Lab Experiment # 07

The Story of Minterms and Maxterms

Objectives

Learn how implement logic functions using the standard forms: Sum of Products and Product of Sums.

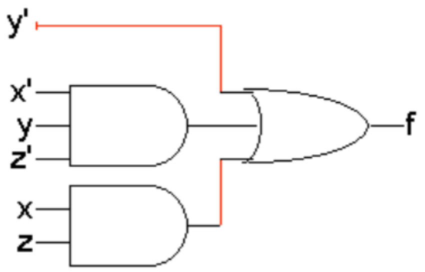
Background

We can write expressions in many ways, but some ways are more useful than others
A sum of products (SOP) expression contains: Only OR (sum) operations at the "outermost" level and each term that is summed must be a product of literals
The advantage is that any sum of products expression can be implemented using a three-level circuit

- literals and their complements at the first level
- AND gates at the second level
- a single OR gate at the third level

Example:

f(x,y,z) = y' + x'yz' + xz



Notice that the NOT gates are implicit and that literals are reused.

A minterm is a special product of literals, in which each input variable appears exactly once.
A function with n variables has 2^n minterms (since each variable can appear complemented or not)

Example:

A three-variable function, such as f(x,y,z), has $2^3 = 8$ minterms:
Each minterm is true for exactly one combination of inputs:
Those minterms are: x'y'z' x'y'z x'yz' x'yz xy'z' xy'z xyz' xyz
A Minterm is true when:

Minterm	When the minterm is True			Minterm ID
x'y'z'	x=0,	y=0,	z=0	m0
x'y'z	x=0,	y=0,	z=1	m1
x'yz'	x=0,	y=1,	z=0	m2
x'yz	x=0,	y=1,	z=1	m3
xy'z'	x=1,	y=0,	z=0	m4
xy'z	x=1,	y=0,	z=1	m5
xyz'	x=1,	y=1,	z=0	m6
xyz	x=1,	y=1,	z=1	m7

Sum of Minterms (or Sum of Products)

Every function can be written as a sum of minterms, which is a special kind of sum of products form
The sum of minterms form for any function is unique

If you have a truth table for a function, you can write a sum of minterms expression just by picking out the rows of the table where the function output is 1.

Example

f = x'y'z' + x'y'z + x'yz' + x'yz + xyz'
 = m0 + m1 + m2 + m3 + m6
 = Σm(0,1,2,3,6)

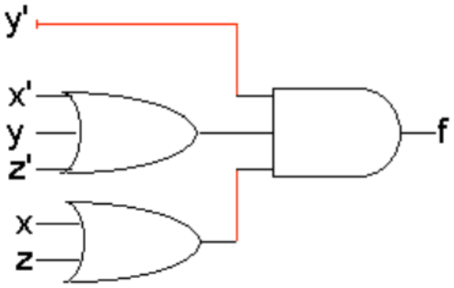
The dual idea: products of sums

A product of sums (POS) expression contains: Only AND (product) operations at the "outermost" level, Each term must be a sum of literals.
Product of sums expressions can be implemented with three-level circuits
- literals and their complements at the first level
- OR gates at the first level
- a single AND gate at the second level

• Compare this with sums of products

Example

f(x, y, z) = y' . (x'+y+z') . (x+z)



A maxterm is a sum of literals, in which each input variable appears exactly once.
A function with n variables has 2^n maxterms

Example

A three-variable function f(x,y,z) has 8 maxterms
Each maxterm is false for exactly one combination of inputs:
Those materms are: x'+y'+z' x'+y'+z x'+ y+z' x'+ y+z x+y'+z' x+y'+z x+y+z' x+y+z
Maxterm Is false when:

Maxterm	When the maxterm is false	Maxterm ID
x + y + z	x=0, y=0, z=0	M0
x + y + z'	x=0, y=0, z=1	M1
x + y' + z	x=0, y=1, z=0	M2
x + y' + z'	x=0, y=1, z=1	M3
x' + y + z	x=1, y=0, z=0	M4
x' + y + z'	x=1, y=0, z=1	M5
x' + y' + z	x=1, y=1, z=0	M6
x' + y' + z'	x=1, y=1, z=1	M7

Every function can be written as a unique product of maxterms
If you have a truth table for a function, you can write a product of maxterms expression by picking out the

rows of the table where the function output is 0. (Be careful if you're writing the actual literals!)
f = (x' + y + z).(x' + y + z').(x' + y' + z')
= M4. M5.M7 = ΠM(4,5,7)

f' = (x + y + z).(x + y + z').(x + y' + z).(x + y' + z').(x' + y' + z)
= M0. M1. M2. M3. M6 = ΠM(0,1,2,3,6)

Minterms and maxterms are related

Any minterm m_i is the complement of the corresponding maxterm M_i
For example, m4' = M4 because (xy'z')' = x' + y + z

Minterm	Shorthand	Maxterm	Shorthand
x'y'z'	m0	x + y + z	M0
x'y'z	m1	x + y + z'	M1
x'yz'	m2	x + y' + z	M2
x'yz	m3	x + y' + z'	M3
xy'z'	m4	x' + y + z	M4
xy'z	m5	x' + y + z'	M5
xyz'	m6	x' + y' + z	M6
xyz	m7	x' + y' + z'	M7

Converting between standard forms

We can convert a sum of minterms to a product of maxterms
• In general, just replace the minterms with maxterms, using maxterm numbers that don't appear in the sum of minterms:
• The same thing works for converting from a product of maxterms to a sum of minterms

Example

From before
f = Σm(0,1,2,3,6)
and f' = Σm(4,5,7)
= m4 + m5 + m7
complementing (f')' = (m4 + m5 + m7)'
so f = m4' . m5' . m7' [DeMorgan's law]
= M4 . M5 . M7
= ΠM(4,5,7)

Lab Tasks

Task 1: Three-input Boolean functions

Given the following truth table of a three-input logic circuit

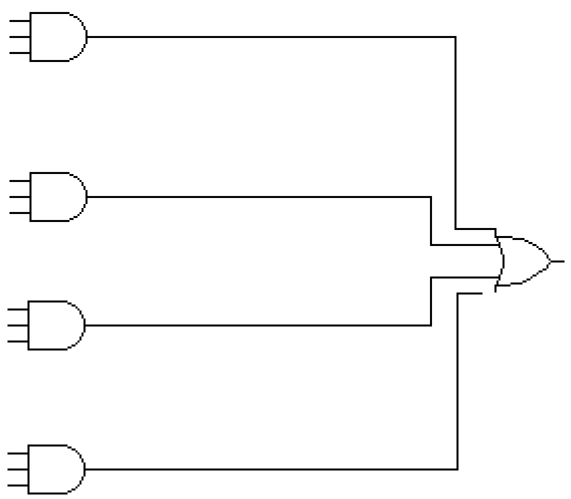
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Write the above function in the two standard forms

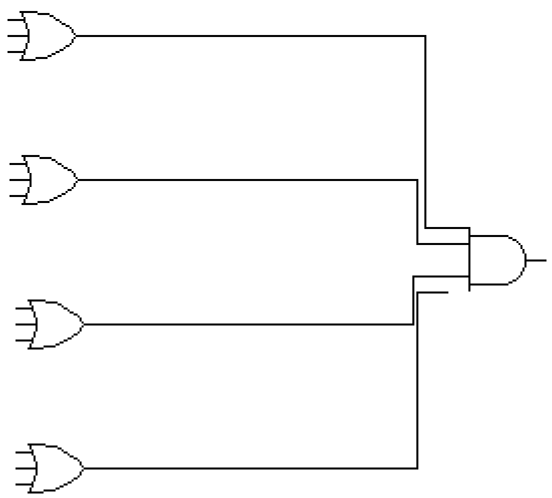
F(A, B, C)= Σ ( )

F(A, B, C) = Π( )

Draw a circuit that implements the above logic function (use minterms only)



Draw a circuit that implements the above logic function (use maxterms only)



Task 2: Three-input Boolean functions

Simplify (using k-maps) the function presented in Task 1 of this lab. Draw the simplified form of the function on EWB. Use the Logic Converter of EWB to generate the truth table of the simplified circuit.

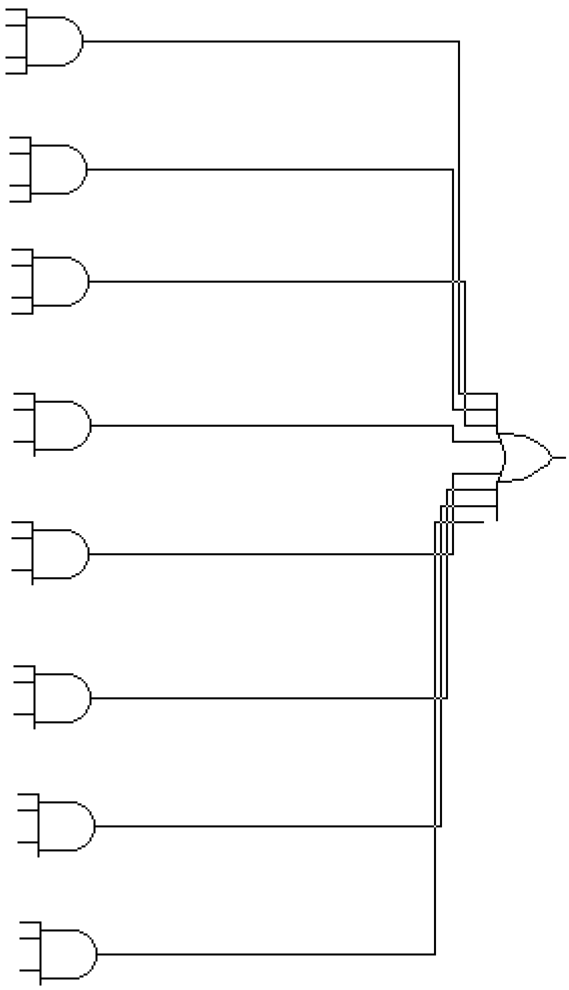
A	B	C	F (simplified)
0	0	0	
0	0	1	

0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Task 3: Four-input Boolean functions

Draw the following logic function using EWB

F(A, B, C, D) = Σ(6, 8, 9, 10, 11, 12, 13, 14)



Task 4: Four-input Boolean functions

Simplify (using k-maps) the function presented in Task 3 of this lab. Draw the simplified form of the function on EWB. Use the Logic Converter of EWB to generate the truth table of the simplified circuit.

	A	B	C	D	F
0	0	0	0	0	

1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

Task 5: Simplifying 4-variable functions

Simplify and implement (using EWB) the following function

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

Draw you circuit below

Task 6: Simplifying 4-variable functions: SOP

Draw a NAND logic diagram that implements the **complement** of the following function

$F(A, B, C, D) = \Sigma(0, 1, 2, 3, 4, 8, 9, 12)$

Draw you circuit below

Task 7: Simplifying 4-variable functions: POS

Draw a logic diagram that implements the following function
F(A, B, C, D) = $\Pi(0, 1, 2, 3, 4, 8, 9, 12)$
Draw you circuit below