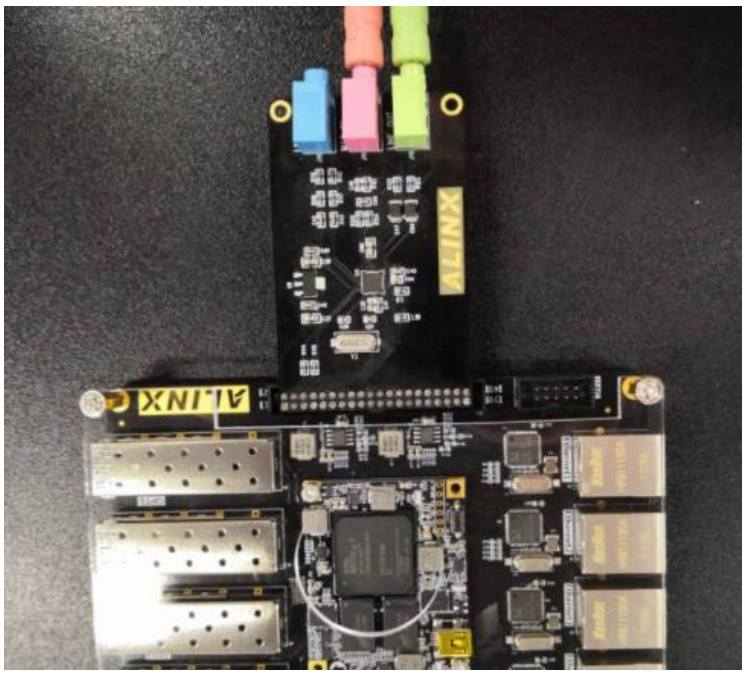
录音与播放例程

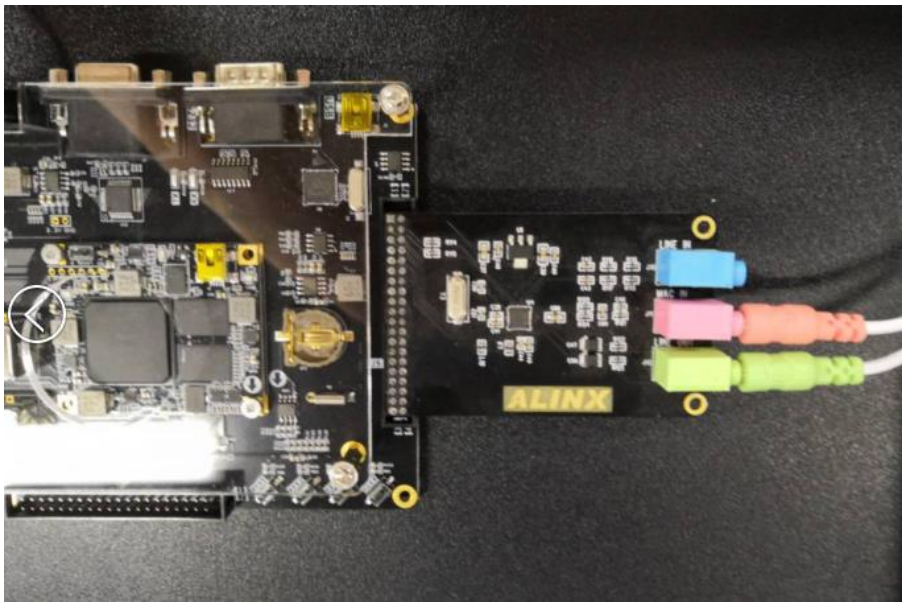
黑金动力社区 2020-03-13

1 实验简介

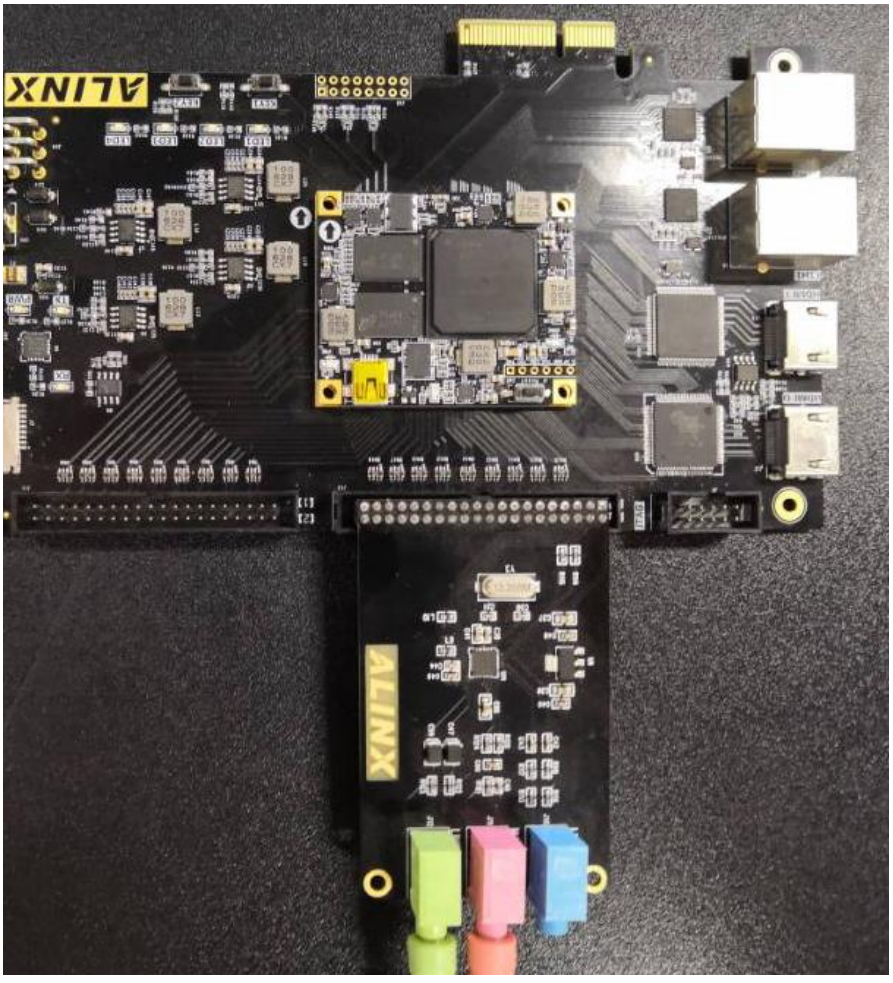
本实验的录音和播放实验因为 AX7101/AX7102/AX7103 开发板上没有音频部分的电路，需要外接一个芯驿电子（黑金）的 AUDIO 音频模块 AN831。



AX7101 音频连接



AX7102 音频连接



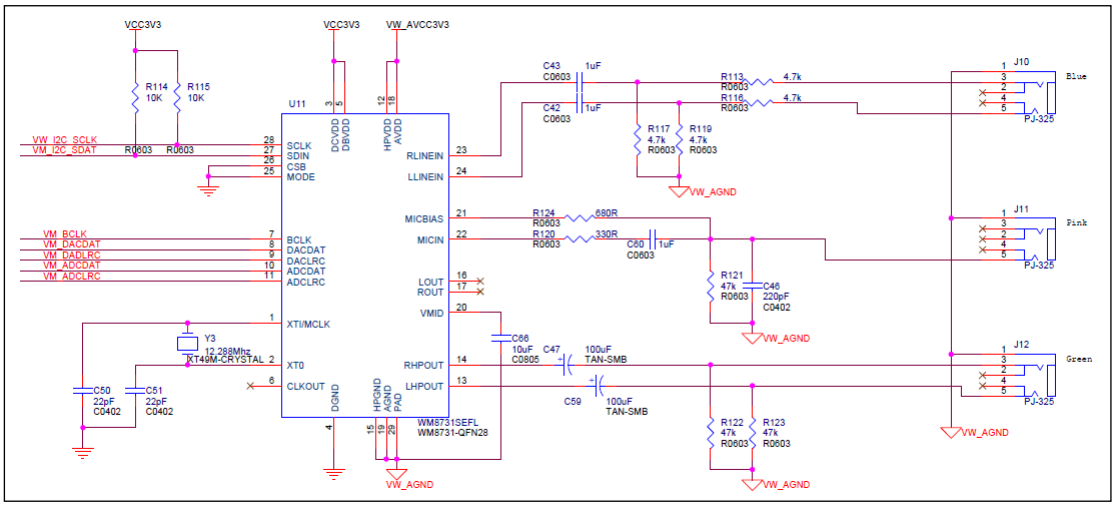
AX7103 音频连接

音频模块上有三个音频连接器,其中粉色的接口为麦克风输入;绿色的接口为耳机输出;蓝色的接口为音频输入，用于连接 DVD 等音频输出口。本实验将实现音频模块和 FPGA 之间的数据通信,通过音频模块把麦克风输入的语音数据存储到 DDR3 存储器里,再把音频数据发送给音频模块,从耳机接口进行语音的播放,从而实现录音和播放的功能。

2 实验原理

2.1 硬件介绍

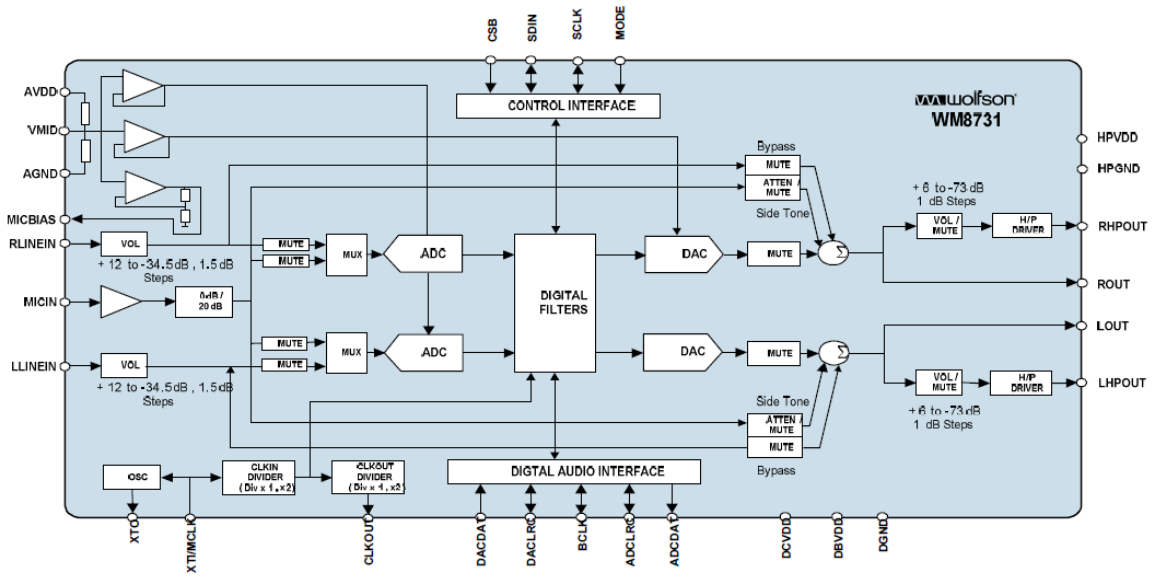
开发板通过 40PIN 的扩展口和 AN831 音频模块连接，AN831 音频模块使用 WOLFSON 公司的 WM8731 芯片实现声音信号的 A/D 和 D/A 转换功能。以下为 AN831 音频模块的硬件电路：



2.2 WM8731 配置和时序

这里简单介绍一下音频模块 AN831 用到的音频编/解码芯片 WM8731。该芯片在本设计中主要完成声音信号在采集和回放过程中的 A/D 和 D/A 转换功能。该芯片的 ADC 和 DAC 的采样频率为 8KHz 到 96KHz 可调，可转换的数据长度为 16-32 位可调。WM8731 的内部有 11 个寄存器。该芯片的初始化以及工作时的工作状态和功能都是通过以 I2C 总线方式对其内部的这 11 个寄存器进行相应的配置来实现的。本设计中 WM8731 工作于主模式，采样频率设为 48KHZ,转换的数据位长度为 16 位。WM8731 的数字音频接口有 5 根引脚，分别为：BCLK(数字音频位时钟)、DACDAT(DAC 数字音频数据输入)、DACLRC(DAC 采样左/右声道信号)、ADCDAT(ADC 数字音频信号输出)、ADCLRC(ADC 采样左/右声道信号)。在本设计中 FPGA 为从设备，WM8731 为主设备。ADCDAT、DACDAT、ADCLRC 和

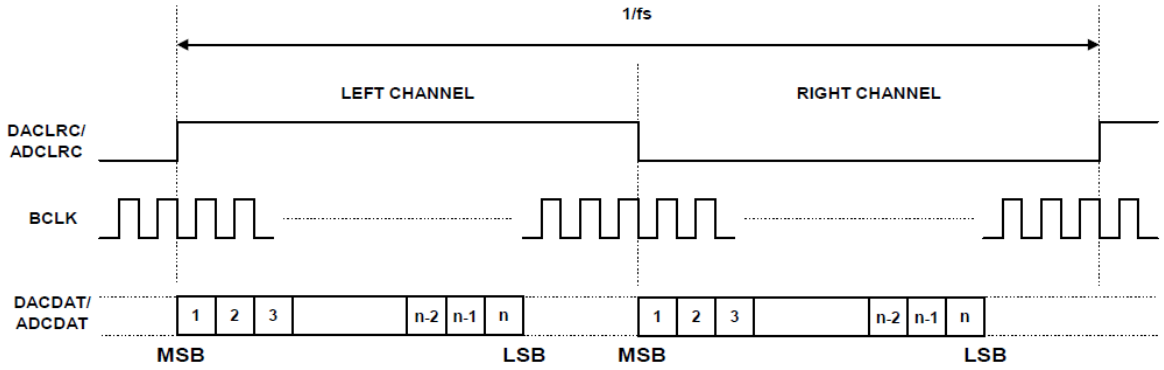
DACLRC 与位时钟 BCLK 同步，在每个 BCLK 的下降沿进行一次数据传输。BCLK、DACDAT、DACLRC、ADCLRC 为 WM8731 的输入信号。ADCDAT 为 WM8731 的输出信号。



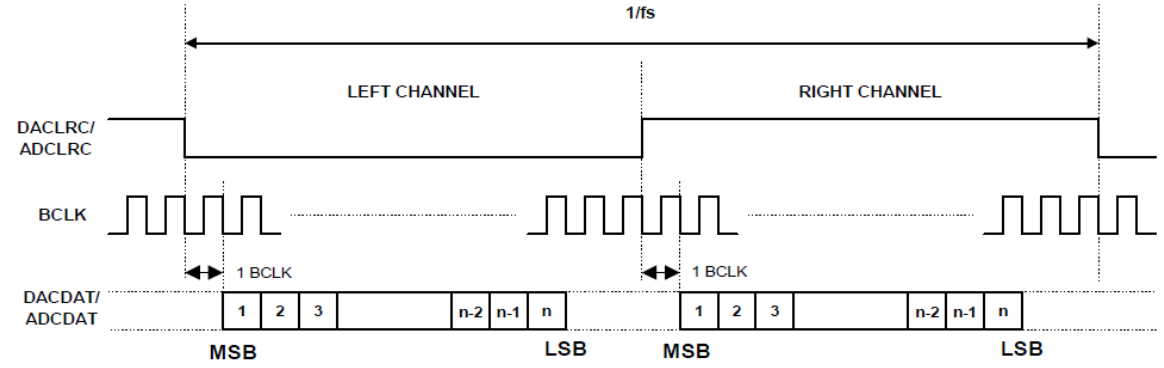
在本系统中 FPAG 和 WM8731 的控制和数据通信将用到 I2C 和数字音频总线接口。FPGA 通过 I2C 接口配置 WM8731 的寄存器，通过 I2S 总线接口来进行音频数据的通信。关于 I2C 接口，其他实验中已经有讲解，下面我们主要来了解数字音频接口。

数字音频接口可提供 4 种模式：

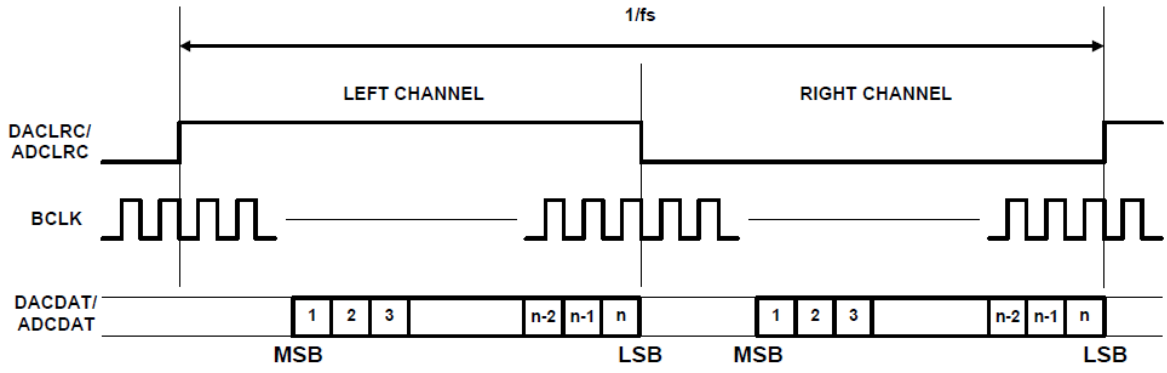
- Right justified
- Left justified
- I2S
- DSP mode



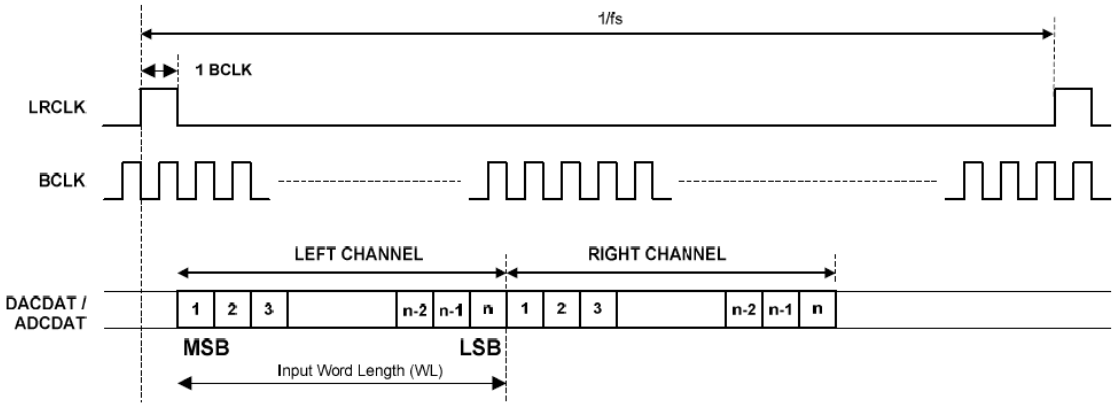
Left justified 模式



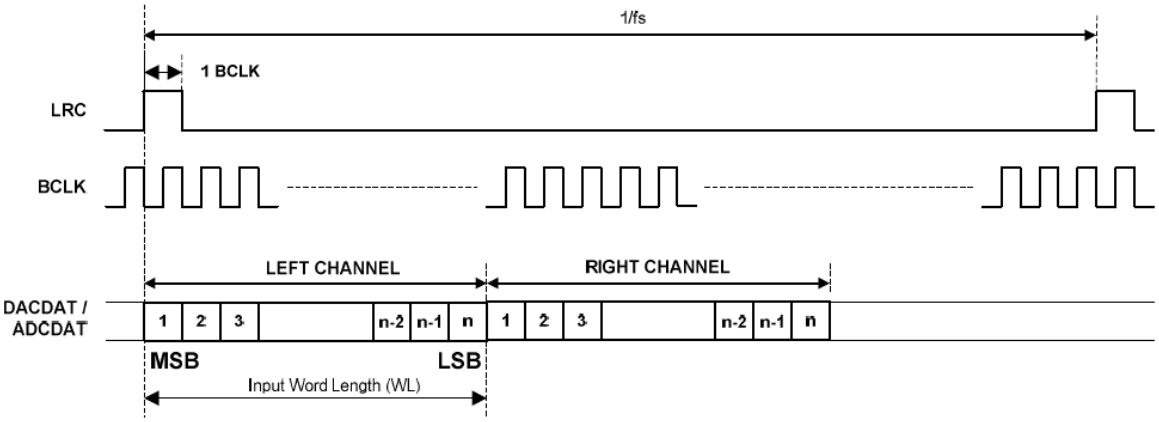
I2S 模式



Right justified 模式



DSP/PCM 模式（MODE A）



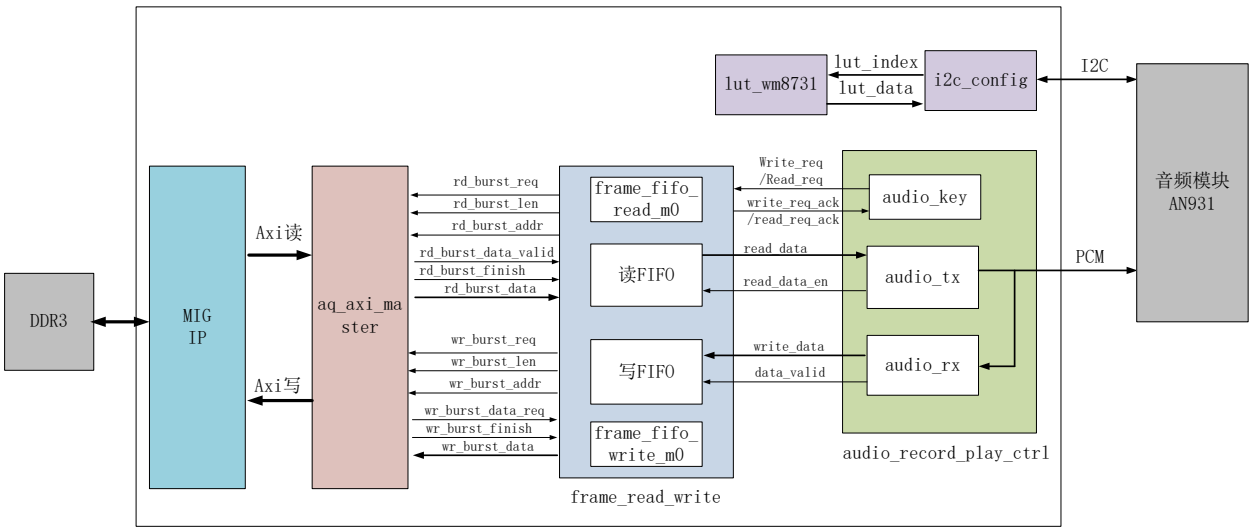
DSP/PCM 模式（MODE B）

本实验选择 Right justified 模式。

REGISTER ADDRESS	BIT	LABEL	DEFAULT	DESCRIPTION
0000111 Digital Audio Interface Format	1:0	FORMAT[1:0]	10	Audio Data Format Select 11 = DSP Mode, frame sync + 2 data packed words 10 = I²S Format, MSB-First left-1 justified 01 = MSB-First, left justified 00 = MSB-First, right justified
	3:2	IWL[1:0]	10	Input Audio Data Bit Length Select 11 = 32 bits 10 = 24 bits 01 = 20 bits 00 = 16 bits
	4	LRP	0	DACLRC phase control (in left, right or I²S modes) 1 = Right Channel DAC data when DACLRC high 0 = Right Channel DAC data when DACLRC low (opposite phasing in I²S mode) or DSP mode A/B select (in DSP mode only) 1 = MSB is available on 2nd BCLK rising edge after DACLRC rising edge 0 = MSB is available on 1st BCLK rising edge after DACLRC rising edge
	5	LRSWAP	0	DAC Left Right Clock Swap 1 = Right Channel DAC Data Left 0 = Right Channel DAC Data Right
	6	MS	0	Master Slave Mode Control 1 = Enable Master Mode 0 = Enable Slave Mode
	7	BCLKINV	0	Bit Clock Invert 1 = Invert BCLK 0 = Don't invert BCLK

3 程序设计

本实验的功能是程序检测按键 KEY2 是否按下，如果检测到 KEY2 按下了，开始录音,录音的最长时间为 20 秒；录音结束后，开始播放刚才录下的音频。本程序设计包含五大部分：WM8731 寄存器配置，MIG IP 调用，AXI 总线转换程序，FIFO 读写控制，音频录音和播放。录音的过程为:采集的音频数据先存储到写 FIFO 中，然后通过 axi 总线和 MIG IP 写入到 DDR3 中; 播放的过程为：通过 axi 总线和 MIG IP 读取 DDR 的音频数据，写入到读 FIFO 中，再并行数据转换成串行数据 PCM 输出音频模块。



➢ WM8731 寄存器配置

lut_wm8731 模块和 i2c_config 模块将寄存器配置地址和配置信息通过查找表的形式通过 I2C 总线写入音频模块中，具体的请参考例程代码和 wm8731 的芯片数据手册。

➢ 音频录音和播放

audio_rx 接收模块，接收从麦克风输入的语音信号，完成左右声道的音频接收，将串行数据转换成并行数据。通过“Right justified”模式的时序图可以看到接收语音信号时在 LRC 信号为高电平，且 BCLK 信号的上升沿时左声道接收数据并完成串行信号转换成并行信号的过程。在 LRC 信号为低电平，且 BCLK 信号的上升沿时右声道接收数据并完成串行信号转换成并行信号的过程。

信号名称	方向	说明
clk	in	时钟输入
rst	in	异步复位输入，高复位
sck_bclk	in	数字音频接口 bit 时钟
ws_lrc	in	ADC 采样时钟
sdata	in	音频数字接口串行数据输入
left_data	out	左声道数据
right_data	out	右声道数据
data_valid	out	音频数据有效

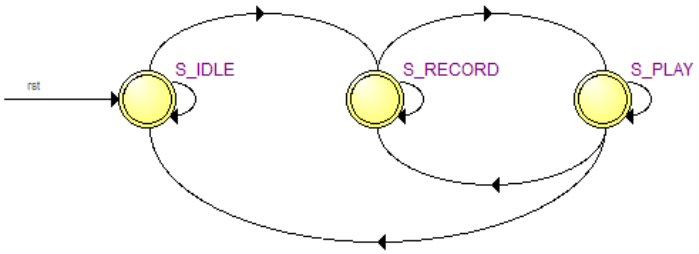
audio_rx 音频接收模块端口

audio_tx 是音频发送模块，完成左右声道音频数据的串行化。同样通过时序图可以看到，语音信号完成模数，数模转换从读 FIFO 输出后进入发送模块，在 LRC 信号上升沿，且即将跳变为高电平时开始发送左声道数据，BCLK 信号下降沿时完成缓存将并行信号转换为串行信号的过程；在 LRC 信号上升沿且即将跳变为低电平时开始发送右声道数据，在 BCLK 信号下降沿时完成缓存将并行信号转换为串行信号的过程。

信号名称	方向	说明
clk	in	时钟输入
rst	in	异步复位输入，高复位
sck_bclk	in	数字音频接口 bit 时钟
ws_lrc	in	ADC 采样时钟
sdata	out	音频数字接口串行数据输入
left_data	in	左声道数据
right_data	in	右声道数据
read_data_en	out	音频数据读取，提前一个采样周期读取

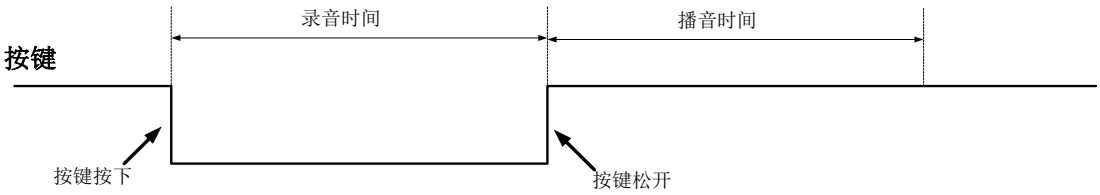
audio_tx 音频发送模块端口

audio_key 模块主要完成录音播放的按键控制，状态转换如下，当按键按下后进入录音状态，当按键松开时进入播放状态。



audio_key 模块状态转换图

程序检测到按键的下降沿的时候，说明按键按下，程序发送录音请求，并开始计时。当程序检测到按键的上升沿的时候，说明按键松开，开始进入播音状态，播放的时间长度等于录音的时间长度。

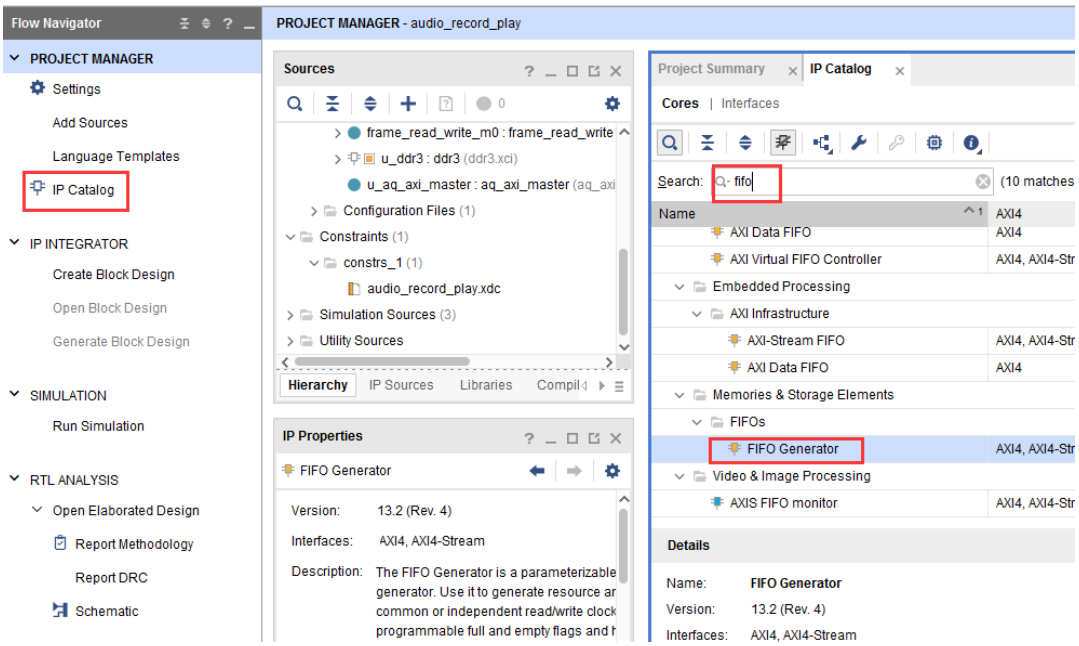


信号名称	方向	说明
clk	in	时钟输入
rst	in	异步复位输入，高复位
key	in	按键输入
record	out	录音状态指示
play	out	播放状态指示
write_req	out	写数据开始
write_req_ack	in	写数据应答
read_req	out	读数据开始
read_req_ack	in	读数据应答

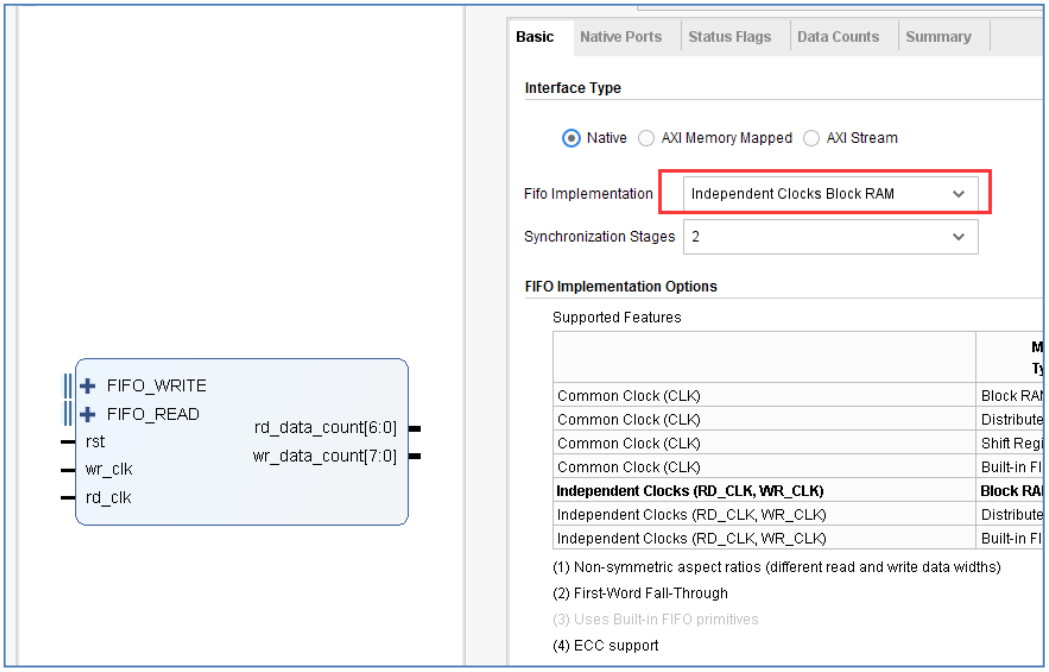
- **FIFO 读写控制**

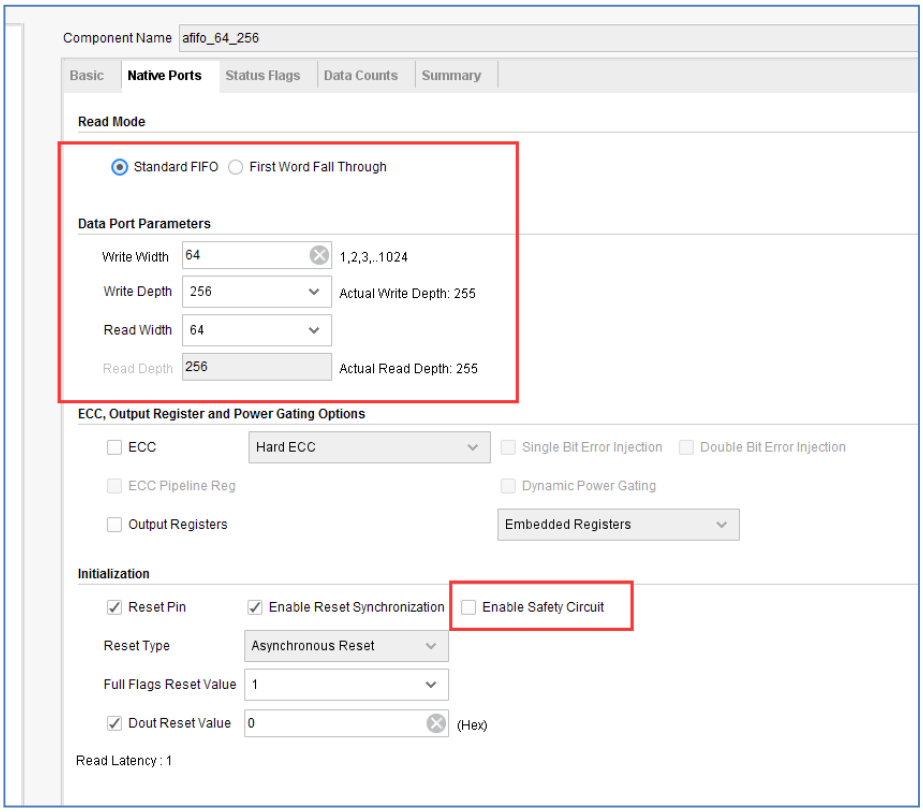
在模块‘frame_read_write’中使用到了 FIFO 的 IP core，通过两个 FIFO 分别作为 DDR3 控制器的读写接口，避免复杂的 DDR3 时序。因为时钟速率的不同，所使用的是异步时钟 FIFO。以 AX7035 的 write_buf 写入模块的 FIFO 为例，在 vivado 中加入 FIFO IP core 的方法和设置如下：

1. 在按下图方式找开 IP Catalog 并双击左键打开 FIFO Generator；



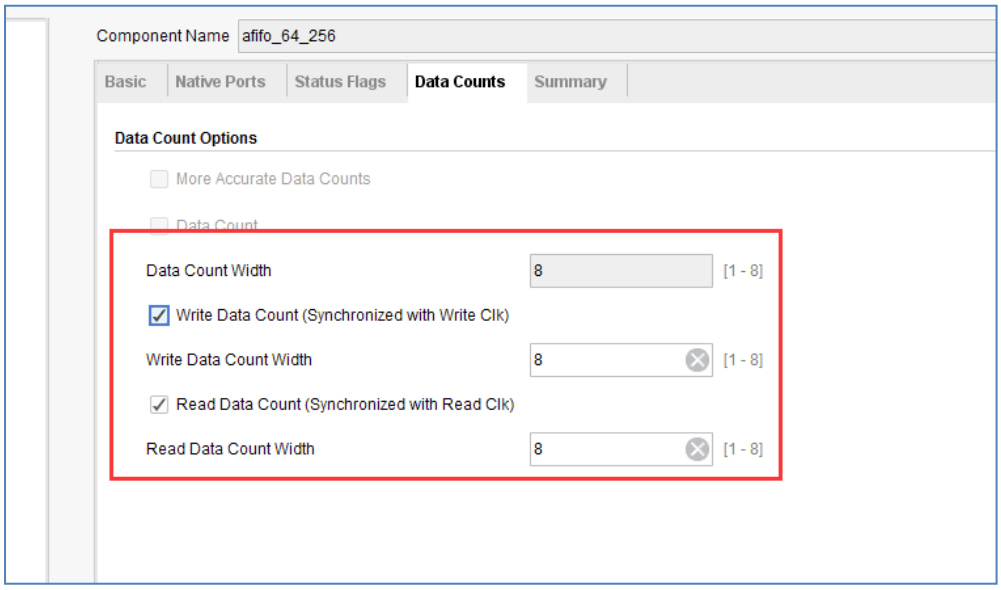
2. 在 Basic、Native Ports 栏中进行如下设置：



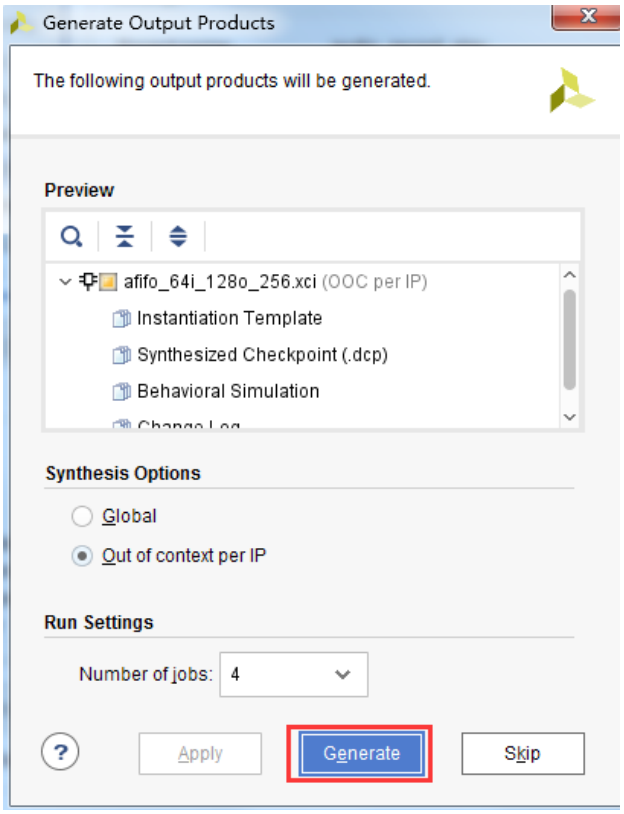


3. Status Flags 、Summary 栏保持默认设置;

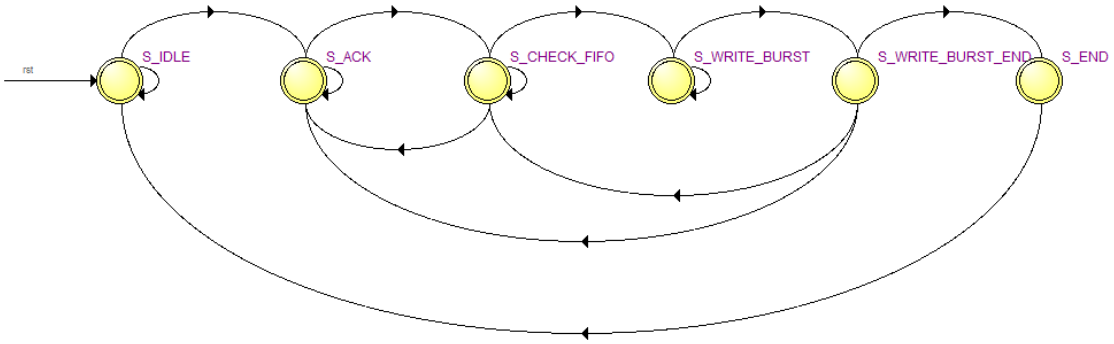
4. Data Counts 进行下图设置即可。



5. 直接点击下图 Generate 结束整个 FIFO IP core 创建过程。接着在模块中直接例化 FIFO 的端口名就可以使用 FIFO 了，在 read_buf 中也是同样的操作步骤，只是我们设置的是异步 FIFO 需要将参数更改一下就好。具体的参数设置和端口例化的信号连接参考例程



frame_fifo_write 模块完成 FIFO 数据到外部存储器的写入，因为 FIFO 接口是异步时钟 FIFO，可以完成写数据的跨时钟域转换。状态机转换图如下，收到写数据请求后进入应答状态“S_ACK”，如果写请求撤销，则进入检测 FIFO 空间大小状态“S_CHECK_FIFO”，检查 FIFO 内数据是否够一次突发写，如果有足够多的数据，进入突发写存储器状态“S_WRITE_BURST”,突发写完成后进入“S_WRITE_BURST_END”状态。



frame_fifo_write 模块状态机

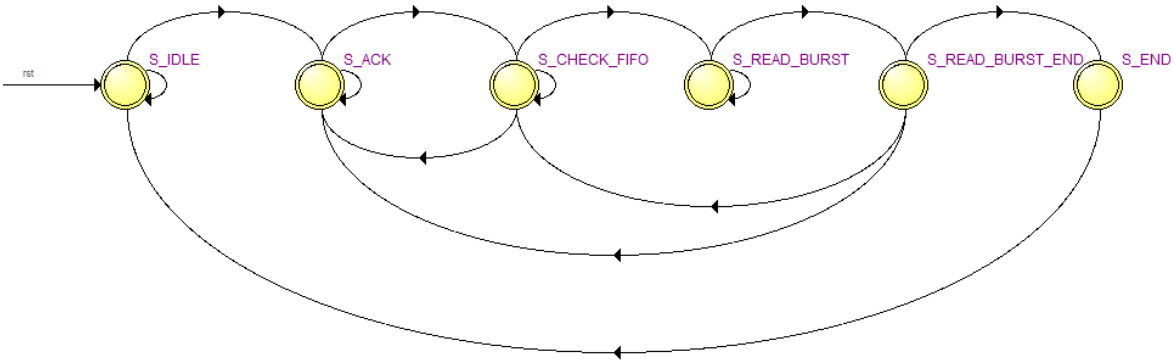
信号名称	方向	说明
mem_clk	in	外部存储器用户时钟输入

rst	in	异步复位输入，高复位
wr_burst_req	out	对接存储器控制器，写请求
wr_burst_len	out	对接存储器控制器，写请求长度
wr_burst_addr	out	对接存储器控制器，写请求基地址
wr_burst_data_req	in	对接存储器控制器，写请求数据索取，提前一个时钟周期发出，用于连接 FIFO 的读数据
wr_burst_finish	in	对接存储器控制器，写请求完整
write_req	in	一帧（大量数据）写开始，收到应答后必须撤销请求，新的请求会中断正在进行的请求
write_req_ack	out	一帧（大量数据）写应答
write_finish	out	一帧（大量数据）完成
write_addr_0	in	一帧（大量数据）写基地址 0
write_addr_1	in	一帧（大量数据）写基地址 1
write_addr_2	in	一帧（大量数据）写基地址 2
write_addr_3	in	一帧（大量数据）写基地址 3
write_addr_index	in	一帧（大量数据）写基地址选择， 0：write_addr_0 1：write_addr_1 2：write_addr_2 3：write_addr_3
write_len	in	一帧（大量数据）写长度
fifo_aclr	out	在收到写请求后，模块会清空 FIFO
rdusedw	in	FIFO 读端的数据使用量

frame_fifo_write 模块端口

frame_fifo_read 读模块完成从外部存储器读取数据，然后写到 FIFO，使用时钟异步 FIFO 可以完成数据从存储器时钟域到其他时钟域的转换。状态机转换图如下图所示，收到读请求以后进入应答状态“S_ACK”，等待读请求撤销后应答，进入 FIFO 深度检测状态“S_CHECK_FIFO”,如果

FIFO 空间足够一次突发读，进入突发读状态“S_READ_BURST”，突发读结束后进入“S_READ_BURST_END”。



frame_fifo_read 模块状态机

信号名称	方向	说明
mem_clk	in	外部存储器用户时钟输入
rst	in	异步复位输入，高复位
rd_burst_req	out	对接存储器控制器，读请求
rd_burst_len	out	对接存储器控制器，读请求长度
rd_burst_addr	out	对接存储器控制器，读请求基地址
rd_burst_data_valid	in	对接存储器控制器，读请求数据有效
rd_burst_finish	in	对接存储器控制器，读请求完全
read_req	in	一帧数据读开始
read_req_ack	out	一帧数据读应答
read_finish	out	一帧数据读完成
read_addr_0	in	一帧数据读基地址 0
read_addr_1	in	一帧数据读基地址 1
read_addr_2	in	一帧数据读基地址 2
read_addr_3	in	一帧数据读基地址 3
read_addr_index	in	一帧数据读基地址选择 0：read_addr_0

		1：read_addr_1 2：read_addr_2 3：read_addr_3
read_len	in	帧数据读长度
fifo_aclr	out	外部 FIFO 异步复位
wrusedw	in	FIFO 写端使用空间大小

frame_fifo_read 模块端口

frame_read_write 模块完成音频帧数据 FIFO 读写的封装，这里使用了异步 FIFO 来解决跨时钟问题，例如 FIFO 输入宽度和输出宽度的不同来完成数据位宽的转换。

- **AXI 总线转换**

总线转换程序 aq_axi_master 为同 open source 下载的第三方的 axi master 读写控制程序，它把 burst 读写请求的信号转换成 axi 总线请求，这里我们使用的 MIG IP 需要使能 AXI 总线接口。这里通过 aq_axi_master 程序和 MIG IP 的 axi 总线接口连接，这样用户可以直接通过 axi 总线来访问 DDR3 的存储空间。

4 实验现象

AX7101 开发板首先将音频模块插入扩展口 J11，AX7102 开发板首先将音频模块插入扩展口 J5，AX7103 开发板首先将音频模块插入扩展口 J13，同时插入麦克风和耳机，然后下载实验程序，按下 KEY2 不放，进行录音，松开按键后可通过耳机回放录音。