

DOI:10.11918/j.issn.0367-6234.201611144

星载存储器吞吐率瓶颈与高速并行缓存机制

董振兴^{1,2}, 朱 岩^{1,2}, 许志宏^{1,2}, 宋 琪^{1,2}

(1.中国科学院国家空间科学中心,北京 100190; 2.中国科学院大学,北京 100190)

摘 要:为解决目前星载存储器无法有效支持多路高速数据并行存储的问题,针对载荷数据高速输入需求,对基于 NAND Flash 的固态存储器的吞吐率瓶颈进行分析,根据固态存储器的固有写操作特性对有效吞吐率的影响,提出了四级流水线操作和总线并行扩展方案;针对多通道数据并行存储、流水线加载连续性等需求,对使用现场可编程门阵列 FPGA(Field-Programmable Gate Array)内部双端口随机存取存储器 RAM(Random access memory)、外置静态随机存取存储器 SRAM(Static Random Access Memory)等已有缓存方案的不足进行分析,完成了基于同步动态随机存储器 SDRAM(Synchronous Dynamic Random Access Memory)的方案可行性分析与新型存储单元架构设计,最终提出了基于 SDRAM 的高速多通道缓存与存储协同调度方案.模型仿真与原型功能验证结果表明,方案在极限工况下可将 4 路高速文件数据连续并行接收缓存至 SDRAM 中,并可根据各分区缓存状态将文件数据按优先级自主动态写入 Flash 中,期间缓存无溢出,并最终进入常规动态平衡调度状态,实现了对多路高速载荷数据的并行接收缓存和自主调度存储,且存储器的数据吞吐率可达 1.2Gbps,能够满足未来星载存储器对多路高速载荷数据存储的需求.

关键词:星载存储器;存储吞吐率;流水线操作;总线并行扩展;高速并行缓存

中图分类号:TN919.5 文献标志码:A 文章编号:0367-6234(2017)11-0052-08

Bottleneck analysis of spaceborne memory throughput and high-speed parallel caching mechanism design

DONG Zhenxing^{1,2}, ZHU Yan^{1,2}, XU Zhihong^{1,2}, SONG Qi^{1,2}

(1. National Space Sciences Center, Chinese Academy of Sciences, Beijing 100190, China;

2. University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: To solve the problem that the current spaceborne memory cannot support parallel storage of multi-channel high-speed data effectively, this paper proposes two methods. To satisfy the requirement of receiving high-speed payload data, this paper analyzes the throughput bottleneck of NAND Flash-based solid-state memory, and proposes a four-stage pipeline operation and bus parallel expansion scheme according to the inherent write operation characteristics. To meet the needs of parallel storing multi-channel data and the continuity of pipeline operation, this paper analyzes the deficiencies of the data cache methods which use RAM (Random access memory) and SRAM (Static Random Access Memory), and proposes a scheduling scheme using high-speed caching and cooperative storing, which is based on SDRAM (Synchronous Dynamic Random Access Memory). Through the model simulation and the prototype function verification, we verify that the proposed scheme can effectively support parallel reception of the 4-way high-speed payload data and the autonomous scheduling storage of solid state memory, and the throughput rate of memory can reach 1.2 Gbps, which satisfies the demands for storing the multi-channel high-speed payload data on spaceborne memory in the future.

Keywords: spaceborne memory; storage throughput; pipeline operation; bus parallel expansion; high speed parallel cache

随着航空航天事业的快速发展,空间探测器的有效载荷种类及任务形式多样化,对星上数据管理系统高速并行存储能力提出更高要求.星载存储器是数管系统的核心设备之一^[1-2].基于 NAND Flash

的固态存储器具有存储密度高、非易失等特点,是星上数据存储的主流方式^[3-5].

目前星载存储器方案主要面向低速有效载荷设计,支持信源量小、固定分区存储灵活性差^[6-7],如暗物质粒子探测卫星、实践十号科学实验卫星存储器有效吞吐率低于 350Mbps,仅支持两路数据分区存储,无法适用于多路载荷数据高速输入、同一时刻多任务等工作模式.徐永刚等^[8]基于 Flash 设计的图像记录系统使用 CPU 内存作为两路外部图像数

收稿日期:2016-11-30

基金项目:中国科学院空间科学先导卫星计划(XDA0402020201)

作者简介:董振兴(1991—),男,博士研究生

朱 岩(1973—),男,博士生导师

通信作者:董振兴,E-mail: dongzhenxing@nssc.ac.cn

据缓存,每路各缓存一幅图像后由 CPU 将数据按顺序逐一写入 Flash 阵列中,其缓存调度方式制约了载荷数据的输入速率。李晴等^[9-10]针对 Flash 存储技术和高速互连技术进行了研究,系统通过多个存储体同时工作实现对多种类型数据记录,虽然提高了系统存储吞吐率,但不支持单个存储体对多路载荷数据的有效存储,无法适用于对功耗、重量等要求严格的航天领域。为此,针对星载存储器实际应用,对固态存储器吞吐率瓶颈进行分析,探讨提高吞吐率关键技术以支持高速载荷数据输入,并对高速并行缓存与任务调度机制进行设计验证,以期解决单板存储器对多路高速数据同步缓存与自主存储方案的需求问题。

1 星载存储器吞吐率分析

1.1 基于 Flash 存储器吞吐率瓶颈

目前星载存储器主要使用经抗辐照加固处理后的宇航级 3DFN64G08VS NAND Flash 叠装芯片,每个叠装模块由 8 片 K9F8G08U0M 基片组成^[11]。设

叠装模块 Flash 数据总线位宽为 $W_{\text{one_die}}$,最大工作频率为 $f_{\text{flash_work_max}}$,则单个模块理论最高吞吐率为

$$H_{\text{one_die_theory_max}} = f_{\text{flash_work_max}} * W_{\text{one_die}} \quad (1)$$

宇航级 NAND Flash 基片最高工作频率为 50 MHz,抗辐照加固后叠装模块最高工作频率降为 40 MHz。根据军用元器件 80%降额使用准则,NAND Flash 实际最高工作频率为 32 MHz,又 $W_{\text{one_die}} = 8\text{bit}$,由式 (1)可得 Flash 叠装模块理论最高吞吐率为 256 Mbps。

固态存储器的固有写操作特性降低了有效吞吐率。K9F8G08U0M 基片以页为单位进行读写操作,一页数据区大小为 4 096 Bytes,数据写入时,先将数据加载到芯片内部缓存区,再经过编程完成数据写入,写操作流程见图 1。

根据写操作时序要求,完成一页数据写操作需要的时间为:

$$t_{\text{one_page_write}} = t_{\text{LOAD}} + t_{\text{PROG}} + t_{\text{CHECK}}, \quad (2)$$

$$t_{\text{LOAD}} = t_{\text{command}} + t_{\text{address}} + t_{\text{ADL}} + t_{\text{data}} \quad (3)$$

时序参数见表 1,其中: T 为时钟周期, $N_{\text{one_page}}$ 为页容量, t_{WHR} 为读取编程结果等待时间。

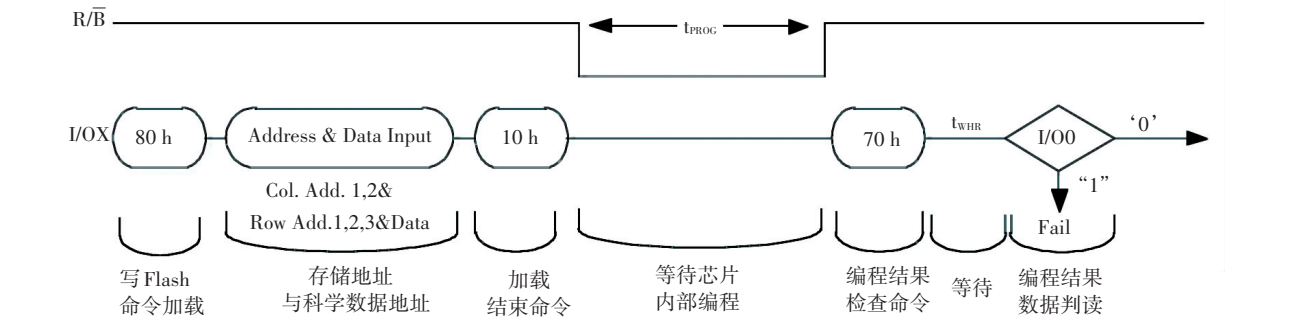


图 1 NAND Flash 写操作流程

Fig.1 Procedure of NAND Flash writing operation

表 1 Flash 写操作时序参数表

参数	说明	时间
t_{command}	命令加载时间	$3 * T$
t_{address}	地址加载时间	$5 * T$
t_{ADL}	地址到数据加载间隔	$\geq 100 \text{ ns}$
t_{data}	数据加载时间	$N_{\text{one_page}} * T$
t_{PROG}	编程等待时间	$200 \text{ us} \sim 700 \text{ us}$
t_{CHECK}	编程结果检查时间	$\geq t_{\text{WHR}} + T$
t_{WHR}	读编程结果等待时间	$\geq 60 \text{ ns}$

设 Flash 有效写操作效率为 $\eta_{\text{one_die_write}}$,实际工作时最大支持写速率为 $H_{\text{one_die_theory_max}}$,则

$$\eta_{\text{one_die_write}} = \frac{T_{\text{WC}} * N_{\text{one_page}}}{t_{\text{one_page_write}}}, \quad (4)$$

$$H_{\text{one_die_work_max}} = H_{\text{one_die_thy_max}} * \eta_{\text{one_die_write}} \quad (5)$$

由式 (2) ~ (5) 可得 Flash 在实际工作中的最高

有效写操作效率为 38.97%,最高写速率为 99.76 Mbps,无法满足多路高速载荷数据的存储需求。

1.2 提高吞吐率关键技术

1.2.1 流水线操作

在 Flash 写操作流程中,芯片编程占用大量时间,大大降低了 Flash 的写入效率,采用流水线操作方式可解决芯片内部编程过程中的长时间等待问题。流水线写 Flash 操作原理见图 2。

每次加载数据到一组 Flash 芯片后,该组芯片进入内部编程状态,在此期间尽管不能对该组芯片进行其它操作,但可立即进行另一组芯片加载。由芯片加载时间与编程时间数值关系知,当第 4 组芯片加载完毕后,第 1 组芯片的编程过程通常已经结束。使用四级流水线操作技术,可以基本保证宏观上实现对存储区的全速率无间断数据写入,Flash 有效写操作速率接近理论值 256 Mbps,解决了存储介质固有写操作特性对存储器有效吞吐率的限制。

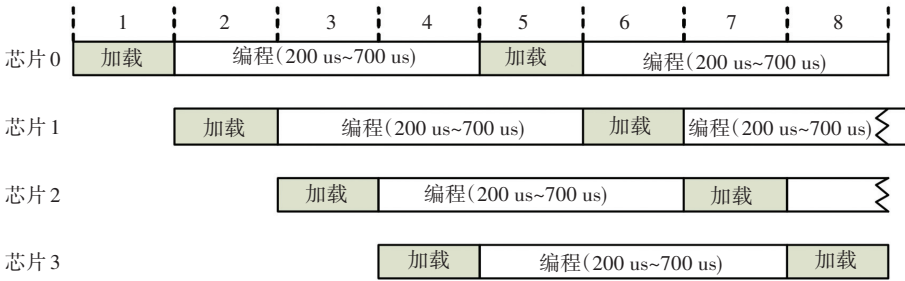


图 2 四级流水线加载 Flash 原理

Fig.2 Four-level pipeline load operation of NAND Flash

1.2.2 总线并行扩展

为进一步提高存储吞吐率,扩展存储容量,在空间横向轴上采用 I/O 总线并行扩展技术. 综合考虑 FPGA 管脚资源和存储器性能需求,设计 8 倍 I/O 总线并行扩展方案,并行扩展结构见图 3.

将 8 片 Flash 叠装模块控制总线、状态总线互连,而 I/O 总线扩展为 64 位,64 位总线不同数

据位段分别对应 8 片并行扩展的 Flash 的 8 位总线,将 Flash 中 8 片并行的页扩展成 1 簇进行读写操作. 由于物理空间扩展与芯片操作时序无关,因此 8 倍总线扩展后系统速率将提高为原来的 8 倍,此时存储器单板理论可支持最高 2Gbps 数据输入,提高了存储系统对高速载荷数据的吞吐能力.

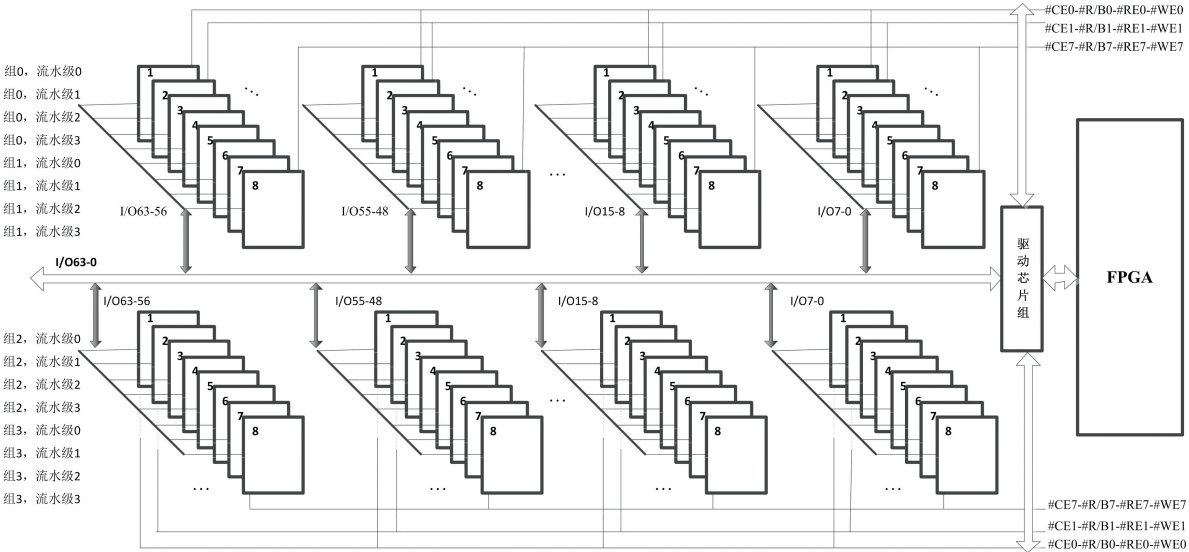


图 3 8 倍 FLASH I/O 并行扩展结构图

Fig.3 8 times parallel I/O bus expansion of NAND Flash

2 高速缓存与存储任务调度

四级流水线的操作特点要求 Flash 在启动写操作时,将加载所需的四簇数据准备完成,以确保四级流水加载连续性,同时为保证每次流水操作均是针对同一载荷,需设计并行缓存机制解决多路载荷数据各自分区缓存问题.

传统方案使用 FPGA 内部双端口 RAM 作为缓存以降低控制复杂度,但对于多路载荷高速并行输入,有限的 RAM 资源难以满足缓存容量需求. 外置缓存芯片 SRAM 虽然有了一定容量提升,但仍然有限,并且在 64 位数据总线下 SRAM

接口速率最高为 2 Gbps,数据吞吐复用 I/O 端口时平均读写速率只有 1 Gbps,限制了 Flash 最高写速率使用. 宇航级 SDRAM (Synchronous Dynamic Random Access Memory,同步动态随机存储器) 芯片最高工作频率达 133 MHz,当数据总线为 64 bits 时,理论接口总吞吐率可高达 8 Gbps,且芯片容量 3 Gbits,能够满足系统高吞吐率需求.

2.1 SDRAM 缓存可行性

SDRAM 芯片一次完整的读/写数据操作包括自刷新、行选通、数据突发读写、预充电及自动刷新等部分^[12],操作流程见图 4.

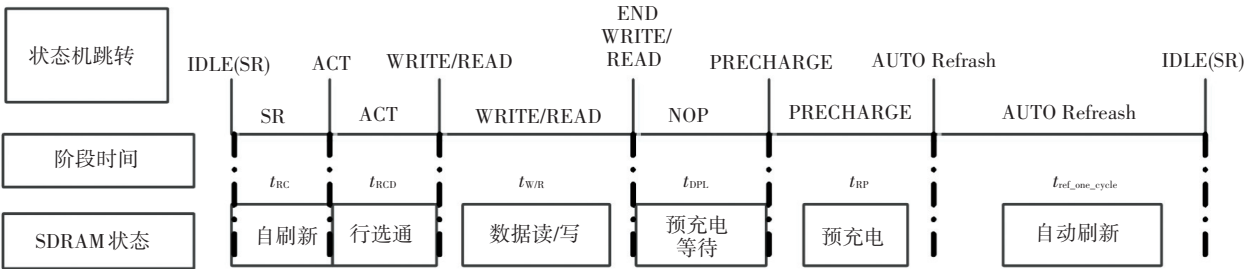


图 4 SDRAM 读写操作流程图

Fig.4 Procedure of SDRAM reading/writing operation

SDRAM 换行操作需再次行选通,为减小时间开销,设计一次读、写所涉及地址均在同一行内,数据吞吐采用 8 字突发传输方式. 由于 SDRAM 芯片每 64 ms 内必须完成 8 192 次自动刷新操作以保证数据有效,为保证连续读/写作过程满足刷新要求,将自动刷新机制分配到每个读/写周期操作后,此时

$$H_{W/R_{operate}} = \frac{N_{W/R} * T * W_{data_bus} * (1/T)}{t_{RC} + t_{RCD} + N_{W/R} * T + t_{DPL} + t_{RP} + N_{ref} * t_{RC}}$$
$$t_{RC} + t_{RCD} + N_{W/R} * T + t_{DPL} + t_{RP} + N_{ref} * t_{RC} \leq \frac{t_{REF(max)}}{N_{cycles}}.$$

(6)

其中: $N_{W/R}$ 为一个读/写周期的数据大小, T 为 SDRAM 工作时钟周期, N_{ref} 为每次读/写操作后自动刷新次数,操作时序参数见表 2.

当 SDRAM 工作时钟为 64 Mhz, N_{ref} 取 2 时,由式 (6) 可得吞吐率 $H_{W/R_{operate}} = 3.848$ Gbps,平均读写

速率 $H_{W/R_{operate_average}} = 1.924$ Gbps,与 NAND Flash 最大支持写速率相匹配,能够满足多通道分区输入速率、空间需求,缓存方案可行.

表 2 SDRAM 时序参数

Tab.2 Timing parameter of SDRAM operation

参数	说明	时间
t_{RC}	自刷新时间	≥ 67.5 ns
t_{RCD}	行选通周期	≥ 20 ns
$t_{W/R}$	数据读写时间	$N_{W/R} * T$
t_{DPL}	预充电等待	≥ 15 ns
t_{RP}	预充电时间	≥ 20 ns
t_{ref}	自动刷新时间	$N_{ref} * t_{RC}$

2.2 星载存储器方案设计

采用四级流水线操作与总线并行扩展技术提高单板存储吞吐率与存储容量,采用 SDRAM 芯片作为缓存支持高速并行数据输入,星载存储器总体设计方案见图 5.

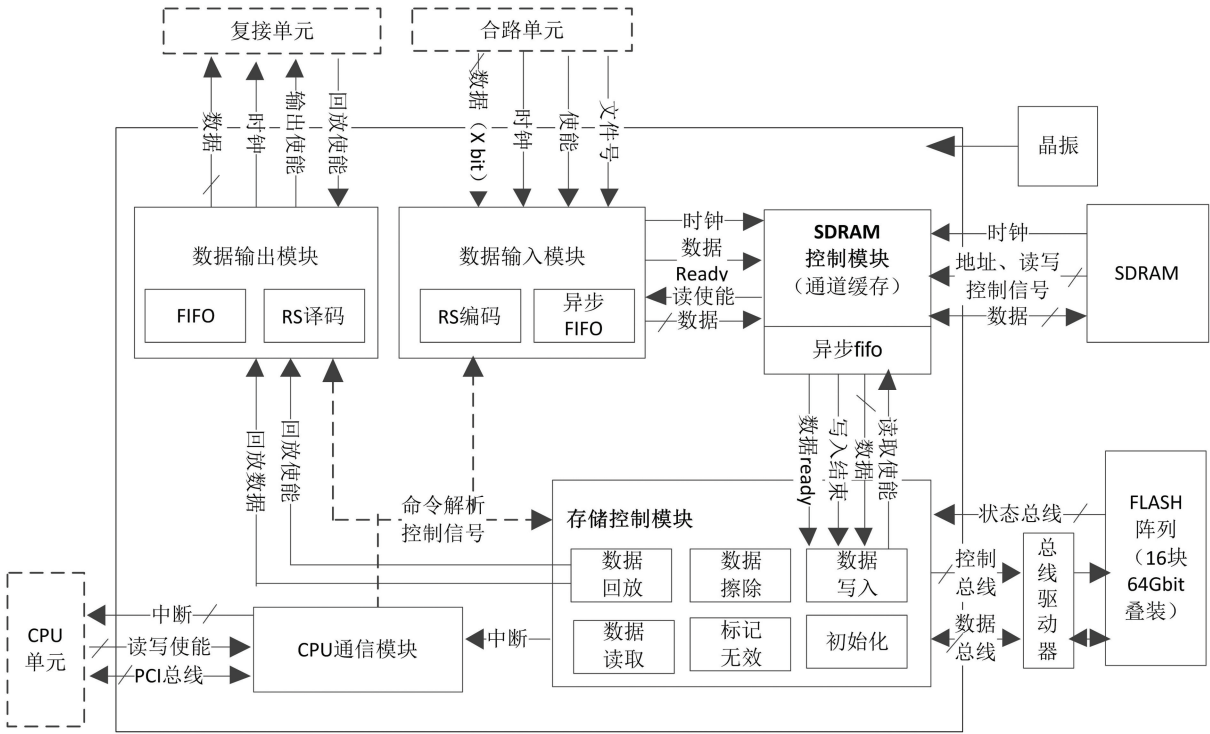


图 5 星载存储器方案框图

Fig.5 Diagram of spaceborne storage system

新型星载存储器设计参数为单板数据吞吐率 2 Gbps,存储容量 1 Tbits. 采用 FPGA 作为控制器载体,配置 SDRAM 高速并行缓存,将各载荷数据动态自主存储于 NAND Flash 中. 数管系统 CPU 单元运行存储管理软件,实现存储系统的文件化管理.

2.3 高速数据并行缓存

在多路高速载荷数据并行输入的工况下,合路单元设置 FIFO 分别接收缓存,同时设定文件号与

之对应. 随后数据被送至存储控制单元,经 RS (252,256)纠错编码后通过双 FIFO 进行乒乓缓存. 任一 FIFO 缓存满 256 字后向 SDRAM 控制模块发出读申请,SDRAM 根据载荷数据文件号将数据送往对应的分区通道进行缓存. 当任一 SDRAM 分区内缓存的数据量满足四级流水需要(即满 4 簇)时,存储控制 FPGA 与 CPU 单元协同管理,启动 Flash 写操作,将数据按四级流水写入固态存储介质. 多通道数据并行接收与缓存流程见图 6.

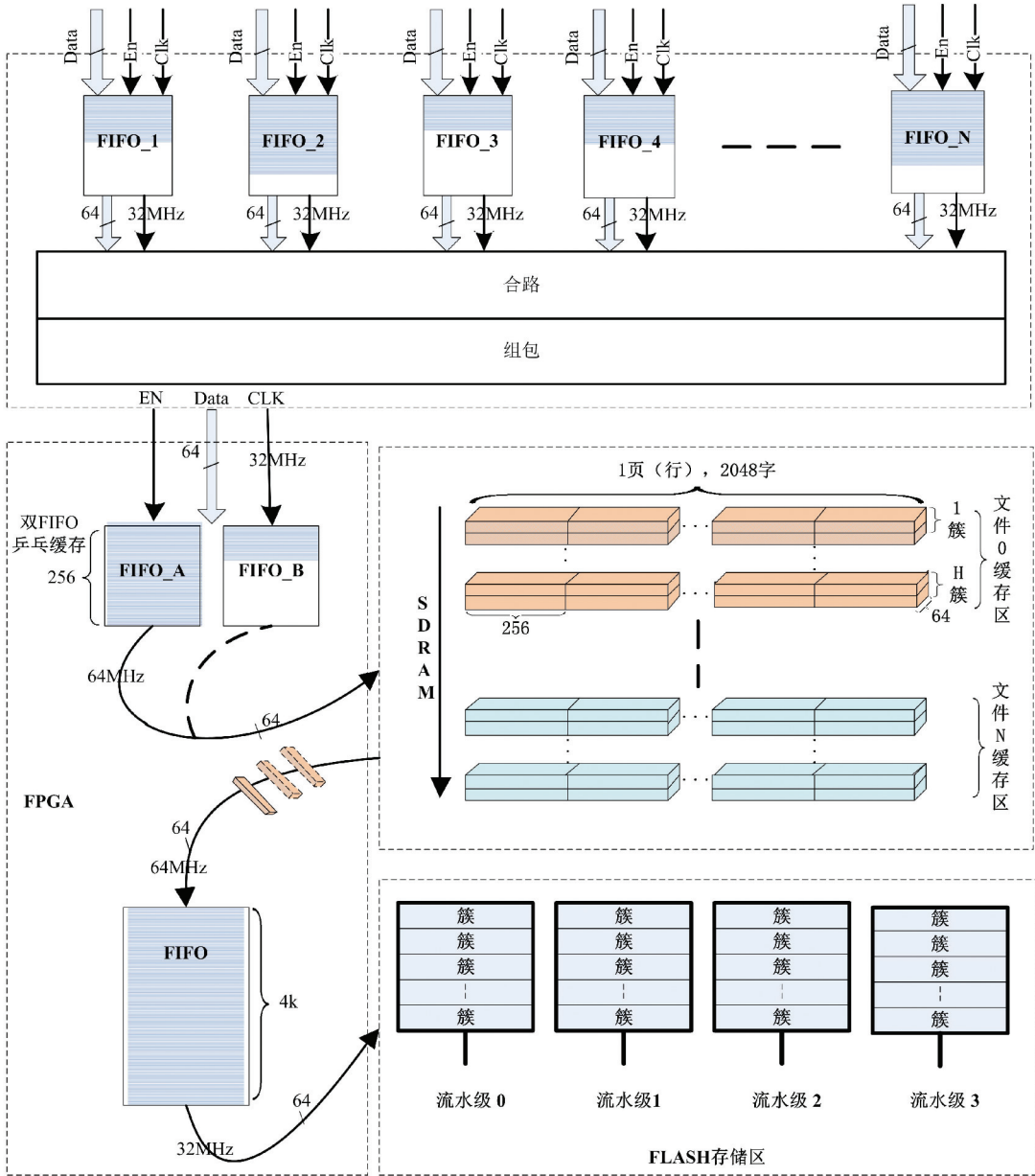


图 6 多通道数据并行接收与缓存流程图

Fig.6 Parallel reception and cache of multi-channel data

2.4 存储任务调度

由于各载荷通道数据速率的差异,导致 SDRAM 各分区内缓存数据量互不相同,当多个缓存通道同时满 4 簇大小,或者一个缓存通道中有多个 4 簇大

小缓存时,可通过通道缓存任务调度机制保证数据被完整有效存储.

设计 Flash 写操作优先级最高,读操作次之,擦除操作优先级最低,以保证 Flash 在复杂工况下

载荷数据被优先存储;设计优先对 SDRAM 通道缓存中速率高的载荷数据进行写 Flash 操作,以防止

数据缓存溢出. 通道缓存任务调度与存储流程见图 7.

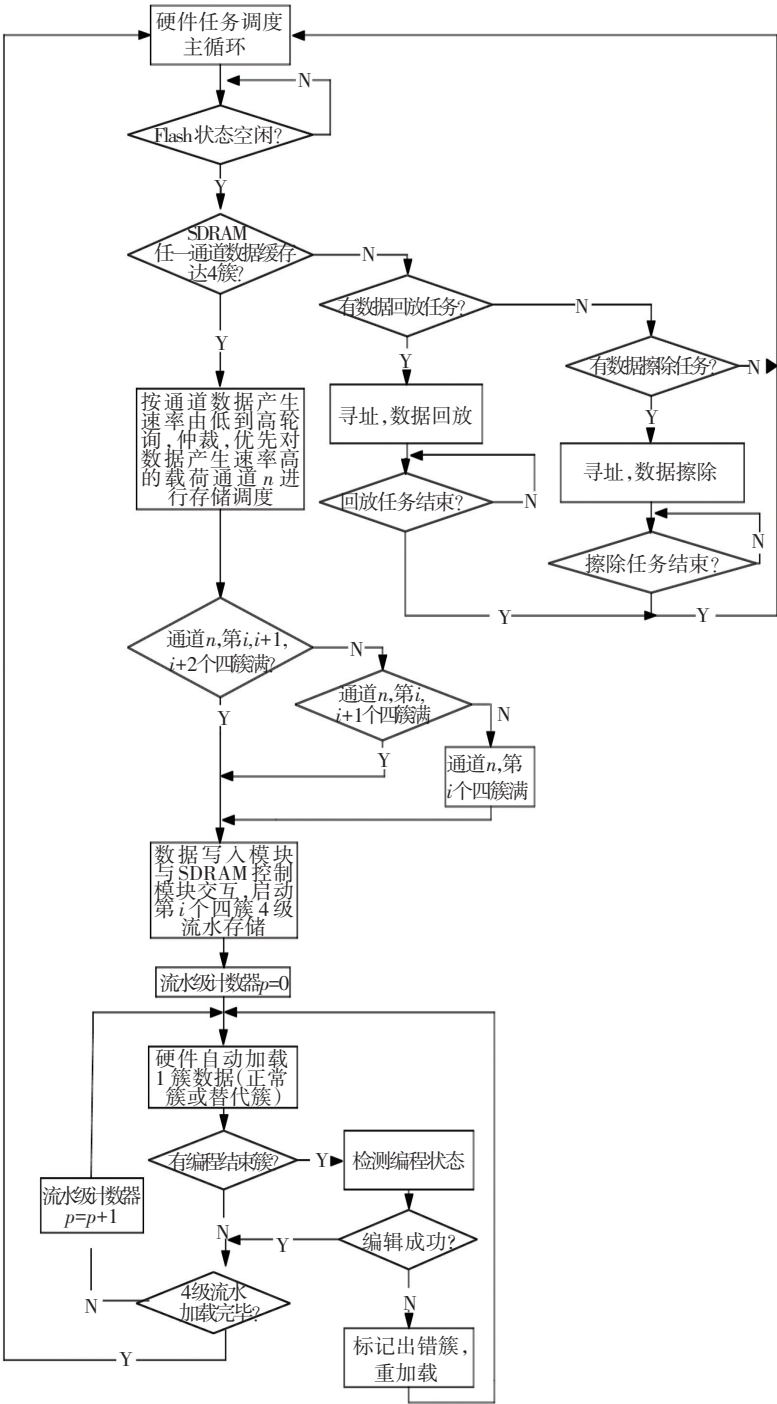


图 7 通道缓存任务调度与存储流程图

Fig.7 Channel cache task scheduling and storage process

存储控制 FPGA 根据载荷数据产生速率高低由小到大依次分配文件号,SDRAM 控制模块按文件号由大到小轮询各通道数据缓存情况,当某通道缓存数据多于 4 簇时,寄存该通道文件号. 完成所有通道遍历后,确定当前被操作通道,如果被操作缓存通道有多个 4 簇数据,则按缓存时间先后顺序进行通道内各 4 簇数据调度读取.

SDRAM 各通道需设置的缓存空间大小由多路有效载荷数据速率综合决定. 保证通道缓存不溢出的条件是存储器在极限工况下工作时 SDRAM 各通道仍有缓存空间,即在 SDRAM 各缓存通道都即将满 4 簇时, NAND Flash 启动了擦除操作,待擦除操作完成后,所有通道缓存数据被完整写入 Flash 存储区.

3 仿 真 验 证

为验证存储器高速并行缓存与任务调度机制设计正确性,使用 Matlab 工具进行模型仿真. 设置输入条件为:1)4 路载荷,数据速率分别为 480 Mbps、400 Mbps、200 Mbps、120 Mbps;2)文件号对应设定为 1~4;3)极限工况;4)载荷数据连续输入;5)Flash 擦除时间 1.5 ms,流水写 4 簇数据时间0.5 ms;6)存储无失败. 对 SDRAM 内部各文件缓存变化情况进行仿真观察,结果见图 8.

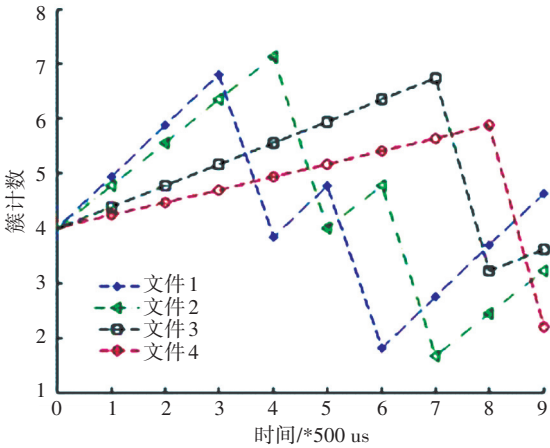


图 8 4 路文件并行缓存与存储调度模型仿真

Fig.8 Model simulation of 4 files parallel cache and storage scheduling

$t = 0$ 时刻,4 路文件缓存均将满 4 簇,由于擦除任务阻塞,Flash 不执行写操作;
 $t = 0 \sim 3$ 时,Flash 擦除操作,4 路文件持续写入 SDRAM 缓存,无写 Flash 操作;
 $t = 3$ 时刻,Flash 擦除结束,4 路文件缓存均多于 4 簇,根据优先级读文件 1 缓存写 Flash;

$t = 4$ 时刻,文件 1 写结束,文件 2、文件 3、文件 4 缓存多于 4 簇,读文件 2 缓存写 Flash;
 $t = 5$ 时刻,文件 2 写结束,文件 1、文件 3、文件 4 缓存多于 4 簇,开始读文件 1 写 Flash;
 $t = 6$ 时刻,文件 1 写结束,文件 2、文件 3、文件 4 缓存多于 4 簇,开始读文件 2 写 Flash;
 $t = 7$ 时刻,文件 2 写结束,文件 3、文件 4 缓存多于 4 簇,开始读文件 3 写 Flash;
 $t = 8$ 时刻,文件 3 写结束,文件 4 缓存多于 4 簇,开始读文件 4 写 Flash;
 $t = 9$ 时刻,因擦除阻塞累积的所有缓存文件全部写入 Flash 中,SDRAM 进入常规动态平衡调度状态.

综上,在极限工况下 4 路文件数据被连续并行接收缓存至 SDRAM 中,并按照存储优先级动态自主调度写入 Flash,期间缓存无溢出,并最终进入常规动态平衡调度状态. 仿真结果符合并行缓存与任务调度机制设计,满足了多路高速数据同时输入需求,说明方案机理设计有效可行.

为进一步验证高速并行缓存与任务调度机制的实用性,基于 FPGA 开发程序进行原型功能仿真. 参考模型仿真结果,并考虑到 FPGA 内部模块间交互时间开销与 Flash 可能存储失败重加载的影响,设置各文件缓存通道大小为 4 个 4 簇(0~3,4~7,8~11,12~15). 在与模型仿真实验相同载荷速率输入条件下,4 路文件并行缓存与存储调度原型功能仿真结果见图 9.

在原型功能仿真结果中选取部分关键时间点,观察分析 SDRAM 内各文件缓存调度变化情况见表 3.

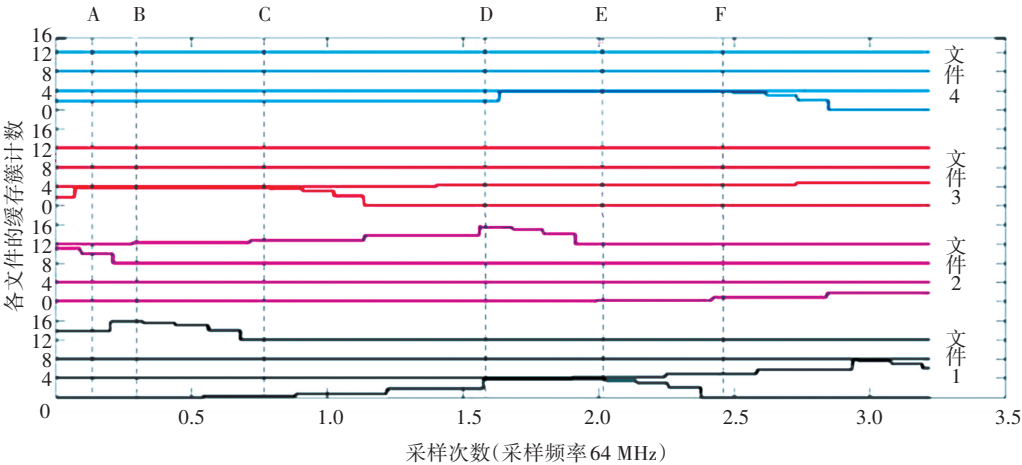


图 9 4 路文件并行缓存与存储调度原型功能仿真

Fig.9 Prototype function simulation of 4 files parallel cache and storage scheduling

表 3 原型功能仿真中各文件缓存与存储调度情况

Tab.3 Caching and storage scheduling status of files in prototype function simulation

项目	关键时间点					
	A 时刻	B 时刻	C 时刻	D 时刻	E 时刻	F 时刻
文件 1 缓存	12~15 簇未满	12~15 簇满	0~3 簇未满	0~3 簇满	0~3 簇满	4~7 簇未满
文件 2 缓存	8~11 簇满	12~15 簇未满	12~15 簇未满	12~15 簇满	0~3 簇未满	0~3 簇未满
文件 3 缓存	0~3 簇满	0~3 簇满	0~3 簇满	4~7 簇未满	4~7 簇未满	4~7 簇未满
文件 4 缓存	0~3 簇未满	0~3 簇未满	0~3 簇未满	0~3 簇未满	0~3 簇满	0~3 簇满
SDRAM 操作	正在读文件 2 并写 Flash	启动读文件 1 并写 Flash	启动读文件 3 并写 Flash	正在读文件 2 并写 Flash	正在读文件 1 并写 Flash	启动读文件 4 并写 Flash
理论存储调度	读文件 2	读文件 1	读文件 3	读文件 2	读文件 1	读文件 4

由原型功能仿真结果可知,在实际工作中 4 路载荷文件被并行接收缓存至 SDRAM 中,存储任务调度机制根据各分区缓存状态将文件按优先级自主动态写入 Flash 中,仿真结果与模型仿真结果吻合,并验证了存储器单板可有效支撑 1.2Gbps 数据输入,证明了高速并行缓存与任务调度方案的可行性和实用性。

4 结 论

针对目前星载存储器无法有效支持多路高速数据并行存储的问题,采用四级流水线操作和总线并行扩展方案,提高了存储系统对多路高速数据的存储能力;设计 SDRAM 高速多通道缓存与存储协同调度机制,实现了多路数据有效并行接收缓存与存储,确保了复杂工况下数据存储完整性. 模型仿真与原型功能仿真结果表明 SDRAM 高速多通道缓存与存储写操作协同调度机制设计有效可行。

参考文献

[1] MORGAN P S. Cassini Mission-to-Saturn Spacecraft overview & CDS preparations for end-of-mission Proximal Orbits [C]// 2015 IEEE Aerospace Conference. Big Sky, MT:IEEE Press, 2015:1-18. DOI: 10.1109/AERO.2015.7118911.

[2] FABIANO M, FURANO G. NAND flash storage technology for mission-critical space applications [J]. IEEE Aerospace and Electronic Systems Magazine, 2013, 28 (9) : 30-36. DOI: 10.1109/MAES.2013.6617096.

[3] 李姗,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 Shan, SONG Qi, ZHU Yan, et al. Design of quick initialization algorithm for space-borne solid state recorder [J]. Journal of Harbin Institute of Technology, 2015, 47 (10) : 1-10. DOI: 10.11918/j.issn.0367-6234.2015.10.019.

[4] REID M, OTTMAN G. Software controlled memory scrubbing for the Van Allen Probes Solid State Recorder (SSR) memory [C]// IEEE Aerospace Conference. Big Sky, MT:IEEE Press, 2014:1-6. DOI: 10.1109/AERO.2014.6836406.

[5] KIM J Y, PARK S H, SEO H, et al. NAND flash memory with multiple page sizes for high-performance storage devices [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016, 24 (2) : 764-768. DOI: 10.1109/TVLSI.2015.2409055.

[6] 宋琪.星载固态存储技术的应用与研究 [D].北京:中国科学院大学, 2015.

SONG Qi. Research and application of spaceborne solid state recorder technology [D]. Beijing: University of Chinese Academy of Sciences, 2015.

[7] HSIEH J W, TSAI Y L, KUO T W, et al. Configurable flash-memory management: performance versus overheads [J]. IEEE Transactions on Computers, 2008, 57 (11) : 1571-1583. DOI: 10.1109/TC.2008.61.

[8] 徐永刚.基于 NAND Flash 的嵌入式图像记录技术 [D].北京:中国科学院大学, 2013.

XU Yonggang. Embedded image recording technology based on NAND flash [D]. Beijing: University of Chinese Academy of Sciences, 2013.

[9] 李晴.高速大容量 NAND FLASH 存储系统的设计与实现 [D].北京:北京理工大学, 2015.

LI Qing. The design and realization of a high speed and large capacity NAND FLASH storage system [D]. Beijing: Beijing Institute of Technology, 2015.

[10] 徐玉杰.高速大容量存储系统的设计与实现 [D].西安:西安电子科技大学, 2014.

XU Yujie. Design and implementation of high-speed and huge-capacity storage system [D]. Xi'an: Xidian University, 2014.

[11] Samsung Electronics. K9F8G08UXM NAND FLASH memory data sheet [EB/OL]. (2007-3-31). www.datasheetpdf.com/datasheet/K9F8G08UXM.html.

[12] Orbita Control Engineering Co., Ltd. VDS3G48XQ114XX6V75 user manual (Version 1.4) [EB/OL]. [2016-11-29]. www.myorbita.net.

(编辑 苗秀芝)