Unit 4: Application of Boolean Algebra
Midterm and Maxterm Expansions

4.1 Conversion of English Sentences to Boolean Equations.

3 steps to design a single output combinational switching circuit

1) Find a switching function that specifies the desired behavior (OR, AND, etc.)

2) Find a simplified algebraic expression

3) Realize the simplified function using available logic elements

Ex 1) Mary watches TV if it is Monday night and she has finished her homework

3 phrases with two variables

F = 1, Mary watches TV	F = 0 otherwise
A = 1, it is Monday night	A = 0 otherwise
B = 1, she has finished her hw	B = 0 otherwise

Because F is true if A and B are both true, we can write

$$F = A \cdot B$$

Ex 2) The alarm will ring iff the alarm switch is turned on and the door is not closed or it is after 6 pm and the window is not closed

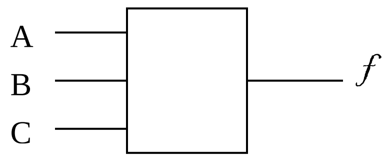
5 phrases

Z = 1, the alarm will ring
A = 1, alarm switch is on
B' = 1, the door is not closed
C = 1, it is after 6 pm
D' = 1, window is not closed

$Z = A \cdot B' + C \cdot D'$

Circuit form

4.2 Combinational Logic Design Using a Truth Table



Where A, B, and C represent 1st, 2nd, and 3rd bits respectively of number N

$f = 1$ for $N \geq 011_2$ and $f = 0$ for $N < 011_2$

A	**B**	**C**	***f***	***f'***
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Derive equation for values of $f = 1$

f = A' B C + A B' C' + A B' C + A B C' + A B C

Simplifying,

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

A' B C + A B' + A B =

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

A' B C + A =

A + BC

Note:
(y + y' x) = x + y

Can also solve by representing $f = 0$

f = (A + B + C) (A + B + C') (A + B' + C)

Can also solve by representing $f' = 1$ then take the complement to get f

f' = A' B' C ' + A' B' C + A' B C '

Take the complement to get f,

$(f')' = (A'B'C')'\ (A'B'C)'\ (A'BC')'$

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

$(x+y)(x+y')=x$

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

$x=A,\ y=B,\ z= B'+ C$

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

$f = A + BC$

4.3 Minterm and Maxterm Expansions

Minterm: term of n variables which is a product of n literals in which each variable appears exactly once in either true or complemented form, but not both.

Row	**A**	**B**	**C**	**Minterms**
0	0	0	0	$A'B'C' = m_0$
1	0	0	1	$A'B'C = m_1$
2	0	1	0	$A'BC' = m_2$
3	0	1	1	$A'BC = m_3$
4	1	0	0	$AB'C' = m_4$
5	1	0	1	$AB'C = m_5$
6	1	1	0	$ABC' = m_6$
7	1	1	1	$ABC = m_7$

Each minterm has a value of 1 for exactly one combination of values of the variables A, B and C.

If $A = B = C = 0$ then $A'B'C' = 1$ and is designated as m_0

$f = A'BC + AB'C + AB'C + ABC' + ABC$ is an example of a function written as a sum of minterms. It is often referred to as minterm expansion or standard sum of products.

The above equation can be rewritten in m-notation,

$f(A, B, C) = m_3 + m_4 + m_5 + m_6 + m_7$

$f(A, B, C) = \Sigma\, m\, (3,4,5,6,7)$

<u>Maxterm</u>: term of n variables which is a sum of n literals in which each variable appears exactly once in its true or complemented form, but not both.

Row	**A**	**B**	**C**	**Maxterms**
0	0	0	0	$A+B+C = M_0$
1	0	0	1	$A+B+C' = M_1$
2	0	1	0	$A+B'+C = M_2$
3	0	1	1	$A+B'+C' = M_3$
4	1	0	0	$A'+B+C = M_4$
5	1	0	1	$A'+B+C' = M_5$
6	1	1	0	$A'+B'+C = M_6$
7	1	1	1	$A'+B'+C' = M_7$

Each maxterm has a value of zero for exactly one combination of values of A, B, and C. Thus $A = B = C = 0$ $A + B + C = 0$ and is designated M_0

Note: The maxterm is the complement of the corresponding minterm

$f = (A + B + C)(A + B + C')(A + B' + C)$ is an example of a function written as a product of maxterms. Referred to as maxterm expansion or standard product of sums

Rewritten in M-notation,
$f(A, B, C) = M_0 M_1 M_2$

Abbreviated
$f(A, B, C) = \Pi M(0, 1, 2)$

In general, switching expressions can be converted to minterm or maxterm expansions either by using a truth table or algebraically.

Note: Another way to obtain the minterm expansion is to first write the expression as a sum of products and introduce missing variables in each term by applying $x + x' = 1$

Ex) Find minterm expansion of $f = (a, b, c, d) = a'(b' + d) + acd'$

$= a'b' + a'd + acd'$

$= a'b'(c + c')(d + d') + a'd(b + b')(c + c') + acd'(b + b')$

$= a'b'cd + a'b'cd' + a'b'c'd + a'b'c'd' + a'bcd + a'bc'd +$ ~~a'b'cd~~ $+$ ~~a'b'c'd~~ $+ abcd' + ab'cd'$

= a'b'c'd' + a'b'c'd + a'b'cd' + a'b'cd + a'bc'd + a'b'c'd + abcd' + ab'cd'

Decimal notation,

a'b'c'd'	a'b'c'd	a'b'cd'	a'b'cd	a'bc'd	a'b'c'd	abcd'	ab'cd'
0000	0001	0010	0011	0101	0111	1110	1010

$$f\ (a, b, c, d) =\ \Sigma\ m\ (0, 1, 2, 3, 5, 7, 10, 14)$$

Maxterms: ones not listed in minterm for n = 4

$$f(a, b, c, d) =\ \Pi M\ (4, 6, 8, 9, 11, 12, 13, 15)$$

Alternate method for maxterm: Factor f to obtain product of sums and introduce missing variables using xx' = 0 then factor again for maxterm

f = a' (b' + d) + acd'
x' z x y

= [a + (b' + d)] (a' +cd')
x yz

= (a + b' +d) (a' + c) (a' + d')

= (a+ b' + cc' + d) (a' + bb' + c + dd') (a' + bb' + cc' + d')

= (a+ b' + c + d) (a+ b' + c' + d) (a' + bb' + c + d) (a' + bb' + c + d')
~~(a' + bb' + c + d')~~ (a' + bb' + c' + d')

= (a+ b' + c + d) (a+ b' + c' + d) (a' + b + c + d) (a' + b' + c + d) (a' + b + c + d')
(a' + b' + c + d') (a' + b + c' + d') (a' + b' + c' + d')

= (a+ b' + c + d) (a+ b' + c' + d) (a' + b + c + d) (a' + b' + c + d) (a' + b + c + d')
0100 0110 1000 1100 1001
(a' + b' + c + d') (a' + b + c' + d') (a' + b' + c' + d')
1101 1011 1111
f = ΠM (4, 6, 8, 9, 11, 12, 13, 15)

Note: in maxterm translation to decimal, primed values equate to a 1 and nonprimed to a zero

An equation can be proven valid by factoring the minterm expansions of each side and showing the expansions are the same

Ex) a'c + b'c' + ab = a'b' + bc + ac'

Left side,

= a'c (b + b') + b'c' (a+ a') + ab (c + c')

= a'bc + a'b'c + ab'c' + a'b'c' + abc + abc'

011 001 100 000 111 110

$= m_3 + m_1 + m_4 + m_0 + m_7 + m_6$

Right side,

= a'b' (c + c') + bc(a + a') + ac' (b + b')

= a'b'c + a'b'c' + abc + a'bc + abc' + ab'c'

001 000 111 011 110 100

$= m_1 + m_0 + m_7 + m_3 + m_6 + m_4$

4.4 General Minterm and Maxterm Expansions

A	B	C	F
0	0	0	a_0
0	0	1	a_1
0	1	0	a_2
0	1	1	a_3
1	0	0	a_4
1	0	1	a_5
1	1	0	a_6
1	1	1	a_7

For a function n, there are 2^n rows of a truth table and $2^\wedge 2^n$ possible functions of n variables.

Minterm expansion for a general function (3 variables)

$$F = a_0m_0 + a_1m_1 + a_2m_2 + \dots a_7m_7 = \sum_{i=0}^{7} (a_im_i)$$

Maxterm expansion for a general function

$$F = (a_0 + m_0)(a_1 + m_1)(a_2 + m_2) \dots (a_z + m_z) = \prod_{i=0}^{7} (a_i + m_i)$$

Generalized

$$F = \sum_{i=0}^{2n-1} a_i m_i = \prod_{i=0}^{2n-1} (a_i + m_i)$$

$$F' = \sum_{i=0}^{2n-1} a_i' m_i = \prod_{i=0}^{2n-1} (a_i' + m_i)$$

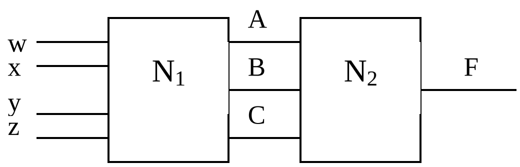
Given two different minterms of n variable $m_i + m_j$ at least one variable appears complemented in one minterm and in uncomplemented in the other.

Therefore if $i \neq j \quad m_i m_j = 0$

Ex) $f_1 = \Sigma m(0, 2, 3, 5, 9, 11)$ $\quad f_2 = \Sigma m(0, 3, 9, 11, 13, 14)$

$f_1 f_2 = \Sigma m(0, 3, 9, 11)$

4.5 Incompletely Specified Functions



Assume that the output of N1 has no combination of values for w, x, y, and z to cause A, B, and C to have a value of 001 and 110. Then when designing N_2 it is not necessary to specify values of F for ABD = 001 and 110.

A	**B**	**C**	**F**
0	0	0	1
0	0	1	X
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	X
1	1	1	1

The X’s indicate “don’t cares” because the values could be 0 or 1 since they will never occur anyway. A function in this format is said to be incompletely specified.

To realize a function, a value must be specified for the “don’t cares”. It is best to choose a value that helps to simplify the function.

Example for both X = 0;

$$F = A'B'C' + A'BC + ABC$$

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

$$= A'B'C' + BC$$

Example for first X = 1 and second X = 0;

$$F = A'B'C' + A'B'C + A'BC + ABC$$

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

$$= A'B' + BC$$

Example for both X = 1;

$$F = A'B'C' + A'B'C + A'BC + ABC' + ABC$$

$$= A'B'(C' + C) + BC (A' + A) + ABC'$$

$$= A'B' + B(C + AC') \qquad (Y + XY' = X + Y)$$

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

(The second example yields the simplest solution.)

Minterm expansion for incompletely specified functions; where m = minterm & d = don’t cares

$$F = \Sigma\, m\, (0, 3, 7) + \Sigma\, d\, (1, 6)$$

Maxterm expansion for incompletely specified functions; where M = Maxterm & D = don’t cares

$$F = \Pi\, M\, (2, 4, 5) \cdot \Pi\, D\, (1, 6)$$

4.6 Examples of Truth Table Construction

Example 1) Design a binary adder using two 1-bit binary numbers

a	**b**	**Sum**		
0	0	0	0	(0 + 0 = 0)
0	1	0	1	(0 + 1 = 1)
1	0	0	1	(1 + 0 = 1)
1	1	1	0	(1 + 1 = 2)

Construct a truth table with logic variables A and B and 2-bit sum logic variables x and y

A	**B**	**X**	**Y**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Because numeric zero is represent by logic zero and numeric 1 by logic 1,

$X = AB$ and $Y = A'B + AB' = A \oplus B$

Example 2) Adder of two 2-bit binary numbers to form a 3-bit sum

N_1		N_2		N_3		
A	B	C	D	X	Y	Z
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Output function:

X (A, B, C, D) = Σ m (7, 10, 11, 13, 14, 15)

Y (A, B, C, D) = Σ m (2, 3, 5, 6, 8, 9, 12, 15)

Z (A, B, C, D) = Σ m (1, 3, 4, 6, 9, 11, 12, 14)

Example 3) Design an error detector for 6-3-1-1 BCD digit. F = 1 iff if the four digits (A, B, C, D) are an invalid code. (page 21)

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

Output

F(A,B,C,D) = Σ m (2,6,10,13,14,15)

= A'B'CD' + A'BCD' + AB'CD' + ABCD' + ABC'D + ABCD

= A'CD'(B + B') + ACD'(B + B') + ABD(C +C')

= A'CD' + ACD' + ABD

= CD' + ABD

Circuit Equation:

Example 4) Design a circuit for an 8-4-2-1 BCD digit (A, B, C, D) where Z = 1 if the decimal is exactly divisible by 3.

0, 3, 6 and 9 are divisible by 3 while 10 – 15 are not valid. (See page 21.)

A	B	C	D	Z
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Output:

$F(A,B,C,D) = \Sigma\, m\,(0,3,6,9) + \Sigma\, d\,(10,11,12,13,14,15)$

To simplify, the easiest solution is a Karnaugh Map which is covered in Unit 5

4.7 Design of Binary Adders and Subtractors

Adders:

Design a parallel adder that adds two 4-bit unsigned binary numbers and a carry input to give a 4-bit sum and carry output.

Option 1: Use a truth table to design the adder … this is very difficult

Option 2: Design a module for a 2-bit adder with a carry (Full Adder).

Then connect four together to form a 4-bit adder.

Truth Table

X	**Y**	**C_{in}**	**C_{out}**	**Sum**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Logic Equation:

$$Sum = X'Y'C_{in} + X'YC_{in}' + XYC_{in}' + XYC_{in}$$

$$= X'(Y'C_{in} + YC_{in}') + X(Y'C_{in}' + YC_{in})$$

$$= X'(Y \oplus C_{in})\ X(Y \oplus C_{in})'$$

$$= X \oplus (Y \oplus C_{in})$$

$$= X \oplus Y \oplus C_{in}$$

$$C_{out} = X'YC_{in} + XY'C_{in} + XYC_{in}' + XYC_{in}$$

$$= (X'YC_{in} + XYC_{in}) + (XY'C_{in} + XYC_{in}) + (XYC_{in}' + XYC_{in})$$

$$= YC_{in}(X' + X) + XC_{in}(Y' + Y) + XY(C_{in}' + C_{in})$$

$$= YC_{in} + XC_{in} + XY$$

Circuit Equation:

Adders for signed numbers: negatives expressed in complement form.

2's complement – the last carry is discarded

1's complement – the last carry is end around carry

Overflow – when two positive numbers yield a negative number
– when two negative numbers yield a positive number

Subtractors: Accomplished by adding the complement of the number to be subtracted. Use either 1's or 2's complement.

Another method is to employ a full subtractor

Where x_i, y_i and b_i are inputs; b_{i+1} and d_i are outputs (b = borrow, d = difference)

Truth Table

x_i	y_i	b_i	b_{i+1}	d_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1