舰船现代化丛书

容错计算原理

主编 袁由光

哈尔滨工程大学出版社

舰船现代化丛书

容错计算原理

袁由光　主编

哈尔滨工程大学出版社

图书在版编目(CIP)数据

容错计算原理/袁由光主编.—哈尔滨:哈尔滨工程大学出版社,2006
ISBN 7-81073-790-2

Ⅰ.容… Ⅱ.袁… Ⅲ.容错技术-高等学校-教材 Ⅳ.TP302.8

中国版本图书馆 CIP 数据核字(2006)第 005585 号

内 容 简 介

本书从理论和工程实践两个方面系统地介绍了容错计算原理。全书共 11 章:首先讨论了容错计算的定义、容错技术的分类及发展状况;其次介绍了故障的表现及分布、编码技术、同步技术;然后在此基础上详细地介绍了故障发生后确保系统正常运行的各种容错技术;最后分析了两种典型的容错计算机结构。

本书可作为研究生、本科生的教材和参考书,也可作为从事可靠性研究的工程技术人员的参考书。

哈尔滨工程大学出版社出版发行
哈尔滨市东大直街 124 号
发行部电话:(0451)82519328 邮编:150001
新华书店经销
黑龙江省教育厅印刷厂印刷
*
开本 787mm×1 092mm 1/16 印张 13 字数 310 千字
2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷
印数:1—1 500 册
定价:28.00 元

前　言

随着计算机速度的提高、容量的增大和应用的推广，系统的可靠性问题越来越受到人们的重视。容错技术是构造高可靠系统的强有力的手段，也是当今十分活跃的一个研究领域。

本书从理论和工程实践两个方面系统地介绍了容错计算原理。第1章讨论了容错技术的定义、分类及发展状况；第2章介绍故障的表现及分布；第3章介绍编码技术，编码技术是一种提高可靠性的信息冗余技术，同时也是理解以后各章介绍的各种容错技术的基础；第4章介绍了同步技术；第5章、第6章、第7章分别介绍了故障检测与诊断技术、故障屏蔽技术以及动态冗余技术；第8章讨论了软件可靠性技术；第9章介绍了分布容错计算技术；第10章介绍了容错系统可靠性的评价；第11章分析了两种典型的容错计算机结构，这不仅可以深化读者对理论知识的理解，也可以对工程技术人员的实践提供指导。

本书可作为研究生、本科生的教材和参考书，也可作为从事可靠性研究的工程技术人员的参考书。

本书由袁由光担任主编，陈以农、周双娥、欧中红、戴新发、方铭参加了本书的编写。其中第2章、第8章和第10章由陈以农编写，第4章由戴新发编写，第9章由周双娥和方铭编写，周双娥对书稿的整理做了大量工作，第11章由欧中红编写，袁由光编写了其余章节并负责全书的统稿工作。在本书的出版过程中，得到了中国船舶重工集团公司709研究所、哈尔滨工程大学领导及相关部门的大力支持，在此一并致谢。

我们在编写本书的过程中，努力跟踪容错技术的新技术、新发展，力求反映当代该领域的最新成果，以保持本书的先进性和实用性。由于作者水平有限，错误之处在所难免，恳请读者批评指正。

编　者

2005年于武汉

目　录

第1章　绪　　论

采用容错和避错两种技术可提高数字系统的可靠性。尤其是容错技术，它是构造高可靠系统的有力手段，也是当今最活跃的一个研究领域。本章简要地回顾了可靠性技术的研究历史，概括地叙述了可靠性技术研究的各个方面及未来研究的广阔前景。

1.1　容错和避错技术的产生及发展

1.1.1　历史的回顾

性能、价格和可靠性是评价一个系统的三大要素。为了提高数字系统的可靠性，人们进行了长期的研究，总结出了两种方法。一种方法是避错(Fault-avoidance)，试图构造出一个不包含故障的"完美"系统，其手段是采用正确的设计和质量控制方法尽量避免把故障引进系统。要绝对做到这一点实际上是不可能的。一旦系统出了故障，则通过检测和修复来消除故障的影响，进而自动或人工地恢复系统。第二种方法叫做容错(Fault-tolerance)，所谓容错就是当出现某些指定的硬件故障或软件错误时，系统仍能执行规定的一组程序(或算法)，或者说程序不会因系统中的故障而中止或被修改，并且执行结果也不包含系统中故障所引起的差错。容错的基本思想是在系统体系结构上精心设计，利用外加资源的冗余技术来达到掩蔽故障的影响，从而自动地恢复系统或达到安全停机的目的。要达到高可靠性的目标，必须综合应用避错和容错两种方法。

人们对避错方法的研究与应用从计算机问世之日就开始了，为了使早期的电子管计算机能满足实际应用的要求，人们在机器的设计和生产过程中，对元器件进行严格的老化和筛选，实行减额使用，并对工艺生产过程严格把关，以使产品能满足设计任务书所规定的可靠性标准。计算机的发展经历了从电子管到晶体管，从晶体管到集成电路，直至目前的大规模集成电路和超大规模集成电路的更新换代，但无论在计算机发展的哪个时代，避错方法都是提高计算机可靠性的基本方法。时至今日，这门技术有了很大的发展，在计算机的研制过程中应用得十分广泛。美国的几家军用计算机公司，如NORDEN公司、EMM公司、ROLM公司和MILTOPE公司等，他们的基本策略就是瞄准当今流行的商用机，研究和利用各种避错技术，使这些计算机具有抗恶劣环境的能力，从而发展系列化、标准化的军用计算机。早期推入市场的有 PDP－11M，VAX－11M，SECS 等军用微机模块系列及军用微机系列。我国从 20 世纪 80 年代中开始先后研制出了自己的抗恶劣环境计算机系列(980JX 及 OPIAC 等)，并通过了国家鉴定，目前正用于军事和工业控制领域。

人们对容错技术的研究也开始得很早，1952 年冯·诺依曼(Von. Neumann) 就在美国加利福尼亚理工学院作过五个关于容错理论研究的报告，他的精辟论述成为以后容错研究的基础。

最初，人们从用四个二级管进行串并联代替单个二极管工作可以提高可靠性这一事实

得到了启发，研制出了四倍冗余线路；从多数元件表决的结果较为可靠这一事实总结出了三模冗余和 N 模冗余结构；在通信中发展起来的纠错码理论也很快地被吸收过来以提高信息在传送、存储以及运算中的可靠性。20 世纪 60 年代末，出现了以自检、自修计算机 STAR 为代表的容错计算机，标志着容错技术从理论上和实践上进入了一个新时期。

20 世纪 70 年代是容错技术研究蓬勃发展的时期，应用和研究范围迅速从宇航领域扩大到交通管制，工厂自动化，电话交换机，医院病人监护，银行资金管理，空港管理，潜艇导航，边界、海岸及领空的保安和监护，战略防卫的控制和数据处理等领域，主要的成果有电话交换系统 ESS 系列处理机，软件实现容错的 SIFT 计算机，容错多重处理机 FTMP，表决多处理机 C.vmp 等。

20 世纪 80 年代是 VLSI 和微计算机迅速发展和广泛应用的时代，容错技术的研究也随着计算机的普及而深入到整个工业界，许多公司生产的容错计算机，如 Stratus 容错计算机系列，IBM System88，Tandem16 等已商品化并推入市场。人们普遍认为，把容错作为每个数字系统的一个主要特征的时代已经到来。

20 世纪 90 年代以来，基于通用硬件（包括 CPU、存储器等）的容错计算机得到重点发展。这一方面是因为受价格因素的制约，日益增长的功能复杂性和迅速增加的集成电路集成度使得开发一种专用容错计算机费用开销很大；另一方面是因为采用通用硬件能缩短开发周期、软件支持丰富等。采用通用硬件的容错计算机具有代表性的是 Stratus 公司推出的 FtServer 系列计算机。

国际电机和电子工程学会（IEEE）从 1971 年起每年召开一次“国际容错计算年会（FTCS）”（2000 年后改为可信计算会议 DCS），并出版论文集，迄今开了三十多届，会议规模越来越大，我国学者也陆续有论文在年会上宣读。此外，我国科技工作者在诊断和容错理论的研究方面，在建造实际的容错计算机系统和应用容错技术方面也取得了不少成果。

1.1.2 展望

随着计算机技术的进一步发展，可靠性设计必将变得越来越重要，其原因如下。

（1）计算机性能的提高（即功能的完备和速度的加快）使系统的复杂性增加，主频加快，也将使系统更容易出错，为了使系统的可靠性不随性能的增高而急剧下降，必须进行精心的可靠性设计。

（2）计算机走向社会，为各行各业所应用，计算机的使用者不再是计算机专业人员，这就要求计算机能够允许各种操作错误。

（3）计算机已从具有良好环境条件的机房迁移到各种应用现场，各种环境因素，如温度、湿度、电磁干扰、机械冲击和振动、盐雾、霉菌等施加于计算机上，使计算机更容易出错，这就要求计算机具有抗恶劣环境的能力。

（4）计算机硬件成本日益降低，维护成本相对增高，因此需提高系统的可靠性以降低维护成本。

由于上述原因，目前和将来的可靠性技术研究将向下面几个方向发展。

（1）瞄准优秀系列结构的商用机，走与商用机兼容的道路，一方面研究和利用各种避错技术，发展抗恶劣环境计算机；另外一方面是研究和利用商用硬件和软件构成高可靠的容错计算机。

目前世界各国研制军用计算机的许多公司，为了减少重复开发的费用，他们根据商用机

与军用机在体系结构、计算类型、数据传输率、响应时间等方面没有本质差别的特点，瞄准当前的主流商用机，在逻辑和软件上全盘翻版，集中力量在计算机的结构组装、系统工艺、质量控制上下功夫，使研制出的计算机能在恶劣环境条件下稳定可靠地工作。这种计算机在工业控制等民用部门也有着广泛的应用。

过去人们设计专用的硬件和软件所研制的多种容错计算机虽然有着广泛的应用，但存在着明显不足：第一，专用容错计算机的成本过高，计算机硬件需要专门设计，软件需要重新规划和设计；第二，专用容错计算机的可扩展能力差；第三，专用容错计算机的编程复杂，使用不方便；第四，升级能力弱，设计周期长，当容错计算机面世的时候它的处理能力已经远落后于同时期的主流计算机，使得容错计算机对于潜在的用户失去了很多吸引力。认识到这些不足，人们开始研究低成本、高扩展能力、紧跟技术进步和尽可能通用的容错计算机。这种容错计算机以通用硬件和软件为基础，充分利用硬件技术和软件的进步所提供的对容错计算的支持，经过精心设计实现容错计算的各个环节。例如美国 NASA 的 JPL 实验室等正在研制用于航空航天应用领域的高性价比的通用容错计算机取代专用容错计算机，欧洲也成立了由多家公司和学术组织一起启动的 GUARDS(Generic Upgradeable Architecture of Real – time Dependable Systems) 计划，共同研制性能卓越的通用容错计算机。

(2) 随着 VLSI 线路复杂性增高，故障埋藏深度增加，发现故障难度增大，为增加芯片可控性和可观测性的可测试性研究已成为重要课题。同时，随着整片集成 WSI(Wafer Scale Integration) 技术和片上系统(SOC) 技术的提出，硅片容错技术应运而生。将动态冗余技术用于 VLSI 的设计，产生了称为 RVLSI(Restructurable VLSI) 的技术。用 PLA 进行容错设计是实现硅片容错的另一途径。由于多值逻辑器件的出现，又提出了一种新的研究方向，即用冗余的逻辑值来实现容错。这些思想和研究课题在新一代计算机的可靠性研究中特别重要。

(3) 在容错系统结构方面，已由单机向分布式系统发展，并尽量采用目前通用的微处理器以及微计算机来实现高性能的分布容错系统，这已成为目前的主要趋向。

由于分布式系统具有模块性、并行性和自治性三大特征，它与集中式系统相比具有可靠、坚固，快速响应，易于修改、扩充，资源共享等明显的优点，因此容错系统结构已由单机向分布式系统发展。利用局部网络的研究成果，采用现有的微处理器及微计算机，在局部网络中注入全局管理、并行操作、自治控制、冗余和错误处理，是研究高性能、高可靠性的分布式容错系统的便利途径。在理论方面，对分布式系统的形式描述不够理想，对程序在分布式环境下的行为特性的研究和理解不够深入；在实际应用方面，实现冗余管理和错误处理方面还有许多困难。总之，要建造一个完全分布容错的计算机系统，无论在理论方面还是在实际应用方面都有许多工作要做。

(4) 对软件可靠性技术将进行更多的研究。随着计算机硬件技术的飞速发展，软件开发的低效率与不可靠性已成为阻碍计算技术继续发展的主要障碍，在这方面有过惨痛的教训。例如，欧洲阿里雅娜火箭发射失败就是由于软件错误引起的。根据统计，在计算机系统中软件故障占系统故障的比例越来越高，甚至达到 80% 以上。面软件领域中的可靠性技术研究，虽然有多年的历史，但进展却非常缓慢，只是到了近些年才受到普遍重视。

提高软件可靠性也有避错和容错两种方法。避错法主要研究生产高可靠软件产品的程序设计方法和软件验证技术；容错法是指开发容错软件的适宜环境和系统方法，其主要目的是提供足够的冗余信息与算法程序，使系统在实际运行中能够及时发现程序设计错误，采取补救措施，保证整个计算机系统的正常运行。目前对软件容错的研究还不成熟。

(5) 在容错性能评价方面,分析法和实验法并重,同时不惜花费昂贵的代价制做试验样机以获得满意的容错系统。

对高可靠系统性能的评价是一项重要而困难的工作。例如对 SIFT 可靠性的评估是基于软件正确运行的假定,由于没有一种满意的方法用来估价软件出错的概率,因此唯一的方法是给出软件正确性的严格的数学证明。尽管 SIFT 系统的软件设计采用了层次结构设计,但这一证明却花了 10 年以上的时间。因此,研究有效的可靠性评估方法就十分迫切了。

为了获得建造一个容错系统的可靠数据,在部分容错系统的研制过程中,不惜花费高昂的代价先建立一个实验系统,通过实测数据确认该容错系统能达到的可靠性目标。自检、自修计算机 STAR 就是为了验证待命储备冗余系统可能得到比单份无冗余系统高达 10 倍以上的寿命增益的预测而研制的实验样机。

故障注入技术是一种重要的容错验证技术,它通过对目标容错计算机系统注入各种硬件故障并观察目标系统对故障的响应,可以获得评价目标系统的各种参数,以辅助系统设计的改进。

为了对高可靠系统进行正确而有效的评估,分析法和实验法的研究都同等重要。

(6) 在理论研究方面,人们企图建立包含"故障"状态的计算机模型,并提出一套容错系统的综合方法论,建立一个广泛的故障病理学和相应的故障防护学。这是一项正在艰难探索的研究课题。

1.2 容错计算的特征及定义

1.2.1 可靠性研究的四论域信息模型

计算机系统极其复杂,为了便于研究,同时清除可靠性研究领域中许多含混不清的概念,可以按照物理的、逻辑的、信息的(统称内部的)、用户的(或称外部的)这样一个递增顺序构造一个层次结构模型来描述一个信息处理系统。层次结构模型中的每一层次都包含各自的一组基本概念、模型和术语。采用这种模型,设计要求、性能度量、正确特性样式、测试方法和概念规范都可以通过给定的论域描述。系统由确定的原子元件以层次的、递归的方式来构造,在给定论域上的一个系统是较高论域上的一个子系统,等等。在每一论域中,我们可以想像一个所有状态的划分,即系统可分为正确的和不正确的两个集合,这个划分借助给定论域的正确特性来获得。此外,每个论域都有一个定时约定,它确定为考察和解释表示信息结构和控制信号的变量的合法的时间间隔。系统的正常功能可由一个不希望事件 UE(Unexpectant Event)(失效、故障、错误、失败)而被破坏,这个不希望事件起源于一个内部的(物理的,逻辑的,信息的)论域,然后在上述论域中产生破坏作用。容错系统的属性和实现它的方法论就可通过四论域,他们的不希望事件,不希望事件的检测算法和恢复算法来解释。因此,我们可以把容错计算定义为当系统出现不希望事件时仍能正确地执行所规定的算法。在这个意义上,所谓容错应当叫做容忍不希望事件,或容忍 UE。

1.2.2 不希望事件 UE 的分类

已经指出,四论域中的每一个都有各自的基本术语,发生在物理域的不希望事件称为失

效。同样，从逻辑域到外部域我们依次把它们的不希望事件叫做故障，差错或错误，以及失败，其因果关系为失效→故障→错误→失败。对于每一个论域，我们都可以根据原因、时间间隔、值和范围对该域的不希望事件等价在逻辑域来描述，并都把它归纳在简单的术语“故障”之下，这就需要建立所谓的“故障模型”，然而这不是一件容易的事。实践证明，对于有些物理失效和信息错误是很难找到其等价的逻辑故障的，尤其是随着技术的进步和不断更新许多具有新的特征的不希望事件不断出现，这样做就更加困难。但是，这毕竟是把复杂的事物进行简化处理的行之有效的方法。因此，我们主要讨论逻辑域中不希望事件（故障）的分类，当不能把其它论域中的不希望事件等价成逻辑论域中的“故障”时，或需要严格的分析讨论时，应就每个论域来讨论，但其方法是类似的。

(1) 按时间间隔分为“永久故障”和“瞬间故障”

永久故障是由元件中的不可逆变化引起的，如固定故障、二极管短路故障等，它永久地将原逻辑变成一个新逻辑。

瞬时故障是持续时间不超过一个确定的最大时间长度的故障。例如短时的外界干扰，粒子对存储单元的影响（由温度和寄生电容引起的），元件参数的暂时变化或程序员的误操作。这类故障只引起元件当前值的变化而不导致不可逆变化。

间歇故障（或伪瞬时故障）是由元件失效、不正确的设计或恶劣环境所引起的，当出现若干逻辑变量的组合时，这类故障才发生，如半导体存储器中的图形敏感故障，由边缘定时和未发现的竞争条件可引起这类故障。

(2) 按值分为“确定值故障”和“非确定值故障”

确定值故障的故障变量保持在许可的一个恒定值上，例如引线开路；而非确定值故障允许故障变量在可能的值之间不断改变，如振荡故障。

(3) 按范围分为“局部故障”和“分布式故障”

局部（单）故障是只影响局部逻辑线路（单逻辑变量）的故障，而分布式故障（相当于多故障）是包含有两个，三个，多个变量，一个子系统或整个系统的故障，分布式故障可能造成灾难性后果。

物理故障是由物理系统内部的（如半导体结的破坏）或外部的（电磁辐射等）原因引起的发生在逻辑域的故障，而人为错误（不正确的设计、误操作和恶意攻击等）是由设计人员或操作人员引起的错误。

值得指出的是，发生在逻辑域中的许多故障的起源点是物理域中的元件失效。

不希望事件及其原因可以简略地用图 1－1 表示。

1.2.3 容忍不希望事件 UE

已经指出，有两种基本的方法可获得可靠的信息处理，第一种叫做避错法，第二种叫做容错法。无论哪种方法，其目的都是控制系统中可能发生的不希望事件。要实现容错法，首先应确认被容忍的不希望事件的规范，其次要选择与该不希望事件的类别相匹配的检测算法，并在此基础上设计出恢复算法，以便由选定的检测算法所调用，并使系统回到正确操作的某个级或者安全停机（系统恢复）。实现方法选定之后，应对其性能进行数量估价（容错性能评价）以确定其有效性并使设计精确化（设计精加工）。

信息处理系统三个内部论域（物理的，逻辑的，信息的）中的防卫方案的一个简单抽象模型如图 1－2 所示。图中所有避错技术形成一个环绕正确行为的防卫圈 A，n 个不同的不希

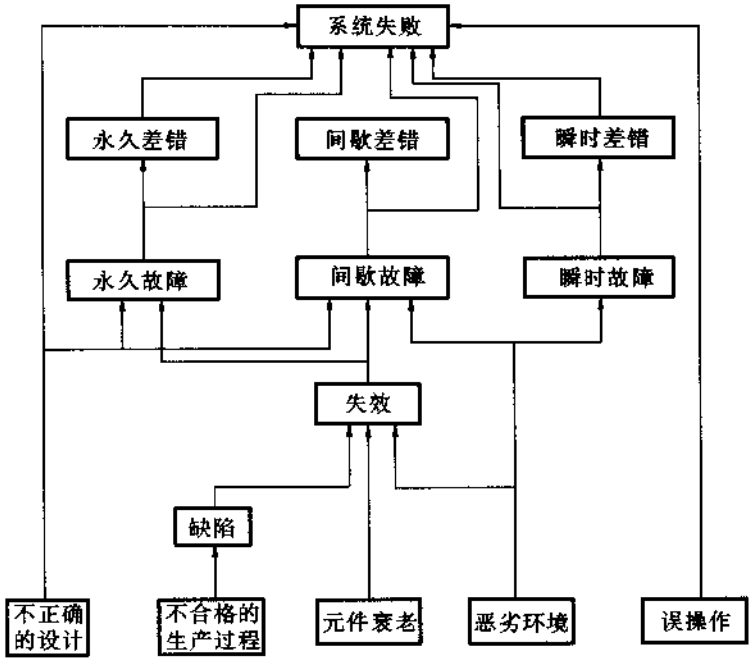

图 1－1　不希望事件及其原因

望事件的检测算法由标号为 $D_1, D_2, \cdots, D_i, \cdots D_n$ 的环来表示，箭头 UE(a)，UE(b)、UE(c) 表示由这些算法检测的 UE，而弧 $R_1, R_2, \cdots, R_i, \cdots, R_n$ 表示由检测算法所调用的恢复算法的成功作用。A 和 D_n 之间的弧表示可恢复的不正确特性，而 D_n 之外的面积表示内部论域中不可恢复的不正确特性。

图中示出的检测和恢复发生在不希望事件 UE(失效、故障、错误) 存在的内部论域中，因此，这些 UE 的出现在所在论域中被有效屏蔽，从面达到容忍 UE 的目的，图中 UE(x) 表示一个 UE，它不可能由 n 个检测算法中的任何一个检测，并且在内部域中产生了一个不可恢复的不正确行为。该 UE 不能被屏蔽，且将在外部论域中产生一个不希望事件，很可能导致灾难性后果。

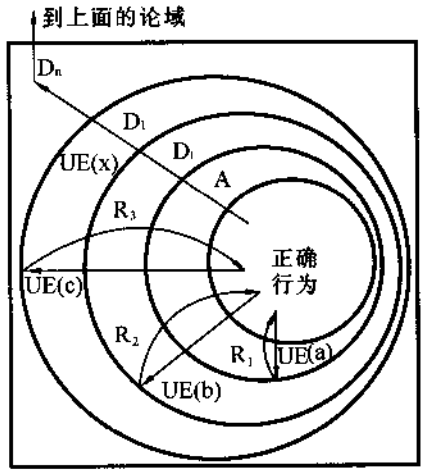

图 1－2　内部论域的防卫模型

1.2.4　容错计算四要素

实现容错计算包括下面四个主要方面。

(1)UE的检测　为了容忍系统中的一个UE，其结果应当首先被检测，当一个失效(或故障) 不能由系统直接检测时，该失效(或故障) 往往表现为系统中某些地方的错误(信息域)，因此，容错技术通常的出发点是错误状态的检测。

(2) 损坏估价 当检测出一个错误时,系统的许多状态都可能被怀疑,而不只是怀疑初始发现的错误状态,由于一个失效(或故障) 的出现和它的错误结果之间可能存在延迟,错误的信息可能已经传播到该系统的其它地方,导致错误的扩大。因此,在作出一个与被检测的错误有关的任何决定之前,有必要鉴定系统已被破坏的程度。这就依赖于系统设计者的策略和已有的探测技术。

(3)UE 的恢复 在 UE 检测和损坏估价之后,应采用 UE 恢复技术,其目的在于把目前的错误系统状态转换成一个确定的无错系统状态,以便继续正常的系统操作,否则,系统损坏可能继续发生。

(4)UE 处理和继续服务 尽管 UE 恢复阶段可能使系统回到无 UE 状态,仍旧需要一种技术确保已被恢复了的 UE 效应不会立即再现,以使系统继续提供规定的服务,UE 处理的第一步是试图精确地定位 UE,紧接着是恢复 UE 或重新配置其余的系统以避免发生 UE。

这四个方面形成所有容错技术的基础,从而也是设计和制造容错系统的基础。各个阶段之间可能有很大的互相影响,导致一个具体系统中这几个阶段识别的模糊。例如,保护机构常常提供一种形式的错误检测,并可能在设计和实现损坏估价阶段也起着重要作用。类似地,对一个系统的损坏估价将利用探测方法来识别可能的破坏,测量本身也采用 UE 检测技术等。

1.2.5 实现容错计算的主要方法

容错计算是依靠外加资源的方法来换取可靠性的。外加资源的方法很多,主要的有外加硬件,外加信息,外加时间和外加软件,这些方法往往要合理使用才能达到高可靠性的目标。

1. 硬件冗余

广泛应用的硬件冗余之一是硬件堆积冗余,在物理级可通过元件的重复而获得(如相同元件的串、并联,四倍元件等)。物理域的恢复作用是自动的,即不需单独的检测,但每一次失效将削弱防卫,在逻辑域可采用多数表决方案,如三模冗余、N 模冗余、分段冗余、修复机构等。

另一硬件冗余方法叫做待命储备冗余,该系统中共有 $m+1$ 个模块,其中只有一块处于工作状态,其余 m 块都处于待命接替状态。一旦工作模块出了故障,立刻切换到一个待命储备模块,当换上的储备模块发生故障时,又切换到另一储备模块,直至资源枯竭,显然,这种系统必须具有检错和切换装置。

将堆积冗余和待命储备冗余结合运用,构成所谓混合冗余系统,当堆积冗余中有一个模块发生故障时,立即将其切除,并代之以无故障的待命模块,这种冗余方式既可达到较高的可靠度,又可达到较长的无故障运行时间。

上述三种容错基本结构统称为 K 出自 N 结构。该结构中共有 N 个相同的模块,其中至少有 K 个是正常的,系统才能运行。这种结构能容忍分别出现在 $(N-K)$ 个模块中的 $(N-K)$ 个独立的故障,或称其容忍故障能力为 $t=N-K$。

对有人维修的系统,一有故障就能排除,两模块就能起到多模块的作用,因此可构成双模冗余系统。在部件级和整机级可实现双模结构,在整机级可采用双机交替工作、双机协同工作和修理不停机等工作方式。

近年来,随着线路密度的大大增加,自动生成测试模式和利用这些测试模式进行故障模拟的能力急剧减弱。解决这个问题的中心思想是提高系统的可控制性和可观测性,人们把对

它的研究归纳在“可测试性设计”这个技术范畴之中，并受到普遍重视。其中大多数是通过硬件资源来达到目的的。

可测试性设计可分为两类。第一类是针对一个具体的设计，提出一个适合于提高该设计的可控性和可观测性的一个特殊方法，该方法没有普遍意义，如划分、增加测试点、总线结构、特征分析等。第二类叫做构造法，它具有通用性，一般包括一组设计规则。构造法的目的是减少一个网络的时序复杂性，往往把一个时序网络当做组合网络来处理，诸如敏感级扫描设计，扫描通路，扫描 / 置位逻辑，随机访问扫描等。

2.时间冗余

时间冗余是通过消耗时间资源来达到容错目的的。时间冗余的一个应用是程序卷回。这种技术用来检验一段程序完成时的计算数据，如有错，则卷回，重算那个部分。如果一次卷回不解决问题，还可多次卷回，直到故障消除或判定不能消除故障为止。

利用诊断程序对系统进行初始检查、联机检查、周期性检查都可看做时间冗余的应用。人们对故障的检测与诊断方面的研究，历史悠久，成果显著，例如 D 算法、布尔差分法、星算法、多值算法、因果分析法、图论法等都是产生故障检测和诊断测试集的有效方法。

3.信息冗余

信息冗余是靠增加信息的多余度来提高可靠性的。这些附加的信息位置具有如下功能：当代码中某些信息位发生错误(包括附加位本身的错误) 时能及时发现错误(检错)，或者能恢复原来的信息(纠错)。一般而言，附加的信息位越多，其检错纠错能力越强。在数字系统中的信息传送，算术逻辑运算中广泛使用的奇偶码、海明码、乘积码、循环码，各种算术误差码都有很强的检错、纠错能力。

信息冗余的优点是增加的冗余度比别的方法低，而且许多码的信息位和校验位在运算中可统一处理。此外，它还能纠正瞬时错误，提供故障的自检测、自定位、自纠错能力，其缺点是产生时延，难于纠正编码器和译码器本身的错误。

4.软件冗余

提高软件的可靠性有两种方法。一种是研究无错误软件，另一种是研究容错软件。

无错软件曾经是过去有关容错的许多文章研究的课题。通常假定，一个可靠的软件一经产生，它在以后的运行中仍保持是可靠的。当然，这个假定取决于使用软件的正确性，支撑软件的系统的正确性，软件维护的正确性等。因此，在上述条件下，无错软件就是在软件的使用过程中无错误的软件。

无错软件的研究主要包括以下三方面的内容。

(1) 寻求导致高可靠软件产品的程序设计方法。目前已为人们广泛接受的结构化程序设计方法就是一例。

(2) 软件测试技术。该方法是在软件设计完成后，交付用户之前施行的。如验收测试技术等。

(3) 程序正确性证明。其主要内容是：应用一种严格的语言，阐明程序要达到的目标，然后以数学论证法证明程序执行的输出与所要达到的目的是否相符。程序的证明可以通过另一个程序(证明程序) 来实施，然而证明程序自身的正确性又如何证明是一个重要的问题。

软件容错技术指开发容错软件的适宜环境和系统方法，其主要目的是提供足够的冗余信息给算法程序，使系统在实际行动中能够及时发现程序错误，采取补救措施，保证整个计算的正确运行。

软件容错的主要任务是研究如何将具体设计差异，对应同一任务采用不同软件程序组成一个有机整体，完成错误检测、程序系统重组及系统恢复等多项功能，达到利用设计差异实现容错的目的。

从系统结构而言，有两种软件冗余方式：静态冗余和动态冗余。静态冗余方式构成的容错软件系统，需要冗余硬件的支持，如 N 个独立程序在独立的硬件模块中运行，其典型代表是“N 份程序”(NVP) 结构。

用动态冗余方式构成的容错软件是通过备份软件来实现的。这种结构由单机支持，其代表是“恢复块”(RB) 结构。

5.各种冗余技术的综合应用

要实现一个容错信息处理系统，必须根据系统特性所确定的可靠性指标，成本诸因素选择适当的冗余方式，将这些冗余方式应用于适当的级别。一般而言，信息冗余的冗余度低，效率高，在逻辑域中获得了广泛的应用。各种硬冗余适合于各个级别，所用级别越低，可靠度越高，但冗余量增加，附加成分又降低了系统的可靠度，增加了成本。在有人维修的系统中，可采用双模冗余。软冗余成本较高，只有当优点超过硬冗余时才使用。在有的场合，可同时使用软冗余和硬冗余以获得最佳效果。对程序比较固定的地方可采用时间冗余，对有些可靠性要求极高的系统中，往往要综合应用各种冗余技术。目前，在计算机网络和各种分布式系统中也广泛应用冗余技术来提高可靠性。总之，冗余要消耗资源，因此在满足所需可靠性的前提下，应尽量减少资源的消耗，在可靠性与资源消耗之间权衡利弊，决定取舍。

1.2.6 可靠性参数

度量系统的可靠性有两个主要的参数：$R(t)$ 和 $A(t)$。

系统的可靠度 $R(t)$ 是指在 $t=0$ 时系统正常的条件下，系统在时间区间 $[0,t]$ 内能正常运行的概率。

数字系统是由众多的元器件构成的，在一个无冗余系统中，元器件的失效将导致系统的失败。系统的可靠度可以表示为

$$R(t) = \mathrm{e}^{-\lambda t}$$

其中，λ 是单位时间内失效元件数与元件总数之比，称为失效率。在系统的正常生命期中，失效率 λ 为常数。

系统的可用度 $A(t)$ 是指系统在时间 t 可运行的概率。当 t 趋于无穷大时，$A(t)$ 的极限存在，则该极限称为系统的稳态可用度，它表示期望系统可用来执行有用计算的时间部分。有些工作，例如预防性维护和修复会减少系统给用户的可用时间。可用度通常用来作为衡量可延迟或停止短时间服务而不导致严重后果的系统的品质因素。

可靠度 $R(t)$ 和可用度 $A(t)$ 都是时间的函数，往往很复杂，有时也引入一些简单的参数来度量系统的可靠性，例如系统的平均无故障运行时间 MTTF 为

$$\mathrm{MTTF} = \int_0^\infty R(t)\mathrm{d}t = \int_0^\infty \mathrm{e}^{-\lambda t}\mathrm{d}t = \frac{1}{\lambda}$$

1.3 避错和容错技术的分类

1.3.1 避错技术的分类

对一个非容错系统而言，任意故障将导致系统失效，其可靠度

$$R(t) = e^{-\lambda t}$$

显然在无冗余的前提下，提高系统可靠性的根本途径是降低 λ 的值。

在大量实验的基础上，得出计算集成电路失效率的 MIL－217B 模型如下：

$$\lambda = \pi_L \pi_Q (C_1 \pi_T + C_2 \pi_E)$$

式中 π_L——学习因子，由器件生产过程的成熟程度决定；

π_Q——质量因子，由器件老化、筛选的项目多少和严格程序决定；

π_T——温度因子，由器件的结温决定；

π_E——环境因子，由系统的应用环境条件决定；

C_1，C_2——器件的复杂程序因子。

避错的目标是尽量减少故障出现的概率，即减少失效率 λ，因此必须努力降低 π_L，π_Q，π_E，π_T 诸因子。其方法有环境防护技术，质量控制技术和提高元件的集成度等。

(1) 环境防护技术　数字系统应用的环境不同，则所受环境因素，如温度、湿度、冲击、振动、电磁场、盐雾、霉菌等的影响也不同。必须采取适当的环境防护措施，如热设计、机械应力防护、化学防护、电磁兼容性设计等，使数字系统能够承受所感受的环境应力，这是提高可靠性的重要途径。

(2) 质量控制技术　要减少质量因子 π_Q，必须进行质量控制。质量控制主要是指在生产线上，对全部材料、工艺及设备有严格的质量管理规范，对产品生产的每一步流程都进行检测，及时发现问题，分析失效形式，改进和控制工艺流程，使混入成品中有缺陷的次品数量尽量减少。

(3) 提高元件集成度　由 MIL－217B 模型可知，元件的失效率 λ 不随复杂性线性增加。复杂性因子 C_1 和 C_2 与门的数目遵从指数规律关系，单个门的失效率随每块封装的门的数目的增加而减小。因此整个系统的失效率随集成度的增大而减小。

1.3.2 容错技术的分类

避错的目的是尽量减小故障出现的概率，是提高系统可靠性的最直观的方法，是任何生产过程和设计过程都必须考虑的。容错则是利用外加的冗余资源来掩盖故障的影响。为了克服故障的影响，一个冗余系统可能经历多达 10 个阶段：

- 故障限制：限定故障的传播范围，防止故障对其它区域的污染；
- 故障检测：尽快发现故障，减小故障潜伏期，可脱机和联机检测；
- 故障屏蔽：掩盖故障对输出的影响；
- 重试：再作一遍或若干遍，消除对不引起物理破坏的瞬时故障的影响；
- 诊断：确定故障的位置；

• 重组:切除故障部件,换上备份部件;

• 恢复:检测和重组后,使系统操作回到故障检测前的处理点;

• 重启:当恢复不能消除故障影响时,采用"热"重启(从故障检测点恢复所有的操作)或"冷"重启(重新引导装入系统);

• 修复:对故障部件进行修理使之复原,修复也可脱机或联机进行;

• 重构:把修复了的部件加入系统,若修复是联机进行的,则重构不能中断系统的运行。

图1-3示出了上述概念,对非容错系统而言,所有故障处理均由系统外部提供,而容错系统自身包含部分或所有故障处理机构。

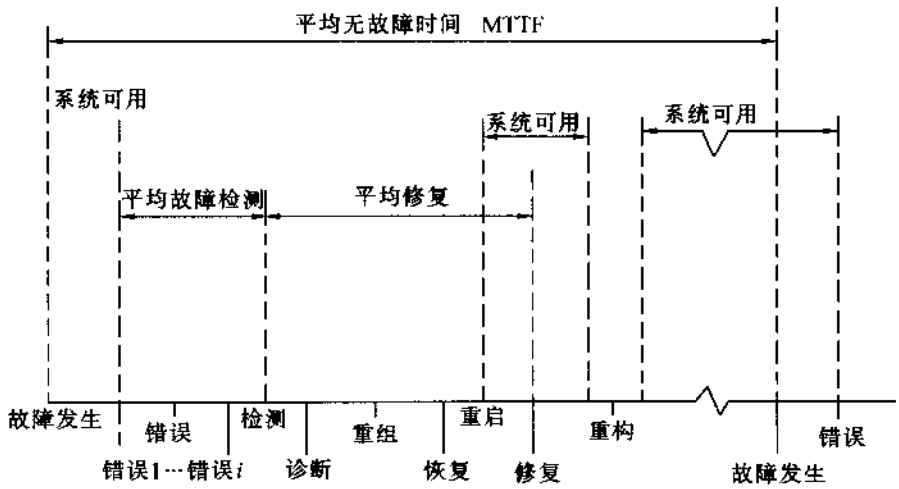

图1-3 冗余系统经历的阶段

根据对故障处理的方式不同,可把容错技术分为故障检测、故障屏蔽和动态冗余三大类。故障检测不提供对故障的容忍,只提供已发生故障的警告。屏蔽冗余(也叫静态冗余)容忍故障但不给出故障警告。动态冗余是最复杂的一类容错技术,它可包含故障处理的所有10个阶段。

1.检错技术

检错技术是检测和定位故障的技术。衡量检测技术的主要指标是检测覆盖率,即任意一故障被检测到的概率。检测技术也包括了诊断,衡量诊断技术的指标是诊断分辨率,即故障定位的精确程度。

最常见的检测技术是检错码,这种技术已广泛用于通讯设备和存储器的设计中。

二倍仿作也是一种常见的检测技术,其基本思想是用两个模块同时操作,对结果进行比较,若不一致则检测出错。

其他一些检错技术包括:自校验,故障保险,安全失败逻辑以及超时监督定时器等。

2.屏蔽冗余

故障检测技术给出出错警告,也可提供诊断能力,分辨到有限的故障位置。但只采用故障检测技术不能提供有效的故障容忍,而故障屏蔽技术提供了容忍故障的冗余,在故障效应到达模块输出以前,通过隔离或校正来消除它们的影响。屏蔽冗余不改变系统的结构,因此