

HiperFace DSL 编码器研发快速指导手册

部门：	产品管理部	类型：	指导说明
编写人：	Carlos Wang	邮箱：	Carlos.wang@sick.net.cn
版本：	V1.0	日期：	2017-12-2

HIPERFACE[®]
DSL

西克中国
官方微信



西克中国
微官网



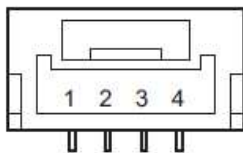
目 录

1. 电气接线.....	3
2. 电路设计	3
3. 硬件配置注意事项	3
4. 系统初始化.....	4
4.1. DSL-Master 启动	5
4.2. 激活 DSL 输出	5
4.3. DSL 同步	5
4.4. 写入 ES 值	5
4.5. 读取 EVENT_H 寄存器，清除所有报警位	6
4.6. 读取 Fast Position 和 Speed 值，进行运动控制.....	6
5. 运行中信号监控.....	6
5.1. Online Status_D 中的报警信息：	7
5.2. 通讯质量.....	7
5.3. Estimator_on 输出信号	7
5.4. Div_thr_Err 输出信息号	7
6. 其他常用功能.....	8
6.1. 设定预设值：	8
7. 故障处理.....	9
7.1. EVENT_H, EVENT_L 包含所有警报和故障的状态位。	9
8. SPI 通讯规则.....	10
8.1. SPI 通讯时序	12
8.2. 读取单独寄存器	12
8.3. 读取多个寄存器	13
8.4. 写入一个寄存器	14
8.5. 写入多个寄存器	14
8.6. 读/写组合	15
8.7. SPI 故障	15

1. 电气接线

按照以下接线图把编码器与驱动器进行连接：

电源 / 通讯针脚分配



独立电缆 = S, T

针脚	信号	说明
1	+U _s	7 ... 12 V 电源
2	DSL+	RS-485 DSL-Data
3	DSL-	RS-485 DSL-Data
4	GND	接地

推荐绞合电缆外径：4 mm +0/-0.3 mm
推荐接头：JST (GHR-04V-S)

整合至电机电缆中 = J, K

针脚	信号	说明
1		不连接
2	+U _s /DSL+	电源 / DSL-Data
3	GND/DSL-	接地 / DSL-Data
4		不连接

- 请注意电源为 DC7-12V，推荐 DC8V。
- 电源功率保证可以满足 3.6A 的瞬时电流。

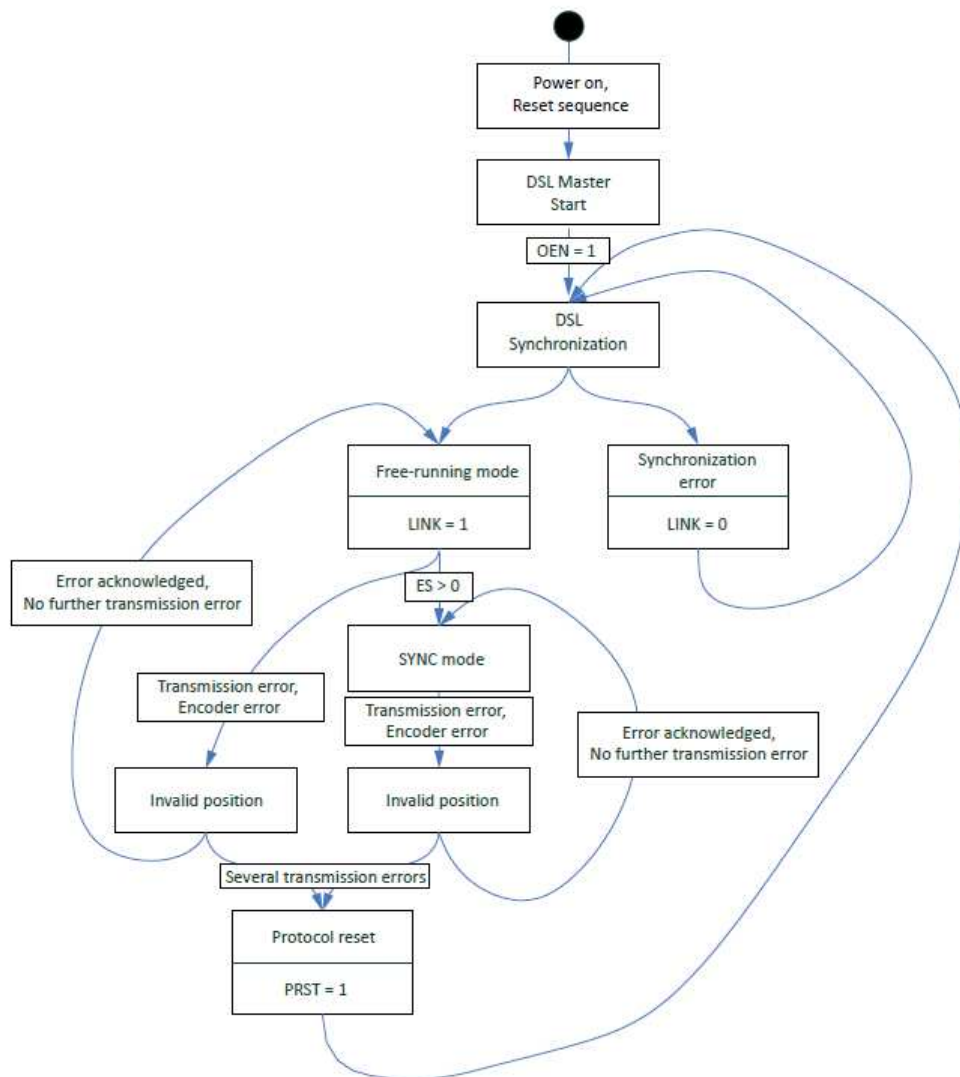
2. 电路设计

关于接口电路、FPGA IP-Core 输入输出以及电缆规格，请参考 Implementation 手册第 4 章。

3. 硬件配置注意事项

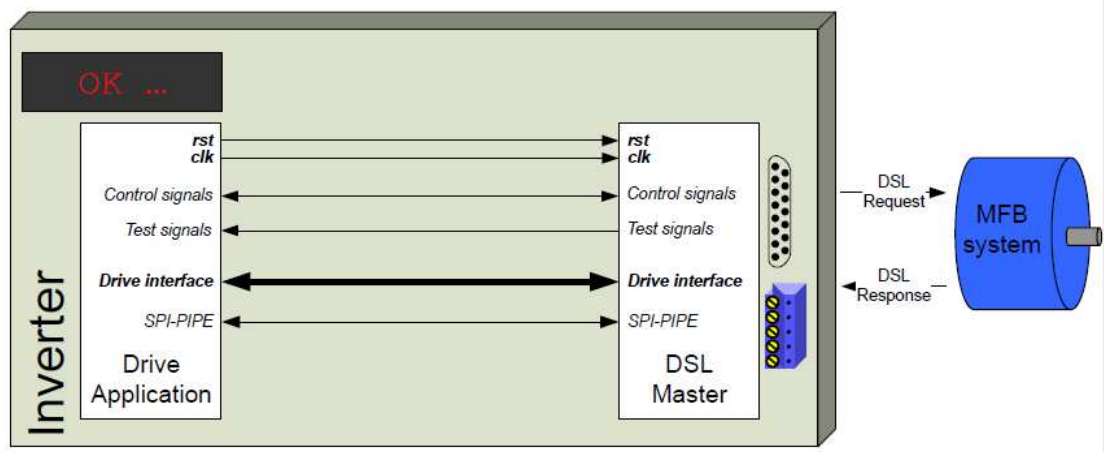
- 需提供 75MHz 的时钟频率，用于驱动 IP-Core 运行
- 推荐驱动器与 IP-Core 同步的频率为 8KHz 或 10KHz

4. 系统初始化



上图为初始化的流程，供参考。以下是详细的介绍：

首先，需要给 IP-Core 接入 75MHz 的时钟，保证其正常运行



时钟的规格如下：

Parameter	Value			Units	Remarks
	Minimum	Typical	Maximum		
System clock	74.9925	75.0000	75.0075	MHz	± 100 ppm

4.1. DSL-Master 启动

通电，系统通过 rst 引脚对 IP-Core 进行 reset，等待大约 500ms。

Signal name	Type	Function
rst*	Input	Master reset (High active)

4.2. 激活 DSL 输出

写寄存器 00h 的 OEN 值 1，在 SYS_CTRL Register 里：

00h	SYS_CTRL	PRST	MRST	FRST	LOOP	PRDY	SPPE	SPOL	OEN	0000 0000
-----	----------	------	------	------	------	------	------	------	-----	-----------

4.3. DSL 同步

读取寄存器 03h 的 LINK 值是否为 “1”，确认 DSL 连接正常：

03h	MASTER_QM	LINK	-	-	-	Quality monitoring	0--- 0000
-----	-----------	------	---	---	---	--------------------	-----------

4.4. 写入 ES 值

ES 值为驱动器每次读取数据时，编码器响应的频率，在寄存器 01h 内写入：

01h	SYNC_CTRL	ES								0000 0001
-----	-----------	----	--	--	--	--	--	--	--	-----------

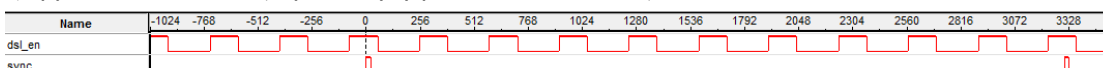
ES 值的计算方法如下：

根据驱动器的 SYNC 扫描频率计算：

Cycle time of the frequency inverter	12.1		1,950	μs	In SYNC mode
Packet cycle time	12.1		27	μs	In SYNC mode

假设驱动器扫描频率 8kHz，那么 Cycle time 是 125us，而编码器的帧频为 12.1-27us，那么最小 ES 值是 4.6，最大 ES 是 10.3，可以把 ES 值写成 10。

注：ES 值只可以写入，不可以读取，写完后，可以监控同步信号和 dsl_en 信号，确保 ES 值已经生效，如下图，ES 值被成功写为 10：



建议有条件的话，把 ES 值设置为 2 的 N 次方，比如 8 或 16，在一个同步请求后，能收到 1 个或 2 个完整的 safe position 值。

4.5. 读取 EVENT_H 寄存器，清除所有报警位

A. EVENT_H 的 SUM 为报警合集，如果 SUM=1，需要找出与之对应的所有报警寄存器，进行清零。寄存器对应关系详见第 7 节，或英文手册。

04h	EVENT_H	INT	SUM	SCE	-	POS	VPOS	DTE	PRST	000- 0000
-----	---------	-----	-----	-----	---	-----	------	-----	------	-----------

B. 读取 POS，确保为 0：说明当前没有 Fast Position 报警；如果是 1，进行清除操作；如果无法清除，则需要查找原因。

C. 读取 PRST 的值，确保为 0：说明当前 IP-Core 正常运行；如果是 1，进行清除操作；如果无法清除，则需要查找 reset 原因。

4.6. 读取 Fast Position 和 Speed 值，进行运动控制

10h	POS4	Fast position, byte 4	0000 0000
11h	POS3	Fast position, byte 3	0000 0000
12h	POS2	Fast position, byte 2	0000 0000
13h	POS1	Fast position, byte 1	0000 0000
14h	POS0	Fast position, byte 0	0000 0000
15h	VEL2	Speed, byte 2	0000 0000
16h	VEL1	Speed, byte 1	0000 0000
17h	VEL0	Speed, byte 0	0000 0000

驱动器利用并口或 SPI 方式读取需要的值：

位置值是寄存器 10h 到 14h 的组合，从高到低位的排序如下所示：

Byte4	Byte3	Byte2	Byte1	Byte0
00	6B	10	0A	06

速度值是寄存器 15h 和 17h 的组合，从高到低位的排序如下所示：

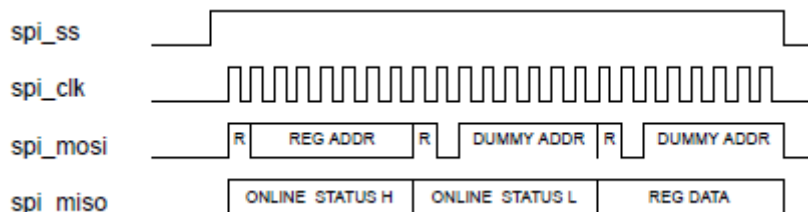
Byte2	Byte1	Byte0
1A	0d	08

5. 运行中信号监控

运行中重要的监控信息有以下几种：

5.1. Online Status_D 中的报警信息：

驱动器每次同步读取位置时，Online status 内容会跟着位置信息一起发送。



位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT	SUM	SCE	FIX1	POS	VPOS	DTE	PRST	POSTX		MIN	ANS	FIX0	QMLW	FREL	FRES
正常	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1

其中，POS 表示 fast position 是否正常。

5.2. 通讯质量

可监控寄存器 03h 的 LINK 为，或者监控 IP 核的 link 输出：

03h	MASTER_QM	LINK	-	-	-	Quality monitoring	0--- 0000
-----	-----------	------	---	---	---	--------------------	-----------

IPcore 的 OUTPUT。

link	Output	Connection indication
------	--------	-----------------------

二者意思一样，如果通讯中断，二者都会变为 1

5.3. Estimator_on 输出信号

这个信号有输出，表明当前的 Fast Position 数值无效，当前驱动器得到的数值是预估值 Estimated value。

监控 IP-Core 的这个输出点，

estimator_on	Output	Position Estimator activated
--------------	--------	------------------------------

5.4. Div_thr_Err 输出信息号

当前位置的偏差值超过了寄存器 3A,3B 的设定值，IPcore 会激活这个输出：

dev_thr_err	Output	Estimator deviation threshold reached
-------------	--------	---------------------------------------

6. 其他常用功能

Resource 集合了编码器所有的功能，驱动器可以用长信息来实现和 resource 通讯，读取或修改相应的参数。

RDB(Resource database)里的每个 Resource 都有自己的 Index，叫做 RID。

6.1. 设定预设值：

设置预设值需要用到长信息写入方法，具体操作如下：

把当前位置值设置为预设值，比如 00h 00h 00h 00h 00h。

a. 写入之前，需要把 Access Level 改为 2，即 Maintenance Level。

详见手册 8.8.3：

Access level	Standard access key	Usage
0	No access key necessary	Publicly accessible system functions
1	31 31 31 31h	Protected system functions - "operator" level
2	32 32 32 32h	Protected system functions - "maintenance" level

b. 检查寄存器 05h，FREL 为 1：

05h	EVENT_L	-	-	MIN	ANS	-	QMLW	FREL	FRES
-----	---------	---	---	-----	-----	---	------	------	------

Bit 1 FREL: Channel status for "long message".

1 = The channel for the "long message" is free.

0 = The channel for the "long message" is in use.

c. 写入预设值到长数据寄存器：

数据写操作寄存器起始地址是 20h。

寄存器 20h~24h，写入的数据为 5 bytes 据 00h。

寄存器 25h~27h 写入数据也是 00h；

d. 指定写入规则

28h	PC_ADD_H	LID	LRW	LOFF	LIND	LLEN	LADD9:8
29h	PC_ADD_L	LADD7:0					

寄存器 28h 写入 1Dh；寄存器 29h 写入 01h；

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1

即：写入模式；无 offset；间接模式；数据长度为 8bytes；地址为 101h。

e. 寄存器 2A、2Bh 为 00h，表示无地址 offset；

f. 最后写入寄存器 2Ch 数据 01h，启动发送 long message。

2Ch	PC_CTRL	-	-	-	-	-	-	-	LSTA
-----	---------	---	---	---	---	---	---	---	------

g. 写完之后，一直读取 FREL，直到 FREL=1。

05h	EVENT_L	-	-	MIN	ANS	-	QMLW	FREL	FRES
-----	---------	---	---	-----	-----	---	------	------	------

h. 当 FREL=1 时，读取当前 Fast Position 值，可以看到当前的位置已经被成功设置为 0。

10h	POS4	Fast position, byte 4	0000 0000
11h	POS3	Fast position, byte 3	0000 0000
12h	POS2	Fast position, byte 2	0000 0000
13h	POS1	Fast position, byte 1	0000 0000
14h	POS0	Fast position, byte 0	0000 0000
15h	VEL2	Speed, byte 2	0000 0000
16h	VEL1	Speed, byte 1	0000 0000
17h	VEL0	Speed, byte 0	0000 0000

Byte4	Byte3	Byte2	Byte1	Byte0
00	00	00	00	00

7. 故障处理

7.1. EVENT_H, EVENT_L 包含所有警报和故障的状态位。

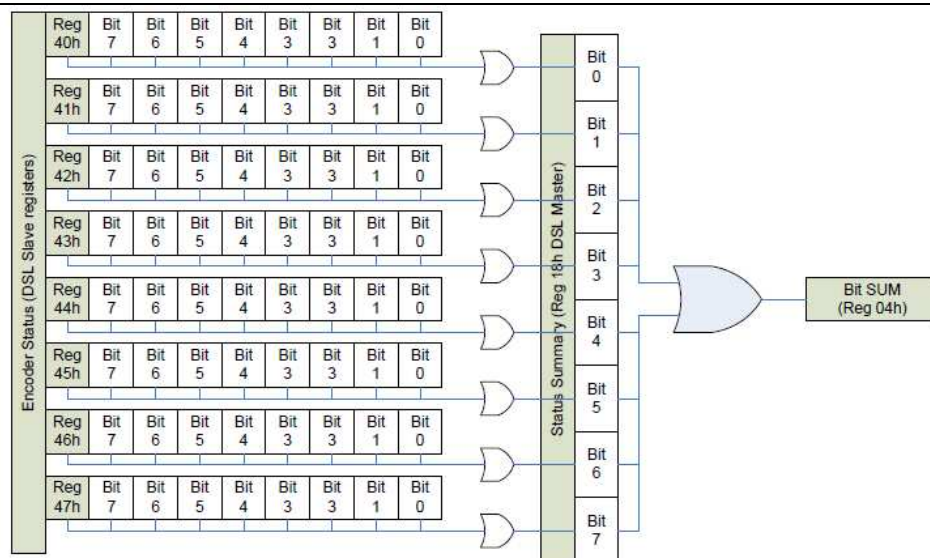
04h	EVENT_H	INT	SUM	SCE	-	POS	VPOS	DTE	PRST	000- 0000
05h	EVENT_L	-	-	MIN	ANS	-	QMLW	FREL	FRES	--00 -000

EVENT_H 的 SUM 的值为 1 的时候，说明 DSL 编码器有报错信息。此时，你需要检查寄存器 DSL Slave(40h...47h),是不是有 bit 为 1 了。

读取之后，你可以把相应的 bit 写入 0。因为寄存器的报警位是无法自己消除的，所以需要驱动器进行消除。

初始化结束后，你需要清除 PRST-bit，这个总是用于表示一个 Protocol Reset 被执行完毕。(EVENT_H，04h，bit0)

如果故障无法消除，那么请参考详细手册 7.6 节，明确故障内容，进行排查。

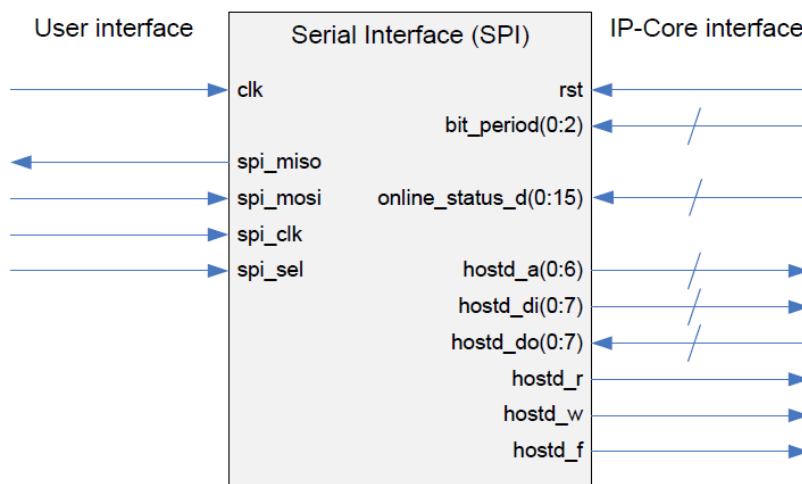


Encoder status	SAFE_SUM bit (DSL Master 36h)
ENC_ST0 (40h)	SUM0
ENC_ST1 (41h)	SUM1
ENC_ST2 (42h)	SUM2
ENC_ST3 (43h)	SUM3
ENC_ST4 (44h)	SUM4
ENC_ST5 (45h)	SUM5
ENC_ST6 (46h)	SUM6
ENC_ST7 (47h)	SUM7

8. SPI 通讯规则

以下为驱动器与 IPCORE 进行 SPI 串口通讯的规则，仅供参考。详细信息请翻阅操作手册。

接口如下：



时钟规格如下：

Parameter	Value			Units	Comments
	Mini- mum	Typical	Maxi- mum		
Clock spi_clk			10	MHz	
Clock phase (PHA)	PHA = 1, scanning during falling clock edge				
Clock polarity (POL)	POL = 0, base value of the clock is 0				
Data endianness	MSB is clocked out first				

接口的定义如下图：

Pin name	Model name	Function	Note
Host interface			
clk	Input	Clock input	
spi_miso	output	SPI data output	
spi_mosi	Input	SPI data input	
spi_clk	Input	SPI clock	
spi_sel	Input	SPI selection	
IP Core interface			
rst	Input	Internal reset	Connect to IUO(1) of the IP Core
bit_period(2:0)	Input	Internal state machine	Connect to IUO(4:2) of the IP Core
online_status_d(15:0)	Input	Internal status IP Core	
hostd_a(6:0)	output	Register block address bus	
hostd_di(7:0)	output	Data bus interface to core	
hostd_do(7:0)	Input	Data bus core to interface	
hostd_r	output	Read access requirement	
hostd_w	output	Write access requirement	
hostd_f	output	Freeze register selection	

SPI 对寄存器有以下交互操作：

- 读取每个寄存器
- 自由读取多个寄存器
- 写入每个寄存器
- 写入多个寄存器(自动增加)
- 读/写组合

注：在进行一个读/写组合时，写必须是最后一步。

在每一次交互的开始，DSL master 通过 spi_miso 传输 online-status，以 2 个字节的形式。

8.1. SPI 通讯时序

时序图如下：

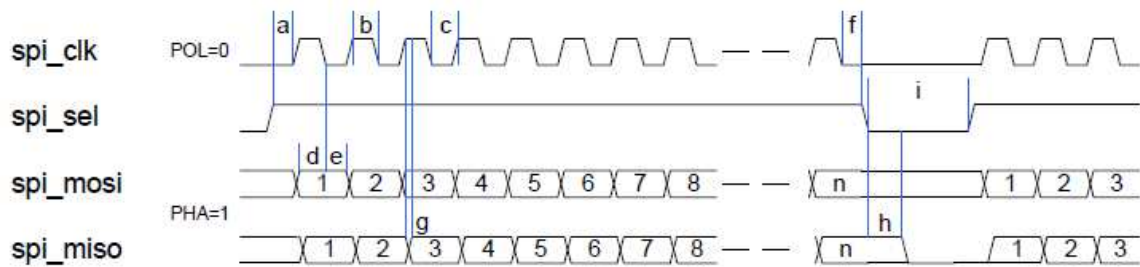


Figure 37: Time control of the SPI

时序要求如下

Diagram position	Description	Minimum	Maximum	Units
a	Setting spi_sel before spi_clk	25		ns
b	Time for spi_clk high	50		ns
c	Time for spi_clk low	50		ns
d	Setting spi_mosi before spi_clk low	10		ns
e	Keep spi_mosi at spi_clk low	25		ns
f	Keep spi_sel at spi_clk low	260		ns
g	Delay spi_miso at spi_clk high	25	60	ns
h	Delay spi_miso at spi_sel low	25	60	ns
i	Time for spi_sel low	50		ns

8.2. 读取单独寄存器

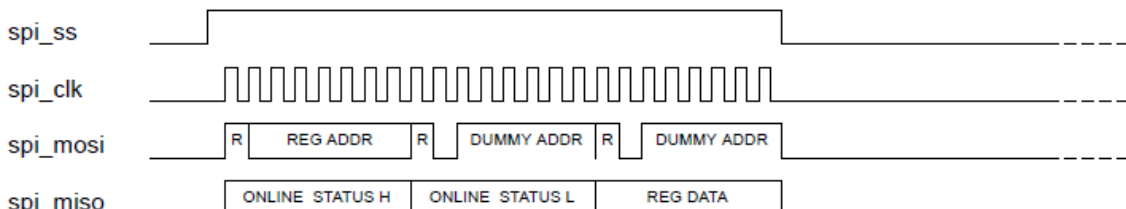
读写的标志位是指寄存器地址的最高位：

读的时候在寄存器地址最高位变为 1，也就是寄存器地址加 80h；写的时候保持寄存器地址最高位为 0。

因为收到的 Status H, Status L，REG DATA 总共 3 个字节，所以读取时要用 DUMMY 虚拟地址填满，防止收到其他信息。

时序如下：

Symbol	Meaning
R	Access bit: Read ("1")
REG ADDR	Register address (00h to 7Fh)
DUMMY ADDR	Register address for the dummy read process (3Fh)
ONLINE STATUS H	Online-status - High byte
ONLINE STATUS L	Online-status - Low byte
REG DATA	Register content



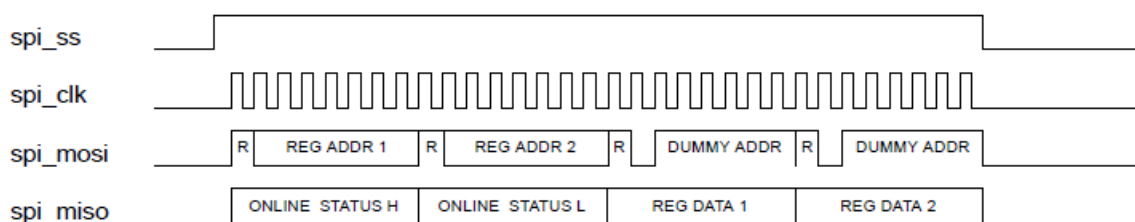
举例：读取寄存器 00h 的值：

Spi_Mosi: 80 BF BF

8.3. 读取多个寄存器

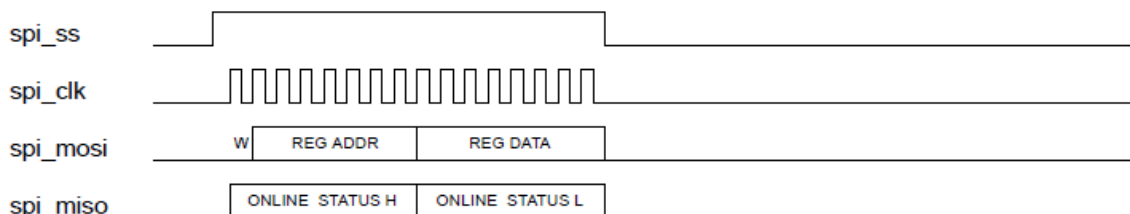
时序如下：

Symbol	Meaning
R	Access bit: Read ("1")
REG ADDR x	Register address (00h to 7Fh), no. x
DUMMY ADDR	Register address for the dummy read process (3Fh)
ONLINE STATUS H	Online-status - High byte
ONLINE STATUS L	Online-status - Low byte
REG DATA x	Content of register x



8.4. 写入一个寄存器

Symbol	Meaning
W	Access bit: Write ("0")
REG ADDR	Register address (00h to 7Fh)
REG DATA	Register content
ONLINE STATUS H	Online-status - High byte
ONLINE STATUS L	Online-status - Low byte

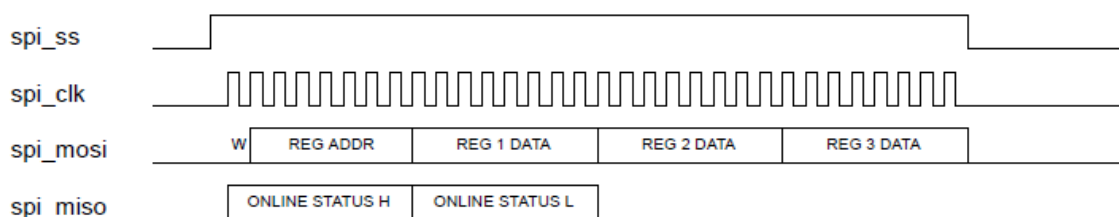


举例：写入寄存器 00h 值 01h

Spi_mosi: 00 00 01h.

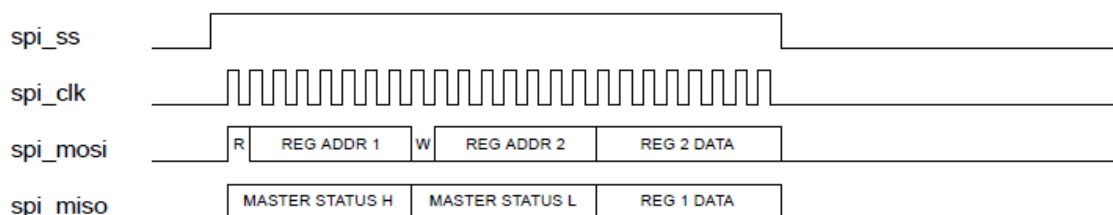
8.5. 写入多个寄存器

Symbol	Meaning
W	Access bit: Write ("0")
REG ADDR	Address of the starting register (00h to 7Fh)
REG DATA x	Content of register x, beginning at REG ADDR
ONLINE STATUS H	Online-status - High byte
ONLINE STATUS L	Online-status - Low byte



8.6. 读/写组合

Symbol	Meaning
R	Access bit: Read ("1")
W	Access bit: Write ("0")
REG ADDR 1	Register address for read access (00h to 7Fh)
REG ADDR 2	Register start address for write access (00h to 7Fh)
REG DATA 2	Register content for write access
MASTER STATUS H	Online-status - High byte
MASTER STATUS L	Online-status - Low byte
REG DATA 1	Register content for read access



8.7. SPI 故障

Fault condition of the SPI	Fault indication
Incorrect number of CLK impulses	<code>spi_miso</code> at high level
Write command without data	<code>spi_miso</code> at high level