高等学校教材

数字系统的测试与容错

陈廷槐　曹泽翰　陈以农
陈廷槐　主编

东南大学出版社

高等学校教材

数字系统的测试与容错

陈廷槐　曹泽翰　陈以农

陈廷槐　主　编

东南大学出版社

内 容 简 介

本书由电子工业部高等学校《计算机与自动控制》教材编审委员会《计算机》教材编审组审定为全国高等院校工科计算机等专业的统编教材。

本书主要介绍容错计算领域中具有实用价值和发展前景的理论和技术，如测试的生成、可测试性设计、系统级诊断、容错技术、故障检测技术、可靠性评价和高可靠系统的设计方法等。全书共分十章，每章末均附有习题和参考文献。

本书可作为计算机专业研究生教材和供本科生及有关专业研究生选修之用，也可供有关人员参考。

数字系统的测试与容错

陈廷槐　曹泽翰　陈以农　编

陈廷槐　主编

东南大学出版社出版

南京四牌楼2号

江苏省新华书店发行　　阜宁印刷厂印刷

开本787×1092毫米 1/16 印张15.25字数378千字

1990年4月第1版　1990年4月第1次印刷

印数：1—3000册

ISBN 7-81023—255—X

TP·20　　定价：3.05元

责任编辑　朱经邦

出 版 说 明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校、中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978年至1985年，已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻“努力提高教学质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革试验的教材”的精神，我部所属的七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制订了1986～1990年的“七五”（第三轮）教材编审出版规划。列入规划的教材、实验教材、教学参考书等近400种选题。这批教材的评选推荐和编写工作由各编委会直接组织进行。

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的。广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处，希望使用教材的单位、广大教师和同学积极提出批评建议，共同为不断提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

前　言

本教材系按电子工业部的工科电子类专业教材1986～1990年编审出版规划，由电子工业部高等学校《计算机与自动控制》教材编审委员会《计算机》编审组征稿，推荐出版，责任编委谢希仁教授。

本教材由重庆大学担任主编，北京计算机学院梁业伟教授担任主审。

随着计算机的广泛使用，计算机自身计算的可靠性愈来愈受到人们的关注。计算机的测试（包含检测与诊断）与容错是提高计算可靠性的有效手段。测试是将故障暴露于执行计算任务之前，而容错则是在执行计算任务时使故障不影响计算结果的正确性，二者是相辅相成的。随着大规模和超大规模集成电路的出现，计算机辅助测试（CAT）也随之兴旺起来。至于容错方法和容错技术几乎被各种不同类型计算机或多或少地采用，以容错为主要设计目标的容错计算机也相继推出（例如国内熟知的STRATUS容错计算机），我国现在也开始着手研制容错计算机产品。因此，测试与容错计算已经成了计算机工作者必需具备的知识，一本以研究生为主要对象的教材就成了当前教学的必需。

1981年曾由国防工业出版社出版过《数字系统的诊断与容错》一书。该书早已脱销，加之材料有些过时，这次我们对全书全部重新编写，更名为《数字系统的测试与容错》，以示区别。

本课程的参考学时数为50学时。本书共含十章，除第一章对全书作基础介绍之外，二至六章为测试（包含诊断）部份，介绍测试的生成、可测试性设计和系统级诊断等各个方面；七至十章为容错部份，介绍容错技术、故障检测技术、可靠性评价和高可靠系统的设计方法等。使用本教材时请注意，该书的全部内容可供容错计算方向的研究生使用，如果其它方向的研究生用它来作50学时的选修课程时，则可略去注有“*”号的小节。至于本科生用它作为50学时选修课程时，则建议根据各校情况选择讲授其中的一个部份，并略去注有“**”号和“*”号的章节。

本教材由曹泽翰和陈廷槐编写第一、三章，曹泽翰编写第二、四、五章，陈廷槐编写第六章，陈以农编写第七至十章。陈廷槐统编全稿。由于编者水平有限，书中难免存在一些缺点和错误，殷切希望广大读者批评指正。

编　者

1989年2月9日

目　录

第一章　测试与容错的基本知识

本章介绍诊断与容错技术的产生、发展和应用，概述诊断与容错技术涉及的一些问题和基本概念，试图使读者对这一领域有概略的了解并有助于对以下各章的理解。

§1-1　诊断与容错技术的产生及发展

计算机自20世纪40年代后期问世以来，发展非常迅速。当前，计算机在速度上已突破10亿次／秒，内存容量达到100亿比特，外存容量接近无穷，加上网络化、微型化及各种软件的采用，使计算机的功能越来越强，使用日益普遍。现代空间技术，国防系统，各种科学研究，财政和商业的自动管理，交通控制系统以及生产过程的控制等对计算机的可靠性提出了更高的要求。例如用于宇宙飞船的自检自修计算机STAR(Self-Teting and Repairing)的设计指标要求可靠运行时间在十年以上。以上各种需要，使高可靠计算机成了计算机的一个重要研究课题。数字系统的诊断及容错正是在这些实际需要下应运而生的一门新兴学科，现已成为计算机科学中一个比较活跃的领域。

通过改进制造工艺来不断提高元件的可靠性只是提高系统可靠性的一个途径。用同样可靠程度的元件去组成系统时，系统越大，则出错率越高。因此，必须研究一种从系统方面提高可靠性的方法，这就是诊断和容错技术。所谓“诊断”，就是让故障尽快地暴露出来，以便在运行之前将它排除，它用于系统的设计、生产和运行的各个阶段。而所谓“容错”，则是采用各种增加备份资源的设计方法，使在元件有故障的情况下，系统的功能仍不受影响。然而增加过多的资源不仅会提高成本，还会降低系统的可靠性。因此，在同一时间系统能够容忍的故障数是有限的，这就需要通过“诊断”对故障的元件进行及时的修复。两种方法是相辅相成的。

早期的计算机主要是靠人工的参与查找故障，随着系统规模的增大，器件集成度的迅速提高，人工诊断就逐渐为机器诊断所代替，产生了系统的诊断理论和各种自动诊断和测试系统。自1959年，埃尔德雷德首先提出的一维通路敏化法到现在，已经有了若干产生组合电路测试码的系统的有效算法。在此基础上对时序电路、模块、子系统乃至系统的诊断都展开了研究，并取得了不少成果。随着LSI和VLSI的出现，进一步对电路的可测试性进行了理论研究，产生了各种可测试性设计的方法。

容错技术是提高系统可靠性的重要途径。早在1952年，冯·诺依曼就指出：人脑工作的可靠性是由结构上较低一级的冗余来保证的[1]。1956年，他又证明：（通过容错技术）可以用较不可靠的器件组成可靠的系统[2]。早期的实用计算机中已经采用了某些容错技术来减缓器件不可靠造成的影响。60年代后期，由于航天、航空等应用的高可靠性要求，容错技术才真正开始受到人们的重视，并逐步显示出巨大的作用。

随着计算机的发展和应用的推广，容错技术显得越重要的原因在于：计算机从具有良好

条件的机房迁到恶劣的实际应用环境，容易出现错误；随着计算机系统复杂性的增加和工作速度的提高，出错概率随之增加；计算机走向社会，使社会对计算机的依赖越来越多，这不仅会导致更多操作上的错误，而且由计算机出现的错误可能造成重大经济损失和严重后果。以上这些都要求对计算机系统进行精心的可靠性设计。

到70年代，各种容错计算机相继出现，如加州理工学院等单位研制的STAR系统，斯坦福研究院研制的SIFT和FTMP系统，Intel公司开发的Intel 432容错计算机系列和Tanden公司推出的Tandem-Nostop系统。1982年，Stratus公司推出了Stratus容错计算机系列，问世后立即获得了广泛的用户，1982年销售额为5百万美元，1983年就达到2千万美元。IBM公司在容错计算机的冲击下，也开始生产容错计算机系列IBM System 88。

总的说来，目前诊断和容错的理论和技术均发展很快，研究工作非常活跃。除每年一次的专门讨论诊断与容错问题的国际容错计算会议(FTCS)外，其它如国际测试会议(ITC)，国际设计自动化会议（DAC）等都涉及大量诊断与容错方面的论文。我国已有了与上述国际会议相对应的全国性学术会议，近年来已取得不少成果。我国这方面的学者有了越来越多的研究成果在国际性刊物和会议上交流。随着计算机事业的发展，诊断与容错技术将向纵深与普及两个方面发展。

§ 1-2 测试技术概述

一、测试的一般模型

计算机从元件、部件到系统的制造乃至运行的过程中都可能发生故障，为了发现故障，必须进行测试。通常，对测试的结果有两种要求：一种只对被测电路中有或没有故障给出答复，这种测试叫**检测**（detection）；另一种测试叫**诊断**（diagnosis），它不仅要对电路中有无故障给出答复，还要指出故障的确切位置，即进行故障定位。与检测相比，诊断测试的难度要大得多。

如何对一个（数字）逻辑电路进行测试呢？一般用穷举法对组合电路进行对比测试的模型如图1-1所示。

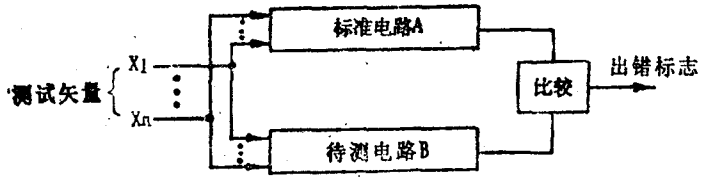

图1-1 对比测试模型

这里假设待测电路有n个输入引脚和m个输出引脚，我们分别称作该电路的**原始输入端**和**原始输出端**。测试时将2^n种输入矢量$\mathbf{X}=(x_1, x_2, \cdots, x_n)$依次加在标准电路$A$和待测电路$B$的原始输入端，两者原始输出端的输出响应经异或电路进行一系列的比较。若对所有的输入矢量$\mathbf{X}$均无出错标志，则说明B的逻辑功能是正确的。反之，如果出现错误标志，则说明B有故障。我们将出现出错标志时所对应的输入矢量称为该故障的**测试矢量**（或**测试码**），简称**测试**。

对于自动测试系统，其模型与图1-1类似。所不同的是由测试机器自动提供的输入矢量$\mathbf{X}$通常是事先计算得到的那些能暴露给定故障集合的所谓测试码；比较是由专门的电路或程序来完成；标准电路亦可用事先得到的正常响应的记录或对正常电路的程序模拟来代替，整个

过程可以快速地自动进行。这种让被测电路脱离正常的工作，由另外的测试设备提供测试激励并对其输出响应进行分析的测试方法称作**离线测试**（Off-Line Testing），主要用于检测和诊断逻辑电路中的永久性故障，它是诊断与测试通常采用的方法。如果在系统运行过程中以电路的正常工作的输入兼作测试激励，用专门的电路完成比较和判断，则称之为**在线测试**（On-Line Testing），在容错系统中为了及时地隔离或切换故障的模块，一般都包含有某些在线测试机构，如奇偶校验电路、多数表决器等。在线测试不仅可检测永久性故障，也可检测许多间歇故障和瞬时故障。

二、故障与故障模型

1．故障

通常，当一个设计正确的元件、电路或系统表现出与设计规定的功能不一致时，人们便说该元件、电路或系统存在故障。我们将那些使元件或电路不能正常工作的物理上和化学上的因素称作**物理故障**，或者说该元件或电路存在缺陷（defect）。计算机的硬件在生产、组装、存放以及工作过程中都会产生缺陷，如氧化穿孔、腐蚀引起的失效，焊接时的短路和开路等。造成缺陷的原因是多种多样的，它们大都与各种制造工艺有关。

元件的缺陷常常会导致电路的逻辑功能发生变化，即出现逻辑上的故障，我们称之为**逻辑故障**，在不引起混淆的情况下**简称故障**（fault）。可以说，逻辑故障是某些缺陷的逻辑等效。例如，在制造过程中某与门的一个输入端由于缺陷形成开路，这一缺陷在逻辑上等效于该输入端永远取1值，或者说这一缺陷造成了固定为1的逻辑故障。当然，有的缺陷可以影响电路的某些性能参数但并不引起逻辑故障。

即使故障已经发生，并不一定就能暴露出来。就上面所说的与门输入固定为1的故障而言，如果电路的某输入矢量恰好使该与门输入端为1值，显然这个故障表现不出来。但是当另一输入矢量使该与门输入为0值且该与门的其它输入端均为1值，则故障与门的输出与正常与门的输出便会不同，出现差错（error），从而暴露出故障。因此，差错是故障在特定输入条件下的表现。如何寻找那些能暴露电路中各个故障的输入矢量的问题，即通常所说的**测试码生成问题**，它是诊断技术中研究的主要问题之一。

2．故障模型

集成电路器件和由它们组装成的电路板的规模都变得越来越大。测试信号仅能加至器件或电路板的原始输入端，要想检测出电路中的任何缺陷已变得十分困难乃至不可能。对组合电路来说，要检测导致逻辑功能改变的各种缺陷，必须穷举所有可能的输入矢量。以一个32位的加法器为例，由于共有65个原始输入，则需穷举2^{65}种不同的输入矢量，即使每毫微秒输入一种矢量也会超过1000年。如果测试大型时序电路，则必须验证所有可能的状态转移，其复杂性急剧增加。

计算机系统的设计和测试人员一般不去直接研究测试物理缺陷的方法，而是根据电路的结构去研究逻辑故障的测试。这是因为：物理缺陷随工艺而改变，而逻辑故障的研究具有相对的稳定性，由它产生的测试具有相对的普遍性；逻辑故障易于理解和求出测试，它的精确定位也有助于发现具体的物理缺陷。这样便可以针对一个给定的逻辑故障的集合去找出它们的测试，从而用较少的测试就能保证被测电路中不存在该故障集合中的任何故障。

现在的问题是如何确定那些应该测试的故障集合呢？我们曾经指出，逻辑故障是物理缺陷的逻辑等效，因此，给定的待测故障集合应能充分地反映实际缺陷的影响。可是当给出一

种实际的物理缺陷后要分析出等效的逻辑故障来并不都是轻而易举的事。有的缺陷可对应于多个逻辑故障，有的缺陷则很难找到恰当的对应故障。为了重点研究某些主要缺陷的影响和测试的方法，通常需要建立一定的故障模型。对于一个故障模型一般都有两方面的要求：一是模型反映实际的程度，即要尽可能如实地反映实际缺陷的影响；二是便于处理，可用于复杂的系统。这两种要求常常是互相矛盾的，因此应根据元件的不同制造工艺、系统的各个阶段（如设计、生产和运行等）以及系统的不同规模和应用，从不同的角度和不同的层次去建立故障模型。

（1）门级故障模型

门级故障模型假设故障出现在逻辑门的输入或输出引线上。

i)固定型故障模型　该故障模型假设逻辑门的输入或输出值永远固定不变，即与电路的原始输入值无关。当某导线的逻辑值固定为0时，则称该导线发生了**固定0故障**(stuck-at-0)，记作：s-*a*-0。若导线发生的是固定1故障记为s-*a*-1。如果假设电路中最多只有一条线存在固定型故障，则称之为单固定型故障模型。这种模型便于处理，能反映绝大部份物理缺陷的影响。在系统运行阶段，由于诊断测试足够频繁，单固定型故障模型是合理的。它是使用得最普遍的模型。

如果允许电路中一个或多个导线同时具有固定型故障，则称之为**多固定型故障模型**。这时由于任一导线都具有正常、固定0或固定1三种可能的状态。若电路中包括n条连线，该模型将包含3^n-1个故障，比起单固定型故障模型来其复杂性大大增加。但是，器件和系统的生产阶段的测试采用该模型还是必要的。

ii)桥接故障模型　该模型允许逻辑电路中任何连线之间的短接并假设短接处的故障值取其“线与”值或“线或”值（由制造工艺决定）。如果要考虑所有可能的桥接的组合，则可能发生的桥接故障数为

$$\sum_{i=2}^{n} C_n^i = 2^n - n - 1$$

式中n为逻辑电路中的连线数。这是一个大得惊人的数，显然是不实际的。实际上总是要对这个一般的桥接故障模型作出若干限制，例如只考虑任何两条连线之间的短接，或只考虑两条邻近线之间的短接等。Mei[3]已经证明：任何固定型单故障的测试集将检测任何一个门的两个或多个输入线之间的桥接故障以及导致反馈且反馈回路的反相次数为奇数的任何桥接故障。这就告诉我们，如果已经采用了单固定型故障模型，需要按桥接故障处理的故障数大大减少，可以简化桥接故障模型。

（2）晶体管级故障模型

虽然门级故障模型可以等效很多晶体管级的故障，甚至针对门级故障模型产生的测试还能检测很多其它的故障。但仍然还有不少的晶体管级的故障未能反映出来。特别是在MOS工艺的情况下，有些晶体管级的电路模块不能用简单的逻辑门去等价，从而导致对晶体管级的故障模型的研究。这对于故障有更精确的描述，使我们对故障的原因以及故障对大型电路的影响有更深入的了解。但与此同时也大大增加了处理的复杂性。然而，在功能模型中如能包含晶体管级故障的影响有可能便于处理非常复杂的系统。

i)固定型断开(stuck-open)故障模型　在CMOS电路中，某些故障的影响表现在某些输入矢量时导致逻辑门的输出呈现高阻状态，使得故障输出响应依赖于前一输入矢量时的输出

响应，这些故障使逻辑门呈现出类似存储元件的性质。

图1-2所示的CMOS或非门，由两个与电源V串联的P通道晶体管和两个并联到地的N通道晶体管串联而成。当输入A和B均为0时，两个P通道管导通，两个N通道管截止，导致输出为1。输入A或B为1，则相应的P通道管截止，相应的N通道管导通，输出为0。

现假设图中标有1的连线是断开的。当AB输入为00，01和11时，该故障对输出没有影响。但当AB输入为10时，正常输出为0，假设的故障出现时，因输出端的上面和下面的晶体管均不导通，使输出端为高阻状态，输出值维持前一输入矢量下的输出值。这样的故障无法用门级故障模型来描述。在图1-2中类似的故障如Q_3，Q_4晶体管以及它们输入引线的断开，Q_1，Q_2以及串联连接线5、6、7的断开。通常将CMOS电路中的这一类故障称之为固定型断开故障（stuck-open）。

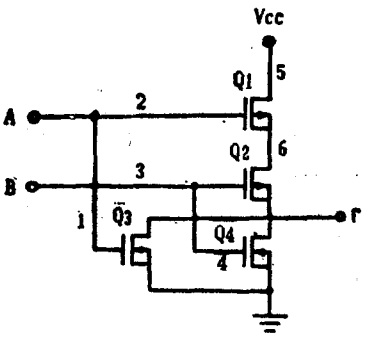

图1-2　CMOS或非门

ii)固定型导通故障（stuck-on）　CMOS电路中的另一类故障是导致电路中的某些晶体管永远导通，如图1-2中Q_1或Q_2的永远导通，从而使输出端的上下晶体管会出现同时导通的情况。这类故障被称之为固定型导通故障。它们能否被检测出来取决于那些固定导通的晶体管的内阻。Bechiera和Courtois[4]得出这样的结论：如果一测试矢量能测试固定型导通故障，它也能测试对应的固定型断开故障。反之则不成立。

iii)其它晶体管级故障模型　不同工艺的电路中的短路或开路故障往往产生不同的影响。如图1-3所示的MOS电路，正常输出函数$F=(A+C)(B+D)$，当出现图中α点开路故障时，输出函数$F_\alpha=A\cdot B+C\cdot D$。说明MOS电路中的某些故障会引起逻辑函数的改变。图1-4所示的DTL电路中的输入二级管D_1短路，使扇出分支线z由正常时的$z=x_1$变成$z=x_1x_2x_3$。

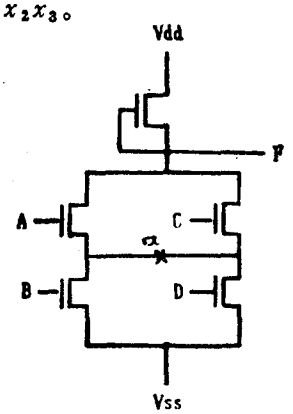

图1-3　MOS电路

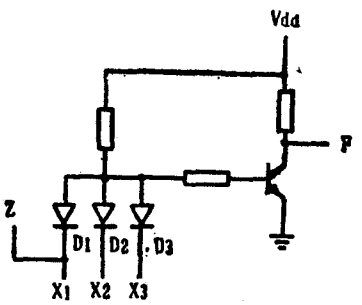

图1-4　DTL电路

以上这些故障都不能简单地用门级故障模型来描述，必要时都得单独考虑。Courtois[6]从电性能的研究中导出了一般化的晶体管级的短路和开路故障模型。

（3）功能级故障模型

一些复杂的集成电路器件或由它们构成的系统，作为用户，一般都不知道它们的详细的门级结构信息。有时即使知道门级结构，但对于复杂的大型时序电路，用门级模型处理起来

效率很低。这就迫使我们只能对电路的某些特定的功能进行测试或验证。这就需要针对不同功能的电路建立功能故障模型。模型的好坏对测试生成、测试效率均有重要影响。

例如，对随机存储器我们可以建立以下故障模型：

- 一个或多个单元固定为0或1
- 两个或多个单元相互耦合（反映了部份短路故障的影响）
- 某种数据模式或某些单元的状态转换引起别的存储单元的状态变化

显然，建立一种包含晶体管及其连线故障影响的功能块故障模型有可能成为处理复杂系统的有效方法。

以上列举的所有故障模型都是模拟的引起电路逻辑功能发生变化的永久性故障。致于影响电路各种电气参数的永久故障、瞬时故障和间歇故障，将通过直流参数测试，动态功能测试，以及在线测试等手段去完成。

三、故障的检测与诊断

测试输入仅能加至待测电路的原始输入端而不允许随意加至内部各点和从内部各点观察测试响应，是测试中存在不可区分故障和不可测故障的重要原因。

定义1-1 设逻辑电路N所实现的函数为f，电路N的两个故障α和β产生的故障函数为f_α和f_β，若$f_\alpha = f_\beta$，则称故障α和β**等价**。所有互相等价的故障构成一**等价类**。等价类中的故障是不可区分的。

我们对故障进行检测和诊断都是以等价类为单位。

定义1-2 若存在输入矢量x_i，使得故障的响应与正常响应不同，即$f_\alpha(x_i) \oplus f(x_i) = 1$，则称故障$\alpha$是**可检测的**。矢量$x_i$称为故障$\alpha$的测试或测试矢量。$\alpha$的所有测试构成故障$\alpha$的**完全测试集**。反之，若对所有输入矢量故障$\alpha$的响应都与正常响应一致，则称$\alpha$是**不可测故障**，并说该电路中存在冗余（redundancy）。

对于复杂的数字电路来说，要检测它的所有可能的故障是不现实的。实际上的故障检测和诊断都是针对某种故障模型所确定的一故障集进行的。

定义1-3 某电路的一测试集能检测电路中某故障集的所有可测故障，则称为该电路此故障集的**完全检测集**。其中所含测试数目最少的完全检测集称为**最小完全检测集**。

定义1-4 对于二故障α和β，若存在一矢量x_i，使得$f_\alpha(x_i) \neq f_\beta(x_i)$，则称故障$\alpha$与$\beta$是**可区分的**，$x_i$称为$\alpha$和$\beta$的区分测试。若一测试集能检测和区分一逻辑电路中某故障集的所有可测等价故障类，则称为该电路的此故障集的**完全诊断集**。含有最少测试数的完全诊断集称为**最小完全诊断集**。

上述的检测集和诊断集都是用作待测电路的输入激励信号，通常将它们连同正常测试响应一起称为**测试模式**（patterns）。

有三种生成测试模式的方法：人工生成；伪随机生成和算法生成。算法生成也称为确定性测试生成。它便于用计算机程序实现自动测试生成，是产生复杂电路测试模式的主要方法。

四、测试费用分析

随着集成工艺的发展，集成度迅速提高，使电路的硬件成本下降，却使测试的难度增大，导致测试费用的增加，减少测试费用成了降低产品成本的重要因素。

测试费用花在两个方面：一是测试模式的生成；二是对具体产品的测试。对于复杂电路来说，测试生成的费用是相当大的，采用不同的测试算法其费用相差也很大。一般说来，为了检测同样的故障集，生成的检测集越小所需的测试生成费用越高，但用于产品实测的时间越短，费用越少。由于测试生成是一次性的，具体测试是重复性的，故产量大的产品应生成尽可能小的检测集。反之，产量小的专用产品则应尽量降低测试生成费用。

影响测试费用的另一重要因素是针对什么样的故障模型生成测试？是否要求生成完全检测集？显然，精确性高的故障模型，对同一故障模型的故障检测率越高，测试生成费用越高。因此，有必要确定一个恰当的的故障检测率，以便选择合适的故障模型并首先为那些最可能发生的故障生成测试。为此，Williams[6]采用了以下的方程

$$DL = 1 - Y^{(1-T)} \qquad (1-1)$$

式中 DL表示缺陷度，即出厂产品中潜在有缺陷的产品的百分率。变量Y表示生产过程中的成品率，变量T表示缺陷的检测率。

例 1-1 如果要求检测所有的缺陷，则

$$T = 1 \qquad DL = 1 - Y^{(1-1)} = 0$$

如果生产过程中不存在有缺陷的器件，则

$$Y = 1 \qquad DL = 1 - 1^{(1-T)} = 0$$

显然，以上两种情况均不会有潜在缺陷的器件出厂。式（1-1）可改写成

$$T = 1 - \frac{\log(1-DL)}{\log Y} \qquad (1-2)$$

例 1-2 假设某集成电路的生产成品率为20%，我们要求出厂产品的缺陷度不超过2%，那么测试必须达到的故障检测率由式（1-2）得

$$T = 1 - \frac{\log(1-0.02)}{\log(0.2)} = 1 - 0.0126 = 0.9874$$

即要求能检测物理缺陷的98.74%。

这是一个相当高的要求，因为测试费用与故障的检测率之间不是线性关系。经验表明，二者之间的关系如图1-5所示。由图可见，当故障检测率接近100%，成本急剧增加。从式（1-2）可知，提高产品的成品率可以降低测试要求和测试费用。

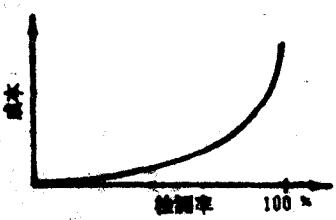

图 1-5 测试成本曲线

在系统的生产和运行维护过程中，还涉及故障的定位和替换有缺陷元件的费用。这是一件费事的工作，不仅是故障诊断比故障测试更困难，而且更换元件有可能产生新的故障，每次更换元件后都得重新测试。为了减少故障定位的费用，最有效的办法就是在系统制造过程中尽早地检测出有故障的元件。这意味着在制造过程中要设置若干层次的测试，在低层次的测试应该尽可能严格。

§1-3 容错技术概述

一、可靠性的概念

可靠性是表征系统正常运行能力的一个通用概念。可靠性可以通过各种可靠性参数来反映。

为了定义可靠性参数，需要首先定义造成系统不可靠的一些因素（和诊断技术相比，容错技术的讨论往往是在系统结构的更高层次上进行，须注意某些名词术语用在不同情况下的含义上的差别），它们是：

- 失效：硬件的物理改变。
- 故障：由于部件失效、环境干扰、操作失误或不正确的设计等在硬件或软件中造成的不正确状态。
- 差错：故障在程序或数据结构中的表现。
- 系统失效：故障在系统输出的表现。

本书主要讨论失效对系统可靠性的影响。下面介绍基本的可靠性参数。

通常情况下，如果一个系统能够正确执行预定的一组任务，则认为该系统是可用的，否则是不可用的。

系统在任意时刻 t 可用的概率 $A(t)$ 称为系统的**可用度**。可用度是系统的失效率和修复率的函数。实验表明：对于非冗余的系统，系统的失效率为某个常数 λ。如果设系统的修复率也为常数 μ，则可求得系统的可用度为

$$A(t)=\frac{\mu}{\lambda+\mu}+\frac{\lambda}{\lambda+\mu}e^{-(\lambda+\mu)t} \tag{1-3}$$

$A(t)$ 是可靠性的一个综合参数，由它可以导出若干具有特殊意义的参数。

当 $t\to\infty$ 时，若 $A(t)\to A$（常数），则称 A 为系统的**稳态可用度**。在无冗余的情况下，

$$A=\frac{\mu}{\lambda+\mu}\approx\frac{\text{可用时间}}{\text{可用时间}+\text{不可用时间}} \tag{1-4}$$

在某些实际应用环境下，联机修复是不可能的，因此 $\mu=0$。记 $A(t)_{\mu=0}=R(t)$。$R(t)$ 通常称为系统的**可靠度**。它是系统在时间区间〔0，t〕内连续正确运行的概率。在实时应用系统中，人们通常更关心系统的可靠度。

由 $R(t)$ 可导出另一个常见的可靠性参数：**平均无故障时间**MTTF。

$$\text{MTTF}\triangleq\int_0^\infty R(t)d(t) \tag{1-5}$$

在无冗余系统中：

$$R(t)=e^{-\lambda t}\qquad \text{MTTF}=\frac{1}{\lambda}$$

在冗余系统中，系统的失效率通常不再是常数。因此系统可靠性参数的计算要困难多。幸运的是，人们已经找到了系统化的方法来解决这一问题。目前常用的方法主要有两种[7]：