

“嫦娥五号”载荷数据存储管理方法

刘攀^{1,2}, 王志国^{1,2}, 刘建峰^{1,2}, 闵康磊^{1,2}, 游红俊^{1,2}

(1. 上海航天电子技术研究所, 上海 201109; 2. 智能计算技术重点实验室, 上海 201109)

摘要: 针对“嫦娥五号”轨道器多载荷数据调度、存储复接、存储管理和高可靠性高要求, 设计了基于NAND Flash芯片的数据复接及存储模块管理方案。采用具有抗辐射特性的Axcelerator系列反熔丝现场可编程门阵列(Field-Programmable Gate Array, FPGA)芯片作为主控制芯片, 用于对多路载荷数据的调度管理和存储控制, 采用大容量NAND Flash芯片用于对多种载荷和星上遥测的存储管理, 实现了对多个数据源按要求进行数据记录、数据管理以及数据回放等功能。“嫦娥五号”任务在轨结果表明, 经过数传帧编码以及可变速率回放, 实现了深空探测对数据复接存储回放的要求, 为后续深空探测数据的高速下传提供参考。

关键词: 嫦娥五号; 数据调度; 存储; 编码

中图分类号: V41

文献标识码: A

文章编号: 2096-9287(2021)03-0299-07

DOI:10.15982/j.issn.2096-9287.2021.20210001

引用格式: 刘攀, 王志国, 刘建峰, 等. “嫦娥五号”载荷数据存储管理方法[J]. 深空探测学报(中英文), 2021, 8(3): 299-305.

Reference format: LIU P, WANG Z G, LIU J F, et al. A payload data storage and management method for Chang'E-5[J]. Journal of Deep Space Exploration, 2021, 8(3): 299-305.

引言

我国已成功采用“长征五号”(CZ-5)运载火箭将“嫦娥五号”(Chang'E-5, CE-5)送到预定轨道, 并实现了探月工程“绕”“落”“回”里程碑式壮举。“嫦娥五号”轨道器各种试验、遥测数据因地球未能覆盖全部的可见弧段导致无法及时传送到地面站, 需要星载固存进行接收、管理和存储再转发。

当前, 国内外学者对基于NAND Flash大容量存储芯片的应用做了大量的研究和探索, 有很多值得推广的成果。Luo等探究了NAND Flash多页并行读取的方法^[3]。基于深空探测的存储方案设计上, 相关学者也做了大量的应用探索工作, 宋琪等提出了卫星固态存储器数据容错设计方案, 实现了在数据存储过程对错误进行检测并实时反馈的功能^[4-5]。张卫东等对星载固态存储控制器的标准化和可配置方面进行了研究, 并设计了通用存储控制器, 可适配多个厂家的NAND Flash芯片^[6]。上述探究主要为芯片设计方面的应用、算法优化方面的探索和通用化设计^[6-10], 对于数据接口的扩展则没有涉及, 本文使用大规模反熔丝工艺FPGA上实现灵活的数据存储管理以及纠错编码来满足

“嫦娥五号”高可靠、集成度高以及小型化应用需求。

本文针对“嫦娥五号”轨道器任务设计了一种基于高级在轨系统(Advanced Orbiting Systems, AOS)虚拟信道存储管理方法, 该器有效载荷数据包括常规遥测、内存下传遥测、返回器遥测、图像遥测以及延时回放遥测等数据, 这些数据传输时效性要求高。在此情况下, 进行合理的数据处理、优先级编排和高效的存储分区, 能够迅速提取高优先级载荷数据, 在突发情况下优先下传重要数据, 并且可以最大化地使用缓存资源^[11-13], 基于AOS信息流的不同数据业务的传输需求, 设计了一种合理的队列管理与存储管理方法。

1 数据存储管理及编码

1.1 数据流处理方法

数据复接及存储模块主要功能为接收异步串行遥测指令并解析; 根据指令完成4路相机数据、常规遥测、内存下传遥测、返回器遥测以及1路器内环境监测探头数据的屏蔽接收处理, 包括数据复接、信息格式化缓存等, 在遥测数据接收完成后, 可以选择存入固存或者进行实时下行传输。同时提供基于虚拟信道标志(Virtual Channel Identifier, VCID)的分区存储管

理、延时数据和实时数据可根据指令进行组合编排,组合数据编排后经过里德索洛蒙(Reed Solomon, RS)编码、伪随机化、卷积编码、最终送测控数传应答机。与“嫦娥三号”(Chang'E-3, CE-3)相比,“嫦娥五号”的数据复接及存储模块载荷数据输入速率和数据输出速率均有较大提升,满足整器遥测数据和业务数据处理需求。

数据复接及存储模块接收各路图像监视数据和整器遥测数据,进行复接处理,并标记对应的虚拟信道标识。在实时传输模式下,由调度单元裁决下发数据类型(载荷数据或者填充帧),保证下传数据的连续性。然后按国际空间数据系统咨询委员会(Consultative Committee for Space Data Systems, CCSDS)格式化数据要求,并存储到Flash中。数据存储完成之后,由路由单元根据当前指令裁决数据下传模式:回放模式(从存储器读取数据)、实时模式(直接调用缓存中的数据)。在数据下传之前需对数据进行RS编码、伪随机化、卷积编码等处理,然后下传给测控数传分系统,系统信息框图如图1所示。

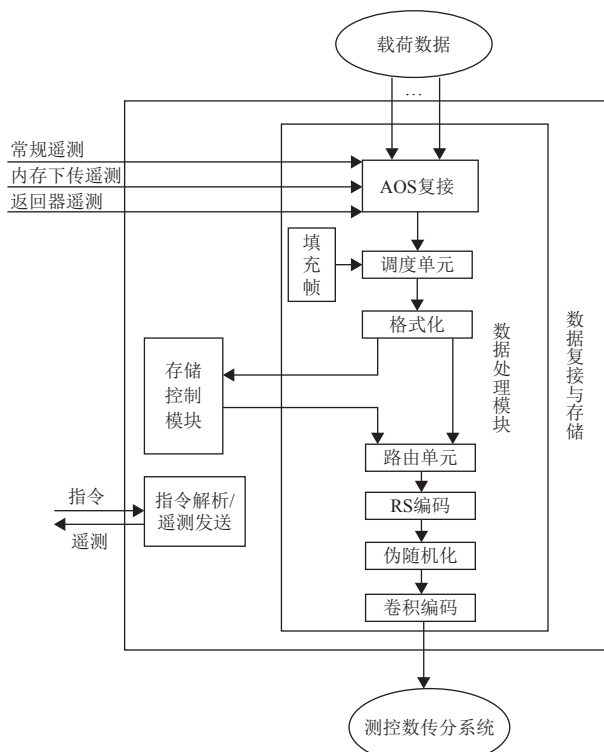


图1 大容量固定及复接模块的信息流程框图

Fig. 1 Diagram of information flow for large-capacity fixed storage and multiplexing module

本系统采用具有抗辐射能力的现场可编程门阵列(Field-Programmable Gate Array, FPGA)作为控制器,设计NAND Flash控制器,用于提高存储控制系统

的存取速率。本文共采用两片NAND Flash芯片,并考虑了NAND Flash芯片可能出现的故障预案,将两片Flash芯片设计为独立控制模式,既可以两片同时用于存储有效数据+校验数据,又可以相互独立用于存储有效数据,默认情况下使用第一种存储模式。

1) 有效数据+校验数据控制模式:该模式下两片FLASH芯片用于将有效数据和校验数据分开存储,第一片存储有效数据,第二片存储错误检测与纠正(Error Detection And Correction, EDAC)校验码(汉明校验),每存储8 bit数据计算一次EDAC存入Flash,每次读出数据时进行一次校验和纠错(每8 bit数据中可以纠正1 bit错误),满足NAND Flash自身对纠错的要求,同时有效降低空间单粒子影响导致数据异常产生的错误。

2) 独立存储有效数据控制模式:该模式下若两片Flash相互独立,则关闭EDAC校验,在两片芯片都正常情况下,可以互为备份进行存储,但是若数据异常则无法判断和纠错。在其中一片芯片异常的情况下,可用另一片芯片存储有效数据,保证大容量存储载荷数据能够正常传输到地面。

1.2 坏块表管理

控制Nand Flash的核心是如何建立和维护坏块表,若坏块表失效,存储系统中的数据将出现乱码,无法实现正常的数据存储和回放。实践证明,非易失性磁性随机存储器(Magnetic Random Access Memory, MRAM)具有较为优秀的抗辐射能力,其内部存储区域的磁特性使存储部分对单粒子翻转(Single Event Upset, SEU)免疫,将坏块表存放在MRAM中,可以有效降低空间辐射环境对系统的影响。因此,本文的FLASH坏块表管理采用MRAM存储,方案框图如图2所示。

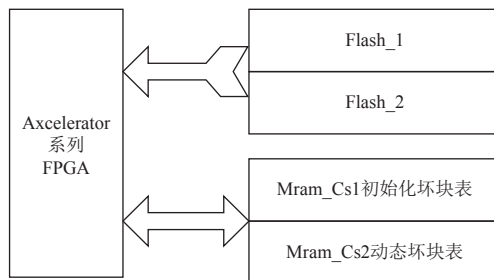


图2 坏块表存储示意图

Fig. 2 Schematic diagram of bad block table storage

本方案的坏块表管理采用MRAM存储2份坏块信息,分别为初始坏块信息(静态、可更新),以及动态坏块信息。2份坏块信息表均可更新,初始坏块信息

在产品出厂时根据Flash芯片实际状态进行设置；动态坏块信息则由初始坏块信息导入或者由地面遥控注入Flash坏块信息，且动态坏块信息在应用的过程中可自动根据FLASH擦除和编程反馈状态进行更新，确保及时存储Flash坏块信息，防止将有效数据写入Flash坏块区。

在初始化坏块表的过程中，由FPGA遍历Flash芯片的坏块标志，并将Flash的每个block（块）在对应的MRAM地址中标记为0x00（表示好块），或者标记为0xff（表示坏块）。该动作仅操作一次，通过加锁指令防止误操作，后续通过注数指令可对初始坏块信息进行局部修改。通过指令FPGA将静态坏块表更新至动态坏块表，Flash数据读、写以及擦除过程中，FPGA将检索动态坏块表的好坏快标志，并更新。

1.3 数据编码应用

系统应用于探月工程中，月地距离大约38万km，深空通信的信道编码技术中，最为经典的方法是内码采用卷积码，外码采用里德-所罗门（RS）码的级联码，相较于低轨道（Low Earth Orbit, LEO）卫星常采用RS编码或者低密度奇偶校验码（Low Density Parity—check Code, LDPC）编码再结合序列伪随机化，虽降低数据传输效率却有效提高数据传输可靠度。

本系统的存储及复接模块下行数传信道编码采用国际空间数据系统咨询委员会（The Consultative Committee for Space Data Systems, CCSDS）标准建议的高级在轨系统对数据进行处理，其编码顺序：①RS编码；②伪随机码；③卷积编码”方式，并采用序列伪随机化。

1) RS编码器

RS编码特性如下：

- (1) RS (255, 223) 纠错编码，交织深度为4；
- (2) 每个RS码元的位数：8；
- (3) 一个RS码字内RS码元的纠错能力：16；
- (4) 每个编码字包含255个码元；
- (5) 域生成多项式： $F(x) = x^8 + x^7 + x^2 + x + 1$ （在GF(2)上定义）；

- (6) 码生成多项式： $g(x) = \prod_{j=112}^{143} (x - \alpha^{11j}) = \sum_{i=0}^{32} G_i X^i$ （在GF(28)上定义， $F(\alpha) = 0$ ）。

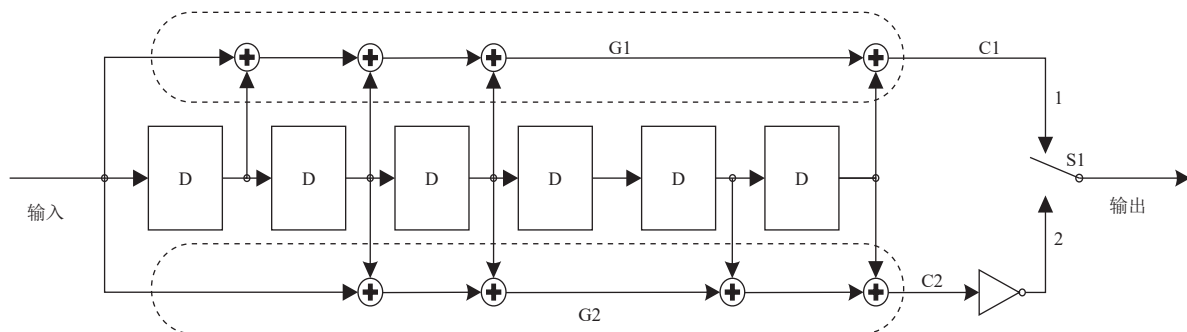
2) 伪随机码发生器

在数据传输的过程中，为确保数据的转变密度合理可靠，防止同时出现全“0”或者全“1”的情况，需要采用伪随机码发生器对数据进行加扰处理的方法，该方法主要是将伪随机序列与数据每位异或起来（不包含帧同步字）。根据CCSDS的处理方法，伪随机序列生成多项式为： $h(x) = x^8 + x^7 + x^5 + x^3 + 1$ 。该序列在处理完成255 bit之后重复，同时在处理每个同步标志期间序列发生器重复初始化为全1状态。该序列的初始40个bit为

1111 1111 0100 1000 0000 1110 1100
0000 1001 1010

3) 卷积编码器

卷积编码器的输入为脉冲编码调制（Pulse Code Modulation, PCM）信号—非归零电平码NRZ-L。卷积编码器原理如图3所示。



注 1: $\rightarrow \boxed{D} \rightarrow$ 1比特延迟；

注 2: 每输入1比特信息，采样开关S1就要完成从位置1到位置2的一个切换循环，同时产生2个输出符号；

注 3: 当采样开关S1位于位置1是，产生于输入信息比特相关的第一个输出符号；

注 4: $\oplus$ = 模二加法器；

注 5: $\rightarrow \triangle \rightarrow$ = 反相器。

图3 卷积编码器

Fig. 3 Convolution encoder

卷积编码器的输出为二元电平符号序列(即比特序列)。编码器每输入一个信息比特,产生二个输出符号,即一个符号对。符号输出顺序如下:

第一输出符号: C1的输出;

第二输出符号: C2的反相输出。

地面应用系统则先进行Viterbi译码,纠正信道误差,然后按照比特移位找到同步帧头,进行逆向操作,从而恢复原始数据。

2 存储调度实现

2.1 载荷数据调度管理

载荷数据调度管理的功能为数据的接收和复接,数据复接模块使用三线制低压差分信号(Low Voltage Differential Signaling, LVDS)和异步串口接收、先位宽转换至8 bit,然后跨时钟域转换,使用本地时钟将数据存入先入先出队列(First Input First Output, FIFO),防止FIFO受载荷数据时钟信号异常干扰,当FIFO数据达到一帧数据量时,FIFO向数据拼接模块帧传输请求,数据拼接模块将从目标FIFO中读取一帧数据,并添加虚拟信道号和帧计数,然后将这帧数据传送到SRAM控制器。数据拼接模块轮询各路FIFO帧传输请求的速率是40 MHz,其数据处理能力可以满足各路数据同时输入的最高速率。

载荷调度管理分为实时模式和延时模式两种。

实时模式为了保证信道的连续性则会在无有效数据时组织填充帧下传,通过指令设置外部通道与实时缓存的映射关系,共设置5个缓存,其中3个固定为常规遥测、内存下传数据遥测、返回器大容量存储数据遥测,另外2路则可设置为器内环境监测探头数据、图像数据以及延时回放数据3路数据中的2路映射到2个FIFO缓存实现各种模式下的数据下传工作,通过该方法使得FPGA内部32 k字节RAM即满足内部多路载荷高速数据缓存需求。

延时模式为了有效利用存储容量,则只会记录有效载荷数据帧。在混合传输模块下则会在实时帧中穿插NAND Flash存储器中的载荷数据帧,有效利用信道。延时模式下,载荷数据在接收、组帧完成后,由存储控制模块根据VCID分区存入Nand Flash芯片指定区域。在数据回读的过程中,则根据遥控指令,按照VCID检索Flash中的数据,将检索出的数据通过实时模式指定的通道下传到地面控制中心。

鉴于本系统中外部输入的数据路数较多,可能出现输入突发速率较大的模式,故采用一片SRAM作为存储和回放的缓冲器,实时延时数据流图如图4所示。

由于载荷数据速率较快、载荷类型较多、而FPGA内部BLOCK RAM资源有限、设计选用外置SRAM作为载荷数据二级缓存,当载荷数据一级缓存达到一定数据量时由数据处理模块进行调度缓存到二级缓存SRAM中。SRAM空间划分为8个8 K字节空间,按照载荷优先级调度,三级缓存为一个8 K字节缓存当SRAM缓存半满即4 K字节准备好后,随即调度到三级缓存中,然后存入Flash芯片中。

2.2 载荷管理的存储控制实现

使用Nand Flash 需要进行擦除、写以及读三种典型操作,为了满足应用需求,使用Nand Flash包含擦除、读、写三个子状态,并设置读写分时复用、分区擦除和全部擦除功能满足数据存储管理需求,通过分区管理可以实现任意载荷数据点播操作。而根据任务需求选择合适的外置缓存以及核算内部缓存是否能够满足应用需求则是任务的关键。

本文采用的Nand Flash存储阵列拓扑结构内部分为8个叠层,各自有独立的片选、读写使能等信号。根据Flash叠层拓扑结构,实现流水擦除和编程操作,Flash操作速率为20*8 Mbps的突发写入速率,流水操作后,当前层编程或者擦除期间可对下一层进行数据写入或擦除,有效提高操作效率。

载荷缓存FIFO的半满信号产生写SRAM请求,SRAM读写调度控制模块根据半满请求,产生写SRAM时序,将载荷数据写入SRAM;SRAM与Flash之间的FIFO的半空信号产生SRAM的读请求,将SRAM中的数据读出到SRAM与Flash之间的FIFO,写优先级高于读优先级,SRAM的读写速率为20*16 Mbps,位宽为16,读写周期为50 ns,速率远高于Flash读写速率和载荷输入速率,实现SRAM数据缓存分时复用的目的。

3 数据存储测试与验证

测试过程中,模拟载荷1~4的LVDS数据源,每路数据时钟频率提高到10 Mbps后,载荷数据也可正常存储,为了快速验证数据存储系统正确性,提高测试效率,设计配置数据输出模式具备2.5、5、10 Mbps三档速率可切换。通过输入输出速率的提升,可快速验证该存储系统设计正确性,系统测试时间减半,大大提高测试效率。

针对模块方案设计了通用地面测试设备,分为数据模拟源板卡、数据接收解码板卡以及可配置的地面数据分析比对软件。通过对模块的全面测试验证,保证设计可靠性和测试覆盖性。图5为“嫦娥五号”工程应用时下传的双谱段相机实时数据。

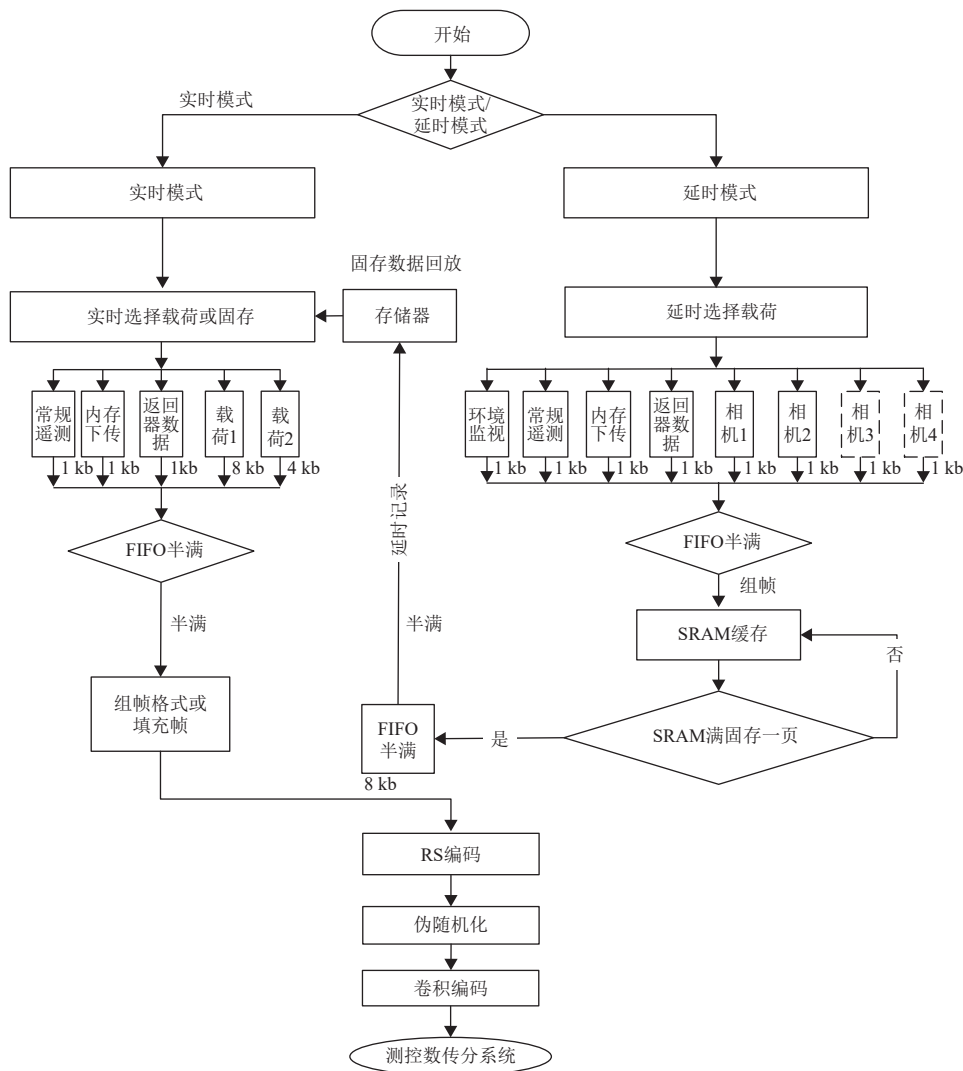


图 4 数据存储信息流程图

Fig. 4 Flow chart of data storage information

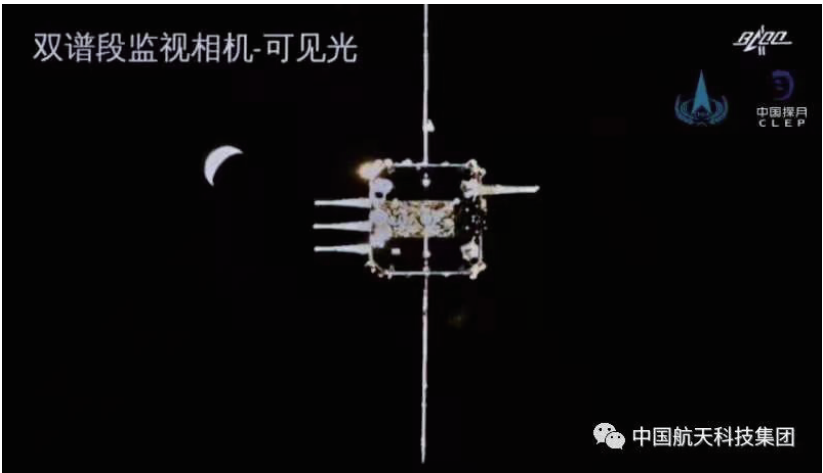


图 5 “嫦娥五号”双谱段相机实时数据

Fig. 5 Real-time data of Chang'e-5 bi-spectrum camera

4 结 论

本文针对探月工程三期“嫦娥五号”任务需求,设计并实现了以反熔丝FPGA为主控制芯片的载荷调度存储管理方法,设置多个存储、回读模式,将数据写入NAND Flash,同采用MRAM存储NAND Flash块地址信息,实现对数据的文件管理,完成载荷数据的调度和存储任务。通过测试表明,该设计切实可行,满足航天大容量、高可靠存储需求。

本文设计的多通道高速数据调度、存储、回读的载荷及遥测处理方式,为适应多通道、多种速率的载荷调度、组帧、分发等工况,在产品设计的过程中综合考虑各种应用需求,将对外输入输出的LVDS接收通道等设计为速率可配置模式,通道数量设计为可扩展模式,目前已在“嫦娥五号”上成功应用,形成的通用IP将推广应用到高分、微纳等系列卫星平台,会产生良好的经济效益。

参 考 文 献

- [1] YANG T, XIA Z, SHI D, et al. Analysis and optimization of threshold voltage variability by polysilicon grain size simulation in 3D NAND Flash memory[J]. IEEE Journal of the Electron Devices Society, 2020(8): 140-144.
- [2] OUYANG Y, XIA Z, YANG T, et al. Optimization of performance and reliability in 3D NAND flash memory[J]. IEEE Electron Device Letters, 2020, 41(6): 840-843.
- [3] LUO T, PELEATO B. Multi-page read for NAND Flash[J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2017, 64(1): 76-80.
- [4] 李娜, 宋琪, 朱岩, 等. 星载大容量固态存储器快速可靠启动算法设计[J]. 哈尔滨工业大学学报, 2015, 47(10): 100-105.
LI S, SONG Q, ZHU Y, et al. Design of quick initialization algorithm for space-borne solid state recorder[J]. Journal of Harbin Institute of Technology, 2015, 47(10): 100-105.
- [5] 宋琪, 邹业楠, 李珊, 等. 卫星固态存储器数据容错设计与机制[J]. 国防科技大学学报, 2016, 38(1): 101-106.
SONG Q, ZOU Y N, LI S, et al. Design and mechanism of the fault tolerance of solid state recorder[J]. Journal of National University of Defense Technology, 2016, 38(1): 101-106.
- [6] 张伟东, 董振兴, 朱岩, 等. 星载固态存储控制器标准化, 可配置通用

设计[J]. 微电子学与计算机, 2019, 36(9): 61-66.

ZHANG W D, DONG Z X, ZHU Y, et al. Standardized, configurable universal design for onboard solid state memory controllers[J]. Microelectronics & Computer, 2019, 36(9): 61-66.

- [7] 刘哲, 邓畅霖, 郭薇, 等. 星座通信网建设过程中的端到端延时及存储需求分析[J]. 宇航学报, 2016, 37(7): 854-861.
LIU Z, DENG C L, GUO W, et al. Analysis on end-to-end delay and storage requirements during progressive construction of satellite constellation communications network[J]. Journal of Astronautics, 2016, 37(7): 854-861.
- [8] 刘光辉, 周军, 孙菲. 皮纳卫星数据存储系统设计与在轨验证[J]. 哈尔滨工业大学学报, 2018, 50(2): 178-183.
LIU G H, ZHOU J, SUN F. Design and in-orbit verification of data storage system for nano-pico satellites[J]. Journal of Harbin Institute of Technology, 2018, 50(2): 178-183.
- [9] 司锋, 张强. 基于数据路由的星载高速数据处理系统研究[J]. 测绘通报, 2014(S1): 27-31.
SI F, ZHANG Q. Research on space-borne high-speed data processing system based on data routing[J]. Bulletin of Surveying and Mapping, 2014(S1): 27-31.
- [10] 黄炜. 卫星数据的分布式存储系统设计与验证[J]. 遥感信息, 2020, 35(5): 66-71.
HUANG W. Design and verification of distributed file system based on satellite data[J]. Remote Sensing Information, 2020, 35(5): 66-71.
- [11] 陈斌, 沈卫华, 朱岩, 等. 嫦娥二号卫星大容量存储器设计[J]. 航天器工程, 2011, 20(5): 99-104.
CHEN B, SHEN W H, ZHU Y, et al. Design of solid state recorder for Chang'e-2 orbiter[J]. Spacecraft Engineering, 2011, 20(5): 99-104.
- [12] 刘红伟. 基于CPCI架构卫星信号数据存储平台的硬件设计[J]. 电子技术与软件工程, 2019(13): 178.
- [13] 许志宏, 董振兴, 安军社. 基于FPGA的星载一体化高速数据复接器设计[J]. 电子设计工程, 2018, 26(5): 184-188, 193.
XU Z H, DONG Z X, AN J S. Design of spaceborne integrated high-speed data multiplexer based on FPGA[J]. Electronic Design Engineering, 2018, 26(5): 184-188, 193.

作者简介:

刘攀(1984—), 男, 高级工程师, 主要研究方向: 星载计算机设计、大容量存储设计。

通讯地址: 上海市闵行区中春路1777号(201109)

E-mail: lp316@163.com

游红俊(1973—), 男, 研究员, 主要研究方向: 星载综合电子。**本文通讯作者。**

通讯地址: 上海市闵行区中春路1777号(201109)

E-mail: snowolf8271@sina.com

A Payload Data Storage and Management Method for Chang'E-5

LIU Pan^{1,2}, WANG Zhiguo^{1,2}, LIU Jianfeng^{1,2}, MIN Kanglei^{1,2}, YOU Hongjun^{1,2}

(1. Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China;

2. Key Laboratory of Intelligent Computing Technology (SAST), Shanghai 201109, China)

Abstract: According to the requirements of multi-load data scheduling, storage multiplexing, storage management and high reliability of lunar exploration project, a management scheme of large-capacity data multiplexing storage module based on NAND Flash chip is designed. Axcelerator series anti-fuse FPGA chip with anti-radiation characteristics is used as the main control chip for scheduling management and storage control of multi-channel load data, and large-capacity NAND Flash chip is used for storage management of multiple loads and on-board telemetry, which realizes the functions of data recording, data management and data playback for multiple data sources as required. The practice results show that the requirements of data multiplexing, storage and playback for deep space exploration can be realized through data transmission frame coding and variable rate playback, which provides reference for the high-speed data downloading for subsequent deep space exploration.

Keywords: Chang'E-5; data scheduling; storage; code

Highlights:

- Design of synchronous storage of multiple load data of Chang'E-5 detector.
- Design of multiple download modes for load data of Chang'E-5 detector.
- Extended design of load data storage channel of Chang'E-5 detector.
- Multi-rate adaptive design of storage control module.
- Flash partition control and high-speed storage design.

[责任编辑: 杨晓燕, 英文审校: 宋利辉]