Website: ycyk.brit.com.cn

一种高速可靠的大体量星载软件重构方案设计

代雅晴,韩晨阳,王泽波,赵 欣,马宏业 (北京遥测技术研究所北京 100076)

摘要:随着星载软件的复杂度与体量不断增加,对软件在轨重构并进行更新维护的功能愈发重要。当软件越来越大时, 使用低速通道进行重构的方案在时间上难以满足在一个测控弧段内重构软件的需求。同时,大软件使得数据存储空间更为 紧缺,无法使用三模冗余等传统方法保证程序数据的可靠安全。因此,本文提出了一种使用高速通道的可靠的大体量星载 软件重构方案。以固化在PROM(可编程只读存储器)上的引导监控程序作为根本保障,构建一个存于MRAM(磁随机存 储器)上专门用于高速重构软件的安全模式程序作为方案核心,并给星载软件加入自重构功能作为最常用的重构方式。通 过地面测试与在轨实验表明:该方案能够保证大体量软件重构功能的高速度与高可靠性,让星载软件的更新与维护更加安 全与便捷。

关键词:在轨重构;软件升级;高速通道;大体量软件;星载软件
中图分类号:V474;TP311 文献标志码:A 文章编号:2095-1000(2023)06-0019-10
DOI: 10.12347/j.ycyk.20230527001
引用格式:代雅晴,韩晨阳,王泽波,等.一种高速可靠的大体量星载软件重构方案设计[J].遥测遥控,2023,44(6):19 -28.

A design of high-speed and reliable refactor scheme for large-scale onboard software

DAI Yaqing, HAN Chenyang, WANG Zebo, ZHAO Xin, MA Hongye (Beijing Research Institute of Telemetry, Beijing 100076, China)

Abstract: With the complexity and volume of onboard software increasing, on-orbit refactor for software update and maintenance comes more important. As the software scale becomes larger, it is too hard to refactor software by low-speed channel in one observation arc. At the same time, large software makes data storage space more scarce, and traditional methods such as triple-mode redundancy cannot be used to ensure the reliability and security of program data. Therefore, a reliable refactor scheme for large-scale onboard software by using high-speed channel is proposed. Let the boot loader which store in read-only PROM be the fundamental guarantee of this scheme. A safe mode program saved in MRAM which is used to refactor the onboard software by high-speed channel is the kernel of this scheme. Meanwhile, adding a self-refactor function in onboard software as the most commonly used refactor method. The ground tests and on-orbit experiments indicated that this scheme can refactor large software with high speed and high reliability, which means it can update and maintain the onboard software safer and more convenient.

Keywords: On-orbit refactor; Software upgrade; High-speed channel; Large-volume software; Onboard software

Citation: DAI Yaqing, HAN Chenyang, WANG Zebo, et al. A design of high-speed and reliable refactor scheme for large-scale onboard software [J]. Journal of Telemetry, Tracking and Command, 2023, 44(6): 19–28.

引 言

随着航天技术的不断发展,卫星上可以承载 的功能也越来越多。与此同时,星载软件系统为 了实现诸多功能,其规模与复杂度也逐渐增加。 卫星在发射之后硬件无法改变,但是由于软件功能需要升级、存储的程序数据被太空中单粒子打翻造成错误,或者软件存在设计缺陷等情况,星载软件在卫星发射之后仍需要进行更新维护^[11]。给卫星的软件系统增加在轨重构功能,可以增加卫

星总体设计的灵活性与可靠性^[2],以便于延长卫星 的使用寿命^[3],即便程序数据出现意外损坏时,使 用重构功能也可以将软件恢复正确,而不会导致 卫星无法正常使用。然而由于星载软件的规模变 大,软件体量也不断增加,所以在一个有限的测 控弧段内进行一次完整的重构,将逐渐变得困难。

传统方案中,综合电子系统从低速或者高速 通道收到完整的软件数据,再通过CAN总线^[4,5]、 1553B^[6-8]等通道分发给对应的设备进行重构。当传 输数据过多的时候,无法在一个测控弧段内传输 完所有的数据,或者无法挤占其他业务时间去完 成所有数据的分发,还需要实现多弧段的断点续 传四与进度控制59,这样会使得传输和分发流程变 得复杂。数据补丁[10,11]或者文件替换[12]等方式,更 适用于由多个小模块构成的系统或者操作系统[12-14] 配置文件和模块的更新。这样下次系统重启的时 候,就可以加载新的功能模块或配置文件。但是 这种方法不适合替换一个完整执行文件的情况。 因此,若想在一个测控弧段内完成一个较大软件 的重构,必须启用高速通道。但是如果让引导监 控程序包含高速通道重构功能,该程序就会变得 过大,不能固化在大小有限的 PROM 上。所以可 以给星载软件加入高速自重构的功能,这样可在 一个测控弧段内完成重构。

但是由于软件体量变大,也使得 MRAM 的存储空间变得更为紧缺。通常情况下,星载软件只有几百 kB,使用三模冗余^[4,11]或者数据分区^[5,10]等方案,可以有效地保证程序的可靠性及安全性。即便某一份程序的数据发生错误,也可以启动另一份程序。但是当星载软件大小达到兆字节,考虑到还有其他业务数据的存储、硬件设计的困难以及卫星的总体造价,很难继续使用三模冗余或分区方法来保证安全性。然而,数据传输出现误码或数据帧丢失等星地传输中的常见问题,也可能导致原程序数据被重构数据替换后无法正常启动的情况。所以不能只开发星载软件的高速重构功能,还要防止其数据被破坏而无法工作。

因此,本文提出了一种高速可靠的大体量星 载软件重构方案。除了为星载软件本身加入高速 自重构功能外,还设计了一个专门用于高速重构 星载软件的小体量、双备份的安全模式程序作为 本方案的核心,以防星载软件无法自重构。二者 均存于MRAM中,且均可被更新重构,除了防止 程序被破坏,还可以更好地适应未来功能与协议 的变化。将引导监控程序固化在PROM上,用于 安全模式程序和星载软件的启动,以及安全模式 程序的重构更新,成为整个重构方案可靠性的根 本保障。

通过真实的工程测试与在轨实验表明:该方 案能够保证软件重构功能的高速度与高可靠性, 让星载软件的更新与维护更加安全便捷。

1 方案设计

1.1 总体设计

与本星载软件重构方案软件运行相关的环境, 如图1所示。其中,CPU连接了3个存储器用于程 序的存储以及运行,并连接一个1553B或CAN总 线控制器(本文以1553B总线为例),可以通过总线 收取综合电子系统从低速通道获取的数据和指令, 或者上报遥测给综合电子系统。新星载软件的数 据通过高速S测控通道,发送给XC7K325T的 LVDS接口,随后重构数据被传递给CPU进行处 理。CPU需要定时给AX2000发送"喂狗"信号, 确保软件正常运行,否则"咬狗"复位。







图2展示了系统中PROM与MRAM的存储空间大小及其包含的数据内容。方案设计思路与存储资源紧缺关系密切。

引导监控程序需要被固化在 PROM 中,不可 以被其他程序改变。该程序只需要完成基本的收 发指令、上报遥测和星载软件数据搬移启动任务, 约为 20 kB。若是为其加上配置高速通道数据、开 启高速转发功能并进行重构的代码,程序大小将



图2 存储空间与内容

Fig. 2 Storage space and contents

超过32kB,无法被固化在PROM中。

MRAM中的星载软件约为2 MB,被引导监控 程序启动后,将自动执行规定任务。MRAM中的 业务数据由星载软件进行管理,这些地址的数据 不会被其他程序进行修改。因为该星载软件本身 就已经配置了高速通道并开启高速转发功能,可 以和地面进行信息交互传输,所以只需要为其加入 重构功能,使其可以进行高速自重构即可。在正常 流程下,星载软件被引导监控程序启动后,就可以 进行软件升级。尽管如此,也需要防止星载软件因 为意外而无法启动,导致无法实现自重构。

为此,本方案设计了一个大小约为220 kB的 安全模式程序,专门用于启动高速通道,并进行 星载软件重构。两份相同的安全模式程序被存于 MRAM中,互为备份。当引导监控程序在30 s内 收到安全模式启动指令之后,就会根据指令中的 地址参数从MRAM中搬运并启动一份安全模式程 序,否则将正常启动星载软件。当其中一份安全 模式程序无法正常启动时,可以更改指令中的地 址参数以启动另一份。为防止星载软件和两份安 全模式的程序都被破坏而无法启动,引导监控程 序可通过1553B总线获取综合电子系统从低速通道 收到的重构包,对安全模式程序更新重构。因为该 程序很小,即使使用低速重构,重构时间在可接受 范围内。当安全模式程序启动后,就可以利用高速 S测控通道来更新存在于MRAM中的星载软件。

安全模式程序和星载软件均可以被更新重构, 即便是之后有业务功能、协议更新等调整,也可 以通过重构进行适应,增强了设计的灵活性。

1.2 程序功能设计

1.2.1 引导监控程序设计

引导监控程序被固化在 PROM 中, PROM 被

设置为只读模式,程序数据永远不会被修改,是 整个方案安全性的根本保障。根据上述的总体需 求与设计,引导监控程序需要包含以下功能:

① 完成芯片初始化工作,完成1553B总线初 始化工作;通过1553B完成最基础的遥测上报功 能;在加电30s之内没有收到任何数据或指令时, 搬运并启动星载软件。

② 加电 30 s之内,收到"引导监控搬运安全 模式程序"指令时,根据指令中的MRAM地址与 长度,搬运对应位置的数据来启动安全模式程序。

③ 当MRAM中安全模式的程序数据因为意外 被破坏的时候,使用指令启动安全模式程序会因 为程序无法正常"喂狗"而导致"咬狗"复位, 无法收到正确的安全模式遥测。此时可以选择通 过1553B总线低速重构安全模式。根据重构包中的 地址,将包中数据存到MRAM对应的地方。

④ 为了验证安全模式是否重构正确,当收到 "在线编程校验"指令时,根据指令中的地址与长 度计算异或校验。当其与自己计算的校验相同时, 可以判断重构正确。

有关引导监控程序的基本流程如图3所示。



1.2.2 安全模式程序设计

安全模式被存储于MRAM中,其可以被引导 监控程序启动、重构,是本方案的核心。星载软 件的高速重构过程,由该程序来实现。根据上述 的总体需求与设计,安全模式程序需要包含以下 功能:

 ① 在启动安全模式之后,自动配置高速S测 控通道,开启数据高速转发功能,等待数据和指 令,并定时上报最基本的遥测。

② 从高速S测控通道接收重构数据包,并根据数据包内给出的DDR地址,将重构数据暂时缓存在DDR中。

③ 收到"烧写指令"的时候,将 DDR 中缓存的数据烧写到 MRAM 中,替换原来的星载软件。

④ 在收齐所有重构数据包之后,需要判断收 到的数据是否完整正确;在烧写完成之后,需要 确认从 DDR 烧写到 MRAM 中的数据是否完全正 确。在这两个时间节点,向程序发送"在线编程 校验"指令,根据指令中的地址与长度计算异或 校验。当其与自己计算的校验相同时,可以判断 重构正确。

⑤ 当烧写到 MRAM 中数据确认正确的时候, 就可以重新启动引导监控程序,倒计时 30 s之后自 动启动星载软件。此时可以选择断电重启,也可 以发送软复位指令通过"咬狗"进行重启。

有关安全模式程序的基本流程如图4所示。 1.2.3 星载软件自重构功能设计

星载软件被存储于MRAM中,为其增加自重 构功能,可以自己对自己进行重构。在星载软件 正常的情况下,无需启动安全模式程序,只需要 在引导监控程序正常搬运启动星载软件后,直接 接收高速S测控通道传来的重构数据包即可。星载 软件重构功能设计与安全模式程序设计类似,除 了本身需要完成的任务外,还需要添加接收、缓 存、烧写、校验重构数据,以实现自重构功能。

1.3 重构包格式设计

1.3.1 高速重构包

若星载软件需要更新,可以直接使用高速S测 控通道进行自重构,因为星载软件本身就具有和 地面通信的高速通道,实现高速数据传输。这样 无需重新定义数据帧格式,只需要使用星载软件 已有的帧格式,并在应用层为其定义一个专属于 重构的数据包即可。对于安全模式程序而言,为 了减少地面在构造重构包时的复杂度,也使用同 样的数据帧格式进行传输。

为了更好地实现天地一体化,本文采用 IP over CCSDS 协议^[15]。链路层采用 CCSDS 高级在轨





系统(AOS)协议^[16]。网络层采用 IPv6 协议,将 LWIP 这个小型开源 TCP/ IP 协议栈移植到软件系 统中^[17]。传输层采用用户数据报(UDP)协议。相关 数据格式如图5 所示。

天地通信时,发送方先把IP包封装进AOS帧 后进行传输,然后接收方收到AOS帧之后将IP包 从中解析出来。在CCSDS空间链路协议上传递IP 包有以下3种方法^[18-20],可根据实际需求,任选 其一:

① 无需任何中介子层,直接将 IP 数据包置入 CCSDS 空间数据链路帧内实现 IP 数据包传递,如 图 6(a)所示。

②将IP数据包放在CCSDS空间包内,再将空间包分割或合并放入一个多路复用协议数据单元 (M_PDU)中,最后将M_PDU封入AOS链帧的数据域内进行传递,如图6(b)所示。

③用户自定义串行流封装。

| | | | | | 传 | 输帧尾 (| (可选) | | | | | |
|---------------|----------------|-----------|-----------------------|---------------------|--------------------|---------------------|--------------------|----------------|---------------------|------------------|--------------------|--------|
| | | 传输帧 帧头 | 传输帧 插入域 (可选) | | | 传输帧数 | 居域 | | · 控 (下 | 操作 制域 可选)(| 帧差错 控制域 (可选) | |
| | 6 | 或8 bytes | 可变 | | | 可变 | | | 41 | bytes | 4 bytes | |
| | I | | I | | | (a) AO (a) AOS f | S帧 rames | | I | I | I | |
| | | | | T 4 | H.B. Mc. | | | | | IPv | 6数据域 | |
| 版本号 4 bits | 通信类型 8 bits | 20 bits | 有效 载荷长度 16 bits | 下一个 头部 8 bits | 跳致 限制 8 bits | 源IP地址 128 bits | 目的IP地址 128 bits | 源端口 16 bits | 目的 端口 16 bits | 包长度 16 bits | 校验和 16 bits | UDP数据均 |

(b)使用UDP协议的Ipv6包

(b) Ipv6 packets using UDP protocol

图5 AOS帧与IPv6包格式

Fig. 5 Data structure of AOS and IPv6



意图



由于星载软件进行天地传输时存在不同业务, 需要在应用层通过包类型号进行区分。所以对于 重构功能来说,在主导头的包类型处为其定义一 个重构包类型号,并在包数据域中定义好重构数 据所需要的各类信息,就可以进行封装传输。包 数据域中需要填写重构程序的相关数据,包括: 重构数据,重构包内重构数据的长度,程序要缓 存在DDR中的首地址,以及包的校验和。最终定 义的重构数据包如表1所示,根据各层协议,除去 各层协议的包头以及高速重构数据包中其它必要

参数所占字节数,将"重构程序数据"设置为740 字节可以让高速重构包占满整个AOS帧,提高 AOS帧的利用率。

表1 高速重构数据包定义

Table 1 High-speed refactor packet

| 序号 | 数据定义 | 字节数 |
|----|----------------|-----|
| 1 | 主包头(包含重构包类型号) | 6 |
| 2 | 副包头 | 8 |
| 3 | 程序在DDR中缓存的起始地址 | 4 |
| 4 | 重构数据长度 | 2 |
| 5 | 重构程序数据 | 740 |
| 6 | 校验 | 2 |

1.3.2 1553B重构包

若安全模式程序无法启动或者需要更新,通 过1553B总线获取综合电子系统从低速通道收到的 重构包,对其进行重构。因为本身1553B总线上就 使用 CCSDS 帧发送指令,为了与指令遥控帧有所 区别,所以需要在1553B包数据域的"功能识别" 处,为重构包定义一个类型号。在"功能识别" 后,需要放入重构程序数据、该数据的长度、注 入到MRAM中的首地址以及校验。1553B重构数 据包格式如表2所示,根据约定的1553B协议,将 "重构程序数据"设置为232字节,可以100%利用 1553B一帧的空间。

2 可靠性与速度分析

2.1 可靠性分析

本方案高可靠性主要体现在三个方面:任务 可靠、数据可靠、程序可靠。

任务可靠:除非PROM被物理破坏,否则无

表2 1553B重构数据包定义

Table 2 1553B refactor packet

| 序号 | 数据定义 | 字节数 |
|----|--------------|-----|
| 1 | 主包头 | 6 |
| 2 | 副包头 | 8 |
| 3 | 功能识别(重构包类型号) | 2 |
| 4 | 程序注入MRAM的首地址 | 4 |
| 5 | 重构程序长度 | 2 |
| 6 | 重构程序数据 | 232 |
| 7 | 校验 | 2 |

论MRAM中任何数据被破坏,都能应用本方案完成重构星载软件的任务。

一般情况下,当星载软件可以正常启动并执 行任务时,直接使用星载软件的自重构功能进行 高速重构。当星载软件因为意外情况无法启动时, 使用引导监控程序启动安全模式程序。安全模式 程序将使用高速通道对星载软件进行重构。由于 安全模式程序较小且只能低速重构,所以在 MRAM中存储两份一模一样的程序,作为冗余备 份。当两份安全模式程序也被意外破坏时,首先 利用引导监控程序低速重构安全模式程序,再利 用安全模式程序对星载软件进行高速重构。

当且仅当固化引导监控程序的 PROM 发生故障,而导致引导监控程序都无法启动时,该卫星才彻底失效报废,无法工作。除此以外,卫星均可以重构星载软件,恢复卫星本身的功能。各软件之间的重构关系如图7所示。

数据可靠: 传输的重构数据是可靠的。与地 面数据传输不同的是, 天地传输的误码率高, 容 易丢帧。针对传输中的误码、丢帧情况, 有以下



Fig. 7 Relationship between three software

三重保障,可以保证上传并烧写的重构数据是正确的,且与地面保持一致的:

① 计算 CCSDS 帧、UDP 包、重构包的校验和 或判断这些数据包的合法性。

② 查看正确收包计数、错误收包计数遥测是 否符合预期。

③ 计算卫星 DDR/MRAM 中重构数据的异或 校验,并与地面计算的校验进行对比。

本方案中软件的 1553B 总线中使用的是 CCSDS帧,高速通道中采用的是IP over CCSDS的 技术,可以通过协议中的校验和保证传输数据的 正确。当判断校验错误,直接舍弃该数据包, MRAM/DDR中不存该帧数据。信息传输过程中, 丢帧是随机发生的常见现象,使用UDP协议传输 也没有重传机制,MRAM/DDR中将缺少该帧的数 据。当收到所有数据后,通过正确收包计数、错 误收包计数和异或校验判断存在MRAM/DDR的程 序数据是否与地面一致。

但是当某一帧出现错误(误码、丢帧)时,本方 案无法确认具体是哪一帧错误,只知道MRAM/ DDR 中缺失数据。可以选择用异或校验,利用二 分法查找到缺失的帧后,再补传一帧,也可以选 择重新上传一次所有数据。由于发生错误是随机 的,两次传输中同一帧发生错误的概率不大,上 一次由于错误而没有存数据的位置会存放本次传 来的新数据,而本次错误的位置,本身又存有上 次正常传输的正确数据。所以一般完整上传两次 数据之后,便能得到完全正确的数据。两种解决 传输问题的方法如图8所示。

程序可靠:本方案中使用到的三个程序,当 收到错误数据、非法指令时,系统能够及时通过





Fig. 8 Two methods to solve transmission problems

遥测报告错误,能够舍弃错误数据以及非法指令, 而不会导致系统执行错误指令、数据溢出、异常 复位等情况。这要求程序中需要针对可能出现的 异常情况进行处理,针对不同的错误进行不同的 操作。

为了验证该方案的可靠性,在某项目型号上 应用此方案。地面测试利用PC机、产品、地检来 模拟环境进行测试。如图9所示,PC机与产品之 间通过1553B总线连接,PC向产品发送指令与 1553B重构包,并接收产品上报的遥测。PC机把 星载软件的数据通过网线发送给地检,地检会模 拟从高速S测控通道向产品发送高速重构包。

实验对引导监控重构安全模式的流程,安全 模式重构星载软件的流程,星载软件自重构的流 程,以及各种异常错误分支进行了测试,测试内 容如表3所示。经过充分测试,所有测试结果均符 合期望结果。只要 PROM 内的引导监控程序可以



Fig. 9 Environment of ground test

正常启动,无论MRAM中数据怎样被破坏,都可 以对星载软件进行重构。当收到数据、指令错误 时,软件可以及时报告异常,并继续正常运行。 目前,该型号于在轨测试场景下已成功重构多次, 效果良好。

| 编号 | 实验内容 | 期望结果 |
|----|--|-----------------------------------|
| 1 | 利用引导监控程序启动星载软件 | 成功启动星载软件 |
| 2 | 利用引导监控程序重构安全模式程序 | 遥测表中,重构包正确计数增长至期望值 |
| 3 | 利用引导监控程序校验MRAM中重构的安全模式程序是否正确 | 遥测表中,正确指令计数+1,校验值与手动计算 一致 |
| 4 | 利用引导监控程序启动安全模式程序 | 成功启动安全模式 |
| 5 | 利用安全模式程序或星载软件高速重构最新版星载软件 | 遥测表中,上注标识变化,重构包正确计数增长 至期望值 |
| 6 | 利用安全模式程序或星载软件校验 DDR或 MRAM 中重构的星载软件是否正确 | 遥测表中,正确指令计数+1,校验值与手动计算 一致 |
| 7 | 安全模式程序或星载软件将DDR中缓存的数据烧写至MRAM | 遥测表中,上注标识变化 |
| 8 | 安全模式程序或星载软件软复位,重启最新版星载软件 | "咬狗"复位,成功启动最新版星载软件 |
| 9 | 引导监控程序重构安全模式程序时,地址非法 | 遥测表中,错误指令计数增加 |
| 10 | 引导监控程序校验重构的安全模式程序时,地址或长度非法 | 遥测表中,错误指令计数+1,校验值无变化(说明 未计算校验) |
| 11 | 引导监控程序重构安全模式程序时,1553B重构包校验错误或不 合法 | 遥测表中,错误数据计数增加 |
| 12 | 利用安全模式程序或星载软件高速重构星载软件时,地址非法 | 遥测表中,错误指令计数增加 |
| 13 | 安全模式程序或星载软件校验重构的星载软件时,地址或长度非法 | 遥测表中,错误指令计数+1,校验值无变化(说明 未计算校验) |
| 14 | 安全模式程序或星载软件校验重构的星载软件时,AOS帧或UDP校 验错误或不合法 | 遥测表中,错误数据计数增加 |

表 3 地面测试内容 Table 3 Content of ground test

2.2 速度分析

传输层的协议会影响传输的速率与可靠性,

一般选择 TCP 协议或者 UDP 协议。TCP 在通信前 建立连接,有确认应答机制、错误重传机制,可 靠性更高。但是天地通信时延长、数据量大、易 丢帧,使用TCP的确认应答机制会大大降低传输 效率。UDP协议通信前无需建立连接,可以直接 进行通信,虽然并不如TCP可靠,但是非常适合 于传输时效性高的业务。因此,本方案选择UDP 作为传输层协议,不建立连接,直接发送重构包 进行传输。

表4展示了各个重构步骤传输时间的理论值。 以2 MB的星载软件和220 kB的安全模式程序为 例,高速通道的理论传输速率为1 Mbit/s,低速通 道的理论传输速率为2 kbit/s,设一个测控弧段平 均时长为10 min。该产品可以直接接收高速通道发 送的数据,但不能从低速通道直接接收数据。产 品的LVDS接口可以接收500 Mbit/s的数据,其接 收能力远高于地面从高速通道发送数据的能力。 1553B重构包需要综合电子系统从低速通道收取后 分发给产品。但是由于综合电子系统与地面收发 协议未知,所以在理论计算时,只计算理论速度 传输全部有效数据的时间,忽略收发协议中包头 包尾的数据长度。

| 表4 各个 | 重构步骤的理论传输时间对比 |
|-------|---------------|
|-------|---------------|

| Table 4 | Theoretical | transmission | time | comparison | of re | efactor | steps |
|---------|-------------|--------------|------|------------|-------|---------|-------|
| | | | | | | | |

| 传输方法 | 传输设置 | 传输时间 | 需要测控弧段数 (重构时间/测控弧段平均时长) |
|-----------------|------------------------|--------------|----------------------------|
| 低速通道传输星载软件数据 | 2 kbit/s,有效数据2 MByte | 136 min 32 s | 14 圏 |
| 高速S测控通道传输星载软件数据 | 1 Mbit/s,有效数据2 MByte | 16 s | 1 圏 |
| 低速通道传输安全模式程序数据 | 2 kbit/s,有效数据220 kByte | 14 min 40 s | 2 圈 |

通过理论计算值可以发现,如果使用低速通 道重构大体量的星载软件,从地面传输到太空中 需要14圈。低轨卫星绕地球一周约为90 min,每 次只有约10 min可以用来上注程序,几乎需要一 天时间才能够将完整的软件上注完成,这是难以 接受的。而使用高速通道传输软件,在1圈内就可 以完成上注,非常便捷。而当星载软件和安全模 式都被破坏,无法高速重构时,需要引导监控程 序花费2圈时间来重构安全模式,再花费1圈来高 速重构星载软件,这是一个相对可以接受的时间 消耗。

为了测试各个重构步骤所花费的时间,依旧 使用图9所示的环境进行实验。表5中展示了地面 测试时被重构软件或程序的大小,表6中展示了地 面测试时各个重构步骤的时间。地检可以模拟地 面站从高速通道发送高速重构包; PC可以模拟综合电子系统缓存所有软件数据后,从1553B总线发送1553B重构包。使用PC模拟综合电子系统分发数据包时,由于1553B总线上只有重构包,所以可以设置最大速率100 ms一帧进行发送。而在真实的太空环境中,总线上还会传输其他的数据,综合电子系统约1 s能够发送一次重构包。所以即便综合电子已经缓存了来自低速通道的全部数据,通过总线分发给产品时也会比表中所测数据慢上10倍左右。

表5 软件/程序大小

| Table 5 | Size of software/program |
|---------|--------------------------|
| 名称 | 大小 |
| 星载软件 | 2.07 MB |
| 安全模式程序 | 予 221 kB |

| 表6 行 | 各个了 | 「构 | 步骤 | 的测 | 试时 | 间对 | 比 |
|------|-----|----|----|----|----|----|---|
|------|-----|----|----|----|----|----|---|

| Table 6 | Test time | comparison | of 1 | refactor | steps |
|---------|-----------|------------|------|----------|-------|
|---------|-----------|------------|------|----------|-------|

| 重构步骤 | 发送设置 | 重构时间 |
|---------------|------------------------|-------------|
| 1553B重构星载软件 | 100 ms一帧,每帧包含232字节重构数据 | 19 min 33 s |
| 高速S测控通道重构星载软件 | 50 ms一帧,每帧包含740字节重构数据 | 1 min 20 s |
| 1553B重构安全模式程序 | 100 ms一帧,每帧包含232字节重构数据 | 2 min 3 s |

通过实际测试可以发现,产品使用高速通道 收星载软件重构包只需要1圈的时间,可以实现快 速重构。当使用低速通道重构安全模式程序时, 除了要考虑到综合电子系统接收低速重构包的速 率,也要考虑到从总线上分发给产品的时间,需 要为综合电子系统设置合理的收发协议与缓存 机制。

3 结束语

本文介绍了一种高速可靠的大体量星载软件 重构方案。固化在 PROM 上的引导监控作为重构 方案的根本保障,只要其程序不被破坏,就可以 完成重构星载软件的任务。同时给星载软件加入 高速自重构功能,让大小达到兆字节的软件也可 以在一个测控弧段内完成重构任务。而体量小、 双备份的安全模式作为本方案的核心要点,旨在 于解决存储空间有限,无法使用三模冗余保证星 载软件安全性的问题。当星载软件数据被破坏而 无法启动,又没有备份能够恢复正常工作时,使 用专门用于高速重构的安全模式来重构星载软件, 可以使卫星继续正常工作,延长卫星的寿命。同 时,安全模式也可以被重构,以应对安全模式也 被破坏或将来的功能升级与协议改动,增加了设 计灵活性。方案中各个程序软件的通信协议设计、 功能设计以及系统设计保证了数据传输与指令执 行的正确性。通过具体型号的地面测试与在轨测 试,验证了本方案的可行性,为存储空间有限的 情况下高速重构大体量软件提供了新的思路。

参考文献

- 王战强, 翟盛华. 星载处理设备软件在轨重构技术研究
 [J]. 空间电子技术, 2013, 10(1): 7-13, 43.
 WANG Zhanqiang, ZHAI Shenghua. Research on software on-orbit reconfiguration of space-borne processing equipment[J]. Space Electronic Technology, 2013, 10(1): 7-13, 43.
- [2] 白亮, 邱源, 韦杰, 等. 基于动态库的星载软件可重构设 计与实现[J]. 上海航天(中英文), 2021, 38(4): 84-91.
 BAI Liang, QIU Yuan, WEI Jie, et al. Design and implementation of on-board software reconfiguration based on dynamic library[J]. Aerospace Shanghai(Chinese & English), 2021, 38(4): 84-91.
- [3] 刘伟伟,李欣,于俊慧,等.一种星载软件重构系统的设计与实现[J/OL]. 计算机测量与控制. 2022-12-30[2023-01-31]. http://kns. cnki. net/kcms/detail/11.4762. TP.20221329.1754.008.html.
- [4] 马传干. 基于 FPGA 和 DSP 的星载软件动态重构设计 [J]. 电子设计工程, 2021, 29(11): 50-55.

MA Chuangan. Dynamic reconstruction design of spaceborne software based on FPGA and DSP[J]. Electronic Design Engineering, 2021, 29(11): 50-55.

- [5] 熊浩伦, 闫国瑞, 李国军, 等. 基于最小系统的小卫星在 轨软件重构系统设计[J]. 遥测遥控, 2020, 41(3): 48-55. XIONG Haolun, YAN Guorui, LI Guojun, et al. Design of on-orbit software reconfiguration system of small satellite based on minimum system[J]. Journal of Telemetry, Tracking and Command, 2020, 41(3): 48-55.
- [6] 张衡,顾泽凌,杨明远,等.基于1553B的DSP在轨编程 设计与实现[J].制导与引信,2020,41(4):33-37,42.
 ZHANG Heng, GU Zeling, YANG Mingyuan, et al. Design and realization of on-board programming for DSP based on 1553B[J]. Guidance&Fuze, 2020, 41(4): 33-42,42.
- [7] 丁磊,刘海峰,张小林.一种星载软件无线电系统的高 可靠智能重构方法[J]. 航天电子对抗, 2018, 34(4): 24-28, 33.
 DING Lei, LIU Haifeng, ZHANG Xiaolin. A high reliability intelligent reconfiguration method for spaceborne software radio systems[J]. Aerospace Electronic Warfare,
- 2018, 34(4): 24-28, 33.
 [8] 李吉锋,杨刚,王戈,等.星载环境FPGA软件在轨重加载的一种方法[J].现代电子技术, 2012, 35(9): 87-90.
 LI Jifeng, YANG Gang, WANG Ge, et al. Method of onorbit reloading for FPGA software in satelliteborne environment[J]. Modern Electronics Technique, 2012, 35(9): 87-90.
- [9] 庞波,郝维宁,张文峰,等.一种SRAM-FPGA在轨重构的工程实现方案[J],航天器工程,2017,26(5):51-56.
 PANG Bo, HAO Weining, ZHANG Wenfeng, et al. Scheme of SRAM-FPGA on-orbit reconfiguration[J].
 Spacecraft Engineering, 2017, 26(5):51-56.
- [10] 韦涌泉,董振辉,张红军.一种基于文件的嵌入式星载 软件在轨升级方法[J].单片机与嵌入式系统应用, 2018, 18(5): 32-35.
 WEI Yongquan, DONG Zhenhui, ZHANG Hongjun. Method of embedded onboard software upgrading based on file[J]. Microcontrollers & Embedded Systems, 2018, 18(5): 32-35.
- [11] 钱方亮,林荣锋,周宇,等.一种基于微小卫星系统软件在轨编程功能的设计方法[J].计算机应用与软件, 2018, 35(12): 16-20.

QIAN Fangliang, LIN Rongfeng, ZHOU Yu, et al. Design of on-orbit programming for microsatellite system software[J]. Computer Applications and Software, 2018, 35(12): 16-20.

- [12] 郭宗芝, 刘彬, 邹玉龙, 等. 基于模块动态加载机制的 航天器软件重构方案研究[J]. 计算机测量与控制, 2018, 26(6): 126-129, 148.
 GUO Zongzhi, LIU Bin, ZOU Yulong, et al. Research on spacecraft software refactoring scheme based on dynamic loading mechanism of module[J]. Computer Measurement & Control, 2018, 26(6): 126-129, 148.
- [13] 张国柱,花秋琴,曹斌,等.基于ARM+uC/OS-II的星载 软件重构技术研究[C]//软件定义卫星高峰论坛,2018.
- [14] 王明亮, 王永, 常亮, 等. 基于VxWorks的星务软件快速加载重构研究[J]. 电子设计工程, 2022, 30(8):26-30, 35.

WANG Mingliang, WANG Yong, CHANG Liang, et al. Research on OBDH software reconstructing technique of quickly load and reconfiguration based on VxWorks [J]. Electronic Design Engineering, 2022, 30(8): 26-30, 35.

- [15] CCSDS. 702. 1-B-1-2012, IP over CCSDS space links[S]. Washington, D.C.: NASA, 2012.
- [16] CCSDS. 732. 0-B-3-2015, AOS space data link protocol[S]. Washington, D.C.: NASA, 2015.
- [17] YU S, QIU Q, ZHOU B, et al. Design and implementation of spacecraft onboard network based on LWIP[C]//

International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMC-CC), 2018.

- [18] 闫朝星, 付林罡, 谌明, 等. 天地信息网络协议融合技术综述[J]. 遥测遥控, 2020, 41(6): 30-38. YAN Chaoxing, FU Lingang, CHEN Ming, et al. Overview of space-ground information network integration techniques[J]. Journal of Telemetry, Tracking and Command, 2020, 41(6): 30-38.
- [19] 董晓臣.基于 FPGA的 IPv6 over AOS 网关的设计与实现[D].保定:河北大学, 2021.
- [20] 赵和平,何熊文,刘崇华,等.空间数据系统[M].北京: 北京理工大学出版社,2018.

[作者简介]

| 代雅晴 | 1997年生, | 硕士, | 助理工程师。 |
|-----|---------|-----|--------|
| 韩晨阳 | 1995年生, | 硕士, | 工程师。 |
| 王泽波 | 1999年生, | 大专。 | |
| 赵欣 | 1993年生, | 硕士, | 工程师。 |
| 马宏业 | 1982年生, | 硕士, | 高级工程师。 |

(本文编辑: 傅 杰)