Switching Functions (E)

Before we start **reviewing** how to simplify **Boolean** expressions, let's take a look at special way to write a **Boolean** expression known as the **Switching Function**. Note the two expressions below:

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

Both of the functions above are **Switching Functions**.

Special note of the order of the variables in the variable list (x, y, z) should be taken. **If the order changes, so does the answer, even when the equation doesn't change.**

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

is not equal to

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

For example, they would each produce different output columns in a truth table.

x	y	z	xy	$\bar{y}z$	$f(x,y,z)$	$f(y,z,x)$	$\bar{y}z$	xy	y	z	x
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0	1
0	1	0	0	0	0	1	1	0	0	1	0
0	1	1	0	0	0	1	1	0	0	1	1
1	0	0	0	0	0	0	0	0	1	0	0
1	0	1	0	1	1	1	0	1	1	0	1
1	1	0	1	0	1	0	0	0	1	1	0
1	1	1	1	0	1	1	0	1	1	1	1

same equation ====== ⥣ ======= ⥣ === but different results

The order of the switching list sets the order of the variables in the table. The rule is that the left most variable is the **MOST SIGNIFICANT** variable while the right most variable is the **LEAST SIGNIFICANT** variable. **This is very important! We'll see more on this topic later**.

Let's look at a simple two variable **switching function**. **f(x, y) = x y**

We can create a very familiar truth table (**AND**) for this switching function as shown:

x	y	f(x,y)	
0	0	0	m_0
0	1	0	m_1
1	0	0	m_2
1	1	1	m_3

From this table, we can now rewrite the expression as: $f(x,y) = \sum m(3)$

This form of expression is known as the "Numerical Canonical" form.

The min-term, m_3, represents the input combination (**x=y=1**) which will cause a **logic 1** on the output. The expression can be written in its algebraic form as: **f(x, y) = x y**

Let's look at another switching expression. Note that the expression itself is already in its **"Reduced"** form (can't be made any simpler).

$$f(x,\ y,\ z) = y \ (\text{reduced form})$$

Note that when in this reduced form, 'product' terms might, or might not have a representative of each variable in the variable list.

The table to the right results from this expression.

x	y	z	f(x,y,z)	
0	0	0	0	m_0
0	0	1	0	m_1
0	1	0	1	m_2
0	1	1	1	m_3
1	0	0	0	m_4
1	0	1	0	m_5
1	1	0	1	m_6
1	1	1	1	m_7

Each row in a truth table is called a **min-term**. We can list the **min-terms** that are associated with **a logic 1** in the output column. When we do this, we can now express this expression in its **Numerical Canonical form**:

$$f(x,\ y,\ z) = m_2 + m_3 + m_6 + m_7$$
$$= \sum m(2,\ 3,\ 6,\ 7)$$

So, we now have **three ways** to represent **switching functions:**

1. **Algebraic Form (Reduced) (Not Canonical)**

$$f(a, b, c) = a\,b\,\bar{c} + b\,c + a\,c$$

2. **Algebraic Canonical Form (fully expanded)(each term has each of the switching list variables in it)**

$$\begin{aligned} f(a, b, c) &= a\,b\,\bar{c} + (a + \bar{a})\,b\,c + a\,(b + \bar{b})\,c \\ &= \underbrace{a\,b\,\bar{c}}_{m_6} + \underbrace{a\,b\,c}_{m_7} + \underbrace{\bar{a}\,b\,c}_{m_3} + \underbrace{\cancel{\underbrace{a\,b\,c}_{m_7}}}_{\text{Duplicate}} + \underbrace{a\,\bar{b}\,c}_{m_5} \\ &= \underbrace{a\,b\,\bar{c}}_{m_6} + \underbrace{a\,b\,c}_{m_7} + \underbrace{\bar{a}\,b\,c}_{m_3} + \underbrace{a\,\bar{b}\,c}_{m_5} \end{aligned}$$

3. **Numerical Canonical Form**

$$\begin{aligned} f(a,b,c) &= \sum m(3,5,6,7) \\ &= \prod M(0,1,2,4) \\ &= m_3 + m_5 + m_6 + m_7 \end{aligned}$$

Algebraic and Numeric Canonical Example 1:

Problem Statement: Expand the following Boolean Expression to its **Algebraic Canonical Form**

$$f(g,\ h,\ i) = h + \overline{g\,\overline{i}} = h + \overline{g} + \overline{\overline{i}}$$

$$= h + \overline{g} + i$$

$$= \left(g + \overline{g}\right)h\left(i + \overline{i}\right) + \overline{g}\left(h + \overline{h}\right)\left(i + \overline{i}\right) + \left(g + \overline{g}\right)\left(h + \overline{h}\right)i$$

$$= \left(gh + \overline{g}h\right)\left(i + \overline{i}\right) + \left(\overline{g}h + \overline{g}\,\overline{h}\right)\left(i + \overline{i}\right)$$

$$+ \left(gh + g\overline{h} + \overline{g}h + \overline{g}\,\overline{h}\right)i$$

$$= \underbrace{\underbrace{g}_{1}\ \underbrace{h}_{1}\ \underbrace{i}_{1}}_{7} + \underbrace{\underbrace{g}_{1}\ \underbrace{h}_{1}\ \underbrace{\overline{i}}_{0}}_{6} + \underbrace{\underbrace{\overline{g}}_{0}\ \underbrace{h}_{1}\ \underbrace{i}_{1}}_{3} + \underbrace{\overline{g}\,h\,\overline{i}}_{2}$$

$$+ \cancel{\underbrace{\overline{g}\,h\,i}_{3}} + \cancel{\underbrace{\overline{g}\,h\,\overline{i}}_{2}} + \underbrace{\overline{g}\,\overline{h}\,i}_{1} + \underbrace{\overline{g}\,\overline{h}\,\overline{i}}_{0}$$

$$+ \cancel{\underbrace{g\,h\,i}_{7}} + \underbrace{g\,\overline{h}\,i}_{5} + \cancel{\underbrace{\overline{g}\,h\,i}_{3}} + \cancel{\underbrace{\overline{g}\,\overline{h}\,i}_{1}}$$

$$= g\,h\,i + g\,h\,\overline{i} + \overline{g}\,h\,i + \overline{g}\,h\,\overline{i} + \overline{g}\,\overline{h}\,i + \overline{g}\,\overline{h}\,\overline{i} + g\,\overline{h}\,i$$

We can now go directly to the Numerical Canonical form:

$$f(g,\ h,\ i) = \sum m(0,1,2,3,5,6,7)$$

or

$$f(g,\ h,\ i) = \sum m(0\text{-}3, 5\text{-}7)$$

The **min-term** list creates the **Sum-of-Products (SOP)** form. Let's next examine **the max-term list** and the **Product of Sums (POS)** form.

Algebraic and Numeric Canonical Example 2:

$$f(a,\ b,\ c,\ d) = a\,\overline{d}$$
$$= a\left(b + \overline{b}\right)\left(c + \overline{c}\right)\overline{d}$$
$$= \left(a\,b + a\,\overline{b}\right)\left(c + \overline{c}\right)\overline{d}$$
$$= \left(a\,b\,c + a\,b\,\overline{c} + a\,\overline{b}\,c + a\,\overline{b}\,\overline{c}\right)\overline{d}$$
$$f(a,\ b,\ c,\ d) = a\,b\,c\,\overline{d} + a\,b\,\overline{c}\,\overline{d} + a\,\overline{b}\,c\,\overline{d} + a\,\overline{b}\,\overline{c}\,\overline{d}$$

We can now convert it to its **Numerical Canonical Form**

$$f(a,\ b,\ c,\ d) = \underbrace{a\,b\,c\,\overline{d}}_{m_{14}} + \underbrace{a\,b\,\overline{c}\,\overline{d}}_{m_{12}} + \underbrace{a\,\overline{b}\,c\,\overline{d}}_{m_{10}} + \underbrace{a\,\overline{b}\,\overline{c}\,\overline{d}}_{m_{8}}$$
$$= \sum m(8,\ 10,\ 12,\ 14)$$
$$= \prod M(0-7,\ 9,\ 11,\ 13,\ 15)$$

Max-terms

Let's take a second look at the Example 1's result:

$$f(g,h,i) = \sum m(0,1,2,3,5,6,7)$$

g	h	i	F	m
0	0	0	1	0
0	0	1	1	1
0	1	0	1	2
0	1	1	1	3
1	0	0	0	4
1	0	1	1	5
1	1	0	1	6
1	1	1	1	7

The one term with an output of "0" is the **max-term**. If we had treated the row with a "0" in it as a **min-term (which it isn't)**, we would have written it as:

$$f(g,h,i) = g\,\overline{h}\,\overline{i}$$

This **is not** the correct way to represent the **max-term** algebraically.

In order to think about **max-terms**, it's helpful to review **De'Morgan's Theorem** again. **De'Morgan's** says that you can invert a function by inverting all the variables and changing all signs.

Let's take a **look at the** '0' in the table on the previous page. We want to **represent this as a max-term** which represents the '0' in the function instead of the '1' that the **min-term** represents. We take the **min-term** expression and **negate it**. We do this because '0' **is the INVERSE of** '1', so if we want to represent the **MAX-TERM** expression we would take the **min-term's** inverse and then we would have:

$$f(g,h,i) = \prod M(4) = \overline{\left(g\,\overline{h}\,\overline{i}\right)} = \overline{g} + h + i$$

To find the **max-terms** we invert the **min-terms** with **De'Morgan's**. The box below demonstrates the **min-terms** and max-terms for a **3-bit switching list**.

a	b	c	min-term	Desig		max-term	Desig
0	0	0	$\bar{a}\,\bar{b}\,\bar{c}$	m_0	$\overline{m_0}$	$a + b + c$	M_0
0	0	1	$\bar{a}\,\bar{b}\,c$	m_1	$\overline{m_1}$	$a + b + \bar{c}$	M_1
0	1	0	$\bar{a}\,b\,\bar{c}$	m_2	$\overline{m_2}$	$a + \bar{b} + c$	M_2
0	1	1	$\bar{a}\,b\,c$	m_3	$\overline{m_3}$	$a + \bar{b} + \bar{c}$	M_3
1	0	0	$a\,\bar{b}\,\bar{c}$	m_4	$\overline{m_4}$	$\bar{a} + b + c$	M_4
1	0	1	$a\,\bar{b}\,c$	m_5	$\overline{m_5}$	$\bar{a} + b + \bar{c}$	M_5
1	1	0	$a\,b\,\bar{c}$	m_6	$\overline{m_6}$	$\bar{a} + \bar{b} + c$	M_6
1	1	1	$a\,b\,c$	m_7	$\overline{m_7}$	$\bar{a} + \bar{b} + \bar{c}$	M_7

Algebraic and Numeric Canonical Example 3:

$$f(a,b,c) = ab + c$$

$$= ab\left(c + \bar{c}\right) + \left(a + \bar{a}\right)\left(b + \bar{b}\right)c$$

$$= abc + ab\bar{c} + \left(ab + a\bar{b} + \bar{a}b + \bar{a}\,\bar{b}\right)c$$

$$= \underbrace{abc}_{7} + \underbrace{ab\bar{c}}_{6} + \underbrace{abc}_{7} + \underbrace{a\bar{b}c}_{5} + \underbrace{\bar{a}bc}_{3} + \underbrace{\bar{a}\bar{b}c}_{1}$$

$$f(a,b,c) = \sum m(1,3,5,6,7)$$

$$f(a,b,c) = \prod M(0,2,4)$$

or you could even write

$$f(a,b,c) = \sum m(1,3,5,6,7) + \prod M(0,2,4)$$

The Inverse (Complement) of Min-term and Max-term lists

Suppose you had the following **min-term** list $f\left(a,b,c\right)=\sum m\left(0,1,5,7\right)$ and you wanted to find its **complement**. All that is necessary is to provide a **min-term** list with all the **min-terms** which were missing from the original list.

$$f(a,b,c)=\sum m\left(0,1,5,7\right)$$
$$\overline{f}(a,b,c)=\overline{\sum m\left(0,1,5,7\right)}$$
$$\overline{f}(a,b,c)=\sum m\left(2,3,4,6\right)$$

The same can be said for a **max-term** list as is shown below:

$$f(a,b,c)=\prod M\left(2,3,4,6\right)$$
$$\overline{f}(a,b,c)=\overline{\prod M\left(2,3,4,6\right)}$$
$$\overline{f}(a,b,c)=\prod M\left(0,1,5,7\right)$$

Note that in both cases, all we changed was the list contents. The **min-term** list stayed a **min-term** list and the **max-term** list stayed a **max-term** list.

Question: **So when do you use the min-term expression and when do you use the max-term expression?**

Answer: **One way to look at it is that the max-term expression is used when you are looking for an Active Low output. Otherwise, use the Min-term expression.**

Incompletely Specified Functions (Don't Cares)

Up until now, we have only discussed functions which had to be able to respond to any possible combination of the input variables. These functions are known as "**completely specified**." However, quite often a function may contain some min-terms but not others. These other terms are called **"don't care"** terms and the function which contains them is an **"incompletely specified function**."

Incompletely Specified functions occur naturally in two different ways.

- The "**don't care**" conditions will never be allowed to occur, so the network can use these don't cares in any way necessary to simplify itself **(BCD for example).**
- Some don't care **conditions may arise** but the output is required to be a '1' or a '0' **only for certain conditions**.

We label these types of **terms** as d_i vice m_i.

The numerical canonical form representing this table is:

$$f(a,b,c) = \sum m(0,2,4,7) + \sum d(3,6)$$

a	b	c	f (a,b,c)	
0	0	0	1	0
0	0	1	0	
0	1	0	1	2
0	1	1	d	don't care
1	0	0	1	4
1	0	1	0	
1	1	0	d	don't care
1	1	1	1	7