Chapter 1

Quine-McCluskey Method

This document is a DRAFT. Please email all corrections and suggestions to:

nicholas.outram@plymouth.ac.uk

with the subject heading:

"QM Document".

The Karnaugh Map (KM) method of logic simplification works very well for 4 variables or less. It is quick and simple, and can be performed by hand on paper. It can be extended to 5 variables or even more, but it becomes far less intuitive and more error prone. An alternative technique that is logically equivalent to Karnaugh Maps that scales to many variables is known as the Quine-McCluskey (QM) method. This technique is tabular and can still be performed by hand on paper. It is also relatively simple to automate with a computer. Although other computer based techniques have since superseded it (reference), it is still a useful technique as it is functioanally equivalent to a Karnaugh Map and can still be performed manually on paper.

In the next section, the origins of the QM technique are demonstrated and contrasted with the KM method to illustrate their equivalence.

1.1 Underlying principles and notation

The fundamental principle underlying both KM and QM is the following identity:

$$A \cdot B + A \cdot \overline{B} = A \cdot (B + \overline{B}) = A \tag{1.1}$$

In this expression, A is the common factor. The terms $A \cdot B$ and $A \cdot \overline{B}$ are known as **min-terms**. Each min-term can be represented as a binary number, where $A \cdot B = 11b$ and $A \cdot \overline{B} = 10b$. Each term can also be represented as an integer value, where $m3 = 11b$ and $m2 = 10b$ (the m stands for min-term). This is also illustrated from the truth table for this Boolean expression.

min term **m**	A	B	Y
$m0$	0	0	0
$m1$	0	1	0
$m2$	1	0	1
$m3$	1	1	1

First write the two terms one above the other, and compare the Logic expression with the binary representation.

Logic Expression	min-term	binary representation
$A \cdot \overline{B}$	m2	10
$A \cdot B$	m3	11

Note that for the binary representation, the first digit is the same (1 in this case) in both terms. This represents the common factor A. In contrast, the second digit is different in each term ($\overline{B}$and B respectively). This is the variable that is removed using (1.1).

> Where two min-terms differ by **only** one bit, then the terms can be merged to form a new simplified term where one variable can be eliminated.

This is illustrated in the following table.

Logic Expression	min-term	binary representation
$A \cdot \overline{B}$	$m2$	10
$A \cdot B$	$m3$	11
A	$(m2, m3)$	1-

The new simplified term is labelled $(m2, m3)$. As will be seen later, this provides important tractability back to the original min-terms. Note also the binary notation. The '-' symbol is used to mark where a variable has been removed through simplification using (1.1). Again, this will become important later.

Note the following:

- Where there is one bit different, it is the variable represented by the bit that differs that is removed
- This process is simply a numerical equivalent to applying the identity in (1.1)

example 1.1.1: Using the approach described above, simplify the following expression:

$$Y = A \cdot B \cdot C + A \cdot \overline{B} \cdot C$$

This can also be written as:

$$Y = m7 + m5$$

In this simple case, it can instantly be simplified using Boolean algebra, whereby $A \cdot C \cdot (B + \overline{B}) = A \cdot C$. Consider the tabular approach. In the table below, the first two rows show the binary representation of each min-term. From the binary representation, the first and third (A and C) are equal (common factors), whereas the second variable (B) is not common. From this, the simplified term 1-1 is derived.

Logic Expression	min-term	binary representation
$A \cdot B \cdot C$	m7	1**1**1
$A \cdot \overline{B} \cdot C$	m5	1**0**1
$A \cdot C$	(m5,m7)	1-1

The simplified expression is therefore $Y = A \cdot C$. Compare this to the Karnaugh Map for the exact same expression. Note the loop spans one column, whereby the variables A and $C$are seen to be constant.

	$\overline{B} \cdot \overline{C}$	$\overline{B} \cdot C$	$B \cdot C$	$B \cdot \overline{C}$
$\overline{A}$	0 0	0 1	0 3	0 2
A	0 4	1 5	1 7	0 6

example 1.1.2: Using a tabular method, simplify the Boolean expression $Y = \overline{A} \cdot B \cdot C + \overline{A} \cdot B \cdot \overline{C}$

Logic Expression	min-term	binary representation
$\overline{A} \cdot B \cdot C$	$m3$	01**1**
$\overline{A} \cdot B \cdot \overline{C}$	$m4$	01**0**
$\overline{A} \cdot B$	$(m3, m4)$	01-

This time the common factor is $\overline{A} \cdot B$ and the variable C differs in the both terms. This can again be seen in the binary representation where bit 0 (C) is the only bit that differs. The simplified expression is therefore $Y = \overline{A} \cdot B$

This tabular approach is equivalent to creating a loop that spans two adjacent rows **or** two adjacent columns in a KM. In the next section, this principle is extended through the repeated application of (1.1) to the remiaining terms, which is equivalent to creating loops that span more than 2 columns on a KM.This the full generalised QM method.

Hopefully it is becoming clear to the reader that the KM and tablar (QM) methods are logically equivalent. For up to 4 variables, the KM is quicker and more intuitive. However, unlike the KM, the QM method scales well beyond 4 variables and can still be performed manually. In the following sections, the complete QM method will be discussed.

There are cases where both KM and QM techniques produce a unique solution and other cases that result in multiple solutions, sometimes with equal merit. Further analysis is sometimes required to find the best solution and this is discussed in section 1.4. To help the reader, the cases with unique solutions will be considered first.

1.2 Simplifications with unique solutions

To assist understanding, the initial focus will be on 3 or 4 variable problems so that the reader can also compare results with Karnaugh Maps. In this section, examples will be used to illustrate the general method.

example 1.2.1: Simplify the following expression:

$$Y = A \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C}$$

$$Y = m7 + m5 + m6$$

Using the same approach as above, the terms are written into a table, and where possible, all possible terms that differ by one bit are merged.

Logic Expression	min-term	binary representation	Number of 1's
$A \cdot \overline{B} \cdot C$	m5	101	2
$A \cdot B \cdot \overline{C}$	m6	110	2
$A \cdot B \cdot C$	m7	111	3
$A \cdot C$	(m5,m7)	1-1	2
$A \cdot B$	(m6,m7)	11-	2

For reasons that will be explained later, the number of 1's in the binary representation of each term are also calculated. This time, there are two combinations of terms that differ by 1 bit: $(m7, m5)$ and $(m7, m6)$. Looking at these in detail:

$$m5 = 101$$

$$m7 = 111$$

Here the two terms differ by one bit (bit 1), so the simplified term is $1 - 1$

$$m6 = 110$$

$$m7 = 111$$

Again the two terms differ by one bit (bit 0), so the simplified term is $11-$

The results of these two simplifications are written at the bottom of the table (see table above). From the table $Y = A \cdot C + A \cdot B$

Now consider the Karnaugh map for this case.

	$\overline{B}\cdot\overline{C}$	$\overline{B}\cdot C$	$B\cdot C$	$B\cdot\overline{C}$
$\overline{A}$	0 $_{0}$	0 $_{1}$	0 $_{3}$	0 $_{2}$
A	0 $_{4}$	1 $_{5}$	1 $_{7}$	1 $_{6}$

Note there are two overlapping loops. The left loop is $(m5, m7)$ and the right loop is $(m6, m7)$. It is also apparent that $m7$ is a common term in both expressions.

example 1.2.2:

Simplify the following expression

$$Y = \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D} + \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot D + \overline{A}\cdot B\cdot\overline{C}\cdot\overline{D} + \overline{A}\cdot B\cdot\overline{C}\cdot D$$

The Karnaugh Map for this expression is as follows:

	$\overline{C}\cdot\overline{D}$	$\overline{C}\cdot D$	$C\cdot D$	$C\cdot\overline{D}$
$\overline{A}\cdot\overline{B}$	1 $_{0}$	1 $_{1}$	0 $_{3}$	0 $_{2}$
$\overline{A}\cdot B$	1 $_{4}$	1 $_{5}$	0 $_{7}$	0 $_{6}$
$A\cdot B$	0 $_{12}$	0 $_{13}$	0 $_{15}$	0 $_{14}$
$A\cdot\overline{B}$	0 $_{8}$	0 $_{9}$	0 $_{11}$	0 $_{10}$

In this simple case, a loop can be drawn around all 4 min-terms. By inspection, this simplifies to $\overline{A}\cdot\overline{C}$. Now, consider the binary representation of the min-terms.

Logic Term	min term	binary representation	Numbers of '1's
$\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}$	m0	0000	0
$\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot D$	m1	0001	1
$\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}$	m4	0100	1
$\overline{A}\cdot B\cdot\overline{C}\cdot D$	m5	0101	2

Looking at the binary representation, it is possible to find combinations of terms that differ by one bit and simplify using the approach described above. The table below lists all possible combinations.

Logic Term	min term	binary representation	Numbers of '1's
$\overline{A} \cdot \overline{B} \cdot \overline{C}$	(m0,m1)	000-	0
$\overline{A} \cdot \overline{C} \cdot \overline{D}$	(m0,m4)	0-00	0
$\overline{A} \cdot \overline{C} \cdot D$	(m1,m5)	0-01	1
$\overline{A} \cdot B \cdot \overline{C}$	(m4,m5)	010-	1

TASK: Consider what these new simplified logic terms mean on a Karnaugh Map. Draw the loops on the Karnaugh Map for all the min-terms in this table

These terms represent all possible loops (spanning two rows or two columns). The next step in the QM method is to (where possible) further combine these terms to obtain simpler expressions.

Terms which can be combined must meet the following criteria:

- They differ by **only** one bit
- They contain the same domain of variables (therefore the '-' symbols must line up)

Given the above, these terms can be combined. Some useful hints to speed up this process are as follows:

- Count the number of 1's in each term,and sort the table rows respectively (see below)
- Terms with n number of 1's can **only** be combined with terms with $n + 1$ number of 1's.
  - Terms that can be merged must come from the same domain[1], so look for aligned '-' digits, and then from those, find the terms which are one bit different.
  - Where they can be merged, the result will have n number of 1's

In this example the following terms can be combined

- $(m0, m1)$ and $(m4, m5)$
- $(m0, m4)$ and $(m1, m5)$

The results are shown in the following table.

Logic Term	min term	binary representation	Numbers of '1's
$\overline{A} \cdot \overline{B} \cdot \overline{C}$	(m0,m1)	000-	0
$\overline{A} \cdot \overline{C} \cdot \overline{D}$	(m0,m4)	0-00	0
$\overline{A} \cdot \overline{C} \cdot D$	(m1,m5)	0-01	1
$\overline{A} \cdot B \cdot \overline{C}$	(m4,m5)	010-	1
$\overline{A} \cdot \overline{C}$	$(m0, m1, m4, m5)$	0-0-	0
$\overline{A} \cdot \overline{C}$	$(m0, m4, m1, m5)$	0-0-	0

[1] They contain the same variables

Note that in this case, the two remaining terms are identical, so one can be deleted. The process stops as there is only one term left. The final result is therefore $\overline{A} \cdot \overline{C}$.

Now, consider a more complex expression, again with only 4 variables (so the reader can directly compare with a Karnaugh Map)

example 1.2.3:

Simplify the following expression:

$$Y = \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D} + \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot D + \overline{A}\cdot B\cdot\overline{C}\cdot\overline{D} + \overline{A}\cdot B\cdot\overline{C}\cdot D + \overline{A}\cdot B\cdot C\cdot D + A\cdot\overline{B}\cdot C\cdot D$$

$$Y = m0 + m1 + m4 + m5 + m7 + m11$$

Draw the Karnaugh Map and we can show by inspection that the simplified expression is:

$$Y = \overline{A} \cdot \overline{C} + \overline{A} \cdot B \cdot D + A \cdot \overline{B} \cdot C \cdot D$$

	$\overline{C} \cdot \overline{D}$	$\overline{C} \cdot D$	$C \cdot D$	$C \cdot \overline{D}$
$\overline{A} \cdot \overline{B}$	1 $_{0}$	1 $_{1}$	0 $_{3}$	0 $_{2}$
$\overline{A} \cdot B$	1 $_{4}$	1 $_{5}$	1 $_{7}$	0 $_{6}$
$A \cdot B$	0 $_{12}$	0 $_{13}$	0 $_{15}$	0 $_{14}$
$A \cdot \overline{B}$	0 $_{8}$	0 $_{9}$	1 $_{11}$	0 $_{10}$

First, tabulate the terms, sorted by increasing number of 1's

Logic Term	min term	binary representation	Numbers of '1's
$\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$	m0	0000	0
$\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D$	m1	0001	1
$\overline{A} \cdot B \cdot \overline{C} \cdot \overline{D}$	m4	0100	1
$\overline{A} \cdot B \cdot \overline{C} \cdot D$	m5	0101	2
$\overline{A} \cdot B \cdot C \cdot D$	$m7$	0111	3
$A \cdot \overline{B} \cdot C \cdot D$	$m11$	1011	3

Now examine the terms that have n number of 1's with terms that have $n + 1$ number of 1's, and look for pairs of terms that differ by one bit.

Logic Term	min term	binary representation	Numbers of '1's
$\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$	m0	0000	0
$\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D$	m1	0001	1
$\overline{A} \cdot B \cdot \overline{C} \cdot \overline{D}$	m4	0100	1
$\overline{A} \cdot B \cdot \overline{C} \cdot D$	m5	0101	2
$\overline{A} \cdot B \cdot C \cdot D$	$m7$	0111	3
$A \cdot \overline{B} \cdot C \cdot D^*$	$m11^*$	1011	3
$\overline{A} \cdot \overline{B} \cdot \overline{C}$	$(m0, m1)$	000-	0
$\overline{A} \cdot \overline{C} \cdot \overline{D}$	$(m0, m4)$	0-00	0
$\overline{A} \cdot \overline{C} \cdot D$	$(m1, m5)$	0-01	1
$\overline{A} \cdot B \cdot \overline{C}$	$(m4, m5)$	010-	1
$\overline{A} \cdot B \cdot D$	$(m5, m7)$	01-1	2

Note that one of the terms, $m11$, cannot be combined. This is marked as a starred item, meaning the term has to be kept in the final result. This can be seen on the KM. For the next phase, repeat the procedure on the combined terms.

Logic Term	min term	binary representation	Numbers of '1's
$\overline{A} \cdot \overline{B} \cdot \overline{C}$	$(m0, m1)$	000-	0
$\overline{A} \cdot \overline{C} \cdot \overline{D}$	$(m0, m4)$	0-00	0
$\overline{A} \cdot \overline{C} \cdot D$	$(m1, m5)$	0-01	1
$\overline{A} \cdot B \cdot \overline{C}$	$(m4, m5)$	010-	1
$\overline{A} \cdot B \cdot D^*$	$(m5, m7)^*$	01-1	2
$\overline{A} \cdot \overline{C}$	$(m0, m1, m4, m5)$	0-0-	0
$\overline{A} \cdot \overline{C}$	$(m0, m4, m1, m5)$	0-0-	0

The starred terms are those that cannot be combined any further. Not that $(m5, m7)$ is the only 3-variable term that cannot be combined with any other. Again, this can be seen on the KM as a loop of two cells. For the simplified two-variable results, both are identical, and using the identity $X + X \equiv X$, only one of them is kept.

A table of all starred and remaining terms is created as follows:

	$m0$	$m1$	$m4$	$m5$	$m7$	$m11$
$A \cdot \overline{B} \cdot C \cdot D^*$						X
$\overline{A} \cdot B \cdot D^*$				X	X	
$\overline{A} \cdot \overline{C}$	X	X	X	X		

Using this table it is confirmed that all the original min-terms are covered. Relating this to the Karnaugh Map, all cells with a '1' are covered by these terms. As will be seen later, there are cases where different combinations of terms cover the set, but where not all are required. Before considering these cases however, it is important to consider the situation where "don't care" are included in the original truth table.

1.3 Simplifications with don't care terms

Some combination logic contains don't care terms, where a set of specific input conditions are defined to never occur or where the output has no consequence. The designer therefore has the choice to choose the actual value of the don't care terms to aid simplification. Consider the following truth table.

	A	B	C	Y
$d0$	0	0	0	X
$m1$	0	0	1	0
$m2$	0	1	0	0
$m3$	0	1	1	0
$m4$	1	0	0	1
$m5$	1	0	1	1
$d6$	1	1	0	X
$m7$	1	1	1	1

One interpretation of this truth table is that the input combinations 000 and 111 never occur, so the output is a don't care condition. The Karnaugh Map for this truth table is shown below. By visual inspection, the simplification is simply $Y = A$, whereby the don't care term $d6$ is taken to be a '1' and the don't care term $d0$ is defined as a 0.

	$\overline{B} \cdot \overline{C}$	$\overline{B} \cdot C$	$B \cdot C$	$B \cdot \overline{C}$
$\overline{A}$	X $_{0}$	0 $_{1}$	0 $_{3}$	0 $_{2}$
A	1 $_{4}$	1 $_{5}$	1 $_{7}$	X $_{6}$

Note the new notation introduced here. The don't care terms are prefixed with a d. The tabular approach is similar to the above. First, find all possible combinations of merged terms.

Logic Term	min term	binary representation	Numbers of '1's
$\overline{A} \cdot \overline{B} \cdot \overline{C}$	$d0$	000	0
$A \cdot \overline{B} \cdot \overline{C}$	$m4$	100	1
$A \cdot \overline{B} \cdot C$	$m5$	101	2
$A \cdot B \cdot \overline{C}$	$d6$	110	2
$A \cdot B \cdot C$	$m7$	111	3
$\overline{B} \cdot \overline{C}$	$(d0, m4)$	-00	0
$A \cdot \overline{B}$	$(m4, m5)$	10-	1
$A \cdot \overline{C}$	$(m4, d6)$	1-0	1
$A \cdot C$	$(m5, m7)$	1-1	2
$A \cdot B$	$(d6, m7)$	11-	2

Note that all possible combinations have been considered at this stage. Also note that all of the original terms have been combined with another. One more iteration is be applied.

Logic Term	min term	binary representation	Numbers of '1's
$\overline{B} \cdot \overline{C}^{*}$	$(d0, m4)^{*}$	-00	0
$A \cdot \overline{B}$	$(m4, m5)$	10-	1
$A \cdot \overline{C}$	$(m4, d6)$	1-0	1
$A \cdot C$	$(m5, m7)$	1-1	2
$A \cdot B$	$(d6, m7)$	11-	2
A	$(m4, m5, d6, m7)$	1–	1
A	$(m4, d6, m5, m7)$	1–	1

The term $(d0, m4)$ cannot be combined so it is starred, The other terms are combined to produce two identical results, of which only one is kept.

The remaining terms are $(d0, m4)$ and $(m4, m5, d6, m7)$. Now create a table of all essential terms

	$m4$	$m5$	$m7$
$\overline{B} \cdot \overline{C}$	X		
A	X	X	X

From this table, note the following:

- Only the minterm columns are included (the don't care terms do not need to be covered by the final result).
- The minterms $m5$ and $m7$ are only covered by the logic term A. Therefore, **the logic term A is compulsary**.
- From this table, all essential minterms are covered by the logic expression A $(m4, m5, d6, m7)$ so no other terms are needed.
- Note also that like the KM, this logic term uses the don't care term $d6$ (but not $d0$),
- The term $\overline{B} \cdot \overline{C}$ $(d0, m4)$ is not needed as $m4$is already covered so the don't care term $d0$ is not used (defined as a zero).
- The final result is therefore $Y = A$

1.4 Simplifications with multiple solutions

It is not always the case that a KM or the QM method will result in one unique solution. There may even be multiple solutions of the same merit (requiring the same number of logical operations).In cases where this is non-trivial, further analysis is required.

Consider the Karnaugh Maps shown below. Both contain the same minterms, but two solutions of equal merit are provided. Note also that these represent the simplest possible solutions without consideration for static hazards (there are no overlapping loops).

Table 1.1: Alternative Karnaugh Map solutions with equal merit

	$\overline{B}\cdot\overline{C}$	$\overline{B}\cdot C$	$B\cdot C$	$B\cdot\overline{C}$
$\overline{A}$	1 0	0 1	1 3	1 2
A	1 4	1 5	1 7	0 6

	$\overline{B}\cdot\overline{C}$	$\overline{B}\cdot C$	$B\cdot C$	$B\cdot\overline{C}$
$\overline{A}$	1 0	0 1	1 3	1 2
A	1 4	1 5	1 7	0 6

The solutions for these maps are $Y = \overline{B}\cdot\overline{C} + A\cdot C + \overline{A}\cdot B$(upper) and $Y = \overline{A}\cdot\overline{C} + B\cdot C + A\cdot\overline{B}$ (lower)

Now consider the QM tabular approach.

Logic Term	min term	binary representation	Numbers of '1's
$\overline{A}\cdot\overline{B}\cdot\overline{C}$	$m0$	000	0
$\overline{A}\cdot B\cdot\overline{C}$	$m2$	010	1
$A\cdot\overline{B}\cdot\overline{C}$	$m4$	100	1
$\overline{A}\cdot B\cdot C$	$m3$	011	2
$A\cdot\overline{B}\cdot C$	$m5$	101	2
$A\cdot B\cdot C$	$m7$	111	3
$\overline{A}\cdot\overline{C}$	$(m0, m2)$	0-0	0
$\overline{B}\cdot\overline{C}$	$(m0, m4)$	-00	0
$\overline{A}\cdot B$	$(m2, m3)$	01-	1
$A\cdot\overline{B}$	$(m4, m5)$	10-	1
$B\cdot C$	$(m3, m7)$	-11	2
$A\cdot C$	$(m5, m7)$	1-1	2

From the result above, no more terms can be combined.

	$m0$	$m2$	$m3$	$m4$	$m5$	$m7$
$\overline{A} \cdot \overline{C}$	X	X				
$\overline{B} \cdot \overline{C}$	X			X		
$\overline{A} \cdot B$		X	X			
$A \cdot \overline{B}$				X	X	
$B \cdot C$			X			X
$A \cdot C$					X	X

The task is now to select the optimum set of logic terms to cover all the minterms (columns).This time, there are no starred items as no logic term exclusively covers any minterm (in this example, all minterms are covered by two logic terms). For such a simple case, the opimum set of logic terms could be selected by inspection.

Some possible solutions are:

$Y = \overline{B} \cdot \overline{C} + A \cdot C + \overline{A} \cdot B$

$Y = \overline{A} \cdot \overline{C} + B \cdot C + A \cdot \overline{B}$

$Y = \overline{A} \cdot \overline{C} + \overline{A} \cdot B + A \cdot \overline{B} + A \cdot C$

The first two are identical to the result from the Karnaugh Map and have equal merit in terms of the number of logical operations. This approach is only manageable because this is a simple example. Such an approach would become unweildy for problems with large numbers of variables. An analytical approach, known as Petrick's method (**ref**), can be used to solve for the simplest solution(s).

1.4.1 Petrick's Method

Petrick's method is a simple analytical solution foridentifying the simplest permutation(s) of terms that cover all minterms. In contrast to the QM method, the underlying Boolean identity is the following:

$$X + XY \equiv X \tag{1.2}$$

Consider the example above:

Table 1.2: Prime implicant table

		$m0$	$m2$	$m3$	$m4$	$m5$	$m7$
$I0$	$\overline{A} \cdot \overline{C}$	X	X				
$I1$	$\overline{B} \cdot \overline{C}$	X			X		
$I2$	$\overline{A} \cdot B$		X	X			
$I3$	$A \cdot \overline{B}$				X	X	
$I4$	$B \cdot C$			X			X
$I5$	$A \cdot C$					X	X

Note that each row has been labelled $I[0..9]$. The next step may seem unfamiliar to those readers with an engineering background, but is reminisent of the origins of Boolean algebra,which is to study logic in a more philosophical context!

The objective is to choose a permutation of simplified terms $I[0..9]$ (known as Implicants) such that alll minterms $\{m0, m2, m3, m4, m5, m7\}$are covered. This can be expressed in terms of Boolean logic:

- The variable $M0$ shall be deemed TRUE if $m0$ is included, that is $I0$ OR $I1$ are selected. This can be written as $M0 = I0 + I1$
  - Similar rules can be written for all remaining minterms $m2, m3, m4, m5, m7$.
- The variable Y is TRUE when $M0$ AND $M2$ AND $M3$ AND $M4$ AND $M5$ AND $M5$ are TRUE

This can be written more consisely as:

$$Y = M0 \cdot M2 \cdot M3 \cdot M4 \cdot M5 \cdot M7$$

Combining both rules:

$$Y = (I0 + I1) \cdot (I0 + I2) \cdot (I2 + I4) \cdot (I1 + I3) \cdot (I3 + I5) \cdot (I4 + I5) \tag{1.3}$$

To meet the criteria that all minterms must be included, then expression for Y must be TRUE. However, the expression for Y may also be simplified using the indentity in equation (1.2).

Algebraic Simplification

The algebraic simplification of (1.3) requires the repeated and systematic application of the identity (1.2).

Consider the first product of sums $(I0+I1)\cdot(I0+I2) = I0+I0\cdot I2+I0\cdot I1+I1\cdot I2$. Two applications of the identity (1.2) reduces this expression as follows:

$$\begin{aligned} I0 + I0 \cdot I2 + I0 \cdot I1 + I1 \cdot I2 &= I0 \cdot (1 + I2) + I0 \cdot I1 + I1 \cdot I2 \\ &= I0 + I0 \cdot I1 + I1 \cdot I2 \\ &= I0 \cdot (1 + I1) + I1 \cdot I2 \\ &= I0 + I1 \cdot I2 \end{aligned}$$

Now multiply this with the next sum

$$\begin{aligned} (I0 + I1 \cdot I2) \cdot (I2 + I4) &= I0 \cdot I2 + I0 \cdot I4 + I1 \cdot I2 + I1 \cdot I2 \cdot I4 \\ &= I0 \cdot I2 + I0 \cdot I4 + I1 \cdot I2 \cdot (1 + I4) \\ &= I0 \cdot I2 + I0 \cdot I4 + I1 \cdot I2 \end{aligned}$$

This process is continued until all terms in (1.3) have been included and all simplification has been applied.

$$
\begin{aligned}
\left(\begin{array}{c} I0 \cdot I2 + I0 \cdot I4 + \\ I1 \cdot I2 \end{array}\right) \cdot (I1 + I3) &= \begin{array}{c} I0 \cdot I2 \cdot I3 + I0 \cdot I1 \cdot I4 + \\ I0 \cdot I3 \cdot I4 + I1 \cdot I2 \end{array} \\
\left(\begin{array}{c} I0 \cdot I2 \cdot I3 + I0 \cdot I1 \cdot I4 + \\ I0 \cdot I3 \cdot I4 + I1 \cdot I2 \end{array}\right) (I3 + I5) &= \begin{array}{c} I0 \cdot I2 \cdot I3 + I0 \cdot I1 \cdot I4 \cdot I5 + \\ I0 \cdot I3 \cdot I4 + I1 \cdot I2 \cdot I3 + \\ I1 \cdot I2 \cdot I5 \end{array} \\
\left(\begin{array}{c} I0 \cdot I2 \cdot I3 + I0 \cdot I1 \cdot I4 \cdot I5 + \\ I0 \cdot I3 \cdot I4 + I1 \cdot I2 \cdot I3 + \\ I1 \cdot I2 \cdot I5 \end{array}\right) \cdot (I4 + I5) &= \begin{array}{c} I0 \cdot I2 \cdot I3 \cdot I5 + I0 \cdot I1 \cdot I4 \cdot I5 + \\ I0 \cdot I3 \cdot I4 + I1 \cdot I2 \cdot I3 \cdot I4 + \\ I1 \cdot I2 \cdot I5 \end{array}
\end{aligned}
$$

The final result is the following:

$$
\left[\begin{array}{c} I0 \cdot I2 \cdot I3 \cdot I5 \\ I0 \cdot I1 \cdot I4 \cdot I5 \\ I1 \cdot I2 \cdot I3 \cdot I4 \\ I0 \cdot I3 \cdot I4 \\ I1 \cdot I2 \cdot I5 \end{array}\right] \tag{1.4}
$$

Referring to table 1.2 and the result in (1.4), there are 5 possible combinations which cover all minterms. There are two solutions with only three implicants, which are $I0 \cdot I3 \cdot I4$ and $I1 \cdot I2 \cdot I5$, both of which have equal merit (so either can be used). Taking the first as the selected solution, this is interpreted as:

> "To cover the set of minterms, the designer needs to select $I0$, $I3$ and $I4$, therefore $Y = \overline{A} \cdot \overline{C} + A \cdot \overline{B} + B \cdot C$"

This is one of the solutions from the KM in table 1.1(lower).

Tabular method

The algebraic application of Petrick's method is cumbersome and error prone and best perormed by a computer. However, it can also be performed using a tabular approach. Returning to equation (1.3)

$$Y = (I0 + I1) \cdot (I0 + I2) \cdot (I2 + I4) \cdot (I1 + I3) \cdot (I3 + I5) \cdot (I4 + I5)$$

Again, consider the first product of sums

$$
\begin{aligned}
(I0 + I1) \cdot (I0 + I2) &= I0 + I0 \cdot I2 + I0 \cdot I1 + I1 \cdot I2 \\
&= I0 \cdot (1 + I2) + I0 \cdot I1 + I1 \cdot I2 \\
&= I0 \cdot (1 + I1) + I1 \cdot I2 \\
&= I0 + I1 \cdot I2
\end{aligned}
$$

This expression includes the first two columns of the implicant table shown below where each implicant I_x, $x = 0 \ldots 5$, is represented in the same way as used in the QM method.

	binary	$m0$	$m2$	$m3$	$m4$	$m5$	$m7$
$I0$	$-----1$	X	X				
$I1$	$----1-$	X			X		
$I2$	$---1--$		X	X			
$I3$	$--1---$				X	X	
$I4$	$-1----$			X			X
$I5$	$1-----$					X	X

Remember that rows in the same column are combined with an 'OR' function and all columns are combined with an AND. The product of the first two columns $(I0+I1)\cdot(I0+I2) = I0+I0\cdot I2+I0\cdot I1+I1\cdot I2$ which simplifies to $I0+I1\cdot I2$

Now consider these terms using the same binary representation as used in the QM method:

$$\begin{aligned} I0 &= -----1 \\ I1 &= ----1- \\ I2 &= ---1-- \end{aligned}$$

Similarly, using the same notation as the QM method:

$$\begin{aligned} I0\cdot I1 &= ----11 \\ I0\cdot I2 &= ---1-1 \\ I1\cdot I2 &= ----11 \end{aligned}$$

When terms or combined with an OR function, they are simply listed as rows in a vector:

$$\begin{aligned} M0 &= (I0+I1) = \begin{bmatrix} -----1 \\ ----1- \end{bmatrix} \\ M2 &= (I0+I2) = \begin{bmatrix} -----1 \\ ---1-- \end{bmatrix} \end{aligned}$$

And by combining with the AND function

$$M0\cdot M1 = \begin{bmatrix} -----1 \\ ---1-1 \\ ----11 \\ ---11- \end{bmatrix} = \begin{bmatrix} I0 \\ I0\cdot I2 \\ I0\cdot I1 \\ I1\cdot I2 \end{bmatrix} \tag{1.5}$$

Consider the first two rows of this result alongside the Boolean algebra

$$\begin{array}{cc} \text{Binary} & \text{Algebraic} \\ \begin{bmatrix} -----1 \\ ---1-1 \end{bmatrix} & I0 + I0 \cdot I2 = I0 \cdot (1 + I2) \\ \hline [-----1] & I0 \end{array}$$

Here the identity $X + XY \equiv X$ has been applied for both representations. Using the binary representation, this can be recognised using the following rules:

- Where two rows X and Y are equal **except** that Y has one or more **additional** term(s), then delete Y
- Where two rows X and Y are identical, delete Y (or X)

Returning to the result in (1.5), application of these rules can be applied to simplify the expression for $M0 \cdot M1$

$M0 \cdot M2$	first simplification	second simplification
-----1	-----1	-----1
---1-1	~~---1-1~~	---11-
----11	----11	
---11-	---11-	

This can be extended to the whole expression in (1.3).

$$Y = (I0 + I1) \cdot (I0 + I2) \cdot (I2 + I4) \cdot (I1 + I3) \cdot (I3 + I5) \cdot (I4 + I5)$$

$M0$	$M2$	$M0 \cdot M2$	$M3$	$M0 \cdot M2 \cdot M3$	$M4$	$M0 \cdot M2 \cdot M3 \cdot M4$
-----1	-----1	-----1	---1--	---1-1	----1-	--11-1
----1-	---1--	---11-	-1----	-1---1	--1---	-1--11
				---11-		-11--1
						---11-

$M0 \cdot M2 \cdot M3 \cdot M4$	$M5$	$M0 \cdot M2 \cdot M3 \cdot M4 \cdot M5$	$M7$	$M0 \cdot M2 \cdot M3 \cdot M4 \cdot M5 \cdot M7$
--11-1	--1---	--11-1	-1----	1-11-1
-1--11	1-----	11--11	1-----	11--11
-11--1		-11--1		-11--1
---11-		--111-		-1111-
		1--11-		1--11-