

POS Map Method

- Recall that Canonical Product of Maxterms Involves the 0's of a Function
- Map Method can be Used to Obtain a Minimized POS Form
- 0's are Circled
- POS Terms are Obtained Similar to SOP but with Complemented Literals

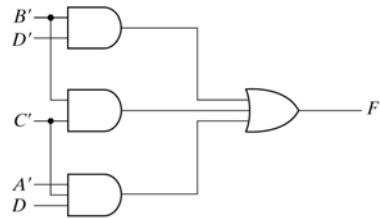
POS Map Method Example

		<i>CD</i>		<i>C</i>	
		00	01	11	10
<i>AB</i>	00	1	1	0	1
	01	0	1	0	0
	11	0	0	0	0
	10	1	1	0	1

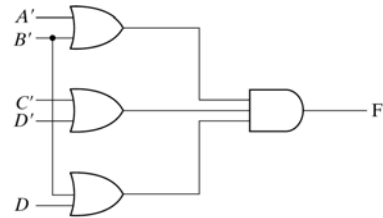
D

Fig. 3-14 Map for Example 3-8; $F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10)$
 $= B'D' + B'C' + A'C'D = (A' + B')(C' + D')(B' + D)$

POS Map Method Example



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



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

Fig. 3-15 Gate Implementation of the Function of Example 3-8

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Find SOP and POS Forms

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

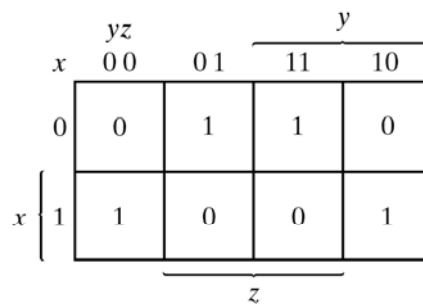


Fig. 3-16 Map for the Function of Table 3-2

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Find SOP Form

Exclusive-OR - y is a Redundant Variable

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$f = x\bar{z} + \bar{x}z = x \oplus z$$

	yz			
x	00	01	11	10
0	0	1	1	0
1	1	0	0	1

Fig. 3-16 Map for the Function of Table 3-2

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Find POS Forms

Exclusive-OR - y is a Redundant Variable

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$f = (x+z)(\bar{x}+\bar{z}) = x \oplus z$$

	yz			
x	00	01	11	10
0	0	1	1	0
1	1	0	0	1

Fig. 3-16 Map for the Function of Table 3-2

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Don't Cares

- All Boolean Functions Evaluate to 0 or 1
- Designers Often Partially Specify a Function
 - Certain Input Combinations May Never Occur
 - The Output for Certain Inputs May not Matter
- In Such Cases a “Don't Care” is Specified
- Leads to a “Don't Care” Minterm
- Usually Denoted by an “X” in Truth Table

Don't Care Example

- Circuit is Needed that Detects When an EVEN-Valued BCD Digit is Input
- Recall BCD is 4-bit Binary Value of Decimal Digits 0-9
- Form Truth Table
- Form Karnaugh Map
- Find Minimized SOP
- Draw Circuit Diagram

Don't Care Example (cont)

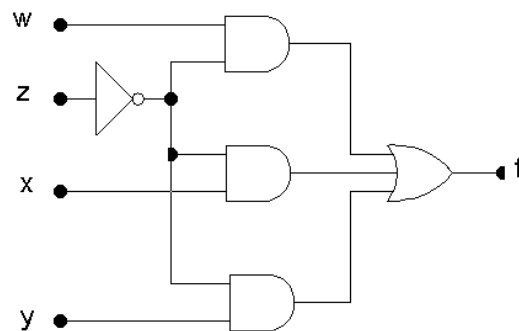
w	x	y	z	f	BCD Digit
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	2
0	0	1	1	0	3
0	1	0	0	1	4
0	1	0	1	0	5
0	1	1	0	1	6
0	1	1	1	0	7
1	0	0	0	1	8
1	0	0	1	0	9
1	0	1	0	X	illegal
1	0	1	1	X	illegal
1	1	0	0	X	illegal
1	1	0	1	X	illegal
1	1	1	0	X	illegal
1	1	1	1	X	illegal

		yz			
		00	01	11	10
wx	00	0	0	0	1
	01	1	0	0	1
	11	X	X	X	X
	10	1	0	X	X

$$f = w\bar{z} + x\bar{z} + y\bar{z}$$

Don't Care Example (cont)

$$f = w\bar{z} + x\bar{z} + y\bar{z}$$

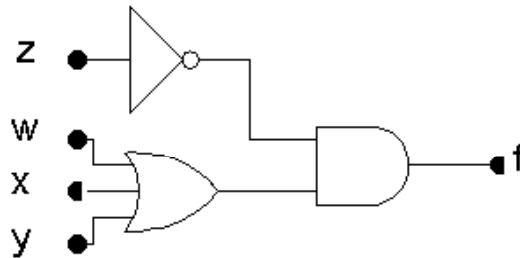


What happens when $wxyz=1100$? How about 1101 ?

Don't Care Example (cont)

$$f = w\bar{z} + x\bar{z} + y\bar{z}$$

$$f = \bar{z}(w + x + y)$$



Another Don't Care Example

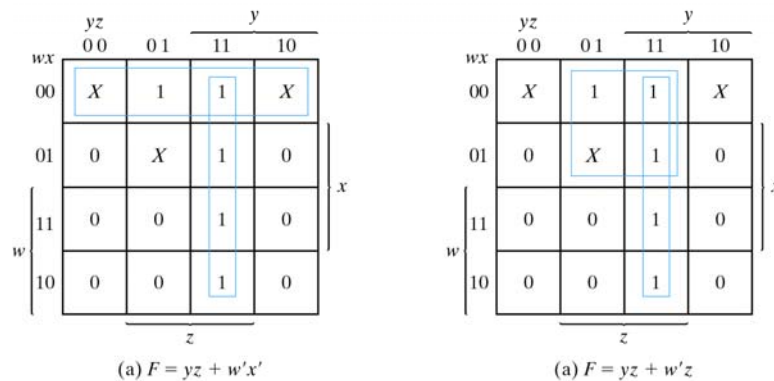


Fig. 3-17 Example with don't-care Conditions

NAND-NAND Logic

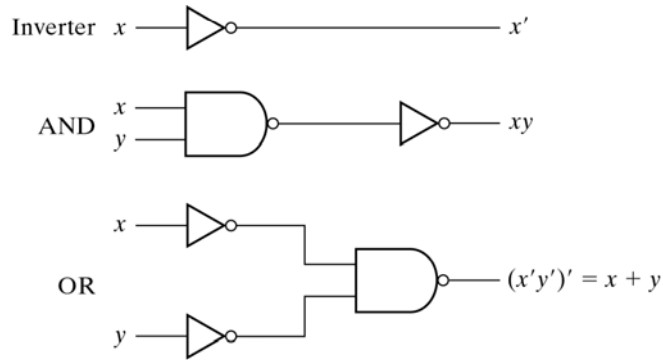


Fig. 3-18 Logic Operations with NAND Gates

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Equivalent NAND Gate Symbols

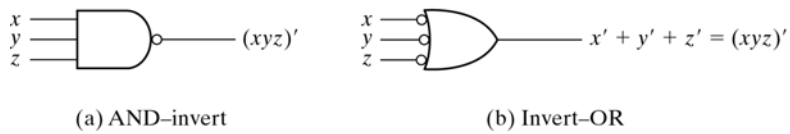
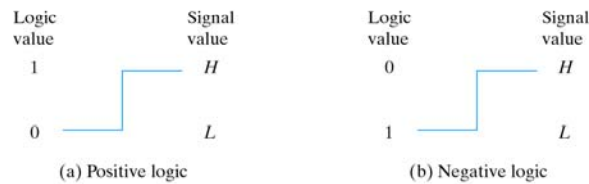


Fig. 3-19 Two Graphic Symbols for NAND Gate

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Positive and Negative Logic

- 0 and 1 are Models to use Algebra
- Circuits use High (H) and Low (L) Values
 - Voltage or Current
- Up to Designer to Interpret if $H \rightarrow 0$ or $H \rightarrow 1$
- Interpretations Called 'Positive'/'Negative' Logic



© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

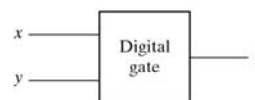
Fig. 2-9 signal assignment and logic polarity

Positive and Negative Logic Example and Notation

*Positive-Logic
AND becomes
Negative-Logic
OR*

<i>x</i>	<i>y</i>	<i>F</i>
<i>L</i>	<i>L</i>	<i>L</i>
<i>L</i>	<i>H</i>	<i>L</i>
<i>H</i>	<i>L</i>	<i>L</i>
<i>H</i>	<i>H</i>	<i>H</i>

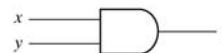
(a) Truth table with *H* and *L*.



(b) Gate block diagram

<i>x</i>	<i>y</i>	<i>z</i>
0	0	0
0	1	0
1	0	0
1	1	1

(c) Truth table for positive logic



(d) Positive logic AND gate

<i>x</i>	<i>y</i>	<i>z</i>
1	1	1
1	0	1
0	1	1
0	0	0

(e) Truth table for negative logic



(f) Negative logic OR gate

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Fig. 2-10 Demonstration of positive and negative logic

AND/OR to NAND/NAND

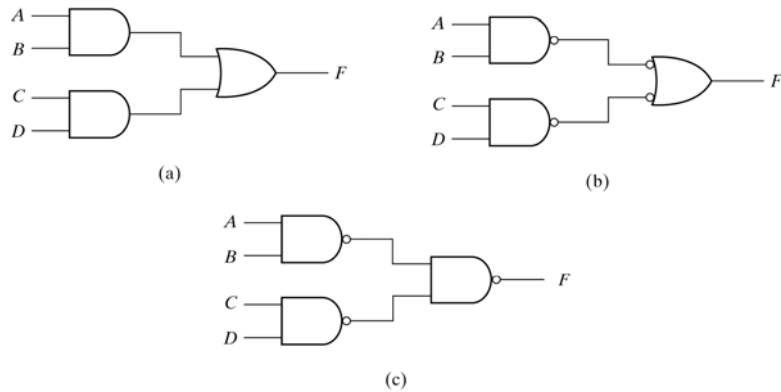


Fig. 3-20 Three Ways to Implement $F = AB + CD$

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

NAND/NAND Example

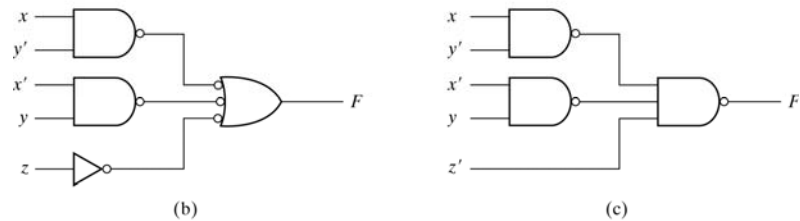
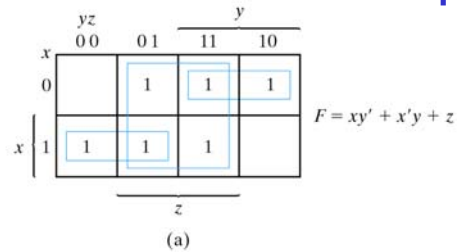
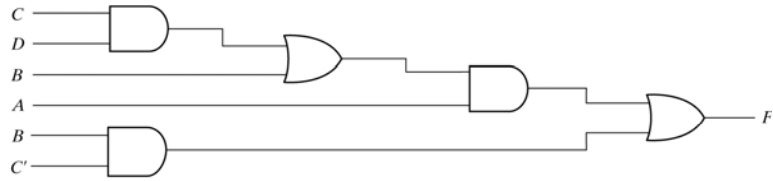


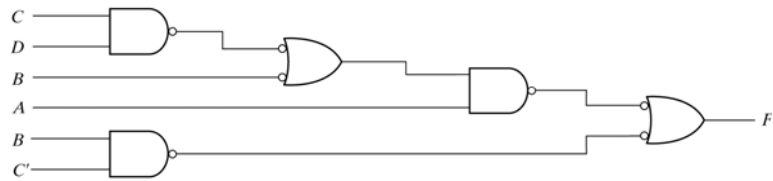
Fig. 3-21 Solution to Example 3-10

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Multilevel NAND Example



(a) AND-OR gates

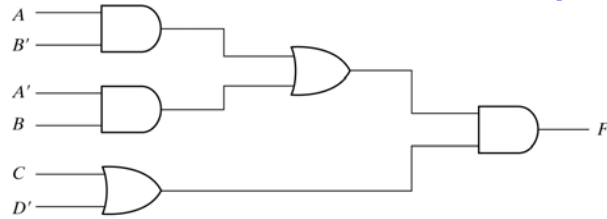


(a) NAND gates

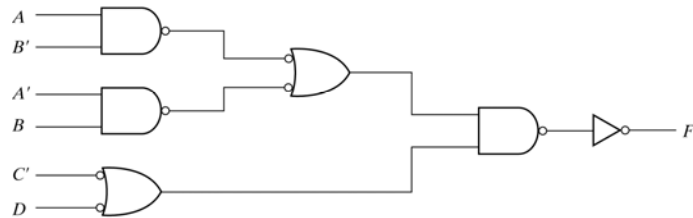
Fig. 3-22 Implementing $F = A(CD + B) + BC$

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Multilevel NAND Example



(a) AND-OR gates



(b) NAND gates

Fig. 3-23 Implementing $F = (AB' + A'B)(C + D')$

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

NOR-NOR Logic

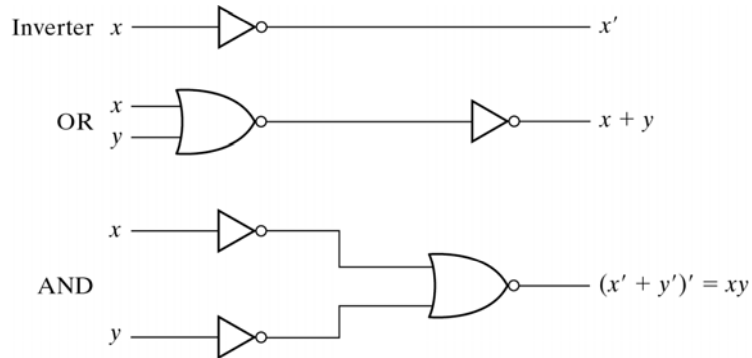


Fig. 3-24 Logic Operations with NOR Gates

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Equivalent NOR Gate Symbols

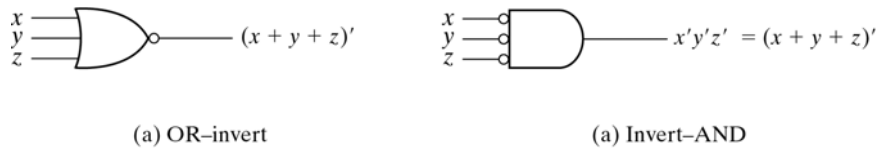


Fig. 3-25 Two Graphic Symbols for NOR Gate

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

NOR/NOR Example

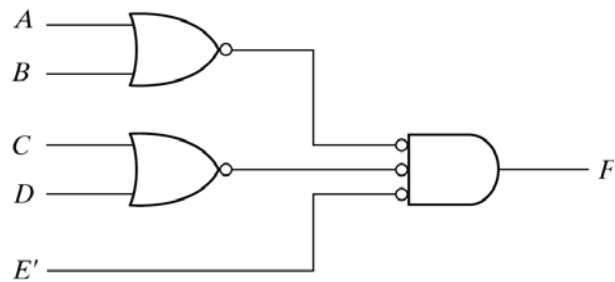


Fig. 3-26 Implementing $F = (A + B)(C + D)E$

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.

Multilevel NOR Example

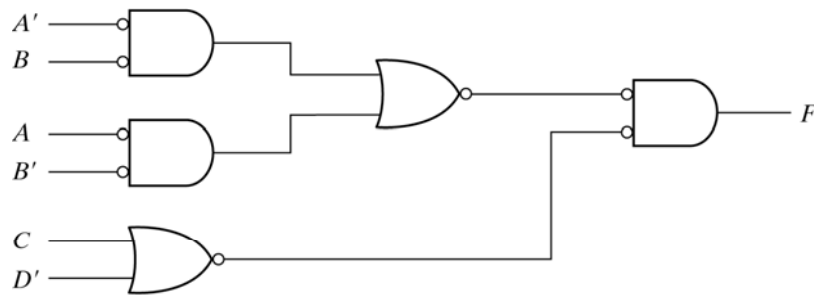


Fig. 3-27 Implementing $F = (AB' + A'B)(C + D')$ with NOR Gates

© 2002 Prentice Hall, Inc.
M. Morris Mano
DIGITAL DESIGN, 3e.