Digital Design

Binary logic consists of binary variables and logical operations.

3 Basic Logical Operations

1. AND 2. OR 3. NOT (complement)

x AND y = z $\quad x \cdot y = z \quad xy = z$

that is, z = 1 if and only if (iff) x = 1 and y = 1

x OR y = z $\quad x + y = z$

that is, z = 1 if x = 1 or if y = 1 or if both x = 1 and y = 1

(i.e. z = 0 iff x = 0 and y = 0)

NOT x = z

that is, z is what x is not

Truth Table:

		x AND y	**x OR y**
x	**y**	**x • y**	**x + y**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

	NOT x
x	**x’**
0	1
1	0

Other Logical Operators:

Functions	Symbol	Name	Remarks
F0 = 0		null	
F1= xy	x • y	and	
F2 = xy’	x/y	inhibition	x but not y
F3 = x		transfer	
F4 = x’y	y/x	inhibition	y but not x
F5 = y		transfer	
F6 = xy’ + x’y	$x \oplus y$	xor	x or y but not both
F7 = x + y	x + y	or	
F8 = (x + y)’	$x \downarrow y$	nor	
F9 = xy + x’y’	x ⊚ y	equivalence*	x equals y
F10 = y’	y’	complement	
F11 = x + y’	$x \subset y$	implication	if y then x
F12 = x’	x’	complement	
F13 = x’ + y	$x \supset y$	implication	
F14 = (xy)’	$x \uparrow y$	nand	
F15 = 1		identity	

Axiomatic Definition of Boolean Algebra

1. Closure
    a. Closure with respect to (wrt) OR (+)
    b. Closure with respect to AND (•)

2. Identity
    a. Identity element wrt to OR : 0
    b. Identity element wrt to AND : 1

3. Commutative Property
    a. Commutative Property wrt to OR : $x + y = y + x$
    b. Commutative Property wrt to AND : $x \cdot y = y \cdot x$

4. Distributive Property
    5. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
    6. $x + (y \cdot z) = (x + y)(x + z)$

5. Existence of Complement
    6. $x + x' = 1$
    7. $x \cdot x' = 0$

Precedence:

(1) Parentheses (2) NOT (3) AND (4) OR

Other Properties:

1. a. $x + x = x$ b. $x \cdot x = x$

2. a. $x + 1 = 1$ b. $x \cdot 0 = 0$

3. Involution: $(x')' = x$

4. Associative Property

    a. $x + (y + z) = (x + y) + z$
    b. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$

5. DeMorgan

    a. $(x + y)' = x' \cdot y'$
    b. $(x \cdot y)' = x' + y'$

6. Absorption

a. $x + x \cdot y = x$
b. $x \cdot (x + y) = x$
c. $x \cdot y + x' \cdot z + y \cdot z = x \cdot y + x' \cdot z$
d. $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$

7. Duality Principle

"Every algebraic expression deducible from the postulates of Boolean Algebra remains valid if the operations and identity elements are interchanged."

+ replaced by •
• replaced by +
0 replaced by 1
1 replaced by 0

examples:

1. $xy + xy' = x (y + y') = x (1) = x$
2. $(x+y)(x+y') = xx + xy' + xy$
3. $xyz + x'y + xyz' = xy + x'y = y$
4. $x + x'y = x + xy + x'y = x + y(x + x') = x + y$
5. $xz + x'yz = z(x + x'y) = z(x + y) = xz + yz$
6. $(x + y)'(x' + y')' = (x'y')(xy) = 0$
7. $wyz' + wyz + xy = y(wz' + wz) + xy = yw + xy$

Simplify the following boolean expressions:

1. $xy + (xy)'z$
2. $wx + xy' + yz + xz'$
3. $(x + y + w)(w + x' + y)(y' + z)(w + z)$
4. A student consults his adviser and finds that he may enroll in Electronics only if he satisfies the following conditions:

    5. He has taken E&M and a Physics major in good standing OR
    6. He has taken E&M and a Physics major and has deparmental approval OR
    7. He has not taken E&M and is a Physics major on probation OR
    8. He is in good standing and has deparmental approval OR
    9. He is a Physics major and does not have deparmental approval.

Solutions:

1. xy + (xy)'z = xy + z (absorption) ans.
2. wx + xy' + yz + xz' = wx + xy' + yz + xz' + xy (reverse absorption)
   = wx + yz + xz' + x
   = x + yz (absorption) ans.
3. (x + y + w)(w + x' + y)(y' + z)(w + z) = (w + y)(y' + z)(w + z) = (w + y)(y' + z) ans.
4. let
   w = student has taken E&M
   x = student is a Physics major
   y = student is in good standing
   z = student has deparmental approval
   F = student is eligible to take the course

   F = wxy + wxz + w'xy' + yz + xz'
   = wxy + w'xy' + yz + x(w + z')
   = wx(y + 1) + w'xy' + yz + xz'
   = wx + w'xy' + yz + xz'
   = x(w + w'y') + yz + xz'
   = x(w + y') + yz + xz'
   = wx + xy' + yz + xz'
   = wx + xy' + yz + (xz) + xz' (rev. abs)
   = wx + xy' + yz + x
   = x + yz (absorption) ans.

Boolean Function

F = binary variables with AND OR NOT ( )

Implementation of Boolean functions with gates.

Canonical & Standard Forms:

x → x or x'

two variables x & y combined with AND:

x,y → xy, xy', x'y, x'y' : minterms or standard product

n variables → forms 2^n minterms

two variables x & y combined with OR:

x,y → x+y, x+y', x'+y, x'+y' : maxterms or standard sum

n variables → forms 2^n maxterms

minterm: if 1 variable is unprimed, if 0 variable is primed (look at 1's).

maxterm: if 1 variable is primed, if 1 variable is unprimed (look at 0's).

	Minterms		Maxterms	
xyz	**term**	**designation**	**term**	**designation**
000	x'y'z'	m_0	x + y + z	M_0
001	x'y'z	m_1	x + y + z'	M_1
010	x'yz'	m_2	x + y' + z	M_2
011	x'yz	m_3	x + y' + z'	M_3
100	xy'z'	m_4	x' + y + z	M_4
101	xy'z	m_5	x' + y + z'	M_5
110	xyz'	m_6	x' + y' + z	M_6
111	xyz	m_7	x' + y' + z'	M_7

Example application of minterms & maxterms:

x	y	f1	f2
0	0	0	1
0	1	0	0
1	0	0	0
1	1	1	1

f1 = xy = m_3 (looking at 1's) OR
f1 = (x + y)(x' + y)(x + y') = M_0 M_1 M_2 (looking at 0's)

f2 = x'y' + xy = m_0 + m_3 (looking at 1's) OR
f2 = (x' + y)(x + y') = M_1 M_2 (looking at 0's)