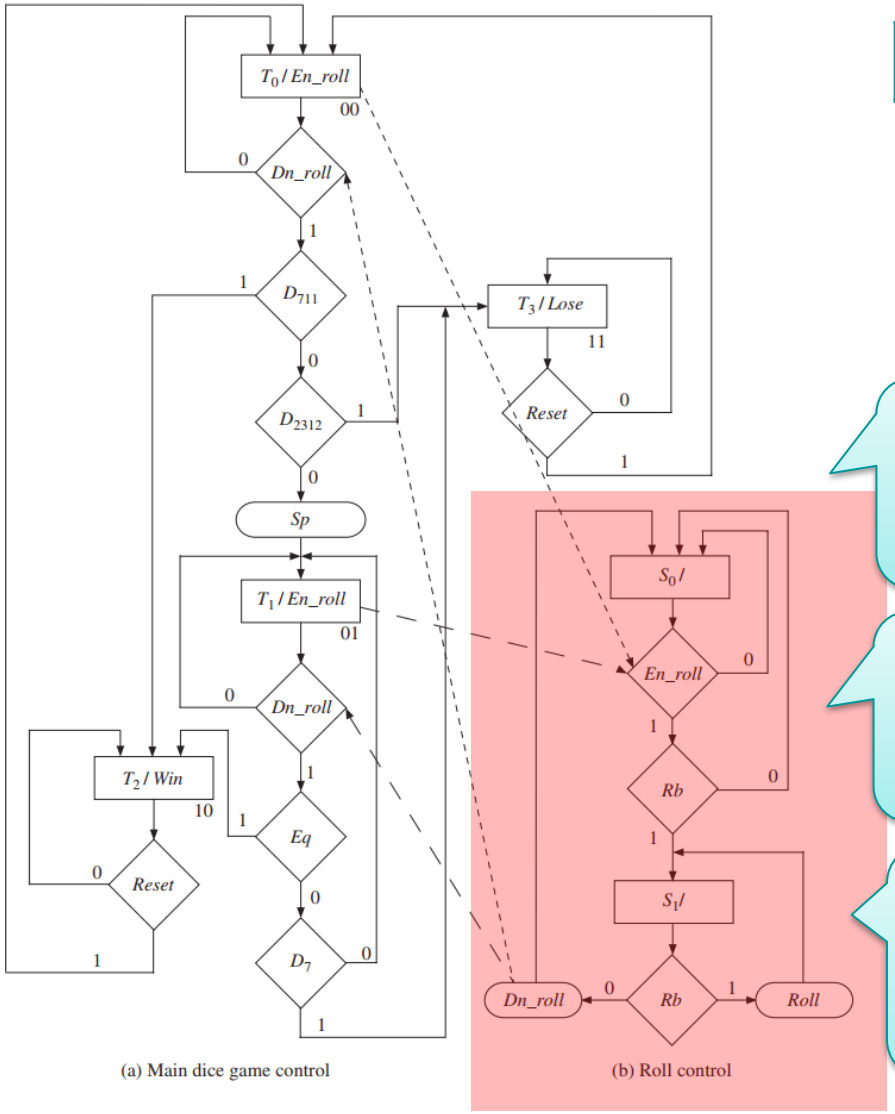



(a) Main dice game control (b) Roll control

The main control generates an En_roll (enable roll) signal in T0 and then waits for a Dn_roll (done rolling) signal before continuing

Similar action occurs in T1

# Linked SM Charts for Dice Game




The roll-control machine waits in S0 until it receives an En_roll signal from the main dice-game control

Then, when the roll button is pressed (Rb = 1), the machine goes to S1 and generates a Roll signal

It remains in S1 until Rb = 0, in which case, Dn_roll signal is generated and the machine goes back to S0