

## PROBLEMS

---

(Answers to problems marked with \* appear at the end of the book. Where appropriate, a logic design and its related HDL modeling problem are cross-referenced.)

Note: For each problem that requires writing and verifying a Verilog description, a test plan is to be written to identify which functional features are to be tested during the simulation and how they will be tested. For example, a reset on the fly could be tested by asserting the reset signal while the simulated machine is in a state other than the reset state. The test plan is to guide the development of a test bench that will implement the plan. Simulate the model using the test bench and verify that the behavior is correct. If synthesis tools and an ASIC cell library or a field programmable gate array (FPGA) tool suite are available, the Verilog descriptions developed for Problems 6.34–6.51 can be assigned as synthesis exercises. The gate-level circuit produced by the synthesis tools should be simulated and compared to the simulation results for the pre-synthesis model.

In some of the HDL problems, there may be a need to deal with the issue of unused states (see the discussion of the **default case** item preceding HDL Example 4.8 in Chapter 4).

- 6.1 Include a 2-input NAND gate in the register of Fig. 6.1 and connect the gate output to the  $C$  inputs of all the flip-flops. One input of the NAND gate receives the clock pulses from the clock generator, and the other input of the NAND gate provides a parallel load control. Explain the operation of the modified register. Explain why this circuit might have operational problems.
- 6.2 Include a synchronous clear input to the register of Fig. 6.2. The modified register will have a parallel load capability and a synchronous clear capability. The register is cleared synchronously when the clock goes through a positive transition and the clear input is equal to 1. (HDL—see Problem 6.35(a), (b).)
- 6.3 What is the difference between serial and parallel transfer? Explain how to convert serial data to parallel and parallel data to serial. What type of register is needed?
- 6.4\* The contents of a four-bit register is initially 0110. The register is shifted six times to the right with the serial input being 1011100. What is the content of the register after each shift?
- 6.5 The four-bit universal shift register shown in Fig. 6.7 is enclosed within one IC component package. (HDL—see Problem 6.52.)
  - (a) Draw a block diagram of the IC showing all inputs and outputs. Include two pins for the power supply.
  - (b) Draw a block diagram using two of these ICs to produce an eight-bit universal shift register.
- 6.6 Design a four-bit shift register with parallel load using  $D$  flip-flops. There are two control inputs: *shift* and *load*. When *shift* = 1, the content of the register is shifted by one position. New data are transferred into the register when *load* = 1 and *shift* = 0. If both control inputs are equal to 0, the content of the register does not change. (HDL—see Problem 6.35(c), (d).)
- 6.7 Draw the logic diagram of a four-bit register with four  $D$  flip-flops and four  $4 \times 1$  multiplexers with mode selection inputs  $s_1$  and  $s_0$ . The register operates according to the following function table. (HDL—see Problem 6.35(e), (f).)

$s_1$	$s_0$	Register Operation
0	0	No change
1	0	Complement the four outputs
0	1	Clear register to 0 (synchronous with the clock)
1	1	Load parallel data

- 6.8\*** The serial adder of Fig. 6.6 uses two four-bit registers. Register  $A$  holds the binary number 0101 and register  $B$  holds 0111. The carry flip-flop is initially reset to 0. List the binary values in register  $A$  and the carry flip-flop after each shift. (HDL—see Problem 6.54).
- 6.9** Two ways for implementing a serial adder ( $A + B$ ) is shown in Section 6.2. It is necessary to modify the circuits to convert them to serial subtractors ( $A - B$ ).
- (a) Using the circuit of Fig. 6.5, show the changes needed to perform  $A + 2$ 's complement of  $B$ . (HDL—see Problem 6.35(h).)
- (b) \*Using the circuit of Fig. 6.6, show the changes needed by modifying Table 6.2 from an adder to a subtractor circuit. (See Problem 4.12). (HDL—see Problem 6.35(i).)
- 6.10** Design a serial 2's completer with a shift register and a flip-flop. The binary number is shifted out from one side and its 2's complement shifted into the other side of the shift register. (HDL—see Problem 6.35(j).)
- 6.11** A binary ripple counter uses flip-flops that trigger on the positive-edge of the clock. What will be the count if
- (a) the normal outputs of the flip-flops are connected to the clock and
- (b) the complement outputs of the flip-flops are connected to the clock?
- 6.12** Draw the logic diagram of a four-bit binary ripple countdown counter using
- (a) flip-flops that trigger on the positive-edge of the clock and
- (b) flip-flops that trigger on the negative-edge of the clock.
- 6.13** Show that a BCD ripple counter can be constructed using a four-bit binary ripple counter with asynchronous clear and a NAND gate that detects the occurrence of count 1010. (HDL—see Problem 6.35(k).)
- 6.14** How many flip-flop will be complemented in a 10-bit binary ripple counter to reach the next count after the following counts?
- (a) \*1001100111
- (b) 1111000111
- (c) 0000001111
- 6.15\*** A flip-flops has a 3 ns delay from the time the clock edge occurs to the time the output is complemented. What is the maximum delay in a 10-bit binary ripple counter that uses these flip-flops? What is the maximum frequency at which the counter can operate reliably?
- 6.16\*** The BCD ripple counter shown in Fig. 6.10 has four flip-flops and 16 states, of which only 10 are used. Analyze the circuit and determine the next state for each of the other six unused states. What will happen if a noise signal sends the circuit to one of the unused states? (HDL—see Problem 6.54.)

- 6.17\* Design a four-bit binary synchronous counter with  $D$  flip-flops.
- 6.18 What operation is performed in the up-down counter of Fig. 6.13 when both the up and down inputs are enabled? Modify the circuit so that when both inputs are equal to 1, the counter does not change state. (HDL—see Problem 6.35(1).)
- 6.19 The flip-flop input equations for a BCD counter using  $T$  flip-flops are given in Section 6.4. Obtain the input equations for a BCD counter that uses (a)  $JK$  flip-flops and (b)\*  $D$  flip-flops. Compare the three designs to determine which one is the most efficient.
- 6.20 Enclose the binary counter with parallel load of Fig. 6.14 in a block diagram showing, all inputs and outputs.
  - (a) Show the connections of four such blocks to produce a 16-bit counter with parallel load.
  - (b) Construct a binary counter that counts from 0 through binary 127.
- 6.21\* The counter of Fig. 6.14 has two control inputs—*Load* ( $L$ ) and *Count* ( $C$ )—and a data input, ( $I_i$ ).
  - (a) Derive the flip-flop input equations for  $J$  and  $K$  of the first stage in terms of  $L$ ,  $C$ , and  $I$ .
  - (b) The logic diagram of the first stage of an equivalent circuit is shown in Fig. P6.21. Verify that this circuit is equivalent to the one in (a).

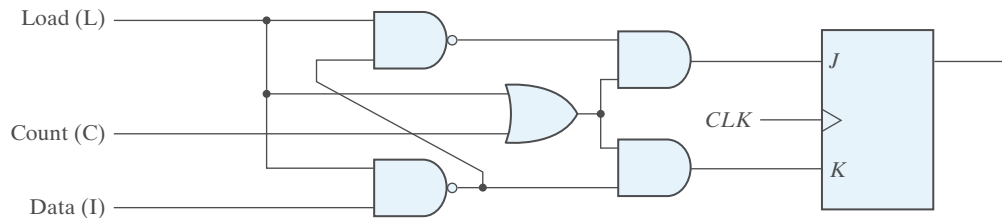


FIGURE P6.21

- 6.22 For the circuit of Fig. 6.14, give three alternatives for a mod-10 counter (i.e., the count evolves through a sequence of 12 distinct states).
  - (a) Using an AND gate and the load input.
  - (b) Using the output carry.
  - (c) Using a NAND gate and the asynchronous clear input.
- 6.23 Design a timing circuit that provides an output signal that stays on for exactly twelve clock cycles. A start signal sends the output to the 1 state, and after twelve clock cycles the signal returns to the 0 state. (HDL—see Problem 6.45.)
- 6.24\* Design a counter with  $T$  flip-flops that goes through the following binary repeated sequence: 0, 1, 3, 7, 6, 4. Show that when binary states 010 and 101 are considered as don't care conditions, the counter may not operate properly. Find a way to correct the design. (HDL—see Problem 6.55.)
- 6.25 It is necessary to generate six repeated timing signals  $T_0$  through  $T_5$  similar to the ones shown in Fig. 6.17(c). Design the circuit using (HDL—see Problem 6.46.):
  - (a) flip-flops only.
  - (b) a counter and a decoder.

- 6.26\*** A digital system has a clock generator that produces pulses at a frequency of 80 MHz. Design a circuit that provides a clock with a cycle time of 50 ns.
- 6.27** Using *JK* flip-flops,  
 (a) Design a counter with the following repeated binary sequence: 0, 1, 2, 3, 4, 5, 6. (HDL—see Problem 6.50(a), 6.51.).  
 (b) Draw the logic diagram of the counter.
- 6.28** Using *D* flip-flops,  
 (a) \*Design a counter with the following repeated binary sequence: 0, 1, 2, 4, 6. (HDL—see Problem 6.50(b).)  
 (b) Draw the logic diagram of the counter.  
 (c) Design a counter with the following repeated binary sequence: 0, 2, 4, 6, 8.  
 (d) Draw the logic diagram of the counter.
- 6.29** List the eight unused states in the switch-tail ring counter of Fig. 6.18(a). Determine the next state for each of these states and show that, if the counter finds itself in an invalid state, it does not return to a valid state. Modify the circuit as recommended in the text and show that the counter produces the same sequence of states and that the circuit reaches a valid state from any one of the unused states.
- 6.30** Show that a Johnson counter with  $n$  flip-flops produces a sequence of  $2n$  states. List the 10 states produced with five flip-flops and the Boolean terms of each of the 10 AND gate outputs.
- 6.31** Write and verify the HDL behavioral and structural descriptions of the four-bit register Fig. 6.1.
- 6.32** (a) Write and verify an HDL behavioral description of a four-bit register with parallel load and asynchronous clear.  
 (b) Write and verify the HDL structural description of the four-bit register with parallel load shown in Fig. 6.2. Use a  $2 \times 1$  multiplexer for the flip-flop inputs. Include an asynchronous clear input.  
 (c) Verify both descriptions, using a test bench.
- 6.33** The following stimulus program is used to simulate the binary counter with parallel load described in HDL Example 6.3. Draw waveforms showing the output of the counter and the carry output from  $t = 0$  to  $t = 155$  ns.

```
// Stimulus for testing the binary counter of Example 6.3
module testcounter;
  reg Count, Load, CLK, Clr;
  reg [3: 0] IN;
  wire C0;
  wire [3: 0] A;
  counter cnt (Count, Load, IN, CLK, Clr, A, C0);
  always
    #5 CLK = ~CLK;
  initial
    begin
      Clr = 0;
      CLK = 1;
      Load = 0; Count = 1;
    end

```

```

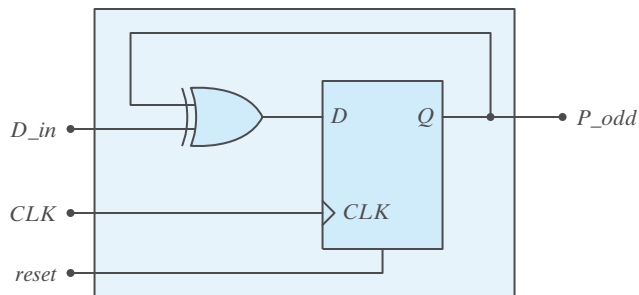
#5 Clr = 1;
#40 Load = 1; IN = 4'b1001;
#10 Load = 0;
#70 Count = 0;
#20 $finish;
end
endmodule

```

- 6.34\*** Write and verify the HDL behavioral description of a four-bit shift register (see Fig. 6.3).
- 6.35** Write and verify
- A structural HDL model for the register described in Problem 6.2
  - \*A behavioral HDL model for the register described in Problem 6.2
  - A structural HDL model for the register described in Problem 6.6
  - A behavioral HDL model for the register described in Problem 6.6
  - A structural HDL model for the register described in Problem 6.7
  - A behavioral HDL model for the register described in Problem 6.7
  - A behavioral HDL model of the binary counter described in Fig. 6.8(b)
  - A behavioral HDL model of the serial subtractor described in Problem 6.9(a)
  - A behavioral HDL model of the serial subtractor described in Problem 6.9(b)
  - A behavioral HDL model of the serial 2's complemeter described in Problem 6.10
  - A behavioral HDL model of the BCD ripple counter described in Problem 6.13
  - A behavioral HDL model of the up–down counter described in Problem 6.18.
- 6.36** Write and verify the HDL behavioral and structural descriptions of the four-bit up–down counter whose logic diagram is described by Fig. 6.13, Table 6.5, and Table 6.6.
- 6.37** Write and verify a behavioral description of the counter described in Problem 6.24.
- \*Using an **if ... else** statement
  - Using a **case** statement
  - A finite state machine.
- 6.38** Write and verify the HDL behavioral description of a four-bit up–down counter with parallel load using the following control inputs:
- \*The counter has three control inputs for the three operations: *Up*, *Down*, and *Load*. The order of precedence is: *Load*, *Up*, and *Down*.
  - The counter has two selection inputs to specify four operations: *Load*, *Up*, *Down*, and no change.
- 6.39** Write and verify HDL behavioral and structural descriptions of the counter of Fig. 6.16.
- 6.40** Write and verify the HDL description of an eight-bit ring-counter similar to the one shown in Fig. 6.17(a).
- 6.41** Write and verify the HDL description of a four-bit switch-tail ring (Johnson) counter (Fig. 6.18a).
- 6.42\*** The comment with the last clause of the if statement in *Binary\_Counter\_4\_Par\_Load* in HDL Example 6.3 notes that the statement is redundant. Explain why this statement can be removed without changing the behavior implemented by the description.
- 6.43** The scheme shown in Fig. 6.4 gates the clock to control the serial transfer of data from shift register A to shift register B. Using multiplexers at the input of each cell of the shift registers, develop a structural model of an alternative circuit that does not alter the clock path. The

top level of the design hierarchy is to instantiate the shift registers. The module describing the shift registers is to have instantiations of flip-flops and muxes. Describe the mux and flip-flop modules with behavioral models. Be sure to consider the need to reset the machine. Develop a test bench to simulate the circuit and demonstrate the transfer of data.

- 6.44** Modify the design of the serial adder shown in Fig. 6.5 by removing the gated clock to the  $D$  flip-flop and supplying the clock signal to it directly. Augment the  $D$  flip-flop with a mux to recirculate the contents of the flip-flop when shifting is suspended and provide the carry out of the full adder when shifting is active. The shift registers are to incorporate this feature also, rather than use a gated clock. The top-level of the design is to instantiate modules using behavioral models for the shift registers, full adder,  $D$  flip-flop, and mux. Assume asynchronous reset. Develop a test bench to simulate the circuit and demonstrate the transfer of data.
- 6.45\*** Write and verify a behavioral description of a finite state machine to implement the counter described in Problem 6.24.
- 6.46** Problem 6.25 specifies an implementation of a circuit to generate timing signals using
- Only flip-flops.
  - A counter and a decoder.
- As an alternative, write a behavioral description (without consideration of the actual hardware) of a state machine whose output generates the timing signals  $T_0$  through  $T_5$ .
- 6.47** Write a behavioral description of the circuit shown in Fig. P6.47 and verify that the circuit's output is asserted if successive samples of the input have an odd number of 1s.



**FIGURE P6.47**  
Circuit for Problem 6.47

- 6.48** Write and verify a behavioral description of the counter shown in Fig. P6.48(a); repeat for the counter in Fig. P6.48(b).
- 6.49** Write a test plan for verifying the functionality of the universal shift register described in HDL Example 6.1. Using the test plan, simulate the model given in HDL Example 6.1.
- 6.50** Write and verify a behavioral model of the counter described in
- Problem 6.27
  - Problem 6.28
- 6.51** Without requiring a state machine, and using a shift register and additional logic, write and verify a model of an alternative to the sequence detector described in Fig. 5.27. Compare the implementations.
- 6.52** Write a Verilog structural model of the universal shift register in Fig. 6.7. Verify all modes of its operation.