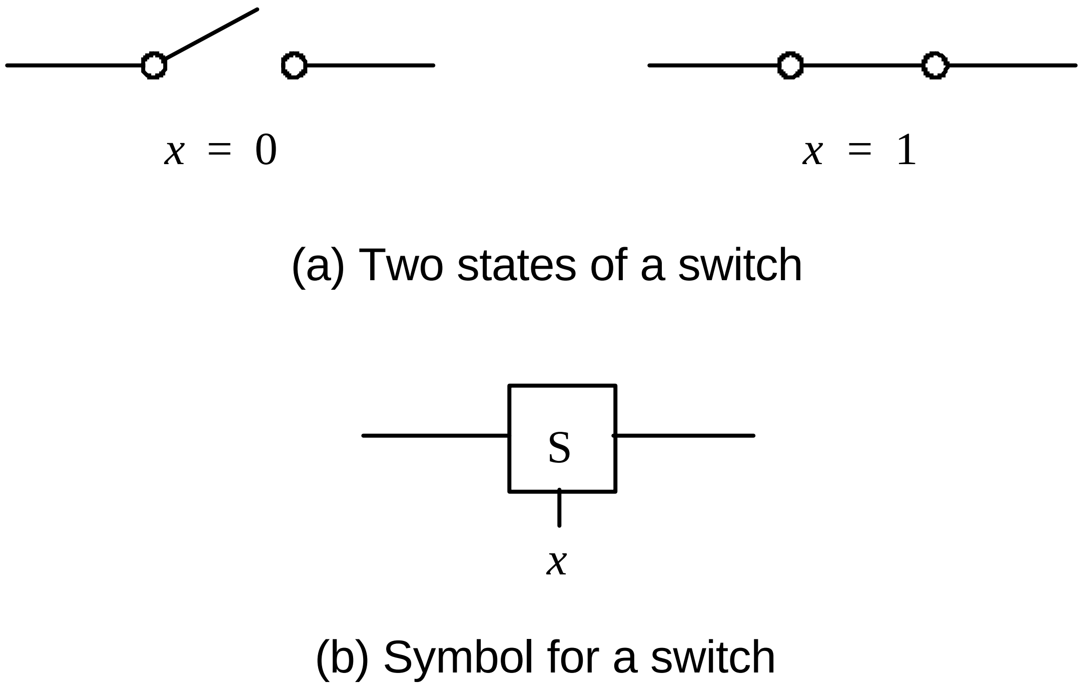
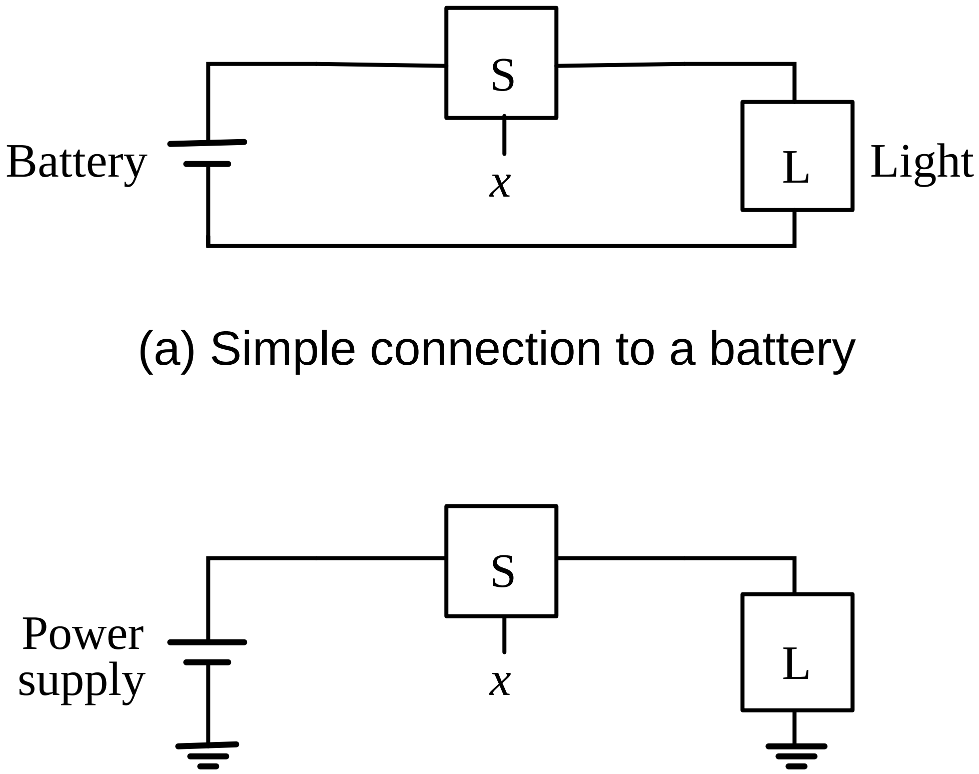


(a) Two states of a switch

(b) Symbol for a switch

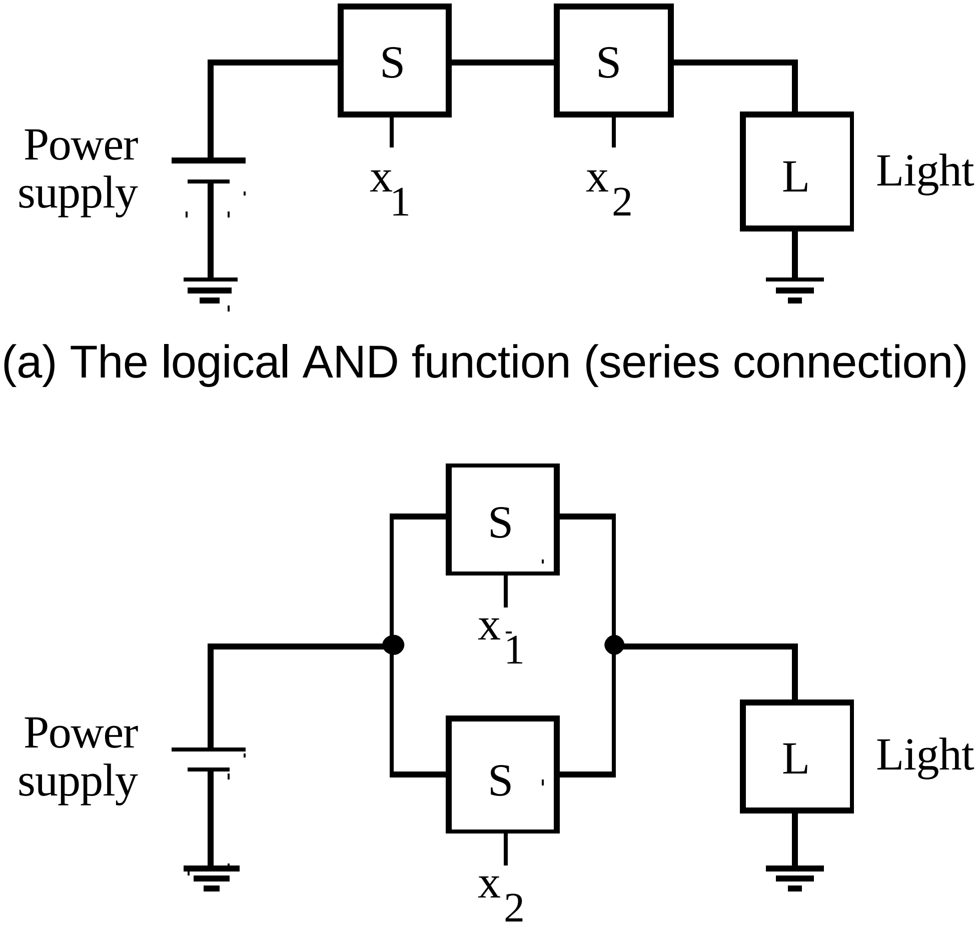
Figure 2.1 A binary switch



(a) Simple connection to a battery

(b) Using a ground connection as the return path

Figure 2.2 A light controlled by a switch



(a) The logical AND function (series connection)

(b) The logical OR function (parallel connection)

Figure 2.3 Two basic functions

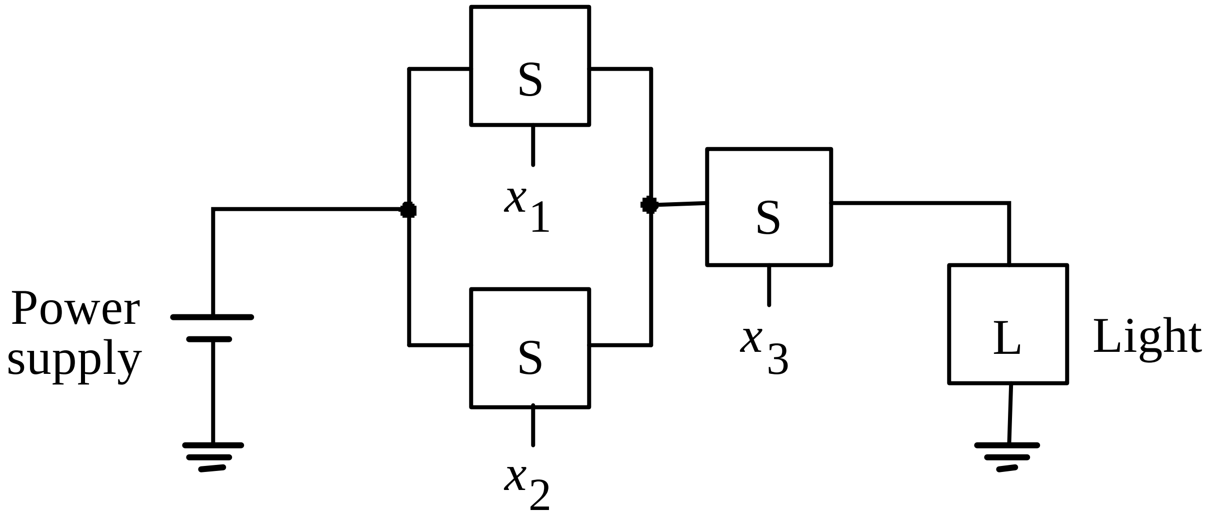


Figure 2.4 A series-parallel connection

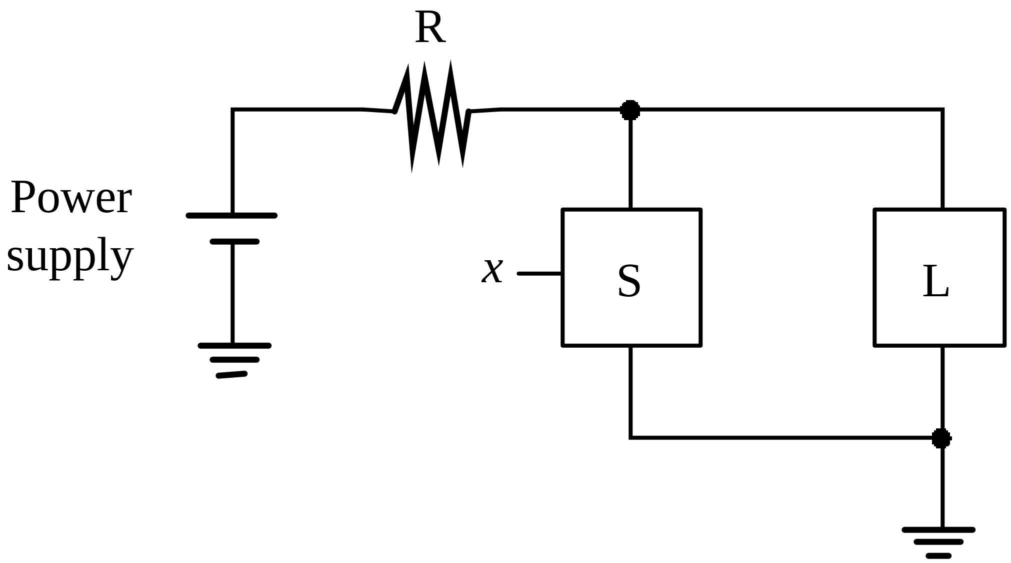
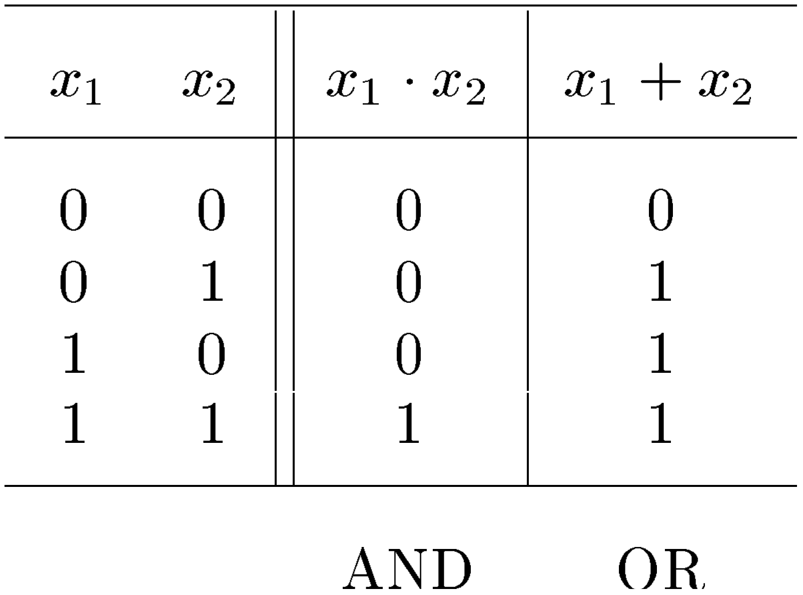


Figure 2.5 An inverting circuit

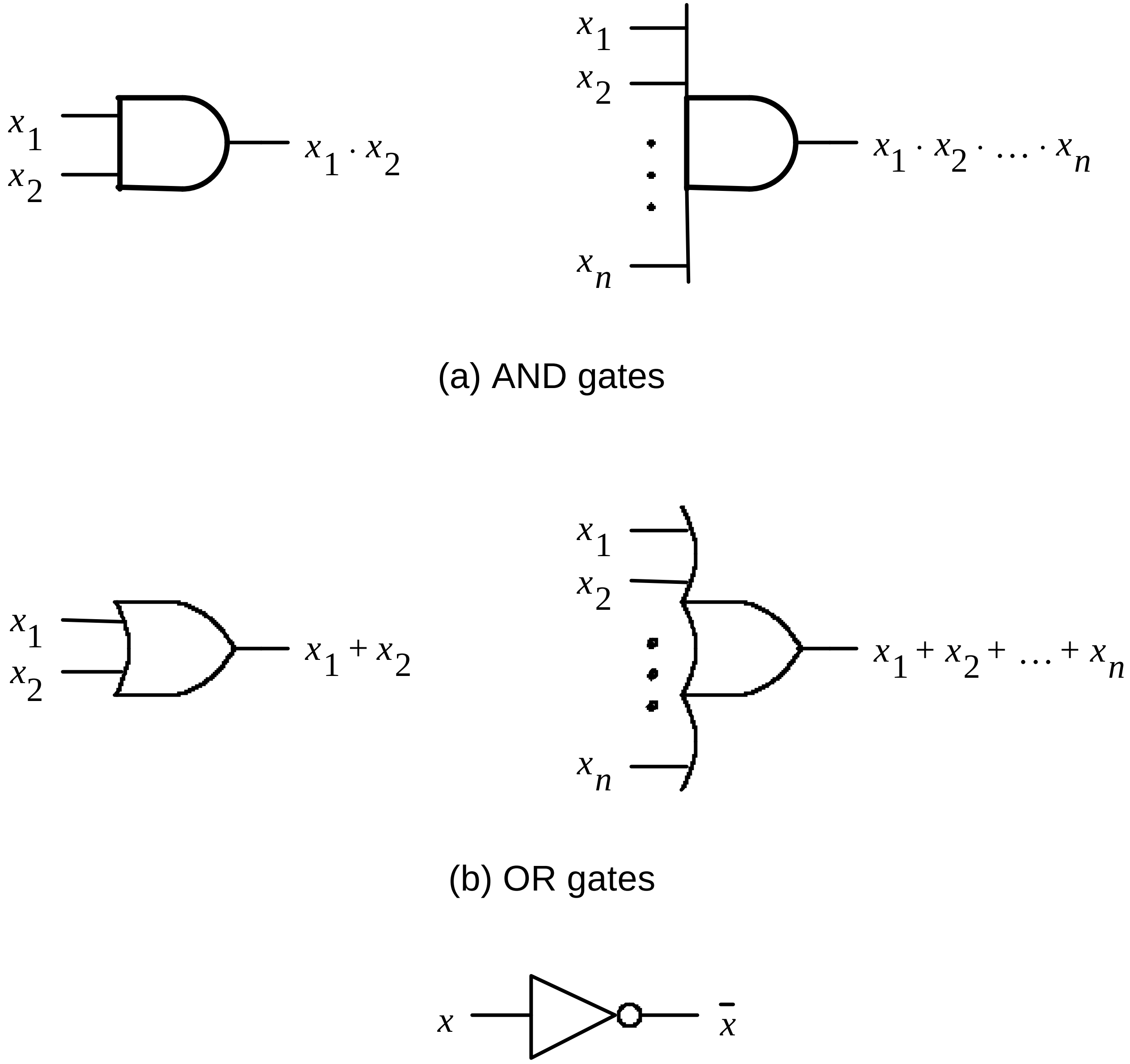


x_1	x_2	$x_1 \cdot x_2$	$x_1 + x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1
		AND	OR

Figure 2.6 A truth table for AND and OR

x_1	x_2	x_3	$x_1 \cdot x_2 \cdot x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 2.7 Three-input AND and OR



(a) AND gates

(b) OR gates

(c) NOT gate

Figure 2.8 The basic gates

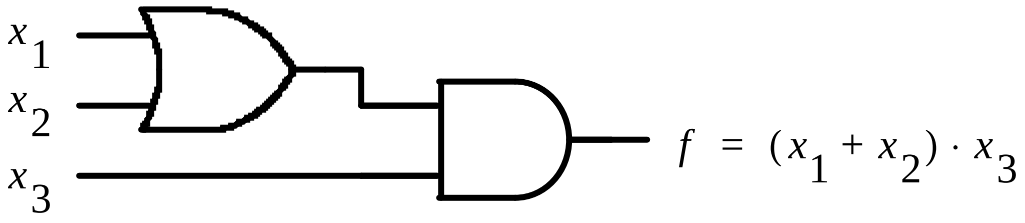
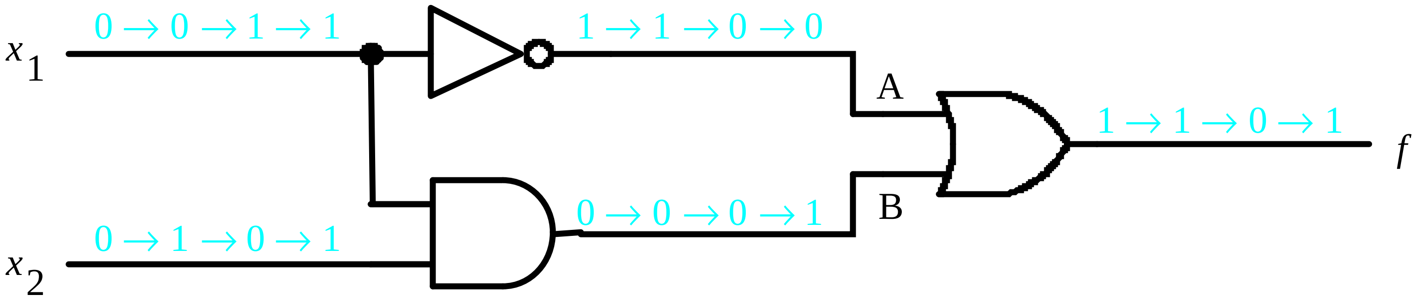


Figure 2.9 An OR-AND function

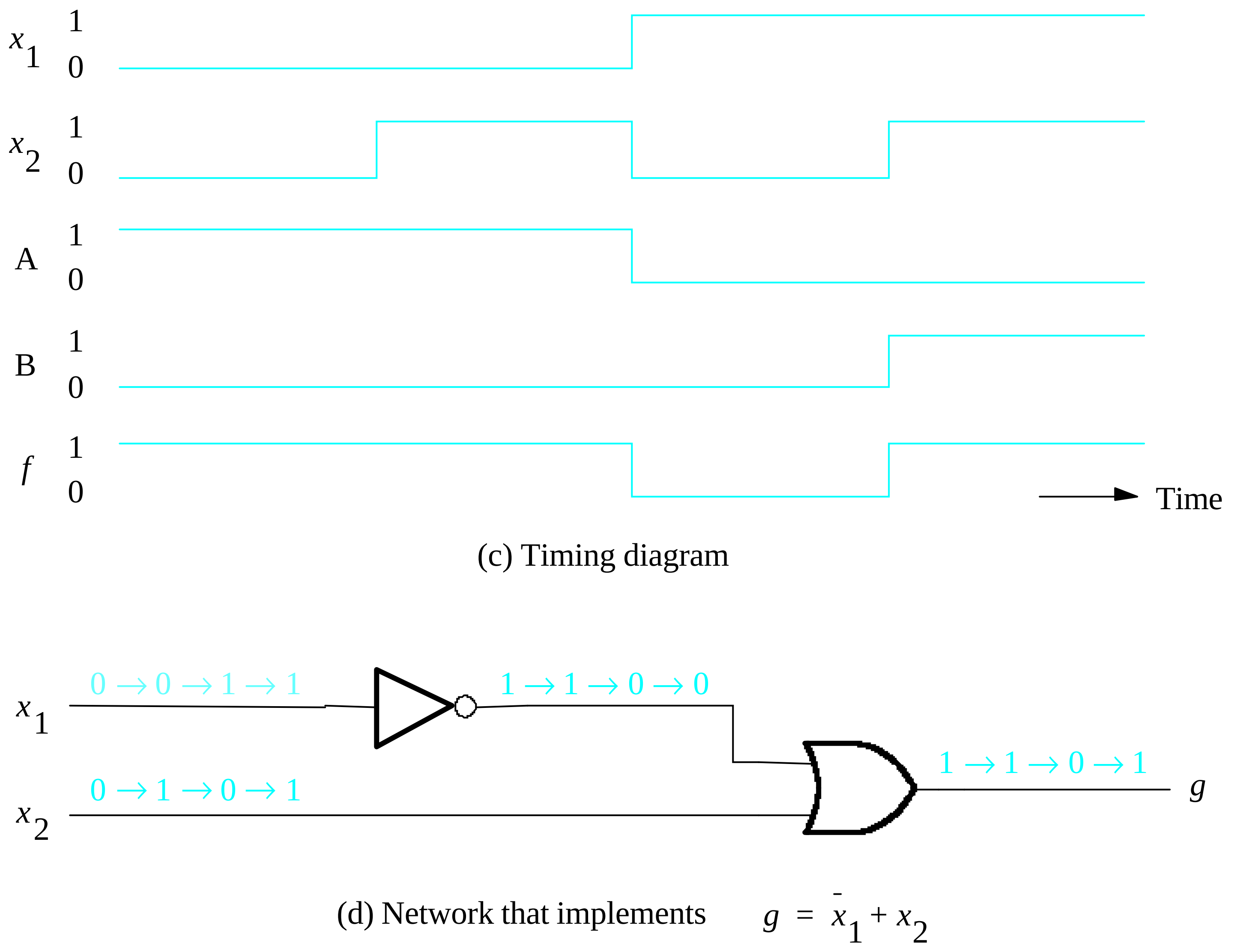


(a) Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

(b) Truth table for f

Figure 2.10 a Logic network



(c) Timing diagram

(d) Network that implements $g = \bar{x}_1 + x_2$

Figure 2.10 b Logic network

x	y	$x \cdot y$	$\overline{x \cdot y}$	$\overline{x}$	$\overline{y}$	$\overline{x} + \overline{y}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0
		LHS		RHS		

Figure 2.11 Proof of DeMorgan's theorem

Please see “portrait orientation” PowerPoint file for Chapter 2

Figure 2.12 The Venn diagram representation

Please see “portrait orientation” PowerPoint file for Chapter 2

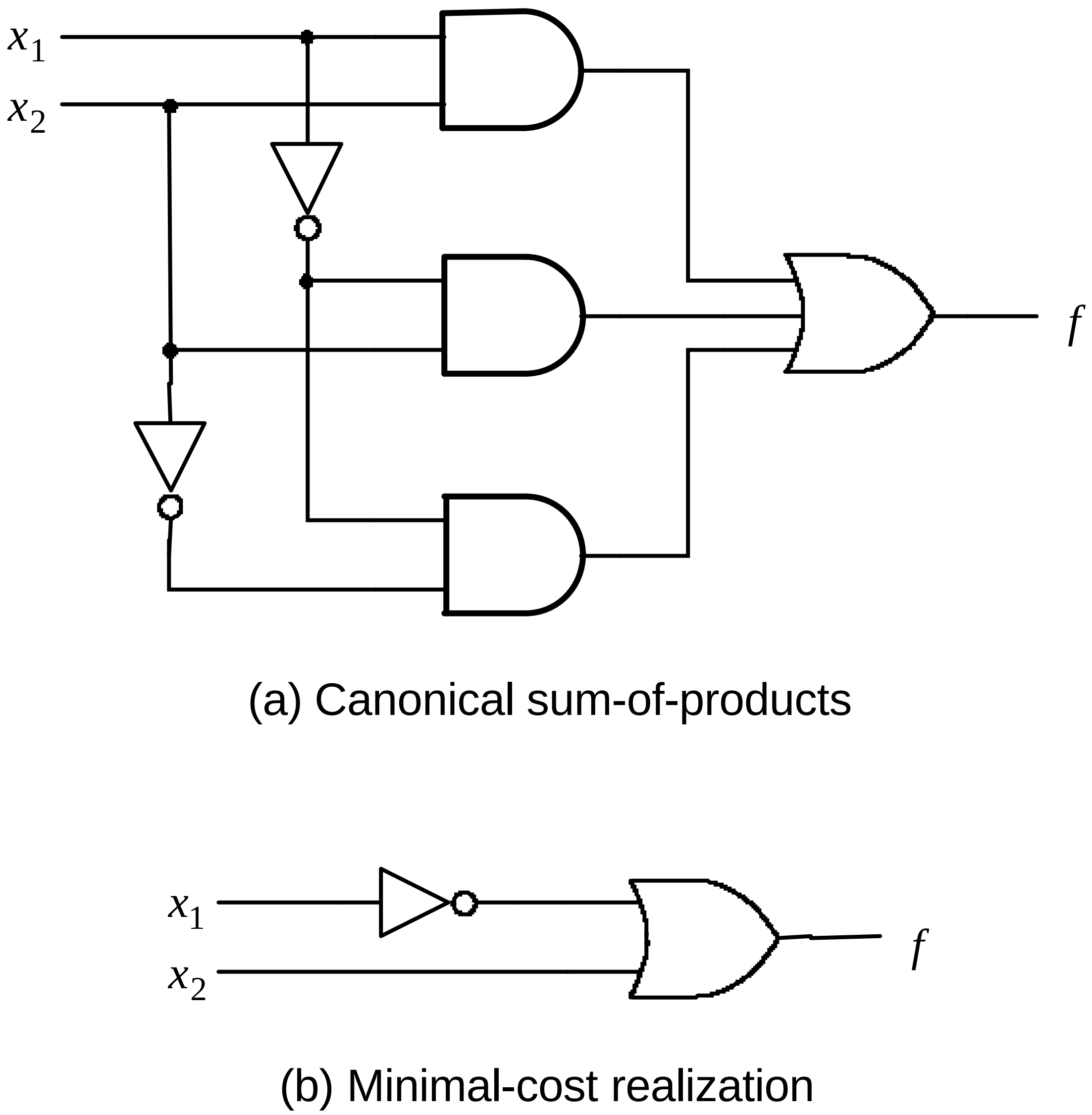
Figure 2.13 Verification of the distributive property

Please see “portrait orientation” PowerPoint file for Chapter 2

Figure 2.14 Verification example

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

Figure 2.15 A function to be synthesized



(a) Canonical sum-of-products

(b) Minimal-cost realization

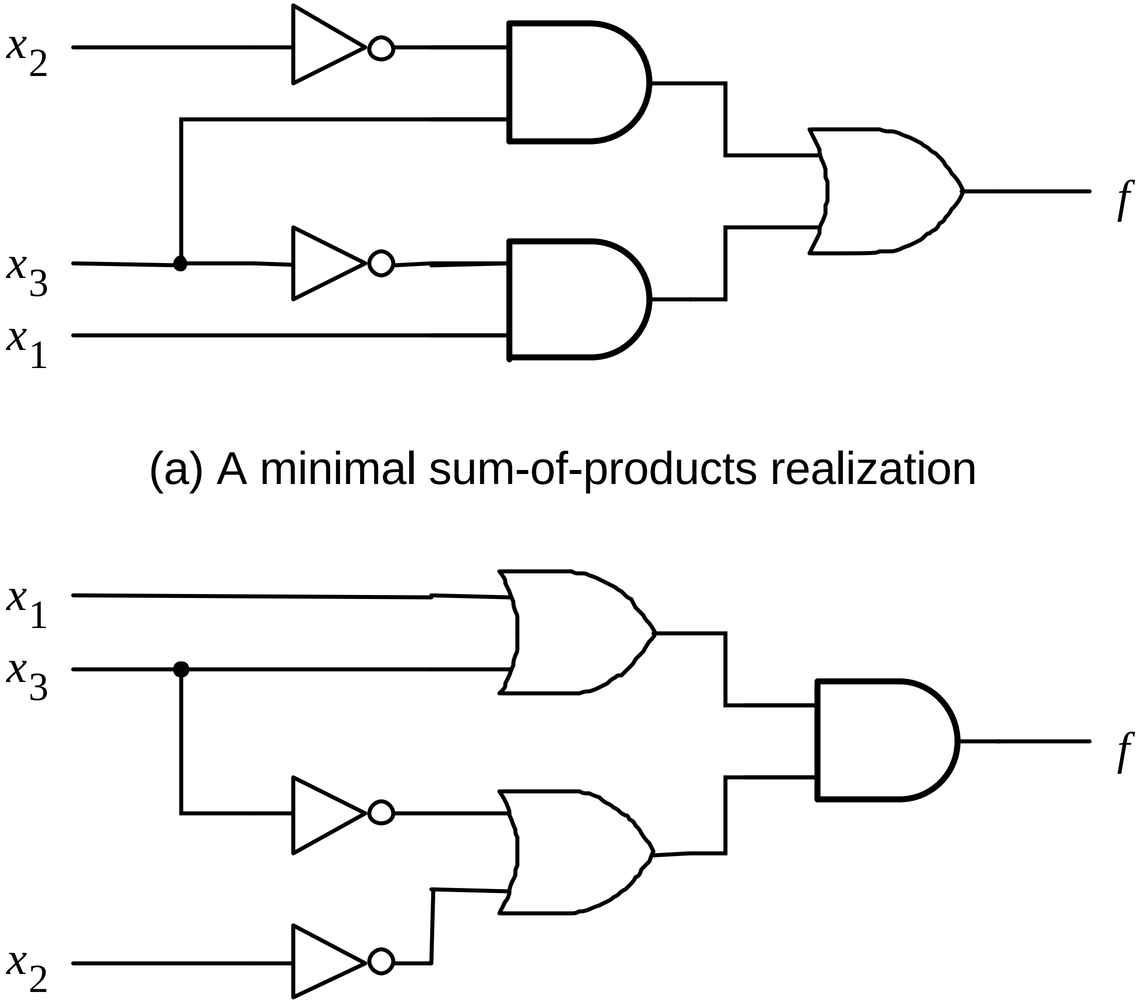
Figure 2.16 Two implementations of a function

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1x_2x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

Figure 2.17 Three-variable Minterms and Maxterms

Row number	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Figure 2.18 A three-variable function



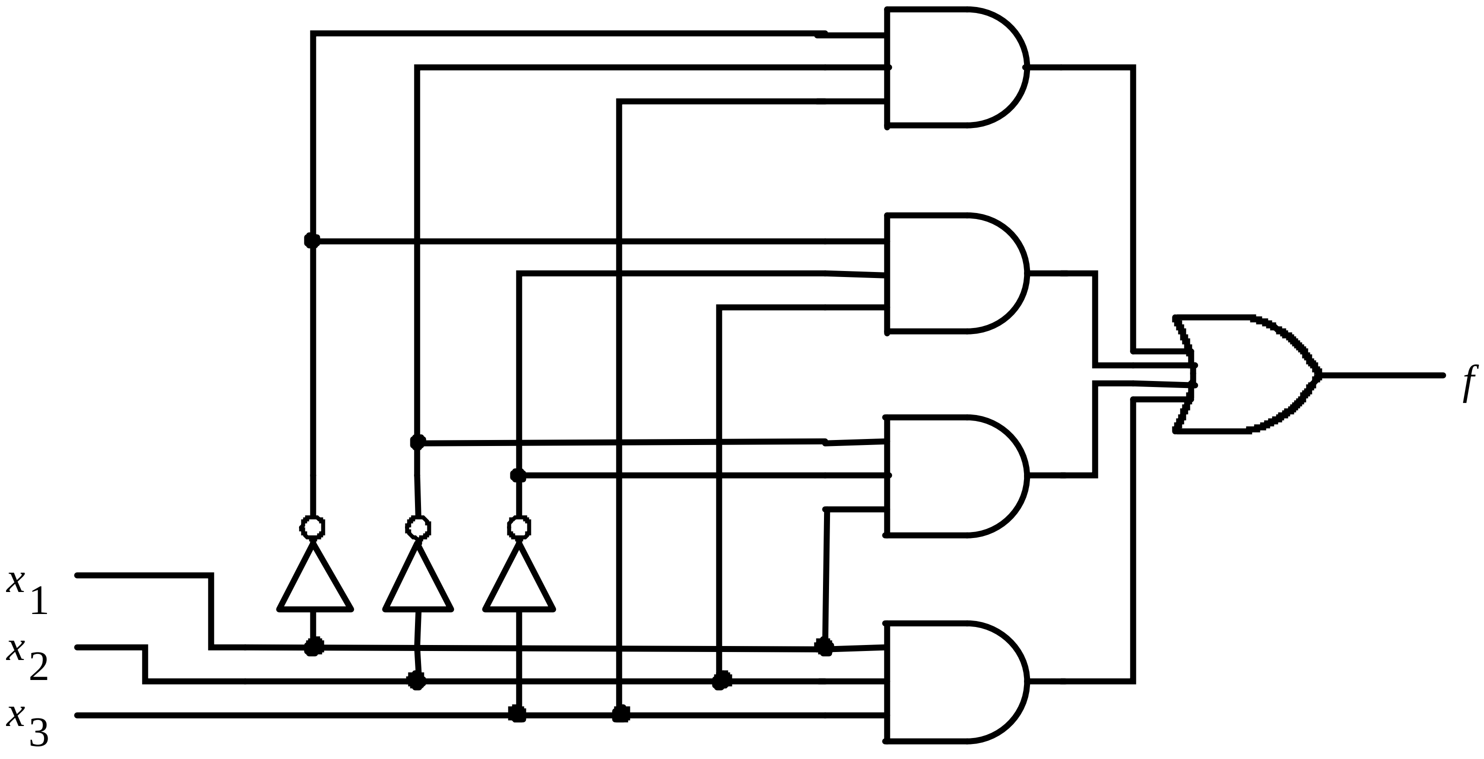
(a) A minimal sum-of-products realization

(b) A minimal product-of-sums realization

Figure 2.19 Two realizations of a function

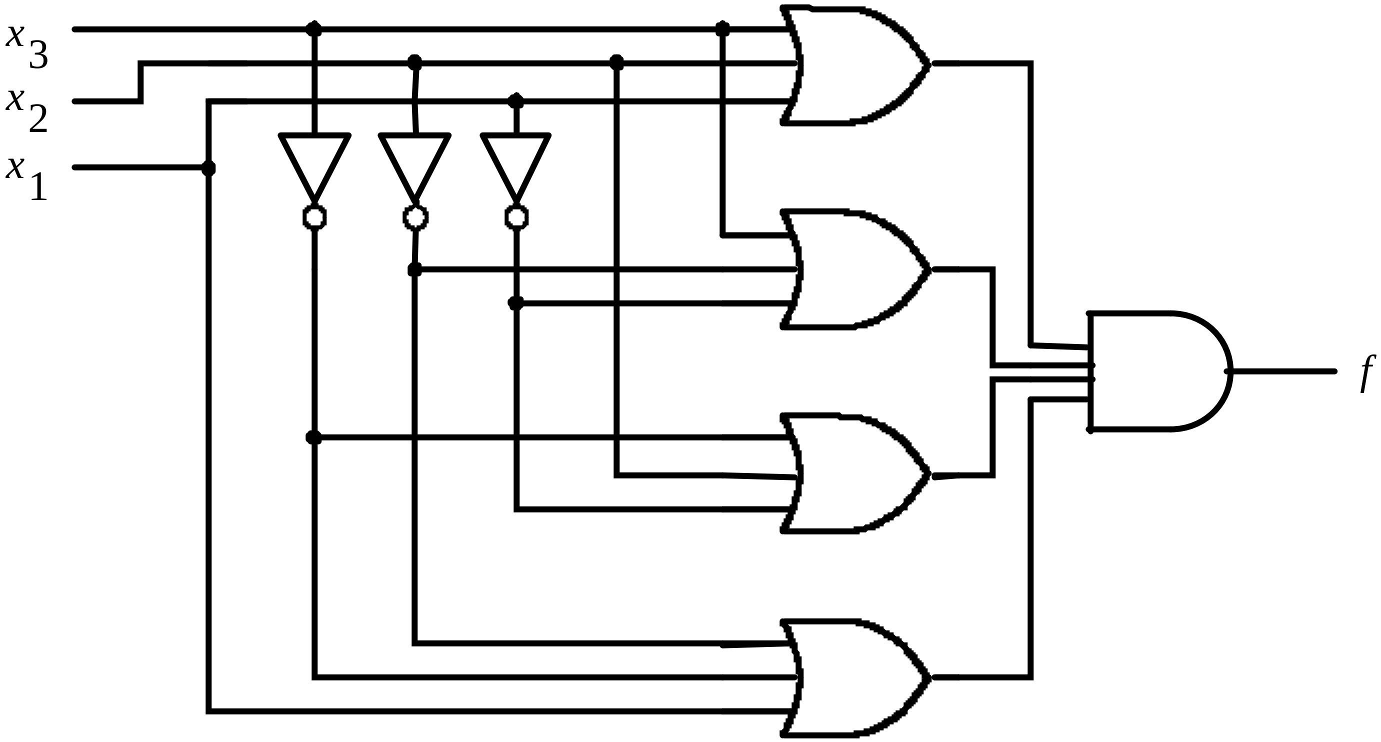
x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Figure 2.20 Truth table for a three-way light controller



(a) Sum-of-products realization

Figure 2.21 SOP implementation of the three-way light controller

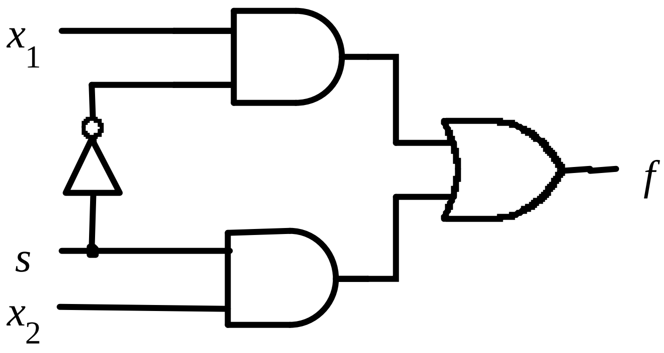


(b) Product-of-sums realization

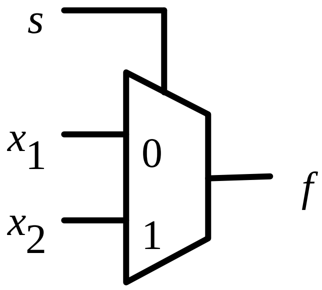
Figure 2.21 POS implementation of the three-way light controller

$s\ x_1\ x_2$	$f(s, x_1, x_2)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

(a)Truth table



(b) Circuit



(c) Graphical symbol

s	$f(s, x_1, x_2)$
0	x_1
1	x_2

(d) More compact truth-table representation

Figure 2.22 Multiplexer

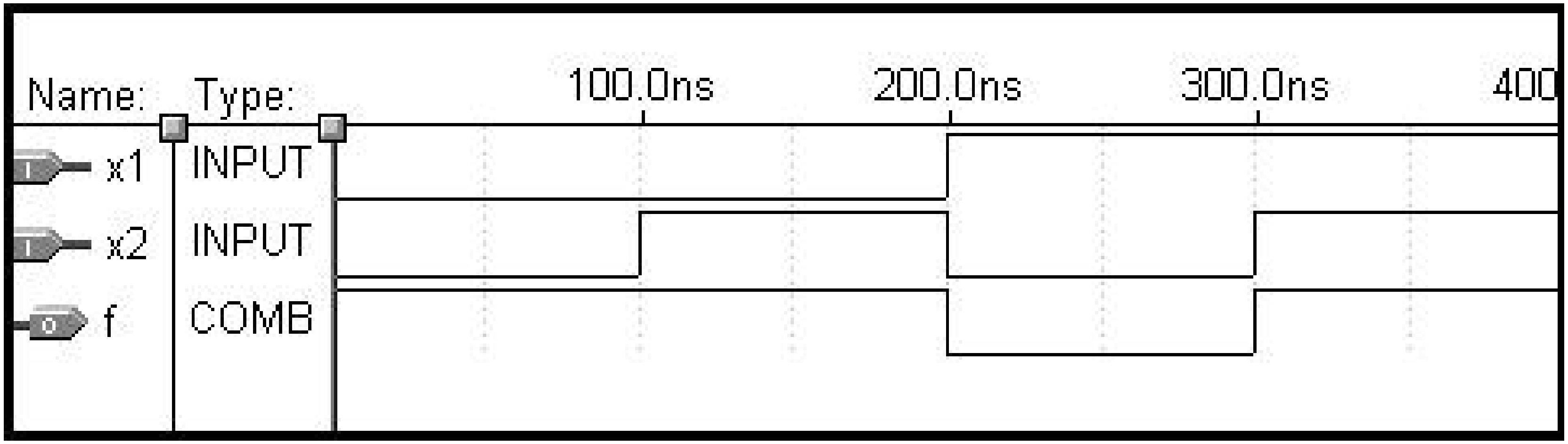


Figure 2.23 Screen capture of the Waveform Editor

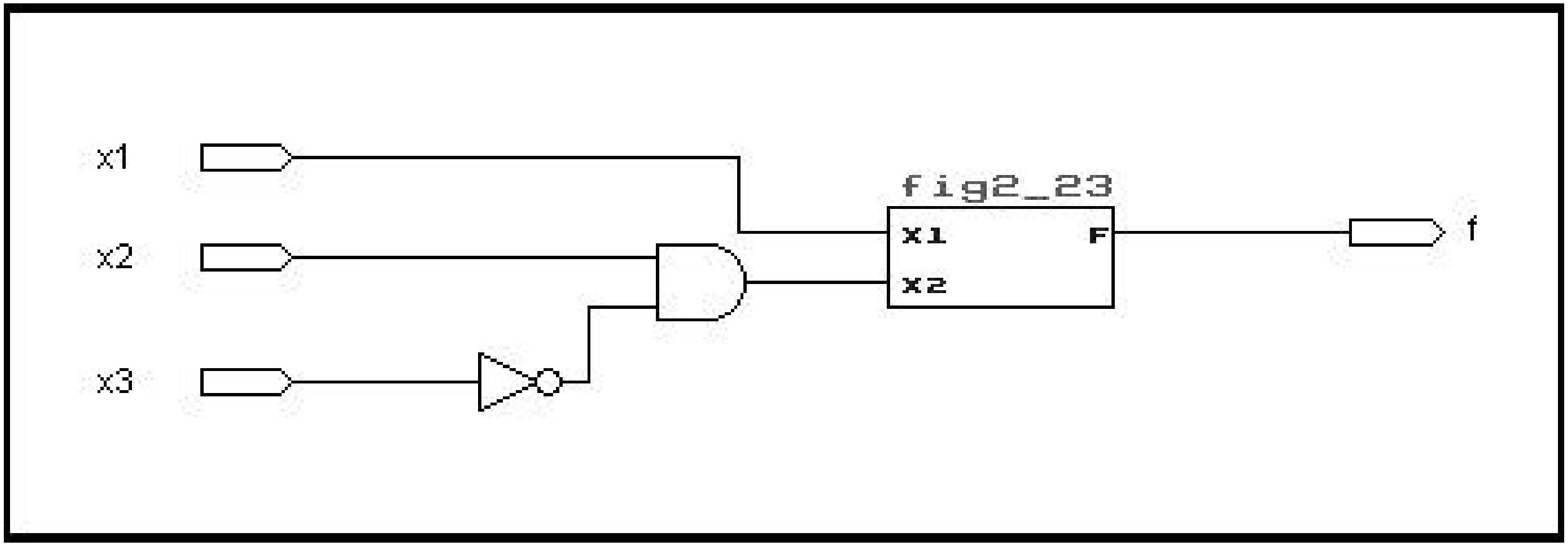
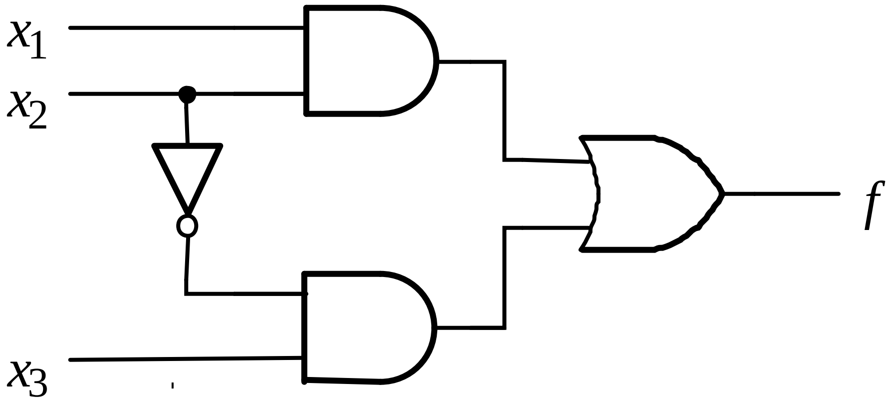
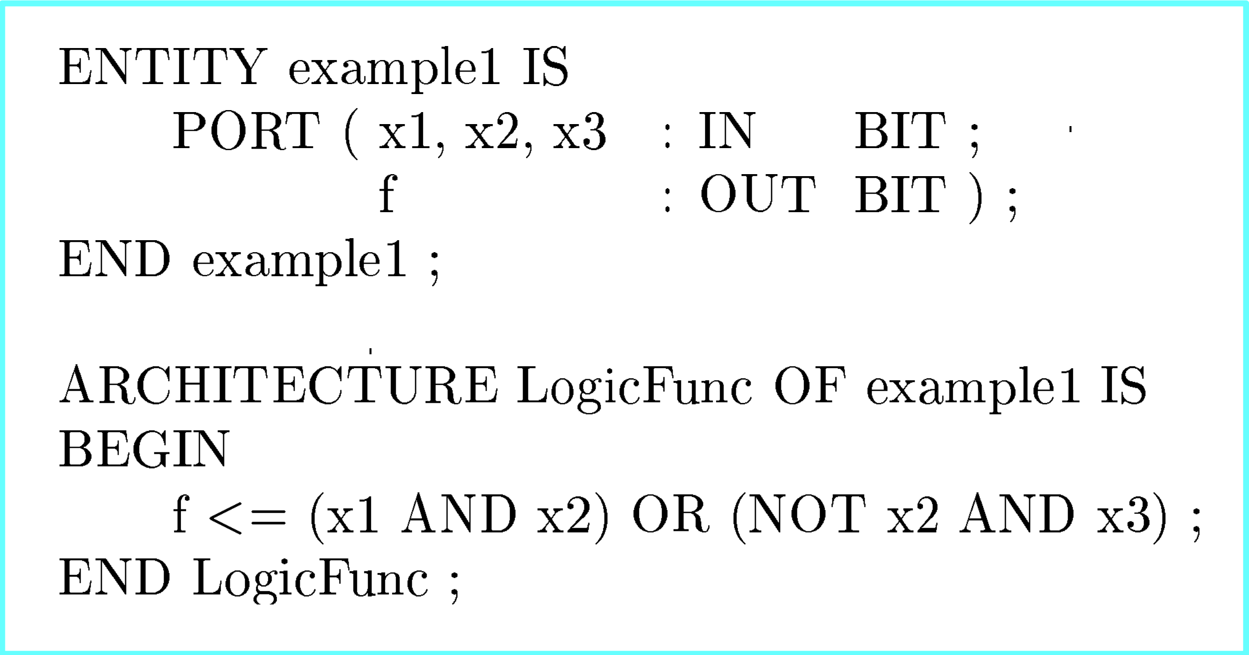


Figure 2.24 Screen capture of the Graphic Editor

Please see “portrait orientation” PowerPoint file for Chapter 2

Figure 2.25 The first stages of a CAD system





```
ENTITY example1 IS
        PORT ( x1, x2, x3   : IN     BIT ;
               f            : OUT  BIT ) ;
END example1 ;

ARCHITECTURE LogicFunc OF example1 IS
BEGIN
        f <= (x1 AND x2) OR (NOT x2 AND x3) ;
END LogicFunc ;
```

Figure 2.26 A simple logic function and corresponding VHDL code

```
ENTITY example2 IS
    PORT ( x1, x2, x3, x4  : IN    BIT ;
           f, g            : OUT   BIT ) ;
END example2 ;

ARCHITECTURE LogicFunc OF example2 IS
BEGIN
    f <= (x1 AND x3) OR (NOT x3 AND x2) ;
    g <= (NOT x3 OR x1) AND (NOT x3 OR x4) ;
END LogicFunc ;
```

Figure 2.30 VHDL code for a four-input function

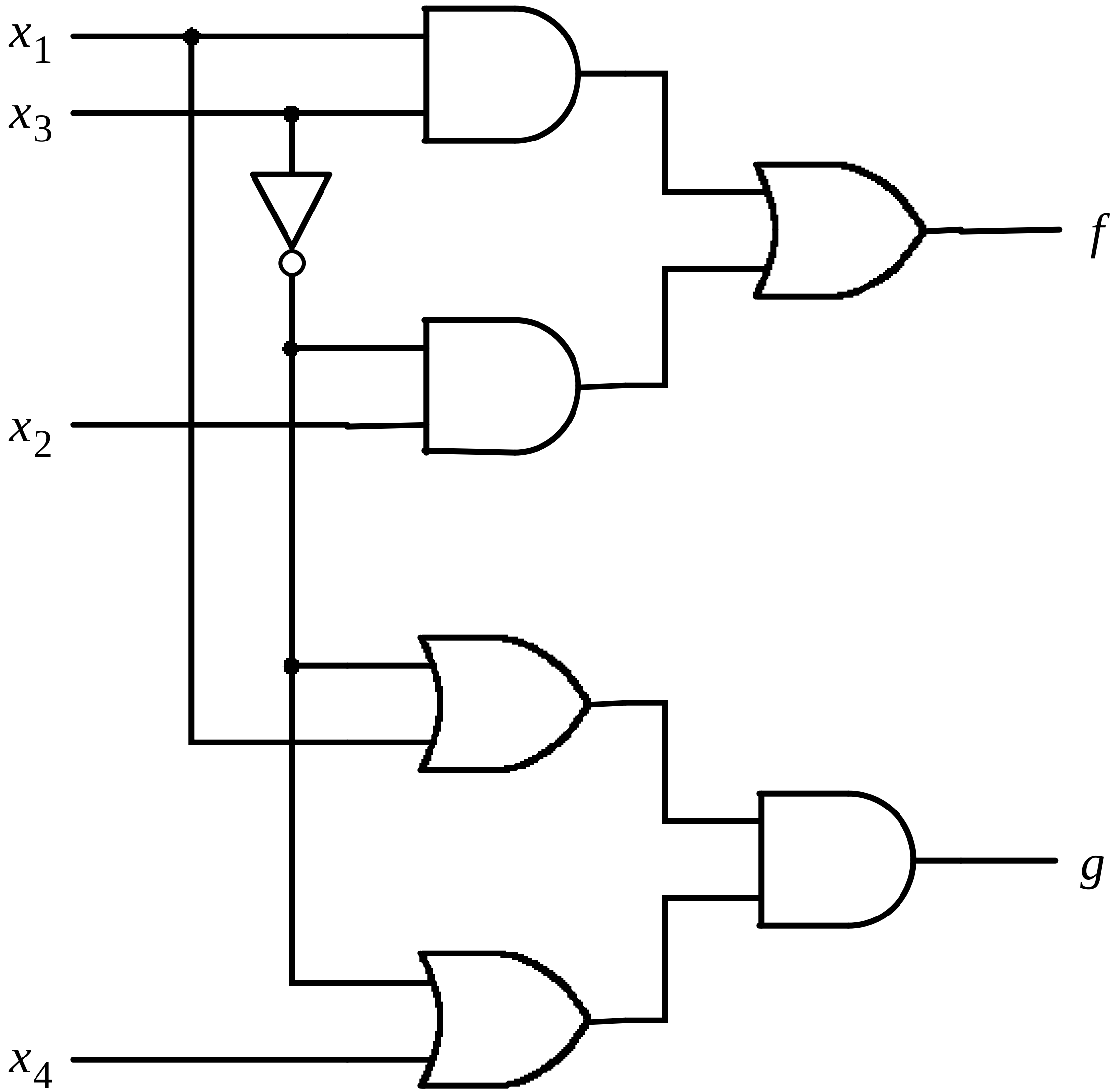


Figure 2. 31 Logic circuit for four-input function

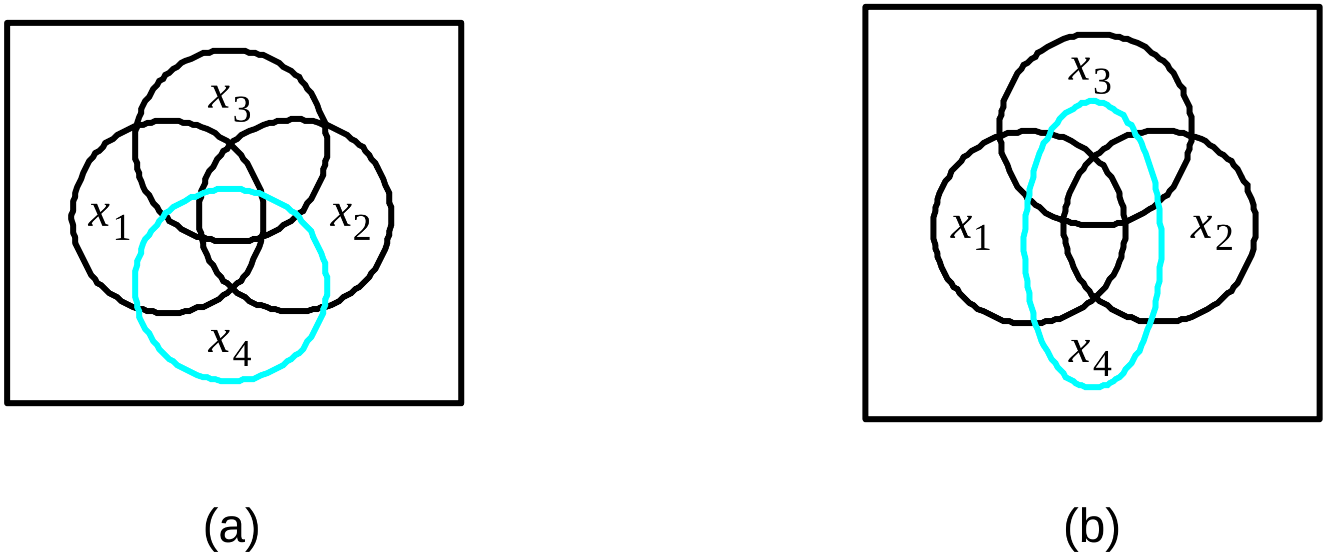


Figure P2. 1   Two attempts to draw a four-variable Venn diagram

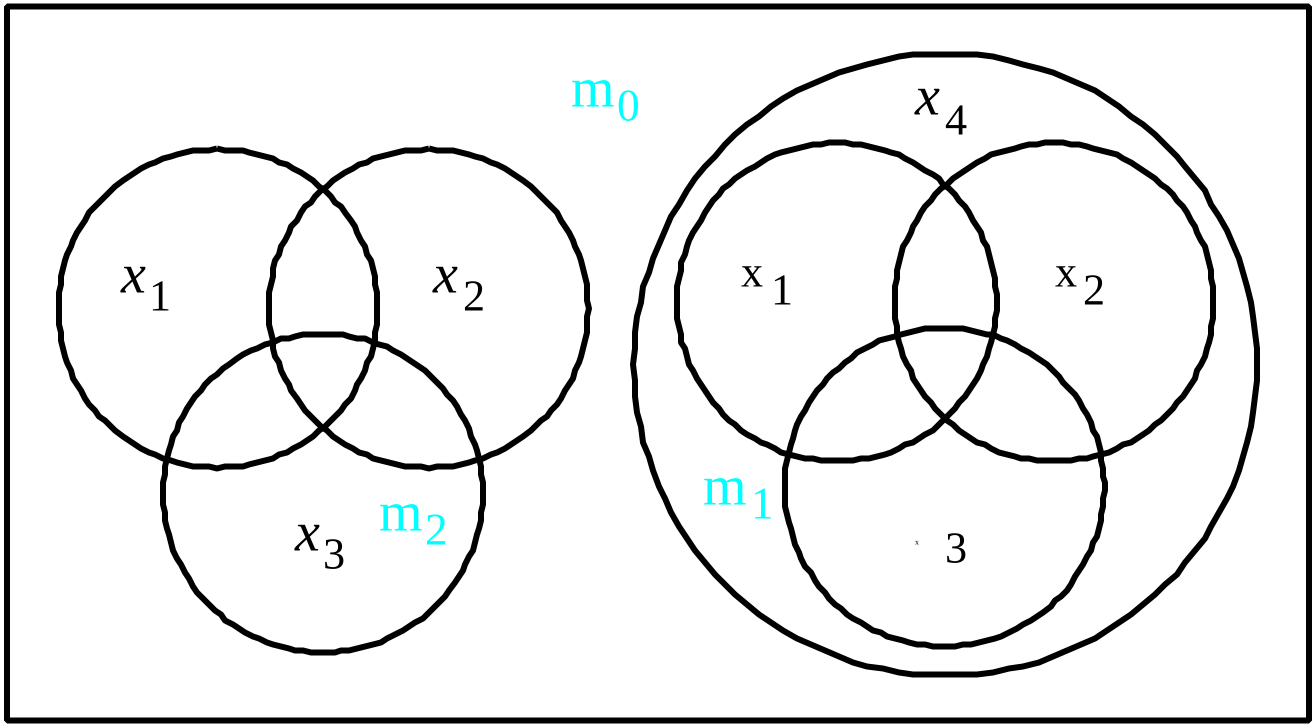


Figure P2. 2 A four-variable Venn diagram

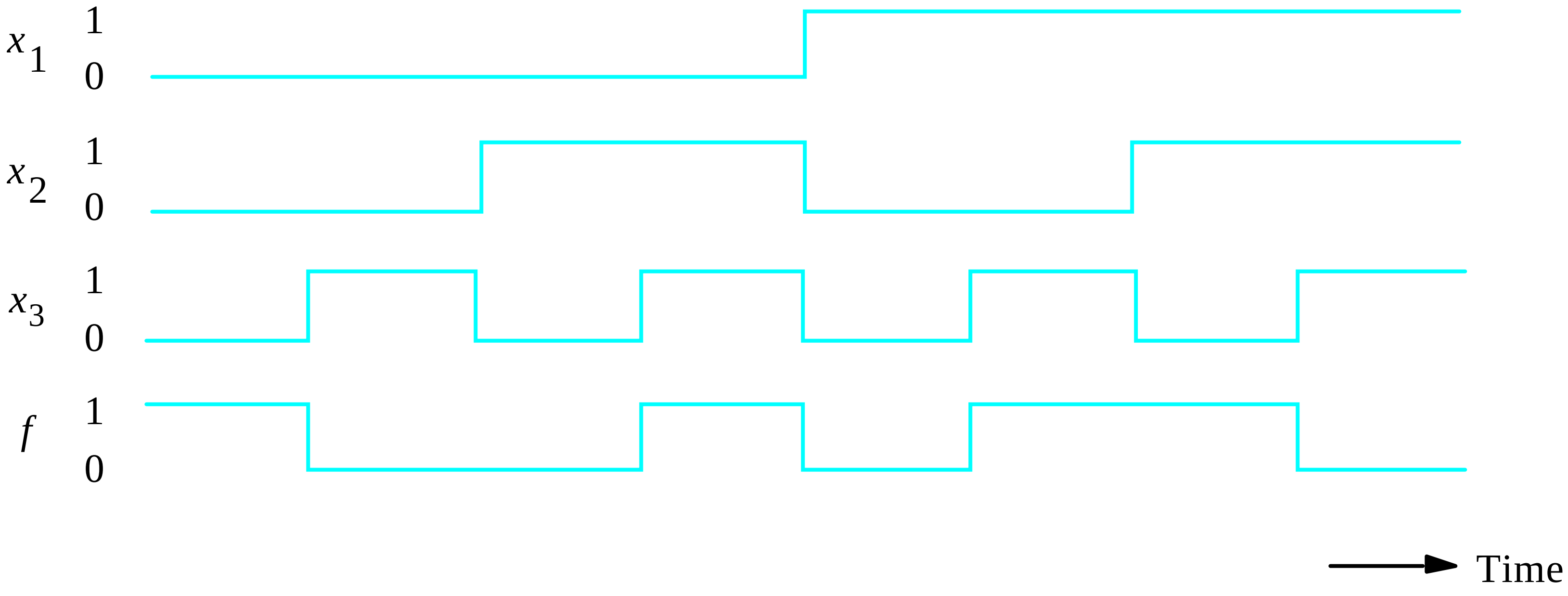


Figure P2. 3 A timing diagram representing a logic function

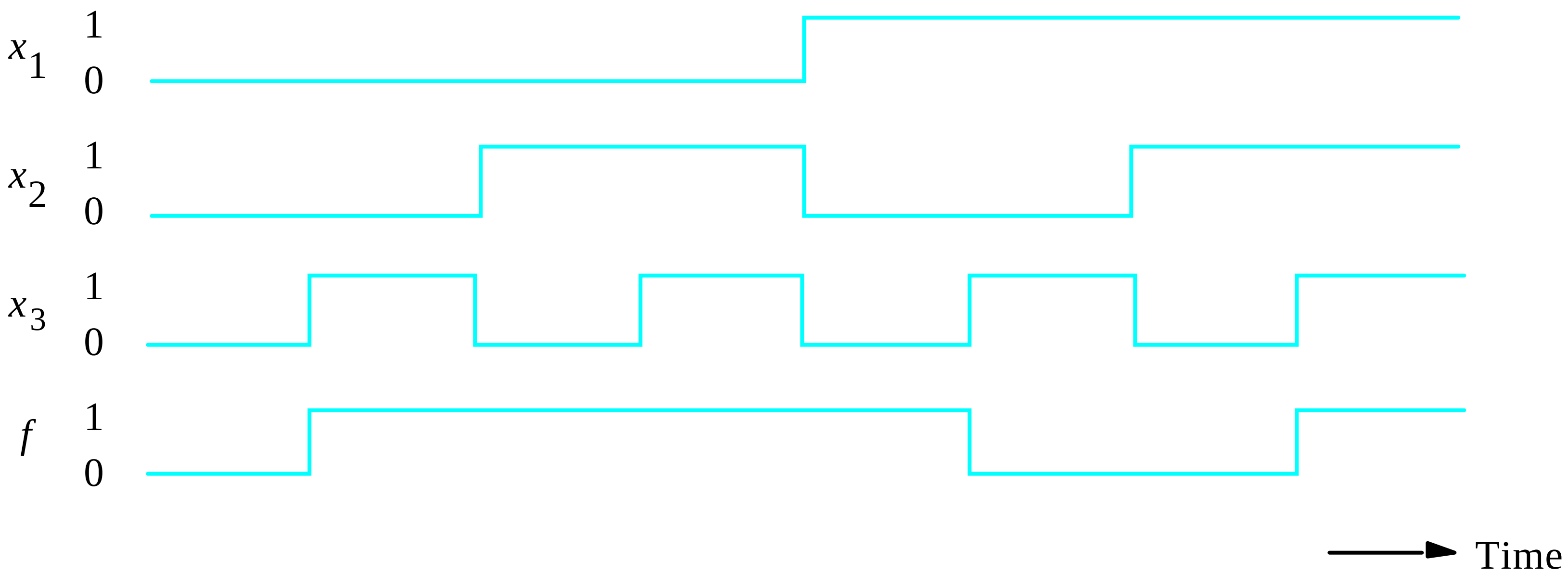


Figure P2. 4 A timing diagram representing a logic function