CHAPTER 4

BOOLEAN FUNCTIONS AND DIGITAL CIRCUITS

Canonical Forms

4.1.1 Canonical Sum-of-Products

Table 4.1 (a) A truth table for two variables. (b) Conversion of of (a) to a table for AND.

(a)

A	B	F
0	0	0
0	1	0
1	0	1
1	1	0

$X = B'$

(b)

A	X	F
0	1	0
0	0	0
1	1	1
1	0	0

$F = AX = AB'$

Table 4.2 (a) A truth table for three variables. (b) Conversion of of (a) to a table in comparison with AND.

(a)

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

X = A’, Y = B’

(b)

X	Y	C	F
1	1	0	0
1	1	1	1
1	0	0	0
1	0	1	0
0	1	0	0
0	1	1	0
0	0	0	0
0	0	1	0

F = XYC = A’B’C

Canonical (standard) product

Canonical (standard) sum-of-products

Table 4.3 Truth table for prime number detector.

A	B	C	F(A,B,C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

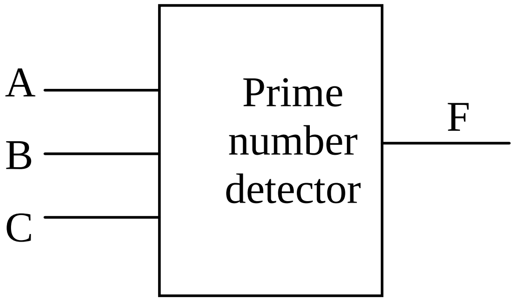


Figure 4.1 Block diagram for prime number detector.

$$F(A,B,C) = A'B'C + A'BC' + A'BC + AB'C + ABC$$

Minterm (Canonical product)
Minterm list representation

Table 4.4 List of canonical products for the prime number detector.

A B C		Canonical product	Minterm
Decimal	Binary		
1	0 0 1	A' B' C	m_1
2	0 1 0	A' B C'	m_2
3	0 1 1	A' B C	m_3
5	1 0 1	A B' C	m_5
7	1 1 1	A B C	m_7

$$F(A,B,C) = m_1 + m_2 + m_3 + m_5 + m_7$$

$$F(A,B,C) = \Sigma\, m(1, 2, 3, 5, 7)$$

4.1.2 Canonical Product-of-Sums

Table 4.5 Truth table for prime number detector.

A B C	F'(A,B,C)	Minterm for F'
0 0 0	1	m_0
0 0 1	0	
0 1 0	0	
0 1 1	0	
1 0 0	1	m_4
1 0 1	0	
1 1 0	1	m_6
1 1 1	0	

Canonical sum Canocical (standard) product-of-sums

Maxterm Product-of-maxterms

Maxterm list representation

$$F'(A,B,C) = m_0 + m_4 + m_6$$

$$F(A,B,C) = (m_0 + m_4 + m_6)'$$

$$= m_0' \bullet m_4' \bullet m_6'$$

$$= (A'B'C')' \bullet (AB'C')' \bullet (ABC')'$$

$$= (A + B + C) \bullet (A' + B + C) \bullet (A' + B' + C)$$

$$= M_0 \bullet M_4 \bullet M_6$$

$$= \pi M(0, 4, 6)$$

❖ Example 4.1

$$F(A,B,C,D) = \Sigma\, m(1, 2, 4, 5, 6, 7, 8, 10, 12, 13, 15)$$

minterm and maxterm numbers for an n-variable function from 0 to 2^{n-1}.

$$F(A,B,C,D) = \pi\, M(0, 3, 9, 11, 14)$$

Decimal	1	2	4	5	6	7
Binary	0 0 0 1	0 0 1 0	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1

F(A,B,C,D) = A'B'C'D + A'B'CD' + A'BC'D' + A'BC'D + A'BCD' + A'BCD

+ AB'C'D' + AB'CD' + ABC'D' + ABC'D + ABCD

Binary	1 0 0 0	1 0 1 0	1 1 0 0	1 1 0 1	1 1 1 1
Decimal	8	10	12	13	15

Decimal	0	3	9	11	14
Binary	0 0 0 0	0 0 1 1	1 0 0 1	1 0 1 1	1 1 1 0

F(A,B,C,D) = (A+B+C+D) (A+B+C'+D') (A'+B+C+D')(A'+B+C'+D')(A'+B'+C'+D)

Order of Variables

$F(A,B,C,D) = \Sigma\, m(1, 2, 4, 5, 6, 7, 8, 10, 12, 13, 15)$

	ABCD	ACBD	
1	0001	0001	1
2	0010	0100	4
4	0100	0010	2
5	0101	0011	3
6	0110	0110	6
7	0111	0111	7
8	1000	1000	8
10	1010	1100	12
12	1100	1010	10
13	1101	1011	11
15	1111	1111	15

$F(A,C,B,D) = \Sigma\, m(1, 4, 2, 3, 6, 7, 8, 12, 10, 11, 15)$

$= \Sigma\, m(1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 15)$

4.1.3 Conversion to Canonical Forms

- Example 4.2

$F(A,B,C,D) = A'B'C + BC' + AC'D + ABCD'$

$A'B'C = A'B'C(D' + D) = A'B'CD' + A'B'CD = m_2 + m_3$

$AC'D = AC'D(B' + B) = AB'C'D + ABC'D = m_9 + m_{13}$

$BC' = BC'(A' + A)(D' + D) = A'BC'D' + A'BC'D + ABC'D' + ABC'D$
$= m_4 + m_5 + m_{12} + m_{13}$

$ABCD' = m_{14}$

$F(A,B,C,D) = (m_2 + m_3) + (m_9 + m_{13}) + (m_4 + m_5 + m_{12} + m_{13}) + m_{14}$
$= \Sigma\, m(2, 3, 4, 5, 9, 12, 13, 14)$

$F(A,B,C,D) = A'B'C + BC' + AC'D + ABCD'$

Table 4.6 Conversion of products to minterms.

Product	Variable A B C D	Minterm number
A B C D'	1 1 1 0	14
A' B' C	0 0 1 0	2
	0 0 1 1	3
A C' D	1 0 0 1	9
	1 1 0 1	13
B C'	0 1 0 0	4
	0 1 0 1	5
	1 1 0 0	12
	1 1 0 1	13

- Example 4.3

To find the maxterm list representation

$$F(A,B,C,D) = (B' + C + D)(A + B')(A' + D')$$

$$\begin{aligned} F(A,B,C,D) &= (B' + C + D)(A + B')(A' + D') \\ &= (B' + C + D)(AD' + A'B') \\ &= AB'D' + ACD' + A'B' + A'B'C + A'B'D \\ &= AB'D' + ACD' + A'B' \end{aligned}$$

$AB'D' = m_8 + m_{10}$

$ACD' = m_{10} + m_{14}$

$A'B' = m_0 + m_1 + m_2 + m_3$

$F(A,B,C,D) = \Sigma\, m(0, 1, 2, 3, 8, 10, 14)$

$F(A,B,C,D) = \pi\, M(4, 5, 6, 7, 9, 11, 12, 13, 15)$

Example 4.3 (Alternative approach)

To find the maxterm list representation

$$F(A,B,C,D) = (B' + C + D)(A + B')(A' + D')$$

$$B' + C + D = A'A + B' + C + D = (A' + B' + C + D)(A + B' + C + D) = M_{12} M_4$$

$$A + B' = A + B' + C'C + D'D = (A + B' + C' + D'D)(A + B' + C + D'D)$$
$$= (A + B' + C' + D')(A + B' + C' + D)(A + B' + C + D')(A + B' + C + D)$$
$$= M_7 M_6 M_5 M_4$$

$$A' + D' = A + B'B + C'C + D' = (A' + B' + C'C + D')(A' + B + C'C + D')$$
$$= (A' + B' + C' + D')(A' + B' + C + D')(A' + B + C' + D')(A' + B + C + D')$$
$$= M_{15} M_{13} M_{11} M_9$$

$$F(A,B,C,D) = (M_{12} M_4)(M_7 M_6 M_5 M_4)(M_{15} M_{13} M_{11} M_9)$$
$$= \pi M(4, 5, 6, 7, 9, 11, 12, 13, 15)$$

$$F(A,B,C,D) = (B' + C + D)(A + B')(A' + D')$$

Table 4.7 Conversion of sums to maxterm numbers.

Sum	A	B	C	D	Maxterm number
B'+C+D	0	1	0	0	4
	1	1	0	0	12
A+B'	0	1	0	0	4
	0	1	0	1	5
	0	1	1	0	6
	0	1	1	1	7
A'+D'	1	0	0	1	9
	1	0	1	1	11
	1	1	0	1	13
	1	1	1	1	15

$$F(x_{n-1}, x_{n-2}, \ldots, x_2, x_1, x_0) = \sum_{0}^{2^n - 1} a_i m_i \qquad (4.1)$$

minterm coefficient a_i ε (0, 1)

a_i = 1 if i is a minterm number.

F(A,B,C,D) = Σ m(0, 1, 2, 3, 8, 10, 14)

$a_0 = a_1 = a_2 = a_3 = a_8 = a_{10} = a_{14} = 1$

$a_4 = a_5 = a_6 = a_7 = a_9 = a_{11} = a_{12} = a_{13} = a_{15} = 0$

4.2 Incompletely Specified Functions

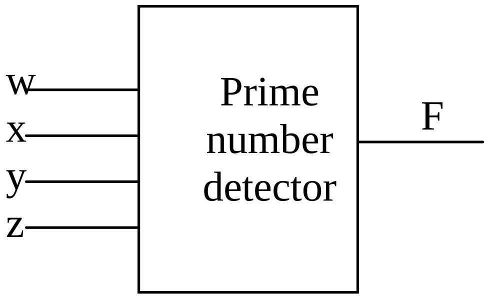


Figure 4.2 Block diagram for prime number detector.

Table 4.8 Truth table for the prime number detector in Figure 4.2.

w	x	y	z	F(w,x,y,z)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	d
1	0	1	1	d
1	1	0	0	d
1	1	0	1	d
1	1	1	0	d
1	1	1	1	d

$F(w,x,y,z) = \Sigma\, m(1, 2, 3, 5, 7) + d(10, 11, 12, 13, 14, 15)$

$F(w,x,y,z) = \pi\, M(0, 4, 6, 8, 9) \bullet D(10, 11, 12, 13, 14, 15)$

$$F(w,x,y,z) = \Sigma\, m(1, 2, 3, 5, 7) + d(10, 11, 12, 13, 14, 15)$$

$$\begin{aligned}
F(w,x,y,z) &= \Sigma\, m(1, 2, 3, 5, 7) \\
&= w'x'y'z + w'x'yz' + w'x'yz + w'xy'z + w'xyz \\
&= w'x'y'z + w'x'yz' + w'x'yz + w'x'yz + w'xy'z + w'xyz \\
&= (w'x'y'z + w'x'yz) + (w'x'yz' + w'x'yz) + (w'xy'z + w'xyz) \\
&= w'x'z + w'x'y + w'xz \\
&= w'x'y + w'z
\end{aligned}$$

$$\begin{aligned}
F(w,x,y,z) &= \Sigma\, m(1, 2, 3, 5, 7, 10, 11) \\
&= w'x'y + w'z + wx'yz' + wx'yz \\
&= w'x'y + w'z + wx'y \\
&= x'y + w'z
\end{aligned}$$

4.3 Expansion of Boolean Functions

Shannon's expansion theorem

$$F(x_{n-1}, x_{n-2}, ..., x_{i+1}, x_i, x_{i-1}, ..., x_2, x_1, x_0) = x_i' F_{x_i=0} + x_i F_{x_i=1} \qquad (4.2)$$

where $$F_{x_i=0} = F(x_{n-1}, x_{n-2}, ..., x_{i+1}, x_i = 0, x_{i-1}, ..., x_2, x_1, x_0) \qquad (4.3a)$$

and $$F_{x_i=1} = F(x_{n-1}, x_{n-2}, ..., x_{i+1}, x_i = 1, x_{i-1}, ..., x_2, x_1, x_0) \qquad (4.3b)$$

are called sub-functions of F and x_i is called an expansion variable.

x_i	Left-hand-side of (4.2)	Right-hand-side of (4.2)
0	$F(x_{n-1}, x_{n-2}, ..., x_{i+1}, x_i = 0, x_{i-1}, ..., x_2, x_1, x_0) = F_{x_i=0}$	$(0)'F_{x_i=0} + (0)F_{x_i=1} = F_{x_i=0}$
1	$F(x_{n-1}, x_{n-2}, ..., x_{i+1}, x_i = 1, x_{i-1}, ..., x_2, x_1, x_0) = F_{x_i=1}$	$(1)'F_{x_i=0} + (1)F_{x_i=1} = F_{x_i=1}$

Binary Tree

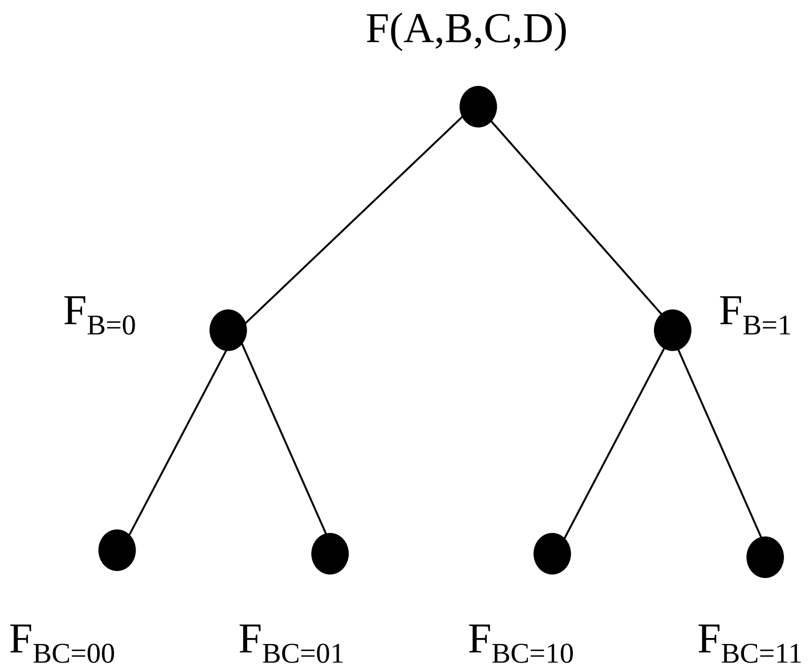


Figure 4.3 A binary tree for the expansion of a Boolean function.

Shannon's expansion theorem

$F(x_{n-1}, x_{n-2}, ..., x_{i+1}, x_i, x_{i-1}, ..., x_2, x_1, x_0) = x_i' F_{x_i=0} + x_i F_{x_i=1}$ (4.2)

$F(A,B,C,D) = A'B'C + BC' + AC'D + ABD'$ (4.4)

B : expansion variable

$F_{B=0} = F(A, B = 0, C, D)$
$= A'(0)'C + (0)C' + AC'D + A(0)D' = A'C + AC'D$ (4.5a)

$F_{B=1} = F(A, B = 1, C, D)$
$= A'(1)'C + (1)C' + AC'D + A(1)D' = C' + AC'D + AD' = C' + AD$ (4.5b)

$F(A,B,C,D) = B'(\underline{A'C + AC'D}) + B(\underline{C' + AD'})$ (4.6)

$F(A,B,C,D) = B'(\underline{A'C + AC'D}) + B(\underline{C' + AD'})$ (4.6)

Sub-functions of $F_{B=0} = A'C + AC'D$ are

$F_{BC=00} = F(A, B = 0, C = 0, D) = A'(0) + A(0)'D = AD$ (4.7a)

$F_{BC=01} = F(A, B = 0, C = 1, D) = A'(1) + A(1)'D = A'$ (4.7b)

$F_{B=0} = A'C + AC'D = C'\ (\underline{AD}) + C\ (\underline{A'})$ (4.8)

Sub-functions of $F_{B=1} = C' + AD'$ are

$F_{BC=10} = F(A, B = 1, C = 0, D) = (0)' + AD' = 1$ (4.9a)

$F_{BC=11} = F(A, B = 1, C = 1, D) = (1)' + AD' = AD'$ (4.9b)

$F_{B=1} = C' + AD' = C'(\underline{1}) + C(\underline{AD'})$ (4.10)

$$
\begin{aligned}
F(A,B,C,D) &= B'(\underline{A'C + AC'D}) + B(\underline{C' + AD'}) \\
&= B'\ [C'\ (\underline{AD}) + C\ (\underline{A'})] + B\ [C'(\underline{1}) + C(\underline{AD'})] \\
&= B'C'\ (\underline{AD}) + B'C\ (\underline{A'}) + BC'(\underline{1}) + BC(\underline{AD'}) \\
&= B'C'\ \underline{F_{BC=00}} + B'C\ \underline{F_{BC=01}} + BC'\ \underline{F_{BC=10}} + BC\ \underline{F_{BC=11}} \qquad (4.11)
\end{aligned}
$$

4.4 Functionally Complete Set

Functionally complete set : AND, OR, NOT

NAND (Not AND) $F(x,y) = (x\ y)'$

Table 4.9 Truth table for NAND.

x	y	F(x,y)
0	0	1
0	1	1
1	0	1
1	1	0

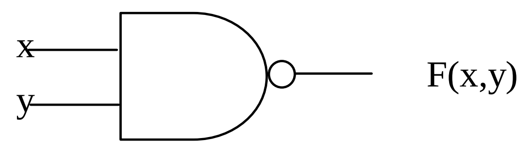


Figure 4.4 Logic symbol for NAND gate.

$x' = (x\ x)'$ $x\ y = [\ (x\ y)'\ ]'$ $x + y = (x'\ y')'$

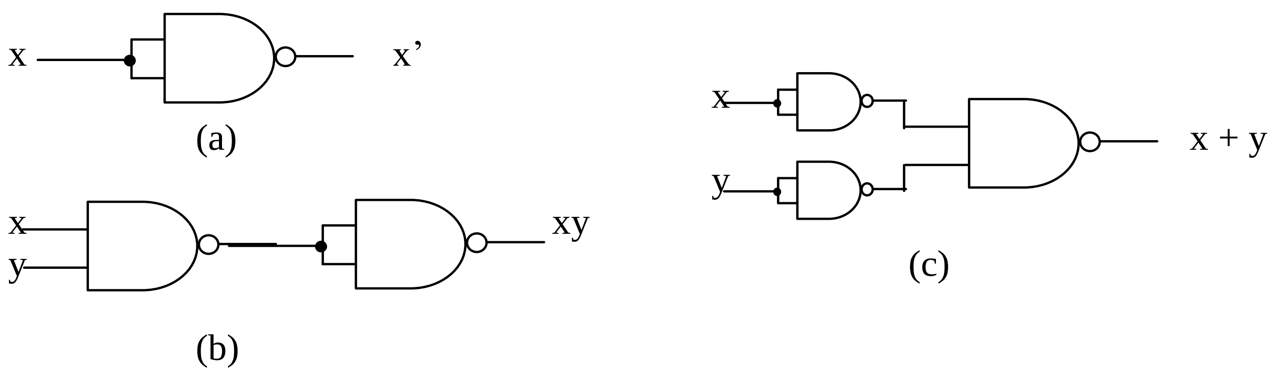


Figure 4.5 Implementation of NOT, AND, OR using NAND gates.

NOR (Not OR) F(x,y) = (x + y)’

Table 4.10 Truth table for NOR.

x	y	F(x,y)
0	0	1
0	1	0
1	0	0
1	1	0

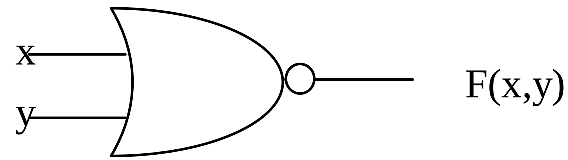


Figure 4.6 Logic symbol for NOR gate.

x’ = (x + x)’ x + y = [(x + y)’]’ x y = (x’ + y’)’

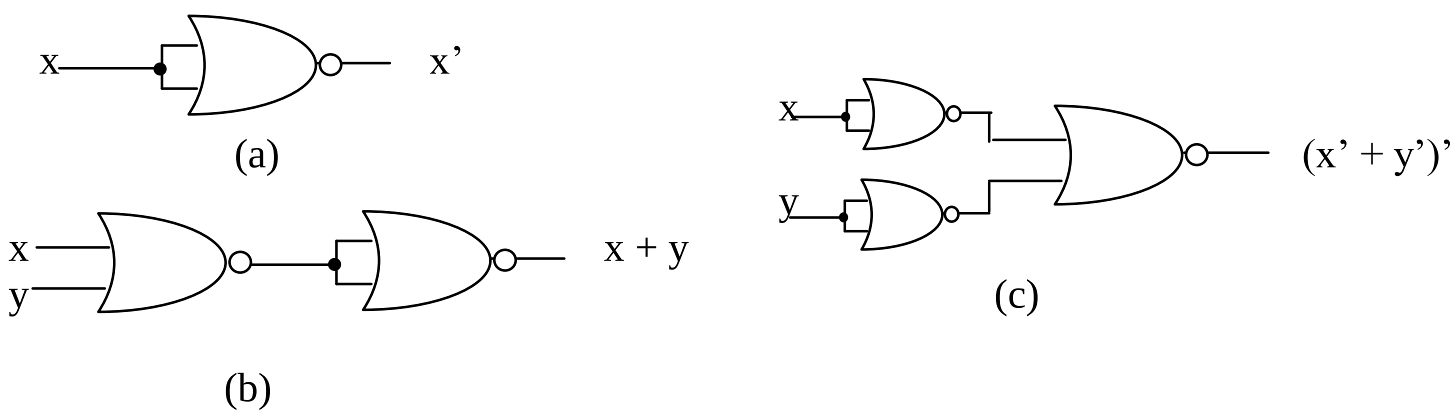


Figure 4.7 Implemantation of NOT, OR, AND using NOR gates.

4.5 Exclusive-OR and Exclusive-NOR

EXCLUSIVE-OR (XOR) $F(x, y) = x \oplus y$

Table 4.11 Truth table for EXCLUSIVE-OR.

x	y	F(x,y)
0	0	0
0	1	1
1	0	1
1	1	0

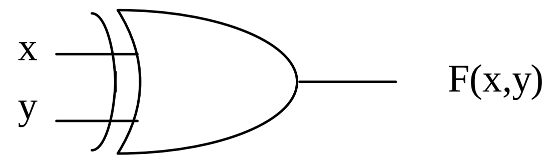


Figure 4.8 Logic symbol for EXCLUSIVE-OR gate.

EQUIVALENCE, Exclusive-NOR (XNOR) $F(x, y) = (x \oplus y)' = x \odot y$

Table 4.12 Truth table for EXCLUSIVE-NOR.

x	y	F(x,y)
0	0	1
0	1	0
1	0	0
1	1	1

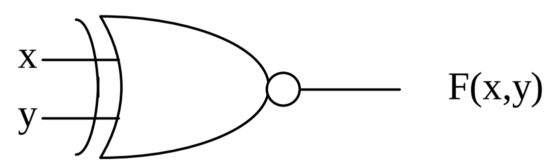


Figure 4.9 Logic symbol for EXCLUSIVE-NOR gate.

Basic laws for XOR

(1) Law of 0	$x \oplus 0 = x$
(2) Law of 1	$x \oplus 1 = x'$
(3) Idempotency law	$x \oplus x = 0$
(4) Complementary law	$x \oplus x' = 1$
(5) Commutative law	$x \oplus y = y \oplus x$

$$x \oplus y = x'y + xy' = (x' + y')(x + y)$$

$$x \odot y = (x \oplus y)' = x'y' + xy = (x' + y)(x + y')$$

$$(x \oplus y)' = x \oplus y' = x' \oplus y$$

4.6 Timing Diagram and Propagation Delay

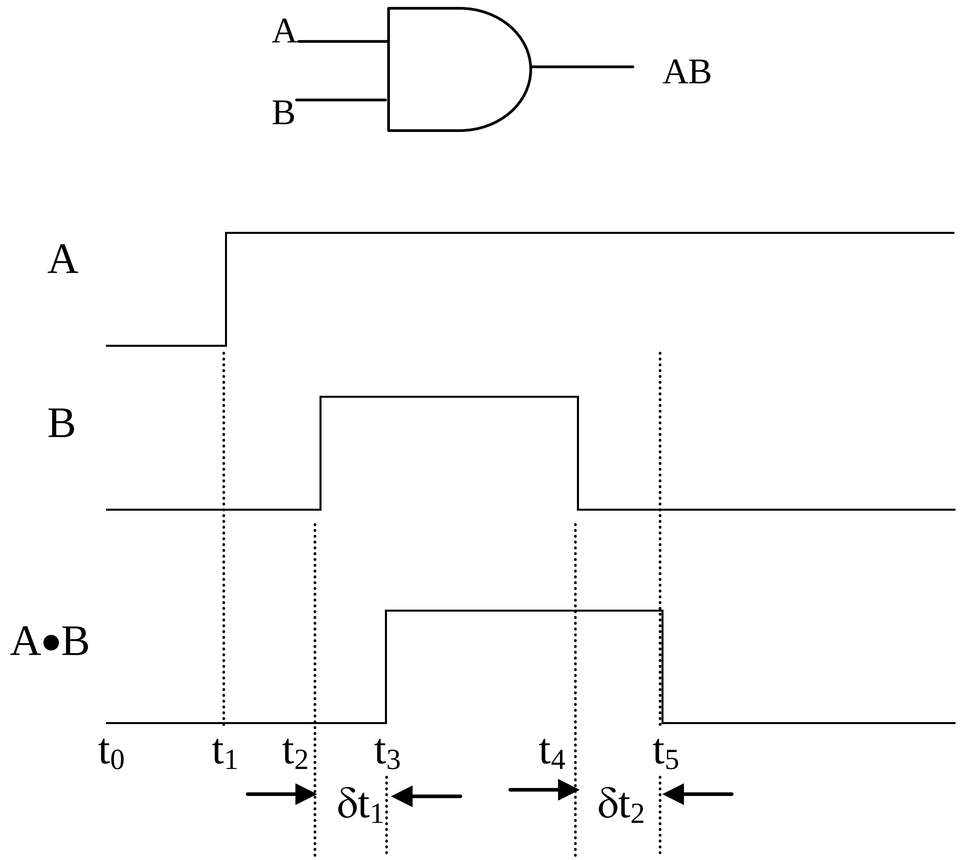


Figure 4.10 Timing diagrams for a 2-input AND gate with propagation delay.

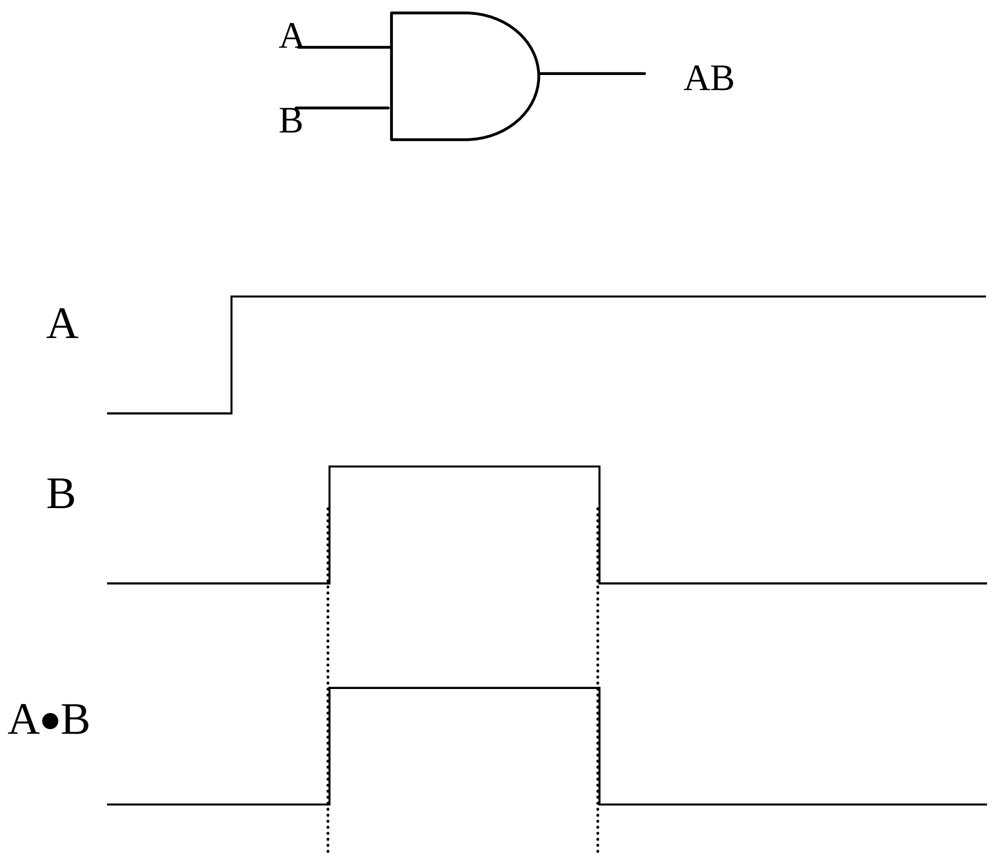


Figure 4.11 Timing diagram for a 2-input AND gate without propagation delay.

Timing diagrams

$$F(A,B,C) = \Sigma\, m(1, 2, 3, 5, 7)$$

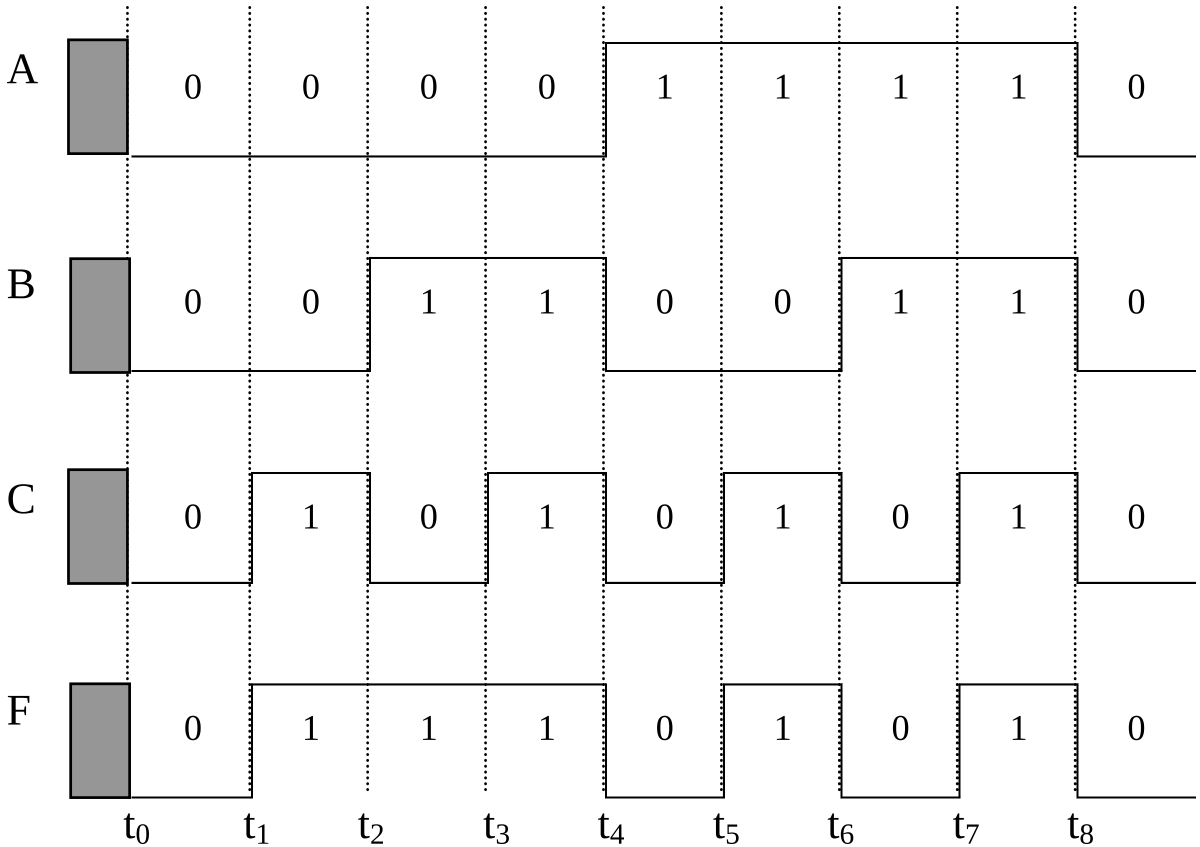


Figure 4.12 Timing diagram for prime number detector.

4.6 Analysis of Combinational Circuits

$P_1 = BC' + D'$

$P_2 = (BD)' = B' + D'$

$P_3 = (A\,P_1)' = [A\,(BC' + D')]'$

$P_4 = (C\,P_2)' = [C\,(B'D + BD')]'$

$F(A,B,C,D) = (P_3\,P_4)'$

$= \{[A\,(BC' + D')]' \bullet [C\,(B' + D')]'\}'$

$= A\,(BC' + D') + C\,(B' + D')$

$= ABC' + AD' + B'C + CD'$

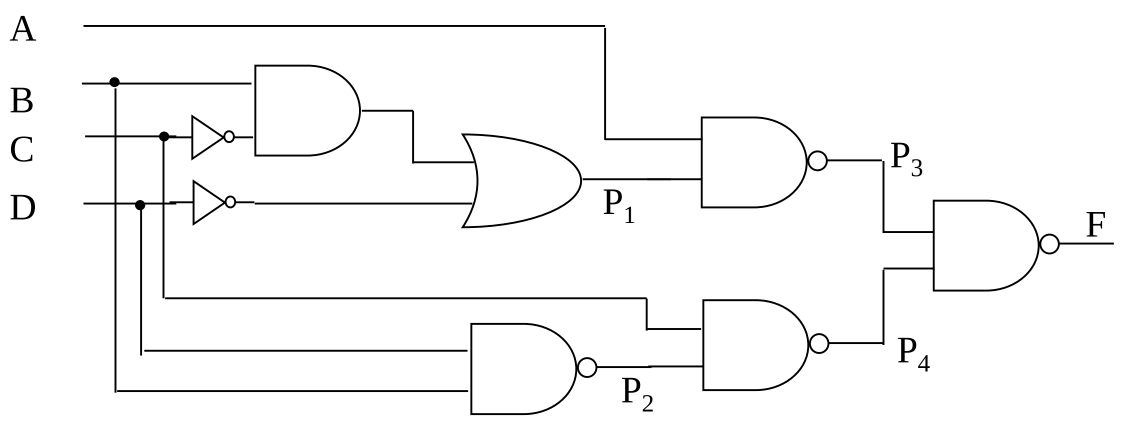


Figure 4.13 Circuit for Example 4.4.

Eliminating of internal inversions by gate equivalencies

(a) $(x_1 \bullet x_2 \bullet x_3 \bullet \ldots\ldots\ldots \bullet x_{n-1} \bullet x_n)' = x_1' + x_2' + x_3' + \ldots\ldots\ldots + x_{n-1}' + x_n'$

(b) $(x_1 + x_2 + x_3 + \ldots\ldots\ldots + x_{n-1} + x_n)' = x_1' \bullet x_2' \bullet x_3' + \ldots\ldots\ldots \bullet x_{n-1}' \bullet x_n'$

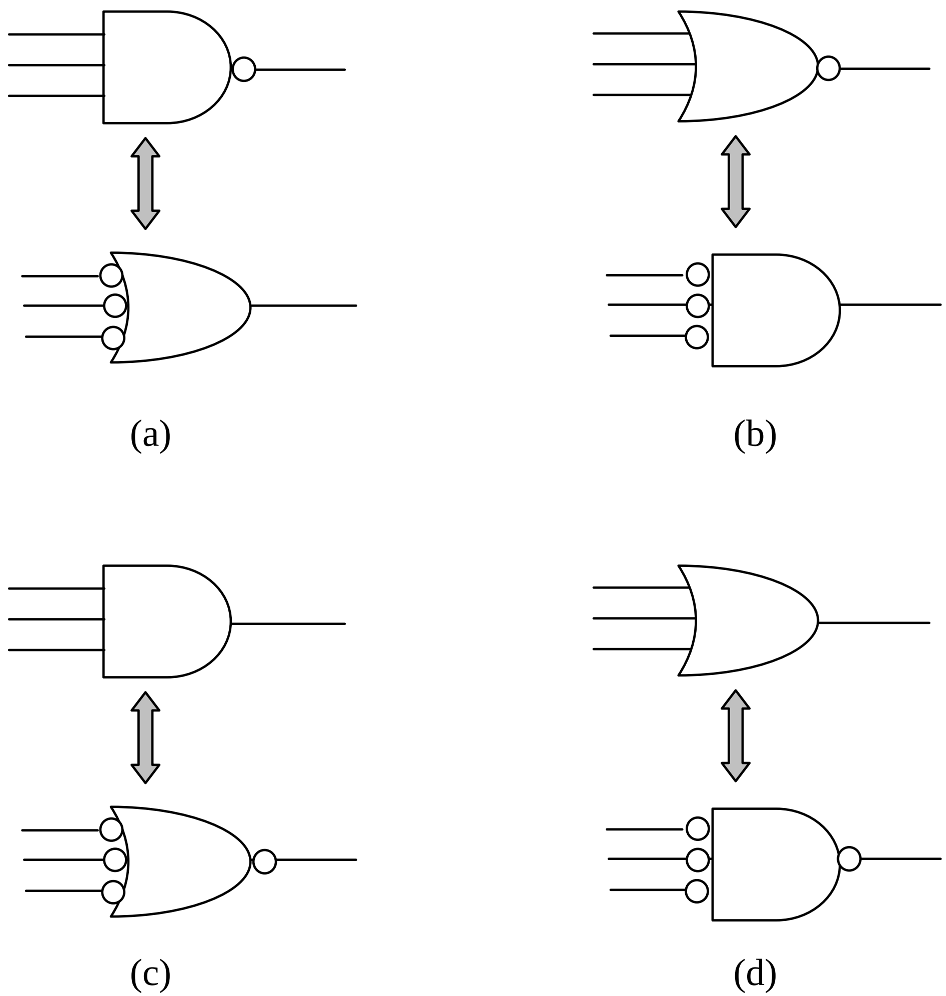


Figure 4.14 Gate equivalencies using DeMorgan's theorem.

❖ Example 4.5

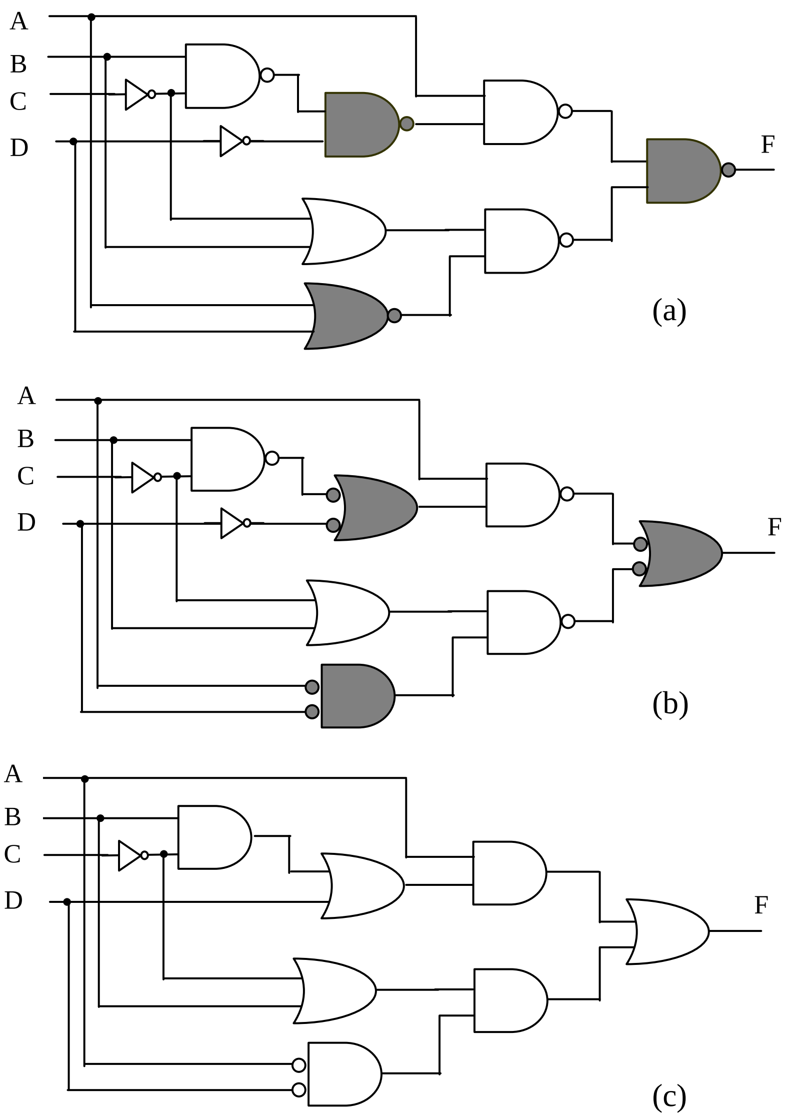
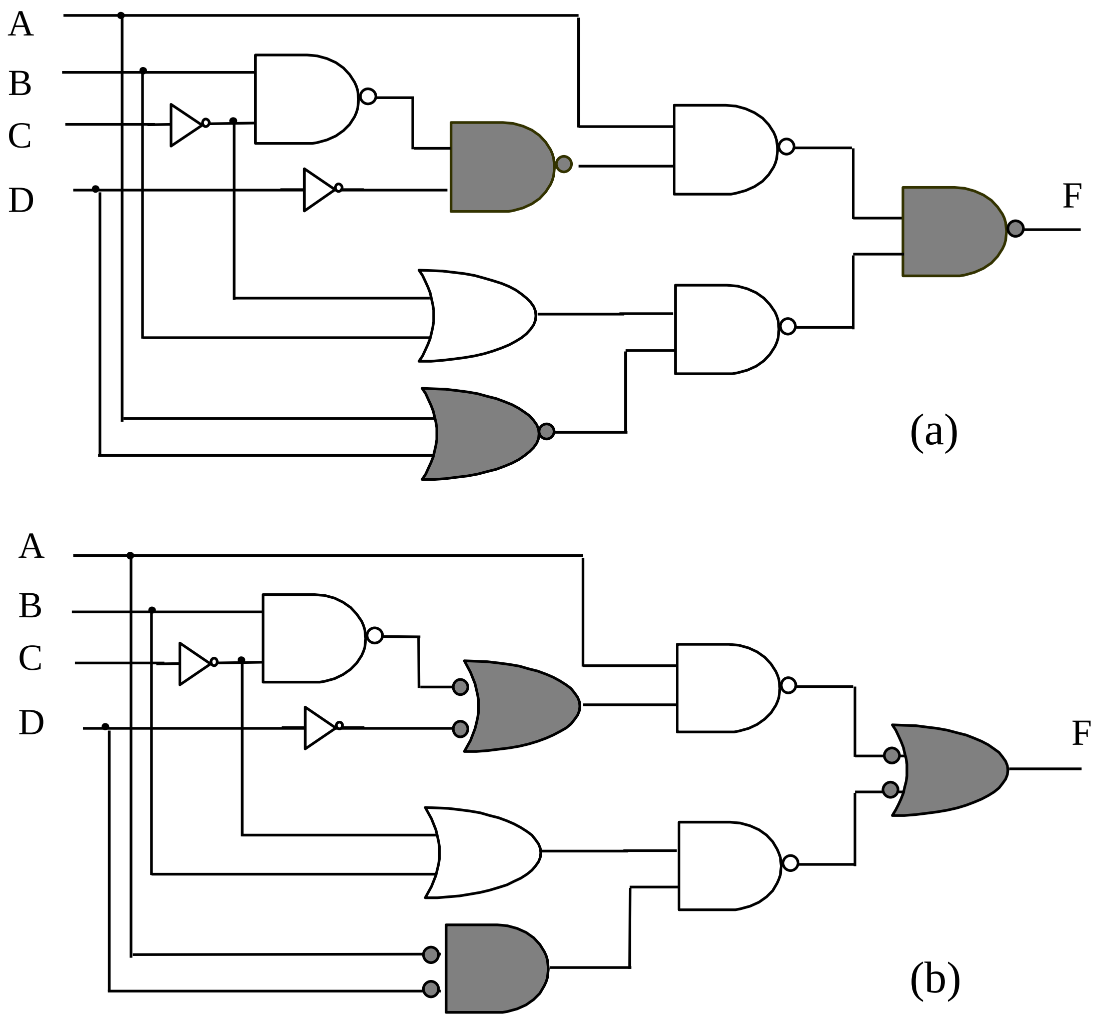


Figure 4.15 Circuit for Example 4.5

F = A(D + BC') +A'D' (B + C') = AD + ABC' + A'BD' + A'C'D'

❖ Example 4.5



(a)

(b)

Example 4.5 (continued)

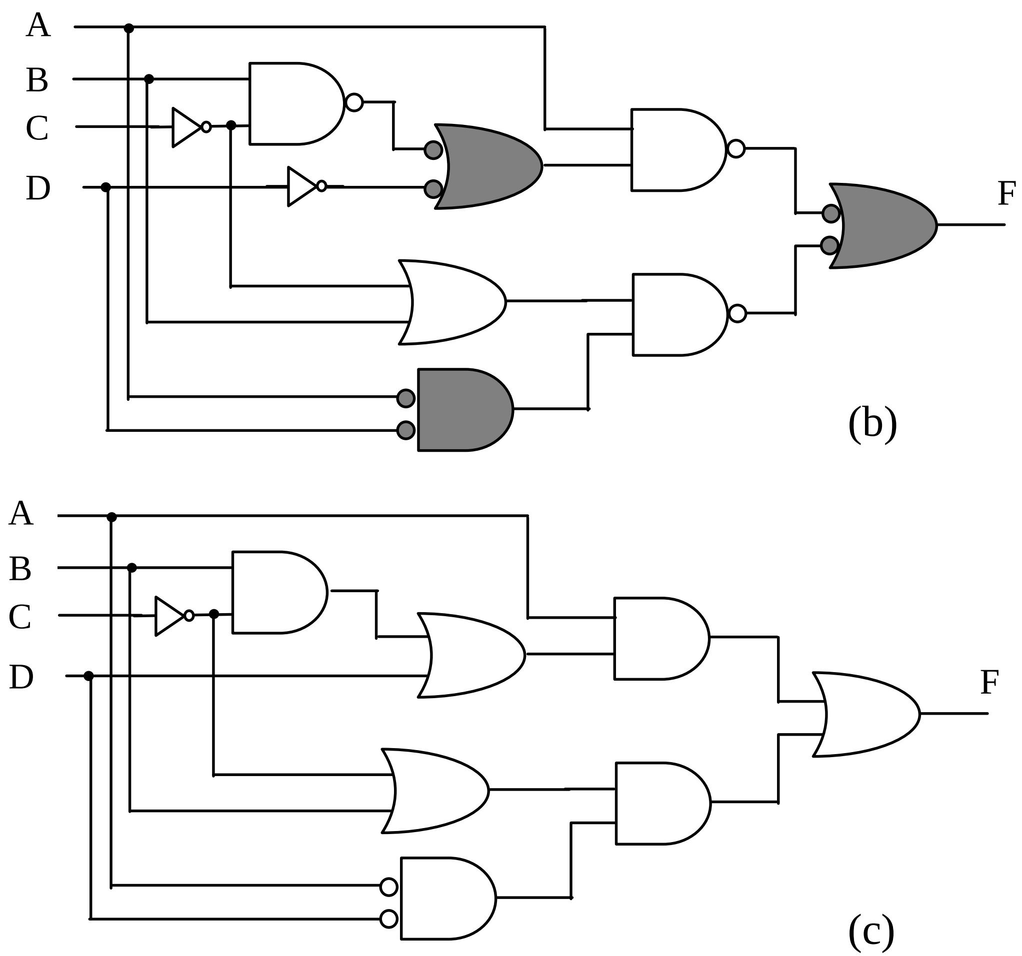


Figure 4.15 Circuit for Example 4.5

$$F = A(D + BC') + A'D'(B + C') = AD + ABC' + A'BD' + A'C'D'$$

❖ Example 4.6

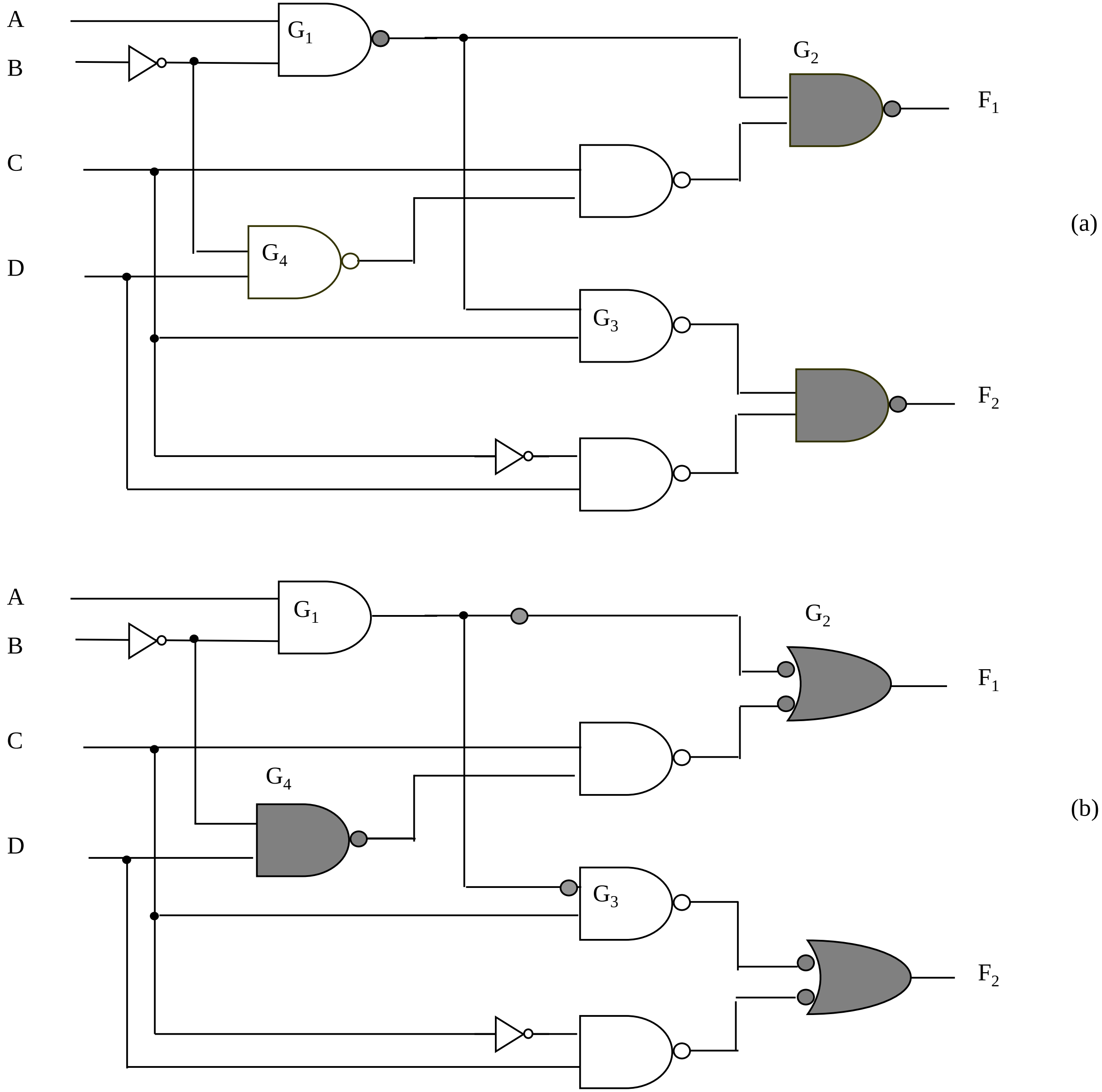
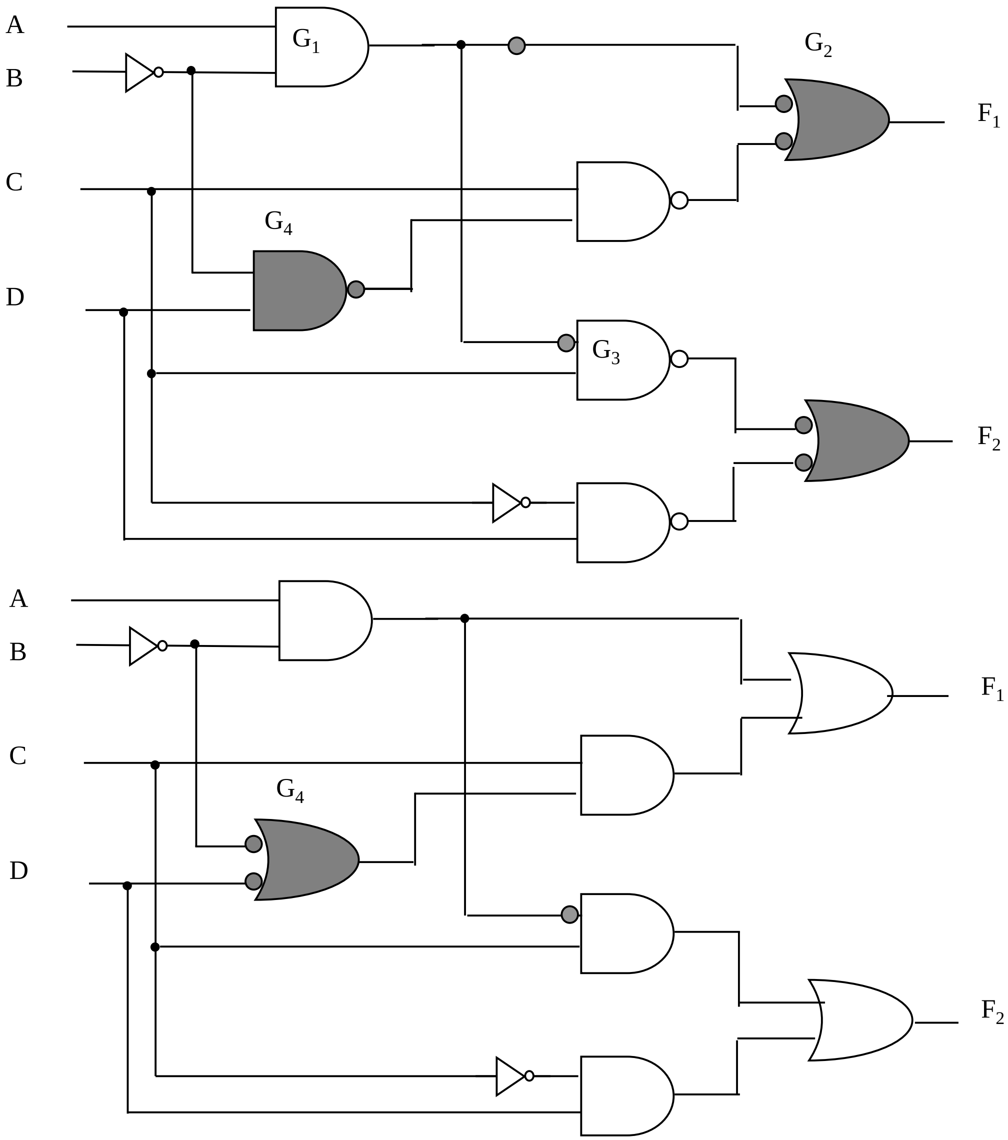


Figure 4.16 Circuit for Example 4.6.



(b)

(c)

Figure 4.16 Circuit for Example 4.6.

$F_1 = AB' + (B + D')C = AB' + BC + CD'$

$F_2 = (AB')'C + C'D = (A' + B)C + C'D = A'C + BC + C'D$

4.8 Assertion and Signal Level

Asserted	Condition : True	Operation : Executed
De-asserted	Condition : False	Operation : Not executed

Active-high signal	Asserted	High (1)
	De-asserted	Low (0)
Active-low signal	Asserted	Low (0)
	De-asserted	High (1)

Table 4.13 Truth table for providing service by a printer.

(a)

Idle	Request	Print
0	0	0
0	1	0
1	0	0
1	1	1

(b)

Idle	Request	/Print
0	0	1
0	1	1
1	0	1
1	1	0

(c)

/Idle	/Request	Print
0	0	1
0	1	0
1	0	0
1	1	0

(d)

/Idle	/Request	/Print
0	0	0
0	1	1
1	0	1
1	1	1

(a) Print = Idle • Request

(b) /Print = (Idle • Request)’

(c) Print = (/Idle)’ • (/Request)’

(d) /Print = (/Idle) + (/Request) = [(/Idle)’ • (/Request)’]’

(a) Print = Idle • Request

(b) /Print = (Idle • Request)’

(c) Print = (/Idle)’ • (/Request)’

(d) /Print = (/Idle) + (/Request) = [(/Idle)’ • (/Request)’]’

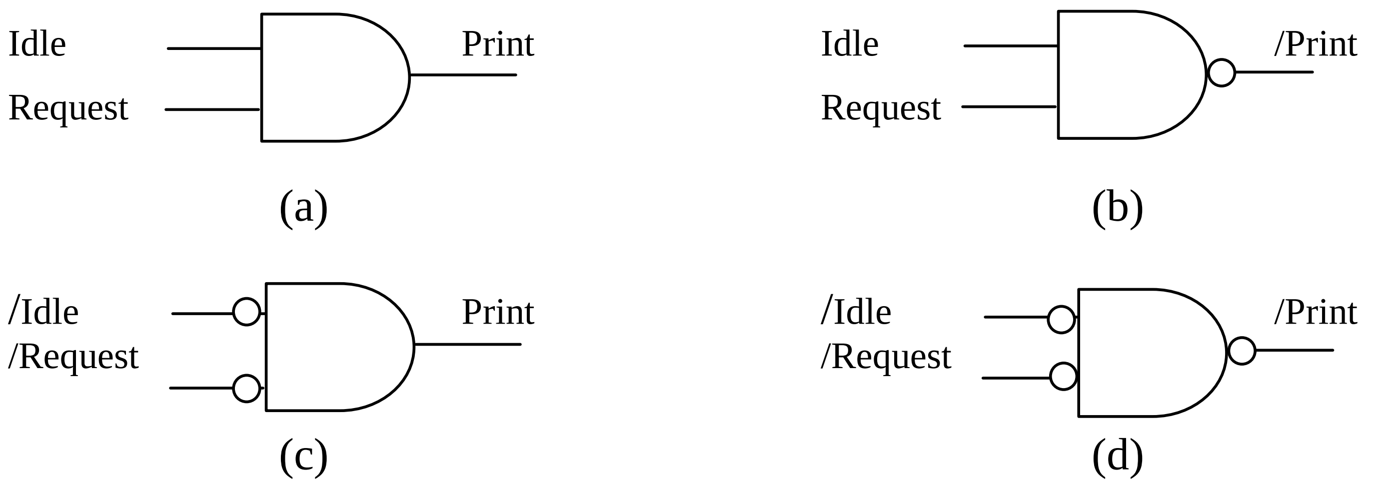


Figure 4.17 Implementation of the truth tables in Table 4.13.