信号处理详细设计

文件编号

本模板的使用建立在假设用户已经基本了解了“软件开发指引（独立文件）”的内容的基础上。在使用本模板时出现关于其用法的问题，请参考“软件开发指引”相关章节

Compliant with Functional safety Standard (ISO26262)

□Required 需要 ☑ No不符合功能安全标准(ISO26262) □

|  |  |  |  |
| --- | --- | --- | --- |
| 批准 | 审核 | 审核 | 编制 |
|  |  |  |  |

发行日期

2021/07/22

发行单位

**变更历史**

| 版本 | | 审批日期 | 作者 | 审核人 | 修改内容 |
| --- | --- | --- | --- | --- | --- |
| 当前 | 新版 |
| --- | V1.0 |  |  |  | 基础版本 |
|  | V1.1 |  |  |  | CR-1：修改3.1静态设计框图  CR-2：修改3.1.11速度解模糊流程  CR-3：修改2.1组件约束  CR-4：修改3.2动态组件设计，增加超时处理动态图 |
|  | V1.2 |  |  |  | CR-1：修改3.7动态组件设计描述 |

目录

[信号处理详细设计 1](#_Toc29604)

[1 概要 2](#_Toc32019)

[1.1 目的 2](#_Toc30756)

[1.2 参考文件 2](#_Toc2505)

[1.3 用语、缩略语等定义 3](#_Toc12675)

[2 制约条件 3](#_Toc13512)

[2.1 组件约束 3](#_Toc19021)

[3 软件组件设计 3](#_Toc16304)

[3.1 组件静态设计 3](#_Toc16861)

[3.2 组件动态设计 4](#_Toc18372)

[3.2.1 波形配置 6](#_Toc13943)

[3.2.2 ADC数据采集 9](#_Toc14361)

[3.2.3 一维FFT 9](#_Toc7475)

[3.2.4 二维FFT 10](#_Toc30294)

[3.2.5 二维幅值矩阵 10](#_Toc15939)

[4 功能函数详述 11](#_Toc26220)

[4.1 配置ADCBUF驱动程序的函数 11](#_Toc16469)

[4.2 一维FFT功能函数 12](#_Toc1246)

[4.3 二维FFT功能函数 12](#_Toc9253)

[4.4 二维幅值矩阵功能函数 13](#_Toc5278)

1. 概要
   1. 目的

本详细设计说明书编写的目的是为了说明程序模块的设计考虑，包括程序描述、输入/输出、算法和流程逻辑等，为软件编程和系统维护提供基础。本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。

* 1. 参考文件

| 编号 | 参考文档 | 版本 | 发行者 | 理由 |
| --- | --- | --- | --- | --- |
| 1 | 系统架构文档 |  |  |  |
| 2 | IWR68xxIndustrial Radar Technical Reference Manual |  |  |  |

* 1. 用语、缩略语等定义

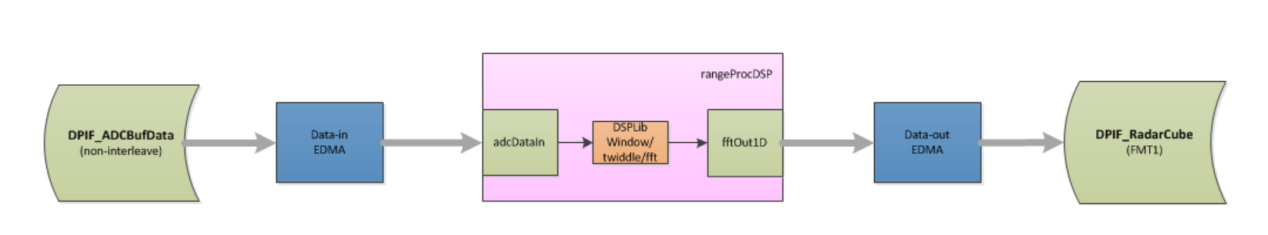
| 编号 | 用语、缩略语 | 含义、定义和正式名称 |
| --- | --- | --- |
| 1 | SPT | 信号处理工具箱 |
| 2 | CFAR | 恒虚警率 |
| 3 | ADC | 模数转换器 |
| 4 | ADAS | 高级驾驶员辅助系统 |
| 5 | APP | 应用程序 |
| 6 | SRAM | 静态随机存取存储器 |
| 7 | FLASH | 闪速存储器 |
| 8 | FFT | 傅里叶变换 |
| 9 | ID | 编号 |
| 10 | RCS | 雷达横截面 |
| 11 | HIST | 直方图 |
| 12 | DML | 确定性最大似然算法 |
| 13 | MMIC | 单片式微波集成电路 |

1. 制约条件
   1. 组件约束

软件设计和实现约束条件如下：

1. 供信号处理使用的L2空间最大为256K, L3空间大小最大为768K；
2. 需要考虑运行时间，在能够输出最大目标个数的情况下，运算时间不得超出系统的调度时间；
3. 需要考虑兼容性以及软件的可移植性，便于后续维护及开发；
4. 模块实现需要遵循MISRA C:2012规范。
5. 软件组件设计
   1. 组件静态设计

1DFFT



2DFFT

|  |
| --- |
|  |

1. 目标模块静态框图
   * 1. 配置

考虑到内存限制此次调试波形设置为1x1

|  |
| --- |
| sensorStop flushCfg dfeDataOutputMode 1 channelCfg 15 6 0 adcCfg 2 1 adcbufCfg -1 0 0 1 1 profileCfg 0 60.25 10 6 80 0 0 46.5 0 128 2000 0 0 40 chirpCfg 0 0 0 0 0 0 0 2 chirpCfg 1 1 0 0 0 0 0 4 chirpCfg 2 2 0 0 0 0 0 2 chirpCfg 3 3 0 0 0 0 0 4 frameCfg 0 3 128 0 20 1 0 lowPower 0 1 sensorStart |

|  |  |  |  |
| --- | --- | --- | --- |
| 配置command | Command details | Command Parameters | Usage in mmW demo  /xwr68xx |
| sensorStop |  |  |  |
| flushCfg | 此命令应在“sensorStop”命令之后发出，以刷新旧配置并提供新配置。 |  |  |
| dfeDataOutputMode 1 | 此命令中的值不应更改 在 sensorStop 和 sensorStart 之间。  这是强制性命令。 | 1 - frame based chirps  2 - continuous chirping  3 - advanced frame config | 支持1 、3 |
| channelCfg 15 1 0 | N/A | <rxChannelEn>  接收天线掩码，例如对于 4 个天线，它是 0x1111b = 15 | 支持 4 个天线 |
| <txChannelEn>  发射天掩码 | 可以使用位掩码 0x5（即 tx1 和 tx3）启用 2 个方位角天线。 方位角和仰角天线都可以使用位掩码 0x7（即 tx1、tx2 和 tx3）启用。 可以使用位掩码 0x3（即 tx1 和 tx2）启用 2 个方位角天线。 |
| <cascading>  SoC 级联，不适用，设置为 0 |  |
| adcCfg 2 1 | N/A | <numADCBits>  1是14位字节2是16为字节 | 只支持16位 |
| <adcOutputFmt>  0是real、1是complex | 只支complex |
| adcbufCfg -1 0 0 1 1 | N/A | <subFrameIdx>  子帧索引 | legacy模式设置为-1； |
| <adcOutputFmt>  ADCBUF 输出格式  0 - Complex, 1-Real； | 只支持1 |
| <SampleSwap>  ADCBUF IQ swap selection:  I in LSB, Q in MSB, 1-Q in LSB, I in MSB； | 只支持1 |
| <ChanInterleave>  ADCBUF channel interleave  0 - interleaved(only supported for XWR14xx),  1 - non-interleaved； | 只支持1 |
| <ChirpThreshold>  用于 ADCBUF 缓冲区的 Chirp 阈值配置以触发 ping /pong 缓冲开关。有效值：0-8，xwr64xx/xwr68xx/xwr18xx：仅支持值 1 | 只支持1 |
| profileCfg 0 60 48.5 5 31.6 0 0 120 0 128 5000 0 0 40 | N/A | profile Identifier | Legacy frame： (dfeOutputMode=1)：可以是任何允许的值，但每个配置只支持一个有效的配置文件 Advanced frame：(dfeOutputMode=3)：可以是任何允许的值，但每个子帧仅支持一个配置文件。 然而，不同的子帧可以有不同的配置文件 |
| <起始频率> 以 GHz 为单位的“频率启动”（允许浮动值） 例子： 77 61.38 | 请参阅上面显示的 chirp 图以了解各种配置文件参数和相互依赖的约束之间的关系。 |
| <空闲时间> 以 u-sec 为单位的“空闲时间”（允许浮点值） 例子： 7 7.15 | N/A |
| <adcStartTime> 以 u-sec 为单位的“ADC 有效开始时间”（允许浮点值） 例子： 7 7.34 | N/A |
| <rampEndTime> 以 u-sec 为单位的“斜坡结束时间”（允许浮点值） 例子： 58 216.15 | N/A |
| <txOutPower> tx 天线的 Tx 输出功率back-off代码 | 测试了“0”的值 |
| <txPhaseShifter> 用于 tx 天线的 tx 移相器 | 测试了“0”的值 |
| <freqSlopeConst> 以 MHz/usec 为单位的线性调频“频率斜率”（允许浮点值） 例子： 68 16.83 | N/A |
| <tx开始时间> 以 u-sec 为单位的“TX 开始时间”（允许浮点值） 例子： 1 2.92 | N/A |
| <numAdcSamples> 在“ADC 采样时间”期间收集的 ADC 样本数为 如上面的啁啾图所示示例： 256 224 | N/A |
| <digOutSampleRate> 以 ksps 为单位的 ADC 采样频率。(<numAdcSamples> / <digOutSampleRate> = "ADC 采样时间") 例子： 5500 | N/A |
| <hpfCornerFreq1> HPF1（高通滤波器 1）转角频率 0: 175 kHz 1：235KHz 2：350KHz 3：700KHz | 具体的值根据数据手册 |
| <hpfCornerFreq2> HPF2（高通滤波器 2）转角频率 0：350KHz 1：700KHz 2：1.4兆赫 3：2.8 兆赫 | 具体的值根据数据手册 |
| <rxGain> 以 dB 为单位的 RX 增益与 RF 增益目标的 OR 值 | 具体的值根据数据手册 |
| chirpCfg 0 0 0 0 0 0 0 1 | N/A | chirp 起始索引； | 根据 mmwavelin doxygen 的任何值 |
| chirp 结束索引； | 根据 mmwavelin doxygen 的任何值 |
| 配置文件标识符； | 应该匹配 profileCfg->profileId |
| 以 Hz 为单位的起始频率变化（允许浮点值） | 只支持0 |
| 以 kHz/us 为单位的频率斜率变化（允许浮点值） | 只支持0 |
| 以 u-sec 为单位的空闲时间变化（允许浮点值） | 只支持0 |
| 以 u-sec 为单位的 ADC 开始时间变化（允许浮点值） | 只支持0 |
| tx 天线启用掩码 (Tx2,Tx1) 例如 (10)b = Tx2 启用，Tx1 禁用。 | 单个 chirp 应仅启用一个不同的 Tx 天线 (MIMO) 或应为所有 chirp 启用相同的 TX 天线 |
| frameCfg 0 0 128 0 50 1 0 | dfeOutputMode 应设置为 1 以使用此命令。此命令中的值可以在 sensorStop 和 sensorStart 之间更改。 当 dfeOutputMode 设置为 1 时，这是一个强制性命令。 | chirp 起始索引 (0-511) | 定义相应的 chirp Cfg 的任何值 |
| chirp 结束索引 (chirp start index-511) | 定义相应的 chirp Cfg 的任何值 |
| number of loops (1 to 255) | 大于等于4并且为4的倍数 |
| 帧数（有效范围是 0 到 65535，0 表示无限） | 定义相应的 chirp Cfg 的任何值 |
| 以毫秒为单位的帧周期（允许浮点值） | 帧不应有超过 50% 的占空比（即活动线性调频时间应 <= 帧周期的 50%） |
| 触发选择 1：软件触发 2：硬件触发。 | 只支持软件触发 |
| 以毫秒为单位的帧触发延迟（允许浮点值） | 定义相应的 chirp Cfg 的任何值 |
| lowPower 0 1 | N/A | <don’t\_care> | 设置为 0 |
| ADC模式 0x00：常规ADC模式 0x01：低功耗 ADC 模式（xwr6xxx 设备不支持） | 使用“0”或“1”的值（取决于 profileCfg- >digOutSampleRate) |
| sensorStart | N/A |  |  |

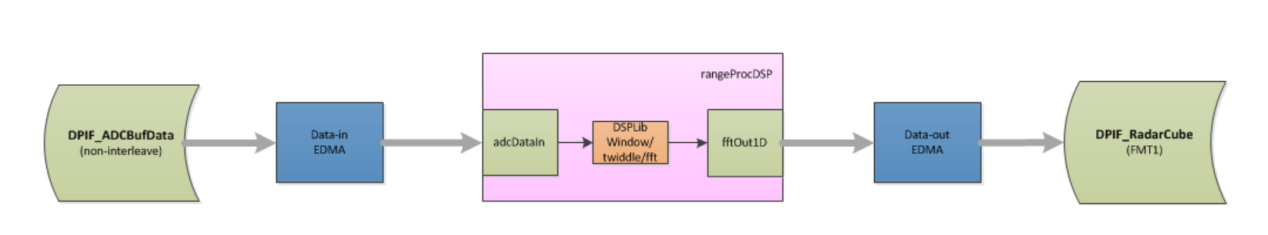
* + 1. ADC数据采集

ADCBuf是在DSS程序中进行的配置，因为它是C674x的外设。内存大小为32KB，如下图：



ADCBuf驱动的配置以及启用：

1. 调用ADCBuf驱动
2. 对adcBufParams进行赋值，完成参数初始化
3. 开启天线接收并讲接受的数据储存
   * 1. 一维FFT



1. 一维FFT处理流程

RangeProcDSP 有 3 个处理阶段：

1.通过dataIn EDMA通道引入ADC数据

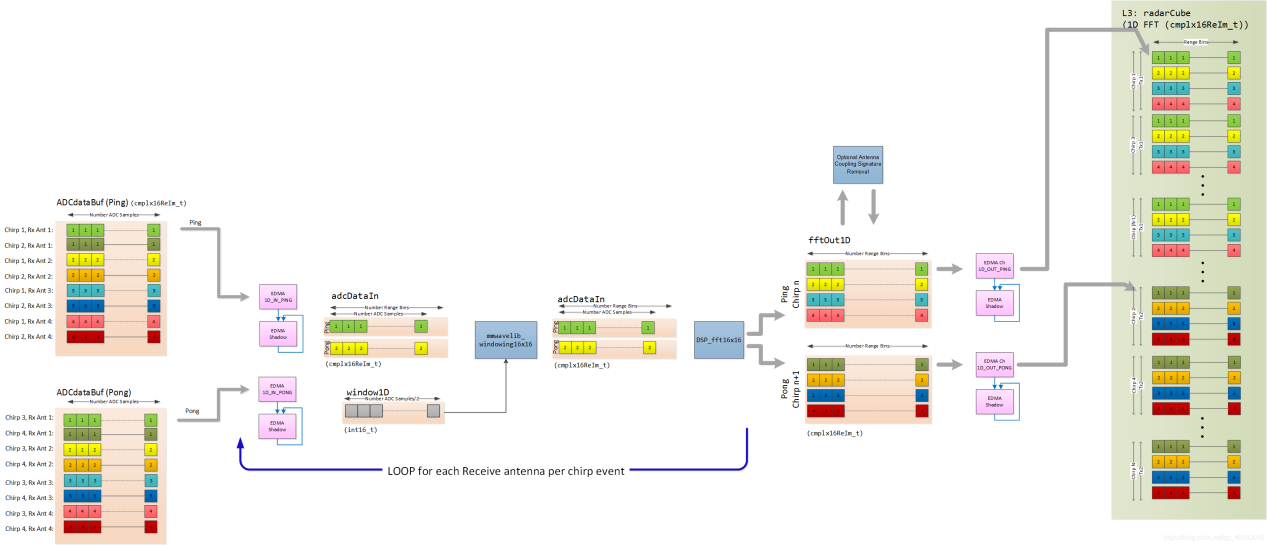
2.使用 DSPlib/mmwavelib 进行 FFT 处理

3.通过 dataOut EDMA 通道将 FFT 结果传输到雷达立方体

使用案例:

无论 TX 天线的数量如何，距离 FFT 都需要 2 个输入 EDMA 通道和 2 个输出 EDMA 通道才能以Ping Pang方式工作。

Ping 输入通道将数据带入本地内存“adcDataIn”的 Ping 区域，pong 通道将数据带入 pong 区域。 FFT 后，对于 2 TX 天线，本地存储器“fftout1D”中的 ping 结果被复制到雷达立方体中的 TX1 区域。 Pong 结果被复制到雷达立方体中的 TX2 区域。

对于 3 TX 天线，ping 总是处理奇数 chirp 数据，pong 总是处理偶数 chirp 数据。 距离 FFT 结果以下列格式复制到雷达立方体： 

* + 1. 二维FFT



1. 二维FFT处理流程

如上图，二维FFT模块的输入为一维FFT的结果，输出为二维FFT的结果。二维FFT主要做速度维的傅里叶变换。首先读取一维FFT的结果，将数据转置为速度维数据存储在一起的格式，之后进行相应点数的FFT操作，最终得到二维FFT的结果并输出到L3中存储。

* + 1. 二维幅值矩阵



1. 二维幅值矩阵计算流程

如上图，二维幅值矩阵的输入值为单个接收通道的二维FFT的结果，输出为单通道的二维幅值矩阵。其中只计算单个通道的二维幅值，最终得出单通道的二维幅值矩阵用以速度解模糊使用。

1. 功能函数详述
   1. 配置ADCBUF驱动程序的函数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能名称 | MmwDemo\_dssDataPathConfigAdcBuf() | | | |
| 输入 | ADCBuf的数据格式 | 当前ADCBuf的格式 | 精度 | 1 |
| 值域 |  |
| 分辨率 | 1 |
| 输入 | 使能Rx channels | 开启Rx天线 | 精度 | 1 |
| 值域 | 0x2100\_0000 – 0x2100\_7FFC |
| 分辨率 | 1 |
| 输入 | 设置ping/pang阈值 | 缓存大小 | 精度 |  |

* 1. 一维FFT功能函数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能名称 | MmwDemo\_processChirp() | | | |
| 输入 | 原始数据地址 | MMIC采样通过总线传输到SRAM（ADC data Buffer）的地址 | 精度 | 1 |
| 值域 | 0x2100\_0000 – 0x2100\_7FFC |
| 分辨率 | 1 |
| 输入 | 发射通道1的chirps数据处理地址 | 发射通道1的原始数据通过ping pang缓存EDMA数据传输在SRAM(L2)中经过一维FFT计算 | 精度 | 1 |
| 值域 | 0x007E\_0000 – 0x0081\_FFFF |
| 分辨率 | 1 |
| 输入 | 帧类型 | 当前帧的制式 | 精度 |  |
| 值域 |  |
| 分辨率 |  |
| 输入 | 波形类型 | 波形的用途 | 精度 |  |
| 值域 |  |
| 分辨率 |  |
| 输出 | 一维FFT结果 | 一维FFT结果地址（L3） | 精度 | 1 |
| 值域 | 0x2000\_0000-0x201F\_FFFF |
| 分辨率 | 1 |
| 解释 | 一维FFT功能 | | | |

1. SptRangeFft
   1. 二维FFT功能函数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能名称 | MmwDemo\_interFrameProcessing | | | |
| 输入 | 一维FFT数据地址 | 一维FFT数据结果在SRAM(L3)中的存放地址 | 精度 | 1 |
| 值域 | 0x2000\_0000-0x201F\_FFFF |
| 分辨率 | 1 |
| 输入 | 二维FFT数据地址 | 一维FFT结果经过二维FFT计算得到的结果在SRAM(L3)中的地址 | 精度 | 1 |
| 值域 | 0x2000\_0000-0x201F\_FFFF |
| 分辨率 | 1 |
| 输入 | 帧类型 | 当前帧的制式 | 精度 |  |
| 值域 |  |
| 分辨率 |  |
| 输入 | 波形类型 | 波形的用途 | 精度 |  |
| 值域 | t |
| 分辨率 |  |
| 输出 | 二维FFT结果 | 对一维FFT结果进行处理之后的二维FFT结果 | 精度 | 1 |
| 值域 | 0 -- 65535 |
| 分辨率 | 1 |
| 解释 | 二维FFT功能函数 | | | |

1. SptDopplerFft
   1. 二维幅值矩阵功能函数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能名称 | SptMagnitudeMatrixExport | | | |
| 输入 | 二维FFT结果地址 | 二维FFT结果的SRAM地址 | 精度 | 1 |
| 值域 | 0x40000000 – 0x4017FFFF |
| 分辨率 | 1 |
| 输入 | 幅值矩阵地址 | 对多通道的二维FFT进行非相参累积之后得到的幅值矩阵在SRAM中的存储地址 | 精度 | 1 |
| 值域 | 0x40000000 – 0x4017FFFF |
| 分辨率 | 1 |
| 输入 | 帧类型 | 当前帧的制式 | 精度 |  |
| 值域 | teMrModeFrame,  teSrModeFrame |
| 分辨率 |  |
| 输入 | 波形类型 | 波形的用途 | 精度 |  |
| 值域 | teReferenceChirp,  teBaseChirp |
| 分辨率 |  |
| 输出 | 二维幅值矩阵结果 | 对二维FFT结果进行处理之后的二维幅值矩阵结果 | 精度 | 1 |
| 值域 | 0 -- 65535 |
| 分辨率 | 1 |
| 解释 | 解模糊波形的幅值矩阵计算功能 | | | |

1. SptMagnitudeMatrixExport