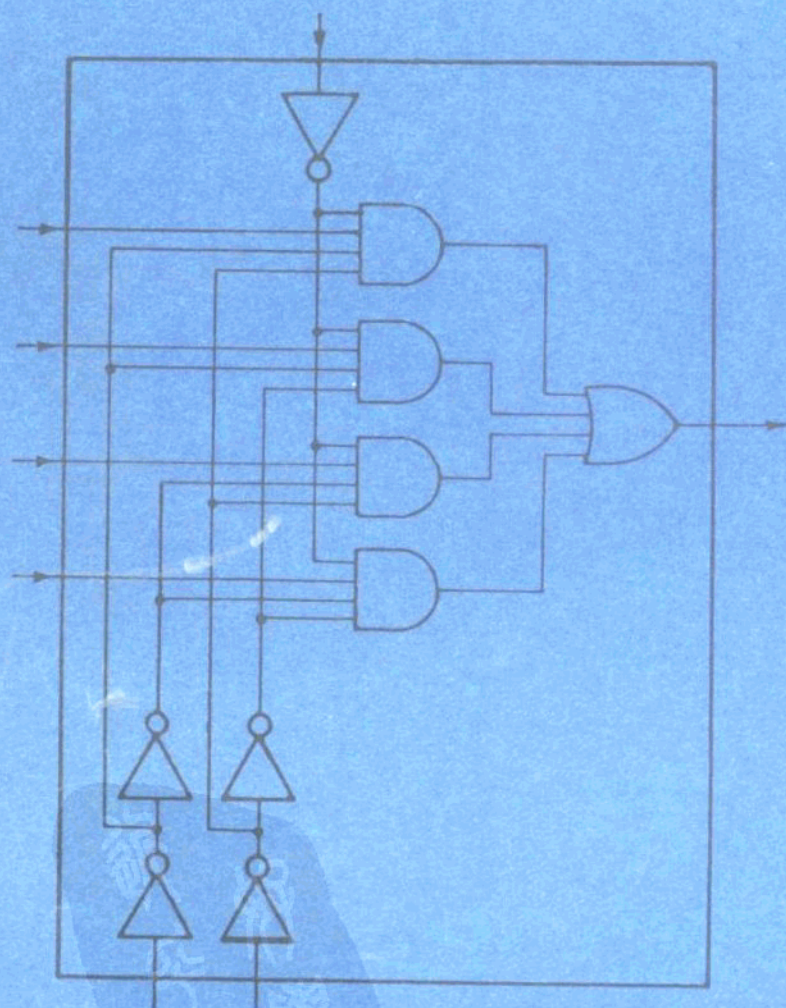


高等学校教材

数字系统 计算机辅助设计

清华大学

薛宏熙
边计年 编著
赵致格



海洋出版社

72
1

序

本书是为高等学校计算机、电子工程等各专业的高年级学生和研究生编著的教科书，着重介绍数字系统设计自动化各个领域的基础知识和发展状况。由于超大规模集成电路（VLSI）技术的迅速发展，一个 32 位的 CPU，或者一个几十万门规模的专用处理器可以制作在一个集成电路芯片上。因此，要人工设计一个复杂的数字系统几乎是一件不可能的事。设计自动化（Design Automation）或计算机辅助设计（Computer Aided Design）技术是解决这个矛盾的唯一方法。

CAD 自计算机问世后便随之产生。最早使用 CAD 技术用于计算机设计是在五十年代美国的 STRETCH 工程，它试图用已运行的第一代电子管计算机来设计第二代的晶体管计算机。当时 CAD 只用于设计文档（如逻辑图）保存、检查和修改。计算机本身已走完了第一代、第二代和第三代的历程，进入了 VLSI 与计算机技术相结合的第四代，并正在向着第五代智能计算机和第六代神经元计算机发展。目前，单片 CPU 的速度可达 50 MIPS，4M 位单片存储器已在市场销售，大容量的光盘已投入使用，一个运算速度每秒几千万次、可作实时可视计算的 CAD 工作站，其售价只有上万美元。计算技术如此迅速的发展，其重要原因之一就是在计算机各设计阶段广泛使用了 CAD 技术。可以这样说，没有 CAD 就没有计算机的今天。反过来，计算机技术的发展也给 CAD 技术提供了坚实的基础。目前，已有大量的 CAD 系统用于计算机和集成电路的设计。从逻辑设计、电路设计、工程设计（包括印刷电路板和集成电路版图）、故障诊断和测试都有 CAD 工具可供使用。良好的图形系统和用户界面，丰富的 CAD 软件，加上高性能的网络系统，给用户设计数字系统带来了很大方便，使用户从繁重的脑力劳动中解放出来。

本书正是从九十年代计算机和集成电路发展的需要，介绍了数字系统设计自动化中主要几个领域：逻辑综合、行为级描述、寄存器级描述和模拟、门和功能块级描述和模拟、故障诊断、形式验证等。较全面地叙述了数字系统设计过程中 CAD 工具的使用。读者可以从中学到如何使用相应的 CAD 工具去帮助设计一个数字系统。但是，CAD 技术还远远满足不了计算机和集成电路发展的需要，新的算法，新的 CAD 工具不断被推出。为此，本书也介绍了一些近几年发展起来新的 CAD 领域，如硬件算法，行为级描述、形式验证、硅编译技术和智能 CAD。书中除了介绍一些国际上标准的和通用的描述语言及其基本算法外，还引入了作者近几年的研究成果。使得本书既有数字系统设计自动化基本原理、基本方法，又有近几年来新发展的研究领域；既有 CAD 工具使用方法，又有实现 CAD 系统的算法和软件技术。全书内容丰富，叙述深入浅出，文字简练，并有不少实例供读者参考。这是一本介绍数字系统设计自动化方面的很好的教科书和参考书。

清华大学计算机科学与技术系

洪先龙

1990 年 4 月 15 日

前 言

数字系统,包括数字电路和集成电路所构成的网络,其中也包括计算机本身,一直在日新月异地发展,遍布各行各业,甚至进入家庭和日常生活之中。VLSI 技术使得数字系统具有相当大的规模和复杂性,利用人工设计,已成为不可能的事情,必须借助于计算机的帮助。而计算机性能的飞速增长又恰好满足了这一需求。数字电路计算机辅助设计正是在这一背景下兴起的一门学科。

本书是在清华大学计算机系教材《数字系统设计自动化》(1987 年编)的基础上修改而成。本书宗旨兼顾讲清基本原理和反映先进水平两个方面。一部分内容以基本原理和实践为基础,作为本学科的基本内容;一部分内容尽量反映国际国内的新领域、新技术,为读者了解各领域研究动向,提供一些帮助。编者在编写时力求深入浅出,信息量大,满足各方面的需要。本书可作为高等学校计算机、电子工程等各专业高年级学生和研究生教材或参考书;也可供从事数字系统设计制造的科技人员、工程技术人员参考。

数字系统的设计大致可分为功能描述、综合、验证、制造、测试等五个阶段。本书着重对系统行为级到门级的几个重要领域加以介绍。第一章为数字系统计算机辅助设计内容的综述。第二章介绍了逻辑综合的基本原理和技术。第三、四、五章分别介绍了行为级、寄存器传输级、门和功能块级以及开关级的描述、模拟方法。第六章介绍了故障诊断,包括测试码生成算法和故障模拟。第七章对新兴的逻辑设计的形式验证领域的几种方法作了介绍。第八章则简要介绍了设计自动化的几个新技术领域的发展状况。除第一章为综述性质外,其他各章内容基本相互独立。在讲授本课程时,对各章内容可根据学时取舍。在学时有限的情况下,建议以第二、四、五、六各章为基本内容,并最好与上机实践相结合。

本书编写工作由清华大学计算机系下列同志承担:薛宏熙负责第一章至第四章;赵致格负责第五章;边计年负责第六章至第八章,以及第五章的开关级模拟部分。全书由洪先龙教授终审。

目前本学科发展迅速,技术水平不断提高,新领域不断出现,由于编者学术水平和实践经验所限,不能完全反映最新水平。特别是对某些新内容是一边学习一边编写的。有些技术术语往往有不同的用词,我们根据自己的理解和常见的用法加以选择,并在书末附有英汉名词对照表,以供参考。对书中的不当甚至错误之处,恳请读者批评指正。

编 者

1990 年 1 月

目 录

第一章 概 论

1.1 数字系统计算机辅助设计的主要领域	1
1.2 模拟技术	2
1.3 逻辑综合	4
1.4 测试与诊断	4
1.5 逻辑设计形式验证	5
1.6 工程实现	5
1.7 硬件描述语言	8
1.7.1 硬件描述语言的描述能力	9
1.7.2 硬件描述语言的分类	11
1.7.3 图形硬件描述语言	14
1.7.4 多级硬件描述语言	15
1.8 CAD 系统简介	18
参考文献	20

第二章 逻辑综合

2.1 引言	21
2.1.1 术语简介	21
2.1.2 布尔函数的立方体表示法	23
2.2 立方运算	25
2.2.1 基本概念	25
2.2.2 相交和包含判断的具体实现	31
2.2.3 锐积运算 (Sharp Product)	33
2.2.4 星积运算 (Star Product)	38
2.3 多输出函数与单输出函数的阵列变换	41
2.3.1 单输出函数的表示形式	42
2.3.2 阵列合并	43
2.3.3 阵列分离	43
2.4 单输出函数质立方体的计算	44
2.4.1 锐积求质立方体	44
2.4.2 迭代星积求质立方体	44
2.4.3 广义星积求质立方体	45

2.4.4 改进列表法求质立方体	48
2.5 单输出函数的综合	50
2.5.1 选拔法求函数的最小化覆盖	51
2.5.2 收缩算法求无冗余覆盖	54
2.6 多输出函数的综合	55
2.6.1 收缩算法求无冗余覆盖	55
2.6.2 选拔法求最小化覆盖	59
2.7 组合逻辑电路的变换	60
2.7.1 多级逻辑电路转化为二级逻辑电路	60
2.7.2 二级逻辑电路转化为多级逻辑电路	63
2.8 时序逻辑电路综合简介	73
2.9 实现锐积运算的程序实例	76
参考文献	80

第三章 并发语言和行为描述

3.1 行为描述	81
3.2 CE 语言简介	82
3.2.1 模块	83
3.2.2 进程和管程	84
3.2.3 进程之间的通讯	85
3.2.4 延迟时间的描述	89
3.2.5 函数 empty	90
3.2.6 变量的绝对地址	91
3.3 模拟实例	91
3.3.1 表决器	92
3.3.2 容错策略简介	96
3.3.3 行为描述和模拟	97
3.4 VHDL 中行为描述的特点	97
3.4.1 进程	98
3.4.2 顺序执行语句	99
3.4.3 决断信号与决断函数	100
3.4.4 并行执行语句	100
3.4.5 实例分析	101
参考文献	103

第四章 寄存器传输级描述和模拟

4.1 引言	104
--------------	-----

— 2 —

4.2 AHPL 语言简介	104
4.3 AHPL 的电路描述	106
4.3.1 数据类型	106
4.3.2 运算符	106
4.3.3 基本结构及描述	108
4.3.4 AHPL 标准函数	113
4.3.5 时序的精细调整	116
4.3.6 编写 AHPL 描述文件的某些规定	117
4.4 AHPL 通讯文件	118
4.5 应用举例	120
小结	128
参考文献	128

第五章 逻辑描述和模拟

5.1 逻辑模拟的目的	129
5.2 逻辑模拟过程	129
5.3 逻辑模拟模型	131
5.3.1 元件模型	131
5.3.2 元件的延迟时间	132
5.3.3 模拟信号的状态值	134
5.3.4 逻辑电平的强度	141
5.4 门级和功能块级硬件描述语言 GFHL	144
5.4.1 GFHL 概述	144
5.4.2 如何定义元件	147
5.4.3 子线路描述	155
5.4.4 逻辑电路网络的描述	161
5.4.5 外部激励信号波形的描述	163
5.4.6 模拟命令	164
5.5 交互式逻辑图输入系统	165
5.5.1 概述	166
5.5.2 符号库和字符库	166
5.5.3 交互逻辑图编辑软件的设计	167
5.6 逻辑模拟算法	168
5.6.1 编译方法和编排级数法	168
5.6.2 表驱动方法	169
5.6.3 提高模拟速度的方法	174
5.6.4 表驱动法的表格结构和调度算法	175
5.6.5 惯性延迟模型的模拟算法	180

5.7 逻辑模拟系统举例	181
5.8 开关级模拟	188
5.8.1 开关级模拟的对象与功能	188
5.8.2 MOSSIM 的强度比较算法	189
5.8.3 RSIM 的等效阻容网络算法	192
5.8.4 开关级电路的延迟计算	194
5.8.5 门级和开关级的混合模拟处理	195
参考文献	197

第六章 故障诊断

6.1 故障诊断的目的和基本方法	199
6.1.1 故障模型	199
6.1.2 测试与测试集	200
6.1.3 压缩测试集的方法	203
6.1.4 故障字典	209
6.1.5 测试过程	210
6.1.6 测试生成问题	211
6.2 组合电路测试生成算法	211
6.2.1 路径敏化法	211
6.2.2 D 算法	213
6.2.3 关键路径法	216
6.2.4 路向判决法	221
6.2.5 布尔差分法	222
6.2.6 其它故障模型的诊断	225
6.3 时序电路的测试生成	228
6.3.1 同步时序电路的迭代展开	229
6.3.2 扩展 D 算法	230
6.3.3 异步时序电路的迭代展开	233
6.4 功能块模型的测试生成	234
6.4.1 功能块模型	235
6.4.2 功能块测试生成举例	236
6.5 故障模拟	240
6.5.1 并行故障模拟	240
6.5.2 演绎故障模拟	241
6.5.3 同时故障模拟	243
6.6 可测试性设计	244
6.6.1 可测试性设计的一般措施	244
6.6.2 设置观察点与控制点	244

6.6.3 组合电路的可测试性电路结构	245
6.6.4 时序电路的扫描方式	248
参考文献	250

第七章 逻辑设计的形式验证

7.1 逻辑设计验证的目的和基本方法	252
7.1.1 设计和验证	252
7.1.2 逻辑模拟与验证的关系	252
7.1.3 形式验证的基本方法	253
7.2 基于符号处理的形式推理方法	254
7.2.1 电路的描述	254
7.2.2 公理系	255
7.2.3 FOL 定理证明系统	257
7.3 基于时态逻辑的验证	260
7.3.1 时态逻辑简介	260
7.3.2 用时态逻辑描述电路的时序关系	263
7.3.3 利用状态迁移表的验证方法	265
7.4 归纳断言法在逻辑验证中的应用	272
7.4.1 归纳断言法简介	272
7.4.2 一个寄存器传输语言及其公理定义	274
7.4.3 验证实例	277
7.5 提取行为表达式的验证方法	278
7.5.1 硬件描述语言 ISPB 简介	279
7.5.2 事件、历史序列和行为	280
7.5.3 行为表达式	281
7.5.4 由 ISPB 程序求行为表达式	283
7.5.5 用行为表达式进行验证	285
附录 7.1 Wagner 形式推理系统的硬件描述语言 BNF 范式	286
附录 7.2 Wagner 形式推理系统的公理式	287
附录 7.3 同步计数器验证过程	289
参考文献	293

第八章 设计自动化的几个新技术领域

8.1 硬件算法	295
8.1.1 硬件算法的概念	295
8.1.2 硬件算法与逻辑设计	296
8.1.3 硬件算法的分析与评价	301

8.2 硅编译器	302
8.2.1 硅编译器的概念	302
8.2.2 硅编译器的主要技术动向	303
8.2.3 几个硅编译器的例子	304
8.3 CAD 专用硬件	312
8.3.1 CAD 专用硬件的背景	312
8.3.2 硬件逻辑模拟器	313
8.3.3 硬件开关级模拟器	319
8.4 智能 CAD	324
8.4.1 智能 CAD 出现的背景	324
8.4.2 智能逻辑设计	325
参考文献	331
英汉名词对照表	333

第一章 概 论

1.1 数字系统计算机辅助设计的主要领域

计算机辅助设计 (Computer Aided Design, 以下简称 CAD) 技术近年来得到迅速发展, 各行各业都在使用 CAD 技术, 而数字系统 CAD 则是指利用已有的计算机系统所提供的 CAD 工具去设计新的数字系统或计算机。CAD 工具可以帮助设计者完成大量繁杂、重复性的工作, 而设计者则可集中精力创造新的数字系统, 从而极大地加快了计算机翻新的速度。可以这样讲, 假如今天还单纯用人力研制新型计算机, 那么很可能在它诞生之前就因落后而夭折。本书主要介绍数字系统 CAD 的算法、实现和应用。

从历史发展过程来看, 由于数字系统的复杂性和 CAD 工具当时的发展还不够充分, 数字系统的设计是分阶段进行的, 在不同的阶段分别由掌握相应专业知识的技术人员进行设计。这就要求前一阶段的工作成果 (输出) 成为后一阶段的工作的依据 (输入), 存在一个相互通讯的问题。

在总体设计阶段, 总体设计师根据预定的目标确定计算机的规模、速度、主要部件、总线结构以及指令系统等等。这里有两个主要问题: 一是如何知道所确定的系统结构是否满足预期的目标; 二是如何精确地描述所确定的系统结构, 使成为下一步工作的依据, 并且使参加这一任务的所有设计人员有共同的认识而没有误解。硬件描述语言就是根据以上要求而产生的。用它既可以精确而简明地描述计算机的系统结构, 又可以被计算机接受进行验证, 检验并修正所设计的系统结构使之满足预定的目标。

在逻辑设计阶段, 设计者要把系统设计阶段确定的各个部件具体化, 以标准的逻辑元件 (与门、与非门、或非门、寄存器、存储器等)、门阵列、标准单元等实现这些部件。

在工程实现阶段, 设计者要把逻辑设计的结果变成实体。所涉及的技术问题包括: 集成电路芯片的版图生成、测试码的生成以及印制电路板的布局布线、机箱底板布线等等。

数字系统的分阶段设计引起了不同级别的 CAD 工具, 各个级别的 CAD 工具都有相应的硬件描述语言, 整个设计过程就是把高层次的抽象描述不断细化为接近物理实现的低层次描述。为了减少不同层次的硬件描述语言相互转化所需的工作量, 近年来发展了多级硬件描述语言, 使同一硬件描述语言能够适应不同设计阶段的需要。

图 1.1 表示了数字系统设计和验证过程的层次结构关系, 它和硬件描述语言级别的划分大体一致。图中表示了两个过程: 一个是由高层次的描述经过验证并逐步向较低层次细化的过程; 另一个是由低层次的描述提取其较高层次的描述, 加以比较, 反向验证的过程。这里暂时把寄存器传输级描述当作最高的级别, 关于更高级别及硬件描述语言的讨论详见 1.7 节。

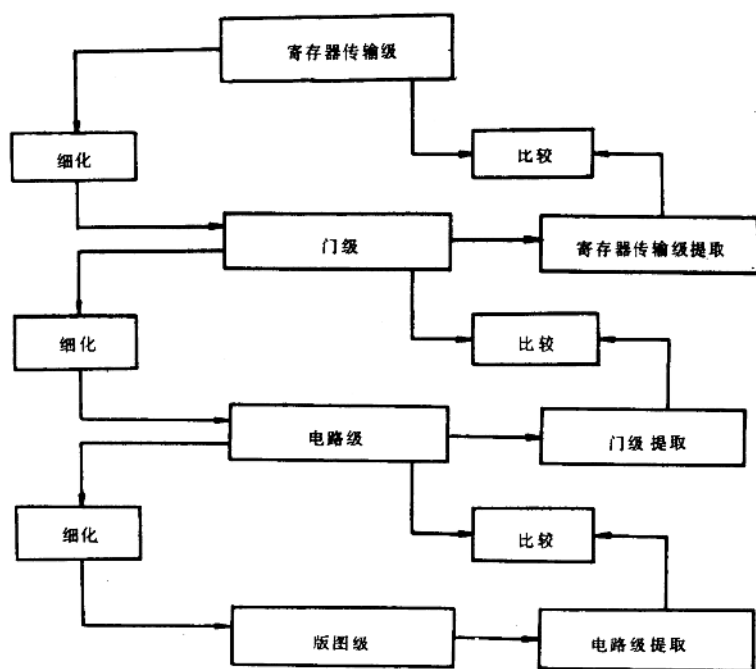


图 1.1 数字系统的设计和验证过程

1.2 模拟技术

所谓模拟技术,是指对实际数字系统加以抽象,提取其模型,将其输入计算机。然后将外部激励信号施加于此模型,通过观察模型在外部激励信号作用下的反应判断该数字系统是否实现了预期的功能。

模拟技术是当前数字系统验证的主要手段。它的局限性在于:模拟器的功能仅是表现在某一组外部激励信号作用下该数字系统的行为,至于加什么样的外部激励信号以及在该外部激励信号作用下系统的反应的正确与否,完全由设计者自己决定。

由此可知,设计者必须给模拟器提供以下信息:

- (1) 数字系统基本元件的功能特性。
- (2) 基本元件的互连关系或相互作用的关系。
- (3) 模拟过程所需的信息,如外部激励信号的名称和波形、观察点的名称以及如何表示被观察信号的命令等。

一般说来,上述信息是通过专门的硬件描述语言提供的。

· 寄存器传输级 RTL (Register Transfer Level)

七十年代中开始用 RT 硬件描述语言书写电路的用户描述文件,其基本元件是寄存器、存储器、总线、运算单元 ALU 等。此时,元件的延时特性、负载特性一般说来是未知的。主要用来表示电路的行为特性,也隐式表达电路的结构,能较为简练地、宏观地描

述系统。

· 门级 (Gate Level) 模拟技术

门级的基本元件是各种门及触发器。门级描述仅表达电路中基本元件的互连关系,即电路的结构信息。当 RTL 验证完成之后,应根据当时所采用的技术将 RTL 电路细化为门级元件或规模稍大的电路块。元件的延时特性、负载特性在门级模拟中应当有较为精细的描述。门级模拟用于验证门级电路的正确性,在门级模拟中,也许还要完成可测试性分析。

· 电路级 (Circuit Level)

电路级的基本元件是晶体管、电阻、电容等,电路级描述仅表达电路基本元件的互连关系。在门级验证完成之后,应将门级元件细化为电路级元件。为了验证电路的正确性,可使用电路模拟(分析)程序或时域模拟(分析)程序,以得到各点波形(表示信号间的相互作用关系)MOTIS-C 和 SPICE 分别是时域模拟和电路模拟的具有代表性的工具。

在同一计算机上分析同一问题,时域模拟程序 MOTIS-C 一般比电路模拟程序 SPICE-2 快两个数量级,如图 1.2 所示。图中纵轴 T 代表为计算一个元件所花费的 CPU 时间,横轴 N 代表该电路的节点数(表示该电路的规模),而逻辑模拟程序则要比 SPICE 快三个数量级左右。因此,希望有一种混合的模拟程序能把逻辑模拟和时域模拟结合起来使用,精度要求高的部分使用时域模拟,精度要求低的地方使用逻辑模拟。SPLICE 就是这样一种混合模拟程序。

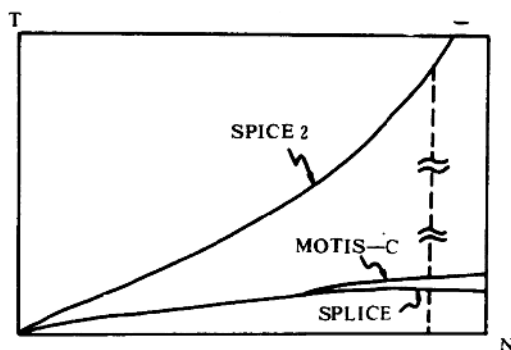


图 1.2 电路模拟所需 CPU 时间的比较

电路模拟程序和时域模拟程序的基本元件都是晶体管、电阻和电容。但是,时域模拟程序的晶体管模型按照 MOS 电路特点进一步简化,并采用了若干种逻辑门的宏元件,在计算时采用查表技术代替模型计算,在求解电路方程时也采取了一些简化措施。其结果是运算时间大大减少,而瞬态波形的精度略有降低,这两种计算所得波形的比较见图 1.3,误差 不甚明显。

根据电路描述可产生集成电路芯片的版图。根据版图和当时所采用的工艺技术,可提取晶体管的参数、寄生电容、引线电阻等寄生参数。根据版图所提取出来的电路必然和原先的电路级描述有所不同,有必要再次进行模拟验证。这就是图 1.1 中所表示的由低层次的描述提取其高层次的描述,再次进行模拟验证的过程。

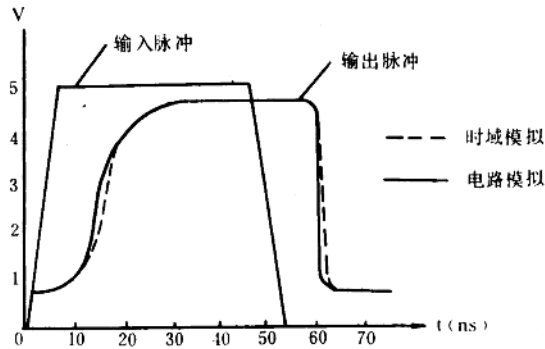


图 1.3 瞬态波形比较

1.3 逻辑综合

模拟或分析的方法是分析数字系统的某一方案是否正确,是否达到了预期的要求,其前提是已有了一个现成的方案。而综合则是分析的逆过程,其前提是给定应实现的逻辑功能和实现此电路所采用的技术,目标是给出一个合理的方案。

(1) 逻辑功能可由 RTL 语言或布尔方程或布尔函数的形式给出,本书的第二章是以覆盖表示的布尔函数给出。

(2) 所采用技术一般是指元件类型以及每种元件的约束条件和电路限制。例如本书第二章所采用的元件类型为与门、或门、非门。

(3) 所谓方案合理是指该方案的成本最低或接近最低。

逻辑综合的另一个内容是电路形式的变换。假如电路 A 已正确实现某一逻辑功能,由于技术条件的改变,例如门类型或元件的约束条件发生变化,则应由电路 A 导出逻辑功能等价而使用不同元件的电路 B。

绝大部分数字电路都是时序电路,但是逻辑设计过程大部分处理的问题都是组合逻辑电路的综合,此外,组合逻辑电路综合所采用的技术正是时序逻辑电路的基础,其技术也比较成熟和实用。因此,本书第二章重点介绍组合逻辑电路的综合。

在 VLSI 领域,基于结构化设计方法,规则电路比随机逻辑电路更为流行和实用。因此,在逻辑综合中,我们对可编程序逻辑阵列 PLA 投入更多的注意。

1.4 测试与诊断

每一件产品制成之后都需要经测试合格后才能成为成品。数字系统的测试对象一般是:集成电路芯片、印制电路板、部件或整机。本书第六章以集成电路芯片的测试为主要对象介绍测试的理论和方法。

随着 VLSI 工艺的进步,芯片内部元件的密度剧增,以致几乎没有空隙允许探针插入。因此,外部激励信号只能从输入引脚引入,观察点只能从输出引脚引出,这就增加了测试的

难度, 需要设法在外部激励信号的作用下, 把内部元件的故障“驱赶”到输出端, 以便能在输出引脚上观察到故障的存在。不幸的是, 并非每一个故障都能用这种方法在输出端表现出来。

对于印制电路板或部件来说, 如果在测试中发现故障, 不是简单的抛弃而是要设法维修。这就要求指出故障的位置和区域, 称为故障定位或故障诊断。

用于故障测试和诊断的 CAD 工具的主要任务是: 对于一个给定的数字系统, 能自动生成一个最小的测试集。利用该测试集能尽可能多地检测出该系统内的故障并指出故障的位置或区域。

1.5 逻辑设计形式验证

在层次化设计过程中, 一般采用“自顶向下”的设计方法。在较高层次的设计完成之后, 都要经过检验, 证明正确以后才进入较低层次的设计。1.2 节中介绍的模拟技术就是最为流行的方法之一。但是, 模拟技术存在以下缺点:

(1) 外加激励信号要由用户给出, 外加激励信号的时序波形决定了所能查出问题的多少。

(2) 输出结果乃是外加激励信号作用下该数字系统的反应, 并不表示正确与否。

(3) 分析输入、输出信号之间的关系, 判断该数字系统是否正确的责任落在用户本人的肩上。

(4) 因此, 受用户给定的外加激励信号质量的影响及用户判断能力的影响, 很难保证现存错误都能查出。

逻辑设计形式验证则不进行动作模拟; 而是用形式验证的方法验证逻辑设计的正确性:

(1) 对数字系统作结构描述, 这是验证的对象。

(2) 对数字系统作功能描述, 作为验证的目标。

验证的过程就是根据描述和一套公理系统证明该结构描述所实现的功能是否与相应的功能描述等价。

逻辑设计形式验证是七十年代中期开始的, 目前还处在发展阶段。本书第七章将介绍其基本理论和方法。

1.6 工程实现

逻辑模拟和验证基本上保证了逻辑设计的正确性。此后的工作如: 集成电路设计和制造、印制电路板的设计和制造、机柜底板的布线等都属于工程实现阶段的任务。这些工作的特点是工作量大、重复性强、极易出错。只有依靠计算机的帮助才能顺利完成。

现以集成电路的设计流程为例 (见图 1.4) 说明工程实现阶段 CAD 系统的作用和任务。从设计方法来看, 可分为“自顶向下”和“自底向上”两种:

(1) “自底向上”法是设计者建立单元库的过程。库中每一标准单元或功能块都经过器件设计、电路设计和版图设计, 并经模拟验证后存入单元库。

(2) “自顶向下”法是设计者先进行最高层次的系统设计, 而后逐级向下。在逻辑设计

和验证完成之后版图设计之前,可能需要进行逻辑划分。即当电路规模很大,整个电路在一个芯片内容纳不下时,把整个电路划分为若干模块,以便每个模块都能在单个芯片内实现。逻辑划分的要求是模块间相互连线最少。版图设计的主要任务是布局和布线:先把标准单元或功能块配置在芯片的适当位置上,然后进行布线,完成单元之间的互连。

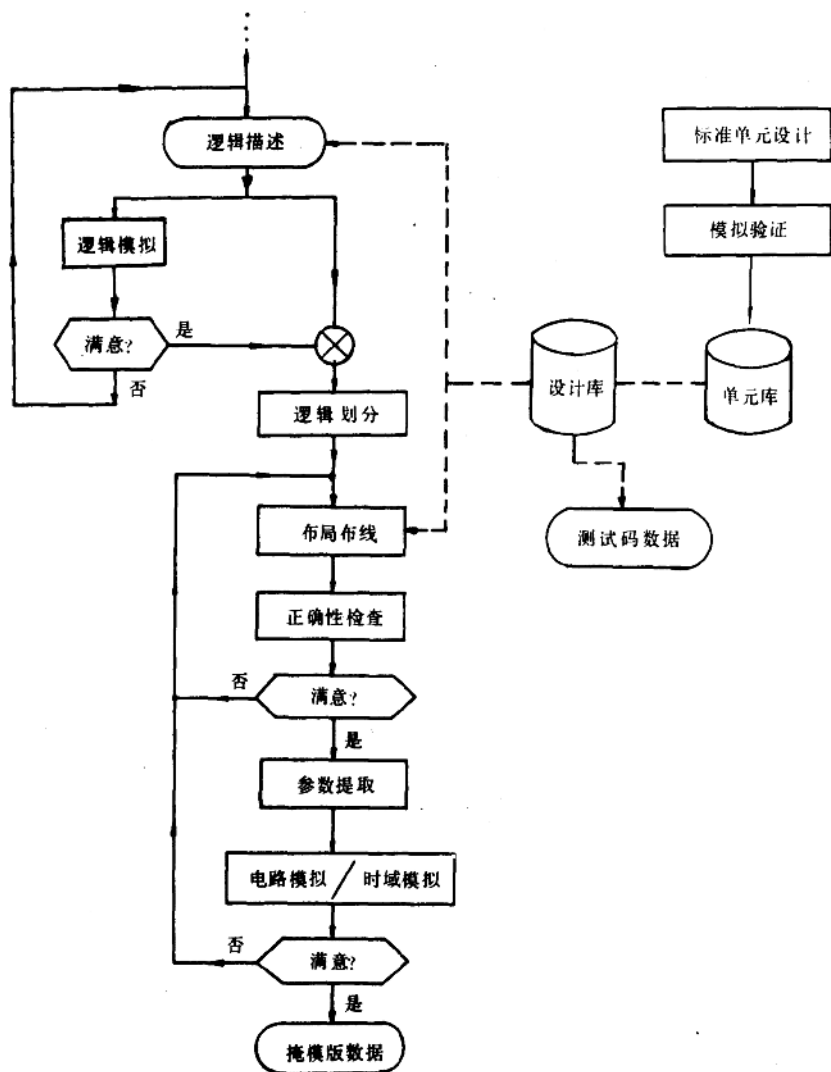


图 1.4 VLSI 设计流程

布局布线的正确性检查包括:设计规则检查和逻辑连通性检查两部分。设计规则是根据所采用的工艺,由设计者事先拟定的最小线宽、最小间距、周长、面积、覆盖、重叠等一系列规定,检查实际布线有无违背这些规定之处。逻辑连通性检查是检查版图中有无短

路、开路及冗余线段,布线结果是否和输入的网络表一致。

布局布线正确性检查完成之后,还要进行电性能检查,即根据版图和工艺提取晶体管参数、寄生电容、寄生电阻等参数,再经电路模拟作最后的验证。

图 1.4 中的设计库的作用是统一管理所有的数据。在层次化设计系统中,各个设计阶段是紧密联系的,其数据也是紧密联系的。例如,逻辑模拟和布局布线软件都需要网络连接信息,如果这些信息放在设计库中统一管理就会方便得多。由设计库统一管理各种数据的好处是:

- (1) 避免数据重复存放和数据之间的矛盾。
- (2) 数据的存放和管理与使用这些数据的应用程序相互独立。数据可被多个程序共享,其访问方式不变;而应用程序本身则可能随时作某种更改。
- (3) 提高处理数据的效率。

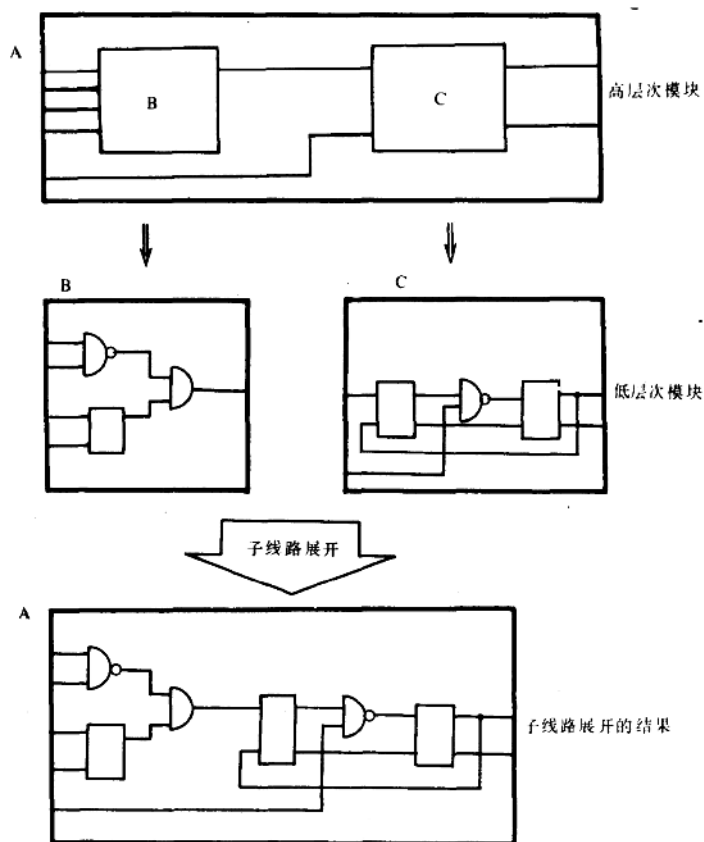


图 1.5 子线路展开

在一些实用的数字系统 CAD 系统中,往往还配有许多实用模块。例如,在 HIDE MAP 系统中有一个子电路展开器,它的任务是把高层次的模块自动展开为低层次的

模块,图 1.5 是实现子电路展开的一例子。转化命令指定高层次模块名和低层次号码(或低层次模块名),只要设计者给出的电路网络描述符合此层次化结构的要求,就可利用子电路展开器和数据库得到低层次的描述。

该系统中还有一个块划分器。在“自顶向下”的层次化设计中,模块的大小不会相等,在高层次中所划分的模块对低层次(例如布局布线)来说不一定合适。块划分器的作用是使模块重构,使其规模适当。

计算机的特长是记忆容量大和处理速度快,它能进行数据转换、模拟验证、资料处理等大量人所力不能及的工作。但是,创造性的工作、设计思想、原始方案还要由人来提出。此外,CAD 工具目前还不够完善。例如,在自动布线过程中,有可能出现两点之间连线遇到障碍而“不通”的情况,经过人工干预就可能解决。

一个新的研究方向是把人工智能技术用于数字系统 CAD 之中。遇到矛盾时求助于面向目标的算法,采取规则导向的搜索算法,把一般问题划分为容易求解的子问题,确定并解决各个子问题,从而得到最终解。

1.7 硬件描述语言

硬件描述语言(Hardware Description Language,简称 HDL)是人们对数字系统抽象化、模型化、形式化的产物,用以描述设计者当前最关心的最本质的特性。次要因素经常被忽略,以便突出主要因素。在不同的设计阶段,设计者所关心的主要因素是不同的,例如,逻辑设计阶段所使用的硬件描述语言应能描述逻辑元件的功能(与、或、非、传输等等)和参数(延时、扇入、扇出等等)。逻辑模拟应反映出信号在逻辑元件间的传输情况,必要时应能反映出因不同路径的延时不同而造成的竞争和冒险等,而对组成逻辑元件的晶体管、电阻、电容的参数以及信号的波形视作次要因素而加以忽略。与此类似,寄存器传输级描述语言注意力集中于描述信息如何在基本元件间的传输,完成哪些操作,它把逻辑元件的延时、扇入、扇出等都看作次要因素而忽略不计。

硬件描述语言可以是文字的或图形的或二者结合的,图形硬件描述语言(Graphic Hardware Description Language,简称 GHDL)用直观的图形描述一个硬件实体的结构,近年来得到迅速发展,它们的共同要求是:

(1) 简练而精确的描述,无二义性,有严格的语法关系,能被计算机所接受。如果再配有模拟器,可用于发现描述错误和设计错误。

(2) 可读性好,便于形成设计文件。这些设计文件可作为设计人员之间或用户和厂家之间的共同约定,准确地传递信息。

(3) 使用宏元件或子系统的方法,实现结构化分层的描述。层次越低,与硬件的物理实现、物理特性的对应关系就越密切。这些预先定义的宏元件或子系统可存放于 CAD 数据库中。

结构化和分层次的要求是基于计算机设计过程带有鲜明的层次特点,设计者首先在最高级别上描述系统的行为和结构,然后在某些约束条件下逐步细化,直到得到其物理实现描述(VLSI 版图以及印制电路板布线等)为止。较高一级的描述与较低一级的描述其实现的功能应该是相同的,而较低一级的描述与硬件实体的“距离”更近一些。