GIVEN f = e IN CSOP, OBTAIN MINIMAL POS

Procedure-CSOP-to-POS

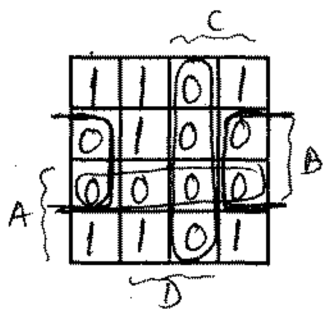
1- MARK f ON THE MAP

2- f′ IS DEFINED BY THOSE SQUARES NOT MARKED AS 1
(f′ IS DEFINED BY THOSE SQUARES MARKED AS 0)

3 - OBTAIN A MINIMAL e FOR f′ IN SOP BY COVERING ALL 0-SQUARES WITH MINIMAL NUMBER OF LARGEST POSSIBLE AREAS OF SIZE 2^k, k = 0, 1, 2,.. FORMED BY COMBINING ADJACENT 0-SQUARES

4- TAKE COMPLEMENT OF f′ IN SOP TO OBTAIN f IN POS

e.g., f(A,B,C,D) = $\sum$m(0, 1, 2, 5, 8, 9, 10) $= B'D' + B'C' + A'C'D$



$$f' = AB + CD + BD'$$

$$f = (f')' = (AB + CD + BD')'$$

$$= (AB)' \cdot (CD)' \cdot (BD')'$$

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

GIVEN f = e in CPOS, OBTAIN MINIMAL SOP

1) Obtain CSOP from the given CPOS

2) Apply **Procedure-CSOP-to-SOP**

GIVEN f = e IN CPOS, OBTAIN MINIMAL POS

1) Obtain CSOP from the given CPOS

2) Apply **Procedure-CSOP-to-POS**

e.g., Given $f(x,y,z) = \Pi M(0, 2, 5, 7)$ in CPOS,

A) Obtain minimal SOP

$f(x,y,z) = \Pi M(0, 2, 5, 7)$ in CPOS is $f(x,y,z) = \Sigma m(1, 3, 4, 6)$ in CSOP

$$\begin{array}{r|c|c|c|c|} \multicolumn{1}{r}{} & \multicolumn{1}{c}{} & \multicolumn{2}{c}{\overbrace{\qquad}^{y}} & \multicolumn{1}{c}{} \\ \cline{2-5} & 0&1&1&0\\ \cline{2-5} x\{ & 1&0&0&1\\ \cline{2-5} \multicolumn{1}{r}{} & \multicolumn{2}{c}{\underbrace{\qquad}_{z}} & \multicolumn{2}{c}{} \end{array}$$

$$f = xz' + x'z$$

B) Obtain minimal POS

$f(x,y,z) = \Pi M(0, 2, 5, 7)$ in CPOS is $f(x,y,z) = \Sigma m(1, 3, 4, 6)$ in CSOP

$$\begin{array}{r|c|c|c|c|} \multicolumn{1}{r}{} & \multicolumn{1}{c}{} & \multicolumn{2}{c}{\overbrace{\qquad}^{y}} & \multicolumn{1}{c}{} \\ \cline{2-5} & 0&1&1&0\\ \cline{2-5} x\{ & 1&0&0&1\\ \cline{2-5} \multicolumn{1}{r}{} & \multicolumn{2}{c}{\underbrace{\qquad}_{z}} & \multicolumn{2}{c}{} \end{array}$$

$$f' = xz + x'z'$$

$$\Rightarrow f = (x'+z')(x+z)$$

SIMPLIFICATION OF FUNCTIONS WITH DON'T CARE CONDITIONS

× = DON'T CARE is a value of a function f corresponding to an input combination that will never occur

e.g.,

w x y z	f
0 0 0 0	×
0 0 0 1	0
0 0 1 0	×
0 0 1 1	0
0 1 0 0	1
0 1 0 1	1
0 1 1 0	1
0 1 1 1	0
1 0 0 0	1
1 0 0 1	1
1 0 1 0	1
1 0 1 1	0
1 1 0 0	1
1 1 0 1	1
1 1 1 0	1
1 1 1 1	×

$\Rightarrow \quad f(w,x,y,z) = \sum m(4, 5, 6, 8, 9, 10, 12, 13, 14) + d(0, 2, 15)$

y

X			X
1	1		1
1	1	X	1
1	1		1

w, x, z

y

X	0	0	X
		0	
		X	
		0	

w, x, z

A) Obtain minimal SOP

$f = z' + xy' + wy'$

B) Obtain minimal POS

$f' = w'x' + yz$

$f = (w+x)(y'+z')$

FROM STANDARD TO CANONICAL FORMS OF EXPRESSIONS

A) SOP $\rightarrow$ CSOP

EXAMINE EACH TERM IN e
IF TERM IS A MINTERM
THEN CONTINUE WITH THE NEXT TERM
ELSE FOR EACH MISSING x_i, PERFORM $(x_i + x'_i)$ **AND** TERM
SIMPLIFY AND ELIMINATE REDUNDANT TERMS

e.g., $f(x,y,z) = x'y + z' + xyz$
$= (x'y) \bullet (z + z') + (x + x') \bullet (y + y') \bullet z' + xyz$
$= x'yz + x'yz' + xyz' + xy'z' + x'yz' + x'y'z' + xyz$
$= x'yz + x'yz' + xyz' + xy'z' + x'y'z' + xyz$
$= m_3 + m_2 + m_6 + m_4 + m_0 + m_7$
$= \sum m(0, 2, 3, 4, 6, 7)$

B) POS $\rightarrow$ CPOS

EXAMINE EACH TERM IN e
IF TERM IS A MAXTERM
THEN CONTINUE WITH THE NEXT TERM
ELSE FOR EACH MISSING x_i, PERFORM $(x_i \bullet x'_i)$ **OR** TERM
SIMPLIFY AND ELIMINATE REDUNDANT TERMS

e.g., $f(x,y,z) = x' \bullet (y' + z)$
$= (x' + yy' + zz') \bullet (y' + z + xx')$
$= [(x'+y+z)\bullet(x'+y+z')\bullet(x'+y'+z)\bullet(x'+y'+z')]\bullet[(x+y'+z)\bullet(x'+y'+z)]$
$= (x' + y + z)\bullet(x' + y + z')\bullet(x' + y' + z)\bullet(x' + y' + z')\bullet(x + y' + z)$
$= M_4, M_5, M_6, M_7, M_2$
$= \prod M(2, 4, 5, 6, 7)$

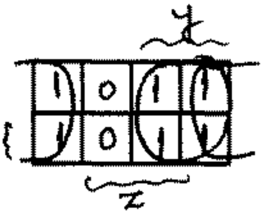
(IF THE VALUE OF A FUNCTION
IS INDEPENDENT OF THE VALUE OF SOME TERM/LITERAL
THEN THAT TERM/LITERAL IS REDUNDANT)

e.g., $f(x,y,z) = x'y'z + yz + xz$
$= z \bullet (x'y' + y + x)$
$= z \cdot (x' + y + x)$
$= z \cdot (y + 1)$
$= z \cdot 1$
$= z$

Since $y + x'y' = (y + x')(y + y')$
$= y' + x'$
by (P4)

SHORT-CUT FOR **SOP** → **CSOP** AND MORE ...

f(x,y,z)= $x'y + z' + xyz$ in SOP



f(x,y,z)= $y + z'$ in minimal SOP

f(x,y,z)= $\sum m(0, 2, 3, 4, 6, 7)$ in CSOP

f(x,y,z)= $\prod M(1, 5)$ in CPOS

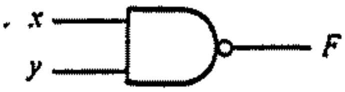
f′(x,y,z)= $\sum m(1, 5)$ in CSOP

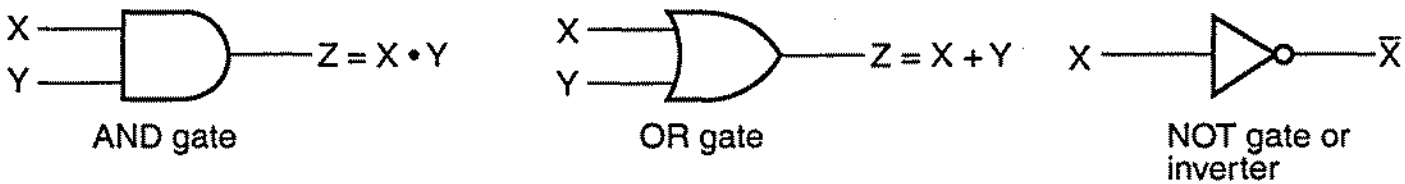
f′(x,y,z)= $\prod M(0, 2, 3, 4, 6, 7)$ in CPOS

5) IMPLEMENTATION OF $f(x_1, x_2, ..., x_n)$

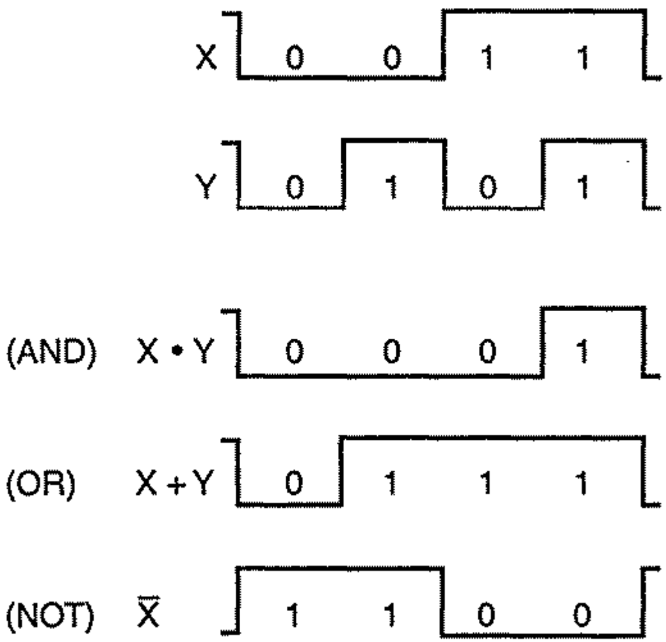
in a BOOLEAN EXPRESSION	in a CIRCUIT
each LITERAL	an INPUT to a gate
each TERM	a GATE

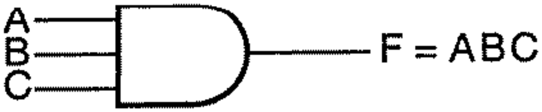
LOGIC GATES

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= x \odot y$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

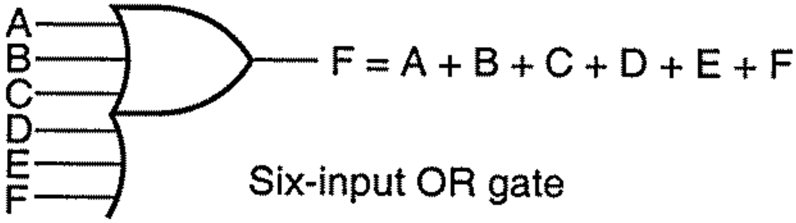


(a) Graphic symbols

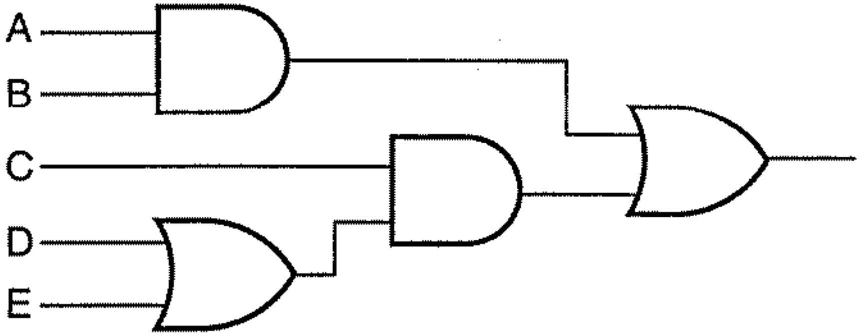




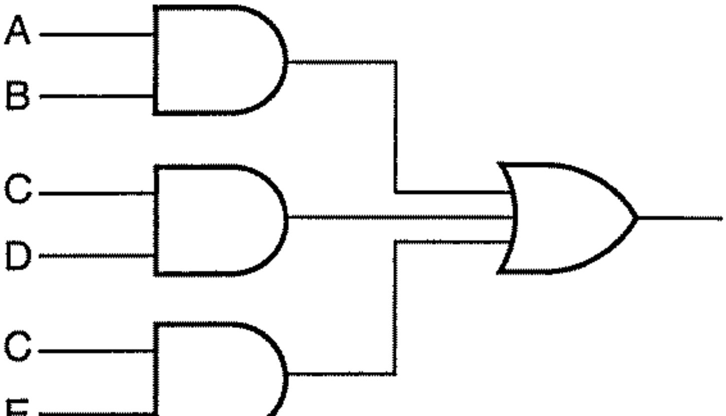
Three-input AND gate

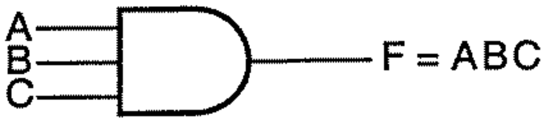


Six-input OR gate

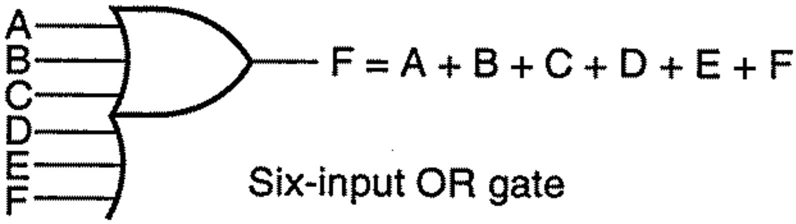


AB + C (D + E)

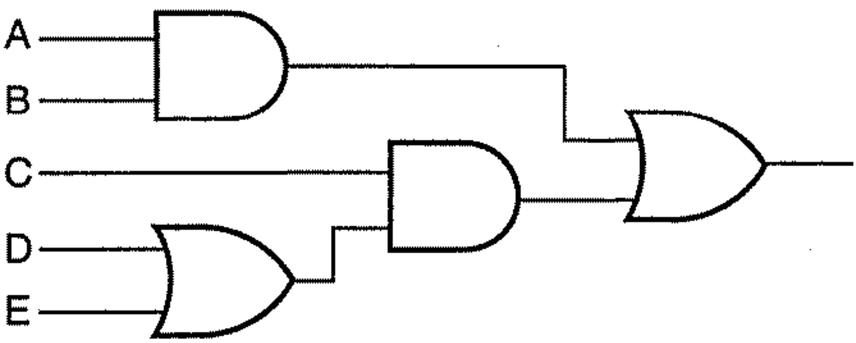




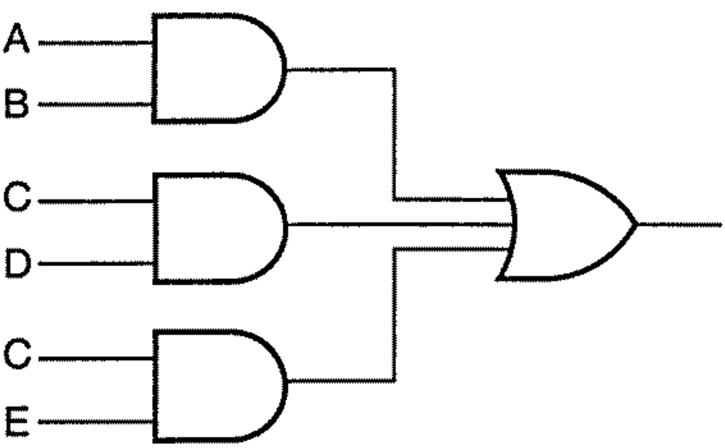
Three-input AND gate



Six-input OR gate



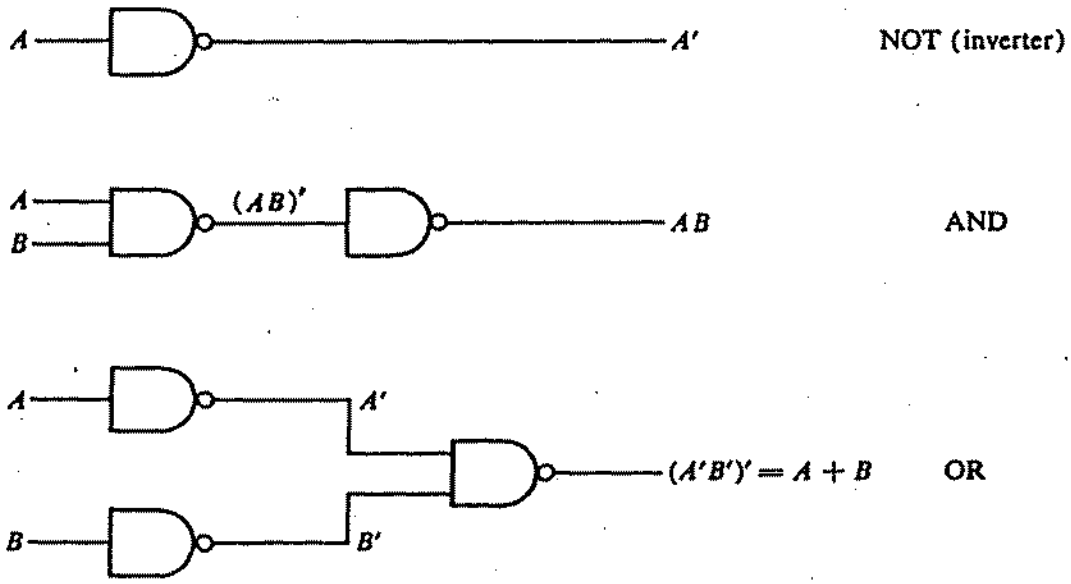
AB + C (D + E)



AB + CD + CE

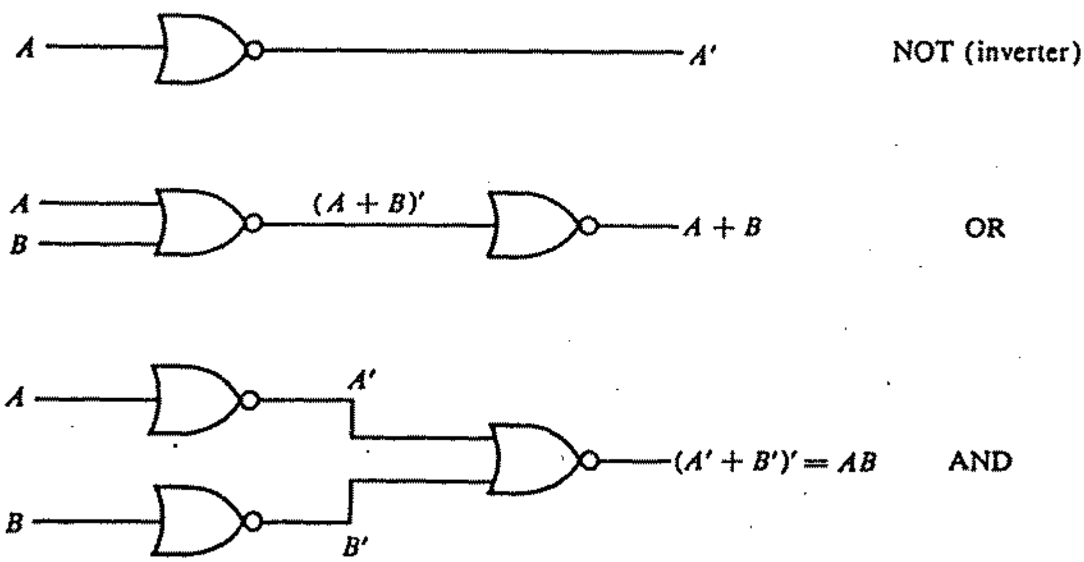
AND - OR - INVERT FORM A FUNCTIONALLY COMPLETE SET

NAND FORMS A FUNCTIONALLY COMPLETE SET



Implementation of NOT, AND, and OR by NAND gates

NOR FORMS A FUNCTIONALLY COMPLETE SET



Implementation of NOT, OR, and AND by NOR gates