5.1 The *D* latch of Fig. 5.6 is constructed with four NAND gates and an inverter. Consider the following three other ways for obtaining a *D* latch. In each case, draw the logic diagram and verify the circuit operation.

(a) Use NOR gates for the *SR* latch part and AND gates for the other two. An inverter may be needed.
(b) Use NOR gates for all four gates. Inverters may be needed.
(c) Use four NAND gates only (without an inverter). This can be done by connecting the output of the upper gate in Fig. 5.6 (the gate that goes to the *SR* latch) to the input of the lower gate (instead of the inverter output).

5.2 Construct a *JK* flip-flop using a *D* flip-flop, a two-to-one-line multiplexer, and an inverter. (HDL—see Problem 5.34.)

5.3 Show that the characteristic equation for the complement output of a *JK* flip-flop is

$$Q'(t + 1) = J'Q' + KQ$$

5.4 A *PN* flip-flop has four operations: clear to 0, no change, complement, and set to 1, when inputs *P* and *N* are 00, 01, 10, and 11, respectively.

(a) Tabulate the characteristic table.
(b)* Derive the characteristic equation.
(c) Tabulate the excitation table.
(d) Show how the *PN* flip-flop can be converted to a *D* flip-flop.

5.5 Explain the differences among a truth table, a state table, a characteristic table, and an excitation table. Also, explain the difference among a Boolean equation, a state equation, a characteristic equation, and a flip-flop input equation.

5.6 A sequential circuit with two *D* flip-flops *A* and *B*, two inputs, *x* and *y*; and one output *z* is specified by the following next-state and output equations (HDL—see Problem 5.35):

$$A(t + 1) = xy' + xB$$
$$B(t + 1) = xA + xB'$$
$$z = A$$

(a) Draw the logic diagram of the circuit.
(b) List the state table for the sequential circuit.
(c) Draw the corresponding state diagram.

5.7* A sequential circuit has one flip-flop *Q*, two inputs *x* and *y*, and one output *S*. It consists of a full-adder circuit connected to a *D* flip-flop, as shown in Fig. P5.7. Derive the state table and state diagram of the sequential circuit.

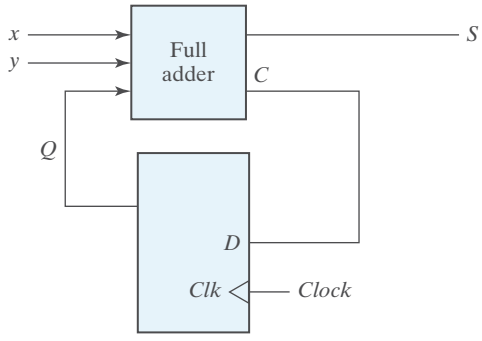

FIGURE P5.7

5.8* Derive the state table and the state diagram of the sequential circuit shown in Fig. P5.8. Explain the function that the circuit performs. (HDL—see Problem 5.36.)

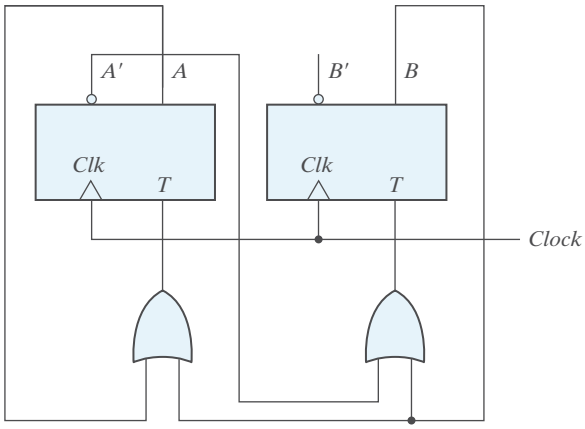

FIGURE P5.8

5.9 A sequential circuit has two *JK* flip-flops *A* and *B* and one input *x*. The circuit is described by the following flip-flop input equations:

$$J_A = x \quad K_A = B$$
$$J_B = x \quad K_B = A'$$

(a) Derive the state equations $A(t + 1)$ and $B(t + 1)$ by substituting the input equations for the *J* and *K* variables.
(b) Draw the state diagram of the circuit.

5.10 A sequential circuit has two *JK* flip-flops *A* and *B*, two inputs *x* and *y*, and one output *z*. The flip-flop input equations and circuit output equation are

$$J_A = Bx + B'y' \qquad K_A = B'xy'$$
$$J_B = A'x \qquad K_B = A + xy'$$
$$z = Ax'y' + Bx'y'$$

(a) Draw the logic diagram of the circuit.
(b) Tabulate the state table.
(c) Derive the state equations for *A* and *B*.

5.11 For the circuit described by the state diagram of Fig. 5.16,
(a)* Determine the state transitions and output sequence that will be generated when an input sequence of 010110111011110 is applied to the circuit and it is initially in the state 00.
(b) Find all of the equivalent states in Fig. 5.16 and draw a simpler, but equivalent, state diagram.
(c) Using *D* flip-flops, design the equivalent machine (including its logic diagram) described by the state diagram in (b).

5.12 For the following state table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

(a) Draw the corresponding state diagram.
(b)* Tabulate the reduced state table.
(c) Draw the state diagram corresponding to the reduced state table.

5.13 Starting from state a, and the input sequence 01110010011, determine the output sequence for
(a) The state table of the previous problem.
(b) The reduced state table from the previous problem. Show that the same output sequence is obtained for both.

5.14 Substitute the one-hot assignment 2 from Table 5.9 to the states in Table 5.8 and obtain the binary state table.

5.15 List a state table for the *JK* flip-flop using Q as the present and next state and J and K as inputs. Design the sequential circuit specified by the state table and show that it is equivalent to Fig. 5.12(a).

5.16 Design a sequential circuit with two *D* flip-flops A and B, and one input *x_in*.
(a)* When *x_in* = 0, the state of the circuit remains the same. When *x_in* = 1, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats.
(b) When *x_in* = 0, the state of the circuit remains the same. When *x_in* =1, the circuit goes through the state transitions from 00 to 11, to 01, to 10, back to 00, and repeats. (HDL—see Problem 5.38.)

5.17 Design a one-input, one-output serial 2's complementer. The circuit accepts a string of bits from the input and generates the 2's complement at the output. The circuit can be reset asynchronously to start and end the operation. (HDL—see Problem 5.39.)

5.18* Design a sequential circuit with two *JK* flip-flops A and B and two inputs E and F. If $E = 0$, the circuit remains in the same state regardless of the value of F. When $E = 1$ and $F = 1$, the circuit goes through the state transitions from 00 to 01, to 10, to 11, back to 00, and repeats. When $E = 1$ and $F = 0$, the circuit goes through the state transitions from 00 to 11, to 10, to 01, back to 00, and repeats. (HDL—see Problem 5.40.)

5.19 A sequential circuit has three flip-flops A, B, C; one input *x_in*; and one output *y_out*. The state diagram is shown in Fig. P5.19. The circuit is to be designed by treating the unused states as don't-care conditions. Analyze the circuit obtained from the design to determine the effect of the unused states. (HDL—see Problem 5.41.)

(a)* Use *D* flip-flops in the design.
(b) Use *JK* flip-flops in the design.

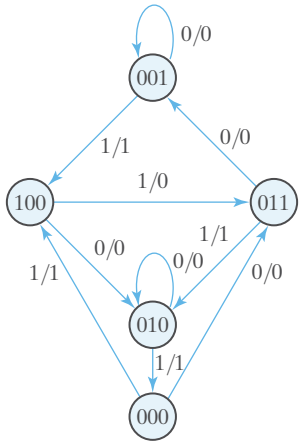

FIGURE P5.19

5.20 Design the sequential circuit specified by the state diagram of Fig. 5.19, using *T* flip-flops.

5.21 What is the main difference between an **initial** statement and an **always** statement in Verilog HDL?

5.22 Draw the waveform generated by the statements below:

(a)
```
initial begin
   w = 0; #10 w = 1; # 40 w = 0;  # 20 w = 1;  #15 w = 0;
end
```

(b)
```
initial fork
   w = 0;  #10 w = 1;  # 40 w = 0;  # 20 w = 1;  #15 w = 0;
join
```

5.23* Consider the following statements assuming that *RegA* contains the value of 50 initially.

(a)
```
RegA = 125;
RegB = RegA;
```

(b)
```
RegA <= 125;
RegB <= RegA;
```

What are the values of RegA and RegB after execution?

5.24 Write and verify an HDL behavioral description of a positive-edge-sensitive *D* flip-flop with asynchronous preset and clear.

5.25 A special positive-edge-triggered flip-flop circuit component has four inputs *D*1, *D*2, *D*3, and *D*4, and a two-bit control input that chooses between them. Write and verify an HDL behavioral description of this component.

5.26 Write and verify an HDL behavioral description of the *JK* flip-flop using an if-else statement based on the value of the present state.

(a)* Obtain the characteristic equation when $Q = 0$ or $Q = 1$.
(b) Specify how the *J* and *K* inputs affect the output of the flip-flop at each clock tick.

5.27 Rewrite and verify the description of HDL Example 5.5 by combining the state transitions and output into one **always** block.

5.28 Simulate the sequential circuit shown in Fig. 5.17.

(a) Write the HDL description of the state diagram (i.e., behavioral model).
(b) Write the HDL description of the logic (circuit) diagram (i.e., a structural model).
(c) Write an HDL stimulus with a sequence of inputs: 00, 01, 11, 10. Verify that the response is the same for both descriptions.

5.29 Write a behavioral description of the state machine described by the state diagram shown in Fig. P5.19. Write a test bench and verify the functionality of the description.

5.30 Draw the logic diagram for the sequential circuit described by the following HDL module:

```
module Seq_Ckt (input A, B, C, E output reg Q,input CLK,);
 reg E;

 always @ (posedge CLK)
 begin
  E <= A || B;
  Q <= E && C;
 end
endmodule
```

5.31 How should the description in problem 5.30 be written to have the same behavior when the assignments are made with = instead of with <= ?

5.32 Using an **initial** statement with a **begin** . . . **end** block write a Verilog description of the waveforms shown in Fig. P5.32. Repeat using a **fork** . . . **join** block.

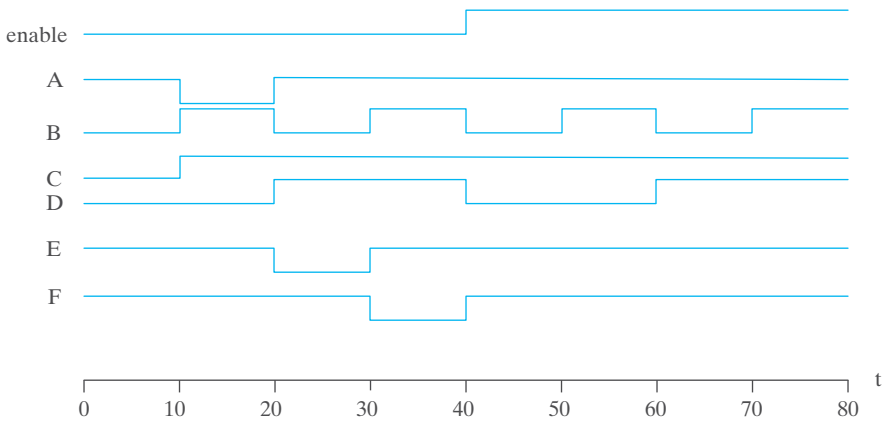

FIGURE P5.32
Waveforms for Problem 5.32

5.33 Explain why it is important that the stimulus signals in a test bench be synchronized to the inactive edge of the clock of the sequential circuit that is to be tested.

5.34 Write and verify an HDL structural description of the machine having the circuit diagram (schematic) shown in Fig. 5.5.

5.35 Write and verify an HDL model of the sequential circuit described in Problem 5.6.

5.36 Write and verify an HDL structural description of the machine having the circuit diagram (schematic) shown in Fig. P5.8.

5.37 Write and verify HDL behavioral descriptions of the state machines shown in Figs. 5.25

and 5.26. Write a test bench to compare the state sequences and input–output behaviors of the two machines.

5.38 Write and verify an HDL behavioral description of the machine described in Problem 5.16.

5.39 Write and verify a behavioral description of the machine specified in Problem 5.17.

5.40 Write and verify a behavioral description of the machine specified in Problem 5.18.

5.41 Write and verify a behavioral description of the machine specified in Problem 5.19. (*Hint*: See the discussion of the **default** case item preceding HDL Example 4.8 in Chapter 4.)

5.42 Write and verify an HDL structural description of the circuit shown in Fig. 5.29.

5.43 Write and verify an HDL behavioral description of the three-bit binary counter in Fig. 5.34.

5.44 Write and verify a Verilog model of a *D* flip-flop having asynchronous reset.

5.45 Write and verify an HDL behavioral description of the sequence detector described in Fig. 5.27.

5.46 A synchronous finite state machine has an input *x_in* and an output *y_out*. When *x_in* changes from 0 to 1, the output *y_out* is to assert for three cycles, regardless of the value of *x_in,* and then de-assert for two cycles before the machine will respond to another assertion of *x_in*. The machine is to have active-low synchronous reset.

(a) Draw the state diagram of the machine.

(b) Write and verify a Verilog model of the machine.

5.47 Write a Verilog model of a synchronous finite state machine whose output is the sequence 0, 2, 4, 6, 8 10, 12, 14, 0 . . . . The machine is controlled by a single input, *Run*, so that counting occurs while *Run* is asserted, suspends while *Run* is de-asserted, and resumes the count when *Run* is re-asserted. Clearly state any assumptions that you make.

5.48 Write a Verilog model of the Mealy FSM described by the state diagram in Fig. P5.48. Develop a test bench and demonstrate that the machine state transitions and output correspond to its state diagram.

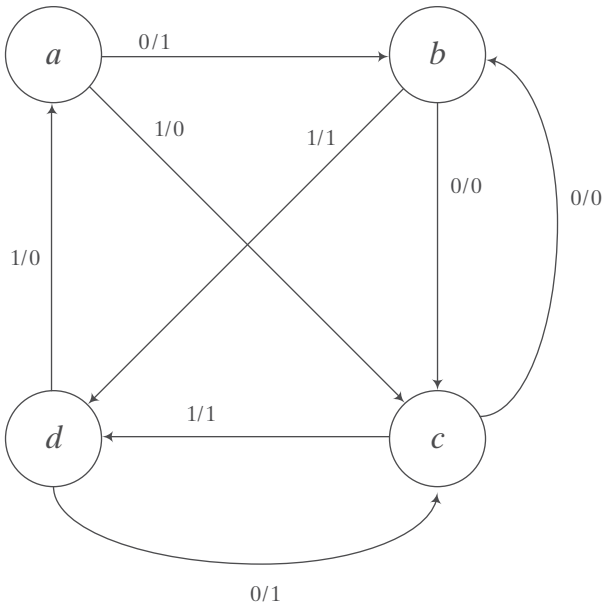

FIGURE P5.48

5.49 Write a Verilog model of the Moore FSM described by the state diagram in Fig. P5.49. Develop a test bench and demonstrate that the machine's state transitions and output correspond to its state diagram.

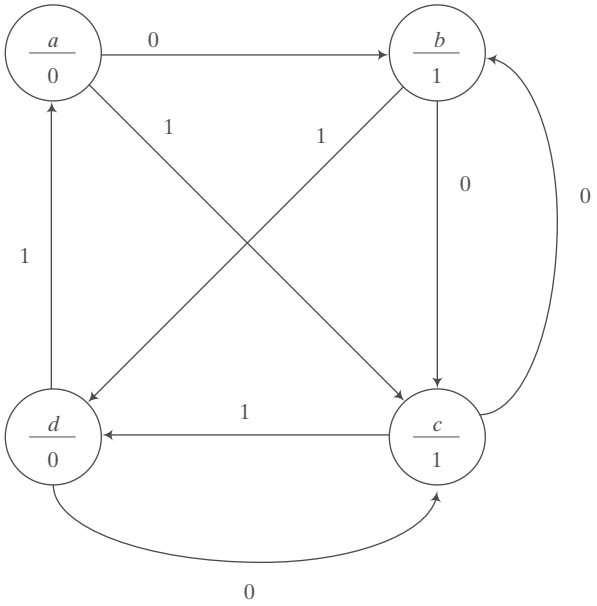

FIGURE P5.49

5.50 A synchronous Moore FSM has a single input, *x_in*, and a single output *y_out*. The machine is to monitor the input and remain in its initial state until a second sample of *x_in* is detected to be 1. Upon detecting the second assertion of *x_in* *y_out* is to asserted and remain asserted until a fourth assertion of *x_in* is detected. When the fourth assertion of *x_in* is detected the machine is to return to its initial state and resume monitoring of *x_in*.

(a) Draw the state diagram of the machine.

(b) Write and verify a Verilog model of the machine.

5.51 Draw the state diagram of the machine described by the Verilog model given below.

```
module Prob_5_51 (output reg y_out, input x_in, clk, reset);
 parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10, s3 = 2'b11;
 reg [1:0] state, next_state;
 always @ (posedge clk, negedge reset) begin
  if (reset == 1'b0) state <= s0;
  else state <= next_state;
 always @(state, x_in) begin
  y_out = 0;
  next_state = s0;
  case (state)
  s0: if x_in = 1 begin y_out = 0; if (x_in) next_state = s1; else next_state = s0; end
  s1: if x_in = 1 begin y_out = 0; if (x_in) next_state = s2; else next_state = s1; end
```

```
   s2: if x_in = 1 begin y_out = 1; if (x_in) next_state = s3; else next_state = s2; end
   s3: if x_in = 1 begin y_out = 1; if (x_in) next_state = s0; else next_state = s3; end
   default: next_state = s0;
   endcase
 end
endmodule
```

5.52 Draw the state diagram of the machine described by the Verilog model given below.

```
module Prob_5_52 (output reg y_out, input x_in, clk, reset);
   parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10, s3 = 2'b11;
 reg [1:0] state, next_state;
 always @ (posedge clk, negedge reset) begin
   if (reset == 1'b0) state <= s0;
   else state <= next_state;
 always @(state, x_in) begin
   y_out = 0;
   next_state = s0;
   case (state)
   s0: if x_in = 1 begin y_out = 0; if (x_in) next_state = s1; else next_state = s0; end
   s1: if x_in = 1 begin y_out = 0; if (x_in) next_state = s2; else next_state = s1; end
   s2: if x_in = 1 if (x_in) begin next_state = s3; y_out = 0;
                   else begin next_state = s2; y_out = 1; end
   s3: if x_in = 1 begin y_out = 1; if (x_in) next_state = s0; else next_state = s3; end
   default: next_state = s0;
   endcase
 end
endmodule
```

5.53 Draw a state diagram and write a Verilog model of a Mealy synchronous state machine having a single input, *x_in,* and a single output *y_out,* such that *y_out* is asserted if the total number of 1's received is a multiple of 3.

5.54 A synchronous Moore machine has two inputs, $x1$, and $x2$, and output *y_out*. If both inputs have the same value the output is asserted for one cycle; otherwise the output is 0. Develop a state diagram and a write a Verilog behavioral model of the machine. Demonstrate that the machine operates correctly.

5.55 Develop the state diagram for a Mealy state machine that detects a sequence of three or more consecutive 1's in a string of bits coming through an input line.

5.56 Using manual methods, obtain the logic diagram of a three-bit counter that counts in the sequence $0, 2, 4, 6, 0, \ldots$.

5.57 Write and verify a Verilog behavioral model of a three-bit counter that counts in the sequence $0, 2, 4, 6, 0, \ldots$.

5.58 Write and verify a Verilog behavioral model of the counter designed in Problem 5.55.

5.59 Write and verify a Verilog structural model of the counter described in Problem 5.56.

5.60 Write and verify a Verilog behavioral model of a four-bit counter that counts in the sequence $0, 1, \ldots, 9, 0, 1, 2, \ldots$.