Chapter 12

Advanced Boolean Algebra

Advanced Boolean Algebra

While much of the logic needed for a vehicle can be programmed using just the three basic functions of Boolean Algebra, you may find need for more complex functions. This chapter deals with more advanced functions that are "built" from the three basic functions. It also introduces one of the more important theorems, De Morgan's.

DE MORGAN'S THEOREMS

De Morgan was a great logician and mathematician, as well as a friend of Boole's. Among De Morgan's important contributions to logic are these two theorems:

A + B = !(!A * !B)
A * B = !(!A + !B)

In words, the first equation says that *the statement A OR B is the same as the statement NOT (NOT A AND NOT B)* This set of equations will allow you to use PMC to create statements having many ANDs and ORs.

An AND function can be restated using OR, by "NOT"ing the inputs and outputs. An OR function can be restated using AND, by "NOT"ing the inputs and outputs. In simple vehicles, you may never require this function. For example, a three (or greater) input OR function can be developed by using several Virtual Channels (extra memory). However, in more complicated systems, utilizing DeMorgan's Theorem, the extra memory can be saved and the entire equation can be written on a single Boolean Editor Screen.

***Q* = A * B * C = !(!A + !B +!C)**

THE EXCLUSIVE-OR FUNCTION

Another example of combining these simple functions is the **Exclusive-OR.** The output of the Exclusive-OR is <u>true</u> whenever the two inputs are different.

The truth table for this function is as follows:

Inputs		Output
A	B	*Q*
0	0	0
0	1	1
1	0	1
1	1	0

The reason for the name Exclusive-OR is this that a 1 output occurs when A or B is 1, <u>*but not both*</u>. Stated another way, the Exclusive-OR function has a 1 output only when both inputs are different; the output is 0 when the inputs are the same.

The Exclusive-OR function can be created from the simpler AND *and* OR functions as follows:

***Q* = (A * !B) + (!A * B)**

Let's go through this equation to examine how this algebra works.

The first case is when both inputs are 0. The first term, A!B is A AND'ed with !B, or 0 AND 1. The result is 0. The second term, !A B is !A AND'ed with B, or 1 AND 0. The result is 0. OR'ing these two zeros gives an output of 0.

The next case is when A is 0 and B is 1. The first term is A!B, or 0 AND 0. The result is 0. The second term is !AB, or 1 AND 1. The result is 1. OR'ing the 0 and the 1 gives an output of 1.

The third case is similar to the second.

The fourth case is when A and B are equal to 1. The first term, A!B is A AND'ed with !B, or 1 AND 0. The result is 0. The second term, !AB is !A AND'ed with B, or 0 AND 1. The result is 0. OR'ing these two zeros gives an output of 0.

Advanced Boolean Algebra

The Exclusive-OR function gives us a new kind of function to work with. We will use the symbol ⊕ to stand for this function.

When we want to describe an Exclusive-OR function, we can write:

Q = A⊕B

An example of the use of this function is making a three way switch for a light. If we use two switches as the inputs for the Exclusive-OR function, they will make the light go ON, or OFF from either switch.

Let's take a look at how this works.

If we let: A = Switch 1
B = Switch 2
Q = Light

Then the exclusive OR Boolean expression will be:

Q = A!B + !AB

If both switches are OFF, or 0, the expression is 0, therefore, the light is off. If either switch is turned ON (they don't match), the expression will be 1, the light will be ON. If both switches are turned ON (they match), the expression will be 0 and the light will be OFF.

An additional switch can be added to the control group by using the output from this expression to be an input to another Exclusive-OR expression (rather than being the light), with it's output being the light.

The two expressions would be:

C = Switch 3
D = Temporary Memory Location (Virtual Module)

D = A * !B + !A * B
Q = C * !D + !C * D

With this expression, the light will be controlled by any of the three switches. Additional switches can be added to the control by writing more of these expressions.

See Chapter 9 for PMC program examples, or call Intellitec. We will be happy to asist you.

Appendix A

Design Worksheets

I/O List

Channel	Input/Output Name	I/O	Input H/L Output Amps	Location/Zone
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				
45				
46				
47				
48				
49				
50				
51				
52				

Channel	Input/Output Name	I/O	Input H/L Output Amps	Location/Zone
53				
54				
55				
56				
57				
58				
59				
60				
61				
62				
63				
64				
65				
66				
67				
68				
69				
70				
71				
72				
73				
74				
75				
76				
77				
78				
79				
80				
81				
82				
83				
84				
85				
86				
87				
88				
89				
90				
91				
92				
93				
94				
95				
96				
97				
98				
99				
100				
101				
102				
103				
104				

Channel	Input/Output Name	I/O	Input H/L Output Amps	Location/Zone
105				
106				
107				
108				
109				
110				
111				
112				
113				
114				
115				
116				
117				
118				
119				
120				
121				
122				
123				
124				
125				
126				
127				
128				
129				
130				
131				
132				
133				
134				
135				
136				
137				
138				
139				
140				
141				
142				
143				
144				
145				
146				
147				
148				
149				
150				

Channel	Input/Output Name	I/O	Input H/L Output Amps	Location/Zone
151				
152				
153				
154				
155				
156				
157				
158				
159				
160				

NOTES

Module and Channel Assignment

Vehicle Identification	
Date	
Module Name	
Module Address A-P	
Module Type or Description	
Module Location or Zone	

Channel Number	Channel Name	In/Out	High/Low Amps	Boolean Expression
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

NOTES