

星载通讯软件的可靠性设计方法

代 霜, 王 槐, 徐抒岩

(中国科学院 长春光学精密机械与物理研究所, 长春 130033)

摘要: 为避免空间环境尤其是单粒子翻转效应 (SEU) 对星载通讯软件的危害性影响, 建立星载通讯软件通用的失效模式分析模型, 并针对典型的失效模式如中断无法响应、帧格式错误等, 给出与一般通讯软件不同的高可靠性软件设计流程, 在设计方法上采取寄存器定时更新、存储区表决法以及冗余设计等可靠性措施避免失效, 通过对固定位置的存储单元模拟 SEU 故障注入的方式进行验证; 实验表明此设计降低了单粒子翻转的危害性, 提高了星载通讯软件的可靠性, 具有普遍意义。

关键词: 星载通讯软件; 单粒子翻转; 失效; 可靠性

Spaceborne Communication Software Reliability Design

Dai Shuang, Wang Huai, Xu Shuyan

(Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China)

Abstract: Firstly, the failure model of Spaceborn communication software is given for space environment, especially SEU (Single Event Upset), unlike common communication software, then the design flow for spaceborn software is set, and some measures are taken such as register renewed periodically, memory voting and redundancy design. Finally fault injection is applied to validate. Practical results show that the design is necessary and effective.

Key Words: spaceborne communication software; SEU; fault tree; reliability

0 引言

星载通讯软件是实时嵌入式软件, 负责接收星上系统的外部控制数据, 并下行表征系统内部重要运行状态的数据, 是星上系统与外界沟通的桥梁, 为保证星上系统顺利完成任务, 必须保证星载通讯软件的准确可靠。星载通讯软件最重要的特点是受空间环境辐射效应的影响^[1]。就目前的认识而言, 影响较大的辐射效应见表 1, 主要有: 总剂量效应、单粒子效应、航天器表面充放电效应、高能电子内带电效应、太阳电池等离子体带电效应等。其中对星载软件造成直接影响的是单粒子效应中的单粒子翻转 (SEU-Single Event Upset) 事件。

表 1 主要辐射效应的对象

空间辐射效应	产生效应的主要对象
电离总剂量效应	几乎所有电子器件及材料
位移总剂量效应	太阳电池、光电器件
单粒子翻转(SEU)	逻辑器件、单/双稳器件
单粒子锁定(SEL)	CMOS 器件
单粒子烧毁(SEB)	功率 MOSFET
卫星表面充/放电	卫星表面包覆材料、涂层
内带电	卫星内部介质材料、器件
太阳电池的等离子体充电	太阳电池

单粒子翻转可引起器件电性能状态的改变, 造成逻辑器件或电路的逻辑错误, 例如: 存储器中数据发生翻转, 进而引起逻辑功能混乱, 计算机程序“跑飞”, 甚至造成灾难性的后果, 对于通讯软件, 可使通讯接口芯片的存储器或寄存器的 0, 1

状态发生翻转, 从而使系统通讯逻辑混乱, 进而使整个系统无法与外界通讯, 无法正常执行任务。

本文主要对空间环境辐射效应中的单粒子翻转从软件角度采取相应的可靠性措施, 保证星载通讯软件可靠性。

1 星载通讯软件失效模式分析

与一般通讯软件相同, 星载通讯软件功能主要有通讯初始化功能、接收功能、发送功能及中断处理等功能。但是由于星载通讯软件会受空间环境的影响, 会引起数据流或控制流的错误, 当发生单粒子翻转而使存储器数据发生 0 或 1 的跳变时, 导致帧格式错误或关键变量错误, 从而使读写过程中传递错误信息, 进而使接收或发送的数据失效。另一方面发生单粒子翻转而使寄存器和存储区初始化遭到修改, 则会使中断无法响应, 也会产生读写数据失效。针对上述情况, 本文进行了与一般通讯软件不同的失效模式分析, 建立了如图 1 所示的星载通讯软件故障树。从图 1 中可以看出, 导致通讯软件失效的中间事件为接收或发送数据失败和读写过程中传递错误信息。子事件有初始化寄存器错误、中断无法响应、帧格式错误、关键变量错误等。

上述典型的失效模式, 具体在下边的设计通讯软件的过程中加以相应的可靠性设计措施。

2 星载通讯软件的组成

2.1 星载通讯软件拓扑结构

参考 ISO 网络七层模型, 与一般通讯软件相似, 将星载通讯软件的总线通讯分为 4 层: 物理层、数据链路层、驱动层和应用层, 如图 2 所示。物理层按照特定总线协议规定的编码格式进行传输。数据链路层按照特定总线协议规定的数据格式进行传输。总线 MBI 包括 MBI 底层驱动程序和 MBI 通讯程序。一般总线通讯有专用的协议芯片支持。MBI 底层驱动程序实现 MBI 初始化、协议芯片的中断、RAM 地址、寄存器地

收稿日期: 2010-09-02; 修回日期: 2010-10-15。
作者简介: 代 霜(1982-), 女, 吉林省东丰县人, 硕士, 主要从事嵌入式系统设计方向的研究。

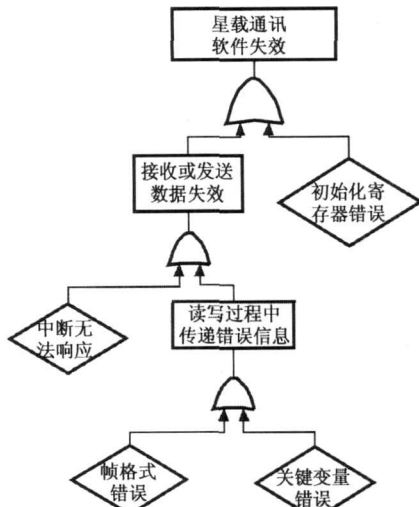


图1 星载通讯软件故障树

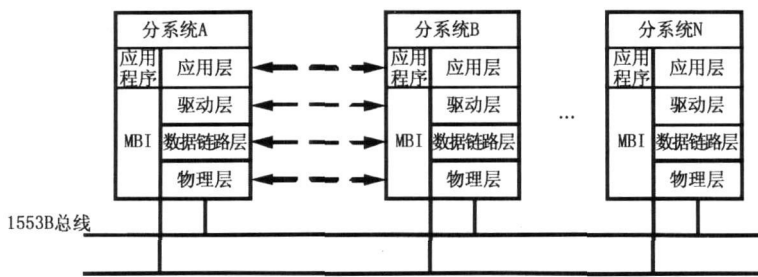


图2 通讯软件拓扑结构

址设置、内部自测试等功能。MBI 通讯程序根据 MBI 底层驱动程序以及载荷通讯需求接口数据单 (ICD) 生成, 完成协议芯片的初始化、管理以及总线数据接收发送功能。不同于一般通讯软件的是对其中 MBI 底层驱动程序和 MBI 通讯程序要进行相应的可靠性设计^[2]。

2.2 星载通讯软件设计流程

对于软件来说, 软件工程化技术是保证软件可靠性的基础, 它提出了软件开发的基本原则和要求。例如, 由不同人员完成软件设计、程序编码和软件测试; 保留设计文档、框图和测试记录, 使软件的开发和使用具有可追溯性。由于软件规模和复杂度是影响软件可靠性的一个主要因素, 在软件设计中采用自顶而下和模块化的设计方法, 可以合理划分模块, 实现低耦合高内聚的要求, 降低系统实现难度, 有利于提高软件的可靠性^[1]。

对于星载通讯软件, 除了依循上述的一般设计原则外, 根据其特点, 给出如图 3 的设计流程。主要包括了 ICD 表的设计、MBI 底层驱动的开发和 MBI 通讯程序设计。其中需明确以下几个问题。首先应确定总线传输的数据量及周期, 主要通过载荷通讯需求接口数据单 (ICD) 完成此项功能。其中 ICD 的内容主要包括上行的定义、下行参数的定义 (包括单位, 取值范围等), 参数地址分配以及周期的确定等。其中对于一些关键参数还要进行针对空间环境的可靠性设计。MBI 通讯程序在 MBI 驱动程序的基础上利用

ICD, 对通讯所需的时间和空间资源进行分配, 将周期性的通讯任务和非周期性的通讯任务划分开, 采取合理的调度方法, 保证通讯满足系统设计的要求, 进而完成通讯程序的设计, 最后通过 MBI 测试联调和系统联调进行验证和完善。

3 星载通讯软件设计

根据上述通讯软件的组成及开发流程, 根据空间辐射中的单粒子翻转的特点, 以在空间环境普遍应用的 1553B 通讯软件为例, RT 端的 1553B 通讯软件为例, 在 ICD 的编写、MBI 底层驱动和 MBI 通讯程序中主要对关键参数、数据存储、寄存器设置等几个方面进行可靠性设计。

3.1 1553B 通讯软件 MBI 实现

以 RT 端的软件为例, 针对 BU-65170 芯片, 主要实现了以下几个方面的功能。

(1) 配置寄存器的初始化

实现了主要工作模式的设定, 存储方式的选择及中断方式

(2) 设定等。

(2) 消息存储方式的选择和存储地址的分配

存储方式主要有单缓冲、双缓冲、循环缓冲方式。对于发送的消息根据消息块的大小可以选择不同的存储方式。数据块小于 32 字的选择单缓冲方式, 数据块大于 32 字应选择循环缓冲方式。对于接收的消息, 一般选用双缓冲方式, 但是对于发送周期间隔小或消息长度长的 (大于 32 字) 的应选用循环缓冲方式。

(3) 对于非法消息设置非法化表

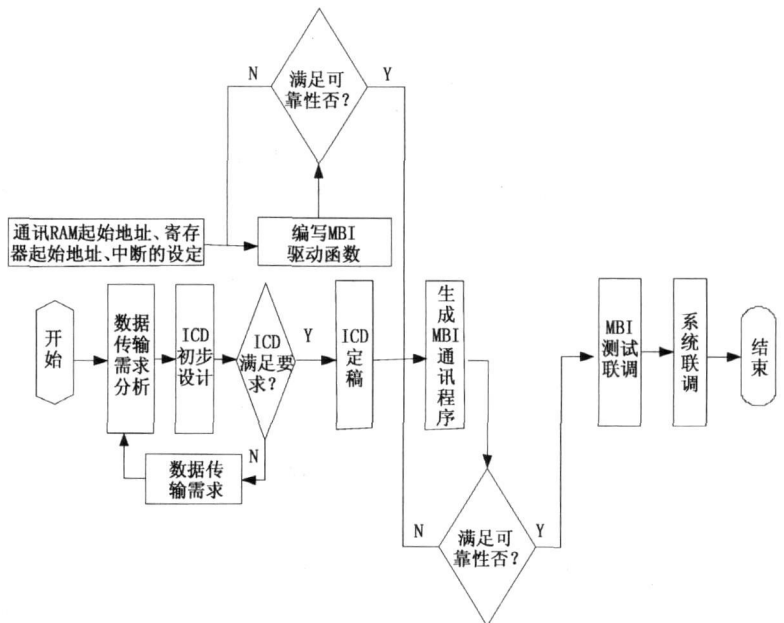


图3 星载通讯软件设计流程图

设置非法化表是保证 1553B 通讯可靠性的方法, 可以将不满足预先设定的子地址和消息块长度的消息设定为非法消息, 在通讯中这类消息将被芯片自动排除。

(4) 消息中断的处理。

消息中断主要实现了数据的移动。对于接收消息, 将在接口芯片上的消息移至用户处理区内。对于发送消息则是要求预先将消息移至设定的接口芯片缓冲区, 然后置相应的服务请求位。

下面以子地址 3 接收 32 个字, 广播方式; 子地址 5 接收 5 个字, 广播方式; 子地址 1 发送 21 个字为例的 1553B 通讯软件初始化 Rtnit () 的示例代码如下:

```
/* 寄存器初始化 */
AceRegWrite( CONFIG_3_REG, 0x8000); //设置 RT 为禁止运行模式
AceRegWrite( INTERRUPT_MASK_REG, 0x0010);
// 允许消息结束中断
AceMemWrite( (int *) StackPtrA, 0); // 置堆栈指针为 0
AceRegWrite( CONFIG_1_REG, 0x0cf80);
AceRegWrite( CONFIG_3_REG, 0x8019);
AceRegWrite( CONFIG_2_REG, 0x981a);
AceRegWrite( CONFIG_4_REG, 0x0000);
AceRegWrite( CONFIG_5_REG, 0x0700);
/* 接收和发送缓冲区的地址初始化 */
AceMemWrite
(( (int *) 0x0143, (int) AddrR3);
// 设定 R3 存储区首址
AceMemWrite
(( (int *) 0x0145, (int) AddrR4);
// 设定 R5 存储区首址
AceMemWrite
(( (int *) 0x0161, (int) AddrT1);
// 设定 T1 存储区首址
/* 存储方式的设定 */
AceMemWrite
(( (int *) 0x01a1, 0x4000);
// T1 单缓冲方式;
AceMemWrite
(( (int *) 0x01a3, 0x8200);
// R3 双缓冲方式
AceMemWrite
(( (int *) 0x01a4, 0x8200);
// R5 双缓冲方式
/* 非法化表的设定 */
AceMemWrite
(( (int *) 0x03c3, 0x0ffdf); // T1 21word
AceMemWrite
(( (int *) 0x0306, 0x0fffe);
// R3 32word 广播方式
AceMemWrite
(( (int *) 0x0308, 0x0ffdf);
// R4 5word 广播方式
其余设定为 0xffff;
```

初始化芯片后, 编写适当的通讯中断程序就可以正常的进行通讯, 这里不再详述。

3.2 星载通讯软件可靠性设计

3.2.1 ICD 表的设计

关键变量的定义, 不能采用一位来表示, 尤其不能用 0, 1 来表示不同的状态。一般应采取多位来表示。如设置系统加电标志= 0x5a5a, 表征未加电, 0x6161 表示已加电。

3.2.2 MBI 可靠性设计

对于这种情况, 一方面需要对取值的合法性进行判断, 另一方面, 下文将采取存储区表决法来消除单粒子翻转造成的影响。

而中断无法响应或初始化寄存器错误将采取定期更新的方式。

(1) 将接口芯片初始化相关的寄存器和存储区定期更新。

对一些关键寄存器, 如接口芯片的控制寄存器, 如果发生单粒子翻转, 将使整个通讯接口芯片失灵。而中断屏蔽寄存器的相应位发生翻转, 可能会使中断服务请求得不到响应。对这些寄存器应采取定期更新的方法, 以避免寄存器值发生跳变造成的后果。

由于通讯芯片的寄存器在更新期间将无法进行正常通讯, 因此应考虑在整个大系统内通过广播消息, 统一进行定期更新, 并在更新期间内, 停止通讯。

(2) 数据存储区的表决选取

将接收的数据存放在 3 个存储区上, 然后采用 3 取 2 的原则来选择数据。

(3) 数据冗余设计。

关键变量主要包括大循环控制变量、有限状态机控制变量、重要的全局指针、重要的全局标志等。上述变量关系到程序运行的总体进程, 发生单粒子翻转的概率比一些临时局部变量大很多, 造成的后果更严重。采用数据冗余设计的方法可降低发生翻转的概率。以全局变量 iSystemState 的设计为例具体说明如下:

未采用冗余时:

```
if(iSystemState== 0x33)
{
// 执行信号处理
iSystemState= 0x55;
}
```

采用冗余后:

```
iSystemState = voter(is1, is2, is3);
if(iSystemState== 0x33)
{
// 执行信号处理
iSystemState= 0x55;
is1= iSystemState;
is2= iSystemState;
is3= iSystemState;
}
```

采用冗余方法, 通过复制变量 iSystemState, 在使用时用 is1, is2, is3 的多数表决 (3 取 2), 增加了关键变量 iSystemState 的可信度。

3.2.3 其它

另外, 从整个星载软件可靠性考虑, 还应对程序存储区进行验证, 比较常用的方法是对整个程序代码进行求和校验, 并将结果作为常量存储在 3 个位置, 程序引导到 RAM 时, 对程序区进行校验, 并将结果与 3 个校验值的表决结果比较, 如果两者不一致, 应考虑重新引导。考虑到程序可能受到干扰, 还应加软件陷阱和软件看门狗。

4 验证

通常采用故障注入的方法, 对采取的可靠性措施进行验证。对固定位置的存储单元模拟 SEU 故障注入, 应用程序对此单元进行读操作时, 监视应用程序的运行状态, 从而可以判断应用程序对此位置的故障处理能力^[3]。具体的测试实例, 这里不再详述。

(下转第 735 页)

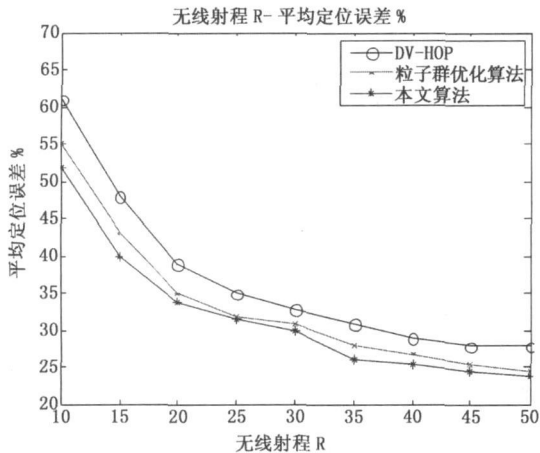


图4 无线射程R—平均定位误差

度,数量从150递增变化到400,无线射程R为15米,锚节点数量固定为20,测距误差设置为最小。从整体仿真效果来看,本文的算法比基于粒子群的定位算法和DV-HOP算法的平均定位误差更小。

图3给出了信标节点数量与平均定位误差之间的关系,实验中,在仿真区域内随机分布了200个节点,无线射程R为15米,测距误差设置为最小值。信标节点的数量从10到50,经过比较这3种定位算法,本文算法产生的定位误差有明显最小。

图4给出了无线射程R与平均定位误差的关系,在该仿真实验中,在仿真区域内随机分布200个节点,信标节点固定为20个,改变节点的无线射程R,变化范围为15米到40米。通过比较这3种定位算法,从仿真结果可以看出,当R增加时,平均定位误差都会减小,但是本文算法的效果更好,产生的平均定位误差较小。

综上所述,根据其仿真结果图可以看出,对本文提出的算法(ACOPSO-HOP)在相同的仿真条件下,产生的平均定位误差比DV-HOP算法和基于粒子群的定位算法更低。

4 结束语

本文中提出了一种新的WSN定位算法ACOPSO-Hop,这种定位算法是蚁群粒子群算法和DV-Hop算法的结合,DV-Hop算法用来估量未知节点与锚节点的测量距离,并用蚁群粒子群算法来最小化与DV-Hop的适应度函数,进一步优化DV-Hop算法,提高其定位精度。仿真结果表明该算法可有效的应用于WSN的定位算法中。

(上接第715页)

5 总结

本文从软件角度出发,根据空间环境的特点,针对星载通讯软件提出了一般的设计流程,并提出了有效的防单粒子翻转的设计,实际应用表明,此种方法提高了星载通讯软件的可靠性。

参考文献:

[1] 段星辉, 华建文, 代作晓, 等. 一种提高星载软件可靠性的开发

参考文献:

[1] Nieulescu D, Nath B. Ad-hoc positioning system [A] //In Proceeding of IEEE Global Communications Conference (GLOBECOM) [C], 2001, 2926-2931.

[2] 李文锋. 无线传感器网络与移动机器人控制 [M]. 北京: 科学出版社, 2009.

[3] Nieulescu D, Nath B. DV based positioning in ad-hoc networks [J]. Telecommunication Systems, Kluwer Academic Publishers, 2003, 267-280.

[4] 白凤娥, 姜晓荣, 牟汇慧. 无线传感器网络DV-Hop定位算法的研究 [J]. 计算机与数字工程. 2010, (3): 34-36.

[5] 张佳, 吴延海, 石峰, 等. 基于DV-HOP的无线传感器网络定位算法 [J]. 计算机应用. 2010, 30 (2): 323-326.

[6] Xiao G, Li S Z, Wang X H, et al. A solution to unit commitment problem by ACO and PSO hybrid algorithm [A]. Proceedings of the 6-th World Congress on Intelligent Control and Automation [C], 2006, 223-224.

[7] Simon S P, Padhy N P, Anand R S. An ant colony system approach for unit commitment problem [J]. Electrical Power and Energy Systems, 2006, (28): 315-323.

[8] Kumar V, Kumar R V R. Performance Analysis of a Finite Word Length Implemented CCK Modem with Rake Receiver for WLAN System [J]. IEEE, 2005, 62-67.

[9] Kannan A, Mao G Q, Vucetic B. Simulated annealing based localization in wireless sensor network [A] //Proceedings of the 30th IEEE Conference on Local Computer Networks [C], 2005, 154-157.

[10] 陈烨, 赵国波, 刘俊勇, 等. 用于机组组合优化的蚁群粒子群混合算法 [J]. 电网技术. 2008, 32 (6): 52-56.

[11] Nicholas Holden, Alex A. Freitas. A Hybrid PSO/ACO Algorithm for classification [A]. Genetic And Evolutionary Computation Conference Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation [C]. 2007, 2745-2750.

[12] Xiao G, Li S Z, Wang X H, et al. A solution to unit commitment problem by ACO and PSO hybrid algorithm [A]. Proceedings of the 6-th World Congress on Intelligent Control and Automation [C], 2006, 87.

[13] 柴宝杰, 刘大为. 基于粒子群优化的蚁群算法在TSP中的应用 [J]. 计算机仿真, 2009, 26 (8): 89-91.

[14] 赵仕俊, 孙美玲, 唐懿芳. 基于遗传模拟退火算法的无线传感器网络定位算法 [J]. 计算机应用与软件. 2009, 26 (10): 189-192.

方法 [J]. 计算机工程, 2009, 35 (12): 73-75.

[2] 代霜, 王槐, 徐抒岩, 等. 一种多总线通讯系统的实现 [J]. 计算机测量与控制, 2009, 17 (9): 1834-1836.

[3] 刘海峰. 星载软件容错设计及验证技术 [J]. 中国电子科学研究院学报, 2009, 4 (3): 313-316.

[4] 袁春如, 廖泰安, 贺红卫. 基于测试覆盖的嵌入式软件可靠性评估 [J]. 计算机工程与设计, 2009, 30 (9): 2198-2200.

[5] 龚健, 杨孟飞. 一种星载控制计算机智能容错方法 [J]. 空间控制技术与应用, 2008, 34 (6): 29-33.