**基带处理概要设计说明书**

|  |  |
| --- | --- |
| 拟制/日期 |  |
| 标准化/日期 |  |
| 审核/日期 |  |
| 会签/日期 |  |
| 批准/日期 |  |

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **修订版本** | **日期** | **作者** | **修改描述** | **备注** |
| V1.0 | 20230724 | 孙东/蔡占辉 | 初稿发布 |  |
| V1.1 | 20230731 | 孙东/蔡占辉 | 增加总体设计工作流程 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[1 引言 1](#_Toc141791263)

[1.1 目的 1](#_Toc141791264)

[1.2 范围 1](#_Toc141791265)

[1.3 缩略语定义 1](#_Toc141791266)

[1.4 参考资料 1](#_Toc141791267)

[2 总体设计 2](#_Toc141791268)

[2.1 组成原理 2](#_Toc141791269)

[2.2 工作流程 3](#_Toc141791270)

[2.2.1 启动准备 3](#_Toc141791271)

[2.2.2 启动仿真 5](#_Toc141791272)

[3 模块设计 7](#_Toc141791273)

[3.1 时钟单元 7](#_Toc141791274)

[3.1.1 命令通道 7](#_Toc141791275)

[3.1.2 网络中断 8](#_Toc141791276)

[3.2 ARM0方案设计 9](#_Toc141791277)

[3.2.1 数据传输 9](#_Toc141791278)

[3.2.2 程序结构说明 17](#_Toc141791279)

[3.3 ARM1方案设计 23](#_Toc141791280)

[3.3.1 数据传输 23](#_Toc141791281)

[3.3.2 程序结构说明 27](#_Toc141791282)

[3.4 FPGA逻辑方案设计 33](#_Toc141791283)

[3.4.1 功能模块设计 34](#_Toc141791284)

[3.4.2 接口协议 34](#_Toc141791285)

[4 风险分析及应对策略 37](#_Toc141791286)

[5 其他设计说明 37](#_Toc141791287)

[6 附录 数仿数据文件格式 38](#_Toc141791288)

[6.1 数据文件头 38](#_Toc141791289)

[6.2 载体轨迹数据文件 38](#_Toc141791290)

[6.3 观测数据接口文件 39](#_Toc141791291)

[6.4 电文数据接口文件 40](#_Toc141791292)

# 

# 引言

## 目的

本文是无人机导航诱导信号源初样机基带处理部分的概要设计，用于指导嵌入式和FPGA工程师进行程序设计。

## 范围

初样机目前采用已有硬件平台进行信号生成，后续在产品设计中相关内容还需要进行修改。

## 缩略语定义

|  |  |  |
| --- | --- | --- |
| **缩略语** | **全称** | **描述** |
| PS | processing system | 处理系统 |
| PL | programmable logic | 可编程逻辑 |
|  |  |  |

## 参考资料

略。

# 总体设计

## 组成原理

SSH-100导航诱导信号源初样机功能框图如下图所示。



图2.1功能结构框图

在系统基本原理如下：

1. 上位机：读取星历文件生成卫星导航电文和观测数据；分解数据，生成600ms周期的卫星导航电文和20ms周期的观测数据，并传输数据和命令到ARM0；并在界面上显示通道状态，卫星状态，轨迹状态及软件运行时间等信息；
2. 时钟单元：接收外部输入的参考时钟、同步信息，生成基带处理所需要的参考时钟、同步触发信号、1KHz信号及可编程时钟；解析上位机通过以太网发送的控制信息，周期性向上位机请求20ms中断。
3. ARM0：接收600ms周期的电文和20ms周期的观测数据，把数据保存在指定的缓存空间；
4. ARM1：从缓存空间获取电文与观测量，对基带数据进行处理，生成1ms的基带参数，通过中断发送到FPGA；
5. FPGA：接收ARM1传输来的基带参数，生成数字基带信号。

系统运行还包括功率调整、通道控制等控制命令的传输，ARM0接收命令包转发给ARM1；ARM1接收并响应部分指令，把指令响应结果发送给FPGA直接响应命令。

## 工作流程

工作流程如下图所示。



图2‑1主流程时序图（初样）

### 启动准备

上位机首先设置读取数学仿真数据的路径，数据包括观测数据和导航电文，数据格式见附录。上位机软件开始运行后首先进入启动准备阶段。在启动准备阶段进行设备硬件连接检查和硬件初始化，为仿真运行做准备。其中硬件初始化包括时钟单元初始化和基带模块初始化。

1. **时钟单元初始化**

* 时钟单元UDP初始化

时钟单元与上位机建立一个周期UDP软中断，以固定周期（例如20ms）发送一个简单UDP数据包，上位机周期接收到UDP数据包作为上位机的时钟周期，达到上位机与下位机时钟同源。

UDP通信端口号：7070；

UDP 通信内容： int32 params[4]；

下位机初始化时创建一个UDP接收线程，由上位机发送第一包通信参数数据包并建立一个UDP接收线程，下位机接收第一包通信参数包后，获取上位机通信地址信息"192.9.200.95"，并根据参数包完成初始化。

|  |  |
| --- | --- |
| 序号 | 数据内容说明 |
| params[0] | 缓存包数 |
| params[1]~ params[3] |  |

时钟单元等待上位机的启动命令，接收启动命令后，中断任务周期向上位机发送UDP时间中断数据包，内容如下。

|  |  |
| --- | --- |
| 序号 | 数据内容说明 |
| params[0] | 中断计数 |
| params[1]~ params[3] | 扩展 |

* 时钟单元TCP初始化

命令通道接口采用TCP通讯协议。命令的数据结构：

typedef struct

{

INT32 cmdtype; //命令分类一级类别

INT32 cmdid; //命令分类二级类别

INT32 item[7]; //参数

INT32 dataLen;

}CTRL\_CMD；//40 Byte

|  |  |
| --- | --- |
| **参数** | **数据内容说明** |
| 1 初始化(cmdtype = 0x12120000,cmdid=0x01) datalen =40 | |
| cmd.item[0] | 触发模式：1：内部触发 |
| cmd.item[1] | 系统周期 1：表示20ms |
| cmd.item[2] | 系统延迟：单位毫秒。0表示0毫秒。 |
| cmd.item[3] | True：表示启动网络软中断 |
| cmd.item[4]~ cmd.item[6] | 任意 |

上位机查收应答信息，若收到应答且应答命令ID正确，则表示时钟单元初始化成功。

1. **基带模块初始化**

|  |  |
| --- | --- |
| **参数** | **数据内容说明** |
| 1 初始化(cmdtype = 0x12120000,cmdid=0x01) datalen =40 | |
| cmd.item[0] | 系统周期  **1：表示20ms**，2：表示10ms，3表示5 ms； |
| cmd.item[1] | 系统频点信息，**低8位Y侧频点号**，高8位X侧频点号 |
| cmd.item[2] ~ cmd.item[5] | 0 |
| cmd.item[6] | 14：GPS相对北斗时的跳秒14秒。 |

上位机查收应答信息，若收到应答且应答命令ID正确，则表示时钟基带模块初始化成功。

### 启动仿真

启动准备阶段完成后，上位机软件进入启动仿真阶段。

1. **首先发送缓存包**

提取缓存包数的数据发送给基带模块ARM0，数据格式如下。

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类型** | **字段名称** | **数据类型** | **数据内容说明** |
| 数据头 | 总数据长度 | int | 数据内容的总长度 |
| Y测Obs数据长度 | int |  |
| X测Obs数据长度 | int |  |
| Y侧 Nav数据长度 | int |  |
| X侧Nav数据长度 | int |  |
| 数据内容 | Y侧Obs数据内容 | char \* |  |
| X侧Obs数据内容 | char \* |  |
| Y侧Nav数据内容 | char \* |  |
| X侧Nav数据内容 | char \* |  |

1. **启动时钟单元**

向时钟单元发送启动命令，格式如下。

|  |  |
| --- | --- |
| 2 启动命令(cmdtype = 0x12120000,cmdid=0x02)datalen = 40; | |
| cmd.item[0]~cmd.item[6] | 内触发：立即产生触发信号；  外触发：等待外部触发信号； |
| 应答初始化命令(cmdtype = 0x12120000,cmdid=0x02) datalen =40 | |
| cmd.item[0]~cmd.item[6] | 扩展 |

时钟单元启动后向基带模块发送启动信号，基带模块开始启动；时钟单元通过UDP线程周期性向上位机发送中断请求，上位机响应中断向下位机发送数学仿真数据。

# 模块设计

## 时钟单元

### 命令通道

命令通道接口采用TCP通讯协议。

typedef struct

{

INT32 cmdtype; //命令分类一级类别

INT32 cmdid; //命令分类二级类别

INT32 item[7]; //参数

Int dataLen;

}CTRL\_CMD;

表3‑1命令列表

|  |  |  |  |
| --- | --- | --- | --- |
| **命令类型cmdtype** | **上位机命令编码cmdid** | **目标机应答命令cmdid** | **命令描述** |
| 0x12120000  （时序控制） | 0x00000001 | 0x01000000 | 初始化 |
| 0x00000002 | 0x02000000 | 启动运行 |
| 0x00000003~0x00000007 | 0x03000000~  0x07000000 | 扩展 |
| 0x00000008 | 0x08000000 | 停止复位 |

表3‑2命令内容

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **数据内容说明** | | |
| 1 初始化(cmdtype = 0x12120000,cmdid=0x01) datalen =0 | | | |
| cmd.item[0] | 触发模式：1内部触发，2外部触发； | | |
| cmd.item[1] | 系统周期 1：表示20ms，2：表示10ms，3表示5 ms； | | |
| cmd.item[2] | 系统延迟：单位毫秒。系统周期系统延迟，如0表示0毫秒，从触发到信号输出0毫秒。 | | |
| cmd.item[3] | 是否启动网络软中断 | | |
| cmd.item[4]~ cmd.item[6] | 扩展 | | |
|  | | | |
| 应答初始化命令(cmdtype = 0x12120000,cmdid=0x01) datalen =0 | | | |
| cmd.item[0] | 按位标示模拟器基带板卡使能状态，从最低位开始标示 基带1，最高32位标示基带32 | | |
| cmd.item[1] | 按位标示模拟器基带板卡使能状态，从最低位开始标示 基带33，最高32位标示基带64 | | |
| cmd.item[2]~cmd.item[6] | 扩展 | | |
|  | | | |
| 2 启动命令(cmdtype = 0x12120000,cmdid=0x02)datalen = 0; | | | |
| cmd.item[0]~cmd.item[6] | | 内触发：立即产生触发信号；  外触发：等待外部触发信号； | |
| 应答初始化命令(cmdtype = 0x12120000,cmdid=0x02) datalen =0 | | | |
| cmd.item[0]~cmd.item[6] | | 扩展 | |
|  | | | |
| 3 停止复位(cmdtype = 0x12120000,cmdid=0x08) datalen = 0 | | | |
| cmd.item[0]~cmd.item[6] | int | | 扩展 |
| 应答停止(cmdtype = 0x12120000,cmdid=0x08)datalen=0 | | | |
| cmd.item[0]~cmd.item[6] | Int | | 扩展 |

### 网络中断

时钟单元由RPU1和FPGA来实现。时钟单元与上位机建立一个周期UDP软中断，以固定周期（例如20ms）发送一个简单UDP数据包，上位机周期接收到UDP数据包作为上位机的时钟周期，达到上位机与下位机时钟同源。

UDP通信端口号：7070；

UDP 通信内容： int32 params[4]；

下位机初始化时创建一个UDP接收线程，由上位机发送第一包通信参数数据包并建立一个UDP接收线程，下位机接收第一包通信参数包后，获取上位机通信地址信息，并根据参数包完成初始化。

|  |  |
| --- | --- |
| 序号 | 数据内容说明 |
| params[0]~ params[4] | 扩展 |

时钟单元等待上位机的启动命令，接收启动命令后，中断任务周期向上位机发送UDP时间中断数据包，内容如下。

|  |  |
| --- | --- |
| 序号 | 数据内容说明 |
| params[0] | 中断计数 |
| params[1]~ params[3] | 扩展 |

## ARM0方案设计

ARM0内核中运行嵌入式Linux系统，利用多线程等方式实现与上位机进行数据通信、数据初步处理以及数据中转功能，是连接上位机与处理计算单元ARM1的通道。

ARM0作为TCP的服务器端与上位机进行数据交互，采用Linux操作系统。操作系统由BOOT、内核、设备树、文件系统四部分构成。

ARM0通过网络从上位机接收各种数据，收到数据后根据需要分别进行处理，将命令和处理后的数据传输到ARM1和其他连接的外设。连接关系如下图所示。



图3‑1 ARM0接口示意图

ARM0通过网口与上位机相连，该接口进行传输命令和数据。

ARM0通过AXI接口与ARM1相连，该接口将接收到的数据向计算单元ARM1传递。

ARM0通过串口与检测调试终端相连，该接口监控调试Linux系统以及应用程序的工作状态。

### 数据传输

#### 与上位机数据传输

ARM0模块从上位机接收数据，处理后生成输出数据供FPGA使用。ARM0通过TCP接收需要处理的命令和数据，包括观测数据、电文数据等，其中命令和数据分别通过两个不同的网络套接字进行传输，系统上电后在指定端口处进行监听，等待上位机的连接请求，在系统开始仿真运行后对数据套接字和命令套接字进行同步接收，一直到仿真结束，上位机断开网络连接，将内存变量初始化。网络接口是100M以太网。如下图所示。



图3‑2 网络通信示意图

1. 命令格式

命令的数据结构：

typedef struct

{

INT32 cmdtype; //命令分类一级类别

INT32 cmdid; //命令分类二级类别

INT32 item[7]; //参数

INT32 dataLen;

}CTRL\_CMD；//40 Byte

表3‑3控制命令字

|  |  |  |  |
| --- | --- | --- | --- |
| **命令类型cmdtype** | **上位机命令编码cmdid** | **目标机应答命令cmdid** | **命令描述** |
| 0x12120000  （时序控制） | 0x01 | 0x01 | 初始化并自检，完毕后接收缓存数据，等待触发运行 |
| 0x02 | 0x02 | 状态反馈 |
| 0x02~0x07 | 0x02~0x07 | 扩展 |
| 0x08 | 0x08 | 停止复位 |
| 0x09 | 0x09 | 自毁 |
| …… | …… | 扩展 |
|  | | | |
| 0x34340000  （RNSS相关功能控制） | 0x13000001 | 无应答 | 调制类型（单载波，BPSK单I，Q，QPSK，BOC） |
| 0x13000007 | 无应答 | 功率调整 |
| 0x13000008 | 无应答 | 失锁重捕 |
| 0x13000009 | 无应答 | 射频功率调整 |
|  |  |  |

（1）通用命令协议

|  |  |  |
| --- | --- | --- |
| 参数 | | 数据内容说明 |
| **1 自检(cmdtype = 0x12120000,cmdid=0x01)** | | |
| cmd.item[0] | | 系统周期 1：表示20ms，2：表示10ms，3表示5 ms； |
| cmd.item[1] | | 系统频点信息，低8位Y侧频点号，高8位X侧频点号 |
| cmd.item[2]~ cmd.item[6] | | 预留扩展 |
| 应答自检**(cmdtype = 0x12120000,cmdid=0x01)** | | |
| cmd.item[0] | | 基带板频点编号，分Y侧、X侧，每个频点编号占十进制2位，Y侧在前，X侧在后。如：Y侧频点为E5b，对应编号为11；X侧频点为B1，对应编号为4，则本字段编码：4\*100+11=411 |
| cmd.item[1] | | 射频时钟、基带板时钟、P××模块状态、硬件版本号、FPGA版本号、DSP/ARM版本号 |
| cmd.item[2] | |
| cmd.item[3] | |
| cmd.item[4] | |
| cmd.item[5]~ cmd.item[6] | | 扩展 |
|  | |  |
| **2 状态反馈(cmdtype = 0x12120000,cmdid=0x02)** | | |
| **状态反馈(cmdtype = 0x12120000,cmdid=0x02)** | | |
| cmd.item[0] | | 下位机状态随机上报：错误、异常、复位等信息 |
| cmd.item[1]~ cmd.item[6] | | 预留扩展 |
|  | | |
| **3停止复位(cmdtype = 0x12120000,cmdid=0x08)** | | |
| cmd.item[0]~cmd.item[6] | | 预留扩展 |
| 应答停止**(cmdtype = 0x12120000,cmdid=0x08)** | | |
| cmd.item[0]~cmd.item[6] | | 预留扩展 |
|  | | |
| **4自毁(cmdtype = 0x12120000,cmdid=0x09)** | | |
| cmd.item[0]~cmd.item[6] | | 预留扩展 |
| 应答复位**(cmdtype = 0x12120000,cmdid=0x09)** | | |
| Cmd.item[0] | 1：成功；0失败 | |
| Cmd.item[1] | 信息类别:当失败时，指定失败类别，上位机根据失败类别，进行用户友好交互； | |
| Cmd.item[2]~ cmd.item[6] | 扩展 | |
|  | | |

（2）RNSS命令协议

|  |  |
| --- | --- |
| **1 调制类型(cmdtype = 0x34340000,cmdid=0x13000001)** | |
| cmd.item[0] | 1：Y侧，2：X侧，3：X侧+Y侧 |
| cmd.item[1] | 通道号1，2，…24； 0：所有通道； |
| cmd.item[2] | 1：普通单载波；2：BPSK（单I）；3 ：BPSK（单Q）；4：QPSK；5：BOC调制； |
| cmd.item[3]~cmd.item[6] | 扩展 |
| **7功率调整(cmdtype = 0x34340000,cmdid=0x13000007)** | |
| cmd.item[0] | 1：Y侧，2：X侧，3：X侧+Y侧 |
| cmd.item[1] | 通道号1，2，…，24；0：所有通道 |
| cmd.item[2] | 功率修改值\*10.0（OBS调整） |
| cmd.item[3]~cmd.item[6] | 扩展 |
| **8 失锁重捕(cmdtype = 0x34340000,cmdid=0x13000008)** | |
| cmd.item[0] | 1：Y侧，2：X侧，3：X侧+Y侧 |
| cmd.item[1] | 通道号1，2，…，24；0：所有通道 |
| cmd.item[2] | 失锁时间，单位秒 |
| cmd.item[3]~cmd.item[6] | 扩展 |
|  | |

1. 数据格式

传输数据格式见下表。

表 3‑4传输数据格式

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类型** | **字段名称** | **数据类型** | **数据内容说明** |
| 数据头 | 总数据长度 | int | 数据内容的总长度 |
| Y测Obs数据长度 | int |  |
| X测Obs数据长度 | int |  |
| Y侧 Nav数据长度 | int |  |
| X侧Nav数据长度 | int |  |
| 数据内容 | Y侧Obs数据内容 | char \* |  |
| X侧Obs数据内容 | char \* |  |
| Y侧Nav数据内容 | char \* |  |
| X侧Nav数据内容 | char \* |  |

表 3‑5观测数据输出文件格式

|  |  |  |  |
| --- | --- | --- | --- |
| **内容** | **字节数** | **格式** | **备注** |
| 周计数 | 4 | 无符号整形 | 北斗周计数 |
| 周内秒计数 | 8 | Double | 周内秒计数（s） |
|  | | | |
| 通道1信息 | 见表 3‑6 | | |
| 通道2信息 | 见表 3‑6 | | |
| **……** | | | |
| 通道24信息 | 见表 3‑6 | | |

说明：数据总长度：4+8+41\*24=996

表 3‑6单通道观测数据格式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **内容** | **变量** | **字节数** | **符号** | **备注** |
| 星号 | starNo | 1 | char |  |
| 伪距 | r0 | 8 | double | m |
| 伪距一阶变化率 | v0 | 4 | float | m/s |
| 伪距二阶变化率 | a0 | 4 | float | m/s2 |
| 伪距三阶变化率 | J0 | 4 | float | m/s3 |
| 载波相位 | phy0 | 4 | float | 弧度 |
| 载波一阶变化率 | pv0 | 4 | float | 弧度/s |
| 载波二阶变化率 | pa0 | 4 | float | 弧度/s2 |
| 载波三阶变化率 | pJ0 | 4 | float | 弧度/s3 |
| 功率 | pow | 4 | float | 以dBm为单位 |

各频点导航电文传输格式见下表。

表 3‑7 I支路导航电文

|  |  |  |  |
| --- | --- | --- | --- |
| **内容** | **字节数** | **格式** | **备注** |
| 周计数 | 4 | 无符号整数 | 北斗周计数 |
| 周内秒 | 8 | Double | 周内秒计数 |
| 24通道映射可见星对应的电文序号 | 24 | 无符号8位整数 | 24通道映射卫星所对应的电文序号。序号从1开始编号，0表示此通道没有卫星，也就没有电文 |
| 可见星1 | 单通道电文，见下表 | | |
| …… | | | |
| 可见星12 | 单通道电文，见下表 | | |
|  |  | | |

表 3‑8 各频点单通道（单星）I电文

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **频点** | **单星** | | **周期（秒）** | **电文速率** | **电文长度**  **(BYTE)** | **备注** |
| **bit** | **BYTE** |
| L1 | 300 | 40 | 6 | 50 | 480 | 40\*12 |
| L2c | 600 | 75 | 12 | 900 | 75\*12 |
| L5 | 600 | 75 | 6 | 100 | 900 | 75\*12 |
| B1 | 300 | 40 | 0.6 | 500/ 50 | 480 | 40\*12 |
| B2 | 300 | 40 | 0.6 |
| B3 | 300 | 40 | 0.6 |
| R1 | 200 | 25 | 2 | 100 | 300 | 25\*12 |
| R2 | 200 | 25 | 2 |
| E1 | 500 | 63 | 2 | 250 | 756 | 63\*12 |
| E5a | 500 | 63 | 10 | 50 |
| E5b | 500 | 63 | 2 | 250 |

#### 与ARM1数据传输

ARM0的功能是向ARM1转发多种数据以及控制命令信息。ARM1与ARM0通过片上RAM连接，在指定内存中开辟一定长度的空间，进行命令和数据传输。

zynq片上有两块RAM，本例用的是位置为0x FFFF1000的一块。具体分配如下：

#define COMM\_BASE 0xFFFF1000

#define COMM\_MAP\_LEN 0xB000

#define COMM\_TX\_FLAG\_OFFSET 0x0000

#define COMM\_TX\_LEN\_OFFSET 0x0001

#define COMM\_TX\_DATA\_TYPE\_OFFSET 0x0005

#define COMM\_TX\_DATA\_OFFSET 0x0006

#define COMM\_TX\_DATA\_END 0x1FFF

#define COMM\_RX\_FLAG\_OFFSET 0x2000

#define COMM\_RX\_LEN\_OFFSET 0x2001

#define COMM\_RX\_DATA\_OFFSET 0x2005

#define COMM\_RX\_DATA\_END 0x20FF

#define COMM\_START\_FLAG\_OFFSET 0x2100

#define COMM\_STOP\_FLAG\_OFFSET 0x2104

#define COMM\_FPGA\_VERSION 0x2108

#define COMM\_CPU1\_VERSION 0x2128

#define COMM\_VERSION\_WRITTEN\_FLAG 0x2148

#define COMM\_NAV\_BASE 0x3000

其中传输数据的类型有5种：

**#define** DATA\_CMD 1 //命令

**#define** DATA\_Y\_OBS 2 //y侧obs

**#define** DATA\_X\_OBS 3 //x侧obs

**#define** DATA\_Y\_NAV 4 //y侧电文

**#define** DATA\_X\_NAV 5 //x侧电文

ARM0先把数据写到片上RAM中，然后置传输标志位，并且等待接收标志位被置。ARM1将数据读过去后，将要返回的数据写到COMM\_RX\_DATA\_OFFSET 处然后置接收标志位，ARM1根据数据内容中的数据类型宏来区分收到的数据是何种类型。

传输数据格式见下表。

表 3‑9 ARM0与ARM1间传输数据格式

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类型** | **字段名** | **数据类型** | **数据内容说明** |
| 数据头  data\_head\_buf[5]; | 总数据长度 | int | 数据内容的总长度 |
| Y测Obs数据长度 | int |  |
| X测Obs数据长度 | int |  |
| Y侧 Nav数据长度 | int |  |
| X侧Nav数据长度 | int |  |
| 数据内容 | Y侧Obs数据内容 | char \* |  |
| X侧Obs数据内容 | char \* |  |
| Y侧Nav数据内容 | char \* |  |
| X侧Nav数据内容 | char \* |  |

电文和观测数格式与上位机相同。上位机可以先发送数据头，然后再发送数据内容；下位机先接收数据头，解析获取数据头信息，获取数据内容总长度，然后再接收数据内容。

#### ARM0显示

ARM0通过TCP接收命令信息和数据信息，然后从观测数据中提取部分需要实时打印监控的数据，通过串口传递到监测调试口进行显示。串口的传输速率是115200。



图3‑3 显示接口示意图

### 程序结构说明

#### 工作流程

主流程图如下图所示。



图3‑4主流程时序图（初样）

1. **主程序main** 
   1. 配置本卡IP地址；
   2. 初始化内存；
   3. 初始化定时器；
   4. 初始化信号量；
   5. 创建数据处理线程；
   6. 主线程进入命令处理循环。
2. **系统启动**
   1. 接收上位机Client的链接请求；
   2. 链接成功后，接收上位机发送的初始化参数；命令参数有：开环闭环类型、内外触发模式、系统周期；给ARM1发送上述初始化参数，同时应答上位机；开始接收观测数据和电文存到各自的缓存队列；全局变量同时计数；完毕后网络任务阻塞等待继续接收数据的信号量；
   3. 给FPGA发送开始时间信息帧；
   4. 给FPGA发送本地时间信息帧；
   5. 系统正常启动运行。
3. **系统运行**
   1. 数据处理线程同步接收一个套接字连接，若此时上位机无数据发送则阻塞在接收函数，若有数据发送，根据接收到的数据头判断所发的数据长度和数据类型；
   2. 根据所获取的信息，在接收相应长度的数据后，将数据放到缓存中；
   3. 待内存标志位COMM\_TX\_FLAG\_OFFSET为1时，将数据写入内存中，发送至ARM1。



图3‑5系统控制流程图

ARM中断设计：

1. ARM0：20ms，内部基带数据处理参考时钟，用于从上位机获取数据，并把数据发送给ARM1，要求上电后即产生此中断信号；
2. ARM1：1ms，用于给FPGA送数据，信号源启动运行后即产生此信号。

ARM0的主要函数有main ()、data\_thread ()、main\_loop ()、process ()等。调用关系如下图所示。



图3‑6 ARM0的main函数调用流程图

#### 文件说明

1. 头文件说明

ARM0工程所有全局变量、全局函数、结构体的声明统一放到global.h文件中，其中包含接口结构体如下。

* **单通道观测量结构体**

**typedef** **struct**

{

**char** starNo; // 星号

**double** r0; // 伪距pseudorange

**float** v0; // 伪距一阶变化率

**float** a0; // 伪距二阶变化率

**float** k0; // 伪距三阶变化率

**float** phy0; // 载波相位

**float** pv0; // 载波一阶变化率

**float** pa0; // 载波二阶变化率

**float** pJ0; // 载波三阶变化率

**float** pow; // 功率

}RNSS\_Main;//共41字节

* + **24通道观测量结构体接口**

**typedef struct**

**{**

**int week\_num;**

**double week\_second;**

**RNSS\_Main rnss\_main[24];**

**}OBS\_CALCDSP;**

* + **电文存储结构体**

**typedef struct**

**{**

**int data\_head\_buf[5];**

**char data\_buf[DATA\_MAX\_LEN];**

**int q\_nav\_length;**

**char q\_nav\_buf[480];**

**}DATA\_SAV\_STRUCT;**

包含宏信息如下：

**#define** DATA\_CMD 1

**#define** DATA\_Y\_OBS 2

**#define** DATA\_X\_OBS 3

**#define** DATA\_Y\_NAV 4

**#define** DATA\_X\_NAV 5

#define COMM\_BASE 0xFFFF1000

#define COMM\_MAP\_LEN 0xB000

#define COMM\_TX\_FLAG\_OFFSET 0x0000

#define COMM\_TX\_LEN\_OFFSET 0x0001

#define COMM\_TX\_DATA\_TYPE\_OFFSET 0x0005

#define COMM\_TX\_DATA\_OFFSET 0x0006

#define COMM\_TX\_DATA\_END 0x1FFF

#define COMM\_RX\_FLAG\_OFFSET 0x2000

#define COMM\_RX\_LEN\_OFFSET 0x2001

#define COMM\_RX\_DATA\_OFFSET 0x2005

#define COMM\_RX\_DATA\_END 0x20FF

#define COMM\_START\_FLAG\_OFFSET 0x2100

#define COMM\_STOP\_FLAG\_OFFSET 0x2104

#define COMM\_FPGA\_VERSION 0x2108

#define COMM\_CPU1\_VERSION 0x2128

#define COMM\_VERSION\_WRITTEN\_FLAG 0x2148

#define COMM\_NAV\_BASE 0x3000

1. sock.c

**功能描述：**

sock.c中函数包括命令处理函数、数据处理函数等。

**函数声明及功能介绍：**

void \*data\_thread(void \*threadid)

int main\_loop()

表3‑10函数声明和内容

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名称** | **函数参数** | **函数返回值** | **函数功能描述** |
| \*data\_thread | void | void | 数据处理函数 |
| main\_loop | void | int | 命令处理函数 |

**函数流程介绍：**

data\_thread函数流程图如下图所示。



图3‑7 data\_thread函数流程图

main\_loop()流程图如下图所示。



图3‑8 main\_loop函数流程图

1. mem\_process.c

**功能描述：**

与内存操作相关的函数集。

**函数声明及功能介绍：**

void send\_stopflag()

void process(int id, char \*buf, int len, int data\_type\_flag)

int init\_mem()

void close\_mem(int fd)

表3‑11函数声明和内容

|  |  |  |  |
| --- | --- | --- | --- |
| **函数名称** | **函数参数** | **函数返回值** | **函数功能描述** |
| send\_stopflag() | void | void | 发送停止信号 |
| Process() | int id, char \*buf, int len, int data\_type\_flag | void | 命令及数据发送到ARM1 |
| init\_mem() | void | int | 初始化内存 |
| close\_mem(int fd) | void | void | 关闭内存句柄 |

void process(int id, char \*buf, int len, int data\_type\_flag)：

将buf所指地址的Len长度的数据写入mm+COMM\_TX\_DATA\_OFFSET开始的内存供ARM1读取。并将data\_type\_flag值写入mm+COMM\_TX\_DATA\_TYPE\_OFFSET。如果是data\_type\_flag是DATA\_CMD命令，那么除了将DATA\_CMD写入mm+COMM\_TX\_DATA\_TYPE\_OFFSET以外，还需要准备返回上位机的数据；如果data\_type\_flag 为DATA\_Y\_OBS、DATA\_X\_OBS、DATA\_Y\_NAV、DATA\_X\_NAV等，则不需要返回给上位机数据。

## ARM1方案设计

### 数据传输

ARM1与ARM0的数据传输见3.2.1.2节。本章介绍ARM1与FPGA的数据传输。

基带板中ARM1与FPGA通过ZYNQ-7000内部的AXI\_HP\_DMA接口连接，这个接口把ARM计算生成的数据下发给FPGA，本接口具有较高的吞吐量，常用于多数据的高性能DMA，理论吞吐量可达1200M/s。

初始化基带工作参数命令包的格式如下。

表 3‑12 初始化基带工作参数命令包格式

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **字节内容** | **含义** | **默认值** |
| 1 | 0xEB90 | 帧头 | 0xEB90 |
| 2 | 16 | 数据包长度 | 16 |
| 3 | 0x0001 | 命令号 | 0x0001 |
| 4 | 开闭环：0或者1 | 0：开环；1：闭环 | 0 |
| 5 | 内外触发：0或者1 | 0：外触发；1：内触发 | 1 |
| 6 | 工作周期：1/2/3 | 3：5ms；2：10ms；1：20ms | 1 |
| 7 | 启动延迟时间（ms）  1~100 | 系统设置的启动等待时间 | 20 |
| 8 | 频点号1：1~21 | 板卡的频点 |  |
| 9 | 频点号2：1~21 | 板卡的频点 |  |
| 10 | 频点号3：1~21 | 板卡的频点 |  |
| 11 | 频点号4：1~21 | 板卡的频点 |  |
| 12~15 | 0 | 保留 | 0 |
| 16 | 0x146F | 帧尾 | 0x146F |

基带1ms工作数据包的格式见下表。工作参数包详细说明见下表。表中提到的副载波、子码、小数码片针对BOC调制信号，对于非BOC信号，这些值不计算。

表 3‑13 基带工作数据包格式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **数据位宽为16位（15:0）** | | | **是否必须计算** |
| 1 | 帧头(0XEB90) | | | Y |
| 2 | 数据包长度 | | | Y |
| 3 | 接收时间低16位 | | | Y |
| 4 | 接收时间高16位 | | | Y |
| 5 | 通道1命令字低16位   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | [15:8] | [7] | [6] | [5:2] | 1 | 0 | | 卫星号 | 通道开关 | 副载波开关 | 调制方式 | 空 | 单通道功率增强 |   调制方式（4位）定义为：  0000　　单载波  0001　　单I  0010　　单Q  0011　　QPSK  0100　　BOC  通道开关为1位控制   1. 打开通道 2. 关闭通道   副载波开关为1位控制  0关闭副载波  1 打开副载波 | | | Y |
| 6 | 通道1命令字高16位  低8位代表工作状态：  0x1：通道复位，FPGA回到上电复位状态；  0x2：初始化时间，设置开始工作时间，单位为毫秒；  0x3：本地时间，设置本地时间，触发有效后才开始计时，单位为毫秒；  0x4：工作参数，可设置为正常工作，关闭输出，停止工作；  0x6：停止，本通道停止工作，转回到复位状态，arm可以发送初始化时间包（0x2）开始重新工作；  高8位保留 | | | Y |
| 7 | 通道1扩频码初始码片地址 | | | Y |
| 8 | 通道1Q支路电文[15:8] | 通道1I支路电文[7:0] | | Y |
| 9 | 通道1对应码NCO累加控制字[15:0] /通道1I支路对应滤波器地址 | | | Y |
| 10 | 通道1对应码NCO累加控制字[31:16] /通道1Q支路对应滤波器地址 | | | Y |
| 11 | 通道1对应码NCO累加控制字[47:32] | | | Y |
| 12 | 通道1对应码NCO累加控制字[63:48] | | | Y |
| 13 | 通道1对应码NCO初始相位[15:0] /通道1对应速度初始值[15:0] | | | Y |
| 14 | 通道1对应码NCO初始相位[31:16] /通道1对应速度初始值[31:16] | | | Y |
| 15 | 通道1对应码NCO初始相位[47:32] /通道1对应速度累加值[15:0] | | | Y |
| 16 | 通道1对应码NCO初始相位[63:48] /通道1对应速度累加值[31:16] | | | Y |
| 17 | 通道1载波累加控制字[15:0] | | | Y |
| 18 | 通道1载波累加控制字[31:16] | | | Y |
| 19 | 通道1载波累加控制字[47:32] | | | Y |
| 20 | 通道1载波累加控制字[63:48] | | | Y |
| 21 | 通道1载波初始相位1[15:0] | | | Y |
| 22 | 通道1载波初始相位1[31:16] | | | Y |
| 23 | 通道1载波初始相位1[47:32] | | | Y |
| 24 | 通道1载波初始相位1[63:48] | | | Y |
| 25 | 通道1载波初始相位2[15:0] | | | Y |
| 26 | 通道1载波初始相位2[31:16] | | | Y |
| 27 | 通道1载波初始相位2[47:32] | | | Y |
| 28 | 通道1载波初始相位2[63:48] | | | Y |
| 29 | 通道1发送时间低16位 | | | Y |
| 30 | 通道1发送时间高16位 | | | Y |
| 31 | 通道1功率控制字低16位 | | | Y |
| 32 | 通道1功率控制字高16位 | | | Y |
| 33 | 通道1数据通道子码初始位置 | | | N |
| 34 | 通道1导频通道子码初始位置 | | | N |
| 35 | [15:8]代表20ms计数，[7:0]代表1.5s计数 | | | N |
| 36 | 通道1副载波BOC(11,1)初始码片地址 | | | N |
| 37 | 通道1副载波BOC(6,1)初始码片地址 | | | N |
| 38 | [15:8]代表通道1副载波BOC(1,1)初始相位（小数码片0~5） | | [7:0]通道1扩频码初始相位（小数码片0~11） | N |
| 39-42 | 保留位用于扩展（64位，8字节） | | | N |
| 43-916 | 序号5~42重复23次，总共24个通道数据。 | | | Y |
| 917 | 帧尾(0X146F) | | | Y |

ARM1生成基带参数所用的观测数据格式如下（与见表 3‑6相同）。

表 3‑14 基带工作数据包所用观测数据格式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **内容** | **变量** | **字节数** | **符号** | **备注** |
| 周数 | week\_num | 4 | Int |  |
| 周内秒数 | week\_second | 8 | Double |  |
| 观测量信息 | rnss\_main[24] | 984 | RNSS\_Main | 24通道 |

每通道观测数据格式如下。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **内容** | **变量** | **字节数** | **符号** | **备注** |
| 星号 | starNo | 1 | char |  |
| 伪距 | r0 | 8 | double | m |
| 伪距一阶变化率 | v0 | 4 | float | m/s |
| 伪距二阶变化率 | a0 | 4 | float | m/s2 |
| 伪距三阶变化率 | J0 | 4 | float | m/s3 |
| 载波相位 | phy0 | 4 | float | 弧度 |
| 载波一阶变化率 | pv0 | 4 | float | 弧度/s |
| 载波二阶变化率 | pa0 | 4 | float | 弧度/s2 |
| 载波三阶变化率 | pJ0 | 4 | float | 弧度/s3 |
| 功率 | pow | 4 | float | 以dBm为单位 |

### 程序结构说明

#### 工作流程

ARM1模块的程序流程框图如下图所示。



图3‑9 ARM1程序的流程框图

ARM1程序工作流程：

1. 系统复位，加载程序开始。
2. 接收命令参数帧，提取开环闭环、触发模式、系统周期。
3. 接收第一个观测数据帧，从该帧中提取时间信息作为系统启动时间的控制参数，处理观测数据帧，产生20个输出帧，并向FPGA发送初始化参数。
4. 接收一个电文帧。
5. 接收一个观测数据帧，向FPGA发送开始时间信息帧。
6. 读取FPGA中断，在FPGA给出的第一个中断后，发送本地时间帧。
7. 读取FPGA中断，向FPGA发送观测数据输出帧和电文信息。
8. 在处理数据时，因为数据和电文是保持同步的，即计算30包观测数据要用掉同步的一包（600ms）电文，所以在处理完30包观测数据之后，检查是否有新的电文。
9. 循环接收和处理观测数据信息和电文信息，重复7）、8）两个步骤。



图3‑10 ARM1的main函数调用流程图

ARM1的数据处理算法如下。

1个输入数据帧包含12个子块，每个子块对应一个通道（卫星）的观测数据，子块结构相同。ARM1对每个子块进行1ms处理，处理结果组成新的帧放到输出缓冲区中。处理后，1个输入数据帧产生20个输出帧。对于每个输入帧，重复计算20次，生成20个输出帧。计算过程如下。

* 1. **每帧基本变量计算**

每帧数据计算时用到的基本参数为时间、伪距、速度、加速度，其计算方式如下，其中n为当前帧序号，取值0~19：

（1）

（2）

（3）

（4）

* 1. **每帧基带参数计算**

1. 从时间中取出毫秒时间，转换成双精度浮点数，仍记为。
2. 将伪距转换为双精度浮点数，并乘以10-7，记为，注意原为8字节整数。
3. 计算，所得为双精度浮点数，可能为负（单位为ms）；
4. 将向下取整位32bit整型数，得到，可能为负；
5. 计算I路扩频码地址其中CodeLenI为I路扩频码长；
6. 计算：I路扩频码初始相位=，即滤波器位置；
7. 计算速度初值： {扩频码初始相位-[扩频码初始相位]}×262144，即量化位宽218的无符号数。
8. 计算：I路扩频码累加值=，量化位宽18bit的有符号数；
9. 载波参数计算：，其中f为载波频率，为伪距；
10. 载波参数计算：其中v为速度，fs为采样率；
11. 载波参数计算：其中a为加速度；
12. 载波相位累加值=，如果结果为负，则将结果加1，直到结果为正，再乘以；
13. 载波相位初值1=，如果结果为负，则将结果加1，直到结果为正，再乘以；
14. 载波相位初值2 =，如果结果为负，则将结果加1，直到结果为正，再乘以264；
15. 计算：幅度=，结果为18bit无符号数，其中Pow为功率值，Att为功率衰减值。
    1. **电文查找过程**
16. 电文的缓存结构为nav\_data[NAV\_BUF\_NUM]，采用最新的电文时间减去当前观测量时间，差值除以电文的周期，即得到电文缓存相对最新电文的偏移地址cnt\_offset\_code；
17. 电文的偏移地址cnt\_offset\_code及最新电文的缓存地址相减，即可得到当前观测量时间对应的电文所在的缓存地址，仍记为cnt\_offset\_code；
18. 用观测量时间减去电文缓冲区起始时间，再除以1bit电文周期，得到电文指针，从该指针处取出1bit电文，作为I路数据码第1bit；
19. 用减去电文缓冲区起始时间，再除以1bit电文周期，得到电文指针，从该指针处取出1bit电文，作为I路数据码第2bit；
20. 用减去电文缓冲区起始时间，再除以1bit电文周期，得到电文指针，从该指针处取出1bit电文，作为I路数据码第3bit；

#### 文件说明

1. 头文件说明

global.h为本工程的头文件，其中包含了本工程中所有宏的定义，对应于ARM1的输入数据帧和输出帧与其他中间临时帧定义了多个结构体；全局变量以及函数也全部在此头文件中声明。

1. input.c

input.c文件包括RecvCmdDataAndProcess()函数，解析命令及数据，为1ms数据计算做准备。recv\_cmd\_process()函数通过调用get\_param(cmd)获得命令字，处理命令字，生成对FPGA的命令字；接收的数据交由data\_calculate.c中的process\_intdata()函数和code\_gen函数生成1ms数据。

1. data\_calculate.c

data\_calculate.c包含process\_intdata()函数和code\_gen函数，函数为软中断“SWI\_process”的服务子函数。该函数根据算法要求，将从上位机收到的观测数据和导航电文进行参数换算以生成FPGA的工作数据，放入输出缓冲区。

int process\_intdata(void)；

函数介绍：参数计算。

输入参数：从输入数据缓冲区取数据。

输出参数：输出到输出数据缓冲区。

void code\_gen(int xinghao,int tongdaohao,int xuliehao,

long ks,unsigned int \* data\_code\_3bitI,unsigned int \* data\_code\_3bitQ);

函数介绍：电文查找，根据当前的信号、通道号、序列号（时间）去查找对应的3bitI路和Q路电文。

输入参数：

int xinghao，星号

int tongdaohao，通道号

int xuliehao，序列号

unsigned long ks，时间

输出参数：

unsigned int data\_code\_3bitI， I路3bit电文

unsigned int data\_code\_3bitQ， Q路3bit电文

返回值：无

1. output.c

output.c主要进行数据的发送，将计算后的数据传输给基带处理FPGA。

void carry\_data(void \*addr\_input , void \*addr\_output , unsigned short length)

函数介绍：将计算好的参数传输到基带处理FPGA中。

输入参数：

addr\_input：输入地址

addr\_output：输出地址

length：传输长度

输出参数：无

void send\_init\_frame(unsigned int init\_time)

函数介绍：向基带FPGA输出初始化时间，基带FPGA在这个时刻开始进行信号处理。

输入参数：init\_time初始化时间，单位为ms

输出参数：无

返回值 ：无

要求：在调用这个函数之前，首先应该将star\_num\_recod[]进行赋值处理。命令参数为0x02000000

void restart\_out\_init\_frame(unsigned int init\_time,unsigned int init\_addr,unsigned change\_channel\_num)

函数介绍：换星时重新向基带FPGA输出初始化时间。

输入参数：

init\_time初始化时间，单位为ms

init\_addr 初始化地址

change\_channel\_num 改变的通道号

输出参数：无

返回值 ：无

要求：在调用这个函数之前，首先应该将star\_num\_recod[]进行赋值处理。命令参数为0x02000000

void send\_localtime\_frame (unsigned long time)

函数介绍：向基带FPGA输出本地时间，基带FPGA内部时间以这个时刻开始。

输入参数：lc\_time本地时间，单位为ms

输出参数：无

返回值 ：无

命令参数为0x03000000

GPIO2\_input\_isr(void)

函数介绍：FPGA1的中断处理函数，每1ms产生一次中断。

输入参数：无

输出参数：无

说明：这个中断处理函数中包含时序控制，只有先配置初始时间帧才能再进行本地时间帧的配置。

## FPGA逻辑方案设计

FPGA基带（PL）逻辑图如下图所示。



图3‑11基带模块框图

PS实时接收来自于上位机的电文和观测量，并将电文和观测量以1ms为单位生成PL运算使用的计算参数，包括如电文、码初始值、电文内容、FIR累加器初始值、FIR累加器累加值等。

PL以1ms为单位，独立计算各通道的仿真数据，实现多通道的基带数据仿真，实现多普勒和时延仿真。PL产生参考时钟、内部触发信号和1kHz、1Hz信号。时钟驯服过程中PL计算内部时钟与外部1PPS间误差，同时将数据反馈至PS。

DAC将数字基带信号调制到预设频段。

### 功能模块设计

基带模块化框图如下图所示。



图3‑12系统模块框图

PL可分为控制、计算、时钟管理、采集四个功能单元。

1. 控制单元完成启动时序控制，生成同步触发信号，向控制PS发送周期性20ms中断信号；
2. 计算单元工作参数解析，将工作参数包解析成各通道参数，数据解析完毕后向计算PL发送周期性1ms中断信号；单路仿真信号的仿真，包括伪码生成、电文调制、多普勒调制及功率制等；多路数字信号合路。
3. 时钟管理单元统计外部1PPS与内部工作时钟间误差，并将结果反馈给PS以对时钟模块进行微调。
4. 采集单元为扩展功能，可采集DAC输出信号，存储在SD后以便数字接收机解调或者分析零值等。

### 接口协议

模块对外接口见下表。

表3‑15 对外接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **接口名称** | **信号/数据方向** | **对应连接器** |
| 1 | 控制参数 | PS→PL，PS→DSP | 数据接口 |
| 2 | 初始化参数包 | PS→PL | 数据接口 |
| 3 | 工作数据包 | PS→PL | 数据接口 |
| 4 | 1Hz | PL→外部 | 物理接口 |
| 5 | 外部1PPS输入 | 外部 → PL | 物理接口 |
| 6 | 外部时钟输入 | 外部 → PL | 物理接口 |
| 7 | Trig\_in | 外部 → PL | 物理接口 |
| 8 | Trig\_out | PL→外部 | 物理接口 |
| 9 | 1ms和20ms中断输出 | PL→PS | 内部 |
| 10 | 启动信号 | PS→PL | 内部 |
| 11 | 系统复位 | PS→PL | 内部 |
| 12 | 数字基带信号 | PL→DAC | 数据接口 |

PS与PL之间使用AXI总线完成数据传输。接口定义见下表。其中初始化基带工作参数命令包见表 3‑12。

其中FPGA与DAC间的数字基带信号：11bit；数据速率40.92Msps。

表3‑16 PS与PL之间数据传输接口

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **命令包名称** | **工作阶段** | **发送内容** | **方向** | **基地址统**  **偏移地址** | **接口长度详细格式** | **命令解析及说明** |
| 自检状态包 | 自检阶段 | 读取基带软件版本号，板卡号，自检状态 | PL→PS |  |  | [3:0]指示自检状态  [0]:1表示时钟锁定；[1]:表示射频本振锁定；  [9:4]指示硬件板卡编号  [15:10]指示软件版本号 |
| 初始化包 | 初始化阶段 | 初始化基带工作参数命令包 | PS→PL |  |  | 见表 3‑12，只发一次 |
| 开始时间包 | 启动准备阶段 | 初始化基带开始时间包 | PS→PL |  |  | 只发一次 |
| 本地时间包 | 启动准备阶段 | 初始化基带本地时间包 | PS→PL |  |  | 只发一次 |
| RNSS工作参数包 | 工作阶段 | 基带1ms工作数据包 | PS→PL |  |  | 循环发送，1ms一次 |
| 启动命令 | 启动阶段 | 启动命令 | PS→PL |  |  | 写0x1执行启动命令操作，其余无效，自动清零 |
| 软复位命令 | 软复位阶段 | 软复位 | PS→PL |  |  | 写0x1执行软复位操作，其余无效，自动清零 |
| 大小信号切换命令 | 工作阶段 | 大小信号切换 | PS→PL |  |  | 表示衰减量，0不衰减，步进0.5dB，衰减量为该参数\*0.5dB。 |
| 休眠控制 | 任意阶段 | 休眠控制指令 | PS→PL |  |  | 使相应模块休眠 |
| 版本控制 |  |  | PL→PS |  |  | [7:0] FPGA版本号  [15:8] ARM版本号 |
| 时钟模式输出 |  |  | PL→PS |  |  | 0：输出1Hz  1：输出10.23MHz  默认：0 |

# 风险分析及应对策略

无

# 其他设计说明

在开始阶段为调试初样机基带处理的需要，可预先生成导航电文和观测量等数学仿真数据并以文件的形式保存在上位机，由上位机读取并以下位机中断请求的方式下发给ARM0进行基带信息的后续处理，所需调试条件见下表。

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **条 件** | | **备注** |
| 1 | **硬件** | XCZU7EV及Linux相关环境部署 |  |
| 2 | 时钟模块并与基带处理、DAC连接完备 |  |
| 3 | **软件** | 数学仿真数据和命令参数 | 命令参数也可预先固化在下位机 |
| 4 | 读取并发送数学仿真数据的调试软件，且需具备网络通信功能 |  |
| 5 | RPU核时钟同步与中断响应功能实现 | 响应FPGA来的20毫秒中断，然后以软中断的方式与上位机进行同步。 |

# 附录 数仿数据文件格式

每个数据文件（包括观测数据文件、电文文件、载体轨迹文件）由文件头和数据组成，其中文件头用来说明文件版本信息、数据更新率等。观测数据文件、电文文件等文件的数据格式在3.2.1.1节已经介绍，这里只介绍文件头的格式及载体轨迹数据格式。

## 数据文件头

|  |  |  |  |
| --- | --- | --- | --- |
| **内容** | **字节数** | **格式** | **备注** |
| 版本信息1 | 4 | Float | 主版本号：如：3.1表示3.1 |
| 版本信息2 | 4 | Float | 次版本号：1.6 版本 |
| 数据更新周期 | 1 | Byte | 5,10,20 |
| 通道数 | 1 | Byte | 16、18、24等值 |
| 保留位 | 10 | --- | 扩展 |

## 载体轨迹数据文件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **内容** | **名称** | **单位** | **数据类型** |
|  | week | 北斗周计数 | ms | uint32 |
|  | secondsOfWeek | 周内秒计数 | S | double |
|  | X | ECEF位置坐标 |  | double |
| Y |  |  | double |
| Z |  |  | double |
|  |  | 速度 |  | double |
|  |  | double |
|  |  | double |
|  |  | 加速度 |  | double |
|  |  | double |
|  |  | double |
| 14 |  | 加加速度 |  | double |
|  |  | double |
|  |  | double |
| 15 | Roll | 滚动角 | 弧度 | double |
| 16 | Pitch | 俯仰角 | 弧度 | double |
| 17 | Yaw | 偏航角 | 弧度 | double |
| 18 | Roll Rate | 滚动角一阶变化率 | 弧度/ s | double |
| 19 | Pitch Rate | 俯仰角一阶变化率 | 弧度/ s | double |
| 20 | Yaw Rate | 偏航角一阶变化率 | 弧度/ s | double |
| 21 | Roll Jerk | 滚动角二阶变化率 | 弧度/ s2 | double |
| 22 | Pitch Jerk | 俯仰角二阶变化率 | 弧度/ s2 | double |
| 23 | Yaw Jerk | 偏航角二阶变化率 | 弧度/ s2 | double |

## 观测数据接口文件

观测数据格式见下表。

表 6‑1 观测数据的后缀名

|  |  |
| --- | --- |
| **类型** | **数据文件后缀** |
| BDS系统观测数据 | \*.B1OBS，\*.B2OBS,\*.B3OBS,\*.BDS |
| GPS频点观测数据 | \*.L1OBS, \*.L2OBS, \*.L5OBS |
| GLONASS频点观测数据 | \*.R1OBS，\*.R2OBS |
| GALILEO频点观测数据 | \*.E1OBS，\*.E5OBS，\*.E6OBS |

表 6‑2观测数据输出文件格式

|  |  |  |  |
| --- | --- | --- | --- |
| **内容** | **字节数** | **格式** | **备注** |
| 周计数 | 4 | 无符号整形 | 北斗周计数 |
| 周内秒计数 | 8 | Double | 周内秒计数（s） |
| 卫星号 | 24 | 无符号整数 | 24通道对应可见星。按照该历元中所含有的卫星填写其编号，0表示该位置卫星为空。 |
|  | | | |
| 通道1信息 | 见表 3‑6 | | |
| 通道2信息 | 见表 3‑6 | | |
| **……** | | | |
| 通道24信息 | 见表 3‑6 | | |

说明：数据总长度：4+8+24+63\*24=1548

表 6‑3单通道观测数据格式

|  |  |  |  |
| --- | --- | --- | --- |
| **内容** | **字节数** | **格式** | **备注** |
| 卫星号 | 1 | UINT8 | 1~32；0标示当前通道没有卫星 |
| 多径号 | 1 | UINT8 | 0：主径，1~多径 |
| 多径类别 | 1 | UINT8 | 1：地面反射  2：定点反射  3：固定偏差 |
| 伪距 | 8 | double | 以m为单位 |
| 伪距速度 | 4 | float | 以m/s为单位 |
| 伪距加速度 | 4 | float | 以m/s2为单位 |
| 伪距加加速度 | 4 | float | 以m/s3为单位 |
| 载波相位 | 4 | float | 以弧度为单位  （一载波周内小数部分） |
| 载波相位变化率 | 4 | float | 弧度/s为单位 |
| 载波相位二阶变化率 | 4 | float | 弧度/s2为单位 |
| 载波相位三阶变化率 | 4 | float | 弧度/s3为单位 |
| 电离层 | 4 | float | m |
| 对流层 | 4 | float | m |
| 钟差 | 4 | float | s |
| 仰 角 | 4 | float | 度 |
| 方位角 | 4 | float | 度 |
| 信号功率 | 4 | float | 以dBm为单位 |

## 电文数据接口文件

各频点导航电文传输格式见下表。

表 6‑4 I支路导航电文

|  |  |  |  |
| --- | --- | --- | --- |
| **内容** | **字节数** | **格式** | **备注** |
| 周计数 | 4 | 无符号整数 | 北斗周计数 |
| 周内秒 | 8 | Double | 周内秒计数 |
| 24通道映射可见星对应的电文序号 | 24 | 无符号8位整数 | 24通道映射卫星所对应的电文序号。序号从1开始编号，0表示此通道没有卫星，也就没有电文 |
| 可见星1 | 单通道电文，见下表 | | |
| …… | | | |
| 可见星12 | 单通道电文，见下表 | | |
|  |  | | |

表 6‑5 各频点单通道（单星）I电文

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **频点** | **单星** | | **周期（秒）** | **电文速率** | **电文长度**  **(BYTE)** | **备注** |
| **bit** | **BYTE** |
| L1 | 300 | 40 | 6 | 50 | 480 | 40\*12 |
| L2c | 600 | 75 | 12 | 900 | 75\*12 |
| L5 | 600 | 75 | 6 | 100 | 900 | 75\*12 |
| B1 | 300 | 40 | 0.6 | 500/ 50 | 480 | 40\*12 |
| B2 | 300 | 40 | 0.6 |
| B3 | 300 | 40 | 0.6 |
| R1 | 200 | 25 | 2 | 100 | 300 | 25\*12 |
| R2 | 200 | 25 | 2 |
| E1 | 500 | 63 | 2 | 250 | 756 | 63\*12 |
| E5a | 500 | 63 | 10 | 50 |
| E5b | 500 | 63 | 2 | 250 |