电子科技大学
University of Electronic Science and Technology of China

求 / 实 / 求 / 真　　大 / 气 / 大 / 为

基于ZYNQ平台的
麦克风阵列声源定位系统

指导老师：张寅
小组成员：曾正启、朱凯杰、林时宁

目录/Contents

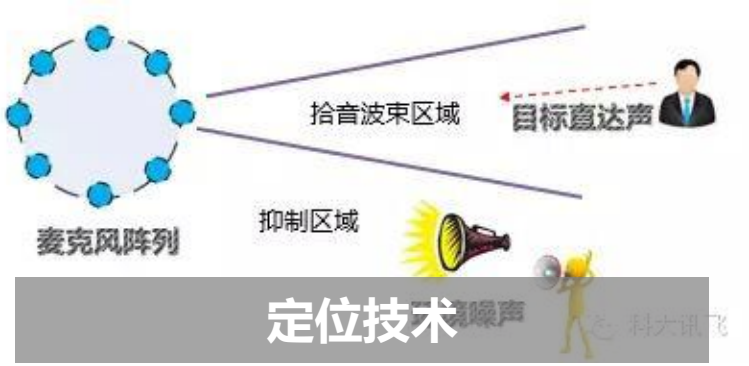
电子科技大学
University of Electronic Science and Technology of China

01 需求分析与总体规划

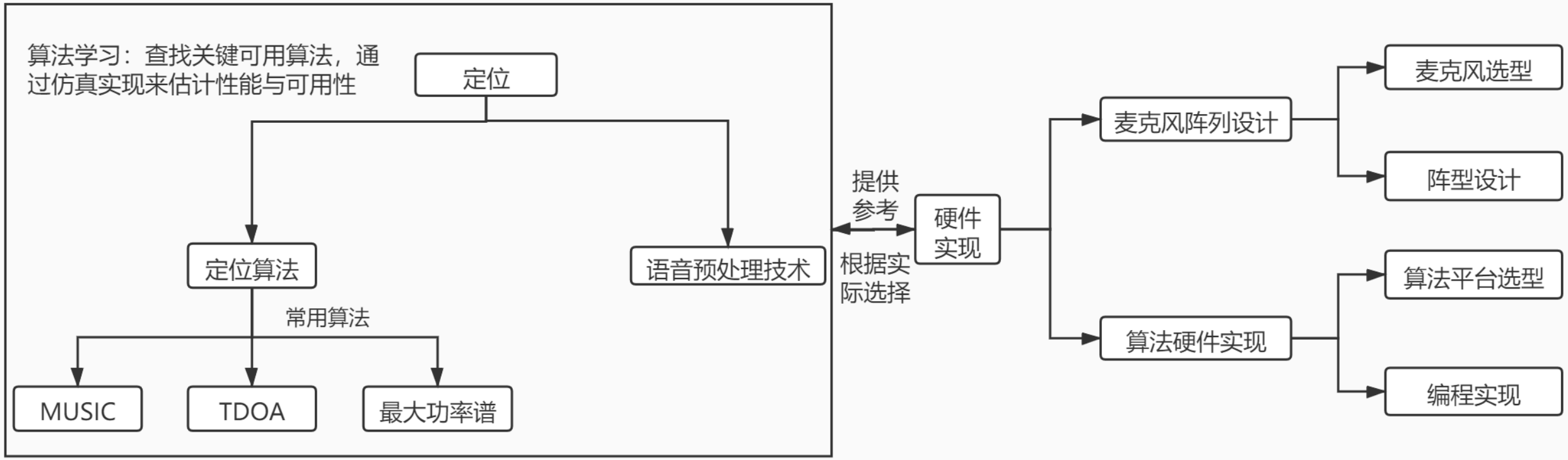
需求分析与总体规划

最终目的：

实现声源定位



需求分析总体规划


算法学习：查找关键可用算法，通过仿真实现来估计性能与可用性
定位
定位算法
常用算法
MUSIC
TDOA
最大功率谱
语音预处理技术
提供参考
根据实际选择
硬件实现
麦克风阵列设计
麦克风选型
阵型设计
算法硬件实现
算法平台选型
编程实现

02 算法选择

主要算法仿真一览

MUSIC算法

TDOA算法

基于最大输出功率的可控波束形成的定位技术

定位-TDOA算法

TDOA：时延估计+位置估计

时延估计：广义互相关时延估计法（GCC）

互相关函数在时延处取得最大值（CC）

互功率谱加权

广义互相关时延估计法（GCC）

$$R_{X_1X_2}(\tau)=\int_0^{\pi}\psi_{12}(\omega)\hat{G}_{X_1X_2}(\omega)e^{-j\omega\tau}d\omega$$

加权函数名	表达式
Roth	$\psi_{12}(\omega)=\frac{1}{\mathrm{G}_{X_1X_1}(\omega)}$
SCOT	$\psi_{12}(\omega)=\frac{1}{\mathrm{G}_{X_2X_2}(\omega)}$
PHAT	$\psi_{12}(\omega)=\frac{1}{\lvert\mathrm{G}_{X_1X_2}(\omega)\rvert}$
Eckart	$\psi_{12}(\omega)=\frac{\alpha\mathrm{G}_{s_1s_2}(\omega)}{\mathrm{G}_{n_1n_1}(\omega)\cdot\mathrm{G}_{n_2n_2}(\omega)}$
Hannon与Thomson	$\psi_{12}(\omega)=\frac{1}{\lvert\mathrm{G}_{X_1X_2}(\omega)\rvert}\cdot\frac{\lvert\gamma_{12}(\omega)\rvert^2}{[1-\lvert\gamma_{12}(\omega)\rvert^2]}$

定位-TDOA算法

麦克风个数：2个

信号形式：语音

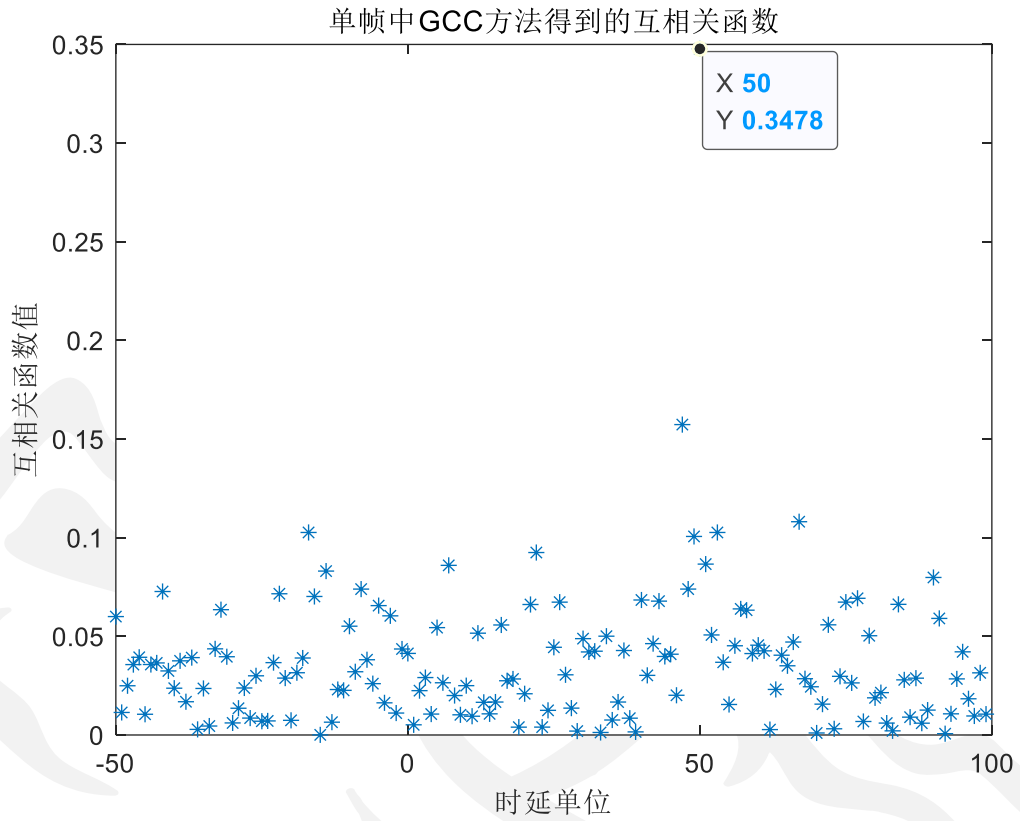
两信号时延差：50个单位

加权函数：PHAT

采样频率：8000Hz

采集点数：48100点

噪声：方差0.03高斯白噪声

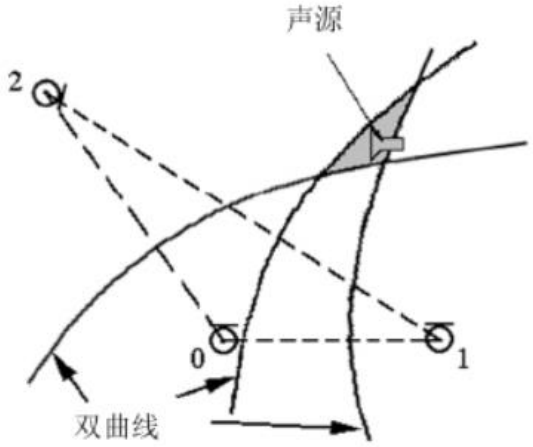


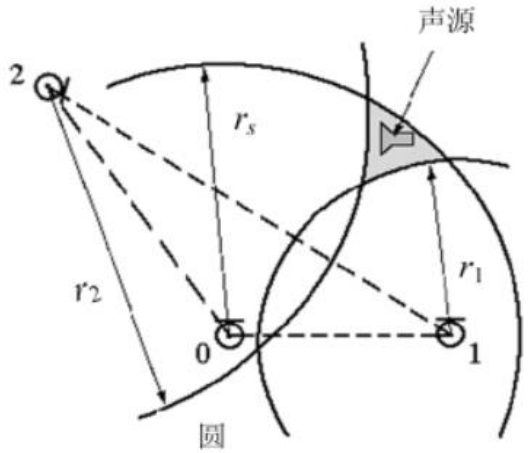
定位-TDOA算法

TDOA：时延估计+位置估计

位置估计：核心：在已知麦克风阵列坐标及麦克风间时延差的情况下，使得理想距离差与实际距离差误差最小。

误差估计方法：
最小二乘估计器：
双曲LS误差：误差图形为双曲面，代价函数最小化得到的声源位置为离双曲线最近的点。
球面LS误差：误差图形为球面，代价函数最小化得到的声源位置后为到球面最近的点。





定位-TDOA算法

麦克风个数：9个（立方体阵+原点）

阵元大小：长宽高0.4m

采集点数：48000点

时延估计算法：GCC

采样频率：8000Hz

噪声：0.05方差高斯白噪声

(0.4,0.6,0.5) → 0.3889 0.5808 0.4834

(0.8,0.9,0.7) → 0.8156 0.9226 0.7063

(1.0,1.0,1.2) → 0.9877 0.9877 1.1831

(1.4,1.5,1.3) → 1.4184 1.5237 1.3131

03 硬件实现路线

硬件实现路线

第一步

实现麦克风阵列音频采集

第二步

实现单声源方向定位

第三步

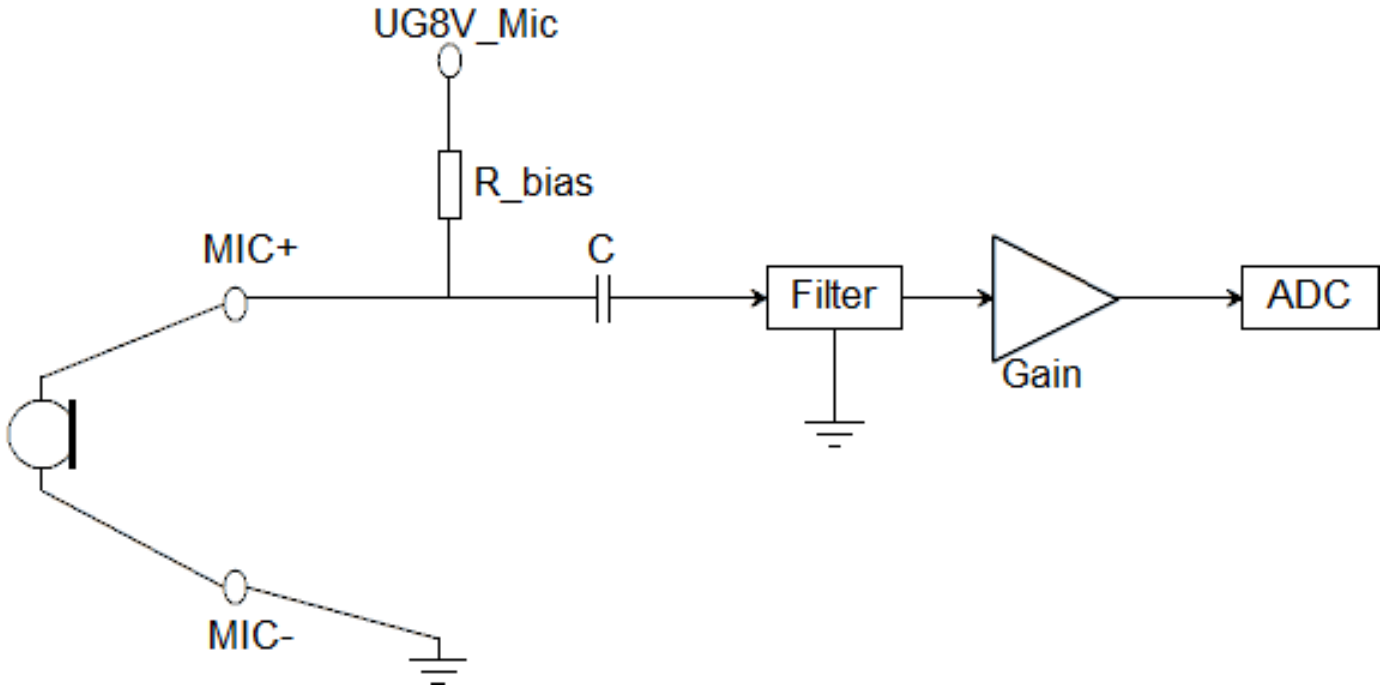
实现多声源定位（展望）

04 系统硬件实现

4.1 电路设计

1.麦克风阵列结构选型

(1). 采用模拟麦克风加模拟前端结构，通过多通道ADC采样。

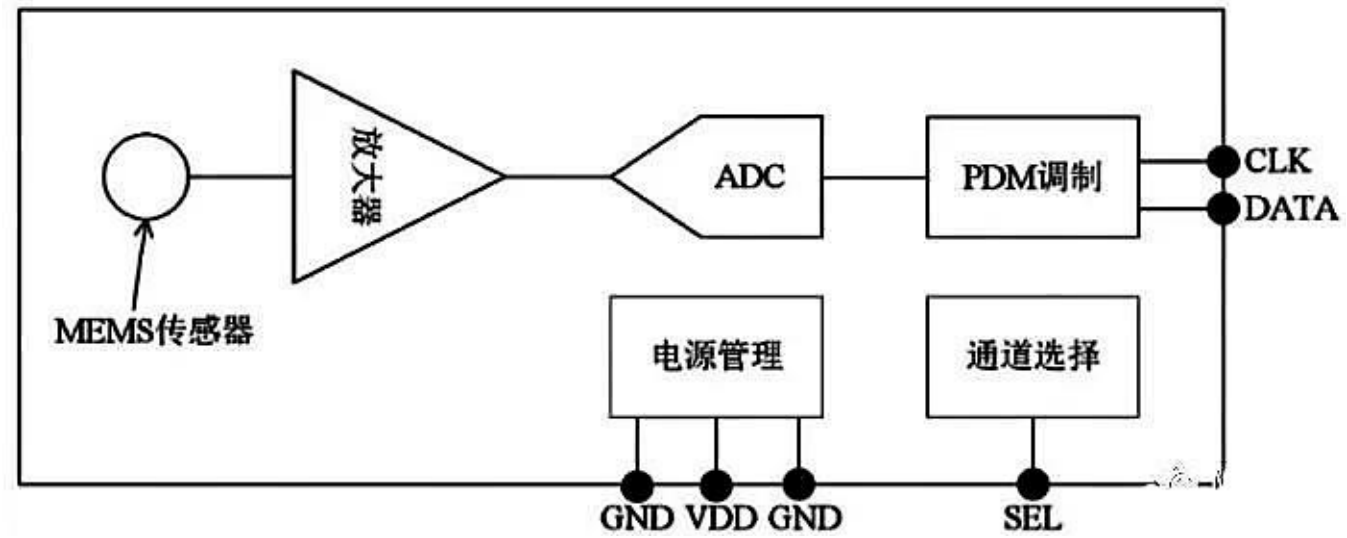


虽然其具有高SNR的优势，但因为滤波器，放大器等模块的匹配工作相对繁琐，对于采用阵列架构的语言信号采集系统来说，并非最理想的选择。

4.1 电路设计

1.麦克风阵列结构选型

(2). 采用数字MEMS麦克风



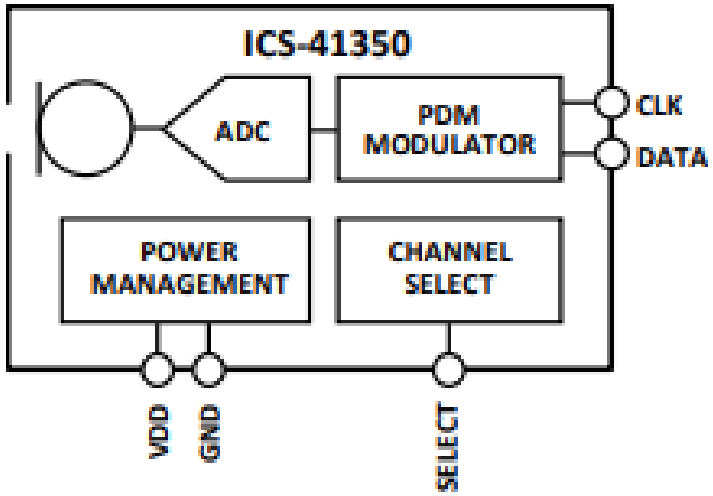
采用MEMS技术的麦克风的一致性更为优良，这样可以大大简化电路设计和电路板布局等，同时仍然可以达到与模拟麦克风类似或更好的信号质量,较为适合运用在采用阵列架构的语言信号采集系统中。

4.1 电路设计

2.麦克风选型

综合成本，性能等因素的考虑，我们选用具有PDM接口的数字硅麦ICS-41350，其硬件特性主要有：

SPEC	LOW-POWER MODE	STANDARD MODE	HIGH PERFORMANCE MODE
Sensitivity	−26 dB FS ±1 dB	−26 dB FS ±1 dB	−32 dB FS ±1 dB
SNR	63 dBA	64 dBA	64 dBA
Current	185 µA	430 µA	650 µA
AOP	120 dB SPL	120 dB SPL	126 dB SPL
Clock	400 – 800 kHz	1.0 – 3.3 MHz	4.1 – 4.8 MHz



其采用四阶 Σ-Δ 调制，在驱动时钟在4.8Mhz时提供高性能模式，信噪比可达64dB。

4.1 电路设计

2.麦克风选型

ICS-41350 PDM接口时序较为简单,便于开发:

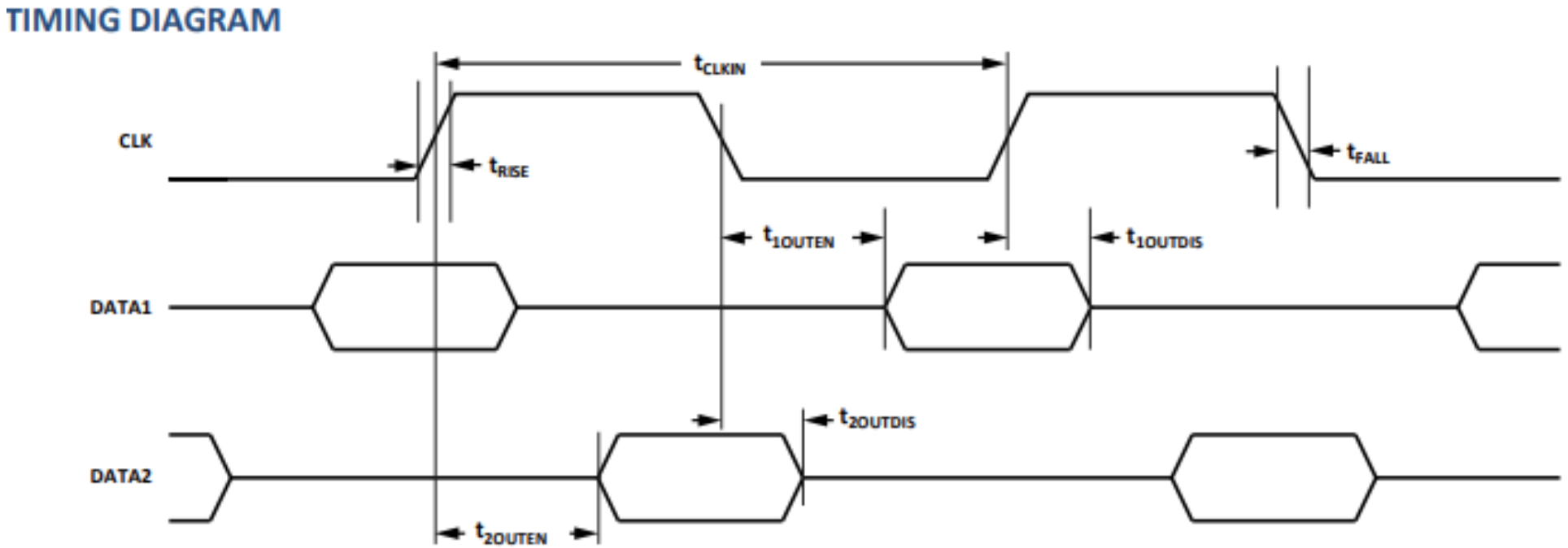


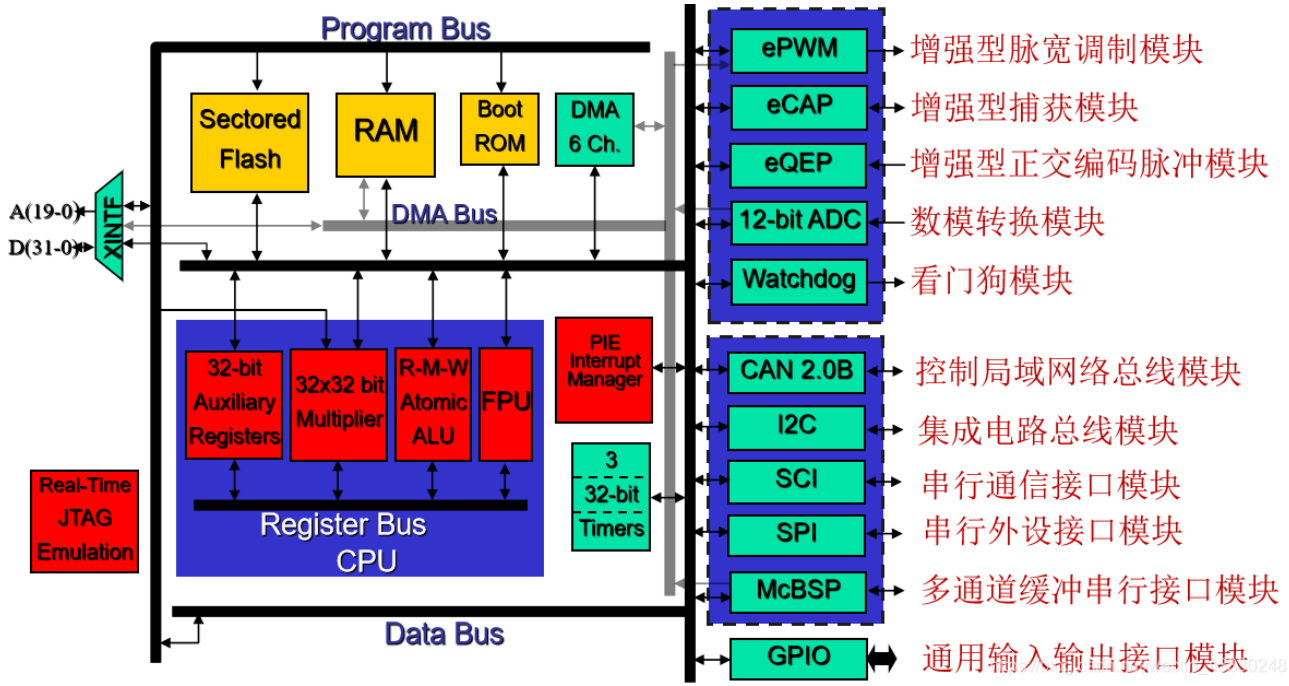
Figure 1. Pulse Density Modulated Output Timing

4.1 电路设计

3.算法实现平台选型

(1). 采用DSP信号处理平台

利用DSP实现数据采集，数据分析、计算等功能；外挂的MCU完成管理、通信、人机接口等功能。两者通过双端口RAM或者并/串口等方式进行数据交互。由于DSP较难实现多通道并行数据采集, 并且其对降采样滤波器实现较为困难,同时项目成员都未接触过DSP开发,所以弃用此方案。

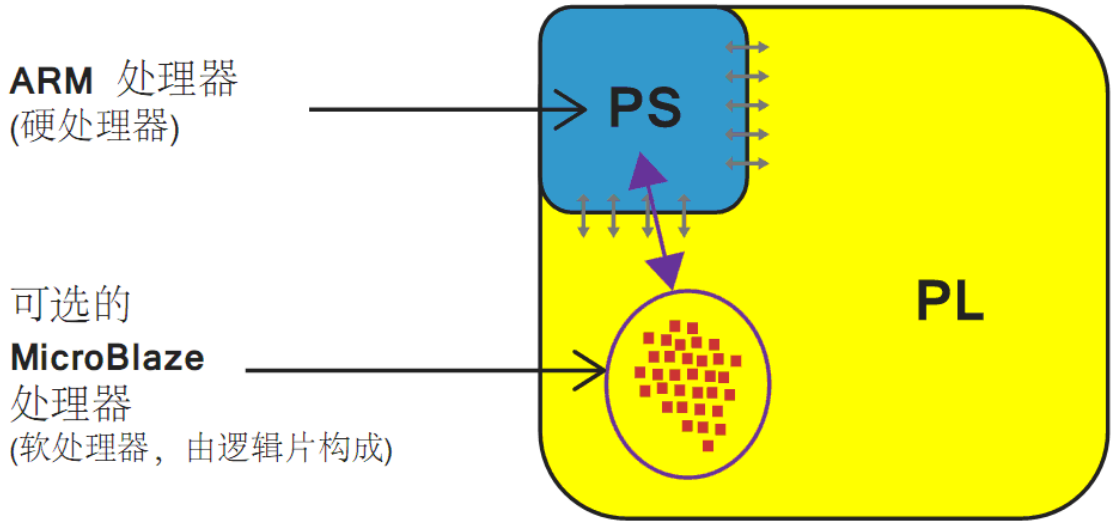


常见DSP结构框图

3.算法实现平台选型

(2). 选用ZYNQ异构FPGA平台

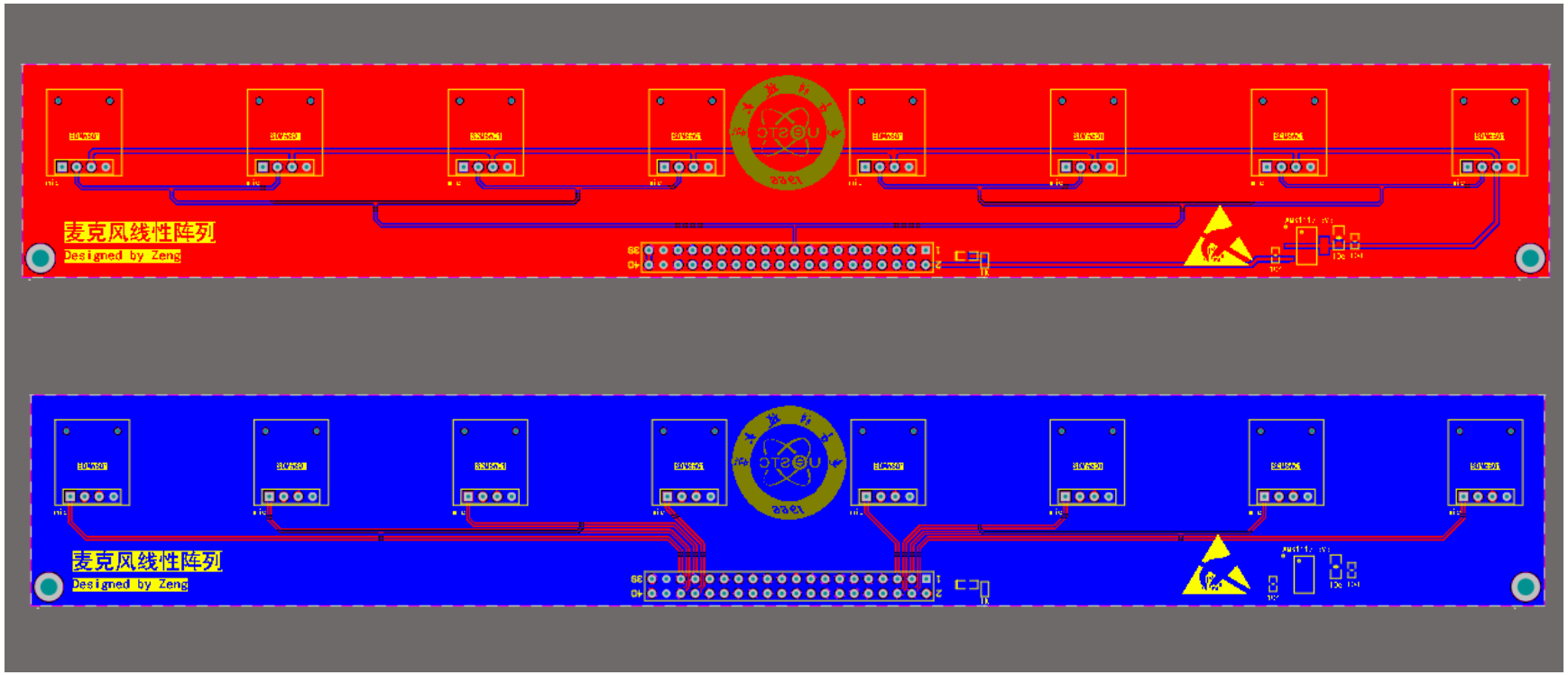
Zynq®-7000 SoC 系列集成 ARM® 处理器的软件可编程性与 FPGA 的硬件可编程性，不仅可实现重要分析与硬件加速，同时还在单个器件上高度集成 CPU、DSP、ASSP 以及混合信号功能。 Zynq®-7000 SoC FPGA+ARM的架构十分灵活,可以同时通过FPGA实现信号处理的硬件加速和通过ARM处理器实现人机交互等控制功能, 并且有项目成员接触过其开发,所以较为适合本项目的硬件实现。



ZYNQ结构框图

4.1 电路设计

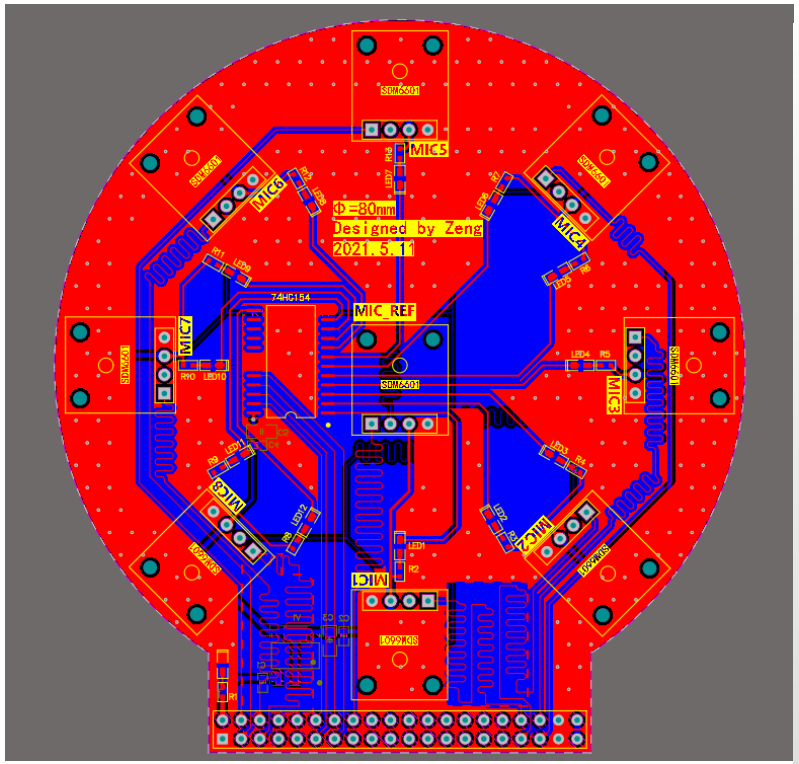
4.麦克风阵列PCB设计1（8元麦克风线阵）

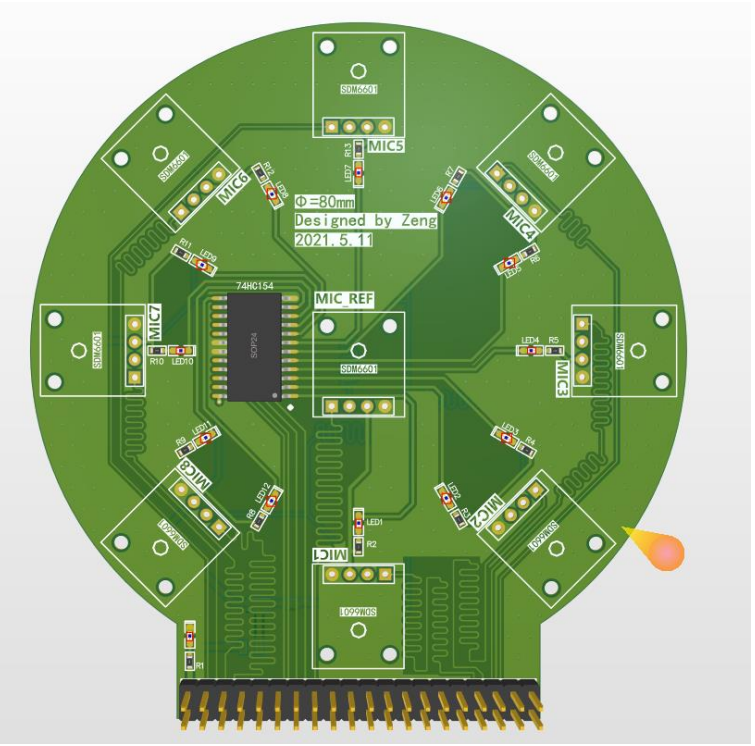


阵元间距： 25mm
接口：黑金ZYNQ7020开发板定义
时钟线进行T形拓扑等长

4.1 电路设计

4.麦克风阵列PCB设计2（9元麦克风圆阵）



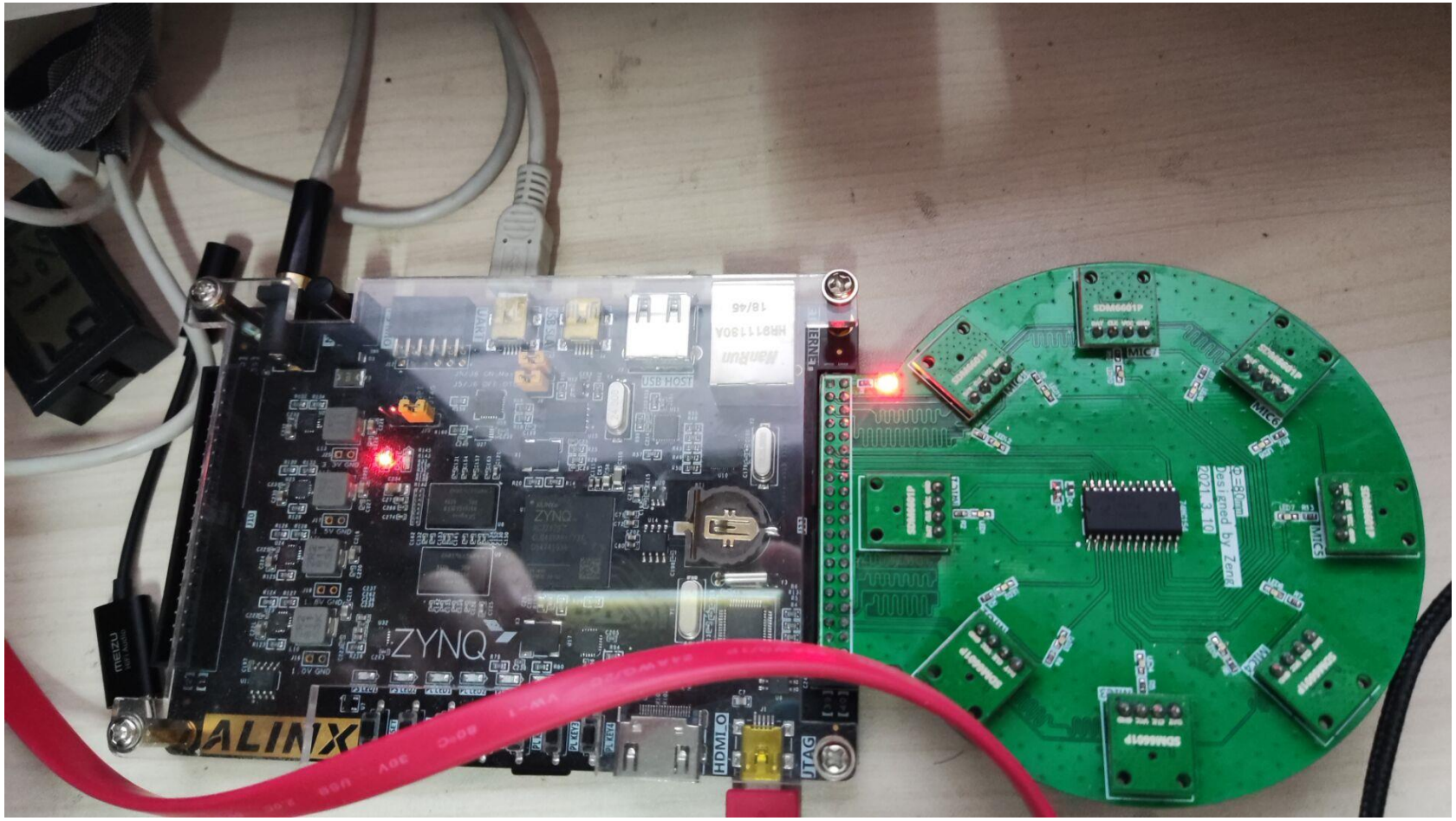


圆阵直径：80mm
接口：黑金ZYNQ7020开发板定义
时钟线进行Flyby拓扑等长

因为线阵只能实现一维DOA估计,为了实现二维定位,我们决定采用平面阵的设计,即**设计2**

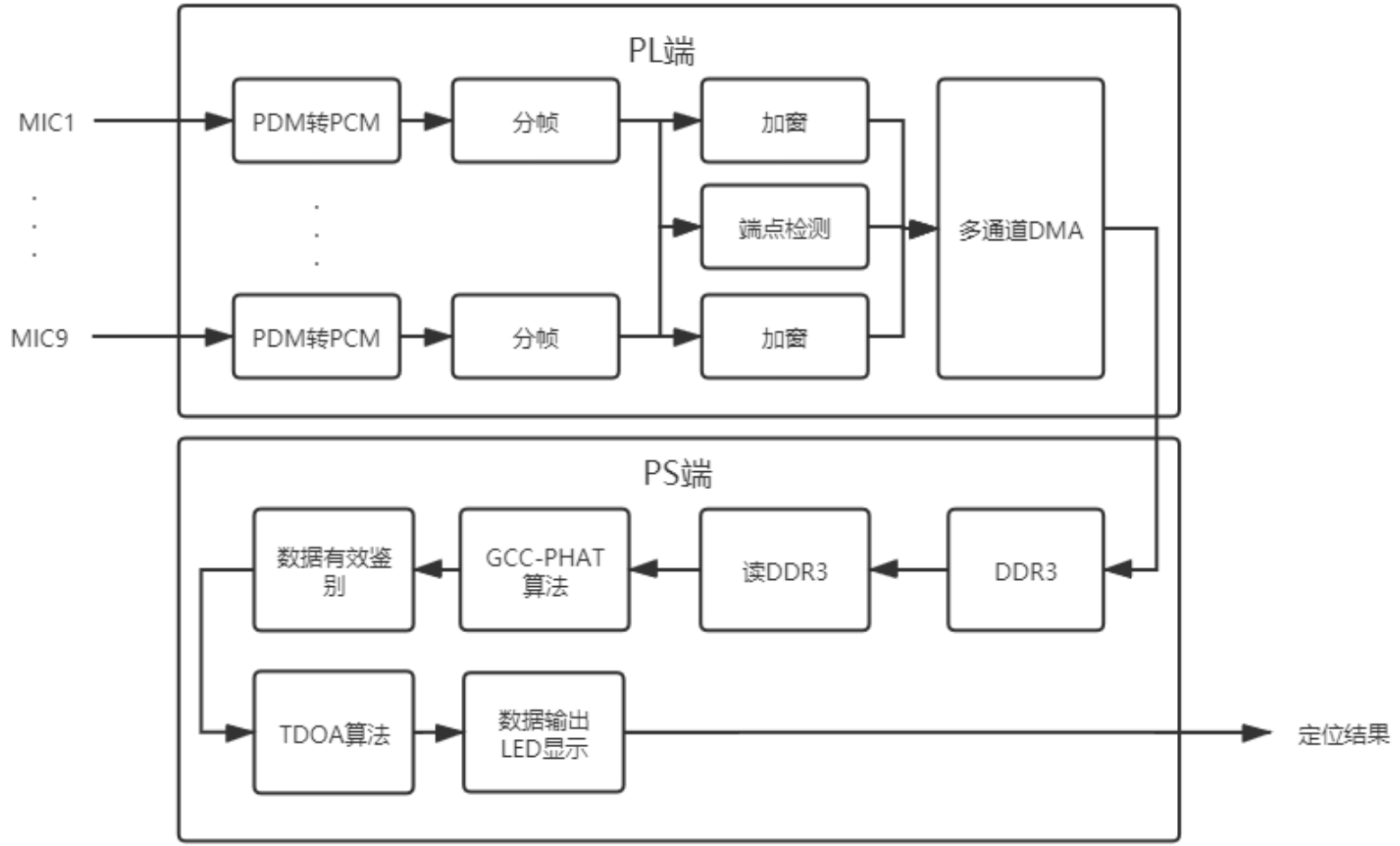
4.1 电路设计

5. 系统搭建实物图



4.2 硬件实现方案一（舍弃）

PL端进行数据采集,PS端进行GCC-PHAT和TDOA的算法实现和数据输出



FPGA硬件结构框图

4.2 硬件实现方案一（舍弃）

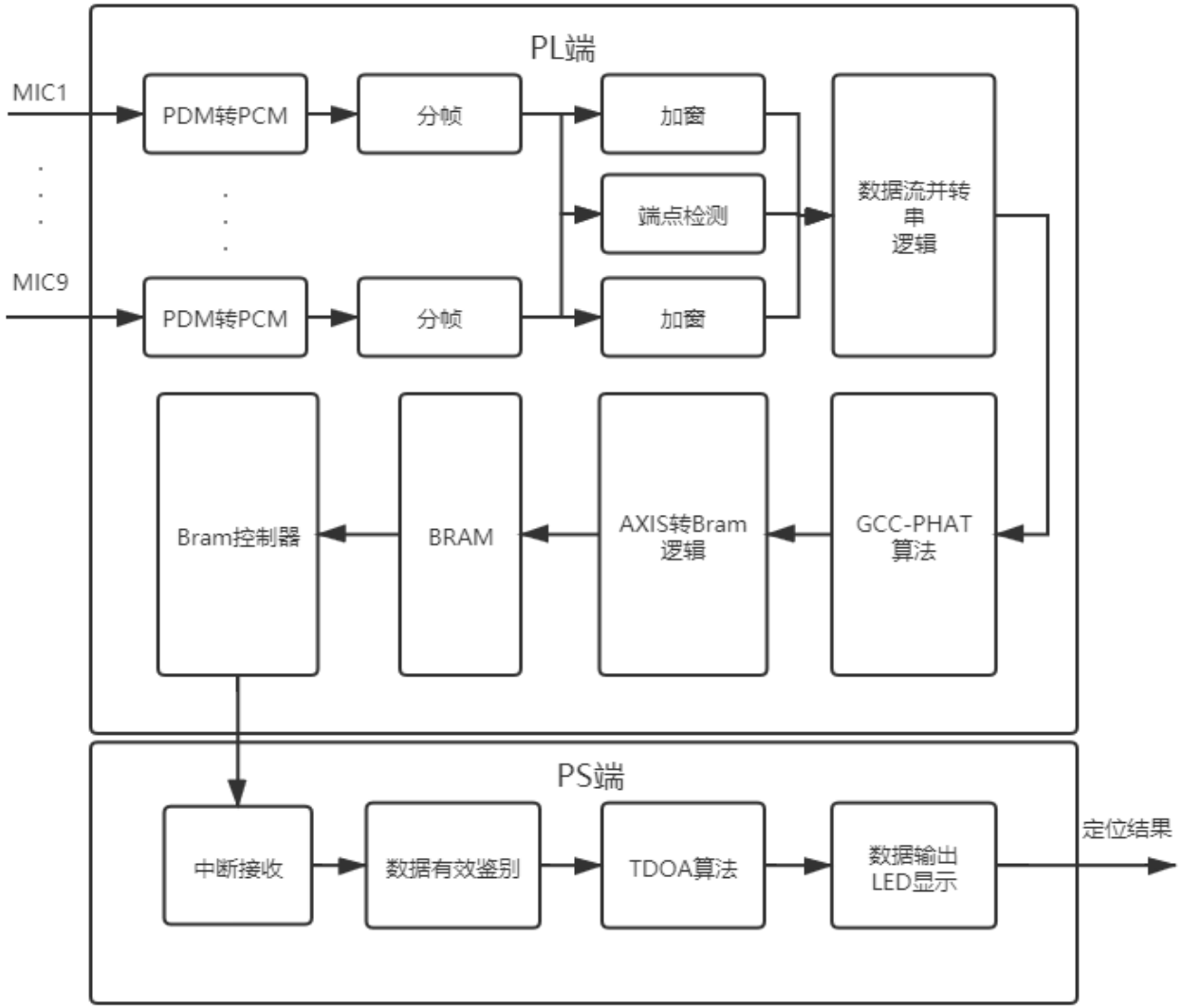
在实际实现的过程中,我们发现了方案一的以下不足:

1. PS端进行较长数据的FFT和IFFT等算法较为耗时，系统实时性难以提高。
2. 多通道数据流的DMA传输由于中断频率较高而出现不稳定的情况，导致系统整体稳定性不佳。
3. 由于PS端一直处于高负载状态,ZYNQ芯片发热较为严重。
4. 未有效利用FPGA资源实现算法加速的效果。

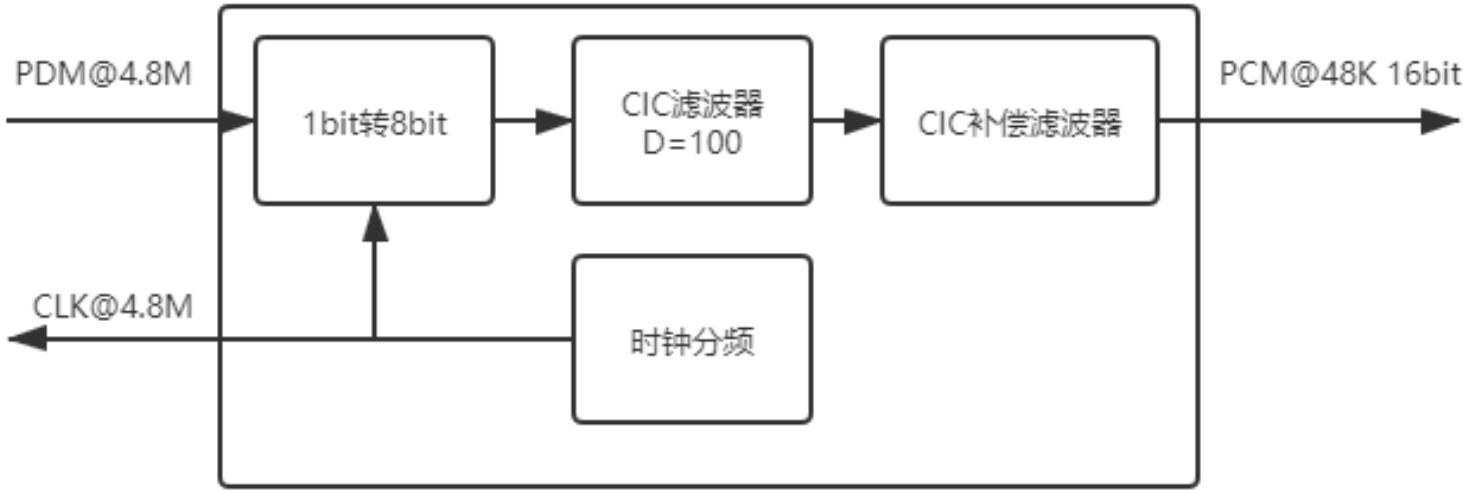
故**舍弃**此方案。

4.2 硬件实现方案二

PL端进行数据采集和GCC-PHAT算法实现，PS端进行TDOA估计和数据输出



4.2.1 PDM转PCM模块



麦克风本底噪声 :94dB SPL - 64dB = 30dB SPL

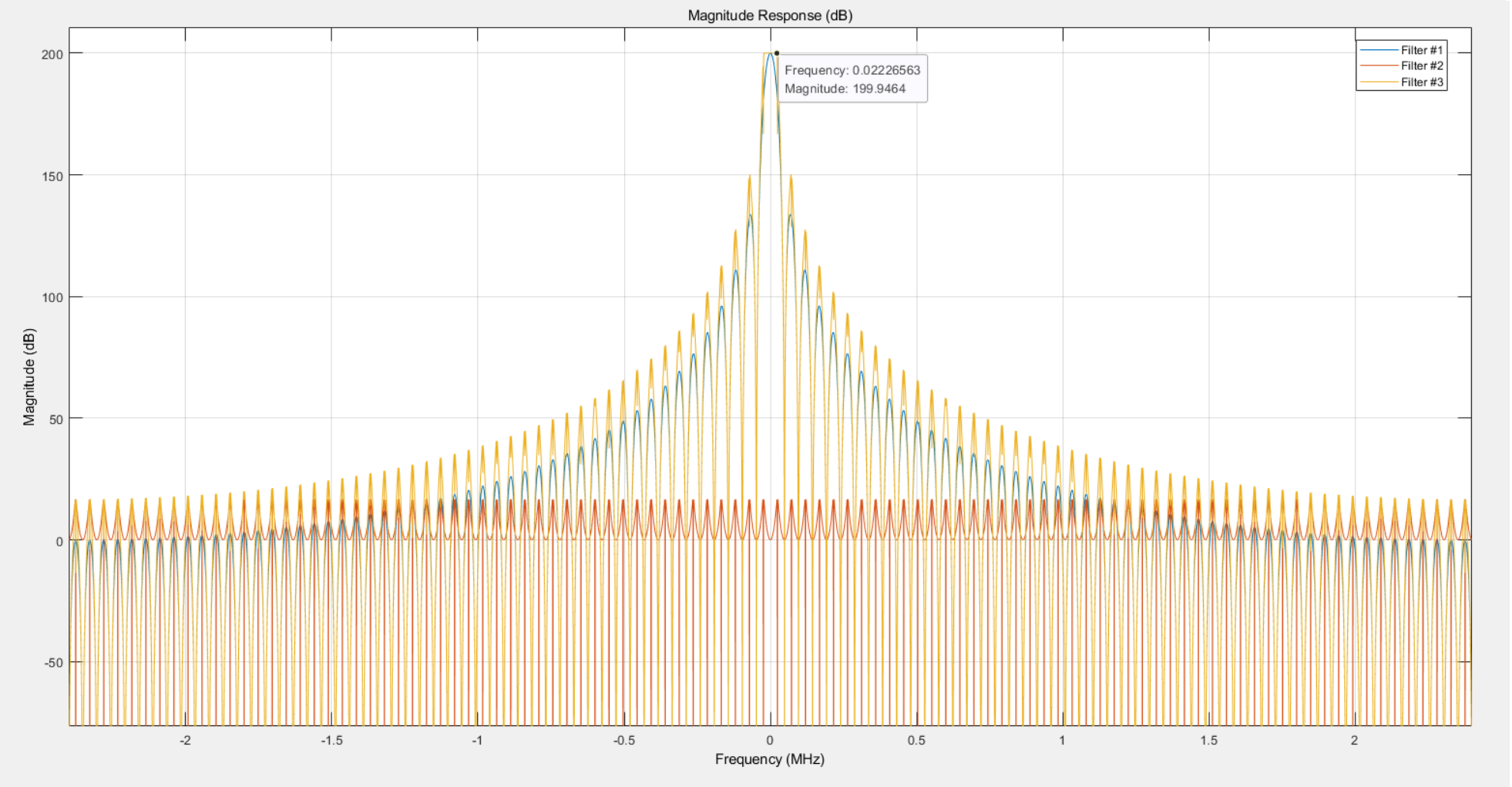
麦克风最大动态范围:126dB SPL - 30dB SPL = 96dB

降采样滤波器输出截位16位，动态范围：20lg(2^16) = 96.3dB

理论上可以完全量化数字麦克风输出，同时位宽较低, 可以降低后级位宽,进而降低FPGA的资源使用率。

4.2.1 PDM转PCM模块

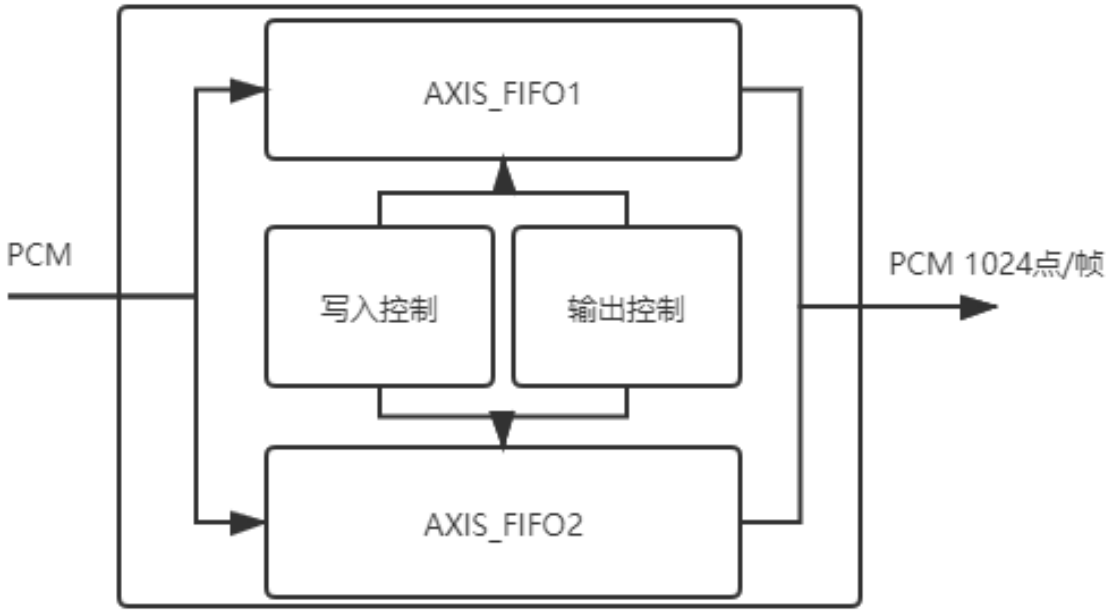
滤波器频谱特性仿真:



4.2.2 分帧模块

为了保证语音信号平稳性，一帧信号的时间窗长度在10 -30ms之间。采样率为48kHz选择帧长是1024点,即21.3ms 。为了保证统计特征的连续性，各帧之间进行50%的重叠，即帧移512点。一帧内的信号可以近似认为是平稳的，且各帧间的过渡平滑。

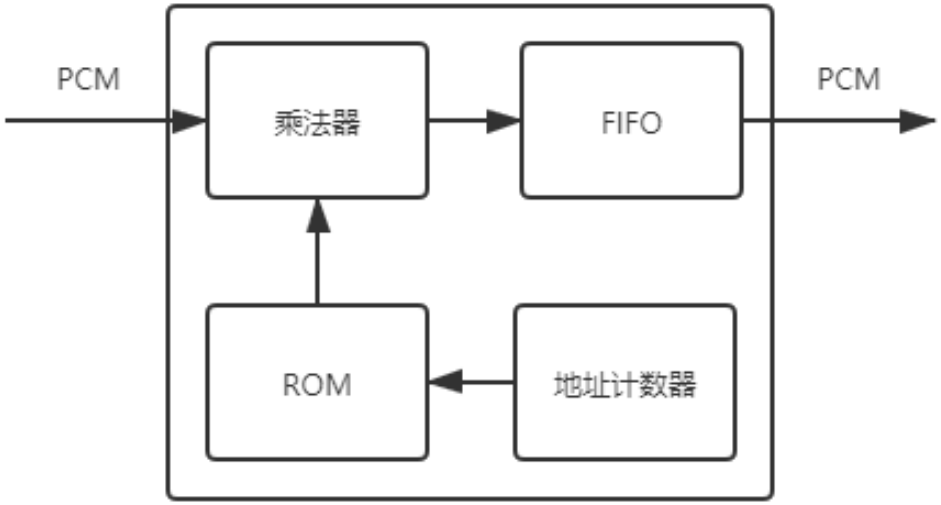
FIFO2相较FIFO1延时512点写入,双FIFO写满后立即读出,从而实现1024点分帧,帧移512点。硬件实现如下图：



4.2.3 加窗模块

由于直接对信号截断会产生频谱泄露，为了改善频谱泄露的情况需要给信号加非矩形窗。汉明窗主瓣峰值与第一个旁瓣峰值衰减可达43db,加汉明窗能够有效改善频谱泄露的问题。

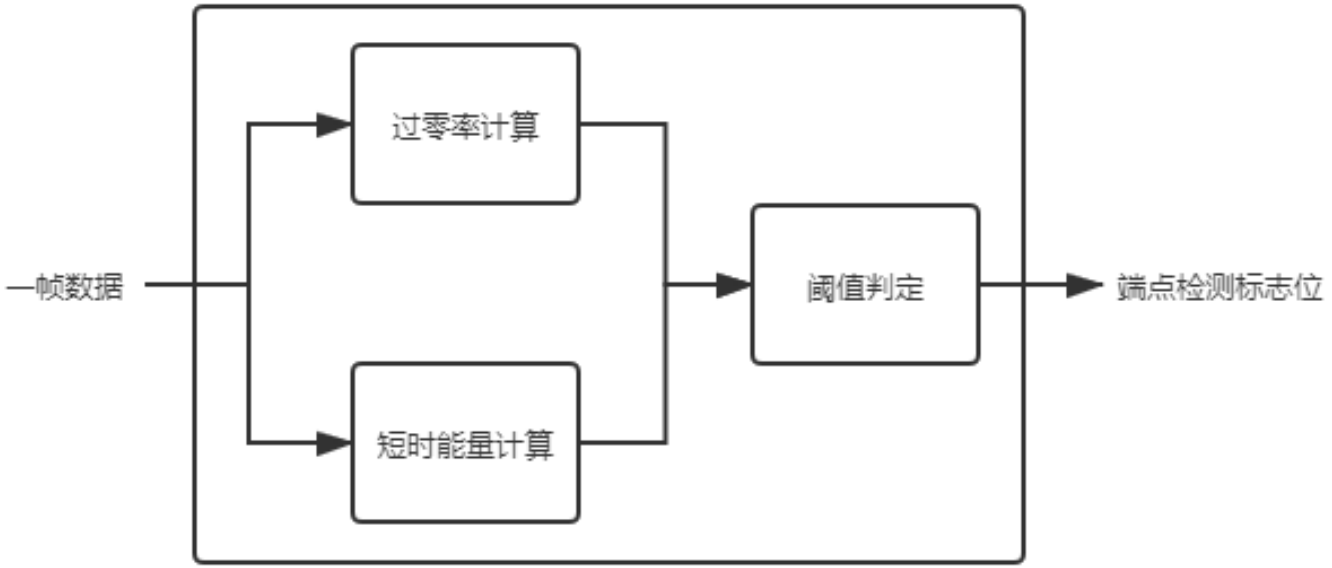
汉明窗值先由Matlab计算出并量化后导入FPGA的ROM中。根据接收数据量计数器产生ROM地址,读取ROM中对应的窗函数值,和信号值相乘并截位后写入FIFO中。FIFO写满后自动读出,即输出了一帧加窗后的数据。 硬件实现如下图:



4.2.4 端点检测模块

音频端点检测一般用于鉴别音频信号当中的音频出现和音频消失,从而抑制无声段的噪音干扰,提高算法输入数据的有效性,降低计算的数据量。

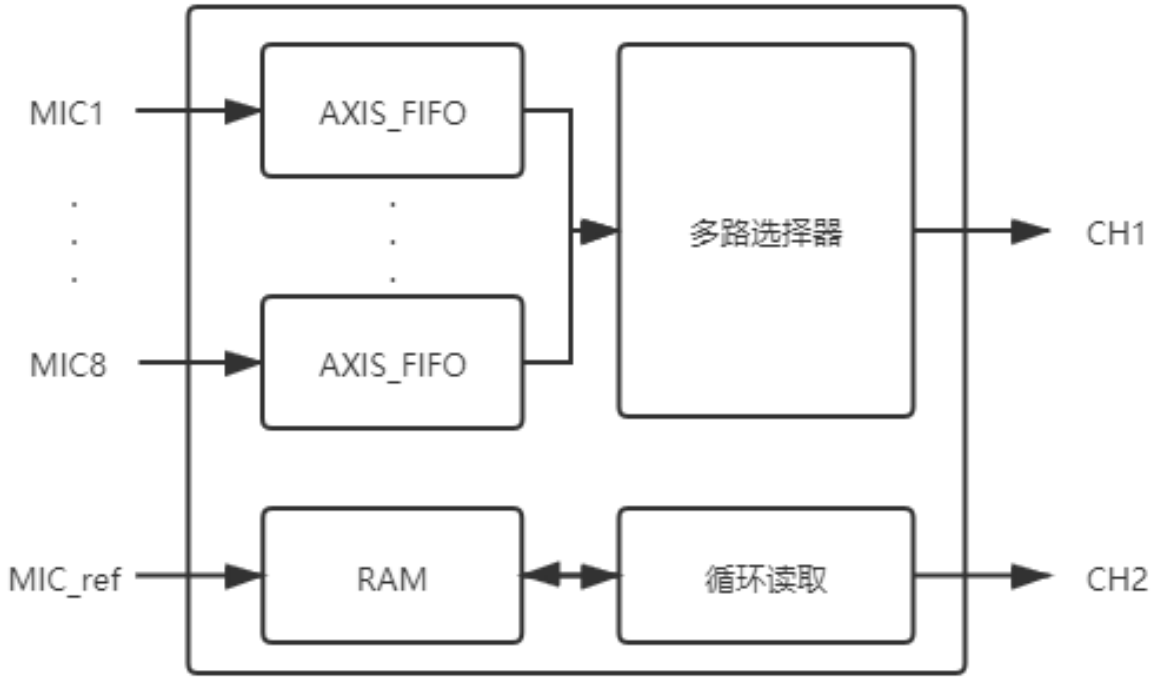
在信噪比不是很低的情况下，因为音频信号能量绝大部分包含在低频带内，而噪音信号通常能量较小且含有较高频段的信息，所以音频片段的短时能量相对较大，而过零率对较小，而非音频片段恰好相反。故而可以通过测量语音信号的这两个特征判断音频片段与非音频片段。硬件实现如下图：



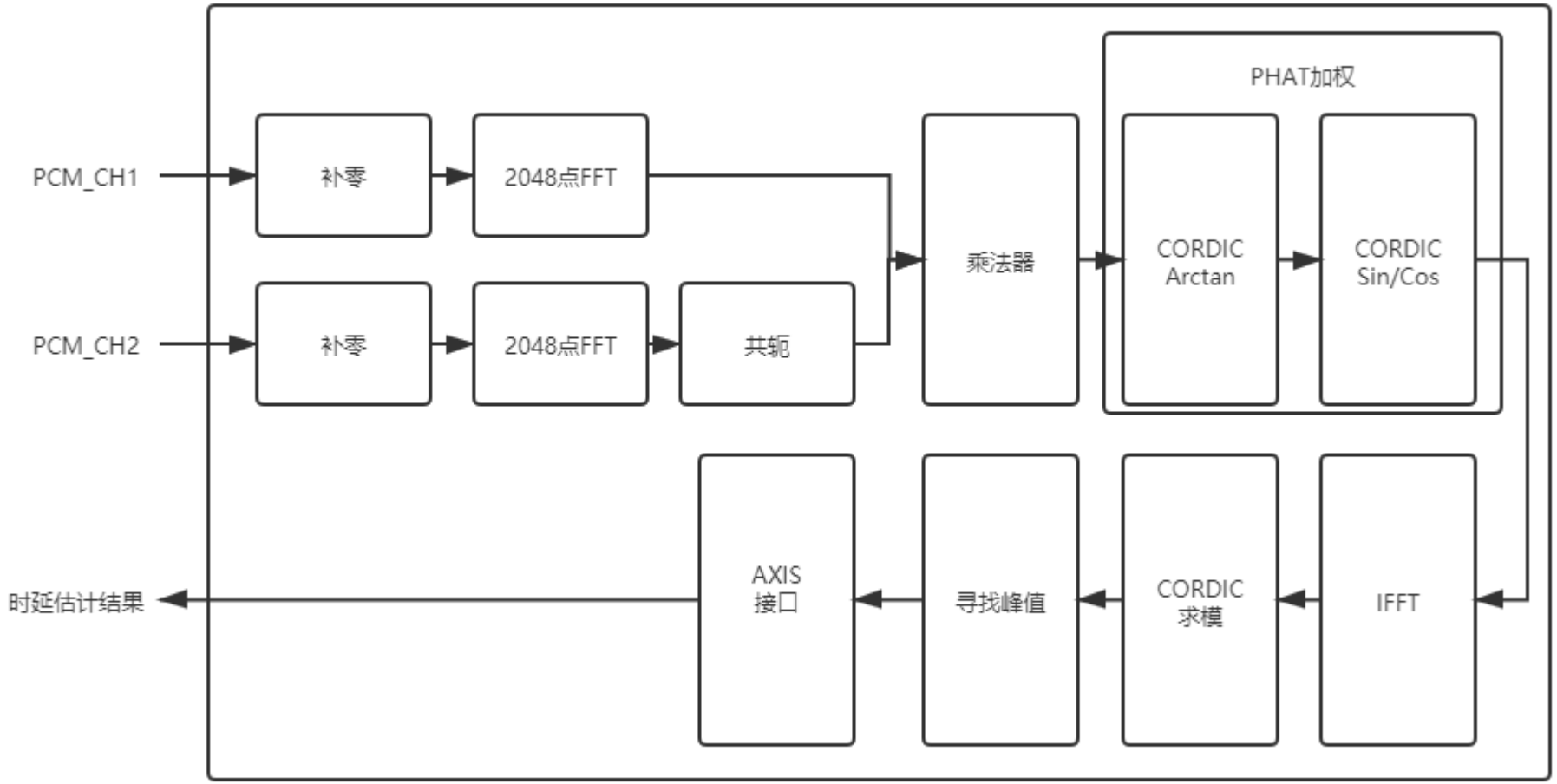
4.2.5 数据流并转串模块

由于GCC-PHAT算法占用了近1/3的FPGA资源,为了实现多通道的时延估计必须以时间换空间,即将并行的数据流转为串行输入到GCC-PHAT算法模块中。

8个FIFO缓存8通道的麦克风数据,并通过多路选择器逐个读出。作为标准的麦克风的数据写入开辟的BRAM中,并通过自定义逻辑循环读出8次数据。 硬件实现如下图:



4.2.6 GCC_PHAT算法模块



4.2.6 GCC_PHAT算法模块

基于广义互相关函数的时延估计算法引入了一个加权函数，对互功率谱密度进行调整，从而优化时延估计的性能。根据加权函数的不同，广义互相关函数有多种不同的变形，其中广义互相关-相位变换方法（GCC_PHAT）方法应用最为广泛。

算法实现步骤概括：

1.信号补零到2048点并进行FFT运算

2.一路信号求共轭

3.相乘截位求互功率谱

4.通过CORDIC算法转化为角度进行快速PHAT加权，避免使用除法器

5.截位进行IFFT运算

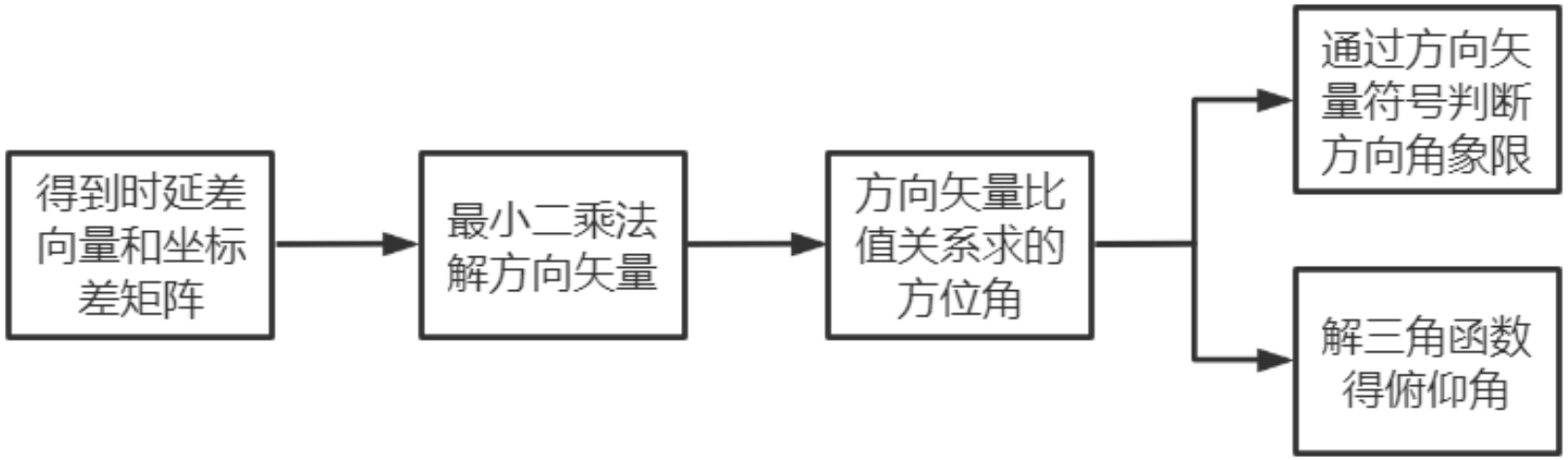
6.CORDIC算法求模

7.寻找峰值位置

8.通过输出逻辑转化为AXIS数据流形式

4.2.7 TDOA算法C语言实现

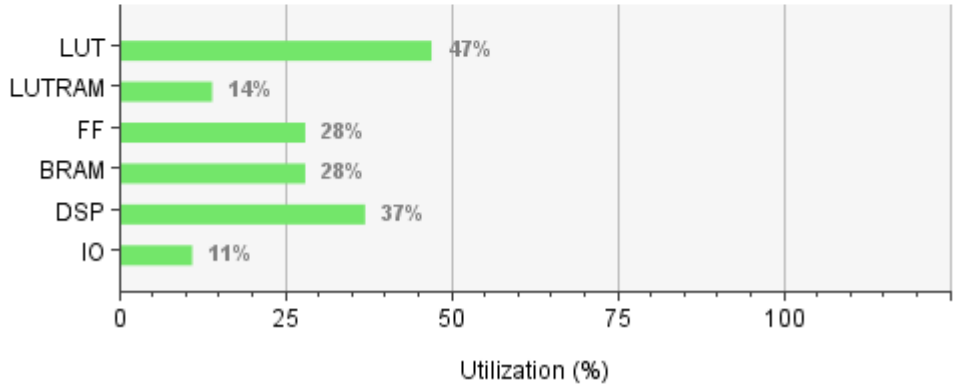
算法框图:



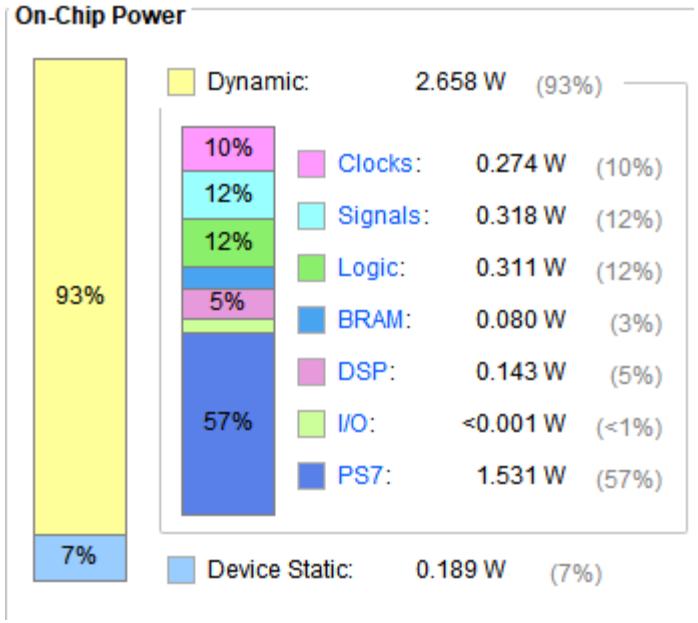
4.3 硬件使用报告

资源使用率：

Resource	Utilization	Available	Utilization %
LUT	25018	53200	47.03
LUTRAM	2471	17400	14.20
FF	30227	106400	28.41
BRAM	38.50	140	27.50
DSP	82	220	37.27
IO	14	125	11.20



功耗：



经过架构和逻辑优化，FPGA资源使用率不到50%,最大功耗不到3W，具有一定的实际应用价值

05 测试结果

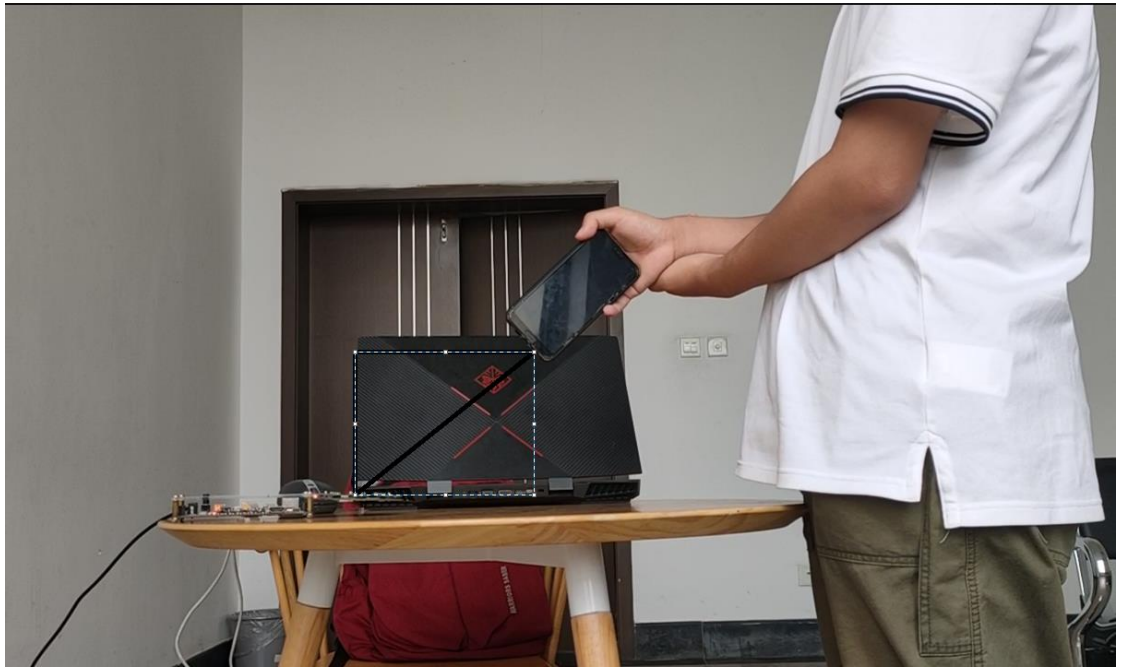
测试结果（静止）

实际角度（粗略估计）：
方向角270度
俯仰角30度

Yaw:267Pitch: 28E
Yaw:267Pitch: 28E
Yaw:269Pitch: 28E
Yaw:267Pitch: 31E
Yaw:269Pitch: 35E
Yaw:267Pitch: 31E
Yaw:269Pitch: 33E
Yaw:270Pitch: 24E
Yaw:272Pitch: 36E
Yaw:269Pitch: 35E
Yaw:269Pitch: 31E
Yaw:269Pitch: 35E
Yaw:269Pitch: 31E
Yaw:266Pitch: 34E
Yaw:269Pitch: 28E
Yaw:272Pitch: 31E

测试结果（静止）

实际角度（粗略估计）：
方向角180度
俯仰角40度



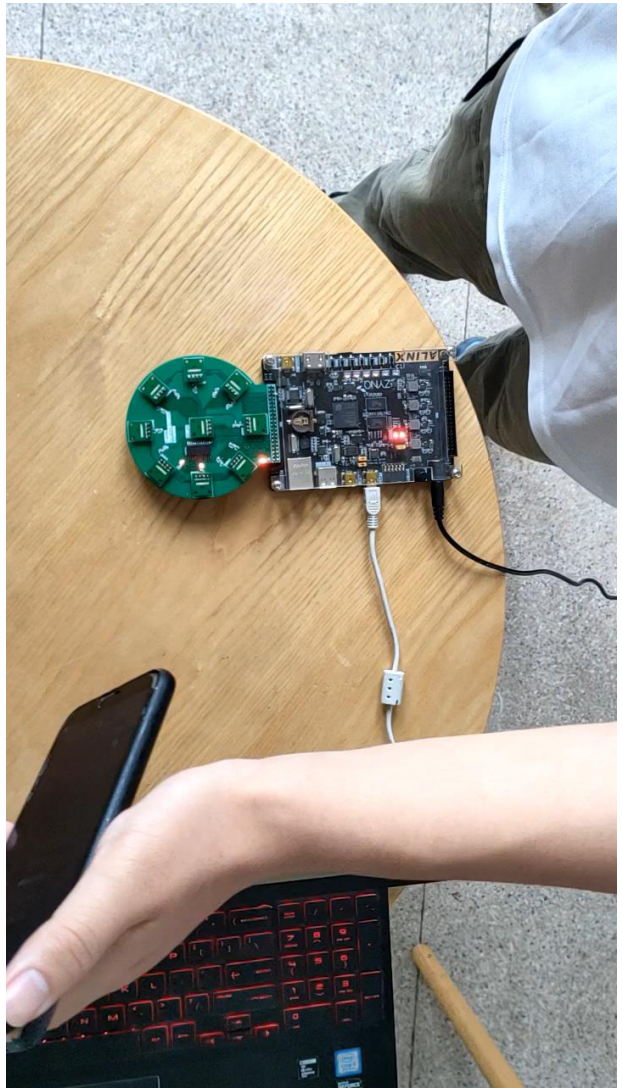
Yaw:179Pitch: 39E
Yaw:182Pitch: 36E
Yaw:186Pitch: 45E
Yaw:179Pitch: 43E
Yaw:185Pitch: 40E
Yaw:187Pitch: 36E
Yaw:184Pitch: 42E
Yaw:179Pitch: 43E
Yaw:183Pitch: 43E
Yaw:179Pitch: 46E
Yaw:179Pitch: 46E
Yaw:182Pitch: 40E
Yaw:184Pitch: 32E
Yaw:189Pitch: 43E
Yaw:189Pitch: 63E
Yaw:182Pitch: 40E

测试结果（静止）

实际角度（粗略估计）：
方向角90度
俯仰角52度

Yaw: 89Pitch: 58E
Yaw: 92Pitch: 49E
Yaw: 96Pitch: 54E
Yaw: 92Pitch: 52E
Yaw: 89Pitch: 51E
Yaw: 89Pitch: 54E
Yaw: 92Pitch: 52E
Yaw: 92Pitch: 52E
Yaw: 89Pitch: 46E
Yaw: 89Pitch: 46E
Yaw: 89Pitch: 50E
Yaw: 87Pitch: 52E
Yaw: 94Pitch: 51E
Yaw: 89Pitch: 54E
Yaw: 93Pitch: 46E
Yaw: 89Pitch: 51E

测试结果（移动粗测）



06 总结与展望

总结与展望

总结

- 完成了对相关算法的有关论文的阅读与学习
- 初步规划好了功能实现的具体过程
- 对一些功能进行了一定的硬件实现

展望

- 在实践中逐步完善或改良现有算法的具体细节
- 逐步按照计划完成更进一步的硬件实现

THANKS

And Your Slogan Here