**CPRI板卡程序设计**

**中国科学院计算技术研究所**

**ICT/CAS-ASTRI 先进无线技术联合研究中心**

**修改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文件编号 | 版本号 | 拟制人/  修改人 | 拟制/修改日期 | 更改理由 | 主要更改内容  （写要点即可） |
| 硬件支撑组 | 0.1 | 曹永敏  李杰 | 2014.5.10 | 创建 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 注1：每次更改归档档（指归档到组内及研究室的档）时，需填写此表。  注2：当第一次归档时，“更改理由”、“主要更改内容”栏写“无”。 | | | | | |

修改列表：

本文档的程序或内容受版权法的保护，未经中科院计算所的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

目录

[1. CPRI协议简介 4](#_Toc387754529)

[1.1 CPRI协议框架 4](#_Toc387754530)

[1.2 CPRI帧结构 4](#_Toc387754531)

[1.2.1 CPRI基本帧 5](#_Toc387754532)

[1.2.2 超帧控制字 6](#_Toc387754533)

[2. 下行链路设计 7](#_Toc387754534)

[2.1 LISU传输信号格式 7](#_Toc387754535)

[2.2 RapidIO接口定义 8](#_Toc387754536)

[2.2.1 RapidIO帧格式 8](#_Toc387754537)

[2.3 CPRI板卡程序设计 10](#_Toc387754538)

[2.3.1 实时性传输的CPRI板卡程序设计 10](#_Toc387754539)

[2.3.1.1 SRIO解帧器设计 11](#_Toc387754540)

[2.3.1.2 下行缓冲器设计 11](#_Toc387754541)

[2.3.2 1.5ms传输一次IQ数据的CPRI板卡程序设计 14](#_Toc387754542)

[2.3.2.1 SRIO解帧器设计 14](#_Toc387754543)

[2.3.2.2 下行缓冲器设计 15](#_Toc387754544)

[2.3.3 组帧模块的设计 17](#_Toc387754545)

[2.3.3.1 CTRL 模块 18](#_Toc387754546)

[2.3.3.2 IQ映射与IQ指示信号映射模块 18](#_Toc387754547)

[2 上行链路的设计 21](#_Toc387754548)

[3.1 CPRI板卡程序设计 22](#_Toc387754549)

[3.1.1 BBU处理定时信息：解帧模块的设计 22](#_Toc387754550)

[3.1.2 BBU处理定时信息：上行缓冲器的设计 24](#_Toc387754551)

[3.1.3 BBU定时信息的获取 25](#_Toc387754552)

[3.1.4 CPRI处理定时信息：解帧模块的设计 25](#_Toc387754553)

[3.1.5 CPRI处理定时信息：上行缓冲器的设计 26](#_Toc387754554)

[3.1.6 SRIO成帧器设计 27](#_Toc387754555)

# CPRI协议简介

为了有效的对无线基站进行产品划分，独立的发展无线设备控制中心（简称REC）及无线设备（简称RE）技术，爱立信、华为、NEC、北电网络及西门子公成立了通用公共无线接口（CPRI）联盟，致力于从事无线基站内部REC及RE之间主要接口规范的制定工作。

## CPRI协议框架

CPRI协议支持多种连接方式。可以满足灵活的网络拓扑结构。几条CPRI 链路可以用来增加系统容量来满足有很多天线和载波的大系统配置要求，这就要求一条CPRI 链路能够完整携带某个天线和某个天线载波的一个IQ 数据流（尽管允许同一个天线载波可以同时在几个链路上传输）。一个REC可以为几个RE提供服务，下图是REC与RE之间的连接，本设计LISU与BBU的连接就是通过这种拓扑方式连接，拓扑方式还可以是链式的，星型，环形等等，这里不做介绍了。

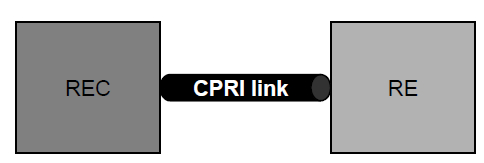
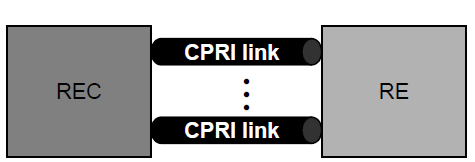
 

图1-2 REC与RE之间的链接

## CPRI帧结构

一个UMTS无线帧由150个超帧组成，一个超帧由256个基本帧组成。CPRI帧层次结构如图1-4所示，Z:超帧数；X：基本帧数；W：基本帧里的字数；Y：每个字里的byte数，W=0为控制字。Z、X、W、Y、B的范围如表1-1.

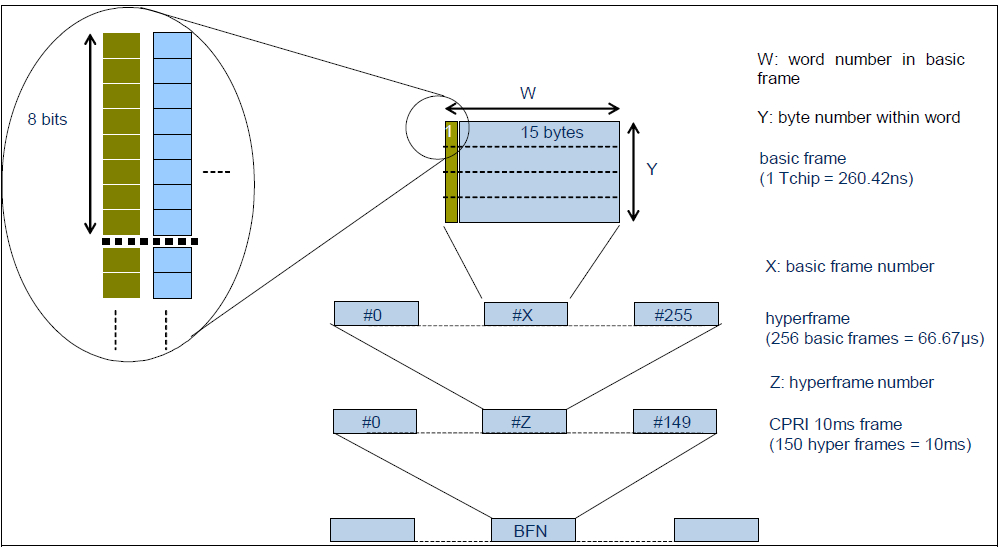
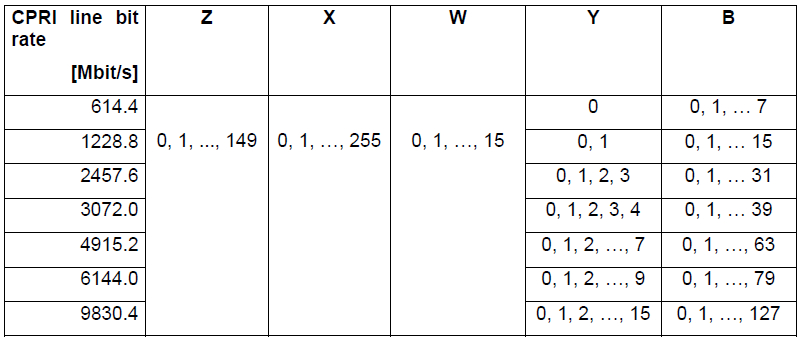


图1-4 CPRI帧层次结构

表1-1 Z、X、W、Y、B取值范围



### CPRI基本帧

基本帧长：1 Tc＝1/3.84 MHz＝260.416667ns一个基本帧包含16个字：W=0…15，字长T 依赖于线比特率。W=0 用于控制字； bit：B=0…T-1；byte＝8bit，Y＝0：B=0…7，Y=1：B=8…15，……。W=1……15用于IQ数据。

CPRI数据在传输过程中首先传送基本帧的控制BYTES，然后依次传送IQ数据。基本帧结构因线比特率而异。图1-5，图1-6描述了不同线比特率的基本帧结构。

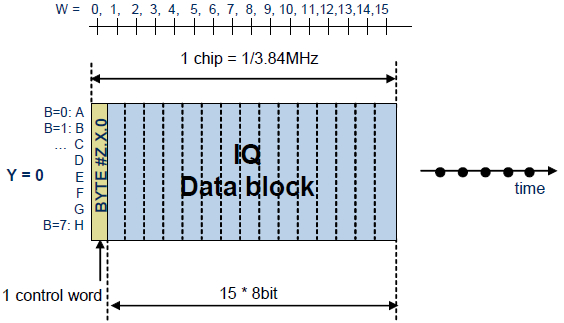


图1-5 线比特率614.4Mb/s的基本帧结构

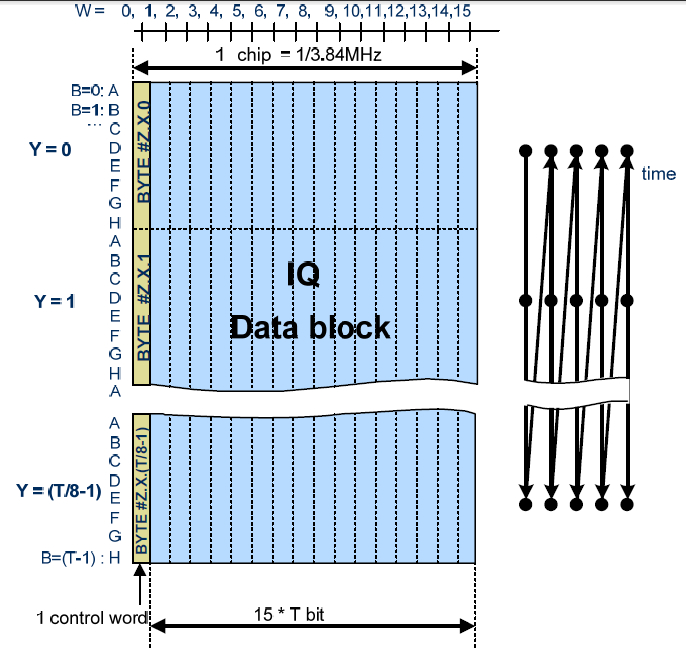
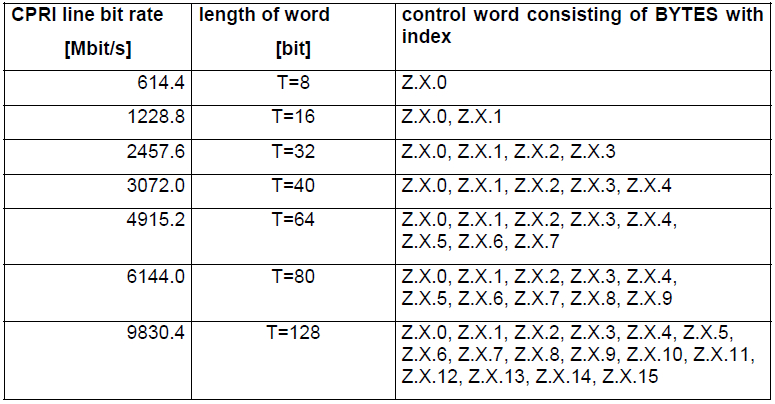


图1-6 不同线比特率基本帧结构（T的值在表1-2中定义）

表1-2 控制字的长度



### 超帧控制字

逐级嵌套的256个控制字按每四个字一组编为64个子信道。子信道序号Ns=0..63，每个

子信道里的控制字序号Xs=0..3，一个嵌套里的控制字序号X=Ns+64\*Xs。图 1-7 阐述了子信道里控制字的组织情况。表1-3说明了每个子信道控制字的意义。

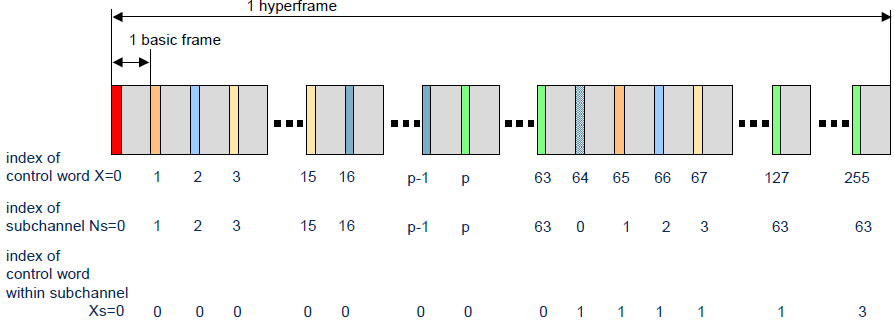
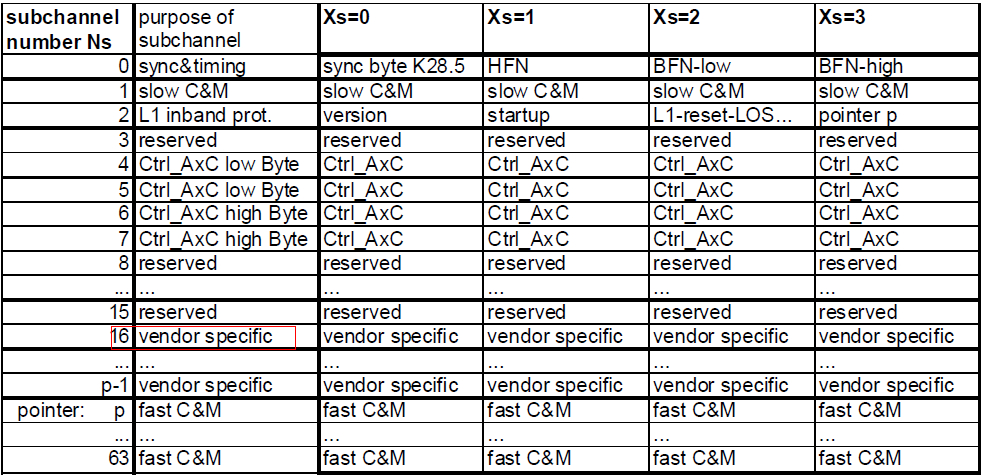


图1-7 超帧的控制字与子信道说明

表1-3 超帧中的控制字



# 下行链路设计

下图2-1是整个下行链路实现的功能框架图。根据CPRI协议的定义，本设计中的LISU板卡作为REC，4个BBU作为RE。在下行发射链路上，每块LISU板卡发射60个通道数据。LISU通过RapidIO接口实现与CPRI板卡的数据交互。SRIO帧进入CPRI板卡，封装成CPRI帧，经光纤发送到BBU。

图2-1 下行链路功能框架图

## LISU传输信号格式

LISU板卡会发送60个通道的数据信息，每个信道发送的信息包括IQ指示信号和IQ编号，IQ指示信号包括IQ编号、帧号、时隙号、指示带宽。IQ编号用于指示IQ数据发送到哪块BBU板卡；指示带宽指示数据传输比特率；时隙号和帧号是IQ数据的定时信息，定义1.5ms一个时隙号，60ms一个帧号，时隙号数值为0---39；帧号数值为1到313344。

下图2-2详细定义了LISU板卡发送的IQ数据的参考时钟clk，数据有效标志和IQ数据之间的对应关系，参考时钟clk采用96Mhz。每个数据有效标志位周期传输IQ数据32bit（I:16bit,Q:16bit）。



图2-2 LISU输出信号

每个信道数据的指示带宽有五种，分别为：

* 1/（64KHz\*1）
* 1/（64KHz\*2）
* 1/（64KHz\*5）
* 1/（64KHz\*10）
* 1/（64KHz\*15）

LISU输出信号的指示带宽变化时间为1.5ms的整数倍。LISU每个通道输出信息是并行的，彼此不关联，他们输出信息的时刻都是不定时的。为了实现LISU信息的实时性传输，使用RapidIO实时性的捕捉每个通道的传输信息。

## 2.2 RapidIO接口定义

### 2.2.1 RapidIO帧格式

RapidIO协议包由可变包头和数据负载组成，其中数据负载根据需求由最小1个字节到最大256个字节组成，如下图2-3，根据RapidIO协议定义的帧格式知道包头是可变的。SRIO定义两种帧格式：HELLO 帧结构，SRIO Stream帧结构，本设计中使用HELLO 帧结构。HELLO 帧的包头包括：Source ID、tt、ftype、size、adress等一共64bit。下行链路发射的IQ数据全部作为SRIO帧数据负载中的传输内容（红框中标出）。

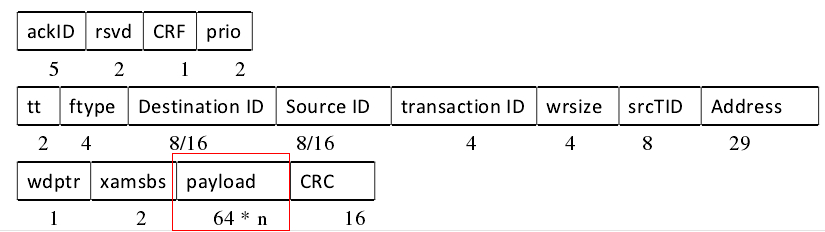


图2-3 RapidIO包结构

RapidIO接口数据同过两种方式传送LISU数据：1、实时性的将LISU的IQ数据发送到CPRI板卡；2、每隔1.5ms发送一次LISU数据到CPRI板卡。

实时性传输定义的SRIO帧结构见下图2-4a。一个包发送一路IQ，每个包负载64bit\*2数据，其中一个64比特传输IQ编号、时隙号、指示带宽和IQ数据，另一个64比特传输该路信息对应的帧号。



图2-4a 实时性传输的RapidIO帧结构

1.5ms定时传输定义的SRIO帧结构如下图2-4b所示（指示带宽1/（64KHz\*1）对应的SRIO包结构），每个包最多负载64\*32字节。如果SRIO包传完一路数据，下一个SRIO包传输另一路数据，一个时隙内的0---59路数据依次通过SRIO包传输。指示带宽不同，封装成的SRIO包个数也不同，指示带宽对应的SRIO包个数如下：

* 1/（64KHz\*1）对应2个SRIO包
* 1/（64KHz\*2）对应4个SRIO包
* 1/（64KHz\*5）对应8个SRIO包
* 1/（64KHz\*10）对应16个SRIO包
* 1/（64KHz\*15）对应23个SRIO包



图2-4b 1.5ms定时传输的RapidIO帧结构

## 2.3 CPRI板卡程序设计

针对实时性传输数据和1.5ms传输一次数据设计了不同的CPRI板卡程序实现方案。

### 2.3.1 实时性传输的CPRI板卡程序设计

下行链路的实时性传输的CPRI板卡程序设计的总体结构框图如图2-5，主要由四部分组成，分别为解帧器，下行缓冲器，CPRI组帧器，CPRI的IP核，。在下行链路，解帧器将串行的SRIO帧解帧，解帧后的数据缓存到下行缓冲器，下行缓冲器中的IQ数据传输到组帧模块，将IQ数据封装成CPRI帧，最后通过CPRI的IP核将CPRI帧发送出去。



图2-5 下行链路CPRI板卡程序结构框图

CPRI板卡将来自LISU的120路数据通过四条CPRI链路分发给四个BBU，本设计实现SRIO解帧器到下行缓冲器的动态映射（图2-5红框），下行缓冲器到组帧模块的静态映射（图2-5蓝框）。SRIO解帧器通过IQ编号实现到下行缓冲器的动态映射，下行缓冲器分配四组缓存单元。如果要求IQ数据发送到BBU1，SRIO解帧后的数据缓存到缓冲器的1组缓存单元，1组缓存单元的数据静态映射到组帧#1；如果要求IQ数据发送到BBU2，SRIO解帧后的数据缓存到缓冲器的2组缓存单元，2组缓存单元的数据静态映射到组帧#2；如果要求IQ数据发送到BBU3，SRIO解帧后的数据缓存到缓冲器的3组缓存单元，3组缓存单元缓冲的数据静态映射到组帧#3；如果要求IQ数据发送到BBU4，SRIO解帧后的数据缓存到缓冲器的4组缓存单元，4组缓存单元缓冲的数据静态映射到组帧#4。组帧#1通过CPRI链路分发给BBU1，组帧#2通过CPRI链路分发给BBU2，组帧#3通过CPRI链路分发给BBU3，组帧#4通过CPRI链路分发给BBU4。

#### 2.3.1.1 SRIO解帧器设计

解帧器的参考时钟clk采用RapidIO的采样时钟。解帧器根据SRIO帧的IQ编号将IQ数据及IQ指示信号发送到缓冲器对应的地址单元，SRIO帧携带的帧号发送到组帧模块的控制信息中传输。

下图2-6解释了这个过程，假设RapidIO接口输出的三组SRIO帧，第一组与第二组SRIO帧携带的IQ数据要发送到BBU1，第三组SRIO帧携带的IQ数据要发送到BBU1。第一组与第二组IQ数据缓存到缓冲器的0-29地址单元，第三组IQ数据缓存到缓冲器的30-59地址单元。这样解帧后的SRIO帧就被缓存到缓冲器中等待传输。



图2-6 解帧器实现方式

#### 2.3.1.2 下行缓冲器设计

下行缓冲器主要有三部分组成： Xilinx提供的真实双端口RAM、FIFO和ADDRA映射模块，如图2-7所示。真实双端口RAM负责SRIO帧的写入和读出，FIFO负责为真实双端口RAM的端口 B提供读地址，ADDRA映射模块实现SRIO解帧数据到RAM的动态映射。

图2-7 下行缓冲器的实现框图

ADDRA映射模块的详细设计如下：IQ编号用来指示IQ数据发送到哪个BBU板卡。ADDRA映射模块的实现方式如下图，ADDRA\_REG1，ADDRA\_REG2，ADDRA\_REG3，ADDRA\_REG4是四个寄存器，ADDRA\_REG1初始化0，ADDRA\_REG2初始化30，ADDRA\_REG3初始化60，ADDRA\_REG4初始化90。假设IQ编号指示当前IQ数据发送到BBU1，那么RAM端口A的写地址ADDRA等于ADDRA\_REG1，如果下一组IQ也发送到BBU1，ADDRA\_REG1自加1，ADDRA等于ADDRA\_REG1；如果下一组IQ发送到BBU2, 那么RAM端口A的写地址ADDRA等于ADDRA\_REG2。通过四个寄存器的值获得端口A的写地址ADDRA。



图2-8 ADDRA映射模块的实现框图

FIFO的详细设计如下：FIFO的写时钟采用SRIO的系统时钟,读时钟采用的时钟是CPRI模块的系统时钟。FIFO的数据宽度设置为16bit，数据深度设置为128，ADDRA映射模块输出的数据输入到FIFO的写端口。

FIFO的实现方式如下图2-9所示，当有SRIO帧传输时，FIFO的写使能信号置于高电平，FIFO开始写入ADDRA映射模块的输出数据，等待一个CPRI基本帧的时间，开始从FIFO读数据。FIFO的读使能信号由组帧模块控制，如果IQ数据要发送到BBU1，读使能信号由组帧模块#1控制。由于缓冲器的数据是64bit，CPRI的IP核是以一个时钟16bit的速率传输，所以FIFO的读使能信号要在一个读时钟周期置高后，下三个读时钟周期置低电平，读使能信号以245.76MHZ/4的频率处理缓冲数据（245.76MHZ是CPRI的系统时钟），FIFO的读数据输入到端口B的读地址ADDRB。根据读地址ADDRB，把缓冲器的读出数据输入到相应的组帧模块，输入到几号组帧模块在图2-5中标识了。



图2-9 下行缓冲器的FIFO实现框图

真实双端口RAM的详细设计如下：真实双端口RAM的写时钟CLKA采用SRIO的系统时钟，读时钟CLKB采用的时钟是CPRI模块的系统时钟。真实双端口RAM的端口A与端口B的数据宽度都设置为64bit，数据深度设置为120。ADDRA映射模块处理后的数据输入到RAM端口A的地址口，SRIO解帧模块处理后的数据输入到端口A的写入数据口。

RAM的实现方式如下图2-10所示，端口A与端口B的地址初始化为0，120个地址单元缓存数据初始化为0。当无SRIO帧传输时，真实双端口RAM的两个端口都处于读出状态，组帧模块不从缓冲器当中读取数据，如图2-9。当有SRIO帧传输时，缓冲器的读写方式如下图2-11所示，端口A一直是写状态，一直到无SRIO帧传输，跳转到读状态。端口B设置为读状态，当FIFO的读使能信号有效时，FIFO的读出数据作为端口B的读地址，组帧模块从端口B中读出数据封装成CPRI帧，然后将缓存此数据的地址单元回填数据‘0’。 当端口A的写使能无效，FIFO无数据读出，端口B的读地址单元缓冲的数据为0时，组帧模块不在从缓冲器中读取数据，端口A与端口B的地址置0。



图2-10 真实双端口RAM的读写方式

### 2.3.2 1.5ms传输一次IQ数据的CPRI板卡程序设计

下行链路的1.5ms传输一次IQ数据的CPRI板卡程序设计的总体结构框图如图2-11，主要由四部分组成，分别为解帧器，下行缓冲器，CPRI组帧器，CPRI的IP核，。在下行链路，解帧器将串行的SRIO帧解帧，解帧后的数据缓存到下行缓冲器，下行缓冲器中的IQ数据传输到组帧模块，将IQ数据封装成CPRI帧，最后通过CPRI的IP核将CPRI帧发送出去。



图2-11 CPRI板卡程序设计的总体结构框图

CPRI板卡将来自LISU的120路数据通过四条CPRI链路分发给四个BBU，本设计实现SRIO解帧器到下行缓冲器的动态映射。SRIO解帧器通过IQ编号实现到下行缓冲器的动态映射，下行缓冲器分配四组缓存单元。如果要求IQ数据发送到BBU1，SRIO解帧后的数据缓存到缓冲器的1组缓存单元；如果要求IQ数据发送到BBU2，SRIO解帧后的数据缓存到缓冲器的2组缓存单元；如果要求IQ数据发送到BBU3，SRIO解帧后的数据缓存到缓冲器的3组缓存单元；如果要求IQ数据发送到BBU4，SRIO解帧后的数据缓存到缓冲器的4组缓存单元。与实时性传输数据相比，只设计了一块组帧模块，在缓冲器缓冲完成一个时隙的数据后，组帧模块把缓存数据封装成CPRI帧。

#### 2.3.2.1 SRIO解帧器设计

解帧器的参考时钟clk采用RapidIO的采样时钟，通过IQ编号动态的缓冲到外部缓冲器中。外部缓冲为SRIO帧每一个64bit数据分配一个地址单元，SRIO帧携带的帧号直接发送到组帧模块。一个时隙内的一路IQ数据至少占用97个缓冲地址。下图2-12是IQ数据发送到BBU1的解帧方式，IQ编号指示此路数据发送到BBU1板卡，将此路的IQ数据依次缓冲到外部缓冲，直到此路数据缓冲完。如果下一路数据仍然发送到BBU1，那么下一个地址单元接着缓冲第二路IQ数据，如果下一路数据发送到其他BBU1，那么下一路IQ数据缓冲到缓冲器分配的相应地址单元。



图2-12解帧器实现方式

#### 2.3.2.2 下行缓冲器设计

1.5ms传输IQ数据的缓冲器使用外部缓冲器，已知外部缓冲的位宽是64位。一个时隙内

LISU每路最多发送46144bit数据，包含IQ数据1440\*32bit，IQ指示信号32\*2bit。为外部缓冲器分配四组缓存单元，分别缓冲传送到BBU1、BBU2、BBU3、BBU4的数据，每组缓冲单元分配21630（30路\*721）个地址单元：

* 0—21629地址单元缓冲发送到BBU1的数据，
* 21630—43259地址单元缓冲发送到BBU2的数据，
* 43260—64889地址单元缓冲发送到BBU3的数据，
* 64890—86519地址单元缓冲发送到BBU4的数据。

将1.5ms内传输的IQ数据以SRIO的系统时钟全部写入缓冲器后，已知SRIO的线速率可达到10Gb/s，传输一个时隙的全部IQ数据，SRIO最多需要0.515ms ((1440\*32+32\*2)\*120bit/10Gbit)。已知CPRI的线速率是4.9152 Gb/s，传输一个时隙的全部IQ数据，CPRI最多需要1.052ms，虽然0.515+1.052=1.567ms>1.5ms，但是120路全部传输最大带宽的IQ数据几乎是不可能的，所以本设计的外部缓冲是可以满足条件的。

外部缓冲器初始化缓冲数据全部为‘0’，通过IQ编号写入相应的缓冲单元，写入方式是每个64bit数据分配一个地址。缓存的方式如下图2-14所示，根据IQ编号确认初始缓存地址，然后每个时钟上升沿，缓存地址自加1，当一个时隙的一路IQ数据缓存完后，此路数据的最后一个IQ数据的缓存地址要用寄存器缓存，下一路数据根据IQ编号写入，如果这两路数据发送到同一块BBU，那么写入的缓存地址是寄存器的缓存数据加1，否则根据IQ编号缓冲到其他的缓存单元。



图2-14下行缓冲器写入的实现框图

一个时隙的数据全部写入缓冲后组帧模块以CPRI的系统时钟读取数据，读取IQ的实现框图如下图2-15所示，从地址0开始读数据，当读到缓冲器中缓冲的数据全部为‘0’时，再从地址21660开始读取数据，依次这样读取，直到从第四组缓存单元的地址中读取到的数据为‘0’，说明一个时隙的数据全部读完。组帧模块从第一组地址单元读取的数据发送到BBU1，从第二组地址单元读取的数据发送到BBU2，从第三组地址单元读取的数据发送到BBU3，从第四组地址单元读取的数据发送到BBU4。

由于组帧模块的数据位宽是16bit，外部缓冲器的数据位宽是64bit，所以每四个读时钟周期读取一次缓存数据，读使能的时序实现如下图2-16所示，数据读出后的地址单元缓冲的数据回填‘0’。



图2-14下行缓冲器读出的实现框图



图2-16 读使能的时序实现

### 2.3.3 组帧模块的设计

本设计中发送的是16比特的IQ（I：16bit，Q：16bit）。系统所用的CPRI线速率是4.9152Gb/s，CPRI协议定义的此线速率对应的基本帧有16个word，第一个word是控制字节，其他15个word都是IQ数据字节，每一个word有64bit。本设计的数据放置如下图2-17所示，每一个基本帧可以配置30个AXC容器。每个容器都是32bit。



图2-17组帧结构图

根据我们定义的CPRI组帧结构， 设计CPRI组帧的实现方式，实现框图如下图2-18所示。组帧模块根据CTRL模块提供的计数信号，把IQ映射模块和IQ指示信号映射模块映射完成的IQ数据和IQ指示数据填充到CPRI帧的载波容器中，同时把控制信息填充到相应的控制字当中。缓冲器每路IQ信息都包含：32bitIQ数据，16bitIQ编号、6bit时隙号，4bit指示带宽以及32bit的帧号，32bitIQ数据在IQ映射模块实现映射，32bit的帧号、16bitIQ编号、6bit时隙号和4bit指示带宽在IQ指示信号映射模块实现映射，同时32bit的帧号作为厂商定制信息填充到控制字当中。



图2-18 CPRI组帧的实现框图

#### 2.3.3.1 CTRL 模块

CTRL模块的作用是根据输入的时钟为组帧模块提供字（word\_cnt）、基本帧（basic\_frame\_num）的计数信息。word\_cnt计数范围是0—15，1—15放置的是IQ数据，0放置的是控制信息，当计数到15时复位为0。basic\_frame\_num计数范围0—255，计数到255时，复位为0。word\_cnt的计数值用于指示填充IQ数据，basic\_frame\_num的计数值的指示用于填充控制信息。

当缓冲器有读出数据，根据IQ编号填充到相应的组帧模块中。word\_cnt等于1，2，3……15时，CPRI基本帧IQ负载区第一列，第二列，第三列……第十五列对应的是我们要插入的交织后的数据位置，word\_cnt等于0时，CPRI基本帧的控制字是插入我们的控制信息，如果没有缓冲数据，不进行数据放置，直接传输。

#### 2.3.3.2 IQ映射与IQ指示信号映射模块

IQ数据映射到CPRI基本帧内需要经过两步：a)AXC容器内的IQ数据映射：在一个AXC容器中，对于每个IQ采样，I采样数据和Q采样数据必须按时间先后交替发送；b)基本帧中AXC容器映射：IQ数据经过交织以后，不同的AXC按规定的映射规则放进基本帧的负载区。IQ映射与IQ指示信号映射的方式是一样的。根据这两步骤，设计了图 2-19 所示的 IQ映射与IQ指示信号映射模块，他们的映射方式是一样的，其中 AxC 容器实现第一步骤，AxC Map模块实现第二步骤。



图2-19IQ数据映射模块

IQ数据映射有两种规则：1、实时性传输的IQ数据映射规则；2、每1.5ms传输一次IQ数据的映射规则。

实时性传输的AXC容器内的IQ数据映射有两种方式，如图2-20所示：a)偶数AXC映射IQ指示信号，奇数AXC映射IQ数据。如果缓冲器的读出数据一直有效，根据word\_cnt的值，将数据映射到相应的AXC容器内，一个基本帧存满之后，按照同样的方式映射到下一个基本帧的容器内。b)一个基本帧的30个容器全部映射IQ数据，下一个基本帧的30个容器全部映射IQ指示信号。这两种方式各有自己的优缺点，方式a的优点是成帧方便，不用对IQ指示信号进行缓存，与IQ信号直接映射到容器中，但是一个基本帧最多只能传输15路IQ数据，效率比较低。方式b的优点是一个基本帧可以传输30路IQ数据，效率高，但是每路IQ数据的指示信息要缓存一个基本帧周期才会映射到AXC容器中，成帧较复杂。选用哪一种方式还有待协商。





图 2-20 实时性传输的IQ数据两种映射方式，上（a）,下（b）

1.5ms传输一次数据的AXC容器内的IQ数据映射规则如下图2-20a，一个时隙内传输的每一路数据占用的的第一个容器用来传输指示信号，其他的容器都用来传输IQ数据，直到该路数据传输完，下一个AXC容器开始传输另一路IQ数据的指示信号，直到1.5ms的LISU数据填充完毕，后面的CPRI基本帧的IQ负载区全部为填充比特“v”。



图 2-20a 1.5ms传输的IQ数据映射规则

IQ 数据在 AxC 容器中经过交织以后，在 AxC Map 模块中把 AxC 容器按映射规则映射到 CPRI 基本帧的用户负载区。按 CPRI 协议的规定，AxC 容器的映射具有两种映射规则可选，如图2-21：（1）packed position：AxC 容器被连续放置，AxC 容器之间没有空白（保留位）。（2）flexible position：AxC 容器的第一位必须放在基本帧中 IQ 数据块的一个偶数序号位上，容器之间留有保留位。本设计选用第一种映射规则，所以就形成了图2-12的数据放置方式。

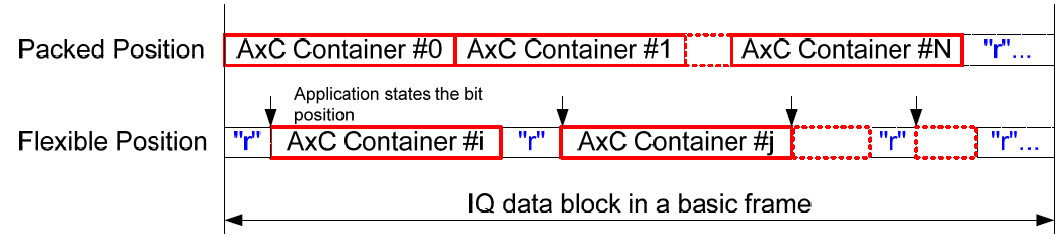


图 2-21 AXC映射规则

根据CPRI协议定义，一个超帧中预留了至少16个（根据以太网的指针会有变化）基本帧的控制信息用于厂商自定义信息，这16个基本帧是：16—19，80—83，144-147和208—211。当basic\_frame\_num的值为这些值时，对应的厂商信息进行传输。在本设计中我们会把帧号作为定制厂商信息传输，当basic\_frame\_num计数到16，word\_cnt是0时，把帧号作为基本帧的控制信息传送。帧号可以通过6个10ms帧后自加1 获得，与传输的帧号进行校对。

# 上行链路的设计

下图3-1是整个上行链路实现的功能框架图。在上行发射链路上，每块BBU板卡发射30个通道数据，根据发射数据的IQ编号发送到LISU板卡的相应通道。LISU通过RapidIO接口实现与CPRI板卡的数据交互。CPRI板卡接收从光纤发送过来CPRI帧，经过CPRI板卡的处理，形成SRIO帧，通过RapidIO接口发送到对应的LISU通道。

图3-1 上行链路功能框架图

## 3.1 CPRI板卡程序设计

在上行链路，CPRI板卡接收来自BBU的数据，通过RapidIO接口，交互给LISU板卡。主要包括CPRI的IP核，解帧模块，上行缓冲器和成帧器，如图3-2。设计实现解帧模块到上行缓冲器的动态映射，通过IQ编号指示IQ数据发送到哪块LISU板卡的哪一路。

上行链路解帧模块和上行缓冲器的设计要考虑两种不同的情况：a) BBU处理定时信息。 CPRI板卡负责将CPRI帧解帧，把解帧后的数据封装成SRIO帧发送到RapidIO接口；b) CPRI板卡负责处理定时信息，BBU的数据发送到CPRI板卡后，CPRI板卡要缓存1.5ms个时间单元后把数据封装成SRIO帧发送到RapidIO接口，这种情况我们要考虑在哪个帧号和哪个时隙号处理数据。



图3-2上行链路CPRI板卡程序设计框图

### 3.1.1 BBU处理定时信息：解帧模块的设计

本设计中接收的是16比特的IQ（I：16bit，Q：16bit），他的帧结构与组帧的帧结构类似,一个基本帧中要放置30个载波数据，占用30个AXC，唯一不同的是AXC填充的数据不一样。解帧结构如下图3-3所示，IQ指示信息包括信道号和指示带宽。每一路数据传输的开始，除了第一个容器填充的是它的指示信号，其他的容器都是IQ数据，一路的数据传输完后，下一个CPRI基本帧传输另一路数据。



图 3-3 解帧映射规则

解帧是组帧的逆过程，它从接收到的 CPRI 帧中分离出 IQ 数据以及控制管理信息，然后传送给接收缓冲器做相应的处理。与组帧模块一样，它也要根据CTRL提供的word\_cnt和basic\_frame\_number对CPRI帧的数据进行分离。一个时隙内，不同指示带宽的数据所占用的基本帧数和容器数是不一样的，对应关系如下表3-1所示。

表3-1 不同指示带宽的数据所占用的基本帧数和容器数

|  |  |  |  |
| --- | --- | --- | --- |
| 指示带宽 | 容器数 | 基本帧数 | 说明 |
| 1/64khz | 97 | 4 | 1.5ms传输96组IQ数据，需要占用97个容器，第一个容器负载指示信息，其他的96个容器负载IQ数据，97/30=3.2，需要4个基本帧。 |
| 1/(64khz\*2) | 193 | 7 | 1.5ms传输192组IQ数据，需要占用193个容器，第一个容器负载指示信息，其他的192个容器负载IQ数据，193/30=6.4，需要7个基本帧。 |
| 1/(64khz\*5) | 481 | 16 | 1.5ms传输96组480数据，需要占用481个容器，第一个容器负载指示信息，其他的480个容器负载IQ数据，481/30=16.03，需要17个基本帧。 |
| 1/(64khz\*10) | 961 | 16 | 1.5ms传输960组IQ数据，需要占用961个容器，第一个容器负载指示信息，其他的960个容器负载IQ数据，961/30=32.03，需要33个基本帧。 |
| 1/(64khz\*15) | 1441 | 16 | 1.5ms传输1440组IQ数据，需要占用1441个容器，第一个容器负载指示信息，其他的1440个容器负载IQ数据，1441/30=48.03，需要49个基本帧。 |

我们上述选择的解帧方式是：一个时隙内的每路数据占用整数个基本帧传输，这样就会导致一些容器是没有传输数据的，还有一种方式就是每个容器都有数据传输，例如第一路数据传输占用了0—2号基本帧的全部容器，3号基本帧的AXC0—AXC6容器，那么3号基本帧的AXC7号容器就开始负载第二路数据，之后的容器和基本帧都用于负载第二路数据，直到一个时隙的第二路数据传完，再传输第三路。这种方式利用率高，但是解帧比较麻烦。

### 3.1.2 BBU处理定时信息：上行缓冲器的设计

CPRI板卡采用握手协议的方式定义4个BBU的优先级，如下图3-4所示。先处理BBU1的数据，最后处理BBU4的数据。CPRI板卡通过CPRI链路1先发送一个请求信号给BBU1，BBU1通过CPRI链路1 发送数据到CPRI板卡，当CPRI板卡处理完来自BBU1的数据，CPRI板卡通过CPRI链路2发送一个请求信号到BBU2，BBU2开始通过CPRI链路2发送数据。同理，当CPRI板卡处理完BBU2的数据后，通过CPRI链路3发送请求信号到BBU3；当CPRI板卡处理完BBU3的数据后，通过CPRI链路4发送请求信号到BBU4；当CPRI板卡处理完BBU4的数据后，再通过CPRI链路1发送请求信号到BBU1。



图 3-4 BBU遍历的实现模式

CPRI基本帧IQ模块以64bit为一拍作为缓冲器的缓存数据，缓存信息包括 IQ数据，IQ编号和指示带宽。FIFO进行跨时钟域的处理，写时钟采用CPRI的系统时钟，读时钟采用SRIO的系统时钟，FIFO的读数据形成SRIO帧通过RapidIO接口交互到LISU板卡。FIFO的宽度设置为64，深度设置为16。

图 3-5上行缓冲器的实现方式

### 3.1.3 BBU定时信息的获取

CPRI协议定义的同步定时信息有超帧号和10ms无线帧号，一个超帧的时间是1/15ms，一个10ms无线帧的时间是10ms。定时信息帧号是10ms无线帧号的整数倍，所以可以通过10ms无线帧获取帧号的定时信息。而定时信息时隙号根据CPRI协议现有的同步定时信息无法得到，为了实现定时信息时隙的获取，提出了一个新的实现方式，步骤如下：

CPRI在1.5ms可以传输5760个基本帧，一个超帧由256基本帧组成，确定5760与256的最小公倍数，经计算最小公倍数是115200，也就是每两个时隙对应45个超帧，也就是根据CPRI协议定义的同步定时信息可以获得3ms的定时信息，而一个时隙对应22.5个基本帧，1.5ms通过上一个超帧+34us或者下一个超帧-34us获得（34us约等于半个超帧的时间）。BBU定时信息的获取如下图3-6所示。



图3-6 BBU定时信息获取实现

### 3.1.4 CPRI处理定时信息：解帧模块的设计

第二种方式的解帧模块与第一种方式的解帧模块是一样的，它的指示信号占用两个容器，帧号独自占用一个容器，其他指示信号包括时隙号，信道号，指示带宽占用一个容器。帧结构如下图3-7所示。



图 3-7 解帧结构

### 3.1.5 CPRI处理定时信息：上行缓冲器的设计

CPRI处理定时信息，同样通过握手协议遍历四块BBU。CPRI基本帧IQ模块以64bit为一拍作为缓冲器的缓存数据，缓存信息包括 IQ数据，IQ编号和指示带宽。缓冲器写时钟采用CPRI的系统时钟，读时钟采用SRIO的系统时钟。因为CPRI板卡处理定时信息，所以解帧后的数据需要经过定时处理发送到RapidIO接口，缓冲器使用外部缓存，实现方式如下图3-8所示。缓冲器以CPRI的系统时钟写入数据，4个时钟周期的IQ填充一个地址单元，缓存数据1.5ms后，CPRI板卡控制BBU板卡停止数据传输，缓冲器以SRIO的系统时钟读出数据，再从缓冲器读取IQ数据，最后通过通过RapidIO接口交互到LISU板卡。

外部缓冲器划分为两个地址区域，一个区域用于传输到LISU1板卡，另一个区域用于传送到LISU2板卡，每个区域分配43260个地址空间，动态映射的方式与下行链路动态映射的方式一样。



图 3-8上行缓冲器的实现方式

### 3.1.6 SRIO成帧器设计

上行链路的IQ数据是以1.5ms为时间单位，也就是每路1.5ms的数据传输完后，另一路数据被传输，不同的指示带宽，一个时隙的数据容量大小不一样：

* 指示带宽64khz，一个1.5ms时隙内会传输96组IQ数据
* 指示带宽64khz\*2，一个1.5ms时隙内会传输192组IQ数据
* 指示带宽64khz\*5，一个1.5ms时隙内会传输480组IQ数据
* 指示带宽64khz\*10，一个1.5ms时隙内会传输960组IQ数据
* 指示带宽64khz\*15，一个1.5ms时隙内会传输1440组IQ数据

已知一个SRIO包最多携带256个字节的数据负载，本设计定义8个字节传输一组数据，所以一个SRIO包可传输32组IQ数据。以指示带宽是64khz为例，一个1.5ms时隙内会传输96组IQ数据，本设计定义的SRIO帧结构如图3-9。



图3-9定义的SRIO帧结构