一种星载计算机操作系统容错引导算法研究

辛 宁[1]，邱乐德[1]，张立华[2]，张宏飞[3]
（1. 中国空间技术研究院通信卫星事业部，北京 100094；2. 航天东方红卫星有限公司，北京 100094；
3. 陆航驻北京地区军代室，北京 100176）

摘 要：将软件冗余备份与SPARC V8构架芯片EDAC相结合，提出了一种新型星载计算机操作系统容错引导算法。该算法首先在EEPROM中备份三份操作系统文件，星载机上电或复位后，利用硬件EDAC功能对引导的操作系统文件进行错误检测及纠错，当检测主操作系统文件错误并且无法纠错时，则屏蔽错误操作系统文件自动引导备份操作系统文件，星载机仍可正常启动。与目前采用的硬件编码容错技术及软件冗余容错算法相比，该算法在有效降低硬件成本和软件消耗的同时，实现了对操作系统文件的检错纠错及错误屏蔽功能，提高了操作系统数据的可靠性。该算法可以为星载机容错设计提供参考。

关键词：星载计算机；操作系统；SPARC V8微处理器；EDAC；容错引导
中图分类号：TN432 **文献标识码**：A **文章编号**：1000-1328(2013)06-0818-06
DOI：10.3873/j.issn.1000-1328.2013.06.011

A Method for Fault-Tolerant Bootloading Osfile for On-Board Computer

XIN Ning[1], QIU Le-de[1], ZHANG Li-hua[2], ZHANG Hong-fei[3]
(1. Institute of Telecommunication Satellite CAST, Beijing 100094, China; 2. DFH Satellite CO. LTD. Beijing 100094, China;
3. Representative office of Army Aviation in Beijing, Beijing 100176, China)

Abstract: A new fault-tolerant osfile bootloading method for on-board computer (OBC) combined with software redundancy backup and EDAC of SPARC V8 MPU is proposed in this paper. First, three osfiles are stored in EEPROM. Then the data of default osfile is detected whether errors occur when the OBC is powered on or reset. If the error occurred can be corrected by EDAC, the OBC can startup normally. Otherwise, the error osfile can be shielded and the backup osfile can be bootloaded automatically, so the OBC can also startup. Compared to the hardware coding(encoding) method and software redundancy algorithms, the function of error detection and correction and error shielding for osfile is realized while the hardware cost is decreased, and the reliability of osfile is improved. This algorithm could play a role of reference in OBC fault-tolerant design.

Key words: On-board computer; Osfile; SPARC V8 MPU; EDAC; Fault-tolerant bootloading

0 引 言

SPARC V8(Scalable Processor ARChitecture，可扩充处理器结构）架构微处理器系列芯片是一种RISC类型的CPU指令集体系结构[1]，因其优异的可扩展性，在32位处理器中占据重要地位，广泛应用于航天领域的星载计算机系统中[2]。

星载计算机软件运行在星载计算机上，是星载计算机的灵魂。星载软件分为两部分，存储在EPROM中的操作系统引导软件和存储在EEPROM的操作系统。在通常情况下，EEPROM中只存储一份操作系统文件。当星载计算机加电或复位后，软件启动流程是先运行EPROM中的引导软件，CPU将EEPROM中的操作系统数据引导到SRAM中，最后整个操作系统在SRAM中运行。在空间辐射环境中，星载计算机内部的EEPROM将受到高能宇宙

收稿日期：2012-06-16； 修回日期：2012-11-29
基金项目：国家自然科学基金(41274041)

射线的严重影响[3],导致存储器内部逻辑状态可能发生改变。这种间歇性的瞬态故障通常指的是单粒子翻转(Single Event Upset, SEU),例如从逻辑 0 变为逻辑 1,或者反之。SEU 可导致 EEPROM 中操作系统文件数据损坏、存储器硬件临时故障[4]等现象,外界也无法及时掌握操作系统文件的最新状态。在这种情况下,如果引导软件没有容错引导的功能,直接引导错误的操作系统文件,则直接会导致星载计算机无法正常启动,同时地面测控站也不能及时掌握操作系统工作的最新状态进行故障的定位。

目前星载计算机的容错设计技术主要包括硬件和软件技术,硬件技术通常采用硬件冗余编码技术[5],目前存储器结构使用了 ECC 编码[6]或奇偶校验码[7],其中的冗余位用来定位甚至恢复错误。但硬件容错需要增加特定形式的硬件逻辑及满足冗余计算的需求,但是很多时候成本、功耗等原因限制了硬件容错的使用范围。软件技术具有操作灵活,不需另附硬件的特点[8],目前主要为各种信息冗余技术,如 CRC 实验室的 EDDI 方法[9]、三取二比对算法[10]等。EDDI 方法是在不违背时间约束的前提下,将操作系统文件并行读取多次并比较结果,相同则说明操作系统文件正确,否则文件肯定出现了错误。但是 EDDI 算法强调检错功能,无法进行纠错。三取二比对算法需要将 EEPROM 中存储的三份文件全部读取,逐一比对更新后才能进行引导,开销较大。其次,三取二比对算法虽然具有错误纠正的能力,但当操作系统文件中有两份文件同时出现错误时,则无法引导正确的操作系统,不具备错误屏蔽的能力。已有研究表明软件和硬件相结合是容错技术的一个合理解决方案[11]。基于上述思想,本文提出了一种利用 SPARC V8 芯片 EDAC 功能与操作系统冗余备份的容错引导算法,该算法在有效降低硬件成本和软件开销的同时,实现了对操作系统文件的纠错检错及错误屏蔽功能,提高了操作系统数据的可靠性。

1　容错引导算法原理

容错引导软件采用 SPARC V8 汇编语言编程[12],存储在 EPROM 中。宇航级 EPROM 对于单粒子效应具有免疫性。因此,只要容错引导软件在地面测试充分,即可保证在太空环境中的可靠性、准确性。容错引导算法首先采用软件冗余在 EEPROM 中存储三份操作系统文件,设备上电后,引导程序利用硬件 EDAC 纠错检错功能对需引导的操作系统数据进行校验,当检测操作系统数据错误时,如果错误可以纠错,则进行数据纠错;如果无法纠错,则触发软件陷阱自动屏蔽发生错误的操作系统,引导备份的操作系统到 SRAM 中,星载机仍正常启动。如果算法引导操作系统失败,可通过 CAN 总线输出异常信息报文定位故障。

2　容错引导算法设计流程

容错引导算法总流程如图 1 所示。算法设计流程为设备加电/看门狗复位;初始化硬件基本资源;使能看门狗计数器;打开 CACHE;使能软件陷阱;获取当前看门狗咬计数值,根据狗咬计数值选择引导模式(狗咬计数值 $N < 2$)还是异常模式(狗咬计数值 $N \geqslant 2$)。异常模式为通过 CAN 总线输出提示信息,说明当前软件的状态;容错引导模式为利用 SPARC V8 CPU 提供的 EDAC 功能,来检测操作系统文件是否正确,若正确,则引导操作系统;若不正确,则跳转到异常模式,输出异常报文,地面可根据异常报文进行切机或重新上注程序操作。

3　容错引导算法模块设计

由图 1 可知,容错引导算法主要分为三个模块,硬件初始化模块、容错引导模块、异常模块。

3.1　硬件初始化模块设计

星载计算机加电/看门狗复位后,进行一系列的初始化工作。即通过硬件资源基本初始化后,设备能执行简单的操作。由于 SPARC V8 CPU 的特点,初始化有严格的执行顺序。

软件的初始化主要包括:建立中断向量表并对 256 个陷阱进行相应的处理措施;初始化 CPU 内部寄存器% ASR16、% ASR17、% PSR、% WIM、% TBR 和% FSR;打开 CACHE;打开存储器的 EDAC;使能陷阱;使能 CPU 外部的 I/O;初始化 2 路 CAN 总线;使能看门狗计数器等。在初始化的过程中为了提高效率,数据存、取操作使用 CPU 内部寄存器% g0、% g1、% g2 和% g3,详细内容见参考文献[13 - 15]。

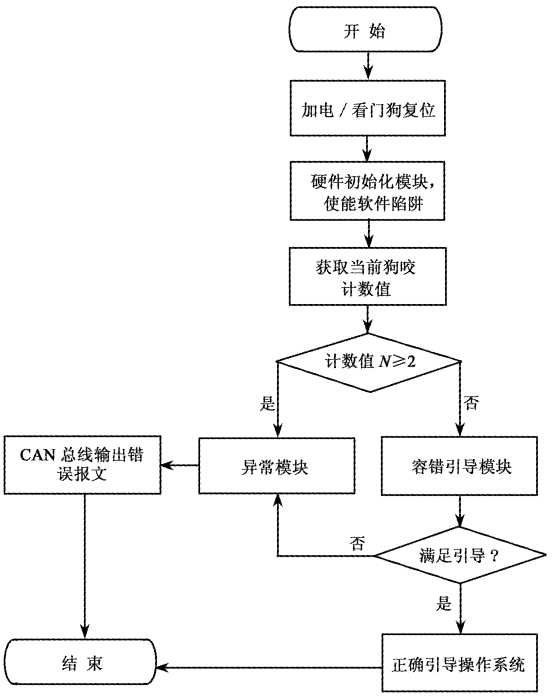

图 1 容错引导软件总流程图

Fig. 1 Generic framework of fault-tolerant bootloading algorithm

3.2 容错引导模块设计

容错引导模块设计的首要问题是选择 EDAC 的纠错编码方式，常用的纠检错码有汉明码、RS 编码等，汉明码可以纠正一个字中的任何一位错误，检测出两位错误，但是对于多于两位的错误不具有任何的纠错和检错能力。RS 编码可以发现并纠正一个码字中的多位错误，但是其相应的编译码电路和延迟开销也更大。目前星载计算机中采用的宇航级 EEPROM 芯片中一个码字中多位出错发生的概率极低[16]。因此，在纠错能力和现实的开销的折中下，汉明码更适合于空间环境中 EEPROM 的 EDAC 保护。

当看门狗狗咬计数值 $N < 2$ 时，执行容错引导模块。该模块主要完成操作系统启动之前的检测工作，利用 CPU 的 EDAC 汉明码纠正一位错检测两位错的特点，检测操作系统文件是否正常。若正常，则引导操作系统；若不正常，则执行异常模块，关于异常模块详见 3.3 。

容错引导模块设计思路：在 EEPROM 中存储 3 份操作系统文件（即 OS1、OS2、OS3）的条件下，使用 CPU 自身的 EDAC 特点，首先对第一块操作系统文件（OS1）数据，以 4 字节为单位进行刷新，在刷新过程中，若出现可以纠正的数据错误时，将纠正后的数据进行加载。若一直没有出现 Trap = 0x09 的陷阱类型，说明没有检测到不可纠正的数据错误，则设置 SRAM 的启动标志并正确引导 OS1；若出现过 Trap = 0x09 的陷阱类型，则停止后续的数据刷新。然后，用同样的方法去刷新 OS2 及 OS3。假如，到最后 3 块操作系统都出现过 Trap = 0x09 的陷阱类型，那么说明操作系统文件已全部损坏，最后执行异常模块的软件。3 块操作系统的刷新优先级是 OS1 > OS2 > OS3，若 OS1 没有出现 Trap = 0x09，则直接引导 OS1 并设置启动标志，只有在 OS1 出现过 Trap = 0x09 的情况下，OS2 才有数据刷新的机会，同理，OS3 也是一样。

为可靠性考虑，操作系统的启动标志存放于 SRAM 的 4 个地址当中。在软件的设计过程中，对于 Trap = 0x09 的中断服务程序，程序计数器 PC 使用寄存器% l2，中断服务程序标志存放于寄存器% g7[13-15]。

容错引导模块流程如图 2 所示。

3.3 异常模块设计

本模块主要是通过 2 路 CAN 总线输出星载计算机的异常信息，信息的输出格式见表 1。表 1 中 ID10-ID3 为仲裁场寄存器，ID2-ID0、RTR 及 DLC 为控制场寄存器。异常模块软件设计流程：当看门狗狗咬计数 $N \geqslant 2$ 或者三块操作系统文件均出现 EDAC 两位错时，才有异常信息输出。异常信息输出的顺序为首先是第一路总线输出，等待大约 2s 后，第二路总线输出，然后执行 CPU 死循环指令。

4 试验验证

通过前面的分析及软件设计，在实验室里对嵌入式设备平台进行了软件测试。

4.1 硬件基本配置

容错引导算法的嵌入式硬件基本配置：CPU 采用国产的 SPARC V8 构架的 BM3803MG，时钟主频 100MHz，该款 CPU 是目前唯一经过在轨搭载验证的国产 CPU；外部接口有 2 路串口、1 路 DSU 口和 2 路 CAN 总线；三种存储器（包括 EPROM、EEPROM、SRAM）。设备上电或热复位后，软件启动流程是先

运行 EPROM 中容错引导软件,然后运行 EEPROM 的操作系统,最后整个软件在 SRAM 中运行。EPROM 存放容错引导软件,是 8 位操作;EEPROM 存放操作系统,是 32 位操作;SRAM 存放系统运行时的数据,是 32 位操作。根据强实时性、高可靠性、容错性要求,操作系统采用 VxWorks5.5 并且在 EEPROM 内存放 3 份,编译环境为 Tonado2.0[17]。

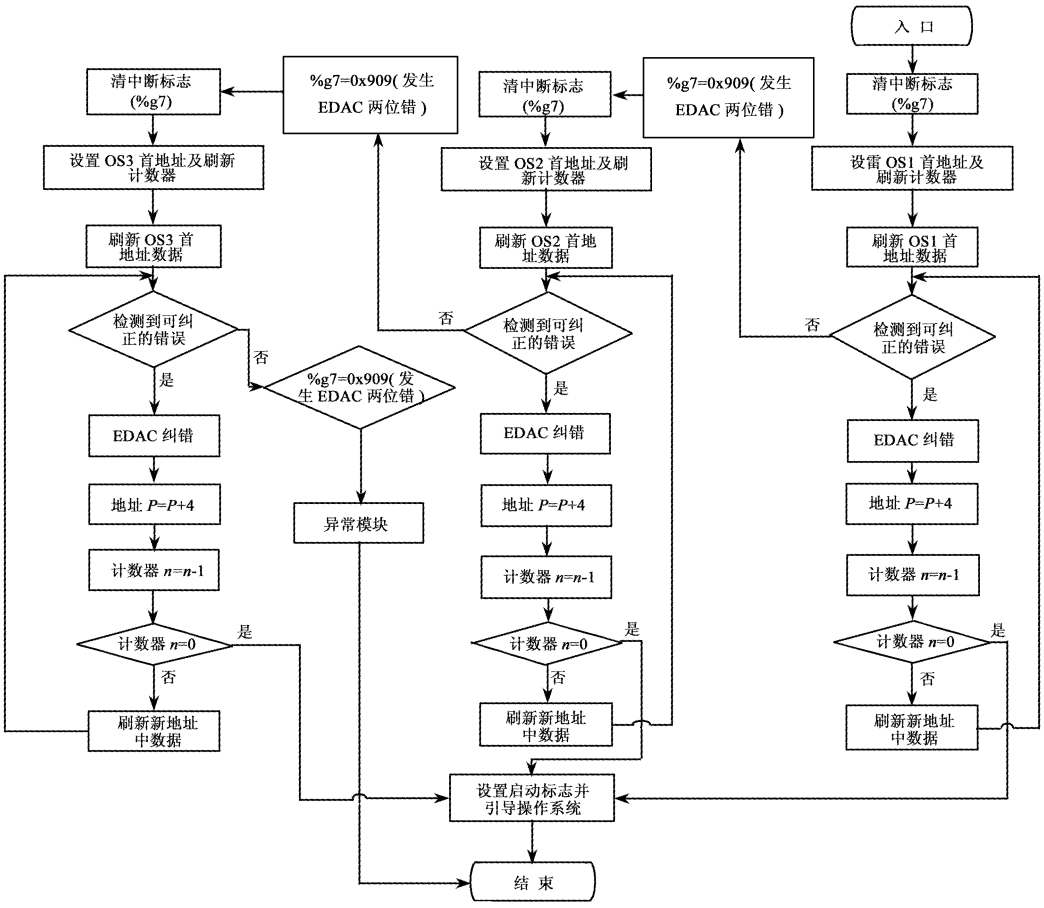

图 2　容错引导模块流程图

Fig. 2　Fault-tolerant bootloading block processing block

表 1　CAN 总线信息输出格式表

Table 1　CAN frame format

异常模式	总线标识	仲裁场	控制场	有效数据场							
		ID10-ID3	ID2-ID0 RTR DLC	data1	data2	data3	data4	data5	data6	data7	data8
看门狗狗咬计数 $N \geqslant 2$	CAN A	02H	08H	02H	02H	01H	01H	AAH	01H	AAH	01H
	CAN B	02H	08H	02H	02H	02H	02H	BBH	02H	BBH	02H
3 块操作系统文件都有 EDAC 两位错误	CAN A	03H	08H	03H	03H	01H	01H	AAH	01H	AAH	01H
	CAN B	03H	08H	03H	03H	02H	02H	BBH	02H	BBH	02H

4.2　验证步骤及结果

(1) 通过 DSU 口,在线注入正确的 OS1、OS2、OS3;

(2) 设备加电,查看操作系统是否启动并获取 SRAM 中的启动标志是否为 OS1 标志;

(3) 设备断电,然后加电。进入调试模式,利用 BM3803 提供的数据位造错功能对 OS1 人为造 1 位错。造错的顺序为首先关闭造错的校验位、设置哪

几位造错、打开造错的校验位、对造错位置写入正确数据、清存储器容错配置寄存器1、清存储器容错配置寄存器2、清存储器容错配置寄存器3；

（4）设备加电，进入正常启动模式。查看操作系统是否启动并获取SRAM中的启动标志是否为OS1标志；

（5）设备断电，然后加电。进入调试模式，对OS1人为造2位错。造错的顺序与第3步一样，只是造错的数量变化；

（6）设备加电，进入正常启动模式。查看操作系统是否启动并获取SRAM中的启动标志是否为OS2标志；

（7）设备断电，然后加电。进入调试模式，对OS2人为造1位错。造错的顺序与第3步一样，只是造错的位置变化；

（8）设备加电，进入正常启动模式。查看操作系统是否启动并获取SRAM中的启动标志是否为OS2标志；

（9）设备断电，然后加电。进入调试模式，对OS2人为造2位错。造错的顺序与第3步一样，只是造错的位置和数量变化；

（10）设备加电，进入正常启动模式。查看操作系统是否启动并获取SRAM中的启动标志是否为OS3标志；

（11）设备断电，然后加电。进入调试模式，对OS3人为造1位错。造错的顺序与第3步一样，只是造错的位置变化；

（12）设备加电，进入正常启动模式。查看操作系统是否启动并获取SRAM中的启动标志是否为OS3标志；

（13）设备断电，然后加电。进入调试模式，对OS3人为造2位错。造错的顺序与第3步一样，只是造错的位置和数量变化；

（14）设备加电，进入正常启动模式。查看操作系统是否启动，若不能启动，再查看2路CAN总线是否有数据输出，同时输出数据是否满足表1的格式及数据。

（15）在实验过程中，还针对看门狗计数器溢出的情况进行了测试。测试手段为设备加电后，进入正常启动模式，应用层软件一直不清看门狗计数器，在等待大约5s后，会发现2路CAN总线有数据输出且输出模式满足表1数据格式。

根据以上步骤的测试，文章所设计的软件算法是可行的，验证了EDAC检错纠错及错误操作系统屏蔽功能的正确性，实现了三备份操作系统容错引导的功能。

5 结 论

本文从软硬件联合容错角度提出了一种操作系统容错引导新算法，在有效降低硬件成本和软件开销的同时，实现了对EEPROM中操作系统文件的纠错检错及错误屏蔽的功能。当主份操作系统出现错误时，可以从备份操作系统启动，而不影响设备的正常运行，除非所有的EEPROM全部损坏。为了实时性方面考虑，算法设计采用SPARC V8汇编语言实现，对于参数传递，数据存、取操作等，采用CPU内部寄存器。当出现星载计算机未能启动的情况时，星载机可通过CAN总线输出故障报文，供研发人员故障分析使用。后续工作主要考虑当EEPROM全部异常的情况下，如何把EEPROM中的工作搬移到EPROM中，保证设备也能正常运行。文章的研究结果对后序航天型号设计有较高的参考价值。

参 考 文 献

[1] Gaisler J. A portable and fault-tolerant microprocessor based on the SPARC V8 architecture[C]. International Conference On Dependable Systems and Networks, Washington DC, USA, April 9-15, 2002.

[2] 施蕾，刘波，周凯. 基于SPARC V8结构处理器的计算机系统设计[J]. 空间控制技术与应用，2008，34(3)：46-50.[Shi Lei, Liu Bo, Zhou Kai. The computer system design based on the SPARC architecture processor[J]. Aerospace Control and Application, 2008, 34(3): 46-50.]

[3] Baumann R C. Radiation-induced soft errors in advanced semiconductor technologies[J]. IEEE Trans. on Device and Materials Reliability, 2004, 5(3): 305-316.

[4] 张小林，杨根庆，李华旺. 星载计算机可靠性和低功耗的均衡优化研究[J]. 宇航学报，2009. 30(5)：1992-1997.[Zhang Xiao-lin, Yang Gen-hua, Li Hua-wang. Reliable and low power simultaneous optimization for on-board computer[J]. Journal of Astronautics, 2009, 30(5): 1992-1997.]

[5] 傅忠传，陈红松，崔刚. 处理器容错技术研究与展望[J]. 计算机研究与发展，2007. 44(1)：154-160.[Fu Zhong-chuan, Chen Hong-song, Cui Gang. Processor fault-tolerance technology research and prospect[J]. Journal of Computer Research and

Development, 2007,44(1):154 - 160.]

[6] Weaver C, Emer J, Mukherjee S S. Techniques to reduce the soft error rate of a high-performance microprocessor[C]. The 31st Ann International Symp on Computer Architecture, USA, June 28 - 30, 2004.

[7] Reinhardt S K, Mukherjee S S. Transient fault detection via simultaneous multithreading[C]. The 27st Ann International Symp on Computer Architecture, USA, June 3 - 5, 2000.

[8] 徐建军,谭庆平,熊庆平. 面向瞬态故障的软件容错技术[J]. 计算机工程与科学,2011. 33(11):132 - 139. [Xu Jan-jun, Tan Qing-ping, Xiong Qing-ping. Software fault-tolerance techniques for transient faults[J]. Computer Engineering and Science, 2011. 33(11):132 - 139.]

[9] Oh N, Shirvani P P, McCluskey E J. Control-flow checking by software signatures[J]. IEEE Trans. on Reliability, 2002,51(1):111 - 122.

[10] 李爱国,洪炳,王司. 一种星载计算机数据流软故障纠正算法[J]. 宇航学报,2007,28(4):284 - 288. [Li Ai-guo, Hong Bing, Wang Si. A software based method for soft error correction in space computer[J]. Journal of Astronautics, 2007,28(4):284 - 288.]

[11] Hu J, Li F, Degalahal V, et al. Compiler-assisted soft error detection under performance and energy constraints in embedded system[J]. ACM Trans on Embedded Computing Systems,2009, 8(4):27 - 56.

[12] SPARC Assembly Language Reference Manual[M]. May 2002.

[13] Atmel Corporation. Rad-hard 32-bit SPARC embedded processor user's manual[M]. Nantes: Atmel Nantes S. A, 2001.

[14] Atmel Corporation. Rad-hard 32 bit SPARC V8 processor preliminary information[M]. San Jose: Atmel Corporation, 2009.

[15] 北京微电子技术研究所. BM3803MG 32 位空间处理器用户手册,Ver2. 4. 2[M]. 北京:北京微电子技术研究所,2010. [Beijing Institute of Microelectronics Technology. User manual of BM3803MG 32-bit space processor Ver2. 4. 2[M]. Beijng: Institute of Microelectronics Technology, 2010.]

[16] 丁义刚. 空间辐射环境单粒子效应研究[J]. 航天器环境工程,2007,24(5):283 - 290. [Ding Yi-gang. Single event effects in space radiation environment[J]. Spacecraft Environment Engineering, 2007,24(5):283 - 290.]

[17] 周红波. 嵌入式系统软件开发环境中调试器的设计[J]. 微计算机信息,2006,22(5):61 - 62. [Zhou Hong-bo. The design of debugger in software development environment of embedded system[J]. Control & Automation, 2006, 22(5):61 - 62.]

作者简介:

辛宁(1982 -),男,中国空间技术研究院博士生,主要从事嵌入式操作系统研究。

通信地址:北京市海淀区友谊路 102 号(100094)

电话:(010)68744245 - 239

E-mail:xinning7@sina.com

(编辑:余　未)