

精解 SPI 的 CPHA 时钟相位 与 CPLK 时钟极性

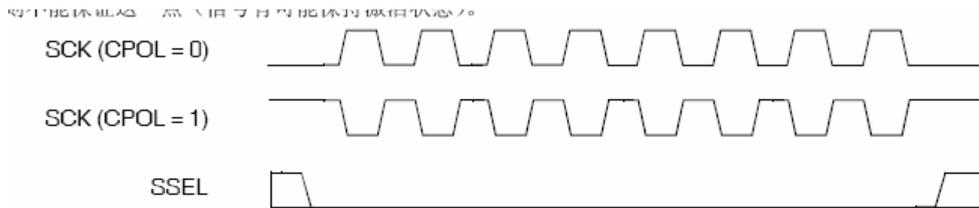
Netjob @ 2009-08-01

使用 SPI 模块都要设 CPHA 相位 与 CPLK 时钟有效极性，这个确实是烦到好多人的。其实这个设置也是比较简单的。

先来讲讲 SPI 的 CPLK 时钟有效极性

CPLK 时钟有效极性，有效当然是相对 无效 或是空闲而言的。

要传输 8 位数据，需要 8 个时钟脉冲。那么这 8 个脉冲之前和之后的时钟状态，可以认为是时钟空闲状态 或 无效状态。因为此事 SCK 没变化，保持在某个状态。



从上图中可以看出，(CPOL=0)的 SCK 波形，它有（传输）8 个脉冲，而在脉冲传输前和完成后都保持在【低电平状态】。此时的状态就是时钟的空闲状态或无效状态，因为此时没有脉冲，也就不会有数据传输。

同理得出，(CPOL=) 1 的图，时钟的空闲状态或无效状态时 SCK 是保持【高电平的】。

再来讲解何为 SPI CPHA 相位。

简单的讲，就是数据线上，MCU 或 FLASH 外设对数据的采样时刻。

例如 ATMEL ARM7的CPHA 时钟相位定义如下：

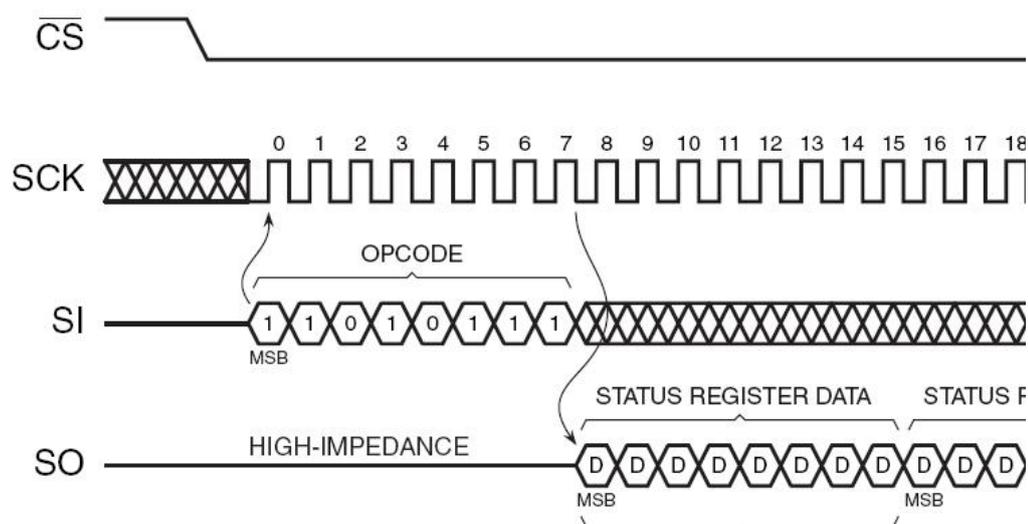
0= 数据在SPCK起始边沿改变，在SPCK 下一个边沿捕获。

1 = 数据在SPCK起始边沿捕获，在SPCK 下一个边沿改变。

我们知道，数据线上可以分两种时刻：【采样-保持时刻】和【变化更新 时刻】

要对数据进行采样，当然就要求数据要保持 稳定，不要发生变化。是吧？这个大家都明白。

atus Register Read (Opcode D7H)



我们看上面的图，发现 数据 SI 是对应 SCK 的第一个时钟沿，再仔细看，数据是在 SCK 的第一个时钟边沿保持稳定【数据被采样捕获】，在下一个边沿改变【SCK 的下降沿数据改变】

因此我们得出结论：AT45DB642DB 系列 FLASH 是【数据在第一个时钟沿被采样捕获】或【数据在 SPCK 起始边沿捕获，在 SPCK 下一个边沿改变】

我们再看看 CC2500 的时序图：

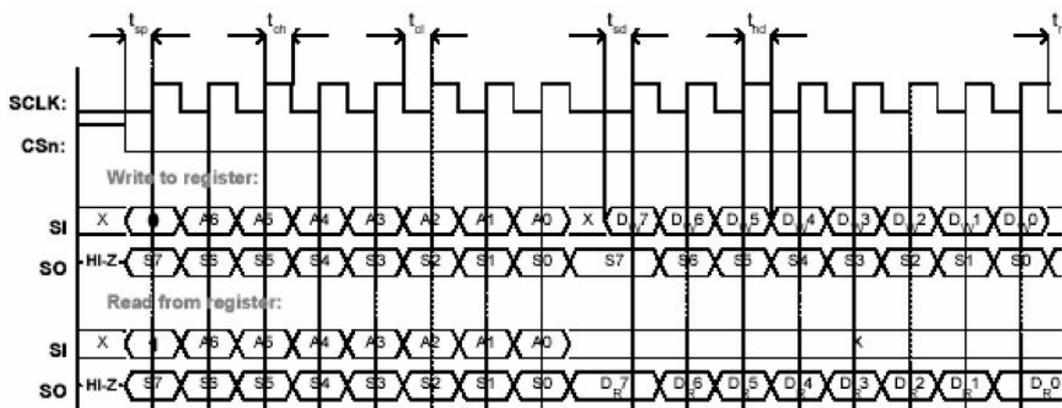


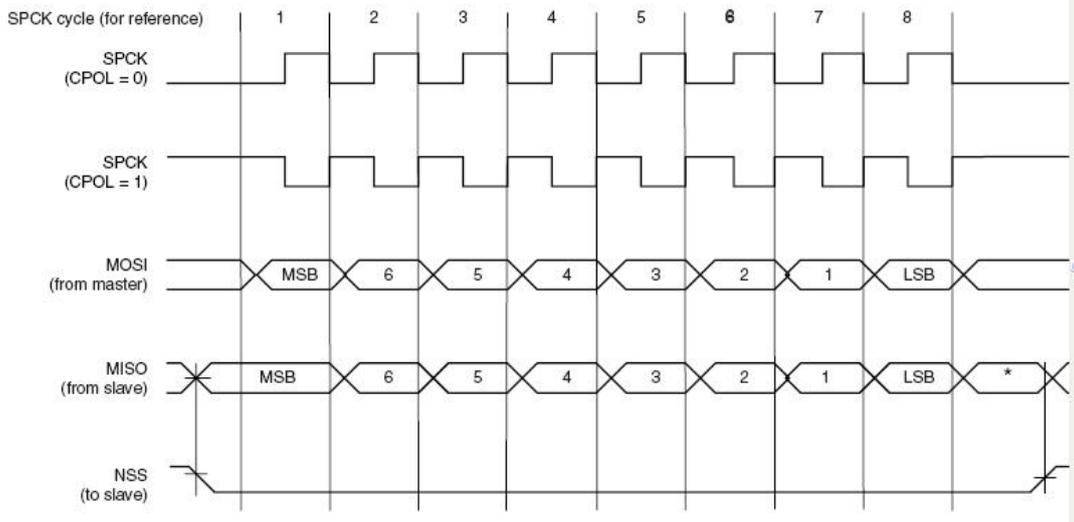
图 6 配置寄存器写和读操作

看上面图，明显看出 SI 和 SO 都是在 SCLK 的第一个时钟边沿【上升沿】数据被采样捕获。在下一个时钟边沿结束并数据发生变化。

而且可以看出，SCLK 在脉冲开始前是低电平的空闲状态或叫无效状态。

下面看图说明：

Figure 83. SPI 传输格式 (NCPHA = 1, 每次传输 8 位)



看上面图，可以看出，

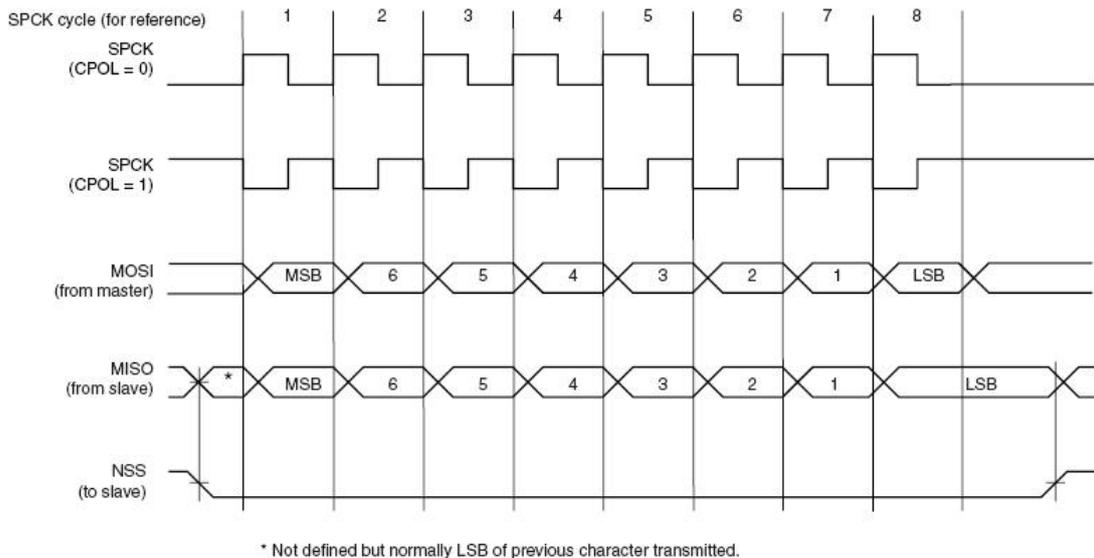
首先(CPOL=0)的 SPCK 图，它的时钟开始是低电平的(空闲时保持低电平)，结束后也是低电平的。由于 PHA=1,就是说 MOSI 和 MISO 的数据保持和被采样捕获是在 SPCK 的第一个时钟边沿（起始边沿）数据被捕获。第二个时钟边沿改变（数据锁存）。

当 CPOL=0 的是上升沿数据捕获，下降沿数据锁存。（开始的 SPCK 是低电平空闲）

当 CPOL=1 的是下降沿数据捕获，上升沿数据锁存。（开始的 SPCK 是高电平空闲）

再看图：

Figure 84. SPI 传输格式 (NCPHA = 0, 每次传输 8 位)



看上面图，可以看出，

首先(CPOL=0)的 SPCK 图，它的时钟开始是低电平的(空闲时保持低电平)，结束后也是低电

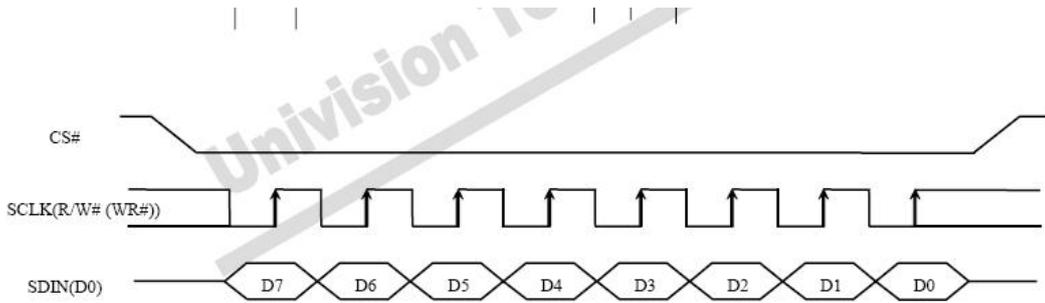
平的。

由于 $PHA=0$ ，就是说 MOSI 和 MISO 的在 SPCK 的第一个时钟边沿数据改变锁存（起始边沿）。第二个时钟边沿数据被捕获。

当 $CPOL=0$ 的 是上升沿数据锁存，下降沿数据捕获。（开始的 SPCK 是低电平空闲）

当 $CPOL=1$ 的 是下降沿数据锁存，上升沿数据捕获。（开始的 SPCK 是高电平空闲）

下面是晶门科技的 SS1D1351 SPI 接口方式的时序图：



这款控制器使用在 TFT LCD ,OLED 上面。 它的 SPI 方式：

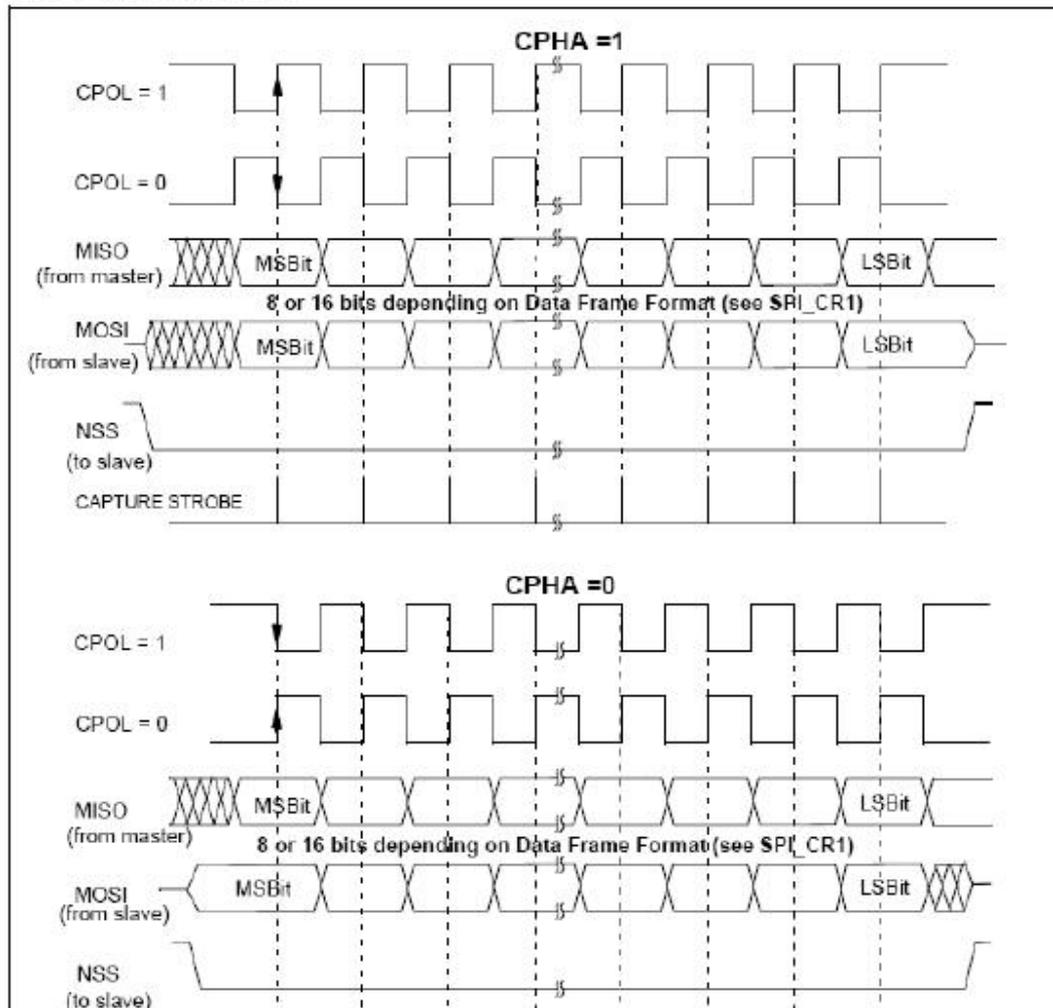
SCLK 开始的时刻可以是高或低电平。SCLK 的粗实线是上升沿箭头，说明是数据被捕获。下降沿数据改变（锁存）。由于 SCLK 的空闲无效状态可以是低电平，也可以是高电平。

因此：

如果 SCLK 的 ($CPOL=0$) 的空闲无效是低电平，那么数据是在第一个时钟沿（上升）被捕获。在下一个边沿（下降）数据改变（锁存）。

如果 SCLK 的 ($CPOL=1$) 的空闲无效是高电平，那么数据是在第一个时钟沿（下降）数据改变（锁存）。在下一个边沿（上升）数据被采样捕获。

图164 数据时钟时序图



看上面的图 STM32 的 SPI 这个图，

【CPOL=1】时钟脉冲的【开始】，【结束】都是无效状态、空闲的，CLK 是高电平的。ATMEL 说的 时钟脉冲在空闲时的电平是高电平的 则 CPOL=1;

STM32 说的“如果 CPOL 被‘复位’，SCK 引脚在空闲状态保持高电平;”

【CPOL=0】时钟脉冲的【开始】，【结束】都是无效状态、空闲的，CLK 是低电平的。ATMEL 说的”时钟脉冲在空闲时的电平是低电平的 则 CPOL=0;“

STM32 说的“如果 CPOL 被‘置位’，SCK 引脚在空闲状态保持低电平;”

【CPHA=1】表示第二个边沿数据被采样捕获，第一个边沿被锁存【数据改变时刻】

【CPHA=0】表示第一个边沿数据被采样捕获，第二个边沿被锁存【数据改变时刻】

STM32 的初始化代码:

```
SPI_InitStructure.SPI_Direction = SPI_Direction_1Line_Tx;
```

```
SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
```

```
SPI_InitStructure.SPI_DataSize = SPI_DataSize_8b;
```

//时钟脉冲的【开始】，【结束】都是高电平： 无效状态、空闲的

```
SPI_InitStructure.SPI_CPOL = SPI_CPOL_High;
// 数据在第二个边沿没采样捕获，第一个边沿被锁存【改变时刻】
SPI_InitStructure.SPI_CPHA = SPI_CPHA_2Edge;
```

```
SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_4;
SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB;
SPI_InitStructure.SPI_CRCPolynomial = 0;
SPI_Init(SPI1, &SPI_InitStructure);
```

最后要注意，NXP 的 ARM 对 SCK 的时钟有效极性说明的设置方法是与 ATMEL 的相反的。NXP 的说明是这样的：

SPI 控制寄存器 (S0SPCR - 0xE0020000)

SPCR 寄存器根据每个配置位的设定来控制 SPI 的操作。

表 110 SPI 控制寄存器 (S0SPCR - 0xE0020000)

SPCR	功能	描述	复位值
2:0	保留	保留，用户软件不要向其写入 1。从保留位读出的值未被定义。	NA
3	CPHA	时钟相位控制决定 SPI 传输时数据和时钟的关系并控制从机传输的起始和结束。当该位为 1 时，数据在 SCK 的第二个时钟沿采样。当 SSEL 信号激活时，传输从第一个时钟沿开始并在最后一个采样时钟沿结束。当该位为 0 时，数据在 SCK 的第一个时钟沿采样。传输从 SSEL 信号激活时开始，并在 SSEL 信号无效时结束。	0
4	CPOL	时钟极性控制。当该位为 1 时，SCK 为低有效。为 0 时，SCK 为高有效。	0
5	MSTR	主模式选择。为 1 时，SPI 处于主模式。为 0 时，SPI 处于从模式。	0
6	LSBF	LSBF 用来控制传输的每个字节的移动方向。为 1 时，SPI 数据传输 LSB (位 0) 在先。为 0 时，SPI 数据传输 MSB (位 7) 在先。	0
7	SPIE	SPI 中断使能。为 1 时，每次 SPIF 或 MODF 置位时都会产生硬件中断。为 0 时，SPI 中断被禁止。	0

它们的意义是一样，但 NXP 的是把 CPOL 设为 1，而 ATMEL 的是把 CPOL 设为 0。

我个人比较喜欢 ATMEL 说法。由于本人水平有限，不妥的地方恳请大家多多指教！

谢谢大家花时间来阅读此文！