

## FFT\_SoC 芯片级算法 IP 核——项目计划书

### 1 项目简介

设计一款 **256 点定点 FFT IP 核**（Q24.8 格式），支持 AXI4-Stream 数据流接口。项目聚焦核心功能实现与基础验证流程，包含 Verilog RTL 设计、SystemVerilog 验证环境搭建及基础覆盖率收集，适合初学者快速掌握 IP 开发全流程。

### 2 项目目标

- 核心功能**
  - 实现 256 点 FFT 计算（基-2 算法）
  - 支持 AXI4-Stream 输入/输出（含反压处理）
  - 定点 Q24.8 数据格式
- 验证目标**
  - 搭建 SystemVerilog 验证环境
  - 实现基础功能覆盖率（>85%）
  - 完成关键场景测试
- 资源目标**
  - 开发周期：2 个月
  - 代码规模：≤ 3000 行 Verilog

### 3 项目内容

#### 3.1 算法建模

基于 matlab 软件进行算法建模

基-2 DIT FFT 算法的核心是蝶形计算单元：

$$\begin{aligned}X(k) &= A + W * B \\X(k+N/2) &= A - W * B\end{aligned}$$

其中：

- A 和 B 是第一和第二输入数据点
- W 是旋转因子（复数）
- X(k)和 X(k+N/2)是输出数据点

#### 3.2 系统架构设计

- 模块划分**
  - 蝶形运算单元（BFLY）：支持定点复用设计
  - 旋转因子 ROM：预计算 Twiddle Factor
  - 数据缓冲双口 RAM：输入和输出数据流控制单元
  - 控制状态机：支持连续/单次触发模式
- 接口定义**
  - 数据流接口：AXI4-Stream

#### 3.3 RTL 设计与优化

模块	实现内容	关键技术
蝶形运算核心	流水线复数乘法器+加法树	Booth 编码乘法器/CSA 加法链
地址生成器	基-2 算法索引控制	位反转寻址+状态机调度
数据流控制器	双口 RAM	数据缓存及读取
AXI4-Stream 适配	数据打包/解包+流控握手	反压处理

3.4 验证平台搭建

//验证环境架构

```
TB_top
├── fft_driver.sv      // AXI4S 驱动
├── fft_monitor.sv    // 输出采集
├── ref_model.py       // 参考模型
├── scoreboard.sv     // 幅值误差检查(±1 LSB)
└── test_cases.sv     // 手动编写测试场景
```

3.5 验证关键指标

关键测试场景

- 1. 基础功能测试
  - 正弦波输入→验证频谱峰值
  - 冲激函数输入→验证平坦频谱
  - 全零输入→输出全零验证
- 2. 接口测试
  - AXI 反压测试（随机 TREADY 拉低）
  - 连续帧处理测试
- 3. 覆盖率收集
  - 代码覆盖率
  - 简单功能覆盖点：
    - 状态机覆盖率
    - 数据路径激活

4 项目实施计划

阶段	时间	核心任务
架构设计	1 周	确定模块划分+算法模型验证
RTL 编码	3 周	实现蝶形单元/控制 FSM/AXI 接口
验证环境搭建	2 周	搭建 SV TB+编写基础测试用例
测试与调试	1.5 周	执行测试+收集覆盖率+修复 BUG

阶段	时间	核心任务
交付准备	0.5 周	整理代码+编写简易文档
5 交付成果		
<ul style="list-style-type: none"><li>可综合 RTL 代码</li><li>验证环境与测试用例</li><li>简易设计文档（含模块说明）</li></ul>		