综合工程设计铁电存储测试仪：嵌入式系统设计

1. 功能概述

通过软件C语言编程，我们实现了基于上位机控制的铁电存储器全零全一测试、简单数据或文件发送与接收比对测试、铁电存储器疲劳测试、铁电存储器断电测试、修改SPI速率，基于串口屏控制的铁电存储器全零全一测试、铁电存储器疲劳测试、铁电存储器断电测试。

1. 实现平台

嵌入式系统设计采取Keil MDK-ARM开发环境、C语言编程、调用HAL库进行开发，采用STM32F302系列的单片机。

1. 系统整体设计

整个系统我们采用并行运行的方式，根据不同模块响应速度的要求，划分为三类：每次主函数中循环一次即调用；每个1ms执行一轮；每100ms执行一轮。

所有执行模块均在主循环中调用，其中主循环每次调用的部分只包含了单次的串口接收转发模块，每100ms执行一次的模块只包含了程序指示灯的闪烁功能，其余主体功能均在每1ms执行一轮的模块中执行，其中每1ms执行一轮的模块利用系统时钟每1ms中断一次的中断回调函数判断执行进度。

主函数执行部分流程图展示如下：

主函数流程图

图1 主函数流程图（左）与每1ms执行一轮的模块的流程图（右）

1. 各功能实现算法设计

本实验中，首先利用串口中断回调函数解析收到的指令，每收到1字节即进行一次回调，同时利用switch语句判断当前接收到的指令的位置以及指令的参数，由此更新标志位在主函数中执行相应操作。

在实际算法编写过程中，为了达成类似FPGA的多线程并行处理，我们采用的是类似时序逻辑的编写方法，设定1ms为一个时钟周期，在每个时钟周期中进行相应功能的处理，每个功能基本都会占用多个时钟周期，我们将每个功能的步骤进行拆解，分散在多个时钟周期中进行避免某一功能长期占用主函数影响其他效果的实现。由于我们的实验中只是对单个铁电芯片进行处理，并不能同时对铁电芯片进行多种操作，主要并行处理的部分是对铁电芯片读写的操作与整个项目的LED灯闪烁这两部分，以及单片机对铁电芯片的SPI读写操作与单片机借由UART串口与上位机或串口屏的通信并行处理。

1. 全零全一测试

全零全一测试意在对铁电存储器芯片进行遍历，测试芯片中是否有某个存储单元处于常态0或常态1的状态，若如此则说明该铁电存储器存在问题，无法正常进行存储。我们在此将0x00或0xFF依次存储进铁电存储器各个地址对应的内存中，并依次读取，查看读取到的数据是否为0x00或0xFF，如果是则测试通过，向上位机反馈每个存储部分的错误个数，如果是串口屏发来指令则反馈每个存储部分错误与否。

上面描述的是所要求的功能的大致实现思路，全零全一测试在主函数中每隔1ms进行调用一次，此流程图反映的是被调用的函数的处理流程。流程图展示如下。

全零全一流程图

图2 全零全一流程图

1. 数据收发测试

数据收发测试用于测试较大文件是否可以正确存储于铁电存储器中，我们先利用发送数据包的编号得到可以存储的地址并保存进数组，再存储进对应地址，读取数据时我们将利用要获取的数据包的编号从数组中获取对应地址，再从铁电存储器芯片中获取存储到的数据并反馈。大致操作过程的流程图展示如下。

数据收发流程图

图3 数据收发流程图

1. 疲劳测试

疲劳测试是在短时间内连续对铁电存储器进行发送读取效果，检测是否会出错。我们采取对其连续进行全零全一测试的方式进行疲劳测试，每隔特定次数向上位机或串口屏返回错误次数，然后由串口屏或上位机发送对应指令结束疲劳测试。开启疲劳测试指令收取后的效果是置位疲劳测试开启标志位，结束疲劳测试指令收取后的效果是复位疲劳测试进行标志位，一旦停止疲劳测试将会终止当前疲劳测试的进度直接结束。

上面描述的是所要求的功能的大致实现思路，疲劳测试在主函数中每隔1ms进行调用一次，此流程图反映的是被调用的函数的处理流程。疲劳测试大致的流程图展示如下。

疲劳测试流程图

图4 疲劳测试流程图

1. 断电测试

断电测试是为了查看该芯片在断电的情况下是否能够进行有效存储。先对铁电存储器发送全一，断电一定时间后再度开启电源，从中读取数据，查看存储情况，然后再发送全零，断电一定时间后再度开启电源，从中读取数据，查看存储情况，若两次全部正确，即可判定该芯片存储性能正常。

上面描述的是所要求的功能的大致实现思路，断电测试在主函数中每隔1ms进行调用一次，此流程图反映的是被调用的函数的处理流程。断电测试大致流程图如下展示。

断电测试流程图

图4 断电测试流程图

1. 修改SPI速率的测试

为了查看不同SPI速率情况下铁电芯片是否能够正常存储，我们修改SPI传输速率，然后在不同SPI速率的情况下对铁电芯片进行读写，查看结果。SPI速率主要依赖系统时钟，我们通过修改预分频器的数值可以修改SPI的速率。当预分频器数值为4的时候，SPI传输速率为18Mbps；当预分频器数值为4的时候，SPI传输速率为18Mbps；当预分频器数值为8的时候，SPI传输速率为9Mbps；当预分频器数值为16的时候，SPI传输速率为4.5Mbps；当预分频器数值为32的时候，SPI传输速率为2.25Mbps；当预分频器数值为64的时候，SPI传输速率为1.125Mbps；当预分频器数值为128的时候，SPI传输速率为562.5Kbps；当预分频器数值为256的时候，SPI传输速率为281.25Kbps。修改预分频器后重新对SPI模块进行初始化，即可达到此效果。

修改SPI速率大致流程图展示如下。

SPI速率修改功能流程图

图4 修改SPI速率功能流程图