**802.11a发送端模块设计**

**吴浩洋**

**2013年2月2日**

目录

[1 总述 1](#_Toc349640205)

[2 文件与格式 2](#_Toc349640206)

[2.1 文件读取 2](#_Toc349640207)

[2.2 格式转换 2](#_Toc349640208)

[3 编码 2](#_Toc349640209)

[3.1 扰码（scrambler） 2](#_Toc349640210)

[3.2 卷积编码（BCC） 2](#_Toc349640211)

[3.3 交织（Interleaver） 3](#_Toc349640212)

[3.4 星座映射（QAM） 3](#_Toc349640213)

[4 IFFT 模块 3](#_Toc349640214)

[4.1 IFFT\_mapper 3](#_Toc349640215)

[4.2 IFFT 3](#_Toc349640216)

[5 循环前缀与训练字 3](#_Toc349640217)

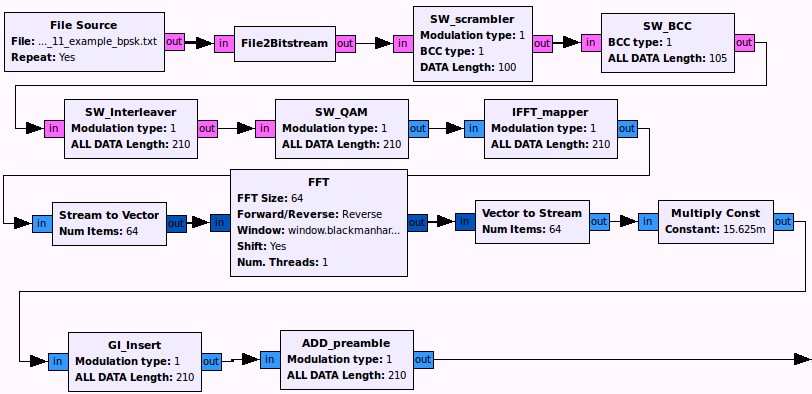
[5.1 循环前缀 3](#_Toc349640218)

[5.2 训练字 3](#_Toc349640219)

# 总述

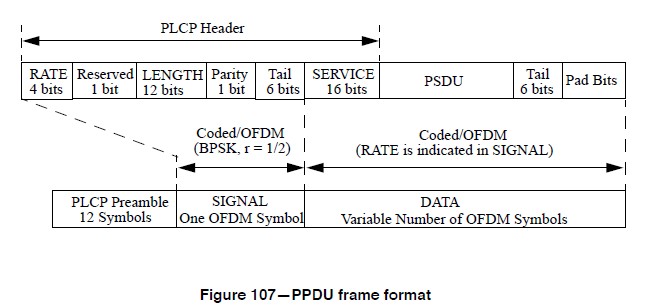
总的来说，发送端的设计是要比接收端的设计简单的多，因为发送端的设计省去了同步、均衡与解码等繁琐的步骤，只要遵循协议的步骤完成所有编码的过程，即可完成发送端的设计。

发送端模块的流图如下图所示：



File Source中预存一帧未处理的数据，通过一个文件转换器将ASIC码的txt文件转换为二进制数据流，并以8bit的位宽向后输送。该文件中包含了未处理的SIGNAL部分与DATA部分。之后数据依次经过扰码器（scrambler）、卷积编码器（BCC）、交织器（Interleaver），之后经过星座映射（QAM）模块将数据流变为复数。之后将频域数据经过IFFT转变为时域上的数据，最后加上循环前缀（GI\_Insert）和长短训练序列（ADD\_preamble）即完成了完整的OFDM的一帧。

帧格式如下图所示：



整个发送端的过程可以看为将上图的上排数据转变为下排数据。

在PLCP Header中，存在采样率、数据长度等有用信息，协议的要求是要利用这些信息，动态的调整模块的参数，例如调制方式（Modulation type）和卷积编码的编码率(BCC rate)，但在目前的设计中，仍然是用手动的方式调节这几个参数的。在使用整个系统的过程中，需要手动调整多个参数，且多个参数有一些换算关系。参数的换算关系详见文档《GNU Radio软件参数调整规则》；模块的使用方式、参数的调整与一些注意事项详见文档《GNU Radio软件测试细节》，在此不展开叙述。

另外，在PLCP Header中，前24 bit数据的编码方式始终是固定的，这就需要和后面的部分区分开来。所以在程序中设置了一个状态，通过状态的跳转来区分一帧中前后不同的调制方式与编码率。

# 文件与格式

## 文件读取

初始文件以字符形式存取，字符‘1’代表二进制的1，字符‘0’代表二进制的0，每一个byte数据放为一组，在同一byte里面，按左边高位、右边低位的顺序排列。文件在读取的时候，空格和回车略去不读。‘#’作为行注释。

## 格式转换

在软件的设计中，为了调测的方便，编写了一些格式转换的工具，如Bitsteam2StreamBIN（以二进制，同一byte由低到高的顺序显示数据）、Bitsteam2StreamHEX（以十六进制，同一byte由低到高的顺序显示数据）、Bitsteam2UserBIN（以二进制，同一byte由高到低的顺序显示数据）、Bitstream2UserHEX（以十六进制，同一byte由高到低的顺序显示数据）、Complex2String（显示复数数据）。

# 编码

## 扰码（scrambler）

数字通信系统中接收端的码元同步（位同步）信号通常是从接收到的数字信号的0和1的交变时刻中提取的，如果数字信号序列中经常出现长游程（0游程或1游程），则将会长时间不出现0和1码元的交变点，从而影响码元同步的建立和保持。因此希望传送的数字信号序列中不出现长游程；另外也不希望数字信号序列中存在周期性分量，因为这些周期分量的不同频率的谐波会由于电路中的非线性而产生交叉干扰。而扰码就是用于解决以上问题的方法之一。

扰码模块的流程是用程序生成一串伪随机序列，用该随机序列与原数据流进行异或操作即可。但要注意其中的Pad bits部分不进行异或操作。

伪随机序列采样m序列的方式生成，它有一串随机种子，该种子应该是不固定的，但在写程序时将该随机种子写成固定的了，这是有待修改的地方。

## 卷积编码（BCC）

采用卷积编码的优点是在编码率较高的情况下，保证解码时能够有效的纠正传输过程引入的偶然差错。因为卷积编码在整个序列中引入的前后的相关性，提高了一定的纠错能力。

卷积编码器共有1/2、2/3、3/4三种编码率，通过外部参数调整。编码过程包括循环移位、异或以及压缩等过程。

## 交织（Interleaver）

BCC的作用是纠正很长序列中的一个突发差错，但是有时候传输过程中会出现连续的突发差错，交织技术就是用于纠正这种突发差错的编码措施。它通过信号设计将原来属于突发差错的有记忆信道改造成为了独立差错的无记忆信道。也就是说，当序列中出现连续的误码时，交织可以将这些误码乱序排列，使得错误变成了间断的、不连续的，也就将连续误码改成了单次误码，这样通过接收端的Vitebi译码就可以纠正这种差错。

在设计程序时，按照协议的规定，对原序列进行重新排列即可。

## 星座映射（QAM）

星座映射模块共有四种信号的调制模式，它们分别是BPSK、QPSK、16QAM、64QAM。该模块将之前的二进制信号转变为了复数信号。从性能角度来说，BPSK的抗噪声性能最好，但是速率最慢；64QAM的抗噪声性能最差，但是速率最快。

该模块将二进制的信号映射到了二维的实数空间，实数空间的各点之间的汉明距离代表了其抗噪声的能力。设计程序时，也只需按照协议的规定，将对应二进制数据映射到实数空间即可。

# IFFT 模块

## IFFT\_mapper

在IFFT计算之前，需要将原来的数据中插入对应的导频数据，因此需要在IFFT之前加入IFFT\_mapper的模块，用于插入导频与调整导频次序。

注意在IFFT\_mapper模块中，-21、-7、7、21四个位置的导频的符号是会变化的，需要通过程序去动态的调整其符号。

## IFFT

IFFT直接采用了GNU Radio软件自带的模块。将频域数据变换到了时域上。

# 循环前缀与训练字

## 循环前缀

为了有效对抗多径信道的时延扩展与子载波间的干扰问题，可以采用循环前缀的方法。循环前缀就是将每个OFDM符号的信号波形的最后时间内的波形复制到前面原本是空闲保护间隔的位置上。对于IFFT实现来说，就是将最后若干个样值复制到前面，形成前缀。

## 训练字

训练字分为短训练字和长训练字，它们用于接收端的同步与信道估计，有着非常重要的作用。

训练字部分的频域上数据都是固定不变的，所以时域上的数据也是固定的，所以最简单的方式就是将320个训练序列的采样值预存在程序中，直接在整个数据的头部加上这些采样值即可。