

目录

目录

| | | |
|-----|---------------------------------------|---|
| 1 | SPI 功能说明..... | 2 |
| 2 | FCE1353 和 LAN9252 的 SPI 差异及应用注意..... | 3 |
| 2.1 | FCE1353 和 LAN9252 的 SPI 快速读指令差异 | 3 |
| 2.2 | 使用示例..... | 4 |

Funcience confidential

1 SPI 功能说明

SPI 从模块提供低引脚数同步从接口，便于器件与主机系统之间的通信。凭借 SPI 从器件，可访问系统 CSR、内部 FIFO 和存储器。该器件支持一条和多条采用递增、递减和静态寻址的寄存器读写命令。SPI 模式支持单、双和四位通道，时钟速率最高达 80 MHz。

在 SPI 模式下，8 位指令在 SCS#变为有效后从输入时钟的第一个上升沿开始。该指令始终以串行方式输入 SI/SIO0。对于读指令和写指令，指令字节后跟两个地址字节。对于一些读指令，地址字节后跟一个或两个长度字节和空字节周期。取决于传输指令，传递长度字节以串行方式输入，或者以每个时钟 2 位或 4 位的方式输入。

芯片支持多种读写指令，常见的一般多使用快速读（FASTREAD）指令。

Funcience confidential

2 FCE1353 和 LAN9252 的 SPI 差异及应用注意

2.1 FCE1353 和 LAN9252 的 SPI 快速读指令差异

FCE1353 和 LAN9252 的差异主要表现在快速读（FASTREAD）指令。根据图 2-1 和图 2-2 可以看到 FCE1353 和 LAN9252 的发送的命令格式存在差异，主要体现在图 2-1 红框内的传输长度字节，而图 2-2 则没有此字节。

FCE1353 进行读操作指令时，首先将 SCS#置为有效，8 位 FASTREAD 指令（0Bh）输入到引脚后，其后跟 2 个地址字节，1 到 2 个传输长度字节以及 1 到 255 个空字节。

LAN9252 进行读操作指令时，首先将 SCS#置为有效，8 位 FASTREAD 指令（0Bh）输入到引脚后，其后跟 2 个地址字节以及 1 个空字节。

FCE1353 在 SPI 模式下单次寄存器快速读操作如下：

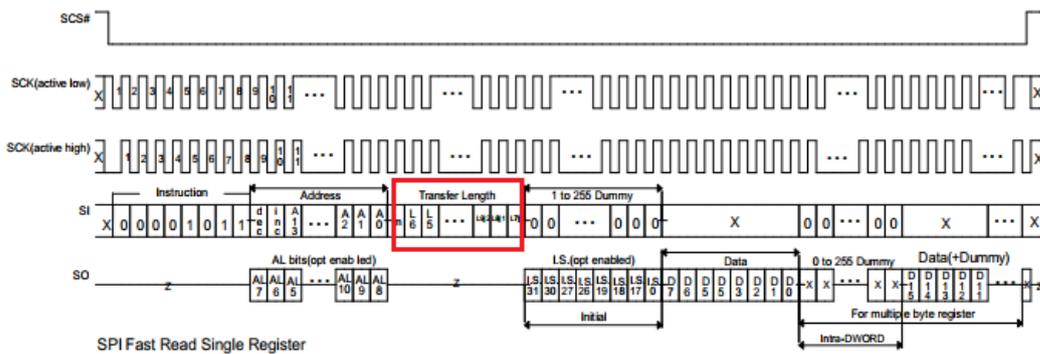


图 2-1 FCE1353 快速读操作

LAN9252 在 SPI 模式下单次寄存器快速读操作如下：

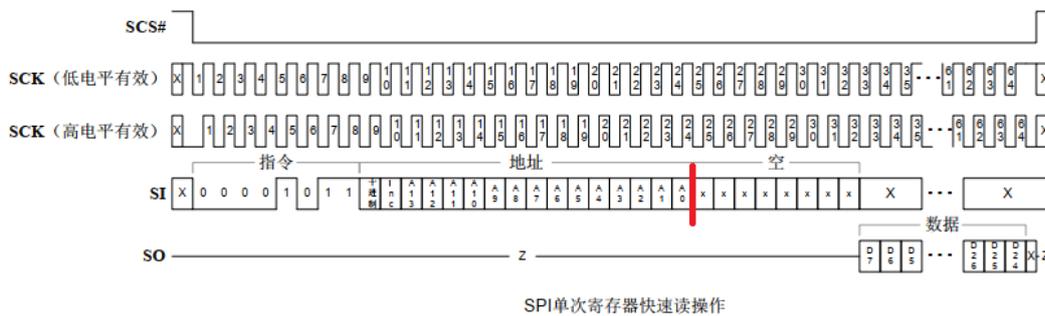


图 2-2 LAN9252 快速读操作

2.2 使用示例

在选择读写指令时，需要结合芯片 **datasheet** 来编写底层读写函数，两者主要差异为 FCE1353 相对于 LAN9252 多发送了一个传输长度字节，其余相同即可。

下图为两者底层快速读函数的使用示例，在 FCE1353 使用时需要多发送一个传输长度字节，在 LAN9252 中则直接注释即可，其余均相同。

FCE1353 底层读函数用例如下：

```
uint32_t SPIReadDWord (uint16_t Address)
{
    UINT32_VAL dwResult;
    UINT16_VAL wAddr;

    wAddr.Val = Address;
    //Assert CS line
    CSLOW();
    //Write Command
    SPIWriteByte(CMD_FAST_READ);
    //Write Address
    SPIWriteByte(wAddr.byte.HB);
    SPIWriteByte(wAddr.byte.LB);
    //Write Length
    SPIWriteByte(4);
    //Dummy Byte
    SPIWriteByte(CMD_FAST_READ_DUMMY); //CMD_FAST_READ_DUMMY
    // SPIReadByte();
    //Read Bytes
    dwResult.byte.LB = SPIReadByte(); //
    // printf("\r\n SPIReadByte() is 0x%x", dwResult.byte.LB);
    dwResult.byte.HB = SPIReadByte();
    // printf("\r\n SPIReadByte() is 0x%x", dwResult.byte.HB);
    dwResult.byte UB = SPIReadByte();
    // printf("\r\n SPIReadByte() is 0x%x", dwResult.byte.UB);
    dwResult.byte.MB = SPIReadByte();
    // printf("\r\n SPIReadByte() is 0x%x", dwResult.byte.MB);
    //De-Assert CS line
    CSHIGH();

    return dwResult.Val;
}
```

图 2-3 FCE1353 底层读函数

LAN9252 底层读函数用例如下:

```
uint32_t SPIReadDWord (uint16_t Address)
{
    UINT32_VAL dwResult;
    UINT16_VAL wAddr;

    wAddr.Val = Address;
    //Assert CS line
    CSLOW();
    //Write Command
    SPIWriteByte(CMD_FAST_READ);
    //Write Address
    SPIWriteByte(wAddr.byte.HB);
    SPIWriteByte(wAddr.byte.LB);
    //Write Length
    // SPIWriteByte(4);
    //Dummy Byte
    SPIWriteByte(CMD_FAST_READ_DUMMY); //CMD_FAST_READ_DUMMY
    // SPIReadByte();
    //Read Bytes
    dwResult.byte.LB = SPIReadByte();
    // printf("\r\n SPIReadByte() is 0x%x", dwResult.byte.LB);
    dwResult.byte.HB = SPIReadByte();
    // printf("\r\n SPIReadByte() is 0x%x", dwResult.byte.HB);
    dwResult.byte UB = SPIReadByte();
    // printf("\r\n SPIReadByte() is 0x%x", dwResult.byte UB);
    dwResult.byte.MB = SPIReadByte();
    // printf("\r\n SPIReadByte() is 0x%x", dwResult.byte.MB);
    //De-Assert CS line
    CSHIGH();

    return dwResult.Val;
}
```

图 2-4 LAN9252 底层读函数