

Exercises

Exercise 2.1 Write a Boolean equation in sum-of-products canonical form for each of the truth tables in Figure 2.80.

(a)			(b)				(c)				(d)					(e)				
A	B	Y	A	B	C	Y	A	B	C	Y	A	B	C	D	Y	A	B	C	D	Y
0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0
1	0	1	0	1	0	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0
1	1	1	0	1	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1
			1	0	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0
			1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	1
			1	1	0	0	0	1	1	0	0	1	1	0	0	0	1	1	0	1
			1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	0
											1	0	0	0	1	1	0	0	0	0
											1	0	0	1	0	1	0	0	1	1
											1	0	1	0	1	1	0	1	0	1
											1	0	1	1	0	1	0	1	1	0
											1	1	0	0	0	1	1	0	0	1
											1	1	0	1	0	1	1	0	1	0
											1	1	1	0	1	1	1	1	0	0
											1	1	1	1	0	1	1	1	1	1

Figure 2.80 Truth tables

Exercise 2.2 Write a Boolean equation in product-of-sums canonical form for the truth tables in Figure 2.80.

Exercise 2.3 Minimize each of the Boolean equations from Exercise 2.1.

Exercise 2.4 Sketch a reasonably simple combinational circuit implementing each of the functions from Exercise 2.3. Reasonably simple means that you are not wasteful of gates, but you don't waste vast amounts of time checking every possible implementation of the circuit either.

Exercise 2.5 Repeat Exercise 2.4 using only NOT gates and AND and OR gates.

Exercise 2.6 Repeat Exercise 2.4 using only NOT gates and NAND and NOR gates.

Exercise 2.7 Simplify the following Boolean equations using Boolean theorems. Check for correctness using a truth table or K-map.

(a) $Y = AC + \overline{A}BC$

(b) $Y = \overline{A}B + \overline{A}BC + \overline{(A + C)}$

(c) $Y = \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C} + \overline{A}B\overline{C}D + ABD + \overline{A}B\overline{C}D + B\overline{C}D + \overline{A}$

Exercise 2.8 Sketch a reasonably simple combinational circuit implementing each of the functions from Exercise 2.7.

Exercise 2.9 Simplify each of the following Boolean equations. Sketch a reasonably simple combinational circuit implementing the simplified equation.

(a) $Y = BC + \overline{A}BC + B\overline{C}$

(b) $Y = \overline{A + \overline{A}B + \overline{A}B} + \overline{A + B}$

(c) $Y = ABC + ABD + ABE + ACD + ACE + (\overline{A + D + E}) + \overline{B}CD + \overline{B}CE + \overline{B}DE + \overline{C}DE$

Exercise 2.10 Give an example of a truth table requiring between 3 billion and 5 billion rows that can be constructed using fewer than 40 (but at least 1) two-input gates.

Exercise 2.11 Give an example of a circuit with a cyclic path that is nevertheless combinational.

Exercise 2.12 Alyssa P. Hacker says that any Boolean function can be written in minimal sum-of-products form as the sum of all of the prime implicants of the function. Ben Bitdiddle says that there are some functions whose minimal equation does not involve all of the prime implicants. Explain why Alyssa is right or provide a counterexample demonstrating Ben's point.

Exercise 2.13 Prove that the following theorems are true using perfect induction. You need not prove their duals.

- (a) The idempotency theorem (T3)
- (b) The distributivity theorem (T8)
- (c) The combining theorem (T10)

Exercise 2.14 Prove De Morgan's Theorem (T12) for three variables, B_2, B_1, B_0 , using perfect induction.

Exercise 2.15 Write Boolean equations for the circuit in Figure 2.81. You need not minimize the equations.

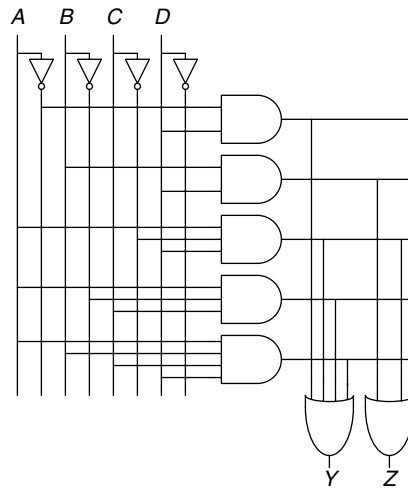


Figure 2.81 Circuit schematic

Exercise 2.16 Minimize the Boolean equations from Exercise 2.15 and sketch an improved circuit with the same function.

Exercise 2.17 Using De Morgan equivalent gates and bubble pushing methods, redraw the circuit in Figure 2.82 so that you can find the Boolean equation by inspection. Write the Boolean equation.

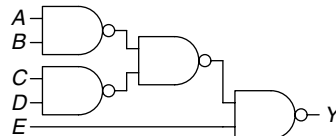


Figure 2.82 Circuit schematic

Exercise 2.18 Repeat Exercise 2.17 for the circuit in Figure 2.83.

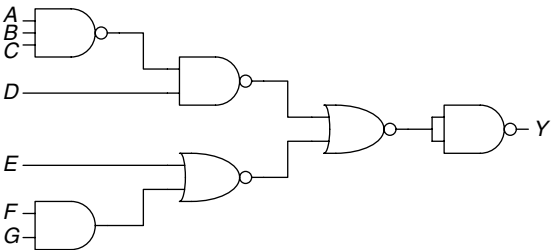


Figure 2.83 Circuit schematic

Exercise 2.19 Find a minimal Boolean equation for the function in Figure 2.84. Remember to take advantage of the don't care entries.

A	B	C	D	Y
0	0	0	0	X
0	0	0	1	X
0	0	1	0	X
0	0	1	1	0
0	1	0	0	0
0	1	0	1	X
0	1	1	0	0
0	1	1	1	X
1	0	0	0	1
1	0	0	1	0
1	0	1	0	X
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	X
1	1	1	1	1

Figure 2.84 Truth table

Exercise 2.20 Sketch a circuit for the function from Exercise 2.19.

Exercise 2.21 Does your circuit from Exercise 2.20 have any potential glitches when one of the inputs changes? If not, explain why not. If so, show how to modify the circuit to eliminate the glitches.

Exercise 2.22 Ben Bitdiddle will enjoy his picnic on sunny days that have no ants. He will also enjoy his picnic any day he sees a hummingbird, as well as on days where there are ants and ladybugs. Write a Boolean equation for his enjoyment (E) in terms of sun (S), ants (A), hummingbirds (H), and ladybugs (L).

Exercise 2.23 Complete the design of the seven-segment decoder segments S_c through S_g (see Example 2.10):

- Derive Boolean equations for the outputs S_c through S_g assuming that inputs greater than 9 must produce blank (0) outputs.
- Derive Boolean equations for the outputs S_c through S_g assuming that inputs greater than 9 are don't cares.
- Sketch a reasonably simple gate-level implementation of part (b). Multiple outputs can share gates where appropriate.

Exercise 2.24 A circuit has four inputs and two outputs. The inputs, $A_{3:0}$, represent a number from 0 to 15. Output P should be TRUE if the number is prime (0 and 1 are not prime, but 2, 3, 5, and so on, are prime). Output D should be TRUE if the number is divisible by 3. Give simplified Boolean equations for each output and sketch a circuit.

Exercise 2.25 A *priority encoder* has 2^N inputs. It produces an N -bit binary output indicating the most significant bit of the input that is TRUE, or 0 if none of the inputs are TRUE. It also produces an output *NONE* that is TRUE if none of the input bits are TRUE. Design an eight-input priority encoder with inputs $A_{7:0}$ and outputs $Y_{2:0}$ and *NONE*. For example, if the input is 00100000, the output Y should be 101 and *NONE* should be 0. Give a simplified Boolean equation for each output, and sketch a schematic.

Exercise 2.26 Design a modified priority encoder (see Exercise 2.25) that receives an 8-bit input, $A_{7:0}$, and produces two 3-bit outputs, $Y_{2:0}$ and $Z_{2:0}$. Y indicates the most significant bit of the input that is TRUE. Z indicates the second most significant bit of the input that is TRUE. Y should be 0 if none of the inputs are TRUE. Z should be 0 if no more than one of the inputs is TRUE. Give a simplified Boolean equation for each output, and sketch a schematic.

Exercise 2.27 An M -bit *thermometer code* for the number k consists of k 1's in the least significant bit positions and $M - k$ 0's in all the more significant bit positions. A *binary-to-thermometer code converter* has N inputs and $2^N - 1$ outputs. It produces a $2^N - 1$ bit thermometer code for the number specified by the input. For example, if the input is 110, the output should be 0111111. Design a 3:7 binary-to-thermometer code converter. Give a simplified Boolean equation for each output, and sketch a schematic.

Exercise 2.28 Write a minimized Boolean equation for the function performed by the circuit in Figure 2.85.

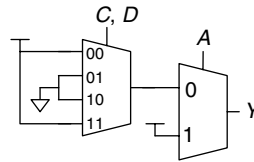


Figure 2.85 Multiplexer circuit

Exercise 2.29 Write a minimized Boolean equation for the function performed by the circuit in Figure 2.86.

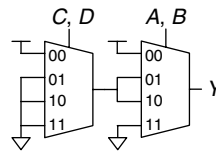


Figure 2.86 Multiplexer circuit

Exercise 2.30 Implement the function from Figure 2.80(b) using

- (a) an 8:1 multiplexer
- (b) a 4:1 multiplexer and one inverter
- (c) a 2:1 multiplexer and two other logic gates

Exercise 2.31 Implement the function from Exercise 2.9(a) using

- (a) an 8:1 multiplexer
- (b) a 4:1 multiplexer and no other gates
- (c) a 2:1 multiplexer, one OR gate, and an inverter

Exercise 2.32 Determine the propagation delay and contamination delay of the circuit in Figure 2.83. Use the gate delays given in Table 2.8.

Table 2.8 Gate delays for Exercises 2.32–2.35

Gate	t_{pd} (ps)	t_{cd} (ps)
NOT	15	10
2-input NAND	20	15
3-input NAND	30	25
2-input NOR	30	25
3-input NOR	45	35
2-input AND	30	25
3-input AND	40	30
2-input OR	40	30
3-input OR	55	45
2-input XOR	60	40

Exercise 2.33 Sketch a schematic for a fast 3:8 decoder. Suppose gate delays are given in Table 2.8 (and only the gates in that table are available). Design your decoder to have the shortest possible critical path, and indicate what that path is. What are its propagation delay and contamination delay?

Exercise 2.34 Redesign the circuit from Exercise 2.24 to be as fast as possible. Use only the gates from Table 2.8. Sketch the new circuit and indicate the critical path. What are its propagation delay and contamination delay?

Exercise 2.35 Redesign the priority encoder from Exercise 2.25 to be as fast as possible. You may use any of the gates from Table 2.8. Sketch the new circuit and indicate the critical path. What are its propagation delay and contamination delay?

Exercise 2.36 Design an 8:1 multiplexer with the shortest possible delay from the data inputs to the output. You may use any of the gates from Table 2.7 on page 88. Sketch a schematic. Using the gate delays from the table, determine this delay.

Interview Questions

The following exercises present questions that have been asked at interviews for digital design jobs.

Question 2.1 Sketch a schematic for the two-input XOR function using only NAND gates. How few can you use?

Question 2.2 Design a circuit that will tell whether a given month has 31 days in it. The month is specified by a 4-bit input, $A_{3:0}$. For example, if the inputs are 0001, the month is January, and if the inputs are 1100, the month is December. The circuit output, Y , should be HIGH only when the month specified by the inputs has 31 days in it. Write the simplified equation, and draw the circuit diagram using a minimum number of gates. (Hint: Remember to take advantage of don't cares.)

Question 2.3 What is a tristate buffer? How and why is it used?

Question 2.4 A gate or set of gates is universal if it can be used to construct any Boolean function. For example, the set {AND, OR, NOT} is universal.

- (a) Is an AND gate by itself universal? Why or why not?
- (b) Is the set {OR, NOT} universal? Why or why not?
- (c) Is a NAND gate by itself universal? Why or why not?

Question 2.5 Explain why a circuit's contamination delay might be less than (instead of equal to) its propagation delay.