

基于 USBHS 通信的低功耗存储系统设计

王继贤,任勇峰,刘利鹏

(中北大学 电子测试技术国家重点实验室,山西 太原 030051)

摘要:通过研究大容量、低功耗、高通讯速度的存储器,采用 USBHS 通信技术,该文介绍了一种大容量、低功耗的高速存储系统。系统以 GD32 为控制核心,将采集到的数据从上位机以 USBHS 通信传送到芯片,通过 EXMC 接口存储到 NAND FLASH 中。重点描述了 USBHS 通信的具体原理和 NAND flash 的存储逻辑,简要说明了系统硬件设计和整体结构。经过测试,系统功耗 90 mA,存储速率 9 MByte/s,存储容量达 32 GByte,性能良好。

关键词:USB 通信;GD32;NAND FLASH;大容量存储;高速

中图分类号:TN98

文章编号:1000-0682(2025)02-0054-06

文献标识码:A

DOI:10.19950/j.cnki.CN61-1121/TH.2025.02.010

Low-power storage system design based on USBHS communication

WANG Jixian, REN Yongfeng, LIU Lipeng

(State Key Laboratory of Electronic Test Technology, North University of China, Shanxi Taiyuan 030051, China)

Abstract: A high-capacity, low-power, high-communication-speed storage system is introduced by researching a large-capacity, low-power, high-communication-speed memory and using USBHS communication technology. The system takes GD32 as the control core, transmits the collected data from the upper computer to the chip via USBHS communication, and stores it in NAND FLASH through EXMC interface. Focusing on describing the specific principle of USBHS communication and the storage logic of NAND flash, the system hardware design and overall structure are briefly explained. After testing, the system power consumption is 90 mA, the storage rate is 9 MByte/s, and the storage capacity is up to 32 GByte with good performance.

Keywords: USB communication; GD32; NAND FLASH; mass storage; high speed

0 引言

随着现代电子技术的不断发展,各个领域所需的存储空间越来越大,存储数据的增加导致各个领域对存储器的容量、功耗、存储速率等性能提出了更高的要求^[1]。因此,低功耗、大容量存储技术成为现代电子技术的重要组成部分。

现在的存储系统多采用 FPGA、STM32 作为主控芯片,存在功耗较高、成本昂贵的问题,GD32 作为国产微控制器不断推陈出新,性能优越、成本低

廉,可以替代国外芯片作为控制核心^[2]。NAND Flash 作为新型半导体器件,具有非易失性、低功耗、大容量的优点,成为现代电子系统首选的存储器件^[3]。所以,选用合适的微控制器和高性能的 NAND Flash 可以满足很多需要低功耗、大容量存储项目的需求。同时在系统设计中,常用通信方式有 SPI、USART、以太网、USB 等,SPI、USART 通信速度最高只有 4 MB/s 左右,不适用于大容量数据传输,以太网通信成本高^[4]。USB 是一个外部总线标准,是设备与 PC 端之间常用的通信方式,它的大容量(Bulk)传输方式适用于大容量数据传输,且设备与 PC 端之间只需用一根 USB 线连接,方便系统开发,成本低廉^[5]。

因此,为解决存储器中造价昂贵、容量小、功耗

收稿日期:2024-07-23

第一作者:王继贤(1999—),女,山西忻州人,硕士研究生,研究方向为测试计量技术及仪器,电路与系统等。

高等问题,该文在现有研究的基础上,将 FPGA 替换成 GD32,通过 USBHS 通信,选择使用 EXMC 接口连接多片 NAND Flash 并行处理实现对大容量数据存储,并在 PC 端进行实时控制。

1 总体方案设计

该系统采用国产微控制器 GD32F470VGT6 作为核心,两片 NAND Flash 为存储介质。GD32F4 系列性能优于 GD32F1、GD32F2 等系列,主频高达 240 MHz,包含 EXMC 接口、USBHS 通信等,同时 GD32F470 在 GD32F4 系列中功耗低,成本不高,还具有大容量存储优势,GD32F470VGT6 引脚为 100,体积较小,因此符合系统集成度高、低功耗、低成本的需求^[6]。

USB 一直在随着时代发展,到现在已经发展出 USB1.0/2.0/3.0 等多个版本的通信协议,其速度也在不断翻倍提升。GD32F470 支持 USB2.0 协议,在 USBHS 模式下通讯速率可以到达 60 MB/s,远高于 USART、SPI、I²C 的通讯速率^[7]。USBHS 为 USB 的高速模式需要外接物理层芯片(USB PHY),该系统选用 USB3300 外接来进行通信。

外部存储器控制器(EXMC),GD32 系列微控制器通过其访问 SRAM、NAND Flash 等存储器,其一端与 AHB 总线连接,另一端与外部存储器相连,微控制器可通过总线与外部存储器进行数据交换。EXMC 可以采取低功耗设计,在保证存储速度的前提下降低功耗,符合存储器低功耗的需求。

该系统使用 PC 端的串口助手将采集到的数据使用 USBHS 通信方式发送到主控制器,通过 GD32 内部的 EXMC 接口访问 NAND Flash,采用乒乓缓存将数据存储到 NAND Flash 中,同时对 NAND Flash 存储进行坏块管理、ECC(Error Checking and Correction)纠错确保数据的可靠性。在 PC 端需要时,发送指令,将 NAND Flash 中存储的数据通过 USBHS 发送到 PC 端进行处理和观测。该系统通过 Type-C 接口数据线与 PC 端进行连接。系统整体架构如图 1 所示。

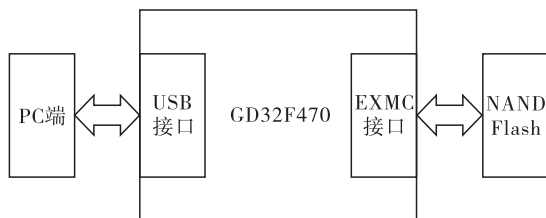


图1 系统整体架构图

系统通过 PC 端发送擦除、记录锁定、下载、复位的指令来控制存储器的操作。存储器有四种工作状态,分别是空闲状态、记录锁定状态、擦除状态、下载状态。当系统上电后,PC 端无指令发送时,存储器自动进入空闲状态,一直保持写数操作,进行数据存储,同时可响应所以指令,首次上电存储器从 NAND Flash 首地址开始存储数据;当 PC 端发送下载指令时,存储器进入下载状态,从 NAND Flash 首地址开始读出所存储的数据;当 PC 端发送到复位指令时,存储器从下载状态退出,停止读数,返回空闲状态;当 PC 端发送到擦除指令时,存储器进入擦除状态,擦除保存的数据,并进入记录锁定状态;当 PC 端通过 USB 发送记录锁定指令时,存储器从当前状态退出,进入记录锁定状态,不再响应任何指令,直到掉电后再上电。存储器工作逻辑如图 2 所示。

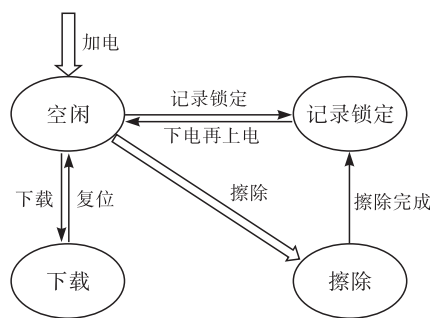


图2 存储器工作逻辑图

2 USBHS 通信设计

USBHS 通信部分主要负责数据传输和指令接收、识别。根据 USB2.0 协议,USB 被分为设备(USB Device)和主机(USB Host)、互连(USB Hub),因为系统需要与 PC 端进行通信,所以设置为 USB 设备类方便操作。而 USB 设备可以设置成鼠标、键盘、虚拟串口(CDC)等,因为虚拟串口方便与 PC 端通信,且虚拟串口不受 PC 端设置的波特率的限制,可以到达 USB 接口传输本身的速率,所以系统将 USB 枚举为虚拟串口进行通信。

同时为防止在传输过程中,接收的指令某一位或某几位出现错误导致存储器无法识别指令,系统采用 LRC 校验进行指令识别,异或校验和计算简单迅速,可以在不影响数据传输速度的前提下,进行指令识别和响应,提高 USBHS 通信的可靠性。

USB2.0 协议通过 D+ 和 D- 这两根数据线采用差分电压的方式进行数据传输,存储器为高速模

式,需在 D+ 上拉电阻接到 VCC,让主机判断到有高速设备接入,进行对应操作^[8]。

因为 PC 端一般不具备 USB 虚拟串口的驱动程序,要想正常识别,需编写驱动程序,该系统选择从官网下载驱动。USB 虚拟串口通过 OUT 端点接收数据,放入数据缓存区进行操作,发送数据时通过 IN 端点将处于数据缓存区的数据发送到 PC 端。虚拟串口与 PC 端正常通信后,会在 PC 端以串口形式出现。

USB 虚拟串口采用中断接收,每当有数据传输到 GD32 时,会触发接收中断,记录该次数据长度并将其放入缓存区中,每触发一次接收中断,接收的数据长度就会累加,当其数据总长度满 32 KB 时发送一个标志位,表示 NAND Flash 可以进行写操作,并清空缓存区以便下次接收。USBHS 在接收到指令时,会进行识别并执行相应操作,系统通过纵向冗余校验(LRC)方法识别指令,将指令数据的所有字节进行异或得到一个字节,也就是 XOR 校验和,不同指令通过数据校验得到各自的校验和来执行对应的程序。USB 虚拟串口通信程序框图如图 3 所示。

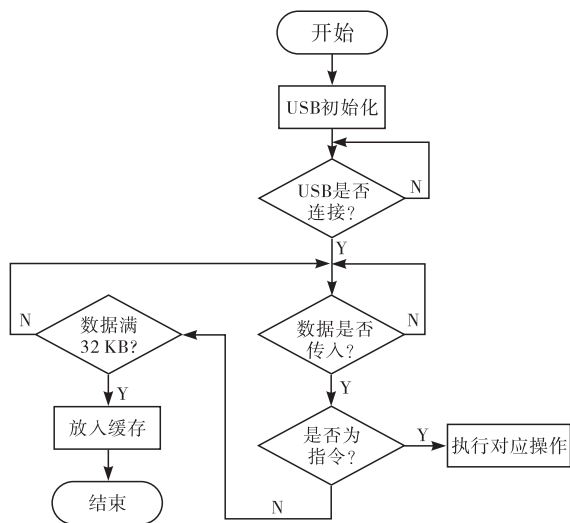


图 3 USB 虚拟串口通信程序框图

3 NAND Flash 存储设计

GD32 通过 EXMC 接口将数据存储到 NAND Flash 中,构建控制逻辑控制存储器的读写、擦除,采用乒乓缓存提高存储速度,设计坏块管理提高数据存储的可靠性和存储速度,降低系统功耗,并对数据进行 ECC 校验,确保传输到 PC 端的数据的正确。

该系统选用两片大容量 NAND Flash 为镁光公

司的 MT29F128G08AJAAA 芯片,一片存储容量为 16 GB,两片共 32 GB,满足系统需求。同时为保证数据的存储速度,该系统采用两片 NAND Flash 并行的办法,由模拟开关 CH444 控制两片 NAND Flash 的使能。

3.1 存储器控制逻辑设计

MT29F128G08AJAAA 内部结构如图 4 所示,含有两个 CE 和 R/B,每个 CE 和 R/B 共同控制一个 Target,LUN 为芯片最小的独立单元,两片 Flash 共 4 个 CE 和 4 个 R/B。模拟开关 CH444 控制 CE 和 R/B 的开关,从而控制开启对应的 Target。模拟开关 CH444 通过两个输入端 IN0、IN1 进行 8 个引脚的切换,两个输入端接在 GD32 的 IO 口上通过程序使能,IN0、IN1 有 00、01、10、11 四种状态,每种对应一个 Target。

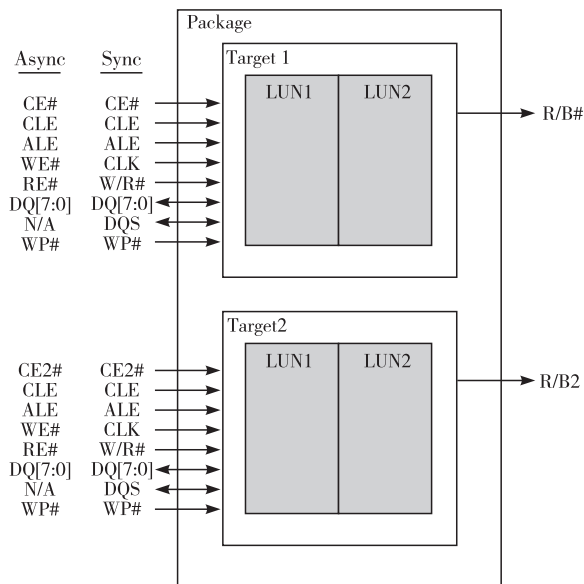


图 4 MT29F128G08AJAAA 内部结构图

该系统以 Target 为单位,每个 Target 大小为 8GB,共有 8192 块,每页大小为 8 KB,两片 NAND Flash 共四个 Target,LUN 内部结构如图 5 所示。每 LUN 分为两个 plane,NAND Flash 芯片为按页写入,每次进行读写操作按顺序向每个 Target 的同一页进行写入和读出,使能顺序为第一片 NAND Flash (CE, R/B) Target1、第一片 NAND Flash (CE2, R/B2) Target2、第二片 NAND Flash (CE, R/B) Target3、第二片 NAND Flash (CE2, R/B2) Target4,因此一次读写数据量为 32 KB。存储器首次写入从首地址开始,写满 8192 块后会重新返回首地址覆盖之前的数据进行存储。

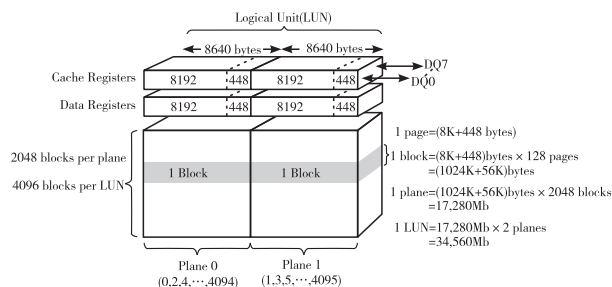


图5 LUN内部结构图

系统在连接到PC端后,执行NAND Flash初始化操作,并进行软件复位,使能Target1,读取Target1的ID,在读出的ID与数据手册所给ID一致后,进行后续操作。

为保证NAND Flash存储速度,在参照数据手册后选择了读写操作延迟时间最小的模式5。

在对页进行写操作前,需要进行擦除操作,NAND Flash芯片为按块擦除,系统每写满一块提前对下一块进行擦除,擦除时分别使能Target,对每个Target的块进行擦除,擦除时会根据坏块表跳过坏块。

存储器上电后进入空闲状态,系统会不断检测USBHS通信接收缓存区的数据是否满32KB,满足条件时返回一个标志位,系统开始进行写操作。在写操作之前会遍历坏块表检测所操作的块是否为坏块,若是则跳过该块,对下块进行相同操作。

为了避免NAND Flash存储太慢导致数据丢失,系统采用乒乓缓存。乒乓缓存是使用两个缓存区来交替存储数据,以实现数据的快速读写,避免缓存冲突。USB虚拟串口接收缓存区分为1和2,在缓存区1接收满32KB后,返回标志位,系统开始写操作,同时虚拟串口继续接收数据放在缓存区2,在系统完成存储后,缓存区2满32KB,系统继续对缓存区2数据进行存储,虚拟串口清除缓存区1的数据继续进行接收,不断循环往复,提高了系统存储速度,减少了存储数据的丢失。

在接收到下载指令后,系统从NAND Flash中读取数据,在读操作前会进行坏块检测,防止数据出现错误。因为读出速度小于通讯速度,所以不用担心数据溢出问题,为避免读空现象出现,读出数据后会返回一个标志位,在接收标志位且虚拟串口连接到PC端时,USB虚拟串口才开始向PC端发送数据。同时系统执行擦除指令后,数据都被清除,不会再响应下载指令,当重新有数据写入

后,可响应下载指令。系统写操作程序逻辑如图6所示。

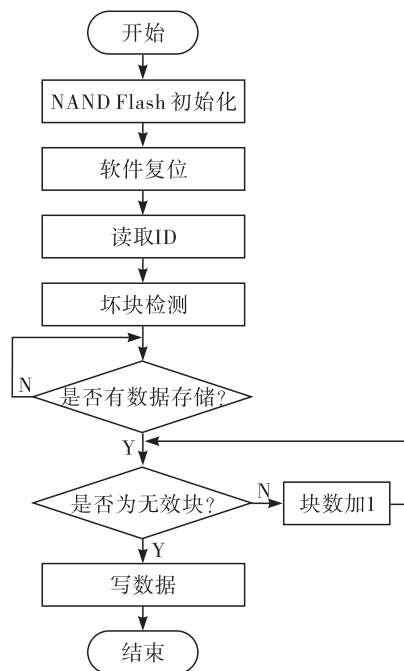


图6 系统写操作程序框图

3.2 存储器坏块管理设计

NAND Flash一般在出厂时就存在坏块,且在实际存储过程中随着读写、擦除次数变多会产生新的坏块,造成存储器无法正常工作,会影响系统稳定,降低存储速度^[9]。因此该系统对NAND Flash进行坏块管理,存储前需要对所有的块进行遍历扫描,进行读写测试并创建坏块表,降低系统功耗,提升NAND Flash存储速度和数据可靠性^[10]。

通常厂商在出厂时会对坏块进行标记,不同厂商存放标记的位置会有差异,该系统所用的MT29F128G08AJAAA会将坏块的值标记为0x00^[11]。系统在初始化完成后会进行坏块检测,读取每个Target的第一页备用区域的第一个字节,每次分别使能四个Target进行读操作,每次四个Target读出来的字节相与,结果不为0xFF,标记为坏块,当检测出坏块时,该次四个Target的块均标记为坏块,不再对其进行读写操作。每标记一个坏块就将其放入数组中,建立坏块表,每次上电后都会重新进行一次坏块检测建立新的坏块表,减少因擦除、读写新产生的坏块对数据读写的影响。系统每次读写新的块时,会遍历一次坏块表,若为坏块则跳过此块,对下一块进行操作。系统坏块管理程序框图如图7所示。

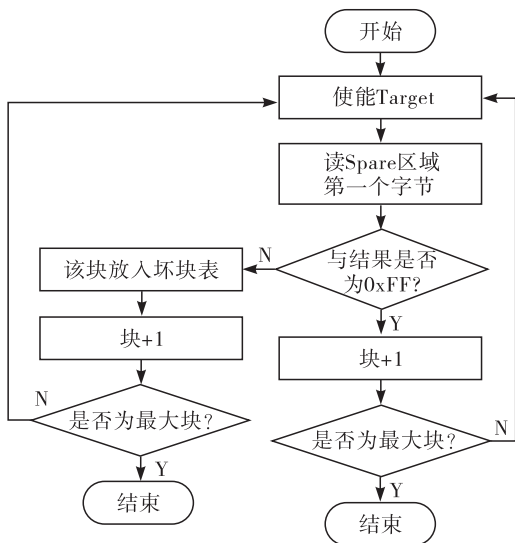


图 7 坏块管理程序框图

3.3 存储器 ECC 校验设计

由于系统在大量数据存储的情况下出现错误的概率会变高,为保证数据可靠性,需要采用校验算法对其进行检错、纠错。由于 NAND Flash 读写出错时一般不会造成一整页或整块的错误,而是一页里有一个或几个位出错,因此系统选用 ECC 校验。ECC 校验计算速度快,可及时检测,可以在保证存储速度的前提下提高数据存储的可靠性。

基于汉明码的 ECC 校验算法能纠正单比特错误并检测双比特错误,但对单比特以上的错误无法纠正,不能保证检测到两比特以上的错误^[12]。ECC 校验将数据块看作一个矩阵,利用矩阵的行列奇偶信息生成 ECC 校验码。ECC 每 256 字存储数据生成 3 字节的 ECC 校验数据,即 6Bit 的列校验和 16Bit 的行校验,多余 2Bit 置 1^[13]。系统 EXMC 接口对于 NAND Flash 内置硬件 ECC, NAND Flash 一页 8192 字节,在进行按页写入操作时,系统会将计算结果存储在 ECC 结果寄存器 (EXMC_NECCx) 的 ECC[31:0] 位域。

系统在进行写入操作时,在数据写入后会生成 ECC 校验码,从寄存器中提取出计算结果并放入该页的 Spare 区域。在读操作时,读出的数据再次生成新的 ECC 校验码并与从 Spare 区域读出的原有 ECC 校验码进行按位异或,若结果为 0,则表示在存储过程中数据未出现错误,若结果有一个 1,则表示存储的 ECC 有一位出现错误,若出现其他情况,则表示出现一个以上错误无法纠正。系统在检测到一个 Bit 错误后,通过定位将其进行翻转纠正。系统 ECC 校验程序框图如图 8 所示。

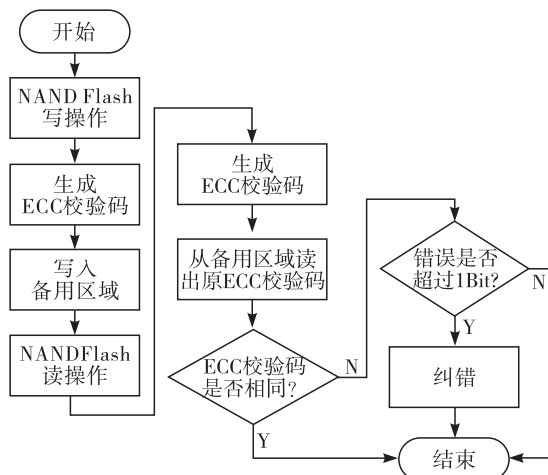


图 8 ECC 校验程序框图

4 系统测试

系统采用市面上的 UartAssist 串口调试工具进行通信,电脑识别到 USB 虚拟串口“GD32 Virtual Com Port”,开始进行数据存储。USB 虚拟串口如图 9 所示。



图 9 USB 虚拟串口

存储器在上电初始化后,进行坏块检测,检测到的坏块放入数组 ivld_blc_addr 中,数组保存的坏块数据可在 keil 软件调试的监控窗口看到,坏块管理软件测试如图 10 所示。

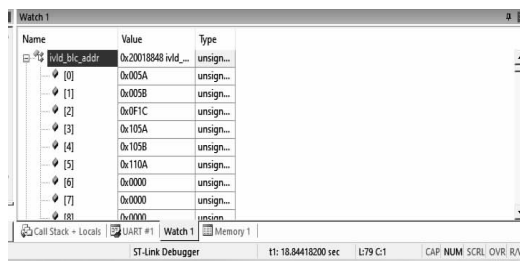


图 10 坏块软件测试图

表 1 不同数据量下存储速度

数据量	时间	存储速度
100 MB	10.6 s	9.4 MB/s
500 MB	53.1 s	9.4 MB/s
1 GB	111.3 s	9.2 MB/s
5 GB	568.8 s	9.0 MB/s
10 GB	1163.6 s	8.8 MB/s

存储器的存储速度如表 1 所示,存储器进入空闲状态,通过 PC 端将数据通过 USB 虚拟串口发送到 GD32,再写入 NAND Flash,然后 PC 端发送下载指令,将存入的数据读出。存储器设计有 LED 灯,LED 灯亮表示开始存储,LED 灯灭表示存储完成,通过记录亮灭时间,计算平均写入的速度,测得存储速度为 9 MB/s 左右。

系统测试所用数据为帧计数,数据从 0 开始自增,每 256 个数为一次组,下次重新开始从 0 自增,数据末尾两个字节保持 0xEB、0x90 不变,倒数第三到六个字节从 0x00、0x00、0x00、0x00 开始每进行一次循环加 1,通过判断最后六个字节的数字是否连续来看数据存储有无出现错误,存储逻辑是否正确。经过系统测试写入和读出的数据相同,传输过程中未出现丢帧、错帧问题。数据下载结果如图 11 所示。

B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF	01
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	8B	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	8C	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	8D	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	8E	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	8F	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	90	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	91	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	92	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	93	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	94	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	95	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	96	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	97	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	98	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	99	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	9A	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	9B	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	9C	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	9D	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	9E	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	9F	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A0	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A1	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A2	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A3	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A4	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A5	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A6	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A7	EB	90	
.B	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	00	00	4E	A8	EB	90	

图 11 数据下载结果

系统通过电源发生器供 5 V 电压,使用 2 通道连接存储器,实验室所用电源发生器测存储器功耗只能显示到安培小数点后两位,因此系统时钟 240 MHz 时,整体功耗在 90 mA 左右。存储器功耗测试如图 12 所示。

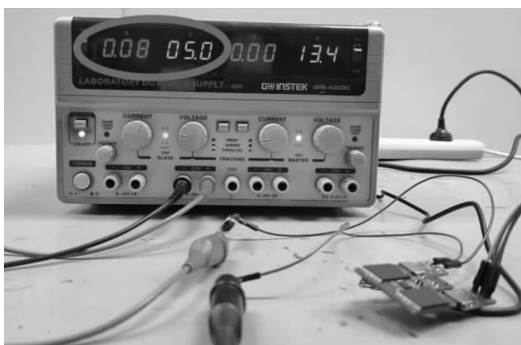


图 12 存储器功耗测试图

5 结论

该系统采用 GD32、USBHS 通信技术、多片 Flash 并行技术,克服了之前存储器功耗过高、体积大的问题,同时选用国产 GD32 微控制器降低成本,是一款体积小、功耗低、容量大、成本低的存储器。经过验证,该系统数据存储功能正常,可靠性好,满足低功耗、大容量存储需求。

参考文献:

- [1] 程晓航. 基于 FPGA 的 SATA 盘大容量存储系统的设计与实现[D]. 西安: 西安电子科技大学, 2019.
- [2] 于志刚,侯伟盟,倪文龙. 基于 GD32F407 的国产化电机驱动系统设计[J]. 集成电路与嵌入式系统, 2024, 24(06): 61-66.
- [3] 徐磊,王保成. 基于地址映射的 NAND Flash 控制器设计[J]. 计算机测量与控制, 2023, 31(06): 109-122.
- [4] 蒲泽坤,沈勇,陈旅超. USB 接口的 CDC&MSC 复合设备设计与应用[J]. 单片机与嵌入式系统应用, 2023, 23(7): 53-56+61.
- [5] 庞彬尧. 基于 STM32 和 USB 的大容量存储器的设计[D]. 太原: 中北大学, 2020.
- [6] 冯帅,左官芳,范阳. 基于 GD32F450 的视频监控终端的设计[J]. 电子设计工程, 2023, 31(05): 189-193.
- [7] 陈旭辉,杨红云. USB 接口的虚拟多串口通信设备设计[J]. 单片机与嵌入式系统应用, 2020, 20(4): 18-21.
- [8] 赵计贺,李杰,张德彪,等. 基于 USB3.0 的高速数据回读系统设计[J]. 中北大学学报(自然科学版), 2021, 42(2): 187-192.
- [9] 申晓敏,钱礼华,杜剑英,等. 基于 USB2.0 总线的 NAND Flash 检测及控制方法[J]. 电子技术应用, 2023, 49(6): 44-48.
- [10] 张雪钰,焦新泉,刘东海. Flash 存储阵列坏块管理方法的设计与实现[J]. 电子测量技术, 2020, 43(20): 21-25.
- [11] 李中,周加谊,曹睿. 基于 NAND Flash 固态硬盘的坏块管理方法[J]. 电子设计工程, 2022, 30(23): 24-27+32.
- [12] 段曦冉. 一种基于 flash 的硬件 ECC 设计原理及实现[J]. 微处理机, 2023, 44(06): 15-18.
- [13] 谭继鹏. 一种基于 FPGA 的 NAND FLASH 控制器的校验方法[J]. 航空计算技术, 2023, 53(2): 84-87.