

基于最小系统的小卫星在轨软件重构系统设计

熊浩伦, 闫国瑞, 李国军, 吕 达
(航天东方红卫星有限公司 北京 100094)

摘要: 提出一种基于星载终端处理设备最小系统的小卫星在轨软件重构系统设计方案。依据星上信息流实现现状, 设计重构系统的系统框架, 提出星载终端处理设备的最小系统组成、通用功能集和需要满足的整星安全性功能需求。系统中, 应用软件程序数据按照注入帧格式通过遥控通道注入到星上, 可缓存在星务存储器中再分发给终端目的设备, 通过最小系统软件对应用程序进行引导和加载管理, 整个过程均有差错控制设计以保证其可靠性和安全性。软件重构试验和容错试验表明, 利用该系统可实现星载微计算机软件和 FPGA 配置数据的可靠上注、可靠存储和可靠加载。

关键词: 最小系统; 在轨重构; 软件升级; 小卫星

中图分类号: V447 **文献标识码:** A **文章编号:** CN11-1780(2020)03-0048-08

Design of on-orbit software reconfiguration system of small satellite based on minimum system

XIONG Haolun, YAN Guorui, LI Guojun, LYU Da
(DFH Satellite Co.Ltd., Beijing 100094, China)

Abstract: Based on the minimum system of onboard terminal processing equipment, a scheme of on-orbit software reconfiguration system is designed. According to the characteristics of current satellite information flow, the system framework is designed, and the minimum system composition, general function set and the whole satellite security functional requirements of the onboard terminal equipment processor are proposed. In the system, the application software program data is made into data transfer frame and transmitted into the satellite through the telecommand channel, which can be stored in the memory of OBDH subsystem and send to the terminal equipment. Application program can be bootload and managed by the minimum system with error control design to ensure its reliability and safety. The test shows that the system can realize the reliable transfer, storage and bootload of onboard microcomputer software and FPGA configuration data.

Key words: Minimum system; On-orbit reconfiguration; Software upgrade; Small satellite

引 言

随着软件无线电技术、综合电子技术的日益发展和星上信息处理能力的不断提升, 硬件功能软件化已成为当前航天技术重要的发展趋势。小卫星因其高度集成的特点, 软件(含 FPGA 产品)的作用和地位越来越重要, 且随着小卫星寿命的增长和任务难度的加大, 软件产品的复杂度越来越高, 需要具备在轨重构和更新等维护能力, 以应对可能出现的任务扩展和软件潜在缺陷^[1]。目前, 小卫星星载软件多使用 PROM、EEPROM 或 FLASH 作为软件的存储介质, 软件固化在 PROM 中的星载计算机只具备有限的 RAM 区数据在轨维护能力, 如配置参数修改、注入补丁程序; 软件存储在可在线编程器件(如 EEPROM、FLASH)中的星载计算机可实现软件配置参数修改、注入补丁程序和在线完整重构^[2,3]。可重构 FPGA 主要通过更改 SRAM-FPGA 的外部配置存储器的内容实现, 维护时是以完整替换的方式进行^[4,5]。受空间环境影响, 与可重复编程存储器相比, 使用只读存储器作为星载软件的存储器的可靠性更高, 但不利于软件的在轨维护^[6]。小卫星的航天器属性和高集成特点要求在工程设计上既要使用有限的各种资源确保卫星的寿命和高可靠性, 也要不断增加设计的灵活性使电子设备小型

化和产品化。软件在轨维护的最高追求是可靠性和灵活性的协调统一，将星载设备的基本功能与型号专属应用功能剥离开，基本功能作为生存功能，具有高可靠、通用化的特点，型号专属应用功能具有软件可定义、可重构的特点，两者兼备则可以实现这一目标。本文提出了一种利用可固化星载处理设备最小基本功能实现小卫星软件在轨重构的系统，实现软件可靠上注、可靠存储和可靠加载，提高了小卫星软件在轨维护的可靠性和灵活性。

1 总体设计

1.1 系统框架设计

当前，小卫星所涉及的领域已包括了空间通信与导航、地球遥感、战略武器、空间技术试验与验证、深空探测等^[7]。将小卫星分为公用平台和有效载荷两大模块，分别进行设计，一种公共平台可满足具有相似重量、相似姿态控制和相似能源需求的多个空间飞行任务的要求^[8]。虽然不同载荷使用的某一卫星公用平台的星载软件功能差异性往往较小，但新载荷的工作需求也会导致公用平台星载软件功能进行定制式修改，破坏卫星平台设备的一致性，需要将共性功能产品化并支持软件重构。小卫星载荷综合电子技术的发展，使载荷机构、天线或镜头共用技术和星载载荷数据实时处理技术得以实现，这些均是建立在电子设备具有重构功能的基础之上^[9]。

与地面系统软件比，星载软件在轨维护具有较多的条件限制：软件注入依赖的星地测控弧段短，通信速率低，有限时间注入的数据量有限，进行较大规模软件注入时注入和确认过程漫长，效率低下；空间环境复杂，对星载软件可靠性要求高，可编程存储器存储单元数据存在单粒子翻转或失效的风险，软件注入及加载必须进行正确性和完整性检查；关键星载软件，如星务、姿控和电源分系统的软件，需要在维护失败时能及时回退到以前的版本，并且在维护时依然能支持卫星平台业务的运转^[10]。

基于小卫星上述特点和在轨软件重构的限制条件，在不改变现有小卫星信息数据流的基础上进行在轨软件重构系统设计。采用现场总线（CAN）技术，国内小卫星构建了以星务管理计算机为控制和调度中心、以面向控制对象的各分系统计算机为执行单元的网络分布式系统。小卫星在轨软件重构系统充分利用这一信息流基础进行设计。该系统中，星载计算机或FPGA具备只读存储器存储该产品的最小系统功能软件，具备可重复编程存储器存储该产品的完整应用功能软件；最小系统功能软件能实现完整应用功能软件的注入和引导加载；待重构软件可通过高速遥控通道上注到软件公用临时存储区，由星务计算机实现软件的分发调度和管理。该系统的总体信息流如图1所示。每个挂载在整星现场总线上的终端设备其软件实现功能可概括为两个阶段：第一阶段为最小系统阶段，该阶段能完成基本的设备安全控制、应用软件的注入修改和应用软件的引导加载；第二阶段为应用软件阶段，该阶段实现设备承担的应用功能，并具有升级程序接口，能接收和存储新的软件程序；第二阶段的软件通过复位或切机指令可以进入第一阶段运行，当第二阶段软件引导不成功或接收到驻留第一阶段指令，设备才会长期运行于第一阶段；否则，成功引导加载应用软件后，直接运行应用软件。测控系统因其功能固定，无需进行软件重构，上注软件可通过测控的低速信道慢速立即注入的方式注入到终端设备中，也可通过测控的高速上行通道快速注入到软件临时存储区中，注入完毕后经由星务计算机调度分发，达到上注通道冗余和适应补丁程序注入、完整程序注入的目的。软件数据的注入、分发和引导加载均须经过正确性和完整性检查确认。

1.2 最小系统设计

最小系统是在完整的硬件环境上运行的，但是并不实现完整的硬件功能。最小系统除了使用终端设备的总线通信接口、存储器资源和必须的卫星安全控制部件外，不使用其它硬件资源。最小系统在硬件描述和软件功能上具有一致性。

终端设备采用星载计算机作为软件重构载体时，最小系统在硬件资源和软件功能上的共同点为：

- ① 存储器包含 PROM、FLASH 和 SRAM，最小系统软件存储在 PROM 中并运行在 PROM 中，

应用软件存储在 FLASH 中并被引导加载到 SRAM 中运行;

② 最小系统软件包含 BootLoad 功能、CAN 总线通信功能和安全功能;

③ 应用软件注入时在 FLASH 中存储 3 份;

④ FLASH 中至少有 2 个存储分区, 非重构软件分区和重构软件分区, 仅重构软件分区的 FLASH 可进行在轨擦除和写入操作;

⑤ 应用软件引导加载时进行三取二比对, 比对通过后加载到 SRAM 中执行, 如果重构软件分区的软件无法引导加载则自动回退到加载非重构软件分区, 如果非重构软件分区软件也无法完成引导加载则直接放弃引导加载, 停留在最小系统中运行;

⑥ 最小系统软件支持将完整应用软件直接注入到 SRAM 中运行。

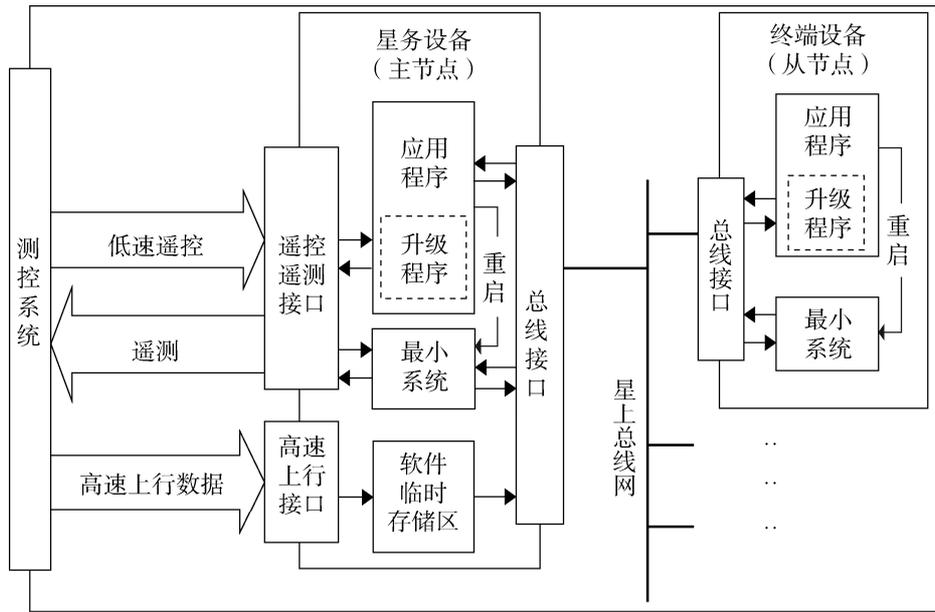


图 1 小卫星软件重构系统信息流图

Fig. 1 Information flow chart of small satellite software reconfiguration system

终端设备采用 FPGA 作为重构载体时, 最小系统在硬件资源和软件功能上的共同点为:

① 使用从并配置模式, 利用专用控制电路或微处理器来控制重构时序对 FPGA 进行配置, 当使用专用控制电路时优先选用反熔丝型 FPGA 实现功能, 其次可选择 FLASH 型 FPGA, 当使用微处理器时优先使用 PROM 存储器存储控制程序, 待配置 FPGA 的配置信息存储在 FLASH 中, 如图 2 所示;

② 最小系统功能通过配置 FPGA 或通过微处理器实现, 最小系统功能包括程序加载、程序重构、在线实时刷新、CAN 总线通信和非使用硬件的安全管理;

③ FPGA 配置数据在 FLASH 中存储 3 份;

④ FLASH 中至少有 2 个存储分区, 非重构软件分区和重构软件分区, 仅重构软件分区的 FLASH 可进行在轨擦除和写入操作;

⑤ FPGA 配置数据加载时进行三取二比对, 比对通过后进行在线重配置, 如果无法完成重配置则自动读取非重构区的数据进行在线重配置, 如果非

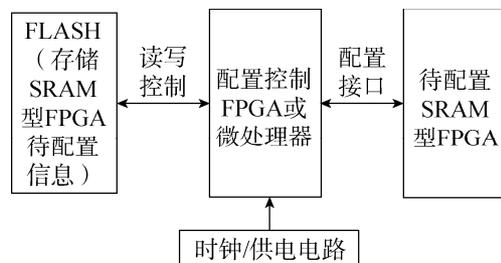


图 2 FPGA 重构最小系统

Fig. 2 Minimum system of FPGA reconfiguration

重构数据也有错误, 则待配置 FPGA 暂时无法实现功能, 等待地面通过指令实现在线重配置。

1.3 整星安全性设计

小卫星整星在轨的基本安全性主要包括以下三点:

- ① 整星上下行通道的安全性;
- ② 整星能源系统的安全性;
- ③ 天上信息传输通道的安全性。

当某设备一段时间内处于最小系统运行时, 必须确保上述三条基本安全性不受影响。星上提供星务管理、电源管理、姿态控制和导航信息生成等业务的卫星平台设备以及载荷设备的最小系统功能集至少如表 1 所示。

表 1 各设备最小系统功能集

Table 1 Function set of minimum system for different equipments

设备业务	最小系统功能
星务管理	遥控与遥测、总线通信、时间信息、整星安全模式控制、软件注入与引导加载
电源管理	能源状态采集、总线通信、安全管理、软件注入与引导加载
姿态控制	姿态采集、总线通信、安全管理、软件注入与引导加载
导航信息生成	总线通信、安全管理、软件注入与引导加载
载荷	总线通信、安全管理、软件注入与引导加载

2 软件注入与加载设计

2.1 注入帧格式

星载计算机的嵌入式软件一般规模不大, 可执行文件的大小往往小于 4MB, FPGA 配置文件的大小随着星载 FPGA 使用的门电路越来越多, 最大已接近 32MB。星载计算机嵌入式软件一般还支持补丁程序的注入运行, 一个小补丁可能小于 240B。软件注入策略与注入软件的大小相关, 数据量很小的补丁软件通过测控慢速上行通道即可实现当圈注入并立即执行; 当数据量很大时, 使用测控系统的高速上行通道实现快速注入可减少测控时间, 受限于总线通信速率和各设备的数据接收能力, 上注软件无法立即送达目的终端, 需暂时缓存在星务存储器中, 软件的分发和执行根据延时指令进行安排。重构软件以数据帧的形式注入到星务管理系统中, 注入帧格式必须兼顾低速通道、高速通道和整星信息流的特点。针对小卫星软件重构系统而设计的数据帧结构如表 2 所示, 帧长为 250B。其中, 首帧

表 2 软件注入数据帧格式

Table 2 Transfer frame of software program data

帧域内容	字段名称	字段长度	内容说明
	帧标识	1	EOH
	帧路由信息	1	XXH: 指示数据块的目的终端编号 高 4bit: 帧序号标志 0001B-首帧 0110B-中间帧 1111B-尾帧
帧头	帧类型	1	低 4bit: 帧存储说明 0000B-本数据块注入到 SRAM 存储区 0001B-本数据块注入到非易失性存储区 0011B-本数据块按地址规则同时注入三个非易失性存储区
	帧有效字节	1	帧数据区内容有效字节
	帧首地址	4	注入软件内容写入的地址
帧数据区	注入软件内容	240	首帧: 4B 运行地址+4B 软件长度+后续为软件文件 中间帧: 软件文件 尾帧: 软件文件+16 B 完整文件校验码 (MD5 码)
帧校验区	帧校验	2	帧数据域内容 CRC16, 初值为 0

中包含本软件的起始运行地址、软件总字节数等重要信息, 尾帧包含软件文件的完整性校验码。同时, 为了保证每帧的正确性, 需要对每帧信息进行校验值计算。

2.2 数据分发设计

对于存储在软件临时存储区的数据, 需要通过 CAN 总线分发到目的终端。整星 CAN 总线受星务计算机调度, 主要用来进行遥测数据的轮询、指令和运行参数的发送以及整星工作状态的广播等工作。受降额使用要求, 星上 CAN 总线实际每秒传输有效应用数据小于 6KB, CAN 总线常规业务每秒占用 3KB, 终端设备的额外数据接收和处理能力分布在每秒 500B ~ 6KB 字节。对于一个 32MB 的载荷 FPGA 配置数据, 如果总线常规业务一直在进行, 则按最快速度分发, 需分发 10900s, 按最慢速度分发, 需分发 65536s, 分发时间较长。为了应对不同数据的数据分发需求, 数据分发采用动态分发的方式进行。动态分发是指利用星载周期性任务的时间间隙进行数据分发控制, 分为快速分发和业务空闲分发, 快速分发时, 整星不提供载荷业务运行能力, 将无需工作的设备断电并仅保持加电设备的关键遥测轮询, 减少常规业务的总线占用时间, 最大化软件数据分发时间; 业务空闲分发时, 不影响卫星正常业务, 星务计算机每秒对总线数据量进行监视, 利用各任务的空闲间隙插入软件数据分发操作。各终端设备接收到非易失性存储区软件注入数据帧后, 立即将其在线编程写入, 对于需要整体断电的载荷设备, 可以在每次工作之后断电或工作到最长限制时间后自动断电, 由星务计算机实现注入的断点续传控制和进度控制; 对于无需断电的设备, 无论何种软件数据注入帧均可连续分发完成。数据分发任务的执行流程如图 3 所示。数据分发和普通遥控通道上行注入帧在总线上传送的格式都一样, 各终端在最小系统模式下和应用软件模式下均能接收该数据, 进行新版本应用程序的接收和本地存储。最小系统软件具有注入重构区软件和非重构区软件的功能权限, 应用软件仅具有注入重构区软件的功能权限。

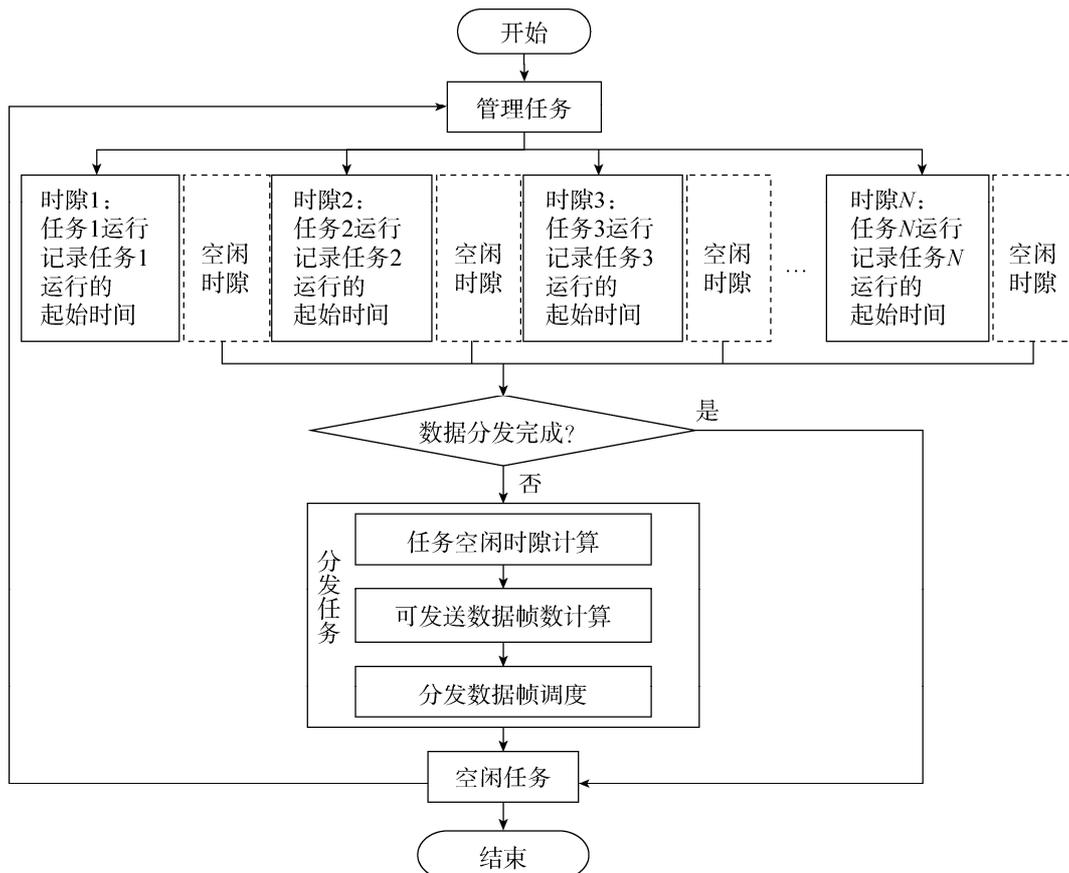


图 3 数据分发流程

Fig. 3 Data distribution flow chart

2.3 软件加载设计

各终端功能设备的最小系统均具有应用软件加载能力,星载计算机嵌入式应用软件也可根据需要设计在应用软件下进行其它版本应用软件的加载。最小系统进行应用软件加载时有自动加载和指令加载两种方式,自动加载通过监测星箭分离状态设置地面研制模式和在轨模式的不同最小系统驻留时长,指令加载是指在最小系统中通过指令重新启动应用软件的加载,星箭分离状态和默认加载存储区标志可通过指令配置写入软件启动配置文件中。当星箭分离状态为未分离时,标识该设备处于地面研制模式,可以在最小系统中自动停驻 30s,等待地面发送最小系统指令进行操控或进行应用软件注入;无驻留指令干预的情况下,最小系统 30s 后即自动引导加载默认加载存储区的应用软件。当星箭分离为已分离时,最小系统无自动停驻功能,启动后即开始引导加载默认存储区的应用软件,应用软件和最小系统软件均有指令接口更改默认星箭分离状态。引导加载时首先从存储区检查软件运行的地址信息和长度信息,然后三取二读取并计算软件文件的 MD5 校验码,通过后将数据搬移到目的 SRAM 地址中运行;当三取二比对不通过或校验不成功时,给出错误类型信息,如果初次引导为重构区软件且未引导成功则自动切换为引导加载非重构分区软件,如果非重构区依然引导加载不成功,则软件自动停留在最小系统中运行,放弃应用软件的引导加载。应用软件在进行其它版本应用软件的引导加载时,读取软件的过程与最小系统软件一致,当引导加载成功后可跳转到新的应用软件起始地址运行,如果引导加载不成功,则继续运行本应用软件。FPGA 配置数据的上注更新不支持运行时读取新的配置数据并进行跳转,只能在最小系统中通过配置状态实现重构区和非重构区配置参数的加载选择。配置数据从 FLASH 中读取到 SRAM 中进行配置时也需进行三取二比对和 MD5 校验码比对,选择过程逻辑与嵌入式软件一样。

2.4 数据差错控制设计

为确保软件程序数据块注入和加载的可靠性、安全性和好用易用性,进行如下差错控制设计。

① 对注入数据的地址连续性进行检查,如果检测出地址不连续,则拒收当前程序块及其它非期望接收的程序块,并将期望接收的块序号作为遥测下传,待地面处理;直至地面发送恢复上注指令,或接收到地址连续且正确的程序块,恢复程序块接收工作;地面从期望接收的块地址开始重新发送程序块,上注继续。

② 对注入数据的数据正确性进行检查,如果校验错,则拒收当前程序块及其它后续程序块,并将校验错的块地址作为遥测下传,等候地面处理;直至地面发送恢复上注指令,或接收到正确且地址连续的程序块,恢复程序块接收工作;地面从校验错的块地址重新发送程序块,上注继续。

③ 具备对程序块存储器进行按扇区校验功能,并能够通过遥测下传存在校验错的扇区号或地址信息,支持按单个扇区擦除及程序块重注。

④ 数据块注入完成后根据地址信息和长度信息自动进行 MD5 码计算,计算通过时才表明完整的程序块已经注入,如果计算不正确,给出遥测显示,等待地面检查;所有的内存数据均可以通过遥测通道下传到地面。

⑤ 数据加载时三取二比对检查,当出现 2 份数据正常一份数据异常时,记录数据的地址并通过遥测下传,提醒地面处理;也可设置禁止/允许类指令实现可重构区存储数据错误时的自动擦除和正确回写的控制,该操作必须按扇区为最小单位进行。

3 试验验证

本文提出的软件在轨重构系统已经在某新技术验证卫星上研制使用,在地面测试期间进行了多次软件更新和容错测试。

3.1 软件重构试验

某新技术验证卫星共有 10 个整星一级 CAN 总线终端节点,其中,6 个为星载微控制器实现软件功能,星务计算机、姿控计算机和相机综合电子计算机均具有 PROM 固化的最小系统软件,且可进行

应用软件的完整更新, 其它 CPU 软件均具有补丁程序的注入执行功能; 4 个为 FPGA 实现产品功能, 数传处理器、实时处理器和星间通信终端均采用最小系统架构, 能实现 FPGA 配置数据的多份存储和 FPGA 在线重配置。在整星测试阶段, 通过高速遥控上行通道进行了多次设备软件更新试验, 如表 3 所示。试验结果表明, 该系统能正确完成整星各设备的软件重构, 灵活性强。

表 3 软件重构试验结果
Table 3 Software reconfiguration test results

设备	应用软件大小	注入时间	分发时间	更新结果
星务计算机	384KB	24s	业务空闲分发 820s	正常
相机综合电子计算机	193KB	12s	业务空闲分发 410s	正常
数传处理器	18MB	1152s	快速分发 9830s	正常
实时处理器	26MB	1664s	快速分发 11360s	正常
星间通信终端	29MB	1856s	快速分发 15840s	正常

3.2 容错试验

软件重构系统的可靠性由星务设备和各终端节点设备进行设计保证。容错试验包括程序数据块上注、分发存储和引导加载三个阶段的内容, 如表 4 所示。试验结果表明, 该系统具有较强的容错能力, 除非硬件失效, 重构软件的程序数据导致的异常均有恢复处理措施, 系统在轨安全性和可靠性高。

表 4 容错试验结果
Table 4 Error control test results

测试项目	测试内容	测试结果
星务计算机程序数据块上注	上注帧校验错误 上注帧地址不连续	停止数据帧写入存储区操作
星务计算机程序数据块分发	通过修改内存数据, 使软件临时缓存区待分发的程序数据块注入帧地址不连续, 或地址连续但最终的程序数据 MD5 码校验不通过	数据分发允许后检测到错误后不进行数据的分发
星务计算机程序数据引导加载	修改星务重构区程序, 使重构区程序数据 MD5 码校验不通过; 在最小系统下修改非重构区内容, 使应用软件重构区和非重构区均校验错误	最小系统自动加载非重构区软件; 无法进行应用软件的加载, 停留在最小系统中
数传处理器程序数据注入存储	上注帧校验错误 上注帧地址不连续	停止数据帧写入存储区操作
数传处理器程序数据引导加载	修改数传重构区程序, 使重构区程序数据 MD5 码校验不通过	最小系统自动加载非重构区软件

4 总 结

本文设计的基于最小系统的小卫星在轨软件重构系统其本质是落实小卫星电子设备的通用功能产品化、专属功能可配置化的要求, 在不降低产品可靠性、安全性和寿命的基础上, 使同一款产品能满足更多不同功能的卫星当前甚至未来可拓展的使用需求。该系统对小卫星整星的快速研制也具有积极意义, 即单机层面上, 将软件存储、引导加载的共同点与设备基本功能相结合, 实现终端设备最小系统的产品固化, 应用功能可在产品的全寿命周期进行持续开发迭代而不影响整星 AIT 进度; 卫星整星层面, 可以在不影响整星常规业务的情况下, 通过测控通道实现不同大小的软件功能重构, 使星载终端设备可适应业务变化、新技术应用及故障修复等需求。

参考文献

- [1] 张然峰, 郝贤鹏, 金龙旭, 等. 空间相机软件在轨重注方法研究与实现[J]. 光机电信息, 2011, 28(6): 30–34.
ZHANG Ranfeng, HAO Xianpeng, JIN Longxu, et al. Study and realization on method of software in space camera on-board reprogramming[J]. OME Information, 2011, 28(6): 30–34.
- [2] 安军社, 刘艳秋, 孙辉先. 软件的动态维护与实现[J]. 计算机工程, 2003, 29(2): 238–239.
AN Junshe, LIU Yanqiu, SUN Huixian. Implementation of on-board software maintenance[J]. Computer Engineering, 2003, 29(2): 238–239.
- [3] 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, 1: 7–13.
WANG Zhanqiang, ZHAI Shenghua. Research on software on-orbit reconfiguration of space-borne processing equipment[J]. Space Electronic Technology, 2013, 1: 7–13.
- [4] 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 on-orbit reloading for FPGA software in satelliteborne environment[J]. Modern Electronics Technique, 2012, 35(9): 87–90.
- [5] 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.
- [6] 李雁斌, 吉峰, 黄勇. 一种星载 DSP 软件的在轨编程方法[J]. 制导与引信, 2011, 32(4): 37–41.
LI Yanbin, JI Feng, HUANG Yong. The design of on-board programming for satellite DSP software[J]. Guidance & Fuze, 2011, 32(4): 37–41.
- [7] 白照广. 中国现代小卫星发展成就与展望[J]. 航天器工程, 2019, 28(2): 1–8.
Bai Zhaoguang. Development achievements and prospects of China modern small satellite[J]. Spacecraft Engineering, 2019, 28(2): 1–8.
- [8] 张科科, 朱振才, 夏磊. 小卫星模块化设计技术分析[J]. 航天器工程, 2015, 24(6): 107–115.
ZHANG Keke, ZHU Zhencai, XIA Lei. Analysis of small satellite modular design technology[J]. Spacecraft Engineering, 2015, 24(6): 107–115.
- [9] PETRICK D, ESPINOSA D, RIPLEY R, et al. Adapting the reconfigurable SpaceCube processing system for multiple mission applications[C]//Proceedings of Aerospace Conference, New York, IEEE, 2014: 1–4.
- [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, 5: 32–35.
Wei Yongquan, Dong Zhenhui, Zhang Hongjun. Method of embedded onboard software upgrading based on file[J]. Microcontrollers & Embedded Systems, 2018, 5: 32–35.

[作者简介]

熊浩伦 1987年生, 硕士, 工程师, 主要研究方向为小卫星综合电子技术、星载计算机软件技术。

闫国瑞 1982年生, 硕士, 高级工程师, 主要研究方向为小卫星综合电子技术、小卫星软件系统设计。

李国军 1983年生, 硕士, 高级工程师, 主要研究方向为小卫星综合电子技术、星载计算机软件技术。

吕 达 1980年生, 硕士, 高级工程师, 主要研究方向为小卫星综合电子技术。