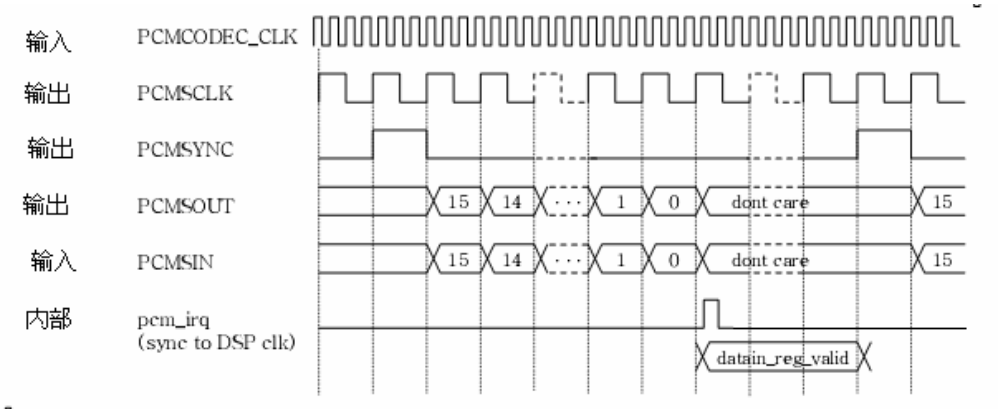

图 37-2 PCM 的时序，POS_MSB_WR/RD = 1

S3C6410X 可以提供多种时钟的 PCM。PCM 能选择两个时钟 PCLK 或 AUDIO（来自系统控制器）。我们还可以选择音频时钟里的 PLL 或外部输入时钟。如图 37-3 所示。

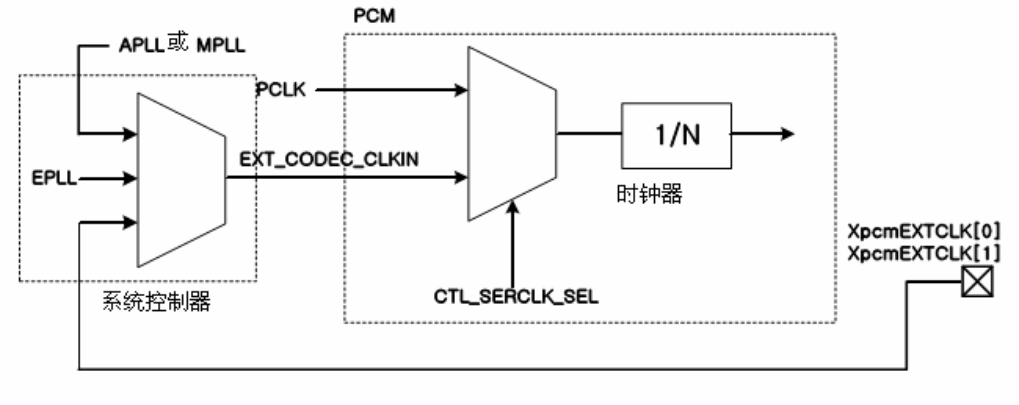

图 37-3 PCM 输入时钟图

37.3 PCM 寄存器

37.3.1. PCM 寄存器概要

寄存器	地址	读/写	描述	复位值
PCM_CTL	0x7F009000 0x7F00A000	读/写	PCM 主控制器。	0x00000000

PCM_CLKCTL	0x7F009004 0x7F00A004	读/写	PCM 时钟和移位控制。	0x00000000
PCM_TXFIFO	0x7F009008 0x7F00A008	读/写	PCM Tx FIFO 写入端口。	0x00000000
PCM_RXFIFO	0x7F00900C 0x7F00A00C	读/写	PCM Rx FIFO 读取端口。	0x00000000
PCM_IRQ_CTL	0x7F009010 0x7F00A010	读/写	PCM 中断控制器。	0x00000000
PCM_IRQ_STAT	0x7F009014 0x7F00A014	读	PCM 中断状态。	0x00000000
PCM_FIFO_STAT	0x7F009018 0x7F00A018	读	PCM Tx 默认值。	0x00000000
PCM_CLRINT	0x7F009020 0x7F00A020	写	PCM 中断清除。	0x00000000

37.3.2. PCM 控制寄存器

PCM_CTL 寄存器主要用于控制各个方面的 PCM 模块。它也提供一个状态位，该位用于提供用轮询代替基于中断的控制的选项。

寄存器	地址	读/写	描述	复位值
PCM_CTL	0x7F009000 0x7F00A000	读/写	PCM 主控制器	0x00000000

为 PCM_CTL 控制寄存器的位定义说明如下：

PCM_CTL	位	描述	初始状态
Reserved	[31:19]	保留。	
TXFIFO_DIPSTICK	[18:13]	当 Almost_full, Almost_empty 的标记为 TXFIFO 启动时，使确定。 Almost_empty: fifo_depth < fifo_dipstick 。	0

		Almost_full: fifo_depth > (32 - fifo_dipstick)。 注意：如果 fifo_dipstick == 0 Almost_empty ，Almost_full 是无效的。 注：对 TX FIFO 的 DMA 装载，Txfifo_dipstick >=2。 PCM_TXDMA 几乎都用作 DMA 请求（保持请求数据，直到 FIFO 差不多满了为止），该寄存器被请求。在一些条件下，DMA 请求完成后，DMA 写入更多的字。因此，FIFO 几乎为满的标志在只剩最少的空间用于额外的字时有效。	
RXFIFO_DIPSTICK	[12:7]	当 almost_full, almost_empty 的标记为 RXFIFO 启动有效时，使确定。 Almost_empty: fifo_depth < fifo_dipstick 。 Almost_full: fifo_depth > (32 - fifo_dipstick)。 注意：如果 fifo_dipstick == 0 Almost_empty ，Almost_full 是无效的 注：对 DMA，RXFIFO_DIPSTICK 是不注意 RX FIFO 的 DMA 卸载的，使用 rx_fifo_empty 标记作为 DMA 请求。	0
PCM_TX_DMA_EN	[6]	启动 DMA 接口为 TXFIFO DMA 必须在需求模式下操作。只要 TXFIFO 不是几乎充分的，DMA_TX 将请求发生。	0
PCM_RX_DMA_EN	[5]	启动 DMA 接口为 RXFIFO DMA 必须在需求模式下操作。只要 RXFIFO 不是几乎充分的，DMA_RX 将请求发生。	0
TX_MSB_POS	[4]	在串行输出流中，控制 MSB 位的位置，相对的 PCMSYNC 信号。 0：MSB 发送在同一时钟,PCMSYNC 为高位	0

		1：MSB 发送在下一个 PCMSCLK 周期后，PCMSYNC 为高位	
RX_MSB_POS	[3]	在串行输入流中，控制 MSB 位的位置，相对的 PCMSYNC 信号。 0：在同一周期，MSB 被夺取在 PCMSCLK 的下降边缘上，PCMSYNC 为高位。 1：在期间周期后，MSB 被夺取在 PCMSCLK 的下降边缘上，PCMSYNC 为高位。	0
PCM_TXFIFO_EN	[2]	启动 TXFIFO。	0
PCM_RXFIFO_EN	[1]	启动 RXFIFO。	0
PCM_PCM_ENABLE	[0]	PCM 启动信号。启动串行转移状态机。 启动必须设置为高位，以便 PCM 的操作。 当启动为低，PCMSOUT 将不会切换。 此外，当启动为低，内部分隔计数器将被保存重置。 当启动为低，内部 FIFO 清除和初始化。	0

37.3.3. PCMCLK 控制寄存器

寄存器	地址	读/写	描述	复位值
PCM_CLKCTL	0x7F009004 0x7F00A004	读/写	PCM 时钟和移位控制	0x00000000

定义位为 PCM_CLKCTL 控制寄存器说明如下：

PCM_CLKCTL	位	描述	初始状态
Reserved	[31:20]	保留。	
CTL_SERCLK_EN	[19]	启动串行时钟分工逻辑。 为 PCM 操作必须为高位（如果是高位，操作 SCLK 和 FSYNC）。	0
CTL_SERCLK_SEL	[18]	选择串行时钟的初始 0：外部编解码器时钟输入	0

		1：PCLK	
SCLK_DIV	[17:9]	控制分频器来创建基于 PCMCODEC_CLK 的 PCMSCLK。时钟是 source_clk / 2×(sclk_div+1)。	000
SYNC_DIV	[8:0]	控制 PCMSYNC 信号的频率基于 PCMSCLK 上。 PCMSYNC 频率 = PCMSCLK 频率/（SYNC_DIV+1)。	000

37.3.4. PCM TxFIFO 寄存器

寄存器	地址	读/写	描述	复位值
PCM_TXFIFO	0x7F009008 0x7F00A008	读/写	PCM Tx FIFO 写入端口	0x00000000

为 PCM_ TXFIFO 控制寄存器的定义位说明如下：

PCM_TXFIFO	位	描述	初始状态
Reserved	[31:17]	保留。	
TXFIFO_DVALID	[16]	TXFIFO 数据是有效的。 写：无效。 读：TXFIFO 读取数据有效。 1：有效 0：无效（可能读一个空的 FIFO）	0
TXFIFO_DATA	[15:0]	TXFIFO 数据。 写：TXFIFO 数据被写入 TXFIFO。 读：TXFIFO 被用于读 APB 界面。	0

37.3.5. PCM RxFIFO 寄存器

寄存器	地址	读/写	描述	复位值
PCM_RXFIFO	0x7F00900C 0x7F00A00C	读/写	PCM Rx FIFO 读取端口。	0x00000000

为 PCM_ RXFIFO 控制寄存器的定义位说明如下：

PCM_RXFIFO	位	描述	初始状态
Reserved	[31:17]	保留。	
RXFIFO_DVALID	[16]	RXFIFO 数据是有效的。 写：无效。 读：TXFIFO 读取数据有效。 1：有效 0：无效（可能读一个空的 FIFO）	0
RXFIFO_DATA	[15:0]	RXFIFO 数据 写：RXFIFO 数据被写入 TXFIFO。 读：RXFIFO 被用于读 APB 界面。 注：写 RXFIFO 意思是支持调试。	0

37.3.6. PCM 中断控制寄存器

寄存器	地址	读/写	描述	复位值
PCM_IRQ_CTL	0x7F009010 0x7F00A010	读/写	PCM 中断控制器。	0x00000000

为 PCM_IRQ_CTL 控制寄存器的定义位说明如下：

PCM_IRQ_CTL	位	描述	初始状态
Reserved	[31:15]	保留。	
EN_IRQ_TO_ARM	[14]	控制是否 PCM 中断发送到 ARM。 1：PCM IRQ 转交 ARM 子系统 0：PCM IRQ 不转交 ARM 子系统	0
Reserved	[13]	保留	0
TRANSFER_DONE	[12]	每次，一个字的串行移位完成，中断产生。	0

		1：IRQ 初始启动 0：IRQ 初始禁止	
TXFIFO_EMPTY	[11]	只要 TxFIFO 是空，产生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_ALMOST_EMPTY	[10]	只要 TxFIFO 几乎为空，产生中断。定义几乎为空，类似 FIXME 字保留。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_FULL	[9]	TxFIFO 为充分，产生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_ALMOST_FULL	[8]	只要 TxFIFO 几乎为充分，产生中断。定义几乎为充分，类似 FIXME 字保留。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_ERROR_STARVE	[7]	由于 TxFIFO starve 错误，中断产生。 当它仍是空白时，只要发生 TXFIFO 被读取就发生这个情况。 这被视为一种错误，并会有意外的结果。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_ERROR_OVERFLO	[6]	由于 TXFIFO 溢出错误，产生中断。 当它已经完整时，只要发生 TXFIFO 写入就发生这种情况。 这被视为一种错误，并会有意外的结果。 1：IRQ 初始启动	0

		0：IRQ 初始禁止	
RXFIFO_EMPTY	[5]	只要 RXFIFO 是空，产生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
RXFIFO_ALMOST_EMPTY	[4]	只要 RxFIFO 几乎为空产生中断。定义几乎为空类似 FIXME 字保留。 1：IRQ 初始启动 0：IRQ 初始禁止	0
RX_FIFO_FULL	[3]	只要 RxFIFO 为充分产生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
RX_FIFO_ALMOST_FULL	[2]	只要 RxFIFO 几乎为充分 中断产生。 定义几乎为充分类似 FIXME 字保留 1：IRQ 初始启动 0：IRQ 初始禁止	0
RXFIFO_ERROR_STARVE	[1]	由于 TxFIFO STARVE 错误，中断产生。当它仍是空白时，只要发生 RXFIFO 被读取就发生这个情况。 这被视为一种错误，并会有意外的结果。 1：IRQ 初始启动 0：IRQ 初始禁止	0
RXFIFO_ERROR_OVERFLOW	[0]	因为 RXFIFO 溢出错误产生中断。 当它已经完整时，只要发生 RXFIFO 写入就发生这种情况。 这被视为一种错误，并会有意外的结果。 1：IRQ 初始启动 0：IRQ 初始禁止	0

37.3.7. PCM 中断状态寄存器

PCM_IRQ_STAT 寄存器用于报告 IRQ 状态。

寄存器	地址	读/写	描述	复位值
PCM_IRQ_STAT	0x7F009014 0x7F00A014	读	PCM 中断状态。	0x00000000

为 PCM_IRQ_ STAT 控制寄存器的定义位说明如下：

PCM_IRQ_STAT	位	描述	初始状态
Reserved	[31:14]	保留	
IRQ_PENDING	[13]	控制是否将 PCM 中断发送到 ARM 上。 1：PCM IRQ 被转发到 ARM 0：PCM IRQ 不被转发到 ARM	0
TRANSFER_DONE	[12]	每次串行移位的一个字完成，发生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_EMPTY	[11]	只要 RxFIFO 几乎为空，产生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_ALMOST_EMPTY	[10]	只要 TxFIFO 几乎为空，中断产生 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_FULL	[9]	只要 TxFIFO 为充分，产生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_ALMOST_FULL	[8]	只要 TxFIFO 几乎为充分产生中断。	0

		定义几乎为充分类似 FIXME 字保留。 1：IRQ 初始启动 0：IRQ 初始禁止	
TXFIFO_ERROR_STARVE	[7]	由于 TxFIFO STARVE 错误，中断产生。 当它是仍是空白时，只要发生 TxFIFO 被读取就发生这个情况。 这被视为一种错误，并会有意外的结果。 1：IRQ 初始启动 0：IRQ 初始禁止	0
TXFIFO_ERROR_OVERFLOW	[6]	因为 TxFIFO 溢出错误产生中断。 当它已经完整时，只要发生 TxFIFO 写入就发生这种情况。 这被视为一种错误，并会有意外的结果。 1：IRQ 初始启动 0：IRQ 初始禁止	0
RXFIFO_EMPTY	[5]	只要 RXFIFO 是空，产生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
RXFIFO_ALMOST_EMPTY	[4]	只要 RxFIFO 几乎为空，产生中断。定义几乎为空类似 FIXME 字保留。 1：IRQ 初始启动 0：IRQ 初始禁止	0
RX_FIFO_FULL	[3]	只要 RxFIFO 为充分，产生中断。 1：IRQ 初始启动 0：IRQ 初始禁止	0
RX_FIFO_ALMOST_FULL	[2]	只要 RxFIFO 几乎为充分，产生中断。 定义几乎为充分类似 FIXME 字保留	0