GLOBAL EDITION

Logic and Computer Design Fundamentals

FIFTH EDITION

Morris Mano • Charles R. Kime • Tom Martin

ALWAYS LEARNING

PEARSON

Logic and Computer Design Fundamentals

Fifth Edition
Global Edition

M. Morris Mano
California State University, Los Angeles

Charles R. Kime
University of Wisconsin, Madison

Tom Martin
Virginia Tech

PEARSON

Boston Columbus Indianapolis New York San Francisco Hoboken
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

3-7. +A traffic light control at a simple intersection uses a binary counter to produce the following sequence of combinations on lines A, B, C, and D: 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000. After 1000, the sequence repeats, beginning again with 0000, forever. Each combination is present for 5 seconds before the next one appears. These lines drive combinational logic with outputs to lamps RNS (red—north/south), YNS (yellow—north/south), GNS (green—north/south), REW (red—east/west), YEW (yellow—east/west), and GEW (green—east/west). The lamp controlled by each output is ON for a 1 applied and OFF for a 0 applied. For a given direction, assume that green is on for 30 seconds, yellow for 5 seconds, and red for 45 seconds. (The red intervals overlap for 5 seconds.) Divide the 80 seconds available for the cycle through the 16 combinations into 16 intervals and determine which lamps should be lit in each interval based on expected driver behavior. Assume that, for interval 0000, a change has just occurred and that GNS = 1, REW = 1, and all other outputs are 0. Design the logic to produce the six outputs using AND and OR gates and inverters.

3-8. Design a combinational circuit that accepts a 3-bit number and generates a 4-bit binary number output equal to double of the input number.

3-9. +Design a combinational circuit that accepts a 4-bit number and generates a 3-bit binary number output that approximates the square root of the number. For example, if the square root is 3.5 or larger, give a result of 4. If the square root is < 3.5 and ≥ 2.5, give a result of 3.

3-10. Design a circuit with a 3-bit input A, B, and C, in the form of Gray code, that produces a 3-bit output 0_0, 0_1, and 0_2 in binary form. For example, if the Gray code inputs are 001 and 011, then the circuit will produce 001 and 010, respectively.

3-11. A traffic metering system for controlling the release of traffic from an entrance ramp onto a superhighway has the following specifications for a part of its controller. There are three parallel metering lanes, each with its own stop (red)–go (green) light. One of these lanes, the car pool lane, is given priority for a green light over the other two lanes. Otherwise, a "round robin" scheme in which the green lights alternate is used for the other two (left and right) lanes. The part of the controller that determines which light is to be green (rather than red) is to be designed. The specifications for the controller follow:

Inputs

PS	Car pool lane sensor (car present—1; car absent—0)
LS	Left lane sensor (car present—1; car absent—0)
RS	Right lane sensor (car present—1; car absent—0)
RR	Round robin signal (select left—1; select right—0)

Outputs

PL	Car pool lane light (green—1; red—0)
LL	Left lane light (green—1; red—0)
RL	Right lane light (green—1; red—0)

Operation

1. If there is a car in the car pool lane, PL is 1.
2. If there are no cars in the car pool lane and the right lane, and there is a car in the left lane, LL is 1.
3. If there are no cars in the car pool lane and in the left lane, and there is a car in the right lane, RL is 1.
4. If there is no car in the car pool lane, there are cars in both the left and right lanes, and RR is 1, then LL = 1.
5. If there is no car in the car pool lane, there are cars in both the left and right lanes, and RR is 0, then RL = 1.
6. If any PL, LL, or RL is not specified to be 1 above, then it has value 0.

(a) Find the truth table for the controller part.

(b) Find a minimum multiple-level gate implementation with minimum gate-input cost using AND gates, OR gates, and inverters.

3-12. Complete the design of the BCD–to–seven-segment decoder by performing the following steps:

(a) Plot the seven maps for each of the outputs for the BCD–to–seven-segment decoder specified in Table 3-9.

(b) Simplify the seven output functions in sum-of-products form, and determine the total number of gate inputs that will be needed to implement the decoder.

(c) Verify that the seven output functions listed in the text give a valid simplification. Compare the number of gate inputs with that obtained in part (b) and explain the difference.

3-13. Design a circuit to implement the following pair of Boolean equations:

$$F_0 = Z(XY + \overline{Y}\,\overline{X} + \overline{Z}(X\overline{Y} + \overline{X}Y)$$

$$F_1 = \overline{W}(X\overline{Y} + \overline{X}Y) + W(XY + \overline{Y}\,\overline{X})$$

To simplify drawing the schematic, the circuit is to use a hierarchy based on the factoring shown in the equation. Three instances (copies) of a single hierarchical circuit component made up of two AND gates, an OR gate, and an inverter are to be used. Draw the logic diagram for the hierarchical component and for the overall circuit diagram using a symbol for the hierarchical component.

3-14. A hierarchical component with the function is to be used along with inverters to implement the following equation:

$$F_0 = \overline{Z}(X\overline{Y} + \overline{X}Y) + Z(XY + \overline{X}\,\overline{Y})$$

$$F_1 = \overline{X}(\overline{W}Z + \overline{Z}W) + X(WZ + \overline{W}\,\overline{Z})$$

The overall circuit can be obtained by using Shannon's expansion theorem,

$$F = \bar{X} \cdot F_0(X) + X \cdot F_1(X)$$

where F_0 is F evaluated with variable $X = 0$ and F_0 is F evaluated with variable $X = 1$. This expansion F can be implemented with function H by letting $Y = F_0$ and $Z = F_1$. The expansion theorem can then be applied to each of F_0 and F_1 using a variable in each, preferably one that appears in both true and complemented form. The process can then be repeated until all F_i's are single literals or constants. For F_1, use $X = A$ to find G_0 and G_1 and then use $X = B$ for F_1 and F_1. Draw the top-level diagram for G using H as a hierarchical component.

3-15. +A NAND gate with eight inputs is required. For each of the following cases, minimize the number of gates used in the multiple-level result:

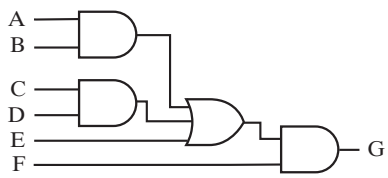
(a) Design the 8-input NAND gate using 2-input NAND gates and NOT gates.

(b) Design the 8-input NAND gate using 2-input NAND gates, 2-input NOR gates, and NOT gates only if needed.

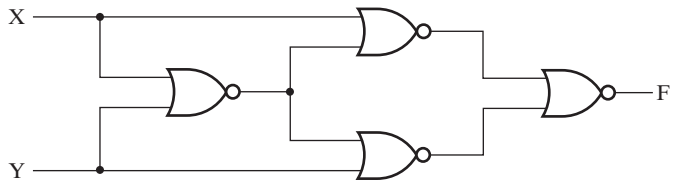
(c) Compare the number of gates used in (a) and (b).

3-16. Perform technology mapping to NAND gates for the circuit in Figure 3-54. Use cell types selected from: Inverter ($n = 1$), 2NAND, 3NAND, and 4NAND, as defined at the beginning of Section 3-2.

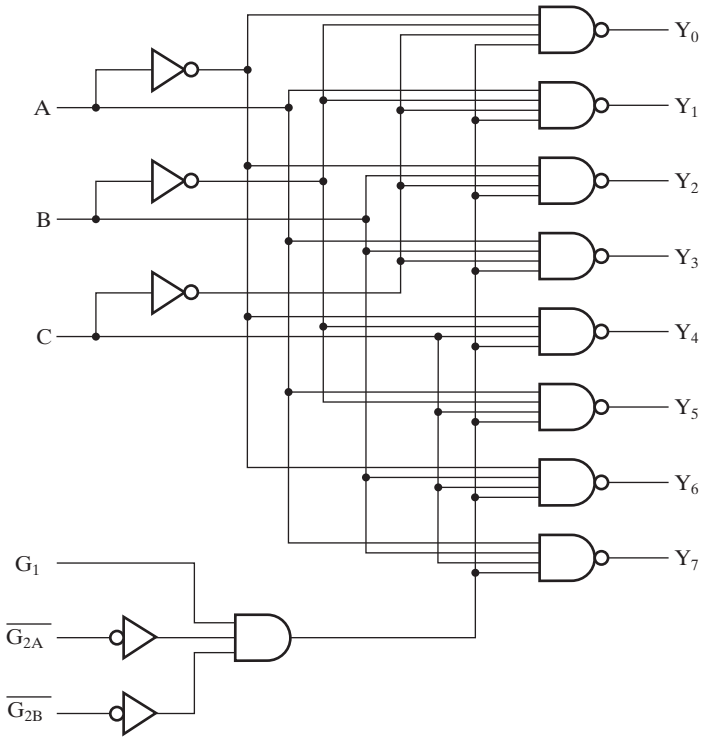
3-17. Repeat Problem 3-16, using NOR gate cell types selected from: Inverter ($n = 1$), 2NOR, 3NOR, and 4NOR, each defined in the same manner as the corresponding four NAND cell types at the beginning of Section 3-2.



☐ **FIGURE 3-54**
Circuit for Problems 3-16 and 3-17



☐ **FIGURE 3-55**
Circuit for Problem 3-20



□ **FIGURE 3-56**
Circuit for Problems 3-21 and 3-22

3-18. **(a)** Repeat Problem 3-16 for the Boolean equations for the segments *a* and *c* of the BCD to seven-segment decoder from Example 3-18. Share common terms where possible.

(b) Repeat part (a) using only Inverter ($n = 1$) and 2NAND cell types.

3-19. **(a)** Repeat Problem 3-18, mapping to NOR gate cell types as in Problem 3-17. Share common terms where possible.

(b) Repeat part (a) using only Inverter ($n = 1$) and 2NOR cell types.

3-20. By using manual methods, verify that the circuit of Figure 3-55 generates the exclusive-NOR function.

3-21. The logic diagram for a 74HC138 MSI CMOS circuit is given in Figure 3-56. Find the Boolean function for each of the outputs. Describe the circuit function carefully.

3-22. Do Problem 3-21 by using logic simulation to find the output waveforms of the circuit or a partial truth-table listing, rather than finding Boolean functions.

3-23. **(a)** Use logic simulation to verify that the circuits described in Example 3-18 implement the BCD–to–seven-segment converter correctly.

(b) Design the converter assuming that the unused input combinations (minterms 10–15) can be don't cares rather than 0s. Simulate your design and compare it to your simulation from part (a).

3-24. ***(a)** Draw an implementation diagram for a constant vector function $F = (F_7, F_6, F_5, F_4, F_3, F_2, F_1, F_0) = (1,0,0,1,0,1,1,0)$ using the ground and power symbols in Figure 3-7(b).

(b) Draw an implementation diagram for a rudimentary vector function $G = (G_7, G_6, G_5, G_4, G_3, G_2, G_1, G_0) = (A, \overline{A}, 0, 1, \overline{A}, A, 1, 1)$ using inputs $1, 0, A$, and $\overline{A}$.

3-25. **(a)** Draw an implementation diagram for rudimentary vector function $F = (F_7, F_6, F_5, F_4, F_3, F_2, F_1, F_0) = (A, \overline{A}, 1, \overline{A}, A, 0, 1, \overline{A})$, using the ground and power symbols in Figure 3-7(b) and the wire and inverter in Figures 3-7(c) and (d).

(b) Draw an implementation diagram for rudimentary vector function $G = (G_7, G_6, G_5, G_4, G_3, G_2, G_1, G_0) = (\overline{F}_0, \overline{F}_1, F_3, \overline{F}_2, 1, 0, 0, 1)$, using the ground and power symbols and components of vector F.

3-26. **(a)** Draw an implementation diagram for the vector $G = (G_0, G_1, G_2, G_3, G_4, G_5, G_6, G_7) = (F_4, F_5, F_6, F_7, F_0, F_2, F_1, F_3)$

(b) Draw a simple implementation for the rudimentary vector $H = (H_7, H_6, H_5, H_4, H_3, H_2, H_1, H_0) = (F_0, F_1, G_3, G_2, G_1, G_0, F_3, F_4)$

3-27. A home security system has a master switch that is used to enable an alarm, lights, video cameras, and a call to local police in the event one or more of six sets of sensors detects an intrusion. In addition there are separate switches to enable and disable the alarm, lights, and the call to local police. The inputs, outputs, and operation of the enabling logic are specified as follows:

Inputs

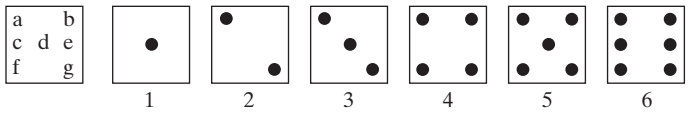
$S_i, i = 0, 1, 2, 3, 4, 5$: signals from six sensor sets (0 = intrusion detected, 1 = no intrusion detected)
M: master switch (0 = security system enabled, 1 = security system disabled)
A: alarm switch (0 = alarm disabled, 1 = alarm enabled)
L: light switch (0 = lights disabled, 1 = lights enabled)
P: police switch (0 = police call disabled, 1 = police call enabled)

Outputs

A: alarm (0 = alarm on, 1 = alarm off)
L: lights (0 = lights on, 1 = lights off)
V: video cameras (0 = video cameras off, 1 = video cameras on)
C: call to police (0 = call off, 1 = call on)

Operation

If one or more of the sets of sensors detect an intrusion and the security system is enabled, then outputs activate based on the outputs of the remaining switches. Otherwise, all outputs are disabled.



□ **FIGURE 3-57**
Patterns for Dice for Problem 3-32

Find a minimum-gate-input cost realization of the enabling logic using AND and OR gates and inverters.

3-28. Design a 3–to–8-line decoder using two 2–to–4-line decoders and eight 2-input AND gates.

3-29. Design a 4–to–16-line decoder with enable using two 3–to–8-line decoders with enable and two AND gates and one OR gate.

3-30. *Design a 5–to–32-line decoder using a 3–to–8-line decoder, a 2–to–4-line decoder, and 32 2-input AND gates.

3-31. A special 3–to–16-line decoder is to be designed. The input codes used are in BCD format, i.e., from 000 to 1001. For a given code applied, the output D_i, with i equal to the decimal equivalent of the code, is 1 and all other outputs are 0. Design the decoder with a 3–to–8-line decoder, using AND gates and a NOT gate.

3-32. An electronic game uses an array of seven LEDs (light-emitting diodes) to display the results of a random roll of a die. A decoder is to be designed to illuminate the appropriate diodes for the display of each of the six die values. The desired display patterns are shown in Figure 3-57.

(a) Use a 3–to–8-line decoder and OR gates to map the 3-bit combinations on inputs X_2, X_1, and X_0 for values 1 through 6 to the outputs a through g. Input combinations 000 and 111 are don't-cares.

(b) Note that for the six die sides, only certain combinations of dots occur. For example, dot pattern $A = \{d\}$ and dot pattern $B = \{a, g\}$ can be used for representing input values 1, 2, and 3 as $\{A\}$, $\{B\}$, and $\{A, B\}$. Define four dot patterns A, B, C, and D, sets of which can provide all six output patterns. Design a minimized custom decoder that has inputs X_2, X_1, and X_0 and outputs A, B, C, and D, and compare its gate-input cost to that of the 3-to-8 decoder and OR gates in part (a).

3-33. Draw the detailed logic diagram of a 2–to–4-line decoder using only NAND gates. Include an enable input.

3-34. To provide uphill running and walking, an exercise treadmill has a grade feature that can be set from 0.0% to 15.0% in increments of 0.1%. (The grade in percent is the slope expressed as a percentage. For example, a slope of 0.10 is a grade of 10%.) The treadmill has a 10 high by 20 wide LCD dot array showing a plot of the grade versus time. This problem concerns only the vertical dimension of the display.