电 子 学 基 础 系 列

ELECTRONICS

DIGITAL LOGIC
FOUNDATION COURSE

数字逻辑基础

学习指导与教学参考

陈光梦 王勇 编著

复旦大学出版社
www.fudanpress.com.cn

电子学基础系列
ELECTRONICS

DIGITAL LOGIC
FOUNDATION COURSE

数字逻辑基础

学习指导与教学参考

陈光梦　王勇　编著

復旦大學出版社
www.fudanpress.com.cn

图书在版编目(CIP)数据

数字逻辑基础学习指导与教学参考/陈光梦,王勇编著.
—上海:复旦大学出版社,2004.6
(电子学基础系列)
ISBN 7-309-04034-1

Ⅰ.数… Ⅱ.陈… Ⅲ.数字逻辑-高等学校-教学参考资料
Ⅳ.TP302.2

中国版本图书馆 CIP 数据核字(2004)第 047335 号

数字逻辑基础学习指导与教学参考

陈光梦 王 勇 编著

出版发行 复旦大学 **出版社**
上海市国权路 579 号 邮编 200433
86-21-65118853(发行部) 86-21-65109143(邮购)
fupnet@fudanpress.com http://www.fudanpress.com

责任编辑 梁 玲
装帧设计 孙 曙
总 编 辑 高若海
出 品 人 贺圣遂

印 刷 上海第二教育学院印刷厂
开 本 787×960 1/16
印 张 9.25 插页 2
字 数 188 千
版 次 2004 年 6 月第一版第一次印刷
印 数 1—3 100

书 号 ISBN 7-309-04034-1/T·287
定 价 16.00 元

作者简介

陈光梦，男，1950年出生。1966年因文化大革命辍学，进入工厂。1977年恢复高考后考入复旦大学，毕业后留校至今。留校以后一直从事电路与系统的教学和科研工作。长期从事电子线路基础教学，曾参加过国家教委组织的中华学习机系列的研制工作，参加过上海多家工厂的工业自动化改造项目，在自动控制技术、可编程逻辑器件应用技术、声音与图像的处理与应用技术等领域开展过不少工作，编著有《可编程逻辑器件的原理与应用》、《数字逻辑基础》。

王勇，男，1964年出生。现为复旦大学信息科学与工程学院电子工程系副教授，教育部电子信息与电气信息类基础课教学指导分委员会委员，长期从事电子信息科学与技术专业的教学和研究工作。

内容提要

本书是《数字逻辑基础》一书的教学参考书，基本章节完全按照《数字逻辑基础》一书安排。每章分为三部分：第一部分是本章内容的重点和难点；第二部分内容是对于《数字逻辑基础》一书中习题与思考题的解答；第三部分以一定的篇幅介绍在《数字逻辑基础》一书中没有涉及的一些扩充内容。这些内容可以作为教师在讲课时的参考资料，也可以供能力比较强的学生或者工程技术人员参考。

前　言

数字逻辑是一门电子信息方面的重要课程.考虑到在数字逻辑课程的教学中,教师除了需要掌握数字逻辑的一般知识以外,需要对使用的教科书的内容包括其中的习题做到心中有数,所以我们在编写了教科书《数字逻辑基础》以后编写了本书.

本书是《数字逻辑基础》一书的教学参考书,基本章节完全按照《数字逻辑基础》一书安排.编写本书的主要目的是作为教师进行教学活动时的参考.由于本书是一本教学参考书,所以对于在《数字逻辑基础》一书中已经展开的内容以及学生容易掌握的内容,均不再展开讨论.

本书的特点是:

1. 列出了每章的重点与难点.

每章的第一部分首先简要地回顾了本章的内容、学生应该掌握课程内容的情况,以及本章内容的重点和难点,然后对内容的难点进行详尽的分析.在分析过程中或者列举例题,或者指出这些难点的内容在哪些习题中得到解答.这些内容可供教师参考,也可以作为教师讲课的材料.

2. 分析和解答了每章的习题,并对其中某些问题展开讨论.

每章的第二部分内容是对于《数字逻辑基础》一书中的习题与思考题的解答.对于这些习题与思考题,比较简单的仅提供解题过程和答案,复杂的则附以一定说明文字,对于具有典型意义的问题则进行比较详尽的分析,有些问题还在原来题目的基础上加以引申以开拓读者的思路.

3. 增加了补充和加深的内容.

为了帮助读者更好地了解数字逻辑的内容,除了第 3 章、第 6 章外,每章在习题解答之后的第三部分均以一定的篇幅介绍在《数字逻辑基础》一书中没有涉及的一些扩充内容.这部分内容可以作为教师在讲课时的参考资料,也可以作为能力比较强的学生或者工程技术人员的参考.

本书由陈光梦负责撰写每章第一部分和第三部分的内容,王勇撰写第 1 章到第 5 章的第二部分(习题解答),第 6 章的习题解答由陈光梦撰写.最后由陈光梦对全书进行整理.

由于《数字逻辑基础》一书的特点是涉及面比较广,有些内容在一般的教科书

中较少讨论,而本书是针对《数字逻辑基础》一书习题的解析和补充,所以本书可以作为数字逻辑课程教学的补充资料,也可以作为一般工程技术人员乃至学生在从事数字逻辑设计和学习时的参考资料.

由于编写时间紧促,书中难免有挂一漏万及错误之处,望读者指正.

陈光梦　王勇

2004 年 4 月

目　录

第 1 章　逻辑代数基础

§1.1　要点与难点分析

本章是数字逻辑的基础理论部分.在教材中讨论了逻辑代数中的基本内容:逻辑函数、基本逻辑运算及其定律与定理、逻辑函数的化简以及逻辑函数目标形式的转换.其中基本逻辑运算定律与定理是逻辑运算和化简的基础,尤其是其中 3 个逻辑定理,学生应该牢固掌握.

逻辑函数的化简是本章的重点.

逻辑函数的化简可以用公式法进行,也可以用卡诺图进行.由于卡诺图化简比较直观,用手工化简逻辑函数时通常用卡诺图方法.

单变量逻辑函数用卡诺图化简成 SOP 形式是卡诺图化简的基础,可以按照一定的步骤进行,过程比较机械,教材中对此做了比较详细的描述,学生很容易掌握.本章大部分有关卡诺图的习题属于这种类型.

单变量逻辑函数化简的难点是给定结果形式的化简.

逻辑函数化简的另一个问题是带任意项的函数化简和多输出函数的化简.带任意项的函数化简比较容易,唯约束条件的判断稍有难度.多输出函数的化简不是很困难,但是要求学生有比较敏锐的观察力.

用卡诺图化简逻辑函数过程中的一个难点是要求按照特定的目标函数形式(非 SOP 形式)对逻辑函数进行化简.

如果要求化简成“或-与”形式,可以在化简时直接按照 POS 的形式进行.在用这种方法化简时特别要注意和通常的 SOP 形式加以区别,原则是:“0”、“1”交换;“或”、“与”交换;原变量、反变量交换.具体例子可以参见本章习题 9.

如果要求化简成其他形式,或者在化简时提出对于输入变量的限制、逻辑门形式与数量的限制等附加的约束条件,则化简过程有一定的技巧.

一、利用逻辑函数的转换进行化简

将逻辑函数按照特定的目标函数进行化简的一个方法是先按照最小项化简,

然后将化简得到的逻辑函数进行转换.可以应用这个方法进行化简的常见的目标逻辑函数形式有:“与-或”形式、“或-与”形式、“与非-与非”形式、“或非-或非”形式、“与或非”形式等.由于在这个方法中,按照最小项化简得到一个最小覆盖的逻辑函数一定是“与-或”形式,所以要研究将“与-或”形式转换成其他形式的方法.

1 将“与-或”形式转换为“或-与”形式

利用对偶定理,可以先写出化简后的“与-或”形式逻辑函数的对偶逻辑函数并展开,然后再次写出展开后的逻辑函数的对偶逻辑函数.这样得到的逻辑函数就是原函数的最简“或-与”形式.

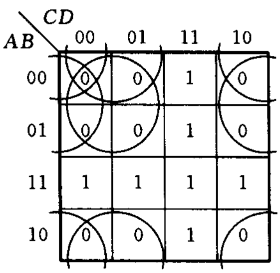


图 1-1 例 1-1 的卡诺图化简

例 1-1 将逻辑函数 $Y = AB + CD$ 转换为“或-与”形式.

解:利用对偶定理,

$$Y^* = (A+B)\cdot(C+D)$$
$$= AC + BC + AD + BD$$
$$Y = (Y^*)^*$$
$$= (A+C)(B+C)(A+D)(B+D)$$

本例若用卡诺图直接化简,采用圈最大项的办法,也可以得到相同的结果,如图 1-1 所示.

2 将“与-或”形式转换为“与非-与非”形式

利用自反律和反演定理,将原来的“与-或”形式逻辑函数进行两次求非,可以得到原函数的“与非-与非“形式.

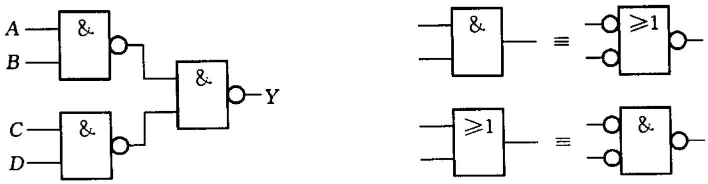
例 1-2 将逻辑函数 $Y = AB + CD$ 转换为“与非-与非”形式.

解:利用自反律和反演定理,

$$Y = \overline{\overline{AB + CD}} = \overline{\overline{AB} \cdot \overline{CD}}$$

上例结果的逻辑图如图 1-2(a)所示.

实际上,运用反演律,可以将与门等效成带输入反相的或非门,也可以将或门等效成带输入反相的与非门,如图 1-2 (b)所示.这个特性常常用来将一个逻辑函数转换成用同一种门实现.



(a)　　　　　　(b)

图 1-2　例 1-2 的逻辑图描述

3　将"与-或"形式转换为"或非-或非"形式

综合上面两个例子的方法，先将原函数利用对偶定理转换成"或-与"形式，再两次求非可以得到原函数的"或非-或非"形式.

例 1-3　将逻辑函数 $Y = AB + CD$ 转换为"或非-或非"形式.

解：先对偶，再两次求非：

$$
\begin{aligned}
Y &= (Y^*)^* \\
&= (A+C)(B+C)(A+D)(B+D) \\
&= \overline{\overline{(A+C)(B+C)(A+D)(B+D)}} \\
&= \overline{\overline{(A+C)}+\overline{(B+C)}+\overline{(A+D)}+\overline{(B+D)}}
\end{aligned}
$$

4　将"与-或"形式转换为"与或非"形式

将"与-或"形式转换为"与或非"形式可以这样思考：由于最后的逻辑表达式是一个"非"输出（即在最后逻辑表达式的上面有一个横贯全式的"非"号），所以可以先将全式进行两次求非运算，然后保留最外面的"非"号，利用反演律将中间的"非"号消去，再进行展开整理，最后可以得到需要的函数形式. 一般而言，通过这样转换得到的结果可能带有反变量输入.

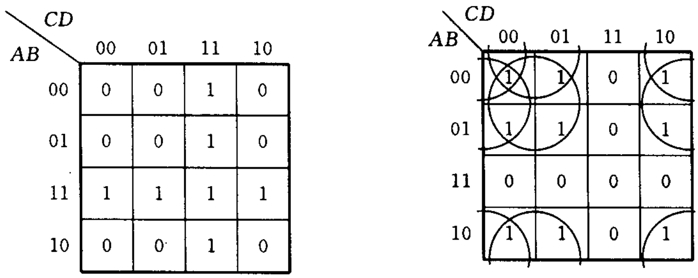
例 1-4　将逻辑函数 $Y = AB + CD$ 转换为"与或非"形式.

解：两次求非、反演、展开：

$$
\begin{aligned}
Y &= \overline{\overline{AB+CD}} \\
&= \overline{\overline{AB}\cdot\overline{CD}} \\
&= \overline{(\overline{A}+\overline{B})\cdot(\overline{C}+\overline{D})} \\
&= \overline{\overline{A}\,\overline{C}+\overline{B}\,\overline{C}+\overline{A}\,\overline{D}+\overline{B}\,\overline{D}}
\end{aligned}
$$

其实上述转换过程可以直接在卡诺图上进行. 由于要求最后的逻辑表达式为

“与或非”形式，所以若将原卡诺图取反，则相当于在取反的卡诺图上化简成“与或”形式. 根据这个想法对上例进行卡诺图化简的过程见图 1-3，化简结果是原函数的反函数“与或”形式，再将结果取反，可以得到与上述转换过程一致的形式.



$Y = AB + CD$　　　$\overline{Y} = \overline{A}\,\overline{C} + \overline{B}\,\overline{C} + \overline{A}\,\overline{D} + \overline{B}\,\overline{D}$

(a)　　　(b)

图 1-3　例 1-4 的卡诺图化简过程

二、利用卡诺图运算进行化简

将逻辑函数按照特定的目标函数进行化简的另一个方法是应用卡诺图运算法进行化简. 此方法一般可以将逻辑函数化简成“与非”、“或非”和“与或非”形式的最简表达式，并且通常在这个目标函数中可以不包含反变量输入. 但必须注意，并不是所有的逻辑函数都可以利用此方法进行化简. 另外也可以利用卡诺图运算将逻辑函数化简成某些特殊形式. 下面就上述几种情况展开讨论.

1　将逻辑函数化简成原变量输入的“与非”形式

由于要求原变量输入，又是化简成“与非”形式，所以所有的卡诺圈(包括阻塞圈)必须圈住“1”重心，根据这个原则，可以得到这种类型逻辑函数化简的一般过程：

(1) 围绕“1”重心(即圈出的卡诺圈中必须包含“1”重心)圈出所有包含“1”的卡诺圈，此圈可以包含部分“0”方格.

(2) 围绕“1”重心画阻塞圈. 此圈应包含所有在上一个步骤中被圈入但是其逻辑值为“0”的方格，也可包含未被上一个步骤中被圈入的“0”方格.

(3) 在上述步骤中，卡诺圈和阻塞圈都要求尽可能大，并且都可以有多个.

(4) 将步骤(1)得到的结果转换成“与非-与非”形式，将步骤(2)得到的结果写成“与非”形式. 然后用步骤(2)的结果去阻塞步骤(1)的结果，得到最后的结果.

例 1-5　将逻辑函数 $Y(ABC) = \sum m(1, 4, 5, 6)$ 化简为原变量输入的“与

非"形式.

解:本例的卡诺图如图 1-4.

根据前面的步骤,先围绕"1"重心圈出所有包含"1"的卡诺圈,如图中实线所圈的两个卡诺圈,它们的函数分别为 $Y_1 = C$ 和 $Y_2 = A$. 由于要求尽可能大,它们分别圈入了多个"0"方格. 然后画阻塞圈,也围绕"1"重心进行,圈出前面卡诺圈中多余的"0"方格,如图中虚线所圈的 Z,它的函数为$\overline{BC}$.

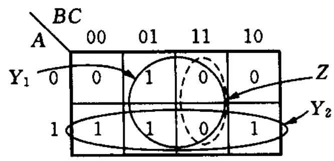


图 1-4　例 1-5 的卡诺图

由于卡诺圈 Y_1 中多圈入的"0"方格可以被 Z 阻塞,根据卡诺图运算规则,SOP 表示的卡诺圈和阻塞圈之间的阻塞关系为"与"关系,所以卡诺圈 Y_1 被阻塞后剩余的最小项 m_1、m_5 可以表示为 $\sum m(1, 5) = C \cdot \overline{BC}$. 同样,卡诺圈 Y_2 被阻塞后剩余的最小项 m_4、m_5、m_6 可以表示为 $\sum m(4, 5, 6) = A \cdot \overline{BC}$. 所以,最后得到化简后的函数为

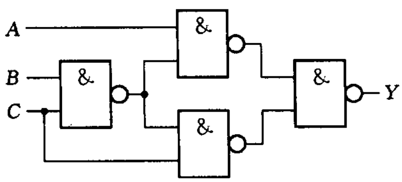


图 1-5　例 1-5 的逻辑图

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

对应的逻辑图如图 1-5. 从图 1-5 可以看到,采用这个方法化简后的逻辑有 3 级,所以可能在速度上要稍稍慢一些. 下面讨论一个稍复杂的例子.

例 1-6　将逻辑函数 $Y(ABCD) = \sum m(2, 6, 8, 9, 10)$ 化简为原变量输入的"与非"形式.

解:本例的卡诺图如图 1-6.

实线所圈的两个卡诺圈分别为 $Y_1 = C$ 和 $Y_2 = A$, 虚线所圈的阻塞圈分别为 $Z_1 = \overline{CD}$ 和 $Z_2 = \overline{AB}$.

由于卡诺圈 Y_1和 Y_2 均被两个阻塞项 Z_1 和 Z_2 所阻塞,根据卡诺图运算规则,卡诺圈 Y_1 被两个阻塞项 Z_1 和 Z_2 阻塞后可以表示为 $\sum m(2, 6, 10) = C \cdot \overline{AB} \cdot \overline{CD}$, 卡诺圈 Y_2 被两个阻塞项 Z_1 和 Z_2 阻塞后可以表示为 $\sum m(8, 9, 10) = A \cdot \overline{AB} \cdot \overline{CD}$, 所以最后得到化简后的函数为

$$Y = A \cdot \overline{AB} \cdot \overline{CD} + C \cdot \overline{AB} \cdot \overline{CD} = \overline{\overline{A \cdot \overline{AB} \cdot \overline{CD}} \cdot \overline{C \cdot \overline{AB} \cdot \overline{CD}}}$$

对应的逻辑图如图 1-7 所示.

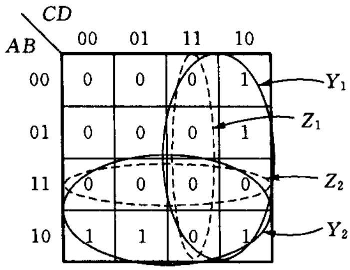


图 1-6 例 1-6 的卡诺图

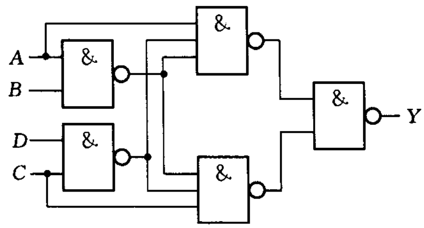


图 1-7 例 1-6 化简后的逻辑图

2 将逻辑函数化简成原变量输入的“或非”形式

原变量输入“或非”形式的卡诺图化简要求所有的卡诺圈(包括阻塞圈)必须围绕“0”重心圈“0”,根据这个原则,可以得到这种类型逻辑函数化简的一般过程:

(1) 围绕“0”重心圈出所有包含“0”的卡诺圈,即按照最大项的方式画卡诺圈,但是此圈可以包含部分“1”方格.

(2) 围绕“0”重心画阻塞圈.此圈应包含所有在步骤(1)中被圈入但是其逻辑值为“1”的方格,也可包含未被步骤(1)中被圈入的“1”方格.

(3) 在上述步骤中,卡诺圈和阻塞圈都要求尽可能大,并且都可以有多个.

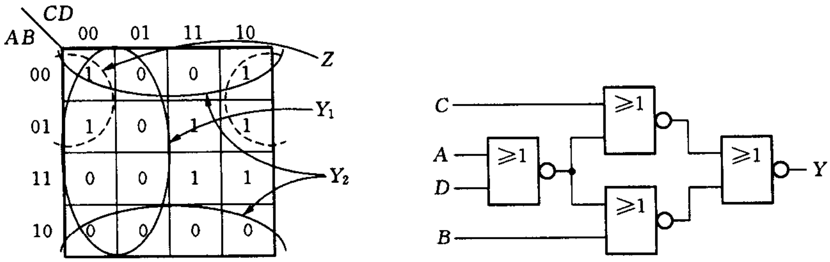
(4) 将步骤(1)得到的结果按照最大项的方式写成“或非-或非”形式,将步骤(2)得到的结果写成“或非”形式.然后用步骤(2)的结果去阻塞步骤(1)的结果,得到最后的结果.

例 1-7 将逻辑函数 $Y(ABCD)=\sum m(0, 2, 4, 6, 7, 14, 15)$ 化简为原变量输入的“或非”形式.

解:按照前面所述的步骤,在图 1-8 的卡诺图中用实线圈出两个卡诺圈,按照最大项之积方式分别记为 $Y_1=C$ 和 $Y_2=B$,它们所构成的“或-与”式应该为 $Y_1\cdot Y_2$.用虚线圈出阻塞圈,按照最大项之积方式记为 $Z=\overline{A+D}$.按照最大项之积方式构成阻塞函数时,阻塞函数和被阻塞函数之间应该进行“或”运算,所以最后得到的函数为

$$\begin{aligned} Y &= (Y_1+Z)\cdot(Y_2+Z) \\ &= (C+\overline{A+D})\cdot(B+\overline{A+D}) \\ &= \overline{\overline{C+(\overline{A+D})}+\overline{B+(\overline{A+D})}} \end{aligned}$$

化简后的逻辑图已经在图 1-8 中给出.将本例同例 1-5 对比,可以看到,只要注意到按照最大项之积方式进行化简,即“0”、“1”交换,“或”、“与”交换,原变量、反变量交换,则两种形式的化简过程完全一致.



(a)　　　　(b)

图 1-8　例 1-7 的卡诺图和逻辑图

3　将逻辑函数化简成原变量输入的“与或非”形式

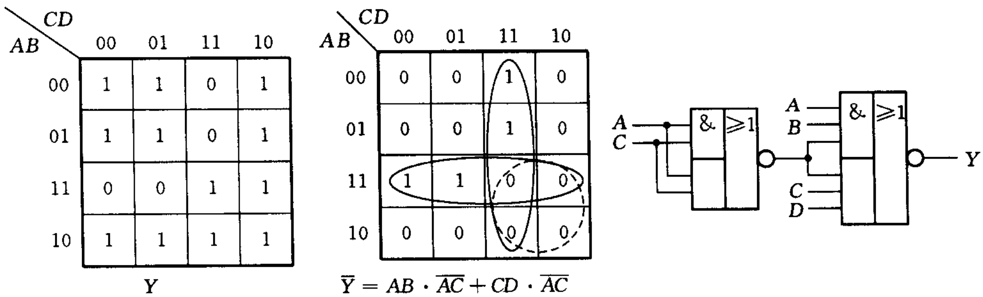
我们在例 1-4 看到，若要求化简的最后结果为“与或非”形式，可以先将卡诺图取反，然后按照“与或”形式进行化简. 所以，要求将逻辑函数化简成原变量输入的“与或非”形式，可以先取反，再按照前面“与或”形式的化简过程进行. 唯一要注意的是，阻塞函数也要转换成“与或非”形式.

例 1-8　将逻辑函数 $Y(ABCD) = \prod M(3, 7, 12, 13)$ 化简为原变量输入的“与或非”形式.

解：本例的化简过程如图 1-9 所示，先将原函数取反，在取反后的卡诺图上按照“与或”形式用阻塞法化简，得到中间结果 $\overline{Y} = AB \cdot \overline{AC} + CD \cdot \overline{AC}$. 再将阻塞项 $\overline{AC}$ 转换成“与或非”形式 $\overline{AC} = \overline{AC + AC}$，就得到了最后结果：

$$
\begin{aligned}
Y &= \overline{AB \cdot \overline{AC} + CD \cdot \overline{AC}} \\
&= \overline{AB \cdot \overline{AC + AC} + CD \cdot \overline{AC + AC}}
\end{aligned}
$$

逻辑图已经画在图 1-9 中.



(a)　　　　(b)　　　　(c)

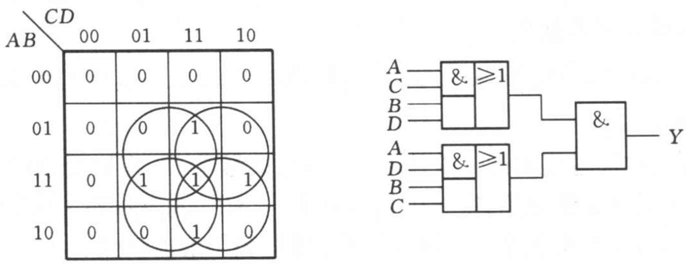
图 1-9　例 1-8 的卡诺图

前面讨论的 3 种利用卡诺图运算进行化简的过程，得到的是 3 种特定的结果形式. 它们的共同特点是：要求最后结果以原变量输入. 由于这个约束条件，在圈卡诺圈和阻塞圈时，都必须注意到围绕某个重心进行："与非"和"与或非"形式围绕"1"重心，"或非"形式围绕"0"重心. 上述步骤是一个比较机械的过程.

然而对于一般的函数形式，未必都能够化简成上述 3 种结果，或者虽然化简成上述形式，但未必就是最简的形式. 所以在运用卡诺图运算化简时，有时候要根据具体的函数特点进行化简. 在教材中讨论了化简成"异或"函数形式的例子，下面举一个化简成"与或"形式但是运用了卡诺图运算方法的例子.

例 1-9 化简逻辑函数 $Y(ABCD)=\sum m(7,\ 11,\ 13,\ 14,\ 15)$，要求化简后的逻辑电路具有最少的输入端.

解：本例的卡诺图和化简结果如图 1-10 所示，最后结果为 $Y=(AC+BD)(AD+BC)$. 若分别计算与门和或门的输入端，共计有 14 个输入端. 若按常规方法化简，得到的结果为 $Y=ABD+ABC+ACD+BCD$，共计有 16 个输入端.



(a)　　(b)

图 1-10　例 1-9 的卡诺图

很明显，在本例的化简中运用了卡诺图运算，利用两组交叉的卡诺圈的"与"，消除了其中 4 个"0"方格，得到比较简单的结果. 但是类似这样的运算，无法完全归纳出规律，还要依赖化简者的观察力，所以是比较困难的化简过程.

§1.2 习题解答

1. 运用基本定理证明下列等式.

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

证明： $AB+\overline{A}C+\overline{B}C=AB+\overline{A}C+BC+\overline{B}C=AB+\overline{A}C+C=AB+C$

(2) $BC+D+\overline{D}(\overline{B}+\overline{C})(DA+B)=B+D$

证明：