School of Engineering and Computer Science

Mathematics Applied to Digital Engineering-2

(Minimisation of Boolean Functions)

Kamel ATTAR
kamel.attar@epita.fr

Week **#2 #3** ✦ 15FEB-01MAR/2024 ✦

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

★ What is minimisation? ★

- Mathematics expressions are simplified for a number of reasons, for instance simpler expression are easier to understand and easier to write down, they are also less prone to error in interpretation but, most importantly, simplified expressions are usually more efficient and effective when implemented in practice.

★ What is minimisation? ★

- Mathematics expressions are simplified for a number of reasons, for instance simpler expression are easier to understand and easier to write down, they are also less prone to error in interpretation but, most importantly, simplified expressions are usually more efficient and effective when implemented in practice.
- A Boolean expression is composed of variables and terms. The simplification of Boolean expressions can lead to more effective computer programs, algorithms and circuits.

★ What is minimisation? ★

- Mathematics expressions are simplified for a number of reasons, for instance simpler expression are easier to understand and easier to write down, they are also less prone to error in interpretation but, most importantly, simplified expressions are usually more efficient and effective when implemented in practice.
- A Boolean expression is composed of variables and terms. The simplification of Boolean expressions can lead to more effective computer programs, algorithms and circuits.

  Minimisation can be achieved by a number of methods, four well known methods are:

  1. Algebraic Manipulation of Boolean Expressions
  2. Karnaugh Maps
  3. Tabular Method of Minimisation
  4. Tree reduction

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

★ Algebraic Manipulation of Boolean Expressions ★

- This is an approach where you can transform one boolean expression into an equivalent expression by applying **Boolean Theorems**.

★ Algebraic Manipulation of Boolean Expressions ★

- This is an approach where you can transform one boolean expression into an equivalent expression by applying **Boolean Theorems**.
- Minimising terms and expressions can be important because electrical circuits consist of individual components that are implemented for each term or literal for a given expression. This allows designers to make use of fewer components, thus reducing the cost of a particular system.

★ Algebraic Manipulation of Boolean Expressions ★

- This is an approach where you can transform one boolean expression into an equivalent expression by applying **Boolean Theorems**.
- Minimising terms and expressions can be important because electrical circuits consist of individual components that are implemented for each term or literal for a given expression. This allows designers to make use of fewer components, thus reducing the cost of a particular system.
- It should be noted that there are no fixed rules that can be used to minimise a given expression. It is left to an individuals ability to apply Boolean Theorems in order to minimise a function.

★ Algebraic Manipulation of Boolean Expressions ★

- This is an approach where you can transform one boolean expression into an equivalent expression by applying **Boolean Theorems**.
- Minimising terms and expressions can be important because electrical circuits consist of individual components that are implemented for each term or literal for a given expression. This allows designers to make use of fewer components, thus reducing the cost of a particular system.
- It should be noted that there are no fixed rules that can be used to minimise a given expression. It is left to an individuals ability to apply Boolean Theorems in order to minimise a function.

Example

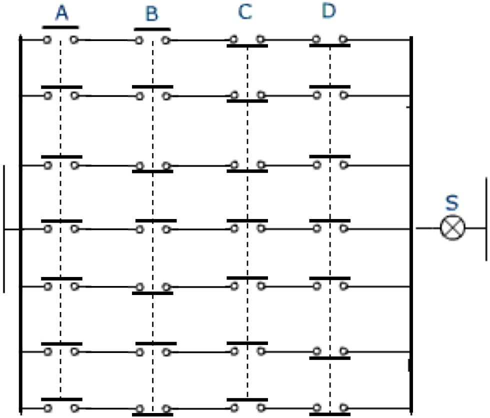
$$(A+\overline{B}+\overline{C})(A+\overline{B}C) = AA + A\overline{B}C + \overline{B}A + \overline{B}\,\overline{B}C + \overline{C}A + \overline{C}\,\overline{B}C = A(1+\overline{B}C+\overline{B}+\overline{C}) + \overline{B}C = A + \overline{B}C$$

✎ **Exercise 4.** Minimise the following functions using algebraic method:

a. $f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B + AB\overline{C} + AC$ b. $f(A,B,C) = \overline{A}B + B\overline{C} + BC + A\overline{B}\,\overline{C}$

★ Algebraic Manipulation of Boolean Expressions ★

✎ **Exercise 5.** Minimise the following switch circuit:



1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

★ Writing a Logic Equation from a Truth Table ★

Any logic problem can be solved using the following step-by-step procedure.

1. Interpret the problem and set up a truth table to describe its operation.

★ Writing a Logic Equation from a Truth Table ★

Any logic problem can be solved using the following step-by-step procedure.

1. Interpret the problem and set up a truth table to describe its operation.
2. Write the AND (product) term for each case where the output is 1.

★ Writing a Logic Equation from a Truth Table ★

Any logic problem can be solved using the following step-by-step procedure.

1. Interpret the problem and set up a truth table to describe its operation.
2. Write the AND (product) term for each case where the output is 1.
3. Write the sum-of-products (SOP) expression for the output.

This is called the **disjunctive normal form** of the boolean function ***f***

★ Writing a Logic Equation from a Truth Table ★

Any logic problem can be solved using the following step-by-step procedure.

1. Interpret the problem and set up a truth table to describe its operation.
2. Write the AND (product) term for each case where the output is 1.
3. Write the sum-of-products (SOP) expression for the output.
4. Simplify the output expression if possible.

★ Writing a Logic Equation from a Truth Table ★

Any logic problem can be solved using the following step-by-step procedure.

1. Interpret the problem and set up a truth table to describe its operation.
2. Write the AND (product) term for each case where the output is 1.
3. Write the sum-of-products (SOP) expression for the output.
4. Simplify the output expression if possible.
5. Implement the circuit for the final, simplified expression.

★ Writing a Logic Equation from a Truth Table ★

Any logic problem can be solved using the following step-by-step procedure.

1. Interpret the problem and set up a truth table to describe its operation.
2. Write the AND (product) term for each case where the output is 1.
3. Write the sum-of-products (SOP) expression for the output.
4. Simplify the output expression if possible.
5. Implement the circuit for the final, simplified expression.

Example

Design a logic circuit that has three inputs, ***A***, ***B***, and ***C***, and whose output will be **HIGH** only when a majority of the inputs are **HIGH**.

❶ **Set up the truth table.**
On the basis of the problem statement, the output x should be **1** whenever two or more inputs are **1**; for all other cases, the output should be **0**.

A	B	C	$f(A,B,C)$	minterms
0	0	0	0	$m0$
0	0	1	0	$m1$
0	1	0	0	$m2$
0	1	1	1	$m3$
1	0	0	0	$m4$
1	0	1	1	$m5$
1	1	0	1	$m6$
1	1	1	1	$m7$

1. **Set up the truth table.**
   On the basis of the problem statement, the output $\boldsymbol{x}$ should be **1** whenever two or more inputs are **1**; for all other cases, the output should be **0**.

2. **Write the AND term for each case where the output is a 1.**
   There are four such cases. The AND terms are shown next to the truth table. Again note that each AND term contains each input variable in either inverted or noninverted form.

A	B	C	$f(A,B,C)$	minterms
0	0	0	0	$m0$
0	0	1	0	$m1$
0	1	0	0	$m2$
0	1	1	1	$m3$
1	0	0	0	$m4$
1	0	1	1	$m5$
1	1	0	1	$m6$
1	1	1	1	$m7$

1. **Set up the truth table.**
   On the basis of the problem statement, the output x should be **1** whenever two or more inputs are **1**; for all other cases, the output should be **0**.

2. **Write the AND term for each case where the output is a 1.**
   There are four such cases. The AND terms are shown next to the truth table. Again note that each AND term contains each input variable in either inverted or noninverted form.

A	B	C	$f(A,B,C)$	minterms
0	0	0	0	$m0$
0	0	1	0	$m1$
0	1	0	0	$m2$
0	1	1	1	$m3$
1	0	0	0	$m4$
1	0	1	1	$m5$
1	1	0	1	$m6$
1	1	1	1	$m7$

3. **Write the sum-of-products expression for the output.**

$$
\begin{aligned}
f(A,B,C) &= \sum m(3,5,6,7)\\
&= m3 + m5 + m6 + m7\\
&= 011 + 101 + 110 + 111\\
&= \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC
\end{aligned}
$$

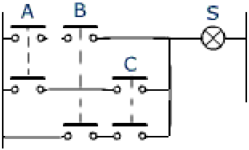
011	is represented by	$\overline{A}BC$
101	is represented by	$A\overline{B}C$
110	is represented by	$AB\overline{C}$
111	is represented by	ABC

4 Simplify the output expression.

This expression can be simplified in several ways. Perhaps the quickest way is to realize that the last term ***ABC*** has two variables in common with each of the other terms. Thus, we can use the ***ABC*** term to pair with each of the other terms:

$$\begin{aligned} f(A,B,C) &= ABC + AB\overline{C} + ABC + A\overline{B}C + ABC + \overline{A}BC \\ &= AB(C+\overline{C}) + AC(B+\overline{B}) + BC(A+\overline{A}) \\ &= AB + AC + BC\,. \end{aligned}$$

5 Implement the circuit for the final expression



✎ **Exercise 6.** Find the disjunctive normal form of the Boolean function for these truth tables:

a.

A	B	$f(A,B)$
0	0	1
0	1	0
1	0	1
1	1	0

b.

A	B	$f(A,B)$
0	0	1
0	1	1
1	0	0
1	1	1

c.

A	B	C	$f(A,B,C)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

d.

A	B	C	$f(A,B,C)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

e.

A	B	C	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

✎ **Exercise 7.** Design a logic circuit that has three inputs, A, B, and C, and whose output will be **HIGH** only when a majority of the inputs are **LOW**.

✎ **Exercise 8.** Find the sum-of-products expansions of the Boolean function $F(x, y, z)$ that equals 1 if and only if a. $x = 0$ b. $xy = 0$ c. $x + y = 0$

✎ **Exercise 9.** Minimize the following boolean function
a. $F(A, B, C) = \sum m(0, 1, 6, 7)$ b. $F(A, B, C) = \sum m(1, 2, 5, 7)$

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
Introduction
Rules of Simplification
Exercises

4 Tabular Method of Minimisation
Rules of Tabular Method
Determination of Prime Implicants
Prime Implicant Chart

- So far we can see that applying Boolean algebra can be awkward in order to simplify expressions.
- Apart from being laborious (and requiring the remembering all the laws) the method can lead to solutions which, though they appear minimal, are not.
- The **Karnaugh** map provides a simple and straight-forward method of minimising boolean expressions.
- With the **Karnaugh** map Boolean expressions having up to four and even six variables can be simplified.
- A **Karnaugh** map provides a pictorial method of grouping together expressions with common factors and therefore eliminating unwanted variables.
- The **Karnaugh** map can also be described as a special arrangement of a truth table.

- The diagram below illustrates the correspondence between the **Karnaugh** map and the truth table for the general case of a two variable problem.

A	B	F
0	0	a
0	1	b
1	0	c
1	1	d

$\Longrightarrow$

		B	
		0	**1**
A	**0**	***a***	***b***
	1	***c***	***d***

- The diagram below illustrates the correspondence between the **Karnaugh** map and the truth table for the general case of a two variable problem.

A	B	F
0	0	a
0	1	b
1	0	c
1	1	d

$\Longrightarrow$

A \ **B**	**0**	**1**
0	***a***	***b***
1	***c***	***d***

- The values inside the squares are copied from the output column of the truth table, therefore there is one square in the map for every row in the truth table. Around the edge of the Karnaugh map are the values of the two input variable. B is along the top and A is down the left hand side. The diagram below explains this:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

$\Longrightarrow$

A \ B	0	1
0	0	1
1	1	1

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $\boldsymbol{f(A,B) = A\overline{B} + AB}$

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $\boldsymbol{f(A, B) = A\overline{B} + AB}$

- Note that values of the input variables form the rows and columns. That is the logic values of the variables ***A*** and ***B*** (with one denoting true form and zero denoting false form) form the head of the rows and columns respectively.

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $f(A, B) = A\overline{B} + AB$

- Note that values of the input variables form the rows and columns. That is the logic values of the variables ***A*** and ***B*** (with one denoting true form and zero denoting false form) form the head of the rows and columns respectively.
- Bear in mind that the above map is a one dimensional type which can be used to simplify an expression in two variables.

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $f(A, B) = A\overline{B} + AB$

- Note that values of the input variables form the rows and columns. That is the logic values of the variables ***A*** and ***B*** (with one denoting true form and zero denoting false form) form the head of the rows and columns respectively.

- Bear in mind that the above map is a one dimensional type which can be used to simplify an expression in two variables.

- There is a two-dimensional map that can be used for up to four variables, and a three-dimensional map for up to six variables.

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $f(A, B) = A\overline{B} + AB$

- Note that values of the input variables form the rows and columns. That is the logic values of the variables A and B (with one denoting true form and zero denoting false form) form the head of the rows and columns respectively.

- Bear in mind that the above map is a one dimensional type which can be used to simplify an expression in two variables.

- There is a two-dimensional map that can be used for up to four variables, and a three-dimensional map for up to six variables.

Using algebraic simplification $f(A, B) = A\overline{B} + AB = A(\overline{B} + B) = A$.

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $f(A, B) = A\overline{B} + AB$

- Going back to the above map, the two adjacent **1**'s are grouped together.

A \ B	0	1
0	0	0
1	1	1

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $\boldsymbol{f(A, B) = A\overline{B} + AB}$

- Going back to the above map, the two adjacent **1**'s are grouped together.
- Through inspection it can be seen that variable ***B*** has its true and false form within the group.

A \ B	0	1
0	0	0
1	1	1

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $\boldsymbol{f(A, B) = A\overline{B} + AB}$

- Going back to the above map, the two adjacent **1**'s are grouped together.
- Through inspection it can be seen that variable ***B*** has its true and false form within the group.
- This eliminates variable ***B*** leaving only variable ***A*** which only has its true form.

A \ B	0	1
0	0	0
1	1	1

Consider the following map

A \ B	0	1
0	0	0
1	1	1

The function plotted is: $f(A, B) = A\overline{B} + AB$

- Going back to the above map, the two adjacent **1**'s are grouped together.
- Through inspection it can be seen that variable B has its true and false form within the group.
- This eliminates variable B leaving only variable A which only has its true form.

A \ B	0	1
0	0	0
1	1	1

The minimised answer therefore is $f(A, B) = A$.

Example

Consider $\boldsymbol{f(A, B) = \overline{A}\,\overline{B} + A\overline{B} + \overline{A}B}$ plotted on the beside Karnaugh map. Pairs of **1**'s are grouped as shown in the table, and the simplified answer is obtained by using the following steps:

A \ B	0	1
0	1	1
1	1	0

1. Note that two groups can be formed for the example given above, bearing in mind that the largest rectangular clusters that can be made consist of two **1**s. Notice that a **1** can belong to more than one group.

Example

Consider $\boldsymbol{f(A, B) = \overline{A}\,\overline{B} + A\overline{B} + \overline{A}B}$ plotted on the beside Karnaugh map. Pairs of **1**'s are grouped as shown in the table, and the simplified answer is obtained by using the following steps:

A \ B	0	1
0	1	1
1	1	0

1. Note that two groups can be formed for the example given above, bearing in mind that the largest rectangular clusters that can be made consist of two **1**s. Notice that a **1** can belong to more than one group.
2. The first group colored red, consists of two **1**s which correspond to $\boldsymbol{(A = 0, B = 0)}$ and $\boldsymbol{(A = 1, B = 0)}$. Put in another way, all squares in this example that correspond to the area of the map where $\boldsymbol{B = 0}$ contains **1**s, independent of the value of $\boldsymbol{A}$. So when $\boldsymbol{B = 0}$ the output is **1**. The expression of the output will contain the term

Example

Consider $\boldsymbol{f(A, B) = \overline{A}\,\overline{B} + A\overline{B} + \overline{A}B}$ plotted on the beside Karnaugh map. Pairs of **1**'s are grouped as shown in the table, and the simplified answer is obtained by using the following steps:

A \ B	0	1
0	1	1
1	1	0

1. Note that two groups can be formed for the example given above, bearing in mind that the largest rectangular clusters that can be made consist of two **1**s. Notice that a **1** can belong to more than one group.
2. The first group colored red, consists of two **1**s which correspond to **(*A* = 0, *B* = 0)** and **(*A* = 1, *B* = 0)**. Put in another way, all squares in this example that correspond to the area of the map where ***B* = 0** contains **1**s, independent of the value of ***A***. So when ***B* = 0** the output is **1**. The expression of the output will contain the term
3. For group colored green corresponds to the area of the map where ***A* = 0**. The group can therefore be defined as . This implies that when ***A* = 0** the output is **1**. The output is therefore **1** whenever ***B* = 0** and ***A* = 0**

Example

Consider $\boldsymbol{f(A,B) = \overline{A}\,\overline{B} + A\overline{B} + \overline{A}B}$ plotted on the beside Karnaugh map. Pairs of **1**'s are grouped as shown in the table, and the simplified answer is obtained by using the following steps:

A \ B	0	1
0	1	1
1	1	0

1. Note that two groups can be formed for the example given above, bearing in mind that the largest rectangular clusters that can be made consist of two **1**s. Notice that a **1** can belong to more than one group.
2. The first group colored red, consists of two **1**s which correspond to **($\boldsymbol{A}$ = 0, $\boldsymbol{B}$ = 0)** and **($\boldsymbol{A}$ = 1, $\boldsymbol{B}$ = 0)**. Put in another way, all squares in this example that correspond to the area of the map where $\boldsymbol{B}$ **= 0** contains **1**s, independent of the value of $\boldsymbol{A}$. So when $\boldsymbol{B}$ **= 0** the output is **1**. The expression of the output will contain the term
3. For group colored green corresponds to the area of the map where $\boldsymbol{A}$ **= 0**. The group can therefore be defined as . This implies that when $\boldsymbol{A}$ **= 0** the output is **1**. The output is therefore **1** whenever $\boldsymbol{B}$ **= 0** and $\boldsymbol{A}$ **= 0**

Hence the simplified answer is $\boldsymbol{f(A,B) = \overline{A} + \overline{B}}$

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

★ Karnaugh Maps ★

Rules of Simplification

The Karnaugh map uses the following rules for the simplification of expressions by grouping together adjacent cells containing ones

★ Karnaugh Maps ★

Rules of Simplification

The Karnaugh map uses the following rules for the simplification of expressions by grouping together adjacent cells containing ones

1. Groups may not include any cell containing a zero

A \ B	0	1
0	0	
1	1	1

Wrong

A \ B	0	1
0	0	
1	1	1

Right

★ Karnaugh Maps ★

Rules of Simplification

The Karnaugh map uses the following rules for the simplification of expressions by grouping together adjacent cells containing ones

❶ Groups may not include any cell containing a zero

A \ B	0	1
0	0	
1	1	1

Wrong

A \ B	0	1
0	0	
1	1	1

Right

❷ Groups may be horizontal or vertical, but not diagonal.

A \ B	0	1
0	0	1
1	1	0

Wrong

A \ B	0	1
0	0	1
1	1	1

Right

❸ Groups must contain **1**, **2**, **4**, **8**, or in general $\mathbf{2^n}$ cells. That is if $\boldsymbol{n} = \mathbf{1}$, a group will contain two **1**'s since $\mathbf{2^1 = 2}$. If $\boldsymbol{n} = \mathbf{2}$, a group will contain four **1**'s since $\mathbf{2^2 = 4}$.

A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	0	0

Group of **3** (Wrong)

A \ B	0	1
0	1	1
1	0	0

Group of **2**

A \ B	0	1
0	1	1
1	1	1

Group of **4**

A \ BC	00	01	11	10
0	1	1	1	1
1	0	0	0	1

Group of **5** (Wrong)

❸ Groups must contain **1**, **2**, **4**, **8**, or in general $\mathbf{2^n}$ cells. That is if $\boldsymbol{n} = \mathbf{1}$, a group will contain two **1**'s since $\mathbf{2^1 = 2}$. If $\boldsymbol{n} = \mathbf{2}$, a group will contain four **1**'s since $\mathbf{2^2 = 4}$.

A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	0	0

Group of **3** (Wrong)

A \ B	0	1
0	1	1
1	0	0

Group of **2**

A \ B	0	1
0	1	1
1	1	1

Group of **4**

A \ BC	00	01	11	10
0	1	1	1	1
1	0	0	0	1

Group of **5** (Wrong)

❹ Each group should be as large as possible.

A \ BC	00	01	11	10
0	1	1	1	1
1	0	0	1	1

Wrong

A \ BC	00	01	11	10
0	1	1	1	1
1	0	0	1	1

Right

❺ Groups may wrap around the table. The leftmost cell in a row may be grouped with the rightmost cell and the top cell in a column may be grouped with the bottom cell.

AB \ CD	00	01	11	10
00	1	0	1	1
01	1	0	0	1
11	1	0	0	1
10	1	0	1	1

❺ Groups may wrap around the table. The leftmost cell in a row may be grouped with the rightmost cell and the top cell in a column may be grouped with the bottom cell.

AB \ CD	00	01	11	10
00	1	0	1	1
01	1	0	0	1
11	1	0	0	1
10	1	0	1	1

❻ There should be as few groups as possible, as long as this does not contradict any of the previous rules.

A \ BC	00	01	11	10
0	1	1	1	1
1	0	0	1	1

Wrong

A \ BC	00	01	11	10
0	1	1	1	1
1	0	0	1	1

Right

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

★ Karnaugh Maps ★

✎ **Exercise 10.** Minimise the following problems using the Karnaugh maps method.

a. $f(A, B, C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B + AB\overline{C} + AC$
b. $f(A, B, C) = \overline{A}B + B\overline{C} + BC + A\,\overline{B}\,\overline{C}$
c. $f(A, B, C) = \sum m(0, 1, 6, 7)$
d. $f(A, B, C) = \sum m(0, 1, 4, 5, 6, 7)$
e. $f(A, B, C, D) = \sum m(0, 2, 8, 10, 14)$
f. $f(A, B, C, D) = \sum m(3, 4, 5, 7, 9, 13, 14, 15)$
g. $f(A, B, C, D) = \sum m(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$
h. $f(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 6, 8, 9, 11, 13, 15)$

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

★ Tabular Method of Minimisation ★

Introduction

- The tabular method which is also known as the *Quine-McCluskey* method is particularly useful when minimizing functions having a large number of variables.

★ Tabular Method of Minimisation ★

Introduction

- The tabular method which is also known as the *Quine-McCluskey* method is particularly useful when minimizing functions having a large number of variables.

- The method reduces a function in standard **sum of products** form to a set of **prime implicants** from which as many variables are eliminated as possible.

★ Tabular Method of Minimisation ★

Introduction

- The tabular method which is also known as the *Quine-McCluskey* method is particularly useful when minimizing functions having a large number of variables.
- The method reduces a function in standard **sum of products** form to a set of **prime implicants** from which as many variables are eliminated as possible.
- These prime implicants are then examined to see if some are redundant. We use a prime implicant chart to select a minimum set of prime implicants.

★ Tabular Method of Minimisation ★

Introduction

- The tabular method which is also known as the *Quine-McCluskey* method is particularly useful when minimizing functions having a large number of variables.
- The method reduces a function in standard **sum of products** form to a set of **prime implicants** from which as many variables are eliminated as possible.
- These prime implicants are then examined to see if some are redundant. We use a prime implicant chart to select a minimum set of prime implicants.
- The tabular method makes repeated use of the law $\boldsymbol{A + \overline{A} = 1}$. Note that Binary notation is used for the function, although decimal notation is also used for the functions. As usual a variable in true form is denoted by **1**, in inverted form by **0**, and the absence of a variable by a dash **(−)**.

★ Tabular Method of Minimisation ★

Introduction

- The tabular method which is also known as the *Quine-McCluskey* method is particularly useful when minimizing functions having a large number of variables.
- The method reduces a function in standard **sum of products** form to a set of **prime implicants** from which as many variables are eliminated as possible.
- These prime implicants are then examined to see if some are redundant. We use a prime implicant chart to select a minimum set of prime implicants.
- The tabular method makes repeated use of the law $\boldsymbol{A + \overline{A} = 1}$. Note that Binary notation is used for the function, although decimal notation is also used for the functions. As usual a variable in true form is denoted by **1**, in inverted form by **0**, and the absence of a variable by a dash **(–)**.

Computer programs have been developed employing this algorithm.

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
Introduction
Rules of Simplification
Exercises

4 Tabular Method of Minimisation
Rules of Tabular Method
Determination of Prime Implicants
Prime Implicant Chart

★ Tabular Method of Minimisation ★

Rules of Tabular Method

① Consider a function of three variables $f(A, B, C)$:

$\overline{A}BC$	is represented by	**011**
$A\overline{B}\,\overline{C}$	is represented by	**100**
$A\overline{C}$	is represented by	**1 – 0**
BC	is represented by	**–11**

★ Tabular Method of Minimisation ★

Rules of Tabular Method

① Consider a function of three variables $f(A, B, C)$:

$\overline{A}BC$	is represented by	**011**
$A\overline{B}\,\overline{C}$	is represented by	**100**
$A\overline{C}$	is represented by	**1 — 0**
BC	is represented by	**—11**

② **Eliminate literals**: Two terms can be combined if they differ in exactly one variable

- Consider the function:
  $f(A, B, C, D) = \sum(m_{14} + m_{15}) = 1110 + 1111 = ABC\overline{D} + ABCD = ABC(\overline{D} + D) = ABC$

  Listing the two **minterms** shows they can be combined

A	B	C	D	
1	1	1	0	$ABC\overline{D}$
1	1	1	1	$ABCD$
1	1	1	—	ABC

Can combine (Differs in one digit position)

★ Tabular Method of Minimisation ★

Rules of Tabular Method

② **Eliminate literals**: Two terms can be combined if they differ in exactly one variable

- Consider the function:
  $\boldsymbol{f(A,B,C,D) = \sum(m_{14} + m_{15}) = 1110 + 1111 = ABC\overline{D} + ABCD = ABC(\overline{D} + D) = ABC}$
  Listing the two **minterms** shows they can be combined

A	B	C	D		
1	1	1	0	$ABC\overline{D}$	Can combine (Differs in one digit position)
1	1	1	1	$ABCD$	
1	1	1	–	ABC	

- Now we consider the following: $\boldsymbol{f(A,B,C,D) = 1101 + 1110 = AB\overline{C}D + ABC\overline{D}}$
  Note that these variables cannot be combined

A	B	C	D		
1	1	0	1	$AB\overline{C}D$	Cannot combine (Differs in more than one digit position)
1	1	1	0	$ABC\overline{D}$	
1	1	?	?		

This is because the *SECOND RULE* of the Tabular method for two terms to combine, and thus eliminate one variable, is that they must differ in only **one digit** position.

★ Tabular Method of Minimisation ★

Rules of Tabular Method

③ Sort into groups according to the number of **1**'s

- Bear in mind that when two terms are combined, one of the combined terms has one digit more at logic **1** than the other combined term.
- This indicates that the number of **1**'s in a term is significant and is referred to as its index

0000	$\Longrightarrow$	Index **0**
0010 10000	$\Longrightarrow$	Index **1**
1010 1100 1001	$\Longrightarrow$	Index **2**
1110 1011	$\Longrightarrow$	Index **3**
1111	$\Longrightarrow$	Index **4**

The necessary condition for combining two terms is that the indices of the two terms must differ by one logic variable which must also be the same.

★ Tabular Method of Minimisation ★

Rules of Tabular Method

Example

Consider the function:

$$f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C$$

To make things easier, change the function into binary notation with index value and decimal value.

$$f(A,B,C) = m_0 + m_1 + m_4 + m_5 = \underbrace{000}_{\text{Index } \mathbf{0}} + \underbrace{001 + 100}_{\text{Index } \mathbf{1}} + \underbrace{101}_{\text{Index } \mathbf{2}}.$$

Tabulate the index groups in a column and insert the decimal value alongside.

First List

		A	B	C
Index **0**	m_0	0	0	0
Index **1**	m_1	0	0	1
	m_4	1	0	0
Index **2**	m_5	1	0	1

- From the first list, we combine terms that differ by **1** digit only from one index group to the next. These terms from the first list are then seperated into groups in the second list.
- Note that the ticks are just there to show that one term has been combined with another term.

★ Tabular Method of Minimisation ★

Rules of Tabular Method

Example

Consider the function:

$$f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C$$

To make things easier, change the function into binary notation with index value and decimal value.

$$f(A,B,C) = m_0 + m_1 + m_4 + m_5 = \underbrace{000}_{\text{Index } \mathbf{0}} + \underbrace{001 + 100}_{\text{Index } \mathbf{1}} + \underbrace{101}_{\text{Index } \mathbf{2}}.$$

Tabulate the index groups in a column and insert the decimal value alongside.

First List

		A	B	C	
Index **0**	m_0	0	0	0	✓
Index **1**	m_1	0	0	1	✓
	m_4	1	0	0	✓
Index **2**	m_5	1	0	1	✓

$\Longrightarrow$

Second List

	A	B	C
$m_0 + m_1$	0	0	–
$m_0 + m_4$	–	0	0
$m_1 + m_5$	–	0	1
$m_4 + m_5$	1	0	–

★ Tabular Method of Minimisation ★

Rules of Tabular Method

Example

Consider the function:

$$f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C$$

To make things easier, change the function into binary notation with index value and decimal value.

$$f(A,B,C) = m_0 + m_1 + m_4 + m_5 = \underbrace{000}_{\text{Index } \mathbf{0}} + \underbrace{001 + 100}_{\text{Index } \mathbf{1}} + \underbrace{101}_{\text{Index } \mathbf{2}}.$$

Tabulate the index groups in a column and insert the decimal value alongside.

Second List

	A	B	C
$m_0 + m_1$	0	0	–
$m_0 + m_4$	–	0	0
$m_1 + m_5$	–	0	1
$m_4 + m_5$	1	0	–

- From the second list we can see that the expression is now reduced to: $Z = \overline{A}\,\overline{B} + \overline{B}\,\overline{C} + \overline{B}C + A\overline{B}$
- Bear in mind that the dash indicates a missing variable and must line up in order to get a third list.

★ Tabular Method of Minimisation ★

Rules of Tabular Method

Example

Second List

	A	B	C	
$m_0 + m_1$	0	0	–	✓
$m_0 + m_4$	–	0	0	✓
$m_1 + m_5$	–	0	1	✓
$m_4 + m_5$	1	0	–	✓

$\Longrightarrow$

Third List

	A	B	C
$(m_0 + m_1) + (m_4 + m_5)$	–	0	–
$(m_0 + m_4) + (m_1 + m_5)$	–	0	–

- The final simplified expression is: $Z = \overline{B}$
- Bear in mind that any unticked terms in any list must be included in the final expression (none occured here except from the last list). Note that the only prime implicant here is $Z = \overline{B}$.
- The tabular method reduces the function to a set of prime implicants.

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

★ Tabular Method of Minimisation ★

Determination of Prime Implicants

Example

Consider the function:

$$\begin{aligned} f(A,B,C) &= m_0 + m_1 + m_2 + m_3 + m_5 + m_7 + m_8 + m_{10} + m_{12} + m_{13} + m_{15} \\ &= 0000 + 0001 + 0010 + 0011 + 0101 + 0111 + 1000 + 1010 + 1100 + 1101 + 1111 \\ &= \underbrace{0000}_{\text{Index } \mathbf{0}} + \underbrace{0001 + 0010 + 1000}_{\text{Index } \mathbf{1}} + \underbrace{0011 + 0101 + 1010 + 1100}_{\text{Index } \mathbf{2}} + \underbrace{0111 + 1101}_{\text{Index } \mathbf{3}} + \underbrace{1111}_{\text{Index } \mathbf{4}} \end{aligned}$$

First List

		A	B	C	D	
Index **0**	m_0	0	0	0	0	✓
Index **1**	m_1	0	0	0	1	✓
	m_2	0	0	1	0	✓
	m_8	1	0	0	0	✓
Index **2**	m_3	0	0	1	1	✓
	m_5	0	1	0	1	✓
	m_{10}	1	0	1	0	✓
	m_{12}	1	1	0	0	✓
Index **3**	m_7	0	1	1	1	✓
	m_{13}	1	1	0	1	✓
Index **4**	m_{15}	1	1	1	1	✓

First List

		A	B	C	D	
Index **0**	m_0	0	0	0	0	✓
Index **1**	m_1	0	0	0	1	✓
	m_2	0	0	1	0	✓
	m_8	1	0	0	0	✓
Index **2**	m_3	0	0	1	1	✓
	m_5	0	1	0	1	✓
	m_{10}	1	0	1	0	✓
	m_{12}	1	1	0	0	✓
Index **3**	m_7	0	1	1	1	✓
	m_{13}	1	1	0	1	✓
Index **4**	m_{15}	1	1	1	1	✓

$\Longrightarrow$

Second List

	A	B	C	D	
$m_0 + m_1$	0	0	0	—	✓
$m_0 + m_2$	0	0	—	0	✓
$m_0 + m_8$	—	0	0	0	✓
$m_1 + m_3$	0	0	—	1	✓
$m_1 + m_5$	0	—	0	1	✓
$m_2 + m_3$	0	0	1	—	✓
$m_2 + m_{10}$	—	0	1	0	✓
$m_8 + m_{10}$	1	0	—	0	✓
$m_8 + m_{12}$	1	—	0	0	$A\overline{C}\,\overline{D}$
$m_3 + m_7$	0	—	1	1	✓
$m_5 + m_7$	0	1	—	1	✓
$m_5 + m_{13}$	—	1	0	1	✓
$m_{12} + m_{13}$	1	1	0	—	$AB\overline{C}$
$m_7 + m_{15}$	—	1	1	1	✓
$m_{13} + m_{15}$	1	1	—	1	✓

Third List

	A	B	C	D	
$(m_0+m_1)+(m_2+m_3)$	0	0	—	—	$\overline{A}\,\overline{B}$
$(m_0+m_2)+(m_1+m_3)$	0	0	—	—	
$(m_0+m_2)+(m_8+m_{10})$	—	0	—	0	$\overline{B}\,\overline{D}$
$(m_0+m_8)+(m_2+m_{10})$	—	0	—	0	
$(m_1+m_3)+(m_5+m_7)$	0	—	—	1	$\overline{A}\,D$
$(m_1+m_5)+(m_3+m_7)$	0	—	—	1	
$(m_5+m_7)+(m_{13}+m_{15})$	—	1	—	1	BD
$(m_5+m_{13})+(m_7+m_{15})$	—	1	—	1	

- The terms that have not been checked off are called prime implicants
- Thus the prime implicants are $\overline{A}\,\overline{B}$, $\overline{B}\,\overline{D}$, $\overline{A}\,D$, BD, $A\overline{C}\,\overline{D}$, $AB\overline{C}$
- Therefore $f(A,B,C) = \overline{A}\,\overline{B} + \overline{B}\,\overline{D} + \overline{A}\,D + BD + A\overline{C}\,\overline{D} + AB\overline{C}$

1 Algebraic Manipulation of Boolean Expressions

2 Writing a logic equation from a truth table

3 Karnaugh Maps
- Introduction
- Rules of Simplification
- Exercises

4 Tabular Method of Minimisation
- Rules of Tabular Method
- Determination of Prime Implicants
- Prime Implicant Chart

★ Tabular Method of Minimisation ★

Prime Implicant Chart

④ **Chart layout:** The chart is used to remove redundant prime implicants.

A grid is prepared having:

- Top row lists minterms of the function.
- All prime implicants are listed on the left side.
- Place ***x*** into the chart according to the minterms that form the corresponding prime implicant.

► **Essential prime implicant**:

If a minterm is covered only by one prime implicant, that prime implicant is called essential prime implicant.

Essential prime implicant must be included in the minimum sum of the function.

★ Tabular Method of Minimisation ★

Prime Implicant Chart

	0	1	2	3	5	7	8	10	12	13	15
$\overline{A}\,\overline{B}$	x	x	x	x							
$\overline{B}\,\overline{D}$	x		x				x	X			
$\overline{A}D$		x		x	x	x					
BD					x	x				x	X
$A\,\overline{C}\,\overline{D}$							x		x		
$AB\,\overline{C}$									x	x	
								$\downarrow$ X			$\downarrow$ X

		$\cancel{0}$	1	$\cancel{2}$	3	$\cancel{5}$	$\cancel{7}$	$\cancel{8}$	$\cancel{10}$	12	$\cancel{13}$	$\cancel{15}$
	$\overline{A}\,\overline{B}$	$\cancel{x}$	x	$\cancel{x}$	x							
$\overline{B}\,\overline{D} \leftarrow$	$\overline{B}\,\overline{D}$	$\cancel{x}$		$\cancel{x}$				$\cancel{x}$	X			
	$\overline{A}D$		x		x	$\cancel{x}$	$\cancel{x}$					
$BD \leftarrow$	BD					$\cancel{x}$	$\cancel{x}$				$\cancel{x}$	X
	$A\,\overline{C}\,\overline{D}$							$\cancel{x}$		x		
	$AB\,\overline{C}$									x	$\cancel{x}$	
									$\downarrow$ X			$\downarrow$ X

- From the above chart, ***BD*** is an essential prime implicant. It is the only prime implicant that covers the minterm decimal **15** and it also includes **5**, **7** and **13**. $\boldsymbol{\overline{B}\,\overline{D}}$ is also an essential prime implicant. It is the only prime implicant that covers the minterm denoted by decimal **10** and it also includes the terms **0**, **2** and **8**.

		$\cancel{0}$	1	$\cancel{2}$	3	$\cancel{5}$	$\cancel{7}$	$\cancel{8}$	$\cancel{10}$	12	$\cancel{13}$	$\cancel{15}$
	$\overline{A}\,\overline{B}$	$\cancel{x}$	x	$\cancel{x}$	x							
$\overline{B}\,\overline{D} \leftarrow$	$\overline{B}\,\overline{D}$	$\cancel{x}$		$\cancel{x}$				$\cancel{x}$	X			
	$\overline{A}D$		x		x	$\cancel{x}$	$\cancel{x}$					
$BD \leftarrow$	BD					$\cancel{x}$	$\cancel{x}$				$\cancel{x}$	X
	$A\overline{C}\,\overline{D}$							$\cancel{x}$		x		
	$AB\overline{C}$									x	$\cancel{x}$	
									$\downarrow$ X			$\downarrow$ X

- From the above chart, $\boldsymbol{BD}$ is an essential prime implicant. It is the only prime implicant that covers the minterm decimal **15** and it also includes **5**, **7** and **13**. $\overline{B}\,\overline{D}$ is also an essential prime implicant. It is the only prime implicant that covers the minterm denoted by decimal **10** and it also includes the terms **0**, **2** and **8**.

- The other minterms of the function are **1**, **3** and **12**.
  - Minterm **1** is present in $\overline{A}\,\overline{B}$ and $\overline{A}D$. Similarly for minterm **3**. We can therefore use either of these prime implicants for these minterms.
  - Minterm **12** is present in $A\overline{C}\,\overline{D}$ and $AB\overline{C}$, so again either can be used.

Thus, one minimal solution is $f(A, B, C, D) = \overline{B}\,\overline{D} + BD + \overline{A}\,\overline{B} + A\overline{C}\,\overline{D}$

★ Tabular Method of Minimisation ★

Exercices

$\bar{B}\bar{C} + C\bar{D} + \bar{A}BC + \bar{A}\bar{C}D$

Exercise 11. Find the essential prime implicant

	0	1	2	5	6	7	8	9	10	14
$\bar{B}\bar{C}$	x	x					x	X		
$\bar{B}\bar{D}$	x		x				x		x	
$C\bar{D}$			x		x				x	X
$\bar{A}\bar{C}D$		x		x						
$\bar{A}BD$				x		x				
$\bar{A}BC$					x	x				
								x		x

Exercise 12. Minimize the function below using the tabular method of simplification:

$$Z = f(A, B, C, D) = \bar{A}\,\bar{B}\,\bar{C}D + \bar{A}\,\bar{B}CD + A\bar{B}CD + A\bar{B}C\bar{D} + \bar{A}BCD + \bar{A}BC\bar{D} + \bar{A}\,\bar{B}CD$$

Exercise 13. Using the tabular method of simplification, find all equally minimal solutions for the function below.

$$Z = f(A, B, C, D) = \sum(m_1 + m_4 + m_5 + m_{10} + m_{12} + m_{14}).$$

- **Literal:** A Literal signifies the Boolean variables including their complements. Such as B is a boolean variable and its complements are B or B', which are the literals.
- **Minterm:** In Boolean algebra, a product term, with a value of 1, in which each variable appears once (in either its complemented or uncomplemented form, so that the value of the product term becomes 1).
- **Prime implicant:** A prime implicant of a function is an implicant (product term) that cannot be covered by a more general, (more reduced, meaning with fewer literals) implicant.
- **Essential prime implicants:** Essential prime implicants (also known as core prime implicants) are prime implicants that cover an output of the function that no combination of other prime implicants is able to cover

AB \ CD	00	01	11	10
00				
01				
11				
10				

Thank you! Questions?