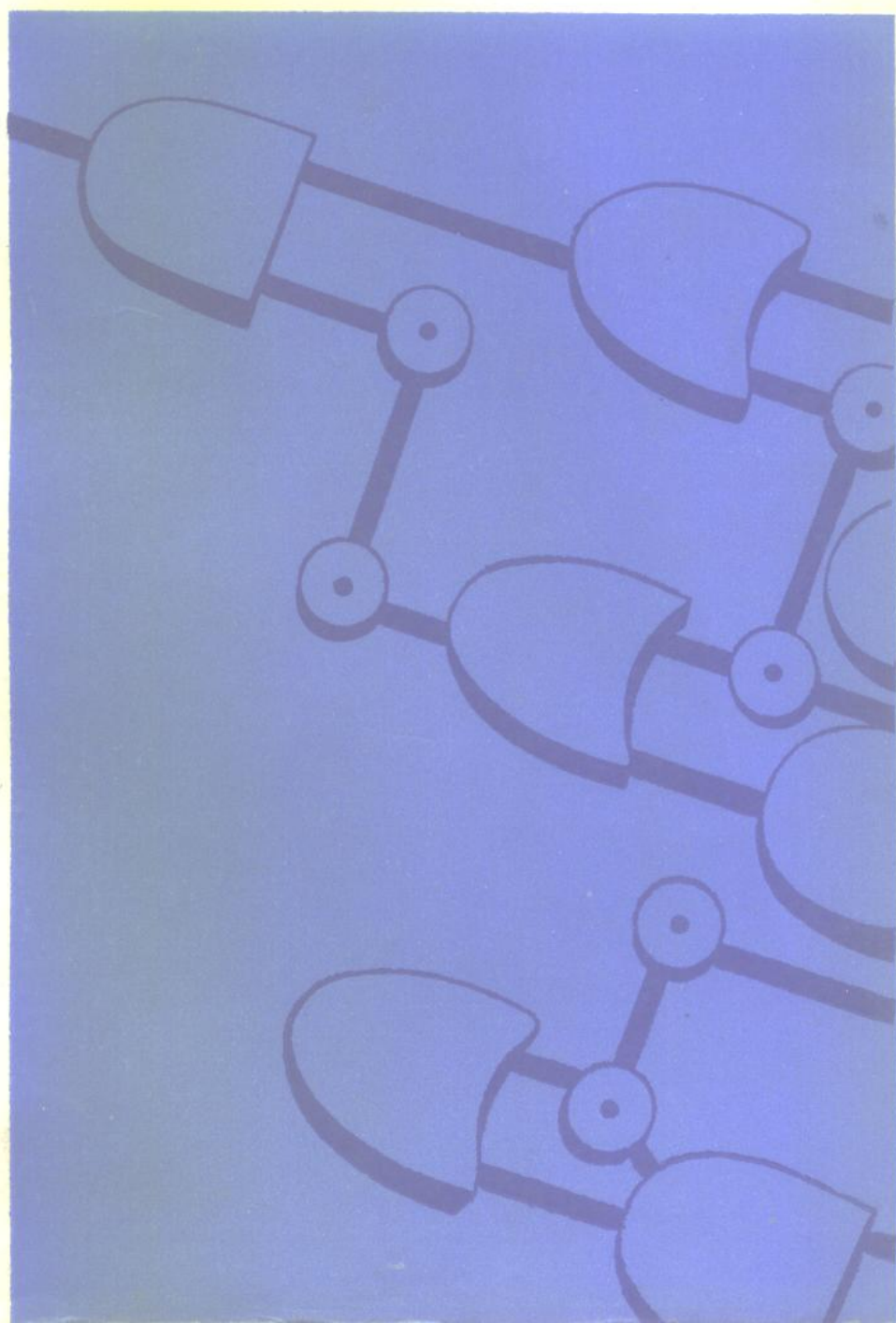


通用逻辑阵列

GAL

周永钊 张雷 陈铭 编译 ● 中国科学技术大学出版社



551
1

通用逻辑阵列(GAL)

周永钊

张 雷 编译

陈 铭

周建民 校

中国科学技术大学出版社

1989·合肥

内 容 简 介

本书向国内工程界和学术界介绍了国际上的最新通用逻辑阵列 (GAL) 技术, 内容包括 GAL 芯片的原理, 基本功能, 软硬件开发工具、实际应用等, 本书列举的若干应用实例是完全可以参照实现的。

本书适于从事工业过程控制、仪表测量、计算机工程、通信工程、电子技术等方面的工程师、科研工作者、教师、高年级大学生和研究生作为参考读物和选择器件的手册, 也可作为开设这门新课程的教学参考书。

DS93/27

通用逻辑阵列(GAL)

周永钊

张 雷 编 译

陈 铭

周建民 校

责任编辑: 黄 德 封面设计: 王瑞荣

*

中国科学技术大学出版社出版

(安徽省合肥市金寨路96号)

中国科学技术大学印刷厂印刷

安徽省新华书店发行

*

开本: 787×1092/16 印张: 15.75 字数: 361 千

1989年7月第1版 1989年7月第1次印刷

印数: 1—13000

ISBN 7-312-00123-8 / TP·6 定价: 5.50 元

前 言

众所周知, ASIC (Application-Specific Integrated Circuit 专用集成电路) 是半定制型的标准单元 (Standard Cell)、门阵列和可编程逻辑器件 (PLD) 的总称。GAL (Generic Array Logic) 器件则是在目前大量行销的 PAL (Programmable Array Logic) 器件的基础上发展起来的崭新的逻辑芯片, 是一种最具竞争能力的 ASIC 器件。

十年来, ASIC 产品无论在性能、集成度, 或是产量、产值方面都已取得了飞速的发展。现在, 世界 IC (集成电路) 市场对 ASIC 的需求急剧增加, 其销售额占 IC 市场的 32.4%。应用方便灵活的 ASIC 产品——PAL 器件, 每年的销售额达数百万芯片。而 GAL 芯片以其最优的性能和 100% 的成品率在 IC 市场上的影响日益增长。

现在世界上应用和开发以 GAL、PAL 为代表的 PLD 技术的势头方兴未艾。自 1985 年起, 国外设计的新产品都应用了 PAL 技术。而 1986 年以来, 国外技术杂志不断发表有关 GAL 设计与开发技术的文章, 每年约数百篇。国内一些大专院校、科研单位和公司也竞相研究 GAL 和 PAL 的开发工具及其应用技术, 已有产品运用了 PLD 技术进行设计。清华大学李三立教授说:“今天的系统设计工程师不会设计 ASIC (GAL) 产品, 就好象五、六十年代的系统设计工程师不会设计印刷电路板一样。”清华大学微电子研究所引进了 CAD 工作站, 旨在设计出新一代的 ASIC 产品; 复旦大学计算机系将 GAL 产品应用于单片机开发系统 SICE 中; 北京四通集团公司 ASIC 事业部及珠海新电集成有限公司着眼于国内外 ASIC 领域的最新 GAL 技术和产品, 研制和开发了四代 GAL 编程器; 中国科学院声学所、北京工业大学电子厂分别在人工智能和 STD 总线方面开展了 GAL 的应用研究; 中国科学技术大学还开设了这方面的课程。

据国内有关专家推测, 未来的 VLSI 市场将属于 ASIC。甚至有人断言, 随着集成度的提高和设计手段的改进, 到 2000 年, ASIC 将在 IC 市场占统治地位。当然, 作为 ASIC 的主要部分及使用方便的器件——GAL 和 PAL 技术, 无疑将成为其发展的先导。

现在, PAL 和 GAL 等 PLD 器件应用技术正向着其广度与深度两个方面发展。就其广度而言, 是随着个人微机的普及而展开的。一个系统设计师只要有一台微机及相应的开发工具, 就可以利用这些芯片, 做成自己特有的 (简单的基本逻辑到复杂的接口) 多种芯片。这种设计的方便性和灵活性, 促使了 PLD 器件的应用向其广度发展。就其深度而言, 各个 PLD 器件的开发工具专业公司正大力开发有关的软件和硬件, 包括能开发各种 PLD 器件的包罗万象的工具, 也包括能与 PCAD 工作站相连接的 PLD 器件库——用此来建立包括 PLD 器件的计算机辅助设计的专家系统。

GAL 和 PAL 器件应用技术广度和深度的发展, 促进和造成了其设计技术的不断改进和完善。目前, 各个等级的 PLD 器件开发工具, 具有与传统设计全然不同的特色。高级的开发工具允许用户用布尔方程、逻辑框图、状态转换图等多种形式的输入, 通过相应的开发工具, 给出所设计芯片的 JEDEC 文件、简化的布尔方程、乘积项说明及矢量测试

文件，然后自动地对PLD芯片进行编程和核对。正在研制的更为高级的开发软件甚至能对整个系统进行芯片的排布、印刷板的设计以及提供整个系统的预先测试。这种设计技术的不断改进和完善，给工程设计人员带来了莫大的方便，同时也给人们提出了一个对这种新的设计方法要尽快适应和学习的要求。

在PLD器件迅速发展、传统的设计方法受到冲击的情况下，我们编译了《通用逻辑阵列(GAL)》一书，以期向电子学界和计算机界的同事们，向高等学校、研究单位及企业的有关人员介绍PLD器件的娇娇者——GAL器件。

GAL器件是1985年由美国LATTICE公司投产的。GAL器件是在PAL器件的基础上，综合了E²和CMOS技术发展起来的一门新技术。GAL器件与PAL器件100%地兼容，同时具有PAL所没有的可擦除、可重新编程及可组态其结构的特点。这些特点就造成了器件100%的成品率、100%的编程可靠性的优异性能；并且使GAL能组成PAL所没有的结构。显然，GAL器件在实验室环境下，在样机试制过程中，具有更大的灵活性。本书围绕GAL器件的这些特点，就其原理进行了阐述；同时本书给出了若干应用的例子，以便读者能了解这种器件的设计方法。当然，读者可直接援引这些例子到自己的系统设计中。为便于读者实际使用，本书也收集了一些有关的技术资料。

在编译过程中，编者对原材料进行了若干甄别与订正，限于成书时间紧迫和编者实践经验，书中难免有错误和不妥之处，敬请读者批评指正。

本书由周永钊、张雷、陈铭编译。初稿承蒙周建民同志校核，最后由周永钊总纂定稿。成书过程中，研究生李勇、赵彤丹做了很多实际工作。

编者

1989年4月

目 次

第一章 GAL 器件概要	(1)
为何要采用可编程逻辑?	(1)
GAL 器件的概念	(2)
GAL 器件的优点	(3)
GAL 器件开发工具	(4)
第二章 逻辑设计基础与 PLD 器件的演变	(6)
逻辑设计基础	(6)
PLD 器件的演变	(12)
第三章 GAL 器件原理及有关技术	(23)
E ² CMOS 可编程逻辑	(23)
GAL 器件的通用结构	(25)
可控硅自锁效应及其消除	(33)
电子标签的使用	(36)
寄存器预加载特性概述	(38)
GAL 器件的功率估算	(40)
GAL 器件的可测试优点	(45)
将 PAL 的模式复制到 GAL	(49)
第四章 E ² CMOS 单元的工艺、电路特性及物理机理	(54)
E ² CMOS 器件工艺和电路性能	(54)
E ² CMOS 单元的工艺和物理模型	(61)
第五章 GAL 器件开发工具	(69)
硬件和软件工具	(69)
设计过程	(73)
举例: 两层电梯控制器	(77)
第六章 GAL 器件的应用	(91)
基本逻辑门	(91)
基本触发器	(99)
移位寄存器	(105)
四位可逆计数器	(109)
七位计数器	(113)
存储器地址译码器	(118)
环形移位寄存器	(122)
四个四输入多路开关	(129)

八一三优先级编码器	(134)
命令译码器	(140)
带有等待状态发生器的译码器	(141)
总线仲裁器	(146)
四位级联加法器	(149)
时钟展宽电路	(151)
双通道动态 RAM 控制器	(151)
三层电梯控制器	(161)
第七章 GAL 器件的隐含成本优势	(187)
取消入库质量保证的检验手续	(187)
简化库存管理	(188)
使用 PLD 成本分析	(191)
第八章 GAL 器件技术规范	(194)
GAL 16V8 系列与 GAL 20V8 系列共同的技术特点	(195)
GAL 16V8 系列与 GAL 20V8 系列的技术规范	(202)
GAL 16V8 系列和 GAL 20V8 系列其它技术规范	(206)
GAL 39V8 与 ispGAL 16V8 技术规范	(225)
第九章 附 录	(233)
附录一 GAL 16V8 器件子结构图	(233)
附录二 GAL 20V8 器件子结构图	(235)
附录三 封装外形图	(237)
附录四 GAL 开发工具一览表	(242)
附录五 软件开发工具专营公司	(243)
附录六 GAL 器件产品型号说明	(244)
附录七 GAL 器件和 PAL 器件速度/功率比较表	(246)

第一章 GAL 器件概要

1986年以来,通用可重编程逻辑阵列(GAL)器件几乎风靡整个可编程逻辑器件(PLD)市场。GAL以其高性能、高可靠性、可擦除及输出逻辑结构可组态的特性,博得了广大用户的偏爱。就历史而言,GAL是在其它PLD器件的基础上发展起来的。但是,GAL不愧为目前最理想的PLD逻辑芯片。

GAL是一种理想的器件,这可从以下四点看出:

1. GAL是采用高速电可擦CMOS(E²CMOS)工艺制造的,它可最大程度地提供可测试功能以及生产工艺质量。另外,可随时擦除,最适合于样机的研制。

2. GAL芯片的速度不低于任何其他TTL可编程逻辑芯片的速度。

3. GAL芯片具有CMOS的低功耗特性。

4. GAL芯片具有输出逻辑宏单元(OLMC),使用户能够按需要对输出组态。

由于GAL芯片兼备以上功能,因此它终将取代TTL/74HC组合逻辑电路,取代低密度门阵列和所有其他可编程逻辑芯片。GAL芯片非常有利于降低系统造价,减小成品体积和功耗,还具有更高的可靠性,并能大大简化系统设计。

为何要采用可编程逻辑?

随着设计方法的不断完善,现在不仅需要简化设计过程,而且越来越需要降低系统的体积和成本,并提高系统的可靠性。正是为此目的,导致了第一个可编程逻辑器件的开发。事实上,可编程逻辑的研究已改变了系统设计的方法,因为,它能给设计师提供一种简便的工具来满足各种要求。可编程逻辑最适于简化系统设计,因为设计师能够借此来实现各种正确的逻辑功能;另外还可减小体积,降低成本,这是因为它比原来的中小规模集成电路具有更大的功能集成度。当然,由于简化了设计,减少了器件的数量,可靠性也得到极大的改善。

简化系统设计

可编程逻辑是理想的设计工具,因为它可由用户来精确规定芯片要执行的某一功能或是多种功能。而一般的中小规模集成电路要做到这一点,就需要把多个功能已预先规定的芯片进行搭配、连接,这是一个很繁琐的过程,中小规模的芯片往往得不到充分利用,且芯片的连接还有个极重要的线路布局问题。可编程逻辑芯片是由用户来定义各种功能;这样就极便于满足各种使用要求和最有效地利用芯片,从而减少了芯片的数量,大大简化布线过程;因此,不仅在构思阶段,而且在实施阶段,都能快速地进行设计。

功能集成度高

由于可编程逻辑芯片比中小规模集成芯片的功能集成度更高，这使它成为降低成本和减小体积的理想设计工具。所谓功能集成度是指在给定的体积内可集成的逻辑功能的数目。可编程逻辑芯片一般可替代4~12个中小规模集成芯片，具有更高的利用率。这些因素使得芯片数量减少，电路板的面积与数量也相应减少，结果就减少了整个系统的体积，自然就导致了研制费用的降低。更重要的是，使得生产成本降低。

可靠性高

芯片、电路板的数量减少，系统的体积减小，这不仅能降低成本，而且还提高了系统的可靠性与稳定性。根据调查统计表明，有较高集成度的系统，例如由可编程逻辑设计的系统，比那些由低集成度的标准芯片组成的同样系统的可靠性要高得多，这是因为出错的因素减少了，潜在出错因素的器件减少了，电路板的连接线减少了，成品的生产变得容易了。所有这一切都提高了可靠性，使设计师受益非浅。

系统设计师总是希望自己的设计工作尽可能简单，希望系统造价最低，可靠性最高。可编程逻辑的应运而生，是解决种种设计难题的理想选择。

GAL 器件的概念

E²C MOS——理想的工艺。

E²C MOS、UVC MOS 和双极型这三种主要可行工艺中，GAL 选定 E²C MOS 的原因是：可测试功能、质量高、速度快、低功耗以及在研制样机时可即时擦除和即时改错。

可测试功能

E²C MOS 在工艺竞争中的最大优势是它的可测试功能。制造厂利用非常快的擦除(50ms)功能，可对各种器件反复编程和擦除，以直接测试包括 AC、DC 功能在内的各种特性，保证程序和功能 100% 地满足用户要求，而无须再改制电路板。正如下面所要讨论的，多种竞争着的工艺都要解决测试方面的重大问题。

低功耗

这种工艺的另一个优点是 CMOS 的低功耗。这使用户直接受益：降低了系统功耗，可靠性高，运行系统的温度低，高性能持久。由于芯片上较低的结温和功耗，CMOS 低功耗就允许较高功能集成度的电路设计。因为较高的功能集成度意味着在系统中进一步减少芯片的数量、应用更小的电路板，所以这会使用户受益的。

高速度

另外一个优点是 E²C MOS 工艺具有极高的速度——其速度可以与除 ECL 电路外

任何工艺生产的任一器件的速度相比。

样机研制和修改错误

最后要说的是，E²CMOS 给用户提供了即时可擦功能，而不必另外处理或使用专用设备，是研制样机的理想产品，使设计的进行可随时修改而不必浪费器件和等待时间。在制造中，即时擦除功能对于改变阵列模式和修改错误来说都有莫大的便利之处。假设一个GAL器件在设计中明显发生错误，则其修改过程是很简便的，并且无须等待时间和浪费器件。只要把出错的器件放入编程器中重新编程即可。其他器件则没有这种功能。

其他工艺简介

下面是关于与E²CMOS 竞争的工艺——即双极型和UVC MOS 工艺和E²CMOS 方案的对比：

双极型

双极型工艺是可编程逻辑器件最先采用的。虽然速度高，但其功耗大，这不仅要加大系统的电源，并且要冷却，这就限制了电路的集成度。

该工艺的另一个缺点是其编程的一次性，对该器件完全测试是不可能的，而且制造商必须利用测试行与测试列来模拟和推断器件的性能，这是由于熔丝阵列不能在编程之前进行测试。由于不能完全测试，可能会引起编程在某个不良熔丝上失效。同时，由于这种器件只能编程一次，所以在样机研制发生错误后就不能再被使用了。任何编程出错的器件都必须报废。

UVC MOS

UVC MOS 虽然消除了一些双极型方案的缺点，但其本身就存在着若干缺点：该工艺虽然需要的功耗相当低，同时也具有可擦除功能，但是它的速度很慢，擦除过程麻烦。因为“熔丝”阵列可擦除和由制造商的程序编程，所以可测试性比双极性好，有直接的测试功能。但问题在于要擦除UVC MOS 器件，必须把该器件放在紫外线照射之下，擦除需要的时间很长（20分钟），这对生产过程而言是很麻烦的工序。由于时间因素，测试和擦除只能进行一次；这种测试与其说是功能的测试，倒不如说仅仅是一种折衷而已。

另外，这种器件必须封装在昂贵带窗口的外壳内，以使用户能够擦除。同时，这种擦除方法很费时间，再加上要紫外线连续照射就显得很麻烦。由于价格方面的原因，UVC MOS PLD 制造商给用户不带窗口外壳的器件，因其不能被擦除，所以封装后不能完全测试，当然用户也无法擦除。这些原因极大地降低了用户对这种产品的需求量。

GAL 器件的优点

GAL 器件是理想的可编程逻辑器件，正如其名称所说明的，它的结构是通用的。阵列输出采用宏单元法，使用户能对每个输出单元规定其组态和功能，使用户最便利的

是，可在每一个输出端上随意实现任何所要求的功能和结构，这不仅是制造过程，也是设计过程的优点。

设计优点

早期的可编程逻辑器件使用户能够定义功能，但不允许预先确定输出组态。把GAL器件与固定输出组态可编程逻辑器件进行对比，就如同把固定组态可编程逻辑器件与中小规模集成电路对比一样。对简化系统设计来说，GAL系列是新一代产品。当用户要设计最适合于某种要求的组态时，GAL系列可编程功能正好能使他们如愿以偿。

制造中的优点

由单一器件代替多种器件的方法大大地简化了生产过程，和不得不设计、生产大量不同种类不同组态的器件相对比，只生产一种可编程GAL器件的方法既省钱又避免了繁重的体力和脑力劳动。由于设计过程简化，生产过程也相当便利。可编程的GAL器件也可以减少由于缺乏原材料而停产的危险性。可编程GAL的另外一个优点是减少原材料的浪费和使保证质量的产品跟踪容易进行。

GAL 器件开发工具

用户在采用该产品时无须配备专用开发工具。因为目前的工业标准开发工具完全能支持GAL器件。用GAL产品进行设计时所需要的设备为两大类：

1. 可编程逻辑开发软件；
2. 编程器。

GAL技术的开发需要通用开发软件。一些器件制造商只能提供支持本厂生产的器件的开发软件包。其缺点是用户必须在每个制造商生产出的新器件时，购买其软件包。而通用软件包，如关于Data I/O公司的ABEL以及Assisted Tech公司的CUPL软件包，则为各种不同的可编程逻辑器件提供了开发支持，并且通过定期修改，用户可对所有制造商生产的各种可编程逻辑器件进行开发。由于这些原因，GAL制造商虽然不提供自己的软件，但是不断与软件开发商保持密切联系，可确保高性能的开发软件始终能用于支持GAL器件。

编程器

如同软件开发工具一样，自然要利用第三方开发的编程器。同样地由于上述通用性的缘故，GAL制造商不提供自己器件的编程器。不过，不断与通用编程器制造商保持密切联系，以确保高性能的编程器始终能用于支持GAL器件。

理想的组件

可编程逻辑器件是设计现代系统的理想器件。理想的设计需要有理想的产品来支持。

正因如此，才设计了GAL 器件，该理想器件——具有通用组态功能，是采用理想生产工艺，即E²CMOS 工艺制造出来的。

由于E²CMOS 可编程逻辑器件的不断开发和改进，带来了更高的速度、更大的灵活性和令人惊叹的新功能：比如用 isp GAL 系列器件有“在系统”（在线，实时）可编程功能，这是制造商敢对可编程逻辑器件和用户所保证的。

第二章 逻辑设计基础与 PLD器件的演变

近十年来，逻辑器件的新产品层出不穷：RTL，TTL，ECL，门阵列，PAL，现场可编程芯片，等等。其中，新产品中的佼佼者是非CMOS GAL器件。GAL器件不仅具有高性能的指标以及崭新的通用结构，而且能与目前现存的开发工具兼容。逻辑系统的设计师们面临着迅速发展的逻辑器件，也即面临着变革设计方法的问题，因为一种新的器件的出现，意味着一种可能的新设计方法的来临。

本章将对使用GAL器件设计逻辑所需要的逻辑基础作一简单的回顾。对PLD器件比较熟悉的读者可以直接阅读本章的后半部的内容。在该处将较详尽地讨论PLD器件的逻辑结构，特别介绍了GAL器件在结构上与其它PLD器件的异同。

逻辑设计基础

数字逻辑的设计过程，事实上是对布尔代数的理解过程。本节仅介绍在PLD器件中实现基础逻辑的功能时所涉及到的布尔代数基础。那些希望更多地了解布尔代数的读者，请参见其它更好的参考书。

布尔代数

布尔理论来自于乔治·布尔和他于1854年出版的《思维规律研究》(An Investigation of the Laws of Thought)一书。布尔研究的状态为：“是”或“非”、“真”或“假”、“高”或“低”；而这些“黑”或“白”状态正是我们经常碰到的问题。

例如：让我们来看一个闹钟，它能早上把你叫醒，它可以被定义为如下的布尔功能：

a) 若闹钟没有设定，它不会闹。

b) 若闹钟已设定，且当时间与设定的相符，则在预定的时刻，它将鸣叫。

这一功能可用表2.1表示，这里0表示“假”或“断开”，1表示“真”或“导通”。

注意：这里的功能仅表示变量的组合，即不但“闹钟设定”，而且要与“时间相符”为真，则导致闹钟蜂鸣。

如上的基本功能可写成如下方程：

$$\text{声音} = \text{设定} * \text{时间相符}$$

这个“与”方程可用语言描述为仅当闹钟设定“与”时间相符时，闹钟才发生蜂鸣。式中“*”表示逻辑“与”的功能。

基本功能

布尔代数的取值通常为[真, 假], 一般可简写为[1, 0], 布尔算符仅对具有两个状态的变量进行运算, 所有布尔运算的结果由它们的两个状态中的一个来表示. 布尔代数的功能远远超过了它的简单表示方法, 在下节中读者将看到其重大意义.

前述方程用了布尔“与”(AND)功能: 三个基本的布尔功能是“与”(AND), “或”(OR)和“非”(或称之为“反”(NOT)), 这三类功能都可用晶体管电路实

表 2.1 闹钟功能表

闹钟设定	时间相符	蜂鸣
0	0	0
0	1	0
1	0	0
1	1	1

现, 例如“或”功能的电路实现如图2.1所示. 双极型工艺和CMOS工艺的差异并不影响“或”运算的结果, 无论其结果为逻辑1还是0. 逻辑“或”运算功能示于真值表(图2.2)第一行. 由图可见, 当A或B二者之一为真时, C也为真, 也即仅仅需要一个运算变量为真就导致结果为真. 若当两个运算变量皆为真时, 其结果自然为真(请记住这里指的是逻辑功能, 而不是算术功能). “与”运算与上相似, 但所有的运算变

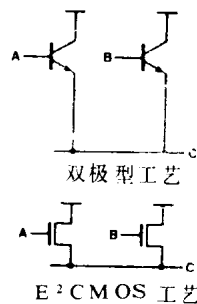


图 2.1 “或”门电路

量为真时, 其结果才为真, 也即, 仅当A和B两者均为真时, C才为真.

“非”运算是三种运算中最简单的一种. 在复杂数据运算中它起基本的作用, “非”的基本功能是简单的反相: 当输入为真时, 输出为假; 当输入为假时, 输出为真.

当演算方程时, 三种布尔运算应遵守一定的次序或优先权: 括号中的功能总是首先




逻辑符号	真值表	方程	功能	浮点运算															
	<table border="1"> <tr><td>A</td><td>B</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	C	0	0	0	0	1	1	1	0	1	1	1	1	$C = A + B$	与	“+”
A	B	C																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
	<table border="1"> <tr><td>A</td><td>B</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	C	0	0	0	0	1	0	1	0	0	1	1	1	$C = A \text{ 和 } B$ $C = A \cdot B$ $C = AB$	或	“·”或“*”
A	B	C																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
	<table border="1"> <tr><td>A</td><td>C</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	C	0	1	1	0	$C = \bar{A}$ $C = \neg A$	非	“/”或“-”									
A	C																		
0	1																		
1	0																		

图2.2 布尔运算

计算，接着是非运算，其后是与运算，最后是或运算。作为一个实例，我们来考察如下方程：

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

$$(2) D = \overline{(A \cdot B)}$$

在第一种情况下，在计算“或”功能之前，首先分别对A值和B值求反，但在第二种情况中，首先应进行括号中的“或”运算，然后对该结果求“反”。

对“与”和“或”运算，布尔代数也有常用的结合律和交换律，利用这两条基本定律，常可使方程达到扩展或简化的目的。

结合律规定：相同的算符可任意组合运算，而不必关心括号在何处：

$$(A + B) + C = A + (B + C)$$

该定律需要算符相同，因此下式是不相等的：

$$(A + B) * C \neq A + (B * C)$$

交换律规定：对相同的算符，可任意交换变量的次序，如：

$$A * B = B * A$$

基本理论

布尔代数理论是建筑在几条公理的基础上的，关于这些公理的证明和分析，因篇幅所限，不予讨论，读者如感兴趣，可参考其它较为详尽的资料。

目前，最先进的PLD软件开发工具已可以自动简化布尔方程，所以，我们对布尔代数方程的演算、简化和扩展规则进行一些简略讨论。

公理

这些公理于1904年由亨庭顿提出：

1. 加法和乘法的同一性原理

$$(a) \quad X + 0 = X$$

$$(b) \quad X * 1 = X$$

2. 交换律:

$$(a) \quad X + Y = Y + X,$$

$$(b) \quad X * Y = Y * X.$$

3. 分配律:

$$(a) \quad X + (Y * Z) = (X + Y) * (X + Z),$$

$$(b) \quad X * (Y + Z) = (X * Y) + (X * Z).$$

4. 互补律:

$$(a) \quad X + \bar{X} = 1,$$

$$(b) \quad X * \bar{X} = 0.$$

注意, 每个公理可表示为两个等式, 事实上是“与”和“或”两个基本逻辑运算的反演.

对偶规则对“与”和“或”运算有类似的结果. 对偶规则可用来产生上述公理的第二部分, 即对逻辑表达式按以下规定进行代换, 则代换后的逻辑表达式自然成立. 对偶规则代数方法规定如下:

(a) 用假来取代所有的真;

(b) 用真来取代所有的假;

(c) 用“或”来取代“与”;

(d) 用“与”来取代“或”.

定理

由公理和对偶规则导出的若干定理, 可以用来简化逻辑方程.

1) 重叠律:

$$(a) \quad X * X = X,$$

$$(b) \quad X + X = X.$$

2) 0-1律:

$$(a) \quad X * 0 = 0,$$

$$(b) \quad X + 1 = 1,$$

$$(c) \quad \bar{0} = 1,$$

$$(d) \quad \bar{1} = 0.$$

3) 吸收律:

$$(a) \quad X * (X + Y) = X,$$

$$(b) \quad X + (X * Y) = X.$$

4) 结合律:

$$(a) \quad X + (Y + Z) = (X + Y) + Z,$$

$$(b) \quad X * (Y * Z) = (X * Y) * Z.$$

5) 若 $X * Y = Y$ 且 $X + Y = Y$, 那么, $X = Y$.

6) 复原律:

$$(\overline{\overline{X}}) = X.$$

7) 德·摩尔根定律

$$(a) \quad (\overline{X + Y}) = \overline{X} * \overline{Y},$$

$$(b) \quad (\overline{X * Y}) = \overline{X} + \overline{Y}.$$

真值表, 卡诺图

真值表可用来描述逻辑功能. 它能向某些开发软件提供资料和设计逻辑功能, 是逻辑设计领域公认的强有力工具.

列出所有输入变量可能的组合状态, 并列出的具有一定逻辑功能的输出, 就形成了所谓真值表. 真值表的输入输出变量是以二进制来表示的. 真值表的长度是输入变量数的指数函数, 两输入变量的真值表长度是 $2^2 = 4$, 而三输入功能表长度为 $2^3 = 8$.

卡诺图是帮助人们简化逻辑方程的强有力工具, 利用卡诺图可以把任何逻辑函数简化为“乘积的和”或“和的乘积”的形式; 而利用该两形式, 很容易把上述逻辑函数所规定的功能转换成PLD逻辑结构图的形式. 我们在下面将详细讨论这两种形式.

因为使用可编程逻辑器件并不需要彻底理解卡诺图 (特别是当前开发的软件工具将自动地精确地简化输入方程), 故这一课题在此不作讨论.

而较早期的以汇编语言为基础的软件包, 需要输入最简的逻辑方程. 如读者只有此类工具可利用, 则充分熟悉卡诺图简化技术是十分必要的.

逻辑方程

在使用编程器对PLD器件编程时, 需要用专门的数据格式来定义器件的功能, 设计者或那些支持开发的软件包的任务即是把通常的逻辑定义形式转换成PLD器件能接受的专门数据格式. “和的积”的逻辑形式 (POS) 可用来描述任何组合逻辑的功能. 这一个两级的逻辑形式是由逻辑“或”和逻辑“与”组成. 如

$$(1) \quad Y = a * (c + d) + b * c + b * d,$$

能简化为 (2) $Y = a * (c + d) + b * (c + d)$, 即 (3) $y = (a + b) * (c + d)$, 这是一个“和的积”逻辑.

逻辑表达式通常可以表示为“和的积”形式, 也可表示为等效的“积的和”形式; 而基本的PLD阵列内部连结即是这种“积的和”的形式.

这种“积的和”形式包含了几个“与”逻辑和“或”逻辑, 方程 1 也可简化成“积的和”的形式: