

使用 STM32 的 SPI 控制器, 对外部的 W25X16 FLASH (2M 字节) 芯片进行操作。本文中的波形为逻辑分析仪抓取的时序图。

以下是该 FLASH 芯片的命令表。

11.2.2 命令表

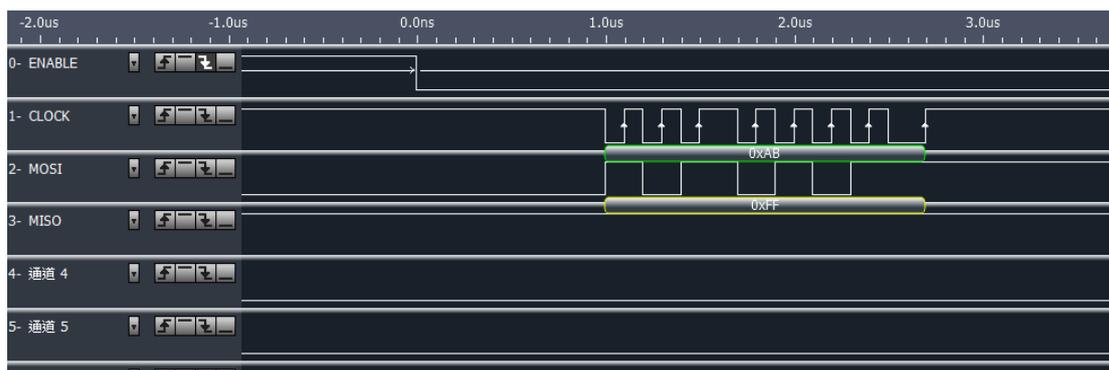
指令名称	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	下一个字节
写使能	06h						
写禁能	04h						
读状态寄存器	05h	(S7~S0)					
写状态寄存器	01h	S7~S0					
读数据	03h	A23~A16	A15~A8	A7~A0	(D7~D0)	下个字节	继续
快读	0Bh	A23~A16	A15~A8	A7~A0	伪字节	D7~D0	下个字节
快读双输出	3Bh	A23~A16	A15~A8	A7~A0	伪字节	I/O=(D6,D4,D2,D0) O=(D7,D5,D3,D1)	每四个时钟一个字节
页编程	02h	A23~A16	A15~A8	A7~A0	(D7~D0)	下个字节	直到 256 个字节
块擦除 (64K)	D8h	A23~A16	A15~A8	A7~A0			
扇区擦除(4K)	20h	A23~A16	A15~A8	A7~A0			
芯片擦除	C7h						
掉电	B9h						
释放掉电/器件 ID	ABh	伪字节	伪字节	伪字节	(ID7~ID0)		
制造/器件 ID	90h	伪字节	伪字节	00h	(M7~M0)	(ID7~ID0)	
JEDEC ID	91h	(M7~M0)	(ID15~ID8)	(ID7~ID0)			

一、读取器件 ID

先拉低片选信号 CS, 再发送命令 0XAB, 再发送三个字节的 dummy。读取第四个字节数据, 数据就是 device ID。最后拉高 CS, 结束一次操作。



以下是发送命令 0XAB 放大的图。可以看出 CLK 空闲状态是高电平 (CPOL = 1), 偶数边沿为采样时刻 (CPHA = 1)。



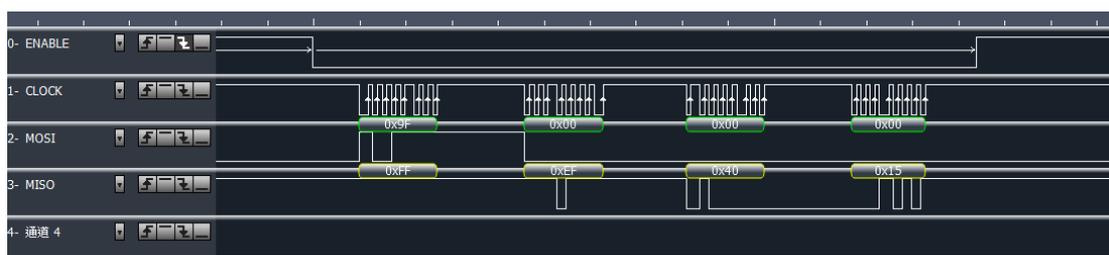
二、 读取 jedec 标准 ID

先拉低片选信号 CS，再发送命令 0x9f，读取三个数据。最后拉高 CS，结束一次操作。

第一个数据：生产厂商

第二个数据：存储器类型

第三个数据：容量



三、 写使能

Flash 在写数据，擦除，写状态寄存器之前，首先要打开写使能。打开写使能的方法也很简单，发送命令 0x06 即可。

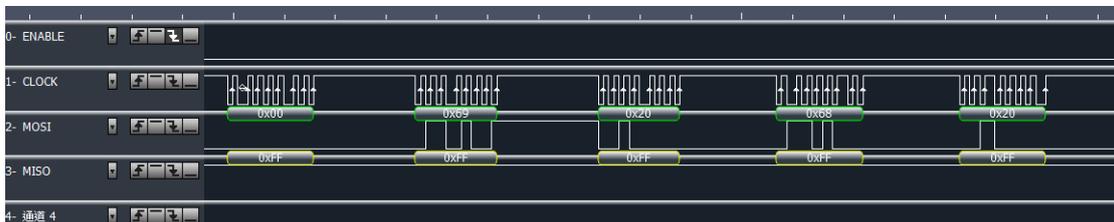


四、写数据

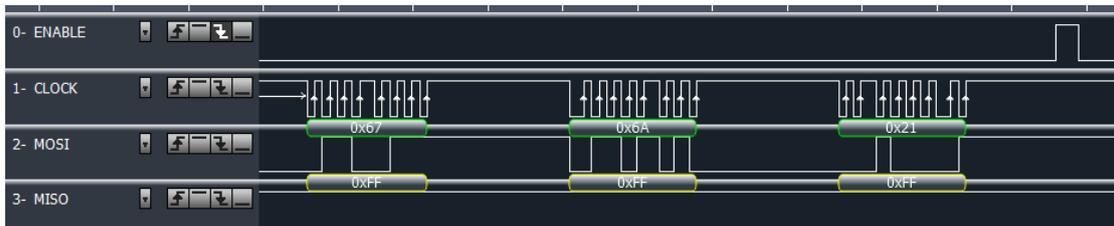
拉低片选，发送命令 0x02，表示写操作。发送 3 个字节的地址。因为选用的 flash 芯片大小为 2M，地址范围为 0x00_0000 – 0x1f_ffff。所以需要 24 位表示地址。因此要发送 3 个字节来表征写入的地址。



三个地址发送完毕后，就开始发送要写入的数据。

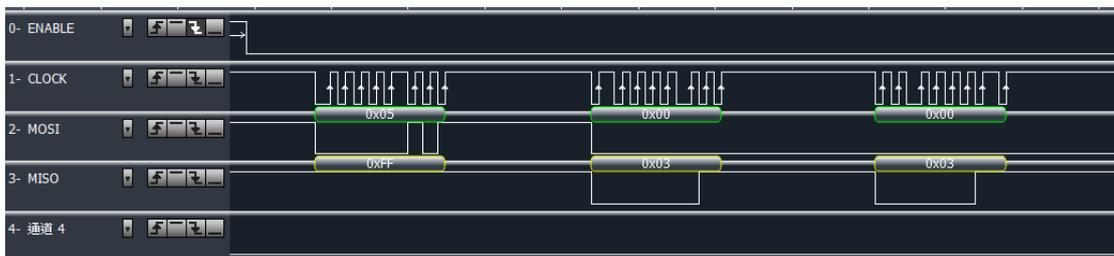


数据写完后，拉高片选即可。

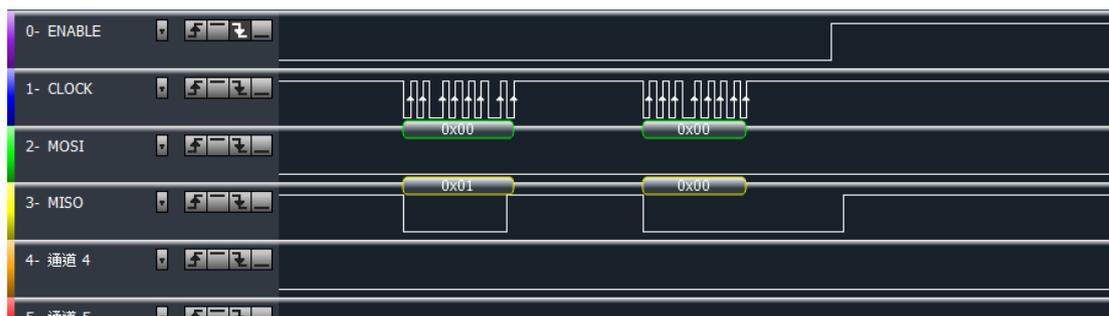


在写数据完成后，要读取 flash 芯片的状态寄存器，判断器件状态。

读取状态寄存器，所用命令为 0x05，第二个 SPI 周期即为读取的状态寄存器值，可以一直读取该状态寄存器的值。



当读到状态寄存器的值为 0x00 后，表示器件完成写操作。



五、扇区擦除

使用命令 0x20 对扇区进行擦除。



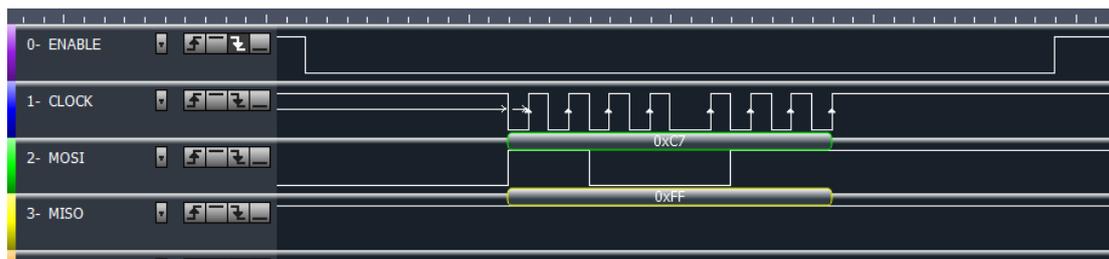
擦除是从写入扇区地址开始到扇区结束地址之间的内容擦除。即如果写入擦除的地址是 0x00_0010, 那么就会将 0x00_0010-0x00_0FFF (一个扇区大小是 4k) 之间的内存区域擦除为 0xff。

六、块擦除

和扇区擦除一样, 只是发的命令为 0xd8, 擦除的最大大小是 64KB。

七、芯片擦除

擦除整个芯片, 发送命令 0xc7。

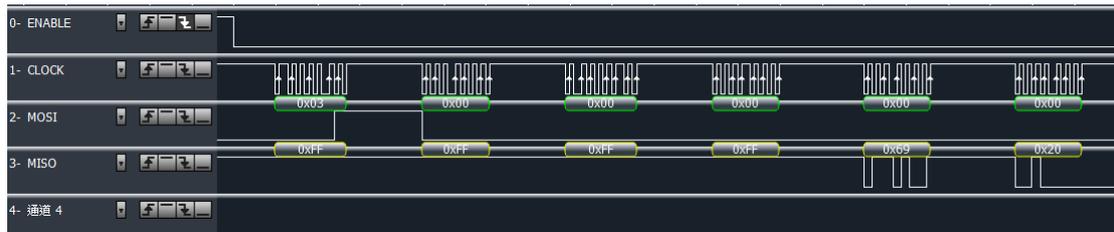


八、 读数据

FLASH 的读数据有三种方式:

普通读方式

先拉低片选信号 CS, 再发送命令 0X03, 发送读取数据的地址, 3 个字节。后面每个 SPI 周期, 就是读取的数据。



快速读方式 (fast read)

先拉低片选信号 CS, 再发送命令 0X0b, 发送读取数据的地址, 3 个字节。发送一个 dummy 的 SPI 周期, 后面每个 SPI 周期, 就是读取的数据。



快速读方式, 要等待 5 个 SPI 周期后, 才开始读取有效信息。而普通读模式下, 只需等待 4 个 SPI 周期后, 就可以读取有效信息了。

快速双通道读方式 (fast read dual output)

拉低片选 CS, 发送命令 0x3b, 发送读取数据的地址, 3 个字节。发送一个 dummy 的 SPI 周期, 后面每个 SPI 周期, 就是读取的数据。



从时序图。看出这种读方式下, MOSI 和 MISO 都参与了读数据的传输。所以在这种模式下, 一个 SPI 可以读取两个字节。

第一个字节

第二个字节

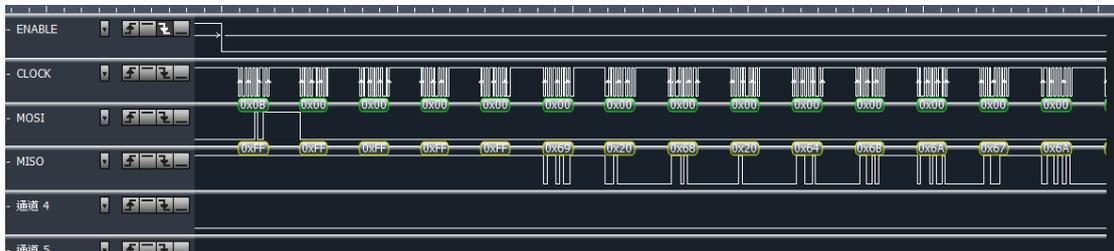
MISO (1D7, 1D5, 1D3, 1D2) (2D7, 2D5, 2D3, 2D2)
MOSI (1D6, 1D4, 1D2, 1D0) (2D6, 2D4, 2D2, 2D0)

下面通过波形图比较一下三种读取数据方式:

普通 read 读取的波形图。第一个有效数据在第 5 个 SPI 处, 值为 0x69, 第二个有效数据在第 6 个 SPI 处, 值为 0x20。

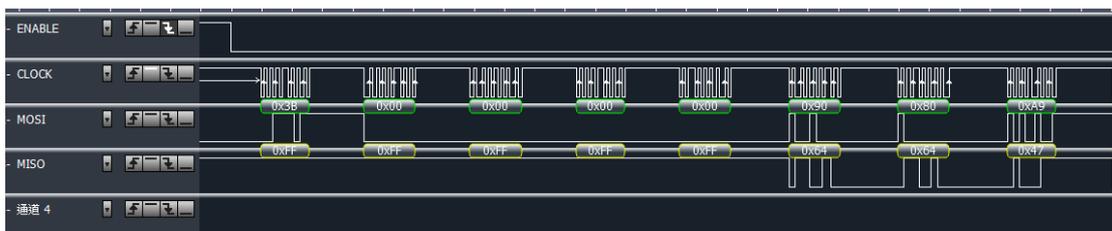


fast read 读取的波形图。第一个有效数据在第 6 个 SPI 处, 值为 0x69, 第二个有效数据在第 7 个 SPI 处, 值为 0x20。



fast read dual 读取的波形图。MOSI 和 MISO 同时参与数据的读取。第一个有效数据和第二个有效数据在第 6 个 SPI 处。

MISO 0 1 1 0 0 1 0 0 (1D7, 1D5, 1D3, 1D1) (2D7, 2D5, 2D3, 2D1)
MOSI 1 0 0 1 0 0 0 0 (1D6, 1D4, 1D2, 1D0) (2D6, 2D4, 2D2, 2D0)
组合得到第一个数据 01 10 10 01 0x69
组合得到第二个数据 00 10 00 00 0x20



利用逻辑分析仪抓取波形图, 来学习 SPI 驱动外部的 FLASH, 是可以很直观的看到数据传输的过程, 从而对 SPI 协议及外部的 FLASH 驱动有更深入的了解。