摘 要

本文对 OFDM 基带调制解调系统的 FPGA 设计进行了研究和论述，重点实现其中的 RS 码编、译码模块和基带成形滤波器模块。本文首先介绍了 OFDM 调制的原理和 OFDM 基带调制解调系统的总体设计，以及 FPGA 设计的基本原则。接着介绍了 RS 码的编码原理和时域迭代译码算法，在此基础上设计实现 RS 码编码器和译码器。然后介绍了成形滤波的原理和多种实现成形滤波器的结构，采用多相结构设计实现了平方根升余弦滚降滤波器。

关键字：OFDM，FPGA，RS 码，成形滤波

Abstract

This paper investigates algorithms and FPGA implementation of the OFDM base-band modem. RS coder 、decoder and square-root raised cosine filter are implemented. Firstly, OFDM modulation theory and top level module of the modem have been introduced, some basic principles for FPGA design have been introduced too. Then, the code theory and time-domain iterative decode algorithm of RS code is introduced, RS coder and decoder are designed and implemented in this base. After that, the principle of base-band signal shaping and some structures of base-band signal shaping filter are introduced, then the square-root raised cosine filter is implemented in polyphase structure.

KEY WORDS:OFDM, FPGA, RS code, signal shaping filter

第一章 绪论

1.1 课题背景和来源

在过去的10年里，移动通信得到了飞速地发展，第三代移动通信系统（3G）的出现更使移动通信前进了一大步。到目前为止，3G各种标准和规范已达成协议，并已开始商用。但也应该看到3G系统尚有很多需要改进的地方，如：3G缺乏全球统一标准；3G所采用的语音交换架构仍承袭了第二代（2G）的电路交换，而不是纯IP方式；流媒体（视频）的应用不尽如人意；数据传输率也只接近于普通拨号接入的水平，更赶不上xDSL等。所以，在第三代移动通信还没有完全铺开，距离完全实用化还有一段时间的时候，已经有不少国家开始了对下一代移动通信系统（4G）的研究。

目前普遍的观点是：下一代的无线通信网络将是基于统一的IPv6包交换方式，向用户提供的峰值速率超过100Mbit/s，并能支持用户在各种无线通信网络中无缝漫游的全新网络。为了支持更高的信息传输速率和更快的用户移动速度，在下一代的无线通信中必须采用频谱效率更高、抗多径干扰能力更强的新型传输技术。在当前能提供高速率传输的各种无线解决方案中，以正交频分复用（OFDM）为代表的多载波调制技术是最有前途的方案之一。

移动通信系统中存在的主要问题是多径效应所引起的码间干扰（ISI）问题。传统上克服ISI的方法有两种：第一种方法是采用单载波调制加时域均衡的方法；第二种方法是直接序列扩频码分多址（DS-CDMA）加RAKE接收技术。这两种方法在各自系统中都能很好地克服因多径效应所引起的码间干扰问题。但是对于高速数据业务来说，传统的单载波系统和CDMA系统都存在很大的缺陷。由于无线信道存在时延扩展，而且高速信息流的符号宽度又相对较短，所以符号之间会存在较严重的ISI。因此对单载波系统中所使用的均衡器提出非常高的要求，即抽头数量要足够大，训练符号要足够多，训练时间要足够长，这样均衡算法的复杂度也会大大增加。对于CDMA系统来说，其主要问题在于扩频增益与高速数据流之间的矛盾。在保证相同带宽的前提下，对高速数据流所使用的扩频增益不能太高，

否则就大大限制了 CDMA 系统噪声平均的优点，从而使得系统的软容量受到一定的影响，如果保持原来的扩频增益，则必须要相应地提高带宽。此外，受系统复杂性的限制，CDMA 系统中 RAKE 接收机的分支数量不能太多（一般为 5 左右），但高速宽带系统中可分解的多径分量较多，此时会有较大的能量损失。

多载波调制技术把一个高速串行数据流分解为若干个独立的子数据流，这样每个子数据流将具有很低的比特速率，用这样的低比特率数据形成的低速率多状态符号再去调制相应的子载波，从而构成多个低速率符号并行发送的传输系统。从而可以有效地对抗多径干扰，这种优势是其它调制方式所无法比拟的。

OFDM 是多载波传输方案的实现方式之一，它利用逆快速傅立叶变换（IFFT）和快速傅立叶变换（FFT）来分别实现调制和解调，是实现复杂度最低、应用最广的一种多载波调制方案。此外，人们还提出了许多其他实现多载波传输的调制方式，如矢量变换方式、基于小波变换的 DWMT 方式、采用滤波器组的滤波多音（FMT）调制方式等，但它们的实现复杂度相对较高，在实际系统中很少使用。正因为 OFDM 潜在的多径对抗能力，且可以灵活地和其他接入方式结合成衍生系统，所以 OFDM 已经被列入 4G 无线通信系统的可能解决方案，受到众多研究者的广泛关注。目前 OFDM 作为核心技术已被多种有线和无线接入标准采纳，包括 IEEE802.11a、HIPERLAN-2、DAB、DVB、IEEE802.16 系列标准等。

本文主要参考 IEEE802.16d 标准，研究和设计全数字 OFDM 基带调制解调系统，并基于 FPGA 实现其中的多个关键模块。

1.2 课题意义

设计全数字通信机，降低了电路复杂度，便于采用先进的算法提高通信机的性能指标，便于采用计算机辅助设计，实现电子设计自动化，便于移植、集成和大规模生产。

自主研发的基带调制解调系统，不需采用专用的调制解调芯片，更有利于自身的技术积累和自我保护，更加灵活自由，不会受到专用调制解调芯片技术和知识产权方面的限制。

IEEE802.16d 标准支持宽带固定接入，这是它最基本的业务模型。全移动接入

是 IEEE802.16 标准发展的最终应用，在这种应用模式下，可以漫游切换，并支持普通车速移动下无中断的无线接入，其市场容量巨大。目前国内采用 IEEE802.16 标准的通信机还没有得到推广，市场对这类设备的需求很大。因此，研究设计基于 IEEE802.16d 标准的全数字 OFDM 基带调制解调系统是一个重要的技术突破，也为基于 OFDM 的 4G 移动通信系统的研究打下基础，具有长远的意义。

本文研究和设计的 OFDM 基带调制解调系统参考了 IEEE802.16d 标准，将 OFDM 基带调制解调所需的扰码、RS 码、卷积码、交织、16QAM 映射、OFDM 调制/解调（IFFT/FFT）、加循环前缀、基带成形滤波器、相关接口电路全部集成到一个 FPGA 芯片上，以实现片上系统（SOC）设计，大大简化系统的复杂度，提高系统的性能。在设计整体系统的前提下，本文重点实现其中的 RS 码编码器和译码器模块，以及基带成形滤波器模块。

1.3 论文内容和结构

本文主要研究和设计了 OFDM 基带调制系统整体设计方案，基于 FPGA 设计和实现了 RS 码编、译码器及基带成形滤波器，工作量较大，具有一定的难度。

本文按照设计要求初步完成了以上模块的设计，并在设计过程中采用了许多先进的技术和设计理念，合理地利用了 FPGA 资源，优化了整个系统的设计，缩短了设计周期。

本文的章节安排如下：

第一章介绍了本文的课题背景、意义、论文内容和结构。

第二章介绍了 OFDM 调制的原理、OFDM 基带调制解调系统发送端和接收端的总体设计方案及参数选择原则。

第三章介绍了基于 Verilog HDL 进行 FPGA 设计的相关内容。

第四章介绍了 RS 码原理、RS 码编码器和译码器的设计及实现。

第五章介绍了 FIR、IIR 滤波器设计原理和平方根升余弦滚降滤波器的设计及实现。

第六章总结全文，并给出了一些改进思路。

第二章 OFDM 基带系统整体设计

2.1 OFDM 简介

多径衰落是影响移动通信系统的主要问题。由于多径传播，会造成信道的时间弥散性效应。在无线电波传输过程中，由于时延扩展，接收信号中的某些符号的波形会扩展到其它符号中，造成了符号间干扰（Inter Symbol Interference ,ISI）。

在单载波系统中,如果数据传输速率较低，多径效应造成的符号间干扰（ISI）不是特别严重，可以利用合理的均衡技术使系统正常工作。但对高速数据业务而言，时延扩展造成的 ISI 很大，这样就会对均衡器提出非常高的要求，特别是均衡器实现的复杂性及收敛速度，导致均衡器难以实现或实现的成本过高。

OFDM，即正交频分复用，是多载波调制（MCM）技术的一种。MCM 的基本思想是把一路高速串行数据流经串并转换为 N 路低速并行的子数据流，用它们分别去调制 N 路子载波后进行并行传输。由于子数据流的速率是原来的 1 /N，即符号周期扩大为原来的 N 倍，远大于信道的最大延迟扩展，这样 MCM 就把一个宽带频率选择性信道划分成了 N 个窄带平坦衰落信道（均衡简单），从而先天具有很强的抗无线信道多径衰落和抗脉冲干扰的能力，特别适合于高速无线数据传输。OFDM 是一种子载波频谱相互混叠的 MCM,因此它除了具有上述 MCM 的优势外，还具有更高的频谱利用率。OFDM 选择时域相互正交的子载波集，它们虽然在频域相互混叠，却仍能够在接收端被分离出来。

OFDM 的历史可以追溯到 20 世纪 60 年代中期，Chang 发表了关于将带限信号合成进行多信道传输的论文，其中阐述了把消息在线性带限信道中进行无信道间干扰（ICI）和符号间干扰（ISI）的并行传输的基本原理。之后，Saltzberg 对这种处理进行了分析，并得出这样的结论：“设计一个有效的并行处理系统的目的应该在于减少相邻信道间的串扰，而不在于完善单个信道，因为减小串扰失真更为重要。”这个非常重要的结论，在几年之后所形成的数字基带处理技术得到了证明。

1971 年 Weinstein 和 Ebert 对 OFDM 的发展做出了重要贡献:他们将 DFT 技术引入 OFDM 中，进行基带调制和解调。他们的工作也不在于“完善单个信道”，而

是引入了更加有效的处理技术，避免了使用大量子载波振荡器，并且为了减少 ISI 和 ICI，他们在 OFDM 符号间加入了保护间隔以及时域升余弦窗函数，但在他们的系统中，子载波在弥散信道环境下不能保持完善的正交性。

1980 年 Peled 和 Ruiz 对 OFDM 技术做出了另一个重要贡献，即把循环前缀（CP）或称循环扩展引入 OFDM 系统以解决子载波间的正交性问题。他们在保护间隔中加入的是 OFDM 符号的循环扩展，而不是使用空白保护间隔，从而有效地模仿了循环卷积信道，当 CP 大于信道冲激响应时间（即信道的最大延迟扩展 τ_{max}）时，就能够保证弥散信道中子载波间的正交性。虽然加入 CP 也同时引入了能量损失，但是相比于其所获得的几乎是零的 ICI，还是值得的。

1999 年，IEEE 将 OFDM 作为无线局域网标准 IEEE802.11a 的物理层的调制标准。目前，OFDM 已被多种数字无线通信标准所采纳，如欧洲的数字音频广播 DAB、数字视频广播 DVB-T，以及无线局域网 HIPERLAN2 等。OFDM 在蜂窝移动通信中的应用始于 20 世纪 80 年代中期，从 1993 年开始兴起了 OFDM 与 CDMA 相结合的研究:欧洲已把 OFDM 作为发展地面数字电视的基础;日本也将它用于发展便携式电视和安装在旅游车、出租车上的车载电视；OFDM 也应用于有线环境的各种高速 PSTN 接入以抗脉冲干扰、防止串话，如 ADSL, RDSL, VDSL 等，当 OFDM 应用于有线时，通常被称为 DMT，即离散多音频;OFDM 技术还应用于 HFC 及 HDTV 传输系统。

OFDM 具有频谱利用率高，抗多径衰落能力强等优点，是一项非常有潜力的高速数据通信技术。OFDM 技术的优点主要归结为以下几个方面:

（1）频谱利用率高。

在 OFDM 系统中，各个子载波之间相互正交，频谱互相重叠，提高了系统的频谱利用率。当子载波个数很大时，频谱利用率趋于 2Baud/Hz。这一优点在频谱资源日益紧张的无线通信应用中尤为重要。

（2）有效克服符号间干扰。

OFDM 系统中，把高速串行的数据流通过串/并转换器变为并行低速的子数据流，使得每个子载波上数据符号的持续周期相对增加，从而有效地减小无线信道时间弥散所带来的 ISI。同时，在数据间插入的循环前缀也有效地减低了 ICI。

（3）使用 IFFT/FFT 方法来实现 OFDM 调制和解调。

大规模集成电路技术和数字信号处理技术的飞速发展，使 IFFT/FFT 的实现变得极为容易，也为 OFDM 技术实用化扫除了障碍。

（4）对抗频率选择性衰落或窄带干扰。

在单载波系统中，单个衰落或干扰能够导致整个通信链路失败，但是在多载波系统中，窄带干扰只会影响到一个或有限的几个子信道。对于这些子信道，可以通过降低受干扰子载波的数据率或放弃受干扰的子载波，来降低窄带干扰对整个 OFDM 系统性能的影响。

（5）具有频率分集能力。

由于 OFDM 系统的频带内存在零点，所以，数据被并行分配在互不相关的子频带上发送时，可将时间分集与频率分集结合起来，提高系统传输的可靠性。

任何一种技术都不可能是十全十美的，OFDM 技术也不例外。OFDM 信号是由多个子信道信号叠加而成的，因此与单载波系统相比，存在以下两大缺点：

（1） 高峰均功率比(Peak-To-Average Power Ratio ,PAPR)。

高峰均功率比对发射机内功率放大器的线性范围提出了很高的要求，且会引起信号频谱的变化，破坏子载波间的正交性。因此，如何降低 PAPR 就成为有效应用 OFDM 技术中的一个难点，

（2） 对频偏和相位噪声比较敏感。

OFDM 系统中，各个子载波必须严格满足频率正交性。无线信号的频率偏移，或者发射机载波频率与接收机本地振荡器之间的频率偏移，都会破坏子载波间的正交性，产生子载波间串扰，导致整个系统性能严重下降。

2.2 OFDM 调制原理

OFDM 调制原理如图 2-1 所示：速率为 $R_b\ bit/s$ 的串行比特流，经过数据编码器，每 $\log_2 M$ 个比特被映射为一个符号（M 为符号空间的符号个数），从而产生速率为 $R_s = \frac{R_b}{\log_2 M}$ 符号/s 的串行符号流，符号周期 $T_s = \frac{1}{R_s}$ s。串行符号串并变换为 N 路并行符号，每一个符号调制 N 个正交子载波中的一个，N 个调制后的子载波相加，再进行传输。

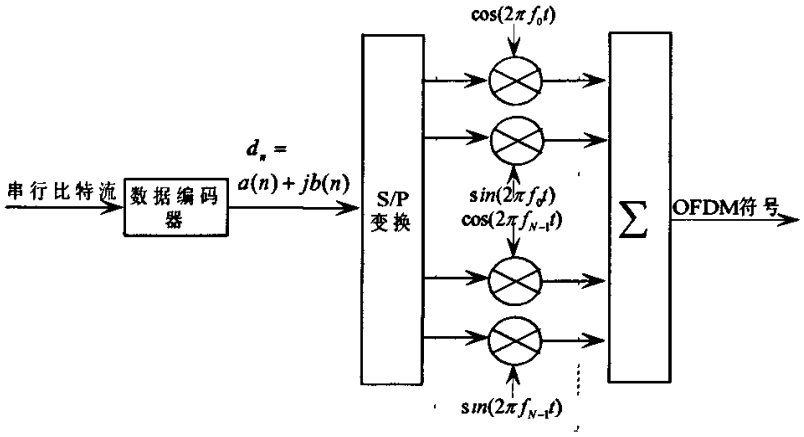

图 2－1 OFDM 调制原理示意图

各子载波间的正交性是通过适当选取 f_0，每个子载波在一个OFDM符号周期内包含整数个周期，各个相邻子载波之间相差一个周期实现的。取子载波间隔 $\Delta f = 1/NT_s$，及 $f_0 = k/T_s$（其中k为大于或等于零的整数，一般取零），$f_n = f_0 + n\Delta f$，则各子载波在一个OFDM符号周期内可保持正交。如下所示。

$$\frac{1}{T}\int_0^T \exp(\ j\omega_n t)\exp(\ -j\omega_m t)\,dt = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$$

在接收端，如图2-2所示，OFDM符号经过混频器/积分器组进行解调和判决。

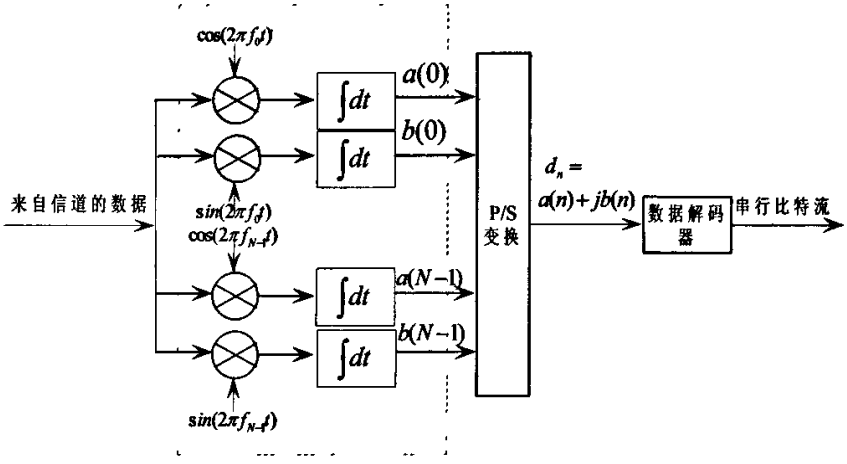

图2-2　OFDM系统接收端的解调部分

在不考虑同步误差以及信道干扰的理想情况下，进入每一路混频器/积分器的信号为相互正交的正弦信号和余弦信号的和，其形式为：

$$a(0)\cos(2\pi f_0 t) + b(0)\sin(2\pi f_0 t) + ,....,+a(N-1)\cos(2\pi f_{N-1} t) + b(N-1)\sin(2\pi f_{N-1} t)$$

考虑 f_n 子载波上同相分量：频率为 f_n 的同相混频器将整个上式乘以 $\cos(2\pi f_n t)$，然后将所得的积在时间 $0 \sim NT_s$ 内进行积分。因为载波间的正交性，积分运算只对位于 f_n 处的同相子载波有非零结果，从而提取出有用数据 $a(n)$。同样的道理可以从各正交支路恢复出 $b(n)$。所有被恢复的符号经并串变换后，再进行解码，即得到所发送的原始数据比特。可见，虽然子信道频谱相互混叠，子载波间的正交性却使得各个子信道依然能够被分离出来。子载波间的正交性还可以从频域的角度进行理解，如不加特殊的脉冲成型，即矩形脉冲成型的情况，由于并行符号的每一路的周期为 NT_s，则每一路的频谱都是取样函数，如图2-3所示，取样函数的峰值出现在各个子载波频率 f_n 处，而此处，所有其它子信道的频谱恰好为0。在解调的过程中需要计算的是这些峰值，因此可以从多个互相混叠的子信道符号频谱中提取出每个子信道符号，而不受其它子信道的干扰。

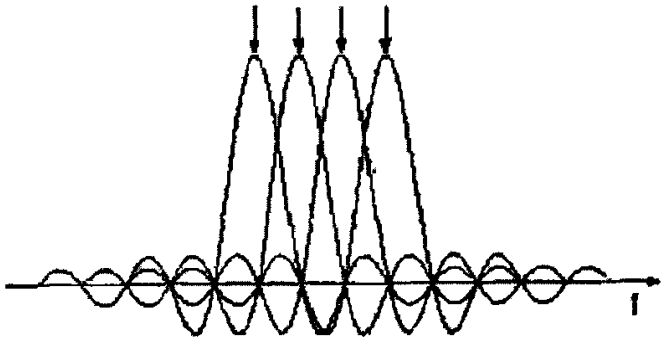

图 2-3 OFDM子信道的频谱(矩形脉冲成型)

采用 IFFT/FFT 来实现 OFDM 调制解调，可以省掉大量的振荡器和积分器，无论是算法还是硬件实现都可以采用比较成熟的 DSP 技术，从而有效地降低系统的成本和复杂度。大大促进了 OFDM 技术的实用化。

OFDM调制的输出信号可写作：

$$X(t)=\sum_{n=0}^{N-1}[a(n)\cos 2\pi f_n t+b(n)\sin 2\pi f_n t], \quad \text{其中} 0\le t\le NT_s \tag{2-1}$$

对其以 $\Delta t = T_s$ 进行采样，得到

$$X(mT_s)=\sum_{n=0}^{N-1}[a(n)\cos 2\pi f_n (mT_s)+b(n)\sin 2\pi f_n (mT_s)] \tag{2-2}$$

其中 $m=0,.....N-1$。将这些分量在 Δt 的时间内进行低通滤波，即进行 D/A 变换，则又可以恢复为原来的模拟信号 $X(t)$。

将 f_n 代入，（2-2）式可以写作：

$$X(m)=\mathrm{Re}\left\{\sum_{n=0}^{N-1}[a(n)+jb(n)]e^{\frac{-j2\pi nm}{N}}\right\}\qquad m=0,1,2,.....,N-1 \tag{2-3}$$

而大括号内的表达式正是序列 $X(n)=a(n)+jb(n),(n=0,1,.....N-1)$ 的离散傅立叶变换 DFT。可见，图 2-1 的方框内部分可用 DFT 实现，取 DFT 输出的实部，再经 D/A 变换和重建平滑滤波又可恢复成原来的模拟信号 $X(t)$，经上变频即可送入信道进行传输。由于只传 DFT 输出的实部，在接收端(下变频之后)则须以 $\Delta t/2$ 时间间隔进行采样，并进行 2N 点 DFT 才能恢复出原来的数据。与发送端相同，图 2-2 接收端的方框内部分也可以用 DFT 模块代替。为使 DFT 输出为实数，可以将 $X(t)$ 的共轭反转序列 $X^*[-n]$ 右移 2N 点置于 $X(n)$ 之后从而构成 2N 点的共轭偶对称序列，其 2N 点的 DFT 即为实数。另外，如果上变频调制的输入既有 I 通道又有 Q 通道的话，则不仅可以传实部(I 通道)，还可以传虚部(Q 通道)。

由于将 $a(n)$ 改为 $-a(n)$ 或将 $b(n)$ 改为 $-b(n)$ 不影响信息的携带，又由于 OFDM 作为 MCM 的一种被视为一种频域技术，则图 2-1 所示的虚线框内部分，也可用 IDFT 来实现。使用 IDFT 的输出为：

$$X'(m)=\mathrm{Re}\left\{\frac{1}{N}\sum_{n=0}^{N-1}[a(n)+jb(n)]e^{\frac{j2\pi nm}{N}}\right\} \tag{2-4}$$

经 D/A 变换后得：

$$X'(t)\approx\frac{1}{N}\sum_{n=0}^{N-1}[a(n)\cos 2\pi f_n t-b(n)\sin 2\pi f_n t] \tag{2-5}$$

与(2-1)相比，$X'(t)$ 与 $X(t)$ 的差异仅在于幅度以及 $b(n)$ 的符号，因而不会影响信息的传输。由于 DFT/IDFT 有快速算法 FFT/IFFT，如果 $N\neq 2^m$，可补零，从而图 2-1 中的虚线框中部分也可用 IFFT 实现，图 2-2 接收端的框内部分可用 FFT 实现。这种方法更为常用。

2.3 OFDM 基带调制解调系统整体设计

发送端整体示意图和接收端整体示意图分别如图 2—4 和图 2—5 所示。根据 IEEE 802.16d 标准，数据经过扰码（随机化）、RS 码编码、卷积码编码、交织、16QAM 映射，进行 OFDM 调制，再加入循环前缀，基带成形滤波，最后通过天线进行发射。解调过程按相反顺序逐步进行。图中自 DA/AD 以下各部分都集成在单片 FPGA 上。设计中暂未考虑同步、峰均比抑制等问题。

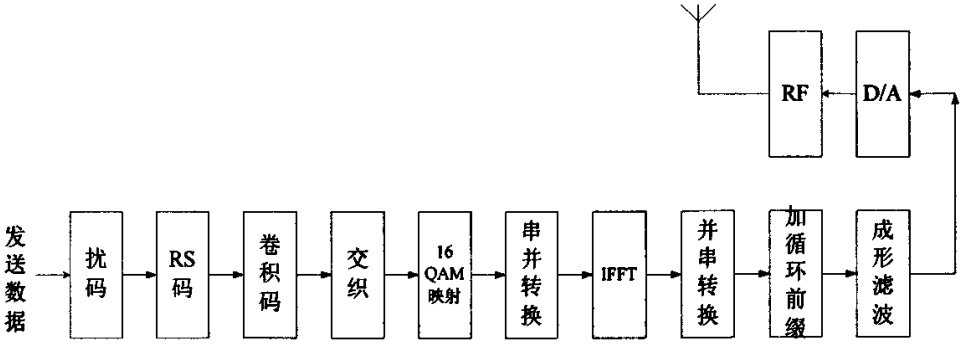

图 2—4 发送端整体示意图

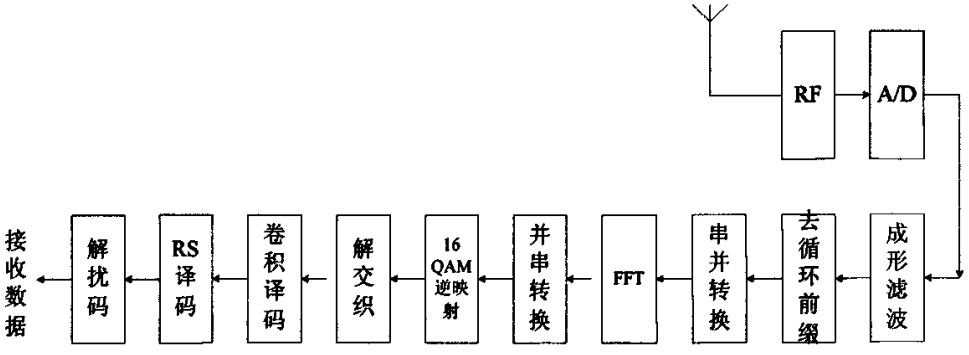

图 2—5 接收端整体示意图

2.3.1 扰码模块

进行基带传输的信号如果出现长的连“1”和连“0”串，则会包含较大的低频分量，则很难通过信道进行传输。对数据进行离散化，也称之为“扰码”，使数据变成伪随机序列，可解决这一问题。它的原理是将待传送的数据与一个伪随机序列进行逐比特异或运算，然后再进行随后的处理步骤并通过信道传送。在接收端把经过解调解码的数据流与相同的一个伪随机序列进行异或，则可以恢复出原

始的发送信息。扰码器可用一个线性反馈移位寄存器构成，如图 2−6 所示。

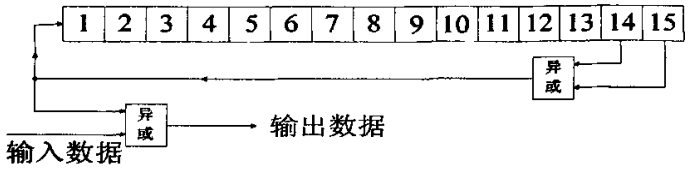

图 2−6 扰码模块示意图

2.3.2 RS 码模块

RS 码是一类有很强纠错能力的多进制 BCH 码，也是一类典型的代数几何码。它最早是由 Reed 和 Solomon 应用 MS 多项式在频域构造出来的。利用 MS 多项式构造的 RS 码是非系统码，我们可以利用 BCH 码的构造方法在时域产生系统码。对于线性分组码来说，在其他指标相同的情况下，非系统码和系统码的纠错能力相同，但在译码时系统码可直接读出信息元，非系统码还要增加一个信息元的转换电路。所以本文采用系统码编码方案。

RS(255,239)码是在有限域 GF（2^8）上运算得到的。它的编码参数如下：

码长 n=255，信息位个数 k=239，

校验位 $r=n-k=16$，纠错能力 t=8，码距 d=17

本原多项式 $p(x)=x^8+x^4+x^3+x^2+1$

生成多项式 $g(x)=(x-\alpha^0)(x-\alpha^1)(x-\alpha^2)\cdots(x-\alpha^{15})$

RS 码译码器的结构较为复杂，通常包括伴随式计算电路、关键方程求解电路、钱氏搜索电路、福尼算法电路和译码输出等部分。其中关键方程的求解是整个译码器的核心，本文采用 Berlekamp-Massey（BM）算法来求解关键方程。

本文在信道编码部分，采用 RS（255，239）码与（2，1，6）卷积码级联的结构。级联码一定程度上解决了纠错码理论中性能与设备复杂性的矛盾，它不仅有极强的纠突发错误的能力，也有极强的纠随机错误的能力。可以根据需要，对编码采用不同的增信删余方案，可以得到不同的编码速率。交织技术对经过编码的数据序列重新排列，解交织后使突发错误在时间上被分散，从而有利于纠错编码有效的进行纠错。

RS 码的编、译码器的设计和实现是本文的重点。

2.3.3 卷积码模块

卷积码可表示为(n_0，k_0，m)，参数的意义如下：

*码长n_0，信息元个数k_0，校验元个数n_0-k_0。

*码率k_0/n_0，表示卷积码传输信息的有效性。

*编码存储m，表示输入信息组在编码器中的存储周期（即编码器中寄存器的级数）。

*编码约束度$N=m+1$，表示编码过程中相互约束的信息组的个数。

*编码约束长度$N_a=n_0N$，表示编码过程中相互约束的码元个数。

(n_0，k_0，m)卷积码编码器通常分为两类：一类是$m(n_0-k_0)$级的串行编码器；另一类是mk_0的串行和并行编码器。对于最常见的$(2,1,m)$卷积码来说，$m(n_0-k_0)$级的串行编码器最常用。特别是在使用子生成元的方式来描述卷积码时，可直接根据子生成元方便地画出相应的$m(n_0-k_0)$级的串行编码器。对$(2,1,m)$卷积码进行增信删余处理，即可方便地获得其它码率的卷积码。

卷积码在编码过程中，充分利用了各组之间的相关性，k_0和n_0较小，在与分组码同样的码率和设备复杂性条件下，从理论和实际上均以证明卷积码的性能不比分组码差，并且实现最佳和准最佳译码也比较简单。由于卷积码各组之间相互有关，因此在卷积码的分析过程中，至今没有找到像分组码那样有效的数学工具，所以对卷积码的性能分析比较困难，从分析中得到的成果也不像分组码那么多，实际应用中往往要借助计算机的搜索来寻找好码。

卷积码的译码通常有门限译码、序列译码、Viterbi算法这三种较好的译码算法。特别是Viterbi算法是基于码的网格（trellis）图基础上的一种最大似然译码算法，是一种最佳的概率译码算法。

2.3.4 16QAM映射

QAM正交幅度调制是用两路独立的信号分别去调制同相与正交的两个载波，QAM已调信号可以表示为：

$$\begin{aligned} s_m(t) &= \mathrm{Re}[(A_{mc} + jA_{ms})g(t)e^{j2\pi f_c t}], (m = 1,2,...,M, 0 \le t \le T) \\ &= A_{mc}g(t)\cos 2\pi f_c t - A_{ms}g(t)\sin 2\pi f_c t \end{aligned} \quad (2-6)$$

式（2−6）中，A_{mc}和A_{ms}是承载信息的正交载波的信号幅度，$g(t)$是信号脉冲。

用另一种方法可将 QAM 信号波形表示为：

$$\begin{aligned} s_m(t) &= \mathrm{Re}[V_m e^{j\theta_m} g(t)e^{j2\pi f_c t}] \\ &= V_m g(t)\cos(2\pi f_c t + \theta_m) \end{aligned} \quad (2-7)$$

式（2−7）中$V_m = \sqrt{A_{mc}^2 + A_{ms}^2}, \theta_m = \tan^{-1}(A_{ms}/A_{mc})$。该表达式表明，QAM 信号波形可以看做组合幅度和相位调制。

图 2-7 是 16QAM 的一个矩形信号点星座图。

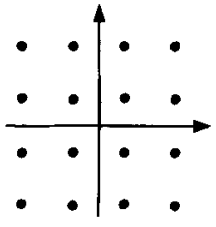

图 2-7 16QAM 矩形信号星座图

16QAM 映射的实现框图如图 2-8 所示。

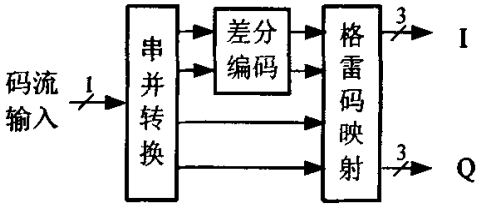

图 2-8 16QAM 映射框图

2.3.5 IFFT/FFT 模块

DFT 的出现大大地增加了数字信号处理的灵活性，有效的 FFT 算法的出现使得硬件实现高速实时的 DFT 运算成为可能。FFT 是数字信号处理中应用最为广泛的工具之一。根据不同的实现方法，可有多种 FFT 算法。

快速傅立叶(FFT)变换利用旋转因子W_N^m具有周期性和对称性的特点对运算进行简化。旋转因子W_N^m的周期性表现为

$$W_N^{m+lN} = e^{-j\frac{2\pi}{N}(m+lN)} = e^{-j\frac{2\pi}{N}m} = W_N^m$$

旋转因子W_N^m的对称性表现为

$$W_N^{-m} = W_N^{N-m} \quad 或者 \quad [W_N^{N-m}]^* = W_N^m \text{ 或者} W_N^{m+\frac{N}{2}} = -W_N^m$$

通过简化，复数乘法运算量从N^2次减少到（$N/2$）$\log_2 N$次。在硬件实现中，乘法所需运算时间远远大于加法，因此以乘法运算量作为整个算法的运算量。当N趋向较大值时，FFT 所占用的时间远远小于 DFT。

FFT 的基本思想在于，将原有 N 点序列分解成为两个或更多的较短序列，这些短序列的 DFT 可重新组合成原序列的 DFT，而总的运算次数却比直接的 DFT 少的多，从而达到提高运算速度的目的。FFT 算法按照分解方式基本上可分成两大类：时域抽取法 FFT(Decimation－In－Time FFT，简称 DIT－FFT)和频域抽取法 FFT(Decimation－In－Frequency FFT，简称 DIF－FFT)。DIT－FFT 算法运算流图如图 2－9 所示。

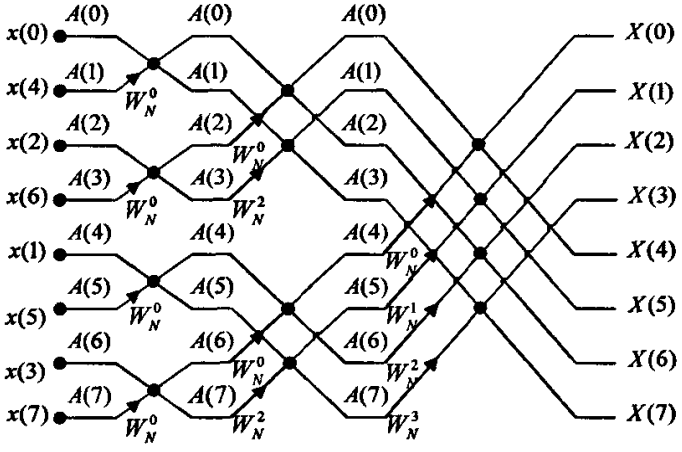

图 2－9 N 点 DIT—FFT 运算流图(N=8)

IFFT 运算的实现有多种方式，既可以经预处理后直接调用已设计好的 FFT 模块来实现，也可以直接采用基-2 的 DIT—IFFT 运算流图来实现。

如果要调用已设计好的 FFT 模块实现 IFFT，可根据

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-kn} \tag{2—8}$$

所以

$$x^*(n) = \frac{1}{N}\sum_{k=0}^{N-1} X^*(k) W_N^{kn} \tag{2—9}$$

对式（2－8）和式（2－9）两边同时取共轭，得

$$x(n)=\frac{1}{N}[\sum_{k=0}^{N-1}X^*(k)W_N^{kn}]^*=\frac{1}{N}\{DFT[X^*(k)]\}^* \qquad (2—10)$$

这样，可以先将 $X(k)$ 取共轭，然后直接调用已设计好的 FFT 模块进行处理，最后取共轭并乘以 $1/N$ 得到序列 $x(n)$。

若直接采用基-2 的 DIT—IFFT 实现 IFFT：

将 FFT 算法流图稍加变换，就可以用于离散傅立叶逆变换（Inverse Discrete Fourier Transform，简称 IDFT）。比较 DFT 和 IDFT 的运算公式：

$$X(k)=DFT[x(n)]=\sum_{n=0}^{N-1}x(n)W_N^{kn} \qquad (2—11)$$

$$x(n)=IDFT[X(k)]=\frac{1}{N}\sum_{n=0}^{N-1}X(k)W_N^{-kn} \qquad (2—12)$$

只要将 DFT 运算式中的系数 W_N^{kn} 改为 W_N^{-kn}，最后 乘以 $1/N$，就是 IDFT 的运算公式。所以，只要将上述的DIT—FFT与DIF—FFT算法中的旋转因子 W_N^p 改为 W_N^{-p}，最后的输出再乘以 $1/N$ 就可以用来计算 IDFT。这两种方法各有优势，根据具体应用中的实际要求灵活选用。

2.3.6 加循环前缀

OFDM系统数据流被分配到N个子载波，符号速率变成单载波系统的1/N，低符号速率使得OFDM能有效抵抗由于多径传播而引起的符号间干扰（ISI）。ISI对OFDM符号的影响还可以通过在每个符号的开始位置加上额外的保护间隔来进一步改善。如果所加的保护间隔是扩展符号波形的循环复制，则每个子载波在符号的数据部分都有一个整数的循环，所以称为循环前缀。插入循环前缀更重要的作用是可以有效消除子载波间干扰（ICI）。ICI的产生如图2－8所示。

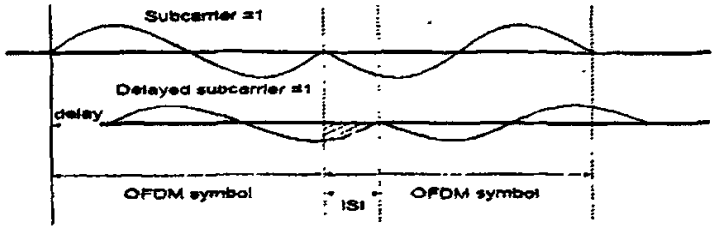

图 2-8　多径效应引起子载波间干扰 ICI

由图2－8知，在不加循环前缀的情况下，在一个OFDM符号周期里，第二子载波相对第一子载波有时延。这样使得在FFT计算周期内，第一子载波和第二子载波之间相差的周期数不是整数。这种情况下，对其中任一个子载波进行解调都会受到另一个子载波的干扰。由于多径传播使子载波之间出现相对时延，从而破坏子载波间的正交性，造成子载波间干扰（ICI）。循环前缀就是把每个OFDM符号的后T_G时间中的样点复制到OFDM符号的前面以形成前缀，在交接点没有间断。

图2-9显示了插入循环前缀的情况。

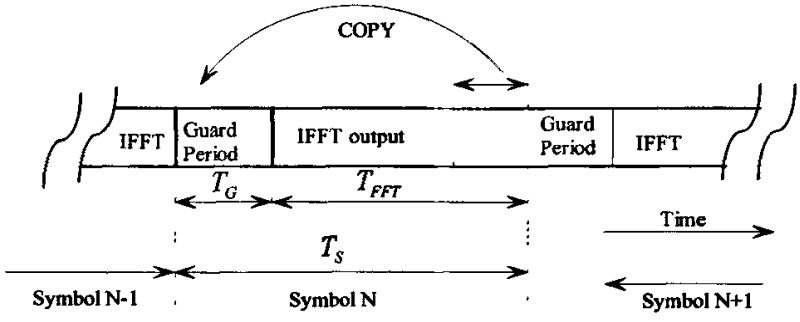

图 2-9　插入循环前缀的 OFDM 信号

整个符号长度为$T_S = T_G + T_{FFT}$，此处T_S为符号总长度，T_G为保护间隔，T_{FFT}为用于生成OFDM信号的FFT的长度。在接收端抽样开始的时刻T_X应该大于信道的最大多径时延扩展，并小于循环前缀的长度T_G。这样可以保证在子载波间有相对时移的情况下，每个FFT计算周期内子载波间相差的周期数仍为整数。这样子载波间的正交性就不会由于多径效应而受到破坏。

2.3.7 基带成形滤波器模块

为了提高频谱利用率，频谱成形技术被广泛采用。对发送信号的频谱进行加工，在消除码间干扰和实行最佳检测的前提下，压缩信号频带，提高频谱的利用率。为了有效利用频谱，要求每个码元波形的频谱扩展尽可能小。但信号的频谱分析表明，任何信号的频谱与它的时间宽度不能同时被限制在任意小的有限值以内。这要求我们在设计波形时要加以权衡，折衷考虑。

成形滤波可以在调制后对调制波以带通滤波方式完成，也可以在调制前对基带以低通滤波方式完成。虽然也可以在中频和射频实现，但由于中频和射频信号

频率较高，难以采用数字处理技术，实现难度较大，因而很少采用。

与基带模拟成形滤波器相比，基带数字成形滤波器具有高精度、高可靠性、高灵活性的优点，便于大规模集成、易于实现线性相位等特点。现代数字通信系统中，成形滤波大多在数字域进行。通过设计数字滤波器来实现成形滤波。

数字滤波器是完成信号滤波处理功能的，用有限精度算法实现的离散时间线性非时变系统，其输入是一组数字量，其输出是经过变换的另一组数字量。数字滤波器具有稳定性高、精度高、灵活性大等突出优点。随着数字技术的发展，用数字技术设计实现滤波器越来越受到人们的注意，得到广泛的应用。

本文研究了 FIR 和 IIR 滤波器的设计和实现结构，在此基础上设计实现了平方根升余弦滚降滤波器。这部分是本文的重点。

2.4 OFDM 系统的参数选择原则

在 OFDM 系统中，通常需要确定以下参数：保护间隔、符号周期、子载波的数量等。这些参数的选择取决于给定信道的带宽、时延扩展以及所要求的信息传输速率。一般按照以下步骤来确定 OFDM 系统的各参数：

（1）确定保护间隔：根据经验，一般选择保护间隔的时间长度为均方根时延扩展值的 2 到 4 倍。

（2）选择符号周期：考虑到保护间隔所带来的信息传输效率的损失和系统的实现复杂度以及系统的峰值平均功率比等因素，在实际系统中，一般选择符号周期长度至少是保护间隔长度的 5 倍。

（3）确定子载波的数量：子载波的数量可以直接利用-3dB 带宽除以子载波间隔（即去掉保护间隔之后的符号周期的倒数）得到。或者，可以利用所要求的比特速率除以每个子信道中的比特速率来确定子载波的数量。每个子信道中传输的比特速率由调制类型、编码速率以及符号速率来确定。

选择了以上参数之后，还要保证在 FFT/IFFT 运算时间内和符号间隔内的采样数量必须为整数，如不能满足要求，可适当改变以上参数，以满足采样数量的要求。

第三章　基于Verilog HDL进行FPGA设计

设计电路的经典方法是依赖于电路原理图的人工设计方法，而现在的大规模复杂电路则广泛采用基于计算机语言的现代设计方法。这种变革实际上有几方面的原因，其中最重要的一个原因就是没有一个设计工程师队伍能够用人工方法有效、全面、正确地设计和管理包含几百万个门的现代集成电路（IC）。而利用硬件描述语言（HDL），工程师们能够很容易地实现对大型复杂电路系统的设计和管理。在设计过程中，为了能抓住市场机遇，设计工程师还必须快速完成能应对市场需求的正确设计，因而即使是小型的电路设计，也提倡采用基于语言描述的设计方法。

3.1 FPGA设计

3.1.1 数字设计方法简介

随着微电子设计技术与工艺的发展，数字集成电路从电子管、晶体管、中小规模集成电路、超大规模集成电路（VLSIC）逐步发展到今天的专用集成电路（ASIC）。ASIC的出现降低了产品的生产成本，提高了系统的可靠性，缩小了设计的物理尺寸，推动了社会的数字化进程。但是ASIC因其设计周期长，改版投资大，灵活性差等缺陷制约着它的应用范围。硬件工程师希望有一种更灵活的设计方法，根据需要，在实验室就能设计、更改大规模数字逻辑，研制自己的ASIC并马上投入使用，这是提出可编程逻辑器件的基本思想。

随着微电子制造工艺的发展，可编程逻辑器件取得了长足的进步。从早期的只能存储少量数据，完成简单逻辑功能的可编程只读存储器（PROM）、紫外线可擦除只读存储器（EPROM）和电可擦除只读存储器（EEPROM），发展到能完成中大规模的数字逻辑功能的可编程阵列逻辑（PAL）和通用阵列逻辑（GAL），今天已经发展成为可以完成超大规模的复杂组合逻辑与时序逻辑的复杂可编程逻辑器件（CPLD）和现场可编程门阵列（FPGA）。随着工艺技术的发展与市场需要，超大规模、高速、低功耗的新型FPGA/CPLD不断推陈出新。新一代的FPGA甚

至集成了中央处理器（CPU）或数字处理器（DSP）内核，在一片FPGA上进行软硬件协同设计，为实现片上可编程系统（SOPC,System On Programmable Chip）提供了强大的硬件支持。

FPGA是在CPLD的基础上发展起来的新型高性能可编程逻辑器件，它一般采用SRAM工艺，也有一些专用器件采用Flash工艺或反熔丝（Anti-Fuse）工艺等。FPGA的集成度很高，其器件密度从数万系统门到数千万系统门不等，可以完成极其复杂的时序与组合逻辑电路功能，适用于高速、高密度的高端数字逻辑电路设计领域。FPGA的基本组成部分有可编程I/O单元、基本可编程逻辑单元、嵌入式块RAM、丰富的布线资源、底层嵌入功能模块和内嵌专用硬核等。

使用FPGA进行数字设计，开发过程中投资较少，研制和开发的时间较短，FPGA的开发工具功能强大且易学易用，使设计人员可以集中精力进行电路功能设计。因此FPGA被越来越多地用于通信、计算机及其他相关领域的设计中。

3.1.2 FPGA设计流程介绍

设计过程是首先是对设计项目进行整体分析设计，然后用硬件描述语言对各个功能模块进行描述，再将用硬件描述语言写的模型进行编译、综合、功能仿真、布局布线、时序仿真等步骤，对模型的结构、功能、时序等进行验证，保证其满足设计的要求。最后将设计模型下载到实际的FPGA芯片上，再进行后续的测试。FPGA设计流程图如图3－1所示。

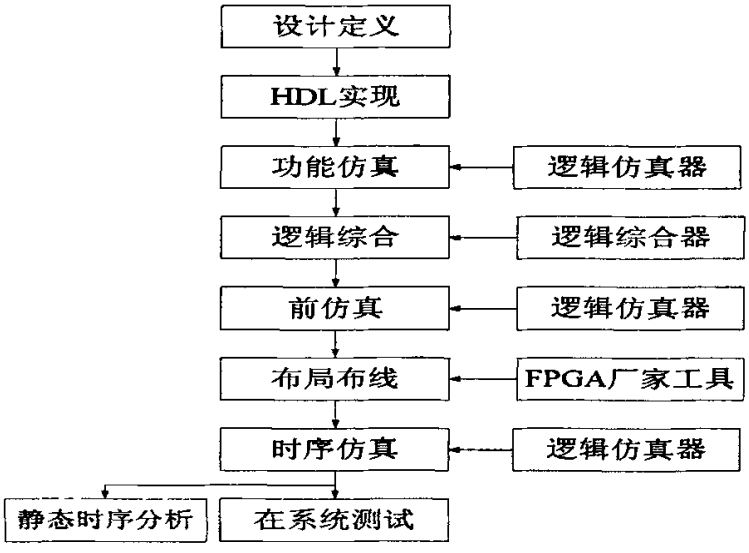

图3－1　FPGA设计流程图

在设计定义步骤中，需要写出设计规范，要对设计所要实现的功能特性进行描述，通常还要包括状态转移图、时序图和用来描述时序机的算法状态机（ASM）图表等。对一个大型的复杂设计，要将其逐渐划分成较小而且较为简单的功能单元。这样一个过程被称为自顶向下的设计方法，或者是分层设计法。

HDL（硬件描述语言）可以为需要进行划分、综合和验证的大型复杂系统设计提供一个通用框架，支持具有混合抽象级别的自顶向下设计。基于HDL的设计要比用电路原理图的设计更容易调试。封装复杂功能特性的行为描述隐藏了许多底层的门级细节，所以在隔离功能设计中出现的问题时，需要处理的信息更少。此外，如果行为描述在功能上是正确的，它就会成为后续门级电路实现的可靠标准。

综合和仿真是数字设计中关键的两个步骤，是现代自动化设计方法的关键。综合工具可以完成的工作包括：（1）检测并消除冗余逻辑，（2）查找组合反馈环路，（3）使用无关紧要条件，（4）检测出未使用状态，（5）查找和避免相等的状态，（6）进行状态分配，（7）在物理工艺满足面积和/或速度限制的条件下，综合出最佳多级逻辑实现。最后还包括了最优化和工艺映射。而HDL是利用面向综合的现代设计方法进行ASIC和FPGA设计的切入点。

仿真是指在软件环境下，验证电路的行为和设计意图是否一致。仿真和验证是一门科学，在逻辑设计领域，仿真与验证是整个设计流程中最重要、最复杂与最耗时的步骤。特别是ASIC设计中，仿真与验证投入的资源与初期逻辑设计的比重约为10：1。虽然FPGA/CPLD设计灵活，可以反复编程，这种灵活性在一定程度上可以弥补仿真与验证的不足，但是对于大型、高速或复杂的系统设计，仿真和验证仍是整个流程中的最重要的环节，目前国内外知名公司仿真验证和逻辑设计人员的配置比率超过4：1。仿真主要分为功能仿真和时序仿真。功能仿真主要在于验证电路功能是否符合设计要求，其特点是不考虑电路门延迟与路径延迟，考察重点为电路在理想环境下的行为和设计构想是否一致。时序仿真是指电路已经映射到特定的工艺环境后，综合考虑电路的路径延迟与门延迟的影响，验证电路的行为是否能够在一定时序条件下满足设计构想的过程。时序仿真的主要目的在于验证是否存在时序违规。

3.2 Verilog HDL语言介绍

基于语言的设计方法简便而且独立。随着器件物理尺寸的缩小，更密集、具有更高性能的电路仍能够通过已有的基于HDL的模型进行综合设计。

对于从许多具有设计专利的原始资料中整合知识产权（IP）来说，HDL是人们首选的一种便于使用的中间工具。有了这样一种通用的设计语言，可以方便地通过模型的整合来进行电路测试。如果想缩短设计周期，可以将电路模型分别或者放在一起来进行综合设计。有些模拟器还支持基于多种语言的混合描述。

采用HDL最显著的意义在于：能够通过基于语言的描述，对于正在进行设计的电路自动进行综合，而不用经历人工设计方法中那些费力的步骤（如利用卡诺图求解最小逻辑等）。

基于HDL的综合方法是现代工业上采用的主流设计方法。设计者可以通过构建一个软件原型或模型来验证其功能，然后利用一种综合工具自动对所设计的电路进行优化，并且可以创建一个针对物理实现的网表（netlist）。

HDL可以作为各种不同的工具平台使用，用来完成设计输入、设计验证、测试生成、故障分析和模拟、定时分析和/或校验，以及原理图的自动生成等任务。这种较宽的覆盖范围使得设计通过工具链路时，由于不再需要设计描述的翻译过程而大大提高了设计流程的工作效率。

Verilog HDL是一种硬件描述语言，用于从算法级、门级到开关级的多种抽象设计层次的数学系统建模。被建模的数学系统对象的复杂性可以介于简单的门和完整的电子数字系统之间。数字系统能够按照层次描述，并可在相同描述中显式地进行时序建模。

采用Verilog HDL输入最大的优点是其与工艺无关性。这使得工程师在功能设计，逻辑验证阶段，可以不必过多考虑门级工艺实现的具体细节。把逻辑验证与具体工艺库匹配，布线及时延计算分成不同的阶段来实现，从而减轻了人们的繁琐劳动。

Verilog HDL能形式化地抽象表示电路的行为和结构；支持逻辑设计中层次与范围的描述；可借用高级语言的精巧结构来简化电路行为的描述；具有电路仿真与验证机制以保证设计的正确性；支持电路描述由高层到低层的综合转换；硬件

描述与实现工艺无关，易于理解和设计重用。Verilog HDL是一种容易掌握的硬件描述语言，只要有C语言的编程基础，经过一段时间的实际操作就可掌握这种设计技术。

3.3 FPGA芯片介绍

现场可编程门阵列（FPGA，Field Programmable Gate Array）的出现是超大规模集成电路（VLSI）技术和计算机辅助设计（CAD）技术发展的结果。FPGA器件集成度高、体积小，具有通过用户编程实现专门应用的功能。它允许电路设计者用基于计算机的开发平台，经过设计输入、仿真、测试和校验，直至达到预期的结果。使用FPGA器件可以大大缩短系统的研制周期，减少资金投入。更吸引人的是，采用FPGA器件可以将原来的电路板级产品集成为芯片级产品从而降低了功耗，提高可靠性，同时还可以很方便地对设计进行在线修改。FPGA器件成为研制开发的理想器件，特别适合于产品的样机开发和小批量的生产。如今，FPGA器件广泛应用于通信、自动控制、信息处理等诸多领域，越来越多的电子设计人员在使用FPGA，熟练掌握FPGA设计技术已经是对电子设计工程师的基本要求。

FPGA的基本组成部分有可编程I/O单元、基本可编程逻辑单元、嵌入式块RAM、丰富的布线资源、底层嵌入功能模块和内嵌专用硬核等。

可编程I/O单元是芯片与外界电路的接口部分，完成不同电气特性下对输入/输出信号的驱动与匹配需要。值得一提的是，随着ASIC工艺的飞速发展，目前可编程I/O支持的最高频率越来越高，一些高端FPGA通过DDR寄存器技术，甚至可以支持高达2Gbit/s的数据速率。

基本可编程逻辑单元是可编程逻辑的主体，可以根据设计灵活地改变其内部连接与配置，完成不同的逻辑功能。FPGA一般是基于SRAM工艺，其基本可编程逻辑单元几乎都是由查找表（LUT）和寄存器（Register）组成的。FPGA内部查找表一般为4输入，查找表一般完成纯组合逻辑功能。FPGA内部寄存器结构相当灵活，可以配置成为带同步/异步复位或置位、时钟使能的触发器，也可以配置成锁存器。FPGA一般依赖寄存器完成同步时序逻辑设计。学习底层配置单元的LUT和Register比率的一个重要意义在于器件选型和规模估算（一般来说两者比

率为1：1)。简单的用系统门权衡基本可编程逻辑单元的数量是不准确的，比较简单科学的方法是用器件的LUT或Register的数量衡量。

器件选型是一个综合性问题，需要将设计的需求、成本压力、规模、速度等级、时钟资源、I/O特性、封装、专用功能模块等诸多因素综合考虑。

FPGA内嵌的块RAM一般可以灵活配置为单端口RAM(SPRAM)、双端口RAM(DPRAM)、内容地址存储器（CAM)、FIFO等常用存储结构。

布线资源连通FPGA内部所有单元，连线的长度和工艺决定着信号在连线上的驱动能力和传输速度。由一个设计所实现的电路性能取决于芯片上单元电路的物理布局与布线，以及底层器件的特性等。互连延迟对于0.18μm以下的亚微米设计的性能很关键。

FPGA的特点包括：

（1）规模越来越大。单片逻辑门数已超过千万。芯片规模越大所能实现的功能就越强，同时也更适于实现片上系统（SOC)。

（2）开发过程投资小。FPGA芯片在出厂之前都做过严格的测试，设计灵活，发现错误时可直接更改设计，减少了投片风险，节省了许多潜在的花费。不但许多复杂系统使用FPGA完成，甚至设计ASIC时也要把实现FPGA功能样机作为必要的步骤。

（3）某些领域的相关标准协议发展的太快，设计ASIC跟不上技术的更新速度，只能依靠FPGA完成系统的研制与开发。

（4）FPGA开发工具智能化，功能强大。这些工具易学易用，可以使设计人员更能集中精力进行电路功能设计。

（5）新型FPGA内嵌CPU或DSP内核，支持软硬件协同设计，可以作为片上可编程系统（SOPC）的硬件平台。

下一代可编程逻辑器件的四大发展趋势包括：

（1）先进工艺。在工艺上第一个显著的进步是90nmCOMS芯片加工工艺被广泛地应用于FPGA产品。这意味着器件密度提高、工作频率提高和器件价格降低。此外还使用了低介电常数（Low K Dielectric）和全铜层（Copper Metal）等工艺。

（2）内嵌处理器内核。FPGA内部嵌入CPU或DSP等处理器，使FPGA在一定程度上具备了实现软硬件联合设计的能力，FPGA正逐步成为SOC的高效设

计平台。

（3）硬核与结构化 ASIC。高端 FPGA 的另一个重要特点是集成了功能丰富的硬知识产权核（Hard IP Core）。这些 Hard IP Core 一般完成高速、复杂的设计标准。通过这些 Hard IP Core，FPGA 正在逐步进入一些过去只有 ASIC 才能完成的领域。

（4）低成本器件。在以市场为导向的电子产品开发中，低成本的重要性不言而喻。

本文选用的开发板是 Altera 的 Stratix 系列的 EP1S80 DSP 开发板。如图 3－2 所示。

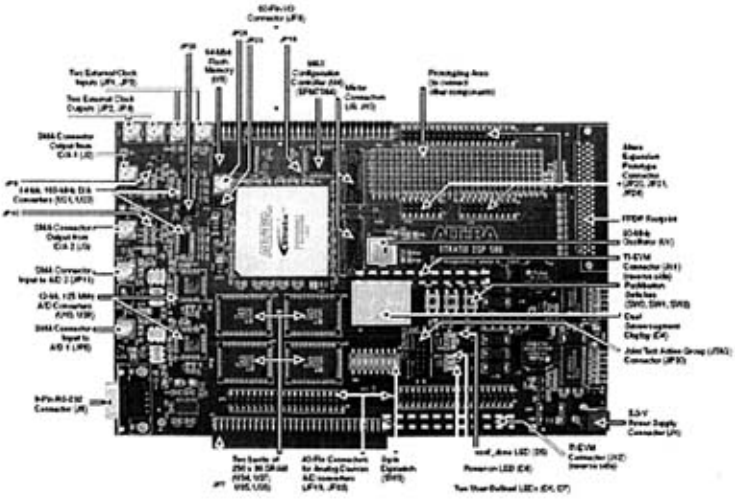

图 3－2　EP1S80 DSP 开发板的组成

Stratix EP1S80 DSP 开发板除包含 Stratix EP1S80 FPGA 外，还包括众多组成部分，主要元件有：两个 12bit，125MHz A/D 转换器；两个 14bit，165MHz D/A 转换器；2M 字节同步 SRAM，64Mbit 闪存；RS232 接口；晶体振荡器等。

Stratix EP1S80 是最新一代基于 SRAM 工艺的大规模 FPGA，集成硬件乘加器，芯片内部结构比 Altera 以前的产品有很大变化。EP1S80 能提供 79040 个逻辑单元，内部存储器超过 7Mb，与其他竞争组件相比，它能提供更多的内存空间，I/O 管脚和更强大的数字信号处理能力，是目前 0.13μm 工业级容量最高的 FPGA 器件。它具有 5 种不同的速度，能提供高性能数字信号处理（DSP）速度 300MHz。专门为恶劣环境设计复杂可编程电子系统的设计师可用此大容量器件提高系统性能，同时使系统设计更平滑。

EP1S80采用了1.5V内核，0.13μm全铜工艺，由Quartus II 2.0以上版本软件支持，可以重复编程，通过JTAG接口或者EPROM加载程序，内部有DSP模块、PLL、大带宽高速I/O接口和大容量存储模块，主要内部资源包括：逻辑单元79040个，M512 RAM模块767个，模块9个M4K RAM 364个，RAM总容量7,9427,520bit，DSP模块22个，嵌入式乘法器（9*9）176个，锁相环12个，用户最多可用引脚1234个。其中，512bit M512模块（512×1bit到32×18bit）：512位模块加上校验，可用于接口速率适配的FIFO。4Kbit M4K模块（4096×1bit到128×36bit）：4K位模块加上校验，可用于小型数据块存储和多通道I/O协议。512Kbit MegaRAM模块（64K×9bit到4K×144bit）：512K位RAM加上校验，可用于存储大型数据块或者Nios TM嵌入式处理器软核等。4Kbit M4K模块和512Kbit MegaRAM模块支持完全的双端口模式。所有存储资源分布在整个器件中，设计者可根据设计的存储器类型和容量大小，灵活选择不同参数，配置成特定存储容量的RAM、DPRAM、FIFO等特殊模块。

第四章　RS码编码器和译码器的设计及实现

RS（Reed－Solomon）码是一类有很强纠错能力的多进制BCH码。它首先由里德（Reed）和索洛蒙（Solomon）应用MS多项式于1960年构造出来。不过，用MS多项式构造的RS码是非系统码，而用BCH码构造方法能产生系统码。本文先对BCH码的原理做简要介绍，然后对RS码编码器和译码器的设计及实现做详细说明。

4.1 BCH码简介

4.1.1 BCH码编码原理

1959年由霍昆格姆（Hocquenghem），1960年由博斯（Bose）和雷－查德胡里（Ray－Chaudhuri）分别提出了能纠正多个随机错误的循环码——BCH 码的构造方法。BCH 码是一类很好的线性纠错码类，纠错能力很强，特别在短和中等码长下，其性能很接近于理论值，并且构造方便，编码简单。特别是它具有严格的代数结构，因此它在编码理论中起着重要作用。BCH码是迄今为止研究得最为详尽，分析得最为透彻，取得的成功也最多的码类之一。

1960年彼德逊（Peterson）从理论上解决了二进制BCH码的译码算法，奠定了BCH码译码的理论基础。格林斯坦（Gorensten）和齐勒尔（Zierler）把它推广到多进制。1966年伯利坎普（Berlekamp）利用迭代算法对BCH码译码，大大加快了译码速度，从实际上解决了BCH码的译码问题。

定义：给定任一有限域$GF(q)$及其扩域$GF(q^m)$，其中q是素数或素数的幂，m为某一正整数。若码元取自$GF(q)$上的一循环码，它的生成多项式$g(x)$的根集合R中含有以下$\delta-1$个连续根

$$R \supseteq \{\alpha^{m_0}, \alpha^{m_0+1}, \cdots, \alpha^{m_0+\delta-2}\}$$

时，则由$g(x)$生成的循环码成为q进制BCH码。

其中，α是$GF(q^m)$域中的n级元素，m_0是任意整数，对于最常见的情况m_0

=0 或 1。设 $m_i(x)$ 和 e_i 分别是 α^{m_0+i}（i=0，1，…，δ−2）元素的最小多项式和级，则 BCH 码的生成多项式和码长分别是：

$$g(x)=LCM(m_0(x),m_1(x),\cdots m_{\delta-2}(x))$$

$$n=LCM(e_0,e_1,\cdots,e_{\delta-2})$$

如果生成多项式 $g(x)$ 的根中，有一个 $GF(q^m)$ 中的本原域元素，则 $n=q^m-1$,称这种 BCH 码为本原 BCH 码；否则，称为非本原 BCH 码。$GF(q^m)$ 中元素的级一定是 q^m-1 的因子，所以非本原 BCH 码的码长也一定是 q^m-1 的因子。

在实际中应用得较多的是码元取自 $GF(2)$ 中的二进制 BCH 码。由 BCH 码的定义可知，对任一个正整数 m，一定可以构造出如下的二进制码：

取 m_0=1，δ=2t+1，又设 α 是 $GF(2^m)$ 的本原域元素，则由 BCH 码的定义可知：若以 α，α^2，…，α^{2t} 为根，则二进制 BCH 码的生成多项式

$$g(x)=LCM(m_1(x),m_2(x),\cdots m_{2t}(x))$$

式中，$m_i(x)$ 是 α^i（1<=i<=2t）的最小多项式，该 BCH 码一定能纠正 t 个错误。

4.1.2 BCH 码的时域迭代译码

1960 年彼得逊奠定了二进制 BCH 码译码的理论基础，稍后戈雷（Gore）和齐勒尔（Zierler）推广到多进制情况。1965 年福尼（Forney）解决了 BCH 码的纠错纠删码。1966 年伯利坎普提出了迭代译码算法，节省了计算量，加快了计算速度，因而从实际上解决了 BCH 码的译码问题。此后，很多研究人员又提出了新的译码算法，如欧几里德、连分式等译码算法。1978 年勃莱哈特用数字信号处理技术中常用的频谱方法，基于码的 MS 多项式提出了频域译码。但目前用得最普遍、实际应用中最重要的还是时域中的迭代译码算法。本文对 RS 码的译码采用时域迭代译码算法。

BCH 码的译码与线性分组码的译码步骤相同，分为三步：第一步是由接收到的 $R(x)$ 计算出伴随式 S；第二步由伴随式找出错误图样 $E'(x)$；第三步由 $R(x)$ − $E'(x)$ 得到最可能发送的码字 $C'(x)$，完成译码。

设$GF(q)$上的$[n, k, d]$BCH 码以α^{m_0}，α^{m_0+1}，…，α^{m_0+d-2}为根，则它的生成多项式

$$g(x)=(x-\alpha^{m_0})(x-\alpha^{m_0+1})\cdots(x-\alpha^{m_0+d-2}) \qquad d=2t+1，\ \alpha\in GF(q^m) \tag{4-1}$$

发送的码字$C(x)=q(x)g(x)$，接收的n重为$R(x)=C(x)+E(x)$，设错误图样为

$$E(x)=e_{n-1}x^{n-1}+e_{n-2}x^{n-2}+\cdots+e_1x+e_0 \tag{4-2}$$

若信道产生t个错误，则

$$E(x)=Y_tx^{l_t}+Y_{t-1}x^{l_{t-1}}+\cdots+Y_1x^{l_1}=\sum_{i=1}^{t}Y_ix^{l_i} \qquad Y_i\in GF(q) \tag{4-3}$$

x^{l_i}称为错误位置数，说明错误发生在$R(x)$中的第$n-l_i$位，错误值是Y_i。如果$R(x)$中有γ个错误，则$E(x)$共有γ项$Y_ix^{l_i}$，$i=1,2,\cdots,\gamma$。

由伴随式定义可知：

$$S^T \triangleq H\bullet R^T=H\bullet E^T$$

$$=\begin{bmatrix}(\alpha^{m_0})^{n-1} & (\alpha^{m_0})^{n-2} & \cdots & \alpha^{m_0} & 1\\ (\alpha^{m_0+1})^{n-1} & (\alpha^{m_0})^{n-2} & \cdots & \alpha^{m_0+1} & 1\\ \vdots & \vdots & & \vdots & \vdots\\ (\alpha^{m_0+2t-1})^{n-1} & (\alpha^{m_0})^{n-2} & \cdots & \alpha^{m_0+2t-1} & 1\end{bmatrix}\begin{bmatrix}0\\ \vdots\\ 0\\ Y_t\\ \vdots\\ \vdots\\ 0\\ 0\\ \vdots\\ Y_1\\ 0\\ \vdots\\ 0\end{bmatrix}$$

$$=\begin{bmatrix}Y_1(\alpha^{m_0})^{l_{11}}+Y_2(\alpha^{m_0})^{l_{12}}+\cdots+Y_t(\alpha^{m_0})^{l_{1t}}\\ Y_1(\alpha^{m_0+1})^{l_{11}}+Y_2(\alpha^{m_0+1})^{l_{12}}+\cdots+Y_t(\alpha^{m_0+1})^{l_{1t}}\\ \vdots\\ Y_1(\alpha^{m_0+2t-1})^{l_{11}}+Y_2(\alpha^{m_0+2t-1})^{l_{12}}+\cdots+Y_t(\alpha^{m_0+2t-1})^{l_{1t}}\end{bmatrix}$$

$$= \begin{bmatrix} s_{m_0} = R(\alpha^{m_0}) = E(\alpha^{m_0}) \\ s_{m_0+1} = R(\alpha^{m_0+1}) = E(\alpha^{m_0+1}) \\ \vdots \\ s_{m_0+2t-1} = R(\alpha^{m_0+2t-1}) = E(\alpha^{m_0+2t-1}) \end{bmatrix} \tag{4-4}$$

式中，

$$s_j = \sum_{i=1}^{t} Y_i (\alpha^j)^{li} = R(\alpha^j) \qquad j = m_0, m_0+1, \cdots, m_0+2t-1 \tag{4-5}$$

通常情况下取 $m_0 = 1$，则

$$\begin{aligned} s_1 &= Y_1 x_1 + Y_2 x_2 + \cdots + Y_t x_t = \sum_{k=1}^{t} Y_k x_k \\ s_2 &= Y_1 x_1^2 + Y_2 x_2^2 + \cdots + Y_t x_t^2 = \sum_{k=1}^{t} Y_k x_k^2 \\ &\vdots \\ s_{2t} &= Y_1 x_1^{2t} + Y_2 x_2^{2t} + \cdots + Y_t x_t^{2t} = \sum_{k=1}^{t} Y_k x_k^{2t} \end{aligned} \tag{4-6}$$

称式（4—6）中的 s_j 为 x 的加权幂和对称函数。通过上述 $2t$ 个方程求出 $2t$ 个未知数错误位置 x_i 和错误值 Y_i（i=1，2，…，t），即可完成译码。直接求解非常困难，因此分两步进行，先求错误位置数 x_i，再求解错误值 Y_i。为此引入错误位置多项式

$$\begin{aligned} \sigma(x) &= (1-x_1 x)(1-x_2 x)\cdots(1-x_t x) = \prod_{i=1}^{t}(1-x_i x) \\ &= 1-(x_1+x_2+\cdots+x_t)x+(x_1x_2+x_1x_3+\cdots+x_{t-1}x_t)x^2 \\ &\quad -\cdots+(-1)^t x_1 x_2 \cdots x_t \end{aligned} \tag{4-7}$$

令

$$\begin{aligned} \sigma_1 &= -(x_1+x_2+\cdots+x_t) \\ \sigma_2 &= (x_1x_2+x_1x_3+\cdots+x_{t-1}x_t) \\ &\vdots \\ \sigma_t &= (-1)^t x_1 x_2 \cdots x_t \end{aligned} \tag{4-8}$$

则上式表示为

$$\sigma(x) = 1+\sigma_1 x+\sigma_2 x^2+\cdots+\sigma_t x^t = \prod_{i=1}^{t}(1-x_i x) \tag{4-9}$$

若 x_k^{-1} 为错误位置，则

$$\sigma(x_k^{-1})=1+\sigma_1 x_k^{-1}+\sigma_2 x_k^{-2}+\cdots+\sigma_t x_k^{-t}=0 \tag{4—10}$$

两边乘以 x_k^t，则

$$x_k^t+\sigma_1 x_k^{t-1}+\sigma_2 x_k^{t-2}+\cdots+\sigma_t=0 \qquad \text{k=1，2，…，t} \tag{4—11}$$

上式两边再乘以 $Y_k x_k^j$，$j=m_0,m_0+1,\cdots,m_0+t-1$，则

$$Y_k x_k^{j+t}+\sigma_1 Y_k x_k^{j+t-1}+\cdots+\sigma_t Y_k x_k^j=0 \qquad \text{k=1，2，…，t} \tag{4—12}$$

对 k 求和得

$$\sum_{k=1}^{t} Y_k x_k^{j+t}+\sigma_1\sum_{k=1}^{t} Y_k x_k^{j+t-1}+\cdots+\sigma_t\sum_{k=1}^{t} Y_k x_k^j=0 \qquad \text{k=1，2，…，t} \tag{4—13}$$

由式（4—6）s_j 的定义知上式成为

$$s_{j+t}+\sigma_1 s_{j+t-1}+\sigma_2 s_{j+t-2}+\cdots+\sigma_t s_j=0 \quad j=m_0,m_0+1,\cdots,m_0+t-1 \tag{4—14}$$

把上式展开，则

$$\begin{aligned}
&s_{m_0+t}+\sigma_1 s_{m_0+t-1}+\sigma_2 s_{m_0+t-2}+\cdots+\sigma_t s_{m_0}=0\\
&s_{m_0+1+t}+\sigma_1 s_{m_0+t}+\sigma_2 s_{m_0+t-1}+\cdots+\sigma_t s_{m_0+1}=0\\
&\qquad\vdots\\
&s_{m_0+2t-1}+\sigma_1 s_{m_0+2t-2}+\sigma_2 s_{m_0+2t-3}+\cdots+\sigma_t s_{m_0+t-1}=0
\end{aligned} \tag{4—15}$$

式（4—15）是一组线性方程，有 t 个方程和 t 个未知数，该方程组有解的充要条件是其系数矩阵满秩。通过求解该方程组，即可得到错误位置多项式 $\sigma(x)$。

得到 $\sigma(x)$ 后，通过求解 $\sigma(x)$ 的根就可以确定 $R(x)$ 中的哪几位产生了错误。钱搜索过程是求解 $\sigma(x)$ 的根的一个实用方法。设

$$R(x)=r_{n-1}x^{n-1}+r_{n-2}x^{n-2}+\cdots+r_1 x+r_0 \tag{4—16}$$

为了要检验第一位 r_{n-1} 是否错误，相当于译码器要确定 α^{n-1} 是否是错误位置数，即要检验 $\alpha^{-(n-1)}$ 是否是 $\sigma(x)$ 的根。若 $\alpha^{-(n-1)}=\alpha$ 是 $\sigma(x)$ 的根，则

$$\sigma(\alpha^{-(n-1)})=\sigma(\alpha)=1+\sigma_1\alpha+\sigma_2\alpha^2+\cdots+\sigma_t\alpha^t=0 \tag{4—17}$$

根据这个准则就可以判定 r_{n-1} 是否发生错误。这样依次对每一个 r_{n-l} $(l=1,2,\cdots,n)$ 进

行检验，就可以求得$\sigma(x)$的根。这个过程就称为钱搜索。

求得错误位置数后，接着求解错误值。这一步对二进制 BCH 码来说可以省略，但对多进制 BCH 码是必需的。

由以上讨论可知，BCH 码的译码过程分为五步：

（1） 由接收到的$R(x)$，求得$s_j=\sum_i Y_i x_i^j$，$j=m_0,m_0+1,\cdots,m_0+2t-1$。

（2） 由s_i求出错误位置多项式$\sigma(x)$。

（3） 用钱搜索解出$\sigma(x)$的根，得到错误位置数。

（4） 由错误位置数求得错误值，得到错误图样$E'(x)$。

（5） $R(x)-E'(x)=C'(x)$，完成纠错过程。

4.2 RS 码编码器设计和实现

4.2.1 RS 码编码器结构

RS 码是一类有很强纠错能力的多进制 BCH 码，所以可以利用 BCH 码的构造方法设计 RS 码编码器。对于线性分组码来说，在其他指标相同的情况下，非系统码和系统码的纠错能力相同，但在译码时系统码可直接读出信息元，非系统码还要增加一个信息元的转换电路。本文对不同设计方案进行比较，采用$(n-k)$级系统码方案来实现 RS 码编码器。

RS(255,239)码是在有限域$GF(2^8)$上运算得到的。它的编码参数如下：

码长n=255，信息位个数k=239，

校验位$r=n-k$=16，纠错能力t=8，码距d=17

本原多项式$p(x)=x^8+x^4+x^3+x^2+1$

生成多项式$g(x)=(x-\alpha^0)(x-\alpha^1)(x-\alpha^2)\cdots(x-\alpha^{15})$

根据 RS 码的每一个码字$C(x)$都是生成多项式$g(x)$的倍式，RS 码的编码电路通常可分为两大类：k 级编码器和 $n-k$ 级编码器。k 级编码器是根据校验多项式

$h(x)=h_k x^k+h_{k-1}x^{k-1}+\cdots+h_1 x+h_0$ 构造的。经过必要的推导，得到它的电路结构如图 4－1 所示。

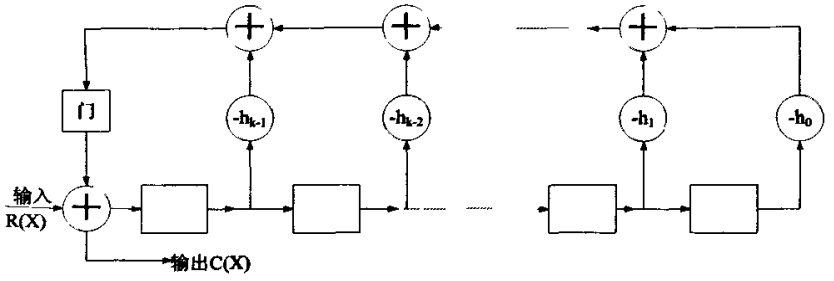

图 4－1　RS 码 k 级系统码编码器

由于 RS（255，239）码的参数 k 的值很大，如果用 k 级编码器来实现，需要非常多的延迟单元、有限域乘法器和有限域加法器，实现不方便，并且要占用大量的资源。这种编码器适合于 k 值比较小的 RS 码，并且可以得到系统码。

$n-k$ 级编码器有两种实现结构：一种是生成多项式 $g(x)$ 的乘法电路。由线性分组码理论可知，$C(x)=m(x)g(x)$，即信息多项式 $m(x)$ 与生成多项式 $g(x)$ 直接相乘就可以得到需要的码字。不过这种编码器产生的是非系统码。它的电路结构如图 4－2 所示。

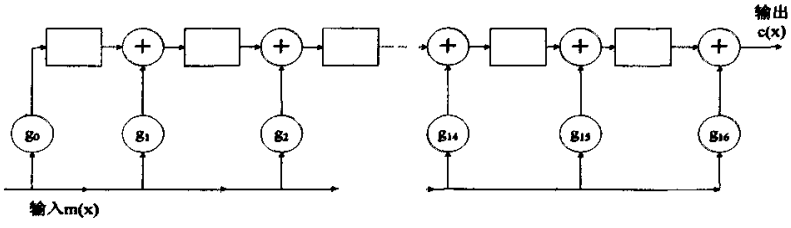

图 4－2　RS 码 n－k 级非系统码编码器

这种编码器的结构简单而且容易理解，与 k 级编码器相比，占用的资源很少，并且编码速度比较快。但它产生的是非系统码，在译码时需要额外的信息元转换电路，并会增加译码时延。

另一种 $n-k$ 级编码器是 g（x）的除法电路。由线性分组码理论可知，

$$m(x)x^{n-k}=q(x)g(x)+r(x)$$

$$C(x)=m(x)x^{n-k}-r(x)$$

即先将信息组 $m(x)$ 乘以 x^{n-k}，然后再用 $g(x)$ 除，求得相应的余式 $r(x)$。余式

$r(x)$ 的系数作为校验位，放在信息码组后一起输出。$n-k$ 系统码编码器就是乘 x^{n-k} 除 $g(x)$ 的电路，其电路结构如图 4－3 所示。该结构易于理解，并可获得系统码。

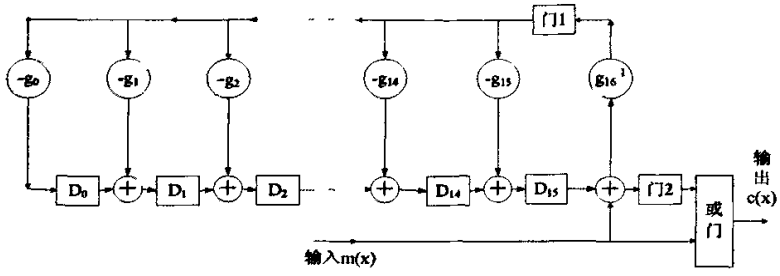

图 4－3　RS 码 n－k 级系统码编码器

该电路的工作过程如下：

（1）将所有移位寄存器清零，门 1 开，门 2 关，然后送入信息码组并进行移位。信息码组一边经过或门输出，一边乘以 x^{n-k} 后进入 g(x)除法电路。

（2）k 次移位后信息码组全部送入电路，完成除法运算，移位寄存器内保留的就是余式 $r(x)$的系数。

（3）信息码组暂停输入，门 1 关，门 2 开，经过 $n-k$ 次移位后，所有校验元依次输出，与原先长为 k 的信息码组一起构成一个长为 n 的码字。

本文所设计的编码器就是基于这种结构。由于 RS（255，239）码是建立在 $GF(2^8)$ 上，所以电路中的计算单元、移位寄存器和数据通路都是 8 位并行的。

4.2.2 基于 Verilog HDL 设计 RS 码编码器

由图 4－3 可知，RS 码编码器主要包括了有限域加法模块、有限域乘法模块和 16 级移位寄存器结构。这些模块也大量使用于 RS 码译码器的实现中。

有限域元素间的加法运算比较简单，用 2 进制形式来表示有限域中的元素，将加数和被加数按对应位进行异或运算就可以实现有限域加法。用 Verilog HDL 进行描述，如下：

```
assign sum＝a[7:0]^b[7:0]
```

有限域乘法器的设计是 RS 编码器实现的关键。可以把有限域 $GF(p^m)$ 中的每一个元素都用它的一组自然基底{1，α，α^2，…，α^{m-1}}来表示。设 α 是本原多项式 $p(x)$ 的根，则可得 $\alpha^8=\alpha^4+\alpha^3+\alpha^2+1$。按此方法可以求得有限域 $GF(2^8)$ 中的全

部元素，表 4—1 列出了 $p(x)$ 的 $GF(2^8)$ 域的部分元素。

表 4—1　$p(x)=x^8+x^4+x^3+x^2+1$ 的 $GF(2^8)$ 域的部分元素

幂	多项式表示	十进制值	二进制值
α	x	2	0000_0010
α^2	x^2	4	0000_0100
α^3	x^3	8	0000_1000
α^4	x^4	16	0001_0000
α^5	x^5	32	0010_0000
α^6	x^6	64	0100_0000
α^7	x^7	128	1000_0000
α^8	$x^4+x^3+x^2+1$	29	0001_1101
α^9	$x^5+x^4+x^3+x$	58	0011_1010
α^{10}	$x^6+x^5+x^4+x^2$	116	0111_0100
α^{11}	$x^7+x^6+x^5+x^3$	224	1110_1000
α^{12}	$x^7+x^6+x^3+x^2+1$	205	1100_1101
α^{13}	x^7+x^2+x+1	135	1000_0111
α^{14}	x^4+x+1	19	0001_0011

这样，可以计算出生成多项式，其中 $\alpha=02_{HEX}$。

$$g(x)=(x-\alpha^0)(x-\alpha^1)(x-\alpha^2)\cdots(x-\alpha^{15})$$

$$=x^{16}+\alpha^{120}x^{15}+\alpha^{104}x^{14}+\alpha^{107}x^{13}+\alpha^{109}x^{12}+\alpha^{102}x^{11}+\alpha^{161}x^{10}+\alpha^{76}x^9+\alpha^3x^8$$
$$+\alpha^{91}x^7+\alpha^{191}x^6+\alpha^{147}x^5+\alpha^{169}x^4+\alpha^{182}x^3+\alpha^{194}x^2+\alpha^{225}x+\alpha^{120}$$
$$=x^{16}+59x^{15}+13x^{14}+104x^{13}+189x^{12}+68x^{11}+209x^{10}+30x^9+8x^8$$
$$+163x^7+65x^6+41x^5+229x^4+98x^3+50x^2+36x+59$$

一种设计有限域乘法器的方法是利用查找表。对于 RS（255，239）码，需要 16 个查找表，一个查找表有 16 个输入，在有限域 $GF(2^8)$ 中每个元素占用 8bit，每一个查找表就需要 64KB 的存储空间，资源占用太多。利用 $g(x)$ 中系数固定的特点，本文将有限域乘法器设计成一个乘数固定的乘法器，大大减少所需占用的资源，原理易于理解，实现起来简单方便。

例如，对于系数 $g_{15}=(59)_{10}=(111011)_2=\alpha^5+\alpha^4+\alpha^3+\alpha+1$，

设输入 $B=b_7\alpha^7+b_6\alpha^6+b_5\alpha^5+b_4\alpha^4+b_3\alpha^3+b_2\alpha^2+b_1\alpha+b_0$。

令 $C=B\times g_{15}=c_7\alpha^7+c_6\alpha^6+c_5\alpha^5+c_4\alpha^4+c_3\alpha^3+c_2\alpha^2+c_1\alpha+c_0$，则可以得到一组异或方程来表示 C 的系数，如式（4－18）所示，其中的加号表示异或运算。将有限域乘法运算转换为异或运算，提高了运算速度。

$$\begin{aligned}
c_7&=b_7+b_4+b_3+b_2\\
c_6&=b_5+b_3+b_2+b_1\\
c_5&=b_7+b_2+b_1+b_0\\
c_4&=b_7+b_6+b_4+b_1+b_0\\
c_3&=b_7+b_2+b_0\\
c_2&=b_5+b_4+b_3+b_2+b_1\\
c_1&=b_6+b_5+b_4+b_1+b_0\\
c_0&=b_5+b_4+b_3+b_0
\end{aligned}\qquad(4-18)$$

按照同样的计算方法可以得到其它 15 个有限域乘法器。在使用 Verilog HDL 进行设计时，可以将每一个乘法器封装成一个函数，然后在编码器模块中进行调用。这样设计出的编码器程序结构简单，易于理解，不容易出错。基于 Verilog HDL 的系数 g_{15} 的乘法器函数模型如下所示：

```
function [7:0] multiplication_g15;
input      [7:0] data_in;
reg        [7:0] data_out;
begin
data_out[7]=data_in[7]^data_in[4]^data_in[3]^data_in[2];
data_out[6]=data_in[5]^data_in[3]^data_in[2]^data_in[1];
data_out[5]=data_in[7]^data_in[2]^data_in[1]^data_in[0];
data_out[4]=data_in[7]^data_in[6]^data_in[4]^data_in[1]^data_in[0];
data_out[3]=data_in[7]^data_in[2]^data_in[0];
data_out[2]=data_in[5]^data_in[4]^data_in[3]^data_in[2]^data_in[1];
data_out[1]=data_in[6]^data_in[5]^data_in[4]^data_in[1]^data_in[0];
data_out[0]=data_in[5]^data_in[4]^data_in[3]^data_in[0];
multiplication_g15=data_out;
```

```
end
endfunction
```

4.2.3 设计仿真结果

本文设计的RS（255，239）编码器使用Verilog HDL对整个模型进行描述，以ALTERA公司的FPGA芯片EP1S80B956C6为硬件平台进行实现。利用Quartus II对设计进行综合，结果如表4－2所示：

表4－2　RS（255，239）模型综合结果

综合结果	Total logic element	Total pins	最高时钟频率
RS(255, 239)	180	18	190MHz

利用Quartus II对设计进行仿真，结果如图4－4、图4－5所示。图4－4显示了仿真初期的仿真波形，出入数据是从00000001到11101111的239个符号，可以看到信息码组依次输出。图4－5清楚地显示了校验元输出的情况，在输出校验元的过程中，信息码组停止输入。输出的校验元与利用MATLAB计算出的正确数据一致。

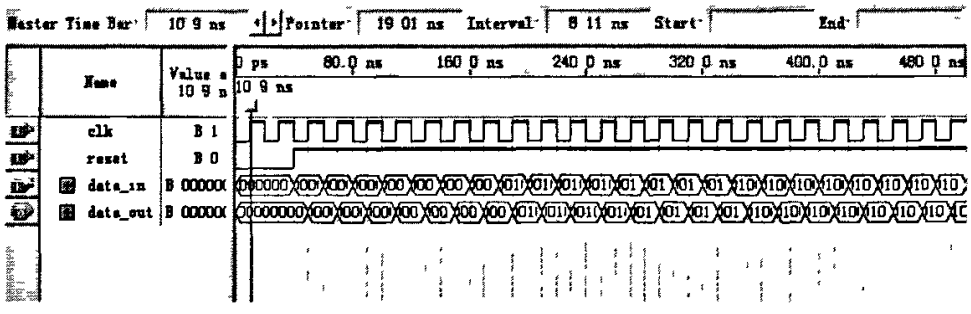

图4－4　仿真初期的仿真波形

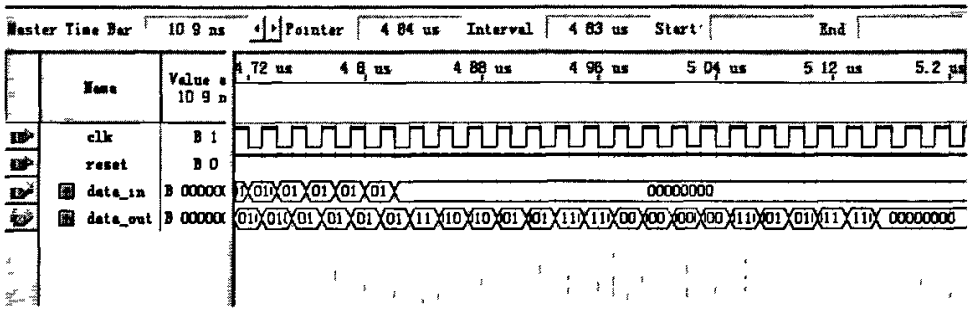

图4－5　仿真末期校验元输出仿真波形

4.3 RS码译码器设计

根据4.1节的分析可知，RS码的译码过程包括求伴随式、求错误位置多项式、求错误位置数、求错误值和译码输出（利用接收多项式减去错误图样后得到译码输出结果）这五个部分。RS码译码器整体结构如图4—6所示。

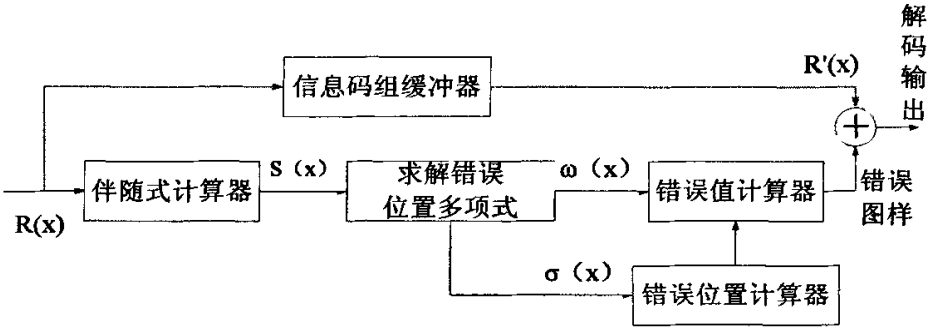

图4—6　RS码译码器结构框图

4.3.1 伴随式计算器设计

译码的第一个步骤是用译码器接收到的$R(x)$来计算伴随式多项式$S(x)$。$R(x)$是由发送的码字$C(x)$叠加噪声$e(x)$后传送到译码器接收端的码字。其中

$$R(x)=r_{n-1}x^{n-1}+r_{n-2}x^{n-2}+\cdots+r_1x+r_0 \tag{4—19}$$

$$S(x)=s_{2t-1}x^{2t-1}+s_{2t-2}x^{2t-2}+\cdots+s_1x+s_0 \tag{4—20}$$

由于

$$s_i=R(\alpha^i)=r_0+r_1\alpha^i+r_2(\alpha^i)^2+\cdots+r_{n-1}(\alpha^i)^{n-1},\ i=0,1,\cdots,2t-1 \tag{4—21}$$

可将式（4—21）改写为

$$s_i=\{\cdots[(r_{n-1}\alpha^i+r_{n-2})\alpha^i+r_{n-3}]\alpha^i+\cdots+r_1)\alpha^i+r_0 \tag{4—22}$$

这样可采用迭代的方法计算出伴随式，计算s_i的单元电路如图4—7所示。

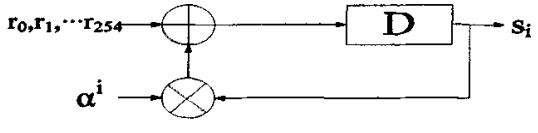

图4—7 计算伴随式s_i的单元电路

各个伴随式的计算是相互独立，使用 16 个这样的单元电路进行并行计算，经过 255 个时钟周期就可以得到所需的 16 个伴随式 S_0，S_1，…，S_{15}。针对每个具体的 s_i 而言，其计算电路中的乘数 α^i 是固定的，可根据这一点可对电路中的有限域乘法器进行优化。

4.3.2 关键方程计算电路设计

接下来，利用得到的伴随式多项式 $S(x)$ 求解错误位置多项式 $\sigma(x)$。决定译码器复杂度和速度的主要因素在于这一步。简化和加快 $\sigma(x)$ 计算电路是 RS 码译码器设计的关键。根据 4.1 节的分析，由式（4－6）直接求解在接收的码字中发生错误的位置和相应的错误值是非常困难的。所以引入错误位置多项式

$$\sigma(x)=1+\sigma_1 x+\sigma_2 x^2+\cdots+\sigma_t x^t=\prod_{i=1}^{t}(1-x_i x) \tag{4－23}$$

假设第 k 个错误位置 $x=x_k^{-1}$，则 $\sigma(x_k^{-1})=0$。因此，求错误位置就是求解错误位置多项式 $\sigma(x)$ 的根。做乘积 $S(x)\sigma(x)$，用 $\omega(x)=1+\omega_1(x)+\omega_2(x)+\cdots$ 来表示该乘积，得到方程

$$S(x)\sigma(x)=\omega(x) \pmod{x^{2t+1}} \tag{4－24}$$

式（4－24）称为求解 $\sigma(x)$ 的关键方程。其中的 $\omega(x)$ 称为错误值多项式，在求解错误值时将会用到。

1966 年伯利坎普提出了基于关键方程，由 $S(x)$ 求 $\sigma(x)$ 的迭代译码算法，极大的加快了求 $\sigma(x)$ 的速度，易于用计算机完成译码，因而从工程上解决了 RS 码的译码问题。1969 年梅西指出了迭代译码算法与序列的最短线性移位寄存器综合之间的关系，并对算法进行了简化，之后就将这种算法称为 BM 迭代译码算法。

利用迭代方法求解式（4－24），就是首先选择一组或两组合理的初值如 $\sigma^{(0)}(x)$ 和 $\omega^{(0)}(x)$，然后开始第一次迭代运算求得 $\sigma^{(1)}(x)$ 和 $\omega^{(1)}(x)$，并用 $\sigma^{(0)}(x)$ 和 $\omega^{(0)}(x)$ 表示它们。这样依次进行，由 $\sigma^{(i)}(x)$ 和 $\omega^{(i)}(x)$ 求得 $\sigma^{(i+1)}(x)$ 和 $\omega^{(i+1)}(x)$，即首先求得满足式（4－24）的 $\sigma(x)$ 和 $\omega(x)$ 的低次项，通过迭代逐步得到 $\sigma(x)$ 和 $\omega(x)$ 的高次

项，最后解出满足式（4—24）的$\sigma(x)$和$\omega(x)$。

迭代过程中需定义一个修正项d_n（n 表示迭代第 n 次）：

$$d_n = s_n + \sum_{i=1}^{n} C_{n,i} s_{n-i} \tag{4—25}$$

迭代过程为：

$$\sigma^{(j+1)}(x) = \sigma^{(j)}(x) - d_j d_i^{-1} x^{j-i} \sigma^{(i)}(x) \tag{4—26}$$

$$\omega^{(j+1)}(x) = \omega^{(j)}(x) - d_j d_i^{-1} x^{j-i} \omega^{(i)}(x) \tag{4—27}$$

本文采用的实现方法是 KUANG YUNG LIU 设计的 BM LFSR 算法。电路结构如图 4—8 所示。

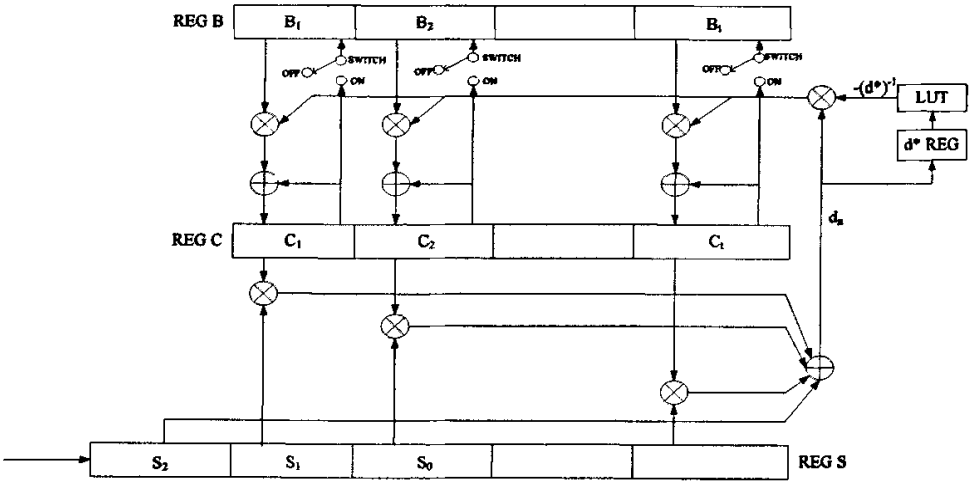

图 4—8　BM LFSR 算法电路结构

BM 算法运算电路工作原理：

（1）在寄存器 S 移位时钟的上升沿，一个新的伴随式符号进入扩展的 J bit 寄存器，从伴随式 S_0 开始。同时，寄存器组 S 向右移动一个符号，扩展寄存器中的 J bit 符号同时移入寄存器组 S 的第一级寄存器里。寄存器组 C 和 S 的内容逐级对应相乘后乘积再相加，通过加法树产生差值 d_n。d_n 再与查找表的输出（即为负的 $d*$ 的逆元素）相乘，将结果传送回 LFSR 电路逻辑。$-(d_n / d*)$ 与寄存器组 B 中的内容相乘，乘积与寄存器组 C 中的内容相加后反馈回寄存器组 C 的输入端。

在寄存器组 C 的载入时钟的上升沿，新数据被载入寄存器组 B 和 C，与此同时，寄存器组 S 的移位时钟到来并载入一个新的伴随式。

（2）在d_n=0 的条件下，$-(d_n/d*)$=0。则在寄存器组 C 载入时钟到来时寄存器组 C 中的内容不发生改变。此时寄存器组 B 进入自动移位状态，寄存器 B 向右移动一个符号，最低级移入有限域元素 0。这个过程中，寄存器 l 和寄存器 $d*$ 均保持不变。

（3）在d_n≠0 并且 n>=2l（寄存器 l，表示错误位置多项式的次数）的条件下，寄存器 C 载入时钟到来时把寄存器 C 输入端的数据载入寄存器组 C。此时寄存器组 B 进入数据载入状态，即在寄存器 C 载入时钟到来时，把寄存器组 C 中的内容向右移位一个符号后并载入寄存器组 B,同时寄存器组 B 的最低位载入有限域元素 1。在寄存器组 S 移位时钟到来时，将数据 d_n 和 $n-l+1$ 分别载入寄存器 $d*$ 和 l。

（4）在d_n≠0 并且 n<2l 的条件下，当寄存器 C 载入时钟到来时把寄存器 C 输入端的数据载入寄存器组 C。此时寄存器组 B 进入自动移位状态。

BM 算法电路的工作流程图如图 4−9 所示。

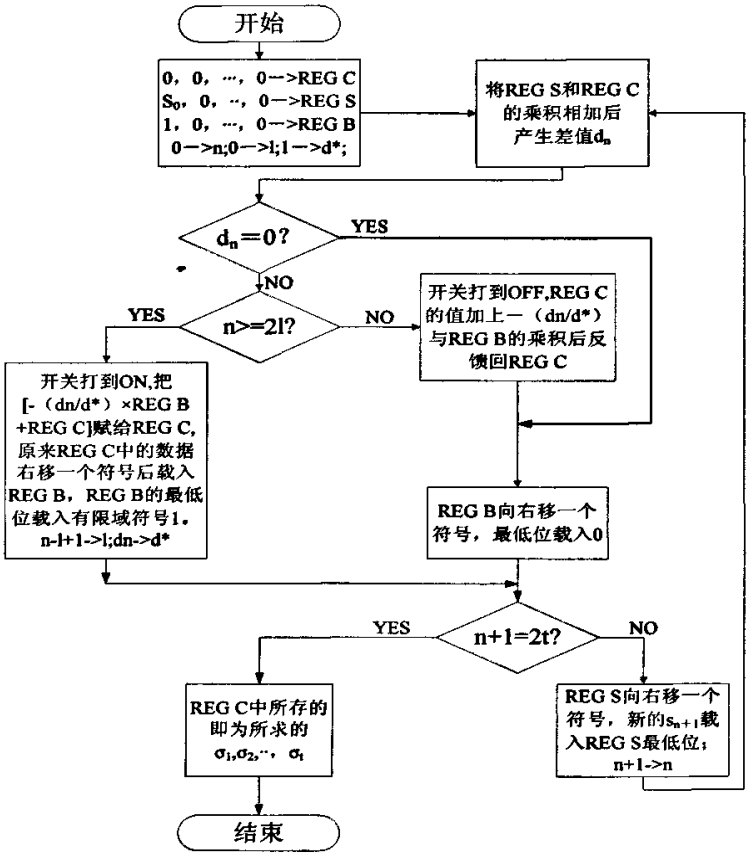

图 4−9　BM 算法电路工作流程图

求得$\sigma(x)$后，根据关键方程$S(x)\sigma(x)=\omega(x)\ (\mathrm{mod}\,x^{2t+1})$，可计算$\omega(x)$如下：

$$\omega_i(2t)=s_i\cdot\sigma_0(2t)+s_{i-1}\cdot\sigma_1(2t)+\cdots+s_0\cdot\sigma_i(2t)\quad(i=0,1,\cdots t-1)\qquad(4-28)$$

在迭代过程中，涉及到有限域元素求逆的问题。根据有限域的性质，对于$\alpha\in GF(2^8)$，有$\alpha=\alpha^{2^8}$，因此有

$$\alpha^{-1}=\alpha^{2^8-2}=\alpha^{2+2^2+2^3+2^4+2^5+2^6+2^7}=\alpha^{2}\cdot\alpha^{2^2}\cdot\alpha^{2^3}\cdot\alpha^{2^4}\cdot\alpha^{2^5}\cdot\alpha^{2^6}\cdot\alpha^{2^7}\qquad(4-29)$$

根据式(4－29)，至少需要用13个有限域乘法器来实现$GF(2^8)$域的求逆电路。这需要占用大量的硬件资源，计算速度也非常慢。为了避免这个求逆运算，产生了多种对BM算法的改进形式，如串行无逆BM算法等。但那些算法增加的电路的复杂度，也大大增加了计算所需的时钟周期数。

基于查找表的方法是实现有限域求逆运算的一种较为简便的实现方法。首先将有限域元素的逆计算出来，存放在查找表中，利用待求逆元素为地址进行查找。这种方法计算速度很快，特别是目前高端FPGA中都内嵌了大容量的RAM块，为采用查找表方法提供了有力的支持。对于$GF(2^8)$域，采用查找表实现求逆运算总共需要8×2^8 bit存储空间。对于RS（255，239）码来说，用$GF(2^8)$上的本原多项式$p(x)=x^8+x^4+x^3+x^2+1$作除法电路，电路进行自发运算就可以得到255个非零元素。电路结构如图4－10所示。给寄存器赋初值（D8,D7,...,D1）=00000001=α^0，电路自发运算，每时钟右移一次，就会不断在8.位寄存器中得到α^1，α^2，α^3，.....。

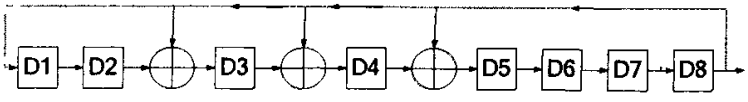

图4－10　　有限域元素生成电路

有限域元素生成电路每向右移动一次就相当与乘以α，每左移一次就相当与除以α。将图4－10中数据流动方向反转，如图4－11所示，就得到逆元素生成电路。电路的初值仍然赋为（D8,D7,...,D1）=00000001=α^0，电路每时钟向左移动一次，就会不断得到α^{-1}，α^{-2}，....。这样可以得到正、逆元素对照表。

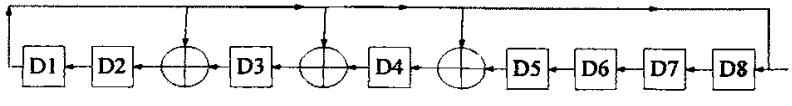

图4－11 有限域逆元素生成电路

4.3.3 钱搜索电路和福尼算法电路设计

设 $R(x)=r_{n-1}x^{n-1}+r_{n-2}x^{n-2}+\cdots+r_1x+r_0$。为了要检验第一位 r_{n-1} 是否错误，相当于译码器要确定 α^{n-1} 是否是错误位置数，即要检验 $\alpha^{-(n-1)}$ 是否是 $\sigma(x)$ 的根。若 $\alpha^{-(n-1)}=\alpha$ 是 $\sigma(x)$ 的根，则

$$\sigma(\alpha^{-(n-1)})=\sigma(\alpha)=1+\sigma_1\alpha+\sigma_2\alpha^2+\cdots+\sigma_t\alpha^t=0 \qquad (4-30)$$

根据这个准则就可以判定 r_{n-1} 是否发生错误。利用已经得到的 $\sigma(x)$，这样依次对每一个 r_{n-l} $(l=1,2,\cdots,n)$ 进行检验，就可以求得 $\sigma(x)$ 的根。这个过程就称为钱搜索。钱氏搜索电路结构如图 4－12 所示。

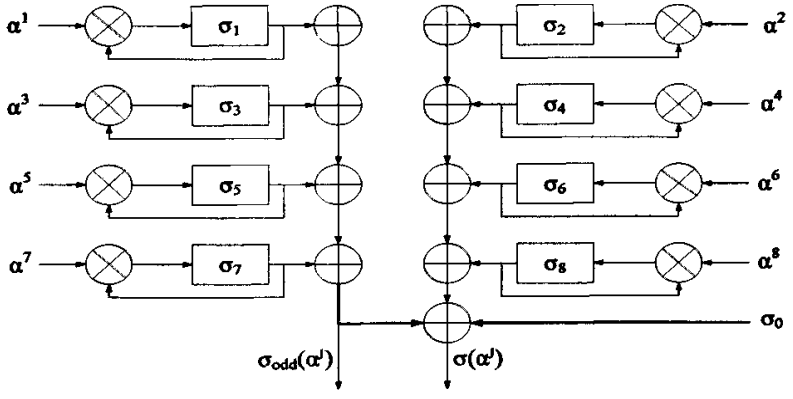

图 4－12 钱氏搜索电路结构

福尼算法用于求解错误值，根据推导可得出：

$$Y_j=\omega(z)/z\sigma'(z),\ (z=x_j^{-1}) \qquad (4-31)$$

方程中的 $\sigma'(x)$ 表示 $\sigma(x)$ 的导数，由有限域元素的性质可知：

$$\sigma_{odd}(x)=x\cdot\sigma'(x) \qquad (4-32)$$

则所求的错误值可表示为

$$Y_j=\omega(x)/\sigma_{odd}(x) \qquad (4-33)$$

$\sigma_{odd}(x)$ 如图 4－12 所示，表示 $\sigma(x)$ 表达式中的奇数次方部分之和。利用钱氏搜索电路先求出 $\sigma_{odd}(x)$ 和 $\sigma(x)$，可减少计算量。

福尼算法电路如图 4－13 所示。

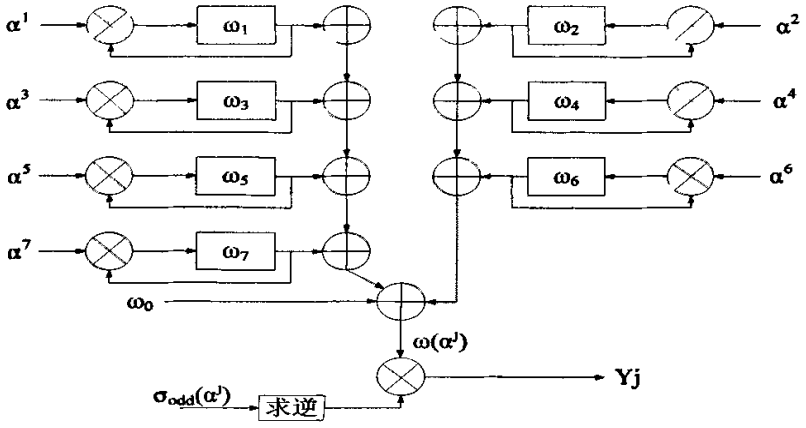

图 4－13 福尼算法电路结构

4.3.4 译码输出电路设计

在译码输出部分，利用计算出的错误位置和错误值对发生错误的数据进行修正，输出正确的码字,电路结构如图 4－14 所示。算法如下：

For i=0 to 254

If $(\sigma(\alpha^i)==0)$ then

$$cc_{254-i}=r_{254-i}-Y_i=r_{254-i}-\omega(\alpha^i)/\sigma_{odd}(\alpha^i)$$

end

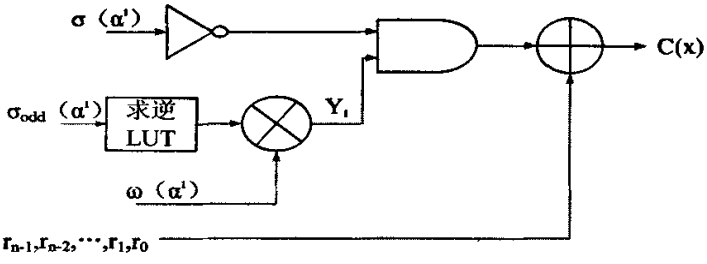

图 4－14 译码输出电路

整个设计要用时钟进行精确的同步。

4.3.5 RS 译码器设计结果

在本设计中，采用自顶向下的设计方法，先分别设计伴随式模块、BM 算法模块、钱搜索模块和福尼算法模块，通过编译、综合、验证后，在 RS 译码器模块中

进行调用。译码过程中，伴随式模块的计算需要 255 个时钟周期，BM 算法模块需要 16 个时钟周期，输出部分需要 255 个时钟周期。

本文设计的 RS（255，239）译码器使用 Verilog HDL 对整个模型进行描述，以 ALTERA 公司的 FPGA 芯片 EP1S80B956C6 为硬件平台进行实现。利用 Quartus II 对设计进行综合和仿真。RS 译码器的实现结果如表 4－3 所示。

表 4－3　RS 译码器实现结果

设计模块	Total logic element	最高工作频率
RS 译码器	6014	39MHz
伴随式模块	309	161 MHz
BM 算法模块	2607	43 MHz
钱搜索模块	102	309 MHz
福尼算法模块	94	259 MHz

将 RS 编码器的编码结果作为输入送入 RS 译码器中，译码结果与编码器的输入一致，即证明译码结果正确。仿真波形图如图 4－15 所示。

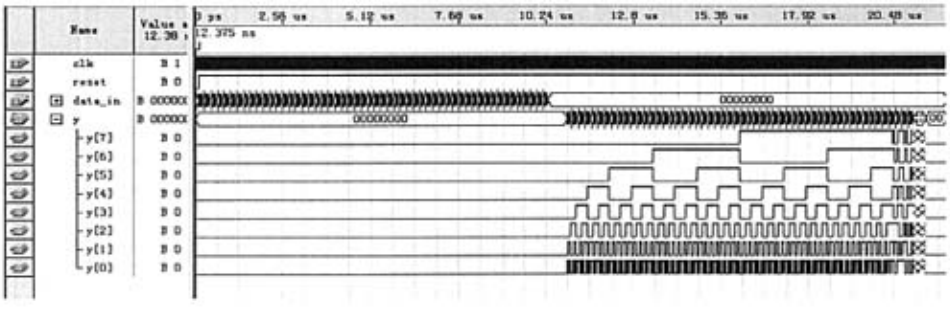

图 4－15 RS 译码器仿真波形图

第五章 基带成形滤波器设计及实现

为了提高频谱利用率，频谱成形技术被广泛使用。与基带模拟成形滤波器相比，基带数字成形滤波器具有高精度、高可靠性、高灵活性的优点，便于大规模集成、易于实现线性相位等特点。随着数字技术的发展，用数字技术设计滤波器越来越受到人们的注意，并得到广泛地应用。本文介绍了 FIR 和 IIR 数字滤波器设计及基于 FPGA 的实现结构，在此基础上利用多相结构实现平方根升余弦滚降滤波器。

5.1 数字滤波器介绍

数字滤波器在信号滤波处理、检测与参数估计等方面有重要应用。它利用有限精度算法实现离散时间线性非时变系统，其输入是一组数字量，输出是经过变换的另一组数字量。数字滤波器具有稳定性高、精度高、灵活性大等突出优点。

数字滤波器的系统函数可以表示为：

$$H(z)=\frac{\sum_{k=0}^{m}b_k z^{-k}}{1-\sum_{k=1}^{n}a_k z^{-k}}=\frac{Y(z)}{X(z)} \qquad (5-1)$$

直接由 $H(z)$ 得出表示输入输出关系的线性常系数差分方程为：

$$y(n)=\sum_{k=1}^{n}a_k y(n-k)+\sum_{k=0}^{m}b_k x(n-k) \qquad (5-2)$$

数字滤波器根据单位脉冲响应 $h(n)$ 的时间特性可分为无限长单位脉冲响应(IIR, infinite impulse response)数字滤波器和有限长单位脉冲响应(FIR, finite impulse response)数字滤波器两种。若系统的单位取样响应延伸到无穷之长，称之为 IIR 系统。IIR 系统用式（5－2）表示时，系数 a_k、b_k 不全为零；系统函数 $H(z)$ 在有限平面 Z 上有极点存在，可能会出现不稳定的情况；在结构上存在着输出到

输入的反馈网络，即结构是递归的。若系统的单位取样响应是一个有限长序列，则称之为 FIR 系统。FIR 系统函数仅有零点(除 z=0 的极点外)，故一定是稳定的，并可以用因果系统来实现。FIR 系统用式（5－2）表示时系数a_k全为零，故也可用式（5－3）来描述 FIR 系统：

$$y(n)=\sum_{k=0}^{m}b_k x(n-k) \tag{5－3}$$

数字滤波器设计的一个重要步骤是确定一个可实现的传输函数$G(z)$来逼近指定的频率响应。如果要设计一个无限冲激响应滤波器（IIR），$G(z)$必须要保证是稳定的。确定传输函数$G(z)$的过程称为数字滤波器设计。在$G(z)$确定以后，要选用一种合适的滤波器结构来实现它。

5.1.1 IIR 数字滤波器设计

设计 IIR 滤波器时，通常将数字滤波器的设计指标转化成模拟低通原型滤波器的设计指标，从而确定满足这些指标的模拟低通滤波器的传输函数$H_a(s)$，然后再将它变换成所需要的数字滤波器传输函数$G(z)$。这种方法之所以得到广泛应用，是因为模拟逼近技术已经很成熟；通常能产生闭式的解；模拟滤波器设计有大量图表可查；在很多应用中需要模拟滤波器的数字仿真。

将$H_a(s)$变换成$G(z)$的基本思路是要把 s 域映射到 z 域，从而使数字滤波器能模仿模拟滤波器的特性。这种映射函数必须满足以下条件：

（1）s 平面的虚轴（jΩ）必须映射到 z 平面的单位圆上；

（2）稳定的模拟传输函数能变换为稳定的数字传输函数。

双线性变换是目前应用最广泛的一种变换，如式（5－4）所示。

$$s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \tag{5－4}$$

数字传输函数$G(z)$和原模拟传输函数$H_a(s)$之间的关系如式（5－5）所示：

$$G(z)=H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} \tag{5－5}$$

由于双线性变换法是一种单值映射，因此消除了频率混叠现象。通常可选择 $T=2$ 来简化设计过程。根据 s 平面的虚轴 $s=j\Omega$ 与 z 平面单位圆 $(z=e^{j\omega})$ 之间的映射，由式（5—4）可推得：

$$\Omega=\tan(\frac{\omega}{2}) \qquad (5-6)$$

这说明双线性变换这种映射的非线性程度很高，导致了频率轴的失真。因此，在设计满足特定幅度响应的数字滤波器前，要首先利用式（5—6）对临界频带 (ω_p,ω_s) 加以预畸变，再进行后续设计步骤。

在满足相同幅度指标的前提下，IIR 滤波器的阶数比 FIR 滤波器低得多，因而具有较低的计算复杂度。但使用 IIR 数字滤波器逼近理想的滤波器幅度频率特性时，得到的滤波器往往是非线性的，限制了 IIR 数字滤波器在要求线性相位响应的数字系统中的应用。

5.1.2 FIR 数字滤波器设计

FIR 数字滤波器可以在设计任意幅度频率特性滤波器的同时，保证精确、严格的线性相位特性；总是可以独立于滤波器系数保持 BIBO 稳定；允许设计多通带（或多阻带）滤波器。因此在很多领域，FIR 数字滤波器都是首选。

FIR 数字滤波器的设计方法有很多种。一种方法是基于对频率样本进行离散傅立叶逆变换来设计。对于长度为 N 的 FIR 数字滤波器，其冲激响应的 N 点离散傅立叶变换由其频率响应的 N 个等间隔的不同频率样本组成，因此该滤波器的冲激响应序列可以利用它的频率样本的离散傅立叶逆变换来计算。这里的基本假设是指定的频率响应可以由 N 个频率样本来描述，因此就可以通过这个样本完全恢复。

另一种直观的设计方法是基于对指定的频率响应的傅立叶级数进行截短来设计，这种方法最为常用。在均方误差准则下，理想无限冲激响应的最佳和最简单的有限长逼近是通过截短来得到的。

通过截短得到的 FIR 滤波器的幅度响应呈现振动的现象，通常称为吉布斯现象。截短运算可以认为是将无限冲激响应序列 $h_d[n]$ 与一个有限长矩形窗序列 $\omega[n]$ 相乘的结果，如式（5—7）所示。

$$h_t[n]=h_d[n]\cdot\omega[n] \qquad (5-7)$$

出现吉布斯现象的原因就是因为矩形窗有陡峭的下降沿。为了减弱吉布斯现象，可以利用两边都是逐渐平滑减少到零的窗函数，或者在通带到阻带之间设计平滑的过渡带。

FIR 数字滤波器有多种实现结构：

（1）直接型

直接型结构的 FIR 滤波器可用式（5－8）来描述：

$$y[n]=\sum_{k=0}^{N}h[k]x[n-k] \tag{5-8}$$

如图 5－1 所示，乘法器的系数正好是传输函数的系数，这种结构也称为抽头延时线或横向滤波器。它的转置形式如图 5－2 所示，两种结构是等效的。

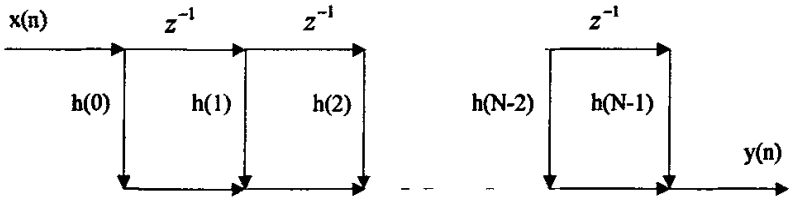

图 5－1 直接型 FIR 滤波器结构

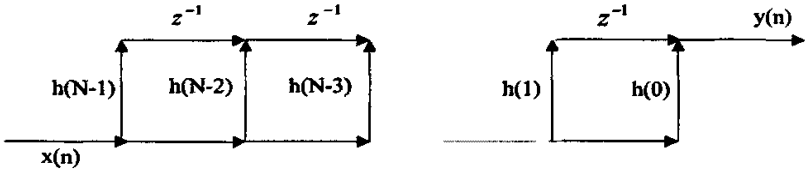

图 5－2　直接型 FIR 结构的转置形式

（2）级联型

高阶 FIR 传输函数可以用每部分都是一阶或二阶传输函数的级联来实现。级联型结构可以用式（5－9）来描述：

$$H(z)=\sum_{n=0}^{N-1}h(n)^{-n}=h(0)\prod_{k=1}^{L}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2}) \tag{5-9}$$

级联型 FIR 滤波器结构如图 5－3 所示，其中的每一个二阶部分也可以用转置的直接型结构来实现。

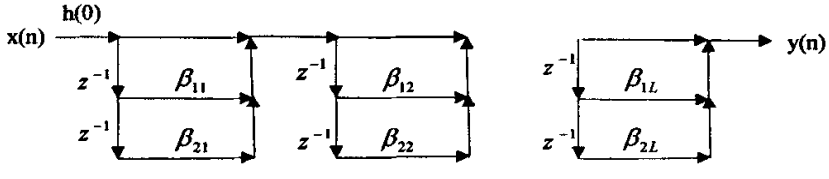

图 5—3 级联型 FIR 结构

（3）线性相位 FIR 结构

N 阶线性相位有限冲激响应滤波器可以用对称冲激响应

$$h[n]=h[N-n] \tag{5—10}$$

或者反对称冲激响应

$$h[n]=-h[N-n] \tag{5—11}$$

来描述。其对称中心在 $n=\frac{N-1}{2}$ 处。所谓线性相位特性是指滤波器对不同频率的正弦波所产生的相移和正弦波的频率成线性关系。

在传输函数的直接型实现中，利用线性 FIR 滤波器的对称（或反对称）性质可以减少近一半的乘法器，如图 5—4 所示。

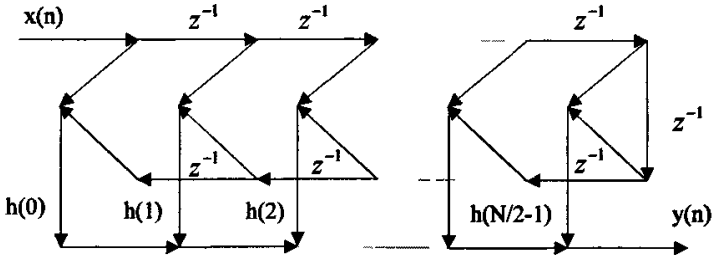

图 5—4　线性相位 FIR 结构

（4）多相实现

L 支 N 阶多相分解的传输函数可以用式（5—12）来描述：

$$H(z)=\sum_{m=0}^{L-1}z^{-m}E_m(z^L) \tag{5—12}$$

式（5—12）中，

$$E_m(z^L)=\sum_{n=0}^{\lfloor (N+1)/L \rfloor}h[Ln+m]z^{-n},\quad 0\le m\le L-1 \tag{5—13}$$

$H(z)$ 的基于式（5—12）的分解实现结构称为多相实现。子滤波器 $E_m(z^L)$ 也是有限冲激响应滤波器，并可用上述任一种结构实现。在多抽样率的数字信号处理中，多相结构经常用于有效计算的实现。

图 5—5 给出了一个传输函数的四支、三支的多相实现。每种结构中的子滤波器都是互不相同的。

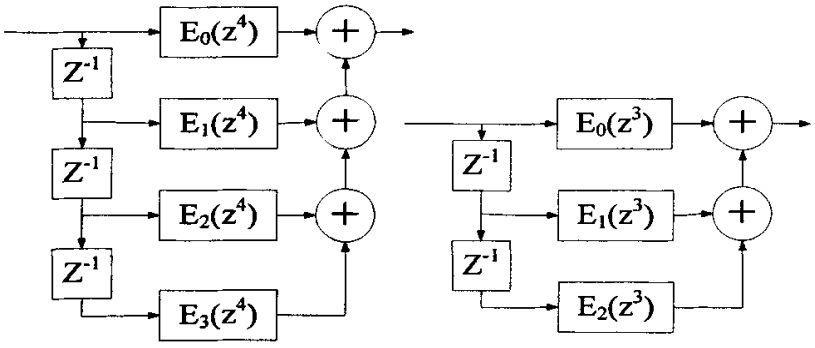

图 5－5　FIR 传输函数的多相实现

数字滤波器结构的计算复杂度取决于实现该结构的乘法器和两输入加法器的数量，这些数量大致说明它的实现成本。但计算复杂度不是对给定的传输函数选择哪一个特定结构来实现的唯一标准。在有限字长的限制下，所选用结构的实现性能也应该与该结构实现成本一起考虑。表 5－1 对 N 阶 FIR 滤波器的各种实现方法的计算复杂度进行了比较。

表 5－1　N 阶 FIR 滤波器各种实现方法的计算复杂度比较

结构	乘法器数目	两输入加法器数目
直接型	N+1	N
级联型	N+1	N
线性相位	$\left\lfloor \frac{N+1}{2} \right\rfloor$	N
多相实现	N+1	N

5.2 基带成形滤波原理

随着现代数字通信技术的发展，频带拥挤的问题日益严重。由于传输信道的带限和非线性，对发送信号的频谱提出了更高的要求。节省频带，提高频谱利用率，已经成为数字通信领域的一个重要课题。为了提高频谱利用率，广泛采用了频谱成形技术，对发送信号的频谱进行加工，使其在消除码间干扰和实行最佳检测的前提下，压缩信号频带，提高频谱利用率。频谱成形技术常常在基带实现。虽然也可以在中频和射频实现，但由于中频和射频信号频率较高，难以采用数字处理技术，实现难度较大。与基带模拟成形滤波器相比，基带数字成形滤波器具

有高精度、高可靠性、高灵活性的优点，便于大规模集成、易于实现线性相位等特点。现代数字通信系统中，成形滤波大多在数字域进行。

二进制数字基带波形都是矩形波，在画频谱时通常只画出了其中能量最集中的频率范围，但这些基带信号在频域内实际上是无穷延伸的。如果直接采用矩形脉冲的基带信号作为传输码型，由于实际信道的频带都是有限的，则传输系统接收端得到的信号频谱必定与发送端不同，这就会使接收端数字基带信号的波形失真。当信道频带严格受限时，波形失真问题就变得比较严重，尤其在传输多元信号时更为突出。因此传输基带信号受到约束的主要因素是系统的频率特性。一种方法是通过加宽传输频带使这种干扰减小到任意程度，但这会导致对频带不必要地浪费；并且如果频带展宽得太多，还会给系统引入额外的大量噪声。另一种方法是通过设计信号波形，或采用合适的传输滤波器，以便在最小传输带宽的条件下大大减小甚至消除这种干扰。

Nyquist 准则给出了数字信号在无噪声线性信道上无失真传输的条件，可以用理想低通滤波器来描述。理想低通滤波器时域上的 Sa(t)波形具有频带利用率高的优点，但它频域上的陡截止特性无法实现，并且 Sa(t)波形的前导和后尾波动较大，衰减较慢，码间串扰严重，即使接收端的同步定时出现较小误差，也会导致严重的码间干扰。若将理想低通滤波器的锐截止特性按照一定规律滚降，同样可以实现信号的无失真传输。如图 5—6 所示。

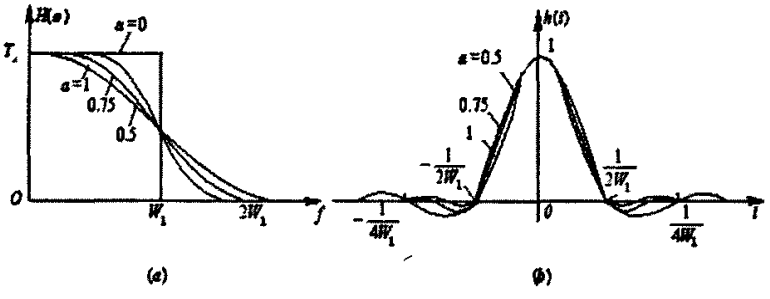

图 5—6　理想低通滤波器滚降示意图

这种滚降特性不仅易于实现，而且其时域响应波形的前导和收尾波动较小，衰减较快，对系统接收端的同步定时精度要求较低。但与理想低通滤波器相比，其频带利用率也下降为 2/(1+α)波特/Hz。α 称为滚降系数，如式（5—14）所示：

$$\alpha = \frac{B - f_N}{f_N}, \quad (0 \le \alpha \le 1) \tag{5—14}$$

实际系统中常采用以 Nyquist 频率为中心，具有奇对称特性的升余弦滚降滤波器。升余弦特性除本码元抽样点不为零外，其余所有码元抽样点上均为零。随滚降系数 α 的增大，波形振荡起伏变小，“尾巴”衰减加快，但其所占用频带也变宽。综合考虑频带使用率和控制衰减振荡，α 一般取 0.22~0.38。升余弦谱如式（5－15）所示：

$$X_{rc}(f)=\begin{cases}T & \left(0\le|f|\le\dfrac{1-\beta}{2T}\right)\\ \dfrac{T}{2}\left\{1+\cos\left[\dfrac{\pi T}{\beta}\left(|f|-\dfrac{1-\beta}{2T}\right)\right]\right\} & \left(\dfrac{1-\beta}{2T}\le|f|\le\dfrac{1+\beta}{2T}\right)\\ 0 & \left(|f|>\dfrac{1-\beta}{2T}\right)\end{cases} \tag{5－15}$$

为了使系统实现最佳检测，使传输误码率最低，要求接收滤波器与发送滤波器相匹配，即要求发送成形滤波器特性 $G_T(f)$ 和接收成形滤波器特性 $G_R(f)$ 均分整个信道滤波器特性 $X_{rc}(f)$。由于升余弦谱的平滑特性，因此设计实用的发送和接收滤波器来近似实现整个期望的频率响应是可能的。在信道是理想的情况下，即 $C(f)=1,|f|\le W$，有

$$X_{rc}(f)=G_T(f)G_R(f) \tag{5－16}$$

式中，$G_T(f)$和 $G_R(f)$分别是发送滤波器和接收滤波器的频率响应。在此情况下，若接收滤波器匹配于发送滤波器，则有 $X_{rc}(f)=G_T(f)G_R(f)=|G_T(f)|^2$。对此理想情况，

$$G_T(f)=\sqrt{|X_{rc}(f)|}e^{-j2\pi ft_0} \tag{5－17}$$

并且 $G_R(f)=G_T^*(f)$，其中 t_0 是某标称延时，用来保证该滤波器的物理可实现性。因此，整个升余弦频谱特性在发送滤波器和接收滤波器之间均等的划分，得到两个平方根升余弦滤波器 $G_T(f)$和 $G_R(f)$。同时，为了保证接收滤波器的物理可实现性，附加的延时是必要的。

5.3 发送端平方根升余弦滚降滤波器设计

本文设计发送端平方根升余弦滚降滤波器，采用窗函数法设计具有线性相位 FIR 滤波器。输入为 25MHz 的 3bit 数据，输出为 100MHz 的 10 比特数据。设计

指标如表 5—2 所示。

表 5—2　发送端平方根升余弦滤波器设计指标

滚降系数α	截止频率f_c	采样频率f_s	阻带衰减α_s
0.35	12.5MHz	100MHz	60dB

用 MATLAB 的 FDATool 来设计满足指标要求的滤波器，对分别加 hann 窗、hamming 窗、blackman 窗、kaiser 窗后的情况进行比较。凯泽（kaiser）窗是使用最广泛的可变窗，如式（5—18）所示：

$$w[n]=\frac{I_0\{\beta\sqrt{1-(n/M)^2}\}}{I_0(\beta)},\ -M\le n\le M \tag{5—18}$$

其中$I_0(u)$是修正的零阶贝塞尔函数；β是可调参数，计算公式为：

$$\beta=\begin{cases}0.1102(\alpha_s-8.7) & \alpha_s>50\\ 0.5842(\alpha_s-21)^{0.4}+0.07886(\alpha_s-21) & 21\le\alpha_s\le50\\ 0 & \alpha_s<21\end{cases} \tag{5—19}$$

根据设计要求$\alpha_s=60dB$，可计算出β=5.65326。加窗结果分别如图 5—7 到图 5—10 所示。

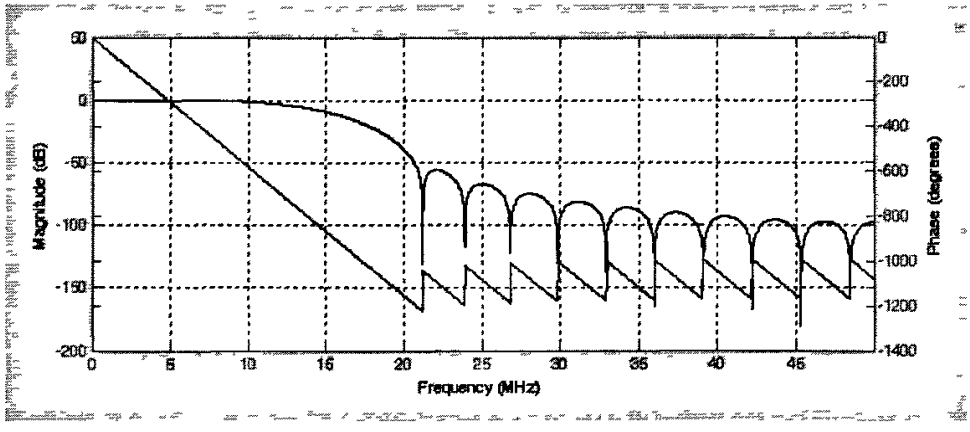

图 5—7　平方根升余弦滚降滤波器，加 hann 窗，32 阶

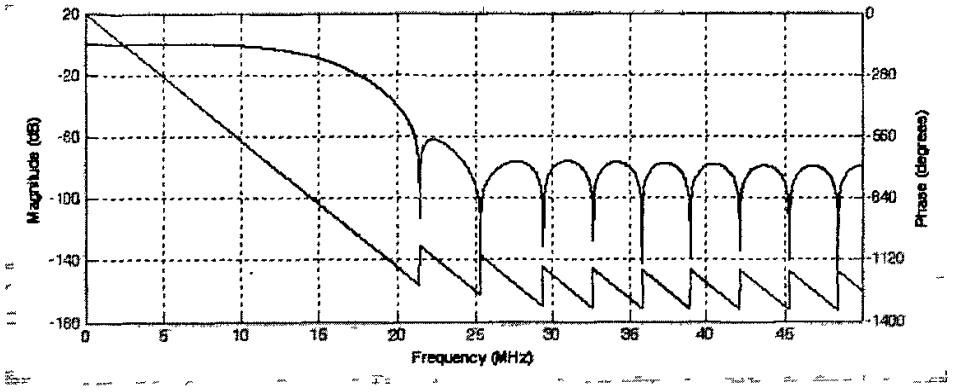

图 5—8　平方根升余弦滚降滤波器，加 hamming 窗，32 阶

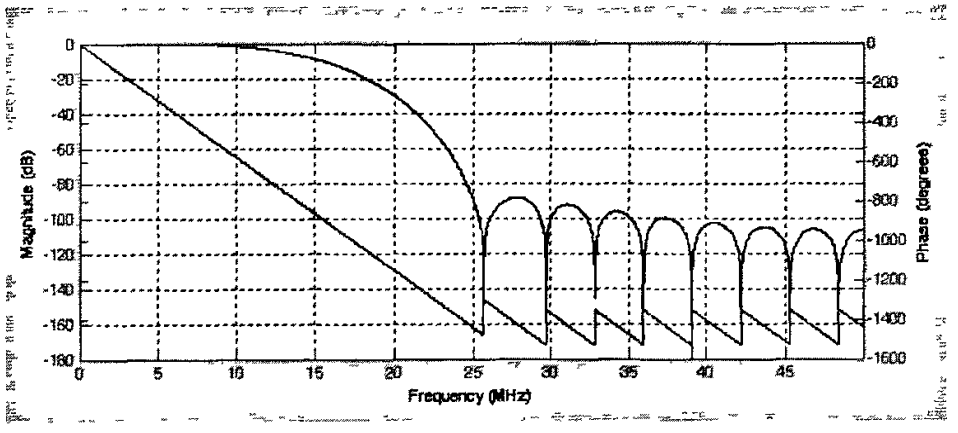

图 5—9　平方根升余弦滚降滤波器，加 blackman 窗，32 阶

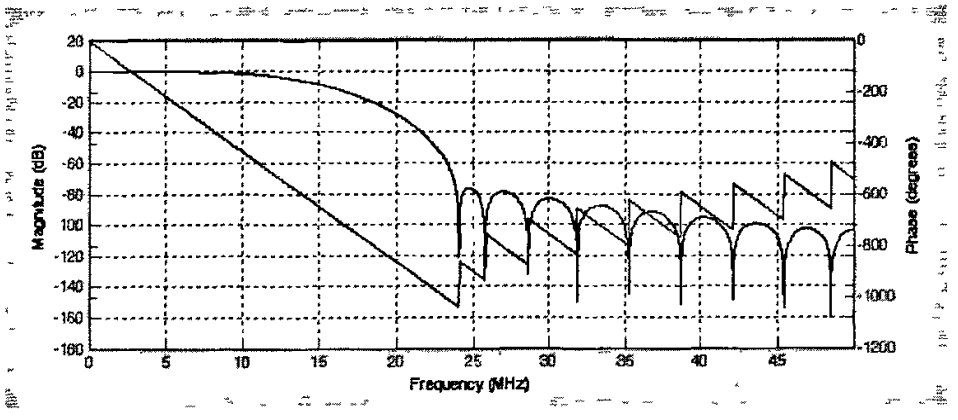

图 5—10　平方根升余弦滚降滤波器，加 kaiser 窗，24 阶，β=5.65326

通过比较可知，加 kaiser 窗，在 24 阶阻带衰减就达到了 79dB，过渡带宽也适中，性能远远超过加 hann 窗、hamming 窗和 blackman 窗时在 32 阶的性能。因此本文选择加 kaiser 窗的设计方案，既充分满足设计要求，并且实现复杂度也较低。

得到理想滤波器后，对它进行量化处理，就成为了可以实际应用的滤波器。

量化过程应在满足设计指标的前提下减少滤波器系数的位数，以降低实现复杂度。图 5－11 显示了对滤波器系数做 14 位定点量化时与理想滤波器幅频响应的比较。在保持严格线性相位的条件下，阻带衰减仍有很大的裕量。

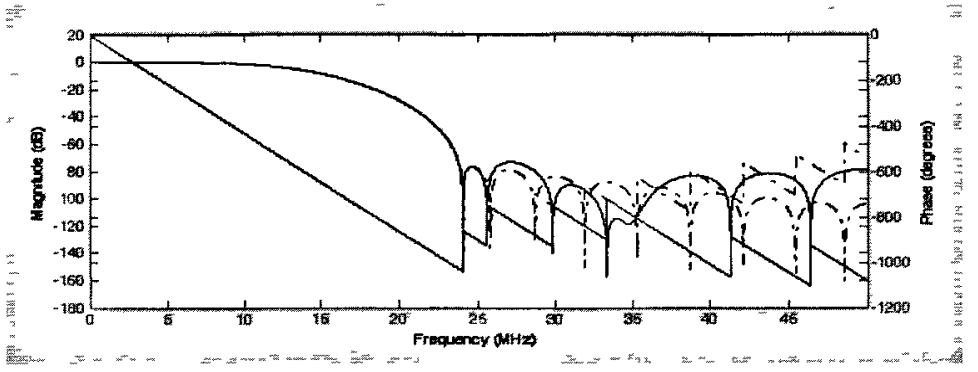

图 5－11　对滤波器系数做 14 位定点量化

将滤波器系数量化为 12 位，如图 5－12 所示。量化后虽与理想滤波器有一定差距，但阻带衰减仍然达到了 65dB，仍可满足设计要求。

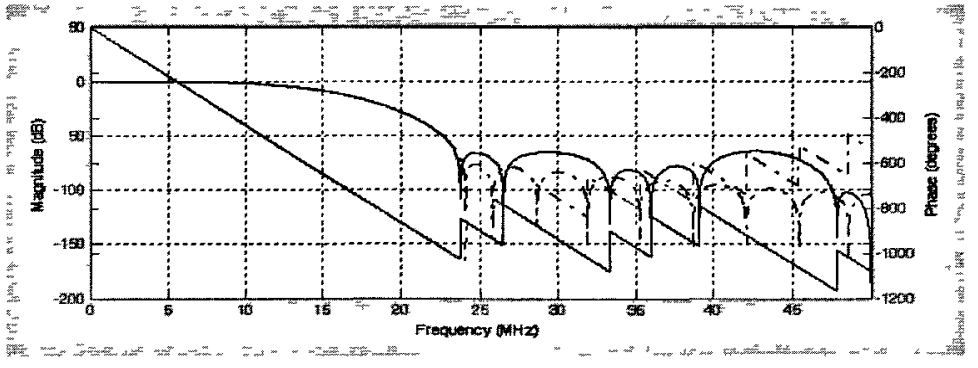

图 5－12　对滤波器系数做 12 位定点量化

但若将滤波器系数量化至 10 位，如图 5－13 所示。则不再满足设计指标，不能采用该设计。

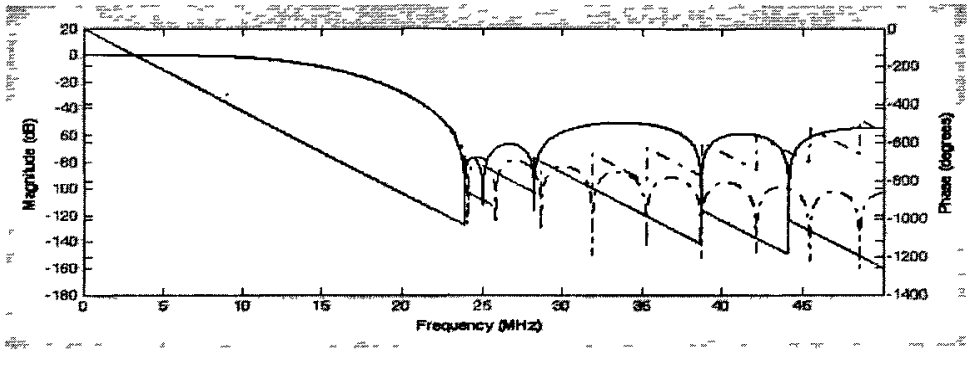

图 5－13　对滤波器系数做 10 位定点量化

根据以上设计过程，本文所设计发送端平方根升余弦滚降滤波器各项参数如表5—3所示。

表5—3　发送端平方根升余弦滚降滤波器各项参数

窗函数	阶数	输入	输出	滤波器系数
Kaiser窗	24阶	3bit	10bit	12bit

滤波器系数如下：

$h(0)=h(24)=111111111111$　　$h(1)=h(23)=111111111111$

$h(2)=h(22)=000000000011$　　$h(3)=h(21)=000000001100$

$h(4)=h(20)=000000010000$　　$h(5)=h(19)=111111110111$

$h(6)=h(18)=111110111010$　　$h(7)=h(17)=111110000110$

$h(8)=h(16)=111110111111$　　$h(9)=h(15)=000010110100$

$h(10)=h(14)=001001000011$　　$h(11)=h(13)=001111000011$

$h(12)=010001100010$

5.4 发送端平方根升余弦滚降滤波器（SRRC）的实现

5.4.1 SRRC滤波器的整体结构

由于输出频率是输入频率的4倍，所设计的滤波器在完成脉冲成形功能的同时还要完成对输入数据的4倍插值功能，即在每两个输入符号之间插入3个零。这是由于在数字滤波器设计过程中对理论上时域无限长的升余弦波形进行截短造成的。实际中升余弦波形是用有限样点的方式进行存储表示的，即取截短长度为M个码元周期，每个码元间隔取L个样点。因此造成输入码流与截短基带波形两序列的码元间隔不一致。本设计中，在每个码元间隔取4个样点，故需对输入数据进行4倍插值。

一种实现方法是先对输入数据流做4倍插值，在每两个输入符号之间插入3个零；将插值后的数据流通过平方根升余弦滤波器（SRRC）滤波输出。其中SRRC的实现结构可采用5.1.2节中FIR滤波器的任意一种实现结构，特别是采用线性相

位结构可在硬件资源方面有很大节省。直接实现结构如图 5－14 所示。

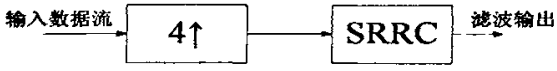

图 5－14 平方根升余弦滚降滤波器直接实现结构

图 5－14 所示的实现结构有下述缺点：

（1）它增加了一个 4 倍插值结构，增加了系统复杂度；

（2）由于是对插值后的数据流进行滤波处理，要求设计的 SRRC 必须稳定工作在 100MHz，增加了设计的难度；

（3）由于数字器件工作频率越高，功耗也越大。可在允许的条件下降低工作频率。

考虑到插值的数据都是零，可采用多相结构来实现 SRRC 滤波器：它不需要添加额外的插值结构；在滤波的同时也实现 4 倍插值功能；使滤波器工作在 25MHz。多相结构常用于多速率数字信号处理的有效计算实现，可以提高效率。对传输函数 $H(z)$ 进行四支的多相分解，如式（5－20）所示：

$$\begin{aligned}H(z)&=h[0]+h[1]z^{-1}+h[2]z^{-2}+\cdots+h[24]z^{-24}\\&=(h[0]+h[4]z^{-4}+h[8]z^{-8}+h[12]z^{-12}+h[16]z^{-16}+h[20]z^{-20}+h[24]z^{-24})+\\&\quad z^{-1}(h[1]+h[5]z^{-4}+h[9]z^{-8}+h[13]z^{-12}+h[17]z^{-16}+h[21]z^{-20})+\\&\quad z^{-2}(h[2]+h[6]z^{-4}+h[10]z^{-8}+h[14]z^{-12}+h[18]z^{-16}+h[22]z^{-20})+\\&\quad z^{-3}(h[3]+h[7]z^{-4}+h[11]z^{-8}+h[15]z^{-12}+h[19]z^{-16}+h[23]z^{-20})\\&=E_0(z^4)+z^{-1}E_1(z^4)+z^{-2}E_2(z^4)+z^{-3}E_3(z^4)\end{aligned}\qquad(5-20)$$

根据式（5－20），可得到基于四支多相分解的 SRRC 滤波器结构如图 5－15 所示，其中子滤波器 $E_0(z)$ 到 $E_3(z)$ 如式（5－21）到式（5－24）所示。

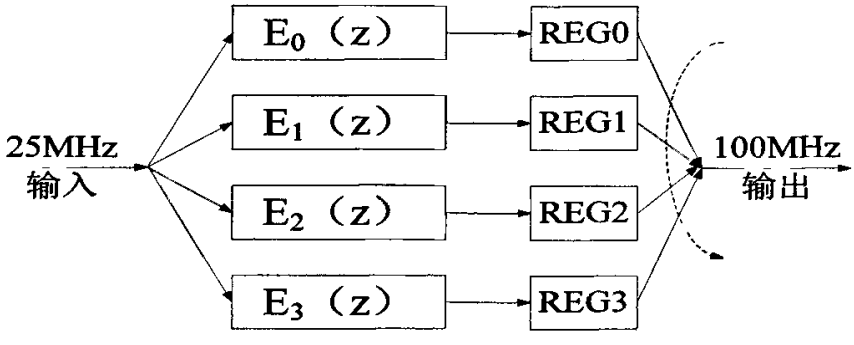

图 5－15 基于四支多相分解的 SRRC 滤波器结构

$$E_0(z)=h[0]+h[4]z^{-1}+h[8]z^{-2}+h[12]z^{-3}+h[16]z^{-4}+h[20]z^{-5}+h[24]z^{-6} \quad (5-21)$$

$$E_1(z)=h[1]+h[5]z^{-1}+h[9]z^{-2}+h[13]z^{-3}+h[17]z^{-4}+h[21]z^{-5} \quad (5-22)$$

$$E_2(z)=h[2]+h[6]z^{-1}+h[10]z^{-2}+h[14]z^{-3}+h[18]z^{-4}+h[22]z^{-5} \quad (5-23)$$

$$E_3(z)=h[3]+h[7]z^{-1}+h[11]z^{-2}+h[15]z^{-3}+h[19]z^{-4}+h[23]z^{-5} \quad (5-24)$$

由图 5－15 可知，基于四支多相分解的 SRRC 滤波器结构具有如下特点：

（1）24 阶的 SRRC 滤波器被等效分解为一个 6 阶和三个 5 阶子滤波器的并行实现，与直接实现结构相比，降低了实现难度，大幅度缩短了计算时间，且没有占用过多的硬件资源；

（2）输入数据同时进入四个子滤波器进行计算，计算结果分别存入寄存器 REG0 到 REG3，四个子滤波器都工作在 25MHz；

（3）在输出端，在一个时钟周期内（指输入频率 25MHz）将上个时钟周期存储在寄存器 REG0 到 REG3 内的数据按顺序输出，完成速率转换。

5.4.2 SRRC 滤波器具体实现

滤波器输出结构涉及时钟频率转换，本文采用的方案是用 100MHz 主时钟对整个电路进行同步，通过一个 4 分频电路产生 25MHz 副时钟对输入和子滤波器进行控制。如图 5－16 所示。

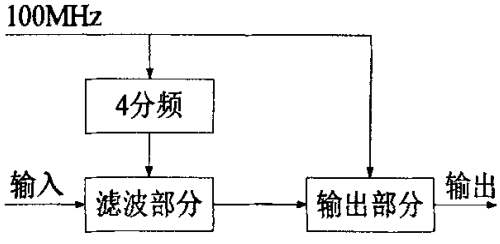

图 5－16 滤波器时钟控制

确定基于四支多相分解的 SRRC 滤波器结构后，主要任务就是实现四个子滤波器 $E_0(z)$ 到 $E_3(z)$ 和滤波器输出结构。由式（5－21）和式（5－23）可知，$E_0(z)$ 和 $E_2(z)$ 仍然保持着系数对称的特性，因此可以采用线性相位 FIR 结构来实现。由表 5－1 可知，线性相位 FIR 结构的实现成本最低。

$E_0(z)$的线性相位 FIR 结构如图 5－17 所示。

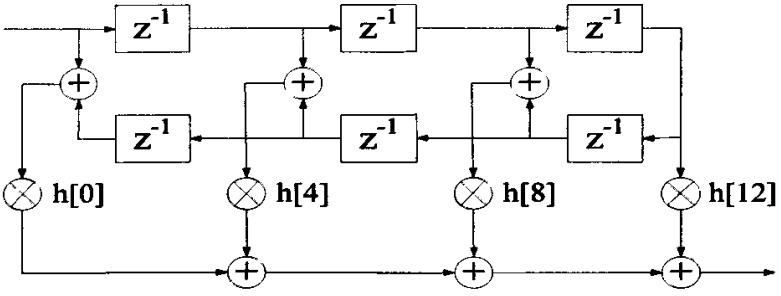

图 5－17 $E_0(z)$线性相位 FIR 结构

$E_2(z)$的线性相位 FIR 结构如图 5－18 所示。

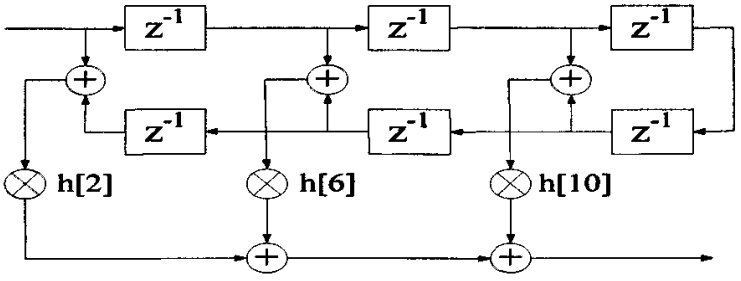

图 5－18 $E_2(z)$线性相位 FIR 结构

由式（5－22）和式（5－24）可知，$E_1(z)$和$E_3(z)$不具有系数对称的特性，不能采用线性相位 FIR 结构来实现。对直接型、转置型、级联型等结构进行比较，在实现成本相近的条件下，转置型结构无需增加寄存器即可实现流水线型计算，提高了它的吞吐能力。$E_1(z)$和$E_3(z)$的结构相同，仅乘法器系数不一致。以$E_1(z)$为例，画出它的转置型结构，如图 5－19 所示：

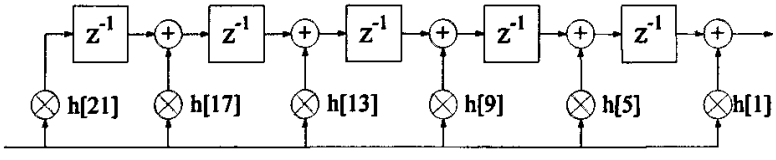

图 5－19 $E_1(z)$转置型 FIR 结构

由图 5－17、图 5－18 和图 5－19 可知，滤波器结构中主要包括乘法器、加法器和移位寄存器结构。特别是乘法器设计，对整体电路的工作频率和资源占用情况有很大影响，应作重点优化设计。

人工计算乘法的过程如下：将移位复制的被乘数依次对准乘数数位的位置进行排列，然后将各列相加。如果乘数的某一位为零，将跳过相应的被乘数，下一个复制被乘数的位置是由向乘数的最高位方向移动时有 1 出现的位置。这个过程用到了将部分积项按列相加的运算，这就需要使用多个加法器来对每一列进行加法运算的硬件方案。由于普通加法器一次只能对两个字进行操作，更可取的实现方案是将被乘数的副本加到累加和上而形成行和序列。

乘法运算可以利用组合电路或时序机来实现。实现两个数相乘的组合电路比时序乘法器所需要的硅片面积更大，但运算速度更快。时序乘法器的特点在于所需硅片面积较小，但完成乘法运算需要几个时钟周期才能得到乘积结果。

提高乘法器性能并简化电路的算法有很多。使用 Booth 算法的乘法器对乘数的位进行重新编码，减少乘数中 1 的个数，以减少完成乘法运算所需的移位加法运算次数。它仅对乘数重新编码，而被乘数保持不变。但 Booth 重编码实际上也有可能会增加所使用的时钟周期，它的有效性取决于数据。正则有符号数字量(CSD：Canonic Signed Digit)编码也是一种优化算法，它与 Booth 乘法器的区别在于 CSD 编码进一步减少了乘数中 1 的个数，其硬件资源比 Booth 乘法器更省。

定点数的表示法包括传统的二进制补码、反码、原码和非传统的有符号数字量编码、对数编码、余数进制编码等。CSD (canonic signed digit)编码就属于非传统的有符号数字量表示方法。

传统二进制编码形式的数字值域只有两个 0 和 1，而对于有符号数字量编码的数字值域却有三个值，0，1 和-1，其中-1 经常表示为$\bar{1}$。这种将十进制数用2^n数相加减的形式表示的编码方式称为 SD 编码(Signed Digit Code)。

如将一个分数 X 用这种方法表示有如下的通用形式:

$$X=\sum_{K=1}^{L}S_K \bullet 2^{-P_K} \qquad (5-25)$$

其中 $S_K \in \{-1,0,1\}$ 并且 $p_K \in \{0,1,2,\cdots M\}$。根据式（5－25），0.375 可以表示为如下所示。

$$0.375=2^{-1}+(-1)\times 2^{-2}+2^{-3}=(0.1\bar{1}1)_{SD}$$

$$0.375=2^{-2}+2^{-3}=(0.011)_{SD}$$

$$0.375 = 2^{-1} - 2^{-3} = (0.10\bar{1})_{SD}$$

可知，SD 编码不是唯一的。我们称具有最少非零元素的表示法为正则有符号数字量表示法(Canonic Signed Digit)，也称作 CSD。

经典的 CSD 编码的方法是：从最低有效位开始，用 $10...0\bar{1}$ 代替所有大于或等于的 1 序列。例如: $111_2 = 100\bar{1}_{CSD}$。这种经典的 CSD 编码是独一无二的，而且另一个特性就是最终表达式在两个非零元素(1 或 $\bar{1}$)之间至少有一个 0。

经典 CSD 编码也不是总能够生成最佳的二进制编码，例如，11_2 的经典 CSD 编码就变成 $10\bar{1}_{CSD}$，非零元素的个数没有改变，但是字长却增加了 1 比特，这样在实现乘法器时就增加了硬件的复杂度。因此，需要对经典 CSD 编码方法进行修改，变为最佳 CSD 编码方法。最佳 CSD 编码将会给出一种非零元素最少的 CSD 编码，但也是减法最少的编码方法。

最佳 CSD 编码：

（1）从最低有效位开始，用 $10...0\bar{1}$ 取代所有大于 2 的 1 序列；

（2）用 $110\bar{1}$ 取代 1011；

（3）从最高有效位开始，用 011 代替 $10\bar{1}$。

常系数 FIR 滤波器中乘法器的硬件成本与系数 c_i 中非零元素的个数直接相关，最佳的 CSD 编码可以使系数 c_i 中非零元素的个数减至最少，因此利用最佳 CSD 编码可以减少常系数 FIR 滤波器中乘法器的资源使用量。

通过前面的讨论，二进制乘法的运算可以用移位相加来实现，CSD编码相对通常的二进制码的优点在于允许表达式中使用负数而能够用较少的非零位来表示乘数，用移位加/减的方式来实现乘法可以大大减少运算量，而减法相对加法只增加了非常少的硬件开销。

对于固定系数的FIR滤波器，将每个抽头系数都表示为最佳CSD码形式，将乘法运算转化为移位运算和加/减运算，实现无乘法滤波器，可以大大降低了硬件成本和功耗，并提高滤波器运算速度。

5.5 SRRC滤波器实现结果

本文所设计SRRC滤波器整体结构示意图如图5－20所示。

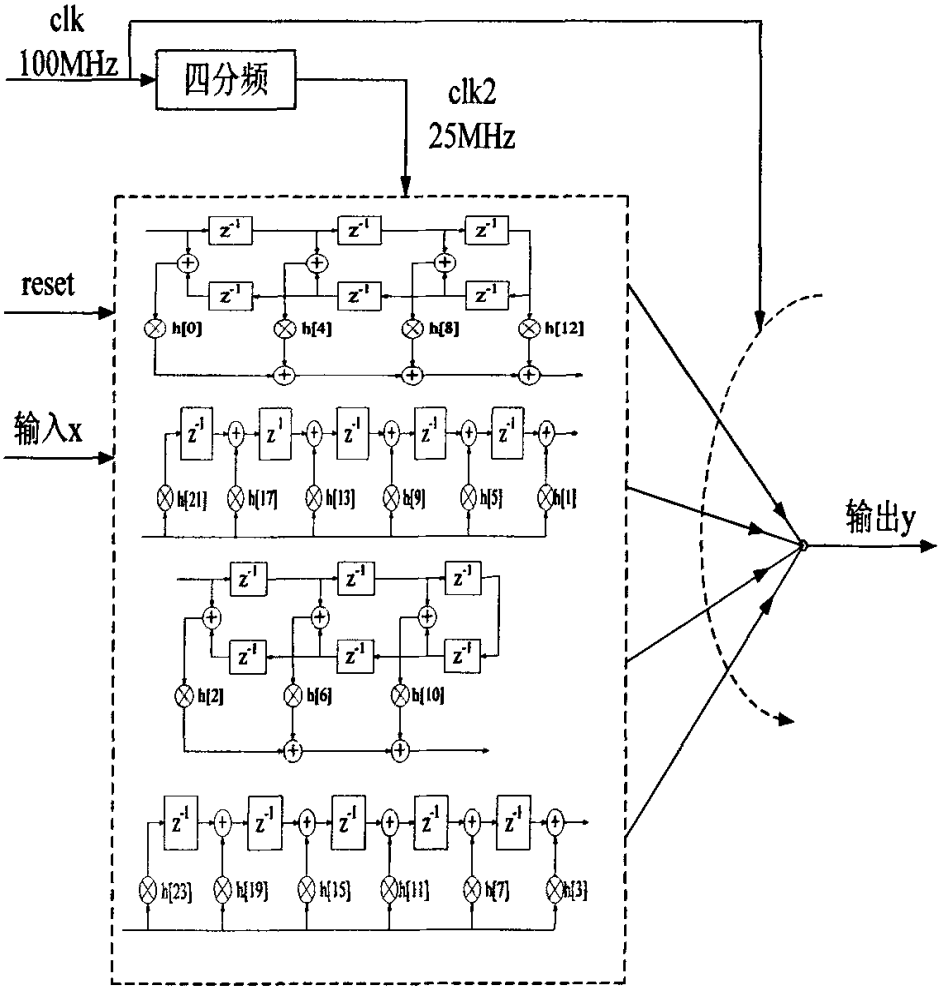

图5－20　SRRC滤波器整体结构示意图

利用 Quartus II 对设计进行综合，本文所设计的 SRRC 滤波器资源占用情况如表 5－4 所示，满足设计要求。

表 5－4　SRRC 滤波器资源占用情况

综合结果	Total logic element	Total pins	最高时钟频率
SRRC 滤波器	125	15	116MHz

利用 Quartus II 对设计进行仿真，波形图如图 5－21 所示。

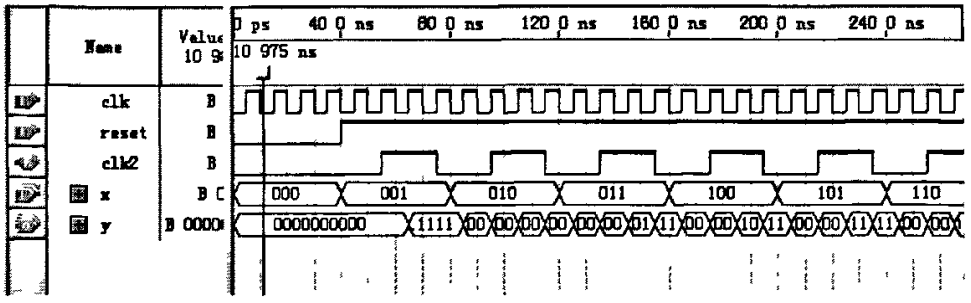

图 5－21　SRRC 滤波器仿真波形图

clk2 是由 100MHz 主时钟 clk 通过四分频电路产生的 25MHz 副时钟。每个 3bit 输入数据 *x* 占一个副时钟周期，每个 10bit 输出数据占一个主时钟周期。将输出的数据与利用 matlab 计算出的滤波后数据进行比对，结果基本一致，证明电路实现了预定的功能。

第六章 总 结

传统的数字信号处理算法是通过通用可编程 DSP 芯片来实现的，但是受到芯片处理能力的限制，只能处理低速数据流应用。本文采用 FPGA 芯片可以完成高速数据流处理。基于 SRAM 工艺的 FPGA 非常适合实现乘累加等最常用的 DSP 功能，以及那些借助于 LUT 来实现的算法功能。

基于 FPGA 设计全数字通信机，降低了电路复杂度，便于采用先进的算法提高通信机的性能指标，便于采用计算机辅助设计，实现电子设计自动化，便于移植、集成和大规模生产。自主研发的基带调制解调系统，不需采用专用的调制解调芯片，更有利于自身的技术积累和自我保护，更加灵活自由，不会受到专用调制解调芯片技术和知识产权方面的限制。

本文研究和设计的 OFDM 基带调制系统参考了 IEEE802.16d 标准，将 OFDM 基带调制解调所需的扰码、RS 码、卷积码、交织、16QAM 映射、OFDM 调制/解调（IFFT/FFT）、加循环前缀、基带成形滤波器、相关接口电路全部集成到一个 FPGA 芯片上，以实现片上系统（SOC）设计，大大简化系统的复杂度，提高系统的性能。

本文介绍了 OFDM 调制的原理和 OFDM 基带调制系统的总体设计，对系统中各个组成部分的设计和实现做了说明；介绍了 Verilog HDL 硬件描述语言和面向综合的现代数字设计方法，以及 FPGA 设计的基本原则；介绍了 RS 码的编码原理和时域迭代译码算法，在此基础上设计实现 RS 码编码器和译码器，对有限域乘法、有限域求逆运算等做了优化处理；介绍了成形滤波的原理和多种实现数字滤波器的硬件结构，采用多相结构设计实现了发送端平方根升余弦滚降滤波器。

本文的主要内容在算法和硬件结构上都得到了充分的验证，是完整可行、正确可靠的设计。本文设计的 RS 码编码器和译码器涉及有限域加法、乘法和求逆运算电路的设计和实现，利用有限域运算的特性和乘法运算中乘数固定的特点，可简化芯片设计，大幅度减小资源消耗。本文设计的平方根升余弦滚降滤波器是一个多速率系统，采用多相结构实现滤波器，大大简化了芯片设计。

本文初步完成了 OFDM 基带调制系统中 RS 码编译码器和成形滤波器的设计，

也达到了较好的性能。进一步的研究包括以下方面：

（1）检查硬件结构中各组合逻辑部分的关键路径，优化设计整体结构，缩短关键时延，从而提高电路工作的最高频率。

（2）在前面工作所搭建的开发平台和积累的丰富时间经验基础上，逐步实现 OFDM 基带系统其他单元如调制解调等部分的硬件结构，满足硬件设计的可移植性、可继承性和通用性要求。

参考文献

[1] 尹长川，罗涛，乐光新 编著，《多载波宽带无线通信技术》，北京邮电大学出版社，2004 年 7 月。

[2] Michael D. Ciletti 著，《Verilog HDL 高级数字设计》，电子工业出版社，2005 年 1 月。

[3] John F.Wakerly 著，《数字设计——原理与实践》（原书第三版），机械工业出版社，2003 年 8 月。

[4] 张欣著，《扩频通信数字基带信号处理算法及其 VLSI 实现》，科学出版社，2004 年 8 月。

[5] 张欣编著，《VLSI 数字信号处理——设计与实现》，科学出版社，2003 年 7 月。

[6] 夏宇闻编著，《Verilog 数字系统设计教程》，北京航空航天大学出版社，2005 年 1 月。

[7] 王金明，杨吉斌编著，《数字系统设计与 Verilog HDL》，电子工业出版社，2002 年 6 月。

[8] Uwe Meyer－Baese 著，《数字信号处理的 FPGA 实现》，清华大学出版社，2003 年 1 月。

[9] Sanjit K.Mitra 著，《数字信号处理——基于计算机的方法》，电子工业出版社，2005 年 1 月。

[10] 王新梅，肖国镇编著，《纠错码——原理与方法》，西安电子科技大学出版社，2003 年 5 月。

[11] EDA 先锋工作室王诚 吴继华等编著，《Altera FPGA/CPLD 设计》，人民邮电出版社，2005 年 7 月。

[12] 亿特科技 编著，《CPLD/FPGA 应用系统设计与产品开发》，人民邮电出版社，2005 年 7 月。

[13] IEEE Std 802.16－2004：Air Interface for Fixed Broadband Wireless Access Systems。

[14] IEEE Std 1364－2001：IEEE Standard Verilog Hardware Description Language.

[15] John G. Proakis, 《数字通信》（第四版），电子工业出版社，2003。

[16] Stephen H.hall,Garrett W.Hall,James A.McCall 著，《高速数字系统设计》，机械工业出版社，2005 年 8 月。

[17] Hanho Lee, High-Speed VLSI Architecture for Parallel Reed-Solomon Decoder,IEEE TRANSACTIONS,VOL. 11,NO.2,APRIL 2003.

[18] KUANG YUNG LIU,Architecture for VLSI Design of Reed-Solomon Decoders,IEEE TRANSACTIONS,VOL. C-33,NO.2,FEBRUARY 1984.

[19] Hanbo Lee,A High-Speed Low-Complexity Reed-Solomon Decoder for Optical Communications,IEEE TRANSACTIONS,VOL. 52,NO.8,AUGUST 2005.

[20] H.M. Shao,T.K. Truong,L.J.Deutsch,J.H.Yuen,I.S. Reed,A VLSI DESIGN OF A PIPELINE REED-SOLOMON DECODER, IEEE TRANSACTIONS,1985.

[21] Andy Miller,A Selection of Filter Case Studies for Wireless Systems,Xilinx Corporation

[22] www.altera.com

[23] Floyd M.Gardner,Interpolation in Digital Modems-PartI:Implementation and Performance,IEEE TRANSACTIONS ON COMMUNICATIONS,VOL. 41,NO.6,June 1993.

[24] Zhang Weiliang,Pan Changyong,Guo Xingbo,Yang Zhixing,Design and FPGA Implementation of High-Speed Square-root-raised-cosine FIR filters,IEEE,2002.

学术论文撰写及录用声明

入学以来，在导师张会生教授的悉心指导下，在完成课题研究的基础上，发表学术论文一篇。

刊物名称：《信息安全与通信保密》

论文题目：《基于 FPGA 实现 RS（255，239）编码器》

论文作者：李健，张会生

国际标准刊号：ISSN1009－8054

国内统一刊号：CN51－1608/TN

论文发表日期：2006 年第 12 期

致 谢

衷心感谢我的导师张会生教授在我论文研究和撰写期间给予的指导，通过他近三年的悉心指导和教诲使我的理论水平和独立科研能力得到了提高。正是听了张老师讲授的《通信原理》课程，使我对通信专业产生了浓厚的兴趣，并有幸在研究生期间继续跟随张老师学习。张老师高尚的品格与敬业精神、坚定果断的办事风格、精益求精的治学态度、循循善诱的悉心教导、敏锐的分析力与洞察力给我以深刻的印象与影响。

非常非常感谢我的父母和亲人，正是由于他们对我一贯无私无悔的关心、支持和鼓励，我才能完成这篇论文。

最后，向百忙之中阅读、评审论文的各位老师致以真诚地感谢！