

摘 要

本项目完成的是基于中国“数字电视地面广播传输系统帧结构、信道编码和调制”国家标准的发射端系统 FPGA 设计与实现。在本设计中，系统采用了 Stratix 系列的 EP1S80F1020C5 FPGA 为基础构建的主硬件处理平台。对于发射端系统，数据处理部分的扰码器（随机化）、前向纠错编码（FEC）、符号星座映射、符号交织、系统信息复用、频域交织、帧体数据处理（OFDM 调制）、同步 PN 头插入、以及信号成形 4 倍插值滚降滤波器（SRRC）等各模块都是基于 FPGA 硬件设计实现的。其中关键技术：TDS-OFDM 技术及其和绝对时间同步的复帧结构、信号帧的头和帧体保护技术、低密度校验纠错码（LDPC）等，体现了国标的自主创新特点，为数字电视领域首次采用。其硬件实现，亦尚未有具体产品参考。

本文首先介绍了当今国内外数字电视的发展现状，中国数字电视地面广播传输国家标准的颁布背景。并对国标系统技术原理框架，发端系统的整体结构以及 FPGA 设计的相关知识进行了简要介绍。在此基础上，第三章重点、详细地介绍了基于 FPGA 实现的发射端系统各主要功能模块的具体结构设计，论述了系统中各功能模块的 FPGA 设计和实现，包括设计方案、算法和结构的选取、FPGA 实现、仿真分析等。第四章介绍了对整个系统的级连调试过程中，对系统结构进行的优化调整，并对级连后的整个系统的性能进行了仿真、分析和验证。作者在项目中完成的工作主要有：

1. 阅读相关资料，了解并分析国标系统的技术结构和原理，分解其功能模块。
2. 制定了基于国标的发端系统 FPGA 实现的框架及各模块的接口定义。
3. 调整和改进了 3780 点 IFFT OFDM 调制模块及滚降滤波器模块的 FPGA 设计并验证。
4. 完成了扰码器、前向纠错编码、符号星座映射、符号交织、系统信息复用、频域交织、帧体数据处理、同步 PN 头插入、以及信号成形 4 倍插值滚降滤波器等功能模块的 FPGA 设计和验证。
5. 在系统级连调试中，利用各模块数据结构特点，优化系统模块结构。
6. 完成了整个发射端系统 FPGA 部分的调试、分析和验证。

关键词：数字地面电视, FPGA, 前向纠错编码, IFFT

Abstract

The FPGA-based hardware implementation of the transmitter of GB20600-2006 “Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System” has been completed in this project. The main hardware-platform is based on the Altra’s Statix series FPGA chip EP1S80F1020C5. The realization of the energy dispersal, FEC-coding, constellation mapping, symbol interleaving, 3780-point IFFT OFDM, synchro-PN insertion, SRRC-FIR filter, etc have been implemented based on FPGA.

The developing status of digital television, which is followed by the structure introduction of the transmitter of the GB20600-2006, is introduced firstly in this dissertation. And then the related knowledge about FPGA design is presented. The FPGA design of the modules of the transmitter is been importantly expatiated in chapter 3. In chapter 4, the amelioration of the system structure is been expatiated firstly. The simulation, analysis and validation of the whole transmitter of the GB20600-2006 is also been importantly expatiated in chapter 4. What I have mainly done in the project is listed as follows:

1. Read papers associated with the project and get familiar with the basic principles of the GB20600-2006.
2. Establish the FPGA design structure of the whole system and finish the interface definition of every sub-module.
3. Ameliorate the FPGA designs of the 3780-point IFFT OFDM module and the SRRC-FIR filter module on FPGA.
4. Complete the FPGA designs and verification the modules of the transmitter system of the GB20600-2006.
5. Ameliorate the system structure of the transmitter.
6. Finish the function simulation, analysis and validation of the whole system of the transmitter base on GB20600-2006.

Key word: Digital Television Terrestrial Broadcasting System, FPGA, FEC, IFFT

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。
· 与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

签名： 陈涛 日期： 2008年5月20日

关于论文使用授权的说明

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后应遵守此规定)

签名： 陈涛 导师签名： 朱继平

日期： 2008 年 5 月 20 日

第一章 引言

在经历了机械电视时代、黑白电子电视和彩色电视时代以后，电视向着数字化和高清晰度向前迈进，模拟电视在全世界范围内正在逐步引退。数字电视相对模拟电视在节目制作、节目存储、信号处理、传输以及播出都发生了革命性变化。其采用数字技术将活动图像和声音等信号加以处理、压缩、编码、经存储或实时广播后，供用户接收。与传统的模拟电视相比，数字电视在图像和声音质量两方面都有重大改进。其根据清晰度可分为：标准清晰度数字电视（SDTV：Standard Definition Television）和高清晰度数字电视（HDTV：High Definition Television），码率分别约为4.4Mbps和20Mbps左右。

经过多年坚持不懈的研究和发展，数字电视地面广播(DTTB: Digital Television Terrestrial Broadcasting)取得了很多的成果，在中国地面数字电视标准颁布之前，世界上已经提出了三个地面数字电视标准：欧洲的DVB-T、美国的ATSC和日本的ISDB-T，且都达到实用阶段。经过数年的研究和探讨，2006年8月30日，国家标准化管理委员会发布公告，中国数字电视地面广播传输系统标准—GB20600-2006《数字电视地面广播传输系统帧结构、信道编码和调制》^[1]（下称国标），于2006年8月18日正式批准成为强制性国家标准，2007年8月1日起实施。

1.1 数字电视的优势和特点

数字电视的优势和特点主要表现在5个方面：

- (1) 视频清晰度高、音频效果好。由于数字电视全过程采用数字信号，不受节目编辑、传输、转播和接收的影响。SDTV数字电视节目可以达到DVD质量，在观看HDTV节目时清晰度是目前电视的4倍以上，如35mm电影般清晰。
- (2) 频带利用率高。原PAL频道可播放4到8套标清数字电视。
- (3) 抗干扰性能好。解决了模拟电视中的闪烁、重影、亮色互串等问题；可以实现城市楼群的高质量接收，移动载体中也可接收到清晰的数字电视节目。
- (4) 便于开展各种综合业务和交互业务（包括因特网业务），有利于构建“三网合一”的信息基础设施。

(5) 节目的加密处理等应用。

1.2 地面数字电视广播的基本需求

对于数字电视地面广播来讲，首先要求数字电视有足够的接收性能，在室内采用简单、小型和低增益天线实现稳定接收。甚至在较强静态和动态多径的环境中，系统仍能够稳定工作。其次，考虑到数字电视日后发展的广阔空间和业务应用的多样性及传输容量需求的不断增长，要有足够高的传输码率，以便在单个6~8MHz 信道中提供高质量高清晰节目（大约 20Mbps）。还要有利于频率规划，使用现有分配的电视频道来传输 DTV 节目，实现和模拟电视节目的同播；当没有额外的频道分配时可使用禁用（Taboo）频道（由于干扰过大，不能用于模拟电视的频道），并具有和现有模拟电视台相当的覆盖范围。

其它的要求包括：需要先进的信道编码和信道估计方案，以便降低系统 C/N 门限，以此降低发射功率，并减少了对现有模拟电视节目的干扰，抵抗各种干扰/失真。高度灵活的操作模式，通过选择不同的调制方案，系统能够支持固定、便携、步行、手持或移动接收。易于和其它媒介或服务器的接口，支持多节目/业务，能够通过分级调制得到分级服务，具有交互性。高度灵活的频率规划和覆盖区域，能够使用单频网和同频道覆盖扩展/缝隙填充。而且系统应允许多种成本价格的接收机实现，包括低成本实现等等。

1.3 国外地面数字电视传输的现状

经过 10 年的深入研究，欧洲、美国和日本相继制定出地面数字电视传输标准 DVB-T、ATSC 和 ISDB-T。3 种传输标准使用相类似的信源编码方案和不同的信道编码调制方案，其传输参数比较如表 1-1 所示^[2]。

表 1-1 美国、欧洲、日本地面广播传输参数比较

	美国的 ATSC 标准	欧洲的 DVB-T 标准	日本的 ISDB-T 标准
传输方案	8VSB 传输方案	OFDM 传输方案	分段 OFDM 传输
载波数	单载波	2K、8K	2K、4K、8K
视频压缩	MPEG-2		
音频压缩	Dolby AC-3 音频编码	MPEG-2 层 II MUSICAM 音频编码	MPEG-2 层 III AAC 音频编码
复用方式	MPEG-2		
信道外码	RS 码(207,187,T=10)	RS 码(204,188,T=8)	RS 码(204,188,T=8)
信道内码	网格编码(TCM)(码率 2/3)	卷积编码(码率: 1/2, 2/3, 3/4, 5/6, 7/8)	
调制技术	8VSB	QPSK/16QAM/64QAM	
总码率	19~28 Mbps	4.98~31.67Mbps	3.68~23.42 Mbps
接收门限	C/N≥15dB	C/N≥19dB	

可以看到, ISDB-T 和 DVB-T 有很多相似的地方, 可以说, 日本使用的是修改的欧洲方案。目前, DVB 成员已经达到 265 个(来自 35 个国家和地区)。ATSC 成员 30 个, 其中美国成员 20 个, 来自阿根廷、法国、韩国等 7 个国家的成员 10 个。ISDB-T 筹划指导委员会委员 17 个, 其他成员 23 个, 都是日本的电子公司和广播机构。由此可以看出, DVB 标准发展最快, 普及范围最广。

从表中我们可以看到三套系统的一些区别和联系。ATSC、DVB-T、ISDB-T 标准的频道宽度均为 6~8MHz, 每个频道都可以设置一个固定接收 HDTV(高清晰度电视)或 4~5 个 SDTV(标准清晰度电视), 数字电视频道的宽度和原有模拟电视频道相同, 处在转换期的同播状况, 均采用“放弃信源兼容, 坚持信道兼容的政策”。ATSC 在 6MHz 内可稳定地传输一套 HDTV 电视。DVB-T 由于保护间隔的插入, 从而减少了频谱的利用率, 在 6MHz 内较难稳定地接收一套 HDTV 电视, 必须加大带宽才能稳定地接收。目前 DVB-T 首先采用的是 SDTV 电视。

ATSC 采用网格编码调制, 理论和实测都表明, 接收门限 (C/N) 达 15 dB, 提高了电视发射机覆盖范围。DVB-T 和 ISDB-T 接收门限 (C/N) 大约在 19 dB 左右。从接收场强看, ATSC 要优于 DVB-T 和 ISDB-T 3~4 dB 左右。模拟接收的门限一般为 40 dB 左右, 可见, 数字接收大大提高了广播的覆盖范围。

ATSC 不支持移动接收，而 DVB-T 和 ISDB-T 都支持移动接收。为增强移动接收能力，DVB 的工作小组从 2002 年前后开始研究 DVB-H，2004 年 1 月确定了该规格的基本框架。DVB-H 是 DVB-T 的扩展，它的主要应用是在手机、掌上电脑等手持移动终端上看电视。相对于 DVB-T、DVB-H 标准主要增加了以下内容：

(1)、引入时间切片 (Time Slicing) 技术，发送数据仍然是连续的，接收机可以选择接收部分 Slicing 而省电。

(2)、蜂窝间握手准备。通过对相邻蜂窝监控，将传输流的切换时间放在接收机的关闭时间，蜂窝转换的握手可以保证服务不中断。

(3)、在 DVB-T 的 2k 和 8k 传输模式之外加入 4k 传输模式，增加组网灵活性。

1.4 国内地面数字电视传输的发展概况

2001 年我国正式在全国范围内广泛征集地面数字电视传输标准，收到了清华大学、广播科学研究院、HDTV 功能样机系统研究开发项目总体组和电子科技大学 4 家单位提交的 5 套技术方案。其中，清华的 DMB-T(地面数字多媒体电视广播传输协议)方案采用时域同步正交频分复用 (TDS-OFDM) 技术，在信道编码方面采用了 TURBO 纠错编码，在时域中插入 PN 序列作为同步信号。DMB-T 由于使用了时域信道估算技术，所以它的同步时间比 DVB-T 快很多，DVB-T 的同步时间为 100 毫秒左右，而清华的 DMB-T 可以做到 5 毫秒以内。DMB-T 是一种多载波方案，共有 3780 个子载波。上海交大的 ADTB-T 标准(高级数字电视广播系统)是一种“单载波”方案，采用 TPC 信道编码技术^[3]。最终，在此基础上打破了单、多载波两种体制的原有方案的组成模块进行融合优化，形成了满足广电应用要求的数字电视地面广播传输融合方案。

在经历多次的实验室测试、场地测试后，相关部门于 2006 年 8 月 18 日颁布了 GB20600- 2006《数字电视地面广播传输系统帧结构、信道编码和调制》国家强制标准。国标的主要关键技术有：TDS- OFDM 技术及其和绝对时间同步的复帧结构、信号帧的头和帧体保护技术、低密度校验纠错码 (LDPC) 等，它们体现自主创新特点，并能提高系统性能。国标支持 4.813~32.486 Mbit/s 的系统净荷传输数据率，支持标准清晰度电视业务和高清晰度电视业务，支持固定接收和移动

接收，支持多频组网和单频组网^[4]。这一国家标准的颁布标志着中国地面数字电视广播的正式启动。

1.5 项目背景与意义

随着中国数字电视地面广播传输系统标准--GB20600-2006《数字电视地面广播传输系统帧结构、信道编码和调制》的正式发布，广电总局规划将在2015年停止模拟信号的传输，全面实现广播电视数字化。在数字电视产业化的必然趋势中，模拟时代的终结和数字时代的兴起，蕴涵着亿万级别的市场前景。

地面广播电视接受系统作为数字电视体系中不可或缺的组成部分，其市场前景伴随广播电视数字时代的到来，炙手可热。本次发布采用国标的主要关键技术有：TDS-OFDM技术及其和绝对时间同步的复帧结构、信号帧的头和帧体保护技术、低密度校验纠错码(LDPC)等，体现了自主创新特点。这些创新技术的采用意味着，在该市场领域，各竞争者在技术上处于同一起跑线。截至本文完成时，仅有国内的凌讯、上海高清两家公司可以提供基于国标的数字电视接收芯片。

在整个电视广播系统中，发射端作为重要组成部分，在实际应用中，其市场需求量相对较少。针对ASIC研发成本高、周期长、功能固化的缺陷，采用FPGA方式实现更适合发射端的市场和技术特点。同时随着整个数字电视产业的发展，FPGA方式有利于对系统进一步升级、优化。

发射端信道编码和调制部分设计，是整个发射端系统的核心。本设计的完成，同时将为接收端芯片的信道解调、解码部分的设计实现，奠定坚实的理论、技术基础，也为接收端设计的实际测试提供了必须的保障。本项目针对国标，实现了基于FPGA传输系统的帧结构，信道编码和调制功能，具有广阔的市场前景，对基于国标的发射、接受端的深入产品研发方面都具有重要的实际参考价值。

1.6 本文的结构和侧重点

本文主要分为两部分：

第一部分，即在本文的第二章节中，首先简要介绍了中国数字电视地面广播传输系统标准--GB20600-2006《数字电视地面广播传输系统帧结构、信道编码和调制》，包括了数字电视地面广播传输发端系统的整体结构、基本原理和实现的技

术指标。之后介绍了本设计中，发端系统的总体实现方案及其实现的硬件处理平台。

第二部分详细介绍了中国数字电视地面广播传输发端系统的 FPGA 设计和实现。在本文的第三章节中，重点论述了系统中各功能模块的 FPGA 设计和实现，包括设计方案、算法和结构的选取、FPGA 实现、仿真分析等。之后的第四章节，介绍了本设计在系统级的设计分析中，根据系统模块数据结构特点对系统进行了一定的结构优化，进一步减少硬件资源消耗成本。

本文以第二部分为重点。

第二章 数字电视地面广播发端系统概述及实现方案

2.1 中国数字电视地面广播传输标准概述^[1]

国家标准定义了数字电视地面广播传输系统发送端完成从输入数据码流到地面电视信道传输信号的转换。地面数字电视广播系统发送端完成从MPEG-TS传送码流到地面电视传输信道信号的转换。输入数据码流经过扰码器（随机化）、前向纠错编码（FEC），然后进行比特流到符号流的星座映射，再进行交织后形成基本数据块，基本数据块与系统信息组合（复用）后并经过帧体数据处理形成帧体，帧体与相应的帧头（PN序列）复接为信号帧（组帧），经过基带后处理转换为输出信号（8MHz带宽内）。该信号经变频转换为射频信号（UHF和VHF频段范围内）。其中体现自主创新具有与国外数字电视地面传输标准不同特点、并能提高系统性能的主要关键技术有：利用特殊设计的PN序列作为同步和信道估计的符号保护间隔填充方法、TDS-OFDM技术及其和绝对时间同步的复帧结构、低密度校验纠错码（LDPC）、系统信息的扩谱传输方法等。本系统的发送端原理如图2-1所示^[1]。

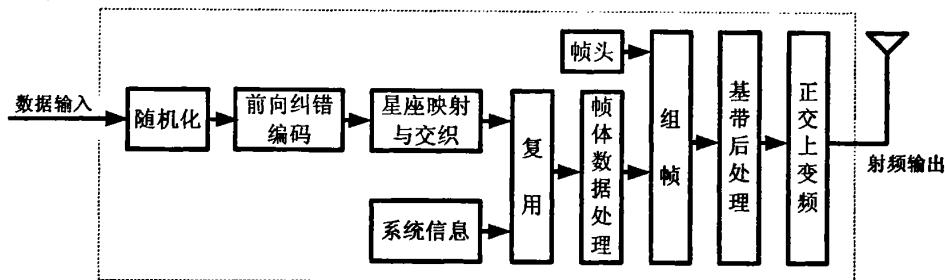


图 2-1 发送端原理框图

2.1.1 编码和调制

2.1.1.1 随机化

为了保证传输数据的随机性以便于传输信号处理，输入的数据码流数据需要用扰码进行加扰。扰码是一个最大长度的二进制伪随机序列。该最大长度二进制伪随机序列线性反馈移位寄存器（LFSR）生成。其生成多项式定义为：

$$G(x) = 1 + x^{14} + x^{15} \quad (2-1)$$

该 LFSR 的初始状态定义为 100101010000000。

输入的比特码流（字节码字从 MSB 到 LSB）与 PN 序列进行逐位模二加后产生数据扰乱码。扰码器的移位寄存器在信号帧开始时复位到初始状态。

2.1.1.2 前向纠错编码

前向纠错编码（FEC）技术是国标相对于国外现有的 3 个地面数字电视广播标准的主要区别之一，由外码（BCH 码）和内码（LDPC）级联实现，采用了三种不同的码率以满足各种服务需求，并且为了降低实现成本，三种不同码率采用的 LDPC 码具有相同的结构，达到了硬件实现的资源共享。

(1) 外码 外码采用的 BCH 码为 BCH(762,752)码，BCH(762,752)码是由 BCH(1023, 1013) 系统码缩短而成的。它在 752 bit 的数据码前加入 261bit 0 补足 1013bit，然后进行 BCH (1023, 1013)编码，BCH(1023, 1013)生成多项式为

$$G_{BCH}(x)=I+x^3+x^{10} \quad (2-2)$$

编码后得到 1023 bit 码字，然后删除前 261 bit 的 0，得到了码长为 762bit 的 BCH 码。根据线性分组码的原理，BCH (1023, 1013) 码可纠 1 bit 的突发错误。

(2) 内码 低密度校验码（LDPC）采用了基于矩阵分解中的两个信息符号的 RS 码法，构造 LDPC 码的循环置换矩阵，得到其生成矩阵 \mathbf{G}_{qc} 。在国家标准中，提供了 3 种不同码率的 LDPC 编码方式可供选择，它们分别是：

- 1) 等效编码码率为 R=0.4 的 LDPC(7493, 3084) 码；
- 2) 等效编码码率为 R=0.6 的 LDPC(7493, 4572) 码；
- 3) 等效编码码率为 R=0.8 的 LDPC(7493, 6016) 码。

国标中定义 LDPC 码信息位在后，校验位在前。

2.1.1.3 符号星座映射

国标包含以下几种符号映射关系：4QAM、4QAM+NR、16QAM、32QAM、64QAM。

4QAM 与 4QAM+NR 的符号映射对应于高速移动服务业务的需求，可以支持标准清晰度电视广播，能够兼顾覆盖范围和接收质量的服务需求。

4QAM 、16QAM 与 32QAM 符号映射可对应于中码率业务的需求，可以支持

多路标准清晰度电视广播，能够兼顾覆盖范围和频率资源利用的服务需求。

32QAM 和 64QAM 符号映射对应于高码率业务的需求，可以同时支持高清晰度电视和多路标准电视的广播，兼顾高档用户和普通用户的服务需求。

2.1.1.4 符号交织

国标系统采用了时域符号交织技术以提高抗脉冲噪声干扰能力。时域符号交织编码是在多个信号帧之间进行的。数据信号的数据块间交织采用基于星座符号的卷积交织编码。国标支持 2 种交织模式，交织深度分别为 240 符号和 720 符号，分别称为短交织和长交织。

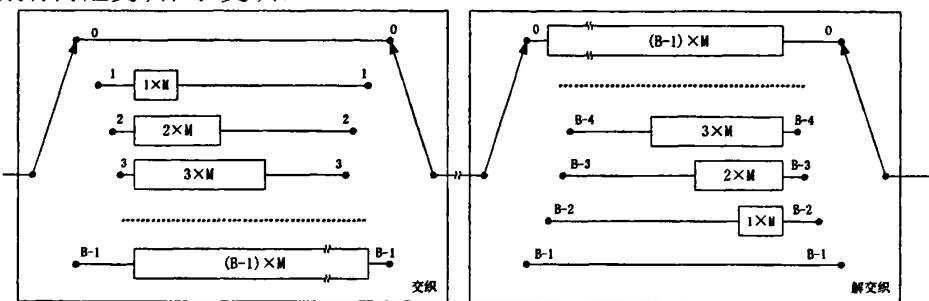


图 2-2 卷积式数据块间交织与解交织

模式 1： $B=52$, $M=240$ 符号，交织/解交织总延迟为 170 个信号帧；

模式 2： $B=52$, $M=720$ 符号，交织/解交织总延迟为 510 个信号帧。

2.1.1.5 频域交织

频域交织仅适用于载波数为 3780 模式，目的是将调制星座点符号映射到帧体包含的 3780 个有效载波上。频域交织为帧体内数据的符号块交织，交织大小等于载波数 3780。具体交织运算过程如下，数组 $X[3780]$ 的前 36 个元素为系统信息符号，后 3744 个元素为数据符号。为了使交织输出时 36 个系统信息符号集中放置，首先将这 36 系统信息符号按一定规律插入到 3744 个数据符号中，插入后得到的序列由数组 $Z[3780]$ 表示。将数组 $Z[3780]$ 通过以下程序进行位置调换得到最终交织输出序列 $Y[3780]$ ：

```
for(i=0; i<3; i=i+1); for(j=0; j<3; j=j+1); for(k=0; k<3; k=k+1); for(l=0; l<2; l=l+1);
for(m=0; m<2; m=m+1); for(n=0; n<5; n=n+1); for(o=0; o<7; o=o+1);
Y[o*540+n*108+m*54+l*27+k*9+j*3+i] = Z[i*1260+j*420+k*140+l*70+m*35+n*7+o];
```

2.1.2 复帧

2.1.2.1 复帧结构

数据帧结构如图 2-3 所示，是一种四层结构。其中，数据帧结构的基本单元为信号帧，信号帧由帧头和帧体两部分组成。超帧定义为一组信号帧。分帧定义为一组超帧。帧结构的顶层称为日帧（Calendar Day Frame, CDF）。信号结构是周期的，并与自然时间保持同步。

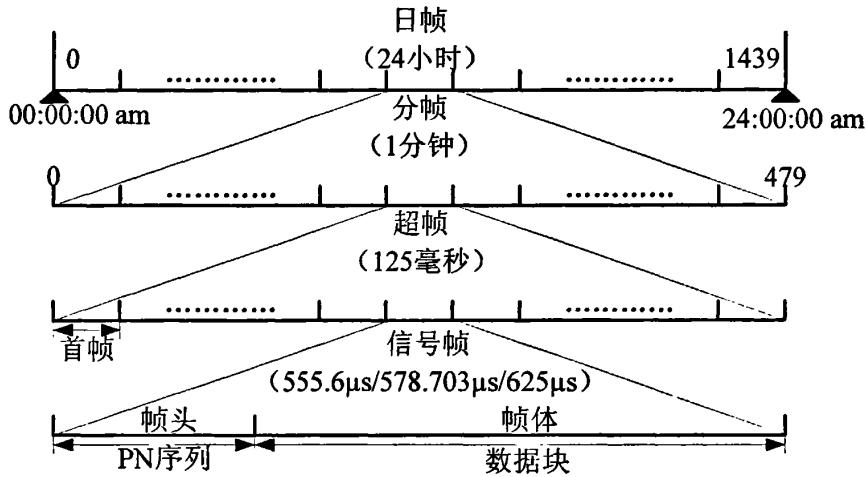


图 2-3 复帧的四层结构

2.1.2.2 复帧各层组成

(1) 信号帧 信号帧是系统帧结构的基本单元，一个信号帧由帧头和帧体两部分时域信号组成。帧头和帧体信号的基带符号率相同 (7.56Msps)。帧头部分由 PN 序列构成，帧头长度有三种选项。帧头信号采用 I 路和 Q 路相同的 4QAM 调制。帧体部分包含 36 个符号的系统信息和 3744 个符号的数据，共 3780 个符号。（详细结构说明见 2.1.3 节）

(2) 超帧 超帧的时间长度定义为 125 毫秒，8 个超帧为 1 秒，这样便于与定时系统（例如 GPS）校准时间。

(3) 分帧 一个分帧的时间长度为 1 分钟，包含 480 个超帧。

(4) 日帧 日帧以一个公历自然日为周期进行周期性重复，由 1440 个分帧构成，时间为 24 小时。在北京时间 00:00:00 AM 或其它选定的参考时间，日帧被复位，开始一个新的日帧。

2.1.3 信号帧

上一小节提到，信号帧是系统帧结构的基本单元，一个信号帧由 3780 点得信息帧体和起保护间隔作用得帧头两部分时域信号组成。如图 2-4 所示

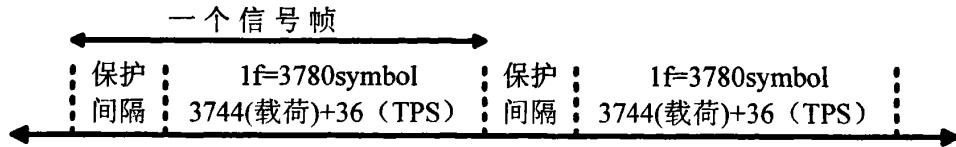


图 2-4 信号帧结构图

国标采用特殊设计的 PN 序列填充保护间隔，利用 PN 序列实现了快速稳健的同步和快速高效的信道估计。该 PN 序列也可以用作为时域均衡器的训练序列，充分发挥判决反馈的作用。

2.1.3.1 帧体构成

3780 个子载波中 3744 个子载波传输载荷数据，36 个子载波传输系统信息信号。系统信息为每个信号帧提供必要的解调和解码信息，包括符号映射方式、LDPC 编码的码率、交织模式信息、帧体信息模式等。在接收机端通过对 TPS 信号的解调可得到发射机信息，方便接收机端自动适应发端的工作模式，实现自动接收功能。

国标系统中预设了 64 种不同的系统信息模式，6 比特扩频前的系统信息将采用扩频技术成为 32 比特长的系统信息矢量，即用长度为 32 的 Walsh 序列和长度为 32 的随机序列来映射保护。对于 3780 个子载波，每个子载波占有的带宽均相同。在 8 MHz 带宽信道中每个子载波占有 2 kHz 带宽，3780 个子载波总共占有 7.56 MHz。

2.1.3.2 帧头构成

国标系统的保护间隔有 3 种模式：1/9 帧体长度(420 符号)、1/6 帧体长度(595 符号) 和 1/4 帧体长度(945 符号)，不同的保护间隔可以适应不同得信道环境，便于组建网络。

帧头模式 1 采用的 PN 序列定义为循环扩展的 8 阶 m 序列。可由一个 Fibonacci 型线性反馈移位寄存器实现，经“0”到+1 值及“1”到-1 值的映射变换为非归零的二进制符号。长度为 420 个符号的帧头信号 (PN420)，由一个前同步、一个 PN255 序列和一个后同步构成。前同步和后同步定义为 PN255 序列的循环扩展。线性反

馈移位寄存器的初始条件值确定所产生的 PN 序列的相位。在一个超帧中共有 225 个信号帧。每个超帧中各信号帧的帧头采用不同相位的 PN 信号作为信号帧识别符。帧头信号的平均功率是帧体信号的平均功率 2 倍。

帧头模式 2 采用 10 阶最大长度的伪随机二进制序列（ m 序列的截短），帧头信号的长度为 595 个符号，是长度为 1023 的 m 序列的前 595 个码片。在一个超帧中共有 216 个信号帧。每个超帧中各信号帧的帧头采用相同的 PN 序列。帧头信号的平均功率与帧体信号的平均功率相同。

帧头模式 3 采用的 PN 序列定义为循环扩展的 9 阶 m 序列。可由一个 Fibonacci 型线性反馈移位寄存器实现，经“0”到+1 值及“1”到-1 值的映射变换为非归零的二进制符号。长度为 945 个符号的帧头信号（PN945），其中 945 个符号由一个前同步、一个 PN511 序列和一个后同步构成。前同步和后同步定义为 PN511 序列的循环扩展。线性反馈移位寄存器的初始条件值确定所产生的 PN 序列的相位。在一个超帧中共有 200 个信号帧。每个超帧中各信号帧的帧头采用不同相位的 PN 信号作为信号帧识别符。帧头信号的平均功率是帧体信号的平均功率 2 倍。

2.1.4 帧体数据处理

映射后 3744 个数据符号复接系统信息后，并经帧体数据处理后形成帧体，用 C 个子载波调制，占用的 RF 带宽为 7.56MHz，时域信号块长度为 $500 \mu s$ 。

C 有两种模式：C=1 或 C=3780；

令 $X(k)$ 为对应帧体信息的符号；

当 C=1 时，生成的时域信号可表示为：

$$FBody(k) = X(k) \quad k = 0, 1, \dots, 3779 \quad (2-4)$$

在 C=1 模式下，作为可选项，对组帧后形成的基带数据在 ±0.5 符号速率位置插入双导频，两个导频的总功率相对数据的总功率为 -16dB。插入方式为从第 1 帧的第一个符号（编号为 0）开始，在奇数符号上实部加 1、虚部加 0，在偶数符号上实部加 -1、虚部加 0。

在 C=3780 模式下，相邻的两个子载波间隔为 2kHz，对帧体信息符号 $X(k)$ 进行频域交织，得到 $X(n)$ ，然后按下式进行变换得到时域信号：

$$FBody(k) = \frac{1}{\sqrt{C}} \sum_{n=1}^C X(n) e^{j2\pi n \frac{k}{C}} \quad k = 0, 1, \dots, 3779 \quad (2-5)$$

2.1.5 基带后处理及射频信号

基带后处理(成形滤波)采用平方根升余弦(Square Root Raised Cosine, SRRC)滤波器进行基带脉冲成形。SRRC 滤波器的滚降系数 α 为 0.05。

平方根升余弦滤波器频率响应表达式如下式所示：

$$H(f) = \begin{cases} 1 & |f| \leq (1 - \alpha)/2T_s \\ \left\{ \frac{1}{2} + \frac{1}{2} \cos \left(\frac{\pi[(2T_s|f|) - 1 + \alpha]}{2\alpha} \right) \right\}^{\frac{1}{2}} & (1 - \alpha)/2T_s < |f| \leq (1 + \alpha)/2T_s \\ 0 & |f| > (1 + \alpha)/2T_s \end{cases} \quad (2-6)$$

其中 T_s 为输入信号的符号周期 ($1/7.56 \mu s$)， α 为平方根升余弦滤波器滚降系数。

调制后的 RF 信号由下式描述：

$$S(t) = \operatorname{Re}\{\exp(j \cdot 2\pi F_c t) \times [h(t) \otimes Frame(t)]\} \quad (2-7)$$

其中， $S(t)$ — RF 信号；

F_c — 载波频率 (MHz)；

$h(t)$ — SRRC 滤波器的脉冲成形函数；

$Frame(t)$ — 组帧后的基带信号，由帧头和帧体组成。

2.1.6 基带信号频谱特性及系统净荷数据率

对应于射频发送信号的成形滤波后基带信号（未含双导频插入）典型频谱特性如图 2-5 所示。

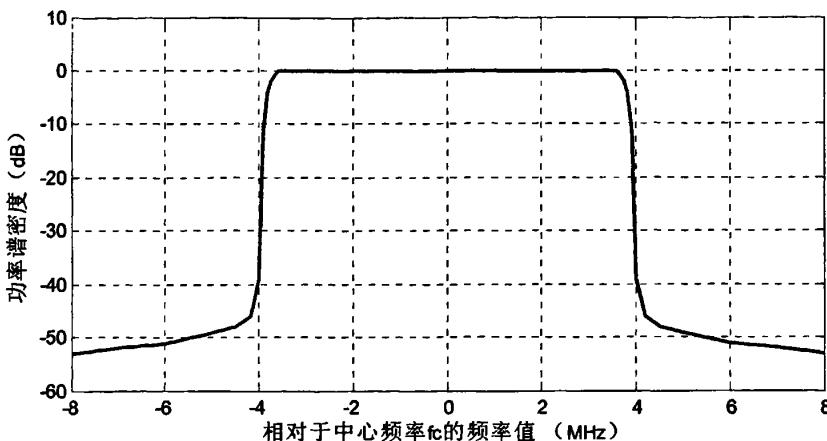


图 2-5 成形滤波后基带信号频谱特性

在不同信号帧长度、内码码率和调制方式下，标准支持的净荷数据率如表 2-1 所示。

表 2-1 系统净荷数据率(Mbps)

		信号帧长度 4200 个符号		
		0.4	0.6	0.8
映射	4QAM-NR			5.414
	4QAM	5.414	8.122	10.829
	16QAM	10.829	16.243	21.658
	32QAM			27.072
	64QAM	16.243	24.365	32.486

		信号帧长度 4375 个符号		
		0.4	0.6	0.8
映射	4QAM-NR			5.198
	4QAM	5.198	7.797	10.396
	16QAM	10.396	15.593	20.791
	32QAM			25.989
	64QAM	15.593	23.390	31.187

		信号帧长度 4725 个符号		
		0.4	0.6	0.8
映射	4QAM-NR			4.813
	4QAM	4.813	7.219	9.626
	16QAM	9.626	14.438	19.251
	32QAM			24.064
	64QAM	14.438	21.658	28.877

注：表中斜线表示该模式组合不在标准规范之内。

2.2 发端系统实现的总体方案与设计层次

2.2.1 总体方案

从图 2-1-1 发送端原理框图可以看出系统的数据传输流程。在本设计方案中，将图中所示各功能模块在基于 FPGA 的发射系统中实现。其中数字基带部分完全在 FPGA 芯片中完成，数字上变频部分和 D/A 转换由专用器件（AD9857）完成。这种处理方式使得系统结构简单灵活，便于调试和移植。完整发端系统的硬件结构简图如图 2-6 所示：

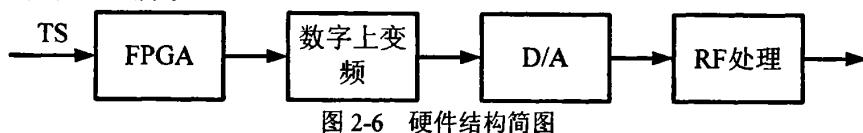


图 2-6 硬件结构简图

本文将详细介绍其中的 FPGA 部分的设计。根据 FPGA 设计的自顶向下的设计思路，在本设计中，作者根据数据的传输流程和功能模块实现的复杂程度，将整个 FPGA 部分分为六个大的功能模块，即能量扩散模块（随机化）、信道编码模块、帧体形成模块、帧体数据处理模块、信号成型模块、基带后处理模块。再根据每部分模块的功能进一步细化处理和设计。FPGA 部分总体方案如图 2-7 所示：

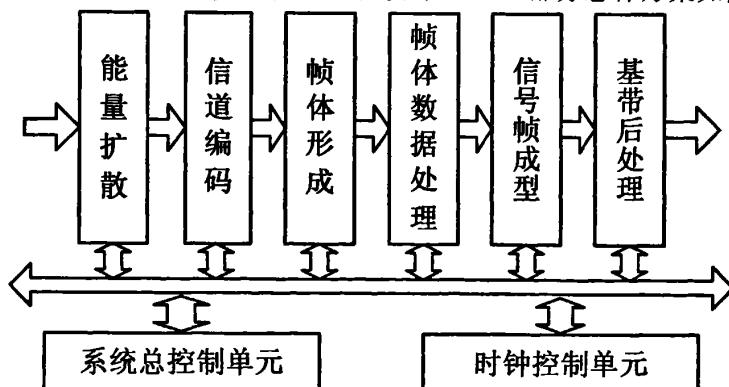


图 2-7 数字电视发端系统 FPGA 设计的总体方案

系统基准时钟源由 FPGA 外部晶振产生，，时钟管理模块以其为参考时钟，通过 FPGA 内部锁相环生成各个模块的工作时钟。总控制单元控制各模块的启动和数据传输。

2.2.2 发端 FPGA 设计流程及层次

根据上节的系统总体方案要求，本文在进行具体的 FPGA 设计时，首先对各个大的系统模块进行接口定义。然后根据各模块功能以及数据流的变换处理过程，对模块进行子模块的划分以及子模块的接口定义。最终，根据 FPGA 设计的自顶向下的设计思路，本设计将基于国标的系统发射端的 FPGA 模块层次划分如图 2-8 所示：

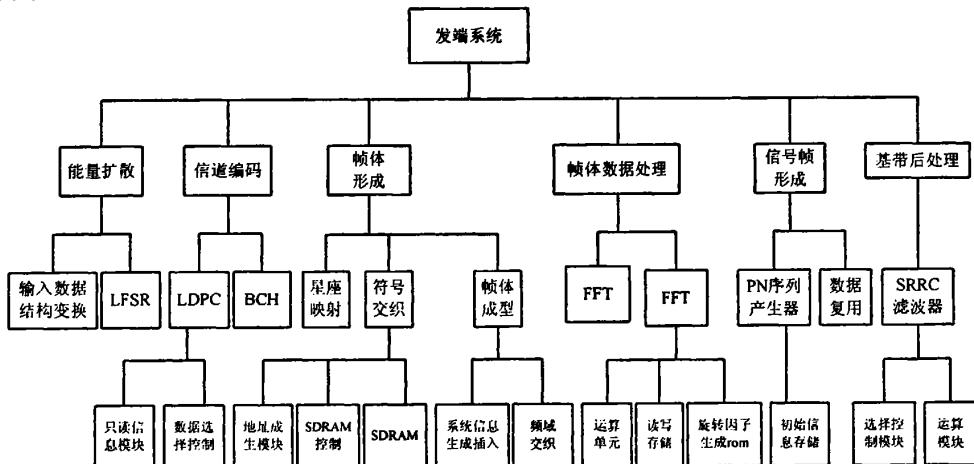


图 2-8 国标系统数字电视发端 FPGA 设计模块层次图

2.3 Stratix 系列 FPGA 及 QuartusII 介绍

2.3.1 Stratix 系列 FPGA 介绍

本项目选用的现场可编程器件是 Altera 公司的 EP1S80F1020C5。EP1S80 是为适应片上系统（SOC）的需求而推出的 Stratix 系列器件之一。该器件内部集成有大量的逻辑单元、大容量存储器、可编程锁相环和高性能的 DSP 模块，同时采用了全新的布线结构，最高工作时钟达到 420MHz。在本方案中，将数字基带处理这个完整的系统集成在一片 FPGA 里，避免了多芯片系统造成的硬件复杂、难于调试、不易移植等弱点。

Stratix 系列器件有如下特点^[5]：

- 1、采用 1.5V 内核电压，0.13-μm 全铜工艺；
- 2、高密度，含有大量的逻辑单元（LE）和存储器资源（RAM bits）；
- 3、内嵌的 DSP 模块，易于实现高性能的乘法器和滤波器；
- 4、支持多种 I/O 标准；
- 5、最多支持 12 个锁相环，最高时钟达 420 MHZ；

6、内部有 16 个专用的全局时钟网络和 16 个区域时钟网络。

Stratix 器件是一个基于行和列的二维的逻辑结构。一系列的行列连线存在于 LAB(逻辑阵列块)之间、存储模块之间和 DSP 模块之间，这些连线以不同的长度和速度为功能模块提供相互之间的信号连接。Stratix 器件的主要单元的功能特点如下：

1. 逻辑阵列块 (LAB)

LAB 由 10 个 LE(逻辑单元)、LE 间的进位链、级连链、LAB 控制信号和 LAB 局部互连构成。LE 是最小的、基本的逻辑单元，用于实现用户定制逻辑功能。每个 LE 包含 1 个 4 输入的查找表、1 个可编程寄存器、1 个进位链和 1 个级连链。

2. 存储器模块结构 (memory block structure)

Stratix 系列器件有三种类型的 RAM 块：M512、M4K 和 M-RAM。

M512 模块是简单的双口存储块，有 512 位外加奇偶校验位（共 576 位）。可以用作简单的专用双口或者单口存储器，最大可达到 18 位宽，318MHz。

M4K 模块是真正的双口 RAM 块，有 4096 位外加奇偶校验位（共 4608 位）。可以用作真正的双口，或者单口存储器，最大可达到 36 位宽，291MHz。

M-RAM 模块 (64K×9bit 到 4K×144bit)：512K 位外加奇偶校验位（共 589,824 位），可以用作真正的双口，或者单口存储器，最大可达到 144 位宽，269MHz。

3. 数字信号处理模块 (Digital Signal Processing Block)

每个数字信号处理模块可以实现 8 个满精度的 9bits×9bits 乘法器，4 个满精度的 18bits×18bits 乘法器，或者 1 个满精度的 36bits×36bits 乘法器，而且有加减法器。数字信号处理模块包含 18 位的输入移位寄存器，这一特点可用在 FIR 和 IIR 滤波等处理中。

4. 锁相环 (PLL) 和时钟管理

Stratix 系列器件提供了一个高级的时钟管理结构和多个 PLL 来满足用户在 FPGA 内部对时钟的需求。分别有全局时钟网络和区域时钟网络，而器件的内部 PLL 也分别有两类：增强型锁相环 (Enhanced PLL) 和快速锁相环 (Fast PLL)。

Stratix 系列器件有 16 个专用的全局时钟网络，所有的资源，包括 IOE、LE、DSP block、memory 都可以用此作为时钟源。而且全局时钟信号也可以作为控制信

号，如时钟使能、同步或者异步清零信号。

Stratix 系列器件有 16 个区域时钟网络。器件可分为 4 个象限，每个象限有 4 个区域时钟网络，它们只在所服务的象限内才有效。

另外，Stratix 系列器件有 16 个专用的时钟管脚，用以驱动全局时钟网络或者区域时钟网络。

5. IOE (I/O 单元)

IOE 由一个双向 I/O 缓冲器和六个寄存器组成。寄存器可用作输入、输出或输出使能信号。当用作专用时钟时，这些寄存器可以为外部存储器（如 DDR SDRAM、QDR SRAM 等）提供优异的接口性能。每个 I/O 单元都可配置成输入、输出或双向引脚。

EP1S80F1020C5 的资源配置如表 2-2，可以看到，该器件的逻辑资源和存储器资源非常丰富，能够把一个完整的信号处理系统放在一片 FPGA 里实现。

表 2-2 EP1S80F1020C5 的资源配置

逻辑单元	79,040	I/O 管脚	782
RAM 总量	7,427,520 bits	锁相环	12
DSP 块	16	全局时钟网络	16
嵌入式乘法器 (9bits × 9bits)	128	专用时钟管脚	16

2.3.2 Quartus II 工具介绍

QuartusII 容易使用，人机界面友好，是支持对 Altera 公司 Stratix 系列 FPGA 进行开发的软件。利用 QuartusII 所提供的设计环境和设计工具，可以灵活高效的完成各种数字电路的设计。

QuartusII 为 FPGA 的设计提供了一个非常完善的开发环境：

- (1) 支持原理图式图形输入、文本设计、内存编辑、模块化设计；
- (2) 支持 EDIT、HDL 和 VQM 等第三方设计工具；
- (3) 强大的逻辑综合，功能和时序仿真。

使用 QuartusII 开发工具进行设计，设计者无需精通器件内部的复杂结构，只

需要自己熟悉设计的输入工具，如 HDL 语言、原理图或者波形进行设计输入。QuartusII 将会自动地把这些设计输入转化为目标结构所要求的格式，由于有关结构的详细知识已经装入开发工具，设计者不需要手工优化自己的设计，因此能大大提高设计效率。

2.3.3 FPGA 设计方法及开发流程

2.3.3.1 FPGA 设计方法

传统的系统硬件设计采用自下而上 (bottom-up) 的设计方法，即系统硬件的设计从选择的具体器件开始，并用这些元器件进行逻辑设计，完成系统各独立功能模块设计，然后再将各功能模块连接起来，完成整个系统的硬件设计。上述过程从最底层开始，直到最高层设计完毕。

目前，我们普遍采用自上而下 (top-down) 的设计方法，就是从系统总体要求出发，自上而下的逐步将设计内容细化，最后完成系统硬件的整体设计。FPGA 设计自上而下可以分为三个层次：

第一层次：行为描述。实质上就是对整个系统数学模型的描述；

第二层次：RTL 方式描述，又称寄存器传输描述，为第三层次的逻辑综合做准备；

第三层次：逻辑综合。利用逻辑综合工具，将 RTL 方式描述的程序转换成用基本逻辑元件表示的文件。其相当于在人工设计硬件电路时，根据系统要求画出的系统逻辑原理图。

2.3.3.2 FPGA 开发流程

FPGA 的开发流程为：

- (1) 设计输入。将所要设计的电路以开发软件要求的某种形式表达出来，并输入到相应软件中，主要包括原理图输入和 HDL 文本输入；
- (2) 综合。将较高层次的设计描述自动的转化为较低层次描述；
- (3) 适配。将综合器产生的网表文件配置于指定的目标器件中，并产生最终的可下载文件；
- (4) 仿真。完成对所设计电路功能的验证，包括时序仿真和功能仿真。
- (5) 编程。把配置后生成的编程文件装入到 PLD 器件中，通常有基于

E²PROM 工艺和 SRAM 工艺两种编程模式。

2.4 本章小结

本章简要概述了数字电视地面广播传输系统国家标准，分析其数据的传输流程和功能模块实现的复杂程度，据此确定发端系统的结构以及实现方案，制定了基本模块结构与设计层次。并对本项目所使用的开发工具和开发器件做了简单的介绍。

第三章 数字电视发端系统的 FPGA 实现

3.1 能量扩散模块（随机化）

能量扩散模块实现的是将扰码与输入的数据码流进行逐位模二加，从而生成数据扰乱码，以保证传输数据的随机性以便于传输信号处理。

扰码为由线性移位寄存器（LFSR）生成的二进制伪随机序列。其生成多项式定义为^[1]：

$$G(x) = 1 + x^{14} + x^{15} \quad (3-1)$$

LFSR 结构如图 3-1 所示：

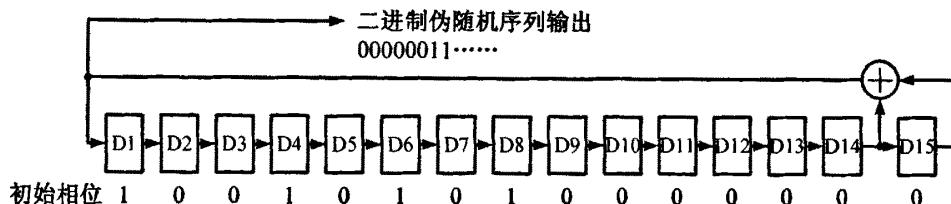


图 3-1 线性反馈移位寄存器

该 LFSR 的初始状态定义为 100101010000000。扰码器的移位寄存器在信号帧开始时复位到初始状态。由帧起始信号进行控制。在数据输入处理中，每一个信号帧需要对线性移位寄存器复位到初始状态。

模块输入码流为位宽 8bits 的 TS 比特码流，输入的 TS 码流首先要进行一次时钟变换，是输入数据与系统模块时钟同步，采用一个异步 FIFO 即可实现。考虑到 TS 输入时钟低于系统工作时钟，FIFO 深度为 2 即可满足要求，避免硬件资源浪费。

从简化模块结构以及编码模块的需要，这里将输入码流改成按位输出的 bit 码流，采用一个数据结构变换模块实现功能。通过计数器控制寄存器按位输出即可实现。从而能量扩散模块的整体结构如图 3-2：

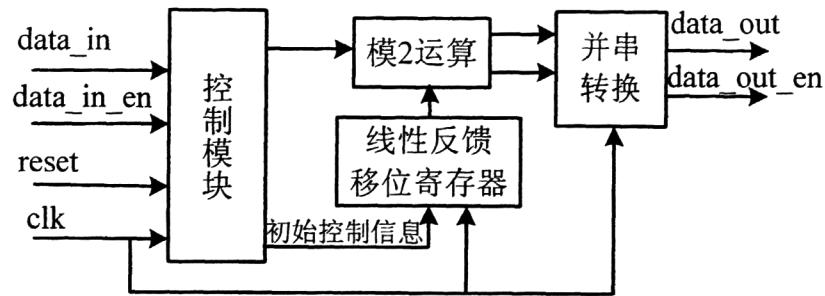


图 3-2 能量扩散 FPGA 实现结构

其结构定义如下：

`data_in`: 输入的 8bits 宽的 TS 流信号;

`data_in_en`: 输入数据有效信号, 高电平有效;

`reset`: 系统复位信号, 低电平有效;

`clk`: 模块工作时钟, 输入数据也同步为此工作时钟;

`data_out`: 输出的数据扰乱码, 按位输出;

`data_out_en`: 输出数据有效信号, 高电平有效。

当输入 8bits 随机数组, 能量扩散模块经 Quartus 仿真, 其输出波形如图 3-3 所示:

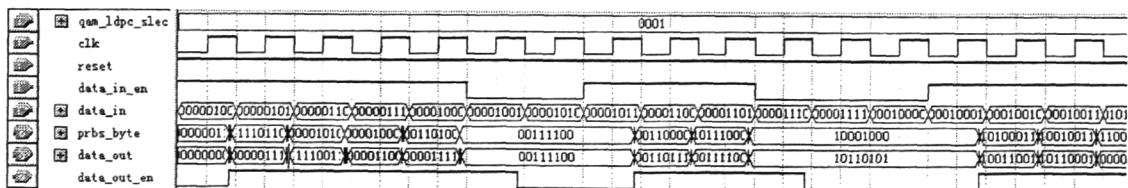


图 3-3 能量扩散仿真波形

为便于观察, 此图中输出数据为能量扩散模块中间数据, 输出信号 ‘`data_out`’ 信号需经过一个并串转换模块后, 方可按 bit 输出所需数据。图中可以清楚的看到, 输入的随机信号 ‘`data_in`’, 与线性移位寄存器生成的扰码数据 ‘`prbs_byte`’ 逐位模二加后, 生成符合系统要求的输出数据 ‘`data_out`’ 。

能量扩散 (energy_dispersal) 模块实现的资源消耗如图 3-4:

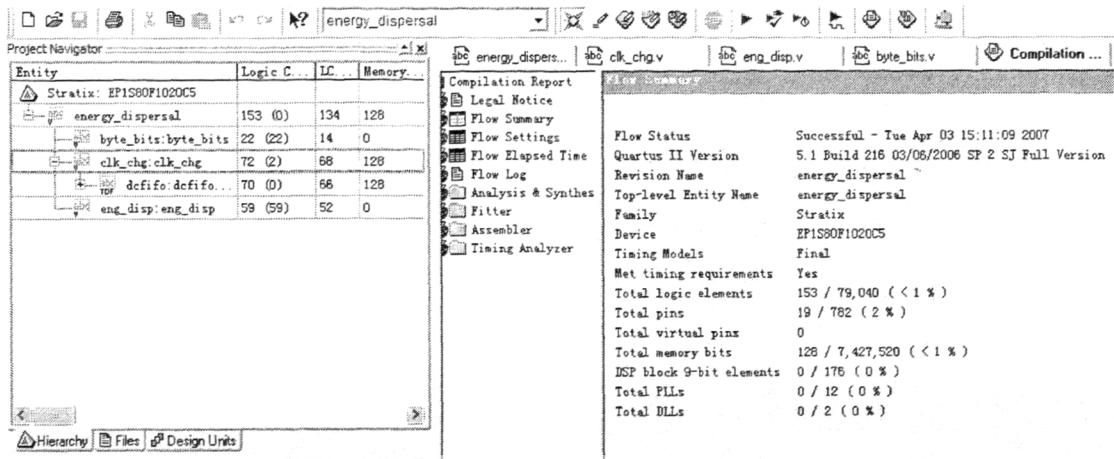


图 3-4 能量扩散模块 FPGA 实现的资源消耗图

3.2 信道编码模块

国标系统采用了前向纠错编码技术(FEC)。前向纠错编码技术是国标相对于国外现有的 3 个地面数字电视广播标准的主要区别之一，由外码 (BCH 码) 和内码 (LDPC 码) 级联实现。

3.2.1 BCH 编码

BCH 码是循环码的一种，它能够纠正随机产生的比特错误^[6]。根据线性分组码的原理，BCH (1023, 1013) 码可纠 1 bit 的突发错误。BCH 码在系统中可以起到两个重要的作用^[4]: 首先 BCH 可以降低误码平层；其次，BCH 码可以完成速率适配，使得在每个信号帧里面有整数个 MPEG 包。在我国地面数字电视传输标准中，FEC 编码的外码 BCH 码为 BCH(762, 752)，BCH(762, 752) 码是由 BCH(1023, 1013) 系统码缩短而成的。它在 752 比特信息位前添加 261 比特 0 成为 1013 长度，然后进行 BCH (1023, 1013) 编码，BCH (1023, 1013) 生成多项式为 $G_{BCH}(x)=1+x^3+x^{10}$ 。编码后得到 1023 bits 码字，然后删除前 261 bits 的 0，得到了码长为 762bits 的 BCH 码。

在本设计中，为实现 BCH 码的编码运算，采用了线性反馈移位寄存器来实现。由循环码基本原理可知，BCH(1023, 1013) 系统码的编码可由图 3-5 的循环码编码器来实现。本文采用的为 n-k 级编码器。

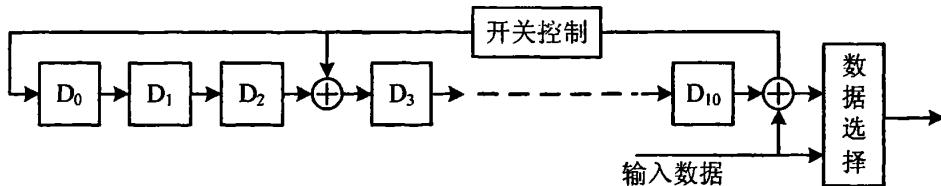


图 3-5 系统码编码电路

移位寄存器初始状态为 0。输入数据时，开关控制关闭，数据选择输出输入的数据，即信息组输出；当数据输入完毕，开关打开，数据选择寄存器内数据移位输出校验元。

由上，则 BCH 编码模块整体结构如图 3-6 所示

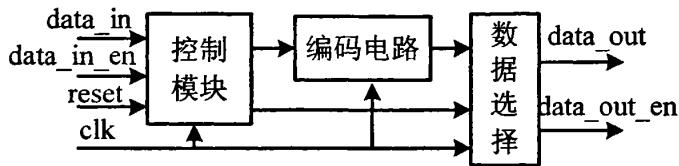


图 3-6 BCH 编码模块实现结构图

控制模块内，由计数器产生控制信号，对输入数据进行缓冲控制，并控制开关电路的状态以及数据选择器的选择。并在每一组编码结束后，将编码电路的移位寄存器进行初始化。

接口定义如下

data_in: 输入数据，1bit 位宽，与 clk 同步；

data_in_en: 输入数据有效信号，高电平有效；

reset: 系统复位信号，低电平有效；

clk: 模块工作时钟；

data_out: BCH 编码完成后的输出数据，1bit 位宽。

data_out_en: 输出数据有效信号，高电平有效。

3.2.2 LDPC 编码

低密度校验（LDPC）码^[7]是 Gallager 提出的一类可用稀疏校验矩阵描述的线性分组码。因为其在迭代译码条件下逼近 Shannon 限的能力^[8]，近年来受到越来越

多的关注，并且，已成功的应用于多个行业标准。

LDPC 码主要分为随机化 LDPC 码和结构化 LDPC 码两大类。前者因为其随机性有较优异的性能，但是，其在编译码时需要完整地存储整个生成矩阵或校验矩阵，从而耗费大量的硬件资源，并带来较大的编译码复杂度。而后者因为其结构性特征，可从存储的生成矩阵和校验矩阵部分行列信息递推完成编译码，为硬件实现带来极大的便利。国标采用的准循环 LDPC 码即是一种结构化 LDPC 码。本文根据其结构，描述了在一个发射机内可实现三种码率编码的编码器的 FPGA 设计方案。

3.2.2.1 LDPC 编码的计算结构

国标系统的 LDPC 编码采用了基于矩阵分解中的两个信息符号的 RS 码法，构造 LDPC 码的循环置换矩阵，得到其生成矩阵 \mathbf{G}_{qc} 如下所示：

$$\mathbf{G}_{qc} = \begin{bmatrix} \mathbf{G}_{0,0} & \mathbf{G}_{0,1} & \cdots & \mathbf{G}_{0,c-1} & \mathbf{I} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{G}_{1,0} & \mathbf{G}_{1,1} & \cdots & \mathbf{G}_{1,c-1} & \mathbf{O} & \mathbf{I} & \cdots & \mathbf{O} \\ \vdots & \vdots & & \mathbf{G}_{i,j} & \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{k-1,0} & \mathbf{G}_{k-1,1} & \cdots & \mathbf{G}_{k-1,c-1} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{I} \end{bmatrix} \quad (3-2)$$

其中， \mathbf{I} 是 $b \times b$ 阶单位矩阵， \mathbf{O} 是 $b \times b$ 阶零阵，而 $\mathbf{G}_{i,j}$ 是 $b \times b$ 循环矩阵，令 $0 \leq i \leq k-1$, $0 \leq j \leq c-1$ 。循环矩阵 $\mathbf{G}_{i,j}$ 内的每一行都是上一行的向右一位循环移位，此方阵的第一行是此方阵的最后一行的向右一位循环移位；此方阵内的每一列都是左一列的向下一位循环移位，并且第一列是最后一列的向下一位循环移位。这就保证了其生成矩阵的每个子矩阵可以由其第一行或者第一列表征^[1]。这种循环矩结构可以极大的减少生成矩阵的存储量。国标提供了生成矩阵中各循环矩阵 $\mathbf{G}_{i,j}$ 的生成多项式，即每个循环矩的第一行的行列式。

由生成矩结构可以看出，国标系统 LDPC 编码输出数据的码结构为信息位在后，校验位在前。

同时，国标系统规定了三种不同内码码率的 FEC 码，其结构分别为：

(1) 码率为 0.4 的 FEC (7488, 3008) 码：

先由 4 个 BCH (762, 752) 码和 LDPC (7493,3048) 码级联构成，然后将 LDPC (7493,3048) 码前面的 5 个校验位删除。LDPC (7493,3048) 码的生成矩阵 \mathbf{G}_{qc} 具有上式所示的矩阵形式，其中参数 $k=24$, $c=35$ 和 $b=127$ 。

(2) 码率为 0.6 的 FEC (7488, 4512) 码:

先由 6 个 BCH (762, 752) 码和 LDPC (7493,4572) 码级联构成, 然后将 LDPC (7493,4572) 码前面的 5 个校验位删除。LDPC (7493,4572) 码的生成矩阵 \mathbf{G}_{qc} 具有上式所示的矩阵形式, 其中参数 $k=36$, $c=23$ 和 $b=127$ 。

(3) 码率为 0.8 的 FEC (7488, 6016) 码:

先由 8 个 BCH (762, 752) 码和 LDPC (7493,6096) 码级联构成, 然后将 LDPC (7493,6096) 码前面的 5 个校验位删除。LDPC (7493,6096) 码的生成矩阵 \mathbf{G}_{qc} 具有上式所示的矩阵形式, 其中参数 $k=48$, $c=11$ 和 $b=127$ 。

3. 2. 2. 2 LDPC 编码的 FPGA 实现结构

三种不同码率的编码, 区别仅在于校验位与信息位的比例不同, 在 FPGA 设计与实现中, 采用同一种运算模式。由标准提供的生成矩阵 \mathbf{G}_{qc} 结构, 我们只要求取前面 $c \times b$ 个校验位就可以得到 LDPC 码编码后码字。后 $k \times b$ 的信息位可直接将输入数据存储后输出即可。

由标准提供的生成矩阵 \mathbf{G}_{qc} , 其内部各个子矩阵 $\mathbf{G}_{i,j}$ 是 $b \times b$ 循环矩阵。这一结构特性大大简化了编码的硬件结构。在本设计中, 以 $b=127$ 的循环矩为基本计算单元。分解整个编码过程, 即为 127bits 的输入与 127×127 的循环矩阵进行矩阵相乘。对同一列的相乘结果进行异或累计, 可以得到 127bits 的校验位。

(1) 并行输入方案:

按上述结构设计, 逻辑最简单的方案为设计一个 1×127 的行列式与 127×127 的矩阵相乘的运算模块。通过串并转换将输入信息结构修改为 127bits 的并行输入。然后通过控制模块按顺序更改运算模块的矩阵信息。并对运算结果按相应顺序进行累积, 即可以完成 LDPC 编码。

但是, 从系统处理时钟速率角度考虑。对于同一行的子矩阵, 对应的输入为相同的 127bits 输入。但是系统需要在两个 127bits 输入的时钟间隔内, 完成 C 次 127 阶的矩阵乘法。这一时钟限定决定了 127 阶矩阵相乘的运算模块需要在短时间内完成运算, 从而需要依靠扩展硬件空间来提高处理速度, 即消耗大量硬件资源的方式。实际设计中, 此方案消耗 19325 个基本逻辑单元, 工作时钟最高为 40.55MHz。虽然满足系统需要, 但是消耗过多系统资源。

通过并行处理运算模块, 增加模块处理时钟数, 降低运算模块复杂度, 以减

小每个运算模块的资源消耗，并且可以进一步提高整体编码模块工作时钟。但是并行需要的运算模块资源消耗也倍增。故此方案虽可满足本系统要求，但成本较高，工作时钟限制较大，不利于模块移植复用。

(2) 串行输入并行输出方案：

综合考虑系统的运行时钟要求以及节省硬件资源，本方案最终采用串行输入并行输出方案。这一方案以简化单个运算模块结构，大量并行处理的方式，以串行 1bit 输入数据，并行 127bits 输出。

对于每一个码字的校验位的生成，其生成公式为 $C = D \times G$, C 为校验位， D 为输入的信息位， G 为生成矩阵 G_{qc} 的前半部校验位生成矩阵部分。针对 G 的特殊结构，可以将公式分解为：

$$[C_0, C_1, \dots, C_{k-1}] = [D_0, D_1, \dots, D_{k-1}] \times \begin{bmatrix} G_{0,0} & G_{0,1} & \cdots & G_{0,c-1} \\ G_{1,0} & G_{1,1} & \cdots & G_{1,c-1} \\ \vdots & \vdots & \ddots & \vdots \\ G_{k-1,0} & G_{k-1,1} & \cdots & G_{k-1,c-1} \end{bmatrix} \quad (3-3)$$

可以看出对于生成矩阵 G 中同一行的子矩阵 $G_{i,0} \sim G_{i,c-1}$ ，其运算所对应的数据 D_i 是相同的。理论上针对同一行的生成子矩阵 可以同步进行运算。但是考虑到硬件实现中，无法同步同一行子矩阵的初始信息更新。因此这种同一行子矩阵同步运算的方式无法实现，本文中采用流水线的方式完成对同一行子矩阵的运算。

其中单个循环矩的运算模块 $D_i \times G_{i,j}$ 结构如图 3-7^[9]:

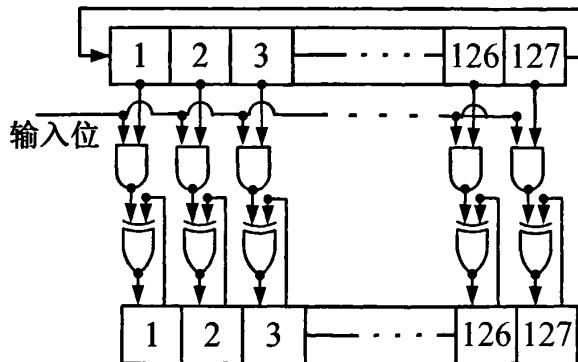


图 3-7 单个循环矩的实现结构

首先，将国标附录中的生成矩的各个循环矩的生成多项式信息（以下简称为 g_{ij} ）存入 ROM 中。利用生成矩的循环特性，通过将循环矩阵行信息存入移位寄存器，然后与输入数据同步循环移位。每个时钟完成一位输入数据与相对应行相乘

的运算，同时将结果与对应寄存器中数据按位异或并存入寄存器。从而并行处理输入数据与循环矩各列的乘运算及结果相加。

由生成矩阵 \mathbf{G}_{qc} 的结构的规则性，同一行的循环矩阵 \mathbf{G}_{ij} 在编码时有着完全同时钟的运算过程，可简单并行采用 c 个如图 3-7 的运算结构即可实现 LDPC 编码。但是考虑到生成矩的 $b=127$, c 最大达到 35，并行读取一行循环矩的位宽达到 $b \times c$ 个比特，显然在硬件实现中，很难满足。故在本设计中，通过对 c 个运算模块进行时序控制，以流水线的运算方式完成整个模块的运算。模块运算结构简易示意图如图 3-8 所示：

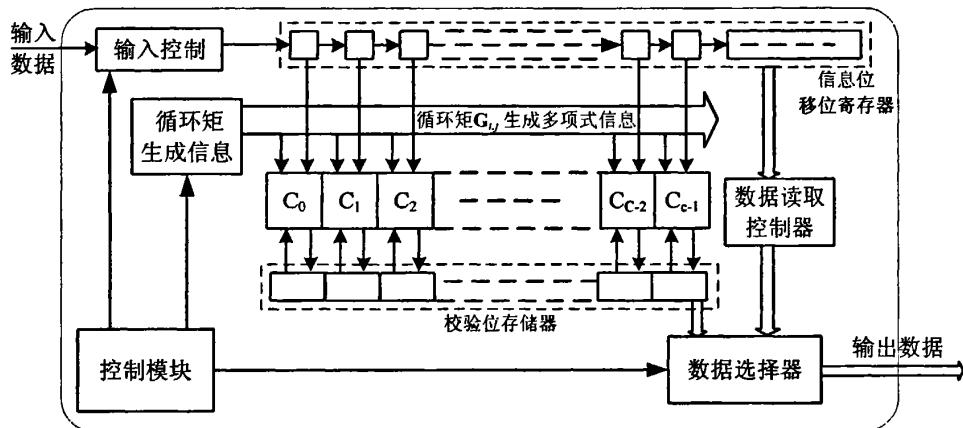


图 3-8 LDPC 模块运算结构示意图

图中， $C_0 \sim C_{c-1}$ 为 c 个并行的如图 3-7 的运算结构。整个编码模块运行操作过程如下：

初始化时，各寄存器置零。

第一个数据输入有效时钟，当第一个有效输入数据 d_0 到来时，将 d_0 写入信息位移位寄存器的第一个存储位（1bits），同时将 $\mathbf{G}_{0,0}$ 的生成多项式信息 $g_{0,0}$ 写入 C_0 。

第二个数据输入有效时钟，此时模块完成两个部分的操作：

运算部分：将数据 d_0 写入 C_0 进行运算，运算结果与校验位存储器中相对应数据按位异或后存储到校验位存储器的第一个存储位置（127bits）。

模块内数据调整部分：信息位移位寄存器进行一次移位，将新输入数据 d_1 存入第一个存储位。 C_0 内行列式进行 1bit 的移位。同时，通过控制模块控制将 $\mathbf{G}_{0,1}$ 的生成多项式信息 $g_{0,1}$ 写入模块 C_1 。

第三个数据输入有效时钟，此时钟操作与上一时钟操作类似：

运算部分：将数据 d_1 、 d_0 分别写入 C_0 、 C_1 进行运算，运算结果与校验位存储器中相对应数据按位异或后存储到校验位存储器的第一和第二个存储位置（127bits）。

模块内数据调整部分：信息位移位寄存器进行一次移位，将新输入数据 a_2 存入第一个存储位。 C_0 、 C_1 内行列式进行 1bit 的移位。同时，通过控制模块控制将 $\mathbf{G}_{0,2}$ 的生成多项式信息 $g_{0,2}$ 写入模块 C_2 。

以后时钟采用同样方式进行运算操作。即每个 C_{i+1} 比 C_i 要晚一个时钟进行同样的运算。当第 C 个时钟后，所有并行的单个循环矩阵同时进行运算。此时 C_0 尚未完成 127 循环。故所有操作步骤与上相同，但是不再写入新的生成多项式信息。

第 128 个数据输入有效时钟时，此时 C_0 已完成 127 次循环，即循环矩阵 $\mathbf{G}_{0,0}$ 参与的编码运算已结束，此时，将第二行的循环矩阵 $\mathbf{G}_{1,0}$ 的生成多项式信息 $g_{1,2}$ 写入 C_0 。开始第二行循环矩阵的运算。

按上述步骤循环操作即可完成一组 LDPC 编码。

数据输出部分的操作：

最后一行循环矩阵参与运算过程时，当循环矩阵 $\mathbf{G}_{k-1,0}$ 的生成多项式信息 $g_{k-1,0}$ 循环结束后，即当第 $k \times 127 + 1$ 个输入有效时钟时，此时输入的数据为参与第二组 LDPC 编码的第一个输入数据，模块操作等同于上述的第一个时钟。此时第一组编码即将完成第一个校验位信息（127bits）。故此时需要将运算后存储到校验位存储器第一个存储单元内的数据直接输出，并将此存储单元清零。同理，此后连续 c 个时钟，将连续完成 c 个 127bits 校验位的生成。并连续并行输出第一组编码的校验位信息。

国标系统的 FEC(7488, 3008) 码是由 BCH(762, 752) 码和 LDPC(7493, 3048) 码级联构成，然后将 LDPC(7493, 3048) 码前面的 5 个校验位删除。故上述输出每一组 LDPC 编码后，需要对每一组编码的第一组校验位信息（127bits）输出做出指示，这里采用一位指示信号 ‘first_127’ 实现。在之后模块通过对这一指示信号判别，在进一步数据处理中删除整个编码前面的 5 个校验位。

c 个时钟后，校验位信息输出完毕。此时信息位移位寄存器中已存储了第一组编码的信息位信息，共 $k \times b$ 位。同时也已输入存储了 c 位参与第二组编码的信息

位。之后，通过数据读取控制器，按时钟将信息位移位寄存器中的信息位数据并行输出（此处需要考虑输入数据的移位影响）。由于输出数据是按 127bits 并行输出，输入数据是串行的 1bit 输入，故信息位移位寄存器深度需要 $k \times b + c$ 位即可。

按上述步骤循环操作，即可完成连续串行输入数据的 LDPC 编码，并按 127bits 并行输出。

3.2.2.3 LDPC 编码模块接口定义

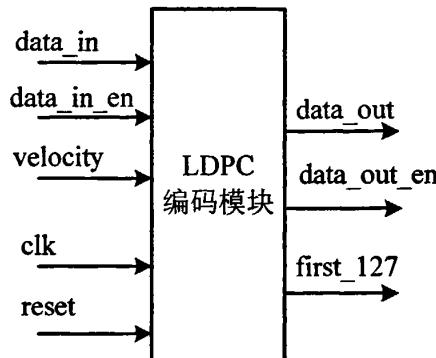


图 3-9 LDPC 编码模块接口定义

clk: 模块工作时钟；

reset: 系统复位信号，低电平有效；

data_in: 输入数据，1bit 位宽，与 clk 同步；

data_in_en: 输入数据有效信号，高电平有效，与 clk 同步；

velocity: LDPC 编码速率选择信号，与 QAM 选择信号共用 4bits 位宽，0.4、0.6、0.8 码率，并由此决定模块中 k 与 c 的大小选择；

data_out: LDPC 编码完成后的输出数据，127bits 位宽，与 clk 同步；

data_out_en: 输出数据有效信号，高电平有效，与 clk 同步；

first_127: 输出编码第一组 127 校验位指示信号，高位有效，与 clk 同步。

3.2.3 信道编码模块的 QuartusII 仿真验证

前向纠错编码（FEC_coder）模块中，BCH 模块结构较为简单，为便于分析 LDPC 模块数据结构，这里给出 LDPC 模块的仿真波形。

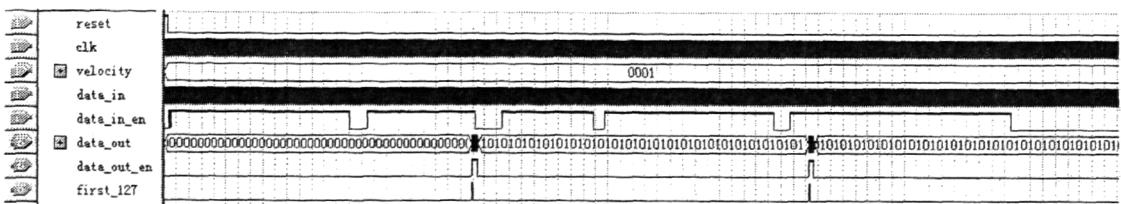


图 3-10 LDPC 模块仿真波形

图 3-10 所示为在 ‘velocity’ 信号为 0001 时（既采用 0.4 码率 LDPC 编码），输入随机数据的仿真输出波形。从图中可以看出，每 3048bits 的输入后，将连续输出 59*127bits 的数据。从第二个数据块可以看出，此结构可以联系处理输入信号，不必进行大量数据缓冲存储。

信道编码模块，即前向纠错编码（FEC_coder）模块由如上所述的 BCH 编码和 LDPC 编码级联构成，其资源消耗如图 3-11 所示：

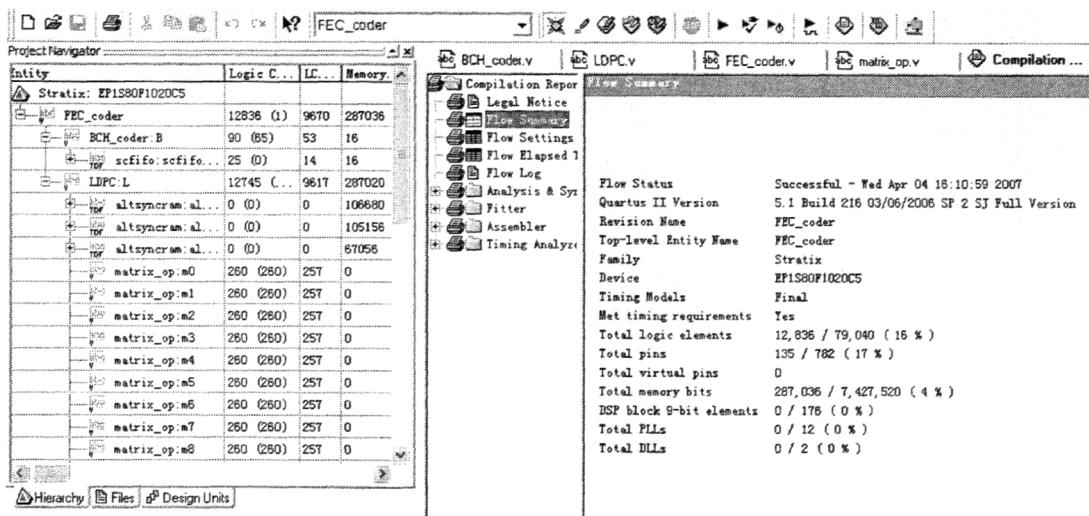


图 3-11 前向纠错编码模块 FPGA 实现资源消耗图

3.3 帧体形成模块

帧体形成模块主要功能为将编码后的数据流经过处理后，形成 3780 点帧体数据，便于后面的帧体数据处理。主要包括符号星座映射、符号交织、系统信息插入三个部分。

3.3.1 符号星座映射

前向纠错编码后的数据流要转换成均匀的 nQAM (n: 星座点数) 符号流。国标系统包含以下几种符号映射关系：64QAM、32QAM、16QAM、4QAM、4QAM-NR。各种符号映射加入相应的功率归一化因子，使各种符号映射的平均功率趋同。

由星座图可得：对于 QPSK，每 2 比特对应于 1 个符号，信号电平的选择分别为 -4.5 、 $+4.5$ ；对于 16QAM，每 4 比特对应于 1 个符号，信号电平的选择分别为 -6 、 -2 、 $+2$ 、 $+6$ ；对于 32QAM，每 5 比特对应于 1 个符号，信号电平的选择分别为 -7.5 、 -4.5 、 -1.5 、 $+1.5$ 、 $+4.5$ 、 $+7.5$ ；对于 64QAM，每 6 比特对应于 1 个符号，信号电平的选择分别为 -7 、 -5 、 -3 、 -1 、 $+1$ 、 $+3$ 、 $+5$ 和 $+7$ ；对于 QPSK+NR，按比特交织之后，根据一定的映射规则进行一个 8 比特到 16 比特的 NR 映射，然后把映射后每 2 个比特根据 QPSK 映射成一个符号。

综合考虑硬件实现中的数据结构以及精确度的要求，本设计将 QAM 映射后的符号扩展为 32bits，高 16 位为实部映射数据，低 16 位为虚部映射数据。利用存储器 ROM 存储各种星座映射对应的符号数据。将输入的数据改造为合适的位宽后作为存储器 ROM 的读地址信号。即可实现将数据流转换成均匀的 nQAM 符号流。

QAM 映射的 FPGA 实现框图如图 3-12：

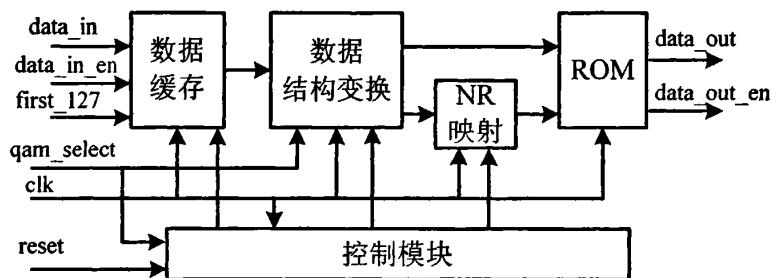


图 3-12 QAM 映射的实现结构框图

数据缓存是考虑到 FEC 编码后输出为 127bits 数据，并且为连续输出，在数据结构变换时，需要对输入数据进行缓存，同时，根据 ‘first_127’ 信号对输入的编码数据的第一个 127 位数据的前五位进行删除。利用一个位宽 127，深度为 59 的 FIFO，以及一个数据选择控制模块即可实现。数据选择控制模块作用是根据 ‘first_127’ 信号，将每组 FEC 编码的第一组数据直接删除前五位后送入结构变换模块，而不存入 FIFO。

数据结构变换模块，是根据 ‘qam_select’ 信号，对输入数据进行结构变换。即，根据不同的 QAM 映射选择，将输入数据变换为 2、4、5、6 位宽的并行数据，并根据不同映射对应的符号数据存储的 ROM 地址不同，将并行数据更改位合适的 ROM 地址信息。同时，如果是 QPSK+NR 调制方式，则还需对输入数据进行相对应的映射调整。

根据国标要求，存储器ROM深度为116，位宽为32。其中地址0~3存储QPSK及QPSK+NR对应的4个符号数据；地址4~19存储16QAM对应的16个符号数据；地址20~51存储32QAM对应的32个符号数据；地址52~115存储64QAM对应的64个符号数据。

以上各模块的协作运行，由控制模块输出的控制信息进行控制，以保证数据流的连续变换。模块接口定义为：

clk: 模块工作时钟；

reset: 系统复位信号，低电平有效；

data_in: 输入数据，127bits 位宽，与 clk 同步；

data_in_en: 输入数据有效信号，高电平有效，与 clk 同步；

first_127: 输入编码数据的组首位指示信号，高位有效，与 clk 同步。

qam_select: 星座映射 QAM 选择信号，与 LDPC 码率选择信号共用 4bits 位宽。

data_out: 映射完成后的输出数据，32bits 位宽，与 clk 同步；

data_out_en: 输出数据有效信号，高电平有效，与 clk 同步。

3.3.2 符号交织

国标系统采用了时域符号交织技术以提高抗脉冲噪声干扰能力。对数据进行符号交织是针对地面无线广播信道的时域和频域选择衰落特性以及收端纠正突发错误解码的需要。国标系统交织部分的时域符号交织编码是在多个信号帧的基本数据块之间进行的，采用基于星座符号的卷积交织编码。有两种交织模式，分别为：B（交织宽度）=52，M（交织深度）=240 和 B=52，M=720 的时域符号交织模式。时域交织可选的两种工作模式结构一致，仅 M 取值不同。

3.3.2.1 实现方案比较

由卷积交织的结构，最简单直接的实现方式是通过一组移位寄存器来实现。QAM 映射厚的数据由数据分配器依次写入不同深度的移位寄存器，并根据不同移位寄存器的存储状态，通过数据选择器将寄存器内移位输出的数据选择输出，结构简图如图 3-13：

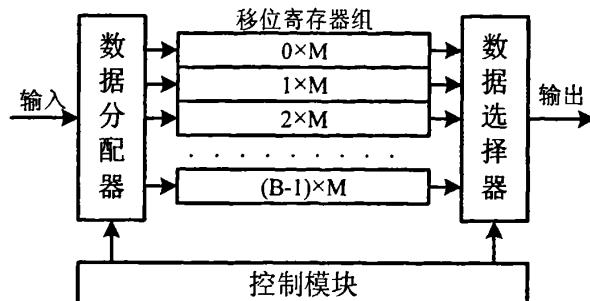


图 3-13 移位寄存器实现简图

移位寄存器方式实现结构简单，但是可以看出，发送端交织部分需要的缓冲存储空间达到 $M \times (B-1) \times B/2$ 个符号位。对于国标系统来说，两种模式需要的缓冲存储空间分别达到 318240 和 954720 个符号。对于本设计，每符号位宽为 32bits，则需要的存储空间过大。对于 FPGA 实现来说，移位寄存器方式实现交织编码是不允许的。

对于大量存储交换数据的实现，有种常用的方式为采用双口 RAM 来实现。对于交织编码来说，结构与图 3-13 类似。不同的是：图中移位寄存器组改为深度为 $\sum_{b=0}^{B-1} b \times M (B = 52)$ 的双口 RAM，输入数据分配器改为由两级计数器实现的数据地址

产生器，产生与输入数据相对应的 RAM 写地址，从而完成存储功能。同时输出数据选择器通过两级计数器，以及 RAM 存储的状态信号来完成输出数据的选择地址进行数据读取。这一工作模式在交织长度适度的情况下常为采用，如 DVB-T 标准中的“outer interleaver”的实现。但对于国标系统，这种方式消耗硬件资源仍然过大。因此本文采用的方式为利用片外 SDRAM 来实现数据的存储。

3.3.2.2 交织模块的 FPGA 实现结构

在各种随机存储器件中，SDRAM 的价格低、体积小、速度快、容量大，是比较理想的器件。所以在本设计中使用外部的 SDRAM 来完成交织部分的数据存储。通过一个控制模块产生存储读取地址，通过一个 SDRAM 控制器对 SDRAM 进行读写操作，从而完成时域交织。模块功能框图如图 3-14 所示：

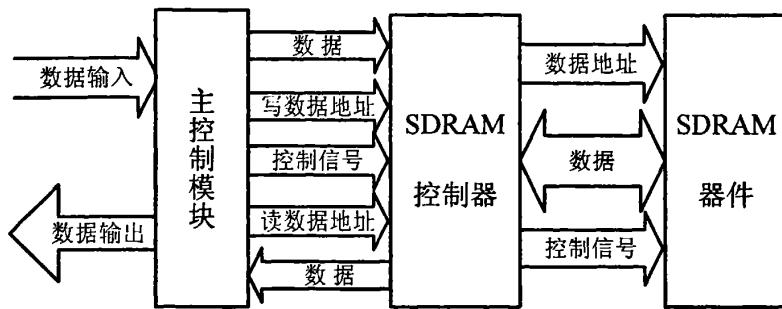


图 3-14 时域卷积交织的模块功能框图

其中，主控制模块内分别采用两级计数器生成输入输出信号的地址信号，同时通过计数器数值判断生成读写控制信号，与数据同步传输给 SDRAM 控制器。由控制器进行读写操作。主控制模块实现结构如图 3-15：

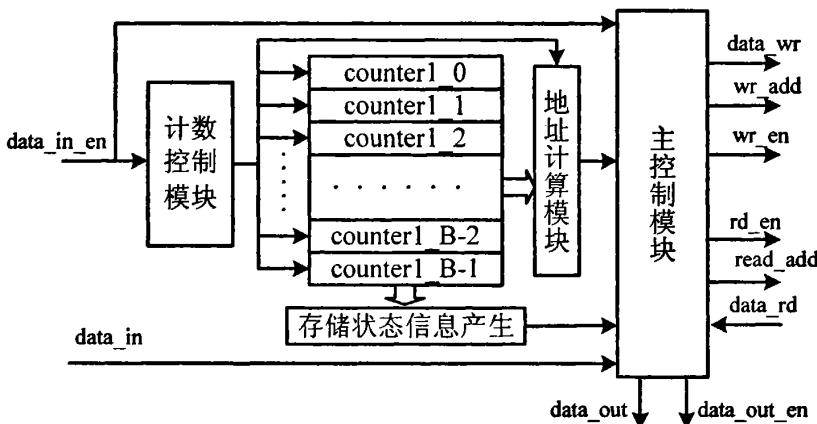


图 3-15 交织控制模块结构图

交织部分 SDRAM 中所需要的存储空间为 $\sum_{b=0}^{B-1} b \times M (B = 52)$ 个符号位，本设计将这

$\sum_{b=0}^{B-1} b \times M (B = 52)$ 个符号分为 B 个组，每组的深度为 $b \times M$ ($b = 0 \sim B-1$)。图中计数

控制模块即为一级计数器 counter0，计数范围为 $0 \sim B-1$ ，通过这机计数器循环计数来分配数据按顺序存储到相对应的存储组中。图中 counter1_i，为一组二级计数器，计数范围为 $0 \sim b \times M - 1$ ，这一组计数器是针对每一个存储组的计数，确定数据按顺序存储到组中相应的位置。所有计数器通过 ‘data_in_en’ 信号激发，与输入数据同步计数。图中地址计算模块是计算出输入数据相对应的 SDRAM 存储地址。

由交织结构可确定组 b 的存储起始位置为 $\sum_{b=0}^{B-1} i \times M$ 。则通过对计数器 counter0 和

counter1 组的计算可得输入数据的存储地：

$$add = \sum_{counter0=0}^{counter0-1} i \times M + counter1_counter0 \quad (3-4)$$

主控制模块得到输入数据，以及相对应的地址信息，生成写使能信号“wr_en”给 SDRAM 控制模块，即可完成数据写入部分的操作。

初始化时，SDRAM 为空，所有计数器为 0。故在开始阶段，当存储组没写满时没有输出信号。只有当每组存储组写满时，才有输出信号。因此，对应每一个存储组，设定一个存储状态信息。初始化时置为低电平。当存储组 b 第一次存满时，即计数器 counter1_b 计数到 $b \times M - 1$ ，第一次循环复位为 0 时将存储状态信息置为高电平，此时意味这下一次将数据存储到这一个存储组时，需要先将 SDRAM 此数据对应位置的数据输出。当所有存储组的状态信息都为高位时，则意味着每一次数据输入存储时，都需要先输出一组数据，输出数据的地址与输入数据的地址相同。

主控制模块得到输入数据，以及相对应的地址信息时，通过对存储信息判断生成写使能信号‘rd_en’，与地址信息同步传送给 SDRAM 控制模块，即可完成数据写入部分的操作。而输入数据和相应地址信息则缓存一个时钟再输给 SDRAM 控制模块。读写操作依次循环，实现符号卷积交织功能。交织模块接口定义为：

clk：模块工作时钟；（图中省略）

reset：系统复位信号，低电平有效；（图中省略）

data_in：输入数据，一个符号位宽，与 clk 同步；

data_in_en：输入数据有效信号，高电平有效，与 clk 同步；

data_out：交织完成后的输出数据，一个符号位宽，与 clk 同步；

data_out_en：输出数据有效信号，高电平有效，与 clk 同步。

3.3.3 帧体成型模块

在 2.1.3 节可看出，国标系统定义了一个信号帧的 3780 个子载波中包含 3744 个子载波传输载荷数据和 36 个子载波传输系统信息信号。系统信息为每个信号帧提供必要的解调和解码信息，包括符号映射方式、LDPC 编码的码率、交织模式信

息、帧体信息模式等。在接收机端通过对 TPS 信号的解调可得到发射机信息，方便接收机端自动适应发端的工作模式，实现自动接收功能。

3.3.3.1 系统信息数据结构

国标系统中预设了 64 种不同的系统信息模式，并采用扩频技术传输。这 64 种系统信息在扩频前可以用 6 个信息比特 ($s_5s_4s_3s_2s_1s_0$) 来表示，其中 s_5 为 MSB，定义如下： $s_3s_2s_1s_0$: 编码调制模式； s_4 : 交织信息； s_5 : 保留。该 6 比特扩频前的系统信息将采用扩频技术成为 32 比特长的系统信息矢量，即用长度为 32 的 Walsh 序列和长度为 32 的随机序列来映射保护。鉴于扩频技术较为繁琐，而信息系统的组合模式不多。对于传输的任何一种系统模式，通过国标附录 G 可以得到需要在信道上传输的 32 比特长的系统信息矢量。故在硬件实现中，通过映射，可以直接简单的得到 32 比特长的系统信息矢量。然后将这 32 比特采用 I、Q 相同的 4QAM 调制映射成为 32 个复符号。

系统信息矢量长度为 32 个复符号，在其前面再加 4 个复符号作为数据帧体模式的指示，全零的四个比特指示子载波数 C=1 对应的帧体数据模式，这四个比特的其他数值保留将来使用。前置的这四个比特也采用 I、Q 相同的 4QAM 映射为 4 个复符号。该 36 个系统信息符号通过复用模块与信道编码后的数据符号复合成帧体数据，其复用结构为：36 个系统信息符号连续的排列于帧体数据的前 36 个符号位置。C=1 和 C=3780 两种模式通用的帧体结构如图 3-16：

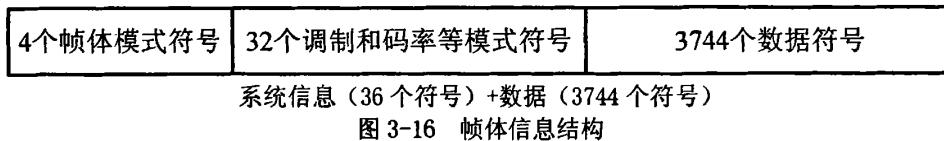


图 3-16 帧体信息结构

对于 C=3780 模式，帧体部分还需要经过频域交织，目的是将调制星座点符号映射到帧体包含的 3780 个有效载波上。频域交织为帧体内的符号块交织，交织大小等于载波数 3780。具体交织运算过程如下^[1]。

(1) 为了使交织输出时 36 个系统信息符号集中放置，首先将这 36 系统信息符号插入到 3744 个数据符号中，其插入位置构成的集合为：

{ 0, 140, 279, 419, 420, 560, 699, 839, 840, 980, 1119, 1259, 1260, 1400, 1539, 1679, 1680, 1820, 1959, 2099, 2100, 2240, 2379, 2519, 2520, 2660, 2799, 2939, 2940, 3080, 3219, 3359, 3360, 3500, 3639, 3779 }。

插入后得到的序列由数组 Z [3780] 表示。

(2) 将数组 Z [3780] 通过以下程序进行位置调换得到最终交织输出序列 Y [3780]:

```
for(i=0; i<3; i=i+1); for(j=0; j<3; j=j+1); for(k=0; k<3; k=k+1); for(l=0; l<2; l=l+1);
for(m=0; m<2; m=m+1); for(n=0; n<5; n=n+1); for(o=0; o<7; o=o+1)
Y[o*540+n*108+m*54+l*27+k*9+j*3+i]=
Z[i*1260+j*420+k*140+l*70+m*35+n*7+o];
```

国标系统附录提供了步骤二的交织图样，大大方便了频域交织的硬件实现。

3.3.3.2 系统信息插入，频域交织的 FPGA 实现

从上一小节的分析可以看出，系统信息插入部分的实现主要分为两个步骤：对发射参数选择的 6bits 系统信息扩频为 32 个复符号，加上 4 个帧体模式符号构成 36 个复符号；将 36 个系统信息符号与输入数据复用为完整的帧体信息。C=3780 模式实现结构如图 3-17：

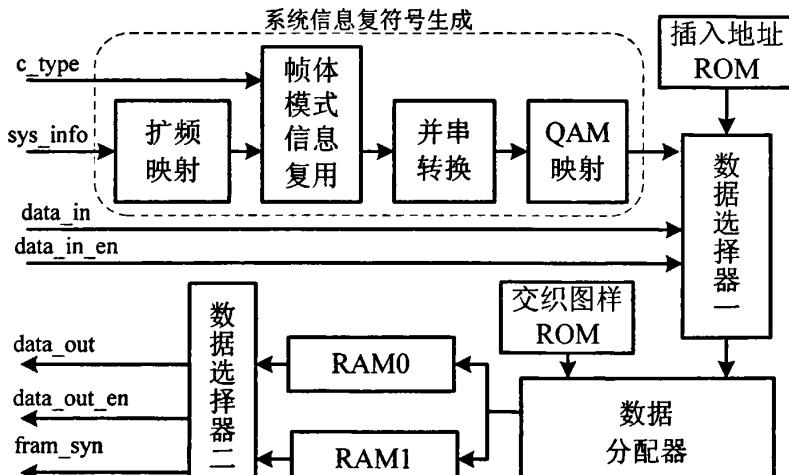


图 3-17 C=3780 模式实现结构图

扩频映射由一个存储国标附录的扩频映射系统信息矢量的 ROM 完成，将输入的系统信息 ‘sys_info’ 作为读取地址。帧体模式信息复用模块根据载波模式信息，生成 36 位符号。符号经过并串转换后，按位进行扩展映射为连续的 36 个系统信息复符号。

对于 C=1 模式，不用进行频域交织，只需将图 3-17 中的数据选择器一改为对

两路数据进行 36+3744 复用后，即可直接输出数据。

对于 C=3780 模式，数据选择器一根据插入地址 ROM 中的数据，将 36 个系统信息符号按地址与输入数据进行复用，输出帧体数据符号。数据分配器，则同理根据交织图样 ROM 中的信息，将 3780 点的帧体，按交织图样地址存储到 RAM0 中。在处理第二个 3780 点帧体数据时，则存储到 RAM1 中，同时，数据选择器将 RAM0 中按交织图样存储的数据按顺序连续读出，并生成帧体起始信号 ‘fram_syn’。通过对 RAM0 和 RAM1 交替存储读取，完成对 3780 点帧体数据的频域交织。

这种处理方式完成了数据的频域交织，同时，由于最后读取 RAM 的方式，能够确定帧起始位置并且保证 3780 点帧体数据符号连续输出，便于下一个模块，帧体数据处理模块的处理。模块接口定义如下：

clk: 模块工作时钟；（图中省略）

reset: 系统复位信号，低电平有效；（图中省略）

c_type: 模式选取信号，表明 C=1 和 C=3780 模式的选择；

sys_info: 系统信息信号，6bits，表示不同的系统参数选择；

data_in: 输入数据，一个符号位宽（32bits），与 clk 同步；

data_in_en: 输入数据有效信号，高电平有效，与 clk 同步；

data_out: 交织完成后的输出数据，一个符号位宽（32bits），与 clk 同步；

data_out_en: 输出数据有效信号，高电平有效，与 clk 同步。

fram_syn: 帧体起始信号，高电平有效，持续一个时钟，标识 3780 点数据输出开始。

3.3.4 帧体形成模块的 QuartusII 仿真验证

帧体形成（BODY_generation）模块，由上述的符号星座映射、符号交织、帧体成型模块级联而成，当输入随机的 127bits 信息，模块经 QuartusII 仿真，输出波形如图 3-18：

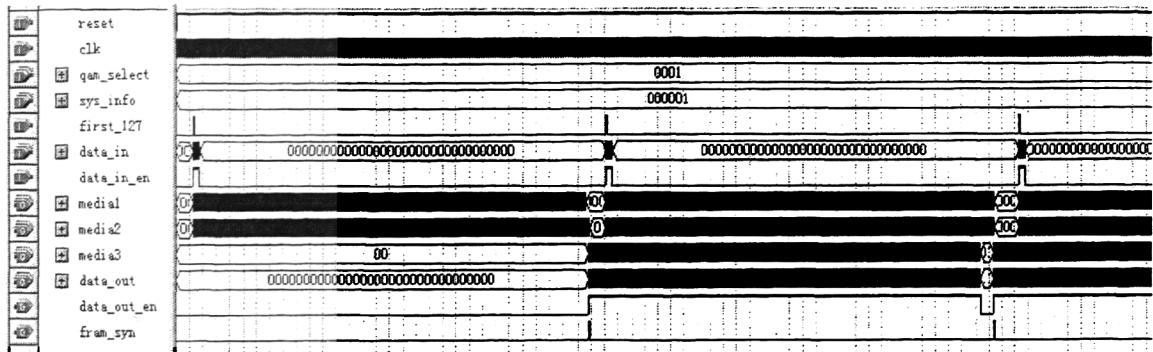


图 3-18 帧体形成模块仿真输出波形

本图为经过模块结构优化后的仿真输出图(优化方式见 3.7.1 节)。图中输入为 59*127bits 随机数据，系统采用 QPSK 星座映射模式。从图中可以看到 ‘medial’ 为将数据块结构改变后的 3744*7bits 的 ROM 地址数据，‘media2’ 为复用 36 符号系统信息后的 3780 点符号数据，‘media3’ 为 3780 点的频域交织后数据，最后输出 ‘data_out’ 为 32 位宽的映射后数据，也就是根据 7bits 的地址数据所读出的 ROM 中的映射数据。

帧体形成 (BODY_generation) 模块 FPGA 实现的资源消耗如图 3-19：

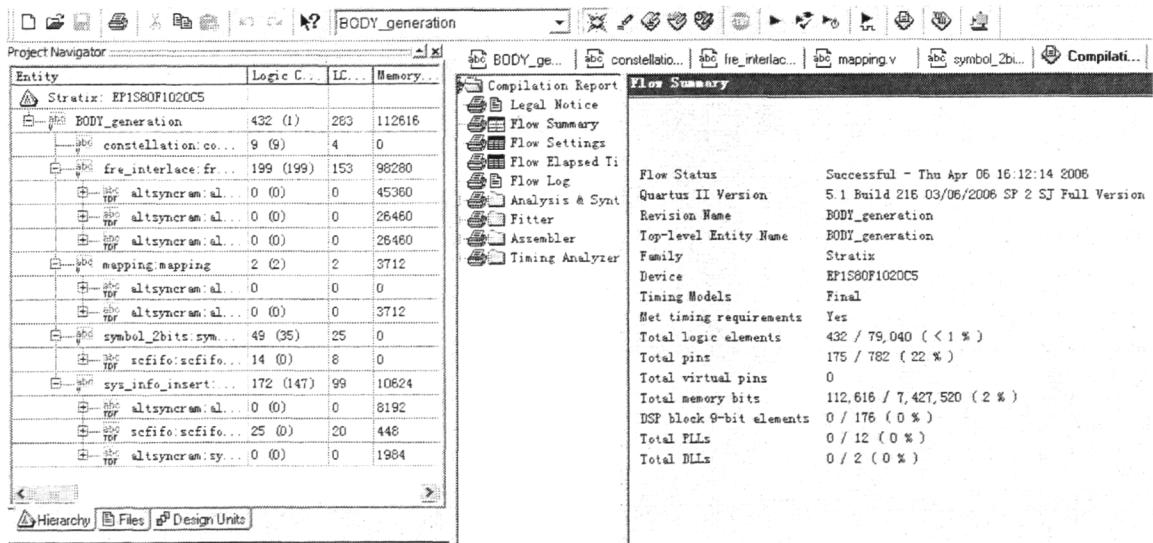


图 3-19 帧体形成模块 FPGA 实现的资源消耗图

3.4 帧体数据处理模块

国标定义了在 C=3780 模式下，需要对帧体数据进行 OFDM 调制，将有效数据调制到 3780 个子载波，相邻的两个子载波间隔为 2kHz。帧体信息符号 $X(k)$ 进行频域交织，得到 $X(n)$ ，然后按下式进行变换得到时域信号：

$$FBody(k) = \frac{1}{\sqrt{C}} \sum_{n=1}^C X(n) e^{j2\pi \frac{k}{C}} \quad k = 0, 1, \dots, 3779 \quad (3-5)$$

可以看出，对数据的 OFDM 调制，采用 3780 点 IFFT 运算来实现。基于 3780 点的特殊性，本文中采用了一种基于存储器的 3780 点 FFT 的 FPGA 实现方法来实现^[10]。根据 FFT 与 IFFT 表达式的对应关系，只需对该模块的输入和输出数据分别取共轭，就可以方便的实现 3780 点的 IFFT 运算。

3.4.1 3780 点 FFT 实现算法

3.4.1.1 3780 点 FFT 算法

由于 3780 不是 2 的幂次方，所以不能采用传统的基 2，基 4 时域或频率抽取设计方法。采用时域下标映射 $n = \langle 60n_1 + n_2 \rangle_{mod 3780}, (0 \leq n_1 \leq 62, 0 \leq n_2 \leq 59)$ 和频域下标映射 $k = \langle k_1 + 63k_2 \rangle_{mod 3780}, (0 \leq k_1 \leq 62, 0 \leq k_2 \leq 59)$ ，可以将一维的 3780 点 DFT 转换为二维 63×60 的带旋转因子的 DFT^[12]，其数学表达式为：

$$X(k_1, k_2) = \sum_{n_2=0}^{59} \sum_{n_1=0}^{62} (x(n_1, n_2) W_{63}^{n_1 k_1}) W_{3780}^{n_2 k_2} W_{60}^{n_2 k_2} \quad (3-6)$$

由式 (3-6) 可以看出：3780 点 DFT 可以先进行 60 次 63 点 DFT，再乘以旋转因子矩阵 $W_{3780}^{n_2 k_2} \left(0 \leq n_2 \leq 59, 0 \leq k_2 \leq 62, W = e^{-j2\pi \frac{n_2 k_2}{3780}} \right)$ ，最后进行 63 次 60 点 DFT 来实现。

根据 PFA 算法（素因子算法）^[12]，一维 63 点和 60 点的 DFT 可以进一步分解为二维 7×9 和三维 $5 \times 3 \times 4$ 不带旋转因子的 DFT：

其中一维 63 点 DFT 采用时域下标映射 $n = \langle 9n_1 + 7n_2 \rangle_{mod 63}, (0 \leq n_1 \leq 6, 0 \leq n_2 \leq 8)$ 和频域下标映射 $k = \langle 36k_1 + 28k_2 \rangle_{mod 63}, (0 \leq k_1 \leq 6, 0 \leq k_2 \leq 8)$ ，其二维 DFT 可表示为：

$$X(k_1, k_2) = \sum_{n_2=0}^8 \sum_{n_1=0}^6 (x(n_1, n_2) W_7^{n_1 k_1}) W_9^{n_2 k_2} \quad (3-7)$$

由式 (3-7) 可以看出：63 点 DFT 可以先进行 9 次 7 点的 DFT，再进行 7 次 9 点的 DFT 来实现；

对于 60 点的 DFT 运算，根据 PFA 算法：一维 60 点 DFT 采用时域下标映射 $n = \langle 4 * \langle 3n_1 + 5n_2 \rangle_{mod 15} + 15n_3 \rangle_{mod 60}, (0 \leq n_1 \leq 4, 0 \leq n_2 \leq 2, 0 \leq n_3 \leq 3)$ 和频域下标映射

$k = \langle 16 * \langle 6k_1 + 10k_2 \rangle_{\text{mod}15} + 45k_3 \rangle_{\text{mod}60}, (0 \leq k_1 \leq 4, 0 \leq k_2 \leq 2, 0 \leq k_3 \leq 3) \rangle$, 其三维DFT可表示为:

$$X(k_1, k_2, k_3) = \sum_{n_3=0}^3 \sum_{n_2=0}^2 \sum_{n_1=0}^4 x(n_1, n_2, n_3) W_5^{n_1 k_1} W_3^{n_2 k_2} W_4^{n_3 k_3} \quad (3-8)$$

由式(3-8)可以看出: 60点DFT可以先进行12次5点的DFT, 再进行20次3点的DFT, 最后进行15次4点的DFT来实现。

综上所述, 3780 点的 FFT 可以由 5 级 (7, 9, 5, 3, 4) DFT 运算以及一个旋转因子矩阵来实现^[9]。为了减少运算量和复杂度, 7、9、5、3、4 点的 DFT 都采用 WFTA 算法^[13]来实现。

3.4.1.2 WFTA 算法介绍

由小 $N (N \leq 16)$ 点循环卷积快速算法可知, 两个 N 点序列 $x(n)$ 与 $h(n)$ 的循环卷积 $y(n)$ 可以通过多项式的积 $Y(Z) = X(Z)H(Z) \bmod P(Z)$ 来实现。其中 $Y(Z)$, $X(Z)$, $H(Z)$ 分别为 $y(n)$, $x(n)$, $h(n)$ 的 z 变换, $P(Z) = z^N - 1$ 。结合中国余数定理 (CRT), $y(n)$ 可以用如下矩阵表示:

$$y = C[Ah \bullet Bx] \quad (3-9)$$

式中 \bullet 表示矩阵与矩阵的元素对应相乘, y , x , h 都是 $N \times 1$ 维的数据向量, A , B 和 C 分别为正变换和反变换矩阵, 其维数和元素值对特定 N 值是确定的。

用向量 $W = [W_N^0, W_N^1, \dots, W_N^{N-1}]^T$, $W_N = e^{-j2\pi/N}$ 代替式 (3-9) 中的向量 h , 则可以得到WFTA计算小 N 点的DFT公式:

$$X = C[AW \bullet Bx] \quad (3-10)$$

式中 x 是输入向量, X 为输出向量, 它们是输入数据 $x(n)$ 及其变换 $X(k)$ 的重新编排。令 $G = AW$, 则有:

$$X = CGBx \quad (3-11)$$

该式是输入向量 x 的线性函数, 也是小 N 点WFTA的标准形式。由于一个 N 点的DFT可以写成 $X = W_N x$ 的形式, W_N 是 $N \times N$ 矩阵, 其元素为 W 因子。因此根据式 (3-11) 有:

$$W_N = CGB \quad (3-12)$$

这样矩阵 C, G, B 可以看作是由 W_N 矩阵分解后的子矩阵。矩阵 C , B 分别叫做前编

排和后编排矩阵，其元素一般取比较简单的数，如 $\pm 1, \pm 2$ 等， G 为对角阵，其元素一般为实数或者纯虚数。以5点DFT为例，其WFTA标准表示式为：

$$X_{wfta} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & 0 \\ 1 & 1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ -1.25 \\ 0.5590 \\ j*0.9511 \\ -j*0.3633 \\ j*1.5388 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix} * x \quad (3-13)$$

3.4.2 3780 点 FFT 的 FPGA 实现

3.4.2.1 3780 点 FFT 的 FPGA 结构与接口定义

3780 点 FFT 模块实现结构如图 3-20 所示

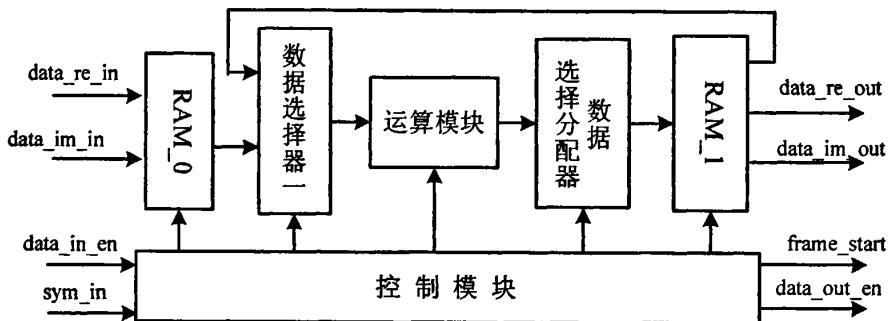


图 3-20 3780 点 FFT FPGA 实现的整体结构

系统工作在双时钟模式，输入为带同步信号 ‘syn_in’ 的 16 位复数信号 ‘data_re_in’ 和 ‘data_im_in’。其包括 2 个读写存储器 RAM_0 和 RAM_1，2 个数据选择模块，一个控制模块和一个运算模块 t。其中 RAM_0 工作在双时钟模式，存储输入的复数数据，每个 ‘syn_in’ 信号到来时启动一次写操作；运算模块主要由 wfta_twid_mult 模块和 wfta_4 模块组成，wfta_twid_mult 完成 7, 9, 5, 3 点 WFTA 运算和乘旋转因子 $W_{3780}^{n_2 k_1}$ 操作，wfta_4 实现 4 点 WFTA 运算；RAM_1 存储每级 WFTA 运算后的中间数据；数据选择器一选择 RAM_0 或者 RAM_1 的数据进入运算模块；数据选择分配器选择 wfta_twid_mult 或者 wfta_4 模块的输出数据写到 RAM_1 存储器中；控制模块产生整个系统的控制信号，其中包括控制 7, 9, 5, 3, 4 点 WFTA 运算的级控制信号 index、RAM_0 与 RAM_1 的读写地址、读写使

能信号以及 FFT 输出数据的起始信号 ‘frame_start’ 和有效信号 ‘data_out_en’。

3780 点 FFT 模块外部接口如图 3-21 所示：

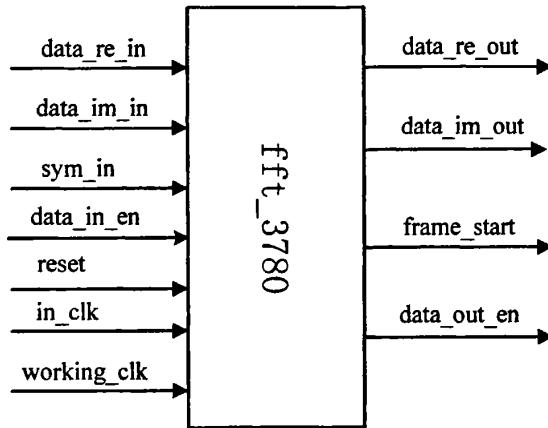


图 3-21 3780 点 FFT 的整体框图

3780 点 FFT 模块接口定义如下：

reset: 系统复位信号，低电平有效，完成系统的初始化和复位操作；

syn_in: 输入有效数据起始信号，高电平有效，持续期为一个输入时钟周期，标识从下一个时钟开始的 3780 个数据为输入有效数据；

data_re_in,data_im_in: 输入复数数据的实部，虚部，宽度为 16 位；

in_clk: 输入数据时钟；

working_clk: 系统工作时钟，也是数据输出时钟；

frame_start: 输出有效数据起始信号，高电平有效，持续期为一个输出时钟周期，标识从下一个时钟开始的 3780 个数据为输出有效数据；

data_out_en: 输出有效数据使能信号，高电平有效；

data_re_out, data_im_out: 输出复数数据的实部，虚部，可以根据系统信噪比要求调整位宽，一般为 16 位。

3. 4. 2. 2 帧体形成模块的 QuartusII 仿真验证

输入 16 位随机复数数据，帧体数据处理（fft_3780）模块经 QuartusII 仿真，其输出波形如图 3-22：

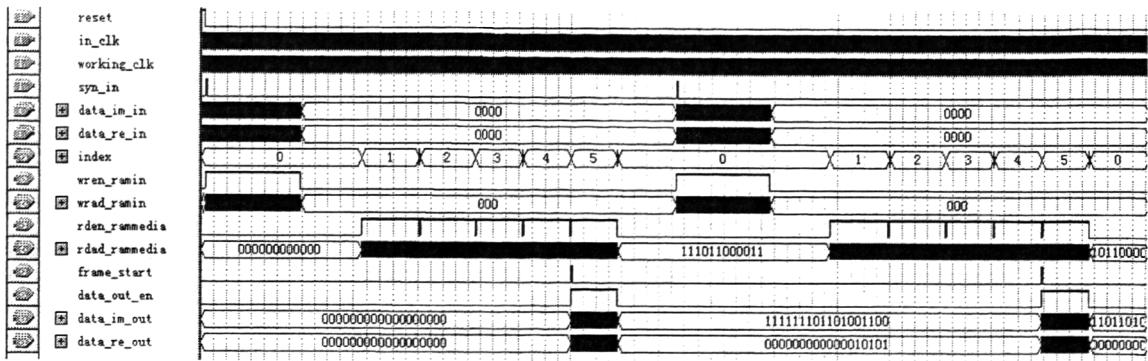


图 3-22 帧体数据处理模块仿真波形

通过波形图,可以清晰的看到本文所采用的基于存储器的 3780 点 FFT 的 FPGA 实现过程,可以清楚的看到,在将 3780 点 FFT 分解为五级 (7, 9, 5, 3, 4) DFT 运算时,对存储器读写的数据转换过程。

帧体数据处理的实现资源消耗如图 3-23 所示:

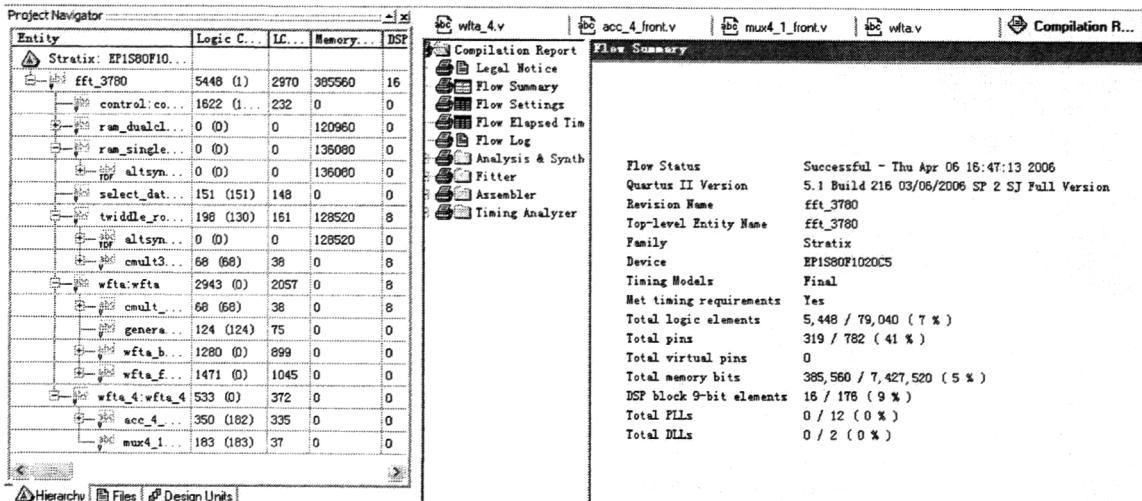


图 3-23 fft_3780 模块 FPGA 实现资源消耗图

3.4.2.3 运算模块的 FPGA 实现结构^[10]

computing unit 模块是整个 3780 点 FFT 系统的关键,其包括 wfta_twiddle_mult 和 wfta_4 两模块。wfta_twiddle_mult 模块完成 7、9、5、3 点的 WFTA 运算及 index=1 时乘 3780 点旋转因子运算; wfta_4 模块完成 4 点 WFTA 运算。本节以 5 点 WFTA 为例,其 FPGA 具体实现框图如下:

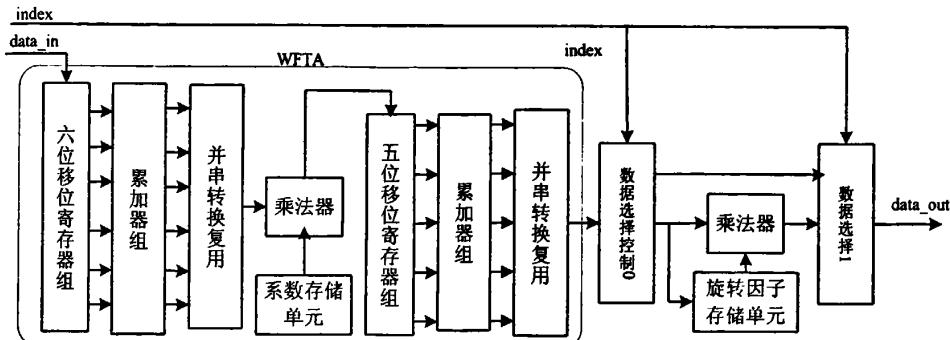


图 3-24 运算模块结构

WFTA_5 运算操作可以分为 C、G、B 三个阶段^[12]，其分别对应 5 点 WFTA 标准表达式中的三次矩阵运算。从存储器读出的数据 $data_in$ 送入 6 位移位寄存器，产生 6 路 time-skewed 信号分别进入六个累加器构成的累加器组，再由一个并串转换复用模块，把累加器输出的 6 路 time-skewed 信号变成一路串行输出，完成 B 阶段的 Bx 运算。B 阶段输出的数据再与系数存储单元的 6 个系数因子相乘，完成 G 阶段的 GBx 运算。C 阶段操作和 I 阶段类似。由式 (3-13) 可以看出，当进行 B 阶段运算时，输入 5 个数据，经 Bx 矩阵运算后，输出为 6 个数据，如果输入数据 $data_in$ 为连续输入，则在 FPGA 具体实现时， Bx 运算后的数据在数据选择过程中会丢失。为了满足 FPGA 设计的时序要求，本方案通过前端存储器的读操作来控制数据 $data_in$ 的输入，即每 6 个时钟周期从存储器中读出 5 个数据。由于 5 点 WFTA 标准表达式中 C、B 矩阵元素只有 0、1、或者-1，则 Bx ， $Cz(z = GBx)$ 矩阵相乘实际上是一个对输入 x, z 的累加运算。所以 WFTA_5 模块中所有累加器均采用 Quartus 器件库的自带累加器，通过控制信号，实现对应的矩阵运算。

7 点、9 点以及 3 点 WFTA 结构及输入数据控制方式与 5 点基本类似，4 点 WFTA 结构与 5 点 WFTA 的 B 阶段类似。为了节省系统的资源消耗，对 7、9、5、3 点的 WFTA 运算，采用累加器和乘法器复用。即当进行 9 点 WFTA 的 B 阶段运算时，11 个累加器全部进行运算；7 点时，其中的 8 个累加器进行运算；5 点时，其中的 6 个累加器进行运算；3 点时，其中的 3 个累加器进行运算。运算后的数据经数据选择器后送入同一个乘法器进行系数因子相乘操作。C 阶段累加器复用和 I 阶段类似。分析表明，采用这样的结构，比单独采用累加器和乘法器进行 WFTA 运算可以节约约 50% 的逻辑单元。

旋转因子处理模块主要实现式(3-6)中旋转因子 $W_{3780}^{n_2 k_1}$ ($0 \leq n_2 \leq 59, 0 \leq k_1 \leq 62$) 的相乘。当级控制信号 ‘index’ 值为 0,2,3 时，WFTA 模块的输出数据直接送到如图

3-4-4 中运算模块输出端, 当 $\text{index}=1$ 时, WFTA 模块的输出数据使能信号启动读 ROM 操作, 与经延迟单元后的输出数据相乘, 完成 3780 个旋转因子乘法运算。ROM 存储由 $W_{3780}^{n_2 k_1}$ 产生的 3780 个旋转因子。为了简化运算模块设计, 所有数据操作均采用定点处理方式, 复数乘法器采用 Quartus 自带的 DSP block 来实现。

3.5 信号帧成型模块

由 2.1.3 节介绍, 可看到为适应不同应用, 国标定义了三种可选帧头模式, 见图 3-25。三种帧头模式所对应的信号帧的帧体长度和超帧的长度保持不变。

帧头 (420 个符号) (55.6μs)	帧体 (含系统信息和数据) (3780 个符号) (500μs)
a) 信号帧结构 1	
帧头 (595 个符号) (78.7μs)	帧体 (含系统信息和数据) (3780 个符号) (500μs)
b) 信号帧结构 2	
帧头 (945 个符号) (125μs)	帧体 (含系统信息和数据) (3780 个符号) (500μs)
c) 信号帧结构 3	

图 3-25 信号帧结构

对于结构 1, 每 225 个信号帧组成一个超帧 ($225 \times 4200 \times 1/7.56\mu\text{s} = 125\text{ms}$); 对于结构 2, 每 216 个信号帧组成一个超帧 ($216 \times 4375 \times 1/7.56\mu\text{s} = 125\text{ms}$); 对于结构 3, 每 200 个信号帧组成一个超帧 ($200 \times 4725 \times 1/7.56\mu\text{s} = 125\text{ms}$) [1]。

信号成型模块的实现即为产生需要的帧头序列, 然后与输入的帧体信息复用。三种帧头采用 PN 序列, 可由不同阶数的 Fibonacci 型线性反馈移位寄存器实现。对于需要在每个超帧中各信号帧的帧头采用不同相位的 PN 信号作为信号帧识别符时, 则需要通过控制模块改变序列产生器的初始相位。信号成型模块实现结构如图 3-26 所示:

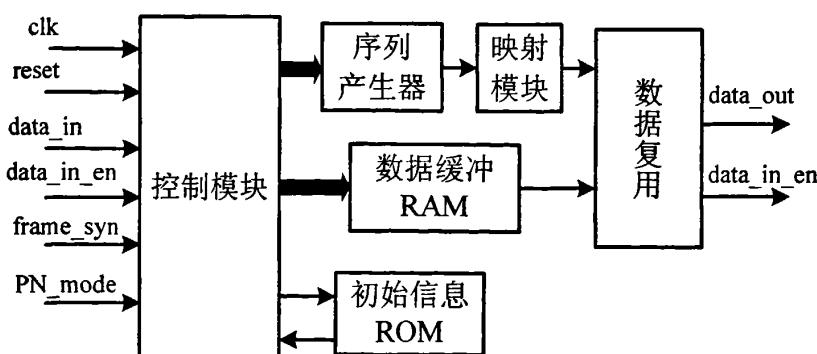


图 3-26 信号成型模块实现结构图

帧头模式 1 采用的 PN 序列定义为循环扩展的 8 阶序列。一个超帧包含 225 个信号帧，每个信号帧采用不同相位的 PN 序列进行标识时，需要 $8\text{bit} \times 225$ 的 ROM 空间；帧头模式 2 采用的 PN 序列定义为循环扩展的 10 阶序列，在一个超帧中共有 216 个信号帧，每个超帧中各信号帧的帧头采用相同的 PN 序列，需要一个 10bits 的初始信息存储空间即可；帧头模式 3 采用的 PN 序列定义为循环扩展的 9 阶 m 序列。一个超帧包含 200 个信号帧，每个信号帧采用不同相位的 PN 序列进行标识时，需要 $9\text{bits} \times 200$ 的 ROM 空间存储初始信息。

控制模块根据 ‘PN_mode’ 信号判断选取的 PN 序列模式。然后根据 ‘frame_first’ 信号对输入帧体进行计数，通过计数器生成地址信息，选取读取初始信息 ROM 中相对应的初始信息，对序列产生器进行初始复位。再利用计数器控制序列产生器依次产生所需要的 PN 序列。同时，将输入的有效数据，存入 RAM 进行缓冲。

PN 序列在复用到信号帧时需要经“0”到+1 值及“1”到-1 值的映射变换为非归零的二进制符号。同时国标系统定义要求不同模式下帧头信号的平均功率需要与帧体信号的平均功率相同或者是其两倍。故在序列产生模块完成一个 PN 序列后，需要经过一个映射模块，将 PN 序列的数据按位扩展为合适的 32bits 符号（与信息数据相同），才生成实际意义的帧头符号。在数据输出后，控制模块控制缓冲 RAM 输出帧体数据。完成将帧头符号与帧体有效数据符号复用，即实现信号帧成型。模块接口定义如下：

clk: 模块工作时钟；

reset: 系统复位信号，低电平有效；

PN_mode: 帧头模式选取信号，根据此信号判断生成一种 PN 序列之；

data_in: 输入数据，一个符号位宽（32bits），与 clk 同步；

data_in_en: 输入数据有效信号，高电平有效，与 clk 同步；

fram_syn: 输入信号，标识一个帧体的起始位，高电平有效，持续一个时钟，
此时钟开始，一个 3780 点的帧体开始输入；

data_out: 帧体成型后的输出数据，一个符号位宽（32bits），与 clk 同步；

data_out_en: 输出数据有效信号，高电平有效，与 clk 同步。

信号帧成型模块经 QuartusII 仿真，输出波形如图 3-27：

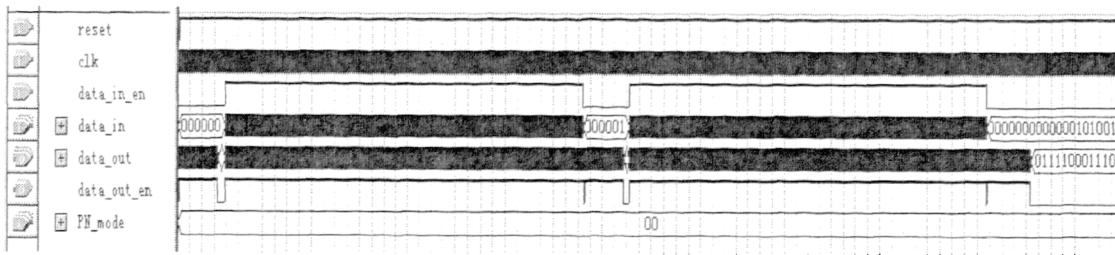


图 3-27 信号帧成型模块仿真波形

如图，此波形为系统采用图 3-25 中所示结构 1 的帧体结构模式，即帧头为 420 点 PN 序列符号的结构。可以看到，本文中此模块的设计利用前面数据处理模块的数据延迟时间，在有效数据输入信号帧成型模块之前，将 PN 序列输出。观察 ‘data_out_en’ 信号，可以看到其中短的 420 点符号为帧头，与其后部的 3780 点符号构成一个完整的信号帧。

信号帧成型（frame_generation）模块 FPGA 实现的实际资源消耗如图 3-28 所示：

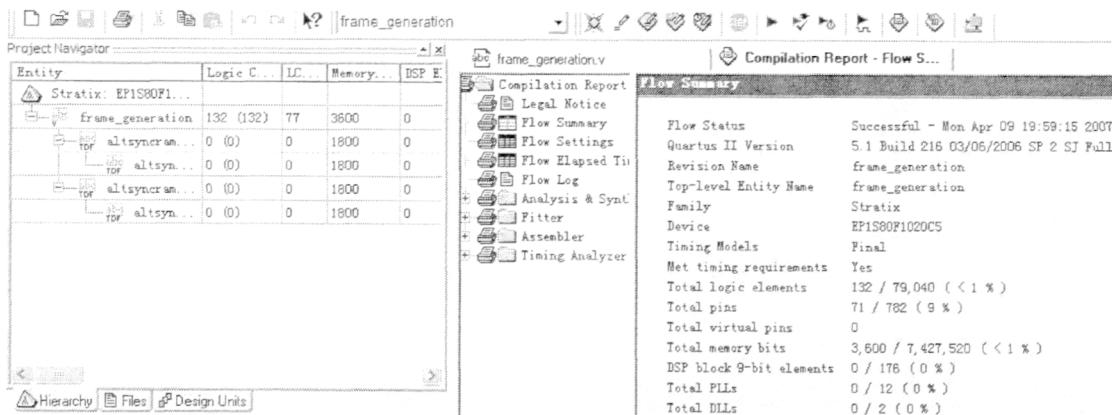


图 3-28 信号帧成型模块 FPGA 实现的实际资源消耗图

3.6 基带后处理模块

数字信号在传输过程中受到叠加干扰与噪声，从而出现波形失真。瑞典科学家哈利·奈奎斯特在 1928 年为解决电报传输问题提出了数字波形在无噪声线性信道上传输时的无失真条件，称为奈奎斯特准则。理想的奈奎斯特滤波器在实际当中只能用来近似，其中常用的是升余弦成形滤波器（具有滚降系数 α ），它的特点就是在取样时刻具有无码间串扰特性，所实现的频谱效率要比理论最高效率下降一个滚降系数 α 。 α 越小，频谱效率就越高，但 α 过小时，升余弦滚降滤波器的设

计和实现比较困难，而且当传输过程中发生线性失真时产生的符号间干扰也比较严重。具体应用时，在发送端和接收端分别放一个平方根升余弦滤波器，其合成传输函数就是升余弦滤波器。

国标系统定义基带后处理（成形滤波）采用平方根升余弦（Square Root Raised Cosine, SRRC）滤波器进行基带脉冲成形。SRRC 滤波器的滚降系数 α 为 0.05。平方根升余弦滤波器频率响应表达式如下式所示^[1]:

$$H(f) = \begin{cases} 1 & |f| \leq (1 - \alpha) / 2T_s \\ \left\{ \frac{1}{2} + \frac{1}{2} \cos \left(\frac{\pi[(2T_s)|f|] - 1 + \alpha}{2\alpha} \right) \right\}^{\frac{1}{2}} & (1 - \alpha) / 2T_s < |f| \leq (1 + \alpha) / 2T_s \\ 0 & |f| > (1 + \alpha) / 2T_s \end{cases} \quad (3-14)$$

其中 T_s 为输入信号的符号周期 ($1/7.56 \mu s$)， α 为平方根升余弦滤波器滚降系数。

本节将介绍在本设计过程中采用的四倍插值 169 阶平方根升余弦滤波器的原理^[14]，以及 FPGA 的实现。本设计采用的滤波器参数性能见附录三。

3.6.1 169 阶平方根升余弦滤波器原理介绍

3.6.1.1 常规 FIR 滤波器原理结构

对于 FIR 数字滤波器，设 $x(n)$ 为 N 阶滤波器输入数据， $y(n)$ 为其输出，则 $y(n)$ 可以表示为：

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \quad (3-15)$$

根据公式 (3-15)，对于 FIR 滤波器的实现，利用加法器，乘法器和移位寄存器即可实现输入数据的滤波操作。直接实现结构如图 3-29 所示

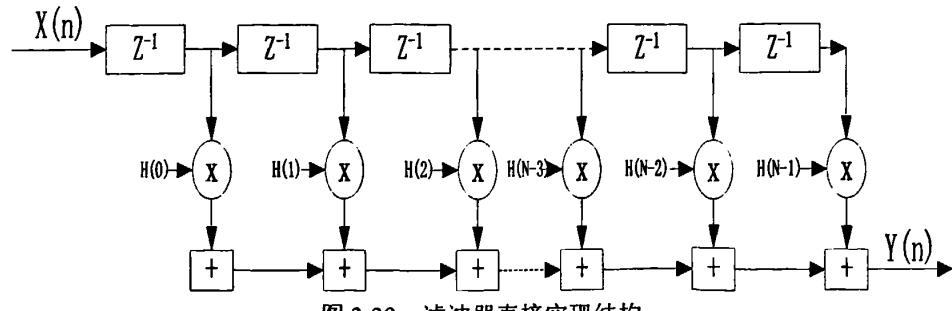


图 3-29 滤波器直接实现结构

但是在实际应用中，由于资源的限制，当滤波器阶数比较高时，采用这样的结构很难在 FPGA 中实现。

利用线性相位 FIR 滤波器系数对称特点，式 (3-15) 可表示为：

$$y(n) = \sum_{i=0}^{(N-1)/2} (x(i) + x(N-1-i))h(i) \quad (3-16)$$

即，通过对相同系数数据通过先叠加，后乘系数的方式实现，其实现框图如图 3-30：

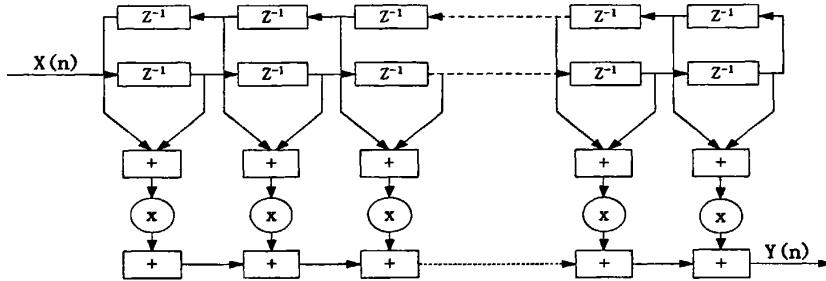


图 3-30 利用滤波器线性相位特点的实现结构

直接对比可发现该结构相对与图 3-29 的结构减少几乎一半的乘法器，因此在 FPGA 实现时约可以节约一半的逻辑资源。但是当滤波器阶数比较高时，如图 3-30 的实现结构也要消耗大量的逻辑资源。因此可以用简单的移位操作替换乘法器单元来降低器件资源的消耗，可以采用比特级流水线等技术来提高滤波器的运算速度^[15]。在本系统方案中，系统的工作时钟不到 100MHz，因此采用一种能够满足时钟速度并尽量降低资源消耗的滤波器是最恰当的，本文采用了一种新的均方根升余弦滚降滤波器设计方案。

3.6.1.2 四倍插值 169 阶平方根升余弦滤波器的原理

对于 FPGA 设计，要求在一定时序要求下，尽可能的利用较少的资源完成功能的实现。这里，利用四倍插值 169 阶平方根升余弦滤波器的一些特点，对其硬件结构进行简化，从而减少硬件资源的消耗：

1) 利用插值输入数据结构特点进行滤波器结构简化

对于 N 阶滤波器，设 $x(n)$ 为输入数据， $y(n)$ 为其输出， $y(n)$ 可以表示为如式 3-15： $y(n) = \sum_{i=0}^{N-1} h(i)x(n-i)$ 。

在本发端设计中，输入 $x(n)$ 为进行 4 倍插值后的数据。即对于 $x(n)$ ，对应于所有 $(n-i) \bmod 4 = 1, 2, 3$ ， $x(n)$ 均为 0，即输入结构如 $\{D_0, 0, 0, 0, D_1, 0, 0, 0, D_2, 0, 0, 0, \dots\}$ 的

形式，则上式可以分解为：

$$y(4n) = \sum_{i=0}^{\text{int}(N/4)} h(4i)x(4n-4i) \quad (3-17)$$

$$y(4n+1) = \sum_{i=0}^{\text{int}(N/4)} h(4i+1)x(4n-4i) \quad (3-18)$$

$$y(4n+2) = \sum_{i=0}^{\text{int}(N/4)} h(4i+2)x(4n-4i) \quad (3-19)$$

$$y(4n+3) = \sum_{i=0}^{\text{int}(N/4)} h(4i+3)x(4n-4i) \quad \text{其中 } n = 0 \sim \text{int}(N/4) \quad (3-20)$$

从上述四个公式可以看出，对于计算输出数据点 $4n, 4n+1, 4n+2, 4n+3$ ($n = 0 \sim \text{int}(N/4)$) 的时，输入值 $x(n)$ 保持不变，仅仅是在计算不同点时对应的系数 $h(n)$ 发生了变化。则对于本设计中的四倍插值 169 阶平方根升余弦滤波器，其输入输出关系公式可由式 3-21 表示：

$$y(4n+k) = \sum_{i=0}^{\text{int}(N/4)} h(4i+k)x(4n-4i), k = 0, 1, 2, 3 \quad (3-21)$$

可以看出，相对于滤波器直接实现结构公式 (3-15)，采用式 (3-21) 的滤波器可以节约 75% 左右的硬件资源。

在 3.6.1.1 节中，提到了利用线性相位滤波器系数具有的对称性，可进一步节省资源。因此，在采用式 (3-21) 的模式进行 4 倍插值滤波器设计时，可以根据系数对称特性进行进一步简化硬件结构。

通过分析对比，可以看到对于式 (3-17) — (3-20) 中的系数，满足一下对应关系：式 (3-17) 中前 21 个系数和后 21 个系数关于第 22 个系数对称；式 (3-18) 中前 21 个系数和式 (3-20) 中后 21 个系数，式 (3-19) 中后 21 个系数和式 (3-20) 中前 21 个系数对称；式 (3-19) 中前 21 个系数和后 21 个系数完全对称。

对于上述公式中相同的系数部分，在硬件实现中则意味着可以利用同一个乘法器完成系数乘法运算，从而合理的复用硬件资源，减少硬件资源的消耗。

通过对相同系数部分的复用，本设计中采用的 4 倍插值 169 阶滤波器还可以进一步变换和简化为：

$$y(4n) = \sum_{i=0}^{\text{int}(N/8)-1} h(4i)(x(4n-4i) + x(168 - (4n-4i))) + h(84)x(84) \quad (3-22)$$

$$y(4n+1) + y(4n+3) = \sum_{i=0}^{\text{int}(N/8)-1} (h(4i+1) + h(168 - (4i+3))) (x(4n-4i) + x(168 - (4n-4i))) \quad (3-23)$$

$$y(4n+2) = \sum_{i=0}^{\text{int}(N/8)-1} h(4i+2) (x(4n-4i) + x(168 - (4n-4i))) \quad (3-24)$$

$$y(4n+1) - y(4n+3) = \sum_{i=0}^{\text{int}(N/8)-1} (h(4i+1) - h(168 - (4i+3))) (x(4n-4i) - x(168 - (4n-4i))) \quad (3-25)$$

其中 $n = 0 \sim \text{int}(N/4)$ 。

$$\begin{aligned} \text{令 } z(4n+1) &= y(4n+1) + y(4n+3), z(4n+3) = y(4n+1) - y(4n+3), \text{ 则有} \\ y(4n+1) &= \frac{(z(4n+1) + z(4n+3))}{2}, y(4n+3) = \frac{(z(4n+1) - z(4n+3))}{2} \end{aligned} \quad (3-26)$$

分析可得，4 倍插值 169 阶滤波器可以统一表示为：

$$z(4n+k) = \sum_{i=0}^{\text{int}(N/8)-1} h'(4i+k)x'(4n-4i), k = 0, 1, 2, 3 \quad (3-27)$$

其中 h' , x' 分别对应不同 k 值时式 (3-22) — (3-25) 中对应的 h , x 的表达式。

从式 (3-27) 可以看出，利用插值输入数据结构特点以及线性相位滤波器系数的对称性。插值 169 阶滤波器只需要进行 22 次乘法运算，比采用传统的系数对称设计方法所需要的 85 次乘法运算量减少了 4 倍运算量。

2) 乘法器的硬件实现设计

对于 FPGA 设计，运算过程中的数据需要以定点方式处理。而对于滤波器的系数 $h(n)$ ，则需要将其由浮点表示方式转化为定点数据。

本文采用的满足系统要求的滚降滤波器阶数为 169 阶，同时根据线性相位特性可知滤波器共有 85 个系数。在设计中，根据滤波器系数值绝对值大小，以及系统数据处理需要，把系数量化设定为 16 位。则对于浮点数据 h ，其转化后的定点量化数据 $h_p = \text{round}(h * 2^{15})$ 。对式 (3-27) 中 h' 定点量化后的系数 h'_p 进行二进制自然编码，即把所有 h'_p 表示为： $h'_p = a_0 * 2^0 + a_1 * 2^1 + \dots + a_{M-1} * 2^{M-1}$ 或 $h'_p = -(a_0 * 2^0 + a_1 * 2^1 + \dots + a_{M-1} * 2^{M-1})$ 的形式，其中 $a_m = 0, 1 (0 \leq m \leq M-1)$ ， M 为用二进制表示滤波器系数时的位数。如 5, -7 就表示为：

$$5 = 2^0 + 2^2;$$

$$-7 = -(2^0 + 2^1 + 2^2); \quad (3-28)$$

则式 (3-6-14) 可以表示为:

$$z(4n+k) = \sum_{i=0}^{\text{int}(N/8)-1} x'(4n-4i)(a_{0i} * 2^0 + a_{1i} * 2^1 + \dots a_{(M-1)i} * 2^{M-1})$$

或者

$$z(4n+k) = \sum_{i=0}^{\text{int}(N/8)-1} -x'(4n-4i)(a_{0i} * 2^0 + a_{1i} * 2^1 + \dots a_{(M-1)i} * 2^{M-1}) \quad (3-29)$$

其中 $a_{mi} = 0,1 (0 \leq mi \leq M-1)$, 其代表第 i 个系数在二进制自然编码中的 2^m 次幂的对应值。进一步推导, 则式 (3-29) 可写为:

$$z(4n+k) = \sum_{m=0}^{M-1} \sum_{i=0}^{\text{int}(N/8)-1} (x'(4n-4i) * a_{mi}) * 2^m$$

或者

$$z(4n+k) = \sum_{m=0}^{M-1} \sum_{i=0}^{\text{int}(N/8)-1} (-x'(4n-4i) * a_{mi}) * 2^m \quad (3-30)$$

由于 a_{mi} 的值只能是 0 或者 1, 乘 2^m 在 FPGA 实现中对应着移位操作。例如对 $D \times 5 = D \times (2^0 + 2^2)$, 在硬件实现中, 即为将 D 向高位移位两个位后加上 D 移位 0 个位的数据。则式 (3-30) 中 $\sum_{i=0}^{\text{int}(N/8)-1} (x'(4n-4i) * a_{mi}) * 2^m$ 或 $\sum_{i=0}^{\text{int}(N/8)-1} (-x'(4n-4i) * a_{mi}) * 2^m$

运算过程即为: 先对 $n (n \leq \text{int}(N/8)-1)$ 个数进行符号控制选择后, 然后将对应 2 的相同次幂的数据进行加法运算, 运算后进行移位处理。最后将所有数据进行加法运算, 即可得到滤波器的输出数据。

3.6.2 滤波器的 FPGA 实现

3.6.2.1 滤波器整体结构

由 3.6.1.2 小节分析, 可以看出, 169 阶平方根升余弦滤波硬件实现过程如下。根据滤波器系数对称特点, 将输入数据, 经过 85 次加法操作完成 $x(i) + x(N-1-i)$ 的对称加操作。对称加操作后的数据根据所乘系数二进制编码中 2 的幂次方的统计, 把所有对应 2 的相同次幂的数据送入并行加法器。对应 2^n 的并行加法器, 相加后的数据进行移位处理, 最后通过并行加法器完成滤波。实现硬件结构图如下:

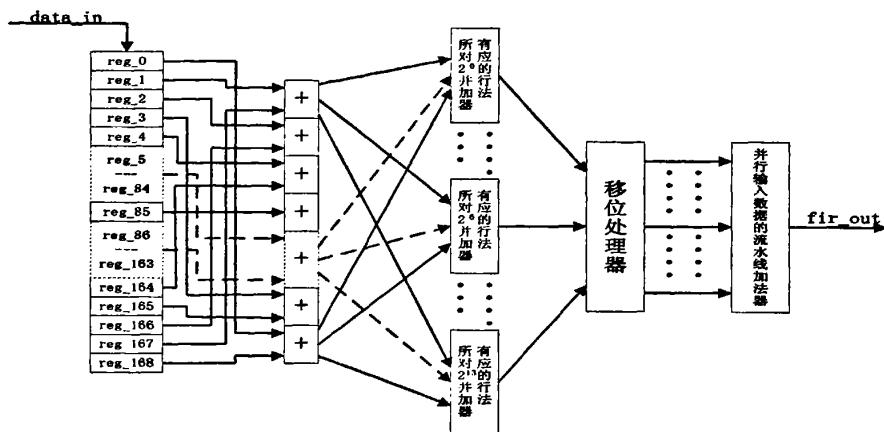


图 3-31 169 阶均方根升余弦滚降滤波器硬件实现结构框图

对于四倍插值的 169 阶平方根升余弦滤波，通过 3.6.1.3 的分析，可以对图 3-6-3 所示结构进行简化，以减少硬件资源的消耗。实现结构系统框如图 3-32 所示：

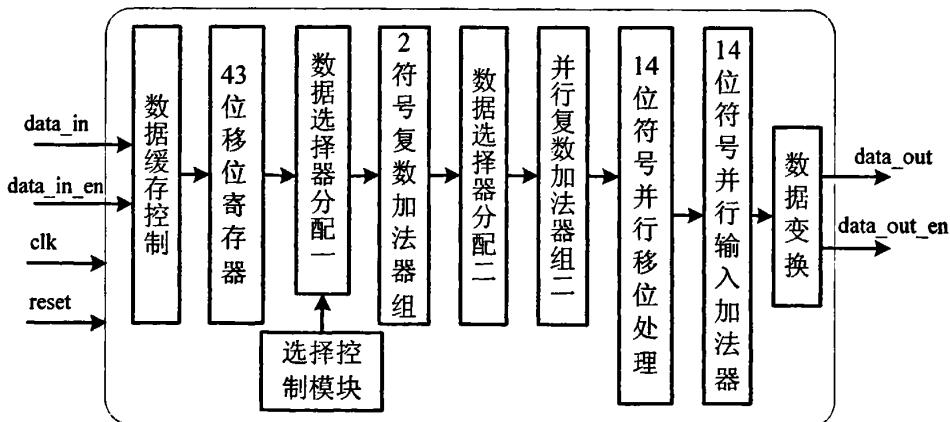


图 3-32 四倍插值 169 阶均方根升余弦滚降滤波器系统框图

由 3.6.1.2 小节分析，通过数据缓存控制模块，对输入有效数据进行缓存，并通过一个计数器对控制数据每四个时钟周期进入移位寄存器。四个时钟即完成四倍插值，在本设计中是为了提供四个时钟周期，便于计算输出数据点 $4n, 4n+1, 4n+2, 4n+3$ ($n = 0 \sim \text{int}(N/4)$) 的值，此时需要的输入值 $x(n)$ 保持不变，仅仅是在计算不同点时对应的系数 $h(n)$ 发生了变化。由于四倍插值原因，所以只需要 43 位符号移位寄存器即可。

数据选择分配器，是利用线性相位滤波器系数具有的对称性，将对应相同系数 $h(n)$ 的数据选择后分配给复数加法器组。这里，在实际计算数据点

$4n, 4n+1, 4n+2, 4n+3$ ($n = 0 \sim \text{int}(N/4)$) 的值时, 移位寄存器组中数据不发生变化。故而通过一个选择控制模块控制选择器的选择。根据公式 (3-22) ~ (3-25), 当 $k = 0, 1, 2, 3$ 时, 分别根据式选择合适的 x 值分配输入到复数加法器。

数据选择分配器二, 功能与一类似, 不过这里完成的是, 根据系数 $h(n)$ 的二进制编码, 把所有对应 2^k 的相同次幂的数送入并行加法器。即实现公式 $\sum_{i=0}^{\text{int}(N/8)-1} (x(4n-4i) * a_m)$ 中 $\sum_{i=0}^{\text{int}(N/8)-1} x(4n-4i)$ 的运算, 为下一步移位做好准备。通过对

$h(n)$ 的二进制编码统计分析, 可知, 从 $2^0 \sim 2^{13}$ 共有 14 个移位方式, 即并行输出 14 个符号位。移位处理模块, 根据不同数据对应的 2^k 的不同幂进行相应的移位处理。

移位后的数据经入并行加法器后, 即可输出公式 (3-27) $z(4n+k) = \sum_{i=0}^{\text{int}(N/8)-1} h(4i+k)x(4n-4i), k = 0, 1, 2, 3$ 的结果。最后数据变换模块根据公式 $y(4n+1) = (z(4n+1) + z(4n+3)) / 2, y(4n+3) = (z(4n+1) - z(4n+3)) / 2$ 对 $z(4n+k)$, 序列进行处理, 即可按时钟输出滤波后数据序列 $y(n)$, 完成滤波功能。

模块接口定义如下:

clk: 模块工作时钟;

reset: 系统复位信号, 低电平有效;

data_in: 输入数据, 一个符号位宽 (32bits), 高 16bits 为实部, 低 16bits 为虚部, 与 **clk** 同步;

data_in_en: 输入数据有效信号, 高电平有效, 与 **clk** 同步;

data_out: 帧体成型后的输出数据, 一个符号位宽 (32bits), 高 16bits 为实部, 低 16bits 为虚部, 与 **clk** 同步;

data_out_en: 输出数据有效信号, 高电平有效, 与 **clk** 同步。

3.6.2.2 基带后处理模块的 QuartusII 仿真验证

当输入为 16 位带符号 4 倍插值后的顺序数据, 基带后处理模块经由 QuartusII 仿真, 模块输出波形如图 3-33 所示:

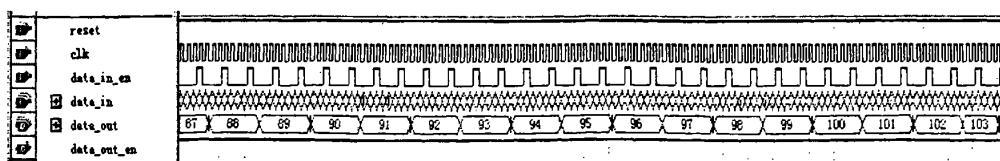


图 3-33 基带后处理模块输出数据波形图

基带后处理模块 (SRRC_filter) 模块的实现资源消耗如图 3-34 所示：

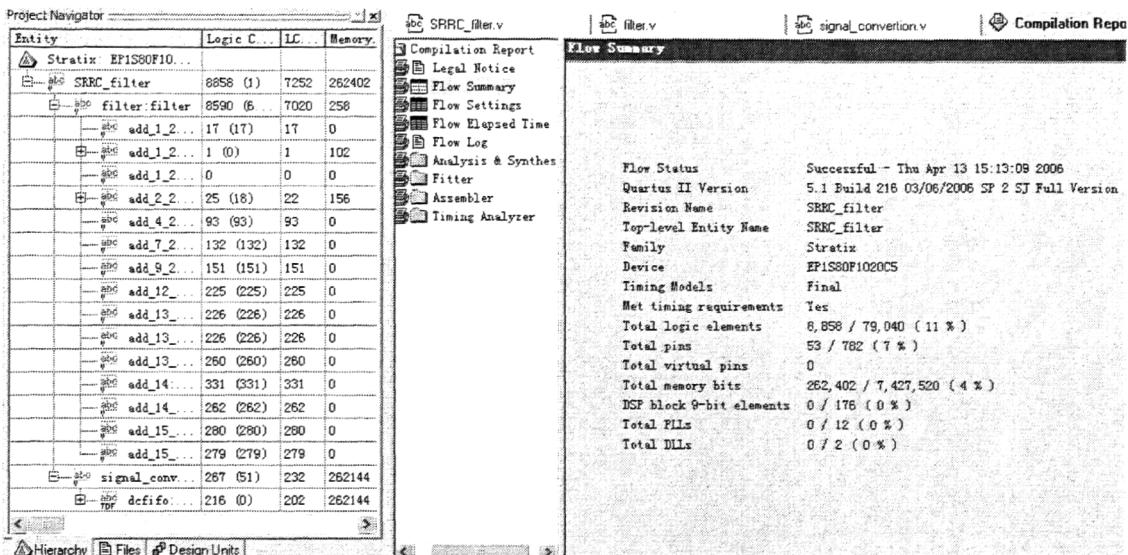


图 3-34 SRRC_filter 模块 FPGA 实现资源消耗图

3.6.2.3 并行复数加法器结构

在本设计中，所用到的并行复数加法器，其并行的位数是由滤波器系数 $h(n)$ 决定的。在用二进制自然码表示 $k = 0,1,2,3$ 时的系数时，对其 2 的不同幂次方进行统计，然后得出所有 2 的不同幂次方的最大个数。这个个数就意味着在对应这个幂次方的并行加法时的并行输入数，即结构图 3-6-4 中并行复数加法器组的最大并行输入数。在本设计中采用的 169 阶均方根升余弦滚降滤波器，将系数量化为 16 位，其对应 $k = 0,1,2,3$ 时 2 的不同幂次方最大个数统计见表 3-1：

表 3-1 $k = 0,1,2,3$ 时 2 的不同幂次方最大个数统计

2 的幂次方	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
最大个数值	13	13	15	13	12	14	15	9	7	4	2	1	1	1	0

由表可得，并行复数加法器组的最大并行输入数为 15。同时从 2 的幂次方个数可以看出最后的并行数据加法器的并行输入为 14。

对于这种并行输入的数据加法处理，为了提高整个滤波器的工作频率，所有并行加法处理模块都采用常见的流水线结构。输入数据每 2 个一组送入第一级进行一次加法运算，所得结果再送入第二级进行下一次 2 个数据的加法运算，以此类推。对 N 个并行输入数据的流水线加法器，其所需流水线级数与输入数据个数的关系为： $N_{class} = \text{int}(\log_2 N) + 1$ 。结构如图 3-35：

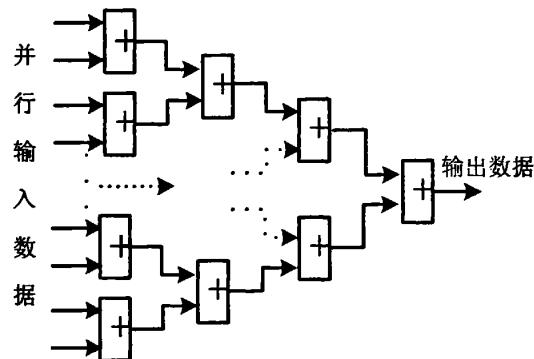


图 3-35 并行输入数据的加法器结构示意图

对于单个的复数加法器，在实现时需要对其正负号选择，以及实部虚部进行分离操作。对于本设计的 32 位数据，运算时将其高 16 位与低 16 位分离并行处理，运算后合并。

3.7 本章小结

本章详细地介绍了基于 FPGA 实现的发射端系统各主要功能模块的具体结构设计，论述了系统中各功能模块的 FPGA 设计和实现，包括设计方案、算法和结构的选取、FPGA 实现、仿真分析等。通过分析验证，可以看出，各功能模块较好的满足了设计需求。

第四章 发端系统整体性能分析和验证

4.1 系统模块级联结构优化调整

分析系统级联的各模块，可以看到，在本设计中，通过符号星座映射模块，将符号数据映射输出定位 32bits 位宽，即采用的存储器 ROM 的深度为 116，位宽为 32。比较可以看到：符号星座映射模块的输出符号为 32bits 位宽，而读取 ROM 内数据时，需要的地址信息符号则只有 7bits 位宽。另外我们可以看到，符号交织模块、系统信息插入模块、帧体频域交织模块，都是基于星座符号的数据处理。即对符号星座映射模块中 ROM 的输出进行数据结构的调整处理。通过比较可以发现，符号交织模块数据处理并不改变一个符号的内部结构，只是在宏观上对数据排列顺序进行调整。同时，在系统信息插入模块，系统信息也需要进行一次 QAM 映射。这里将系统信息的映射数据同样处理成星座映射模块的 ROM 地址结构，然后进行数据插入。则系统信息插入模块、帧体频域交织模块也都是对 ROM 地址数据符号进行顺序上的调整而不改变数据内容。

因此，在本文的设计中将帧体成型模块中数据处理的顺序进行调整，即将符号交织模块、系统信息插入模块、帧体频域交织模块的位置提前到符号星座映射的存储器 ROM 读取之前。结构简图如图 4-1：

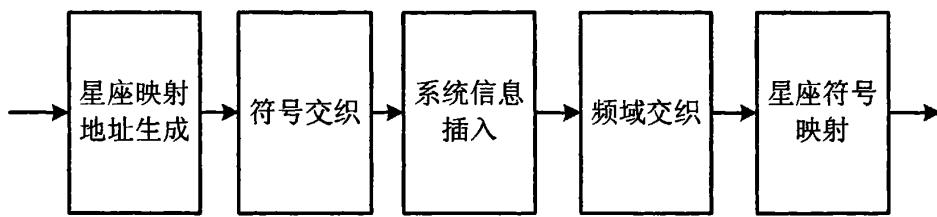


图 4-1 帧体成型模块结构调整后简图

结构改变键入从而可以将这些模块中数据存储器以及片内逻辑寄存器的每一个符号的位宽从 32bits 降低为 7bits。通过这种模块结构顺序上的处理，所需要的硬件资源约为原结构设计的四分之一，大大的节省了硬件资源。

在帧体数据处理 (IFFT_3780) 模块，为了在数据处理过程中，不丢失输入数据，需要一个 3780*32 的缓存 RAM。通过分析，可以看出，在图 3-7-1 中的频域

交织模块时，通过对两个不同 RAM 分别读写完成频域交织的同时，也保证了 3780 点帧体的连续输出。并且保证了两个连续帧体之间的空间。因此，在合适的时钟设定下，可以保证在帧体数据处理模块进行数据运算到数据输出过程中，没有数据输入。故在系统级联中，将帧体数据处理模块前端的缓存 RAM 取消，从而节约硬件资源。

在 LDPC 编码部分，可以看到，其中的独立运算模块被调用 35 次，而在 0.4 和 0.6 码率模式下，只需要使用 11 跟 23 个运算模块。大量的并行运算模块，占用了巨大的硬件资源，且利用率不高。观察三种码率生成矩阵的校验位生成部分，其每列上子矩阵个数分别为 35, 23, 11，都近似为 12 的倍数。因此，在设计中可以选择调用 12 个并行运算模块。通过对运算模块的复用，即三种码率对该结构分别重复使用 3, 2, 1 次即可完成全部校验位的生成。从而节约硬件资源。

通过对表 4-1 的 LDPC 码字参数表进行分析可得，0.4 和 0.6 码率复位后都需要 9156 个时钟才能完成第一个码字的编码，此后每隔 9144 个时钟输出一个码字。0.8 码率复位后需要 6107 个时钟完成第一个码字的编码，此后每隔 6096 个时钟输出一个码字。再对表 2-1 三种码率系统净荷数据的分析，三种码率系统净荷数据率最高分别为 16.243Mbps, 24.365Mbps, 32.486Mbps。故要求系统时钟最低为 48.73MHz 即可满足运算要求。

表 4-1 LDPC 码字生成矩阵参数

码率	k	c	b
0.4	24	35	127
0.6	36	23	127
0.8	48	11	127

通过调整控制信号，控制输入数据的反馈运算，复用运算模块。大大降低了 LDPC 模块的资源消耗。其资源消耗如图 4-2 所示：

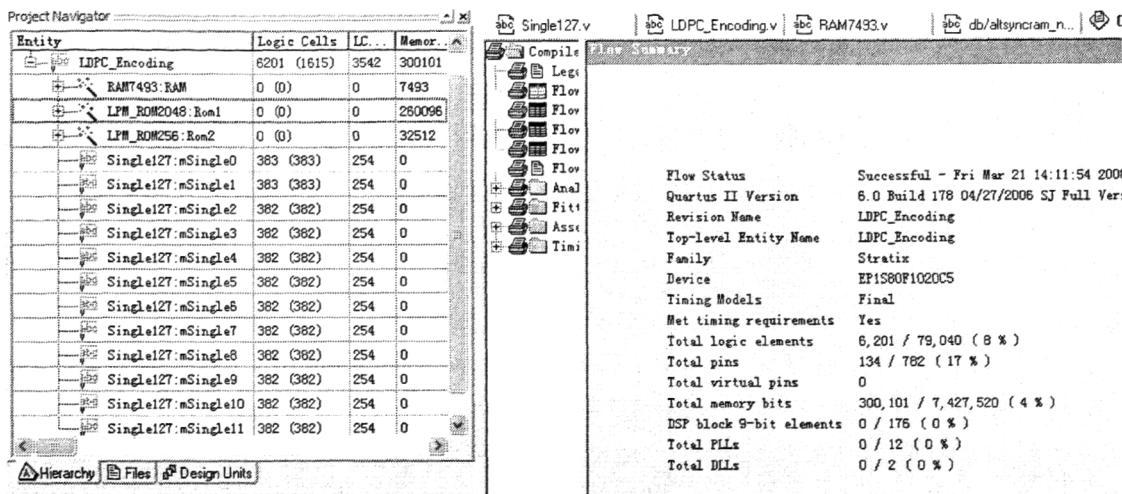


图 3-11 LDPC 模块 FPGA 实现资源消耗图

与图 3-11 对比可见，通过复用技术，节省了一半的逻辑单元消耗。

4.2 系统时钟的设定

在整个发端系统硬件设计中，采用 36 MHz 晶振作为系统时钟源。FPGA 内部采用锁相环 PLL 锁定时钟提供工作时钟。

其中，根据国标要求的 7.56M 带宽，以及本设计采用的四倍插值的数字滤波器设计，需要锁定一个 21/25 输出 30.24MHz 作为基带采用时钟输出给 AD9857。

对于内部处理部分，通过分析如表一所示的标准支持的净荷数据率，以及考虑到数据处理所需要的时钟，本设计采用个不同的时钟。鉴于 LDPC 编码部分模块最高时钟频率为 60.24MHz，而帧体数据处理以及基带后处理模块需要较高的时钟周期来完成运算。

本设计将系统发端模块分为两个部分：从能量扩散模块到系统信息插入模块为一个低时钟模块，频域交织到基带信号处理为一个高时钟模块。通过 PLL 锁定的两个时钟为：21/15 输出的 50.4 MHz 和 21/9 输出 84MHz。两部分之间通过一个异步 FIFO 即可完成时钟的变换。同时可以看到，低时钟在前，高时钟在后，故异步 FIFO 的深度为 2 即可，可以看出这种处理方式消耗硬件资源极小。并且这一时钟设定可以完全满足数据处理速度与硬件模块设计的时钟要求。

4.3 发端系统的性能分析和验证

通过随机序列产生器生成伪随机信号，将随机信号数据输入系统。对于输入随机数据的系统模块经 QuartusII 仿真后，输出波形如图 4-3：

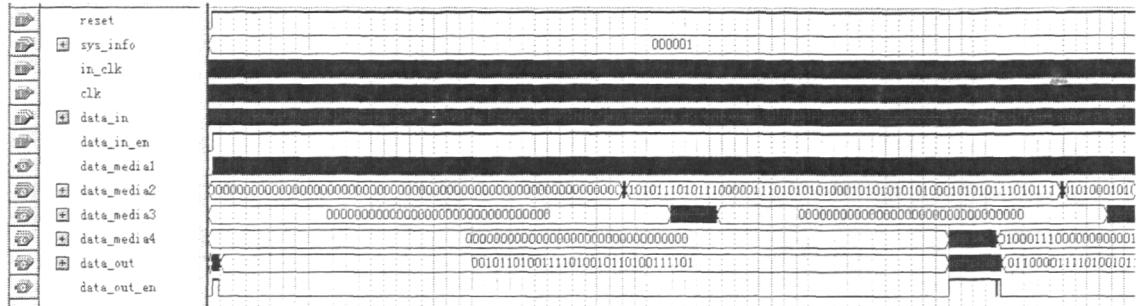


图 4-3 系统仿真输出波形

此图为系统在 QPSK 映射，0.4FEC 编码模式下的数据输出结构图。这里截取了一个完整信号帧的生成过程。从图中，可以清楚的看到数据在系统中的结构变化，‘data_media2’ 为 59×127 bits 的 FEC 编码输出。‘data_media3’ 为 IFFT 的输入数据，‘data_media4’ 为 IFFT 输出数据结构，输出数据 ‘data_out’ 为添加帧头信息后的完整信号帧。此时数据尚未经过时钟变换的数据，便于直接观察数据结构的变化。输出数据在进行 D/A 变化前，要经过一个异步 FIFO 将数据时钟降为国标系统要求的 30.24MHz。

将输出数据（专用器件 AD9857 的输入数据）提取后导入 Matlab，进行分析验证。可以看到随机数据经过系统后形成的基带信号的频谱结构如图 4-4：

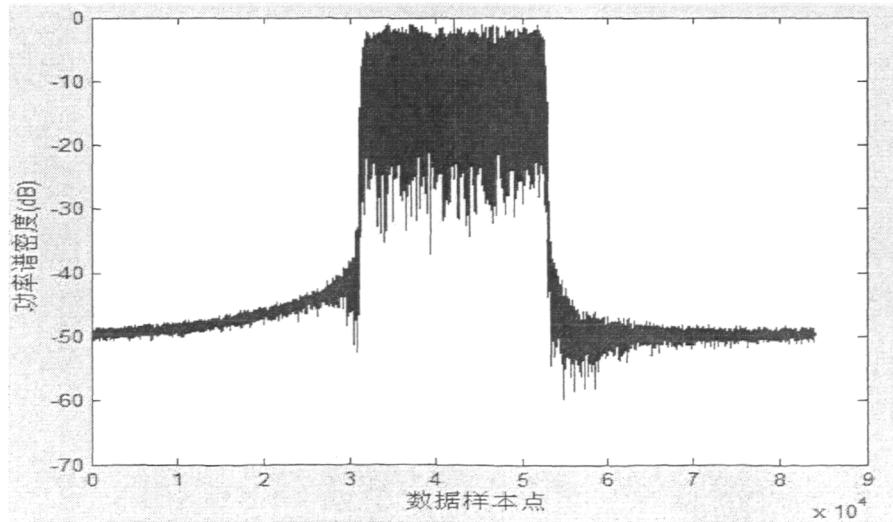


图 4-4 基带信号频谱

利用插入的 PN 序列，通过 PN 序列与数据相关，找寻相关峰值，通过峰值可

以判断信号帧位置，即对信号进行帧同步。在采用相同的 420 点 PN，四倍插值后信号帧点数为 4*4200 点，峰值间距固定，相关后数据波形图如图 4-5 所示：

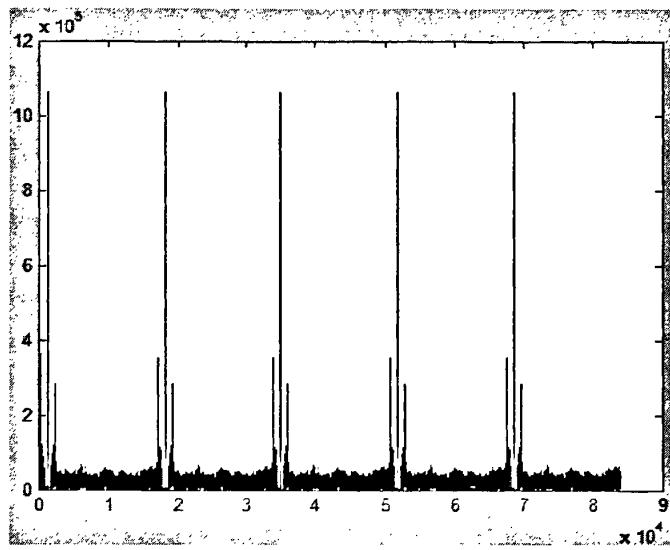
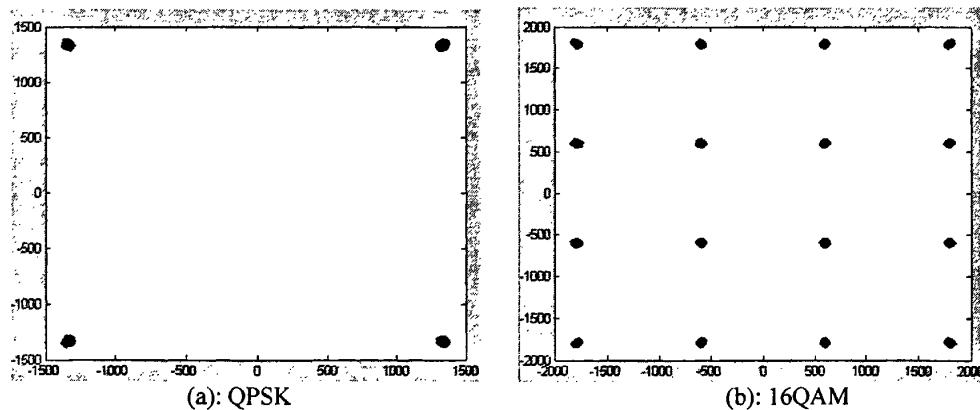


图 4-5 相关后数据波形图

从图中可以清楚的看到相关峰值，以及序列前后同步形成的次级峰值。这一结果表明系统帧体结构的帧头数据，完全符合设计要求。

通过相关峰值确定信号帧位置后，对数据进行去帧头、FFT 运算、解频域交织、去系统信息处理后，得到的为有效数据信息。

采用 QPSK、16QAM、32QAM、64QAM 映射的数据点图如图 4-6 所示：



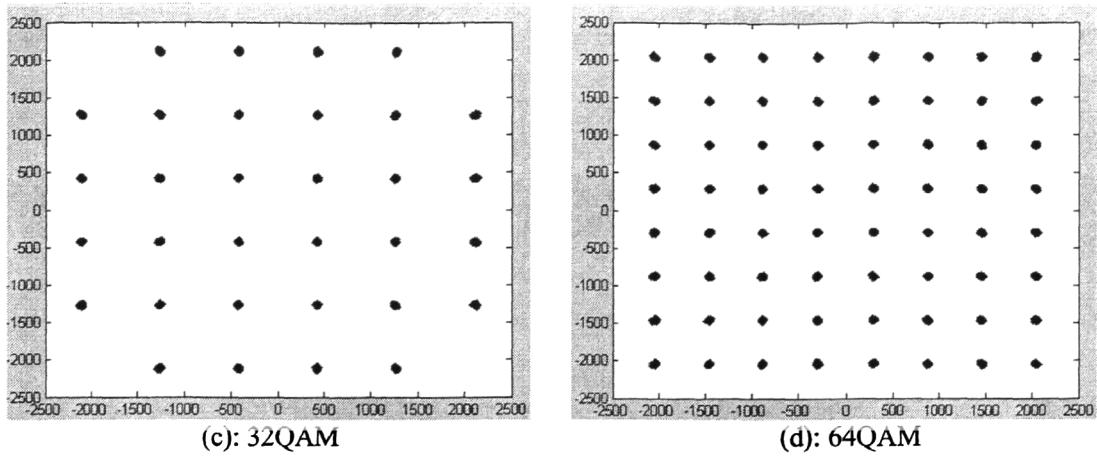


图 4-6 系统发端数据时频转换后点图

分析点图即可有效的对系统设计结果进行判断。通过对点图的分析可以看到，系统可以较好的满足设计要求，但是与理论图相比，可以看到，还存在一定的星座点扩散现象。通过对数据的分析可知，点图的扩散现象是因为在帧体数据处理和基带信号处理模块中，数字运算的截位处理带来了误差。将数据导入 Matlab 分析，在最终采用 14 位输出（既对专用器件 AD9857 的输入）时，系统的信噪比约为 50dB，这一结果已经满足本系统对信躁比的要求。在对信躁比要求更高的条件下，可以通过增加输出数据位宽，以及在帧体数据处理模块中添加一个块浮点处理模块来提高性能。

系统级联 FPGA 实现资源消耗如图 4-7 所示：

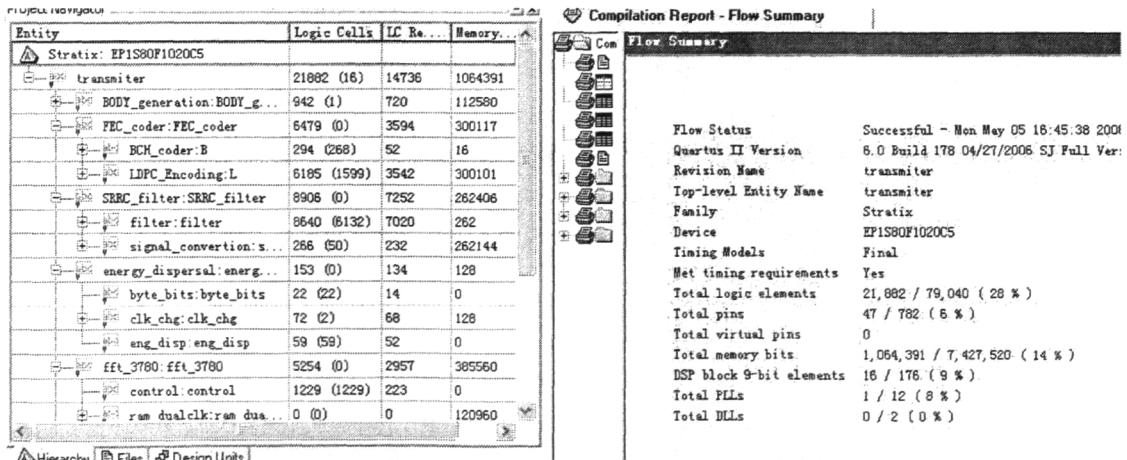


图 4-7 系统级联 FPGA 实现资源消耗图

注：现有硬件条件不含片外 SDRAM，因此无法在现有硬件条件下实现符号交织，故此系统级联调试中不包括符号交织部分。且符号交织部分实现功能不改变数据结构，不包含此部分不会对系统级联测试产生实质影响。

第五章 结束语

本文主要介绍了基于中国“数字电视地面广播传输系统帧结构信道编码和调制”国家标准的发端系统的 FPGA 设计与实现。文中首先简述了国标系统的要求和技术结构框架。针对国标系统的要求，介绍了发端系统的 FPGA 设计的基本结构与各基本模块功能。

首先，针对国标系统的技术要求，完成了系统 Matlab 建模、FPGA 设计、FPGA 仿真以及整个系统的验证。介绍了基于 FPGA 实现的发射端系统各个功能模块的具体结构设计、实现与验证。在设计过程中，以节约系统资源消耗和提高运算精度的角度出发，设计和实现了能量扩散、信道编码、帧体形成、信号帧成型等模块。同时，对已有的基于存储器的 3780 点 FFT 和 4 倍高阶插值滤波器的 FPGA 实现结构进行了优化和调整。经过验证，在保证模块功能时序的前提下，节约了大量系统资源消耗。从而完成了整个发射端系统各功能模块的实现。

在系统级联调试中，通过系统中若干功能模块的数据结构特点，作者已通过调整结构顺序，对系统进行了一定优化，节约了硬件资源。但是由于作者经验和能力有限，发端系统的整体结构的实现和具体算法还有待进一步的改进和优化。现阶段可进一步考虑优化的角度有：根据各模块处理数据的时钟延时特点，进一步简化模块，理论上可以利用这一特性减少一定量的数据缓存空间，对 LDPC 编码模块的算法进一步优化，提高 LDPC 编码模块的最高速率，使得系统处理能够采用单时钟模式等。

参考文献

- [1]. 国家标准化管理委员会. GB20600-2006. 数字电视地面广播传输系统帧结构、信道编码和调制. 北京: 中国标准出版社, 2006-8-18
- [2]. 李日勤. 地面数字电视广播标准的原理分析与思考. 西部广播电视台. 2004, NO.6: 9-10.
- [3]. 广州电子技术网. 上海数字电视与无线多媒体通讯国际论坛的情况.
- [4]. 潘长勇, 王军, 宋健. 中国地面数字电视广播传输标准概要. 电视技术. 2006, No.10: 45-47
- [5]. Stratix Device Handbook, Volume 1, 2, & 3
- [6]. 王新梅, 肖国镇. 纠错码—原理与方法. 西安: 西安电子科技大学出版社, 2006, 242-243
- [7]. Gallager R. G. Low Density Parity Check Codes[M]. Cambridge MA: MIT Press, 1963
- [8]. D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," Electro. Lett., vol.32, Aug. 1996. 1645-1646.
- [9]. Li Z, Chen L, Zeng L Q, et al. Efficient Encoding of Quasi-Cyclic Low-Density Parity-Check Codes. IEEE Trans. Commun., 2006, 54(1): 71-81.
- [10]. 蒋冰, 刘怀宇. 基于存储器的 3780 点 FFT 的 FPGA 设计与实现. 中国有线电视, 2005, No.23: 2340-2342
- [11]. 胡广书. 数字信号处理—理论、算法与实现. 北京: 清华大学出版社, 1997. 195-198
- [12]. Zhi-Xing Yang, Yu-Peng Hu, Chang-Yong Pan, and Lin Yang. "Design of a 3780-Point IFFT Processor for TDS-OFDM". IEEE TRANSACTIONS ON BROADCASTING, VOL. 48, NO. 1, MARCH 2002: 57-60
- [13]. DEAN P. KOLBA AND THOMAS W. PARKS. "A Prime Factor FFT Algorithm Using High-speed Convolution". IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-25, NO. 4, AUGUST 1977: 281-294
- [14]. 赵文亮, 蒋冰. 《基于 FPGA 的高阶高速 FIR 滤波器设计与实现》. 西安: 中国有线电视, 2006, No.3: 329-331
- [15]. Zhang Weiliang, Pan Changyong, Guo Xingbo, Yang Zhixing. Design and FPGA implementation of high-speed square-root-raised-cosine FIR filters. IEEE Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop,

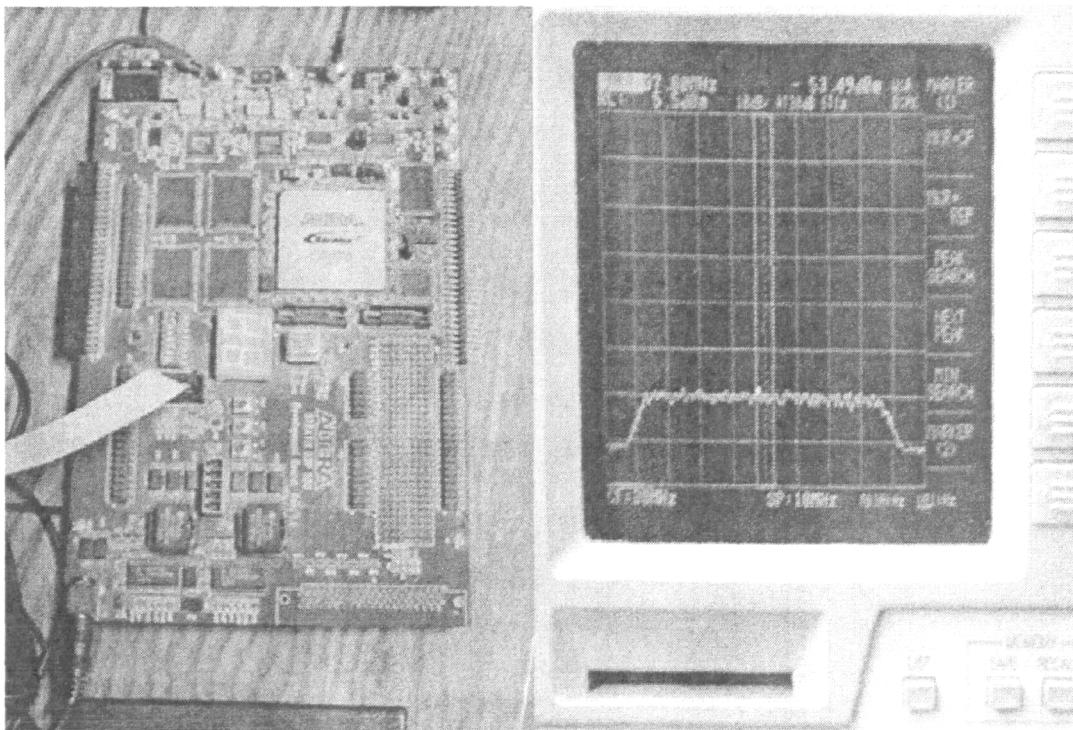
- Proceedings of 2002 IEEE 10th, 2002: 232-235.
- [16].Digital Video Broadcasting (DVB) ; Framing structure ,channel coding and modulation for digital terrestrial television. ETSI EN 300 744 V1.4.1, 2001-01.
- [17].夏宇闻. Verilog 数字系统设计教程, 北京: 北京航空航天大学出版社, 2005
- [18].佟学俭 罗涛. OFDM 移动通信技术原理与应用.北京: 人民邮电出版社, 2003
- [19].Michael D.Ciletti. Advanced Digital Design with the Verilog HDL(张雅绮 等译).北京: 电子工业出版社, 2003

附录

附一：星座映射中映射点电平能量归一后扩展对应幅度值：

QPSK	4.5 : 11586		
16QAM	2 : 5181		6 : 15543
32QAM	1.5 : 3663	4.5 : 10900	7.5 : 18317
64QAM	1 : 2528	3 : 7584	5 : 12640
			7 : 17696

附二：硬件测试平台与输出频谱图



硬件测试平台与输出频谱图

附三：169阶均方根升余弦滚降滤波器的性能指标

利用 Matlab 工具设计的 169 阶均方根升余弦滚降滤波器的性能指标为：

通带：[0, $\pi/4$]，对应实际频率[0, 3.78e6]；

阻带：[0.3 π , π]，对应实际频率[4.536e6, 15.12e6]；

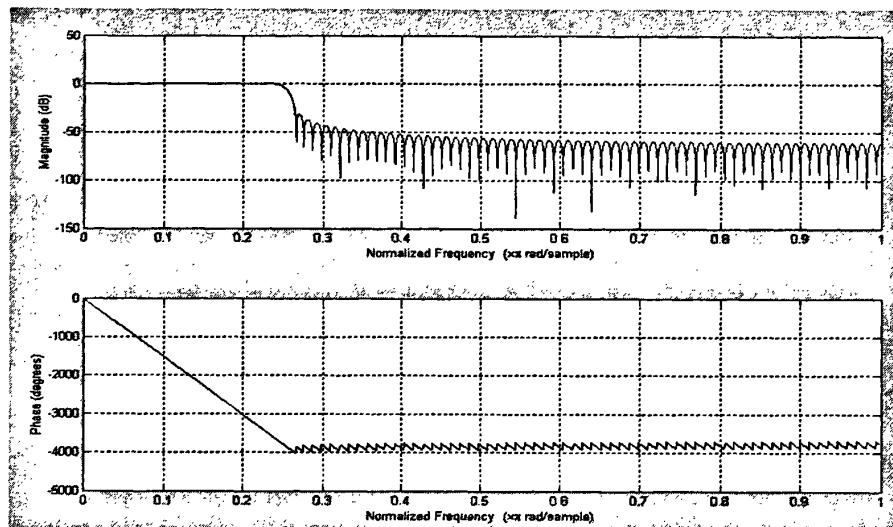
过度带：[$\pi/4$, 0.3 π]，对应实际频率[3.78e6, 4.536e6]；

通带波动： $R_p \leq 0.083$ dB；

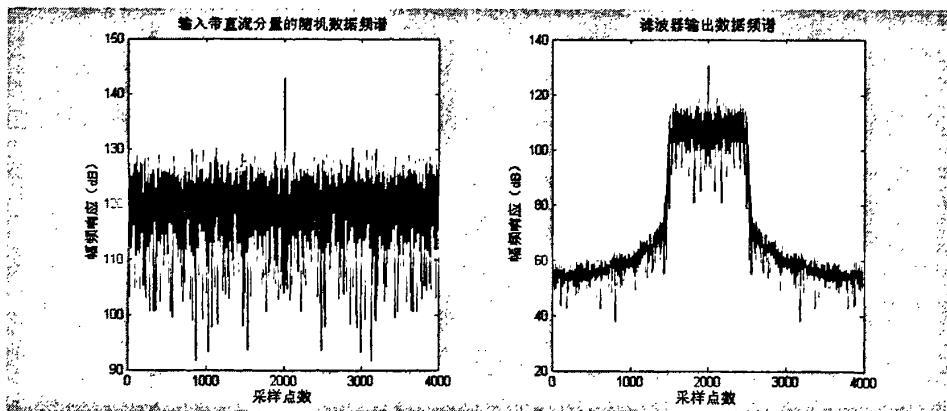
阻带衰减： $A_s \geq 45$ dB；

浮点转定点滤波器实际信噪比：SNR > 69 dB；

其频率响应特性如图所示。

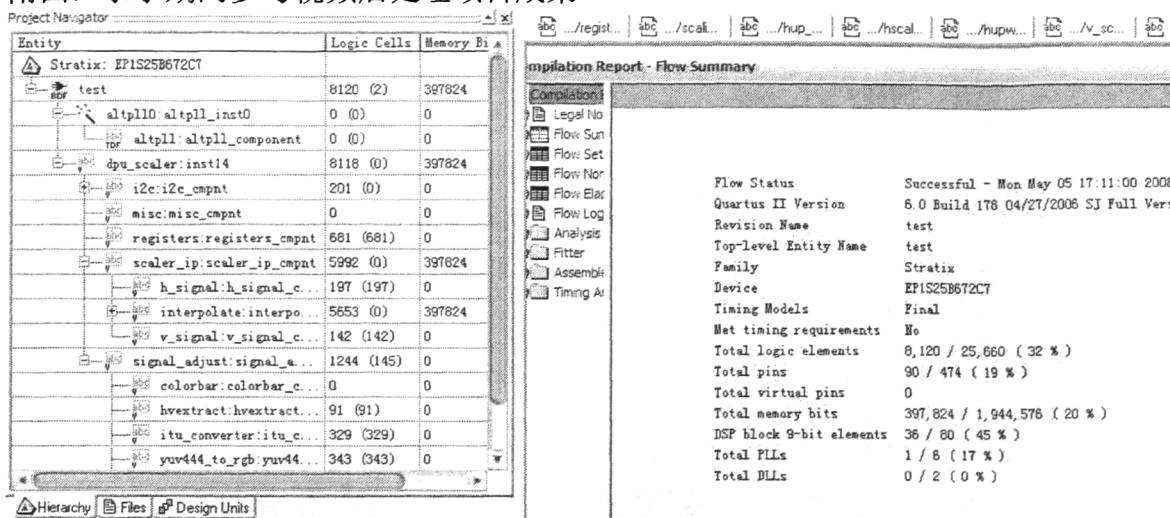


169 阶均方根升余弦滚降滤波器频响特性图

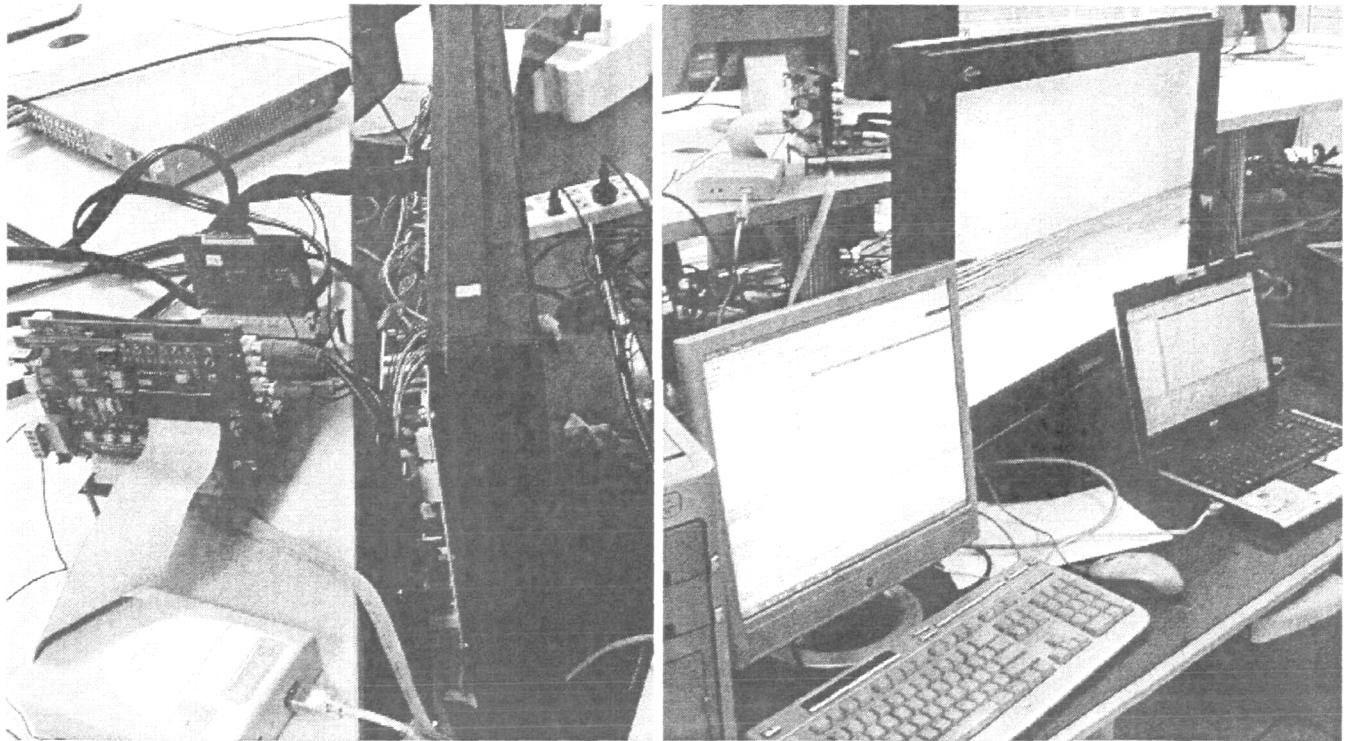


随机输入数据及滤波器输出数据频谱图

附四：求学期间参与视频后处理项目成果



视频后处理 Scaler 功能模块 FPGA 实现资源消耗图



视频后处理硬件平台与演示图象

致 谢

首先深深地感谢我的导师朱维乐教授。本项目的论证、总体规划都是在朱老师的指导下进行和完成的。在与朱老师共处的时间里，朱老师一丝不苟、严谨治学的学者风范，严谨的学习态度，渊博的学识，敏捷的思维都值得我学习，并将激励我在以后的人生道路上不断进取。

特别感谢课题组的许林峰、顾庆水和陈伟老师，他们在学习工作中给予了我很大的指导和帮助。没有他们的指导，我的学习研究工作将寸步难行。另外还要感谢同课题组的蒋冰，李颖异，贺杰，赵文亮、胡浩在课题进展过程中的协助。感谢文中参考文献的作者，本文中方案的设计以及实现是在他们工作的基础上进行的。

深深感谢我的家人对我的鼓励、鞭策与关怀，感谢他们在生活上对我的关心和帮助。

最后，衷心地感谢为评阅本论文而付出辛勤劳动的各位专家和学者。

学习经历、在学期间发表的学术论文

个人简历：

陈涛，男，1981年12月生。

2003年七月毕业于电子科技大学电子工程学院，获工学学士学位；

2004年九月至今就读于电子科技大学电子工程学院，攻读信号与信息处理专业硕士学位，师从朱维乐教授。

在学期间的研究成果

2005年3月—2007年6月，中国地面数字电视融合方案发端系统的FPGA设计。完成了整个系统的软件建模，仿真以及Verilog代码编写，仿真，测试和验证；

2007年7月至今，进行数字图象缩放算法研究，申请算法专利一篇。并实现基于这一算法的数字电视视频后处理系统Scaler部分的功能模块。采用Stratix系列的EP1S25B672C7 FPGA为基础的主硬件处理平台。项目成果见附录。

发表的学术论文

第一作者：《数字电视地面广播传输系统发射端的FPGA设计与实现》，《中国有线电视》，2007年第5期。

第一发明人：《基于三次样条函数的插值方法》，专利申请号200710201882.8