

一种星载软件重构系统的设计与实现

刘伟伟, 李欣, 于俊慧, 穆强, 庞亚龙

(北京空间飞行器总体设计部, 北京 100094)

摘要: 针对航天器寿命延长与功能复杂度提升带来的对软件在轨重构和升级的迫切需求, 基于 SpaceWire 设计了一种对星载软件升级与重构进行统一管理的网络, 并设计实现了一款通用的软件维护控制器; 星载软件重构网络通过 SpaceWire 路由器连接大容量存储模块以及星载计算机、载荷处理器等需要进行软件重构的终端设备, 大容量存储器作为共享的存储资源池, 用于接收各类星载设备的数据和升级软件程序, 直接将软件程序通过 SpaceWire 分发到相应的设备, 软件维护控制器位于终端设备内部, 对软件重构数据进行统一格式封装, 支持软件升级数据的片段化更新、软件 ECC 编码存储、软件 APP 切换控制等功能, 解决星载软件升级缺乏统一操作流程、缺乏通用通信协议、缺乏高可靠设计支持的困境, 并且该控制器采用抗辐射加固 ASIC 设计, 满足宇航使用环境要求, 为实现星载软件的灵活、便捷在轨重构奠定基础。

关键词: SpaceWire; 星载软件; 星载计算机; 大容量存储; 软件重构; 软件维护控制器

Design and Implementation of A Spaceborne Software Reconfiguration System

LIU Weiwei, LI Xin, YU Junhui, MU Qiang, PANG Yalong

(Beijing Institute of Spacecraft System Engineering, Beijing 100094, China)

Abstract: In response to the urgent demands for software reconfiguration and upgrade in orbit, it is brought about by the extended spacecraft life and increased functional complexity, on the basis of SpaceWire, a unified management network for the on-board software upgrade and reconfiguration is designed, and a universal software maintenance controller is designed and implemented; the on-board software reconfiguration network connects mass storage modules, on-board computers, payload processors and other software reconfiguration terminal devices through SpaceWire routers. The mass storage is used as a shared pool of storage resources to receive the reconfiguration and upgrade the software program from various types of on-board devices, and can directly distribute the software programs to the corresponding devices via the SpaceWire. The software maintenance controller is located inside the terminal device, encapsulates the software reconfiguration data in a unified format, and supports the functions of fragmented update for software upgrade data, software error correcting code (ECC) storage, software application (APP) switching control, then solves the troubles of lacking unified operation process, lacking common communication protocol and lacking high reliable design support for on-board software upgrade, and the controller is designed with the radiation-hardened application specific integrated circuit (ASIC). It meets the requirements of space environment, which lays a foundation for the flexible and convenient on-orbit reconfiguration of on-board software.

Keywords: SpaceWire; on-board software; on-board computer; mass storage; software reconfiguration; software maintenance controller

0 引言

软件是航天器电子系统的重要组成部分, 也是实现航天器功能与人器交互的控制载体与信息处理中枢。随着航天技术的不断发展, 一方面航天器的运行寿命不断延长, 所承担的任务需求也不断演变, 另一方面星载软件的功能复杂度越来越高, 地面测试期间难以避免无法发现深层次隐藏的 bug, 在这种情况下, 对星载软件进行在轨重构、升级、更

新将成为航天器电子设备必不可少的功能之一^[1-3]。

为了快速、便捷、可靠地实现星载软件在轨重构与升级, 首先需要解决 3 方面的问题。第一, 星载软件的复杂度越来越大, 相应的软件规模也越来越大, 尤其是大规模 FPGA 的应用越来越广泛, 其软件规模已经从最初的几 Mbit 提升为几百 Mbit, 这种情况下无论是对卫星-地面之间的通信速率, 还是航天器内部设备与设备之间的传输速率都提出了较高的要求^[4-7]。第二, 传统软件设计大多针对

收稿日期: 2022-09-20; 修回日期: 2022-10-24。

作者简介: 刘伟伟(1985-), 男, 山东东营人, 高级工程师, 主要从事航天器电子技术及高性能计算技术方向的研究。

引用格式: 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]. 计算机测量与控制, 2023, 31(5): 207-213.

CPU 内存中的软件进行更新, 存在重启后升级软件丢失的缺陷, 为了支持重构与升级软件的固化, 需要采用可重复编程存储器(如 EEPROM 或 FLASH)实现星载软件的存储, 但是航天器在轨运行面临着空间辐射环境的影响, 容易导致存储器中存储的星载软件被改写, 进而出现星载软件无法正常启动与运行的故障^[8-11], 因此, 保证星载软件的容错存储也是需要解决的一个关键环节。第三, 航天器中的电子设备众多, 这些设备大多由不同的承制商进行研制, 不同厂商对于星载软件的重构和升级处理机制不尽一致, 为了支持不同处理机制下的星载软件重构与升级, 不得不为每一类设备设置专用的接口和处理系统, 进一步提升了航天器系统的复杂度, 也更加容易地引入故障点和设计的薄弱环节, 因此, 对星载软件构建统一的在轨重构与升级机制, 采用统一的传输网络与接口对于保证星载软件重构与升级的便捷性和可靠性具有至关重要的作用。

除此之外, 传统航天器大多采用 1553B、CAN 等低速串行总线实现星载数据传输与处理^[12]。但是, 面对着航天器的高精度姿态控制、高分辨率遥感影像的计算处理、高通量星间网络的路由管理等新的复杂需求, 航天器在轨信息处理数据量急剧增大, 传统的串行通信总线已难以支撑数据传输速率的需求, 单一计算机的集中管理模式已难以满足数据计算量的需求, 在这种情况下, 航天器电子网络互联技术已经逐步从传统的低速子系统间总线互联逐渐向高速交换式系统网络方向演变, 由多台计算机分别实现不同功能的独立处理, 由交换式的网络实现计算机之间的信息交互与融合^[13]。同时, 凭借多个计算机的独立智能工作能力, 除了可通过高速交换网络实现并行计算、协同处理之外, 也为故障情况下的任务迁移与系统重组奠定了基础, 进一步提升了航天器的可靠性。

在交换网络的选择中, SpaceWire 凭借全双工通信、通信协议简单、链路速率高(2~400 Mb/s)、容错能力强、网络拓扑灵活的特点^[14-16], 已经有越来越多的航天器选择采用 SpaceWire 作为数据通信的骨干网络。其中, 国内高分辨率遥感卫星以及风云系列气象卫星中均采用了 SpaceWire 网络, ESA 与 NASA 的多个空间科学探测器中也大量采用了 SpaceWire 作为数据传输网络^[17-18]。

在这种背景下, 本文以 SpaceWire 高速交换网络作为基础, 构建了面向星载软件重构的管控网络, 并在此基础上研制了一款软件维护控制器, 可以实现星载软件的在线维修、升级、检错、纠错、APP 加载/切换等功能, 并针对 FPGA 在空间辐射环境中应用的薄弱环节, 基于软件维护控制器实现对其 SEU、SEFI 等故障的检测和恢复, 形成了覆盖星载软件在轨重构与升级全过程的完整解决方案。

1 星载软件重构网络系统

1.1 基于 SpaceWire 的星载软件重构网络架构

星载软件重构网络架构如图 1 所示, 主要由 SpaceWire 网络、串行数据总线、数管计算机以及其他计算机、传感

器和执行机构等终端设备组成。

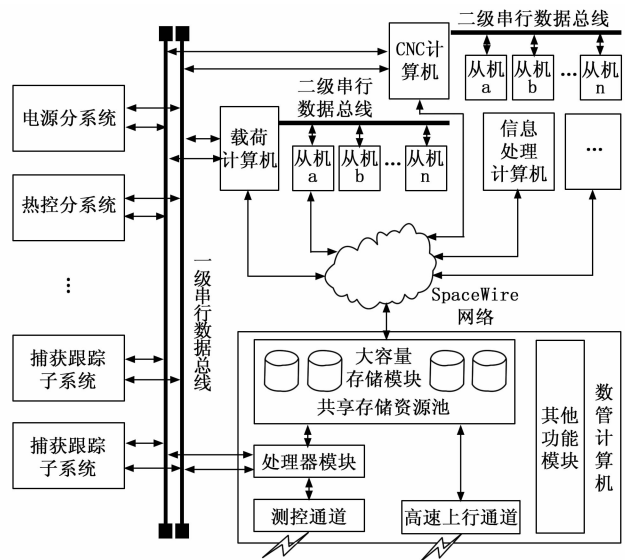


图 1 星载软件重构网络架构

SpaceWire 作为分布式计算机设备之间的高速数据传输骨干网络, 可以实现小型载荷数据、信息处理数据、软件升级数据等信息的传输与交互^[3]。

数管计算机作为航天器与地面进行数据交互以及航天器内部信息处理的中枢, 实现对维护软件以及升级软件的统一管理。数管计算机采用硬件模块化设计模式, 内部集成遥测遥控模块、处理器模块、指令驱动模块、大容量存储模块等模块化产品, 可根据任务需求实现模块化功能扩展。其中数管计算机内部的大容量存储模块作为整个航天器平台的共享存储资源池, 实现对小载荷数据、航天器运行日志数据、软件 APP 等数据的存储, 该大容量存储模块可以通过高速上行通道直接接收地面站上注的控制指令以及软件升级数据。数管计算机内部的处理器模块作为 SpaceWire 网络以及串行数据总线的主控计算机, 完成 SpaceWire 路由表维护、任务规划、健康管理、能源管理等功能, 同时, 处理器模块也负责通过测控通道与地面站之间进行通信, 将航天器的工作状态信息下传地面站, 接收地面站上注的控制指令以及软件升级数据等信息。

其他计算机以及传感器与执行机构作为终端设备, 通过 SpaceWire 网络或者串行数据总线与数管计算机进行通信, 一方面接收数管计算机发送的控制指令, 并向数管计算机返回本计算机的工作状态, 另一方面, 接收数管计算机分发的星载软件重构及升级数据, 完成本计算机内部 CPU、FPGA 等器件软件的升级与重构。

1.2 星载共享存储资源池

在星载软件重构网络中, 大容量存储模块作为整个航天器平台的共享存储资源池对星载软件的重构与升级具有至关重要的作用。大容量存储模块既是软件重构与升级数据的存储中心, 也是进行软件分发的中转中心, 因此, 为了提升不同种类数据在大容量存储模块中进行存储的好用

性与易用性, 并且保证各类数据之间的相互隔离以提升存储的可靠性, 在大容量存储模块中设计实现了一种混合文件系统, 可实现静态文件与动态文件两种不同类型文件的管理^[3], 如图 2 所示。

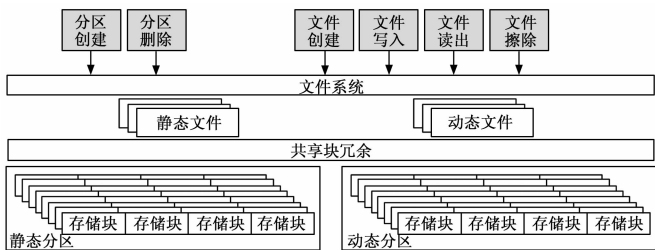


图 2 混合文件系统结构框图

混合文件系统基于 NAND Flash 的并行以及流水设计, 具有完备的容错机制, 对存储数据进行纠错编码, Flash 存储块的坏块自动替换等。其中, 静态文件支持循环写入、边擦边记, 存储容量在轨可调整, 用于长期日志数据记录; 动态文件根据在轨存储需求实时创建, 用于注入程序、环境监测数据的记录, 容量根据写入数据量需求自动追加, 使用完毕后自动删除。

同时, 针对大容量存储模块还设计了统一的访问控制指令集, 实现所有文件统一访问接口, 屏蔽了 NAND FLASH 的底层访问细节, 大幅提升了用户对大容量存储模块访问的便捷性。所设计的访问指令表 1 所示。

表 1 存储器访问指令集

指令名称	参数
创建	file number, attribute
打开	file number
关闭	file number
读	Logic address, file number
写	Logic address, file number, data
回放	Logic address, file number, destination address, rate
停止回放	file number
删除	file number
清空	file number

1.3 软件维护机制

当需要对航天器进行软件重构或升级更新时, 地面站可通过测控信道将软件发送至航天器数管计算机, 由数管计算机中的处理器模块进行接收以及完整性校验后, 直接通过串行数据总线分发给相应的终端计算机, 或者通过数管计算机内部硬件模块之间的通信总线传送给大容量存储模块, 由大容量存储模块通过 SpaceWire 网络进行分发。同时, 地面站也可通过高速传输信道直接将软件升级数据发送至大容量存储模块, 由大容量模块接收并进行完整性校验, 将接收到的软件升级数据存储到统一的存储资源池中, 之后在合适的时机或根据控制指令的指示通过 SpaceWire 网络进行分发。

除此之外, 在各个终端设备中配置软件维护控制器, 软件维护控制器能够接收串行数据总线或者 SpaceWire 网络上分发的软件升级数据, 并将接收到的数据进行正确性校验后写入本地的存储器中。

2 软件维护控制器的设计

软件维护控制器位于各个终端设备中, 能够在终端设备正常运行过程中, 接收重构或升级的软件数据, 完成软件的在线维修、升级、检错、纠错等功能, 实现对星载软件的无感重构, 还可完成对 CPU、DSP、FPGA 等器件的软件 APP 选择、启动以及加载配置管理等功能。并且采用标准化的格式实现对软件维护数据的包装, 实现不同设备和软件维护信息接口的统一化, 简化了软件在线维护操作复杂度, 提升航天器电子系统软件在轨维护的效率和可靠性。

2.1 软件维护控制器组成

软件维护控制器主要由网络通信接口单元、CPU 接口单元、FPGA 接口单元 (包括 SEFI 检测单元、FPGA 动态刷新单元、FPGA 定时回读单元、FPGA 加载配置单元)、存储器接口单元、控制寄存器以及状态寄存器等几部分组成, 如图 3 所示。

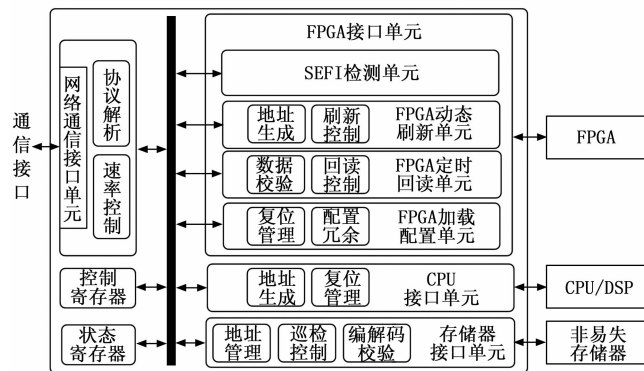


图 3 软件维护控制器组成框图

网络通信接口单元是软件维护控制器接收控制指令、接收软件升级数据、返回工作状态的通信接口, 是软件维护控制器融入星载软件重构网络的桥梁, 为此, 软件维护控制器提供了 SpaceWire、UART、SPI 接口, 通过 SpaceWire 接口可使软件维护控制器直接接入星载软件重构网络, 通过 UART、SPI 接口可使软件维护控制器通过桥接电路接入 SpaceWire 网络或者 1553B、CAN 等串行数据总线。网络通信接口的连接关系如图 4 所示。

网络通信接口单元提供的 SpaceWire 网络接口使得软件维护控制器可以直接接入整个航天器的 SpaceWire 网络中, 直接从 SpaceWire 网络以及大容量存储模块中接收软件升级数据, 直接在数管计算机的控制下完成软件的升级与重构等维护操作。网络通信接口单元提供的 UART、SPI 接口使得软件维护控制器可直接与计算机内部的处理器进行通信, 此时, 各个终端计算机中的处理器负责与整个航天器的 SpaceWire 网络以及一级串行数据总线进行通信, 处理器完

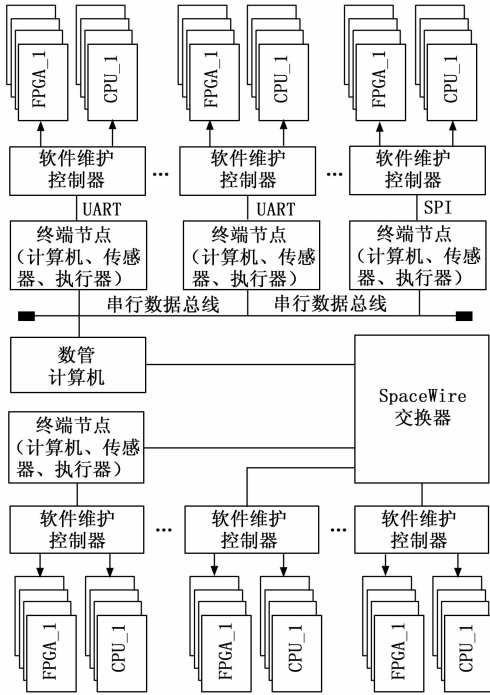


图 4 软件维护控制器网络通信接口连接关系示意图

成软件升级数据的接收后通过 UART 或 SPI 接口转发给软件维护控制器，这种连接方式一方面可以降低对 SpaceWire 交换器接口数量的需求，另一方面可以在只有 1553B 以及 CAN 总线接口的传统设备中使用，保证了软件控制器具有更加灵活的接入手段和更加广泛的应用环境。

FPGA 接口单元负责完成与 SRAM 型 FPGA 之间的通信，完成对 SRAM 型 FPGA 的软件加载配置，同时，为了提升 SRAM 型 FPGA 在空间辐照环境下的适应能力，避免单粒子效应对 FPGA 功能的影响，在 FPGA 接口单元中集成了 SEFI 检测单元以及 FPGA 动态刷新和 FPGA 定时回读单元，其中 SEFI 检测单元用于对 SRAM 型 FPGA 的单粒子功能中断进行检测，定时回读单元用于对 FPGA 已加载的软件位流进行回读来判断是否发生单粒子翻转，动态刷新单元用于对 FPGA 软件配置存储区重复写入软件位流，实现对单粒子翻转的纠正。

CPU 接口单元负责完成对 CPU/DSP 软件的加载，加载控制通过对 CPU/DSP 的复位控制以及 SPI 接口通信实现，在复位完成后，在 CPU/DSP 和存储器之间建立 SPI 透传通路，实现 CPU/DSP 对存储器中软件数据的访问和读取加载。

存储器接口单元实现对非易失存储器的控制，包括对非易失存储器的控制包括擦除、编程、读取等功能。同时，对非易失存储器支持分区管理模式，在一个分区内存储一份独立的软件数据，通过网络通信接口单元接收到的指令控制对其中一个分区的操作，以及从哪一个分区中读取软件数据，并将其加载到 CPU、DSP 或者 FPGA 中，从而实现多个软件 APP 的选择与管理，提升航天器对多任务处

理以及软件定义能力的支持。

2.2 软件维护控制器工作模式

由于软件维护控制器支持对多个器件的软件加载以及非易失存储器的管理，具有多种不同的工作模式，为了避免多个工作模式之间的冲突，将不同的工作模式按照先后顺序结合优先级的方式进行设计，如图 5 所示。

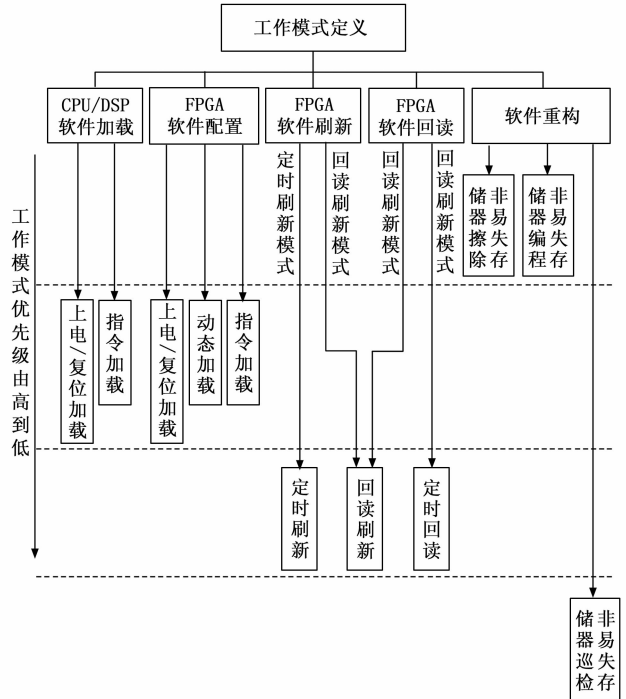


图 5 软件维护控制器工作模式

处于最高优先级的为对非易失存储器的擦除以及编程操作，在此期间不会对 FPGA 以及 CPU/DSP 进行任何操作，直到对存储器完成擦除以及写入新的软件数据为止。

处于第二优先级的为对 FPGA 以及 CPU/DSP 的软件加载。对 CPU/DSP 的软件加载支持上电/复位后的自动加载以及通过指令控制的重加载。对 FPGA 的软件加载除了支持上电/复位后的自动加载以及指令控制的重加载之外，还支持动态加载，也就是在对 FPGA 进行 SEFI 检测过程中，发现单粒子功能中断故障后就会自动对 FPGA 进行重加载，达到及时消除故障的目的。

处于第三优先级的为对 FPGA 的定时回读、定时刷新以及回读刷新操作。由于 SRAM 型 FPGA 在空间辐照环境中应用时，一旦 FPGA 内部的配置存储器发生单粒子翻转将有可能导致 FPGA 的功能紊乱^[19-21]，为此，软件维护控制器可通过回读方式实现对 FPGA 中软件数据是否存在单粒子翻转故障进行检测，通过刷新方式实现对单粒子翻转故障的恢复，支撑 SRAM 型 FPGA 在航天器中的高可靠应用。其中定时刷新为周期性的无条件对 FPGA 内部存储的软件数据进行重复性写入覆盖，回读刷新是只有在回读检测到单粒子翻转故障后才对 FPGA 内部存储的软件数据进行写入覆盖。对 FPGA 进行加载配置、SEFI 检测以及回读

与刷新操作流程如图 6 所示。

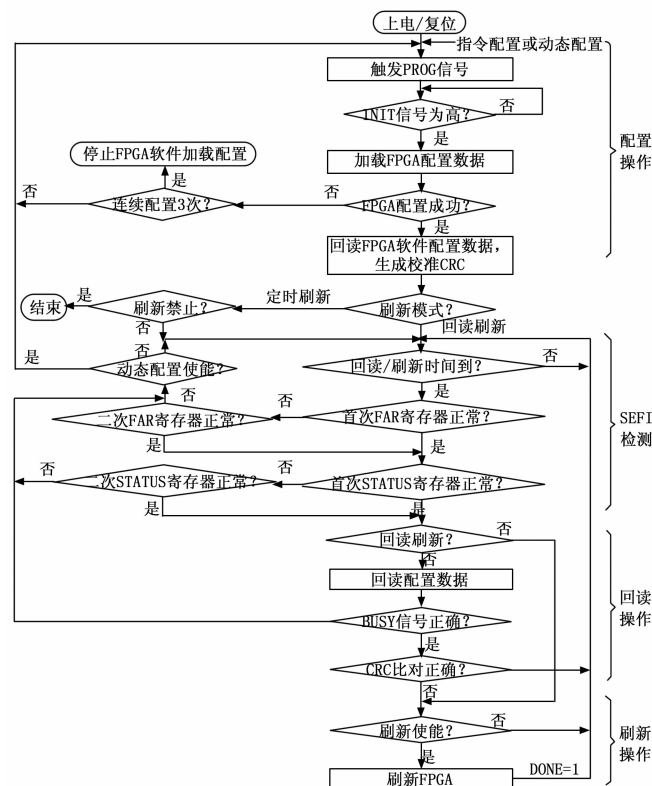


图 6 FPGA 配置及回读刷新操作流程

为了保证对 FPGA 软件加载的可靠性, 在对其进行加载时采取连续 3 次加载的方式, 即在一次加载失败后会自动进行第二次加载, 直到连续 3 次加载均失败时才认为对 FPGA 的加载失败, 避免瞬时故障或干扰造成加载失败的影响。对 FPGA 的 SEFI 检测支持对 FPGA 的 POR、SMAP、FAR 以及全局信号的单粒子功能中断故障检测, 同样, 为了避免瞬时故障或干扰造成的影响, 对每种类型的 SEFI 检测也都采取连续 2 次检测的方式, 只有 2 次均故障的情况下才认为发生 SEFI 故障。

对 FPGA 的回读周期和刷新周期可通过寄存器进行配置, 如公式 (1) 所示。

$$2^{(2 \times n)} \times 2^{20} \times \frac{1}{f_{clk}} \quad (1)$$

其中: n 可设置为 0~15 之间的任意值, 由控制寄存器进行设置。

处于第四优先级的为对非易失存储器的巡检操作, 主要用于在软件维护控制器的空闲时间段内对非易失存储器中的数据内容进行错误检测和纠正。

2.3 软件维护控制器可靠性增强及优化设计

软件维护控制器的使用方式对于星载软件重构与升级的好用性和易用性具有重要的作用, 为此, 在软件维护控制器设计中进行了针对性的优化设计, 主要集中在以下几个方面。

1) 引入通用化设计措施: 航天器电子设备中所采用的

CPU、DSP、FPGA 等器件的种类繁多, 不同类型的器件对软件加载的要求也不尽一致, 在这种情况下, 软件维护控制器将内部的控制寄存器在外部存储器中进行了完整映射, 允许用户对存储器中映射寄存器预存控制信息 (预存控制信息更新的方式与软件重构与升级方式一致), 实现软件维护控制器满足不同器件的软件加载要求功能。控制寄存器映射关系如图 7 所示。

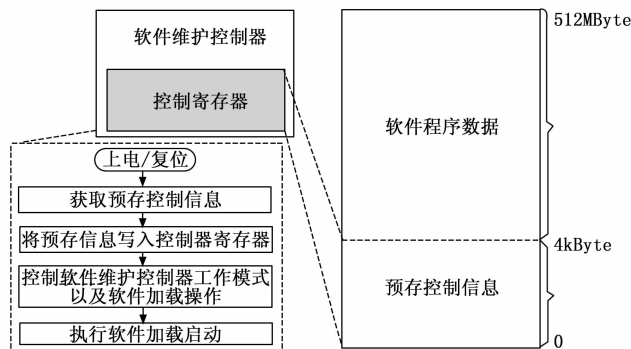


图 7 寄存器映射关系框图

2) 引入软件压缩设计: 软件规模越来越大, 尤其是大规模 FPGA 器件的配置码流更是高达几百 Mbit, 在星载软件重构和升级过程中对卫星一地面通信带宽提出了很高的要求, 为此, 软件维护控制器实现对软件的压缩设计, 降低上注软件规模。

3) 引入片段化更新设计: 有些时候星载软件并不需要对整个软件进行重构与升级, 而是只需更新其中的一小部分, 另外, 在星载软件重构与升级过程中, 可能会出现某一部分数据错误的情况, 需要重新上注, 因此软件维护控制器实现片段化更新, 实现星载软件数据包的中断续传和乱序注入, 避免对整个软件进行二次上注的操作。

4) 引入纠错编码设计: 为了实现星载软件的重构与升级, 需要采用可重复编程的非易失存储器, 使得存储器中的数据会受到空间辐射环境的影响, 为了提升软件存储的可靠性, 软件维护控制器通过 BCH 实现对存储数据的纠错编码, 可实现在线的错误检测和错误纠正。

5) 引入存储器巡检设计: 星载软件一般仅在启动的时刻对存储器进行读取实现软件加载, 之后会有很长一段时间不再对存储器进行访问, 这时候即使对存储器中的数据进行了 BCH 编码也有可能出现 SEU 积累, 一旦积累错误超出了 BCH 纠错能力, 将造成无法对错误的数据进行纠正, 为此, 软件维护控制器实现对存储器的周期性遍历, 及时检测存储器中存在的错误, 并且一旦检测到发生错误, 及时通过 BCH 进行纠正, 并将纠正后的数据重新回写到非易失存储器中。

6) 引入完整性校验设计: 软件维护控制器接收软件升级或重构等维护数据时是按照数据帧的方式进行接收和存储, 一旦在软件升级或重构数据在传输过程中丢失或发生

错误, 导致软件维护控制器在写入非易失存储器过程中发生软件数据丢失或错误, 将导致后续 FPGA 或 CPU/DSP 等无法正常加载启动, 因此软件维护控制器通过设置软件数据整体 CRC 校验方式实现对软件数据完整性的校验, 只有在通过完整性校验的情况下才会执行后续的软件加载以及 FPGA 回读、刷新等操作, 保证不出现设备无法正常启动的风险和故障模式。

2.4 通信协议

软件维护控制器的网络通信接口提供了多种类型的通信接口, 为了降低用户使用的复杂度, 多个网络通信接口之间采用相同的通信协议。其中 UART 和 SPI 接口的数据包格式完全一致, SpaceWire 接口在 UART 和 SPI 数据包的基础上进一步打包成 SpaceWire 数据包 (UART 和 SPI 接口数据包格式作为 SpaceWire 数据包的 cargo)。详细的数据包格式如图 8 所示。

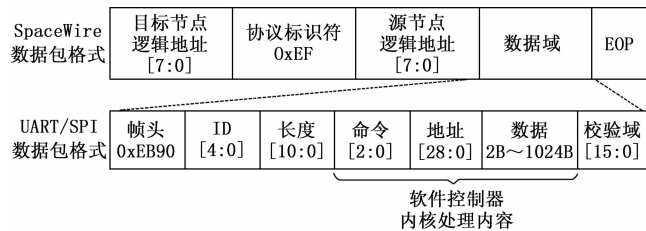


图 8 数据包格式

SpaceWire 数据包格式中, 各部分的功能说明如下:

- 1) 目标节点逻辑地址用于表明软件维护控制器在 SpaceWire 网络中的路由地址, 用于数管计算机向软件维护控制器发送控制指令以及软件升级数据;
- 2) 源节点逻辑地址用于表明数管计算机在 SpaceWire 网络中的路由地址, 用于软件维护控制器向数管计算机返回工作状态和操作结果等信息;
- 3) 协议标识符固定为 0xFE, 代表 SpaceWire 数据包格式遵循 SpaceWire 标准中指定的串行传输通用协议;
- 4) 数据域中的内容为完整的 UART/SPI 数据包格式规定的格式;
- 5) EOP 为 SpaceWire 数据包的结尾标识。

UART/SPI 数据包格式中, 各部分功能说明如下:

- 1) 帧头固定为 0xEB90;
- 2) ID 域代表软件维护控制器的节点号, 当 UART 接口采用 RS485 总线连接方式以及 SPI 接口采用总线连接方式, 并且在总线上有多个软件维护控制器时, 该节点号用于指定需要操作的软件维护控制器;
- 3) 长度域代表数据包中数据域的字节数;
- 4) 命令域代表要对软件维护控制器进行的操作, 如写寄存器、读寄存器等;
- 5) 地址域代表要对软件维护控制器进行操作的寄存器或者存储器地址;
- 6) 数据域代表要对软件维护控制器中寄存器或存储器

进行操作的有效数据;

- 7) 校验域代表从帧头开始到数据域的异或校验值。

3 测试与验证

在星载软件重构系统中, 软件维护控制器是实现软件可靠重构以及统一重构机制的关键, 因此测试验证首先针对软件维护控制器的功能进行测试验证, 并在此基础上开展航天器工程应用验证。

3.1 软件维护控制器测试验证

软件维护控制器支持对 CPU/DSP 以及 FPGA 等器件的软件维护。其中对 CPU/DSP 器件的软件维护以 TMS320C6678 系列 DSP 作为软件维护对象, 通过 SPI 接口作为 DSP 软件加载接口进行测试, 测试中软件维护控制器能够正确接收来自 UART/SPI 接口的软件维护数据帧, 并将其写入非易失存储器中, 之后通过指令控制在 DSP 与存储器之间建立 SPI 透传链路, 并通过复位控制实现对 DSP 软件的加载启动。

软件维护控制器对 FPGA 器件的软件维护以 Xilinx 公司的 Virtex、Virtex-II、Virtex-4、Virtex-5、Virtex-7、Kintex-7 等系列的 FPGA 作为软件维护对应进行测试, 测试中将不同系列的 FPGA 作为独立的测试子板分别与软件维护控制器进行连接, 软件维护控制器能够自主识别所连接的 FPGA 器件, 并根据识别的器件结果完成对 FPGA 的软件加载以及回读和刷新等操作, 在对 FPGA 进行回读与刷新操作器件, FPGA 工作正常。

除此之外, 在软件维护控制器外部分别配置 FLASH 以及 PROM 存储器情况下, 针对软件维护控制器引入的软件压缩、片段化更新、纠错编码、完整性校验等相应的功能进行了专项测试, 测试过程中各项功能正常。

表 2 软件维护控制器软件压缩及纠错编码测试结果

存储器	PROM	FLASH
软件位流		
源码	支持	支持
用户自定义源码	支持	支持
BCH 码	不进行 BCH 编码, 按照原码输出	支持
用户自定义 BCH 码	不进行 BCH 编码, 按自定义原码输出	支持
压缩码	支持	支持
用户自定义压缩码	支持	支持
BCH 压缩码	不进行 BCH 编码, 按自定义压缩码输出	支持
用户自定义 BCH 压缩码	不进行 BCH 编码, 按压缩码输出	支持

软件维护控制器是面向空间辐照环境中运行的航天器进行软件维护, 其自身的可靠性和空间环境适应性对于保证航天器在轨的软件可靠重构具有重要的作用。为此, 针对软件维护控制器的抗辐射能力进行了专项测试验证。利

用中国原子能科学研究院 HI-13 串列加速器和中科院近代物理研究所 HIRL 回旋加速器对 FPGA 维护控制器进行了单粒子效应试验, 利用钴-60 γ 射线源对 FPGA 维护控制器进行了电离总剂量辐照试验, 通过试验结果表明, 软件维护控制器的抗电离总剂量能力大于 100 krad (si), SEU 阈值大于 37.3 MeV · cm²/mg, SEL 阈值大于 81.35 MeV · cm²/mg, 满足宇航空间辐射环境下的应用需求。

3.2 航天器应用验证

基于 SpaceWire 的星载软件重构网络基于星形拓扑搭建, 并在高分辨率遥感卫星中开展工程应用, 如图 9 所示。

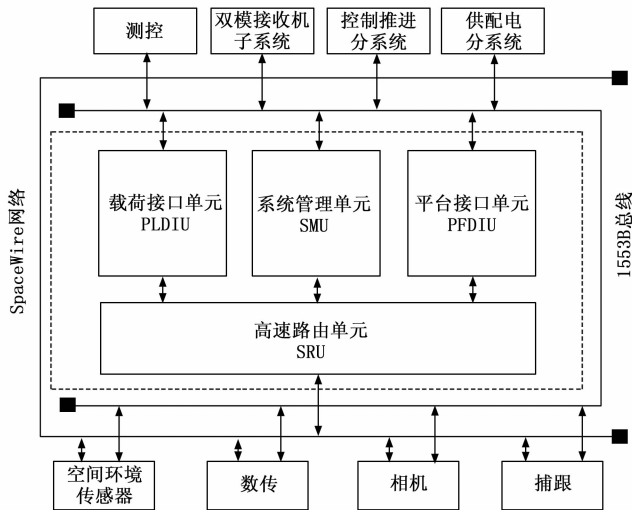


图9 星载软件重构网络连接示意图

在 SMU 中配置大容量存储模块, 在 SpaceWire 以及 1553B 总线的各个终端节点设备中内部配置软件维护控制器实现对 DSP 以及 FPGA 器件的软件在线升级管理, 通过软件压缩, 可将 28 Mbyte 的 FPGA 码流压缩为 4 Mbyte, 压缩率达到 80% 以上。SpaceWire 网络采用 200Mbps 链路速率, 传输任务数据以及软件升级数据, 软件升级数据传输时间由 30 分钟缩减为 10 秒钟。

4 结束语

本文以 SpaceWire 为基础实现了航天器升级的管控网络, 通过 SpaceWire 网络中的大容量存储实现了星载软件升级的集中存储, 通过软件维护控制器实现了对 CPU/DSP/FPGA 等各类器件软件加载、重构、检错、纠错、APP 选择等功能的就近管理与分布式控制, 并支持对传统 1553B 以及 CAN 总线等航天器设备的支持, 使星载软件升级网络成为能够紧密协作, 用户无感的虚拟化一体网络, 实现了整个航天器软件升级接口和操作方式的统一, 改善了星载软件升级方式不一致带来的操作复杂弊端, 大幅提升了软件升级的效率和灵活度。

参考文献:

[1] 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, 15 (5): 32-35.

- [2] 胡萍, 刘世佳, 马宏业. 基于 1553B 总线的星载嵌入式软件在轨升级方法 [J]. 遥测遥控, 2022, 43 (1): 66-69.
- [3] 穆强, 牛跃华, 于俊慧, 等. 应用 SpaceWire 网络的航天器分布式存储系统设计 [J]. 航天器工程, 2019, 28 (2): 78-84.
- [4] 吴军伟, 梁涛涛, 秦代成, 等. 一种 GNSS 信号捕获中的比特压缩与重构方法 [J]. 太赫兹科学与电子信息学报, 2014, 12 (5): 687-701.
- [5] 伍卫国, 王超辉, 王今雨, 等. 基于混合编码的 FPGA 系统配置文件压缩算法 [J]. 计算机研究与发展, 2018, 55 (5): 1049-1064.
- [6] 刘仕东. 一种通用 FPGA 配置数据流压缩与解压缩系统的研究 [D]. 哈尔滨: 哈尔滨工业大学, 2011.
- [7] 徐勇, 李珂, 冯国平, 等. 一种 FPGA 在轨重构配置数据压缩算法 [J]. 航天器工程, 2015, 24 (6): 75-78.
- [8] 袁素春, 滕泽旭, 邵应昭. 一种低成本高可靠 FPGA 在轨可重构加载管理方案 [J]. 空间电子技术, 2017, 14 (3): 92-96.
- [9] 张秀宁, 刘斌, 李澎, 等. 基于反熔丝型 FPGA 的有效载荷可重构技术 [J]. 遥测遥控, 2021, 42 (6): 65-70.
- [10] 王浩波, 郭金鹏. 星载环境下可重构技术分析 [J]. 无线电工程, 2012, 42 (1): 40-42.
- [11] 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.
- [12] 于俊慧, 穆强, 牛跃华, 等. 基于 1553B 和 SpaceWire 组合网络的星载数据管理软件设计 [J]. 航天器工程, 2021, 42 (4): 53-59.
- [13] NIU YUEHUA, LIU WEIWEI, WANG LUYUAN, et al. Distributed Storage System for Satellite Platform Based on SpaceWire Network [R]. Yokohama, International SpaceWire Conference, 2016.
- [14] 龚泉铭, 闫梦婷, 姜秀杰, 等. SpaceWire 网络热点通信模式的缓存资源分配算法 [J]. 国防科技大学学报, 2018, 40 (2): 41-47.
- [15] 姚睿, 王美群, 吴军, 等. 使用 FPGA 设计高可靠 SpaceWire 路由器 [J]. 2019, 41 (4): 86-93.
- [16] 冯国平, 周东, 牛跃华. 航天器通用 SpaceWire 总线路由单元的研究与实现 [J]. 计算机测量与控制, 2019, 27 (1): 241-249.
- [17] 田堂胜, 李陆, 文星, 等. 基于 SpaceWire 总线的 γ 射线监视器数据管理模块设计 [J]. 核技术, 2018, 41 (4): 1-5.
- [18] 牛跃华, 赵文彦. 一种星载强容错 SpaceWire 总线网络拓扑设计与分析 [J]. 计算机应用, 2014, 34 (9): 2497-2500.
- [19] 吴珊, 周学功, 王伶俐. 基于 SRAM 型 FPGA 单粒子效应的故障传播模型 [J]. 电子学报, 2017, 45 (8): 1976-1983.
- [20] 郑晓云, 王绍举. SRAM 型 FPGA 单粒子翻转模拟系统研究 [J]. 红外与激光工程, 2014, 43 (S): 164-168.
- [21] 冯兴, 王大鸣, 张彦奎, 等. 基于 SRAM 型 FPGA 的 SEU 敏感型研究 [J]. 电子技术应用, 42 (5): 53-56.