

高速大容量固态存储器设计

陆 浩^{1,2}, 王振占²

(1. 中国科学院研究生院, 北京 100049; 2. 中国科学院空间科学与应用研究中心, 北京 100190)

摘 要: 为满足信息的高速大容量存储需求, 提出基于闪存(FLASH)的固态存储器设计方法。介绍 FLASH 的结构、存储操作的实现方法和高速存储等相关技术。以通用串行总线和现场可编程门阵列(FPGA)可编程设计为基础, 通过 FPGA 对多片 FLASH 的编程控制实现高速大容量存储。仿真结果证明, 该方法能实现 80 MB/s 的数据记录速度和 20 MB/s 的数据回放速度, 以及 256 GB 的存储容量。

关键词: 现场可编程门阵列; 通用串行总线; 闪存; 大容量; 流水线技术

Design of High-rate Mass-capacity Solid-state Memory

LU Hao^{1,2}, WANG Zhen-zhan²

(1. Graduate University of Chinese Academy of Sciences, Beijing 100049, China;

2. Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing 100190, China)

[Abstract] To satisfy high-rate mass-capacity data recording, a novelty record device is introduced by the paper based on FLASH. The method to operate FLASH array and the skill about high-rate record are described. Based on Universal Serial Bus(USB) and FPGA technique, several FLASH slices work together controlled by FPGA to complete high-rate mass-capacity. The design utilizes module method and pipelining technology. Simulation result shows that the system can achieve data recording speed of 80 MB/s, data playback speed of 20 MB/s, storage capacity of 256 GB.

[Key words] Field Programmable Gate Array(FPGA); Universal Serial Bus(USB); FLASH; mass-capacity; pipelining technique

DOI: 10.3969/j.issn.1000-3428.2011.15.073

1 概述

高速、大容量、高密度、低功耗、低成本的信息存储技术是卫星及其他航天设备信息获取、信息融合、信息传输和信息处理中的关键部件之一。在无人机高空飞行对目标进行侦查的时候, 有许多重要的参数和数据需要记录。该系统可以作为用于记录采集器编码输出的数据存储设备。基于FLASH(闪存)的固态大容量存储器具有容量密度大、体积小、功耗低、成本低、掉电数据不丢失、抗震动和冲击、温度适应范围宽等特点^[1]。

随着电子技术的发展, FLASH 在功耗、成本、集成度等方面相较 DRAM、磁带机等其他存储介质具有明显的优势, 已成为大容量存储器的主流存储介质。由于 FLASH 存在写入(编程)速度慢、存在无效块等问题, 本文提出了一种基于 FLASH 的实时高速大容量存储器的设计与实现方法。

2 系统设计方案

系统设计方案功能框图如图 1 所示。

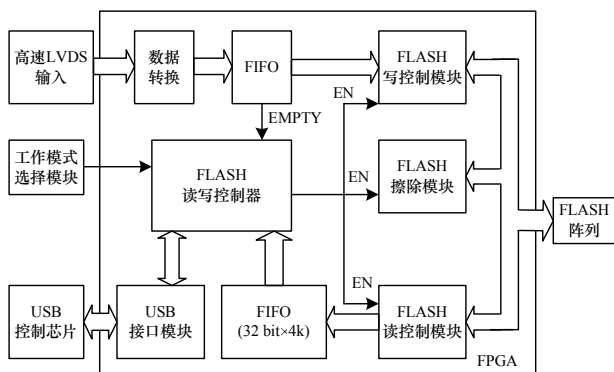


图1 系统设计方案结构

针对特定应用场合, 独立工作是本系统的重要设计目标。整个数据存储系统由 FLASH 控制模块、有效块表建立与使用模块、USB 控制模块、串/并联转换模块、片内 FIFO 等模块组成。

工作模式选择有数据记录模式、数据回放模式 2 种。模式选择通过硬件开关控制。在数据记录模式下, 系统首先进行无效块列表建立等系统初始化操作, 之后进行擦除、编程等操作。在数据回放模式下, 系统通过 USB2.0 接口与上位机相连, 接收上位机的控制指令, 读取存储设备的存储信息, 将目标数据上传到上位机本地硬盘中。系统设计的目标是: 数据记录速度为 80 MB/s, 数据回放速度为 20 MB/s, 存储容量为 256 GB。

3 FLASH 主控功能模块的设计与实现

3.1 无效块标识

无效块是指包含有一位或者多个无效位信息的存储块, NAND FLASH 在出厂时就标记本身的无效块, 每个块的第 1 页或第 2 页的空闲区的第 1 个数据如果不是 FFh, 则表示此块为无效块^[2]。为了保证 FLASH 的正确操作, 必须在操作之前建立无效块信息表, 在编程过程中产生的无效块也应及时更新到无效块列表中。

建立无效块列表的流程如图 2 所示。对于 8 片 FLASH 组成的并行扩展, 如果对并行的 8 片 FLASH 做独立的无效块列表, 不仅增加存储空间还将不利于后续的操作, 因此将它们对应块的无效块列表相与组成新的无效块列表即可。

基金项目: 中国科学院“百人计划”基金资助项目

作者简介: 陆 浩(1988—), 男, 博士研究生, 主研方向: 嵌入式系统, 集成电路设计; 王振占, 研究员、博士生导师

收稿日期: 2011-01-25 **E-mail:** luhao0408@126.com

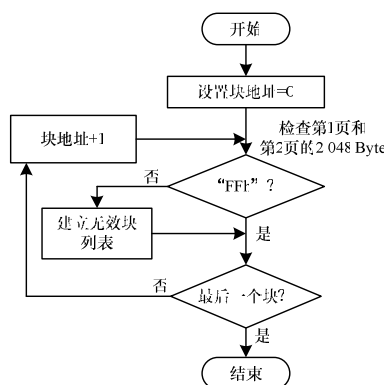


图2 无效块列表的建立流程

3.2 FLASH 基本功能实现

FLASH 擦除：器件的擦除操作是以块为单位的。块地址加载由擦除建立命令 60h 启动，然后输入确认命令 D0h，执行内部擦除过程。这一先建立再执行的 2 步命令时序，确保了存储内容不会因外部的干扰而意外擦除，擦除时间典型值为 1.5 ms。操作结束后通过检测状态寄存器的 I/O 位来判断操作是否成功，0 表示正确，1 表示有错误发生。

FLASH 读写：对 FLASH 的读写操作都是基于页的。写 FLASH 时，外部采集的数据输入 FIFO，在写模块中通过状态机实现产生 ALE、CLE、WE、RE 等控制写 FLASH 的相应时序，将数据从 RAM 中编程到 FLASH 内存，并令页地址加 1，依次循环。当总线上的第 4 组 FLASH 的页地址为 64 时，块地址加 1，并读取无效块信息，如果为无效块，则屏蔽此块，重新读取下一块信息，直至读取到正常块为止。如果是正常块，则向此块中写入数据，同时将页地址清零。读操作与写操作类似，只不过是写 FLASH 中的数据读出先送到 FIFO，然后读取 FIFO 即可。图 3 和图 4 为 FLASH 写操作 Modelsim 仿真时序界面图。

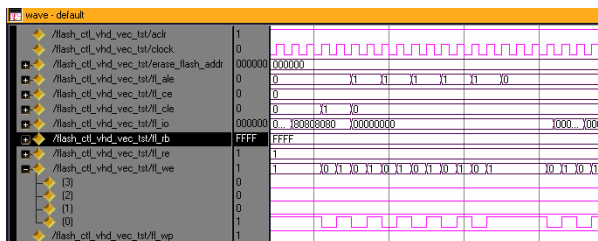


图3 FLASH 写操作仿真时序图(命令加载)

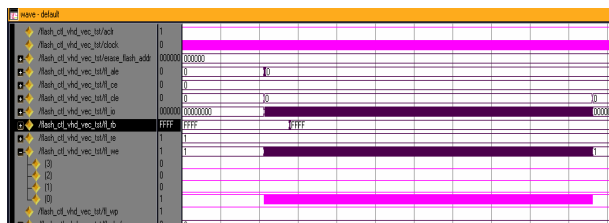


图4 FLASH 写操作仿真时序图(数据加载)

3.3 ECC 校验

在实际信道传输数据时，由于信道传输特性不理想，并且会受到噪声和干扰的影响，会造成接收端误判而发生差错，或者由于个别芯片出现读写失败，也会造成数据的丢失。为了尽可能地降低通信的误码率，提高数据存储的完整性，本系统使用循环冗余校验码 CRC 来检查和恢复错误数据，保障存储系统数据的可靠性^[3]。

循环冗余码采用 CRC-CCITT 的国际标准，其生成多项

式为： $G(x)=x^{16}+x^{12}+x^5+1$ 。硬件电路由 16 级移位寄存器和 3 个加法器组成，编码解码前将各寄存器初始化为“1”，信息位随着时钟移入。当信息位全部输入后，从寄存器组输出 CRC 结果，存储在每一页的数据共享空间内。

3.4 大容量存储扇区管理策略

本系统采用日志文件的形式存储数据信息。一片 FLASH 内部有 8 192 个存储块，同时三星公司保证第 1 个存储块为有效块。将无效块信息以及日志文件信息按特定格式存放在 FPGA 的第 1 个存储块中，当 FLASH 将要工作在数据记录模式时，FPGA 读取 FLASH 第 1 个数据块内的无效块列表信息以及已存日志文件信息，将最新日志文件的存储地址的下一个存储块作为新采集的数据的起始存储地址，当全片 FLASH 空间用完时，控制器将从 FLASH 的第 2 个块开始擦除旧数据，写入新的采集数据，数据采集结束后将当次采集的整块数据在 FLASH 内的首末地址更新在日志文件中。在 FLASH 工作在数据回放模式时，FPGA 首先读取 FLASH 第 1 个数据块内的日志文件，供上位机选择调用^[4]。

4 FLASH 高速操作技术

从前面的介绍可以看到，尽管 FLASH 芯片的读、写时钟周期为 25 ns，但是读、写以及擦除操作过程中都需要一定的等待过程，分别约为 20 ns、200 μs、1.5 ms；如果需要对存储区进行连续的操作，必将严重影响平均速率。因此，在本方案中引入了并行扩展以及流水线操作的模式，以提高存储区的平均操作速率。

4.1 流水线操作

K9K8G08U0A 的编程操作可分为 3 个步骤^[5]：(1)加载操作，主要完成编程命令、地址和数据的载入工作，时间约为 52 μs。(2)自动编程操作，即由 FLASH 芯片自动完成将载入到页数据寄存器的数据写入到内部存储单元的编程操作，耗用时间 200 μs(type)，700 μs(max)。(3)检测操作，自动编程结束后，需要检测写入的数据是否编程正确，如果不正确，需要重新编程；如果正确，才能继续后面的操作。由于每次加载(写)数据后芯片总要有大约 200 μs 的编程时间，在此期间可以立即进行另一组芯片的加载。如图 5 所示，FLASH 加载一页数据的时间大约为 200 μs 左右，最高不超过 700 μs，加载一页数据到 FLASH 寄存器约为 52 μs，700 μs/52 μs=13.5，所以理论上 K9K8G08U0A 最大可进行 14 级流水线操作，从而实现其理论最高编程速度约为 40 MB/s，本设计中 8 片 FLASH 并行扩展为 64 位总线，按记录速度 80 MB/s 的要求，单片 FLASH 的编程速度应为 10 MB/s，所以加载一页数据到 FLASH 寄存器的时间为 2 048 Byte/10 MB/s=204.8 μs，要求流水线最低级数为 700 μs/204.8 μs=3.42，可见，采用 4 级流水线操作即可保证满足设计要求。流水线控制要求外围电路必须有相当的控制逻辑实现 4 级流水之间的各步骤的交替更换，不仅要给出恰当的控制信号使各级流水线有序的运作，还要给每级流水线提供正确的命令、地址和数据，同时正确记录每级流水线的状态信息(编程和擦除是否正确的信息)。这些控制逻辑由专门的 FPGA 来完成。

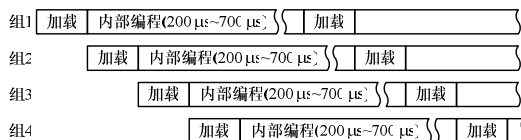


图5 流水线写操作示意图

(下转第 231 页)