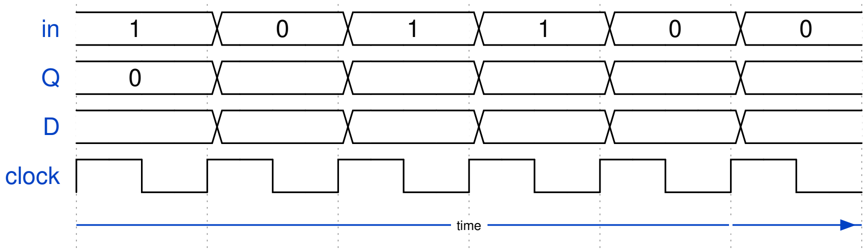
Applications of State Machines

Exercise 1: The example above is an N-bit shift register that shifts the bits right. Draw a block diagram and write the Verilog for a 6-bit shift register that shifts left.

Exercise 2:



Fill in the diagram above for a 4-bit ($N = 4$) right-shift shift register. Assume the initial value is zero. Which bit is the oldest (first) value in the waveform? Which bit of the shift register holds the oldest value?

Exercise 3: Draw a block diagram and write the Verilog for a circuit that sets an output named **`detect`** high when the sequence of values 1, 1, 0, 1 has appeared on an input named **`in`** on successive rising edges of the clock.

Exercise 4: How could you modify the code so that **`digits`** is only updated when an **`enable`** input is asserted?

Exercise 5: How many states can this state machine have?

Exercise 6: Draw the state transition diagram for this simpler implementation. How many states are there? Write the Verilog using a 3-bit `count` state variable.

Exercise 7: For which states would a **`fell`** output be asserted? A **`rose`** output? Draw the schematic and write the Verilog for this state machine. Assume an input **`in`** and a 2-bit register **`bits`** that holds the two most recent input values.

Exercise 8: Can you design an edge detector that uses only one bit?
Is this a Mealy or a Moore state machine?

Exercise 9: Write `always_ff` statements that implement these state machines.

Exercise 10: Write the state transition table for this state machine.