

3.1 Consider the circuit shown in Figure P3.1.

(a) Show the truth table for the logic function f .

(b) If each gate in the circuit is implemented as a CMOS gate, how many transistors are needed?

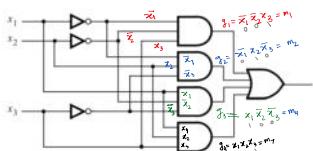


Figure P3.1 A sum-of-products CMOS circuit.

(a) Truth Table			
x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\text{Total Transistors} = 2 \times 3 + 8 \times 4 + 10 = 48$$

3.2 (a) Show that the circuit in Figure P3.2 is functionally equivalent to the circuit in Figure P3.1.

(b) How many transistors are needed to build this CMOS circuit?

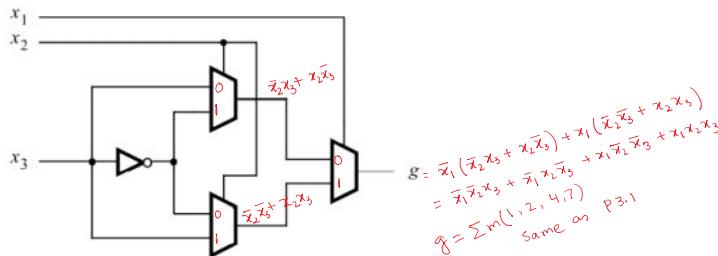


Figure P3.2 A CMOS circuit built with multiplexers.

3.3 (a) Show that the circuit in Figure P3.3 is functionally equivalent to the circuit in Figure P3.2.

(b) How many transistors are needed to build this CMOS circuit if each XOR gate is implemented using the circuit in Figure 3.61d?

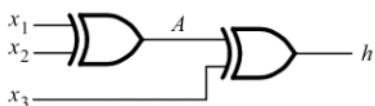


Figure P3.3 Circuit for problem 3.3.

$$\begin{aligned} A &= x_1 \oplus x_2 = x_1 \bar{x}_2 + \bar{x}_1 x_2 \quad \bar{A} = \overline{x_1 \oplus x_2} = x_1 x_2 + \bar{x}_1 \bar{x}_2 \\ h &= A \oplus x_3 = \bar{A} x_3 + A \bar{x}_3 \\ &= (\bar{x}_1 \bar{x}_2 + x_1 x_2) x_3 + (x_1 \bar{x}_2 + \bar{x}_1 x_2) \bar{x}_3 \\ &= \cancel{\bar{x}_1 \bar{x}_2 x_3} + \cancel{x_1 x_2 x_3} + \cancel{x_1 \bar{x}_2 \bar{x}_3} + \cancel{\bar{x}_1 x_2 \bar{x}_3} \\ &= \sum(1, 2, 4, 7) \Rightarrow \text{same as P3.1} \end{aligned}$$

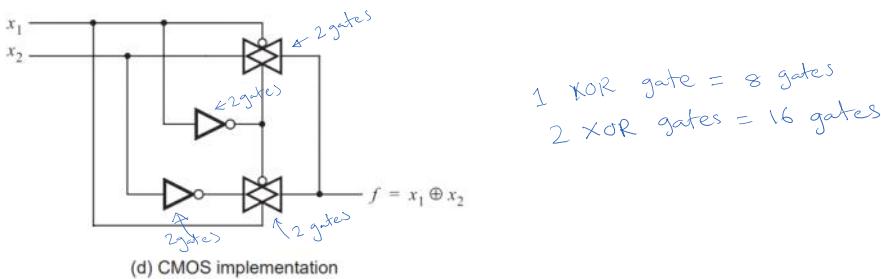
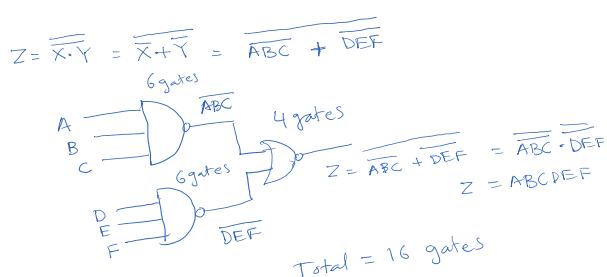
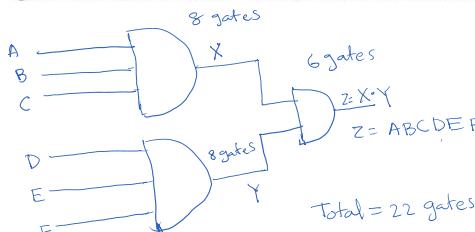


Figure 3.61 Exclusive-OR gate.

***3.4** In Section 3.8.8 we said that a six-input CMOS AND gate can be constructed using two three-input AND gates and a two-input AND gate. This approach requires 22 transistors. Show how you can use only CMOS NAND and NOR gates to build the six-input AND gate, and calculate the number of transistors needed. (Hint: use DeMorgan's theorem.)



- 3.9** Figure P3.7 shows half of a CMOS circuit. Derive the other half that contains the NMOS transistors.

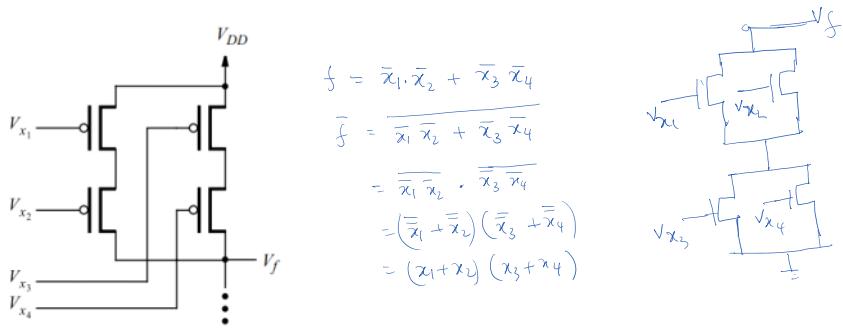
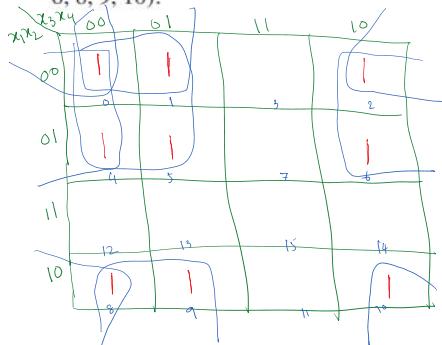


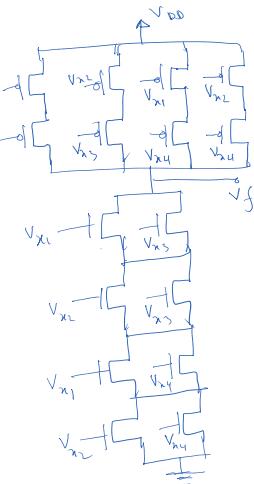
Figure P3.7 The PUN in a CMOS circuit.

- 3.10** Derive a CMOS complex gate for the logic function $f(x_1, x_2, x_3, x_4) = \sum m(0, 1, 2, 4, 5, 6, 8, 9, 10)$.

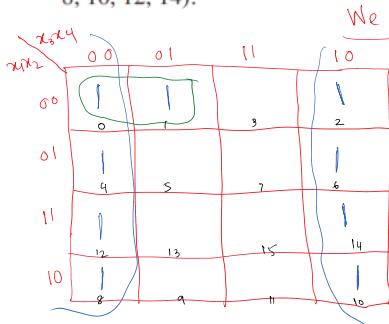


$$f = \bar{x}_1 \bar{x}_3 + \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_4 + \bar{x}_2 \bar{x}_4$$

$$\bar{f} = \frac{1}{\bar{x}_1 \bar{x}_3 + \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_4 + \bar{x}_2 \bar{x}_4} = \frac{1}{\bar{x}_1 \bar{x}_3} \cdot \frac{1}{\bar{x}_2 \bar{x}_3} \cdot \frac{1}{\bar{x}_1 \bar{x}_4} \cdot \frac{1}{\bar{x}_2 \bar{x}_4} = (x_1 + x_3)(x_2 + x_3)(x_1 + x_4)(x_2 + x_4)$$



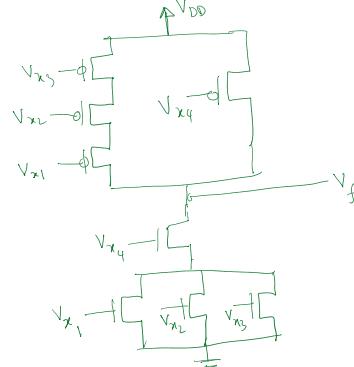
- 3.11** Derive a CMOS complex gate for the logic function $f(x_1, x_2, x_3, x_4) = \sum m(0, 1, 2, 4, 6, 8, 10, 12, 14)$.



We will use K-map

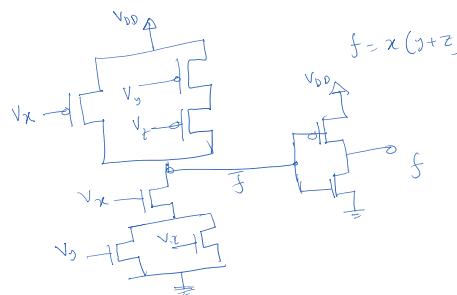
$$f = \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3$$

$$\bar{f} = \frac{1}{\bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3} = \bar{x}_4 \cdot \frac{1}{\bar{x}_1 \bar{x}_2 \bar{x}_3} = x_4 (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) = x_4 (x_1 + x_2 + x_3)$$



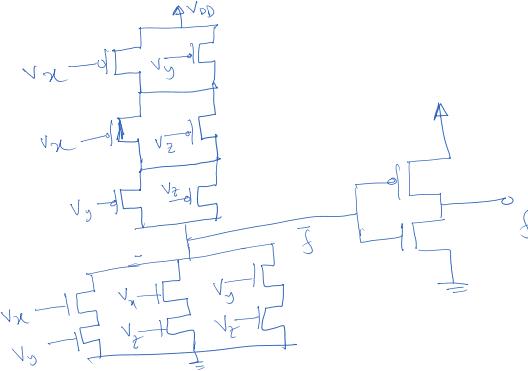
- *3.12** Derive a CMOS complex gate for the logic function $f = xy + xz$. Use as few transistors as possible (Hint: consider \bar{f}).

$$\bar{f} = \overline{xy + xz} = \overline{xy} \cdot \overline{xz} = (\bar{x} + \bar{y}) \cdot (\bar{x} + \bar{z}) = \bar{x} + \bar{x}\bar{z} + \bar{x}\bar{y} + \bar{y}\bar{z} = \bar{x}(1 + \bar{z} + \bar{y}) + \bar{y}\bar{z} = \bar{x} + \bar{y}\bar{z}$$



- 3.13** Derive a CMOS complex gate for the logic function $f = xy + xz + yz$. Use as few transistors as possible (Hint: consider \bar{f}).

$$\bar{f} = \overline{xy + xz + yz} = \overline{xy} \cdot \overline{xz} \cdot \overline{yz} = (\bar{x} + \bar{y})(\bar{x} + \bar{z})(\bar{y} + \bar{z})$$



- 3.36** Using the style of drawing in Figure 3.66, draw a picture of a PLA programmed to implement $f_1(x_1, x_2, x_3) = \sum m(1, 2, 4, 7)$. The PLA should have the inputs x_1, \dots, x_3 ; the product terms P_1, \dots, P_4 ; and the outputs f_1 and f_2 .

$$f_1 = \overline{x_1} \overline{x_2} x_3 + \overline{x_1} x_2 \overline{x_3} + x_1 \overline{x_2} \overline{x_3} + x_1 x_2 x_3$$

The function generally implement in AND-OR logic.

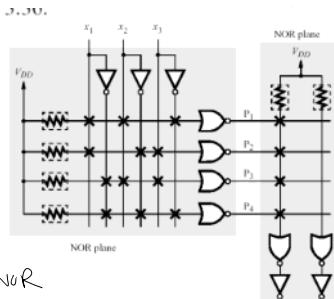
But to implement in NOR-NOR-NOT

logic, we should replace -

AND \rightarrow NOR

OR \rightarrow NOR + OR

where AND \Leftrightarrow inverted inputs NOR



$$\text{So, } P_1 = \overline{x_1 + x_2 + \overline{x_3}} \quad P_2 = \overline{x_1 + \overline{x_2} + x_3} \quad P_3 = \overline{\overline{x_1} + x_2 + x_3} \quad P_4 = \overline{\overline{x_1} + \overline{x_2} + \overline{x_3}}$$

- 3.37** Using the style of drawing in Figure 3.66, draw a picture of a PLA programmed to implement $f_1(x_1, x_2, x_3) = \sum m(0, 3, 5, 6)$. The PLA should have the inputs x_1, \dots, x_3 ; the product terms P_1, \dots, P_4 ; and the outputs f_1 and f_2 .

$$f_1 = \overline{x_1} \overline{x_2} \overline{x_3} + \overline{x_1} x_2 x_3 + x_1 \overline{x_2} x_3 + x_1 x_2 \overline{x_3}$$

$$P_1 = \overline{x_1 + x_2 + x_3}$$

$$P_2 = \overline{x_1 + \overline{x}_2 + \overline{x}_3}$$

$$P_3 = \overline{\overline{x}_1 + x_2 + \overline{x}_3}$$

$$P_4 = \overline{\overline{x}_1 + \overline{x}_2 + x_3}$$

$$g_1 = P_1 + P_2 + P_3 + P_4$$

$$f_1 = \overline{g_1} \quad \text{lets validate } f_1 = \overline{g_1} = \overline{P_1 + P_2 + P_3 + P_4} = \overline{\overline{x_1 + x_2 + x_3} + \overline{x_1 + \overline{x}_2 + \overline{x}_3} + \overline{\overline{x}_1 + x_2 + \overline{x}_3} + \overline{\overline{x}_1 + \overline{x}_2 + x_3}}$$

$$= \overline{x_1 x_2 \overline{x}_3} + \overline{x_1 x_2 x_3} + x_1 \overline{x_2} x_3 + x_1 x_2 \overline{x}_3 \quad (\text{Given function})$$

- 3.38** Show how the function f_1 from problem 3.36 can be realized in a PLA of the type shown in Figure 3.65. Draw a picture of such a PLA programmed to implement f_1 . The PLA should have the inputs x_1, \dots, x_3 ; the sum terms S_1, \dots, S_4 ; and the outputs f_1 and f_2 .

Given $\text{SOP } f_1 = \sum m(1, 2, 4, 7)$

POS: $f_1 = \overline{x_1} M(0, 3, 5, 6)$

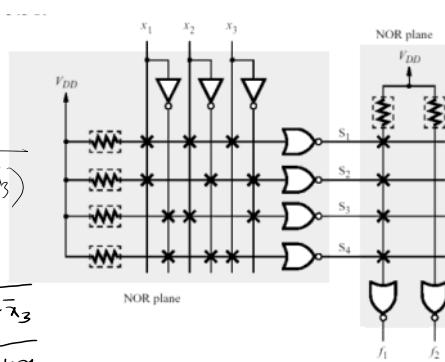
$$S_1 = (\overline{x_1} + x_2 + x_3)(x_1 + \overline{x}_2 + \overline{x}_3)(\overline{x}_1 + x_2 + \overline{x}_3)(\overline{x}_1 + \overline{x}_2 + x_3)$$

$$= (\overline{x_1} + x_2 + x_3) + (\overline{x_1} + \overline{x}_2 + \overline{x}_3) + (\overline{x}_1 + x_2 + \overline{x}_3) + (\overline{x}_1 + \overline{x}_2 + x_3)$$

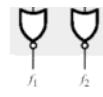
$$= \overline{S_1} + S_2 + S_3 + S_4$$

$$\text{where } S_1 = \overline{x_1 + x_2 + x_3} \quad S_3 = \overline{x_1 + x_2 + \overline{x}_3}$$

$$S_2 = \overline{x_1 + \overline{x}_2 + \overline{x}_3} \quad S_4 = \overline{x_1 + \overline{x}_2 + x_3}$$



where $S_1 = \overline{x_1 + x_2 + x_3}$ $S_3 = \overline{\bar{x}_1 + x_2 + \bar{x}_3}$
 $S_2 = \overline{x_1 + \bar{x}_2 + \bar{x}_3}$ $S_4 = \overline{\bar{x}_1 + \bar{x}_2 + x_3}$



NOR plane

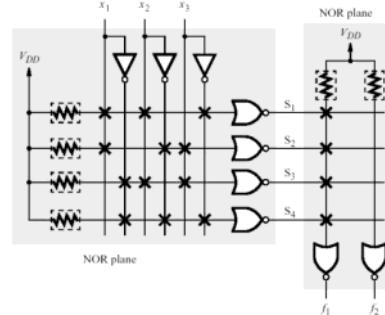
- 3.39** Show how the function f_1 from problem 3.37 can be realized in a PLA of the type shown in Figure 3.65. Draw a picture of such a PLA programmed to implement f_1 . The PLA should have the inputs x_1, \dots, x_3 ; the sum terms S_1, \dots, S_4 ; and the outputs f_1 and f_2 .

Given function $f_1 = \sum m(0, 3, 5, 6)$

The POS function $f_1 = \prod M(1, 2, 4, 7)$

$$f_1 = (x_1 + x_2 + \bar{x}_3)(\bar{x}_1 + \bar{x}_2 + x_3)(\bar{x}_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)$$

Apply the idea used in 3.38



- 3.44** Consider the function $f(x_1, x_2, x_3) = x_1\bar{x}_2 + x_1x_3 + x_2\bar{x}_3$. Show a circuit using 5 two-input lookup-tables (LUTs) to implement this expression. As shown in Figure 3.39, give the truth table implemented in each LUT. You do not need to show the wires in the FPGA.

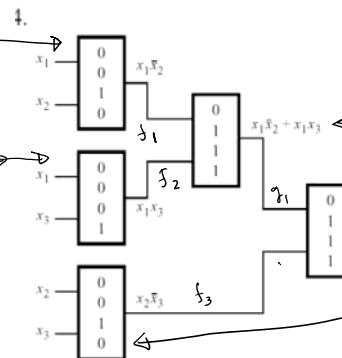
LUT for $x_1\bar{x}_2$

x_1	x_2	f_1
0	0	0
0	1	0
1	0	1
1	1	0

LUT

LUT for x_1x_3

x_1	x_3	f_2
0	0	0
0	1	0
1	0	0
1	1	1



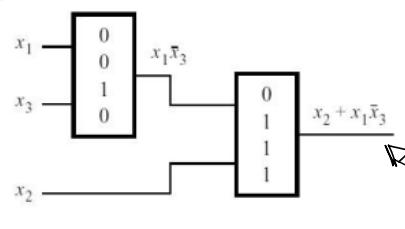
LUT for $x_2\bar{x}_3$

x_2	x_3	f_3
0	0	0
0	1	0
1	0	1
1	1	0

LUT for $\bar{x}_1 + \bar{x}_2$

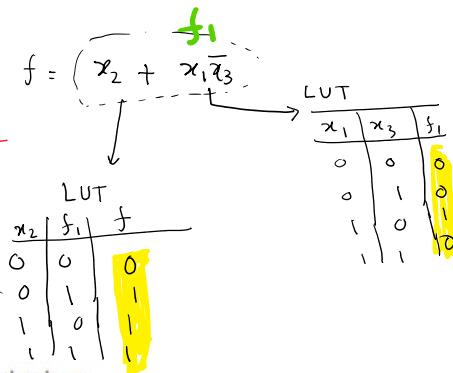
\bar{x}_1	\bar{x}_2	f_4
0	0	0
0	1	1
1	0	1
1	1	1

- 3.45** Consider the function $f(x_1, x_2, x_3) = \sum m(2, 3, 4, 6, 7)$. Show how it can be realized using two two-input LUTs. As shown in Figure 3.39, give the truth table implemented in each LUT. You do not need to show the wires in the FPGA.

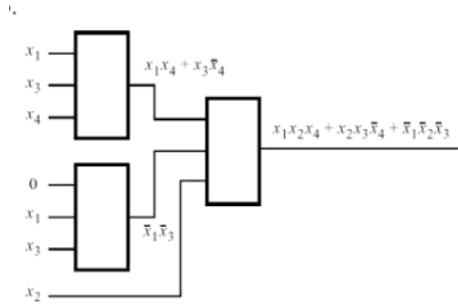


LUT for x_2x_3

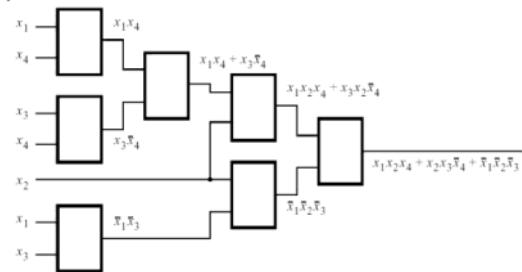
x_2	x_3	f_5
0	0	0
0	1	0
1	0	1
1	1	1



- 3.46** Given the function $f = x_1x_2x_4 + x_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3$, a straightforward implementation in an FPGA with three-input LUTs requires four LUTs. Show how it can be done using only 3 three-input LUTs. Label the output of each LUT with an expression representing the logic function that it implements.



- 3.47** For f in problem 3.46, show a circuit of two-input LUTs that realizes the function. You are to use exactly seven two-input LUTs. Label the output of each LUT with an expression representing the logic function that it implements.



- 3.49** Assume that a gate array contains the type of logic cell depicted in Figure P3.9. The inputs in_1, \dots, in_7 can be connected to either 1 or 0, or to any logic signal. (a) Show how the logic cell can be used to realize $f = x_1x_2 + x_3$. (b) Show how the logic cell can be used to realize $f = x_1x_3 + x_2x_3$.

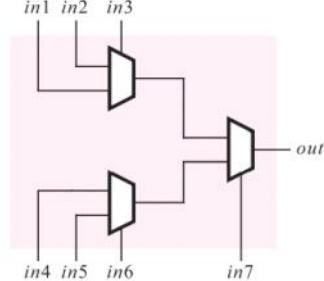
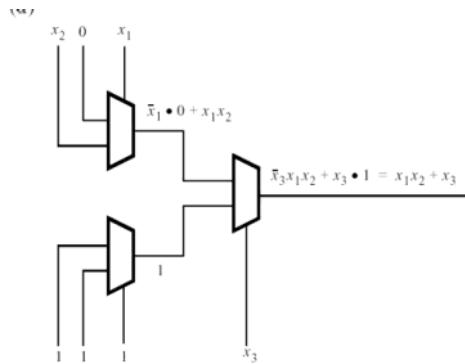
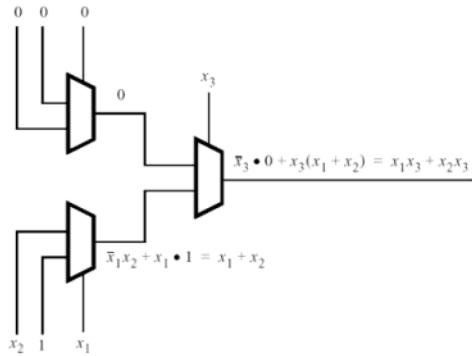


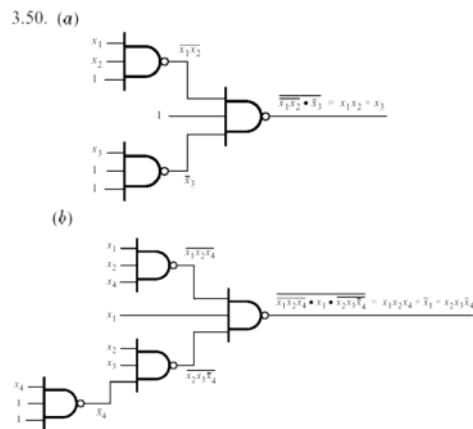
Figure P3.9 A gate-array logic cell.





3.50 Assume that a gate array exists in which the logic cell used is a three-input NAND gate. The inputs to each NAND gate can be connected to either 1 or 0, or to any logic signal. Show how the following logic functions can be realized in the gate array. (Hint: use DeMorgan's theorem.)

- (a) $f = x_1x_2 + x_3$
- (b) $f = x_1x_2x_4 + x_2x_3\bar{x}_4 + \bar{x}_1$



3.51 Write VHDL code to represent the function

$$f = x_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_4 + \bar{x}_1x_2x_3 + x_1x_2x_3$$

- (a) Use your CAD tools to implement f in some type of chip, such as a CPLD. Show the logic expression generated for f by the tools. Use timing simulation to determine the time needed for a change in inputs x_1 , x_2 , or x_3 to propagate to the output f .
- (b) Repeat part (a) using a different chip, such as an FPGA for implementation of the circuit.

```

LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY prob3_51 IS
    PORT ( x1, x2, x3, x4 : IN STD.LOGIC ;
           f : OUT STD.LOGIC ) ;
END prob3_51 ;

ARCHITECTURE LogicFunc OF prob3_51 IS
BEGIN
    f <= (x2 AND NOT x3 AND NOT x4) OR
          (NOT x1 AND x2 AND x4) OR
          (NOT x1 AND x2 AND x3) OR (x1 AND x2 AND x3) ;
END LogicFunc ;

```

3.52 Repeat problem 3.51 for the function

$$f = (x_1 + x_2 + \bar{x}_4) \cdot (\bar{x}_2 + x_3 + \bar{x}_4) \cdot (\bar{x}_1 + x_3 + \bar{x}_4) \cdot (\bar{x}_1 + \bar{x}_3 + \bar{x}_4)$$

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY prob3_52 IS
    PORT ( x1, x2, x3, x4 : IN STD.LOGIC ;
           f : OUT STD.LOGIC ) ;
END prob3_52 ;

ARCHITECTURE LogicFunc OF prob3_52 IS
BEGIN
    f <= (x1 OR x2 OR NOT x4) AND
          (NOT x2 OR x3 OR NOT x4) AND
          (NOT x1 OR x3 OR NOT x4) AND
          (NOT x1 OR NOT x3 OR NOT x4) ;
END LogicFunc ;
```