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.

4.1 Consider the combinational circuits shown in Fig. P4.1 (HDL — see Problem 4.49).

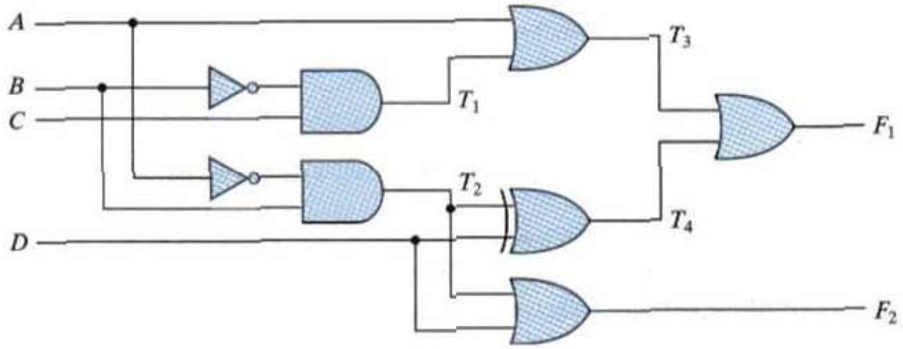


FIGURE P4.1

(a)* Derive the Boolean expressions for T_1 through T_4. Evaluate the outputs F_1 and F_2 as a function of the four inputs.

(b) List the truth table with 16 binary combinations of the four input variables. Then list the binary values for T_1 through T_4 and outputs F_1 and F_2 in the table.

(c) Plot the Boolean output functions obtained in part (b) on maps, and show that the simplified Boolean expressions are equivalent to the ones obtained in part (a).

4.2* Obtain the simplified Boolean expressions for outputs F and G in terms of the input variables in the circuit of Fig. P4.2.

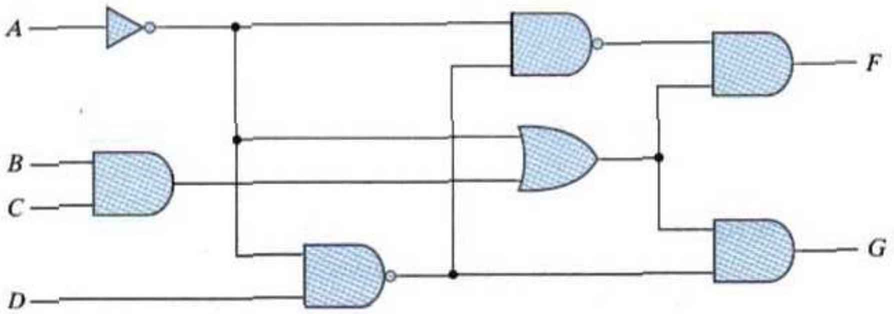


FIGURE P4.2

4.3 For the circuit shown in Fig. 4.26 (Section 4.11),

(a) Write the Boolean functions for the four outputs in terms of the input variables.

(b)* If the circuit is listed in a truth table, how many rows and columns would there be in the table?

4.4 Design a combinational circuit with three inputs and one output.

(a)* The output is 1 when the binary value of the inputs is less than 3. The output is 0 otherwise.

(b) The output is 1 when the binary value of the inputs is an odd number.

4.5 Design a combinational circuit with three inputs, x, y, and z, and three outputs, A, B, and C. When the binary input is 0, 1, 2, or 3, the binary output is two greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is three less than the input.

4.6 A majority circuit is a combinational circuit whose output is equal to 1 if the input variables have more 1's than 0's. The output is 0 otherwise.

(a)* Design a three-input majority circuit by finding the circuits truth table, Boolean equation, and a logic diagram.

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

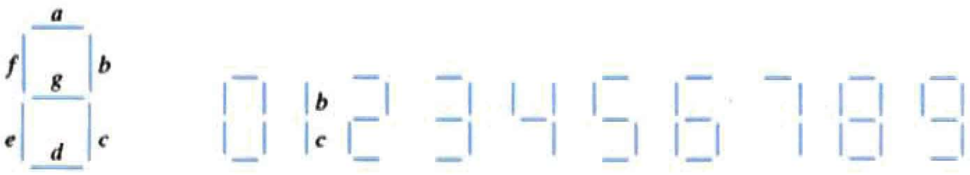
4.7 Design a combinational circuit that converts a four-bit Gray code (Table 1-6) to a four-bit binary number.

(a)* Implement the circuit with exclusive-OR gates.

(b) Using a case statement, write and verify a Verilog model of the circuit.

4.8* Design a code converter that converts a decimal digit from the 8, 4, −2, −1 code to BCD (see Table 1.5). (HDL — see Problem 4.50.)

4.9 An ABCD-to-seven-segment decoder is a combinational circuit that converts a decimal digit in BCD to an appropriate code for the selection of segments in an indicator used to display the decimal digit in a familiar form. The seven outputs of the decoder *(a, b, c, d, e, f, g)* select the corresponding segments in the display, as shown in Fig. P4.9(a). The numeric display chosen to represent the decimal digit is shown in Fig. P4.9(b). Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates. The six invalid combinations should result in a blank display. (HDL—see Problem 4.51.)



(a) Segment designation

(b) Numerical designation for display

FIGURE P4.9

4.10* Design a four-bit combinational circuit 2's complementer. (The output generates the 2's complement of the input binary number.) Show that the circuit can be constructed with exclusive-OR gates. Can you predict what the output functions are for a five-bit 2's complementer?

4.11 Using four half-adders (HDL — see Problem 4.52).

(a) Design a four-bit combinational circuit incrementer (a circuit that adds 1 to a four-bit binary number).

(b) Design a four-bit combinational circuit decrementer (a circuit that subtracts 1 from a four-bit binary number).

4.12 (a) Design a half-subtractor circuit with inputs x and y and outputs *Diff* and B_{out}. The circuit subtracts the bits $x - y$ and places the difference in *Diff* and the borrow in B_{out}.

(b)* Design a full-subtractor circuit with three inputs, x, y, B_{in}, and two outputs *Diff* and B_{out}. The circuit subtracts $x - y - B_{in}$, where B_{in} is the input borrow, B_{out} is the output borrow, and *Diff* is the difference.

4.13* The adder-subtractor circuit of Fig. 4.13 has the following values for mode input M and data inputs A and B:

	M	A	B
(a)	0	0111	0110
(b)	0	1000	1001
(c)	1	1100	1000
(d)	1	0101	1010
(e)	1	0000	0001

In each case, determine the values of the four SUM outputs, the carry C, and overflow V. (HDL—see Problems 4.37 and 4.40.)

4.14* Assume that the exclusive-OR gate has a propagation delay of 10 ns and that the AND or OR gates have a propagation delay of 5 ns. What is the total propagation delay time in the four-bit adder of Fig. 4.12?

4.15 Derive the two-level Boolean expression for the output carry C_4 shown in the lookahead carry generator of Fig. 4.12.

4.16 Define the carry propagate and carry generate as

$$P_i = A_i + B_i$$
$$G_i = A_i B_i$$

respectively. Show that the output carry and output sum of a full adder becomes

$$C_{i+1} = (C_i'G_i' + P_i')'$$
$$S_i = (P_iG_i') \oplus C_i$$

The logic diagram of the first stage of a four-bit parallel adder as implemented in IC type 74283 is shown in Fig. P4.16. Identify the P_i' and G_i' terminals and show that the circuit implements a full adder.

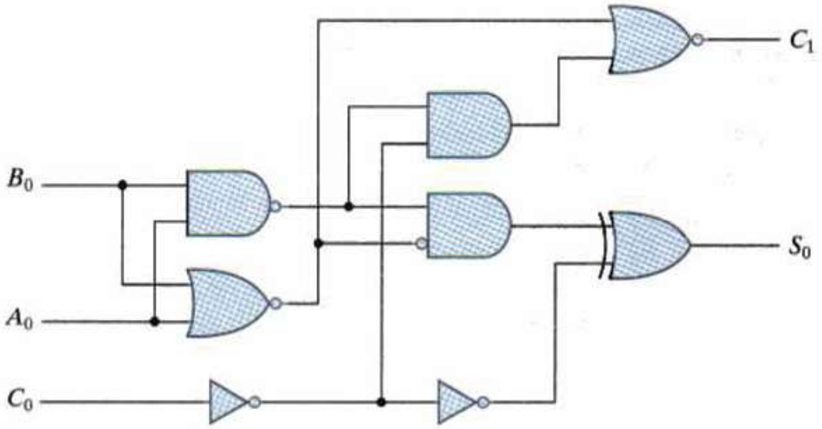


FIGURE P4.16
First stage of a parallel adder

4.17 Show that the output carry in a full-adder circuit can be expressed in the AND-OR-INVERT form

$$C_{i+1} = G_i + P_iC_i = (G_i'P_i' + G_i'C_i')'$$

IC type 74182 is a lookahead carry generator circuit that generates the carries with AND-OR-INVERT (AOI) gates (see Section 3-8.) The circuit assumes that the input terminals have the complements of the G's, the P's, and of C_i. Derive the Boolean functions for the lookahead carries C_2, C_3, and C_4 in this IC. (*Hint:* Use the equation-substitution method to derive the carries in terms of C_i')

4.18* Design a combinational circuit that generates the 9's complement of a BCD digit. (HDL — see Problem 4.54.)

4.19 Construct a BCD adder-subtractor circuit. Use the BCD adder of Fig. 4.14 and the 9's complementer of Problem 4.18. Use block diagrams for the components. (HDL — see Problem 4.55.)

4.20 A binary multiplier multiplies two unsigned four-bit numbers.

(a) Using AND gates and binary adders (see Fig. 4.16), design the circuit.

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

4.21 Design a combinational circuit that compares two four-bit numbers to check if they are equal. The circuit output is equal to 1 if the two numbers are equal and 0 otherwise.

4.22* Design an excess-3-to-binary decoder using the unused combinations of the code as don't-care conditions. (HDL — see Problem 4.42.)

4.23 Draw the logic diagram of a two-to-four-line decoder using (a) NOR gates only, and (b) NAND gates only. Include an enable input.

4.24 Design a BCD-to-decimal decoder using the unused combinations of the BCD code as don't-care conditions. (HDL — see Problem 4.60.)

4.25 Construct a 5-to-32-line decoder with four 3-to-8-line decoders with enable and a 2-to-4-line decoder. Use block diagrams for the components.

4-26 Construct a 4-to-16-line decoder with five 2-to-4-line decoders with enable.

4.27 A combinational circuit is specified by the following three Boolean functions:

$$F_1(A, B, C) = \Sigma(3, 5, 6)$$
$$F_2(A, B, C) = \Sigma(1, 4)$$
$$F_3(A, B, C) = \Sigma(2, 3, 5, 6, 7)$$

Implement the circuit with a decoder constructed with NAND gates (similar to Fig. 4.19) and NAND or AND gates connected to the decoder outputs. Use a block diagram for the decoder. Minimize the number of inputs in the external gates.

4.28 Using a decoder and external gates, design the combinational circuit defined by the following three Boolean functions:

(a) $F_1 = x'y'z' + xz$
$F_2 = xy'z' + x'y$
$F_3 = x'y'z + xy$

(b) $F_1 = (y' + x)z$
$F_2 = y'z' + xy' + yz'$
$F_3 = (x' + y)z$

4.29* Design a four-input priority encoder with inputs as in Table 4.8, but with input D_0 having the highest priority and input D_3 the lowest priority. (HDL—see Problem 4.57.)

4.30 Specify the truth table of an octal-to-binary priority encoder. Provide an output V to indicate that at least one of the inputs is present. The input with the highest subscript number has the highest priority. What will be the value of the four outputs if inputs D_2 and D_6 are 1 at the same time?

4.31 Construct a 16 × 1 multiplexer with two 8 × 1 and one 2 × 1 multiplexers. Use block diagrams.

4.32 Implement the following Boolean function with a multiplexer (HDL—see Problem 4.46):

(a) $F(A, B, C, D) = \Sigma(0, 2, 5, 7, 11, 14)$
(b) $F(A, B, C, D) = \Pi(3, 8, 12)$

4.33 Implement a full adder with two 4 × 1 multiplexers.

4.34 An 8 × 1 multiplexer has inputs A, B, and C connected to the selection inputs S_2, S_1, and S_0, respectively. The data inputs I_0 through I_7 are as follows:

(a)* $I_1 = I_2 = I_7 = 0; I_3 = I_5 = 1; \mathrm{I}_0 = I_4 = D$; and $\mathrm{I}_6 = D'$.
(b) $I_1 = I_2 = 0; I_3 = I_7 = 1; I_4 = I_5 = D$; and $I_0 = \mathrm{I}_6 = D'$.

Determine the Boolean function that the multiplexer implements.

4.35 Implement the following Boolean function with a 4 × 1 multiplexer and external gates.

(a)* $F(A, B, C, D) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$
(b) $F(A, B, C, D) = \Sigma(1, 2, 4, 7, 8, 9, 10, 11, 13, 15)$

Connect inputs A and B to the selection lines. The input requirements for the four data lines will be a function of variables C and D. These values are obtained by expressing F as a function of C and D for each of the four cases when $AB = 00, 01, 10,$ and 11. The functions may have to be implemented with external gates and with connections to power and ground.

4.36 Write the HDL gate-level description of the priority encoder circuit shown in Fig. 4.23. (HDL—see Problem 4.45.)

4.37 Write the HDL gate-level hierarchical description of a four-bit adder-subtractor for unsigned binary numbers. The circuit is similar to Fig. 4.13 but without output V. You can instantiate the four-bit full adder described in HDL Example 4.2. (See Problems 4.13 and 4.40.)

4.38 Write the HDL dataflow description of a quadruple two-to-one-line multiplexer with enable. (See Fig. 4.26.)

4.39* Write an HDL behavioral description of a four-bit comparator with a six-bit output Y[5:0]. Bit 5 of Y is for "equals," bit 4 is for "not equal to," bit 3 is for "greater than," bit 2 is for "less than," bit 1 for "greater than or equal to," and bit 0 for "less than or equal to."

4.40 Using the conditional operator (? :), write an HDL dataflow description of a four-bit adder subtractor of unsigned numbers. (See Problems 4.13 and 4.37.)

4.41 Repeat Problem 4.40, using a cyclic behavior.

4.42 (a) Write an HDL gate-level description of the BCD-to-excess-3 converter circuit shown in Fig. 4.4 (see Problem 4.22).
(b) Write a dataflow description of the BCD-to-excess-3 converter, using the Boolean expressions listed in Fig. 4.3.