

第六届

全国大学生集成电路创新创业大赛

报告类型*: 系统设计方案

参赛杯赛*: 安谋科技杯

作品名称*: 智能娱乐收音机

队伍编号*: CICC2662

团队名称*: 夜的第七章

目录

1 项目背景	4
2 系统功能介绍	5
2.1 Cortex_M0 片上系统的基础设计	5
2.2 FM 电台	5
2.3 自动搜索、存储和换台功能	6
2.4 发挥部分	6
2.4.1 音质改善	7
2.4.2 SD 卡播放音乐	7
2.4.3 FM 对讲机	7
2.4.4 录音存储	7
2.4.5 电源管理模块	7
2.4.6 K 歌模式	7
2.4.7 FM 游戏	7
2.4.8 机械臂远程操控	8
2.4.9 语音识别	8
2.4.10 立体声传输	8
3 系统框架图	9
3.1 系统结构	9
3.2 地址映射	9
4 技术路线	11
4.1 FM 电台	11
4.1.1 SPI 通信	11
4.1.2 基带信号处理 IP	11
4.1.3 PWM_DAC 模块	13
4.2 自动搜台	14
4.2.1 自动搜台原理	14
4.2.2 自动搜台接口信息	14
4.2.3 仿真验证	15
4.2.4 资源消耗	16
4.3 存储与切换频道	16

4.4 音质提升.....	17
4.4.1 软件提升.....	17
4.4.2 硬件提升.....	19
4.5 SD 卡音乐播放.....	21
4.5.1 需求分析.....	21
4.5.2 音乐播放模块接口信息	21
4.5.3 实现原理.....	22
4.5.4 仿真验证.....	23
4.5.5 资源消耗.....	24
4.5.6 内核交互.....	24
4.6 电源管理模块.....	25
4.6.1 原理分析.....	25
4.6.2 实物测试.....	25
4.7 对讲机	26
4.7.1 IIC 通信.....	26
4.7.2 实现复用.....	26
4.8 录音存储.....	27
4.8.1 需求分析.....	27
4.8.2 录音存储模块接口信息	27
4.8.3 实现原理.....	27
4.8.4 仿真验证.....	28
4.8.5 仿真验证.....	29
4.8.6 内核交互.....	29
4.9 K 歌模式.....	30
4.9.1 需求分析.....	30
4.9.2 音乐播放模块接口信息	30
4.9.3 实现原理.....	31
4.9.4 仿真验证.....	32
4.9.5 资源消耗.....	33
4.9.6 内核交互.....	33
4.10 FM 游戏.....	34
4.10.1 设计原理	34

4.10.2 硬件设计	35
4.10.3 串口屏设计	37
4.11 机械臂远程操控	40
4.11.1 控制原理	40
4.11.2 通信方式	42
4.12 语音识别	42
4.12.1 需求分析	42
4.12.2 语音识别模块	42
4.12.3 通信格式	43
4.13 立体声传输	43
4.13.1 立体声广播原理	43
4.13.2 立体声数字解调方案	44
4.13.3 导频提取方案	48
4.13.4 数字解调方案	50

1 项目背景

在科技发展迅速的现在，不断有新电子产品迭代更新。如今的人们每天手机不离手，使用最多 APP 是微信，头条，支付宝，即使想听广播电台，人们也可以方便的在手机上利用网络收音机收听到各地的电台，这使得传统收音机慢慢淡出人们的日常生活的视野。如何在收音机的基础上增添拓展功能，让传统收音机再次焕发生机是本项目主要考虑的。

本项目设计了一种基于 FPGA 的片上收音机，并依托于此 SOC 设计了区别于传统收音机更为娱乐化、多元化、实用化的智能收音机。在完成基础的收音机功能下，通过硬件加速搜台和设计滤波，将收音机达到远超出市面收音机的音质和搜台速度。

人们收听电台通常是为了放松身心，为了让收音机进一步娱乐化，我们增添了诸如打兔子、碰手指等游戏，利用 FM 通信完成多人之间的游戏对战，让人们找回童年里快乐的身影。在游戏的基础上，我们进一步利用 FM 通信的原理实现更为实用的远程控制机械臂操作，此举将主要代替人工从事场景危险的工作或者是顶替密集型、重复性高的动作，在制造业、医学远程控制手术上都会有着极大实用价值。

最后，我们根据软件无线电理论知识，创造出立体声解调，设计专用的 FIR 滤波器、乘法器、Cordic 取模运算等，实现了对立体声副载波的解调(AM 解调)，完成了立体声传输，展示效果良好，实践创新性强，相较于传统相干解调方式节省大量资源。

2 系统功能介绍

2.1 Cortex_M0 片上系统的基础设计

基于 ARM DesignStart 计划开放的 Eval 版 Cortex-M0 搭建片上系统，下载程序后，可在 Keil 界面中完成调试及运行。使用内部总线连接存储器与 FPGA 板载的外设，如数码管、矩阵键盘、拨动开关、LED、蜂鸣器、FLASH 等。

具体驱动方式如下：

- 数码管：M0 通过写总线实现对数码管动态显示数据的更改。
- 矩阵键盘：M0 通过读总线实现对 16 个按键的扫描检测。
- 拨动开关：M0 通过外部中断实现对开关上拨动作的检测。
- LED：M0 通过写总线实现对 LED 流水模式的控制。
- 蜂鸣器：M0 通过写总线实现对蜂鸣器音乐播放的控制。
- FLASH：M0 通过读写总线模拟 SPI 时序，完成对 FLASH 的读写控制。

2.2 FM 电台

在片上系统中，完成了 SPI 控制模块、FM 解调模块、PWM 型 DAC 模块的设计，并成功驱动喇叭，实现 FM 电台的播放。同时，可利用按键手动调频，步进为 0.1MHz，并将当前电台频率实时显示在数码管上。

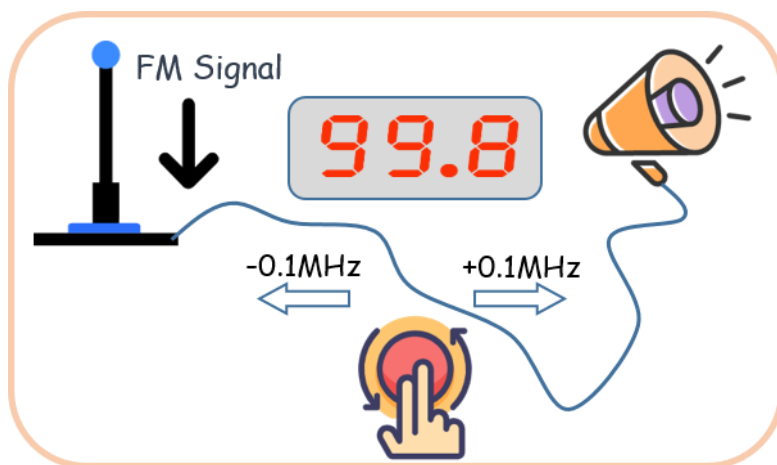


图 2.1 FM 电台示意图

2.3 自动搜索、存储和换台功能

在 FM 电台的基础上，由按键触发，自动搜索当前区域电台。扫描完成，将搜索到的电台显示在串口屏上。接收情况良好，在实验环境中，可接收到的电台个数远大于 5。

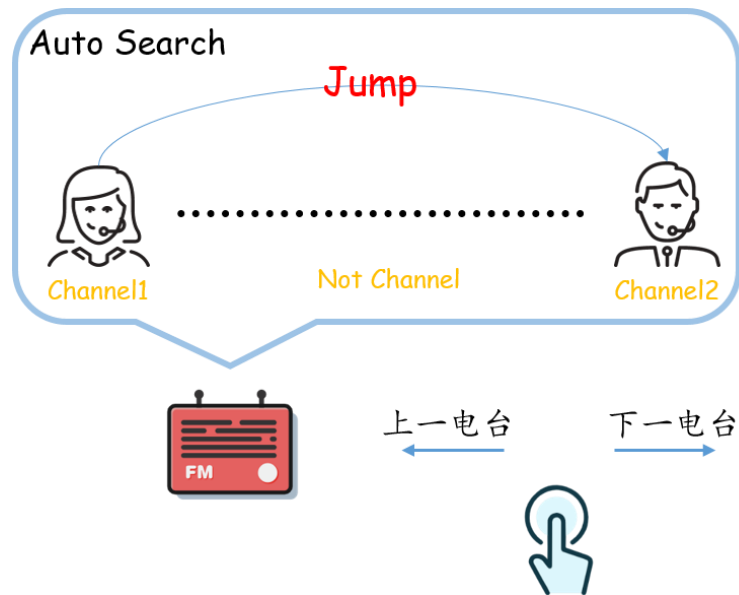


图 2.2 自动搜台示意图

搜索完成后，拨动开关，即可将当前的电台频点存储到 FLASH 中。对多个电台依次存储后，通过按键即可对存储的电台进行选择与切换。

增添串口屏显示模块，展示出美观便捷的人机交互功能。

2.4 发挥部分

发挥部分为五个部分，分别为音质改善、K 歌模式（由 SD 卡播放和录音存储组成）、FM 游戏、机械臂远程操控和立体声传输。其示意图如 2.3 所示。



图 2.3 提升部分功能

2.4.1 音质改善

软件上，设计 16 阶 FIR 带通滤波器，实现对语音信号的处理；硬件上，为 FM 射频板配备了性能更加优良的射频天线与大音腔喇叭，使音质得到了提升。综合处理后，喇叭播放的 FM 电台音频声音洪亮，清晰无噪声。

2.4.2 SD 卡播放音乐

为满足个性化需求，可先行在 SD 卡中存储自己喜欢的信息和歌曲。通过读取 SD 卡内的格式化音频数据，处理后由喇叭播放，即可得到一个便携式音箱。音乐清晰洪亮，无失真，且可由按键执行歌曲的切换与暂停。

2.4.3 FM 对讲机

将 FM 射频板完全开发，设计 IIC 控制模块，实现对 QN8027 发射芯片的控制。在当前区域内，选择两个非电台频点，避开电台信号干扰，即可实现 FM 全双工对讲机。经实验测试，对麦克风发声或播放音乐，经调制发射、接收解调后，均可在喇叭端得到良好接收。

2.4.4 录音存储

利用 FPGA 芯片内合封的 SDRAM 资源，由拨动开关触发，实现对电台音频数据的存储及输出。受 SDRAM 容量的限制，最大录制时长为 40s。录制完成后，人为拨动开关，即可实现对先前录音的循环播放。

2.4.5 电源管理模块

本系统为片上收音设备，却需电脑 USB 接口为其供电运行，不能称得上完整、独立的系统。因此，为其增加单独的电源管理模块进行供电，让整个系统变得便携独立。

2.4.6 K 歌模式

将已实现的 SD 卡播放与录音存储功能合并，左右声道分别输出人声演唱及伴奏，使分立功能协调统一化，构成 K 歌模式。同时根据乐理知识，引入评分机制，构建了一个 KTV 式的歌唱平台。

2.4.7 FM 游戏

FM 对讲机受制于语音信号的微弱，效果不够显著。因此我们在此基础上，制定通信协议，以调制信号的频率、占空比为信息，进行调制传输。最后，在串口屏上设计游戏 UI 界面，实现 FM 双人对战游戏，充分发挥了 FM 通信的潜力。

2.4.8 机械臂远程操控

进一步围绕 FM 通信开展实际应用，利用 FM 传输机械臂角度信息，配合自制的体感手套，实现了对机械臂的实时远程控制。最终，可实现远程操纵机械臂夹取物品并搬移，具有广阔的应用前景，充分证明了 FM 通信的传输能力。

2.4.9 语音识别

采用芯片 CI1122，通过训练完成对特定语句的离线识别功能。设备通过串口与 CI1122 进行通信，实现了语音调频功能和语音存储功能。语音识别功能简化了操作步骤，使设备更加智能化、产品化。

2.4.10 立体声传输

作为专用的 FM 发射芯片，QN8027 支持立体声传输，而 MSi001 仅支持 FM 基带解调。在充分学习通信理论的基础上，我们采用立体声数字解调的方案，设计专用的 FIR 滤波器、乘法器、Cordic 取模运算等，实现了对立体声副载波的解调（AM 解调），完成了立体声传输，展示效果良好，实践创新性强。

3 系统框架图

3.1 系统结构

在方案设计时，核心思想为软硬件协同，充分利用开发板资源完成整个系统的搭建，结构如图 3.1 所示。

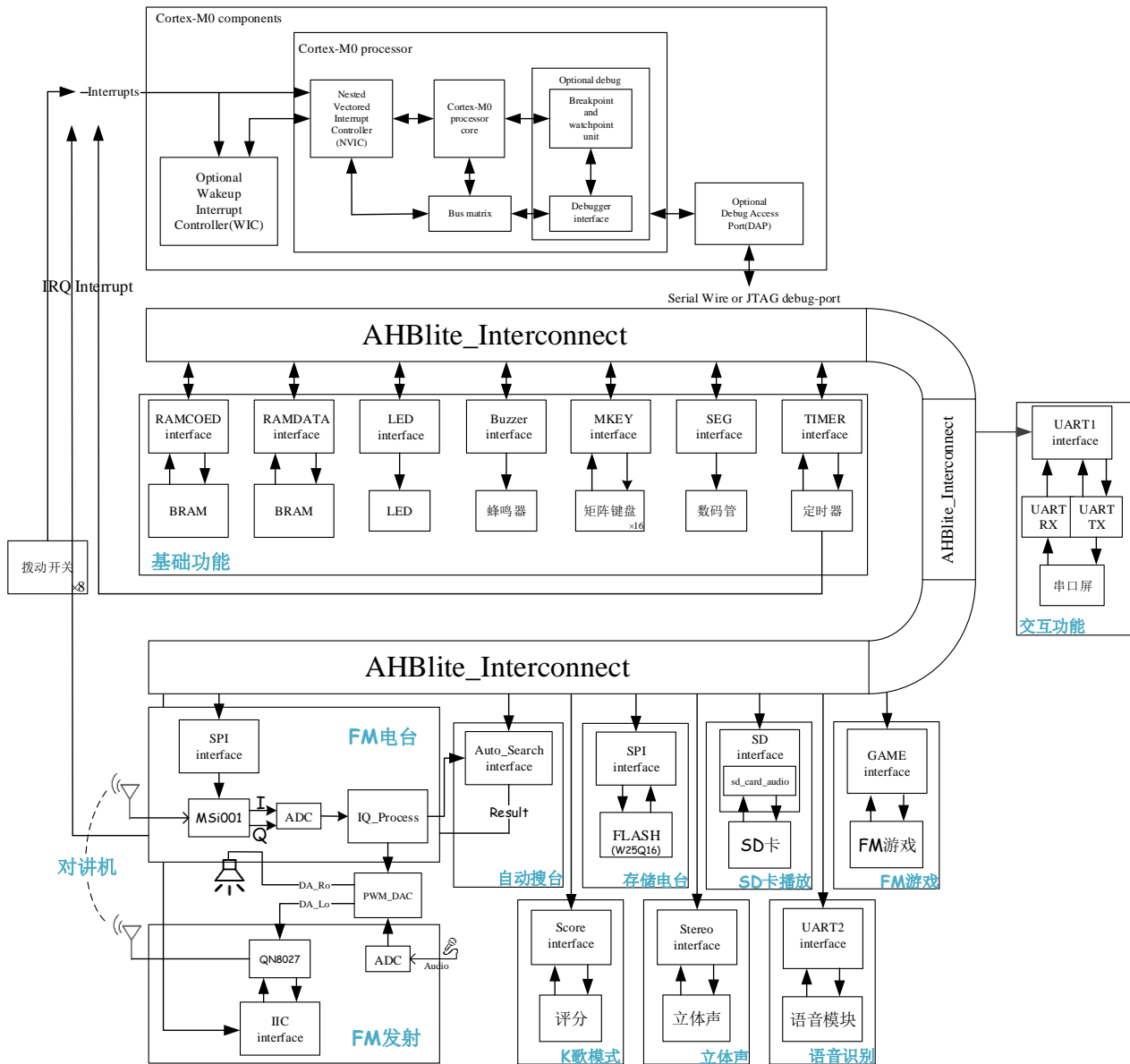


图 3.1 系统结构框图

3.2 地址映射

Cortex-M0 提供了 4GB 的存储器寻址空间，也规定了这些区域的使用方法，程序设计人员可以根据具体要求灵活定义存储器映射空间，本次设计的 Memory Map 如图 3.2 所示。

MEMORY MAP	
0xFFFFFFFF	
0xE0100000	
0xE0000000	PPB
0x40100000	UART2
0x400F0000	Stereo
0x400E0000	PWM
0x400D0000	Score
0x400C0000	GAME1
0x400B0000	SD_card
0x400A0000	Buzzer
0x40090000	Speaker
0x40080000	W25Q16_SPI
0x40070000	Auto_Search
0x40060000	QN8027_IIC
0x40050000	MSi001_SPI
0x40040000	TIMER
0x40030000	SEG
0x40020000	Matrix_Key
0x40010000	UART
0x40000000	LED
0x20000000	RAMDATA 16K
0x00000000	RAMCODE 16K

图 3.2 地址映射图

4 技术路线

本节将详细论述系统功能的设计原理，软硬件功能的划分。根据上文的结构图可知，整个片上系统通过 AHBLite 总线，实现 M0 对各外设模块的控制。受篇幅限制，在此仅对关键的模块，围绕软硬件协同的设计思想，进行展开说明。

4.1 FM 电台

4.1.1 SPI 通信

对于通信时序的实现，有软件与硬件两种方式。硬件即在 FPGA 内编写通信模块，在状态机的控制下逐步实现通讯；而软件则将 IO 挂载总线上，由 M0 模拟出基本时序，层层调用实现复杂的读写过程。

本次采用软件实现的方式，因此硬件上只需将 SPI 接口挂置总线，供 M0 操控。而软件上，先遵循 SPI 时序设计出寄存器写入函数，再结合芯片手册，反复调用此函数，即可完成对 MSi001 芯片的初始化配置。

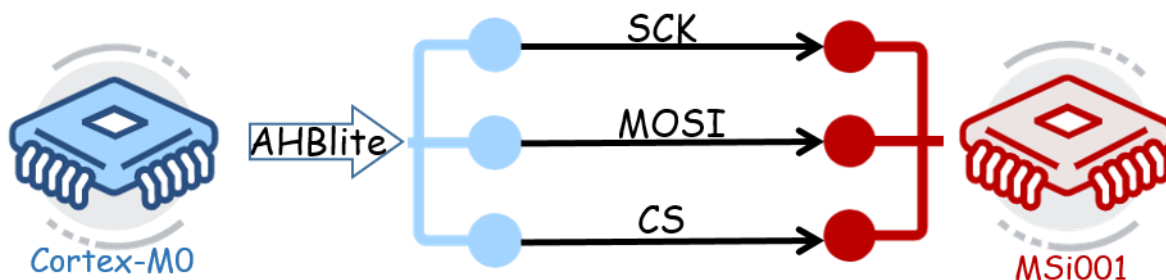


图 4.1 SPI 通信

4.1.2 基带信号处理 IP

基带信号处理 IP 是整个 FM 电台的核心模块，采用零中频的方案，从而降低 ADC 的采样率，以下分三方面论述。

(1) IQ 信号接收

根据 FPGA 板自带的 ADC，位宽为 12bit，采样率选择 1MSPS，双通道轮询串行采样。为得到最大采样速率，借助 TD 软件自带的 IP 核，使用 PLL 模块生成 16MHz 的 ADC 驱动时钟。再者，通过在 Chip Watcher 中观察相关信号关系，由 Channel 变量指示当前的采样结果——为 I 数据亦或 Q 数据。

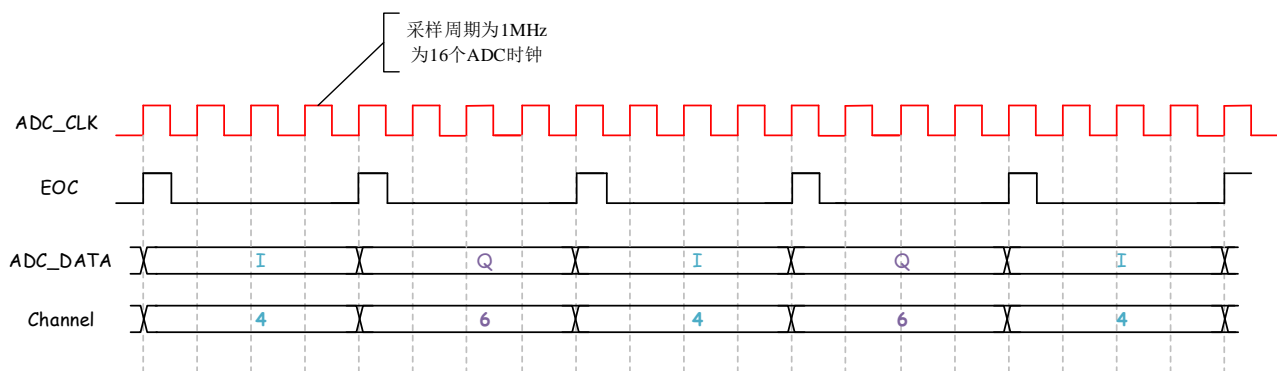


图 4.2 IQ 接收时序

(2) 解调算法

由 FM 的数字解调公式可知，整个解调算法仅是一个由乘法和减法组成的迭代系统。

$$f(n) = X_I(n-1)X_Q(n) - X_I(n)X_Q(n-1) \quad (1)$$

因此，在接收到更新的 IQ 数据后，对其进行打一拍寄存，即得到上一时刻的 IQ 数据，解调算法的时序图如 4.3 所示。

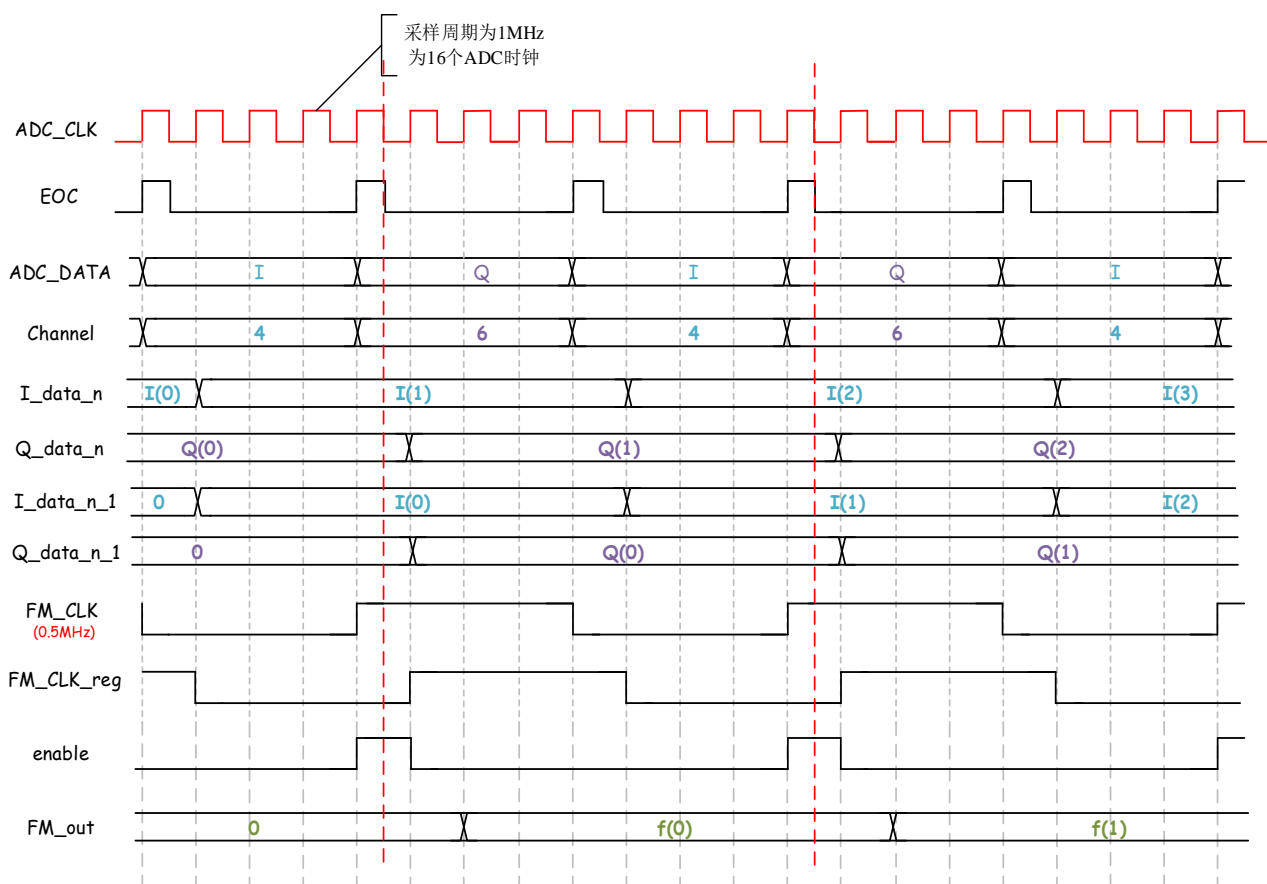


图 4.3 解调时序

(3) 符号转换

在进行解调运算时，由于涉及减法，为防止计算溢出，参与运算的数据应为有符号数。而 ADC 采样数据为无符号，因此需要进行无符号向有符号的转换。同理，在解调得到音频数据后，需要送入 PWM_DAC 模块，因此再将结果的有符号数转换为无符号数。最后，四则运算会增大数据范围，所以应对解调的音频数据结果进行数据位宽拓展。



图 4.4 符号转换

4.1.3 PWM_DAC 模块

与 STM32 定时器的 PWM 输出模式类似，即实现占空比可调的 PWM 输出。核心思想就是计数值与参考值循环比较，从而产生占空比与参考值成正比的 PWM 波。因数据位宽为 12 位，循环计数值为 4095。

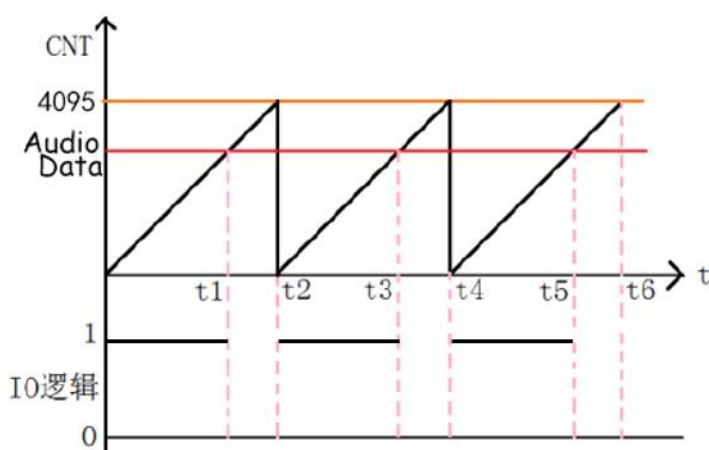


图 4.5 PWM 产生示意图

我们采用的计数时钟为 200MHz，仍是借助 IP 核，利用 PLL 倍频实现。选择 200MHz 出于以下考虑：由公式 (2)，FM 射频板上的二阶低通 RC 滤波器的截止频率为 31.2KHz。（根据 FM 射频板电路得知， $R = 510\Omega$ ， $C = 0.01\mu F$ ）

$$f = 1/2\pi RC \approx 31.2KHz \quad (2)$$

在采用 200MHz 的前提下，由公式（3），对 PWM 进行傅里叶级数展开，可以由此计算得到 PWM 的一次谐波分量为 48.8KHz，大于截止频率，受到抑制。且此频率在人耳辨别范围之外，保证了 PWM 的高阶谐波分量不会对直流分量产生干扰，实现了 PWM 向 DAC 电平的转化，从而精准启动喇叭。

$$f(t) = \frac{n}{N} \times 3.3 + 2 \frac{3.3}{\pi} \sin\left(\frac{n}{N} \pi\right) \cos\left(\frac{2\pi}{NT} t - \frac{n\pi}{N} k\right) + \sum_{k=2}^{\infty} 2 \frac{3.3}{k\pi} \left| \sin\left(\frac{n\pi}{N} k\right) \right| \cos\left(\frac{2\pi}{NT} kt - \frac{n\pi}{N} k\right) \quad (3)$$

$$f_1 = \frac{1}{NT} = \frac{200}{4096} M \approx 48.8KHz \text{ (一次谐波分量)}$$

4.2 自动搜台

4.2.1 自动搜台原理

自动搜台一般有两种方式。

第一种是计算当前频率下的中频输出值，在得到特定值即表示此处有一个信号，结合当前的信号强度及信噪比判定是否要停下来收这个台；第二种是计算当前接收频率的信号大小及信噪比，大于指定的值会被认为收到了一个电台。

本系统运用第二种方式，对 IQ 解调后的语音信号进行强度分析，数值大小对应的幅值进行统计，设定范围寻找出存在电台的频率，实现自动搜台。

4.2.2 自动搜台接口信息

自动搜台的接口信号，方向与功能如下所示，包含进行平均的 Mean_filter 模块和对电台判断的范围常数 parameter range:

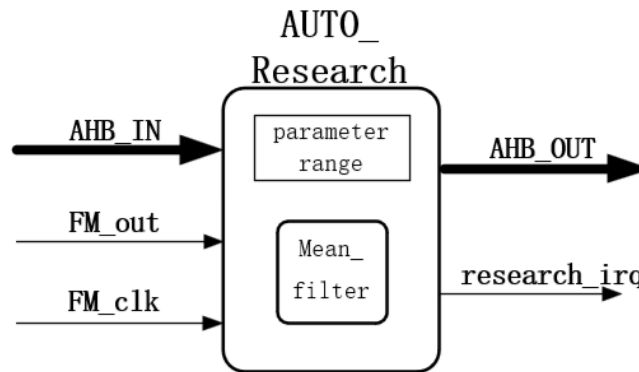


图 4.6 自动搜台信号流图

表 4.1 信号详细功能

信号	方向	功能说明
AHB_IN	In	AHB 总线的一系列输入信号
FM_out	In	音频数据
FM_clk	In	音频数据时钟 0.5MHz
AHB_OUT	Out	AHB 总线的一系列输出信号
research_irq	Out	搜到台，中断标志位

表 4.2 parameter 参数

参数	功能说明
NOISE_BELOW	低于直接视为噪声
NOISE_COUNT_LOW	噪声范围下限
NOISE_COUNT_HIGH	噪声范围上限
NOISE_COUNT	计数统计超过视为噪声
SIGNAL_ABOVE	高于直接视为电台
SIGNAL_COUNT_LOW	电台信号范围下限
SIGNAL_COUNT_HIGH	电台信号范围上限
SIGNAL_COUNT	计数统计超过视为电台

4.2.3 仿真实验

FM 电台输出的语音信号为了后续进行统计判断，先通过 Mean_filter 模块进行 1000 次数值平均，使得信号中偶然信号对整体数据统计的影响。由于语音信号整体为正数，那么信号在 12bit=4096 的量化单位下，基准电平值应该为 2048。

图 4.7 为无电台时幅值，可见在没有电台的情况下，语音信号幅值稳定在 2048 左右，通过对数值的统计可以确定当前频率不存在电台。

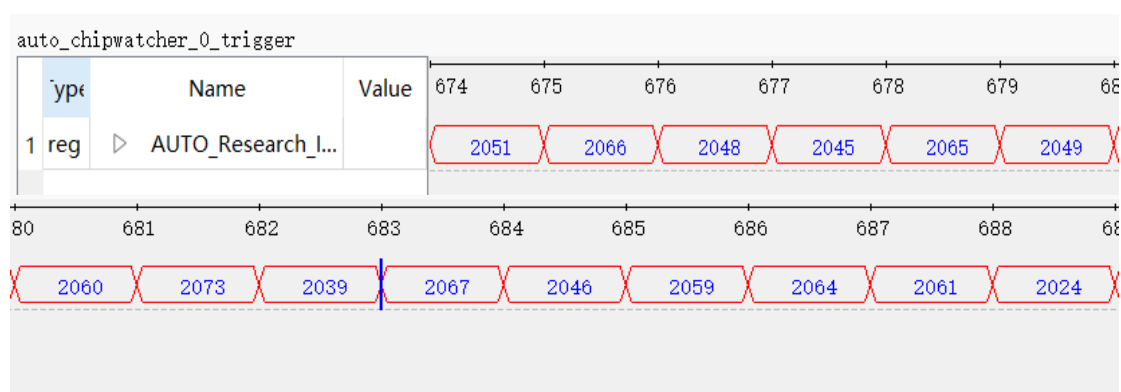


图 4.7 无台语音信号强度

图 4.8 为有电台时幅值，可见在有电台的情况下，语音信号幅值在 2048 上下大幅度跳动，通过对电台数据统计判断，可以确定当前存在电台。

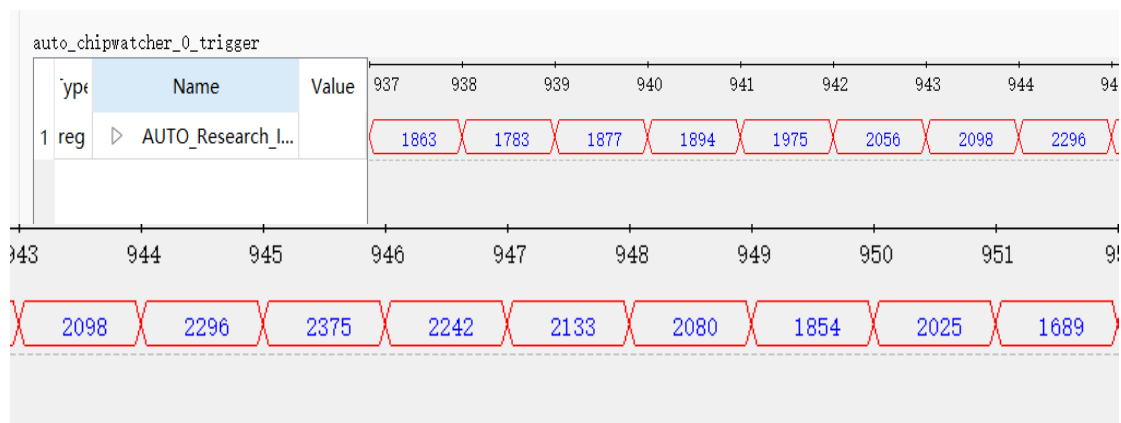


图 4.8 有台语音信号强度

图 4.9 为有电台前一 0.1Mhz 频率时幅值，可见在电台临近台，语音信号幅值在 2048 上大幅度跳动，可以很清晰的判断临近存在电台，可以用于对电台进行辅助判断作用。

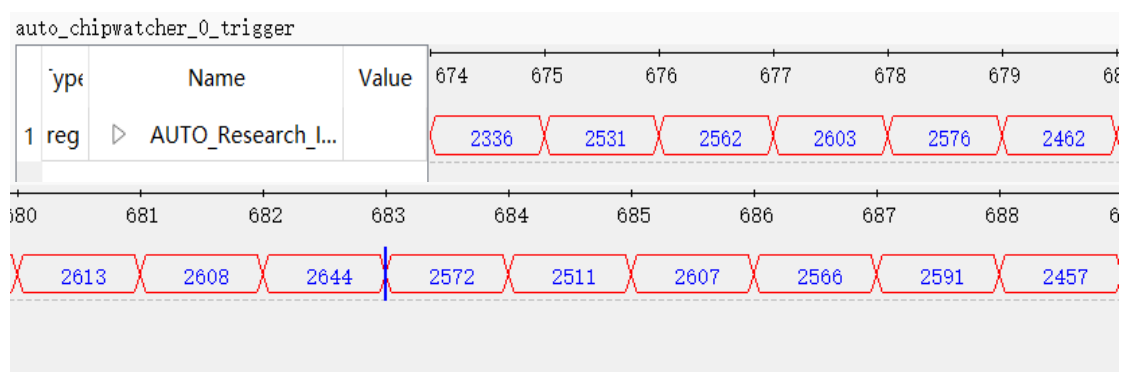


图 4.9 邻近台语音信号强度

4.2.4 资源消耗

由于整个模块采用对解调后信号幅值的统计判断进行自动搜台，并不分析 IQ 信号的功率大小，因此减少了乘法器和开发 IP 核而消耗的资源，故整个模块逻辑资源消耗很少，节约了板上资源实现了自动搜台功能。

表 4.3 资源消耗

资源	数值	占比
Lut	164	0.84%
Reg	71	0.36%

4.3 存储与切换频道

使用 FPGA 板载的 FLASH，型号 W25Q16，实现对电台频道的存储与读取。

设计的主要重点在于 SPI 通信，仍是采用软件的方案，基本思想与 MSi001 一致：先实现 SPI 基础的读写时序函数，再进而编写出 FLASH 的连续写字节和连续读字节函数，在此不再

赘述。

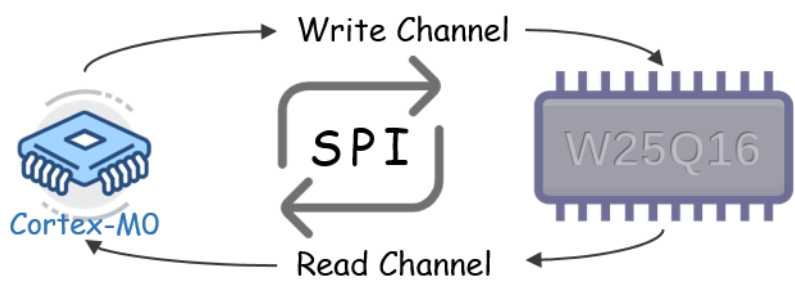


图 4.10 SPI 读写示意图

在对 FM 电台频点进行存储时，由于频点数据为浮点数据，需通过 Union 类型的内存共用性，再存储入 FLASH 中，读取时亦然。在此基础上，即可调用函数轻易地将当前自动搜索到的电台频点存入 FLASH 中，实现掉电存储，同时利用频点读取完成频道的切换。

4. 4 音质提升

4. 4. 1 软件提升

(1) 语音滤波器设计

在 FM 解调算法出来后的信号，仍然带有大量的 IQ 谐波成分，因此需要使用数字滤波器对 FM 解调后的信号进行滤波，才能得到更好的音质。

研究表明人耳能听见 20Hz 到 22000Hz 的声音频率，其中电台播放的声音频率为 300Hz 到 3400Hz，因此我们将设计阶数为 16 阶、截止频率为 4000Hz 的低通滤波器，滤除 IQ 谐波成分和非语音信号信息，让得到的音质得到提升。

(2) 滤波器接口信息

滤波器的接口信号，方向与功能如下所示：

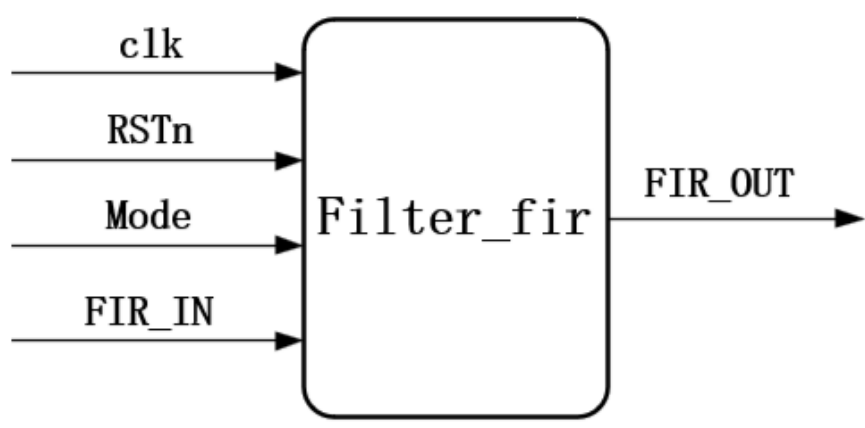


图 4.11 滤波器信号流图

表 4.4 信号详细功能

信号	方向	功能说明
clk	In	滤波器工作时钟，0.5MHz
RSTn	In	复位信号，高电平有效
Mode	In	收音/对讲模式转换
FIR_IN	In	未滤波信号输入
FIR_OUT	Out	滤波后信号输出

(3) 用户读写数据时序

Mode=0 时，clk=0.5Mhz，数据读写入于上升沿；Mode=1 时，clk=0.33Mhz，数据读写入于上升沿；FIR_OUT 与 FIR_IN 数据之间有 16 拍的延时。

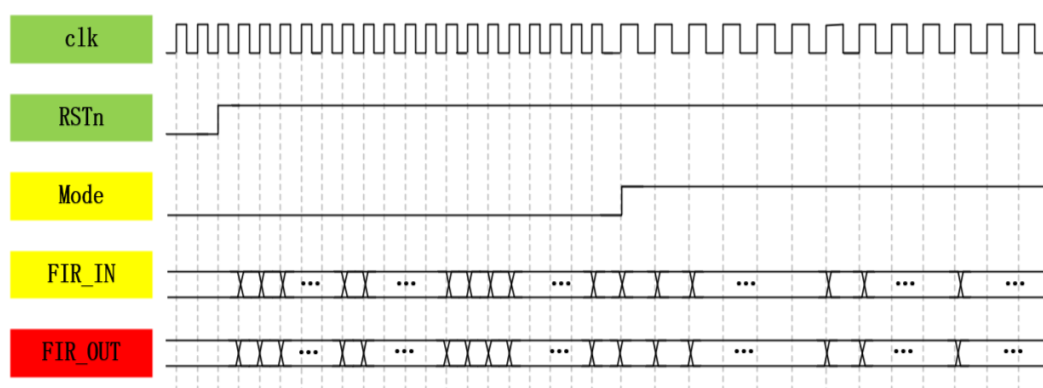


图 4.12 读写时序图

(4) 仿真验证

在 Matlab 中读取一段语音信号，并叠加一个频率为 10KHz 的正弦信号，设计 FIR 滤波器为 16 阶，截至频率为 4K。如图 4.13，可见 10KHz 高频信号被很好的滤除。

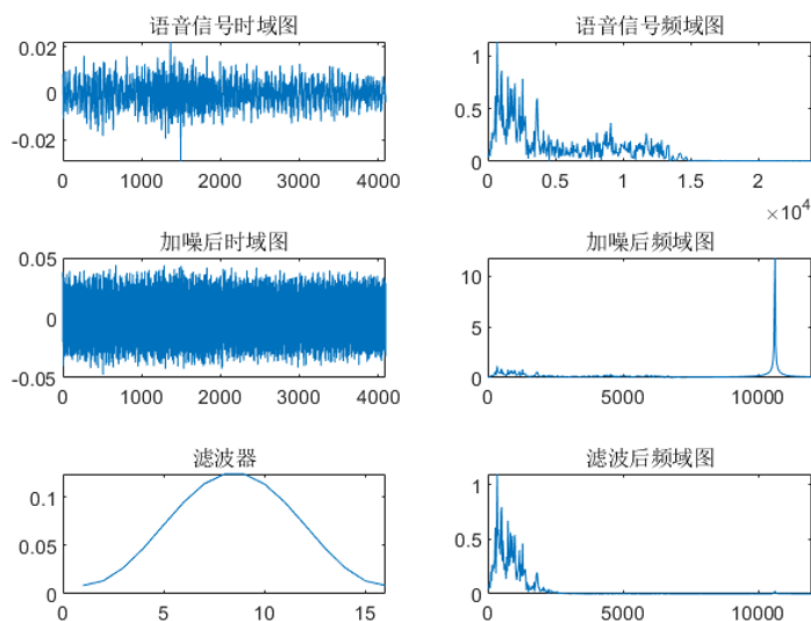


图 4.13 Matlab 滤波器仿真

将叠加后含 10KHz 噪音的信号生成到 txt 文本中，将信号传入到 Verilog 设计的 filter_fir 模块中，编写 Testbench 文件可得到信号如图 4.14。

其中 FIR_MUSIC 为原语音信号信息，FIR_IN 为加噪后信息，FIR_OUT 为滤波后信息，可见滤波器模块成功去除了高频噪声。

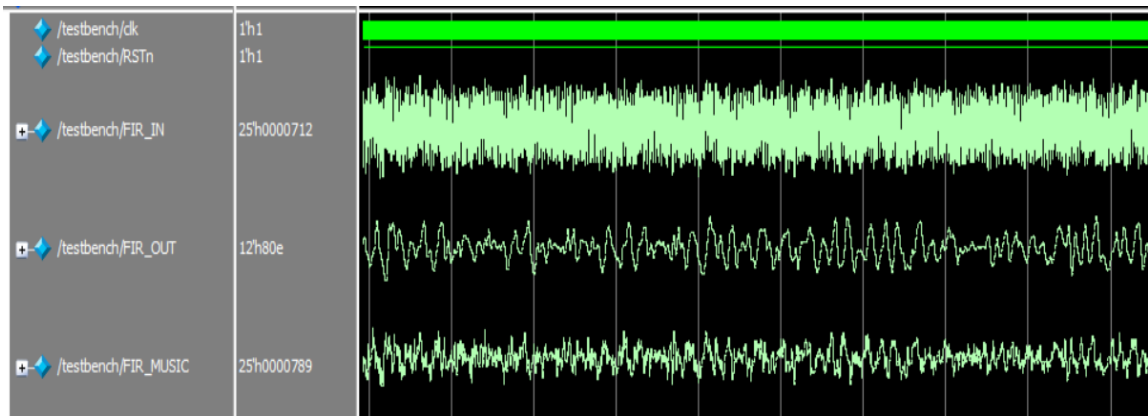


图 4.14 Modelsim 滤波器仿真

(5) 资源消耗

单独编译滤波器文件，主要的资源消耗如表 4.5，可以观察到 16 阶滤波器消耗了 68.97% 的 DSP 资源，实现了乘法器合理使用。在提升整体音质下让芯片更好的发挥了自身性能。

表 4.5 资源消耗

资源	数值	占比
Lut	575	2.93%
Reg	889	4.54%
Dsp	20	68.97%

4.4.2 硬件提升

(1) 天线选择

根据理论和实践证明，当天线的长度为无线电信号波长的 1/4 时，天线的发射和接收转换效率最高。因此，天线的长度将根据所发射和接收信号的频率即波长来决定，若 L 为天线长度，C 为光速，f 为接收频率，则有关系：

$$L = \frac{c}{4f} \quad (4)$$

根据 FM 的频率范围为 87-108MHz，计算得到 L 范围为 0.694m 到 0.862m 之间。选取长度适合的天线能收获更好的 IQ 信号，为音质的提示提供很大帮助。



图 4.15 可伸缩天线

(2) 喇叭选择

根据硬件电路，解调后的语音信号经过滤波后将进入 FM_SDR 模块上的喇叭功放模块，在不改变板上硬件的情况下，提升音质可以从外设的喇叭进行提高。

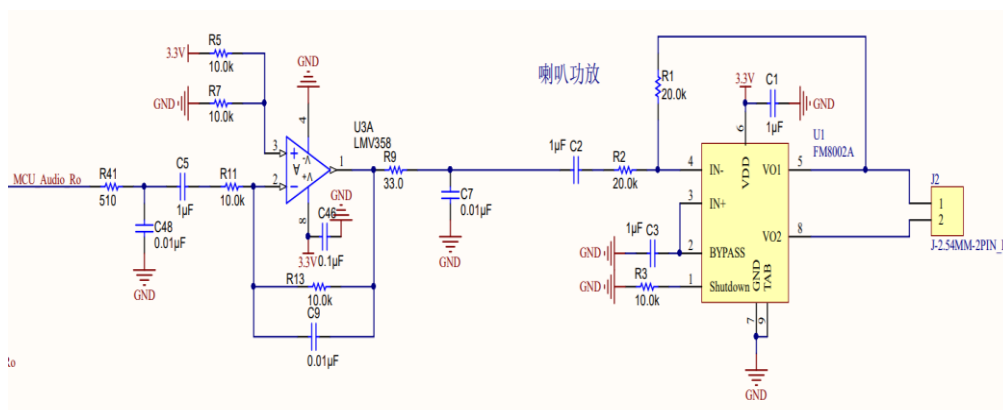


图 4.16 喇叭功放电路

选择的喇叭首先需要阻抗匹配，根据 FM8002A 参考手册，最大驱动功率为 $8\Omega/1.5w$ 和 $4\Omega/2w$ 。经过测试， $4\Omega/2w$ 含有音腔的喇叭发声效果最好，实现了在音质上的进一步提升。



图 4.17 含音腔喇叭

4.5 SD 卡音乐播放

4.5.1 需求分析

本设备已经通过 FM 解调模块进行了对 FM 广播的输出，通过一系列的信号处理后，我们已经得到便捷的收音功能和优良的音质。

那么除了接收 FM 广播信号，我们也希望能够读取自己喜欢的信息和歌曲。由于整个系统已经有很好的音质，进行读取 SD 卡内音乐并播放，我们就能得到一个有着音质的音箱。

对喇叭在放 FM 广播和 SD 卡内音乐的切换，很好的对喇叭功放模块进行了复用，使得整个系统更加完整，资源更加充分使用。

4.5.2 音乐播放模块接口信息

音乐播放接口信号，方向与功能如下所示：

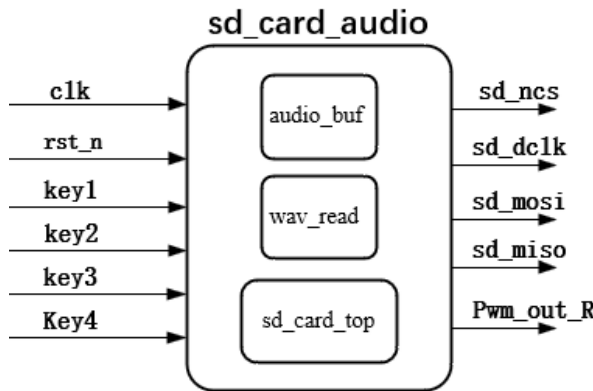


图 4.18 音乐播放信号流图

表 4.6 信号详细功能

信号	方向	功能说明
clk	In	音乐播放工作时钟，50MHz
rst_n	In	复位信号，高电平有效
key1	In	找音乐使能
Key2	In	下一首
Key3	In	上一首
Key4	In	播放暂停
sd_ncs	output	SPI 通信
sd_dclk	output	SPI 通信
sd_mosi	output	SPI 通信
sd_miso	output	SPI 通信
Pwm_out_R	output	音乐输出
state_code	output	显示当前工作状态，调试使用

4.5.3 实现原理

sd_card_top 模块通过 SPI 通信硬件初始化 SD 卡并进行读写，要想对 SD 内音乐进行播放，最好采用没有经过压缩的 wav 文件。wav 文件有着不同的采样率和量化单位和声道数，对应着不同的存放格式，选择 wav 文件量化单位的不同要考虑 FIFO 读取时候选择大端或小端读取。

由于音乐中各种乐器可能存在高频声音，取人耳最高听力 20KHz 频率范围，故 wav 的采样率根据奈奎斯特定理要高于 40KHz，为了音乐的失真度较低，时候采用 16bit 采样率进行 PCM 编码。

经过测试，选用 16bit，采样率为 48000 的 wav 文件能获得高音质，为了和系统形成整体，采用与 FM 同样的单声道输出。

PCM 数据类型	采样	采样
8 Bit 单声道	声道0	声道0
8 Bit 双声道	声道0	声道1
16 Bit 单声道	声道0低位, 声道0高位	声道0低位, 声道0高位
16 Bit 双声道	声道0低位, 声道0高位	声道1低位, 声道1高位

图 4.19 wav 声道存储格式

wav_read 模块用于检测 wav 文件的存在，WAV 文件作为多媒体中使用的声波文件格式之一。它是以 RIFF 格式为标准的，此标准定义每个 WAV 文件的头四个字节是“RIFF”，紧接着 4 个字节表示文件的大小，然后是“WAVE”标志，这样我们就可以确定音频文件位置要读取的数据量。WAV 文件头大小是 88 字节，在播放时 要把前 88 字节的文件头去掉。

如下图为 Winhex 读出的 wav 文件，以“RIFF”开头，DA0B2A=14,289,706 文件大小，而后为“WAVE”。

驱动器 G: 手写的从前-周杰伦_01.wav																	手写的从前-周杰伦_01.wav																
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII	G:\															
00000000	52	49	46	46	2A	0B	DA	00	57	41	56	45	66	6D	74	20	RIFF* U WAVEfmt	文件大小: 13.6 MB															
00000010	10	00	00	00	01	00	01	00	80	BB	00	00	80	BB	00	00	€» €»	14,289,714 字节															
00000020	01	00	08	00	64	61	74	61	8D	F5	D9	00	81	80	81	7F	data 0U €	逻辑大小: 14,289,714 字节															

图 4.20 Winhex 读 wav 文件

audio_buff 是一个 8 位输入、16 位输出、深度为 1024 的 FIFO，由于 wav 文件存储音乐为小端存储，此 FIFO 用于将 16bit 音频信息调整成正确的输出，并缓存 sd 卡一次读地址输出的 512 个字节数据。

4.5.4 仿真验证

用 chip_watcher 观察 sd_sec_read 读使能、sd_sec_read_address 读地址、read_data 读地址数据，state_code 观察当前状态。

初始化仿真如图 4.21，state_code=01 表示 sd 卡成功初始化。此时 sd_sec_read_address 处于 8192，为 FAT32 文件存储格式中 sd 卡物理首地址，没有按下 key1 使读音乐使能，故 sd_sec_read=0 且 read_data 没有读出数据。

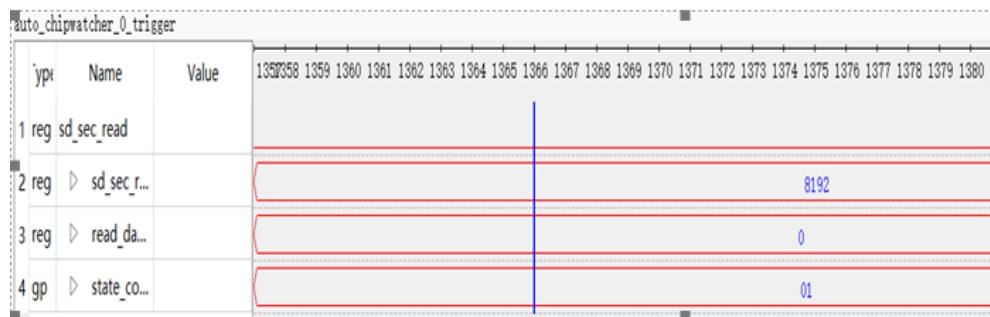


图 4.21 sd 卡初始化成功

搜音乐仿真如图 4.22，按下 key1 使读音乐使能。state_code=10 表示正在进行搜索音乐，此时 sd_sec_read_address 一直递增寻找 wav 文件头地址，sd_sec_read=1，read_data 读出数据并传入 wave_read 模块进行文件头寻找。

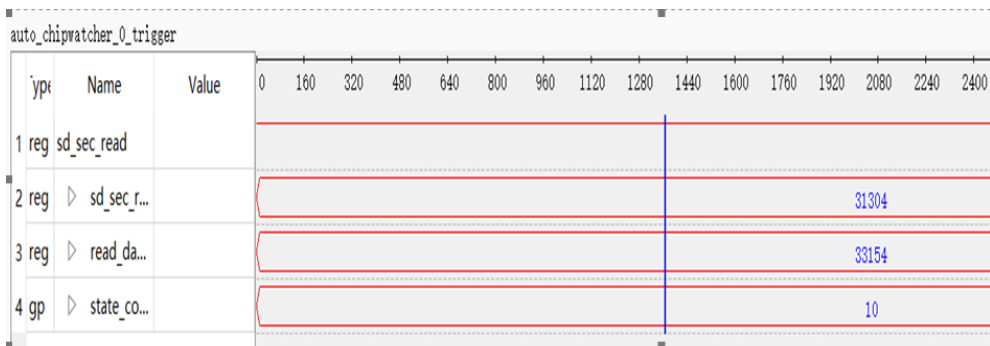


图 4.22 搜音乐使能

搜到音乐仿真如图 4.23，state_code=11 表示成功搜索音乐。此时写一次 sd_sec_read_address 地址，read_data 读出 512 字节的数据，读出数据并传入 PWM 模块进行音乐播放。

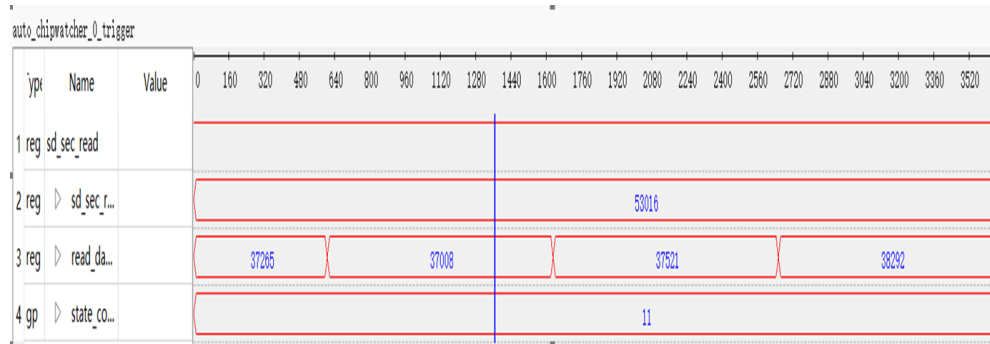


图 4.23 搜到音乐

暂停音乐仿真如图 4.24，此时写一次 sd_sec_read_address 地址停止递增，read_data 停止读数据，音乐暂停播放。

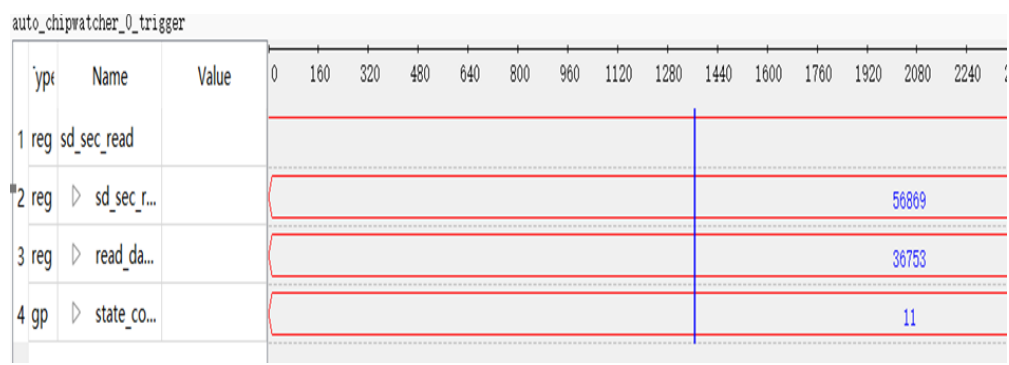


图 4.24 暂停播放

4.5.5 资源消耗

单独编译音乐播放模块文件，主要的资源消耗如表 4.7，可以观察到整个模块只占用了较少的资源就实现了音乐播放功能，由于 FIFO 的引入消耗了仅 1 个 bram 资源，高效的实现了音乐播放功能

表 4.7 资源消耗

资源	数值	占比
Lut	1314	6.70%
Reg	488	2.49%
Bram	1	1.56%

4.5.6 内核交互

音乐播放模块引出的 key1-4 四个输入能控制整个音乐模块的使能、歌曲切换、播放暂停的功能，将其与 cortex-M0 以 AHB 桥挂载在一起，实现以 M0 内核进行对音乐模块各种功能的控制。

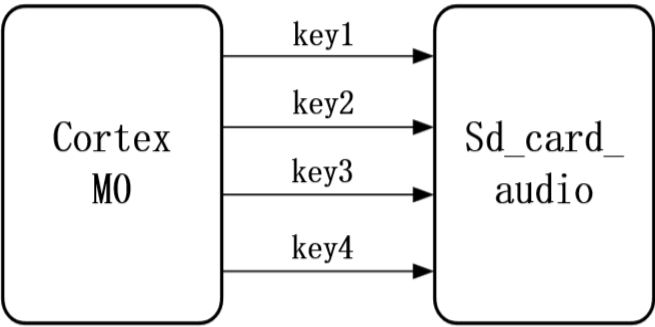


图 4.25 与 M0 交互

4.6 电源管理模块

4.6.1 原理分析

为整个系统进行供电，选择合适的电源管理芯片将使得整个电路的设计变得简便实用。本模块选择 ip5306 芯片作为主芯片构建电源管理模块，两节并联的 18650 电池作为供电，进行整个 pcb 板的设计制作。

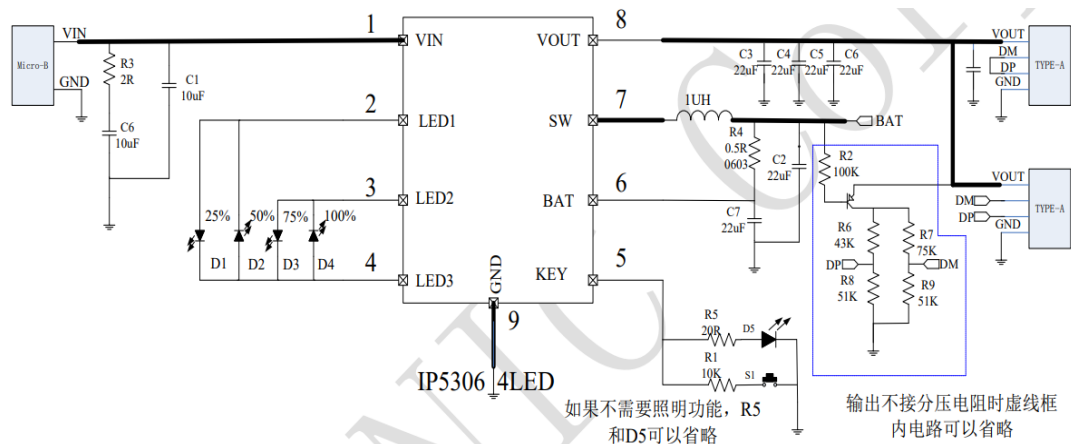


图 4.26 IP5306 应用电路

4.6.2 实物测试

通过 type-c 口进行供电，可以看见 LED 指示灯进行电量指示，模块为两节 18650 进行充电，此时 USB 口可同时向外界供电。

在不带负载时，VOUT 端输出电压为 5.20v，进行对系统供电运行后，电压稳定在 5.03V，能成功为整个设计的片上系统进行电压和电流的支持。



图 4.27 PCB 实物

4.7 对讲机

4.7.1 IIC 通信

此处的设计思想与前面的 SPI 相同，即软件模拟时序。唯一需要注意的是，由于从机需要应答，IIC 的 SDA 信号应设置为 inout 类型。且在主机需要接收应答时，释放 SDA 信号线，移交控制权，设计思路如公式（5）。

$$SDA = (ACK == 1'b1) ? 1'bz : SDA_reg; \quad (5)$$

4.7.2 实现复用

当 MSi001 与 QN8027 芯片分别调通后，即可分别实现射频信号的接收和发射。显而易见，当发射的信号频率与接收的信号频率相同时，便可实现短距离的无线通信，即对讲机。其实现是对两个芯片的联合调试，工程上需要将二者融合。然而，这并非简单的线性叠加，需要在对软硬件充分理解的基础上，实现复用，以下展开说明。

硬件上，发射与接收均为单声道，这就为复用创造了条件。由于左右声道完全相同，无论是输入信号用以发射，还是输出信号用以接收，本质上仅需单个通道，即可获得全部信息量。所以，在 FM 射频板上的两个输出信号 MCU_DA_Lo 和 MCU_DA_Ro，可分别作为接收解调后的语音信号和麦克风输入经采样后的待调制语音信号，完成硬件上的复用。

软件上，ADC 由原来的两通道（IQ 信号）变为三通道，增加了麦克风语音信号的输入。采样通道的变化，不仅意味着 IQ 信号对应采样标志时刻的变化，同样也是 IQ 采样频率，即解调语音信号的输出频率的变化。因此，在更改信号接收标志的同时，还需改变滤波器的系数，从而实现整体系统的联合搭建，如图 4.28。

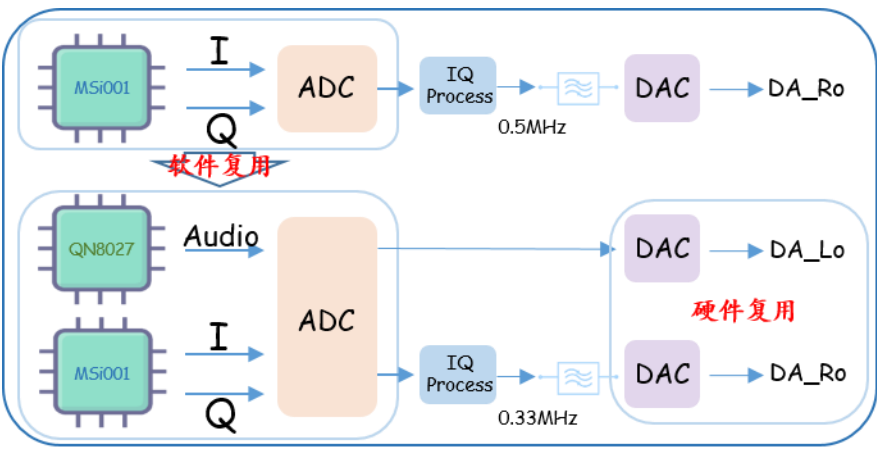


图 4.28 复用示意图

4.8 录音存储

4.8.1 需求分析

本设备已经通过 FM 解调模块进行了对 FM 广播的输出，通过一系列的信号处理后，我们已经得到优良的音质。那么对于接收到的 FM 广播信号，我们希望能够实现存储一段录音，使整个收音系统更加完整。由于整个系统已经有很好的音质，对广播信号进行录音存储并播放，我们就能得到一个有着录音功能的收音机。

对喇叭在放 FM 广播和录音切换，很好地对喇叭功放模块进行了复用，使得整个系统更加完整，资源更加充分使用。

4.8.2 录音存储模块接口信息

录音存储模块接口信号，方向与功能如下所示：

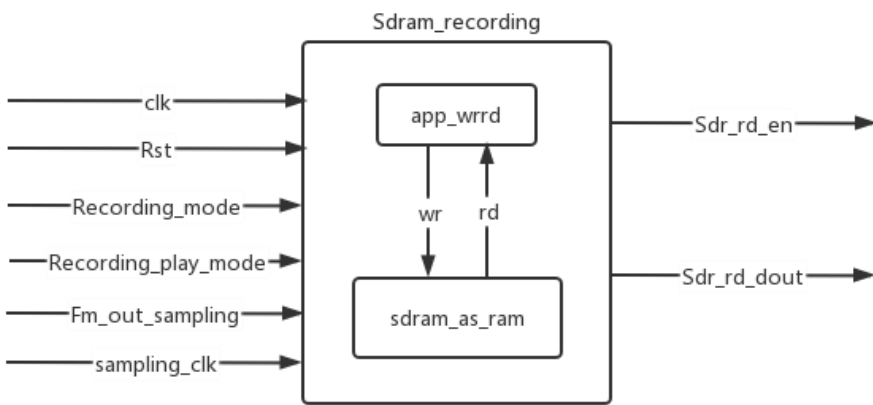


图 4.29 录音存储模块接口信息

表 4.8 信号详细功能

信号	方向	功能说明
clk	In	音乐播放工作时钟，100MHz
rst_n	In	复位信号，高电平有效
key1	In	录音使能
Key2	In	录音播放使能
Fm_out_sampling	In	写入的 FM 采样信号数据
Sampling_clk	In	FM 信号的采样时钟
Sdr_rd_en	output	有效读数据输出使能
Sdr_rd_dout	output	有效读出数据

4.8.3 实现原理

app_wrrd 模块顶层提供一种类似于 FPGA 内的 ERAM 读写接口给用户，方便用户对数据做读写保存，不需做额外多余的存储器件读写和调试工作。用户操作分 2 个通道，读通道和

写通道。因硬件机制问题，应用方案中有一个限制，即不支持读写同时操作。在用户对虚拟 ERAM 做操作前，若使用 ip 方案提供的刷新机制，要先判断当前虚拟 ERAM 是否忙，如果忙，需要等待 Sdr_init_ref_vld 信号变为低电平后，再做读写操作。

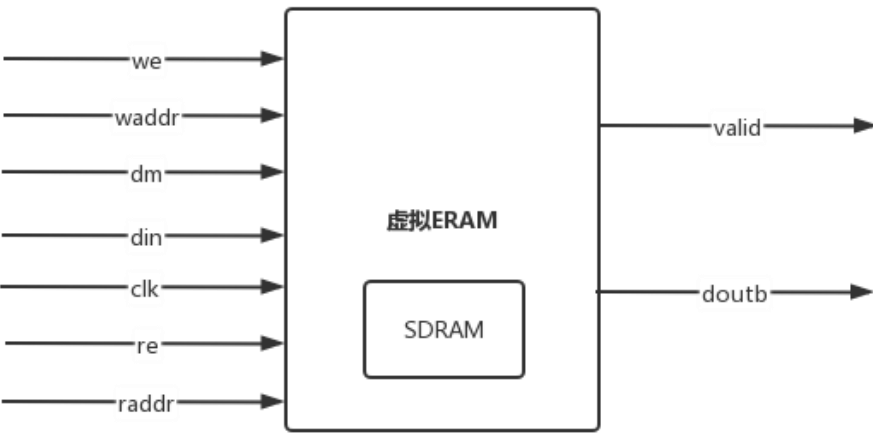


图 4.30 虚拟 ERAM

由于 Sdram_as_ram 封装成的虚拟 eram 的大小为 64M，地址位宽为 21，数据位宽为 32，而 FM 解调输出信号位宽为 12，且速率是 0.5MHz，若不对 FM 解调输出信号采样直接存储，最多只能存储 4s。所以要对 FM 解调输出信号进行抽样，将其速率降为 50kHz，这样最多可以存储录音 40s。Fm_out_sampling 为抽样后的信号，sampling_clk 为 50kHz 的抽样时钟。由于人耳可以听到的声音频率范围一般在 20Hz~20KHz 之间，根据奈奎斯特采样定理，采样频率 fs 大于信号中最高频率 fmax 的 2 倍时，采样之后的数字信号便能完整的反应真实信号。所以 50KHz 是满足条件的采样率。

4.8.4 仿真实验

用 chip_watcher 观察 Recording_mode 录音使能、Recording_play_mode 录音播放使能，App_wr_din 写数据、App_wr_addr 写地址，App_rd_addr 读地址，Sdr_rd_en 有效读数据输出使能，Sdr_rd_dout 有效读出数据，tx_vld 初始化有效。

初始化仿真如图 4.31，tx_vld 为高电平表示 sd 卡成功初始化并等待了一段时间。此时 Recording_mode 为低电平，即录音使能的按键还没有被拨动，App_wr_addr 为 0，App_wr_din 为 0，还没有开始写入数据。

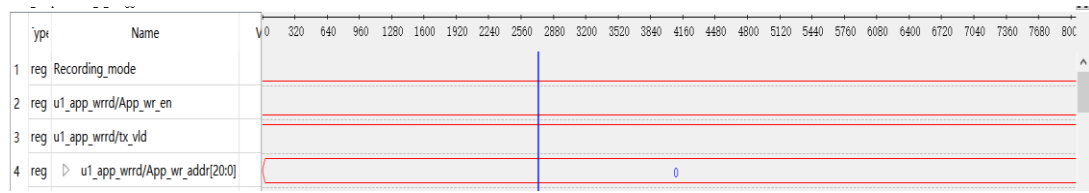


图 4.31 初始化仿真

录音存储仿真如图 4.32，按下 key1 写数据使能。Recording_mode 为高电平表示正在进行录音，此时 App_wr_addr 一直递增，App_wr_din 写入要存储的数据。

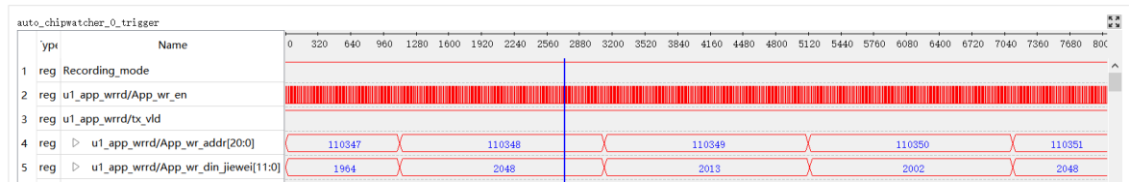


图 4.32 录音存储仿真

录音播放仿真如图 4.33，Recording_mode 为低电平表示录音结束，Recording_play_mode 为高电平表示播放录音。此时 App_rd_addr 一直递增，Sdr_rd_dout 有效读出数据并传入 PWM 模块进行录音播放。

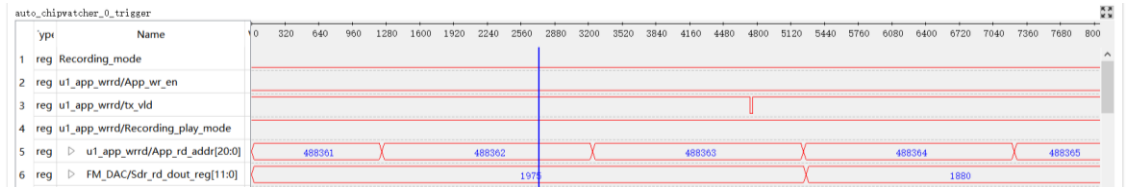


图 4.33 录音播放仿真

录音播放切换回播放电台仿真如图 4.34。Recording_play_mode 为低电平表示停止播放录音。此时写地址 App_rd_addr 停止递增，Sdr_rd_dout 停止读数据，录音暂停播放，开始播放收到的电台信号。

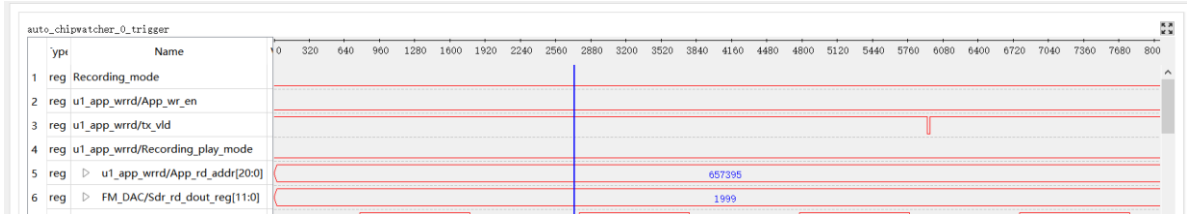


图 4.34 播放切换仿真

4.8.5 仿真实验

单独编译音乐播放模块文件，主要的资源消耗如表 4.9，可以观察到整个模块只占用了较少的资源就实现了录音存储功能，高效地实现了录音存储功能。

表 4.9 资源消耗

资源	数值	占比
Lut	242	1.23%
Reg	533	2.72%
PLL	1	25%

4.8.6 内核交互

音乐播放模块引出的 key1-2 两个输入能控制整个录音存储模块的使能、录音播放的功能，

将其与 cortex-M0 以 AHB 桥挂载在一起，实现以 M0 内核进行对音乐模块各种功能的控制。

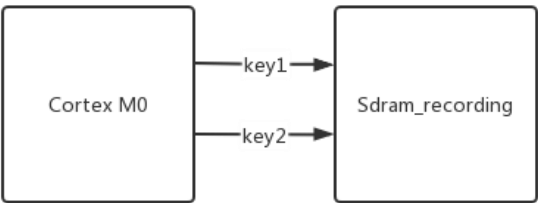


图 4.35 内核交互

4.9 K 歌模式

4.9.1 需求分析

本设备已经在各方面硬件软件的加持下，有着优秀的音质并能够进行 sd 卡的音乐播放和麦克风对讲。观察板上资源，我们可以了解到，板上除了基础题所用的无源音箱外，还存在一个外接高阻抗耳机或有源音箱，综合整体考虑，可以设计一个 K 歌平台，以 sd 卡进行伴奏歌曲演奏，麦克风输入人声进行评分，耳机或有源音箱来个人倾听或者外接功放随时随地 K 歌。

4.9.2 音乐播放模块接口信息

K 歌平台信号，方向与功能如下所示：

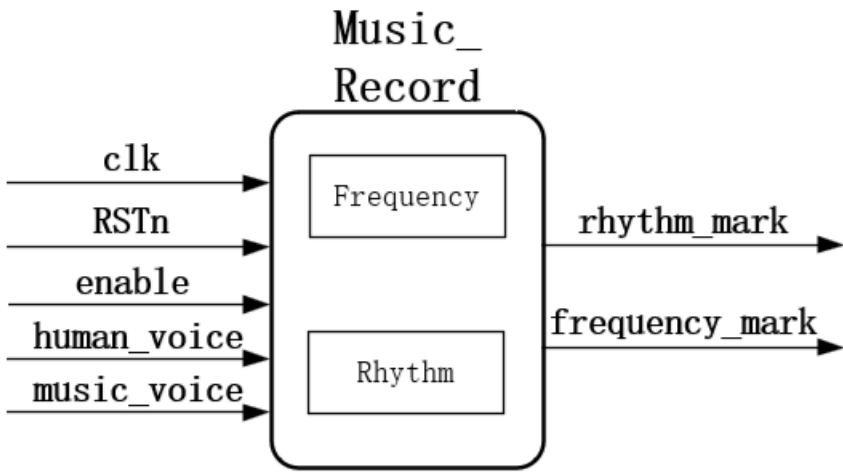


图 4.36 音乐播放信号流图

表 4.10 信号详细功能

信号	方向	功能说明
clk	In	音乐播放工作时钟，50MHz
RSTn	In	复位信号，高电平有效
enable	In	音乐评分使能
human_voice	In	麦克风输入人声
music_voice	In	音乐人声原声
rhythm_mark	output	节奏得分
frequency_mark	output	频率得分

4.9.3 实现原理

Wav 文件可以设置多声道，将其设置为双声道模式，其中一边保存原唱人声，一边保存伴奏，将伴奏通过有源音响传出，同时把麦克风中自己的人声与原唱人声进行比较，打出对应分数。

分数的评判可由多个方面组成，一段语音信号可以评判的特征如图，其中较为常见的有响度、过零率，同时音乐中节奏也时常计入考虑。

频率可以通过声音的过零率表示，节奏可以根据声音高低起伏判断，响度由于不同伴奏和人声的不匹配可能造成较大误差，不计入考虑。综上，分数的评判由频率和节奏来进行打分综合。

表 4.11 语音信号特征

类型	名称	物理或音乐意义
能量特征	均方根能量	信号在一定时间范围内的能量均值
时域特征	起音时间	音符能量在上升阶段的时长
	过零率	信号在单位时间通过 0 点的次数
	自相关	信号与其时间位移后版本的相似度
频域特征	谱质心	信号频谱中能量集中点。描述信号音色的明亮度
	MFCC	语音识别最常用特征，考虑人耳对不同频率感受程度
	频谱平坦度	量化信号和噪声之间的相似度
	频谱通量	量化信号相邻帧间变化程度
乐理特征	基音频率 f0	音高的频率
	失谐度	信号泛音频率和基音频率的整数倍间偏离程度
感知特征	响度	人耳主观感受到的信号强弱
	尖锐度	高频部分能量越大，人耳感觉越尖锐

4.9.4 仿真验证

以节奏评分为例，频率评分同理，通过 chip_watcher 在 8192 次采样中观察 music_rhythm_count 原唱节奏值、human_rhythm_count 人声节奏值，以 differ 人声和原唱之间节奏差值进行评分。

无人声无原唱仿真如图 4.37，此时 music_rhythm_count 原唱节奏值、human_rhythm_count 人声节奏值均为 0，差值 differ=0，当前分数定义为 100 分。

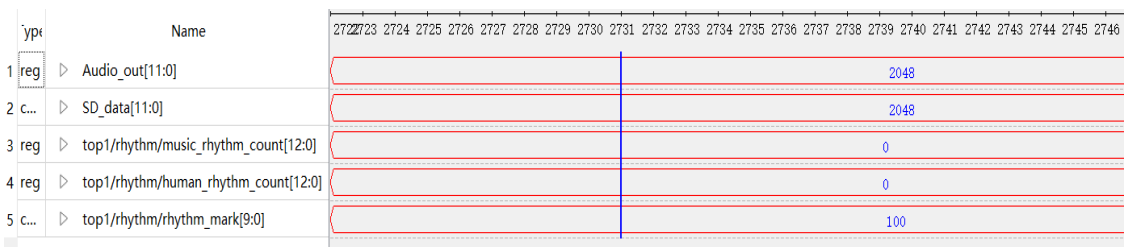


图 4.37 无人声无原唱

无人声有原唱仿真如图 4.38，此时 music_rhythm_count 原唱节奏值跟随音乐的播放进行增加、human_rhythm_count 人声节奏值保持为 0，差值 differ=music_rhythm_count，当前分数按分段进行 60 到 100 的评分。

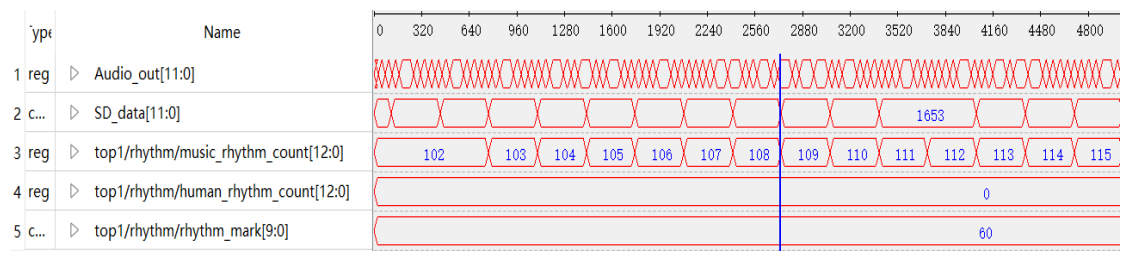


图 4.38 无人声有原唱

有人声无原唱仿真如图 4.39，此时 music_rhythm_count 原唱节奏值保持为 0、human_rhythm_count 人声节奏值跟随麦克风人声进行增加，差值 differ= human_rhythm_count，当前分数按分段进行 60 到 100 的评分。

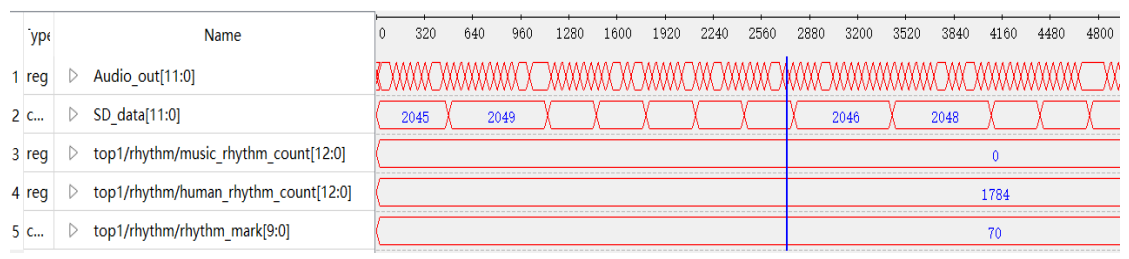


图 4.39 有人声无原唱

有人声有原唱仿真如图，即真实的 k 歌模式，此时 music_rhythm_count 原唱节奏值跟随音乐原声增加、human_rhythm_count 人声节奏值跟随麦克风人声进行增加，差值 $differ=|human_rhythm_count-music_rhythm_count|$ ，以差值进行分段评分。



图 4.40 有人声有原唱

4.9.5 资源消耗

单独编译 K 歌模块文件，主要的资源消耗如表 4.12，可以观察到整个模块只占用了较少的资源就高效的实现了 K 歌功能。

表 4.12 资源消耗

资源	数值	占比
Lut	586	2.99%
Reg	109	0.56%

4.9.6 内核交互

K 歌评分模块引出的 rhythm_mark、frequency_mark 两个信号控制整个评分模块获取相关分数的功能，将其与 cortex-M0 以 AHB 桥挂载在一起，实现以 M0 内核进行对 k 歌模块相关分数的读取。

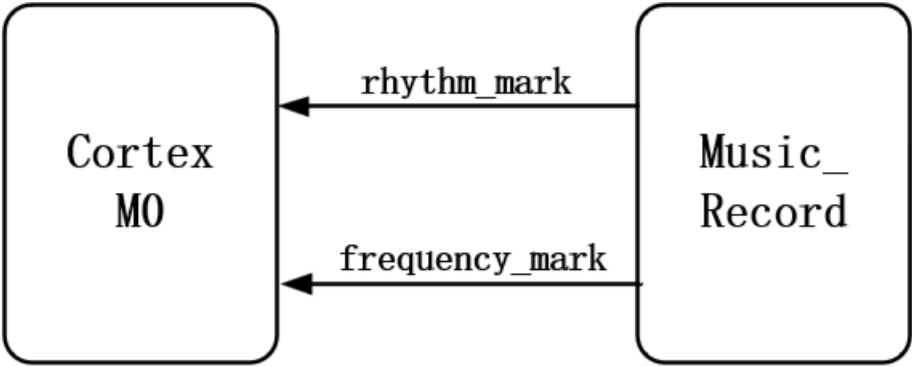


图 4.41 与 M0 交互

4. 10 FM 游戏

共设计两个游戏，第一个是以打地鼠为原型的打兔子游戏，第二个是童年经典游戏——碰手指。规则较为简单，且均为双人模式，轻松诙谐。

4. 10. 1 设计原理

游戏双方分别使用一个串口屏作为游戏操纵器，借助 FM 通信使双方玩家的游戏界面统一，即任意一方操作时的结果需要实时传输到另一方的游戏界面。

为达成这一目的，我们以矩形脉冲作为调制信号，其频率、占空比作为传输信息，调制到 FM 载波上进行传输。考虑到 FM 射频模块二阶 RC 电路对基带信号的滤波作用，我们采用的频率集中在 500Hz~1000Hz，此时解调还原出的信号较强，可识别性较好。

既然涉及到通信，必然牵扯到协议的制定。由于游戏规模、复杂度、信息量都较小，本次采取的协议本着简单高效的原则，如图 4.42 所示。一帧数据分为头帧和数据帧，头帧作为启动标志位，采用 1000Hz；数据帧采用 500Hz 及以上，视传输的信息种类而定。

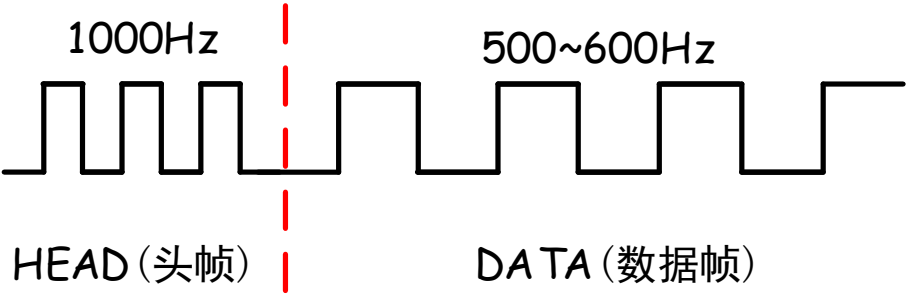


图 4.42 通信协议

具体到游戏，对于打兔子，双方玩家一方为锤子，一方为兔子。共设置 6 个洞口，因此只需采用六种不同的频率即可。经实验测试，采用间隔为 10Hz，依次递进，即节省频谱资源，又有不错的判断效果。

表 4.13 打兔子映射值

洞口	频率值
1	500Hz
2	510Hz
3	520Hz
4	530Hz

5	540Hz
6	550Hz

对于碰手指，双方玩家每人两只手，共四只手，且每只手可表达 0~9，10 个数字。因此信息量从原来的一维上升至二维，我们采取的策略是：数字信息由频率表征，手信息由占空比表征。频率仍以 10Hz 递进，占空比原本设置为 20%、40%、60%、80%，但考虑到在每个频率下若要达到相同的占空比，则需构造 40 个数值来映射（共 10 个频点，每个频点四个占空比数值）。大大增加了复杂度，这里便采用模糊处理的思想：将占空比设定为四个固定数值，由于频点间隔为 10Hz，因此只要选取合适的数值，即可满足这四个固定数值在每个频率下产生矩形波的占空比近似为 20%、40%、60%、80%。这种模糊处理的思想，将问题规模大大简化，无论是矩形波的产生，还是后面的接收处理，都变得较为容易。

表 4.14 碰手指映射值

数字	频率值	手	占空比
0	500Hz	1	近似 20%
1	510Hz	2	近似 40%
2	520Hz	3	近似 60%
3	530Hz	4	近似 80%
4	540Hz		
5	550Hz		
6	560Hz		
7	570Hz		
8	580Hz		
9	590Hz		

4.10.2 硬件设计

为完成上述的发射接收过程，共划分为三个模块：`game_emit.v` 完成信号的产生、`wav_scan.v` 完成对解调后信号的检测扫描、`wav_fsm.v` 根据检测结果进行判断。

首先,game_emit 的整个信号产生过程由状态机实现,如图 4.43。若 Number_emit 被赋值,即不为 0 时,即开启一次发射过程,状态机由空闲进入头帧、数据发送过程。在发送时,根据需要发送的数值,设置矩形波对应的频率和占空比。最后,进入结束状态,在此状态下,会将 Number_emit 硬件置零,供下次触发使用。

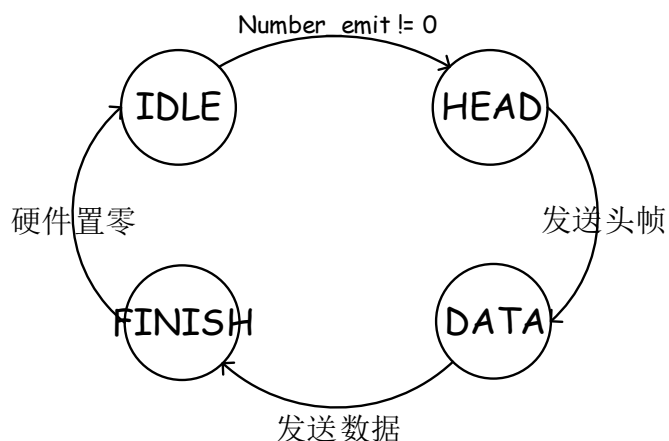


图 4.43 发射状态机

wav_scan.v 的作用是对解调接收的信号进行检测,得到实时的频率值和占空比。整个检测原理比较简单,与前面自动搜台有异曲同工之妙:当没有接收到信号时,数值在 2048 左右;而接收到信号时,数值可看作是对矩形波的采样,正占空比时数值明显高于 2048,负占空比则明显低于 2048。根据这一特性,便很容易得到信号的正负脉冲时间,进而计算出周期与占空比。

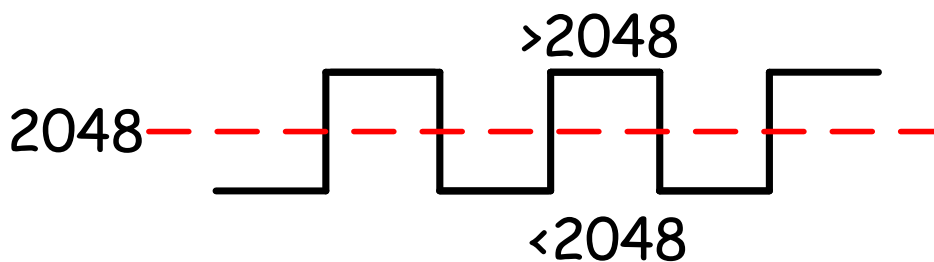


图 4.44 检测原理

在检测到信号的频率和占空比后，即可依据头帧和数据各自的信息特征进行判断，此处也采用了两段式状态机，分 HEAD 和 GET 两个状态。与发射状态机思想类似：首先等待头帧，一旦检测到头帧的频率，即跳转至下一状态，对数据值进行判断。最后完成整个状态机的过程，得到与频率和占空比相相应的数值，挂起中断反馈给 M0 读取。为了清晰展示整个检测过程，我们对状态机进行了仿真测试，结果如图 4.45 所示。

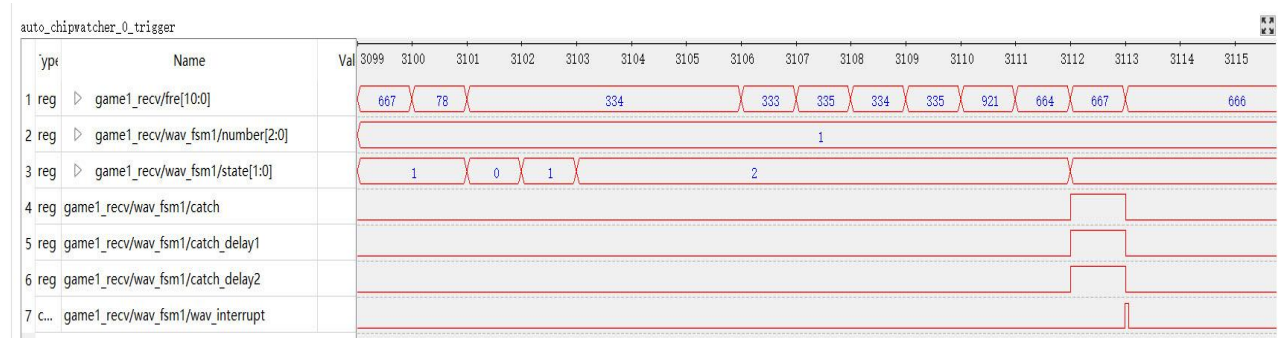


图 4.45 仿真测试

4. 10. 3 串口屏设计

4. 10. 3. 1 打兔子

打兔子游戏 UI 设计一方为兔子，一方为锤子，如图 4.46。

进行打兔子游戏时，游戏双方分别选择当兔子或当锤子的一方，通过设定的 FM 无线通信协议完成打兔子游戏的双人交互。



图 4.46 打兔子主界面

(1) 当兔子

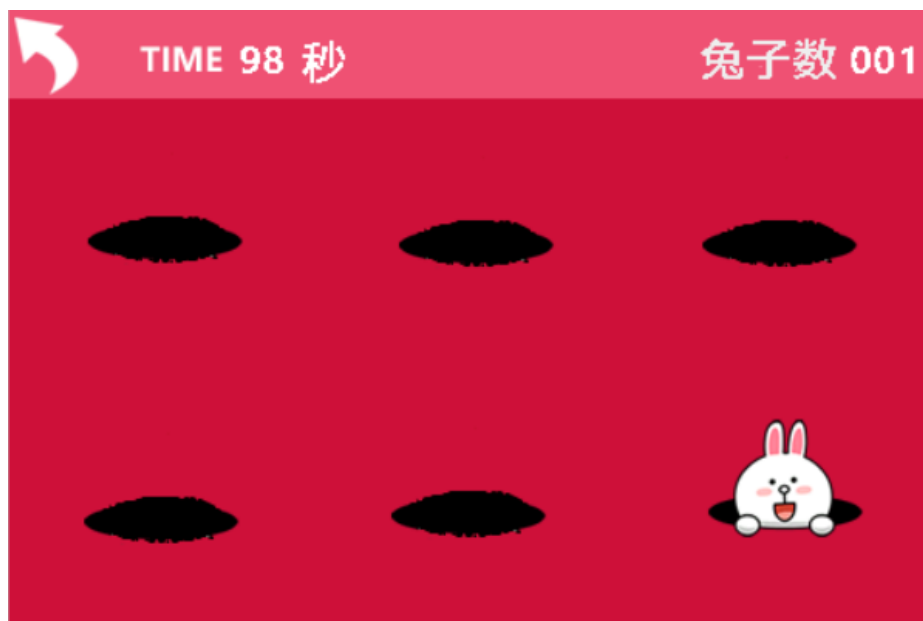


图 4.47 当兔子主界面

当兔子时候界面如图 4.47，按下 **start** 键后游戏开始。按下任意一个洞口时将会出现维持 0.4 秒的兔子，对应锤子界面将在对应洞口出现兔子。0.4 秒后，对应洞口兔子将会消失，当兔子玩家可进行下一次洞口出现兔子的选择。

TIME 可以设置游戏时间，进行任意时长游戏设置。当游戏时间结束，所有的洞口将会消失，产生的兔子数将显示在左上角。

(2) 当锤子

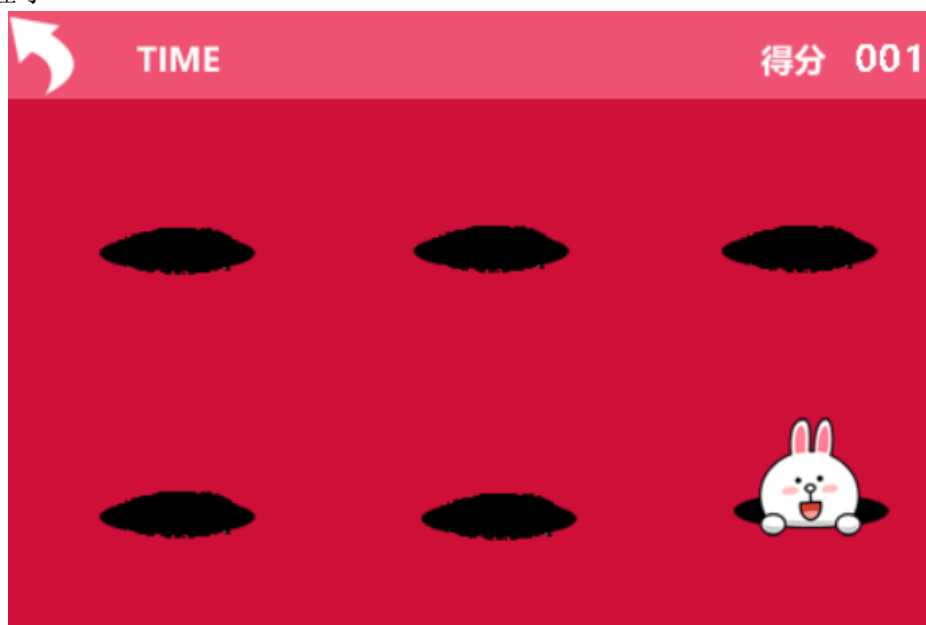


图 4.48 当锤子主界面

当锤子时候界面如图 4.48，在当兔子端按下任意一个洞口时，锤子端将会在对应洞口出现兔子，对应洞口兔子在兔子端没有进行下一次选择时不会消失。敲击对应洞口兔子，兔子将会消失并进行得分并显示在右上角。

4.10.3.2 碰手指

碰手指游戏 UI 设计为双人四根手指。进行碰手指游戏时，游戏双方分别选择一方，通过设定的 FM 无线通信协议完成碰手指的双人交互。

碰手指界面如图，中央显示当前是碰手指中第几个手指，在碰完后，对应位置的手指将会进行数值变换，当任意一方完成两个手指均达到 0 的成就时，对应边将会显示 WIN 的胜利标志。

点击返回将退回到打兔子游戏，重开可让所有手指回到 1 的初始状态，进行新一轮的碰手指游戏。

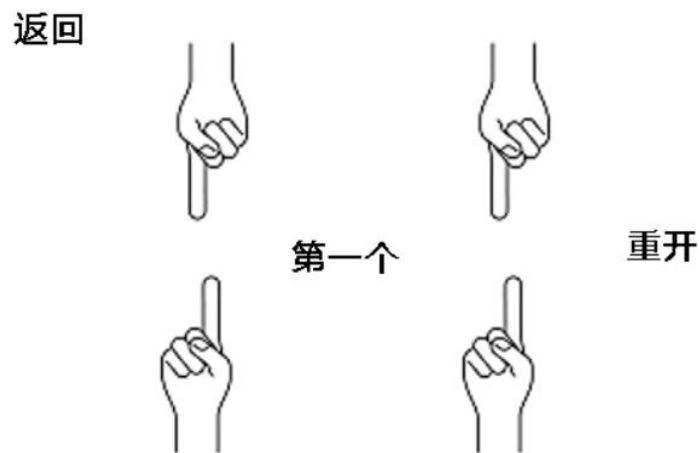


图 4.49 碰手指主界面



图 4.50 碰手指胜利界面

4.10.3.3 通信格式

打兔子游戏中通信格式如下：

表 4.15 打兔子通信格式

组成	头帧		数据帧			尾帧
数值	2C	12	01	00	01 - 06	5B
意义	准确识别数据帧 开头		第一个 游戏	无效位	对应洞 口数值	准确识别数据帧 结尾

碰手指游戏中通信格式如下：

表 4.16 碰手指通信格式

组成	头帧		数据帧			尾帧
数值	2C	12	02	01 - 04	01-0A	5B
意义	准确识别数据帧 开头		第二个 游戏	对应手 指位置	对应手 指数值	准确识别数据帧 结尾

4.11 机械臂远程操控

4.11.1 控制原理

本设计基于我们制定的 FM 通信协议，一帧数据分为头帧和数据帧，头帧作为启动标志位，采用 1000Hz；数据帧采用 700Hz 及以上，视传输的信息种类而定。

借助 3D 打印技术制作体感手套，并在体感手套上嵌入姿态传感器 JY901，JY901 能输出三轴角度，且传感器内部能测量 D0~D3 四个端口的电压值，我们选取了端口 D0 作为数字开关的输入端口，此数字开关安装在体感手套上，当我们做出握拳的手势触碰开关后，输出电平随着受力大小发生相应改变，所以可以根据数字开关输出的高低电平控制机械臂爪子的开合，并根据陀螺仪的姿态变化控制机械臂舵机的转动。

将体感手套作为开发板的外设，这样我们便可以将传感器传来的角度值与端口电压值作为信息进行 FM 通信传输，信息传输到与机械臂相连的开发板上，开发板收到三个轴的角度后将其转换为对应舵机的 PWM 占空比值从而控制舵机转动对应的角度，收到 D0 端口电压值控制爪子开合，利用 FM 通信实现了体感控制机械臂。

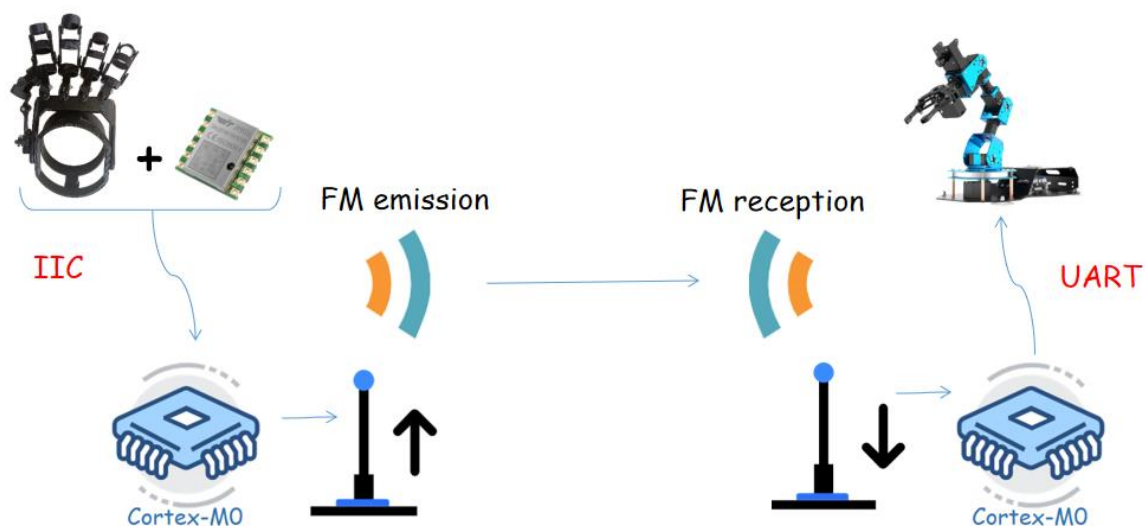


图 4.51 操控示意图

如图 4.52 所示，模块内部集成了姿态解算器，配合动态卡尔曼滤波算法，能够在动态环境下准确输出模块的当前姿态，姿态测量精度静态 0.05 度，动态 0.1 度，稳定性极高。

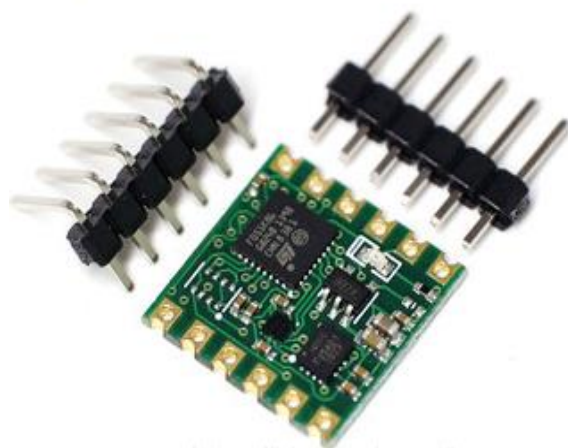


图 4.52 JY901 实物图

对于机械臂舵机的控制，我们采取增量式控制法，即把当前姿态传感器传来的角度值与上一次的角度值作比较，若当前的角度值或端口电压值比上一次的大，则发送占空比近似 25% 的方波信号，反之则发送占空比近似 75% 的方波信号。由于受到外界因素的干扰，传感器的传输数据不是特别稳定，所以当我们检测到当前角度与上一次角度的差值大于 3° 时，才认为这一次传输的数据是有效的。与机械臂相连的开发板收到信号后控制相应的舵机正转或者反转，实现对位置的调节。需要实时调节 PWM 占空比值的舵机一共有四个，三个轴的角度值和 D0 端口的电压值对应四个舵机，所以舵机由频率表征，频率仍以 10Hz 递进。

表 4.17 机械臂映射

舵机	频率值	角度值或端口电压 值比较上次	占空比
1	700Hz	更大	近似 25%
2	710Hz	更小	近似 75%
3	720Hz		
4	730Hz		

4.11.2 通信方式

体感手套与开发板通过 IIC 协议进行通信，将传感器的角度数据传输给开发板，开发板对角度数据进行处理后通过 FM 通信传输给与机械臂相连的开发板，开发板通过串口与舵机驱动板相连，实现对机械臂的控制。

机械臂体感控制中通信格式如下：

表 4.18 机械臂体感控制通信格式

组成	头帧		数据帧			尾帧
数值	2C	12	03	01 - 04	01 - 02	5B
意义	准确识别数据帧开头		机械臂控制	对应舵机位置	对应舵机 PWM 占空比相对大小	准确识别数据帧结尾

4.12 语音识别

4.12.1 需求分析

语音识别是产品走向智能化的最主要的方向，让本设备听懂人类语言并进行对应动作目标，简化整个系统繁琐的按键操作。语音识别让设备拥有一次语音指令就完成相应动作的能力，使得整个系统更加产品化、智能化。

4.12.2 语音识别模块

本设备通过串口与 CI1122 芯片进行通信，实现整体的智能化。CI1122 是一颗专用于语音处理的人工智能芯片，可广泛应用于家电、家居、照明、玩具等产品领域，实现语音交互及控制。CI1122 内置自主研发的脑神经网络处理器 BNPU，支持 200 条命令词以内的本地语音识别。

通过串口通信，进行 M0 和 CI1122 芯片之间的交互。通过语言芯片实现了“自动搜台”、“上一个电台”、“下一个电台”、“存储电台”、“清空电台”和对语音说出的频点进行调台的语音识别能力。

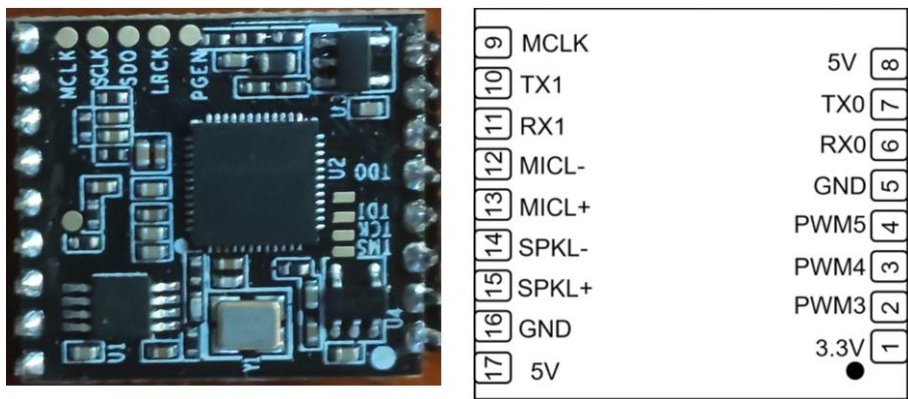


图 4.53 CI1122 实物和引脚图

4.12.3 通信格式

CI1122 与 FPGA 通信格式如下：

表 4.19 CI1122 通信格式

名称	长度	描述
Head	2	头帧 0xA5 0xFC
Length	2	Data 的长度
Msg_type	1	0xA0 命令消息类型
Msg_cmd	1	0x91 命令词
Msg_seq	1	消息序列号
Data	7	6 位语义 ID，1 位识别分值
Check_sum	2	校验帧
Tail	1	尾帧 0xFB

4.13 立体声传输

4.13.1 立体声广播原理

随着技术的发展与进步，调频广播不仅仅传送单一节目，而可传送多套节目，也可传送数据，复合调制了其他信息，充分利用了频谱资源和技术设备，这些技术统称为调频多工技术。单声道调频广播，只需要传送 40Hz~15KHz 的音频信号，一个传声器来捡拾信号就够了，其调制频谱比较简单。而调频立体声则不然，必须采用两套传声器来分别捡拾舞台左右两边的信号，并采用两个声道依次传输。

在导频制立体声广播中，从调频载波上解调出来的信号，还不是已分解了的左、右声道音频信号，而是立体声复合信号。此信号中还有抑载平衡调幅波（DDS），需要进一步解调才能分解出左、右声道信号，这一过程称为立体声解调。

调频立体声广播是兼容 FM 单声道广播的复合调制制式，单声道收音机接收立体声广播

信号时，解调的音频是左右两声道的和信号 $L+R$ 。立体声的基带频谱分布如下，立体声信号的两个声道，相加得到和信道 $L+R$ ，相减得到差信道 $L-R$ 。其中 $L+R$ 信号占据基带信号的 $0\sim 15\text{kHz}$ 的带宽。在基带信号 19kHz 处插入一个正弦导频，以便接收机倍频恢复 38kHz 的副载波。 $L-R$ 信号通过抑制载波调幅调制在 38kHz 的副载波上，占据 $23\sim 53\text{kHz}$ 的带宽。最后， $L+R$ 、导频、 $L-R$ 信号共同作为 53kHz 带宽的基带信号，通过 FM 调制发送出去。

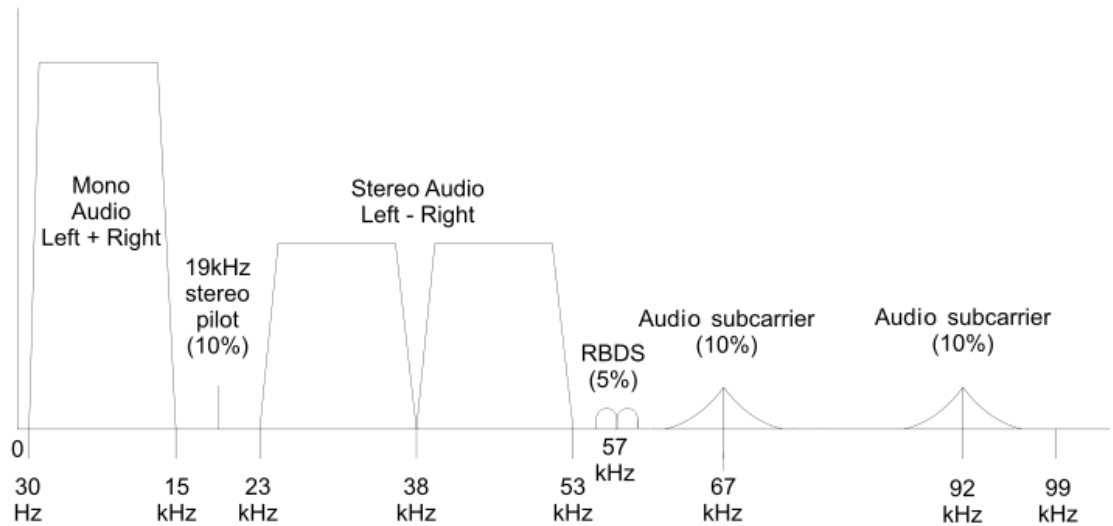


图 4.54 调频多工频谱分布

除立体声之外，在比 $L-R$ 信号更高的频段，工程师开发了多种利用子载波/副载波的数据传输模式。RDS(Radio Data System)和 RBDS(Radio Broadcast Data System)在 57kHz 的副载波上使用 2PSK 调制 1187.5bps 的数据，占据大约 4kHz 的带宽。此外在 DARC 中，可以使用 67kHz 和 92kHz 的副载波。

4.13.2 立体声数字解调方案

4.13.2.1 导频解调

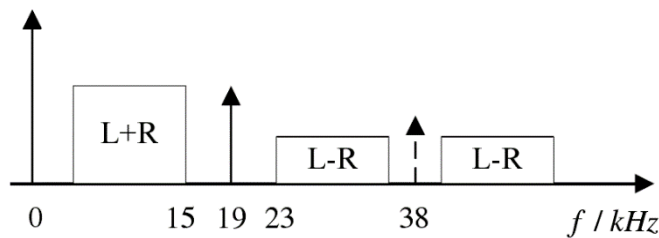


图 4.55 立体声频谱分布

双声道立体声广播的频谱分布如图，为分离立体声信号，采取与合成立体声相逆的处理过程。充分利用导频，分别采用低通、带通滤波器取得相应的频谱分量，再恢复左右声道数

据，解调框图如图 4.56 所示。

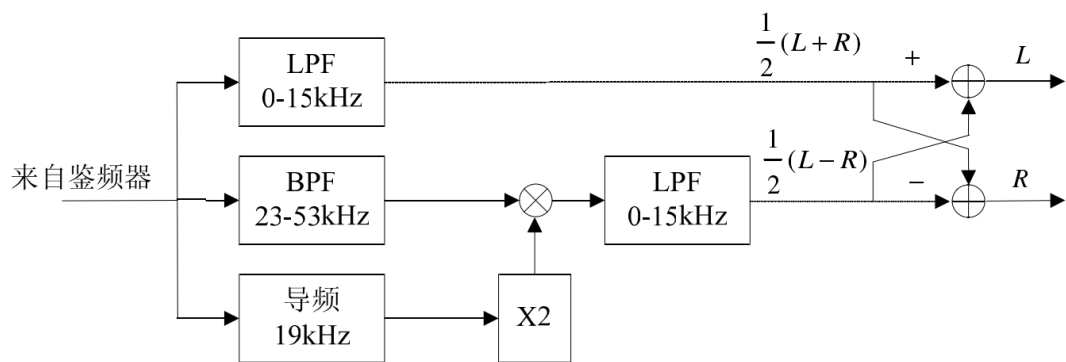


图 4.56 导频方案解调框图

根据框图，分析可得，解调导频方案共需四个滤波器、两个乘法器，其中滤波器采用 FIR 滤波器，保证信号相位的稳定性。同时，FIR 模块可利用 MATLAB 的滤波器向导生成参数，免去了手工求解传递函数和构建滤波器拓扑，解调设计如图 4.57 所示。

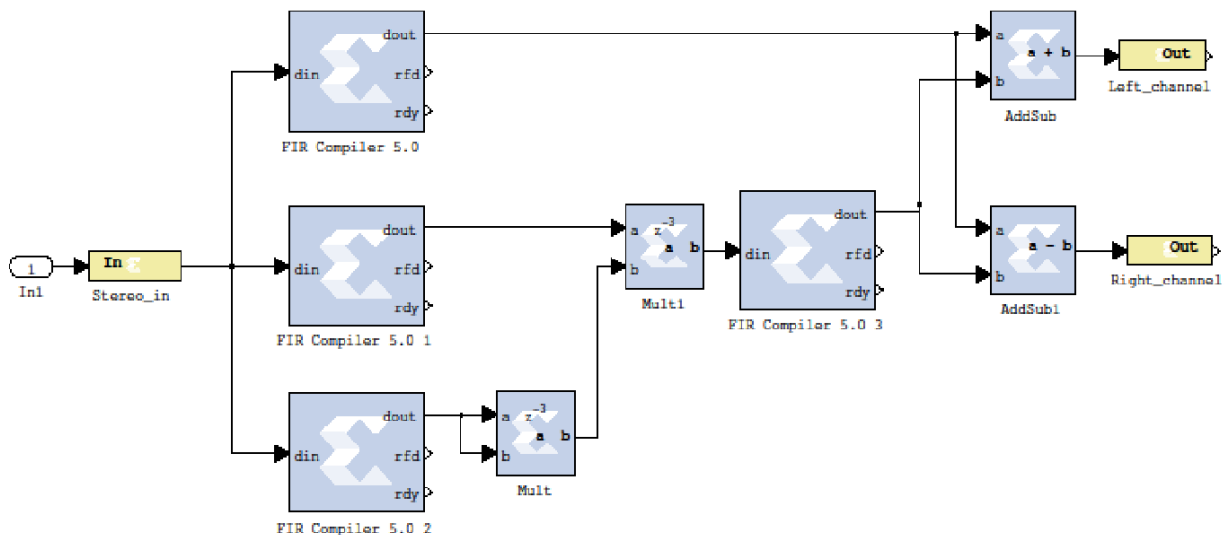


图 4.57 导频方案设计框图

4.13.2.2 数字化正交解调

基于软件无线电的数字化正交解调方法，不仅有较低的复杂度，较好的抗噪声性能，还可以解决本地载波与调制载波之间的频偏和相偏带来的信号失真问题。上文的 FM 解调，采取的就是这种方案。作为通用解调系统，无论哪种调制（AM、DSB、FM、PM 等），都要经过 IQ 正交解调后产生 IQ 两路信号，再视调制信号采取不同的解调算法即可完成输出。

对于 AM/DSB 调制，数字正交解调的原理框图如 4.58 所示。

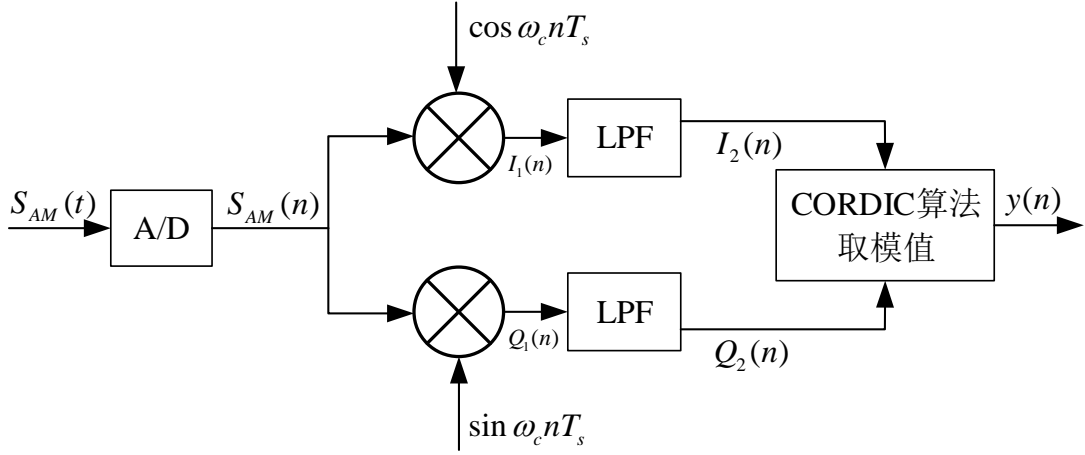


图 4.58 数字正交解调框图

AM 调制模拟信号表达式为:

$$S_{AM}(t) = (A + m(t)) \cos(\omega_c t + \theta_0) \quad (6)$$

中频信号经过 A/D 采样后转化为数字信号 $S_{AM}(n)$ ，表达式为:

$$S_{AM}(n) = (A + m(n)) \cos(\omega_c n T_s + \theta_0) \quad (7)$$

式中， $T_s = 1/f_s$ ， f_s 为信号的采样频率。

将公式 7 与本地载波振荡器产生的两个正交正弦载波信号分别相乘后，可得:

$$\begin{aligned} I_1(n) &= S_{AM}(n) \cos(\omega_c n T_s + \theta_1) \\ &= (A + m(n)) \cos(\omega_c n T_s + \theta_0) \cos(\omega_c n T_s + \theta_1) \\ &= \frac{1}{2} (A + m(n)) [\cos(2\omega_c n T_s + \Delta\theta) + \cos \Delta\theta] \end{aligned} \quad (8)$$

$$\begin{aligned} Q_1(n) &= S_{AM}(n) \sin(\omega_c n T_s + \theta_1) \\ &= (A + m(n)) \cos(\omega_c n T_s + \theta_0) \sin(\omega_c n T_s + \theta_1) \\ &= \frac{1}{2} (A + m(n)) [\sin(2\omega_c n T_s + \Delta\theta) - \sin \Delta\theta] \end{aligned} \quad (9)$$

式中， $\Delta\theta = \theta_0 - \theta_1$ 。通过低通滤波器后，因 $\cos(2\omega_c n T_s + \Delta\theta)$ 分量和 $\sin(2\omega_c n T_s + \Delta\theta)$

分量是高频分量，所以被滤除了，得到 IQ 两路信号 $I_2(n)$ 和 $Q_2(n)$ ，表达式如下:

$$I_2(n) = \frac{1}{2} (A + m(n)) \cos \Delta\theta \quad (10)$$

$$Q_2(n) = \frac{1}{2} (A + m(n)) \sin \Delta\theta \quad (11)$$

把 IQ 两路信号取平方和后得：

$$I_3(n) = I_2^2(n) = \frac{1}{2}(A + m(n))^2 \cos^2 \Delta\theta \quad (12)$$

$$Q_3(n) = Q_2^2(n) = \frac{1}{2}(A + m(n))^2 \sin^2 \Delta\theta \quad (13)$$

相加后开方得：

$$y(n) = \sqrt{I_3(n) + Q_3(n)} = \frac{1}{2}(A + m(n)) \quad (14)$$

$y(n)$ 就是 AM 调制信号的包络，在后续的信号处理中，通过隔直滤波器滤除掉 $y(n)$ 信号中的直流分量 A ，就可得到原始的调制信号 $m(n)$ 。

上述推导过程中，本地载波与调制载波的频率相同，如果设定本地载波频率与调制载波频率不相等，设本地载波角频率为 ω_2 ，则 IQ 信号推导过程如下：

$$\begin{aligned} I_1(n) &= S_{AM}(n) \cos(\omega_L nT_s + \theta_1) \\ &= (A + m(n)) \cos(\omega_c nT_s + \theta_0) \cos(\omega_L nT_s + \theta_1) \\ &= \frac{1}{2}(A + m(n)) [\cos((\omega_c + \omega_L)nT_s + \Delta\theta) + \\ &\quad \cos((\omega_c - \omega_L)nT_s + \Delta\theta)] \end{aligned} \quad (15)$$

$$\begin{aligned} Q_1(n) &= S_{AM}(n) \sin(\omega_L nT_s + \theta_1) \\ &= (A + m(n)) \cos(\omega_c nT_s + \theta_0) \sin(\omega_L nT_s + \theta_1) \\ &= \frac{1}{2}(A + m(n)) [\sin((\omega_c + \omega_L)nT_s + \Delta\theta) - \\ &\quad \sin((\omega_c - \omega_L)nT_s + \Delta\theta)] \end{aligned} \quad (16)$$

通过低通滤波器后，得到 IQ 两路信号 $I_2(n)$ 和 $Q_2(n)$ ，公式如下：

$$I_2(n) = \frac{1}{2}(A + m(n)) \cos(\Delta\omega nT_s + \Delta\theta) \quad (17)$$

$$Q_2(n) = \frac{1}{2}(A + m(n)) \sin(\Delta\omega nT_s + \Delta\theta) \quad (18)$$

经过取模运算后，仍然可以推出：

$$y(n) = \sqrt{I_2^2(n) + Q_2^2(n)} = \frac{1}{2}(A + m(n)) \quad (19)$$

由此可见，这种正交解调方法允许本地载波与调制载波之间存在一定的频偏和相偏，不需要保持严格的同频同相就能无失真的恢复出原调制信号。

4.13.3 导频提取方案

为了提取导频，我们选择对速率是 0.5MHz 数据源进行带通滤波，通带频率为 18800Hz-19200Hz，利用 Matlab 设计 FIR 带通滤波器，阶数为 16 阶，设计后发现滤波器幅频响应较差，近似于低通滤波器，如下图所示。

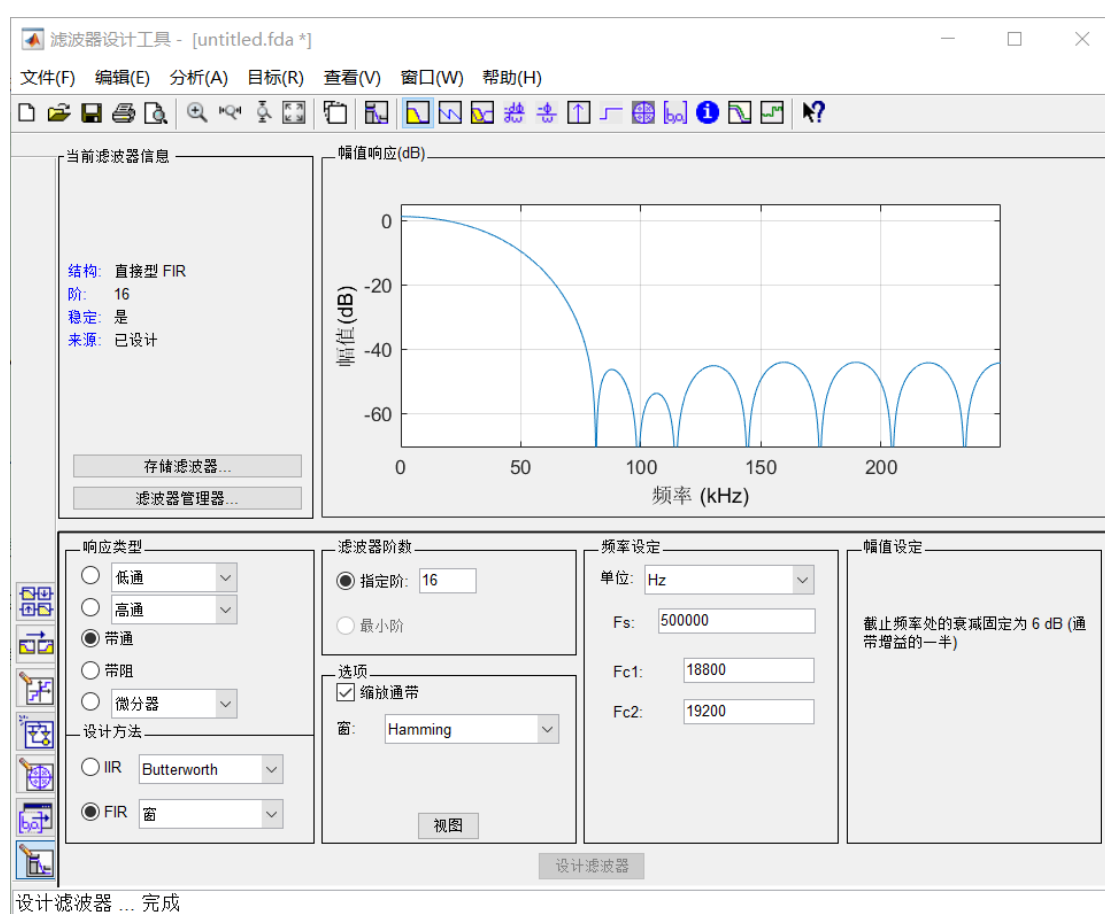


图 4.59 16 阶 FIR 带通滤波器

所以我们决定对数据源进行抽样以降低速率，并提高 FIR 带通滤波器的阶数，观察是否能改善效果。经过多次试验，我们发现当抽样速率为 50k，阶数为 64 阶时效果较好。滤波器的设计参数及幅频响应如下图所示。

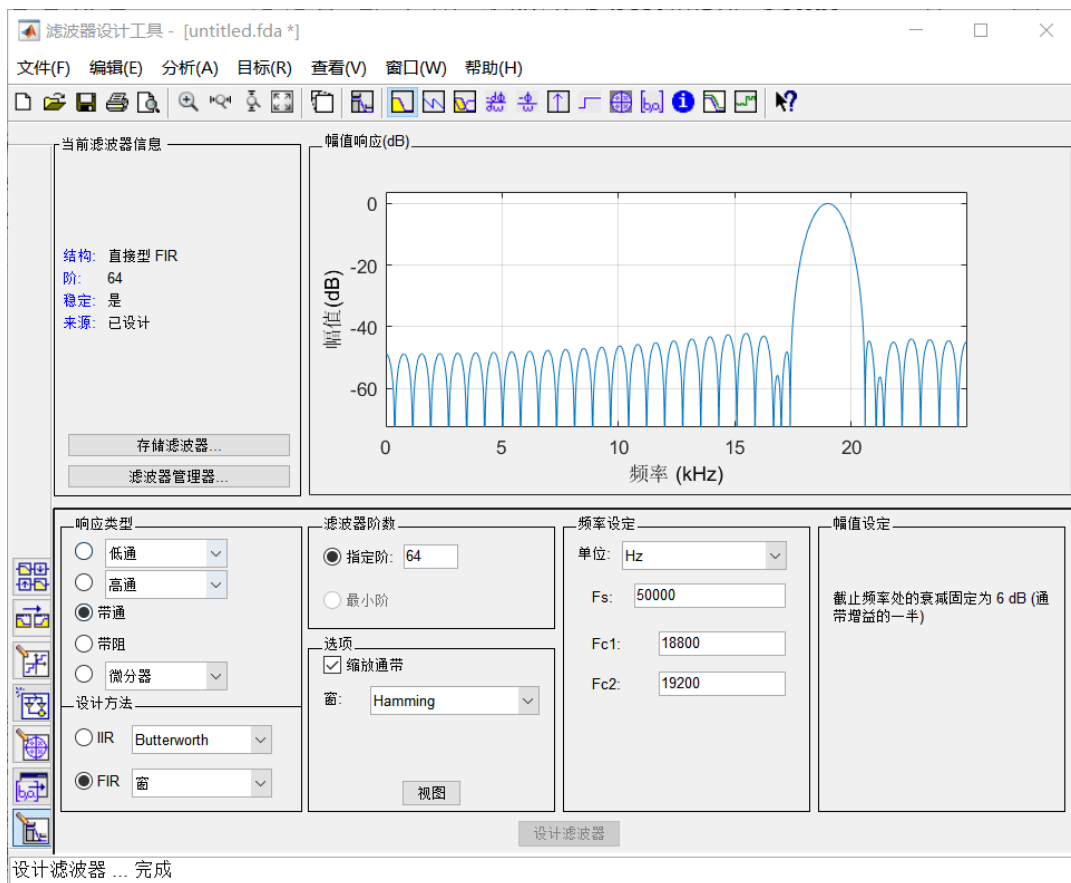


图 4.60 64 阶 FIR 带通滤波器

将带通滤波后输出的数据送到 **Bram** 中，通过调试工具 **Bram editor** 导出此数据并将此数据导入 **Matlab**，对该信号进行频谱分析，如下图所示。此时滤波效果不错。

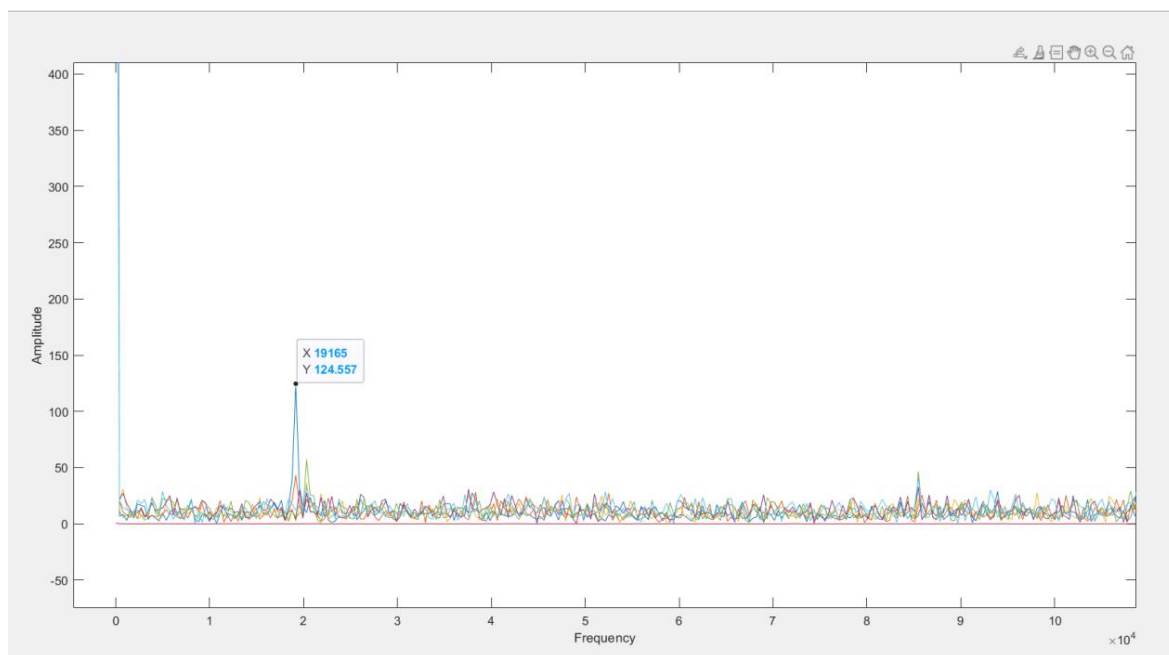


图 4.61 信号频谱分析

观察已使用资源，可以看到查找表资源已经使用了 88.16%，Dsp 也全部用完，由于方案后面还要设计三个滤波器，包括一个带通滤波器和两个低通滤波器，所以资源不够。

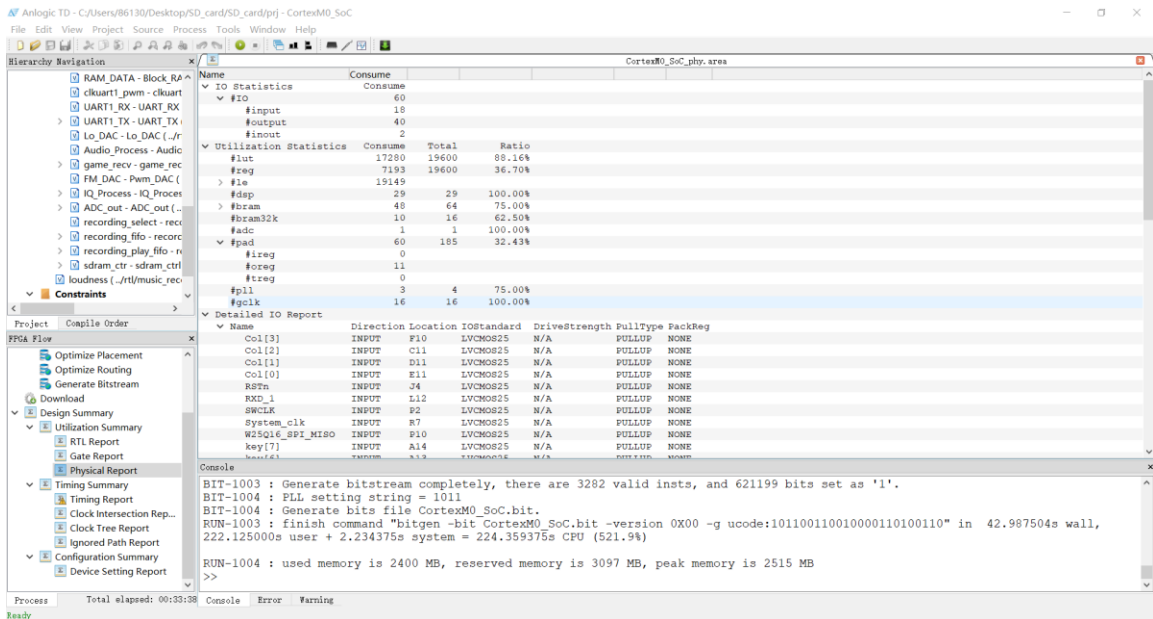


图 4.62 资源消耗

4. 13. 4 数字解调方案

4. 13. 4. 1 DDS 组成原理

此方案将方案一的提取导频改成利用 DDS 生成一个 38Khz 的正弦信号与余弦信号。

DDS 频率合成方法具有低成本、低功耗、高分辨率和快速转换时间等优点。DDS 的基本结构主要由相位累加器、相位调制器、波形数据表 ROM、D/A 转换器四大结构组成，其中较多设计还会在数模转换器之后增加一个低通滤波器。DDS 结构示意图如图 4.63。

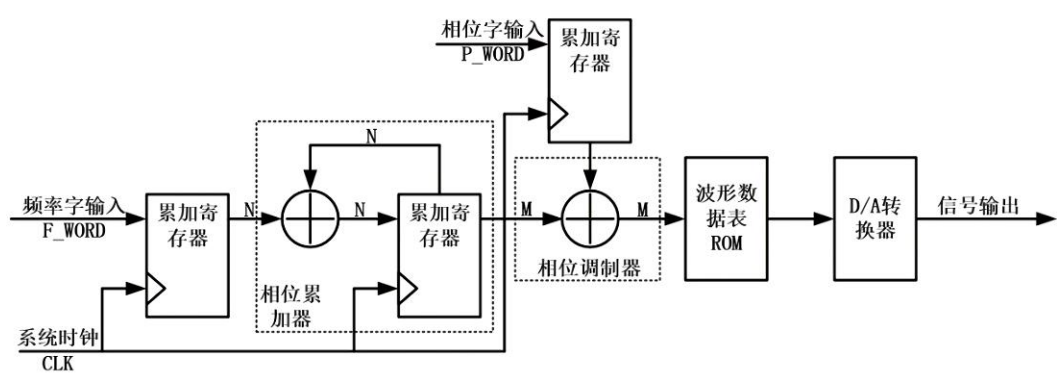


图 4.63 dds 结构

系统时钟 CLK 为整个系统的工作时钟， 频率为 fCLK；频率字输入 F_WORD，一般为整数，数值大小控制输出信号的频率大小，数值越大输出信号频率越高，反之，输出信号频率

越低；相位字输入 P_WORD，为整数，数值大小控制输出信号的相位偏移，主要用于相位的信号调制；输出信号设为 CLK_OUT，频率设为 fOUT。

相位累加器用于完成相位累加，生成相位码。相位累加器的输入为频率字输入 K，表示相位增量，设其位宽为 N。其在输入相位累加器之前，在系统时钟同步下做数据寄存，数据改变时不会干扰相位累加器的正常工作。

3.13.4.2 本地载波设计

根据立体声解调原理，本设备需要实现 38KHZ 的本地载波进行相关解调，本地载波信号通过运用 DDS 产生 38KHZ 正弦信号和余弦信号。

由于所有语音信号都是由 12 位的 ADC 采样得到，故用 matlab 设计位数为 12 位的正弦和余弦信号，深度设置为 4096，生成对应的 ROM 网表。

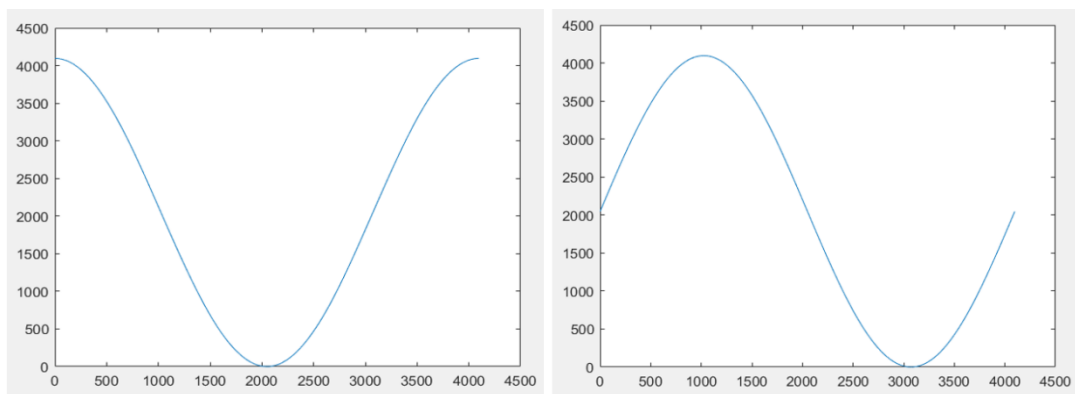


图 4.64 Matlab 生成正弦余弦 Rom 表

输出信号 CLK_OUT 的信号频率为：

$$f_{OUT} = K * f_{CLK} / 2^N \quad (20)$$

其中 N=32，fCLK=50MHZ，为了生成 fOUT=38KHZ 的信号计算得到 K=3264175，不需要进行相位偏移故 P_WORD=0。

自此 DDS 中各个参数已经确定完毕，进行对应模块编写可获得生成的 38KHZ 正弦和余弦信号。

调用乘法器 IP 将生成的正弦信号与余弦信号分别与带通滤波器输出的双边带信号相乘进行数字解调，然后低通滤波获得 I、Q 两路正交信号。

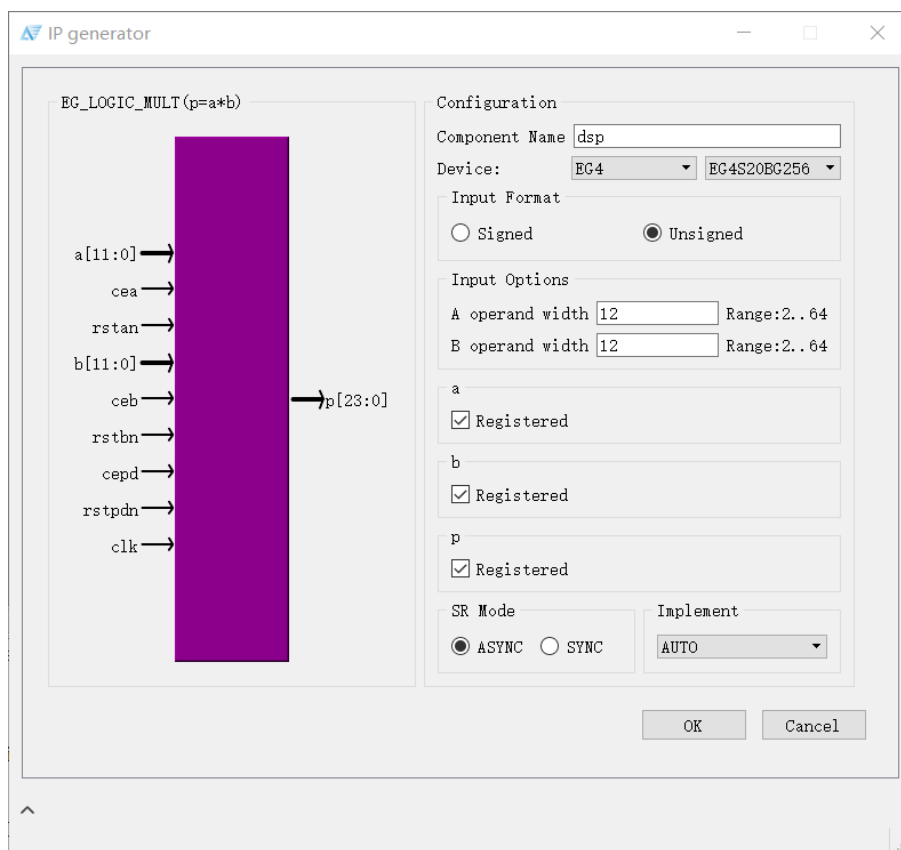


图 4.65 乘法器 IP

带通滤波器与低通滤波器设计如下图，幅值响应较好，符合要求。

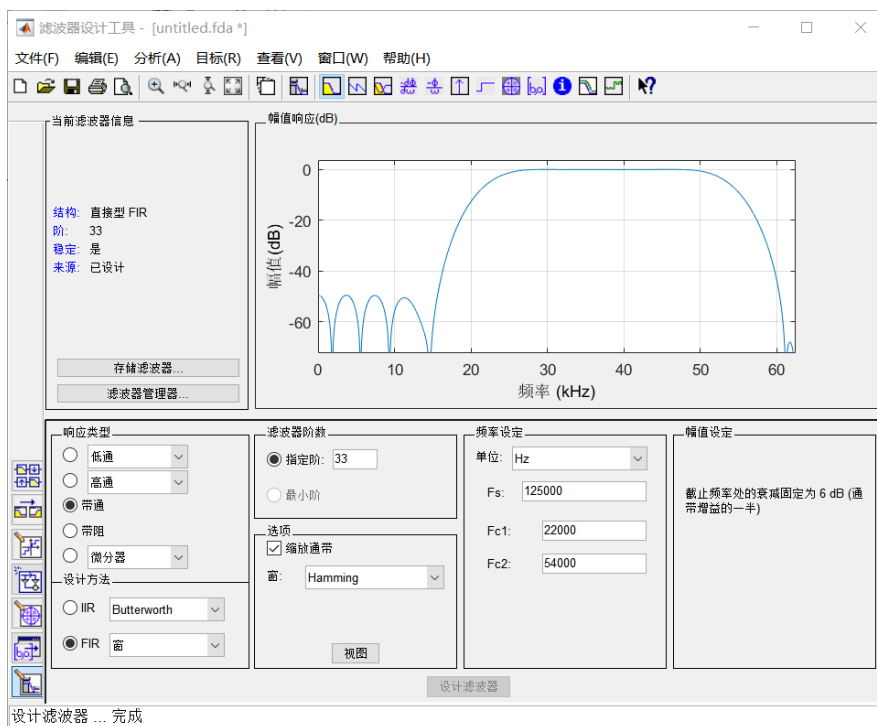


图 4.66 带通滤波器

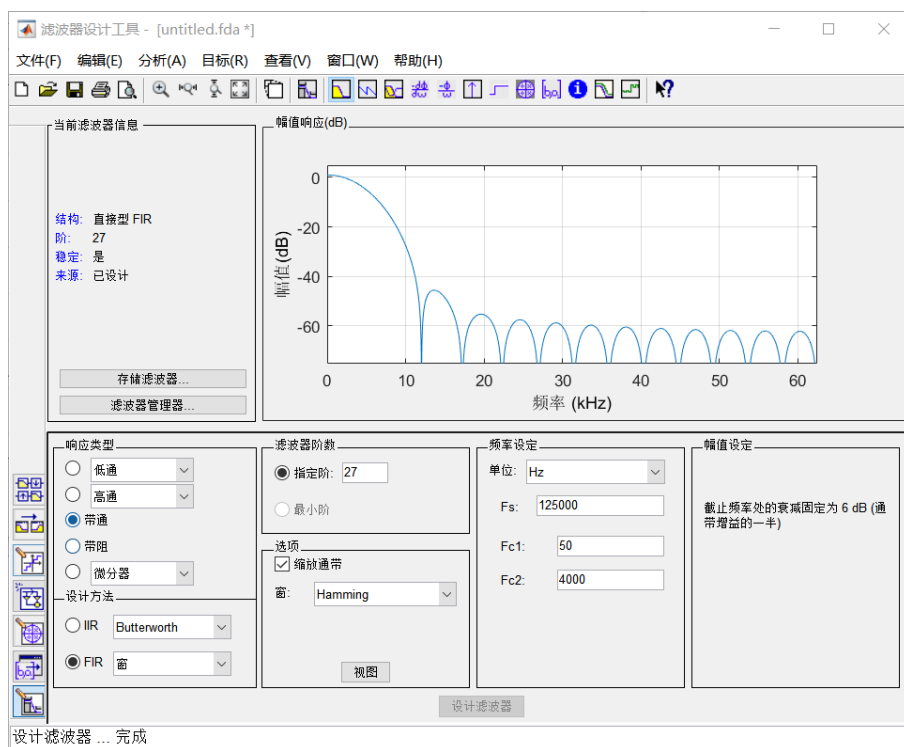


图 4.67 低通滤波器

低通滤波后的解调数据存入 Bram 中，利用 Bram editor 导出到 Matlab 后频谱分析得频谱图如下。

BRAM-EditorD:/Anlogic/TD5.0.30786/bin/td.exe

Physical ☒ Logic ☐ Bram list Hardware: EC4S20BC256

	Logic Name	Status	Width	Depth	Data Path
1	bram/inst	READABLE	12	2048	
2	auto_chipwatcher_0_logicbram	READABLE	51	8192	
3	RAM_DATA/ram_mem_al_u30	OCCUPIED	8	4096	
4	RAM_DATA/ram_mem_al_u20	OCCUPIED	8	4096	
5	RAM_DATA/ram_mem_al_u10	OCCUPIED	8	4096	

instance 1:bram/inst ☐ Dec ☒ Bin ☐ Hex

0x0000~0x03ff			0x0400~0x07ff		
	11~4	3~0		11~4	3~0
0x0000	10011111	1100	0x0400	10000000	0010
0x0001	10111111	1111	0x0401	00100010	0100
0x0002	11011111	0111	0x0402	00001000	0100
0x0003	11110101	1111	0x0403	00001010	0001
0x0004	01111111	1111	0x0404	00010000	1010
0x0005	11100110	0010	0x0405	00000010	0111
0x0006	00110011	1101	0x0406	00111010	1010
0x0007	00110111	0011	0x0407	00100101	1010

instance 1:bram/inst

图 4.68 Bram editor

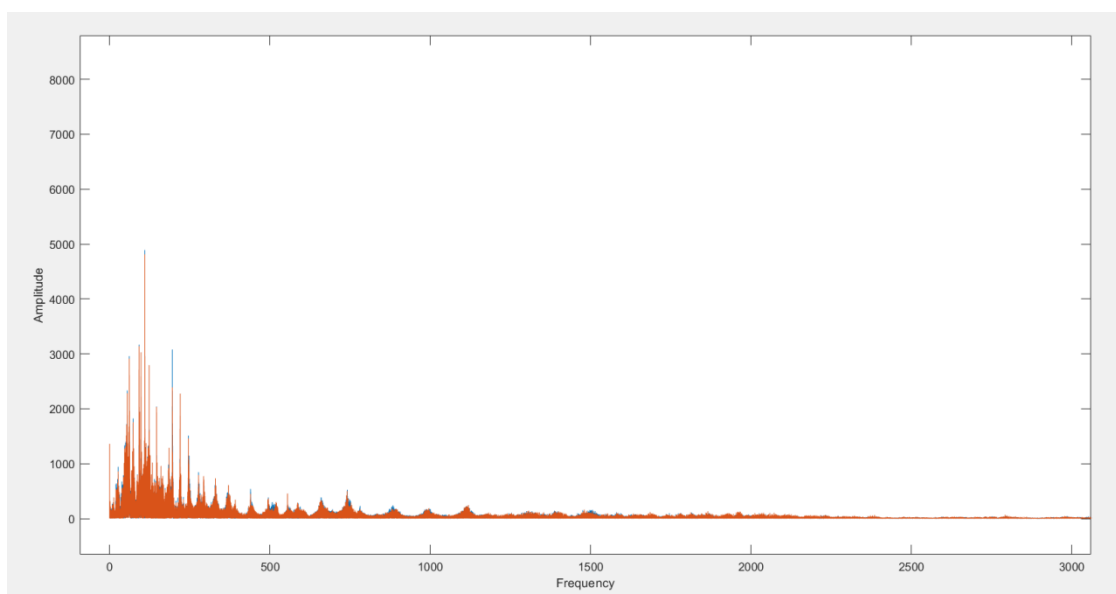


图 4.69 频谱图

调用 CORDIC IP 核, 对 I、Q 两路信号进行取模, 输出即为双边带信号的解调结果。CORDIC IP 核如下图所示。

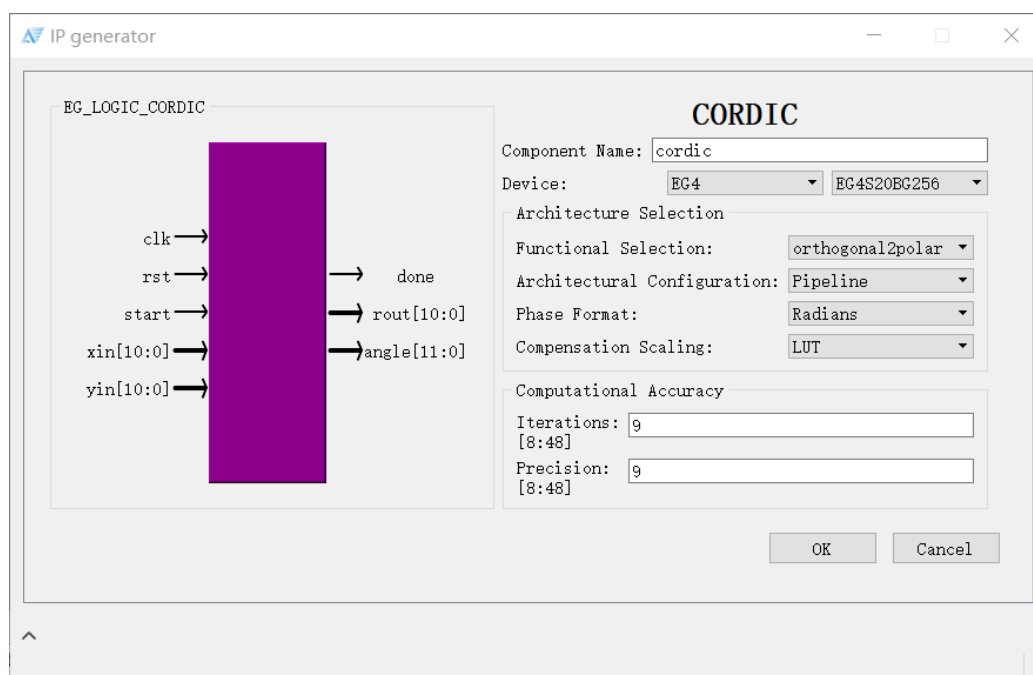


图 4.70 CORDIC IP 核

编译后查看 Physical report, 资源基本上全部用完。

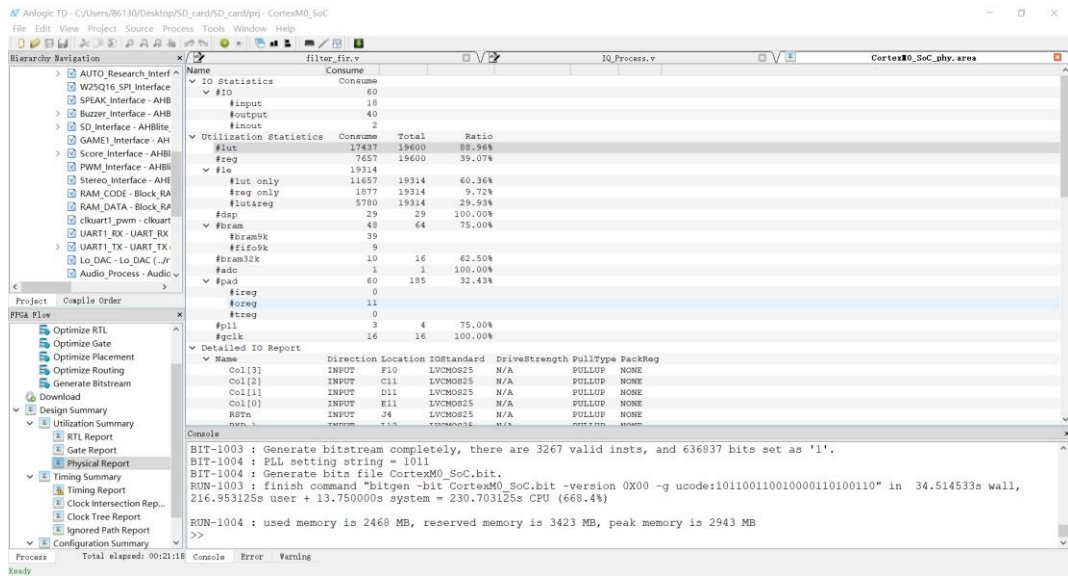


图 4.71 Physical report