核电安全级数字化仪控系统内存诊断设计与实现

Design and Implementation of Memory Diagnosis of Nuclear Power Safety Digital I&C System

★北京广利核系统工程有限公司 马忠刚，李萌，张智慧，石桂连，王晓伟

摘要：核电厂数字化仪控系统根据标准要求需要对内存进行诊断。为了解决物理内存地址操作问题，提高内存诊断覆盖率，并提升诊断效率，设计了一种基于实时操作系统的内存诊断方法。采用March C-算法，诊断范围可覆盖全部内存，在不影响操作系统中其他进程执行的情况下，能够快速、安全地检测内存。该方法已经应用在防城港核电厂3#、4#机组的事故后监视系统中，对我国后续核电厂数字化仪控系统的开发具有借鉴和指导意义。

关键词：核电厂；数字化；仪表与控制；内存诊断；March C；堆栈检查

Abstract: The digital I&C system of nuclear power plant needs to diagnose the memory according to the requirements of the standard. In order to solve the problem of physical memory address operation, improve the memory diagnosis coverage, and improve the diagnosis efficiency, a memory diagnosis method based on real-time operating system is designed. Using the March C- algorithm，the diagnosis range can cover all the memory, and the memory can be tested quickly and safely without affecting the operation of other application software in the operating system. This method has been applied to the post-accident monitoring system of Fangchenggang nuclear power plant No.3 and No.4 units, and has reference and guiding significance for the development of my country's subsequent nuclear power plant I&C system.

Key words: Nuclear power plant; Digitization; Instrumentation and control; RAM diagnosis; March C; Stack check

1 引言

根据国际标准IEC 60880[1]对核电厂执行A类功能的软件相关要求，对安全性、可靠性要求很高的核电安全级软件，必须采用针对性的设计来实现对硬件的诊断。在系统处于运行状态时对内存（Random Access Memory，RAM）进行诊断，是提高系统可靠性的有效方式之一。设备应能够周期性地进行内存诊断，当发现故障时，及时将诊断信息传送上报。

针对目前内存诊断覆盖范围不完整，而且诊断效率偏低的问题进行了探讨。采用March C-算法，设计了一种可应用在实时操作系统中的内存诊断方案。该方案能够确保较低的系统资源占用率、较高的检测速度和覆盖率，并且已经在防城港核电厂3#、4#机组工程项目中进行了应用。

2 内存诊断现状

核电领域早期的数字化仪控系统仅能实现简单的算法、收发网络数据和输入输出控制功能。其中的嵌入式软件不使用操作系统[2]，采用顺序执行方式，所有功能都按照固定周期顺序执行，实现的内存诊断相对简单[3]。新一代的数字化仪控系统，例如安全显示系统普遍使用大尺寸屏幕和实时操作系统软件，这种设计能够应对功能和性能的大幅提升，以显示更加丰富的信息和更加复杂的画面，例如模拟流程图和历史趋势曲线等画面，也更加便于操纵员操作，减少人为错误。但是需要解决三个主要的问题：第一个是如何进行物理内存地址操作问题，需要解决虚拟内存与物理内存的映射，同时需要关闭高速缓冲存储器（Cache）；第二个是如何覆盖全部内存的诊断问题，需要实现内存堆栈区的诊断；第三个是内存诊断效率偏低问题，需要实现内存诊断既能够快速检查内存，又能够做到较低系统资源的占用。

3 基于实时操作系统的内存诊断方法

3.1 整体设计

以实际工程应用项目使用的实时操作系统为例。操作系统中内存诊断功能以进程的方式独立运行，该进程优先级很高，默认设置为255，可以防止内存诊断进程阻塞，进程阻塞是指由于内存诊断进程发生某事件而暂时无法继续执行时便放弃CPU资源而处于的暂停状态。进程初始化阶段创建1个管理线程和1个时钟中断服务程序。

内存诊断整体架构图如图1所示。主要分为三个模块：初始化模块、接口模块、诊断运行模块。诊断进程执行的初始化阶段开启内存诊断管理线程和时钟中断服务程序，并初始化对外接口消息队列。

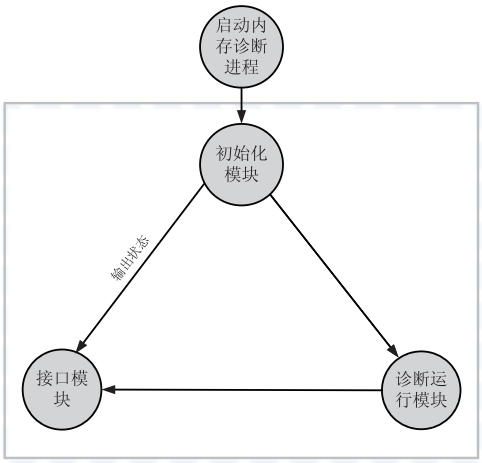

图1 内存诊断架构示意图

初始化模块用于检查内存地址范围有效性、初始化接口信息状态、内存地址映射，将物理内存映射到进程地址空间。等待接口模块消息开启诊断指令，如果开启诊断指令到来，则创建内存诊断管理线程，同时启动时钟中断服务程序。

接口模块与外部软件采用消息机制进行通信，获取内存诊断状态信息，包括：内存诊断状态、错误码、错误内存地址、错误次数计数、数据线诊断计数、地址线诊断计数、内存诊断进度、当前诊断内存地址。

诊断运行模块包含诊断管理线程和时钟中断服务程序。诊断管理线程负责计算内存诊断地址，包括地址线地址和数据地址，将虚拟地址转换为物理地址。时钟中断服务程序产生1ms周期的时钟中断。在中断处理函数中，使用March C-算法对内存进行在线检测，为了确保时钟中断服务程序在1ms内执行完成，默认每次检测内存256字节。检测步骤如下：

（1）时钟中断触发时钟中断服务程序；

（2）判断内存检测间隔时间是否符合预期，例如：100ms进行一次内存检测；

（3）时钟中断服务程序调用中断关闭函数，禁止系统内所有中断，禁止任务运行和接收其他中断；

（4）判断是地址线检测还是数据检测；

（5）使用March C-算法对相应的内存进行分段检测，如果发现错误，记录错误信息并报警，然后继续运行；

（6）时钟中断服务程序调用中断使能函数使能系统内所有中断，正常接收中断，主任务继续运行。等待时钟中断，如果中断再次到来，执行第一步。

通过创建多个内存检测处理函数，可以实现内存检测覆盖完整内存区，在中断处理函数诊断内存前判断当前地址是否在堆栈段恰巧与检测处理函数地址相同，若存在，则跳到另外一个内存检测处理函数运行。

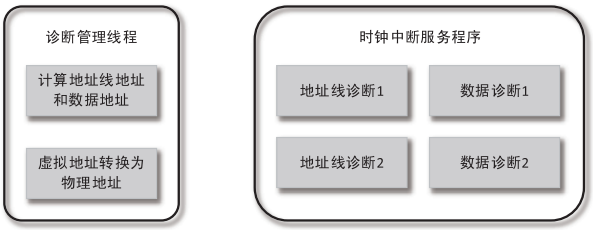

图2 内存诊断模块示意图

3.2 内存诊断算法分析

March算法[4]是存储器测试时常用的一种测试算法，基本原理是对所有的地址逐个进行读写操作。算法的指令比较简单，只有读写0/1和地址变化的指令。通过对存储器不断地读写，用以检测存储器故障。步骤如下：

（1）在整个存储器中写入全部为0的数据背景；

（2）读出第一个存储单元，结果应为0；

（3）将1写入第一个存储单元，再读出第一个单元，结果应为1；

（4）对第二、三、……单元重复步骤（2）和（3），直到最后单元N；

（5）这时候全部单元的内容均为1，即所有单元

以1为背景；

（6）读出第一个存储单元，结果应为1；

（7）把0写入第一个存储单元，再读出这第一个单元，结果应为0；

（8）对第二、三、……单元重复步骤（6）和（7），直到最后一个单元N。

从第（6）步也可以倒序执行，即先从最后一个单元开始一次进行读写直到第一个单元为止。公式表示如图3所示。

$$\sum_{\forall ij}[RN_{ij}, W1N_{ij}]_0$$

$$\sum_{\forall ij}[RN_{ij}, W0N_{ij}]_1$$

图3 March算法表达式

其中，W表示写入操作，R表示读出操作，RN_{ij}表示读出第i行第j列的存储单元N，$W1N_{ij}$表示把1写入第i行第j列的存储单元N，$\forall_{ij}$表示全部N_{ij}的集合，$\sum$表示$\forall_{ij}$集内的总和。中括号外下标0或者1表示成功写入存储单元的数据背景。March算法一方面取得了比较大的故障覆盖率，另一方面，在操作次数的简化上也有较大的优势。

March算法有多种改进版本，例如：March C算法、March C-算法等。各自算法表达式如图4、图5所示。

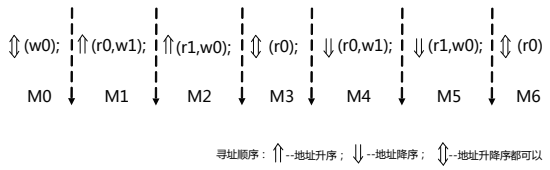

图4 March C算法

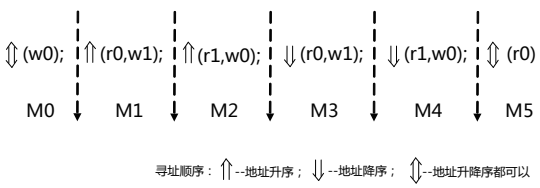

图5 March C-算法

March C在时间操作的复杂度和长度上相对冗长。March C-算法改进了March C算法，去掉了上述第（4）步“读0”，减少这个操作使得原来的11次操作变成10次，而并没有牺牲任何的测试覆盖率，从而降低了复杂度。March C-算法能探测到较多的故障，如固定故障、转换故障、寻址故障和绝大多数的耦合故障，操作复杂度低，故得到了较为广泛的应用。常见单个存储单元故障模型[5]如下：

（1）固定故障（Stuck- at fault，SAF）是指存储器的某个单元或某条线固定为逻辑0或者逻辑1不变。

（2）转换故障（Transition fault，TF）是固定故障中的一个特殊情况，指某个单元或某条线在经过一个写操作以后不能实现0到1或者1到0的转换，分别叫做上升状态转换故障和下降状态转换故障。

（3）耦合故障（Coupling fault，CF）是指多个单元故障，包括倒置耦合故障CFin、固化耦合故障CFid、桥耦合故障BF等。

（4）寻址故障（Address decoder fault，AF）指行或列译码器可能访问不到寻址的单元；或者多个地址访问同一个存储单元；或者一个地址同时访问多个单元；或不访问所指定的单元而访问其他的单元。

3.3 重点解决的问题

3.3.1 物理内存地址操作问题

目前大多数操作系统都使用的虚拟内存映射是计算机系统内存管理的一种技术。它使得应用程序认为它拥有连续的可用内存，即一个连续完整的地址空间。而实际上，它通常是被分隔成多个物理内存碎片，还有部分暂时存储在外部磁盘存储器上，这部分数据只在需要的时候才被交换进物理内存，这种交换是操作系统经由内存管理单元完成的。物理内存和外围存储器之间的数据交换以页（page）为单位进行。

Cache是存在于CPU和内存之间的存储体，用于弥补内存速度的不足。从CPU看来，整个存储体系的速度接近Cache，而容量却接近内存。CPU在取数据时，先从Cache中寻找数据，如果要找的数据存在于Cache中，就称为命中，否则称为不命中。命中的次数与总访问次数的比称为命中率。此时CPU把数据从Cache中取出进行处理。如果没有在Cache中找到需要的数据，就直接到内存中查找，然后对找到的数据进行处理，同时把这些数据回写到Cache中，以备下次使用时直接从Cache中取数据。因为Cache的存取速度比内存的存取速度高得多，所以提高了存储数据的速度。如果CPU从Cache中查找数据的命中次数提高，也就是命中率高，就可大大地提高计算机的存储速度，从而提高计算的性能。

为了能够直接诊断物理内存，而不是虚拟内存，可以调用操作系统提供的物理内存映射接口函数，获取到物理内存地址，并通过内核I/O接口函数访问物理内存地址。另外，调用操作系统提供的关闭Cache接口函数，以避免物理内存读写过程中一直访问的是Cache中的数据。

3.3.2 内存检查覆盖率问题

进程运行时用到的堆栈内存空间会随着程序执行而变化。其中栈空间为操作系统自动分配内存空间，其中数据的大小在编译时确定，数据的分配和释放也由编译器在函数进入和退出时插入指令完成，存储在栈的数据包括本地变量、返回地址、函数参数、编译器临时空间、中断环境等。堆空间由程序员手动分配内存空间，其中数据的大小和初始值在运行时确定。

为了避免当前内存诊断进程执行的诊断函数入栈地址恰巧与当前诊断的内存地址重合，导致函数执行内存诊断的同时改写了栈空间内容，从而造成诊断函数执行异常，直接导致进程崩溃的情况，地址线诊断函数和内存数据诊断函数分别有2个完全一样的副本，例如：AddressLine1()、AddressLine2()和DataTest1()、DataTest2()。程序执行流程如下：

（1）备份AddressLine1()或DataTest1()函数入栈地址；

（2）执行AddressLine1()或DataTest1()前判断当前内存诊断地址与函数入栈地址是否重合；

（3）如果地址不同则继续执行AddressLine1()或DataTest1()函数；

（4）如果地址相同，则执行AddressLine2()或DataTest2()函数。

上述方法可以避免诊断内存数据地址与数据检测汇编程序段地址相同。每次内存诊断完成后保存此次诊断函数在栈内地址，在下一次内存诊断之前判断当前内存诊断地址恰巧与该函数所在栈内地址是否重合。一旦重合则跳过此诊断函数，执行另外一个诊断函数副本。内存检查覆盖率可以达到100%。

3.3.3 内存诊断效率偏低问题

在2.2内存诊断算法分析章节提到的March C和March C-算法是针对以位为基准的存储器的，但目前的存储器多是以字为基准的。为了减少测试时间，在测试中采用以字为基准来进行测试，但组成一个字的多个位都有可能发生耦合故障或者其他类型的故障，即字内故障。面对这种情况就必须增加测试数据背景的组数来弥补这个不足，对于N位的存储器，它的备用测试数据背景一共有$\log_2 N+1$组。以字为基准的March C-算法，极大提升了诊断效率。本设计中采用的是256×8的存储器模型，那么就需要4组数据背景[6]，如表1所示。

表1 以字为基准的数据背景

序号	DB	DBbitnot
1	00000000	11111111
2	01010101	10101010
3	00110011	11001100
4	00001111	11110000

内存诊断进程采用很高的优先级以避免被其他进程阻塞。在时钟中断服务程序中执行内存诊断，采用以字为基准的March C-算法，1ms内能够诊断256字节内存，可以很好地利用CPU空闲片段时间，有限度地占用CPU资源，从而提高了系统效率。

4 测试与验证

内存诊断测试硬件配置使用工控机，采用i7处理器，CPU主频为1.7GHz，内存2GB。

为了验证上述方法的可行性，时钟中断服务程序每次检测内存大小为256字节，设置诊断间隔时间分别为10ms、40ms、60ms、100ms，用于测试CPU资源的使用率。编写多个独立测试进程，分别执行内存读写、共享内存的读写、消息队列传输、创建网络连接传输数据、创建串口连接传输数据、文件创建读写操作等。这些测试进程与内存诊断进程同时运行，用于验证内存诊断是否影响这些测试进程的运行。

表2给出了不同诊断频率下CPU负荷和诊断2GB内存所需时间。

表2 内存诊断性能对比

序号	间隔时间	CPU负荷	诊断2GB内存耗费时间
1	10ms	7%	24h
2	40ms	4%	94h
3	60ms	3%	140h
4	100ms	1%以下	232h

根据表2提供的测试数据和长期运行观察，该内存诊断方法能够稳定运行，长时间监测系统资源占用率很低，具有较高检测速度，并可以诊断全部内存。完全满

足工程项目的应用需求。

5 结束语

本文介绍了基于实时操作系统的核电仪控内存诊断方法的设计、实现和验证情况。该方法解决了操作系统中物理内存隔离保护、内存诊断覆盖范围不完整和诊断效率偏低的问题，尤其是采用了以字为基准的March C-算法，可以有效降低系统资源使用率（CPU使用率7%以下）。目前已经应用在防城港核电厂3#、4#机组的反应堆事故后监视系统中[7]，取得了良好的效果，具备较高的推广价值。AP

作者简介：

马忠刚（1979-），男，河北三河人，学士，工程师，现就职于北京广利核系统工程有限公司，主要从事核安全级数字化仪控系统研究工作。

参考文献：

[1] IEC 60880-2006. Nuclear power plants – Instrumentation and Control Systems important to safety – Software aspects for Computer - Based Systems performing category A functions[S].

[2] 杜乔瑞, 齐敏, 石桂连, 李萌. 核安全级仪控系统 (FirmSys) 平台软件设计技术研究与实现[J]. 自动化博览, 2018, 36 (5) : 66 - 71.

[3] 李萌, 王晓伟, 马忠刚, 等. 核电厂安全级控制显示装置的软件设计[J]. 自动化博览, 2017, 35 (8) : 68 - 73.

[4] 刘振田, 李维波, 林城美, 等. 一种改进的随机存储器自检测 March算法[J]. 船电技术, 2014, 34 (12) : 53 - 55.

[5] 徐军, 王澜, 耿进龙, 等. 一种提高安全计算机可靠性的内存检测设计[J]. 铁路计算机应用, 2014, 23 (10) : 52 - 57.

[6] 孙华义, 郑学仁, 闾晓晨, 等. 嵌入式存储器内建自测试的一种新型应用[J]. 中国集成电路, 2007, (11) : 82 - 87.

[7] 李萌, 李亮, 石桂连, 马忠刚, 王晓伟. 核电厂数字化大尺寸安全显示系统设计与应用[J]. 自动化仪表, 2020, 41 (11) : 58 - 61.