Figure 4.1. The function $f(x_1, x_2, x_3) = \sum m(0, 2, 4, 5, 6)$.

Figure 4.2. Location of two-variable minterms.

Figure 4.3. A simple logic function.

Figure 4.4. Location of three-variable minterms.

Figure 4.5. Examples of three-variable Karnaugh maps.

Figure 4.6. A four-variable Karnaugh map.
Figure 4.13. POS minimization of $f = \Pi M(4, 5, 6)$.

Figure 4.14. POS minimization of $f = \Pi M(0, 1, 4, 8, 9, 12, 15)$.

Figure 4.15. Two implementations of $f = \Sigma m(2, 4, 5, 6) + D(12, 13, 14, 15)$.

Figure 4.16. An example of multiple-output synthesis.

Figure 4.18. Implementation in a CPLD.

Figure 4.19. Implementation in an FPGA.
Figure 4.20. Using 4-input AND gates to realize a 7-input product term.

Figure 4.21. A factored circuit.

Figure 4.22. A multilevel circuit.

Figure 4.23. A multilevel circuit.

Figure 4.24. The structure of a decomposition.

Figure 4.26a. Implementation of XOR.
\( f = x_1 \oplus x_2 = x_1 x_2 + x_1 \bar{x}_2 + x_2 \bar{x}_1 \)  

(c) Optimal NAND gate implementation

Figure 4.26b. Implementation of XOR.

\[ f = \overline{x_1} \oplus \overline{x_2} = x_1 \bar{x}_2 + x_2 \bar{x}_1 \]

Figure 4.29. Circuit for Example 4.10.

Figure 4.30. Circuit for Example 4.11.

Figure 4.31. Circuit for Example 4.12.

Figure 4.32. Circuit for Example 4.13.

Figure 4.33. Representation of \( f(x_1, x_2) = \sum m(1, 2, 3) \).
Figure 4.34. Representation of $f(x_1, x_2, x_3) = \sum m(0, 2, 4, 5, 6)$.

Figure 4.35. Representation of $f = \sum m(0, 2, 3, 6, 7, 8, 10, 15)$.

Figure 4.36. Generation of prime implicants for $f = \sum m(0, 4, 8, 10, 11, 12, 13, 15)$.

Figure 4.38. Generation of prime implicants for $f = \sum m(0, 2, 5, 6, 7, 8, 9, 13) + D(1, 2, 15)$.

Figure 4.41. The coordinate *-operation.

Figure 4.42. The coordinate #-operation.
module func1 (x1, x2, x3, f);
  input x1, x2, x3;
  output f;
  assign f = (~x1 & ~x2 & x3) | (x1 & ~x2 & ~x3) | (x1 & x2 & ~x3);
endmodule
Figure 4.50. Implementation of the Verilog code in Figure 4.49.

Figure 4.51. Implementation using XOR synthesis ($f = x_3 \oplus x_1 x_2 x_4$).

Figure 4.52. Verilog code in Figure 4.49 implemented in a LUT.

Figure 4.53. Verilog code for $f_1$ in Figure 4.7.

module example4_22 (x1, x2, x3, x4, f);
  input x1, x2, x3, x4;
  output f;
  assign f = (~x1 & ~x2 & x3 & ~x4) | (~x1 & ~x2 & x3 & x4) |
            (x1 & ~x2 & ~x3 & x4) | (x1 & ~x2 & x3 & ~x4) |
            (x1 & ~x2 & x3 & x4) | (x1 & x2 & ~x3 & x4) |
            (x1 & ~x2 & ~x3 & x4) | (x1 & x2 & ~x3 & x4) |
            (x1 & x2 & ~x3 & x4) | (x1 & x2 & ~x3 & x4) ;
endmodule

Figure 4.54. Verilog code for the function of section 4.6.

module example4_23 (x1, x2, x3, x4, x5, x6, x7, f);
  input x1, x2, x3, x4, x5, x6, x7;
  output f;
  assign f = (x1 & x3 & ~x6) | (x1 & x4 & x5 & ~x6) |
            (x2 & x3 & x7) | (x2 & x4 & x5 & x7) ;
endmodule

Figure 4.55. Two implementations of a 7-variable function.
Figure P4.1. Expansion of implicant \( x_1x_2x_3 \).

Figure P4.2. Circuit for problem 4.33.

Figure P4.3. Circuit for problem 4.34.