

状态寄存器值

000: IDLE	001: RxACK
010: Rx	011: RxHOLD
100: RxBREAK	101: 保留状态
110: RxRST	111: RxERR

2. RX_CONFIG0_REG

RX_CONFIG0_REG 用于设置 Rx 控制器的配置

表 28-14 RX_CONFIG0_REG 寄存器描述

寄存器	地址	读/写	描述	复位值
RX_CONFIG0_REG	0x7E00_7004	读/写	MIPI HIS Rx 控制器配置寄存器	0x0FFFFFF02

位	名称	描述	读/写	复位值
[31:30]	Reserved	保留位	读	0x0
[29:28]	DREQ_thres_val	DMA 请求阈值 当 FIFO 的有效数据为下列情况时，DMA 请求信号被激活 0x00: 满 0x01: 多于四个字 0x10: 多于 8 个字 0x11: 多于 16 个字	读/写	0x00
[27:16]	Rx_State time	Rx 状态定时器设置值	读/写	0xFFF
[15:8]	RxACK time	RxACK 状态定时器设置值	读/写	0xFF
[7]	Reserved	保留位	读/写	0x0
[6]	RxACK time_en	RxACK 状态定时器使能 0: 禁止 1: 使能	读/写	0x0
[5]	Break_clr	RxBREAK 状态清除位使能 0: 禁止 1: 使能	读/写	0x0
[4]	Err_clr	清除产生的错误	读/写	0x0

		0: 保留 1: 清除		
[3:2]	Width of CHID	通道 ID 的宽度	读/写	0x0
[1]	Burst_mode	固定通道 ID 模式 0: 突发通道 ID 模式 1: 单通道 ID 模式	读/写	0x1
[0]	Frame_mode	帧 模式 0: 流模式 1: 帧 模式	读/写	0x0

3. RX_CONFIG1_REG

RX_CONFIG1_REG 用于设置 Rx FIFO 的配置

表 28-15 RX_CONFIG1_REG 寄存器描述

寄存器	地址	读/写	描述	复位值
RX_CONFIG1_REG	0x7E00_7008	读/写	MIPI HIS Rx 控制器配置寄存器	0x00FFFFFF

位	名称	描述	读/写	复位值
[31]	RxFIFO_clr	打破帧接收定时器设置值	读/写	0x0
[30:28]	Reserved	保留位	读	0x0
[27]	RxFIFO_timer_en	RxFIFO 定时器使能	读/写	0x0
[26:24]	Reserved	保留位	读	0x0
[23:0]	RxFIFO_time	RxFIFO 定时器设置值	读/写	0xFFFFFFFF

4. RX_INTSRC_REG

INTSRC_REG 是中断源悬挂寄存器

表 28-16 RX_INTSRC_REG 寄存器描述

寄存器	地址	读/写	描述	复位值
RX_INTSRC_REG	0x7E00_700C	读/写	MIPI HIS Rx 控制器中断源寄存器	0x00000000

位	名称	描述	读/写	复位值
[31:8]	Reserved	保留位	读	0x0000000
[7]	Break_done	帧模式下接收的打破帧（设置 1 进行清除）	读/写	0x0
[6]	Added_clock	增加的时钟输入（设置 1 进行清除）	读/写	0x0
[5]	Missed_clock	丢失的始终输入中断（设置 1 进行清除）	读/写	0x0
[4]	RxACK_timeout	RxACK 状态超时中断（设置 1 进行清除）	读/写	0x0
[3]	Brframe_err	接收的数据不是打破帧（设置 1 进行清除）	读/写	0x0
[2]	RxDONE	数据接收完成（设置 1 进行清除）	读/写	0x0
[1]	RxFIFO_timeout	RxFIFO 超时但是 RxFIFO 不为空（设置 1 进行清除）	读/写	0x0
[0]	RxFIFO_full	RxFIFO 满中断（设置 1 进行清除）	读/写	0x0

5. RX_INTMSK_REG

RX_INTMSK_REG 是中断屏蔽和 DMA 请求使能寄存器

表 28-17 RX_INTMSK_REG 寄存器描述

寄存器	地址	读/写	描述	复位值
RX_INTMSK_REG	0x7E00_7010	读/写	MIPI HIS Rx 控制器中断屏蔽寄存器	0x800001FF

位	名称	描述	读/写	复位值
[31]	DMA_req_en	DMA 请求信号使能 0: 使能 1: 禁止	读/写	0x1
[30:9]	Reserved	保留位	读	0x0000000

[8]	Wakeup_enn	MIPI 唤醒使能 0: 使能 1: 禁止	读/写	0x1
[7]	Break_done_mak	打破帧完成中断屏蔽 0: 不屏蔽 1: 屏蔽	读/写	0x1
[6]	Add_clock_mak	增加的时钟输入中断屏蔽 0: 不屏蔽 1: 屏蔽	读/写	0x1
[5]	Missed_clock_mak	丢失的时钟中断屏蔽 0: 不屏蔽 1: 屏蔽	读/写	0x1
[4]	RxACK_timeout_mak	RxACK 状态超时中断屏蔽 0: 不屏蔽 1: 屏蔽	读/写	0x1
[3]	Brframe_err_mak	打破帧错误中断屏蔽 0: 不屏蔽 1: 屏蔽	读/写	0x1
[2]	RxDONE_mak	RxDONE 中断屏蔽 0: 不屏蔽 1: 屏蔽	读/写	0x1
[1]	RxFIFO_timeout_mak	RxFIFO 超时中断屏蔽 0: 不屏蔽 1: 屏蔽	读/写	0x1
[0]	RxFIFO_full_mak	RxFIFO 满中断屏蔽 0: 不屏蔽 1: 屏蔽	读/写	0x1

6. RX_SWRST_REG

SWRST_REG 是软件复位

表 28-18 RX_SWREST_REG 寄存器描述

寄存器	地址	读/写	描述	复位值
RX_SWRST_REG	0x7E00_7014	读/写	Rx 控制器软件复位	0x00000000

位	名称	描述	读/写	复位值
[31:1]	Reserved	保留位	读/写	0x00000000
[0]	Sw_rst	软件复位 0: 设置 1: 复位	读/写	0x0

7. RX_CHID_REG

CHID_REG 用于转换通道 ID

表 28-19 RX_CHID_REG 寄存器描述

寄存器	地址	读/写	描述	复位值
RX_CHID_REG	0x7E00_7018	读/写	MIPI HIS Rx 控制器通道 ID 寄存器	0x00000000

位	名称	描述	读/写	复位值
[31:3]	Reserved	保留位	读	0x00000000
[2:0]	CURR_ID	当前通道 ID	读	0x0

注：因为 RxFIFO 内的数据有相同的通道 ID，在是 RxFIFO 为空以后改变通道 ID。

8. RX_DATA_REG

RX_DATA_REG 是 RxFIFO 输入

表 28-20 DATA_REG 寄存器描述

寄存器	地址	读/写	描述	复位值
RX_DATA_REG	0x7E00_701C	读	MIPI HIS Rx 控制器数据寄存器 (FIFO 输出)	0x00000000

位	名称	描述	读/写	复位值
[31:0]	RxFIFO out	RxFIFO 数据输出	读	0x0

注：一旦数据下载到 RxFIFO 上，RxFIFO 内的数据将通过 MIPI HIS Rx 控制器被转换到其他部分的 RX，直到 RxFIFO 完全变空为止。

29 SPI 接口

串行外设接口（SPI）能进行串行数据传输。SPI 包括两个 8、16、32 位的移位寄存器，分别用于传输和接收。在 SPI 传输期间，数据同步传输（串行输出）和接收（串行输入）。

29.1 SPI 接口的特性

SPI支持下面特性：

- 全双工。
- 用于发送和接收的8/16/32位移位寄存器。
- 8位预分频逻辑。
- 三个时钟源。
- 8/16/32位总线接口。
- 两个独立的发送和接收FIFO。
- 主控器模式和从属器模式。

29.2 信号描述

如表 29-1 所示，列出了 SPI 和外部设备之间的外部信号。当无效时，SPI 的所有端口可以作为通用 I/O 端口。

表 29-1 外部信号描述

名称	方向	描述
XspiCLK	输入输出	XspiCLK 是串行时钟，用于控制传输数据的时间。
XspiMISO	输入输出	在主控器模式，该端口是输入端口。输入模式用于从从属器输出端口获得数据。数据通过从属器模式下的端口传输到主控器。
XspiMOSI	输入输出	在主控器模式，该端口是输出端口。该端口用于从主控器输出端口传输数

		据。数据通过从属器模式下的端口被接收。
XspiCS	输入输出	从属器选择信号，当 XspiCS 是低电平时，所有数据发送/接收依次被执行。

29.3 SPI 操作

S3C6410 中，SPI在S3C6410和外部设备之间传输1位串行数据。S3C6410中的SPI支持CPU或DMA分别发送或接收FIFO，并且同时双向传输数据。SPI有两个通道，发送通道和接收通道。

CPU（或DMA）必须在SPI_TX_DATA寄存器写入数据以将数据写入到FIFO。寄存器的数据自动移动到发送FIFO。为了从接收FIFO读数据，CPU（或DMA）必须访问寄存器SPI_RX_DATA，并且数据被自动发送到寄存器SPI_RX_DATA。

1. 操作模式

HS_SPI有两个模式，主控器模式和从属器模式。在主控器模式下，HS_SPICLK产生，并被发送到外部设备。PSS选择从属器的信号，当它为低电平时，表示数据有效。在信息包开始发送和传输前PSS必须设置为低电平。

2. FIFO访问

S3C6410的SPI支持CPU和DMA访问FIFO。CPU访问和DMA访问FIFO的数据大小可以选择8位或者32位。如果选择8位数据大小，有效位为0到7位。通过触发用户定义的阈值，CPU访问正常打开和关闭。每个FIFO的触发器级别被设置从0~64字节。SPI_MODE_CFG 寄存器的TxDMA0n或RxDMA0n位必须设置DMA访问。DMA访问仅仅支持单传输和四个脉冲传输。在发送FIFO时，DMA要求信号为高电平直到FIFO满。在接收FIFO时，如果FIFO 非空，DMA要求信号是高电平。

3. 在接收FIFO中的结尾字节

在中断模式下，接收FIFO中采样的数量小于阈值的值，或者是DMA 四个脉冲模式，并且没有其它数据被接收，该保留字节被称做结尾字节。为了在接收FIFO中移动这些字节，需要用到内部时钟和中断信号。基于APB总线时钟，内部时钟的值能被设置到1024时钟。当定时器值为0时，中断信号发生，并且CPU能在FIFO中移动结尾字节。

4. 信息包数目控制

在主控器模式下，SPI可以控制接收的信息包的数量。如果有信息包要被接收，则设置SFR(Packet_Count_reg)有多少包要接收。当包的数量和设置的数量相同时，SPI 停止产生SPICLK。在该功能被再次装载前，它严格遵循软件和硬件复位（软件复位能清除了特殊功能的寄存器外的所有的寄存

器，但是硬件复位清除所有的寄存器）。

5. NCS控制

nCS可以选择自动控制和手动控制。对于手动控制，Auto_n_Manual必须设置为默认值0。nCS电平设置和设置的nSSout位相同。自动控制下，nCS能被固定在包与包之间。Auto_n_Manual设置为1，只要nCS不活动，nCS_time_count必须设置。这时nSSout是可用的。

6. SPI传输格式

S3C6410支持4种不同的格式来传输数据。如图29-1所示，描述了SPICLK的4种波形。

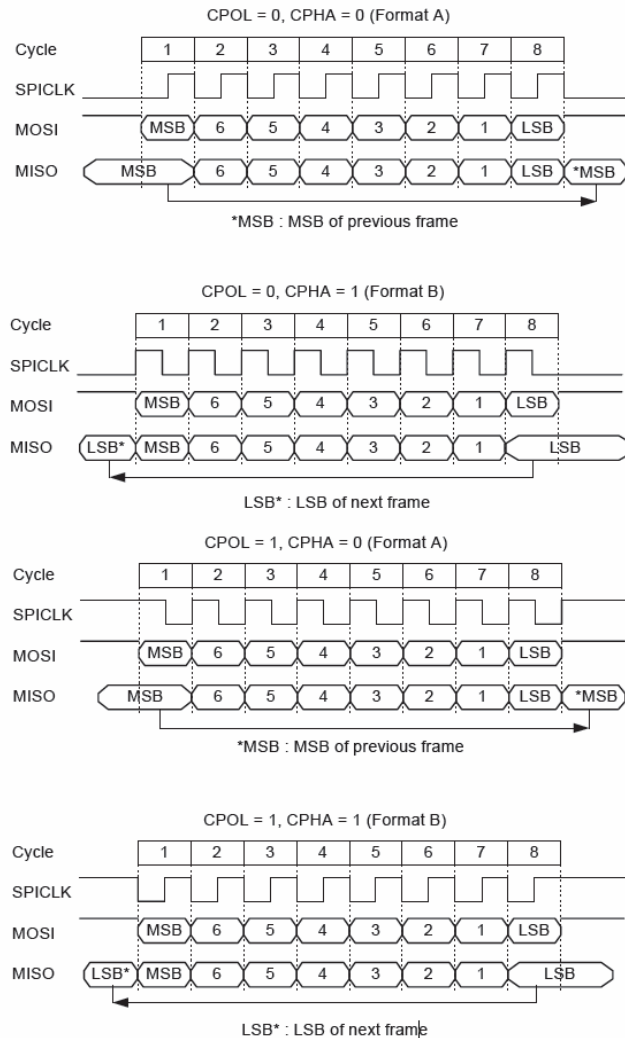


图 29-1 SPI 传输格式

29.4 特殊功能的寄存器描述

特殊功能的寄存器必须以下面的顺序设置 (nCS 为手动模式)：

- (1) 设置传输类型 (CPOL 或 CPHA 设置)。
 - (2) 设置时钟设置寄存器。
 - (3) 设置 SPI MODE 设置寄存器。
 - (4) 设置 API INT_EN 寄存器。
 - (5) 如果需要的话，设置信息包数量配置寄存器。
 - (6) 设置发送和接收通道启动。
 - (7) 设置 nSSout 为低位来开始发送或接收操作。
- 设置 nSSout 为低位，那么启动发送数据写。
 - 如果自动芯片选择位被设置，则不应当控制 nCS。

特殊功能寄存器

寄存器	地址	读/写	描述	初始值
CH_CFG (Ch0)	0x7F00B000	读/写	SPI配置寄存器。	0x0
CH_CFG (Ch1)	0x7F00C000	读/写	SPI配置寄存器。	0x0

CH_CFG	位	读/写	描述	初始状态
SW_RST	[5]	读/写	软复位。 0:不活动 1: 活动	1' b0
SLAVE	[4]	读/写	SPI通道是主控器还是从属器。 0: 主控器 1: 从属器	1' b0
CPOL	[3]	读/写	确定一个有效的高位或者低位时钟。 0: 有效的高位 1:有效的低位	1' b0
CPHA	[2]	读/写	从两个基本的不同传输格式中选择一个。 0: 格式A 1: 格式 B	1' b0
RxCh0n	[1]	读/写	SPI 接收通道打开。 0: 通道关闭 1: 通道打开	1' b0

TxCh0n	[0]	读/写	SPI 发送通道打开。 0: 通道关闭1: 通道打开	1' b0
--------	-----	-----	-------------------------------	-------

寄存器	地址	读/写	描述	初始值
Clk_CFG (Ch0)	0x7F00B004	读/写	时钟配置寄存器。	0x0
Clk_CFG (Ch1)	0x7F00C004	读/写	时钟配置寄存器。	0x0

Clk_CFG	位	读/写	描述	初始状态
ClkSel	[10:9]	读/写	选择时钟源来产生SPI 时钟输出。 00 : PCLK 01 : USBCLK 10 : Ep11 时钟 11 : 保留 *对于使用 USBCLK 源, USB_SIG_MASK 在系统控制器必须设置为打开。	2' b0
ENCLK	[8]	读/写	时钟开/关。 0 : 无效 1 : 有效	1' b0
Prescaler Value	[7:0]	读/写	SPI 时钟输出分频频率。 SPI 时钟输出 = 时钟源 / (2 × (预分频值 +1))	8' h0

寄存器	地址	读/写	描述	初始值
MODE_CFG (Ch0)	0x7F00B008	读/写	SPI FIFO控制寄存器。	0x0
MODE_CFG (Ch1)	0x7F00C008	读/写	SPI FIFO控制寄存器。	0x0

MODE_CFG	位	读/写	描述	初始状态
Ch_tran_size	[30:29]	读/写	00 : 字节 01 : 半字 10 : 字 11 : 保留	2' b0
Trailing Count	[28:19]	读/写	计数值, 从接收FIFO中写入的最后数据到覆盖FIFO中结尾的字节。	10' b0
BUS transfer size	[18:17]	读/写	00: 字节 01: 半字	2' b0