Advanced Boolean Logic and Applications to Control Systems

Course No: E02-011
Credit: 2 PDH

Jeffrey Cwalinski, P.E.

CED engineering.com
PDH FOR THE PROFESSIONAL

Continuing Education and Development, Inc.

P: (877) 322-5800
info@cedengineering.com

Advanced Boolean Logic and Applications to Control Systems

I. Number Systems

Prior to explaining the binary number system the decimal number system will be reviewed. In both cases the discussion will be limited to positive integers.

A. Decimal Number System

The decimal number system consists of digits which are assigned values of 0 thru 9. Each digit is weighted by a power of 10 according to its position in the decimal number. The general format of a decimal number is as follows:

$$K_n * 10^n + K_{n-1} * 10^{n-1} + \ldots + K_3 * 10^3 + K_2 * 10^2 + K_1 * 10^1 + K_0 * 10^0 .$$

Where the *K*'s represent digits with values between 0 thru 9 and the index *n* is equal to one less than the number of digits in the decimal number. As an example the number 93716 will be written in the general format. Since there are 5 digits the index *n* will be 4, one less than the number of digits in the decimal number. The general format is:

$$9*10^4 + 3*10^3 + 7*10^2 + 1*10^1 + 6*10^0 .$$

B. Binary Number System

The binary number system consists of digits which are assigned values of 0 and 1. In the binary number system the digits are called bits. Each bit is weighted by a power of 2 according to its position in the binary number. The general format of a binary number is as follows:

$$K_n * 2^n + K_{n-1} * 2^{n-1} + \ldots + K_3 * 2^3 + K_2 * 2^2 + K_1 * 2^1 + K_0 * 2^0 .$$

Where the *K*'s represent bits with values of 0 or 1, and the index *n* is equal to one less than the number of bits in the binary number. A binary number can be easily converted to its decimal equivalent using the general format. As an example the number 11011 will be written in the general format and converted in to its decimal equivalent. Since there are 5 digits the index *n* will be 4, one less than the number of bits in the binary number. The general format and decimal equivalent is:

$$1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 16 + 8 + 0 + 2 + 1 = 27 .$$

At this point the manner in which the positions of each bit are referred to needs to be discussed. While it is obvious that the binary number above has 5 bits, it is not obvious which bit is defined as the "first" bit. That is, is it the bit on the right? Or is it the bit on the left? Also, is the position count 0 based or 1 based? That is, is the "first" bit considered to be bit number 1 or bit number 0? Inconsistencies exist. What is agreed on are the terms Most Significant Bit (MSB) and Least Significant Bit (LSB). The MSB is the bit with the most weighting, highest power of 2, the left most bit. The LSB is the bit with the least weighting, the lowest power of 2, the right most bit. As for the count being 0 based or 1 based, that is a matter of preference.

There are formal methods for converting from decimal to binary. One such method involves repeated division by 2. For the first division the process uses the decimal number to be converted as the dividend, 2 as the divisor, and the remainder, 1 or 0, is the LSB. Subsequent divisions use 2 as the divisor, the previous quotient as the dividend and the remainder, 1 or 0, is the next bit in the binary number. As an example the number 33 will be converted from decimal to its binary equivalent. The first division results in, 33 ÷ 2 = 16, with a remainder of 1, this 1 is the LSB of the binary number. The division is continued with, 16 ÷ 2 = 8, with a remainder of 0. Continuing with the division, 8 ÷ 2 = 4, with a remainder of 0. Again, 4 ÷ 2 = 2, with a remainder of 0. Again, 2 ÷ 2 = 1, with a remainder of 0. The division continues as long as there is a quotient greater than or equal to 1. As such, 1 ÷ 2 = 0, with a remainder of 1 which is the MSB. The remainders are written in order starting with the MSB to the LSB, 100001. For a small number of bits, approximately 5 (this number is arbitrary and depends upon a person's aptitude), trial and error may be quicker.

II. Simplification

Logic equations can be derived from a specification of requirements. An initial attempt to write an equation to satisfy a requirement will most likely result in an equation that can be simplified. The simplification can reduce the number of terms in the equation which is of a benefit if the equation is to be implemented in software. Simplification also reduces the number of logic gates, if the logic equation is implemented in hardware. Methods for simplification are discussed below.

A. Basic Theorems

The following basic theorems can be used to simplify Boolean equations:

1. $A + 0 = A$
2. $A + 1 = 1$

3. $A + A = A$
4. $A + \overline{A} = 1$
5. $A \bullet 1 = A$
6. $A \bullet 0 = 0$
7. $A \bullet A = A$
8. $A \bullet \overline{A} = 0$
9. $A \bullet B = B \bullet A$, and $A + B = B + A$, communicative property
10. $A \bullet (B \bullet C) = (A \bullet B) \bullet C = A \bullet B \bullet C$, and
    $A + (B + C) = (A + B) + C = A + B + C$, associative property
11. $A \bullet (B + C) = (A \bullet B) + (A \bullet C)$, and
    $A + (B \bullet C) = (A + B) \bullet (A + C)$, distributive property
12. DeMorgan's Theorem, $\overline{A \bullet B \bullet C \bullet \cdots} = \overline{A} + \overline{B} + \overline{C} + \cdots$ and
    $\overline{A + B + C + \cdots} = \overline{A} \bullet \overline{B} \bullet \overline{C} \cdots$.

An example using $\mathbf{Y = AD + \left(\overline{B \cdot \bar{C}}\right)A\bar{C}}$ follows. First, DeMorgan's Theorem can be used to simplify the $\left(\overline{B \cdot \bar{C}}\right)$ term, which results in $(\overline{B} + C)$. The equation is now rewritten as $Y = AD + (\bar{B} + C)A\bar{C}$. It can be further simplified using the distributive property, $Y = AD + (A\bar{B}\bar{C} + AC\bar{C})$. From the theorems above the $C\bar{C}$ term always equals "0" and the equation is now $Y = AD + (A\bar{B}\bar{C} + 0)$. Using the theorem $A + 0 = A$ the final equation is rewritten as $Y = AD + A\bar{B}\bar{C}$, which can be implemented with two NOT gates, two AND gates, and one OR gate.

B. Truth Table

A truth table lists every combination of input variables. The Boolean equation is evaluated with these input values and the corresponding output value is recorded. It is useful to use truth tables when a Boolean equation is simplified. By generating a truth table for the original Boolean equation and a truth table for the simplified Boolean equation, both tables can be compared to ensure that the simplified equation produces the same results as the original equation and thereby ensuring correctness. The input variables start with all "0" values and are listed in ascending order to all "1" values. This is identical to considering the input values to be a binary number initially equal to all "0"s and incrementing the LSB until the number is all "1"s. Then the equation is evaluated for each set of input variables.

A truth table will be generated for the following equation:

$$f(A, B, C, D) = Y = B(AC + AD + \overline{A} \bullet \overline{C}) + CD$$

A	*B*	*C*	*D*	*Y*	Decimal Equivalent of *ABCD* (min terms)
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	2
0	0	1	1	1	3
0	1	0	0	1	4
0	1	0	1	1	5
0	1	1	0	0	6
0	1	1	1	1	7
1	0	0	0	0	8
1	0	0	1	0	9
1	0	1	0	0	10
1	0	1	1	1	11
1	1	0	0	0	12
1	1	0	1	1	13
1	1	1	0	1	14
1	1	1	1	1	15

The variables *A*, *B*, *C*, and *D* can be combined to form a 4 bit binary number, *ABCD*, with *A* being the MSB and *D* being the LSB. This number increments as one moves down the table from 0000 to 1111, which is decimal 15. The method for converting a binary number to a decimal number, discussed in Section I. B above, can be used to determine the remaining values of the 4 bit binary number, *ABCD*. This 4 bit binary number is called a min term and its decimal equivalent is written in the rightmost column. The notation for a min term is m_x, where x is the min term number which is the decimal equivalent of *ABCD*. For example, min term 12, which corresponds to *ABCD* = 1100, is written as m_{12}. The min term is a link between the lines in a truth table and boxes in a Karnaugh Map, which is discussed below.

C. Karnaugh Map

The Karnaugh Map is a table that represents a Boolean equation. Karnaugh Maps can be used for any number of variables, but will be limited to three and four variables in this lesson.

A three variable Karnaugh Map is shown below:

C \ *AB*	00	01	11	10
0	0	2	6	4
1	1	3	7	5

The variables *A* and *B* are on the top row, with variable *A* being the MSB, and variable *B* being the LSB. Note that they are in a peculiar order, $AB = 11$, (binary representation of decimal 3) precedes $AB = 10$, (binary representation of decimal 2). The variable *C* is on the left side of the table. The numbers in the upper left corners are the decimal values of the 3 bit binary number *ABC* with *A* being the MSB and *C* being the LSB. These numbers are the min terms. As before, the notation for a min term is m_x, where x is the min term number. For example min term 6 is written as m_6. It is located where *AB* = 11 and *C* = 0 intersect and the number 6 appears in the box in the table. This is the decimal representation of *ABC* = 110, which equals 6. These numbers are the same min terms that are contained in a three variable truth table and provide a link which allows mapping of the output of the equation from the truth table to the Karnaugh Map.

A four variable Karnaugh Map is shown below:

CD \ *AB*	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

As before, the variables *A* and *B* are on the top row, with variable *A* being the MSB, and variable *B* being the LSB. Again, note that they are in a peculiar order, $AB = 11$, (binary representation of decimal 3) precedes $AB = 10$, (binary representation of decimal 2). The variables *C* and *D* are on the left side of the table, with variable *C* being the MSB, and variable *D* being the LSB. Again, note the peculiar order. The numbers in the upper left corners are the decimal values of the 4 bit binary number *ABCD* with *A* being the MSB and *D* being the LSB. Again, these numbers are the min terms and the notation is the same as it was before, m_x, where x is the min term number. For example, min term 14 is written as m_{14}. It is located where, where *AB* = 11 and *CD* = 10 intersect, and the number 14 appears. This is the decimal representation of *ABCD* = 1110. These numbers are the same min terms that are contained in a four variable truth table and provide a link which allows mapping of the output of the equation from the truth table to the Karnaugh Map.

To map the Boolean equation that generated the truth table in section II.B into a Karnaugh Map place a “1” in every box which is labeled with the min term from the truth table where the value of the function, *Y*, is “1”. In this case a “1” will be placed in the boxes labeled 3, 4, 5, 7, 11, 13, 14, and 15. The “0”s can be neglected.

CD \ *AB*	00	01	11	10
00	0	4 1	12	8
01	1	5 1	13 1	9
11	3 1	7 1	15 1	11 1
10	2	6	14 1	10

D. Standard Sum of Products Form of Equations

Both truth tables and Karnaugh Maps can be used to write Boolean equations in what is called the Standard Sum of Products Form. As the name implies, this form is similar to an algebraic sum of the products of variables. In Boolean terms the Standard Sum of Products Form is the OR of the min terms. This form of the equation makes it obvious which combination of variables will cause the equation to evaluate to “1”. At least one min term must evaluate to a “1” for the equation to evaluate to a “1”.

Each individual min term is an AND of variables, as such, for the min term to be a “1” all variables must be a “1”. Using min term 14, which is abbreviated as $m_{14,}$ as an example the 4 bit binary number formed by the variables is *ABCD* = 1110. The AND of these will result in a “0”. The only way the AND of these variables will result in a “1” is if *D* is complemented, that is, $ABC\overline{D} = 1 \cdot 1 \cdot 1 \cdot \overline{0} = 1 \cdot 1 \cdot 1 \cdot 1 = 1$. This min term is written as a combination of variables, $m_{14} = ABC\overline{D}$ and is defined by this unique equation consisting of the four variables that will evaluate to a “1” only for the following combination of variables, *A*=1, *B*=1, *C*=1, and *D*=0. This is true of all min terms. As another example consider min term 7, which is abbreviated as m_7. For this min term *ABCD* = 0111, decimal 7. For this min term to be “1” the *A* must be complemented, $\overline{A}BCD$. This min term has a unique equation, $m_7 = \overline{A}BCD$ and will evaluate to a “1” only for the following combination of variables *A*=0, *B*=1, *C*=1, and *D*=1.

The equations for min terms can be obtained from truth tables as follows, locate the line on the truth table which corresponds to the min term. Where the variable is equal to “0” write that variable as a complemented variable and where the variable is equal to “1” write that variable as an uncomplemented variable. Using m_1 as an example, in the truth table of section II. B, $A = 0$, $B = 0$, $C = 0$, and $D = 1$. As such, $m_1 = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D$.

The equations for min terms can be obtained from Karnaugh Maps as follows, from the box in the map note the binary value of variables *A* and *B*, from that column; and the binary values of variables *C* and *D* from that row. Where the variable is equal to “0” write that variable as a complemented variable and where the variable is equal to “1” write that variable as an uncomplemented variable. Using m_{11} as an example, in the four variable Karnaugh Map of section II. C, $AB = 10$, and $CD = 11$, and $m_{11} = A \cdot \overline{B} \cdot C \cdot D$.

To write a Boolean equation in the Standard Sum of Products form two steps are required. First, the min terms which evaluate to a “1” are written as an equation, as described above. Second, they are OR’d together. For this example the min terms that evaluate to a “1” are min terms 3, 4, 5, 7, 11, 13, 14,and 15. As such, the equation is:

$$f(A,B,C,D) = Y = \overline{A}\,\overline{B}\,C\,D + \overline{A}\,B\,\overline{C}\,\overline{D} + \overline{A}\,B\,\overline{C}\,D + \overline{A}\,B\,C\,D + A\overline{B}CD + A\,B\,\overline{C}\,D$$
$$+ ABC\overline{D} + ABCD$$

Now the combinations of variables that cause the equation to evaluate to a “1” are obvious. Since the Standard Sum of Products Form of the equation can become quite long there is a shorter way of writing this equation using min term notation. This is done by writing the min terms in their abbreviated notation then OR them together. For this example the equation in a Sum of Products Standard Form using min term notation is:

$$f(A,B,C,D) = Y = m_3 + m_4 + m_5 + m_7 + m_{11} + m_{13} + m_{14} + m_{15}$$
$$= \sum m(3,4,5,7,11,13,14,15)$$

where m_3, etc. are the min terms that evaluate to a 1. Note that each product term contains every variable *A*, *B*, *C*, and *D*.

E. Karnaugh Maps and Simplified Sum of Products Form of Equations

The Karnaugh Map can be used for simplifying logic equations. The difference between the Standard Sum of Products Form of Equations and simplified Sum of Products Form of Equations has to do with the contents of the min terms. In the Standard Sum of Products Form the min terms contain every variable. In the simplified Sum of Products Form the "min terms" may not contain every variable. As such, they are not min terms in the strict sense and should be referred to as "product terms". The simplified equations implemented in software are simpler to program since they contain fewer terms. They are also simpler to implement in hardware design since they require fewer logic gates. Just as with the standard form of the equation, it is obvious from the simplified form which combination of variables will cause the equation to evaluate to "1". A detailed example is below.

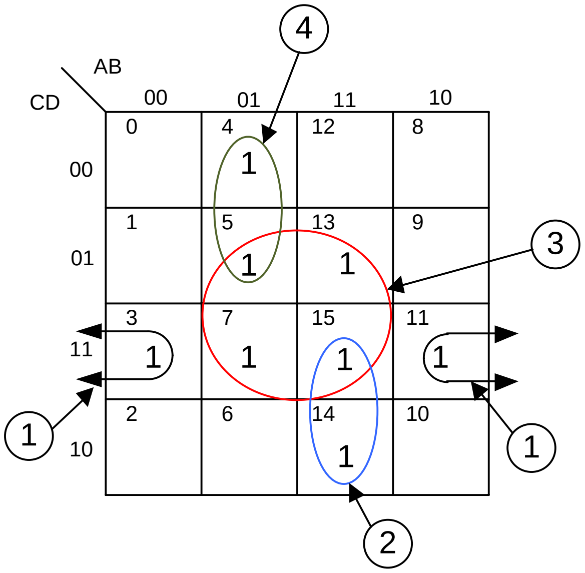
The following example will demonstrate how to use a Karnaugh Map to simplify an equation and write it in the simplified Sum of Products Form. The equation used in section II.B will be used and its truth table and Karnaugh Map are repeated below.

$$f(A,B,C,D) = Y = B(AC + AD + \overline{A} \bullet \overline{C}) + CD$$

A	*B*	*C*	*D*	*Y*	Decimal Equivalent of *ABCD* (min terms)
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	2
0	0	1	1	1	3
0	1	0	0	1	4
0	1	0	1	1	5
0	1	1	0	0	6
0	1	1	1	1	7
1	0	0	0	0	8
1	0	0	1	0	9
1	0	1	0	0	10
1	0	1	1	1	11
1	1	0	0	0	12
1	1	0	1	1	13
1	1	1	0	1	14
1	1	1	1	1	15

CD \ AB	00	01	11	10
00	0	4 1	12	8
01	1	5 1	13 1	9
11	3 1	7 1	15 1	11 1
10	2	6	14 1	10

To simplify the equation the following steps are performed using the Karnaugh Map. The first step is to draw circles around groups of the number "1" in adjacent boxes. Adjacent boxes are those which are to the right and left of each other, and to the top and bottom of each other. Boxes on the diagonal are not included in this definition. The top row is considered to be adjacent to the bottom row, and the right column is considered to be adjacent to the left column. The number of "1"s must be grouped in powers of 2, that is $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, etc. The larger the groups that can be circled, the better, since this will reduce the number of variables in the product terms of the simplified equation. Also, the same "1" can be included in more than one group.



The circled "1"s represent the terms that make up the product terms in the simplified equation. To write the equation for the product terms the row(s) and the columns(s) where the circled group of "1"s are located must be examined.

The circle #1 will be evaluated first. As previously stated, the left and right columns are considered to be adjacent. It is noted that this circle passes through columns where A takes on values of "1" and "0"; and where B = "0". That is, where AB = 00 and AB = 10. The fact that variable A takes on both values indicates that A is irrelevant and should not be included in this product term. Whenever a circle includes "0" and "1" values for a variable, then that variable is not included in the product term. Since B = 0 the product term includes $\overline{B}$. Whenever a circle includes only the "0" value for the variable, then the complement of that variable is included in the product term. The circle #1 lies entirely in the row where C = 1 and D = 1. That is, where CD = 11. This indicates that C and D are in the product term. Whenever a circle includes only the "1" value for the variable, then that variable is included in the product term. The product term that corresponds to the circle #1 is $\overline{B}CD$.

For the circle #2, this circle lies entirely in the column where $A=1$ and B = 1. That is, where AB = 11. As such, A and B are in the product term. It also lies in the rows where CD = 11, and CD = 10. Since D = 0 and D = 1 D is irrelevant. It also lies in the rows where C = 1, as such C is in the product term. The product term that corresponds to the circle #2 is ABC.

For the circle #3, this circle lies in the columns where A = 0 and A = 1, as such, A is irrelevant; and where B =1, as such, B is in the product term. The circle also lies in the rows where C = 0 and C = 1, as such the C term is irrelevant. The circle also lies in the rows where D = 1, as such D is in the product term. The product term that corresponds to the circle #3 is BD. This demonstrates the point that the larger groupings result in product terms with fewer variables.

The product term that corresponds to the circle #4 is $\overline{A}B\overline{C}$. The complete simplified equation in Sum of Products form is:

$$Y = ABC + \overline{B}CD + BD + \overline{A}B\overline{C}.$$

As is the case with the standard form, the combinations of variables that cause the equation to evaluate to a "1" are obvious. However, there are fewer terms.

There are several simple rules for choosing sets of 1's to be circled. First, start by circling those groups that can be circled only in a single way. For example, the "1" in box 14 can only be combined with box 15. Also, the "1" in box 4 can only be combined with box 5. This tends to leave larger groups that can be combined remaining. Second, use the fewest number of circles since this will result in the least number of product terms. Third, circles may overlap. In the example above boxes 14 and 15 were circled, and box 15 was also included in the group with 5, 7, and 13. The advantage to this is that although box 14 could have been in a group by itself, by combining it with box 15 the *D* variable was eliminated. Fourth, once a group of 1's have been included in a circle, do not circle them again. An example of this is the circle that includes boxes 5, 7, 13, and 15. There is no advantage to add additional circles, for example, to circle boxes 5 and 13 as another set and 7 and 15 as another set. Finally, it should be stated that good judgment for choosing circles comes with experience. This method may result in an equation that can be further simplified using the basic theorems discussed in Section II.A.

III. Example

A control system operates a contactor that energizes resistance heaters. Also, temperature, pressure, and flow are inputs monitored by this control system. For the inputs a "1" represents normal operating conditions and a "0" represents abnormal operating conditions. The temperature sensor, flow sensor, and pressure sensor normally send a "1" to the control system. Whenever flow is less than 3 GPM the output of the flow sensor changes to a "0". Whenever temperature is greater than 500 °F the output of the temperature sensor changes to a "0". Whenever pressure is greater than 1500 psig, the output of the pressure sensor changes to a "0". There are two conditions that cause the heaters to de-energize. First, flow is less than 3 GPM. Second, pressure is above 1500 psig and temperature is above 500 °F. These inputs are to be used in an equation. When the equation evaluates to a "1" the contactor energizes the heaters, and a "0" causes the contactor to de-energize the heaters. Write the Boolean equation to control the heaters, that is, keep the heaters energized during normal operating conditions.

To solve this type of problem perform the following steps:

1) Define input variables and their states, for this problem:

A – Flow, "0" when less than 3 GPM, the abnormal operating condition; and a "1" for the normal operating condition
B –Temperature, "0" when greater than 500 °F, the abnormal operating condition; and a "1" for the normal operating condition
C – Pressure, "0" when greater than 1500 psi, the abnormal operating condition; and a "1" for the normal operating condition
One should be consistent with the assignment of logic states for normal and abnormal conditions. In this example and the quiz problems that follow a "1" will represent a normal condition and a "0" will represent an abnormal condition.

2) Define the output variable and its state, for this problem:

Y – A “1” causes the contactor to energize the heaters, and a “0” causes the contactor to de-energize the heaters.

Write the truth table listing all combinations of input variables, the output variable (leave blank for now), the min terms (this is convenient since a Karnaugh Map will be involved in solving these problems), and a column which explains in words what the inputs represent.

A	*B*	*C*	*Y*	Min Term	Inputs
0	0	0		0	
0	0	1		1	
0	1	0		2	
0	1	1		3	
1	0	0		4	
1	0	1		5	
1	1	0		6	
1	1	1		7	

The next step is to fill in the Inputs column, this relates the “1”s and “0”s of the input variables to physical conditions of the input.

A	*B*	*C*	*Y*	Min Term	Inputs
0	0	0		0	Low flow, high temperature, high pressure
0	0	1		1	Low flow, high temperature, normal pressure
0	1	0		2	Low flow, normal temperature, high pressure
0	1	1		3	Low flow, normal temperature, normal pressure
1	0	0		4	Normal flow, high temperature, high pressure
1	0	1		5	Normal flow, high temperature, normal pressure
1	1	0		6	Normal flow, normal temperature, high pressure
1	1	1		7	All conditions normal

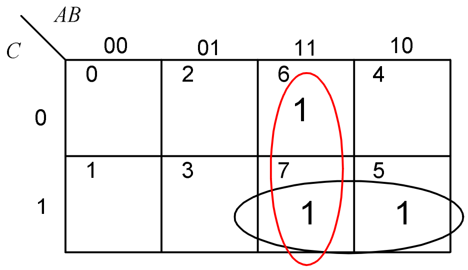
From the problem statement determine which combinations of input variables will cause the heaters to de-energize, *Y*=0. Alternatively, one could determine which combinations of input variables will cause the heaters to energize, *Y*=1, and work the problem that way. For this problem high temperature and high pressure must exist simultaneously for the heaters to de-energize, also low flow alone will cause the heaters to de-energize. All other conditions will cause the heaters to remain energized, *Y*=1.

Note, that one could have started the solution by determining which combination of input values would have caused the heaters to energize.

A	*B*	*C*	*Y*	Min Term	Inputs
0	0	0	0	0	Low flow, high temperature, high pressure
0	0	1	0	1	Low flow, high temperature, normal pressure
0	1	0	0	2	Low flow, normal temperature, high pressure
0	1	1	0	3	Low flow, normal temperature, normal pressure
1	0	0	0	4	Normal flow, high temperature, high pressure
1	0	1	1	5	Normal flow, high temperature, normal pressure
1	1	0	1	6	Normal flow, normal temperature, high pressure
1	1	1	1	7	All conditions normal

The value of min terms m_0, m_1, m_2, and m_3 are "0", heaters are de-energized, since the flow condition of less than 3 GPM exists. The output of min term m_4 is "0" since the high temperature and high pressure conditions exist. The heaters remain energized for min terms m_5, m_6, and m_7.

The "1"s are plotted on a Karnaugh Map and groups are circled. This will provide the equation which keeps the heaters energized, *Y*=1, and when the equation evaluates to "0" the heaters are de-energized.



The output equation is $Y = AB + AC$. As a final step generate a Truth Table using the simplified equations to verify that it is correct.