

GL5116 的音频通路总体分为两部分，音频输入和音频输出。其分别对应于 ADC_FIFO 和 DAC_FIFO，也就是说，输入的数据都是先被放到 ADC_FIFO 中，输出的数据先被放到 DAC_FIFO 中再被取走。

ADC_FIFO 的输入源如下：

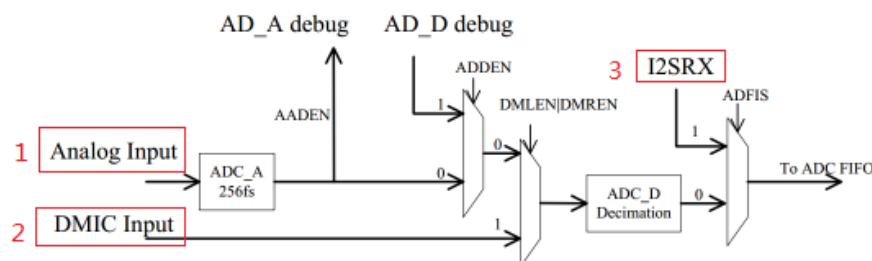


Figure 13-4 ADC Main Block

图 1：ADC_FIFO 的输入源

主要分为 2 部分：

ADC 输入：Analog Input：AUX0；AUX1；AMIC

DMIC Input：DMIC

I2S 输入： I2S_RX

ADC_FIFO 的访问方式：

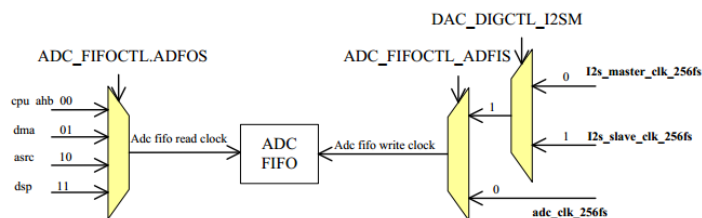


Figure 13-6 ADC Clock

图 2：ADC_FIFO 的访问方式

从上图可以看出 ADC_FIFO 的访问方式分为两种读取写入：

读取：可以将 ADC_FIFO 配置给不同的控制器来访问，如：CPU, DMA, ASRC, DSP，需要配置寄存器 ADC_FIFCTL 中的输出方式。5116 方案中，linein 和录音都是将 ADC_FIFO 配置给 ASRC 的，BTCALL 是配置给 DSP，这样有一个好处直接将 CPU 给解放了，CPU 不用关输入源的数据了，DSP 可以直接从 ADC_FIFO 中读数据或者从 ASRC 这个协处理器将数据取走进行后处理。

问题：为什么 linein 和录音不是 dsp 直接取数从 ADC_FIFO 中？？？

写入：可以通过 ADC 或 I2S 写入，需要配置寄存器 ADC_FIFCTL 中的输入方式，以实现时钟的切换。

向 ADC_FIFO 中写入数据的方法

从上图 1 可以看出首先要确定音频的输入源头，以实现将数据写入到 ADC_FIFO 中：

即步骤 1：确定音频的输入源头。

是 ADC 输入还是 I2S 输入，如果是 ADC 输入的话，还要确定是 AMIC 还是 DMIC

如果是 AMIC 的话，还要确定 AUX0，AUX1，AUX_IN 是单独用还是一起用。

音频输入源头：AUX0；AUX1；AMIC；DMIC；I2S_RX

音频输入源头选择实现方法：

| | |
|--------|---|
| 音频输入源头 | 寄存器:ADC_DIGCTL(0xc0051000) ADC_FIFOCTL(0xc0051004) ADC_ANACTL(0xc0051010) |
| AUX0 | ADC_FIFOCTL Bit3:ADFIS 复位为 0 ADC_ANACTL Bit0-1:ADLEN 和 ADREN 置位为 1 ADC_DIGCTL Bit8-9:DMLen 和 DMREN 复位为 0 ADC_ANACTL Bit2-3:OP0LEN 和 OP0REN 置位为 1 |
| AUX1 | ADC_FIFOCTL Bit3:ADFIS 复位为 0 ADC_ANACTL Bit0-1:ADLEN 和 ADREN 置位为 1 ADC_DIGCTL Bit8-9:DMLen 和 DMREN 复位为 0 ADC_ANACTL Bit7-8:OP1LEN 和 OP1REN 置位为 1 |
| AMIC | ADC_FIFOCTL Bit3:ADFIS 复位为 0 ADC_ANACTL Bit0-1:ADLEN 和 ADREN 置位为 1 ADC_DIGCTL Bit8-9:DMLen 和 DMREN 复位为 0 ADC_ANACTL Bit12-13:MOPLEN 和 MOPREN 置位为 1 |
| DMIC | ADC_ANACTL Bit0-1:ADLEN 和 ADREN 复位为 0 ADC_DIGCTL Bit8-9:DMLen 和 DMREN 置位为 1 |
| I2S_RX | ADC_FIFOCTL Bit3:ADFIS 置位为 1 |

步骤 2：确定音频的输入方式

音频输入方式：单端和差分，输入方式需要硬件支持，并且对模拟信号才是有效的，所以过来的是数字信号的话，像 DMIC；I2S_RX 就没有单端和差分的概念了，对 AUX0；AUX1；MIC 才是有效的，但是 GL5116 的 AUX0 和 AUX1 不支持差分输入，仅仅支持单端输入，MIC 既支持单端又支持差分输入。见下图 3，其中 OP0，OP1，和 MICOP 可以控制输入的数据是否可以到 MIXER，同时还可以对输入进来的模拟信号进行放大或衰减，op0 和 op1 一共 8 级从-12dB-7.5dB；MIC_OP 也是 8 级从 26db-39db。

Left Channel

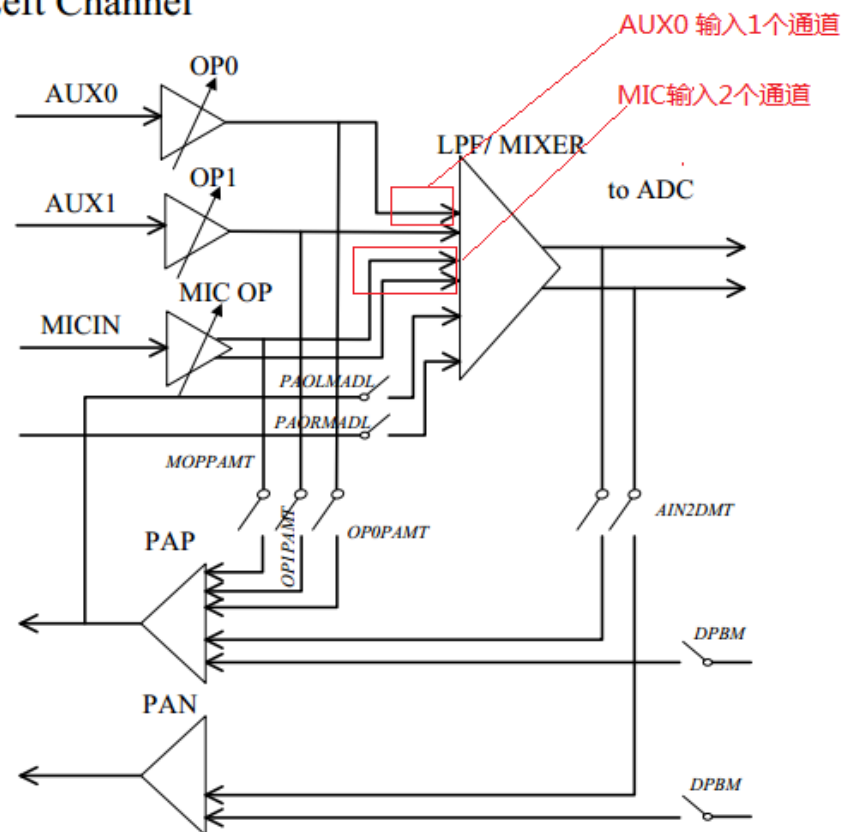


图 3：ADC 输入的连接方式

音频输入方式选择实现方法：

| 音频输入源头 | 寄存器：ADC_ANACTL(0xc0051010) |
|----------|------------------------------|
| MIC 单端输入 | ADC_ANACTLBit15:MFDSES 置位为 1 |
| MIC 差分输入 | ADC_ANACTLBit15:MFDSES 复位为 0 |

步骤 3 音频输入增益：有时候需要对输入进来的模拟信号进行衰减或放大，只需要配置下 ADC_ANACTL 寄存器中对应的 op 增益即可。有时候也需要对数字信号进行放大，只需要配置下 ADC_DIGCTL 中的 ADCGC 即可。

我的理解：ADCGC 实现数字信号放大的原理是，先从 ADC_FIFO 中读取数据，然后放大，将其给相应的控制器。

步骤 4 ADC_FIFO 输入时钟：通过前 3 个步骤实现了输入信号的准备工作，下面就是采集数据到 ADC_FIFO 中了，这样就需要找时钟源，以确定往 FIFO 中写入的时钟频率。

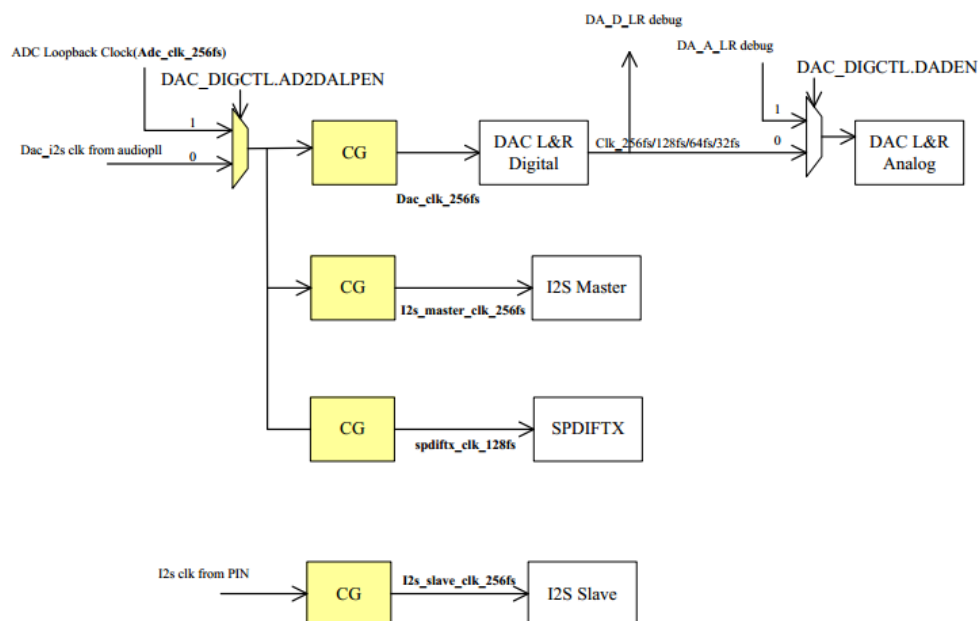


图 4 I2S 时钟

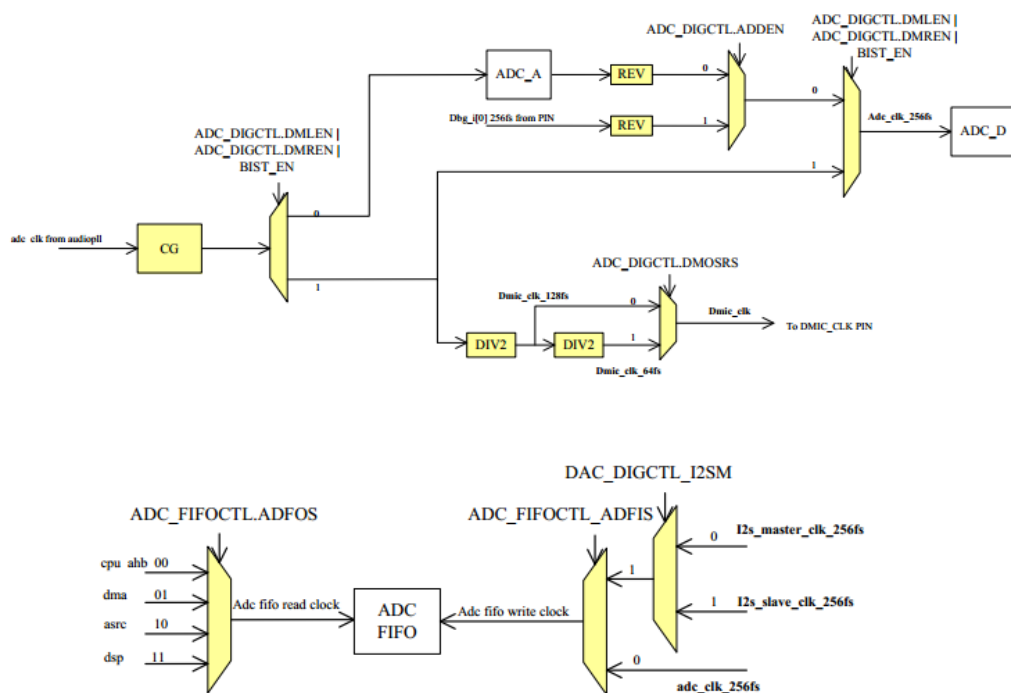


Figure 13-6 ADC Clock

图 5 ADC 时钟

从上图可以看出，ADC_FIFO 的输入时钟有三种：

- (1) I2s_master_clk_256fs
- (2) I2s_slave_clk_256fs
- (3) adc_clk_256fs

其中 (1) 和 (2) 对应 I2S 输入，IC 通过 DAC_DIGCTL 中的 I2SM 位来决定 I2S 的输入方式。

其中 (3) 对应 ADC 输入。

从上图还可以看出，I2S_master_clk_256fs 和 adc_clk_256fs 的源头均是 audio_pll，而 I2S_slave_clk_256fs 来自于外部控制器提供的输入时钟。

audio_pll 的时钟配置：配置 AUDIO_PLL_CTL 寄存器，其中 APS (3:0) bit 决定了 audio_pll 的频率如果其值为 0-3，audio_pll = 22.5792M；如果其值为 4-7，audio_pll = 24.576M。

ADC_FIFO 输入时钟源的选择：ADC_FIFOCCTL Bit3:ADFIS, 0x00 ADC; 0x01 I2SRX。

步骤 5 设置采样率

(1) 如果输入源选择 Analog Input，此时还需要确定 ADC 的采样率

使能 ADCCLKEN

配置 CMU_ADDACLK 中的 ADCCLKDIV[4-6]

注：5116 中支持 ADC 的采样率为 8k/12k/11.025k/16k/22.05k/24k/32k/44.1/48kHz。

(2) 如果输入源选择 DMIC，需要确定下 IC 读取外挂 DMIC 控制器中数据时所需要的频率即上图中的 DMIC_CLK_PIN，通过配置 ADC_DIGCTL 中的 DMOSRS[10] 决定时钟频率是 audio_pll 的 128 还是 64 分频。

从 ADC_FIFO 中读取数据的方法

选择将要从 ADC_FIFO 中读取数据的控制器类型，配置 ADC_DIGCTL 中的 ADFOS[4-5], 0x00: CPU; 0x01: DMA; 0x02: ASRC; 0x03: DSP。

- (1) 如果是 CPU，CPU 有两种方式从 ADC_DAT 中取走数据，中断触发或查询的方式读走数据。中断触发：配置 ADC_FIFOCCTL 中 IRQ 使能，当 ADC_FIFO 满的时候，触发中断给 CPU，CPU 收到后，就可以将数据读走，至于读走的方式可以通过配置协处理器，也可以通过 CPU 直接读取均可，看自己怎么配置。查询方式，CPU 在一个死循环中一直查询 ADC_FIFO 的状态，如果不为空的话，就读走数据，直至为空。
- (2) 如果是 DSP，只能通过查询的方式从 ADC_FIFO 中读走数据，具体实现和 CPU 类似。
- (3) 如果是 DMA，此时牵涉到 DMA 的相关配置，可以参考 DMA 的配置方法，思路如下：
 - 1: DMA 的源地址是 ADC_FIFO
 - 2: 目的地址可以根据需求定义为 FIFO, MEMORY 均可
 - 3: 长度可以根据需求定义
 - 4: reload 模式还是单次触发
 - 5: 看是否使能 DMA 半空半满中断

6:使能 DMA

7:务必使能 ADC_FIFCTL 中的 DRQ,使得 ADC_FIFO 中的数据满后,能够给 DMA 一个信号,让其开始工作。

- (4) 如果是 ASRC,此时牵涉到 ASRC 的相关配置,现在只考虑 ASRC 怎么将数据取走,ASRC 的写时钟应该选择为 IIS-rx clock

DAC_FIFO 的输入源和输出源如下:

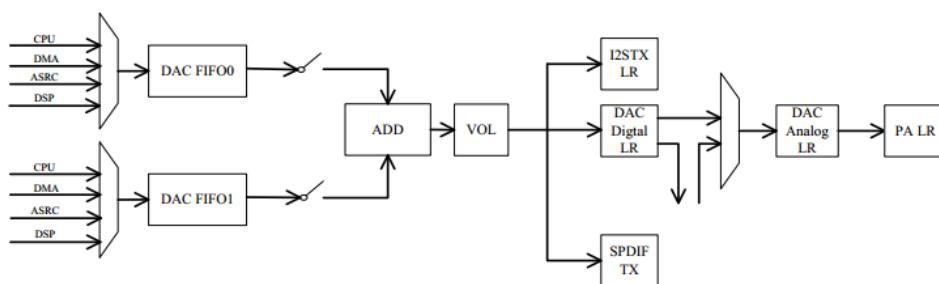


图 6 DAC_FIFO 中的数据输入源

从上图也可以得出以下结论:

- 1: 输入源: 和 ADC_FIFO 类似,可以有多个控制器 CPU,DMA,ASRC 和 DSP,往 DAC_FIFO 中写入数据。
- 2: 输出源: I2S_TX; 内部 DAC; SPDIF_TX
- 3: DAC_FIFO 存在两个,其里面的数据是经过叠加后,在被输出源取走,注意叠加的时候溢出的处理。

注: 使用场景: 播歌或 TTS 播报的时候,同时播放按键音。

向 DAC_FIFO 中写入数据的方法

配置向 DAC_FIFO 中写入数据的控制器类型,是 CPU,DMA,ASRC 还是 DSP?

配置 DAC_FIFCTL 中的 DAF1IS[12-13]和 DAF0IS[4-5]即可。其中 0x00: CPU;0x01: DMA;0x02: ASRC;0x03: DSP。

- (1) 如果是 CPU, CPU 有两种方式向 DAC_FIFO 中写入数据,中断触发或查询的方式。
中断触发:配置 DAC_FIFCTL 中 IRQ 使能,当 DAC_FIFO 空的时候,触发中断给 CPU, CPU 收到后,就可以往 DAC_FIFO 写入数据,至于写入的方式可以通过配置协处理器,也可以通过 CPU 直接写入均可,看自己怎么配置。查询方式, CPU 在一个死循环中一直查询 DAC_FIFO 的状态,如果为空的话,就写入数据,直至为满。
- (2) 如果是 DSP,只能通过查询的方式从 DAC_FIFO 中写入数据,具体实现和 CPU 类似。
- (3) 如果是 DMA,此时牵涉到 DMA 的相关配置,可以参考 DMA 的配置方法,思路如下:
1: DMA 的目的地址是 DAC_FIFO

- 2: 目的地址可以根据需求定义为 FIFO, MEMORY 均可
 - 3: 长度可以根据需求定义
 - 4: reload 模式还是单次触发
 - 5: 看是否使能 DMA 半空半满中断
 - 6: 使能 DMA
 - 7: 务必使能 DAC_FIFOCCTL 中的 DRQ, 使得 DAC_FIFO 中的数据为空后, 能够给 DMA 一个信号, 让其开始工作。
- (5) 如果是 ASRC, 此时牵涉到 ASRC 的相关配置, 现在只考虑 ASRC 怎么将数据取走, ASRC 的读时钟应该选择为 IIS-tx clock

从 DAC_FIFO 中读取数据的方法

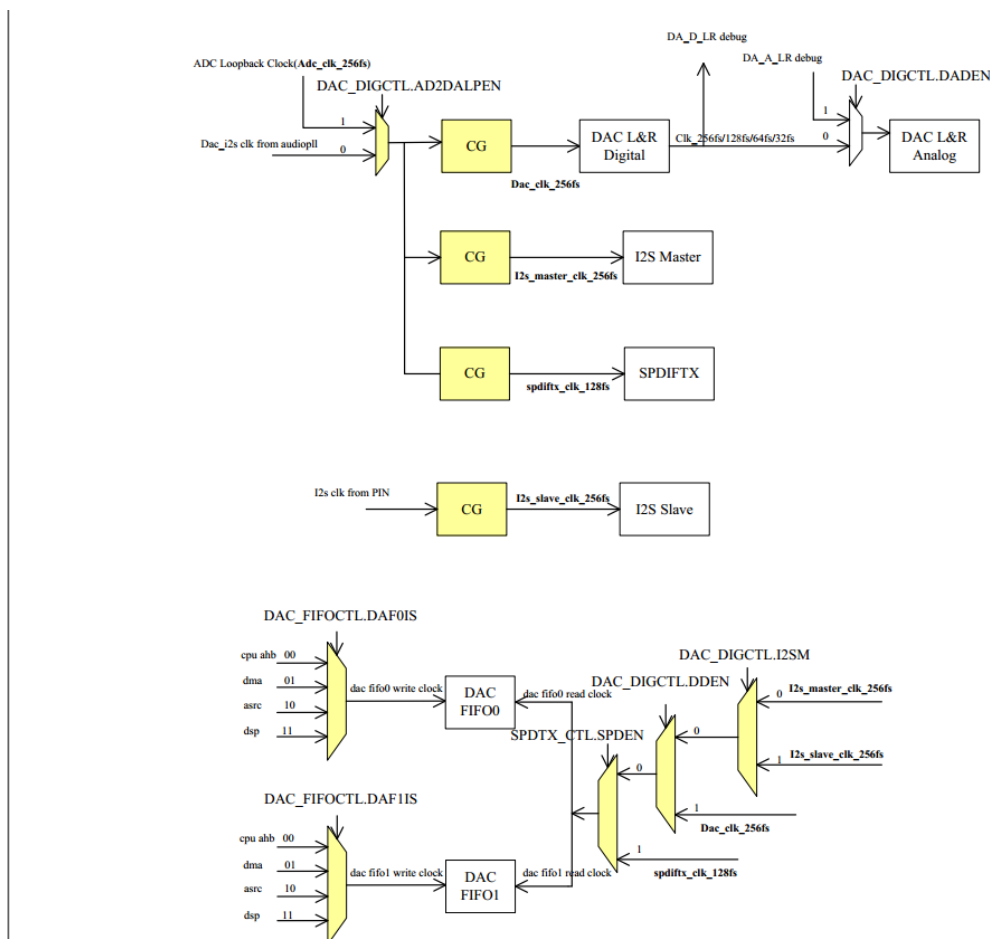


图 7 DAC 时钟

1: 确定输出源头

| 音频输出源头 | 音频输出源头配置 |
|--------|--|
| DAC_PA | CMU_DEVCLKEN 中 DAC 时钟使能 支持 FIFO0 和 FIFO1 数据 mix |

| | |
|----------|--|
| | DAC_DIGCTL 中 DDEN 使能（数字部分使能） |
| I2S_TX | CMU_DEVCLKEN 中主时钟或从时钟使能 DAC_DIGCTL 中 I2STXEN 使能 DAC_DIGCTL 中 I2SM 选择是主模式还是从模式 配置相关 GPIO，通过 I2S 总线传递音频数据 |
| SPIDF_TX | CMU_DEVCLKEN 中 SPDIF 时钟使能 配置相关 GPIO，通过 SPDIF_tx 接收数据 SPDCTX_CTL 中使能 SPDIF |

2：输出方式

| 音频输出源头 | 音频输出方式 |
|----------|----------------------|
| DAC_PA | 单端或差分 直驱或非直驱 高摆幅或低摆幅 |
| I2S_TX | XXX |
| SPIDF_TX | XXX |

3：输出增益

| 音频输出源头 | 输出增益 |
|----------|--|
| DAC_PA | 模拟部分：PA_VOLUME 中 PALRVOL[0-5] 数字部分：VOL_LCH 和 VOL_RCH[0-7] |
| I2S_TX | 和外部 DAC 控制器相关 |
| SPIDF_TX | --- |

4：从图 7 DAC 时钟可以看出，DAC_FIFO 的输出时钟有三种

- (1) I2S_master_clk_256fs
- (2) I2S_slave_clk_256fs
- (3) dac_clk_256fs
- (4) spdif_tx_clk_128fs

其中（1）和（2）对应 I2S 输出，IC 通过 DAC_DIGCTL 中的 I2SM 位来决定 I2S 的输入方式。

其中（3）对应内部 DAC 输出。

其中（4）对应 SPDIF 输出。

从上图 7 还可以看出，spdif_tx_clk_128fs，I2S_master_clk_256fs 和 adc_clk_256fs 的源头均是 audio_pll，而 I2S_slave_clk_256fs 来自于外部控制器提供的输入时钟。

5：设置采样率

如果输出源选择的是内部 dac_pa，此时还需要确定 dac 的采样率

- (1) 使能 ADCCLKEN
- (2) 配置 CMU_ADDACLK 中的 ADCCLKDIV[4-6]

注：5116 中支持 DAC 的采样率为 8k/12k/11.025k/16k/22.05k/24k/32k/44.1/48kHz。