ICS 25.040
N 10

中华人民共和国国家标准

GB/T 20438.7—2006/IEC 61508-7:2000

电气/电子/可编程电子安全相关系统的功能安全　第7部分:技术和措施概述

Functional safety of electrical/electronic/programmable electronic safety-related systems—Part 7:Overview of techniques and measures

(IEC 61508-7:2000,IDT)

2006-07-25 发布　　2007-01-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

中 华 人 民 共 和 国
国 家 标 准
电气/电子/可编程电子安全相关系统的
功能安全 第7部分:技术和措施概述
GB/T 20438.7—2006/IEC 61508-7:2000
*
中国标准出版社出版发行
北京复兴门外三里河北街16号
邮政编码:100045
网址 www.spc.net.cn
电话:68523946 68517548
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 4.75 字数 154 千字
2007年2月第一版 2007年2月第一次印刷
*

目　次

前　言

GB/T 20438 由下列 7 部分构成：

——第 1 部分：一般要求；

——第 2 部分：电气/电子/可编程电子安全相关系统的要求；

——第 3 部分：软件要求；

——第 4 部分：定义和缩略语；

——第 5 部分：确定安全完整性等级的方法示例；

——第 6 部分：GB/T 20438.2 和 GB/T 20438.3 的应用指南；

——第 7 部分：技术和措施概述。

本部分是 GB/T 20438 的第 7 部分。

本部分等同翻译国际标准 IEC 61508-7:2000-03(第 1 版)《电气/电子/可编程电子安全相关系统的功能安全　第 7 部分：技术和措施概述》(英文版)。

附录 A、附录 B、附录 C、附录 D 为资料性附录。

本部分与 IEC 61508-7:2000 在技术内容上没有差异，为便于使用做了下列编辑性修改：

a)　将“IEC 61508”改为“GB/T 20438”；

b)　本“国际标准”一词改为“本标准”；

c)　删除国际标准中 1.2 中注 2，因为此注所表述的是 IEC 61508 在美国和加拿大等国的应用情况，与我国的实际不符，所以删除；

d)　用小数点“.”代替原标准中作为小数点的逗号“,”。

本部分由中国机械工业联合会提出。

本部分由全国工业过程测量和控制标准化技术委员会(SAC/TC 124)归口。

本部分由机械工业仪器仪表综合技术经济研究所负责起草。

本部分主要起草人：欧阳劲松、冯晓升、王莉、蔡廷安、马光武、梅恪、郑旭等。

引　言

由电气和电子器件构成的系统，多年来在许多领域中执行其安全功能，以计算机为基础的系统（一般指可编程电子系统(PES)）在许多领域中用于非安全目的，但也越来越多地用于安全目的，为使计算机系统技术更有效安全的使用，有必要进行安全方面的指导。

GB/T 20438 针对由电气或电子和可编程电子部件构成的、起安全作用的电气/电子/可编程电子系统(E/E/PES)的整体安全生命周期，提出了一个通用的方法。建立统一的方法的目的是为了针对以电子为基础的安全相关系统提出一种一致的、合理的技术方针，主要目标是促进应用领域标准的制定。

在许多情况下，可用多种基于不同技术的防护系统来保证安全（如机械的、液压的、气动的、电气的、电子的、可编程电子的，等等）。从安全战略角度，不仅要考虑各系统中元器件的问题（如传感器、控制器、执行器等），而且要考虑构成组合安全相关系统的所有安全相关系统。因此 GB/T 20438 对电气/电子/可编程电子(E/E/PE)安全相关系统进行了规定。GB/T 20438 还提出了一个框架，在这个框架内，基于其他技术的安全相关系统也可同时被考虑进去。

在各种应用领域里，存在着许多潜在的危险和风险，包含的复杂性也各不相同，从而需应用不同的 E/E/PES。对每个特定的应用，则根据应用的不同而确定所需的安全量。GB/T 20438 仅是使这些量值规范化。

GB/T 20438

——考虑了当使用 E/E/PES 执行安全功能时，所涉及到的整体安全生命周期、E/E/PES 安全生命周期以及软件生命周期的各阶段（如初始构思，整个设计、实现、运行和维护到停用）。

——针对飞速发展的技术，建立一个足够健壮而广泛的能满足今后发展需要的框架。

——有利于促进 E/E/PES 安全相关系统在不同领域中相关标准的制定，各应用领域和交叉应用领域相关标准应在 GB/T 20438 的框架下制定，使之具有高水平的一致性（如基础原理，术语等的一致性），并将既安全又经济。

——为达到 E/E/PE 安全相关系统所需的功能安全，提供了编制安全要求规范的方法。

——使用了一个安全完整性等级，此安全完整性等级规定了 E/E/PE 安全相关系统要实现的安全功能的目标安全完整性等级。

——采用了一种可确定安全完整性等级要求的基于风险的方案。

——建立了 E/E/PE 安全相关系统的数值目标失效量，这些量都同安全完整性等级相联系。

——建立了危险失效模式中目标失效量的一个下限，此下限是对单一 E/E/PE 安全相关系统的要求。这些系统运行在：

1) 低要求操作模式下，为了执行它的设计功能，一旦要求时，就把下限设定成平均失效概率为 10^{-5}；

2) 高要求操作模式或者连续操作模式下，下限设定成危险失效概率为 10^{-9}/h。

注：单一 E/E/PE 安全相关系统不一定是单通道结构。

——采用广泛的原理、技术和措施以达到 E/E/PE 安全相关系统的功能安全，但不使用失效-安全的概念，这个概念是在很好定义了失效模式，并且复杂性相对较低时的一个数值。由于 E/E/PE 安全相关系统的复杂性均在 GB/T 20438 范围之内，因此不适用失效-安全的概念。

电气/电子/可编程电子安全相关系统的功能安全
第7部分:技术和措施概述

1 范围

1.1 GB/T 20438的本部分包含了GB/T 20438.2和GB/T 20438.3有关的各种安全技术和措施的概述。

注:参考文献仅作为各种方法和工具或示例的基本参考,不一定代表当前技术水平。

1.2 GB/T 20438.1、GB/T 20438.2、GB/T 20438.3和GB/T 20438.4是基础安全标准,虽然它们不适用于简单的E/E/PE安全相关系统(见GB/T 20438.4—2006的3.4.4),但作为基础安全标准,各技术委员会可以在IEC导则104和ISO/IEC导则51的指导下制定相关标准时使用。对于每个技术委员会,都有责任在其制定的标准中使用基础标准。同时,GB/T 20438也是一个可独立使用的标准。

在适用的情况下,技术委员会在制定其标准时都应使用基础安全标准。也就是说,本基础安全标准涉及的要求、测试方法或测试条件,只有在相关技术委员会制定标准时加以引用或包含时,才能得到应用。

注:只有在满足所有有关要求时,才能实现E/E/PE安全相关系统的功能安全,因此,仔细考虑和适当参考所有有关要求是很重要的。

1.3 图1是GB/T 20438的整体框架图,并指出了GB/T 20438.7在实现E/E/PE安全相关系统功能安全中所起的作用。

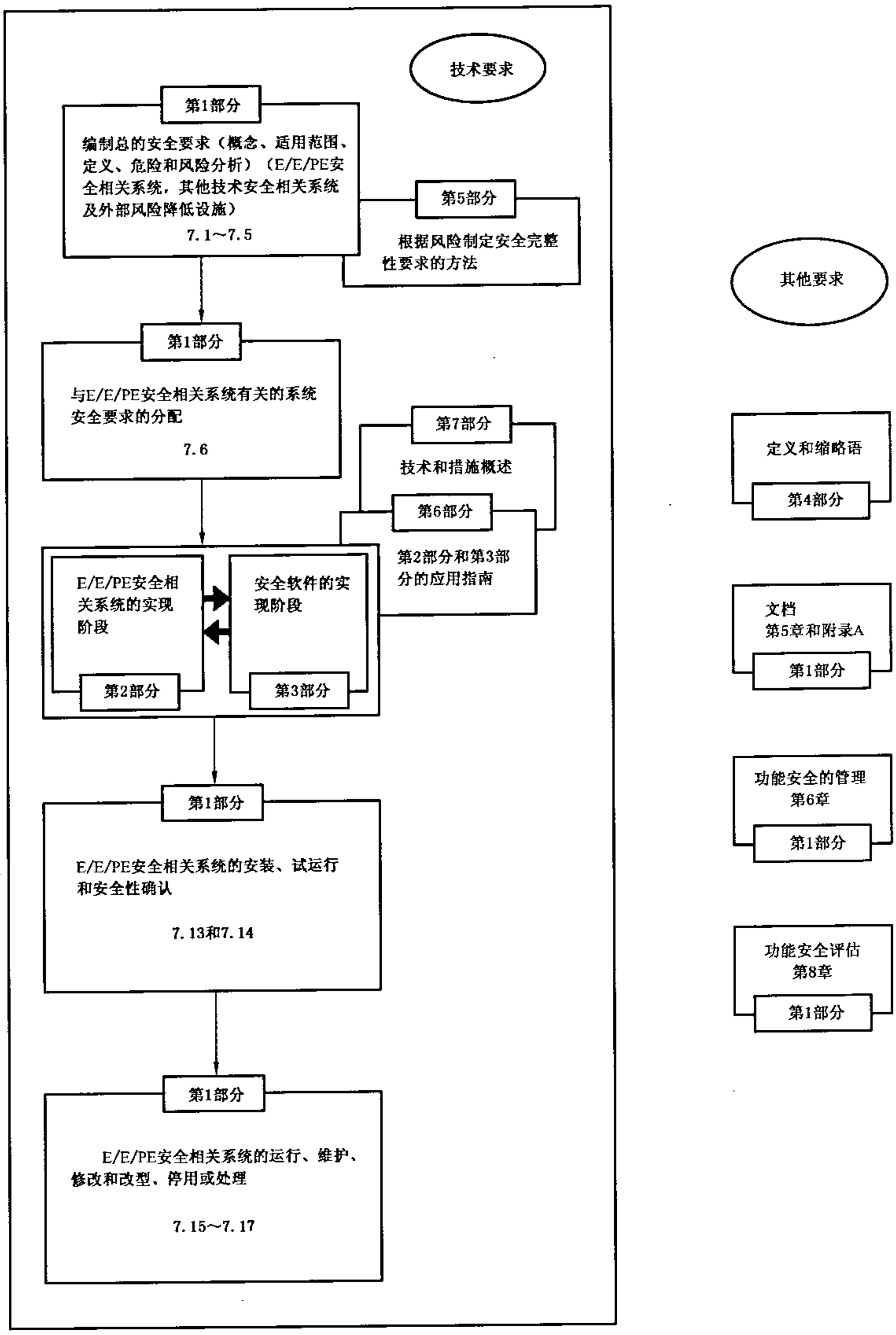

图 1　GB/T 20438 的总体框架

2 规范性引用文件

下列文件中的条款通过 GB/T 20438 的本部分的引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分，然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

GB/T 20438.1—2006 电气/电子/可编程电子安全相关系统的功能安全 第1部分：一般要求(IEC 61508-1:1998,IDT)

GB/T 20438.2—2006 电气/电子/可编程电子安全相关系统的功能安全 第2部分：电气/电子/可编程电子安全相关系统的要求(IEC 61508-2:2000,IDT)

GB/T 20438.3—2006 电气/电子/可编程电子安全相关系统的功能安全 第3部分：软件要求(IEC 61508-3:1998,IDT)

GB/T 20438.4—2006 电气/电子/可编程电子安全相关系统的功能安全 第4部分：定义和缩略语(IEC 61508-4:1998,IDT)

GB/T 20438.5—2006 电气/电子/可编程电子安全相关系统的功能安全 第5部分：确定安全完整性等级的方法示例(IEC 61508-5:1998,IDT)

GB/T 20438.6—2006 电气/电子/可编程电子安全相关系统的功能安全 第6部分：GB/T 20438.2和 GB/T 20438.3 的应用指南(IEC 61508-6:2000,IDT)

IEC 导则 104:1997 安全出版物的编写及基本安全出版物和分类安全出版物的应用

ISO/IEC 导则 51:1990 安全方面 在标准中引入安全条款的指南

3 定义和缩略语

GB/T 20438 的本部分的定义和缩略语已在 GB/T 20438.4 中给出。

附 录 A
（资料性附录）
E/E/PES 的技术和措施概述：随机硬件失效控制
（参看 GB/T 20438.2）

A.1 电气

整体目标：控制机电元件中的失效。

A.1.1 利用在线监视检测失效

注：在 GB/T 20438.2—2006 的表 A.2、表 A.3、表 A.7 和表 A.14 及表 A.19 中引用了本技术/措施。

目的：通过监视 E/E/PE 安全相关系统在响应受控设备（EUC）正常（在线）运行时的行为来检测失效。

描述：在某些条件下，可用（例如）EUC 的时间行为信息来检测失效，例如，一般是由 EUC 启动 E/E/PE 安全相关系统组成部分的一只开关，如果在预定的时间开关并不改变状态，则将检测到一次失效，通常要测定失效部位是不可能的。

A.1.2 继电器触点监视

注：在 GB/T 20438.2—2006 的表 A.2 和表 A.15 中引用了本技术/措施。

目的：检测继电器触点的失效（例如被熔接）。

描述：强制接触（或者可靠的导向接触）继电器可使它们的触点刚性地接在一起，假定有两组转换触点，分别为 a 和 b，a 是常开触点，b 是常闭触点，被熔接时，随继电器线圈断电，a 不能闭合，因此，当继电器线圈断电时，监视常闭触点 b 的吸合可用来证明常开触点 a 已打开。常闭触点 b 闭合的失效表明触点 a 的一次失效，所以对任何受触点 a 控制的机器而言，监视电路应保证安全关机或保持关机状态。

参考文献：

Zusammenstellung und Bewertung elektromechanischer Sicherheitsschaltungen für Verriegelungseinrichtungen. F. Kreutzkampf, W. Hertel, Sicherheitstechnisches Informations und Arbeitsblatt 330212, BIA-Handbuch. 17, Lfg. X/91, Erich Schmidt Verlag, Bielefeld.

Anlagensicherung mit Mitteln der MSR-Technik. G. Strohrman, Oldenburg. 1983.

A.1.3 比较器

注：在 GB/T 20438.2—2006 的表 A.2、表 A.3、表 A.4 中引用了本技术/措施。

目的：为了尽早检测一个独立处理单元或者比较器中的（非同时的）失效。

描述：利用一个硬件比较器周期性地或者连续地比较独立处理单元的信号。可以在外部测试比较器，或者它本身可使用自监视技术。监测到的处理器行为的差异将产生一条失效报文。

A.1.4 多数表决器

注：在 GB/T 20438.2—2006 的表 A.2、表 A.3 和表 A.4 中引用了本技术/措施。

目的：为了检测和防护至少三个硬件通道之一中的失效。

描述：使用多数原理（3 个中有 2 个、3 个中有 3 个、或者 n 个中 m 个）的一个表决单元被用来检测和防止失效。可在外部测试表决器，也可使用自监视技术。

参考文献：

化学过程的安全自动化指南. CCPS, AIChE, New York, 1993.

Anlagensicherung mit Mitteln der MSR-Technik. Praxis der Sicherheitstechnik, Vol1, Dechema, 1988.

Sicherung von Anlagen der Verfahrenstechnik mit Mitteln der Mess-, Steuerungs-und Regelu-ng-

stechnik. VDI/VDE Blatt 1 to 5,1984 to 1988.

A.1.5 非工作电流原理(断电跳闸)

注：在 GB/T 20438.2—2006 的表 A.2、表 A.9、表 A.14 和表 A.15 中引用了本技术/措施。

目的：为了在切断电源或掉电时执行安全功能。

描述：当触点断开并且没有电流流过时就执行安全功能。例如，当使用制动器来刹住一台电机的一个危险运动机件时，制动装置将随安全相关系统中触点闭合而开闸并随安全相关系统中触点打开而闭闸。

参考文献：

化学过程的安全自动化指南. CCPS,AIChE,New York,1993.

A.2 电子

整体目标：控制固态部件中的失效。

A.2.1 利用冗余硬件进行测试

注：在 GB/T 20438.2—2006 的表 A.3、表 A.16、表 A.17 和表 A.19 中引用了本技术/措施。

目的：为了使用硬件冗余(即使用不必执行过程功能的附加硬件)检测失效。

描述：冗余硬件可用来以一个适当的频度检测指定的安全功能。要实现 A.1.1 或 A.2.2，通常必须使用这种方法。

参考文献：

DIN V VDE 0801：Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben(安全系统中计算机遵循的准则)，Beuth-Verlag，Berlin，1990.

A.2.2 动态原理

注：在 GB/T 20438.2—2006 的表 A.3 中引用了本技术/措施。

目的：通过动态信号处理来检测静态失效。

描述：强制改变其他的静态信号(内部或外部产生的)可帮助检测部件中的静态信号。通常这种技术与机电部件相联系。

参考文献：

Elektronik in der Sicherheitstechnik. H. Jürs，D. Reinert，Sicherheitstechnisches Informations und Arbeitsblatt 330220，BIA-Handbuch，Erich-Schmidt Verlag，Bielefeld，1993.

A.2.3 访问存取端口和边界扫描结构的标准测试

注：在 GB/T 20438.2—2006 的表 A.3、表 A.16 和表 A.19 中引用了本技术/措施。

目的：为了控制和观测一片 IC(集成电路)的每个引脚处发生的情况。

描述：边界扫描测试是一种 IC 设计技术，此技术通过解决怎样增加访问 IC 内的电路测试点的问题提高 IC 的可测试性。在由核心逻辑、输入/输出缓冲器组成的一个典型边界扫描 IC 中，核心逻辑和与每个 IC 引脚相邻的输入/输出缓冲器之间插入了一个移位寄存器级。一个边界扫描组元中包含一个移位寄存器级。通过标准测试存取端口，边界扫描组元可控制和观测一个 IC 的每个输入/输出引脚处发生的情况。把芯片内的核心逻辑同接收到的外围部件的激励隔离开然后执行一次内部自测试，可完成 IC 核心逻辑的内部测试。这些测试可用来检测 IC 中的失效。

参考文献：

IEEE 1149.1：1990　标准测试存取端口和边界扫描结构.

A.2.4 失效-安全硬件

注：在 GB/T 20438.2—2006 的表 A.3 中引用了本技术/措施。

目的：为了在发生一次失效时，使系统进入一种安全状态。

描述：在硬连线系统中，如果一个定义的故障集合将导致一种安全工况，并且它们被检测到，则可认

为一个单元是以一种故障-安全方式运行的。

示例:定义的故障集合可包括固定型故障,固定开型故障,部件内部和部件之间短路以及定向短路故障。

参考文献:

关键计算机系统的可靠性 1. F. J. Redmill,Elsevier Applied Science,1988,ISBN 1-85166-203-0.

Elektronik in der Sicherheitstechnik. H. Jürs,D. Reinert,Sicherheitstechnisches Informations und Arbeitsblatt 330220,BIA-Handbuch,Erich-Schmidt Verlag,Bielefeld,1993.

A.2.5 监视冗余

注:在 GB/T 20438.2—2006 的表 A.3 中引用了本技术/措施。

目的:通过配备几个功能单元,监视每个单元的行为,并在检测到这些单元的行为有任何差异时就启动到一种安全工况的转换,从而检测失效。

描述:至少由两个硬件通道来执行安全功能。监视这些通道的输出,当检测到一个故障时(即当各通道的输出信号不相同时)就启动一个安全工况。

参考文献:

关键计算机系统的可靠性 1. F. J. Redmll,Elsevier Applied Science,1988,ISBN 1-85166-203-0.

Elektronik in der Sicherheitsechnik. H. Jürs,D. Reinert,Sicherheitstechnisches Informations und Arbeitsblatt 330220,BIA-Handbuch,Erich-Schmidt Verlag,Bielefeld,1993.

A.2.6 带自动检验的电气/电子部件

注:在 GB/T 20438.2—2006 的表 A.3 中引用了本技术/措施。

目的:为了通过定期检验安全功能来检测失效。

描述:在起动过程之前测试硬件,并且按适当的间隔时间反复测试该硬件。只有在每次测试都无问题时,受控设备(EUC)才继续运行。

参考文献:

关键计算机系统的可靠性 1. F. J. Redmill,Elsevier Applied Science,1988,ISBN 1-85166-203-0.

Elektronikin der Sicherheitstechnik. H. Jürs,D. Reinert,Sicherheitstechnisches Informations und Arbeitsblatt 330220,BIA-Handbuch,Erich-Schmidt Verlag,Bielefeld,1993.

A.2.7 模拟信号监视

注:在 GB/T 20438.2—2006 的表 A.3 和 A.14 中引用了本技术/措施。

目的:提高所测信号的置信度。

描述:凡是有选择时,应使用模拟信号而不使用数字开/关状态。例如,在使用常见的信号电平容差监视时,都是用模拟信号电平来表示跳闸或者安全状态。该技术提供了连续监视以及发送器置信度的较高级别,减少了发送器传感功能必要的验证试验的频度。外部接口,例如脉冲线路也需要测试。

参考文献:

基于仪表的系统的 UKOOA 指南. UK Offshore Operators Association Limited,December 1995.

A.2.8 降额

目的:提高硬件部件的可靠性。

描述:通过把系统设计成在低于最大规范额定值下运行来保证硬件部件在某一应力级下运行。降额是保证在所有额定工作环境下,部件可在低于它们的最大应力级下工作的实际作法。

A.3 处理单元

整体目标:辨别导致处理单元结果错误的失效。

A.3.1 利用软件进行自测试:有限模式数(单通道)

注:在 GB/T 20438.2—2006 的表 A.4 中引用了本技术/措施。

目的:尽早检测出处理单元中的失效。

描述:使用不需考虑任何特殊安全要求的标准技术构建硬件。借助附加的软件功能完全可实现失效检测,这种软件功能使用至少两个互补的数据模式(例如十六进制的 55 和十六进制的 AA)来执行自测试。

A.3.2 利用软件进行自测试:漫步位(WALKING BIT)(单通道)

注:在 GB/T 20438.2—2006 的表 A.4 中引用了本技术/措施。

目的:尽早检测出处理单元的物理存储器(例如寄存器)和指令译码器中的失效。

描述:利用附加软件功能完全可实现失效检测,该软件功能使用可测试物理存储器(数据和地址寄存器)和指令译码的一个数据模式(例如漫步位模式)执行自测试,但诊断覆盖率只有 90%。

参考文献:

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.3.3 由硬件支持的自测试(单通道)

注:为了尽早检测出处理单元中的失效,使用了特殊硬件来提高失效检测速度和扩大失效检测范围。

描述:附加的特殊硬件装置支持自测试功能以便检测失效。例如,该装置可能是一个硬件单元,它根据看门狗原理周期性地监视某个位模式的输出。

参考文献:

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.3.4 编码处理(单通道)

注:在 GB/T 20438.2—2006 的表 A.4 中引用了本技术/措施。

目的:尽早检测出处理单元中的失效。

描述:设计处理单元时使用了特殊的失效辨别或者失效校正电路技术。迄今,这些技术还只适用于比较简单的电路而未普及;但不排除将来的发展。

参考文献:

编码微处理机认证. P. Ozello, Proc. SAFECOMP'92, 185-190, 1992.

各种转接系统的核心编码处理机原理和应用. P. Forin, IFAC Control Computers Communications in Transportation, 79-84, 1989.

Le processeur codé: un nouveau concept appliqué á la Sécurité des systémes de transports. Gabriel, Martin, Wartski, Revue Génerale des chemins de fer, No. 6, June 1990.

A.3.5 利用软件进行相互比较

注:在 GB/T 20438.2—2006 的表 A.4 中引用了本技术/措施。

目的:尽早检测出处理单元中的失效,本方法使用了动态软件比较。

描述:两个处理单元互相交换数据(包括结果、中间结果和测试数据)。在每个单元中使用软件对数据进行一次对比,检测到的差异将产生一条失效报文。

参考文献:

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.4 不可变的存储区

整体目标:检测不可变存储区中信息的修改。

A.4.1 字保存多位冗余(例如使用一个改进的汉明码进行 ROM 监视)

注:参见 A.5.6 和 C.3.2,在 GB/T 20438.2—2006 的表 A.5 中引用了本技术/措施。

目的:检测一个16位字中所有的单位失效,所有的双位失效,某些3位失效和某些全位失效。

描述:为了产生汉明距离至少为4的一个改进的汉明码,可用几个冗余位来扩充存储器的每一个字。每当读一个字时,检验冗余位就能确定是否发生了错误。当发现有一个差异时,就会产生一条失效信息。此程序也可通过计算数据字和它的地址的连接冗余位来检测寻址失效。

参考文献:

错误检测和纠正码. R. W. Hamming, Bell System Technical Journal 29(2), 147-160, 1950.

Prüfbare und Korrigierbare Codes. W. W. Peterson, München, Oldenburg, 1967.

A.4.2 修改的校验和

注:在GB/T 20438.2—2006的表A.5中引用了本技术/措施。

目的:检测所有奇数位失效,即全部可能位的大约50%失效。

描述:借助一种合适的算法来创立一个校验和,此算法使用了一个存储器字组中的全部字。校验和可作为一个附加字存储在ROM中,或者把一个附加字附加给内存块以保证校验和算法产生一个预定的值。在后面的存储器测试中,再使用同一算法建立一个校验和,并把结果同存储的或者定义的值进行对比。如果发现有差异,则产生一条失效报文。

参考文献:

安全技术中的微机——面向开发者和生产者的一个工具. H. Hölscher, J. rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.4.3 单字(8位)的签名

注:在GB/T 20438.2—2006的表A.5中引用了本技术/措施。

目的:检测一个字的所有单位失效和所有多位失效,以及所有可能位约99.6%的位失效。

描述:(既可使用硬件也可使用软件)通过一种循环冗余检验(CRC)算法把一个内存块的内容压缩成一个内存字。一种典型的CRC算法是把块的整个内容当作字节串行或者位串行数据流来处理,根据这种算法,使用一个多项式发生器来执行一个连续的多项式除法。除得的余项就代表压缩的存储内容——即是存储器的"签名"——并被存储起来。在后面的测试中又计算一次签名,并把这个签名同早先存储的签名进行比较。当它们有差异时,就产生一条失效报文。

参考文献:

计算软件中的一个错误校验字符. S. Vasa, Computer Design, 5, 1976.

Berechnung von Fehlererkennungswahrscheinlichkeiten bei Signaturregistern. D. Leisengang, Elektronische Rechenanlagen 24, H. 2, S. 55-61, 1982.

A.4.4 双字(16位)的签名

注:在GB/T 20438.2—2006的表A.5中引用了此技术/措施。

目的:检测一个字中所有的单位失效和所有的多位失效,以及所有可能位的99.998%的位失效。

描述:本程序使用一种循环冗余检验(CRC)算法来计算一个签名,而结果值至少有两个字长,像单字情况中那样,扩展的签名被存储、重新计算和比较。当存储的和重算的签名之间有差异时就产生一条失效报文。

参考文献:

Signaturanalyse in der Datenverarbeitung. D. Leisengang, M. Wagner, Elektronik 32, H. 21, S. 67-72, 1983.

Signaturregister für Selbsttestende ICs. B. Könemann, J. Mucha, G. Zwiehoff, Größtintegration/NTG - Fachtagung Baden-Baden, S, 109-112, April 1977.

A.4.5 块复制(例如利用硬件或者软件进行比较的双重ROM)

注:在GB/T 20438.2—2006的表A.5中引用了本技术/措施。

目的:检测全部位失效。

描述:在两个存储器中复制地址空间。第一个存储器以正常方式工作。第二个存储器包含同样的信息并且同第一个并行存取。比较它们的输出,当检测到有差异时就产生一条失效报文。为了检测某类位错误,必须在两个存储器中的一个中逆向存储数据,当读数时,再次反向。

参考文献:

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

计算机现在已能安全地执行关键的安全功能. Otto Berg von Linde, Railway Gazette international, Vol. 135, No. 11, 1979.

A.5 可变的存储区

整体目标:检测寻址、写、存储和读过程中的失效。

A.5.1 "检测板"法或"跨步"法 RAM 测试

注:在 GB/T 20438.2—2006 的表 A.6 中引用了本技术/措施。

目的:检测主要的静态位失效。

描述:把一个 0 s 和 1 s 模式型式的检测板写入一个面向位的存储器的存储单元中。成对地检查存储单元以保证它们的内容相同和正确无误。这个对的第 1 单元的地址是可变的,而此对的第 2 单元的地址则由第 1 单元的地址逐位反向形成。第 1 次运行时,存储器地址区从可变地址朝高地址扩展,而第 2 次运行时则朝低地址蔓延,在预先指定一个反向值的情况下,两次运行应是一样的。如出现有差异则产生一条失效报文。

在一次"跨步"法 RAM 测试中,一个面向位的存储器的单元将由一个均匀的位流初始化。在第一次运行时,按升序检查这些单元,检查每个单元的内容是否正确并将它的内容反向。在第二次运行时按降序和同样的方式处理第一次运行中建立的后台。在第 3 次或第 4 次运行中反向预赋值的情况下,将重复头两次运行。如果出现有差异就会产生一条失效报文。

参考文献:

存储器测试. W. G. Fee,大规模集成电路测试(Tutorial at the COMPCON 77 in San Francisco), IEEE Computer Society, W. G. Fee(ed), 81-88, 1978.

存储器测试. P. Rosnfield, Electronics and Power, H. 1, P26-31, 1979.

Halbleiterspeicher-Testfolgen. Th. John, E. Schaefer, Elektronipraxis, H. 6, 18-26 and H. 7, 10-14, 1980.

A.5.2 "漫步路径"法 RAM 测试

注:在 GB/T 20438.2—2006 的表 A.6 中引用了本技术/措施。

目的:为了检测静态和动态位失效,以及存贮单元之间的串扰。

描述:用一个均匀的位流初始化要测试的存储范围,第 1 个单元被反向并检查其余的存储区以确保后台是正确无误的。此后第 1 单元再次反向从而使它回复到它的初始值,对下一个单元也重复整个操作过程。在反向的后台预赋值情况下执行"漫游位模型"的第 2 次运行。当出现有差异时就会产生一条失效报文。

参考文献:

存储器测试. W. G. Fee,大规模集成电器测试(Tutorial at the COMPCON 77 in San Francisco), IEEE Computer Society, W. G. Fee(ed), 81-88, 1978.

测试微处理机族的技术. W. Barraclough, A. Chiang, W. Sohl, Proceedings of the IEEE 64(6), 943-950, 1976.

A.5.3 "galpat"法或者"透明的 galpat"法 RAM 测试

注:在 GB/T 20438.2—2006 的表 A.6 中引用本技术/措施。

目的:为了检测静态位失效和大比例动态耦合。

描述:在“galpat”RAM 测试法中,先一致地(即全为 0 或者为 1)初始化选择的存储范围。测试最初存储单元,然后倒置第 1 个存储单元,并检查所有剩余的单元以确保它们的内容正确无误。每读取一个剩余单元之后,也检验反向的单元。对选定的存储范围中的每个单元都重复此操作过程。在相反的初始化情况下执行第二次运行。任何差异就会产生一条失效报文。

“透明的 galpat”测试法是上面的操作过程的一种变种:更换初始化所选存储范围内的所有单元,现存内容保持不变,使用签名来比较单元集合的内容。选择在所选范围内要测试的第 1 个单元,计算范围中其余所有单元的签名 S1,并把它存储起来。然后把要测试的单元反向,重新计算所有其余单元的签名 S2(在每读取一个剩余单元之后,也检验反向的单元。)。比较 S2 和 S1,任何差异就产生一条失效报文。重新反向被测单元使之重建原先的内容,重新计算其余所有单元的签名 S3,把 S3 同 S1 进行比较,任何差异就产生一条失效报文。按同样的方法测试所选存储范围中的所有存储单元。

参考文献:

Entwurf von Selbsttestprogrammen für Mikrocomputer. E. Maehle, Microcomputing. Berichte der Tagung lll/79 des German Chapter of the ACM, W. Remmele, H. Schecher, (ed.), Stultgart, Teubner, 204-216, 1979.

Periodescher Selbsttest einer mikroprocessorgesteuerten Sicherheitsschaltung. U. Stinnesbek, Diplomarbeit am Institut für theoretische Elektrotechnik der RWTH Aachen 1980.

A.5.4 “Abraham”RAM 测试法

注:在 GB/T 20438.2—2006 的表 A.6 中引用了本技术/措施。

目的:为了检测存储单元之间所有的固定型失效和耦合失效。

描述:检测故障的比例超过“galpat”RAM 测试法。整个存储器测试需要执行的操作次数约为 $30n$,n 是存储器中的单元数。为了在操作循环过程中使用透明法进行测试,需要通过分割存储器并在不同时间段测试每个分区。

参考文献:

测试半导体随机存取存储器(RAM)的有效算法. R. Nair, S. M. Thatte, J. A. Abraham, IEEE Trans. Comput. C-27(6), 572-576, 1978.

A.5.5 1 位冗余(例如,使用一个奇偶校验位进行 RAM 监视)

注:GB/T 20438.2—2006 的表 A.6 中引用了本技术/措施。

目的:为了在被测试的存储范围内检测所有可能的位失效的 50%。

描述:把存储器的每个字都扩展 1 位(奇偶校验位),此位给每个字补齐偶数个或奇数个逻辑 1。每次读数据字时将检验它的奇偶性。如发现 1 的个数有错时,就产生一条失效报文。应这样选择偶或奇奇偶性,使得在一次失效事件中,无论是 0 字(全 0)还是 1 字(全 1)都不是有效的,此时字也不是有效代码。当计算数据字和它的地址连接的奇偶性时,奇偶校验也可用来检测寻址失效。

参考文献:

Integrierte Digitalbausteine. K. Reiß, H. Liedl, W. Spichall, Berlin, 1970.

A.5.6 利用一个修改的汉明码进行 RAM 监视,或者利用差错检测和纠错码(EDC)检测数据失效

注:可参看 A.4.1 和 C.3.2,在 GB/T 20438.2—2006 的表 A.6 中引用了本技术/措施。

目的:为了检测所有的奇位失效,所有的 2 位失效,某些 3 位和多位失效。

描述:把存储器的每个字扩展几个冗余位从而产生具有一个汉明距离至少为 4 的一个修改过的汉明码。每读一个字时,通过检验冗余位可以确定是否发生了一次讹错。当发现有差异时,就产生一条失效报文。当计算数据字和它的地址连接的冗余位时,此程序也可用来检测寻址失效。

参考文献:

错误检测和纠错码. R. W. Hamming, The Bell System Technical Journal 29(2), 147-160, 1950.

Prüfbare und korrigierbare codes. W. W. Peterson, München, Oldenburg, 1967.

A.5.7 具有硬件或者软件比较和读/写测试的双重 RAM

注：在 GB/T 20438.2—2006 的表 A.6 中引用本技术/措施。

目的：为了检测全位失效。

描述：在两个存储器中复制地址空间。第一个存储器以常规方式工作。第 2 个存储器包含同样的信息并且同第一个存储器并行存取。比较两个输出，当检测到差异时就产生一条失效报文。为了检测某些类型的位错误，两个存储器中的一个存储的数据必须反向，当读出时再次反向。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Holscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986 ISBN 3-88585-315-9.

计算机现在已能安全地执行关键的安全功能. Otto Berg von Linde, Railway Gazette international, Vol. 135, No. 11, 1979.

A.6 I/O 单元和接口(外部通信)

整体目标：为了检测输入和输出单元(数字、模拟、串行或者并行)中的失效以及防止不允许的输出传送给过程。

A.6.1 测试模式

注：在 GB/T 20438.2—2006 的表 A.7、表 A.14 和表 A.15 中引用了本技术/措施。

目的：为了检测静态失效(固定型失效)和串音。

描述：它是一种与数据流无关的输入和输出单元的循环测试。它用一种定义的测试模式来比较观测值和对应的预计值。测试模式信息、测试模式接受及测试模式评价都必须相互独立。受控设备不应受到不允许的测试模式的影响。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.6.2 代码保护

注：在 GB/T 20438.2—2006 表 A.7、表 A.16、表 A.17 和表 A.19 中引用了本技术/措施。

目的：为了检测输入/输出数据流中随机硬件和系统性失效。

描述：本程序可保护输入和输出信息免受系统和随机硬件引起的失效。代码保护提供了以信息冗余和时间冗余为基础的、与数据流有关的输入、输出单元的失效检测。典型地是把冗余信息叠加在输入或输出数据上。它提供了监视输入或输出电路正确运行的一种方法。许多技术都能用，例如，在一个传感器输出信号上可以叠加一个载频信号。为了监视逻辑单元和最后的执行器之间流经的一个信号的有效性，逻辑单元可以检验附加到一个输出通道上的载频或者冗余码位的存在。

参考文献：

标准的输入/输出测试和监视程序——安全技术中的微机——开发者和生产者取向的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.6.3 多通道并行输出

注：在 GB/T 20438.2—2006 的表 A.7 中引用了本技术/措施。

目的：为了检测随机硬件失效(固定型失效)、外部影响引起的失效、定时失效、寻地失效、漂移失效和瞬态失效。

描述：它是用于检测随机硬件失效的一个具有独立输出的、与数据流有关的多道并行输出。通过外部比较器进行失效检测。当发生一次失效时，立即关掉受控设备。这种措施只有在诊断测试间隔期间数据流改变时才有效。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rneinland, Köln, 1986, ISBN 3-88585-315-9.

A.6.4 受监视的输出

注：在 GB/T 20438.2—2006 的表 A.7 中引用了本技术/措施。

目的：为了检测个体失效，由外部影响引起的失效，定时失效，寻址失效，漂移失效(例如，模拟信号)和瞬态失效。

描述：输出同无关的输入之间有关数据流的比较从而保证同一个定义的容差范围(时间、值)相符。检测到的一次失效并不总是同输出故障有关。只有在诊断测试间隔期间数据流改变时，此措施才有效。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV *Rheinland*, Köln, 1986, ISBN 3-88585-315-9.

MSR—Schutzeinrichtungen. Anforderungen und Massnahmen *zur* gesicherten Funktion. DIN V 19251, February 1995.

A.6.5 输入比较/表决

注：在 GB/T 20438.2—2006 的表 A.7 和表 A.14 中引用本技术/措施。

目的：为了测量个体失效，由外部影响引起的失效，定时失效，寻址失效，漂移失效(对于模拟信号)和瞬态失效。

描述：独立输入的一种与数据流有关的比较，从而确保同一个定义的容差范围(时间、值)相符。2 个中将有 1 个、3 个中将有 2 个或者更多的冗余。此措施只有在诊断测试间隔期间数据改变时才有效。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

MSK-Schutzeinrichtungen. Anforderungen und Massnahmen zur gesicherten Funktion. DIN V 19251, February 1995.

A.7 数据通路(内部通信)

整体目标：为了检测由信息传送中的一个故障引起的失效。

A.7.1 一位硬件冗余

注：在 GB/T 20438.2—2006 的表 A.8 中引用了本技术/措施。

目的：为了检测所有的奇位失效，即数据流中所有可能失效位的 50%。

描述：用一行(位)扩充总线并且借助奇偶校验可把这个附加行(位)用于检测失效。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.7.2 多位硬件冗余

注：在 GB/T 20438.2—2006 的表 A.8 中引用了本技术/措施。

目的：为了检测通信过程中总线上和串行传输链路中的失效。

描述：用 2 行(位)或多行(位)扩充总线，为了借助汉明码技术检测失效，将使用这些附加行(位)。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.7.3 完全硬件冗余

注：在 GB/T 20438.2—2006 的表 A.8 中引用了本技术/措施。

目的：利用比较两条总线上的信号检测通信过程中的失效。

描述：为了检测失效，总线被加倍并使用了附加行(位)。

参考文献：

安全技术中的微机——面向开发者和生产者取向的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.7.4 使用测试模式进行检查

注：在 GB/T 20438.2—2006 的表 A.8 中引用了本技术/措施。

目的：为了检测静态失效(固定型失效)和串音。

描述：这是一种与数据流无关的数据通路循环测试。它使用一个定义的测试模式来比较观察值和相应的预期值。

测试模式信息、测试模式接受、测试模式评价必须彼此完全无关。受控设备不应受到测试模式不允许的影响。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.7.5 传输冗余

注：在 GB/T 20438.2—2006 的表 A.8 中引用了本技术/措施。

目的：为了检测总线通信中的瞬态失效。

描述：依次把信息传送几次。这种重复法只对于瞬态失效才有用。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, ISBN 3-88585-315-9.

A.7.6 信息冗余

注：在 GB/T 20438.2—2006 的表 A.8 中引用了本技术/措施。

目的：为了测量总线通信中的失效。

描述：成块地并连同每个块的一个计算出的检验和传输数据。接收机重新计算接收数据的检验和，并把结果同接收到的检验和作比较。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986 ISBN 3-88585-315-9.

A.8 电源

整体目标：为了检测或者允许一个电源故障引起的失效。

A.8.1 使用安全断电的过压保护

注：在 GB/T 20438.2—2006 的表 A.9 中引用了本技术/措施。

目的：为了保护安全相关系统免受过压。

描述：及早检测过压使之能通过掉电例行程序或者转换到第二电源把所有输出转换到一个安全工况。

参考文献：

化学过程的安全自动化指南. CCPS, AIChE, New York, 1993.

A.8.2 电压控制(次级)

注：在 GB/T 20438.2—2006 的表 A.9 中引用了本技术/措施。

目的:为了监视次级电压并当电压超过规定范围时就启动一个安全工况。

描述:监视次级电压,当它超出规定范围时,就启动一次掉电或转换到第2电源。

A.8.3 具有安全断电的掉电

注:在GB/T 20438.2—2006的表A.9中引用了本技术/措施。

目的:使用存储的所有安全临界信息来切断电源。

描述:及早检测过压或欠压使之内部状态能保存在非易失性存储器中(如必要的话),并使所有的输出能通过掉电例程转换到一个安全工况,或者借助掉电例程或者转换到第2电源把所有的输出转换到一个安全工况。

A.9 时序的和逻辑的程序序列监视

注:在GB/T 20438.2—2006的表A.16、表A.17和表A.19中引用了本技术/措施。

整体目标:为了检测一个有缺陷的程序序列。当在错误的序列或时段中处理一个程序的各个单元(例如软件模块、子程序或者命令)时或者当处理机的时钟有问题时,就会存在一个有缺陷的程序序列。

A.9.1 具有分离时基而无时间窗的看门狗

注:在GB/T 20438.2—2006的表A.10和表A.12中引用了本技术/措施。

目的:为了监视程序序列的行为和似真性。

描述:为了监视计算机的行为和程序序列的似真性,定期触发具有分离时基的外部定时单元(例如看门狗)。在程序中正确地设置触发点是很重要的。不按一个固定的周期触发看门狗,但规定了最大时间间隔。

参考文献:

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.9.2 具有分离时基和时间窗的看门狗

注:在GB/T 20438.2—2006的表A.10和表A.12中引用了本技术/措施。

目的:为了监视程序序列的行为和似真性。

描述:为了监视计算机的行为和程序序列的似真性,定期触发具有分离时基的外部定时单元(例如看门狗)。在程序中正确设置触发点是很重要的。给出了看门狗的一个上、下限。如果程序序列占用的时间比预定的时间长或者短,就会采取应急行动。

参考文献:

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-135-9.

A.9.3 程序序列的逻辑监视

注:在GB/T 20438.2—2006的表A.10和表A.12中引用了本技术/措施。

目的:为了监视各个程序段的正确顺序。

描述:使用软件(计数程序、临界程序)或者使用外部监视设备监视各个程序段的正确顺序。在程序中正确设置校验点是很重要的。

参考文献:

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölscher, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.9.4 程序序列的时序和逻辑监视的组合

注:GB/T 20438.2—2006的表A.10和表A.12中引用了本技术/措施。

目的:为了监视各个程序段的行为和正确的顺序。

描述:只有在也正确执行程序段的顺序时才会触发监视程序序列的一台时序设备(例如看门狗)。

参考文献：

安全技术中的微机——面向开发者和生产者的一种工具. H. Hölsechr, J. Rader, Verlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

A.9.5 具有在线检验的时序监视

注：在 GB/T 20438.2—2006 的表 A.10 和表 A.12 中引用本技术/措施。

目的：为了检测时序监视中的失效。

描述：在起动时检验时序监视，并且只有当时序监视正常工作时，才可能启动。例如，在起动时，可以用一个加热的电阻来检验一个温度传感器。

A.10 通风和加热

注：在 GB/T 20438.2—2006 的表 A.17 和表 A.19 中引用了本技术/措施。

整体目标：为了控制通风和加热中的失效，和/或 监视它们是否与安全有关。

A.10.1 温度传感器

注：在 GB/T 20438.2—2006 的表 A.11 中引用了本技术/措施。

目的：当系统开始在规定的温度范围之外工作之前，检测温度过高还是温度过低。

描述：温度传感器监视 E/E/PES 的大多数临界点处的温度。在温度偏离规定范围之前就采取应急行动。

A.10.2 风扇控制

注：在 GB/T 20438.2—2006 的表 A.11 中引用了本技术/措施。

目的：为了检测风扇运转的不正常。

描述：监视风扇的运转是否正常。当一台风扇工作不正常时，就要采取维护(或者最终得采取应急)行动。

A.10.3 通过热熔断器启动安全断电

注：在 GB/T 20438.2—2006 的表 A.11 中引用了本技术/措施。

目的：在系统的工作温度超过技术条件的规定之前，就断掉安全相关系统的电源。

描述：一个热熔断器被用来切断安全相关系统的供电，对一个可编程电子系统(PES)来说，可通过一个存储有应急行动所需的全部信息的掉电例行程序来引起断电。

A.10.4 来自温度传感器和条件报警的交错报文

注：在 GB/T 20438.2—2006 的表 A.11 中引用了本技术/措施。

目的：为了指示安全相关系统正工作在技术条件规定的温度范围之外。

描述：监视温度，当温度超出规定范围时就产生一个报警。

A.10.5 强制风冷的连接和状态指示

注：在 GB/T 20438.2—2006 的表 A.11 中引用了本技术/措施。

目的：利用强制风冷防止过热。

描述：监视温度，当温度高于规定的一个上限时就引起强制风冷却，并告之用户系统此时的状态。

A.11 通信和大容量存储器

整体目标：为了控制在同外部源和海量存储器通信过程中的失效。

A.11.1 分隔开电力线和信息线

注：在 GB/T 20438.2—2006 的表 A.13 中引用了本技术/措施。

目的：为了减小信息线中大电流感生的串音。

描述：分隔开电力供电线同传送信息的线路。可能在信息线上感生电压脉冲的电场随距离增大而减小。

A.11.2 多线路的空间分隔

注：在 GB/T 20438.2—2006 的表 A.13 和表 A.17 中引用了本技术/措施。

目的：为了减少多线路中大电流感生的串音。

描述：载有重复信号的线路彼此分隔开。可能在多线路上感生电压脉冲的电场随距离增大而减小。这种措施也减少了共同原因失效。

A.11.3 提高抗干扰性

注：在 GB/T 20438.2—2006 的表 A.13、表 A.17 和表 A.19 中引用了本技术/措施。

目的：为了减小对安全相关系统的电磁干扰。

描述：可以使用比如屏蔽和滤波这样的设计技术来提高安全相关系统对电力线或信号线或者静电放电产生的辐射或者传导电磁干扰的抗扰性。

参考文献：

IEC 61000-5-2/TR3:1997 电磁兼容性(EMC) 第 5 部分：安装和调节指南 第 2 章：接地和敷设电缆.

降低电子系统中噪声的技术. H. W. Ott, John Wiley Interscience, 2nd Edition, 1988.

产品设计师的 EMC. Tim Williams, Newnes, 1992, ISBN 0-7506-1264-9.

仪表中的接地和屏蔽技术. John Wiley & Sons, New York, 1986.

电磁兼容性的原理和技术. C. Christopoulos, CRC Press, 1995.

Gestaltung von Maschinensteuerungen unter Berücksichtigung der elektromagnetischen Verträglichkeit. F. Börner, Sicherheitstechnisches Informations-und Arbeitsblatt 330260, BIA-Handbuch. 20. Lfg. V/93, Erich Schmidt Verlag, Bielefeld.

A.11.4 抗合成(Antivalent)信号传输

注：在 GB/T 20438.2—2006 的表 A.13 和表 A.17 中引用了本技术/措施。

目的：为了检测在多信号传输线中相同的感应电压。

描述：使用抗合成(Antivalent)信号(例如逻辑 1 和 0)来传输所有的重复信息。可以通过一个抗合成(Antivalent)比较器来检测共同原因失效(例如由电磁发射引起的)。

参考文献：

Elektronk in der Sicherheitstechnik. H. Jürs, D. Reinert, Sicherheitstechnisches Informations-und Arbeitsblatt 330220, BIA-Handdbuch. 20. Lfg. V/93, Erich Schmidt Verlag, Bielefeld.

A.12 传感器

整体目标：为了控制安全相关系统的传感器中的失效。

A.12.1 参考传感器

注：在 GB/T 20438.2—2006 的表 A.14 中引用了本技术/措施。

目的：为了检测传感器的工作不正常。

描述：使用了一个独立的参考传感器来监测过程传感器的工作。参考传感器以适当的时间间隔检验所有的输入信号从而检测过程传感器的失效。

参考文献：

化学过程的安全自动化指南. CCPS, AIChE, New York, 1993.

A.12.2 启动可靠的开关

注：在 GB/T 20438.2—2006 的表 A.14 中引用了本技术/措施。

目的：通过开关凸轮和触点之间的直接机械连接断开一个触点。

描述：通过开关凸轮和触点之间的直接机械连接，启动可靠的开关将断开它的常闭触点。这就保证了任何时候，只要开关凸轮处于吸合位置，开关触点一定是断开的。

参考文献：

Verriegelung beweglicher Schutzeinrichtungen. F. Kreutzkampf, K. Becker, Sicherheitstechniches Informations-und Arbeitsblatt 330210, BIA-Handbuch. 1. Lfg. IX/85 Erich Schmidt Verlag, Bielefeld.

A.13 最终元件(执行器)

整体目标：为了控制安全相关系统的最终元件中的失效。

A.13.1 监视

注：在 GB/T 20438.2—2006 的表 A.15 中引用了本技术/措施。

目的：为了检测一个执行器的工作不正常。

描述：监视执行器的工作(例如通过一个继电器的启动可靠的触点，参看 A.1.2 中继电器触点监视)。此监视采用的冗余可用来触发应急行动。

参考文献：

Zusammenstellung und Bewertung elektromechanischer Sicherheitsschaltungen für Verriegelungseinrichtungen. F. Kreutzkampf, W. Hertel, Sicherheitstechnisches Informations-und Arbeitsblatt 330212, BIA-Handbuch. 17. Lfg. X/91, Erich Schmidt Verlag, Bielefeld.

化学过程的安全自动化指南. CCPS, AIChE, New York, 1993.

A.13.2 多个执行器的交叉监视

注：在 GB/T 20438.2—2006 的表 A.15 中引用了本技术/措施。

目的：为了通过比较结果来检测多个执行器中的故障。

描述：使用一个不同的硬件通道来监视多个执行器中的每一个执行器。当出现差异时，就采取应急行动。

A.14 对于实际环境采取的措施

注：在 GB/T 20438.2—2006 的表 A.17 中引用了本技术/措施。

目的：为了防止实际环境(水、灰尘、腐蚀物)的影响引起失效。

描述：设计的设备的机壳要能耐受预计的环境。

参考文献：

IEC 60529:1989 机壳提供的保护等级(IP 码).

附 录 B
（资料性附录）
E/E/PES 的技术和措施概述：系统失效的避免
（参看 GB/T 20438.2 和 GB/T 20438.3）

注：本附录中的许多技术适用于附录 C 中未重复的软件。

B.1 一般测量和技术

B.1.1 项目管理

注：在 GB/T 20438.2—2006 的表 B.1～表 B.6 中引用了本技术/措施。

目的：为了避免失效，在开发和测试安全相关系统时采用了一种组织模型、一些规则和措施。

描述：最重要和最好的技术和措施是：

a) 建立一个组织模型，特别是质量保证的组织模型（参看标准，比如 ISO 9000-1～ISO 9004-1 系列标准或类似的标准），在质量保证手册中制定有此组织模型。

b) 在交叉项目和项目专用指南中制定建立和确认安全相关系统的规则和措施。

下面确立了许多重要的基本原则：

c) 定义一个设计机构：

——机构的任务和职权；

——质保部门的职权；

——质量保证（内部检验）与开发的独立性。

d) 定义一个序列计划（活动模型）：

——确定在执行项目的过程中所有有关的活动，包括内部检验和它们的日程表；

——项目更新。

e) 定义一个内部检验的标准化程序：

——检验（检验理论）的计划、执行和检查；

——次品的释放机构；

——重复检验的妥善保管。

f) 配置管理：

——管理和版本检查；

——修改效果的检测；

——修改后的一致性检验。

g) 采用质保措施定量评估：

——采集要求；

——失效统计；

——采用计算机辅助通用方法，工具和人员培训。

参考文献：

IEEE：软件工程标准. IEEE/Wiley-Interscience，New York，1987.

关键计算机系统的可靠性 1. F. J. Redmill，Elsevier Applied Science，1988，ISBN 1-85166-203-0.

化学过程的安全自动化指南. CCPS，AIChE，New York，1993.

B.1.2 编制文档

注 1：在 GB/T 20438.2—2006 的表 B.1～表 B.6 中引用了本技术/措施。

注 2：另见 GB/T 20438.1—2006 的第 5 章和附录 A。

目的:通过把开发过程中的每一步编写成文件从而避免失效和便于评估系统安全性。

描述:在评估过程中必须论证操作能力和安全性,以及在开发中牵涉的各方赋与的关心。为了能显示对开发的关注,以及为了保证任何时候都能验证安全证据,文档中编入了特别重要的措施。

重要的共同措施是编入指南和计算机辅助,即

a) 指南:

——规定分类计划;

——要求内容的检验表;

——确定文档的形式。

b) 借助计算机辅助的和组织化的项目库文档管理。

各种措施有:

a) 分开编写文档:

——要求方面的文档;

——系统方面的文档(用户文档);和

——开发文档(包括内部检验)。

b) 根据安全生存周期对开发文档分类。

c) 定义标准化的文档模块,根据这种文档模块可以汇编文档。

d) 清楚的标记文档的各组成部分。

e) 更新定型版本。

f) 选择清楚的和可理解的描述方式:

——确定用的形式符号;

——介绍、证明和表达意图用的自然语言;

——举例用的图形表示;

——图形元素的语义定义;和

——专用字的目录。

参考文献:

IEC 61566:1997 工业过程测量和控制 应用软件的文档.

基于工业计算机系统的 EWICS 欧洲工厂,TC 7:计算机安全 软件开发和系统文档编制. Verlag TÜV Rheinland Köln,1985.

化学过程的安全自动化指南. CCPS,AIChE,New York,1993.

Entwicklungstechnik sicherheitsverantwortlicher Software in der Eisenbahn-Signaltechnik. U. Feucht,Informatik-Fachberichte 86,Springer Verlag,Berlin,184-195,1984.

Richtlinie zur Erstellung und Prüfung sicherheitsrelevanter Software. K. Grimm,G. Heiner,Informatik Fachberichte 86,Springer Verlag,Berlin,277-288,1984.

B.1.3 分离开安全相关系统与非安全相关系统

注:在 GB/T 20438.2—2006 的表 B.1 和表 B.6 中引用了本技术/措施。

目的:为了防止系统的非安全部分以不需要的方式影响安全部分。

描述:在规范中,应判定是否能够把安全相关系统和非安全相关系统分离开。应写清楚对两个部分的衔接的规范。一份清楚的规范可减少测试安全相关系统的工作量。

参考文献:

化学过程的安全自动化指南. CCPS,AIChE,New York,1993.

B.1.4 多种硬件

注:在 GB/T 20438.2—2006 的表 A.16、表 A.17 和表 A.19 中引用了本技术/措施。

目的:使用具有不同失效率和失效型式的多种元件,检测受控设备运行过程中的系统失效。

描述:对一个安全相关系统的各个通道,使用不同型号的部件。这将减少共同原因失效(例如过压、电磁干扰)的概率并提高检测这种失效的概率。

由于存在执行一个所需功能的不同方法,例如不同的物理原理,从而提供了解决同一问题的其他渠道。存在几种多样化形式。功能多样化使用了各种办法来达到同样的结果。

参考文献:

化学过程的安全自动化指南. CCPS,AIChE,New York,1993.

B.2 E/E/PES 安全要求规范

整体目标:为了要编制一份尽可能完备、无歧义、无矛盾、检验简单的规范。

B.2.1 结构化规范

注:在 GB/T 20438.2—2006 的表 B.1 和表 B.6 中引用了本技术/措施。

目的:通过建立部分要求的一种层次结构来减少复杂性。为了避免各要求之间的接口失效。

描述:本技术把功能规范构建成许多部分要求,从而使这些部分要求之间存在的关系变得最可能简单、可视。这种分析一步一步更加精确直到能够区别那些小而清晰的部分要求为止。最后的精确结果是一种部分要求的层次结构,它提供了总体要求的详细说明的一个框架。这种方法着重于各部分要求之间的接口,它对避免接口失效特别有效。

参考文献:

结构化分析和系统规范. T. De Marco,Youdon Press,Englewood Cliffs,1979.

ESA PSS 05-02,需求调研阶段用户指南,European Space Agency,1989.

B.2.2 形式化方法

注 1:关于专用形式方法的细节可参看 C.2.4。

注 2:在 GB/T 20438.2—2006 的表 B.1 和表 B.6 中引用了本技术/措施。

目的:为了无歧义地和一致地表达规范,使之能检测出错误和遗漏。

描述:形式方法提供了在系统的规范或者设计的某一阶段编制描述该系统的一种方法。为了检测各种类型的不一致性或者错误,编成的描述采用了一种数学形式并容易进行数学分析。此外,在某些情况下还能用机器分析该描述,机器分析精确地相似于使用一个编译器对一个源程序进行语法检查;或者为了显示所描述的系统各方面的行为,在某些情况下还能把该描述制作成动画。动画能为系统满足实际要求以及在形式上所规定的要求提供额外的置信度,这是因为动画提高了人们对规定的系统行为的识别能力。

形式方法通常将提供一种符号表示法(离散数学常用的某种形式),用来在该表示法中推导描述的一种技术,以及用来检查各种属性正确性的描述的各种分析形式。

从一种数学上的形式规范开始,通过逐步精确的系列分析,可以把设计变换成一种逻辑电路设计。

参考文献:

关键计算机系统的可靠性 3. P. G. Bishop et al,Elsevier Applied Science,1990,ISBN 1-85166-544-7.

HOL:一种面向机器的高阶逻辑的公式化. M. Gordon,University of Cambrige Technical Report Number 68,1985.

B.2.3 半形式化方法

目的:为了清楚地和一致地表达一份规范的各部分,以便能检测出某些错误和遗漏。

注:在 GB/T 20438.3—2006 的表 A.1、表 A.2 和表 A.4 以及 GB/T 20438.2—2006 的表 B.1、表 B.2 和表 B.6 中引用了本技术/措施。

B.2.3.1 一般要求

目的:为了证明设计满足它的规范。

描述:半形式方法为在开发一个系统的某个阶段编制该系统的描述(即规范、设计或者编码)提供了一种方法。在某些情况下,可用机器分析该描述,或者为了显示系统各方面的行为,可把该描述制作成动画。此动画可对系统满足实际要求以及规定的要求提供了额外的置信度。

在下面的条款中将描述两种半形式方法。

B.2.3.2 有限状态机/状态转换图

注:在 GB/T 20438.3—2006 表 B.5 和表 B.7 中引用了本技术/措施。

目的:模型化、详细规定或者实现一个系统的控制结构。

描述:能够用它们的状态、它们的输入和它们的行动来描述许多系统。当处于状态 S1 时,在接受到输入 I 时,一个系统将会执行动作 A 并转换到状态 S2。通过描述一个系统处于每个状态中时,每个输入导致该系统的动作,我们就能彻底地描述一个系统。产生的系统模型叫作一个有限的状态机。常常是把它画成一个所谓的状态转换图,此图显示系统怎样从一个状态转移到另一个状态;或者把它画成一个矩阵,在矩阵中,维数是状态和输入,矩阵元包含当系统处于给定状态中时,由接受输入导致的动作和新状态。

一个复杂系统或者具有一个自然结构的系统的情况可以在一个分层的有限状态机中反映出来。

可以检验表示成一个有限状态机的规范或者设计的

——完备性(在每个状态、对每个输入,系统必定有一个动作和新状态);

——一致性(每个状态/输入对只描述一次状态改变);以及

——可达性(是否能通过任何输入序列从一个状态到达另一状态)。

它们是关键系统的一些重要属性。很容易开发支持这些检查的工具。也存在用于验证一个有限状态机实现或者制作一个有限状态机模型的动画的算法,这种算法允许自动生成测试实例。

参考文献:

有限状态机理论的介绍. A. Gill, McGraw-Hill, 1962.

B.2.3.3 时间 Petri(佩特里)网

注:在 GB/T 20438.3—2006 的表 B.5 和表 B.7 中引用了本技术/措施。

目的:通过分析和再设计模型化系统行为的有关方面,评估及尽可能地提高安全性和操作要求。

描述:Petri(佩特里)网属于图形理论模型一类,它适用于在呈现并发性和有异步行为的系统中表示信息和控制流。

Petri 网是一个位置和变迁网。位置可被"标记"或"不标记"。当输入该网的所有输入位置被标记时,就"启动"一次变迁。当启动变迁时,就允许(而不是强迫)"点火"。如果它点火,引起变迁的输入位置就变成未标记的了,并且代之以变迁产生的每个输出位置被标记。

在模型中可把潜在的危险表示成特定的状态(标记)。可把 Petri 网模型扩展成可供系统的定时特征用。虽然"传统的"Petri 网集中在控制流方面,已经提出了把数据流包括到模型中去的几种扩展方案。

参考文献:

Petri 网:属性、分析和应用. T. Murata, Proc. IEEE 77(4), 541-580, April 1989.

使用 Petri 网进行安全分析. N. G. Leveson, J. L. Stolzy, Proc. 15th Ann. Int. Symp on Fault-Tolerant Computing, 358-363, IEEE, 1985.

使用 Petri 网对无人驾驶的地铁系统进行安全分析. M. El. Koursi, P. Ozello, Proc. SAFECOMP'92, 135-139, Pergamon Press, 1992.

网络理论和应用. W. Brauer (ed.), Lecture Notes in Computer Science, Vol. 84, Springer Verlag, 1980.

Petri 网理论和建立系统模型. J. L. Peterson, Prentice Hall, 1981.

基于计时 Petri 网的实时软件的要求规范和分析的一个工具. S. Bologna, F. Pisacane, C. Ghezzi, D.

Mandrioli, Proc. SAFECOMP 88, 9-11, November1988. Fulda, Fed. Rep. of Germany, 1988.

B.2.4 计算机辅助的规范工具

注：在 GB/T 20438.2—2006 的表 B.1 和表 B.6 中以及在 GB/T 20438.3—2006 的表 A.1 和表 A.2 中引用了本技术/措施。

B.2.4.1 一般要求

目的：为了使用形式规范技术以便于自动检测歧义性和完备性。

描述：本技术可产生一个数据库形式的规范，可自动被检查这种形式从而评估一致性和完备性。规范工具能把用户指定的系统的各种特征制作成动画。通常，技术支持不仅建立规范，也能建立设计和项目生命周期的其他阶段。可根据以下条款对规范工具进行分类。

B.2.4.2 为了不用特殊方法所面向的工具

目的：通过提供提示和各相关部分间的连接，帮助用户编写一份好的规范。

描述：规范工具可接替用户的一些日常工作并支持项目管理。它不强求任何特殊的规范方法。在方法方面的相对独立性允许用户在建立规范时有很大的自由度，同时为用户提供所必需的专门支持很少，但会使得要精通系统更为困难。

参考文献：

Integrierte Rechnerunterstützung für Entwicklung, Projektmanagement und Prodnktverwaltung mit EPOS. R. Lauber, P. Lempp, Elektron. Rechenanlagen 27, Heft2, 68-74, 1985.

B.2.4.3 面向模型的层次分析程序

目的：通过保证各抽象层次的动作和数据的描述之间的一致性，帮助用户编写一份好的规范。

描述：本方法给出了各抽象层次(精密度)下，要求的系统的一个功能表达式(结构化分析)。各个层次下的分析对动作和数据两者都有作用。在层次之间和同一层次的两个功能单元(模块)之间评估歧义性和完备性是可能的。

参考文献：

定义要求的结构化分析. D. T. Ross, K. E. Schomann jr, IEEE Trans. on SE, January 1977.

B.2.4.4 实体模型

目的：关注系统中的各实体以及它们之间的关系，帮助用户编写一份好的规范。

描述：把要求的系统描述成一些对象和它们之间的关系的一个集合。工具使我们能确定系统能解释哪些关系。通常，这些关系允许描述对象、数据流的层次结构、数据之间的关系，以及哪些数据须经历一些生产过程。为了用于过程控制，已对传统的程序进行了扩充。检查能力和对用户的支持与所说明的各种关系有关。另一方面，大量可能的表达式使本技术的应用变得很复杂。

参考文献：

PSL/PSA 结构化文档和信息处理分析的计算机辅助技术. D. Teichroew, E. A. Hershey, IEEE Trans on SE, Jan 1977.

计算机辅助的软件开发. D. Teichroew, E. A. Hershey, Y. Lamamoto, Beitrag in: Verfahren und Hilfsmittel fur spezifikation und Entwurf Von Prozcßautomatisierungs-systemen. Hommel (ed.), Bericht KfK-PDV 154, Kernforschungszentrum Karlsruhe, 1978.

PCSL und ESPRESO -zwei Ansätze zur Formalisierung der Prozessrechner Software-spezifikation. J. Ludewig, GI-Fachtagung Prozessrechner 1981, Informatik-Fachberichte Bd. 39, Springer Verlag, Berlin. 1981.

B.2.4.5 诱因和回答

目的：通过识别激励-响应关系，帮助用户编写一份好的规范。

描述：系统的各对象之间的关系用“诱因”和“回答”的一种表示法来说明。使用了一种简单和容易的扩充语言，这种语言包含有代表对象、关系、特性和结构的语素。

参考文献：

实时处理要求的一种需求工程方法. M. W. Alford, IEEE Trans on SE, January 1977.

规范系统 X-SPEX—介绍和经验. G. Dahll, J. Lahti, Proc. SAFECOMP'83, 111-118.

B.2.5 检查表

注：在 GB/T 20438.2—2006 的表 B.1、表 B.2 和 GB/T 20438.3—2006 的表 A.10 和表 B.8 中引用本技术/措施。

目的：在安全生命周期阶段，为保证全面的涵盖而不用制定严格的要求，从而引起对系统各重要问题的注意并管理系统各重要方面的认证性评价。

描述：由执行检查表的人来回答一系列问题。许多问题具有普遍性，评估员对这些问题应视为最适合于正在评估的特定系统那样来解释它们。检查表可用于整体安全生命周期、E/E/PES 和软件安全生命周期的各阶段并特别适合作为帮助功能安全评估的一种工具。

为了适应正在确认的系统繁多的类型，大多数检查表包含了能适用于许多类型的系统的一些问题。结果，在使用的检查表中的问题可能和正在讨论的系统无关，这时就应不管这些问题。这等于说，对于一个特定的系统需要补充一个标准的检查表，这个表包含的问题是专门针对正在讨论的系统的。

在任何一种情况下，使用检查表的关键取决于工程师选择和应用检查表的专门知识和判断能力。工程师选择检查表采取的决定和任何附加的或者多余的问题都应全部编入文档并证明是合理的。目的是要保证能评审检查表的应用并且保证使用相同的判据都应能达到同样的结果。

在完成一份检查表时，对象要尽可能简洁。当需要扩充证明时，应通过引用附加文档来进行这种扩充。编写每个问题的结果，应使用合格、不合格和不确定或者一些类似的受限的应答集合。这种简洁大大简化了获得有关检查表评估结果的全部结论的程序。

参考文献：

IEC 61346(所有部分)　工业系统、设备和装置以及工业产品　结构原理和参考命名.

IEC 60880:1986　核电站安全相关系统中计算机的软件.

化学过程的安全自动化指南. CCPS, AIChE, New York, 1993.

在有关安全性的应用中的可编程电子系统(PES). Health and Safely Executive, UK, 1987.

关键计算机系统的可靠性 2. F. J. Redmill, Elsevier Applied Science, 1989, ISBN 1-85166-381-9.

软件测试工艺. G. Myers, Wiley&Sons, New York, 1979.

B.2.6 规范的检查

注：在 GB/T 20438.2—2006 的表 B.1 和表 B.6 中引用了本技术/措施。

目的：为了避免规范中的不完备性和矛盾。

描述：检查是一种普通的技术，在这种技术中，由一个独立的小组来评估一份规范文档的各种质量。检查小组向拟制者提问题，拟制者必须圆满地回答提问人。检验小组的成员不包括规范的拟制人员。要求的独立程度由系统要求的安全完整性等级确定。独立的检查人员应能以一种无可非议的形式并不用再涉及任何规范就可重建系统的运行功能。他们还必须检验在操作和组织措施中包括的所有相关的安全和技术问题，这个程序已被证明它在实际应用中是很有效的。

参考文献：

软件测试工艺. G. Myers, Wiley&Sons, New York, 1979.

IEC 61160:1992　形式设计评审.

B.3 E/E/PES 的设计和开发

整体目标：为了产生一个符合规范的安全相关系统的稳定设计。

B.3.1 遵循指南和标准

注：在 GB/T 20438.2—2006 的表 B.2 中引用了本技术/措施。

目的：为了遵循应用领域标准(GB/T 20438 中未规定)。

描述:在设计安全相关系统的过程中应遵循一些指南。这些指南首先可实现无失效的安全相关系统,其次可简化其后的安全确证。它们可以是通用的,某个项目专用的或者只专用于某单个阶段。

参考文献:

EWICS 基于工业计算机系统的欧洲工厂,TC 7:与安全有关的计算机 软件开发和系统文档编制. Verlag TÜV Rheinland,Köln,1985.

化学过程的安全自动化指南. CCPS,AIChE,New York,1993.

Deutsche Bundesbahn: Richtlinien-Entwürfe 42500 to 42550 für das Handbuch"Grundsätze zur technischen Zulassung in der Signal-und Nachrichtentechnik". Bundesbahn-Zentralamt München,August 1987.

Richtlinie zur Erstellung und Prüfung sicherheitsrelevanter Software. K. Grimm,G. Heiner,Informatik Fachberichte 86,Springer Verlag,Berlin,277-288,1984.

B.3.2 结构化设计

注:在 GB/T 20438.2—2006 的表 B.2 和表 B.6 中引用了本技术/措施。

目的:通过建立部分要求的层次结构来减少复杂性,避免各要求之间的衔接失效,简化验证。

描述:当设计软件时,应使用专门的判据或者方法。例如需要:

——一种分层的结构化电路设计;

——使用已生产的和经测试的电路元件。

类似地,当设计软件时,使用结构图表使之能够建立软件模块的一种无歧义的结构。这种结构可显示模块的相互关系,模块之间传递的精确数据和模块之间存在的精确控制。

参考文献:

实时系统的结构化开发(共 3 卷). D. T. Yourdon,P. T. Yourdon Press,1985.

实时系统的软件设计. J. E. Cooling,Chapman and Hall,1991.

基本的系统分析. St. M. McMenamin,F. Palmer,Yonrdon Inc,1984.

在开发基于软件的大型航空电子系统中使用的结构化方法. D. J. Hatley,Proceedings DASC,Baltimore,1984.

JSD 的一个评述. J. R. Cameron,IEEE Trans SE-12 No. 2,February 1986.

系统开发. M. Jackson,Prentice-Hall,1983.

MASCOT 3 用户指南. MASCOT Users Forum,RSRE,Malvern,England,1987.

实时系统的结构化开发(共 3 卷). P. T. Ward,S. J. Mellor,Yourdon Press. 1985.

要求定义的结构化分析. D. T. Ress,K. E. Schoman Jr,IEEE Trans. Software Eng,Vol SE-3,6-15,1977.

结构化分析(SA):通信理念的一种语言. D. T. Ross,IEEE Trans. Software Eng,Vol,SE-3(1),16-34.

SADT 的应用和扩展. D. T. Ross,Computer,25-24,April 1985.

结构化分析和设计技术——在安全相关系统上的应用. W. Heins,风险评估和控制课件,模块 B1,第 11 章,Delft University of Technology,Safety Science Group,Po Box 5050,2600 GB Delft,Netherlands,1989.

IEC 61346(所有部分) 工业系统、设备和装置以及工业产品 结构原理和参考命名符.

B.3.3 使用经充分试验过的部件

注:在 GB/T 20438.2—2006 的表 B.2 和表 B.6 中引用了本技术/措施。

目的:通过使用具有专门特性的部件减小大量首次和未检测到的故障的风险。

描述:在安全性方面根据部件的可靠性,生产厂选择经充分试验过的部件(例如使用经运行测试过的物理单元来满足高安全要求,或者在安全存储器中存储安全的程序)。存储器的安全性与未经批准的

访问和环境影响(电磁兼容性、辐射等)以及在发生一次失效的事件中部件的响应有关。

参考文献:

工业使用的可靠性测试. W. T. Greenwood, Computer10(7), 26-30, 1977.

独立实验室:Caveat Emptor. E. Rubinstein, IEEE Spectrum, 14(6), 44-50, 1977.

安全技术中的微机——面向开发者和生产者的一种工具. H. Holscher, J. Rader, Varlag TÜV Rheinland, Köln, 1986, ISBN 3-88585-315-9.

IEC 61163-1:1995 可靠性应力屏蔽 第1部分:批量生产的可修复零件.

Zuverlässigkeit elektronischer Komponenten. T. Bajenscu, VDE-Verlag, Berlin, 1985.

B.3.4 模块化

注:在 GB/T 20438.2—2006 的表 B.2 和表 B.6 中引用了本技术/措施。

目的:为了降低与分系统之间连接有关的复杂性和避免失效。

描述:在设计的各层次上的每个分系统都被清楚地定义,并且限制了它们的大小(仅几个功能)。分系统之间的接口尽可能保持简单,并把交叉部分(即共享数据、信息交换)减到最小。还限制了各个分系统的复杂性。

参考文献:

EWICS 基于工业计算机系统的欧洲工厂,TC 7:与安全有关的计算机 软件开发和系统文档编制. TÜV Rheinland, Köln, 1985.

软件测试工艺. G. J. Myers, Wiley & Sons, New York , 1979.

软件可靠性——原理和实践. G. J. Myers, Wiley—Interscience, New york, 1976.

实时系统的软件设计. J. E. Cooling, Chapman and Hall, 1991.

B.3.5 计算机辅助设计工具

注:在 GB/T 20438.2—2006 的表 B.2 和表 B.6 及 GB/T 20438.3—2006 的表 A.4 中引用了本技术/措施。

目的:为了更系统化地执行设计规程。包括已经可用的和经测试过的合适的自动结构元素。

描述:在硬件和软件设计过程中当可买到它们并由系统的复杂性证明它们是合理的时候,应使用计算机辅助设计工具(CAD)。应通过专门测试、一段漫长的成功应用历史或者它们的输出的单独验证来证明这种工具应用于设计安全相关系统的正确性。

参考文献:

验证—实际问题. J. T. Webb and D. T. Mannering, SARSS 87, November 1987, Altrincham, England, Elsevier Applied Science, 1987 ISBN 1-85166-167-0.

反应堆保护系统的软件设计和确认的一次体验. S. Bologna, E. de Agostino et al, IFAC Workshop, SAFECOMP 1979, Stuttgart, 16-18 May 1979, Pergamon Press, 1979.

B.3.6 模拟

注:在 GB/T 20438.2—2006 的表 B.2 和表 B.6 中引用了本技术/措施。

目的:为了对电气/电子电路的部件功能性能和正确地确定尺寸两方面进行一次系统、全面的检查。

描述:借助一个软件行为模型,在一台计算机上模拟安全相关系统电路的功能。电路的各个部件每个都有它们自己的模拟行为,并且通过观察每个部件的边缘数据来检验这些部件连成的电路的响应。

参考文献:

Proc. Working Conference Prototyping(原型设计工作会议会刊). Namur, October 1983, Budde et al, Springre Verlag, 1984.

使用一个信息系统的一种可执行的规范语言. S. Urban et al, IEEE Trans Software Engineering, Vol. SE-11 No7, July 1985.

实时软件的验证和确认. W. J. Quirk(ed.)Springer Verlag, Berlin, 1985.

VDI-Gemeinschaftsausschuß Industrielle Systemtechnik: Software-Zuverlässigkeit. VDI-Verlag,

1993.

B.3.7 检查(复审和分析)

注：在 GB/T 20438.2—2006 的表 B.2 和表 B.6 中引用了本技术/措施。

目的：为了揭示规范和实现之间的不一致。

描述：检验和评价安全相关系统的规定功能以保证安全相关系统符合规范中给出的要求。有关实现的和产品使用的任何疑点都编入文档，因此这些疑点可得到解决。在检查过程中，同走查相比，作者是被动的，而检查员是主动的。

参考文献：

软件测试工艺. G. J. Myers, Wiley & Sons, New York, 1979.

关键计算机系统的可靠性 3. P. G. Bishop et al, Elsevier Applied Science, 1990, ISBN 1-85166-544-7.

VDI-Gemeinschaftsausschuß Industrielle Systemtechnik: Software-Zuverlässigkeit. VDI-Verlag, 1993.

ANSI/IEE Std. 1028:1997 软件评审和审核的 IEEE 标准.

B.3.8 走查

注：在 GB/T 20438.2—2006 的表 B.6 中引用了本技术/措施。

目的：为了揭示规范和实现之间的不一致。

描述：检验和评价安全相关系统设计的规定功能以保证安全相关系统符合规范中给出的要求。有关产品实现和使用的疑点和潜在的弱点都编入文档以便能得以解决。和检查相反，在走查过程中，作者是主动的而检查员是被动的。

参考文献：

工业系统技术：软件可靠性 3. P. G. Bishop el al, Elsevier Applied Science, 1990, ISBN 1-85166-544-7.

Methodisches Testen Von Programmen. G. T. Myers, Oldenbourg Venlag, München, Wien, 1987.

VDI-Gemeinschaftsausschuß Industrielle Systemtechnik: Software-Zuerlässigkeit. VDI-Verlag, 1993.

ANSI/IEE Std. 1028:1997 软件评审和审核的 IEEE 标准.

B.4 E/E/PES 操作和维护规程

整体目标：拟制有助于避免安全相关系统在操作和维护过程中失效的规程。

B.4.1 操作和维护说明书

注：在 GB/T 20438.2—2006 的表 B.4 中引用了本技术/措施。

目的：为了避免安全相关系统在操作和维护过程中出错。

描述：用户说明书包含了使用和维护安全相关系统的最基本的信息。在特殊情况下，这些说明通常还包括安装安全相关系统的例子。所有说明必须要容易读懂。复杂的操作步骤和依从关系可用图表描述。

参考文献：

化学过程的安全自动化指南. CCPS, AIChE, New York 1993.

B.4.2 用户友善性

注：在 GB/T 20438.2—2006 的表 B.4 中引用了本技术/措施。

目的：为了减小在操作安全相关系统过程中的复杂性。

描述：安全相关系统的正确操作与人的操作水平有关。通过考虑有关的系统设计和工作场地的设计，安全相关系统的开发者必须保证：

——尽量减少人为干预的需要；

——必要的干预应尽可能简单；

——把因操作员错误产生的危害的潜力降到最小；

——根据人机工程学的要求来设计干预设备和指示装置；

——操作员设备的标注十分简易，使用很直观；

——即使是在恶劣情况下，操作员也不用过度操劳；

——干预操作规程和设备的培训应适合受训用户的知识水平和机能。

B.4.3 维护友善性

注：GB/T 20438.2—2006 的表 B.4 中引用了本技术/措施。

目的：为了简化安全相关系统的维护规程和设计有效诊断和修理的必要方法。

描述：通常都是根据最后期限在困难的环境条件和压力下进行预防性维护和修理。因此，安全相关系统的开发人员应保证：

a) 即使需要，也要尽可能少要安全维护措施，从理论上讲，完全不必要；

b) 应包含用于那些不可避免的修理所需的足够的、切合实际并易于掌握的诊断工具——这些工具应包括所有必要的界面；

c) 如果必须开发或者购买这些单独的诊断工具，那么，到时候应能使用上这些工具。

B.4.4 受限的操作可能性

注：在 GB/T 20438.2—2006 的表 B.4 中引用了本技术/措施。

目的：为了减少常规用户的操作可能性。

描述：本方法通过以下几点减少了操作可能性：

——限制特殊操作模式中的操作，例如通过按键开关；

——限制操作元素数量；

——限制通常可能的操作模式数量。

参考文献：

化学过程的安全自动化指南. CCPS, AIChE, New York, 1993.

B.4.5 只能由熟练的操作员操作

注：在 GB/T 20438.2—2006 的表 B.4 和 B.6 中引用了本技术/措施。

目的：为了避免因使用错误引起的操作失效。

描述：要把安全相关系统的操作员培训到能适应安全相关系统的复杂性和安全完整性等级的程度。培训包括学习生产过程的基础知识和了解安全相关系统和受控设备之间的关系。

参考文献：

化学过程的安全自动化导则. CCPS, AIChE, New York, 1993.

B.4.6 防止操作员出错

注：在 GB/T 20438.2—2006 的表 B.6 中引用了本技术/措施。

目的：防止操作员操作系统时的各种错误。

描述：通过似真性检查或者监视受控设备，可检测输入错误(值、时间等)。为了把这些设备汇集到设计中，在相当早的一个阶段就有必要说明哪些输入是可能的和哪些输入是允许的。

B.4.7 (未用)

B.4.8 防止修改

注：在 GB/T 20438.2—2006 的表 A.18 中引用了本技术/措施。

目的：利用一些技术手段来防止修改安全相关系统的硬件。

描述：例如利用传感器信号的似真性检查、技术过程检测以及自动起动测试，自动地检测修改或变换。当检测到一个修改时，就采取应急动作。

B.4.9 输入确认

注：在 GB/T 20438.2—2006 的表 A.18 和表 A.19 中引用了本技术/措施。

目的：在启动受控设备之前，操作员自己检测操作过程中的一个错误。

描述：在一个输入被传送给受控设备之前，经安全相关系统传送给受控设备的一个输入将反回给操

作员,使得操作员有可能检测和校正一个错误。同时系统设计应考虑到异常的、无缘无故的人员动作的速度上下限以及人的反应倾向。这样可以避免例如操作员按键比预计的快而引起系统把一次双击读成一次单击,或者因为系统(显示)对快击反应太慢而把两次按键误读成一次。在输入关键数据时,连续地多次同样的击键并不有效。无限多次按"enter(回车)"键或者"yes(肯定)"键也绝不会导致系统不安全的动作。

当操作员还未作决定并让系统等待时,为了提供必要的条件,应包含具有多个选择问题(是/否问题等)的暂停程序。

除非在设计软件及硬件时考虑到了这种需要,否则重新启动一个安全可编程电子系统的能力将使系统十分脆弱。

参考文献:

DIN V VDE 0801:Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben(安全相关系统中的计算机原理). Beuth-Verlag,Berlin,1990.

B.5 E/E/PES 集成

整体目标:为了避免在集成阶段产生失效以及揭示在本阶段和前面阶段产生的失效。

B.5.1 功能测试

注:在 GB/T 20438.2—2006 的表 B.3 和表 B.5 中以及 GB/T 20438.3—2006 的表 A.5、表 A.6 和表 A.7 中引用了本技术/措施。

目的:揭示在规范和设计阶段产生的失效。避免在软件和硬件的实现和集成期间产生失效。

描述:在功能测试过程中,应审查和观察是否已达到规定的系统特性。提供给系统的输入数据足以说明一般预期的工作特性。观察输出并把它们的响应同规范给出的响应作对比。同规范的偏差以及一个不完整的规范的表象都被写进文档。

多通道结构设计的电子部件的功能测试包括正在使用预先已验证过的伙伴部件进行测试的成品部件。除此之外,建议结合同一批的其他伙伴部件一起测试成品部件,以便揭示共同模式故障,而用其他办法则该类故障会被掩藏而不能揭示出来。

另外,系统的工作能力必须足够,参看 C.5.20 中的指导。

参考文献:

功能程序测试和分析. W. E. Howden,McGraw-Hill,1987.

软件测试工艺. G. J. Myers,Wiley&Sons,Nes York,1979.

关键计算机系统的可靠性 3. P. G. Bishop et al,Elsevier Applied Science,1990,ISBN 1-85166-544-7.

B.5.2 黑盒测试

注:在 GB/T 20438.2—2006 的表 B.3、表 B.5 和表 B.6 中和 GB/T 20438.3—2006 的表 A.5、表 A.6 和表 A.7 中都引用了本技术/措施。

目的:为了检验实际功能条件下的动态行为。为了揭示失效从而满足功能规范和评估实用性和键壮性。

描述:在一个规定的环境中,利用规定的测试数据(这些数据是根据制定的判据从规范中系统导出的)执行一个系统或者程序的功能。这将揭露出系统的行为并允许同规范进行比较。不需使用系统内部结构的知识来指导测试。测试的目的是要确定功能单元是否能正确执行规范要求的所有功能。形成等价类别的技术是黑盒测试数据判据的一个例子。借助规范可把输入数据空间分成专用的输入值范围(等值类)。于是由下列情况构成各种测试实例:

——来自允许范围的数据;

——来自不允许范围的数据;

——来自范围极限的数据；

——极值；

——以上各类的组合。

为了选择各种测试活动(模块测试、集成测试和系统测试)中的测试实例，其他判据也可能是有效的。例如，在一个确认的框架内系统测试的判据“极值工作条件”是可信的。

参考文献：

功能测试和分析. W. E. Howden, MeGraw-Hill Book Company, New York, 1987.

软件测试和确认技术. E. Miller, W. E. howden, IEEE Computer Society, New York, 1978.

软件测试工艺. G. J. Myers, Wiley&Sons, New York, 1979.

Methodik Systematischen Testens von Software. K. Grimm, 30(4), 1988.

VDI-Gemeinschaftsausschuß Industrielle System-technik: Software-Zuverlässigkeit. VDI-Verlag, 1993.

B.5.3 统计测试

注：在 GB/T 20438.2—2006 的表 B.3、表 B.5 和表 B.6 中引用了本技术/措施。

目的：为了检查安全相关系统的动态行为和评估它的实用性和健壮性。

描述：本方法使用根据实际操作输入预期的统计分布——操作分布图选择的输入数据来测试一个系统或程序。

参考文献：

通过环境模拟测试软件(CONTESSE 报告). 1998 年 12 月之前可从下处得到：Ray Browne, CIID, DTI, 151 Buckingham Palace Road, London, SW1W 9SS, UK, 1994.

开发和验证可靠的过程计算机软件的有关问题. W. Ehrenberger, IFAC-IFIP Conference Proceedings, 35-48, 1980.

验证和确认实时软件. W. J. Quirk(ed), Springer Verlag, Belin, 1985.

VDI-Gemeinschaftsausschuß Industrielle Systemtechnik: Software-Zuverlässigkeit. VDI-Verlag, 1993.

关键计算机系统的可靠性 1. F. J. Redmill, Elsevier Applied Science, 1988, ISBN 1-85166-203-0.

关键计算机系统的可靠性 3. P. G. Bishop et al, Elsevier Applied Science, 1990, ISBN 1-85166-544-7.

B.5.4 现场经验

注 1：一种类似的办法另见 C.2.10，一种统计方法另见附录 D，二者都在软件的范围中。

注 2：在 GB/T 20438.2—2006 的表 B.3 和表 B.5 中引用了本技术/措施。

目的：无论在 E/E/PES 集成过程中或者 E/E/PES 安全性确认过程中或二者中，各种应用得到的现场经验可用作避免故障的一种办法。

描述：经验表明不会出现或仅可能出现不重要故障的部件或分系统，在许多不同应用中和足够长时期内使用时都不会有实质性改变。特别是对具有许多可能功能的复杂部件(例如操作系统、集成电路)而言，开发者应注意哪些功能已经历现场测试过。例如考虑故障检测的自测试例程：如在操作周期内，没有发生过硬件故障，就不能说例程已被测试过，因为它们还从未经受过故障检测。

为了使用现场经验，必须满足以下要求：

——规范不变；

——在各种应用中要有 10 个系统；

——10^5 个运行小时以及至少 1 年的运行历史。

通过卖方和(或)操作公司的文件证明现场经验，该文件必须至少要包括：

——系统和它的部件的准确名称，包括硬件的版本控制；

——用户和应用时间；

——运行时数；

——选择为完成证明的系统和应用的规程；

——故障检测、故障登记以及故障排除规程。

参考文献：

DIN V VDE 0801 A1:Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben(安全相关系统中的计算机原理). Änderung 1 zu DIN V VDE 0801/01.90. Beuth-Verlag,Berlin,1994.

化学过程的安全自动化指南. CCPS,AIChE,New York,1993.

B.6 E/E/PES 安全性确认

整体目标:为了证明 E/E/PE 安全相关系统符合 E/E/PES 安全要求规范。

B.6.1 在环境条件下测试功能

注：在 GB/T 20438.2—2006 的表 B.5 中引用了本技术/措施。

目的:为了评估安全相关系统是否能耐受典型的环境影响。

描述:把系统置于各种环境条件下(例如根据 IEC 60068 系列或 IEC 61000 系列标准),评价安全功能的可靠性(以及和上述标准的相容性)。

参考文献：

IEC 61000-4-1:1992 电磁兼容性(EMC) 第 4 部分:测试和测量技术 第一章:抗扰性测试概述.

IEC 60068-1:1988 环境试验 第 1 部分:概述和指南.

关键计算机系统的可靠性 3. P. G. Bishop et al, Elsevier Applied Science, 1990, ISBN 1-85166-544-7.

B.6.2 浪涌抗扰性测试

注：在 GB/T 20438.2—2006 的表 B.5 和 B.6 中引用了本技术/措施。

目的:为了检验安全相关系统应付峰值浪涌的能力。

描述:系统装入一个典型的应用程序,使所有外部线路(所有数字、模拟和串行接口以及总线连接和电源等)都必须得到标准的噪声信号。为了获得一个定量的说明,明智的作法是仔细地逼近浪涌极限。如果功能失效,所选择的噪声类型则不符合要求。

参考文献：

测试浪涌耐受能力(SWC)的指南. ANSI C. 37.90-1974.

IEC 61000-4-5:1995 电磁兼容性(EMC) 第 4 部分:测试和测量技术 第 5 章:浪涌抗扰性测试.

B.6.3 (未用)

B.6.4 静态分析

注：在 GB/T 20438.2—2006 的表 B.5 和表 B.6 中及 GB/T 20438.3—2006 的表 A.9 中引用了本技术/措施。

目的:为了避免系统性故障,此故障可导致受试系统在工作多年后,无论早迟都会发生事故。

描述:这种系统性的和可能用计算机辅助的方法可检查原型系统的特有静态特性,从而可保证所讨论的要求的完备性、一致性和无歧义性(例如构造指南、系统规范和仪表数据表)。静态分析是可再现的。它适用于已达到完成一个良定义阶段的原型机。对于硬件和软件的静态分析的一些例子有:

——数据流的一致性分析(比如测试一个数据对象是否在任何地方都被解释成同一值);

——控制流分析(比如确定通路,确定不可存取代码);

——界面分析(比如调查各种软件模块之间变量转移);

——检测可疑的变量建立、引用、删除序列的数据流分析;

——遵守专用指南的测试(例如爬行距离和间隙,装配距离、物理单位安排、力学有关的物理单位,曾经引入的专用物理单位)。

参考文献:

关键计算机系统的可靠性 3. P. G. Bishop et al,Elsevier Applied Science,1990. ISBN 1-85166-544-7.

VDI-Gemeinschaftsausschuß Industrielle Systemtechnik:Software - Zuverlässigkeit. VDI-Verlag,1993.

B.6.5 动态分析

注:在 GB/T 20438.2—2006 表 B.5 和表 B.6 中以及 GB/T 20438.3—2006 的表 A.5 和表 A.9 中引用了本技术/措施。

目的:通过检查在正式完成前期阶段的一个样机的动态特性检测规范失效。

描述:通过把计划的工作环境下的一些典型的数据输入加给安全相关系统的一个接近运行的样机进行安全相关系统的动态分析。当观察到的安全相关系统的行为同要求的行为一致时,分析是满意的。必须校正安全相关系统的任何失效,然后必须重新分析新的运行方案。

参考文献:

关键计算机系统的可靠性 3. P. G. Bishop et al,Elsevier Applied Science,1990,ISBN 1-85166-544-7.

VDI-Gmerinschaftsausschuß Imdustrielle Systemtechnik:Software-Zuverlässigkeit. VDI-Verlag,1993.

B.6.6 失效分析

注:在 GB/T 20438.2—2006 的表 B.5 和表 B.6 中引用了本技术/措施。

B.6.6.1 失效模式和影响分析

目的:通过检验一个系统的部件的所有可能失效源以及确定这些失效对系统的行为和安全性的影响来分析一个系统的设计。

描述:通常是通过一次工程师会议来进行分析。依次分析一个系统的每个部件从而给出部件的一组失效模式,失效的起因和影响,检测程序和建议。如按建议,它们将作为采取的补救行动而编入文档。

参考文献:

IEC 60812:1985 系统可靠性分析技术 失效模式和影响分析(FMEA)规程.

系统可靠性工程方法:技术现状讨论. J. B. Fussel,J. S. Arend,Nuclear Safety 20(5),1979.

可靠性技术. A. E. Greenl,A. J. Bourne,Wiley-Interscience,1972.

故障树手册. W. E. Vesely et al,NUREG-0942,美国核管理委员会,核反应堆管理局,系统安全处,华盛顿,DC20555,1981.

B.6.6.2 因果图

注:在 GB/T 20438.3—2006 的表 A.10、表 B.3 和表 B.4 中引用了本技术/措施。

目的:为了以一种图表形式建立一个系统中可能形成的事件序列的模型,此事件序列是基本事件组合的结果。

描述:此技术可看成是故障树和事件树的一种组合分析。从一个致命事件开始,一幅因果图被向后和向前跟踪。在后向跟踪中,因果图相当于一个具有作为顶端事件给出的致命事件的故障树。在正向跟踪中,确定了由一个事件产生的可能后果。图形包含顶点符号,这些符号描述了从顶点开始沿各分支传播的条件。也能包括时间延迟。还可用故障树来描述这些条件。为了使图形更简洁,传播路线可用逻辑符号组合。定义了一个标准的符号集合以供因果图使用。可用这些图来计算发生某些致命后果的概率。

参考文献:

作为定量分析事故基础的因果图法. B. S. Nielsen,Riso-M-1374,1971.

B.6.6.3 事件树分析

注:在 GB/T 20438.3—2006 的表 B.4 中引用了此技术/措施。

目的:为了用图表形式建立事件序列的模型,这种事件序列是在一个起始事件之后,在一个系统中

可能产生的;并因此也为了指示可能产生的严重后果。

描述:在图顶部记录了紧随起始事件之后的相继事件有关的序列条件。在序列中从起始事件下面开始(它们是分析的目标)到第 1 个条件,画了一条线。在那里,图分成"Yes(是)"和"No(否)"两个分支,它们描述了以后的事件对条件的依从情况。对这两个分支来说,每个都以类似的方式继续到下一条件。但不是所有的条件都和分支有关。一个分支继续到序列的末端,并且按这种方式构造的树的每个分支代表一个可能的后果。根据序列中的条件的概率和数目,可用事件树来计算各种后果的概率。

参考文献:

事件树和及其在 PC 机上的处理. N. Limnious and J. P. Jeannette, Reliability Engineering, Vol. 18, No. 3, 1987.

B.6.6.4 失效模式,影响和危害性分析

注:在 GB/T 20438.3—2006 的表 A.10 和表 B.4 中引用了本技术/措施。

目的:为了确定在设计或者操作过程中需要特别关注哪些部件和采取哪些必要的控制措施,需要评定部件的危害性等级,这种等级通过单点失效将毁坏、损害系统或者使系统功能下降。

描述:有许多评定危害性等级的方法。最费劲的方法是美国汽车工程师协会(SAE)在 ARP926 中描述的方法。在该规程中,由发生在一个危害性模式中每百万次运行过程中预定的某种特殊类型的失效数来表示任一部件的危害性数。危害性数是 9 个参数的一个函数,它们中的大多数都是必须测量的。确定危害性的一种很简单的方法是部件失效的概率乘以可能发生的损坏。这种方法类似于简单的风险因数评估法。

参考文献:

失效模式、影响和危害性分析(FMECA)的设计分析规程. Aerospace Recommended Practice (ARP)926, Society of Automotive Engineers(SAE), USA, 15 September 1967.

IEC 60812:1985 系统可靠性分析技术 失效模式和影响分析(FMEA)的规程.

B.6.6.5 故障树分析

注:在 GB/T 20438.3—2006 的表 B.4 中引用了本技术/措施。

目的:帮助分析将会导致危险的严重结果的事件或事件组合。

描述:从一个可能直接引起危险或者严重后果的事件("顶事件")开始,沿一条树路径执行分析。使用逻辑算符(与,或等)描述起因的组合。按同样的方法分析中间原因,等等,在停止分析处返回到基本事件。

本方法是图形法并使用了一组标准化的符号来画事件树。本技术原来主要用来分析硬件系统,但也可用于软件失效分析。

参考文献:

IEC 61025:1990 故障树分析(FTA).

系统可靠性工程方法:技术水平的讨论. J. B. Fussel and J. S. Arend, Nuclear Safety 20(5), 1979.

故障树手册. W. E. Vesely et al, NUREG-0492. 美国核管理委员会,核反应堆管理局,系统安全处,华盛顿,DC 20555, 1981.

可靠性技术. A. E. Greene and A. J. Bourne, Wiley-Interscience, 1972.

B.6.7 最坏情况分析

注:在 GB/T 20438.2—2006 的表 B.5 和表 B.6 中引用了本技术/措施。

目的:为了避免由环境条件和部件公差不相宜的组合引起的系统失效。

描述:根据一种理论检验和计算系统的工作能力和确定的部件尺寸。把环境条件改变到它们允许的最高边缘值。检查系统最实质性的响应并把这些响应同规范进行比较。

B.6.8 扩展的功能测试

注:在 GB/T 20438.2—2006 的表 B.5 和表 B.6 中引用了本技术/措施。

目的:为了揭示规范、设计和开发阶段的失效。检验安全相关系统在稀有的,或者非规定的输入事件中的行为。

描述:扩展的功能测试将评审安全相关系统响应输入条件的功能行为,这些输入条件预计是很少见的(例如主要失效)或者它们是超出安全相关系统规范之外的(例如操作错误)。对稀有条件的情况,观察到的安全相关系统的行为将同规范进行比较。在未规定安全相关系统的响应的情况下,应检查由观察到的响应维持的设备安全性。

参考文献:

功能程序测试和分析. W. E. Howden,McGraw-Hill,1987.

软件测试工艺. G. J. Myers,Wiley &Sons,New York,1979.

关键计算机系统的可靠性 3. P. G. Bishop et al,Elsevier Applied Science,1990,ISBN 1-85166-544-7.

B.6.9 最坏情况测试

注:在 GB/T 20438.2—2006 的表 B.5 和表 B.6 中引用了本技术/措施。

目的:为了测试在最坏情况分析过程中规定的情况。

描述:在最坏情况下测试系统工作能力和确定的部件尺寸。环境条件改变到它们的最高容限值,检查系统最重要的响应并把它们同规范进行比较。

B.6.10 故障插入测试

注:在 GB/T 20438.2—2006 的表 B.5 和表 B.6 中引用了本技术/措施。

目的:为了在系统硬件中引入或者模拟故障并把响应编入文档。

描述:这是评估可靠性的一种定性方法。为了描述故障位置和类型以及怎样引入它,最好使用详细的功能块、电路和接线图。例如:可切断各模块的电源;可开路或短路电源线、总线、地址线;可开路或短路各部件或者它们的端口;可使继电器闭合或断开失效、或者在错误的时间开合等。例如可按 IEC 60812 的表Ⅰ和表Ⅱ中所示对产生的系统失效进行分类。原则上讲,只引入单稳态故障。然而,在内部诊断测试并未揭示出一个故障或者一个故障并未变得一目了然的情况下,它可能留在系统中并且要考虑到第二个故障的影响。故障数轻易就可增加到几百个。

应参考系统的多学科工作组和卖方目前所作的工作。应计算或估算对有严重后果的故障两次失效之间的平均工作时间。如果算出的时间很短,则应进行修改。

参考文献:

过程控制系统的完整性测试. RJ. Lasher,Control Engineering 36(11),152-164,October 1989.

IEC 61069-5:1994 工业过程测量和控制 用于系统评价的系统性能评估 第 5 部分:系统可靠性评估.

IEC 60812:1985 系统可靠性分析技术 失效模式和影响分析(FMEA)规程.

附 录 C
（资料性附录）
达到软件安全完整性的技术和措施的评述
（参看 GB/T 20438.3）

C.1 一般要求

本附录中包含的技术的评述既不能被认为是完整的也不能认为是详尽的。

一些通用的参考文献有：

系统——美国安全协会系统安全分析手册. System Safety Society, New Mexico Chapter. Po Box 95424, Albuguerque NM, USA.

关键计算机系统的可靠性 3. P. G. Bishop et al, Elsevier Applied Science, 1990, ISBN 1-85166-544-7.

软件工程百科全书. Ed. J. Marciniak. John Wiley & Sons, 1994, ISBN 0-471-54004-8.

软件工程师参考书. Ed. J. McDdermid. Buttlrworth-Heinemann, 1991, ISBN 0-7506-1040-9.

C.2 要求和详细的设计

注：在 B.2 中可看到有关的技术和措施。

C.2.1 结构化方法

注：在 GB/T 20438.3—2006 的表 A.2 和表 A.4 中引用了本技术/措施。

C.2.1.1 一般要求

目的：结构化方法的主要目的是通过把注意力集中到生命周期的初期阶段来提高软件开发的质量。本方法旨在通过精确的和直观的规程和表示法（计算机辅助的）来达到该目的，从而可用一条逻辑命令和一种结构化形式确定要求和实现特征并把它们编成文档。

描述：存在许多结构化方法。一些方法打算用于传统的数据处理和事务处理功能，而另一些方法（MASCOT、JSD、实时 Yourdon）更适于过程控制和实时应用（有助于更安全的临界值）。

结构化方法实质上是系统地发觉和划分一个问题或一些“思维工具”。它们的主要特征如下：

a) 把一个庞大的问题思考成、分解成可管理的一些阶段的一条逻辑命令；

b) 总系统（包括环境及要求的系统）的分析和文档编制；

c) 要求系统中的数据和功能的分解；

d) 检验表，即需要分析的项目的分类表；

e) 低智能内务操作——简单、直观、实用。

问题和系统实体（例如过程和数据流）分析和编制文件的支持符号有助于准确，而表示这些实体执行的处理功能的表示法则有助于更非形式化。但某些方法部分地使用了（数学上的）形式表示法（例如，JSD 使用了正规的表达式；Yourcon、SOM 和 SDL 使用有限状态机）。精确性的提高不仅缩小了错误理解的范围，还提供了自动处理的领域。

结构化表示法的另一好处是它们的直观性，它使用户能够根据其丰富但未明说的知识直观地检验一个规范或者设计。

本评述较详细地描述了 5 种结构化方法：受控的要求表达式、杰克逊系统开发（JSD）、MASCOT、实时 Yourdon 及结构化分析和设计技术（SADT）。

参考文献：

实时系统的软件设计. J. E. Cooling, Chapman and Hall, 1991.

实时系统的结构化开发(共3卷). P. T. Ward and S. J. Mellor, Yourdon Press, 1985.

主要的系统分析. St. M. McMenamin, F. Palmer, Yourdon Inc, New York, 1984.

基于软件的大型航空系统开发中结构化方法的使用. D. J. Hatley. Proc. DASC, Baltimore, 1984.

C.2.1.2 CORE——受控的要求表达式(Controlled Requirements Expression)

目的:为了保证能确定和表示所有的要求。

描述:本方法想用来在买方/最终用户和分析员之间连接一座桥梁。它在数学上并不严密但有助于通信——CORE是为要求表达式而不是为规范而设计的。本方法被构建并且表达式经历各个求精级。CORE法鼓励对问题的各种看法,引入系统使用的环境知识以及各种类型用户的不同观点。CORE包括识别背离"总设计"的导则和战术。可以校正或者明显地标识这些背离并把它们编入文档。因而规范可能不完全,但可检测未辨别出的问题和高风险区域,它们是在其后的设计中必须要考虑的。

参考文献:

实时系统的软件设计. J. E. Cooling, Chapman and Hall, 1991.

C.2.1.3 JSD——杰克逊系统开发(Jackson System Development)

目的:包括从要求一直到代码的软件系统(特别着重于实时系统)的一种开发方法。

描述:JSD是一种分阶段的开发过程,在这个过程中开发者将制定真实世界行为的模型,系统功能则以这些行为为基础,并且确定所需的功能并把它们插入模型,以及把产生的规范变换成目标环境中可实现的规范。因此它包括规范和设计及开发的传统阶段,而采用了有些与传统方法不同的但不是完全不同的观点。

此外,它特别着重于发现真实世界中实体的初始阶段,该真实世界与正在建立的系统有关并关系到建造它们的模型以及它们可能发生什么情况。一旦已完成"真实世界"的这种分析和建立起一个模型,就能分析系统所需功能,从而确定怎样把它们归纳入这个真实世界模型中。产生的系统模型随模型中所有过程的结构化描述而增大,并且整个被变换成将在目标软件和硬件环境中运行的一些程序。

参考文献:

JSD的一个评述. J. R. Cameron. IEEE Transactions on Software Engineering, SE-12, No. 2, February 1986.

系统开发. M. Jackson, Prentice-Hall, 1983.

C.2.1.4 MASCOT——解决软件构建、运行和测试的模块化方法(Modular Approach to software construction, Opration and Test)

目的:实时系统的设计和实现。

描述:MASCOT是一种由一个编程系统支持的设计方法。它是表示实时系统结构的一种系统方法,其表示方法与目标硬件或实现语言无关。它给设计强加一种规定的方法,该方法可产生一种高度模块化的结构,从而保证出现在系统集成中的构造元素和设计中的功能元素之间的严密一致性。借助并行过程的一个网络设计一个系统,该网络可通过一些通道进行通信。这些通道既可以是固定数据池,也可是队列(数据流水线)。利用访向机构描述访向通道的控制时与过程无关,这些访问机构给过程强加了一些调度规则。记住MASCOT的最新版本已设计有ADA实现。

MASCOT支持一种基于测试和确认单一软件模块和在功能上与软件模块有关的较大集合的接受策略。打算把一个MASCOT实现建立在一个MASCOT内核之上,该内核是支持实现和访问机制的一组调度图元。

参考文献:

MASCOT 3用户指南. MASCOT USERS Forum. RSRE, Malvern, England, 1987.

C.2.1.5 实时 Yourdon(Real-time yourdon)

目的:实时系统的规范和设计。

描述:作为本技术基础的开发方案假定正在开发的一个系统的三阶段进化。第一阶段包含建立一

个“基本模型”,此模型描述了系统所需的行为。第二阶段包含建立一个描述结构和机构(即当实现时,具体化要求的行为)的实现模型。第三阶段大体相当于传统的规范和设计及实现阶段,不过更为着重于每一阶段开发者从事于建立一个模型的活动。

基本模型有两部分:

——环境模型,方案包含描述系统和它的环境之间的边界以及描述系统必须作出响应的外部事件;

——行为模型,方案包含描述系统响应事件进行的变换以及描述系统为了响应必须保持的数据。

实现模型还分成一些子模型,它们包含分配给处理机的各个过程以及这些过程分解成的软件模块。

为了捕获这些模型,本技术结合了许多其他众所周知的技术:数据流图、变换图形、结构化英语、状态转换图和 Petri 网。另外,本方法包含根据已画出的模型,在纸上或者用机器模拟一个建议的系统设计的技术。

参考文献:

实时系统的结构化开发(共 3 卷). P. T. Ward and S. J. Mellor, Yourdon Press, 1985.

实时系统规范的策略. D. Hatley, E. Pirbhai, Dorest Publishing House, 1988.

C.2.1.6 SADT——结构化分析和设计技术(Structured Analysis and Design Technique)

目的:为了用一种图形形式,使用信息流,对与一个复杂系统相关的过程和管理任务进行判定从而建立模型和进行分析。

描述:在 SADT 中,活动因子图的概念起一种核心作用。一幅 A/F(活动因子)图由分类成所谓“动作框”的一些活动构成。每个动作框有一个唯一的名称,并通过因子关系(画成箭头)与其他动作框相连接,每个因子关系还有一个唯一的名字。可把每个动作框按层次分解成一些子动作框和子关系。有 4 种类型的因子:输入、控制、机构和输出。

——输入:用右左手边进入一个动作框的一个箭头表示。输入可表示材料或者非物质的东西,它们适于用一个动作框中的一个或几个活动来操纵。

——控制:典型地是一条指令,一个操作规程,一个选择判据等。一个控制可指导一个活动的执行,用进入一个动作框顶边的一个箭头来表示它。

——机构:比如人员、组织单元或设备这些资源,它是为了执行它的任务的一个活动所需要的。

——输出:一个活动产生的任何东西,它用在右手边离开一个动作框的箭头表示。

当这些活动相互间由许多因子关系强相关时,也许最好把这些活动考虑成包含在一个动作框内的一个不可分的组,它的内容也不能再详细了。把活动分组成动作框的指导性原则是只由很少的几个因子把产生的框成双地偶联。

A/F 图的模型分层将继续进行下去直到动作框的进一步细分变得毫无意义为止。当框内的活动不可再分或者当动作框的进一步细分超出系统分析范围之外时就达到了这一步。

参考文献:

要求定义的结构化分析. D. T. Ross, K. E. Schoman Jr, IEEE transactions on Software Engineering, Vol. SE-3, 1, 6-15, 1977.

结构化分析(SA):通信理念用的一种语言. D. T. Ross, IEEE Transactions on Software Engineering, Vol. SE-3, 1, 16-34, 1977.

SADT 的应用和扩展. D. T. Ross, Computer, 24-34, April 1985.

结构化分析的设计技术在安全相关系统上的应用. W. Heins, Risk Assessment and Control Courseware, Module B1, Chapter 11, Delft University of Tecnology, Safety Science Group, Po Box 5050, 2600 GB Delft, Netherlands, 1989.

C.2.2 数据流图

注:在 GB/T 20438.3—2006 的表 B.5 和表 B.7 中引用了本技术/措施。

目的:为了用一个图形形式描述流过一个程序的数据流。

描述:数据流图记录了数据输入是怎样变换成输出的,图中的每一步表示一个不同的变换。

数据流图由三种组元构成:

a) 带注释的箭头——代表进出变换中心的数据流,注释记录了是什么数据;

b) 带注释的泡——代表变换中心,注释记录了变换;

c) 算符(and(与),xor(异或))——这些算符被用来连接带注释的箭头。

数据流图中的每个泡可看作是一个可独立应用的黑盒,只要它的输入可用,它就可把这些输入变换成它的输出。主要的优点之一就是不用对怎样实现这些变换作任何假定它们就能显示变换。一幅纯数据流图并不包括控制信息或者排序信息,但正如实时 Yourdon 中讲述的那样(参看 C.2.1.5),实时扩充成表示法为此提供了必要的条件。

考查系统输入并朝系统输出方向作业是制作数据流图的最佳逼近解决办法。每个泡必须代表一个不同的变换——它的输出应与它的输入在某些方面有所不同。不存在确定图的总体结构的规则,构建一幅数据流图是系统设计方面的一个创造。同所有设计一样,它是伴随分步求精的初期尝试从而产生最后图形的一个迭代过程。

参考文献:

软件工程. I. Sommerville,Addison-Wesley,3rd Edition,1989.

ISO 5807:1985 信息处理 数据、程序和系统流程图、程序网络图和系统资源图的文件符号和规定符号.

ISO/IEC 8631:1989 信息技术 程序结构和设计它们时的约定符号.

C.2.3 结构图

注:在 GB/T 20438.3—2006 的表 B.5 中引用了本技术/措施。

目的:为了用图形显示一个程序的结构。

描述:结构图是一种符号表示法,此法可实现数据流图。结构图描述了编程系统和各部分的一个分层结构,并用树形图显示这种结构。它们记录了怎样按程序单元的一种层次结构来实现数据流图的元素。

一幅结构图显示了程序模块之间的关系而不包含这些单元启动命令的任何信息,画这些图时使用了以下 4 种符号:

——注释有模块名的一个矩形;

——连接这些矩形从而建立结构的一条线;

——带环的箭头(空心的),注释有通入和来自结构图中各元素的数据名(通常平行于连接图中各矩形的线画带环的箭头);

——带环的箭头(实心环),注释有从结构图中的一个模块到另一模块的控制信号名,也是平行于连接图中两个模块的线画箭头。

从任一非平凡数据流图有可能派生出许多不同的结构图。

数据流图描绘了系统中的信息和功能之间的关系,结构图描绘了实现系统元素的途径。从系统的观点来看,虽然并不相同,但两种技术都是有效的。

参考文献:

软件工程. I. Sommerville,Addison-Wesley,3rd Edition,1989.

结构化设计. L. L. Constantine and E. Yourdon, Englewood Cliffs, New Jersey, Prentice Hall,1979.

通过组合设计的可靠软件. G. J. Myers,New York,Van Nostrand,1975.

C.2.4 形式方法

注:在 GB/T 20438.3—2006 的表 A.1、表 A.2、表 A.4 和表 B.5 中引用了本技术/措施。

C.2.4.1 一般要求

目的:采用基于数学的一种方法来开发软件。它包括形式设计和形式编码技术。

描述:形式方法提供了一种在系统规范、设计或实现的某个阶段开发一个系统描述的方法。产生的描述的形式是一种严密的符号表示法,它可通过数学分析检测各种类型的不一致性或者不正确性,此外,在某些情况下可用机器分析该描述,其精确性类似于使用一台编译器对一个原程序进行语法检查的精度,或者可把该描述制作成动画从而显示所描述的系统各方面的行为,动画可给出对系统满足真实要求及形式上规定的要求的额外置信度,这是因为它提高了人们对规定行为的识别能力。

一种形式方法通常将提供一种表示法(通常使用离散数学的某种形式)和一种用来产生该表示法的一种描述的技术,以及用来检查各种正确性属性描述的各种分析形式。

注:在 B.2.2 中也可看到上述描述。

在本评述的以下小节中描述了几种形式方法——CCS、CSP、HOL、LOTOS、OBJ、时序逻辑、VDM 和 Z。注意,其他的技术,比如有限状态机(参看 B.2.3.2)和 Petri 网(参看 B.2.3.3),也可根据使用该技术时,这些技术符合某个精确的数学基础的严密程度,把它们看成是形式方法。

参考文献:

安全关键系统中形式方法的实际作法. S. Liu, V. Stavridon, B. Dutertre, J. Systems Software 28, 77-87, Eisevier, 1995.

形式方法:在开发安全关键系统中的应用及与开发安全关键系统的关系, L. M. Barroca, J. A. McDermid, The Computer Journal 35(6), 579-599, 1992.

怎样生产正确的软件——介绍通过变换开发形式规范和程序. E. A, Boiten et al, The Computer-Journal 35(6)547-554, 1992.

C.2.4.2 CCS——通信系统的计算(Calculus of Communicating Systems)

目的:CCS 是描述和推断并行通信过程系统行为的一种方法。

描述:CCS 是有关系统行为的一种数学计算。按顺序运行或者并行运行的独立过程的一个网络那样建造系统设计模型。这些过程可通过端口(类似于 CSP 的通道)进行通信,只有在两个过程准备就绪时才能进行通信,不确定性也能建立模型,从整个系统的一个高层抽象描述(称为一条轨迹)开始,可能进行系统的一次逐步求精,使之成为通信过程的一个组合体,它的总行为就是整个系统所需的行为。同样,有可能按自下而上的款式作业、组合过程及使用与合成规则有关的推理规则来演绎最后得到的系统的属性。

参考文献:

通信和并行性. R. Milner, Prentice-Hall, 1989.

复杂系统的规范. B. Cohen, W. T. Harwood and M. I. Jackson, Addison Wesley, 1986.

C.2.4.3 CSP——通信顺序过程(Communicating Sequential Processes)

目的:CSP 是用于并行软件系统(即:并行运行的通信过程的系统)的规范的一种技术。

描述:CSP 为说明的过程系统规范提供了一种语言并为验证过程的实现满足它们的规范(描述成一条轨迹——一个允许的事件顺序列)提供了证明。

按独立过程的一个网络建造一个系统的模型,这些过程既可顺序组合也可并行。用每个过程可能的所有行为来描该过程,这些过程可通过通道进行通信(同步或交换数据),只有在两个过程准备就绪时才能进行通信。能建立事件相关时序的模型。

落后于 CSP 的理论早已直接纳入 INMOS 传送机的体系结构中并且 OCCAM 语言允许在一个传送机网络上直接实现一个 CSP 规定的系统。

参考文献:

通信顺序过程. C. A. R. Hoare, Prentice-Hall, 1985.

C.2.4.4 HOL——高阶逻辑(Higher Order Logic)

目的:它是一种形式语言,这种语言打算作为硬件规范和验证的一种依据。

描述:HOL 指的是一种特殊的逻辑符号表示法和它的机器支持系统,这两者是由英国剑桥大学计

算机实验室开发的,逻辑符号表示法基本上取自丘奇(church)的简单类型理论,机器支持系统则以 LCF(可计算的功能逻辑(logic of computable functions))系统为基础。

参考文献:

HoL:一种面向机器的高阶逻辑的公式表示. M. Gordon, University of Cambridge Technical Report, No. 68, 1985.

使用高阶逻辑的规范和验证:一个实例研究. F. K. Hanna and N. Daeche, in: Formal Aspects of VLSI Design: Proceedings of the 1985 Edinburgh workshop on VLSI, pp, 179-213, G. Milne and P. A. Subrahmanyam(Eds.), North Holland, 1986.

形式方法在 VIPER 微处理机上的应用. W. J. Cullyer, C. H. Pygott, proc. IEEE 134, 133-141, 1987.

C.2.4.5 LOTOS

目的:LOTOS 是用于描述和推理并行通信过程系统行为的一种方法。

描述:LOTOS(时间排序规范语言(language for temporal ordering specification)),是以带有来自相关代数学 CSP 和 CIRCAL(电路计算)附加特征的 CCS 为基础的。它通过把它同一个基于抽象数据型语言 ACT ONE 的第二成分组合起来从而克服了 CCS 在处理数据结构和值表示式方面的弱点。然而在描述抽象数据类型时可使用 LOTOS 的过程描述成分并结合其他形式方法。

参考文献:

ISO 8807:1989 信息处理系统 开式系统互连 LOTOS 基于观察的行为时间排序的一种形式描述技术.

C.2.4.6 OBJ

目的:为了在实现之前利用用户反馈信息和系统确认信息提供一个精确的系统规范。

描述:OBJ 是一种代数规范语言。借助代数方程,用户规定要求。通过影响抽象数据类型(ADT)的运算,规定系统行为、结构特征,在操作员工作情况可见而实现细节被"隐藏"的情况下,一个 ADT 就像一个 ADA 包一样。

一种 OBJ 规范和其后的逐步实现适合于同其他形式逼近法一样的形式验证技术。此外,因为 OBJ 规范的结构特征可用机器执行,因此从规范本身可直接实现系统确认。执行过程实质上是通过持续的方程置换(重写)一直到达到规定的输出值为止来评价一个功能。这种可执行性允许设想的系统的最终用户在系统规范阶段赢得对或然系统的一次"考查",而不需精通基础的形式规范技术。

正如其他所有的 ADT 技术的情况一样,OBJ 只适用于顺序系统或者顺序特征的并行系统。OBJ 已被用于小型和大型工业应用的规范。

参考文献:

OBJ 的一个介绍:编写和测试规范的一种语言. J. A. Goguen and J. Tardo, Sepecification of Reliable Software, IEEE Press 1979, reprinted in Software Specification Techniques, N. Gehani, A. McGrettrick(eds), Addison-wesley, 1985.

实际软件生产的代数规范. C. Rattray, Cogan press, 1987.

图形软件的标准化和认证的一种代数逼近法. R. Gnatz, Computer Graphics Forum 2(2/3), 1983.

C.2.4.7 时序逻辑

目的:安全性和运行要求及形式证明的直接表达式,在其后的开发步骤中将保留这些属性。

描述:标准的一阶谓词逻辑不包含时间概念,通过增加情态算符(例如"今后"(henceforth)和"或许"(eventally)),时序逻辑可扩展一阶逻辑。这些算符可用来限定有关系统的一些断言。例如,安全属性可能需要包含"今后",而要求的系统的其他状态可能需要借助某些另外的起始状态达到"或许"。将根据状态(行为)序列来解释瞬时公式。换句话说,根据选择的描述级来构成一个"状态"。它可以指整个系统,一个系统部件或者计算机程序。

在时序逻辑中并不直接处理量化的时间间隔和约束。必须通过建立附加的时间状态把绝对定时处理成状态描述的一部分。

参考文献：

程序的时序逻辑. F. Kroger. EATCS Monographs on Computer Science, Vol. 8, springer Verlag,1987.

使用时序逻辑的安全性设计. J. Gorski. SAFECOMP 86, Sarlat, France, Pergamon press, October 1986.

程序的时序逻辑. A. Pnueli,18th Annual Symposium on Foundations of Computer Science,IEEE, 1977.

使用时序逻辑验证并行过程. Hailpern,T. Brent,Springer Verlag,1981.

C.2.4.8 VDM,VDM++——维也纳开发方法(Vienna Development Method)

目的：顺序(VDM)和并行实时(VDM++)程序的系统性规范和实现。

描述：VDM 是基于数学的一种规范技术和在某种程度上允许对照规范验证其正确性的一种求精实现技术。

规范技术是以模型为基础的，在这个模型中将根据集合理论结构(根据此集合理论结构来描述不变量(谓词))和运算(对某个状态进行运算的那个状态的模型是借助系统状态并通过规定运算的先决条件和后续条件来建造的)建立系统状态模型。为了保留系统不变量，可验证这些运算。

借助目标语言的数据结构，通过具体化系统状态和借助目标语言的程序，通过求精运算可完成规范的实现。具体化和求精步骤将引出验证它们的正确性的责任。设计者应决定是否要履行这些职责。

原则上讲，VDM 是用在规范阶段，但也可在产生源码的设计和实现阶段中使用。它仅适用于并行系统中顺序结构程序或者顺序过程。

面向对象的和并行实时扩充的 VDM、VDM++是基于 ISO 语言 VDM-SL 和面向对象的语言 Smalltalk 的一种形式规范语言。

VDM++提供了一个很宽的结构范围，因而一个用户可在形式上用一种面向对象的格式来规定实时系统。在 VDM++中，一个完整的形式规范由分类规范的一个汇集和任选的一个工作区组成。

VDM++的实时规定是：

——为了表示一个方法主体中的当前时刻和方法引用时刻，应提供时序表达式。

——给方法增加一个定时的后续表达式以便为正确实现规定执行时间的上(或下)限。

——已经引入时间连续变量。利用假设子句和效果子句，可以规定这些时间函数之间的关系(例如微分方程)。这个特征已证明对工作在一个时间连续环境中的系统要求的规范是很有用的。求精步骤可产生这些类型的系统的离散软件解。

参考文献：

ISO/IEC 13817-1:1997 信息技术 程序设计语言及其环境和系统软件界面 维也纳(Vienna)开发方法 规范语言 第一部分：基本语言.

VDM——SL 的符合性规定. G. I. Parkin and B. A. Wichman, Lecture Notes in Computer Science 670, FME' 93 Industrial——Strength Formal Methods, First International Symposium Of Formal Methods in Europe. Editors:J. C. P. Woodcock and P. G. Larsen. Springer Verlag,501-520.

VDM+——语言的主要特点：http:/www. ifad dk/ products/vdmlangchar. html.

使用 VDM 开发系统软件. C. B. Jones,Prentice-Hall. 2nd Edition,1990.

软件开发——一种精密的逼近法. C. B. Jones,Prentice-Hall,1980.

形式规范和软件开发. D. Bjorner and C. B. Jones,Prentice-Hall 1982.

复杂系统的规范. B. Cohen,W. T. Harwood and M. I Jackson,Addison Wesley,1986.

C.2.4.9 Z

目的：Z是用于顺序系统的一种规范语言表示方法和一种设计技术，在某种程度上它允许开发者从一个Z规范开始进行可执行的算法，此算法允许对照规范验证它们的正确性。

原则上Z可用于规范阶段，但已发明出一种也可用于设计和实现的方法。它最适合于开发面向数据的顺序系统。

描述：像VDM一样，规范技术也是以模型为基础的，在这种模型中将根据集合论结构（借助该结构可描述不变量（谓词））及运算（对某个状态进行运算的该状态的模型是通过规定运算的先决条件和后续条件来建造的）来建立系统状态模型。为了保留系统不变量从而演示它们的一致性可验证这些运算，将把一个规范的形式部分分成一些模式，这些模式允许通过求精构建规范。

典型地，一个Z规范是形式Z和用自然语言的非形式说明文本的混合物，（形式文本本身对易读而言是太简洁了，并且常常需要解释它的目的，而非形式自然语言容易变得模糊和不准确。）

与VDM不同，Z是一种表达式而不是一个完整的方法，然而已开发出一种辅助方法（称为B），它可同Z一起使用。B方法依据的是逐步求精的原理。

参考文献：

Z表达式——一本参考手册. J. M. Spivey. Prentice-Hall, 1992.

规范实例研究. Edited by I. Hayes, Prentice-Hall, 1987.

B方法. J. R. Abrial et al, VDM'91 Fomal Software Development Methods, (S. Prehen and W. J. toelenel, eds), Springer Verlag, 398-405, 1991.

UNIX文件存储的规范. C. Morgan and B. Sufrin. IEEE Transactions on software Engineering, SE-10, 2, March 1984.

· C.2.5 防御性编程

注：在GB/T 20438.3—2006的表A.4中引用了本技术/措施。

目的：为了产生一些程序，在执行它们的过程中，这些程序可检测异常控制流、数据或数据值，并且以一种预定的和可接受的方式对这些异常作出反应。

描述：为了检验控制或者数据异常，在程序设计过程中可以使用许多种技术。为了减少错误的数据处理的似真性，在一个系统的整个程序设计过程中，系统地使用了这些技术。

有两个防御技术的重叠区域。设计的固有出错——安全软件可适应它本身设计上的不足，这些缺点可能是设计或编码中的错误或者不正确的要求造成的，下面列出了一些防御技术：

——检查变量的范围；

——在可能的情况下，检查值的似真性；

——在规程入口处检查输入规程的参数的类型、大小和范围。

从程序功能和变量的物理意义这两方面来看，上述三条建议有助于保证由程序管理的数目是合理的。

只读和读写参数应分开并应检查它们的存取。这些功能应把所有参数处理成只读参数。字面常量不可写存取。这有助于检测意外的重写或者误用变量。

打算使容错软件用于“预计”在它特定的环境中的失效或者使用在超出额定的或预定的条件之外以及像预定的那样工作。这些技术包含以下几种：

a) 检查具有物理意义的输入变量和中间变量的似真性。

b) 检查输出变量的效果，最好直接观察相关的系统状态的改变。

c) 检查软件的配置，包括预期硬件的存在和可存取性，还要检查软件本身是完整的——这一点对于在维护过程之后保持完整性是特别重要的。

有些防御程序设计技术，比如控制流序列检查也能对付外部失效。

参考文献：

关键计算机系统的可靠性 1. F. J. Redmill, Elsevier Applied Science, 1988. ISBN 1-85166-203-0.

关键计算机系统的可靠性 2. F. J. Redmill, Elsevier Applied Science, 1989, ISBN 1-85166-381-9.

实时程序设计概念的软件工程状况. E. Scholtsch, Computer physics Communications 41, North Holland, Amsterdam, 1986.

C.2.6 设计和编码标准

注：在 GB/T 20438.3—2006 的表 A.4 中引用了本技术/措施。

C.2.6.1 一般要求

目的：为了提高可验证性，为了促进以群体为中心的目标逼近法以及为了实施一种标准的设计方法。

描述：项目一开始，参加者就协商将遵守的规则。这些规则包含要遵守的设计和开发方法（例如：JSP、MASCOT、Petri 网等）以及有关的编码标准（参看 C.2.6.2）。

制定这些规则是为了使开发、验证、评估和维护更容易。因此，他们应把适用的工具，特别是分析程序和逆向工程工具考虑进去。

参考文献：

IEC 60880:1986 核电站安全相关系统计算机软件.

关键计算机系统的可靠性 1. F. J. Redmill, Elsevier Applied Science, 1988, ISBN 1-85166-203-0.

Verein Dentscher lngenieure. Software-Zuverlassigkeit-Grundlagen, konstruktive Massnahmen, Nachweisverfahren. VDI-Verlag, 1993, ISBN 3-18-401185-2.

C.2.6.2 编码标准

注：在 GB/T 20438.3—2006 的表 B.1 中引用了本技术/措施。

目的：为了提高生成的代码的可验证性。

描述：在编码之前应充分协商要遵守的详细规则，典型地这些规则包括：

——模块化细节，例如界面形状、软件模块大小；

——在面向对象的语言情况下，使用的封装、继承性（深度受约束）和多形性；

——有限制地使用或者回避某些语言结构，例如“go to”（转到），“equivalence”（等价），dynamic objects（动态对象），dynamic data（动态数据），dynamic data structures（动态数据结构），recursion（递归），Pointers（指示字），exits（出口）等；

——在执行安全关键代码期间启用中断的约束；

——代码的布局（列表）；

——在高级语言程序中没有非条件转移（例如“go to（转到）”）。

制定这些规则是为了使软件模块测试、验证、评估和维护更容易。因此，他们应把可用的工具，特别是分析程序考虑进去。

注：有关本题目的其他信息可参看 C.2.6.3～C.2.6.7。

C.2.6.3 无动态变量或者动态对象

注：在 GB/T 20438.3—2006 的表 B.1 中引用了本技术/措施。

目的：为了排除

——不需要的或未发现的存储器重复占位；

——在（安全）运行时间内资源的分配瓶颈。

描述：在本测量的情况下，是这样的一些动态变量和动态对象：在运行时它们有分配给它们的存储器和确定的绝对地址，分配的存储器的值和它的地址与分配瞬时系统的状态有关，这意味着不能用编译器或者任何别的脱机工具来检查它。

因为动态变量和对象的数目以及分配给新动态变量或对象的现存空闲内存空间与分配瞬时的系统状态有关，当分配或使用变量或对象时有可能发生故障。例如，当系统分配存储单元时，闲置内存量不

足，另一变量的存储内容就可能无意中被重写。当不使用动态变量或对象时就可避免这些故障。

C.2.6.4 在建立动态变量或动态对象过程中的在线检验

注1：在 GB/T 20438.3—2006 的表 B.1 中引用了本技术/措施。

目的：为了检查在发生分配前要分配给动态变量和对象的存储器是空闲的，从而保证在运行时动态变量和对象的内存分配不会影响现存的变量、数据或代码。

描述：在本测量中，动态变量是这样的变量，它们在运行时有分配给它们的存储器和确定的绝对地址(从这个意义上讲变量也是对象实例的属性)。

在给动态变量或对象分配内存之前，借助硬件或软件来检查存储器以便保证它是空闲的(例如用以避免堆栈溢出)。如不允许分配(例如当存储器不足以应付所确定的地址时)必须采取适当的行动。在已经使用一个动态变量或者目标之后(例如退出一个子例程之后)，曾经分配给它的整个存储器必须要腾空。

注2：一种替代办法是统计证明对一切情况来说，存储器都是合适的。

C.2.6.5 有限地使用中断

注：在 GB/T 20438.3—2006 的表 B.1 中引用了本技术/措施。

目的：为了保持软件可验证和可测试。

描述：应限制使用中断。当它们能简化系统时才使用中断。在被执行的功能的临界部分(例如时间上的转折点、数据改变的转折点)禁用中断处理软件。当使用中断时，不可中断的部分应有一个规定的最大计算时间，使之能计算禁止一次中断的最大时间。中断使用和屏蔽应全部编成文档。

C.2.6.6 有限地使用指示字

注：在 GB/T 20438.3—2006 的表 B.1 中引用了本技术/措施。

目的：为了避免在没有首先检查指示字的范围和类型的情况下由存取数据引起的问题。为了支持模块测试和软件验证。为了限制失效的后果。

描述：在应用软件中，只有在存取之前检验指针数据类型和值的范围(为了保证参考指针是在正确的地址空间内)时，才在源码级使用指针算术。在任务之间直接引用不能实现应用软件的任务间通信，应通过操作系统来进行数据交换。

C.2.6.7 有限地使用递归

注：在 GB/T 20438.3—2006 的表 B.1 中引用了本技术/措施。

目的：为了避免不可验证的不可测试的子例程调用。

描述：当使用递归时，必须要有一个明显的判据，此判据可预测递归深度。

C.2.7 结构化编程

注：GB/T 20438.3—2006 的表 B.1 中引用了本技术/措施。

目的：为了设计和实现程序，从某种意义上讲，不用执行它就能进行分析是切实可行的，该程序只包含一个统计学上不可测试的行为的绝对最小量。

描述：下面的原则可用来降低结构的复杂性：

——把程序分成适当小的软件模块，从而保证尽可能地使它们去耦并且所有的相互作用都是显式的。

——使用结构化结构来合成软件模块控制流，这个控制流是一些序列、迭代和选择。

——保持到一个软件模块的可能通路要少，输入和输出参数之间的关系要尽量简单。

——避免复杂的分支，特别要避免高级语言中的无条件转移(goto(转到))。

——在可能的情况下，使回路约束和分支同输入参数联系起来。

——作为决定分支和回路依据的计算要简单。

除效率取得绝对优先权的情况(例如某些安全关键系统)之外，应使用有助于上述方法的程序设计语言的特征而不用(据说)是更有效的其他特征。

参考文献：

结构化程序设计的说明. E. W. Dijkstra, Structured Programming, Academic Press, London, 1972, ISBN 0-12-200550-3.

程序设计的一个规范. E. W. Dijkstra. Englewood Cliffs NJ, Prentice-Hall, 1976.

自上而下编程的一种软件工具. D. C. Ince. Software-Practice and Experience, vol. 13, No. 8, August 1983.

验证——实际问题. J. T. Webb Webb and D. J. Mannering, SARSS 87, Nov. 1987, Altrincham, England, Elsevier Applied Science, 1987, ISBN 1-85166-167-0.

反应堆保护系统的软件设计和确认的一次经历. S. Bologna, E. de Agostino et al, IFAC Workshop, SAFECOMP, 1979, Stuttgart, 16-18 May 1979, Pergamon Press, 1979.

C.2.8 隐藏/封闭信息

注：在 GB/T 20438.3—2006 的表 B.9 中引用了本技术/措施。

目的：为了防止无意识的存取数据或规程并因此为了支持一个好的程序结构。

描述：全局可访问的所有软件成分的数据可能被这些成分的任何一个意外地和不正确地修改。这些数据结构的任何改变都需要仔细地检验代码和扩充性修改。

信息隐藏是减少这些困难的一种常用方法。关键数据结构被“隐藏”，只有通过一个定义的存取规程集才能操纵它。这种办法允许修改内部结构或者增加另外的规程而不会影响剩余软件的功能行为。例如，一个姓名目录可以有存取规程“插入”、“删除”和“查找”。可以重写存取规程和内部数据结构（例如使用不同的查找方法或者把名字存放在一张硬盘上）而不会影响使用这些规程的剩余软件的逻辑行为。

在这种联系中，应使用抽象数据的概念，如果不能提供直接支持，则有必要检查还没有在无意中破坏这种抽象。

参考文献：

软件工程：改变计划编制. D. A. Lamb. Prentice-Hall, 1988.

论程序族的设计和开发. D. L. Parnas. IEEE Trans SE-2, March 1976.

C.2.9 模块法

注：在 GB/T 20438.3—2006 的表 A.4 和 B.9 中引用了本技术/措施。

目的：为了限制系统的复杂程度，把一个软件系统分解成一些小的易理解的部分。

描述：一种模块法或者模块化包含一个软件项目的设计、编码和维护阶段的几种规则。根据设计过程中使用的设计方法改变这些规则。大多数方法都包含以下规则：

a) 一个软件模块应有需完成的、定义良好的单项任务或功能；

b) 应限制和严格定义软件模块之间的衔接，在一个软件模块中的相干性应是很强的；

c) 应建立提供有几级软件模块的子程序集合；

d) 应把子程序的大小限定为某个规定值，典型地为 2～4 个屏幕尺寸；

e) 子程序只有一个入口和一个出口；

f) 通过界面，软件模块与其他软件模块通信——在使用全局或者共用变量的情况下，它们应构成良好，存取可控并有理由认为可在每个实例中使用它们；

g) 应把所有的软件模块界面全部编入文档；

h) 任何软件模块的界面只包含它的功能所需的那些参数。

参考文献：

结构化设计——一种计算机程序和系统设计规范的基本原则. E. Yourdon, L. L. Constantine, Prentice-Hall, 1979, ISBN 0-13-854471-9.

C.2.10 使用可信的/经验证的软件模块和成分

注 1：在 GB/T 20438.3—2006 的表 A.4 中引用了本技术/措施。

注 2：关于支持下面的数值估算的某些数学问题可参看附录 D。关于一种类似的方法和一种统计学方法还可参看 B.5.4。

目的：为了避免对每一新应用都需彻底重新确认或者重新设计软件模块和硬件部件。为了利用设计的优点，该设计已被正式或严格验证，此外，它的重要的运行史也是可用的。

描述：本方法可验证软件模块和软件成分完全不会产生系统设计故障以及运行失效。只在极少见的情况下，使用可信的软件模块和成分（即在使用中被证明的那些软件模块和成分）才足以作为保证达到必需的安全完整性的唯一方法。对具有许多可能功能的复杂软件成分（例如一个操作系统）而言，建立在使用中实际上已被充分验证的那些功能是最根本的。例如，在为检测硬件故障而提供有一个自测试程序的情况下，如果在运行期内没有出现硬件失效，我们就不能把检测故障自测试程序看成是经过使用验证过的。

如果一个软件成分或者软件模块已对所要求的完整性水平作过验证或者如果它满足下列判据，就可充分信任它：

——规范未改变；

——各种应用中的系统；

——至少一年的运行史；

——符合安全完整性等级的运行时间或者适当的要求次数；证明非安全失效率小于

——在置信度 95%时，每个要求为 10^{-2}（年），需要的运转期为 300（年）

——在置信度 99.9%时，每个要求为 10^{-5}（年），需要的运转期为 690 000（年）；

——所有的操作经验必须联系一个已知的软件模块功能的要求简要表，以便保证操作经验的增加真正导致与该要求简要表有关的软件模块行为知识的增加；

——无安全失效。

注 3：在某个描述体中一个并非是安全致命的失效在另一描述体中可能是安全致命的，并且反之亦然。

为了启用对一个软件成分或模块满足判据的验证，必须把下列几项编入文档：

——严格标识每个系统和它的组成成分，包括版本号（对软件和硬件两者而言）；

——用户标识和应用时间；

——运行时间；

——用户应用系统的选择规程和应用实例；

——检测和登记失效以及消除故障的规程。

参考文献：

DIN V VDE 0810 A1：Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben（安全相关系统中的计算机原理）。Änderung 1 zu DIN V VDE 0801/01.90. Beuth-Verlag，Berlin，1994.

化学过程的安全自动化指南. CCPS，AIChE New York，1993.

C.3 结构设计

C.3.1 故障检测和诊断

注：在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的：为了检测一个系统中的故障（这些故障可能导致一次失效）从而为减少失效影响的对应措施提供依据。

描述：故障检测是检查一个系统的错误状态的活动（这些错误状态是被检查的子系统内的一个故障引起的）。故障检测的主要目的是阻止错误结果的影响，当一个系统检测它自己的错误结果时，同并行成分共同起作用并放弃控制的这个系统称为自检。

故障检测基于冗余（主要检测硬件故障——参看 GB/T 20438.2—2006 附录 A）和多样性（软件故障）原理。需要用某种表决来确定结果的正确性。可用的特殊方法是：失效断言程序设计，新版本压缩

和安全袋技术，对硬件而言有：引入附加传感器，控制回路，差错检验码等。

可以通过检查各级的值域或时间域、特别是物理的（温度、电压等）、逻辑的（差错检测码）、功能的（断言）或者外部的（似真性检验）来实现故障检测。为了能进行故障跟踪，可把这些检验的结果储存起来并把它们同数据联系起来。

复杂系统是由子系统构成的，故障检测、诊断和故障纠正的效果与各分系统中相互作用的复杂程度有关，这种相互作用的复杂性可影响故障的传播。

应在最小的子系统级使用故障诊断，因为较小的子系统可以更详细地诊断故障（错误状态检测）。

全企业综合信息系统日常能把安全相关系统的状态包括诊断测试信息传递给另一监控系统。当检测到一个异常时，就会加亮它并在它发展成一种危险情况之前就用它来触发纠正行动。最后，当发生一次事故时，这种异常记录文档可帮助其后的调查。

参考文献：

关键计算机系统的可靠性 1. F. J. Redmill ，Elsevier Applied Science，1988，ISBN 1-85166-203-0.

C.3.2 差错校正和纠正码

注：在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的：为了检测和纠正有关信息中的错误。

描述：以对一条 n 位信息而言，生成一个 K 位编码块，此块使之能检测和纠正 r 个错误。两类编码的例子是汉明（Hamming）码和多项式代码。

应注意，在安全相关系统中，通常需要废弃故障数据而不是纠正它，因为只有一部分预定的错误才能得以正确纠正。

参考文献：

纠错码技术. E. R. Berlekamp，Proc. IEEE68(5)，1980.

关于纠错码的简短历程. C. J. A. Sloane，Springer Verlag，wien，1975.

C.3.3 失效断言编程

注：在 GB/T 20438.2—2006 的表 A.8 中和 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的：为了在执行一个程序的过程中检测软件设计的剩余故障，从而防止系统安全致命失效并持续高可靠地运行。

描述：失效断言程序设计方法遵循的理念是检验一个先决条件（在执行一个语句序列之前，对初始条件进行有效性检验）和一个后续条件（在执行一个语句序列之后检查结果）。如不满足先决条件或者后续条件，处理将报告出了错误。

例如：

```
断言<先决条件>；
    动作 1；
    :
    :
    动作 X；
    :
    :
    断言<后续条件>；
```

参考文献：

程序设计规范. E. W. Dijkstra，Prentice-Hall，1976.

程序设计的科学. D. Gries，Springer Verlag，1981.

软件开发——一种精密的方法. G. B. Jones，Prentice-Hall，1980.

C.3.4 安全袋

注：在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:为了防止软件中规范和实现的剩余故障,这些故障对安全有不利的影响。

描述:一个安全袋是对于一个不同规范的外部监视器,该监视器是靠一台独立的计算机实现的。安全袋只涉及保证主计算机执行安全,不一定纠错、动作。安全袋连续地监视主计算机。安全袋防止系统进入一个不安全的状态,此外,当它检测到主计算机进入一种潜在的危险状态时,安全袋或者主计算机必须使系统恢复到一个安全状态。

安全袋的硬件和软件应按适当的 SIL(安全完整性等级)分类和考核。

参考文献:

使用 AI(人工智能)技术提高软件安全性. Proc. IFAC SAFECOMP 88, Sarlat, France, Pergamon Press, October 1986.

C.3.5 软件多样化(多种程序设计)

注:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:为了防止系统的安全致命失效并持续高可靠地运行,在执行一个程序过程中检测和屏蔽剩余软件设计和实现的故障。

描述:在多样化程序设计中,以不同的方式 N 次设计和实现一个给定的程序规范。给 N 个版本都提供同样的输入值,并比较 N 个版本产生的结果,如认为结果有效,将把结果传送到计算机输出。

N 个版本可在单独的计算机上并行运行,换言之,所有版本都能在同一台计算机上运行并且结果须经一次内部表决。对 N 个版本可使用不同的表决策略,这要根据下示应用要求而定。

——如果系统有一个安全状态,则要求完全一致(所有的 N 一致)是可行的,否则将使用一个使系统达到安全状态的输出值。对简单的跳闸系统来说,表决偏向安全。在这种情况下,如果两种版本要求一次跳闸,安全动作就将是跳闸,典型地,这种方法只使用两种版本($N=2$)。

——对没有安全状态的系统而言,可使用多数表决策略。对于不存在集体一致的情况,为了使选择正确值的机会最大化(例如取中间值,在恢复一致性之前,暂时冻结输出等)可使用概率方法。

本技术不能消除软件设计的剩余故障,也不能避免解释规范时的错误,但它提供了在这些故障或错误影响安全性之前检测和屏蔽的一种方法。

参考方献:

可靠的计算:借助设计多样化概念. A. Avizienis and J. C. Laprie, Proc. IEEE 74(5)May1986.

在重合失效的条件下多版本软件分析的一个理论基础. D. E. Eckhardt and L. D. Lee, IEEE trans SE-11(12), 1985.

现在计算机能安全地执行主要的安全功能. Otto Berg Von Linde, Railway Gazette international, Vol. 135, No. 11, 1979.

C.3.6 恢复程序块

注:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:为了增大程序最终执行它预计的功能的似真性。

描述:常常是独立地写几个不同的程序段,打算每个段都执行同一要求的功能。从这些程序段来构造最后的程序,首先执行叫作主程序段的第一段,随后是它计算的结果的一个验收试验。如果试验顺利通过,则结果被接受并且试验就进入系统的其后部分。如不合格,第一段的任何副作用将被消除,然后执行被称为第一替补的第二段。它后面也跟随一个验收试验,并且也按第一段的情况一样地处理,需要时可提供第二、第三甚至更多的替补。

参考文献:

软件故障裕度的系统结构. B. Randall. IEEE Trans Software Engineering Vol. SE-1, No. 2, 1975.

故障裕度——原理和实践. T. Anderson, P. A. Lee, Prentice-Hall, 1981.

C.3.7 反向恢复

注:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:为了在存在一个或多个故障时保障正确的功能运行。

描述:当检测到一个故障时,系统就复位到一个较早的内部状态,该状态的一致性已在过去被证明过。这种方法意味着内部状态通常保存在所谓良定义检验点上。可以全局性地(对整个数据库)或者递增地(只在检验点之间改变)进行。然后系统必须补偿这些改变,这些改变是在平均时间中因使用日记(动作的审核跟踪)和补偿(消除这些改变的所有影响)或者外部(手动)相互作用而产生的。

参考文献:

软件故障裕度(软件发展方向,No. 3). M. R. Lyu(ed),John Wiley & Sons,April 1995,ISBN 0471950688.

C.3.8 正向恢复

注:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:为了在存在一个或几个故障时保障正确的功能运行。

描述:如探测到一个故障,系统的当前状态将被处理从而得到一个状态,这个状态将和稍后某时的状态一致,这种观念特别适用于具有一个小型数据库和内部状态变化速度快的实时系统。假定了至少一部分系统状态受环境的影响并且只有一部分状态受环境的影响(强制)。

参考文献:

软件故障裕度(软件发展方向,No. 3). M. R. Lyu(ed),John Wiley & Sons,April 1995,ISBN 0471950688.

C.3.9 重试故障恢复机制

注:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:利用重试机制从一个已发现的故障状况尝试恢复功能。

描述:在发现一个故障或错误状况的事件中,通过重新执行同样的代码尝试恢复情况。重试恢复可能和一次软件超时,或者一次任务监视动作之后的一次重新引导和一次重新起动过程,或者一次小的重新调度和重新起动任务完全一样。重试技术常用于一个通信故障或者错误恢复中,并且利用一个通信协议错误(检验和等)或者一个通信确认响应时间超时来标记重试状况。

参考文献:

可靠的计算机系统:设计和评价. D. P. Siewiorek and R. S. Schwartz, A. K. Peters Ltd. ,1998,ISBN 156881092X.

C.3.10 存储执行用例

注:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:当软件尝试执行一条不允许的路径时,将强迫软件安全失效。

描述:执行每个程序的所有有关详情都被编成文档。在正常运行过程中,每个程序的执行情况都同先前记录的详情作比较。如果有差异,就采取一个安全动作。

执行文档包含各条判定——判定路径(DD 路径)的顺序或者访问各个数组、记录或者卷册的顺序,或者二者。

各种存储执行路径的方法都是可能的,为了把执行顺序映射成单个大数或数字序列,可以使用散列编码方法。在正常运行过程中,在发生任何输出操作之前,必须对照存储的用例检验执行路径值。

因为在一个程序过程中,判定——判定路径的可能组合是很大的,把这些程序作为整体来处理是行不通的,在这种情况下,在软件模块层可使用本技术。

参考文献:

失效——安全软件——一些原理和一个用例研究. W. Ehrenberger. proc. SARSS 1987, Altrincham, Manchester, UK, Elsevier Applied Science, 1987.

C.3.11 故障弱化

注:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:借助丢掉比较不致命的一些系统功能保持更关键的功能可用,从而忽视失效。

描述:本技术给出将被系统执行的各种功能的优先级,设计保证了当执行所有系统功能的资源不足时,将优先执行高优先级的功能。例如,错误和事件记录功能的优先级比系统控制功能的优先级低,在此情况下,当与错误记录有关的硬件发生故障时,系统控制将继续进行。另外,当系统控制硬件发生故障,而错误记录硬件并未出故障时,错误记录硬件将接管控制功能。

它主要适用于硬件但也适用于总系统。从最上层设计阶段开始就必须考虑本技术。

参考文献:

空间航天飞机软件. C. T. Sheridan, Datamation, V01. 24, July 1978.

容错计算的进展 第1卷:可靠的计算和容错系统. Edited by A. Avizienis, H. Kopetz and J. C. Laprie, Springer Verlag, 1987, ISBN 3-211-81941-X.

故障裕度量、原理和实践. T. Anderson and P. A. Lee, 可靠的计算和容错系统. Springer Verlag, 1987, ISBN 3-211-82077-9.

C.3.12 人工智能故障纠正

注1:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:借助引入方法和过程模型的组合以及某种在线安全性和可靠性分析,以便能用一种很灵活的方式应答可能的危险。

描述:基于人工智能(AI)的系统将以一种很有效的方式,在一个系统的不同通道中支持故障预测(计算发展趋势),故障纠正、维护和监控动作,因为规则可以从规范直接导出并可对照这些规范来检验规则。本方法,特别是当使用功能或者描述形式的模型和方法的一种组合时,可有效地避免某些共同原因故障,记住这些故障隐式地通过已有的某些设计和实现规则引入规范中。

为了满足要求的安全完整性,应这样选择这些方法:即故障可得到纠正,失效的影响将减到最小。

注2:有关纠错数据的警告可参看 C.3.2,关于本技术的反面意见可参看 GB/T 20438.3—2006 的表 A.2 中第5项。

参考文献:

适用于软件开发的自动程序设计技术:基于异常处理的一种方法. M. Bidoit et al, proc. 1st int. conf. on Applications of Artificial Intelligence to Engineering Problems, Southampton, 165-177, 1986.

人工智能和专家系统设计. G. F. Luger and W. A. Stubblefield Benjamin/cummings, 1989.

C.3.13 动态再配置

注:在 GB/T 20438.3—2006 的表 A.2 中引用了本技术/措施。

目的:为保持系统功能性而忽视一个内部故障。

描述:系统的逻辑结构必须要能把它映射成系统可用资源的一个子集,结构要求能检测一个物理资源中的一次失效,然后重新把逻辑结构映射回剩余的起作用的有限资源。虽然本概念传统上更多地局限于用来恢复有故障的硬件单元,但如果有足够的"运行时间冗余"允许一个软件重试或者如有足够的冗余数据使得单独的和隔离开的失效不重要的话,它也适用于有故障的软件单元。

在系统设计的第一阶段就必须考虑本技术。

参考文献:

可再配置的控制计算机设计中的关键问题. H. Schmid, J. Lam, R. Naro and K. weir, FTCS 14 June 1984, IEEE, 1984.

处理机的分配过程:一种容错方法. G. Kar and C. N. Nikolaou, Watson Research Centre, Yorktown, June 1984.

C.4 开发工具和编程语言

C.4.1 强类型编程语言

注:在 GB/T 20438.3—2006 的表 A.3 中引用了本技术/措施。

目的:通过使用一种语言来降低故障概率,这种语言允许使用编译器进行一次高级检验。

描述:当编译一个强类型程序设计语言时,对使用的变量类型进行许多检验,例如,在过程调用和外部数据存取中,对任何不符合预定规则的应用,编译都将会失败并产生一条出错报文。

通常这样的一些语言允许根据基本的语言类型(比如整型、实型)定义用户定义的数据类型。然后按同基本类型严格相同的办法使用这些类型。为了保证使用的类型正确应严格加强检验,即使程序是由分离的编译单元建造的,整个程序都要施行这些检查,甚至当注明资料来自独立编译的软件模块时,这些检查也能保证过程变元的数目和类型相匹配。

强类型语言一般支持良好的软件工程实践的其他特征,比如容易分析的控制结构(例如 if..then..else(如…则…否则…),do..while(在…的同时做……)等),这些结构可产生良结构程序。

强类型语言的典型例子是 Pascal、Ada 和 Modula2。

参考文献:

为了寻求有效的多样性:容错飞行控制软件的 6 种语言研究. A. Avizienis, M. R. Lyu and W. Schutz. 18th Symposium on Fault-Tolerant Computing, Tokyo, Japan, 27-30 June 1988, IEEE Computer SocietyPress, 1988, ISBN 0-8186-0867-6.

ISO/IEC 8652:1995 信息技术 程序设计语言 Ada.

ISO/IEC 10514.1:1996 信息技术 程序设计语言 第 1 部分:Modula-2,基础语言.

ISO 7185:1990 信息技术 程序设计语言 Pascal.

C.4.2 语言子集

注:在 GB/T 20438.3—2006 的表 A.3 中引用了本技术/措施。

目的:为了减少引入程序设计故障的概率和增大检测任何剩余故障的概率。

描述:例如使用静态分析法检验语言以便确定编程结构,这些结构既易出错又难分析。然后定义一种可消除这些结构的语言子集。

参考文献:

在安全和保密软件标准中对编程语言的要求. B. A. Wichmann. Computer Standards and Interfaces. Vol. 14, pp433-441, 1992.

SaferC:高集成度和安全关键系统的开发软件. L. Hatton, McGraw-Hill, 1994, ISBN 0-07-70640-0.

C.4.3 经认证的工具和经认证的翻译器

注:GB/T 20438.3—2006 的表 A.3 中引用了本技术/措施。

目的:在开发软件的各个阶段,这些工具对帮助开发者是有必要的。只要可能,就应认证这些工具使之能假定有关输出正确性的某级置信度。

描述:一般是由一个独立的,通常是国家、团体,对照单独设立的判据(典型地是国家或者国际标准)来认证一个工具,理想地这些工具可使用于所有开发阶段(规范、设计、编码、测试和确认)以及使用于配置管理,并经受检验。

到此为止,只有编译(翻译)器正式经受过检验规程,它们由国家认证团体拟订。并且它们根据国际标准(比如 Ada Pascal 的那些标准)行使编译(翻译)。

注意经认证的工具和经认证的翻译器通常只根据它们各自的语言或过程标准进行过认证,一般并未对安全性进行过任何认证。

参考文献:

Pascal 确认序列. 英国发行商:BSI Quality Assurance, Po Box 375, Milton Keynes, MK146LL.

Ada 确认序列. 英国发行商:National Computer Centre(NCC), Oxford Road Manchester, England.

C.4.4 工具和翻译器:通过使用提高置信度

注:在 GB/T 20438.3—2006 的表 A.3 中引用了本技术/措施。

目的:为了避免在开发、验证和维护一个软件包的过程中出现的翻译器失效引起的任何困难。

描述:在先前的许多项目中并未发现性能异常的证据的情况下使用一个翻译器时,除非存在正确性能的一些其他保证(例如:参看 C.4.4.1),应避免使用没有运行经验或者带有任何已知的严重故障的翻译器。

在一个与安全有关的项目过程中,当翻译器显示的置信度较低时,记录下有关的语言结构并应小心避免使用这些结构。

对于本作业方法的另一种方案是把语言的使用只限定为常用的特征。

这些建议是根据许多项目的经验提出来的。不成熟的翻译器已显示出它对于任何软件开发都是一个严重的障碍。它们使得开发一个安全软件成为不可能。

事实上也了解目前不存在证明所有工具或者翻译器组成部分正确性的方法。

C.4.4.1 源程序和执行代码的比较

目的:为了检验用来产生一幅 PROM(可编程只读存储器)映像的工具,该映像中未引入任何错误。

描述:为了得到成分"对象"模块逆向设计 PROM 映像。把这些"对象"模块逆向设计成汇编语言文件。使用适当的技术,把这些逆向生成的汇编语言文件同最初用来产生 PROM 的实际源文件进行比较。

本技术的主要优点是对所有程序而言,用于产生 PROM 映像的工具不必被确认。本技术可验证用于特殊安全相关系统的源文件的翻译的正确性。

参考文献:

证明源码和 PROM 内容的等价. D. J. Pavey and L. A. Winsborrow. The Computer Journal Vol. 36. No. 7,1993.

源码和 PROM 内容等价性的形式证明:一个工业实例. D. J. Pavy and L. A. Winsborrow. Mathematics of Dependable Systems, Ed. C. Milchell and V. Stavridou, Clarendon Press, 1995, ISBN 0-198534-91-4.

反应堆保护系统软件的形式验证的回顾. D. J. Pavey, L. A. Winsborrow, A. R. Lawrence. Proceedings of the Second Safety Through Quality Conference, 1995, ISBN 1-897851-5.

在一个安全关键软件应用中保证正确性. L. A. Winsborrow and D. J. Pavey. High Integrity Systems, Vel. No. 5, PP453-459, 1996.

C.4.5 可信的/经验证的软件模块和部件库

注:在 GB/T 20438.3—2006 的表 A.3 中引用了本技术/措施。

目的:为了避免对于每个新应用都需要对软件模块和硬件部件设计进行彻底的重新确认或重新设计。也为了促进还未正式或精确确认但都有相当长的运行历史可用的设计。

描述:良设计和良结构的 PES(可编程电子系统)由许多硬件和软件部件和模块构成,这些部件和模块彼此之间有明显的差别并以一些明确规定的方式相互作用。

为不同应用设计的各种 PES 包含许多同样的或很相似的软件模块或部件。建造这种通用软件模块库使得大部分资源在确认不只一个应用共享的设计时是必要的。

此外,这些软件模块在多个应用中使用为成功运行使用提供了实验证明。这种经验证据无疑增强了用户对软件模块大概会产生的信任。

C.2.10 描述了一种方法,利用此法,可按可信度分类一个软件模块。

参考文献:

在实践中软件重用和逆向设计. P. A. V. Hall(ed.), Chapman & Hall, 1992, ISBN 0-412-39980-6.

DIN V VDE 0801 A1:Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben(安全相关系统中的计算机原理), Änderung 1 Zu DIN V VDE 0801/01.90. Beuth-Verlag, Berlin, 1994.

C.4.6 合适的编程语言

注:在 GB/T 20438.3—2006 的表 A.3 中引用了本技术/措施。

目的:为了尽可能多地支持 GB/T 20438 的要求,特别是在防御性编程、强类型编程、结构化编程,或许还有断言编程中,所选编程语言毫不费力地就可产生一个易于验证的代码并有助于程序的开发、验证和维护。

描述:应充分地、清楚地定义该语言。该语言应是面向用户或问题的而不是面向处理机/平台的机器语言。广泛使用的语言或它们的子集比专用语言更好。

除了已经提到的特征之外,该语言应是为下列目的而提供的:

——程序块结构;

——翻译时间检验;和

——运行时间类型和数组限界检验。

本语言应有助于:

——使用小型的和可管理的软件模块;

——限制在专用软件模块中存取数据;

——定义变量的子范围;和

——构建任何其他型式的错误限制结构。

如果系统的安全运行与实时约束有关,那么,也将为异常/中断处理提供语言。

需要用一个适当的翻译器、合适的预存软件模块库、一个调试程序和一些工具(用于版本控制和开发两方面)来支持语言。

当前,在开发 GB/T 20438 时,还不清楚面向对象的语言是否比其他传统的语言更好。

使验证困难并因此应避免的特征是:

——除子程序调用外的无条件转移;

——资源;

——指示字、堆阵或任何类型的动态变量或对象;

——在源码级的中断处理;

——回路、程序块和子程序的多入口和出口;

——隐含变量初始化或说明;

——变体记录及其等效性;以及

——过程参数。

低级语言,特别是汇编语言,存在因它们面向处理机/平台机的固有特性产生的问题。

一种要求的语言属性是它的设计和使用应产生一些程序,执行这些程序的结果是可预知的。给定一种适当定义的编程语言时,就存在一个能保证可预测程序执行结果的子集。(一般)不能统计地确定这个子集,虽然许多统计学的约束有助于保证执行的可预测性。典型地,这需要证明数组索引是在限界之内以及不会发生数字溢出等。

表 C.1 给出了建议的专用编程语言。

表 C.1 建议的专用编程语言

编 程 语 言	SIL1	SIL2	SIL3	SIL4
1 ADA	HR	HR	R	R
2 具有子集的 ADA	HR	HR	HR	HR
3 MODULA-2	HR	HR	R	R
4 具有子集的 MODULA-2	HR	HR	HR	HR
5 PASCAL	HR	HR	R	R
6 具有子集的 PASCAL	HR	HR	HR	HR

表 C.1(续)

编 程 语 言	SIL1	SIL2	SIL3	SIL4
7 FORTRAN77	R	R	R	R
8 具有子集的 FORTRAN77	HR	HR	HR	HR
9 C	R	—	NR	NR
10 具有子集和编码标准并使用静态分析工具的 C	HR	HR	HR	HR
11 PL/M	R	—	NR	NR
12 具有子集和编程标准的 PL/M	HR	R	R	R
13 汇编程序	R	R	—	—
14 具有子集和编码标准的汇编程序	R	R	R	R
15 梯形图	R	R	R	R
16 具有定义的语言子集的梯形图	HR	HR	HR	HR
17 功能块图	R	R	R	R
18 具有定义的语言子集的功能块图	HR	HR	HR	HR
19 结构化文本	R	R	R	R
20 具有定义的语言子集的结构化文本	HR	HR	HR	HR
21 顺序功能图	R	R	R	R
22 具有定义的语言子集的顺序功能图	HR	HR	HR	HR
23 指令表	R	—	NR	NR
24 具有定义的语言子集的指令表	HR	R	R	R

注 1：建议 R、HR 和 NR——在 GB/T 20438.3—2006 的附录 A 中做了说明。

注 2：系统软件包括作为系统组成部分而配置的操作系统、驱动程序、嵌入功能和软件模块，典型地，软件由安全相关系统卖方提供。应小心选择语言子集以避免可能引起实现故障的复杂结构。应检查语言子集使用是否得当。

注 3：应用软件是为专门的安全应用而开发的软件。在许多情况下，由最终用户或者一个面向应用的承包商开发软件。当同一建议有许多编程语言时，开发者应选择在工业或工厂中人们常用的那一种。应仔细选择语言子集以避免可能会引起实现故障的复杂结构。应检查语言子集使用是否得当。

注 4：如果在表中未列出一种专用语言，绝不能认为它被排除在外了。它应符合 GB/T 20438。

注 5：表项 15～24 可参看 IEC 61131-3。

参考文献：

关键计算机系统的可靠性 1. F. J. Redmill, Elsevier Applied Science, 1988, ISBN 1-85166-203-0.

IEC 60880:1986 核电站安全相关系统中的计算机软件.

IEC 61131-3:1993 可编程控制器 第 3 部份:编程语言.

ISO/IEC 1539-1:1997 信息技术 编程语言 Fortran 第 1 部分:基础语言.

ISO/IEC 7185:1990 信息技术 编程语言 Pascal.

ISO/IEC 8652:1995 信息技术 编程语言 Ada.

ISO/IEC 9899:1990 编程语言 C.

ISO/IEC TR 10206:1991 信息技术 编程语言 扩展的 Pascal.

ISO/IEC/ 10514:1996 信息技术 编程语言 第 1 部分:Modual 2,基础语言.

ISO/IEC 10514-3:1998 信息技术 编程语言 第 3 部分:面向对象的 Modual 2.

ISO/IEC 14882:1998 编程语言 C++.

ISO/IEC/TR 15942 Ada 编程语言在高集成系统中的使用指南[1)].

C.5 验证和修改

C.5.1 概率测试

注：在 GB/T 20438.3—2006 的表 A.5、表 A.7 中引用了本技术/措施。

目的：为了得到被研究软件的可靠性属性的定量数值。

描述：该定量数值考虑到了有关的置信度级和显著性，并能给出：

——每一要求的一个失效概率；

——在某一时段的一个失效概率；和

——错误防止的一个概率。

从这些数值可导出其他一些参数，比如：

——无失效执行的概率；

——残存概率；

——可用性；

——MTBF(平均失效间隔时间)或者失效率；以及

——安全执行的概率。

概率考虑是以一种概率测试和操作经验两者为基础的。通常，测试用例或者观察的操作用例数目都很大。典型地，要求操作模式的测试占用的时间比连续操作模式经过的时间少得多。

一般使用自动测试工具来提供测试数据和监控测试输出。大量的测试要在具有适合过程模拟外围设备的大型主计算机上运行。根据系统和随机硬件两种观点来选择试验数据。整个测试控制，例如，保证了一个测试数据分布，而随机选择能管理各个测试实例的细节。

如上所述，各个测试的导线系统、测试执行和测试监控由详细的测试目的确定。其他重要条件由数学前提给出，此数学前提是在测试评价满足预定测试目的时必须满足的。

有关任何测试对象行为的概率系数还可从操作经历导出，倘若满足同样的条件，评价测试结果就可使用同样的数学。

实际上，使用这些技术很难证明超高级可靠性。

参考文献：

通过环境模拟测试软件(CONTESSE Report). Available until December 1998 from: Ray Browne, CIID, DTI, 151 Buckingham Palace Road, London, SW1W9SS, UK, 1994.

基于软件的系统的超高可靠性的确认. B. Littlewood and L. Strigini, Comm. ACM36(11). 69-80, 1993.

软件可靠性工程手册. M. R. Lyu(ed.)IEEE Computer Society Press, McGraw-Hill, 1995, ISBN 0-07-039400-8.

C.5.2 数据记录和分析

注：在 GB/T 20438.3—2006 表 A.5 和表 A.8 中引用了本技术/措施。

目的：为使验证、确认、评估和维护较容易，软件项目中的所有数据、判定和基本原理都编成文档。

描述：在一个项目期间，保持详细的记录文档，该文档包括：

——对每个软件模块执行的测试；

——判定和它们的基本原理；

——问题和解决办法。

---

1) 将出版。

在项目进行过程中和结束时，可分析这个文档以便建立各种各样的信息。特别是当开发项目期间做出某些判定的基本原理还不为维护工程师所知时，数据记录对计算机系统的维护是很重的。

参考文献：

关键计算机系统的可靠性 2. F. J. Redmil，Elsevier Applied Science，1989，ISBN 1-85166-381-9.

C.5.3 界面测试

注：在 GB/T 20438.3—2006 的表中 A.5 中引用了本技术/措施。

目的：为了检测子程序界面中的错误。

描述：测试的详细程度或者完整性分成几级是可行的。最重要的一些级是对下列量的测试：

——处于它们的极限值的所有界面变量；

——分别处于它们的极限值的所有界面变量以及处于额定值的另一些界面变量；

——每个界面变量和处于额定值的其他界面变量范围中的所有值；

——组合(只有小界面这种组合才是可行的)中的所有变量的所有值；

——与每个子程序的每次调用有关的规定的测试条件。

当界面不包括检测到的错误参数值的断言时，这些测试是特别重要的。在已生成预存子程序的新配置后，它们也是重要的。

C.5.4 边界值分析

注：在 GB/T 20438.3—2006 的表 B.2、表 B.3 和表 B.8 中引用了本技术/措施。

目的：为了检测发生在参数极限值或边界值处的软件错误。

描述：根据等价关系(参看 C.5.7)把程序的输入范围分成若干输入类别。测试应包括这些类的边界值和极限值。测试将检查和程序中规范一致的规范的输入域的边界。在直接和间接翻译中使用值 0 通常易出错并要求注意：

——0 作为除数；

——空白 ASCⅡ字符；

——空栈或者表元素；

——满矩阵；

——0 表项。

通常，输入的边界直接对应于输出范围的边界。为了把输出强制成它的限定值，应写入测试用例。还考虑了是否有可能规定一个能使输出超过规范边界值的测试用例。

如果输出是一个数据序列，例如一个打印的表，应特别注意第一个和最后一个元素以及不包含元素、只包含一个和两个元素的表。

参考文献：

IEC 61704 可靠性评估的软件测试方法的选择指南.

软件测试工艺. G. Myers，Wiley&Sons，New York，1979.

C.5.5 错误推测

注：在 GB/T 20438.3—2006 的表 B.2 和表 B.8 中引用了本技术/措施。

目的：为了消除共同编程出错。

描述：测试经验和直觉同受试系统的知识和奇特性结合可把一些未分类的测试实例附加到计划的测试实例集上。

特殊的值或者值的组合易出错。可以从检查检验表得出某些感兴趣的测试实例。也要考虑系统是否足够健壮。例如，在面板上按按钮太快或者太频繁？同时按两个按钮会发生什么情况？

参考文献：

软件测试工艺. G. Myers，Wiley & Sons，New York，1979.

C.5.6 错误播种

注：在 GB/T 20438.3—2006 的表 B.2 中引用了本技术/措施。

目的:为了弄清一组测试事例是否适合。

描述:在程序中插入(播种)一些已知类型的错误并在测试条件下执行测试实例程序。如只发现播种的那些错误,测试实例集是不合适的。发现的播种错误与播种错误的总数之比可用来估算发现的真实错误与错误总数之比。它给出估算剩余错误数的可能性并因此也给出了剩余的测试工作。

$$\frac{\text{发现的播种错误}}{\text{播种的错误总数}} = \frac{\text{发现的真实错误}}{\text{真实错误的总数}}$$

检测所有播种的错误既可指示测试用例集是合适的,还可指示播种的错误太容易发现了。本方法的限制是为了得到任何有用的结果,错误的类型和播种的位置必须反映真实错误的统计分布情况。

参考文献:

软件故障注入. J. M. Voas and G. McGraw, Wiley 1998.

C.5.7 等价类和输入分区测试

注:在 GB/T 20438.3—2006 的表 B.2 和表 B.3 中引用了本技术/措施。

目的:为了使用最少的测试数据恰当地测试软件。通过选择考验软件所需的输入范围的分区来得到测试数据。

描述:本测试策略是以输入的等价关系为基础的,该关系确定了输入范围的一个分区。

为了包括早先规定的所有分区,应对测试实例进行选择。从每个等价类型至少要选一个测试实例。

对于输入分区存在两种基本的可能性,它们是:

a) 从规范得出的等价类型——规范的解释既可是面向输入的,例如选择值按相同的方法进行,也可以是面向输出的,例如值的集合产生相同的功能结果。

b) 从程序的内部结构得出的等价类型——从程序的静态分析确定等价类型结果,例如值的集合产生同样的执行路径。

参考文献:

软件测试工艺. G. Myers, Wiley&Sons, New York, 1979.

C.5.8 基于结构的测试

注:在 GB/T 20438.3—2006 的表 B.2 中引用了本技术/措施。

目的:为了使用考验某些程序结构子集的测试。

描述:在程序分析的基础上,选择一组输入数据,从而可考验大部分(并且常常是预定的目标)程序代码。根据要求的精确程度,代码测量所包括的范围变化如下:

——语句:它是精确性最差的测试,因为不用考验一个条件语句的两个分支就能执行所有代码语句。

——分支:应检验每个分支的两边。对某些类型的防御性代码这可能是不实际的。

——复合条件:考验一个复合条件分支(即用 AND(与)/OR(或)链接的)的每个条件。参看 MCDC(修改的条件判定范围,参考 DO178B)。

——LCSAJ:一个线性代码序列和转移(LCSAJ)是包括条件语句和用一个转移结束的任何代码语句序列。由于执行早期代码强加给输入数据的约束,许多潜在的子路径是不可行的。

——数据流:根据数据的使用情况选择执行路径。例如,在同一变量的情况下,一条路径是写和读二者。

——调用图:一个程序由从其他子程序调出的子程序组成。调用图是程序中子程序调用的树形图。计划的测试包括树中的所有调用。

——基本路径:从开始到结束的一个有限路径的最小集合中的一条路径,因此包括所有弧线。(在这个基本集合中路线的重叠式组合可形成通过程序那一部分的任何路径),已经显示出,所有基本路径的测试对查找错误是有效的。

参考文献：

路径分析测试策略的可靠性. W. Howden. IEEE trans Software Engineering, Vol. SE-3, 1976.

航空系统和设备检定中软件的考虑. DO178B, RTCA, December 1992.

结构测试. McCabe; NBS Special Rublication 500-99, 1982.

软件可靠性研究. Walsh [USA] National Computer Conference, 1979.

C.5.9 控制流分析

注：在 GB/T 20438.3—2006 的表 B.8 中引用了本技术/措施。

目的：为了检测差的和潜在有错误的程序结构。

描述：控制流分析是查找代码可疑区的一种静态测试技术，该代码的可疑区并不遵循好的编程作法。对程序进行分析可产生一个有向图，此图能进一步分析：

——不可存取的代码，例如无条件转移，它留下执行不到的代码块。

——打结的代码：良结构代码有一个可简化的控制图，该代码只能简化成一个单节点。相反，差结构代码只能简化成几个节点构成的一个结。

参考文献：

当程序的信息流和数据流. J. F. Bergeretti and B. A. Carre, ACM Trans. on Prog. Lang. and Syst., 1985.

C.5.10 数据流分析

注：在 GB/T 20438.3—2006 的表 B.8 中引用了本技术/措施。

目的：为了检测差的和潜在有错误的程序结构。

描述：数据流分析是一种静态测试技术，此技术把从控制流分析得到的信息同在各代码分区中读或写的变量有关的信息组合起来，可以检查分析的：

a) 在给变量赋值之前读出的那些变量——可以通过说明一个新变量时总是分配给它一个值来避免这一点。

b) 只写而决不会读的那些变量——这可指示省略的代码。

c) 只写而决不会读的那些变量——这能指示冗余代码。

一个数据流的异常结构不一定直接相应于一个程序错误，但如果避免了这种异常，代码就很少包含错误。

参考文献：

当程序的信息流和数据流. J. F. Bergeretti and B. A. Carre, ACM Trans. on Prog. Lang. Lang. and Syst., 1985.

C.5.11 寄生回路分析

注：在 GB/T 20438.3—2006 的表 B.8 中引用了本技术/措施。

目的：为了检测一个系统中的一条意想不到的路径或逻辑流。在某些条件下，该系统启动一个不希望有的功能或者禁止一个需要的功能。

描述：一条寄生回路路径可以由硬件、软件、操作员动作或者这些要素的组合构成。寄生回路不是硬件失效的结果而是无意中设计人系统的或者编码到软件程序中的潜伏条件，在某些条件下，它们能引起系统或者软件程序出错。

寄生回路的类型有：

——引起电流、能量或逻辑序列沿一条意想不到的通路或者在一个非预定方向流动的潜通路。

——寄生计时，按这种计时，事件以一种不可预料的或者冲突的顺序发生。

——寄生指示，它引起系统运行状况的显示发生歧义和虚假，并因此造成操作员采取一个不需要的行动。

——寄生标记，它错误地或不准确地标记系统功能，例如系统输入、控制、显示、汇集信息等，并因此

误导一个操作员对系统施加一个错误的激励。

寄生回路分析有赖于识别硬件和软件结构的基本拓扑模式(例如对于软件提出了6种基本模式)。借助关于使用问题的一个检验表以及基本拓朴成分之间的关系进行分析。

参考文献:

寄生分析和软件寄生分析. S. G. Godoy and G. J. Engels. J. Aircraft Vol. 15, No. 8, 1978.

寄生回路分析. J. P. Rankin, Nuclear Saftey, Vol. 14, No. 5, 1973.

C.5.12 符号执行

注:在GB/T 20438.3—2006的表B.8中引用了本技术/措施。

目的:为显示源码和规范之间的一致性。

描述:在所有赋值中,在用右手边替换左手边之后再评价程序变量。条件分支和回路翻译成布尔(Boolean)表达式。每个程序变量的最后结果是一个符号表达式。可对照预期的表达式来检验它。

参考文献:

使用符号执行形式程序验证. R. B. Dannenberg and G. W. Ernst. IEEE Transaction on Software Engineering, Vol. SE-8, No. 1, 1982.

符号执行和软件测试. J. C. King, Communications of the ACM, Vol. 19, No. 7, 1976.

C.5.13 形式检验

注:在GB/T 20438.3—2006的表A.9中引用了本技术/措施。

目的:为了证明一个程序或者规范的正确性而不用执行它,证明中使用了理论模型和数学模型及规则。

描述:在程序中的各个位置讲述了许多断言,它们作为先决条件和后续条件被用于程序中的各条路径。验证包括显示根据一组逻辑规则,程序把先决条件转换成后续条件以及程序终止。

在本评述中描述了几种形式方法,例如CCS, CSP, HOL, LOTOS, OBJ, 时序逻辑, VDM和Z(参看描述这些方法的C.2.4)。

形式证明的一种替代技术是精确的变元。制定的形式验证的一个大纲提出了主要步骤,但并不包括所有的数学细节。它是一种较弱的验证技术,这种技术建立了一种试图进行证明时的可行方法。

参考文献:

软件开发——一种精密的方法. C. B. Jones. Prentice-Hall, 1980.

使用VDM开发系统软件. C. B. Jones. Prentice-Hall, Ind Edition, 1990.

C.5.14 复杂性度量

注:在GB/T 20438.3—2006的表A.9中引用了本技术/措施。

目的:为了从软件本身的属性或者从它的开发史或者测试史预测程序属性。

描述:这些模型评价软件的某些结构属性并把它同一个描述的属性比如可靠性或者复杂性联系起来。为了评价大部分量度方法需要一些软件工具。下面给出了可以使用的一些量度:

a) 图形理论复杂性——本量度方法可用于生命周期的初期进行比较评估,它以程序控制图的复杂性为基础,复杂性用它的秩数来表示。

b) 启动某个软件模块的方式数(可访问性)——较好的说法是可访问一个软件模块的次数,更确切的说法是它被调试的次数。

c) Halstead(赫尔斯梯德)型度量学——这种措施通过数操作码和操作数来计算程序长度。它提供了复杂性和长度的一种量度,它为估算今后开发资源时作比较提供了一个基准。

d) 每个软件模块的进出口数——最小化进口/出口点数是结构化设计和编程技术的一个关键特征。

参考文献:

软件量度法:一种精确和实际的方法. N. E. Fenton, International Thomson Computer Press, 1996,

ISBN 1-85032-275-29,2nd Edition.

—种复杂性量度方法. T. J. McCabe. IEEE Trans on Software Engineering, Vol. SE-2, No. 4, December 1976.

软件质量评估的模型和量度方法. S. N. Mohanty. ACM Computing Surveys, Vol. 11, No. 3 September 1979.

软件科学的基本原理. M. H. Halstead. Elsevier, North Holland, New York, 1977.

C.5.15 Fagan(菲根)检查法

注：在 GB/T 20438.3—2006 的表 B.8 中引用了本技术/措施。

目的：为了揭示程序开发的各个阶段中的错误和故障。

描述：以查找错误和故障为目的对质量保证文档的一种“形式”审核。检验过程包括 5 个分阶段：计划、准备、检验、修改、跟踪。每个分阶段有它自己单独的目标。必须检查整个系统开发(规范、设计、编码和测试)。

参考文献：

为减少程序开发中的错误的设计和编码检查. M. E. Fagan, IBM Systems Journal, No. 3. 1976.

C.5.16 走查/设计复审

注：在 GB/T 20438.3—2006 的表 B.8 中引用了本技术/措施。

目的：为了尽早和尽可能经济地检测某个开发的产品中的故障。

描述：IEC 已公布了关于形式设计复审的指南，它包括形式设计复审的一般描述，它们的目标，各种设计复审类型的细节，设计复审小组的组成和相关的任务和职责。IEC 文件还提供了计划和实施形式设计复审的一般导则以及有关一个设计复审小组中单独的专家的任务。专家作用的例子包括：除设计复审外，还有可靠性、维护支持和可用性复审。

IEC 建议“应对所有新产品/过程、新应用和对现存产品及生产过程的修改本进行一次形式设计复审，制造过程将影响功能、性能、安全性、可靠性、检查可维性和可用性的能力、估算价格的可能性，以及影响最终产品/过程、用户或者旁观者的其他特性”。

一次代码走查包括走查小组选择程序的一个小的纸型测试实例文件集，代表性的输入集合和预计的相应输出集合，然后通过程序逻辑手动绘出测试数据。

参考文献：

IEC 61160:1992 形式设计复审. 修订版 1(1994).

软件检查. T. Gilb, D. Graham, Addison-Wesley, 1993, ISBN 0-201-63181-4.

C.5.17 原型设计/动画

注：在 GB/T 20438.3—2006 的表 B.3 和表 B.5 中引用了本技术/措施。

目的：为了根据给定的一些约束检查实现系统的可行性。为了把系统的说明符译本传送给买方以便查出误解。

描述：选择系统功能、约束和性能要求的一个子集。使用高级工具建立一个原型。在这一阶段，不需要考虑比如目标计算机、实现语言、程序规模、可维修性、可靠性和可用性这些约束。对照买方判据来评价原型并且通过这种评价来修改系统要求。

参考文献：

快速原型分析作为一个实时软件开发工具的应急备份. J. E. cooling, T. S. Hughes, Proc. 2nd int . Conf. on Software Engineering for Real-time Systems, Cirencester, UK, IEE, 1989.

通过快速原型分析进化软件. Luqi, IEEE Computer 22(5), 13-27, May 1989.

原型分析入门. R. Rudde et al, springer Verlag, 1984, ISBN 3-540-13490-5.

原型分析工作会议会刊. Namur, October 1983, Budde et al, Springer Verlag, 1984.

把一种可执行的规范语言用于一个信息系统. S. Urban et al. , IEEE Trans Software Engineering,

Vol. SE-11, No. 7, July 1985.

C.5.18 过程模拟

注：在 GB/T 20438.3—2006 的表 B.3 中引用了本技术/措施。

目的：为了测试一个软件系统的功能连同它与外界的界面而不用允许它对真实世界作任何修改。

描述：建立一个系统，该系统只用于测试目的，它模拟受控设备(EUC)的行为。

模拟可以只是软件或者软件和硬件的组合。它必须：

——提供输入，这些输入等同于实际安装受控设备时存在的输入。

——与受试软件的输出在某种程度上相符合，这种符合程度就是要忠实地代表受控设备。

——保证能为操作员输入提供任何干扰，这种干扰是受试系统需要克服的。

当正在测试软件时，模拟可以是带有它输入和输出的目标硬件的模拟。

参考文献：

通过环境模拟测试软件(CONTESSE 报告). 1998 年 12 月之前可从下处得到：Ray Browne, CIID, DTI, 151 Buckingham Palace Road, London SW1W 9SS, UK, 1994.

C.5.19 性能要求

注：在 GB/T 20438.3—2006 的表 B.6 中引用了本技术/措施。

目的：为了制定一个软件系统的可证明的性能要求。

描述：对系统和软件需求规范两者执行一次分析以便确定所有通用和专用、明显的和隐含的性能要求。

检验每个性能要求以便逐个：

——确定获得成功的判据；

——对照成功判据确定一次测量是否能获得成功；

——确定这种测量的可能精度；

——确定项目分段，在这些阶段估算这些测量；

——确定项目分段，在这些阶段进行测量。

为了得到性能要求、成功判据和可能的测量的一张表，则要分析每个性能要求的可行性。主要的目标是：

——与至少一次测量相关的每个性能要求；

——在可能的情况下选择可能用于开发初期的精确、有效的测量；

——规定根本的和任选的性能要求和成功判据；

——在可能的情况下，对于多个性能要求利用使用单次测量的可能性。

C.5.20 性能建模

注：在 GB/T 20438.3—2006 的表 B.2、表 B.5 中引用了本技术/措施。

目的：为了保证系统工作能力足以满足规定的要求。

描述：要求规范包括专用功能的允许量和响应要求，也许还要和使用系统总资源的约束相结合。提出的系统设计将借助以下办法同说明的要求进行对比：

——生产一个系统过程和它们的相互作用的模型。

——通过每个过程确定资源的使用，例如，处理机时间、通信带宽、存储器件等。

——确定平均的和最差情况条件下要求分布在系统上的位置。

——计算在平均的和最差情况下各个系统功能的允许量和响应时间。

对简单的系统而言，一个分析解可能就足够了，而对较复杂的系统而言，某种形式的模拟可能更适于获得精确的结果。

在详细的建模之前，可以使用一种较简单的“资源预算”检验，它可把所有过程的资源需求总和起来。当需求超过设计的系统能力时，设计是不可行的。即使设计通过了这种检验，由于资源不足，性能

建模将显示产生的延迟和响应时间过长。为了避免这种情况，工程师常常使用总资源的一部分(例如50%)来设计系统以便减小资源不足的可能性。

参考文献：

实时系统设计：从规范到实现和验证. H. Kopetz et al, Software Engineering Journal 72-82, 1991.

C.5.21 雪崩/过载测试

注：在 GB/T 20438.3—2006 的表 B.6 中引用了本技术/措施。

目的：为了给测试对象加上一个例外的高工作负荷以便显示测试目标可以毫不费力地承受额定的工作负荷。

描述：存在各种可使用于雪崩/过载测试的测试条件。这些测试条件中的一些是：

——如果是以轮询方式工作，则当处在正常条件下时，每个时间单位测试对象的输入的变化要增大很多。

——当应要求工作时，则每个时间单位向测试对象的要求数目增加超过正常条件。

——如果一个数据库的规模起某种重要作用，则它增大超过正常条件。

——影响装置分别调到它们的最高或者最低速度。

——在极端情况时，所有影响因素尽可能同时提供给边界条件。

在这些测试条件下，可评价测试对象的时间行为，还可观察负荷变化的影响，并能检查内部缓冲器或者动态变量、堆栈等的正确规模。

C.5.22 响应定时和存储约束

注：在 GB/T 20438.3—2006 的表 B.6 中引用了本技术/措施。

目的：为了保证系统满足它的时序和存储要求。

描述：系统和软件的要求规范包含专用功能的存储和响应要求，也许还要结合对使用系统总资源的约束。

为了确定在平均的和最差情况条件下的分配要求而进行了一种分析。该分析需要估算每个系统功能使用的资源和经过时间，有几种办法可得到这些估算，例如，把一个现存系统或者原型设计同基准时间关键系统作比较。

C.5.23 影响分析

注：在 GB/T 20438.3—2006 的表 A.8 中引用了本技术/措施。

目的：为了确定一个软件系统的一个改变或者增强将对该软件系统中的其他软件模块以及其他系统的影响。

描述：在对软件进行一次修改或增强之前，为了确定这种修改或增强对该软件的影响，还为了确定哪些软件系统和软件模块将受影响，应进行一次分析。

在完成分析之后，需要对重新验证软件系统的问题作出决定。这与受影响的软件模块数、受影响的软件模块的临界状态和特性的改变有关。可能的决定有：

——只重新验证被改变的软件模块；

——重新验证所有受影响的软件模块；或者

——重新验证整个系统。

参考文献：

关键计算机系统的可靠性 2. F. J. Redmill, Elsevier Applied Science, 1989. ISBN 1-85166-381-9.

C.5.24 软件配置管理

注：在 GB/T 20438.3—2006 的表 A.8 中引用了本技术/措施。

目的：软件配置管理的目的是为了保证当那些可交付项有改变时，几种开发的可交付项的一致性。一般配置管理可用于硬件和软件开发两方面。

描述：软件配置管理是在整个开发过程中使用的一种技术。实质上，它要求编写每个重要的可交付

项的每个版本及各个可交付项的不同版本之间的每种关系的生产文档。产生的文档允许开发者确定一个可交付项(特别是它的一个成分)的一个改变对其他可交付项的影响。特别是可以从一致的几套成分版本可靠地重建各系统或分系统。

参考文献:

系统、设备、必需品和计算机程序的配置管理实践. MIL-STD-483.

软件配置管理. J. K. Buckle. Macmillan Press,1982.

软件配置管理. W. A. Babich. Addison-Wesley,1986.

国防设备的配置管理要求. UK Ministry of Defence standard 05-57 Issue3,July 1993.

C.6 功能安全评估

注:在 B.6 中也可看到有关的技术和措施。

C.6.1 判定表(真值表)

注:在 GB/T 20438.3—2006 的表 A.10 和表 B.7 中引用了本技术/措施。

目的:为了提供复杂逻辑组合和关系的一个清楚的和有条理的规范和分析。

描述:本方法使用了两种尺寸的表来简洁地描述布尔程序变量之间的逻辑关系。

本方法的简明性和表格特性使它适于作为一种分析用代码表示的复杂逻辑组合的方法。

当把本方法用作一个规范时,它有可能是可执行的。

C.6.2 危险和可操作性研究(HAZOP)

目的:为了确定在一个建议的或者现存的系统中的安全危险及其可能的起因和后果以及为减少它们发生的概率而建议的行动。

描述:一个由所研究的整个系统方面的专家组成的工程师小组通过一系列调度会加入设计的结构化检验中。他们考虑实际中设计的功能方面和怎样操作系统(包括人的活动和维护)两个问题。组长鼓励小组成员创造性地揭示潜在危险和通过提供系统的每一部分连同几个引导词:"none(没有)"、"more of(更多的)"、"less of(更少的)"、"part of(……的一部份)"、"more than(大于……)"(或者"as well as(以及)")和"other than(而不是……)"得出规程。研究它的可行性,怎样产生它,可能的后果(有无危险?),怎样避免,以及避免技术的花费是否很大等每个应用条件或者失效模式。

在以后的时间,常常需要进一步进行危险分析(常称为概率风险评估或者定量风险评估)以便更详细地研究主要危险。

在项目开发的许多分阶段都要进行危险研究,但最有效的执行时间是早到足以影响主要设计和可操作性判定的阶段。为项目内的会议规定一个固定的时间日程表是有帮助的;每次会议安排至少半天,每周安排不多于 4 次,因此可保持伴生文档的流动。会议文档将构成系统危险/安全档案的重要组成部分。

HAZOP 技术是在过程工业中逐渐形成的并且不修改 PES 的软件元素是难于应用的。对于 PES HAZOP(或者计算机 HAZOPs—"CHAZOPs")已提出了各种派生的方法。通常这些方法引进了一些新引导词并提出了系统性地包含系统和软件体系结构的方案。

参考文献:

拟制临时防御标准 00-58/1:对含有一个可编程电子系统的系统的 HAZOP 研究指南. Ministry of Defence(UK). March 1995.

适用于计算机控制的过程设备的危险和可操作性(HAZOP)研究. P. Chung and E. Broomfiele. In "Computer Control and Human Error"by T. kletz,Institution of Chemical Engineers,165-189 Railway Terrace,Rugby,CV1 3HQ,UK,1995,ISBN 0-85295-362-3.

化学工业中可编程电子系统的可靠性和危险准则. E. Johnson,Proc. of Safety and Reliability of PES,PES 3 Safety Symposium, B. K. Daniels (ed.), 28-30, May 1986. Guernsey Channel Islands,

Elsevier Applied Science,1986.

HAZOP 和 HAZAN. T. A. Kletz. Institution of Chemical Engineers,165-189,Railway Terrace, Rugby,CV1 3HQ,UK,3rd Edition 1992,ISBN 0-85295-285-6.

HAZOPs 指南. Chemical Industries Association Ltd,1977.

可靠性工程和风险评估. E. J. Henlty and H. Kumamoto,Prentice-Hall,1981.

系统可靠性和风险分析(系统可靠性和风险分析的工程应用). E. G. Frenkel,Kluwer Academic Pub. ,May 1988,ISBN 90-2473-665X.

过程设备的控制危险研究. K. Walters,in Integrated Risk Assessment-Current Practice and New Directions,edited by R. E. Melchers and M. G. Stewart, The University of Newcastle, NSW Australia. A. A. Balkema publishers,Rotterdam Netherlands 1995,ISBN 90-5410-5550.

C.6.3 共同原因失效分析

注 1：在 GB/T 20438.3—2006 的表 A.10 中引用了本技术/措施。

注 2：另见 GB/T 20438.6—2006 的附录 D。

目的：为了确定多系统或者多子系统中潜在的失效。因为在多个部分中可同时出现相同的失效，所以这种失效可能逐渐削弱冗余的作用。

描述：打算用来解决一台设备安全性的系统的硬件中常常使用冗余和多数表决。这可避免部件和子系统中的随机硬件失效，这些失效势必会防碍数据的正确处理。

但有些失效对不止一个部件或子系统是共同的。例如，当在单间房内安装一个系统时，空调的缺点可能会削弱冗余的作用。系统的其他外部影响，比如火、注水、电磁干扰、平台撞击和地震也同样如此。系统也可能受与操作和维护有关的意外事故的影响。因此，为操作和维护，以及操作和维护人员的全面培训提供合用的和编写得很好的操作规程是最根本的。

内部影响也是共同原因失效的一个主要原因。它们能起源于共同的或同样的部件和它们的接口的设计缺陷以及部件的老化。共同原因失效分析必须搜查系统的这些潜在的共同失效。共同原因失效分析的方法是：通常的质量控制；设计复审；由一个独立小组进行验证和测试；根据类似系统反馈的经验分析实际的意外事故。然而分析范围超出了硬件范围。即使在一个冗余系统的各个通道中使用软件多样化，还是有可能在软件方法中存在一些共性，它们将引起共同原因失效，例如共用的规范中的错误。

当不是严格地在同一时间发生共同原因失效时，可以借助多通道之间的比较方法采取预防措施，这种比较方法可以在这种失效成为所有通道共有之前检测该失效。共同原因失效分析应把这种技术包括在内。

参考文献：

共同原因失效的评述. I. A. Watson,UKAEA,Centre for systems Reliability,Wigshaw Lane,WA3 4NE,England,NCSR R27,July 1981.

冗余系统中的共同模式失效. I. A. Watson and G. T. Edwards. Nuclear Technology Vol. 46,December 1979.

在与安全有关的应用中的可编程电子系统. Health and Safety Executive,Her Majesty's Stationary Office,London,1987.

C.6.4 马尔可夫(Markov)模型

注：对于本技术同硬件安全完整性上下文中的可靠性方框图的一个简要比较可参看 GB/T 20438.6—2006 的表 B.1。

目的：为了评价一个系统的可靠性、安全性或者可用性。

描述：构建系统的一幅图。此图表示了有关系统失效状态(由图的节点代表)的系统状态。表示失效事件或修理事件的节点之间的边缘由相应的失效率或修理率加权。假定状态 N 改变成后一状态 $N+1$ 的一次变化和前一状态 $N-1$ 无关。注意，可以在达到精确描述系统的程度上(例如发现的或未

发现的失效、一个较大的失效征兆等)详细说明失效事件、状态和失效率或修理率。

马尔可夫技术适合模型化多个系统,在这些系统中冗余级因部件的失效和修理随时间变化。其他传统的方法(例如 FMEA(失效模式与影响分析和 FTA(故障树分析))因为不存在计算相应概率的组合公式,不能轻松地用来模型化系统整个生命周期内的失效影响。

在最简单的情况中,描述系统概率的公式很容易用于文献或者能用人工计算。在较复杂的情况中存在简化(即减少状态数)的一些方法。对非常复杂的情况,用计算机图形仿真才能计算结果。

参考文献:

IEC 61165:1995 Markov(马尔可夫)技术的应用.

随机过程的理论. R. E. Cox and H. D. Miller, Methuen and Co. Ltd., London, UK, 1963.

有限的马尔可夫链. J. G. Kemeny and J. L. Snell. D. Van Nostrand Company Inc, Princeton, 1959.

可靠性手册. B. A. Koslov and L. A. Usnakov, Holt Rinenhart and Winston Inc, New York, 1970.

可靠性系统设计的理论和实践. D. P. Siewiorek and R. S. Swarz. Digital Press, 1982.

C.6.5 可靠性方框图

注:在 GB/T 20438.3—2006 的表 A.10 中引用了本技术/措施,并且在 GB/T 20438.6—2006 的附录 B 中也使用了本技术/措施。

目的:为了用一种图形形式建造必定发生的一组事件和为成功运行一个系统或一个任务必须满足的一组条件的模型。

描述:分析目标被表示成一条成功的路径,此路径由方框、线和逻辑结点组成。一条成功的路径从图的一侧开始,经过方框连接到图的另一侧。一个方框代表一个条件或一个事件,如果该条件是真实的或者事件已发生,则路径就可通过。当路径到达一个结点时,如满足结点逻辑,它就继续下去。当路径到达一个顶点时,它可沿所有的引出线继续向前。如至少存在一条成功地通过图的路径,则分析目标就处在正确操作之中。

参考文献:

IEC 61078:1991 可靠性分析技术 可靠性方框图方法.

系统可靠性工程方法:技术发展状况的分段. J. B. Fussel and J. S. Arend, Nuclear safety 20 (5), 1979.

故障树手册. W. E. Vesely et al, NUREG-0942, Division of System Safety Office at Nuclear Reactor Regulation, US Nuclear Regulatory Commission, Washington, DC 20555, 1981.

C.6.6 Monte-Carlo(蒙特-卡络)仿真

注:在 GB/T 20438.3—2006 的表 B.4 中引用了本技术/措施。

目的:为了用随机数模拟软件中的真实世界现象。

描述:蒙特-卡洛仿真用来解决两类问题:

——概率性的,使用随机数来产生随机现象的情况;以及

——确定性的,它被从数学上变换成一个等效的概率问题。

蒙特-卡洛仿真注入随机数流来模拟一个分析信号上的噪声或者加上随机偏置值或者容差。执行蒙特-卡洛仿真可产生一个大的样本,根据这个样本可得到统计结果。

当使用蒙特-卡洛仿真时,必须注意保证偏置值、容差或者噪声有一个合理的值。

蒙特-卡洛模拟的一个普遍原理是重申和重新公式化问题使之得到的结果尽可能精确而不是解决最初所说的问题。

参考文献:

蒙特-卡洛方法. J. M. Hammersley, D. C. Handscomb, Chapman & Hall, 1979.

附 录 D
（资料性附录）
确定预开发软件的软件安全完整性的一种概率法

D.1 一般要求

本附录提供了关于使用一种概率法来确定根据操作经验预开发软件的软件安全完整性的原始指南。本方法被认为是特别适合作为操作系统、程序库组成部件、编译器和其他系统软件的认证的组成部分。本附录提供了一个指示：哪些技术是可能的，而只有能胜任统计分析的那些人才能使用这些技术。

注：本附录使用了术语 Confidence level(置信级)，在 IEEE 352 中对它们作了描述。在 IEC 61164 中使用了一个等效的术语：Significance level(显著级)。

本技术还用来证明软件安全完整性等级的超时增加。例如，按 GB/T 20438.3 的 SIL1 要求建立的软件在大量应用中成功地运行一段适当的时间之后，会显示出达到 SIL2。

按 D.2 中概述的那样对运行经验进行数学处理以便补充或者代替统计测试，可以把几个现场得到的运行经验结合起来(即把处理的请求次数或者运行的小时数加起来)，但只是在下列情况下才有可能：

——在电气/电子/可编程电子(E/E/PE)安全相关系统中使用的软件版本是和正在申请的运行经验的那个版本一样的；

——输入空间的操作简表是相似的；

——有一个报告和文档化失效的有效的系统；以及

——满足有关的先决条件(参看 D.2)。

表 D.1 安全完整性等级的置信度的必要历史

SIL	低要求操作模式	处理的要求数		高要求或者连续操作模式	总的运行时数	
	(应要求执行它的设计功能时的失效概率)	$1-\alpha=0.99$	$1-\alpha=0.95$	(每小时一次危险失效的概率)	$1-\alpha=0.99$	$1-\alpha=0.95$
4	$\geqslant 10^{-5}$ 且 $<10^{-4}$	4.6×10^{5}	3×10^{5}	$\geqslant 10^{-9}$ 且 $<10^{-8}$	4.6×10^{9}	3×10^{9}
3	$\geqslant 10^{-4}$ 且 $<10^{-3}$	4.6×10^{4}	3×10^{4}	$\geqslant 10^{-8}$ 且 $<10^{-7}$	4.6×10^{8}	3×10^{8}
2	$\geqslant 10^{-3}$ 且 $<10^{-2}$	4.6×10^{3}	3×10^{3}	$\geqslant 10^{-7}$ 且 $<10^{-6}$	4.6×10^{7}	3×10^{7}
1	$\geqslant 10^{-2}$ 且 $<10^{-1}$	4.6×10^{2}	3×10^{2}	$\geqslant 10^{-6}$ 且 $<10^{-5}$	4.6×10^{6}	3×10^{6}

注 1：$1-\alpha$ 表示置信级。

注 2：关于先决条件和得出本表的详情可参看 D.2.1 和 D.2.3。

D.2 统计测试公式及其应用举例

D.2.1 低要求操作模式的简单统计测试

D.2.1.1 先决条件

a) 测试数据分布等于在线运行过程中要求的分布。

b) 相对于一次失效的原因而言，各测试运行在统计学上是互不相关的。

c) 存在一个检测可能发生的任何失效的适当机构。

d) 测试实例数 $n>100$。

e) 在 n 个测试实例过程中没有发生失效。

D.2.1.2 结果

在置信级 $1-\alpha$ 时，失效概率 P 由下式给出：

$$P \leqslant 1-\sqrt[n]{\alpha} \text{ 或者 } n \geqslant -\frac{\ln\alpha}{P}$$

D.2.1.3 例子

表 D.2 低要求操作模式的失效概率

$1-\alpha$	P
0.95	$3/n$
0.99	$4.6/n$

在置信度为 95%时，对于 SIL3 应要求运行时的失效概率而言，在先决条件下，使用公式得出的测试实例为 30 000 个。表 D.1 汇总了每个安全完整性等级的结果。

D.2.2 测试一个低要求操作模式的输入空间(域)

D.2.2.1 先决条件

唯一的必要条件是选择测试数据从而在输入空间(域)内给出一个随机的均匀分布。

D.2.2.2 结果

目的是找到一个测试次数 n，这个数是根据正在测试的低要求功能(比如一次安全停机)的输入精度阈值 δ 计算出的必需值。

表 D.3 两个测试点的平均距离

域的大小	在一根任意轴向上两个测试点的平均距离
1	$\delta=1/n$
2	$\delta=\sqrt[2]{1/n}$
3	$\delta=\sqrt[3]{1/n}$
k	$\delta=\sqrt[k]{1/n}$
注：k 为任何正整数，值 1、2 和 3 恰好是例子。	

D.2.2.3 例子

考虑一次刚好只与两个变量 A 和 B 有关的安全停机，如果已对划分输入变量对 A 和 B 分区的阈值被正确地处理成 A 和 B 测量范围的 1%这样的一个精度做过验证，在空间 A 和 B 中所需的均匀分布的测试实例数就是：$n=1/\delta^2=10^4$。

D.2.3 高要求或者连续操作模式的简单统计测试

D.2.3.1 先决条件

a) 测试数据分布等于在线运行过程中的分布。

b) 不失效的概率的相对减少正比于考虑的时间间隔的长度，不然就是一个常数。

c) 存在检测可能发生的任何失效的一个合适机构。

d) 测试延续一段测试时间 t。

e) 在 t 内不发生失效。

D.2.3.2 结果

失效概率 λ、置信级 $1-\alpha$ 和测试时间 t 之间的关系是：

$$\lambda = -\frac{\ln\alpha}{t}$$

失效概率反比于两次失效之间的平均工作时间：

$$\lambda = \frac{1}{MTBF}$$

注：GB/T 20438 对每小时的失效概率和 1 小时内的失效率不加区别。严格地说，失效概率 F 和失效率 f 之间的关系为 $F=1-e^{-ft}$，但在 GB/T 20438 的范围内，失效率小于 10^{-5}，在这样小的值时 $F\approx ft$.

D.2.3.3 例子

表 D.4 高要求或者连续操作模式时的失效概率

$1-\alpha$	λ
0.95	$3/t$
0.99	$4.6/t$

为了检定在一个置信级为 95%的情况下，两次失效之间的平均时间至少为 10^8 h，至少需要 3×10^8 h 的测试时间，并且还必须满足先决条件，表 D.1 汇总了每个安全完整性等级所需的测试数。

D.2.4 完全测试

把程序当作是装有已知球数 N 的一个缸。每个球代表程序的一个重要特性。随机地抽取这些球，并在检查后更换另一个。当抽取完所有的球时，也就达到了完全测试。

D.2.4.1 先决条件

a) 测试数据这样分布：在等概率的情况下测试程序 N 个属性的每一个。

b) 测试的各次执行彼此无关。

c) 发生的每个失效都能发现。

d) 测试事例数 $n\gg N$。

e) 在 n 个测试事例期间不发生失效。

f) 每执行一次测试将测试一个程序属性(一个程序属性是在一次执行中可测试的一个属性)。

D.2.4.2 结果

测试所有程序属性的概率 P 由下式给出：

$$P=\sum_{i=0}^{N-1}(-1)^j\binom{N}{j}\left(\frac{N-j}{N}\right)^n \text{ 或者 } P=1+\sum_{i=1}^{N}(-1)^jC_{j,N}\left(\frac{N-j}{N}\right)^n$$

其中：$C_{j,N}=\dfrac{N(N-1)\ldots(N-j+1)}{j!}$。

在评价本公式时，通常只有第一项才重要，因为实际中的实例数 $n\gg N$。这使所有 j 大的那些项都小。在表 D.5 中也可看到这个结果。

D.2.4.3 例子

考虑一个已在几台设备中使用多年的程序。总计至少已执行了 7.5×10^6 次运行。估计有 1/100 的运行满足上述先决条件。所以已完成的 7.5×10^4 次运行能进行统计评价。估计一次全数测试将执行 4 000 次测试运行。这种估计是保守的。根据表 D.5，所有程序属性不被测试的概率等于 2.87×10^{-5}。

当 $N=4\ 000$ 时，与 n 有关的第一项的值见表 D.5。

表 D.5 测试所有程序属性的概率

n	P
5×10^4	$1-1.49\times10^{-2}+1.10\times10^{-4}\_\cdots$
7.5×10^4	$1-2.87\times10^{-5}+4\times10^{-10}\_\cdots$
1×10^5	$1-5.54\times10^{-8}+1.52\times10^{-15}\_\cdots$
2×10^5	$1-7.67\times10^{-19}+2.9\times10^{-37}\_\cdots$

实际上，所作的这种估计是保守的。

D.3 参考文献

在下列文献中可以看到有关上述技术的更多信息：

a) 实时软件的验证和确认，第 5 章. W. J. Quirk(ed). Springer Verlag, 1985, ISBN 3-540-15102-8.

b) 结合概率性的和判定性的验证工作. W. D. Ehrenberger, SAFECOMP 92, Pergamon Press, ISBN 0-08-041893-7.

c) Ingenieurstatistik. Heinhold/Gaede, Oldenburg, 1972, ISBN 3-486-31743-1.

d) IEEE 352:1987 核电站安全相关系统可靠性分析一般原理的 IEEE 指南.

e) IEC 61164:1995 可靠性发展过程 统计测试和估算方法.

参 考 文 献

[1] IEC 60068-1:1988 环境测试 第一部分:概述和指南.

[2] IEC 60529:1989 由机壳提供的保护级(IP 码).

[3] IEC 60812:1985 系统可靠性分析技术 失效模式和影响分析(FMEA)程序.

[4] ICE 60880:1986 核电站安全相关系统计算机软件.

[5] IEC 61000-4-1:1992 电磁兼容性(EMC) 第 4 部分:试验和测量技术 第 1 章:抗扰性试验概述.

[6] IEC 61000-4-5:1995 电磁兼容性(EMC) 第 4 部分:试验和测量技术 第 5 章:浪涌抗扰性试验.

[7] IEC 61000-5-2:1997 电磁兼容性(EMC) 第 5 部分:安装和减弱指南 第 2 章:接地和铺设电缆.

[8] IEC 61025:1990 故障树分析(FTA).

[9] IEC 61069-5:1994 工业过程测量和控制 用于系统评估的系统属性评价 第 5 部分:系统可靠性评估.

[10] IEC 61078:1991 可靠性分析技术 可靠性方框图法.

[11] IEC 61131-3:1993 可编程控制器 第 3 部分:编程语言.

[12] IEC 61160:1992 形式设计评述.修订版 1(1994).

[13] IEC 61163-1:1995 可靠性应力筛选 第 1 部分:批量生产的可修复零件.

[14] IEC 61164:1995 可靠性发展过程 统计试验和估算方法.

[15] IEC 61165:1995 Markov(马尔可夫)技术的应用.

[16] IEC 61346-1:1996 工业系统、设备、装置和工业产品-构造、原则和参考命名 第 1 部分:基本规则.

[17] IEC 61506:1997 工业过程测量和控制 应用软件文档.

[18] IEC 61704 可靠性评估软件测试方法的选择指南.

[19] ISO/IEC 5807:1985 信息处理 数据、程序、系统流程图、程序网络图和系统资源图的文档符号和约定.

[20] ISO/IEC 7185:1990 信息技术 编程语言 Pascal.

[21] ISO/IEC 8631:1989 信息技术 程序结构及其表达式的约定.

[22] ISO/IEC 8652:1995 信息技术 编程语言 Ada.

[23] ISO/IEC 8807:1989 信息处理系统 开放系统互连 LOTOS 基于观察到的行为的时序排序的一种形式描述技术.

[24] ISO/IEC 9899:1990 编程语言 C.

[25] ISO/IEC/TR 10206:1991 信息技术 编程语言 扩展的 Pascal.

[26] ISO/IEC 10514-1:1996 信息技术 编程语言 第 1 部分:Modula-2 基本语言.

[27] ISO/IEC 10514-3:1998 信息技术 编程语言 第 3 部分:面向对象的 Modula-2.

[28] ISO/IEC 13817-1:1996 信息技术 编程语言、编程、环境和系统软件界面 Vienna(维也纳)开发方法 规范语言 第 1 部分:基本语言.

[29] ISO/IEC 14882:1998 编程语言 C++.

[30] ISO/IEC 1539-1:1997 信息技术 编程语言 Fortran 第 1 部分:基本语言.

[31] ISO/IEC/TR 15942 高集成系统中 Ada 编程语言的使用指南.

索　引

GB/T 20438.7-2006

*
书号:155066・1-28713