Unit 15 Matrix minimization method.

- For the successful operation of the Karnaugh map technique, it is necessary that:
  - Adjacent terms in the map differ in only one literal.
  - All possible adjacencies be discovered.
- The Karnaugh map method has to be modified for five inputs and does not work for more than five inputs.
- An alternative mapping method for minimization of systems having more than four inputs is presented.

The Karnaugh map technique depends on using the Gray code numbering sequence to ensure that adjacent canonical terms in the Karnaugh map differ in only one literal. This is a necessary but not sufficient condition for the success of the method. It is also necessary that all possible adjacencies be discovered by the technique. This second requirement is satisfied for four literals, with two literals assigned to each map axis, as we have seen in Unit 14. When there are five or more literals, then there must be three or more literals on one axis of the Karnaugh map and this new condition means that all adjacencies are not necessarily revealed. To show this, we consider a single row with three literals along the X axis as shown in the table below.

$\overline{x_2}\overline{x_1}\overline{x_0}$	$\overline{x_2}\overline{x_1}x_0$	$\overline{x_2}x_1x_0$	$\overline{x_2}x_1\overline{x_0}$	$x_2x_1\overline{x_0}$	$x_2x_1x_0$	$x_2\overline{x_1}x_0$	$x_2\overline{x_1}\overline{x_0}$
0	1	0	0	0	0	1	0

This map does not show any adjacent pairs so no simplification seems possible and yet it is easily seen that the terms $\overline{x_2}.\overline{x_1}.x_0$ and $x_2.\overline{x_1}.x_0$, while not adjacent, can be combined to give the simplification:

$$\overline{x_2}\overline{x_1}x_0 + x_2\overline{x_1}x_0 = \overline{x_1}x_0$$

So we can now state that for, three or more literals on an axis of a Karnaugh map, the Gray code sequencing of terms does not reveal all possible adjacent terms and therefore the simple form of the Karnaugh map technique may not reveal a maximally simple system. Essentially, we are trying to illustrate or portray an n dimensional problem of n literals on a two dimensional Karnaugh map and it is not reasonable to expect that the portrayal and resulting simplification will be complete and maximal.

There is therefore a problem associated with the problem of the minimization of systems containing five or more variables which cannot be easily handled by the Karnaugh map technique. There is also a range of problems which are not large enough to justify the use of the more powerful algorithms such as the Quine-McCluskey, the Espresso or the BDD methods (which will be discussed in later Units) and computer packages based on these algorithms which are designed for handling large systems. What is required is a pencil and paper method of getting reasonable and quick but not necessarily optimum solutions to medium sized minimization problems.

The following paper and pencil method is suggested for the minimization of Boolean expressions containing up to about ten inputs.

Consider the set of decimal numbers from 0 to 63, spaced linearly along a piece of string. Then fold the string into four and form a number array with positions as shown in the figure.

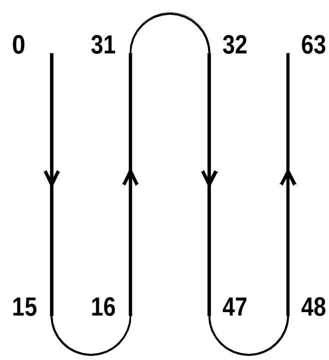

Figure 15.1: Folded ordering of table positions.

The following algorithm is used for generating a table for use in simplifying Boolean minterms. Take the minterm number and convert it from decimal to binary. Treat this binary number as a Gray code number and convert it from Gray to binary using the algorithm described in Figure 7.5 (b). Convert the resulting binary to a decimal number and write the original minterm number in this position in the table.

For instance, minterm number 55 is 110111 in binary. Treat 110111 as a Gray code and convert it to binary using Figure 7.5(b) to obtain 100101. Convert binary 100101 to decimal to obtain 37. Locate the 37th position in the folded table formed as shown in Figure 15.1 and write 55 into that position. This is shown in the table below. The positions of the other minterms are similarly calculated and the complete table is prepared. This table can then be reproduced and used by circling the minterm numbers and searching for pairs, quads etc as in the Karnaugh map method. (See Appendix C)

0	16	48	32
1	17	49	33
3	19	51	35
2	18	50	34
6	22	54	38
7	23	55	39
5	21	53	37
4	20	52	36
12	28	60	44
13	29	61	45
15	31	63	47
14	30	62	46
10	26	58	42
11	27	59	43
9	25	57	41
8	24	56	40

The advantage of this method over the Karnaugh map method is that it can be used for systems having more than four inputs. In this example, we have six inputs but we can handle more if necessary. The only requirement is that we have a preprepared table with the location of the minterms marked on it. Thus for a system with n inputs we have a table with 2^n entries.

Now take as an example a function of six variables specified by the minterm list

$$\begin{aligned} f_1(x_5, x_4, x_3, x_2, x_1, x_0) &= \Sigma m(5, 6, 12, 21, 26, 27, 30, 31, \\ &\quad 37, 44, 45, 46, 47, 53, 58, 62) \end{aligned}$$

Make a copy of the map table for six inputs and circle the minterms which appear in the minterm list as shown.

0	16	48	32
1	17	49	33
3	19	51	35
2	18	50	34
(6)	22	54	38
7	23	55	39
(5)	(21)	(53)	(37)
4	20	52	36
(12)	28	60	(44)
13	29	61	(45)
15	(31)	63	(47)
14	(30)	(62)	(46)
10	(26)	(58)	42
11	(27)	59	43
9	25	57	41
8	24	56	40

Pairs of terms occur symmetrically on either side of lines dividing the table into halves, quarters or eights. Thus the minterms 58 and 62 form a pair. The terms 12 and 44 also form a pair. The terms 49 and 57 would have also formed a pair if they had had occurred in the minterm list. However, terms 55 and 59 could not form a pair because they are not symmetrically spaced.

It can also be seen that the terms (31, 30, 26, 27) form a quad as do (5, 21, 53, 37), (30, 62, 26, 58) and also (44, 45, 47, 46). Note that a group of four such as 49, 51, 50 and 54 could not form a quad because they are not spaced symmetrically about a quartering line but that 49 and 51 could form a pair as could 50 and 54. The minterm 6 does not combine with any other minterm.

When the minterms in the minterm list are grouped and quads and pairs simplified, the function f_1 reduces to:

$$\begin{aligned} f_1(x_5,x_4,x_3,x_2,x_1,x_0) &= \Sigma m((5,21,53,37),(31,30,26,27),\\ &\quad (30,62,26,58),(44,45,47,46),(12,44),6)\\ &= \overline{x_3}x_2\overline{x_1}x_0+\overline{x_5}x_4x_3x_1+x_4x_3x_1\overline{x_0}+\\ &\quad x_5\overline{x_4}x_3x_2+\overline{x_4}x_3x_2\overline{x_1x_0}+\overline{x_5x_4x_3}x_2x_1\overline{x_0} \end{aligned}$$

This tabular map can be extended to the case of 7 inputs by preparing a larger mapping table as shown:

0	16	48	32	96	112	80	64
1	17	49	33	97	113	81	65
3	19	51	35	99	115	83	67
2	18	50	34	98	114	82	66
6	22	54	38	102	118	86	70
7	23	55	39	103	119	87	71
5	21	53	37	101	117	85	69
4	20	52	36	100	116	84	68
12	28	60	44	108	124	92	76
13	29	61	45	109	125	93	77
15	31	63	47	111	127	95	79
14	30	62	46	110	126	94	78
10	26	58	42	106	122	90	74
11	27	59	43	107	123	91	75
9	25	57	41	105	121	89	73
8	24	56	40	104	120	88	72

The valuable feature of this alternative mapping method is that the terms which can be readily combined are revealed as a pattern on the map and can be easily identified. The technique can be applied to functions which have 7 and more inputs whereas the Karnaugh map method is limited to 5 inputs. The technique is also adapted to direct use of the minterm list presentation of the truth table which simplifies the preparation of the map.

There are a number of standard methods which have been applied to this Boolean simplification problem and for which commercial computer program packages are available. These are the Quine- McCluskey method and the Espresso method but, as in all theoretically interesting problems, many other methods have been developed and explored such as methods based on Reed-Muller simplification, Binary Decision Diagrams, Genetic Algorithms, cellular automata and others. We will discuss these methods in the next few units because the techniques used have more general applications that simply the Boolean minimization problem.

15.1 References

Maxfield, C.(1996), *A Reed Muller extraction utility*, EDN (Electronic Design Notes), April 11, 121-135

15.2 Problems

See Appendix C for blank minimization maps.

15.1 Use the method described in this unit to minimize the function

$$f(x_3, x_2, x_1, x_0) = \Sigma m(1, 5, 7, 9, 12, 13, 14)$$

Verify that you obtain the same simplified function that was obtained in Example 14.1.

15.2 How many columns will there be in the table for the case of n different inputs?

15.3 Simplify the expression:

$$f_1(x_3, x_2, x_1, x_0) = \Sigma m(1, 2, 3, 6, 8, 10, 12, 13, 14)$$

15.4 Simplify the expression:

$$f_2(x_5, x_4, x_3, x_2, x_1, x_0) = \Sigma m(19, 23, 27, 31, 55, 59, 63)$$

15.5 Simplify the expression:

$$\begin{aligned} f_3(x_5, x_4, x_3 x_2, x_1, x_0) &= \Sigma m(5, 13, 16, 17, 18, 19, 21, 25, 27, 29, 33, 37, \\ &\quad 45, 48, 49, 50, 51, 53, 57, 59, 61) \end{aligned}$$

15.6 Simplify the expression:

$$\begin{aligned} f_3(x_6, x_5, x_4, x_3, x_2, x_1, x_0) &= \Sigma m(3, 7, 11, 13, 19, 27, 39, 47, 49, 53, 67, \\ &\quad 71, 75, 79, 83, 91, 103, 111, 113, 117) \end{aligned}$$

15.7 Construct the table, similar to that on page 92, which would be suitable for use in minimizing expressions having 8 inputs, that is minterms between 0 and 255. Use the table to minimize the expression

$$f_3(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) = \Sigma m(25, 67, 74, 75, 106, 153, 171, 195, 203)$$