

摘要

专用无线通信系统在当今发展迅速, 由于其具备特有的调度功能、组呼功能和快速呼叫的特性, 因此在通信领域发挥着巨大的作用。随着移动通信技术的发展, 专用无线通信系统正逐步经历由模拟向数字的转变。DMR(Digital Mobile Radio) 是国际 ETSI 组织提出的最新专用数字通信系统标准, 它所用的调制方式为 4FSK。

本文主要研究 DMR 基带传输技术的 DSP 算法实现。文中首先介绍了与 DMR 基带传输相关的物理层和数据链路层协议规定, 以及内插抽取、数字滤波技术、频率调制解调技术, 给出 DMR 端机基带传输的算法实现方案, 发端着重脉冲成形滤波器的设计实现、4FSK 基带四电平调制信号的产生, 接收端着重 4FSK 基带解调信号的抽样和判决, 对于判决, 文中使用自适应门限算法。此外, 在 TMS320VC5510 上, 本文设计基带处理的单工传输和双工实时传输的 DSP 程序。最后, 基于前面论述的基带传输算法和基带处理方案, 在 DMR 端机基带板硬件平台和基带板加射频板硬件平台上, 分别实现了基带传输和无线传输的单、双工语音传输及误码性能测试。

关键词: DMR 4FSK DSP 基带传输

Abstract

Trunked Radio System is growing rapidly today, due to its unique scheduling and group calling function, and characteristic of quick call, it has played a tremendous role in special communication field. During the development of mobile communication technology, trunking system is transferring from analog to digital system. DMR (Digital Mobile Radio) is the latest Digital Trunking Communication System standard proposed by the international organization ETSI, the modulation method of which is 4FSK.

The paper mainly studies DSP algorithm implementation of DMR baseband transmission technology. The paper outlines the physical and the data link layer protocol, interpolation and extraction, digital filtering technology, frequency demodulation technology, which are related to DMR baseband transmission. It gives the DSP algorithm implementation, mainly including pulse shaping filter design, the origination of 4FSK baseband four level modulated signal in transmitter, sampling and judgment of 4FSK baseband demodulated signal in receiver. For judgment, the paper proposes an algorithm for adaptive threshold. In addition, the paper gives simplex and duplex real-time DSP program on TMS320VC5510. Finally, on the hardware platform of baseband circuit board and baseband plus rf circuit board, respectively achieved baseband transmission and wireless transmission of simplex and duplex voice transmission and error performance testing.

Keyword: DMR 4FSK DSP Baseband Transmission

西安电子科技大学 学位论文独创性（或创新性）声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安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做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 邱婷 日期 2010.6.9

西安电子科技大学 关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。本人保证毕业后离校后，发表论文或使用论文工作成果时署名单位仍然为西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密的论文在解密后遵守此规定）

本学位论文属于保密，在___年解密后适用本授权书。

本人签名： 邱婷 日期 2010.6.9

导师签名： 杨彦伟 日期 2010年6月9日

第一章 绪论

1.1 专用无线通信系统发展概述

专用无线通信系统^[1]主要应用于集团调度指挥通信,通常用于群组呼叫、紧急呼叫、发起或接收与公网之间的呼叫等多种呼叫功能,同时可以为用户提供可靠的通信信道、快速建立通话、优先等级划分、动态重组能力等功能。由于具备特有的调度功能、组呼功能和快速呼叫的特性,因此在通信领域发挥着巨大的作用,其主要应用对象包括:

(1)对指挥调度功能要求较高的特殊部门和企业,主要包括政府部门(如军队、公安部门、国家安全部门和紧急事件服务部门)、铁道、水利、电力、民航等单位

(2)普通行业用户,如出租、物流、物业管理和工厂制造业等

早期的专用无线通信主要是由点对点无线电对讲机完成,上世纪八十年代发展成为单频道、单基地台的模拟系统。引入多频道共享技术之后,发展成为第一代模拟专用无线通信系统。

随着数字技术的发展,专用无线通信系统已经向第二代的数字技术发展。数字专用无线通信在信令、多址方式、话音编码、调制解调和信道控制等各个关键技术环节全面采用数字化处理,同时基于数字系统的特点结合同步技术、检错纠错技术以及分集技术等,使数字专用无线通信具有抗干扰能力强、高质量远距离传输、高保密度、高可靠性和高度灵活的业务适应能力和外设接口的连接能力等特殊性能,而且整个数字专用无线通信系统的容量更大,系统联网更方便。与蜂窝移动通信系统的公众网相比较,数字无线专用移动通信系统的专用网更强调组呼、广播呼叫、紧急呼叫、强插强拆和不同的呼叫优先级别等调度指挥功能;更强调缩短呼叫的建立时间,尤其是在紧急情况下,希望一按即通(小于 0.5s),否则会造成相当严重的后果;调度指挥要求简明扼要,一般通话时间为几十秒。随着移动通信技术的发展,人们尽力求同存异,使开发的专业移动通信系统能尽可能满足多个不同专业部门的需求,数字专用无线移动通信系统就是在这种背景下发展起来的。

目前市场上较为成功的数字专用无线系统有欧洲的 TETRA 和摩托罗拉的 iDEN。TETRA 是基于传统大区制调度通信系统的数字化而形成的一个专用移动通信无线电标准,它大量借鉴了 GSM 的概念,采用 TDMA 多址方式和类似的逻辑信道,与传统模拟系统相比,在网络结构中增加了 HLR/VLR,增强了移动性管理;而 iDEN 基于 TDMA 多址方式的调度通信/蜂窝双工电话组合系统,在传统大区制调度通信基础上,大量吸收数字蜂窝通信系统的特点,采用小区复用蜂窝结构,

提高了网络覆盖能力。

从这两个标准可以看出，数字专用无线通信系统发展的一个方向是在传统专用移动通信的调度功能之外，还提供了类似公用移动电话系统的双工通话和短信服务(SMS)，可以多基地台覆盖范围漫游，甚至具有越区切换等功能；除此之外，高速数据通信能力的强化、与 IP 网络的整合等，也是数字专用无线通信系统非常重要的发展方向。

1.2 DMR 通信协议介绍

在数字专用无线通信系统发展的催化作用下，欧洲电信标准协会(ETSI)于 2004 年提出新型数字专用无线通信系统 DMR (Digital Mobile Radio)。DMR 协议文档描述了一个可升级的数字无线移动系统的空中接口，覆盖以下三个阶段的产品^[2]：

阶段1:DMR设备具有一个整体的天线，以直通为工作模式，无个体业务鉴权。

阶段2:在直通或基站转发模式中，需个体业务鉴权的DMR系统操作。

阶段3:第三阶段的设备是指可自动管理通信的DMR专用无线通信系统。

从整体结构上来讲，DMR 协议遵从一般的分层结构，如图 1.1 所示为 DMR 标准所定义的分层结构模型。

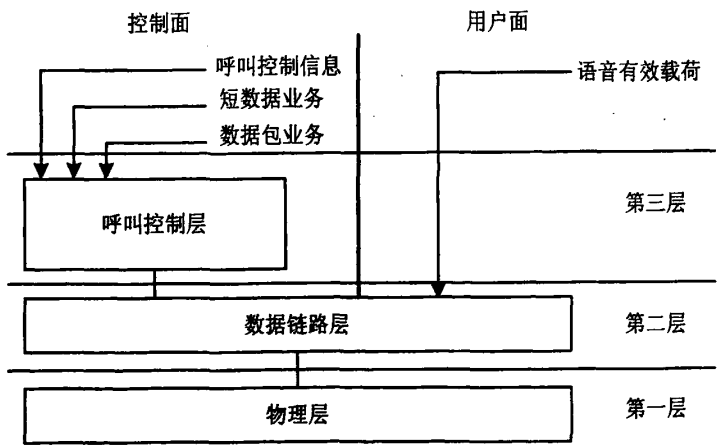


图 1.1 DMR 协议栈

协议栈的第一层即物理层。它支持在物理媒介上传输比特流所需要的全部功能^[3]，其中包含：调制和解调；发送和接收的转换；射频特性；比特和符号的定义；频率和符号的同步；突发的形成等。

协议栈的第二层即数据链路层。它接受物理层的服务，并向第三层提供服务。其中包含：信道编码（FEC，CRC）；交织/去交织和比特排序；确认和重传机制；媒质接入控制和信道管理；帧、超帧的形成和帧同步；突发和参数定义；链路寻

址(源/目的地); 话音应用(编解码数据)与物理层的接口; 数据承载业务; 与呼叫控制层进行信令与用户数据的交换等。

协议栈的第三层即呼叫控制层。DMR 空中接口第三层只适用于控制面, 负责呼叫控制, 提供 DMR 支持的服务, 支持数据业务。呼叫控制层是公共设施和服务的实体, 它得到第二层的支持。DMR 空中接口呼叫控制层的具体功能包含: 基站激活/去激活; 呼叫建立, 呼叫保持, 呼叫中止; 单呼和群呼的发送与接收; 目的寻址; 支持固定业务; 数据呼叫控制; 通知信令等。

DMR 与 TETRA 系统和 iDEN 系统相比具有技术简单, 成本较低的优点, 并且支持从模拟到数字的过渡。DMR 的发展无论是在国内还是在国外都处于起步阶段。由于还没有完全成型, 现在进行跟踪就可以与国际上其它的研究机构在技术上保持同步, 有巨大的研究开发价值。

1.3 论文结构及主要内容

本文所有研究基于 DMR 协议阶段一端到端直通模式, 重点研究端机基带传输技术。主要内容可分为以下几个部分:

第一章介绍了专用无线通信系统发展概况以及发展方向, 并简要介绍了 DMR 协议结构, 最后对本文的章节安排做了介绍。

第二章首先介绍了与 DMR 端机基带传输技术相关的协议规定, 包括 DMR 帧结构、DMR 同步与定时以及 DMR 差错控制技术。在此基础上, 介绍了与 DMR 基带传输紧密相关的理论, 包括内插抽取理论、数字滤波技术, 频率调制与解调技术。

第三章做 DMR 基带传输技术的算法仿真研究, 给出在实际实现中的算法。首先给出了 DMR 基带传输系统发送端的算法研究与实现, 包括基带成形滤波器设计、卷积法与波形存储法产生四电平调制信号的 CCS 仿真研究。然后, 给出 DMR 基带传输系统接收端的算法实现。最后, 将 DMR 基带信号, 设计采用 4FSK 数字正交调制解调的模型, 给出 MATLAB 仿真研究。

第四章首先对 DMR 端机基带处理平台作相关介绍。然后, 主要对 DSP 与语音芯片 AMBE、DSP 与主控制器 MCU 之间的数据传输和中断处理方式给出介绍。最后在 TMS320VC5510 上编写 DMR 基带传输的 C 语言程序, 包括单工通信和双工实时通信的实现。

第五章在硬件平台上, 给出 DMR 端机的系列测试方法和结果, 包括基带板对接测试, 以及基带板加射频板的无线传输测试。

结束语对本文工作进行了总结, 并提出了研究的不足以及将来的研究方向。

第二章 DMR 基带传输技术基础

与传统模拟对讲机不同, DMR 端机是将信号数字化, 并以数字编码形式传播。也就是说, DMR 对语音信号的处理、调制均为数字处理。本章首先介绍与基带传输处理相关的 DMR 协议规定, 然后, 对 DMR 基带传输相关的采样率转换, 滤波处理, 调制解调等原理进行阐述。

2.1 DMR 协议相关介绍

DMR 端机基带传输处理倾向于物理层处理, 同时也涉及了数据链路层中的差错控制技术, 以下分别介绍。

2.1.1 DMR 帧结构

DMR 协议采用一种双时隙的 TDMA 结构, 其时隙结构如图 2.1 所示。一个 TDMA 帧长为 60ms, 包含两个时隙, 一个时隙传输一个突发, 每个时隙长 30ms。一般来说, 两个不同的呼叫占用两个不同的时隙。

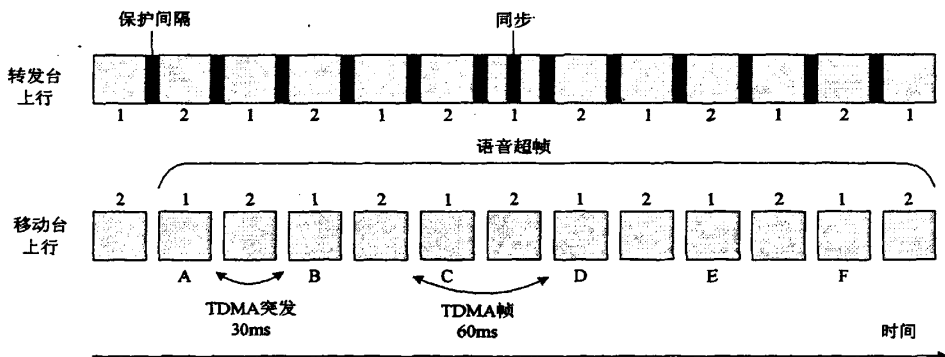


图 2.1 DMR 时隙结构图

DMR 协议中所列的突发结构和信道包括语音突发、一般的数据突发和公共广播信道。这些突发包括用户数据和嵌入协议数据单元的信令。突发结构包含两个 108 比特的有效负荷域和一个 48 比特同步或信令域, 如图 2.2 所示。每一个突发总长度为 30ms, 其中 27.5ms 用于 264 比特内容。在进行语音传输时, 可以利用 216 比特业务承载 60ms 的压缩语音信息。余下的 2.5ms 是保护时间。

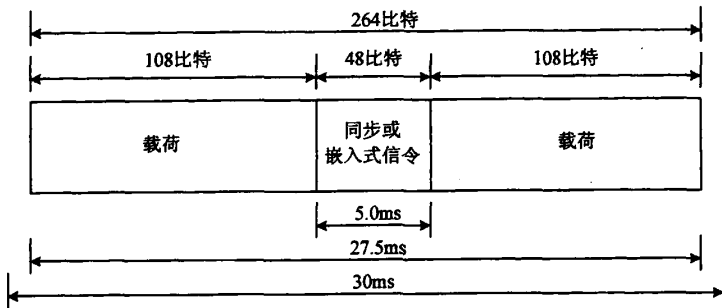


图2.2 一般突发结构

语音帧的一般格式如图 2.3 所示，具体分为两种形式：承载同步信令的语音帧格式和承载嵌入式信令的语音帧格式。每帧承载的语音信息为 216bit，分别在帧的两边，48bit 同步码和嵌入式信令均填充在帧格式的中间。

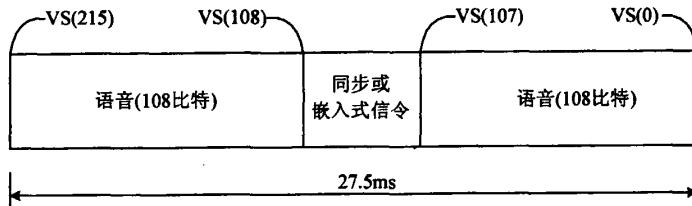


图 2.3 语音帧的一般格式

用于数据和控制信息的有效负载减少为两个 98 的有效负载，留下 20 比特用于附加数据类型域定义。数据和控制帧的一般格式如图 2.4 所示，其中的 196 比特信息可以是各种信令。对于不同的信令，协议规定了不同的编码方式。各种信令的含义是根据 20 比特的时隙类型来定义区分的。而同步码以及嵌入式信令填充在数据帧中的位置与在语音帧中的位置相同，均在帧格式的中间。

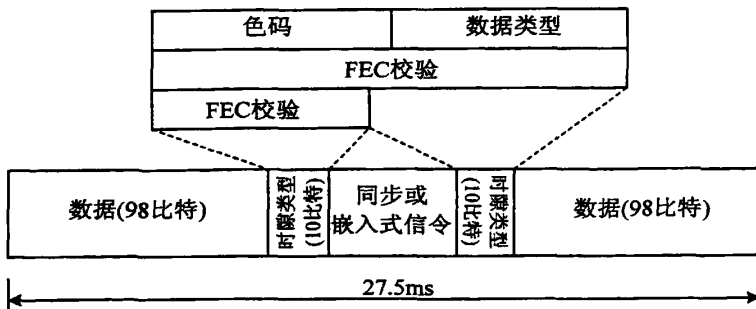


图 2.4 数据帧的一般格式

DMR语音使用360ms的超帧来传输，一个超帧包含6个突发，用符号A~F来表示。数据和控制信息没有超帧结构，这些突发在需要时可含有一个同步模式或者承载嵌入信令。对于数据和控制消息，嵌入区域仅仅是数据同步模式或者反向信道信令；对于话音呼叫，语音同步模式将出现在每个语音超帧的第一个突发，除了标记超帧边界外，周期的插入同步信号可以允许接收方在传输开始之后按同步模式调整好的时隙定时获得话音信息。每次语音开始之前，也就是突发A之前会有

一个语音帧头 (LC Header) 区别传输信息的类别并标志语音的开始。在语音信息传输过程中, 完整的 TDMA 超帧不断重复, 若发送结束时语音长度不能满足超帧结构, 将会延长语音长度组成完整的超帧进行传输后通知结束。突发 A 包含语音同步模式标注超帧的开始, 突发 B~F 以承载信令取代了同步模式。

语音超帧结构如图 2.5 所示。

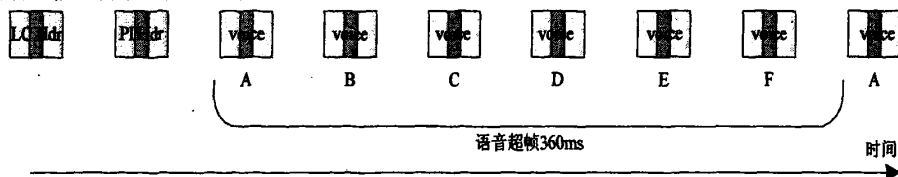


图 2.5 语音超帧结构

2.1.2 DMR 的同步与定时

DMR 的同步包括位同步和帧同步。在数字通信中, 信息流是用若干码元组成一个帧。在接收这些数字信息时, 必须知道这些帧的起止时刻, 否则接收端无法正确恢复信息。为了使接收端能正确分离各路信号, 在发送端必须提供每帧的起止标记, 在接收端检测并获取这一标志的过程, 称为帧同步。接收时还需要知道每个码元的起止时刻, 以便在恰当的时刻进行取样判决。这就要求接收端必须提供一个位定时脉冲序列, 该序列的重复频率与码元速率相同, 相位与最佳取样判决时刻一致。提取这种定时脉冲序列的过程即称为位同步。

实现帧同步, 通常采用的方法是起止式同步法和插入特殊同步码组的同步法。DMR 系统采用连贯式插入法进行帧同步, 又称集中插入法。它是指在每一信息群中集中插入作为帧同步码组的特殊码组, 该码组应在信息码中很少出现, 即使偶尔出现, 也不可能依照帧的规律周期出现。接收端按帧的周期连续数次检测该特殊码组, 这样便可获得帧同步信息。连贯插入法的关键是寻找实现帧同步的特殊码组。对帧同步码的要求包括: 具有尖锐单峰特性的自相关函数、漏同步概率小; 便于与信息码区别、假同步概率小; 码长适当, 以保证传输效率。

DMR 系统对采用的帧同步码组有严格的规定: 帧同步码组插入在帧的中心位置, 并且对于输入输出语音, 数据或控制信息, 由不同的帧同步码组来区分。具体帧同步码组如表 2.1 所示^[3]。根据 DMR 协议规定的映射规则将语音和数据的同步码进行映射, 得到的 24 个符号均为 +3 或 -3, 并且语音同步码与数据同步码的极性相反, 因此帧同步相关器为语音模式提供一个正的结果, 为数据模式提供相等但为负的结果, 通过识别相关器的输出极性即可判断出传输的是语音突发还是数据突发。

表 2.1 帧同步码组模型

下行													
语音	十六	7	5	5	F	D	7	D	F	7	5	F	7
二	0111	0101	0101	1111	1101	0111	1101	1111	0111	0101	1111	0111	0111
数据	十六	D	F	F	5	7	D	7	5	D	F	5	D
二	1101	1111	1111	0101	0111	1101	0111	0101	1101	1111	0101	1101	1101
上行													
语音	十六	7	F	7	D	5	D	D	5	7	D	F	D
二	0111	1111	0111	1101	0101	1101	1101	0101	0111	1101	1111	1101	1101
数据	十六	D	5	D	7	F	7	7	F	D	7	5	7
二	1101	0101	1101	0111	1111	0111	0111	1111	1101	0111	0101	0111	0111
RC同步	十六	7	7	D	5	5	F	7	D	F	D	7	7
二	0111	0111	1101	0101	0101	1111	0111	1101	1111	1101	0111	0111	0111
保留同步模型													
见注释	十六	D	D	7	F	F	5	D	7	5	7	D	D
二	1101	1101	0111	1111	1111	0101	1101	0111	0101	0111	1101	1101	1101
注释	保留同步模型为将来使用												

在DMR的直通模式下，前向路径定时作为反向信道定时的参考基准。一旦源移动台停止传输，另一个移动台将发送消息异步并且建立一个新的独立的定时参考。

图2.6给出了直通模式业务的定时框图^[3]例子。在这个例子中，移动台在前向信道上传输，该信道是TDMA物理信道中的一个。

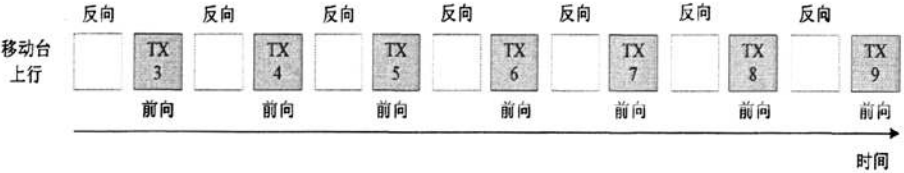


图2.6 直通模式定时

时分双工的定时关系^[3]如图2.7示。在这个例子中，移动台MS在入境(inbound)信道2上传输语音在出境(outbound)信道2上侦听语音。

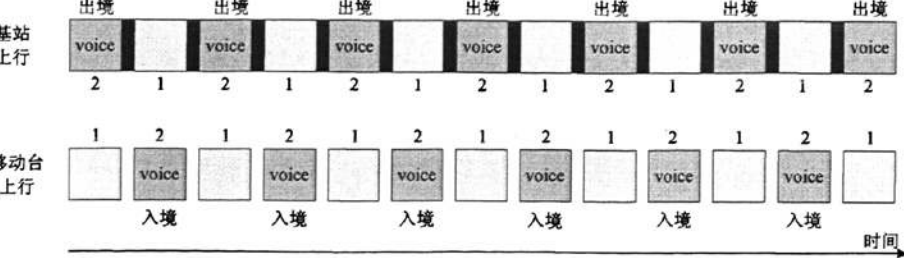


图2.7 TDD语音定时

2.1.3 DMR 差错控制技术

差错控制技术^[4]的基本思想是在发送端将被传输的信息上附上一些监督码元，这些多余的码元与信息码元之间以某种确定的规则相互关联(约束)。接收端按照既定的规则校验信息码元与监督码元之间的关系，一旦传输发生差错，则信息

码元与监督码元的关系就受到破坏,从而接收端就可以发现错误乃至纠正错误。

DMR 系统中涉及的差错控制编码理论,包含校验编码,FEC 前向纠错编码和交织理论。DMR 协议规定不同的信令对应不同的校验编码方式,其主要形式有两种,为 CRC 校验码和 RS 码^[5]。DMR 协议中规定的 FEC 码的种类主要有汉明码(Hamming)、RS 码、golay 码、多块拓扑码(Block Product Turbo Code, BPTC)和可变长度多块拓扑码(Variable Length BPTC)等。

块生成 Turbo 码即 BPTC(Block Product Turbo Code)。BPTC 码作为一种能同时纠正随机错误和突发错误的乘积码,可用图 2.8 所示的矩形阵列来表示。DMR 系统规定在 BPTC(196,96)码中,信息位是由 96 比特添加过校验位的信令和 3 比特保留位(置为 0)构成, C_1 采用 Hamming(15,11,3)码, C_2 为 Hamming(13,9,3)码。所以 $C_1 \times C_2$ 的最小距离有

$$d_{\min} = d_{1,\min} \times d_{2,\min} \quad \text{式 (2-1)}$$

它可以纠正 $(d_{\min} - 1)/2$ 个随机错误。在 BPTC(196,96)码中,即对每一信息块可以纠正 4 个错误。

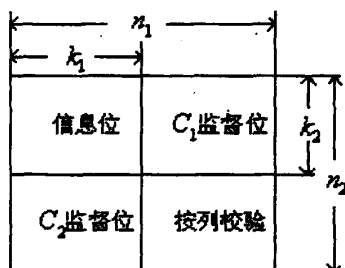


图 2.8 BPTC 码的码阵列

DMR 协议中还规定了三种可变长度 BPTC,分别用于嵌入式信令,反向信道信令和公共广播信令。

用于嵌入式信令的可变长度 BPTC 码包含对行的 Hamming(16,11,4)编码和对列的简单奇偶校验编码。

用于反向信道信令的可变长度 BPTC 码是嵌入式信令编码的一个特例,其编码方式与用于嵌入式信令的 BPTC 编码方式相同,也是包含对行的 Hamming(16,11,4)编码和对列的简单奇偶校验编码,只是采用了不同的交织技术。

用于公共广播信令的可变长度 BPTC 码包含对行的 Hamming(17,12,3)编码和对列的简单奇偶校验编码。

针对突发差错信道的交织器对噪声随机化非常有效,但进一步观察发现,交织前的码字经交织后不再是码字,这种非码字符号直接送入信道传输显然对差错控制不利。例如常用的“行”进“列”出交织,“行”顺序的 (n,k) 分组码换成“列”顺序后就不是码字了。若将交织块的行和列都加以编码,则码字经行、列交织后仍是码

字, 纠错能力一定得以提高, 这就导致了乘积码的产生。乘积码是既能纠正随机错误, 又能纠正突发错误的码类, 特别适合用于信道干扰很复杂的、组合信道所使用差错控制系统中。

2.2 多速率数字信号处理概述

一个数字信号处理系统, 通常不会是从输入到输出都是单一采样率。为满足处理速度要求以及滤波器的可实现性, 需采用多速率处理技术。DMR 基带传输系统中采用的多速率处理技术包括整数倍内插和抽取。

2.2.1 整数倍内插

内插理论是软件无线电发射机中数字上变频技术的理论基础。内插是在保持信号频谱不变的情况下提高信号采样率。内插方法很多, 一个最简单的方法就是在每个有效的输入采样之间插入 $(I-1)$ 个零值, 使原信号采样率增加 I 倍, I 被称为内插因子或内插率。

设有序列 $x(n)$, I 倍内插后得到序列 $y(m)$

$$y(m) = \begin{cases} x(m/I) & m = 0, \pm I, \pm 2I, \dots, \pm nI \dots \\ 0 & m \neq 0, \pm I, \pm 2I, \dots, \pm nI, \dots \end{cases} \quad \text{式(2-2)}$$

对 $y(m)$ 进行 Z 变换, 可推导出:

$$Y(z) = \sum_{m=-\infty}^{+\infty} Y(m)z^{-m} = \sum_{m=-\infty}^{+\infty} Y(mI)z^{-mI} = \sum_{m=-\infty}^{+\infty} X(m)z^{-mI} = X(z^I) \quad \text{式(2-3)}$$

将 $z = e^{j\omega}$ 代入式(2.2), 得到内插前后信号的频谱关系为:

$$Y(e^{j\omega}) = X(e^{j\omega I}) \quad \text{式(2-4)}$$

由上式 2-4 可见, 内插后的信号频谱为原始序列谱经 I 倍压缩后得到的谱。在 $Y(e^{j\omega})$ 中不仅含有 $X(e^{j\omega})$ 的基带分量, 而且还有其频率大于 π/I 的高频成分。因此, 只要对内插的信号进行去镜像滤波 (低通滤波器的带宽为 π/I) 就能恢复出原始谱。经滤波后, 原来插入的零值点就变为 $x(n)$ 的准确内插值。由此可见, 内插可以大大提高时域分辨率。这也是在调制器中经常引入内插模块的原因, 即通过提高时域分辨率来提高信号的采样速率。完整的 I 倍内插器结构如图 2.9 所示。



图 2.9 I 倍内插器结构

2.2.2 整数倍抽取

抽取理论是软件无线电接收机中数字下变频技术的理论基础。整数倍抽取, 就是当信号 A/D 采样后的数据流速率很高时, 为降低速率以便处理和计算。将原

始采样序列 $x(n)$ 每隔 $(D-1)$ 个数据取一个, 以形成一个新序列 $y(m)$, 正整数 D 称为抽取因子。

$$y(m) = x(mD) \quad \text{式(2-5)}$$

假设一个新的信号 $x'(n)$, 当 $n = 0, \pm D, \pm 2D \dots$ 时, $x'(n) = x(n)$; 当 n 为其它值时, $x'(n) = 0$ 。下面对式(2-4)两边进行离散傅里叶变换, 推导过程如下:

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{+\infty} y(m)z^{-m} = \sum_{m=-\infty}^{+\infty} x(mD)z^{-m} \\ &= \sum_{m=-\infty}^{+\infty} x'(m)z^{-\frac{m}{D}} = \sum_{m=-\infty}^{+\infty} \left\{ x(m) \left[\frac{1}{D} \sum_{l=0}^{D-1} e^{j\frac{2\pi lm}{D}} \right] \right\} z^{-\frac{m}{D}} \\ &= \frac{1}{D} \sum_{l=0}^{D-1} X \left[e^{j\frac{-2\pi l}{D}} \cdot z^{\frac{1}{D}} \right] \end{aligned} \quad \text{式(2-6)}$$

将 $z = e^{j\omega}$ 代入式(2-6)可得抽取前后信号的频谱关系:

$$Y(e^{j\omega}) = \frac{1}{D} \sum_{l=0}^{D-1} X \left[e^{j\frac{(\omega-2\pi l)}{D}} \right] \quad \text{式(2-7)}$$

由式(2-7)看出, 抽取后序列的频谱为抽取前原序列的频谱经频移和 D 倍展宽后的 D 个频谱叠加和。显然, 抽取后的频谱会发生混叠现象, 使得无法从 $Y(e^{j\omega})$ 中恢复出原信号的频谱分量。但是如果事先对 $X(e^{j\omega})$ 进行抗混叠滤波处理(低通滤波器的带宽为 π/D), 使得 $X(e^{j\omega})$ 中只含有小于 π/D 的频谱分量, 经过 D 倍抽取后, 信号的频谱就不会发生严重的混叠现象。完整的 D 倍抽取器结构如图 2.10 所示。

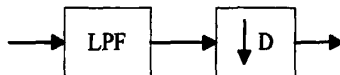


图 2.10 D 倍抽取器结构

2.3 DMR 基带传输中的数字滤波技术

数字滤波是数字信号处理的基本方法。数字滤波与模拟滤波相比有很多优点, 它除了可避免模拟滤波器固有的电压漂移、温度漂移和噪声等问题外, 还能满足滤波器对幅度和相位的严格要求。数字滤波器通常是以有限精度算法实现线性时不变离散系统来完成对信号的滤波处理。它具有精确度高, 稳定性好, 灵活性大等优点。随着各种高速数字处理器的发展, 特别是在 DSP 系统中, 数字滤波器的应用日益广泛。本节中, 我们讨论与 DMR 基带传输紧密相关的脉冲成形滤波技术和有限长脉冲响应(FIR)数字滤波器设计。

2.3.1 基带脉冲成形数字滤波技术

数字基带传输系统的模型^[6]如下图 2.11 所示。在数字基带信号送入调制器前, 波形是矩形脉冲, 由于矩形脉冲上升沿和下降沿是突变的, 高频成分比较丰富,

占用频带宽。通过带限信道时,每个符号的脉冲将扩展到相邻符号的码元内,造成码间串扰(ISI),导致接收机检测误码概率增大。常用的减少码间串扰的方法是增加信道带宽,但是,无线通信系统要求在减少码间串扰条件下,占用带宽小,并尽可能减小调制带外辐射,提高频带利用率。因此需要设计脉冲成形滤波器,产生既减小码间串扰又不会占用过多额外带宽的更适合于信道传输的波形。

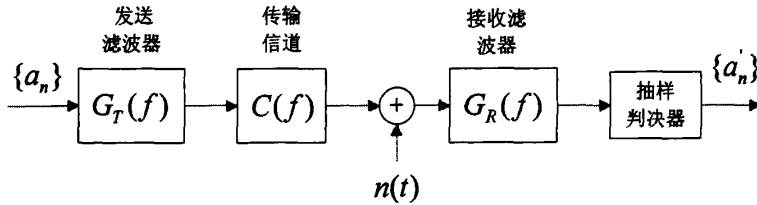


图 2.11 数字基带传输系统模型

根据奈奎斯特第一准则^[7],只要把通信系统(包括发送滤波器,信道和接收滤波器的)整个响应设计为在接收机端的每个抽样时刻只对当前符号有响应,而对其他符号的响应全等于零,那么码间串扰(ISI)的影响就能完全抵消。也就是说,一个信号 $x(t)$ 没有码间串扰的充分必要条件为:

$$x(nT) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad \text{式(2-8)}$$

它的傅里叶变换 $X(f)$ 满足

$$\sum_{m=-\infty}^{\infty} X(f + \frac{m}{T}) = T \quad \text{式(2-9)}$$

式中 $1/T$ 是符号率。

一般来说,很多滤波器都满足上述条件。在无线通信中最常用的满足该条件的脉冲成形滤波器是升余弦滚降滤波器。DMR 协议规定的成形滤波器为下式^[3],其中 α 为滚降系数, $0 \leq \alpha \leq 1$ 。

$$N(f) = \begin{cases} 1 & 0 \leq |f| \leq \frac{1-\alpha}{2T} \\ \frac{1}{2} \left\{ 1 + \cos \left[\frac{\pi(2T|f| - 1 + \alpha)}{2\alpha} \right] \right\} & \frac{1-\alpha}{2T} \leq |f| \leq \frac{1+\alpha}{2T} \\ 0 & |f| \geq \frac{1+\alpha}{2T} \end{cases} \quad \text{式(2-10)}$$

在设计过程中,可认为传输信道是理想的,即图 2.11 中, $C(f) = 1$ 。此时系统频谱特性表达式为:

$$N(f) = G_T(f)G_R(f) \quad \text{式(2-11)}$$

实际通信系统中,升余弦滚降滤波器可以通过在发射机端和接收机端使用同样的平方根升余弦滚降滤波器来实现,即 $G_T(f) = G_R(f)$ 。这样在平坦衰落信道中为实现最佳性能提供了匹配滤波^[8],平方根升余弦滚降滤波器频率响应如下:

$$H(f) = \sqrt{N(f)} = \begin{cases} 1 & 0 \leq |f| \leq \frac{1-\alpha}{2T} \\ \sqrt{\frac{1}{2} \left[1 + \cos \left[\frac{\pi(2T|f| - 1 + \alpha)}{2\alpha} \right] \right]} & \frac{1-\alpha}{2T} \leq |f| \leq \frac{1+\alpha}{2T} \\ 0 & |f| \geq \frac{1+\alpha}{2T} \end{cases} \quad \text{式(2-12)}$$

由式 (2-3) 可以得到平方根升余弦滚降滤波器时域表达式:

$$h(t) = \frac{4\alpha T / \pi}{T^2 - 16\alpha^2 t^2} \cos[\pi(1+\alpha)] + \frac{T / \pi}{T^2 - 16\alpha^2 t^2} \cos[\pi(1-\alpha)] \quad \text{式(2-13)}$$

该滤波器响应属于频域有限, 时域无限, 在实现系统中不可能实现。因此, 采用 FIR 滤波器的设计方法, 对 $h(t)$ 进行对称截断、抽样平移, 得到相应逼近的 FIR 数字滤波器。

2.3.2 FIR 滤波器设计

低通有限冲激响应滤波器(低通 FIR 滤波器)有其独特的优点^[9], 因为 FIR 系统只有零点, 因此, 系统总是稳定的, 而且容易实现线性相位和允许实现多通道滤波器。目前关于 FIR 滤波器的设计方法主要有三种, 即窗函数法, 频率取样法和切比雪夫等波纹逼近的最优化设计方法。其中窗函数设计法简单、直接便于理解。本文对窗函数设计法^[10]的基本思想进行阐述。

FIR 滤波器的设计问题, 就是要使所设计的 FIR 滤波器的频率响应 $H(e^{j\omega})$ 去逼近所要求的理想的滤波器响应 $H_d(e^{j\omega})$ 。由上节知, DMR 要设计的成形滤波器 $h_d(n)$ 为无限长, 所以要用有限长的 $h(n)$ 来逼近, 最有效的方法是用一个有限长度的窗口函数序列 $\omega(n)$ 来截取 $h_d(n)$, 即

$$h(n) = \omega(n)h_d(n) \quad \text{式(2-14)}$$

因此, 实际设计出的 FIR 滤波器的频率特性是理想低通滤波器的幅度频率特性和窗函数的幅度频率特性的复卷积。加窗函数后, 对滤波器的理想特性的影响有以下几点:

- (1)使得 $H_d(e^{j\omega})$ 在截止频率的间断点变成了连续的曲线, 使得设计的实际滤波器出现了一个过渡带, 它的宽度等于窗函数的主瓣的宽度。由此可见, 如果窗函数的主瓣越宽, 过渡带就越宽。
- (2)由于窗函数旁瓣的影响, 使得滤波器的幅度频率特性出现了波动。旁瓣范围的面积越大, 通带波动和阻带的波动就越大。
- (3)增加窗函数的长度 N , 只能相应的减小过渡带, 不能减少主瓣和旁瓣的相对值, 即不能改变滤波器的波动程度。

实际应用中应选择主瓣宽度窄、旁瓣幅度低且下降速率快的窗函数。为了满

足工程上的需要，可以通过改变窗函数的形状来改善滤波器的频率特性。

本文式 2-15、2-16、2-17、2-18 列出了四种窗函数的表达式，表 2.2 列出了四种窗函数的基本技术指标参数，这四种窗均满足 A 类线性相位条件：

$\omega(n)=\omega(N-1-n)$ ，其中 N 为滤波器长度。

矩形窗： $\omega(n)=1$ 式(2-15)

汉明窗： $\omega(n)=0.54-0.46\cos\frac{2\pi n}{N-1}$ 式(2-16)

汉宁窗： $\omega(n)=0.5(1-\cos\frac{2\pi n}{N-1})$ 式(2-17)

布莱克曼窗： $\omega(n)=0.42-0.5\cos\frac{2\pi n}{N-1}+0.08\cos\frac{4\pi n}{N-1}$ 式(2-18)

表 2.2 四种窗函数的基本参数

窗函数	旁瓣峰值幅度/dB	过渡带宽	阻带最小衰减/dB
矩形窗	-13	$4\pi/N$	-21
汉宁窗	-31	$8\pi/N$	-44
汉明窗	-41	$8\pi/N$	-53
布莱克曼窗	-57	$12\pi/N$	-74

2.4 频率调制(FM)与解调技术

调制与解调是现代通信的重要手段。调制是在发射端用一个载波来运载基带信号，使载波信号的某一个(或几个)参量随基带信号的变化而变化。解调是调制的逆过程，它是在接受端从已调制的信号中恢复出基带信号。DMR 所用的调制方式为频率调制，以下对频率调制解调技术进行介绍。

2.4.1 频率调制原理

调频为角度调制的一种，角度调制的一般表达式为^[1]：

$S_m(t)=A\cos(\omega_c t+\varphi(t))$ 式(2-19)

频率调制是利用基带信号调制载波的频率得到的，即瞬时频偏随调制信号 $m(t)$ 线性变化，满足：

$\frac{d\varphi(t)}{dt}=K_fm(t)$ 式(2-20)

则调频信号表达式可以写为：

$S(t)=A\cos\left(2\pi f_c t+2\pi K_f\int_0^t m(t)dt\right)$ 式(2-21)

其中, A 为载波幅度, f_c 为载波频率, K_f 为频偏常数, $m(t)$ 为调制信号。调频信号的产生有直接法和间接法两种, 以下分别介绍。

2.4.2 调频信号的直接法产生

直接法调频就是用调制信号直接去控制一个高频振荡器, 通常使用的是压控振荡器(VCO), 使输出的瞬时频率正比于调制信号 $m(t)$ 的幅度。

$$\omega_i(t) = \omega_c + K_f m(t) \quad \text{式(2-22)}$$

VCO 的优点是可以直接获得大的频偏, 但由于频率-幅度特性的非线性, 频偏受到一定的限制。另外, 调频信号直接作用到振荡器也容易产生中心频率的漂移。在要求载波频率稳定度比较高, 同时频偏比较大的情况下, 采用直接法调频就难以实现。本项目中为了保证调制器具有良好的低频调制特性, 采用两点注入式宽带 FM 调制结构。

2.4.3 调频信号的非相干解调法

由于调频信号的瞬时频率正比于调制信号的幅度, 因而调频信号的解调器必须能产生正比于输入频率的输出电压, 也就是当输入调频信号为

$$s(t) = A \cos \left[\omega_c t + K_f \int_{-\infty}^t m(\tau) d\tau \right] \quad \text{式(2-23)}$$

解调器的输出应当为

$$m_o(t) \propto K_f m(t) \quad \text{式(2-24)}$$

最简单的解调器是具有频率-电压转换特性的鉴频器。理想鉴频器可看成是带微分器的包络检波器, 微分器输出

$$s_d(t) = -A [\omega_c + K_f m(t)] \sin \left[\omega_c t + K_f \int_{-\infty}^t m(\tau) d\tau \right] \quad \text{式(2-25)}$$

用包络检波器将其幅度变化取出, 并滤去直流后输出

$$m_o(t) = K_d K_f m(t) \quad \text{式(2-26)}$$

其中 K_d 称为鉴频器灵敏度。

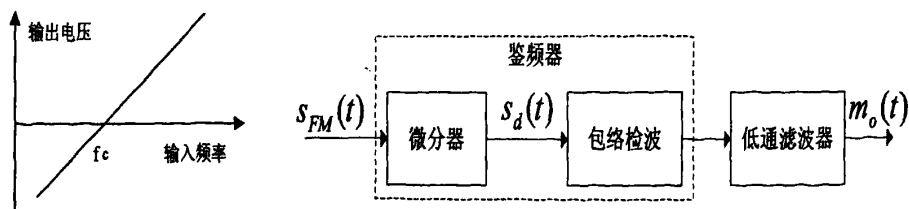


图 2.12 调频信号的非相干解调原理图

2.4.4 调频信号的正交调制解调原理

正交调制是间接法调频^[12]的一种实现方式。原理可以由式(2-21)展开并推得：

$$\begin{aligned}
 S(t) &= A \cos(2\pi f_c t) \cos\left(2\pi K_f \int_0^t m(t) dt\right) - A \sin(2\pi f_c t) \sin\left(2\pi K_f \int_0^t m(t) dt\right) \\
 &= A \cos(2\pi f_c t) \cos \Phi - A \sin(2\pi f_c t) \sin \Phi \\
 &= A \cos(2\pi f_c t) I(t) - A \sin(2\pi f_c t) Q(t)
 \end{aligned} \tag{2-27}$$

其中 $\Phi = 2\pi K_f \int_0^t m(t) dt$ 。可见，在实现 FM 时要先对调制信号进行积分，然后

对积分后的信号分别取余弦和正弦即可得到 $I(t)$ 与 $Q(t)$ 。下图为正交调制实现的一种框图^[11]。

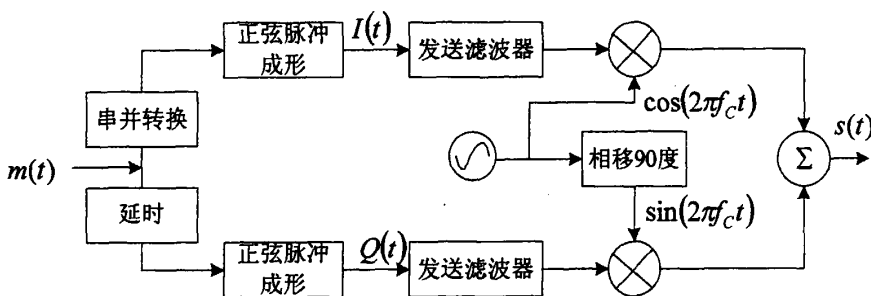


图 2.13 调频信号的正交调制实现框图

频率调制的正交解调是一种相干解调，将式(2-27)所示的调频信号 $S(t)$ 与本地产生的正交载波相乘可得：

$$\begin{aligned}
 S_I(t) &= S(t) \times \cos(2\pi f_c t) \\
 &= A \cos\left(2\pi f_c t + 2\pi K_f \int_0^t m(t) dt\right) \times \cos(2\pi f_c t) \\
 &= \frac{A}{2} \left[\cos\left(4\pi f_c t + 2\pi K_f \int_0^t m(t) dt\right) + \cos\left(2\pi K_f \int_0^t m(t) dt\right) \right]
 \end{aligned} \tag{2-28}$$

$$\begin{aligned}
 S_Q(t) &= S(t) \times \sin(2\pi f_c t) \\
 &= A \cos\left(2\pi f_c t + 2\pi K_f \int_0^t m(t) dt\right) \times \sin(2\pi f_c t) \\
 &= \frac{A}{2} \left[\sin\left(4\pi f_c t + 2\pi K_f \int_0^t m(t) dt\right) - \sin\left(2\pi K_f \int_0^t m(t) dt\right) \right]
 \end{aligned} \tag{2-29}$$

对 $S_I(t)$ 和 $S_Q(t)$ 分别进行低通滤波后可得：

$$I(t) = \frac{A}{2} \cos \left(2\pi K_f \int_0^t m(t) dt \right) \quad \text{式 (2-30)}$$

$$Q(t) = \frac{A}{2} \sin \left(2\pi K_f \int_0^t m(t) dt \right) \quad \text{式 (2-31)}$$

对 $I(t)$ 与 $Q(t)$ 分别求导数可得:

$$I'(t) = -A\pi K_f m(t) \sin \left(2\pi K_f \int_0^t m(t) dt \right) \quad \text{式 (2-32)}$$

$$Q'(t) = A\pi K_f m(t) \cos \left(2\pi K_f \int_0^t m(t) dt \right) \quad \text{式 (2-33)}$$

$$Q(t)I'(t) = -A\pi K_f m(t) \sin \left(2\pi K_f \int_0^t m(t) dt \right) \cdot \frac{A}{2} \sin \left(2\pi K_f \int_0^t m(t) dt \right) \quad \text{式 (2-34)}$$

$$Q'(t)I(t) = A\pi K_f m(t) \cos \left(2\pi K_f \int_0^t m(t) dt \right) \cdot \frac{A}{2} \cos \left(2\pi K_f \int_0^t m(t) dt \right) \quad \text{式 (2-35)}$$

$$m(t) = \frac{1}{2\pi K_f} \cdot \frac{Q'(t)I(t) - Q(t)I'(t)}{I^2(t) + Q^2(t)} \quad \text{式 (2-36)}$$

下图 2.14 为正交解调的一种实现框图。

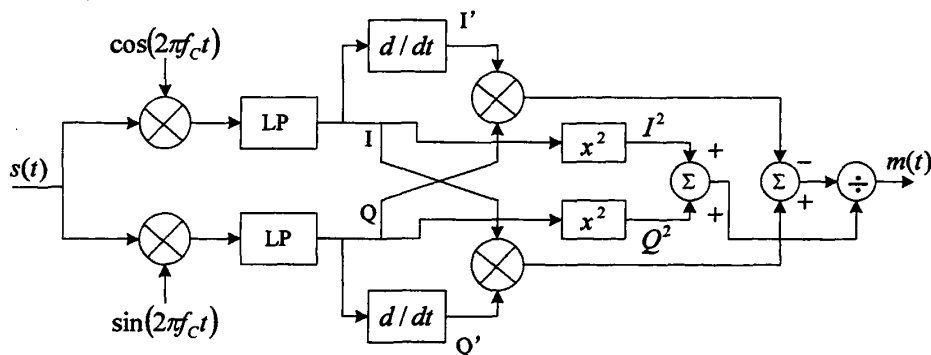


图 2.14 调频信号的正交解调原理框图

第三章 DMR 基带传输实现算法研究及仿真

本章在第二章介绍 DMR 基带处理相关协议规定和相关理论的基础上,给出基带传输算法实现方案,以及相关仿真研究。对发送端,主要包括基带成形滤波器的设计及实现、4FSK 四电平基带信号产生的卷积法与波形存储法 CCS 仿真。在接收端,针对 DMR 端机实际接收的数据,给出 DMR 基带接收算法实现方案。最后,将 DMR 基带产生信号作为输入,使用 4FSK 数字正交调制解调方式,给出 MATLAB 仿真研究。

3.1 仿真平台及运算基础介绍

在 CCS 下,配置 TMS320C5510 的 Device Simulator,本文很多算法和仿真是在此环境中实现的。有关 TMS320VC5510 的详细介绍在第四章中给出。本节首先介绍 CCS 仿真平台,然后对 DSP 定点运算的基本理论进行阐述。

3.1.1 CCS 集成开发环境简介

集成开发环境(Code Composer Studio,CCS)是 TI 公司针对 TMS320 系列 DSP 的集成开发环境,在 Windows 操作系统下,采用图形接口界面,提供有环境配置、源文件编辑、程序调试、跟踪和分析等工具。CCS 具有可扩展的结构,其主要组件包括^[12]:

1)集成开发环境 Code Composer(编辑器、调试器、项目管理器、性能分析工具等);

2)代码生成工具。包括 C 编译器、汇编优化器、连接器等;

3)指令及软件仿真器 Simulator;

4)实时底层软件 DSP/BIOS,可以增强对代码的实时分析能力;

5)实时数据交换工具 RTDX,能在不中断目标系统运行的情况下,实现 DSP 与其它应用系统之间的数据交换;

6)实时分析和数据可视化工具,如:数据的图形显示工具,可以绘制时/频域波形、星座图等;

7)GEL(General Extension Language)工具。

CCS 有两种工作模式,即:

软件仿真器模式:可以脱离 DSP 芯片,在 PC 机上模拟 DSP 的指令集和工作机制,主要用于前期算法实现和调试。

硬件在线编程模式:可以实时运行在 DSP 芯片上,与硬件开发板结合在线编

程和调试应用程序。

CCS 的功能十分强大,它集成了代码的编辑、编译、链接和调试等诸多功能,而且支持 C/C++和汇编的混合编程。用户可通其集成可视化代码编辑界面直接编写 C、汇编、.cmd 等文件。其基本调试工具具有装入执行代码、查看寄存器、存储器、反汇编、变量窗口等功能,并支持 C 源代码级调试。

对于 CCS 的使用^[14],先应建立工程文件,根据需要,添加源程序(C 或汇编)、目标文件、库文件、连接命令文件和包含文件。在 CCS 中集成编辑环境中,编辑头文件(.h 文件)、链接配置文件(.cmd 文件)、和源程序(.c、.asm 文件)等各类文件,完成整个工程的建立。然后对源文件编辑、修改后进行汇编、连接以生成目标文件(.out 文件),对源文件每作一次修改,都必须全部重新建立,生成新的目标文件,接下来,就可用 CCS 调试,先让系统复位,再载入目标文件,调试的方式多种多样,有单步执行、多步执行、断点设置和探测点设置等,用户可以对计算结果/输出数据进行分析,评估算法性能。

3.1.2 DSP 定点算术实现方法

在定点 DSP 芯片中,采用定点数进行数值运算。TMS320VC5510 为 16 位定点 DSP,在一个 16 位表示的有符号数中,最左边的一位是符号位,其余 15 位包括整数位和小数位。用 m 个比特表示整数, n 个比特表示小数($m+n=15$),可记为 Q_n 。不同 Q 值所对应的最大值、最小值和量化精度分别为 $(2^{15}-1)\times 2^{-Q}$ 、 $-(2^{15-Q})$ 、 2^{-Q} 。因此,对定点数而言,数值范围与精度是一对矛盾,一个变量要想表示比较大的数值范围,必须以牺牲精度为代价;而想精度提高,则数的表示范围就相应地减小。采用何种 Q 格式来表示小数就是定标的过程。最常用的是 $Q15$ 的形式,表示 $[-1,1)$ 的数。浮点数与定点数的转换关系可表示为:

浮点数(x)转换为定点数(x_q):

$$x_q = (\text{int})x * 2^Q \quad \text{式(3-1)}$$

定点数(x_q)转换为浮点数(x):

$$x = (\text{float})x_q * 2^{-Q} \quad \text{式(3-2)}$$

将浮点加法/减法转化为定点加法/减法时最重要的一点就是必须保证两个操作数的定标值一样。若两者不一样,需使 Q 值小的数调整为与另一个数的 Q 值一样大。此外,在做加法/减法运算时,必须注意结果可能会超过 16 位表示。如果加法/减法的结果超出 16 位的表示范围,则必须保留 32 位结果。以下为结果不超出 16 位的加法、乘法和除法的定标及 C 语言表达^[12]。

设 x 的 Q 值为 Q_x , y 的 Q 值为 Q_y , 加法结果 $z=x+y$ 的定标值为 Q_z , 则有

$$z_q = [x_q + y_q \cdot 2^{(Q_x - Q_y)}] \cdot 2^{(Q_x - Q_y)} \quad \text{式(3-3)}$$

乘积 $z=xy$ 的定标值为 Q_z , 则有

$$z_q = (x_q y_q) \cdot 2^{Q_z - (Q_x + Q_y)} \quad \text{式(3-4)}$$

商 $z=x/y$ 的定标值为 Q_z , 则有

$$z_q = \frac{x_q \cdot 2^{(Q_z - Q_x + Q_y)}}{y_q} \quad \text{式(3-5)}$$

用 C 语言描述三种运算的表示为:

```
int x,y,z;
long temp;    /*临时变量*/
/*加法 z=x+y 的定标*/
temp=y<<(Qx-Qy);
temp=x+temp;
z=(int)(temp>>(Qx-Qz)), 若 Qx>Qz
z=(int)(temp<<(Qz-Qx)), 若 Qx<Qz
/*乘法 z=xy 的定标*/
temp = (long)x;
z = (temp*y) >> (Qx+Qy-Qz);
/*除法 z=x/y 的定标*/
temp = (long)x;
z = (temp<<(Qz-Qx+Qy))/y;
```

3.2 DMR 端机发送基带信号处理

DMR协议采用4FSK调制, 是一种恒包络调制。调制时每秒发送4800个符号, 其中每个符号携带两比特的信息。最大频偏D定义如下:

$$D = 3h/2T \quad \text{式(3-6)}$$

h 代表每个特殊调制的频偏系数, 定义为0.27; T 代表符号周期, 为(1/4800)s。根据式(3-6)可计算出最大频偏D为1.944kHz。

表3.1列出了特定符号与频偏的对应关系^[3]。

表3.1 符号对4FSK频偏的映射

信息比特		符号	4FSK频偏
Bit1	Bit0		
0	1	+3	+1.944KHz
0	0	+1	+0.648KHz
1	0	-1	-0.648KHz
1	1	-3	-1.944KHz

DMR 端机的发送端主要完成 4FSK 调制, 图 3.1 给出 4FSK 调制器组成。在 DMR 基带发送系统中, 主要完成 4FSK 的四电平基带调制信号产生, 图 3.1 中虚线所示的频率调制器是在射频模块中实现。

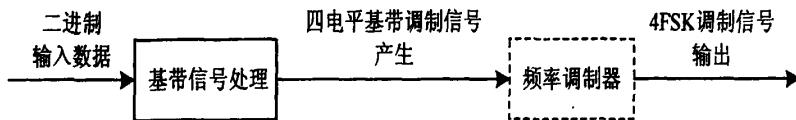


图 3.1 4FSK 调制器组成

3.2.1 4FSK 基带调制信号产生过程

DMR 传输系统分成两部分, 第一部分为基带传输系统, 产生四电平的基带带限信号作为调制信号, 在传输距离不太远的情况下, 数字基带信号可以直接传输; 第二部分是频带传输, 在经过无线信道时, 数字基带信号必须经过载波调制, 把频谱搬移到高频载波处才能在信道中传输。由于本项目采用的是直接调频法, 在接收端则采用鉴频器进行解调。因此在 DMR 端机发送端的基带传输系统中, 只需要产生带限的四电平基带调制信号。图 3.2 给出发送端的基带信号处理过程。

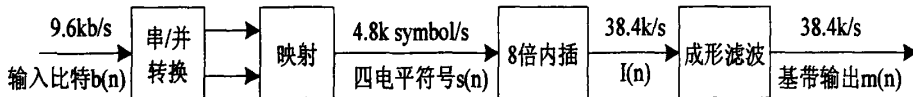


图 3.2 4FSK 基带调制信号产生过程

根据 DMR 协议, 调制时输入速率为每秒 9600 个比特, 即每秒发送 4800 个符号, 每个符号携带两比特的信息。每个符号采样 8 个值, 数据率变为 38.4k/s。由图 3.2 知, 输入的二进制比特流根据表 3.1 进行映射后, 得到四电平的符号流。由第二章内插理论知, 在每两个符号间插入 7 个零值, 实现符号的 8 倍采样。最后经过低通滤波器脉冲成形, 得到带限的基带调制输出信号 $m(n)$ 。该部分的关键在于成形滤波器的设计。

3.2.2 基带成形滤波器设计

DMR 采用滚降系数 $\alpha = 0.2$ 的根升余弦滤波器作为基带成形滤波器, 其幅度频率响应为

$$|F(f)| = \begin{cases} 1 & |f| \leq 1920 \text{ Hz} \\ \cos\left(\frac{\pi f}{1920}\right) & 1920 \text{ Hz} \leq |f| \leq 2880 \text{ Hz} \\ 0 & |f| \geq 2880 \text{ Hz} \end{cases} \quad \text{式(3-7)}$$

由 2.3 节知, 平方根升余弦滚降滤波器的时域表达式 $h(t)$ 是无穷阶数的且系数对称。DMR 基带成形滤波器设计的关键在于, 用阶数有限的 FIR 滤波器去逼近无穷阶数滤波器, 得到适合系统处理实现, 适合实际信道传输的滤波器。由于 FIR

滤波器具有稳定、线性相位特性、只包含实数算法,不涉及复数运算等突出优点,并且窗函数法设计简便,因此使用的脉冲成形数字滤波器设计选用了窗函数设计 FIR 滤波器。

根据给定的滤波器技术指标,选择滤波器阶数 N 和窗函数 $w(n)$ 。本节利用 MATLAB 仿真实现所要设计的滤波器^[15],然后通过分析滤波结果来判断设计是否满足技术指标。

用 Matlab 软件进行根升余弦滤波器的设计非常方便,调用 `firrcos` 函数即可得到滤波器系数^[16]:

$$h = \text{firrcos}(N, F_c, \alpha, F_s, \text{'rolloff'}, \text{'sqrt'}, \text{delay}, \text{window}) \quad \text{式(3-8)}$$

其中 N 是滤波器阶数,滤波器长度为 $N+1$; F_c 是理想低通滤波器的截止频率,为符号率的 $1/2$; α 是滚降系数; F_s 是采样率, `window`为截短滤波器系数所用的窗函数。返回的系数 h 被归一化了,这样通带增益为 1 。根据 DMR 系统的要求,设定截止频率 $F_c = 4800/2 = 2400$ Hz, 滚降系数 $\alpha = 0.2$, 采样率 $F_s = 4800 * 8$, N 的取值可以根据实际情况作出更改。图 3.3、图 3.4 为采用矩形窗截断得到的 16 阶、32 阶和 64 阶滤波器的幅频响应及对应的冲激响应。

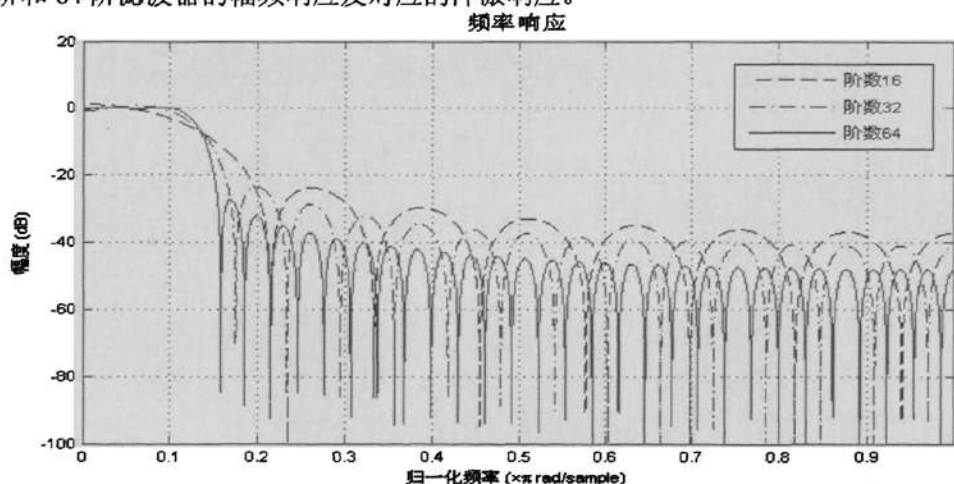


图3.3 16阶、32阶和64阶滤波器的幅频响应

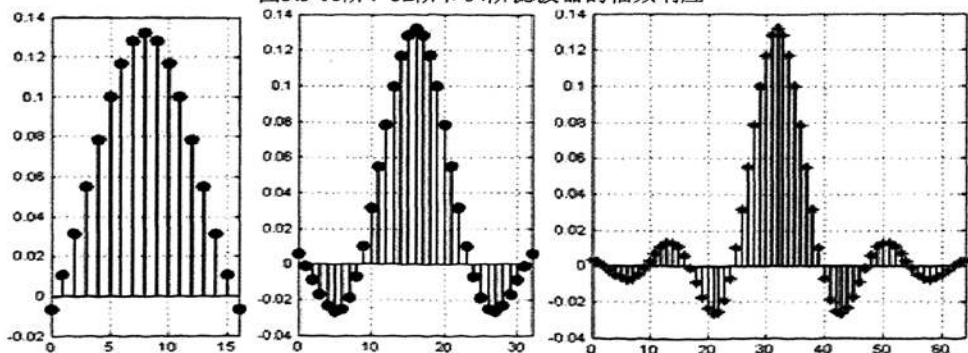


图3.4 16阶、32阶和64阶滤波器的冲激响应

仿真结果可见, N 值越大, 过渡带越陡峭, 通带截止频率越低, 阻带衰减也越大。

在 MATLAB 下, 采用 DMR 规定的一帧 132 个四电平符号信号作为输入, 8 倍内插完成后, 与设计不同阶数成形滤波器按 3.2.3 节所述方法进行卷积滤波, 比较眼图效果。

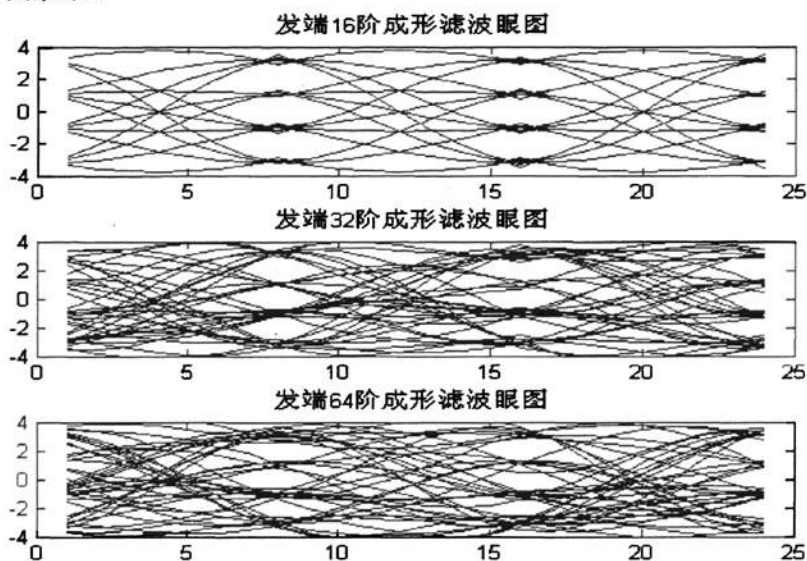


图 3.5 不同阶数滤波器成形滤波眼图

仿真发现, 滤波器阶数越大, 眼图效果越差, 引入的码间串扰越大。此外, 阶数越小, 实现滤波所用的乘法运算也越少, 因此最终选择 $N=16$ 的发端成形滤波器。

本文中比较四种窗: 矩形窗, 汉明窗, 汉宁窗, 布莱克曼窗的发端成形滤波效果。在 Matlab 下四种窗的调用形式为:

矩形窗(Rectangle Window), 调用格式: `window=boxcar(N)`

汉明窗(Hamming Window), 调用格式: `window=hamming(N)`

汉宁窗(Hanning Window), 调用格式: `window=hanning(N)`

布莱克曼窗(Blackman Window), 调用格式: `window=blackman(N)`

下图 3.6 为阶数 $N=16$ 的四种窗函数设计滤波器的频率响应。由仿真结果知, 矩形窗的过渡带宽最小, 布莱克曼窗的过渡带宽最大。同时, 布莱克曼窗的旁瓣衰减幅度最大, 矩形窗的旁瓣衰减幅度最小。

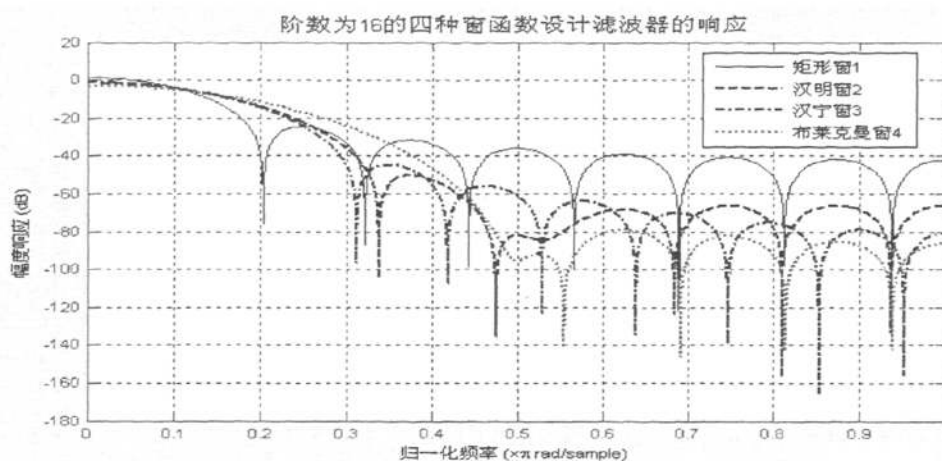


图 3.6 不同窗函数设计滤波器的幅频响应

在MATLAB环境下, 分析采用不同窗函数设计的16阶FIR滤波器对DMR系统性能影响。对图3.13中的同一帧数据用四种滤波器成形滤波, 得到的眼图如下:

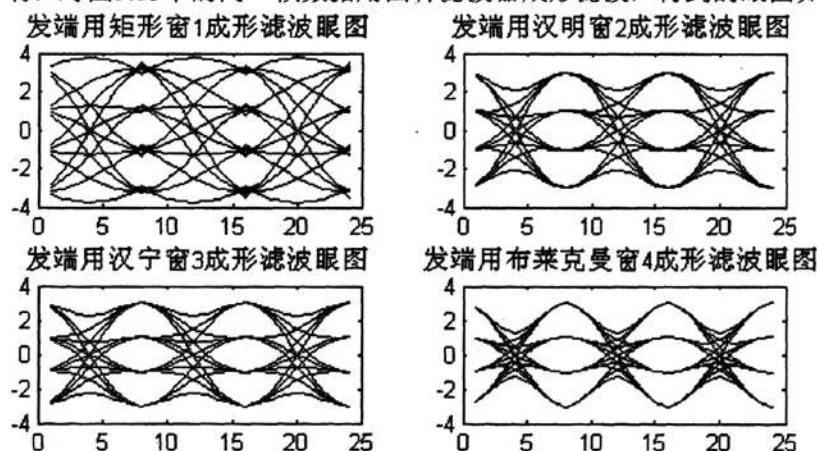


图 3.7 窗函数设计滤波器成形滤波眼图

滤波后的波形分别通过信噪比为 20dB 的 AWGN 信道, 然后再通过与发端相同的成形滤波器匹配滤波, 得到的眼图如下:

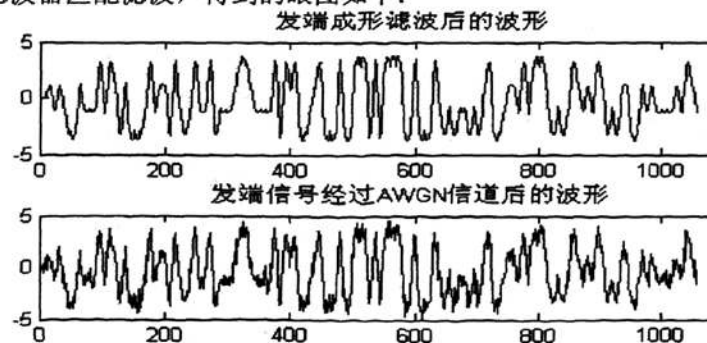


图 3.8 发端成形滤波信号经过 AWGN 信道

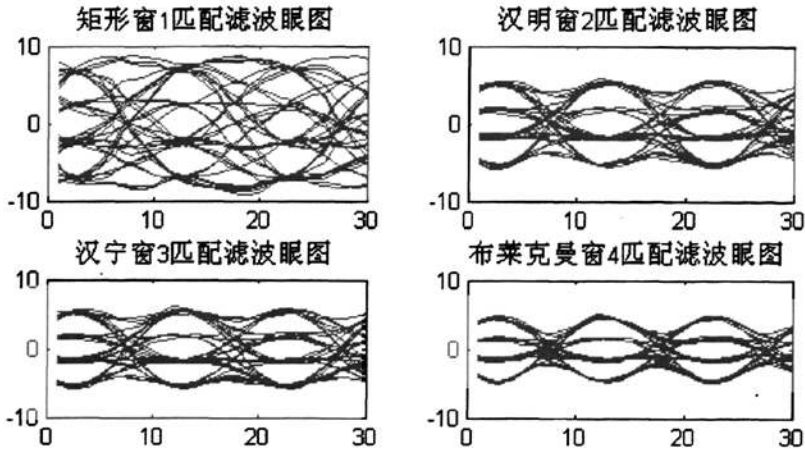


图 3.9 发端信号经过不同滤波器匹配滤波后的眼图

仿真结果分析:

- (1) 发端采用阶数 $N=16$ 的四种窗函数设计的成形滤波器滤波, 眼图效果都很好。
- (2) 在 MATLAB 下, 发送信号通过相同信噪比的高斯白噪声信道, 接收端采用与发端相同的滤波器匹配滤波后, 从眼图效果来看, 汉明窗、汉宁窗和布莱克曼窗稍优于矩形窗。
- (3) 理论上, 应该选择过渡带陡峭, 且阻带衰减大的滤波器, 而在实际实现中, 这二者是一对矛盾。由于矩形窗具有最窄的主瓣宽度, 且成形滤波效果与其它滤波器相差不大, 因此最终选择阶数 $N=16$ 的矩形窗截断的滤波器作为发端成形滤波器。

3.2.3 DMR 端机发送基带信号处理的 DSP 实现

设 FIR 滤波器单位脉冲响应长度为 N , 其系统传输函数 $H(z)$ 为

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \quad \text{式(3-9)}$$

$H(z)$ 是 z^{-1} 的 $(N-1)$ 次多项式, 它在 z 平面上有 $(N-1)$ 个零点, 原点 $z=0$ 是 $(N-1)$ 阶重极点。因此, $H(z)$ 永远稳定。在时域, 输入为 $x(n)$, FIR 滤波器的输出 $y(n)$ 是滤波器冲激响应 h 与 $x(n)$ 的卷积。它的基本算法是一种相乘累加运算, 即不断地输入样本 $x(n)$, 经过 z^{-1} 延时后, 再进行乘法-累加, 最后输出滤波结果 $y(n)$ 。

$$y(n) = h_0x(n) + h_1x(n-1) + \dots + h_{N-1}x(n-N+1) \quad \text{式(3-10)}$$

其中, h_i 为滤波器系数, $x(n)$ 表示滤波器在 n 时刻的输入, $y(n)$ 为 n 时刻的输出。由于本文设计的成形滤波器冲激响应长度 $N=17$, 具有偶对称特性:

$$h(n) = h(N-1-n) \quad \text{式(3-11)}$$

将式(3-11)代入式(3-10), 得到:

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \quad \text{式(3-12)}$$

$$= \sum_{i=0}^{\frac{N-3}{2}} h(i)[x(n-i) + x(n+i-N+1)] + h(\frac{N-1}{2})x(n-\frac{N-1}{2})$$

由式(3-12)可以看出, 采用对称结构的滤波器, 可减少近一半的乘法运算, 而乘法在 DSP 中是非常耗费资源的运算, 因此可减少 DSP 的运行时间。

实现条件:

在 CCS 下, 设置硬件仿真环境 C5510 v2.x XDS510 Emulator, 将工程编译链接完成后, 通过 DSP 仿真器下载到 DSP 芯片。3.2.2 节中在 MATLAB 下仿真产生的滤波器系数都是归一化的小数, 在 CCS 中则采用 Q15 定点数表示, 即将每个系数乘以 32768 (2^{15}), 然后四舍五入取整进行存储。采用 DMR 规定的一帧 264 个 0、1 比特作为输入。按表 3.1 所示的规则, 对输入信号完成串并转换, 得到 132 位的四电平信号, 大小为 ± 1 、 ± 3 。对四电平信号每两个值之间插入 7 个零, 完成 8 倍内插。内插后的信号采用 DSP 的 Q13 定点数表示, 再与存储的 FIR 滤波器系数按式(3-12)所示的方法进行卷积运算, 得到基带四电平调制信号的输出波形。基于图 3.2 给出的发端模型, 以下为在具体实现过程中 CCS 下观察的 $b(n)$ 、 $s(n)$ 、 $I(n)$ 、 $m(n)$ 各处的波形。

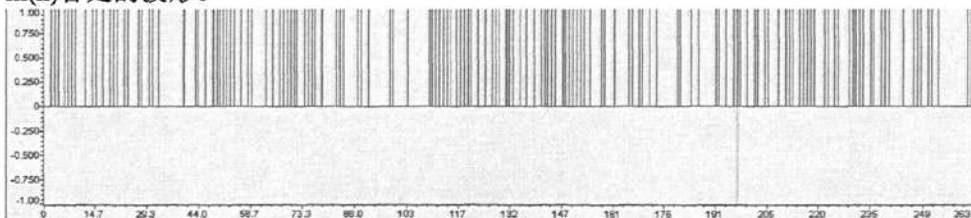


图 3.10 264 位二进制比特输入

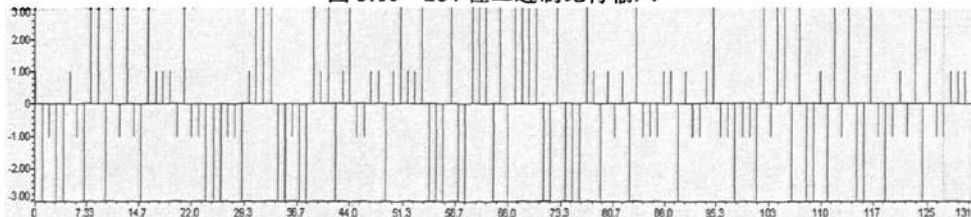


图 3.11 完成映射后产生的四电平信号

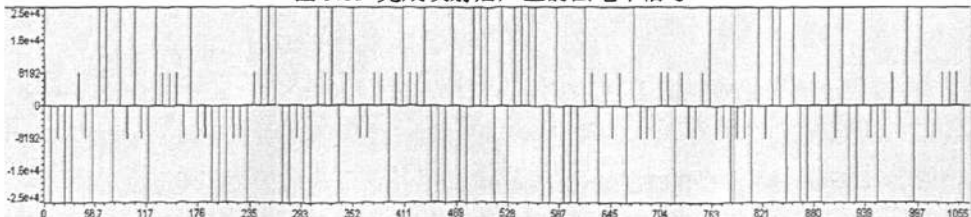


图 3.12 8 倍内插后的四电平信号

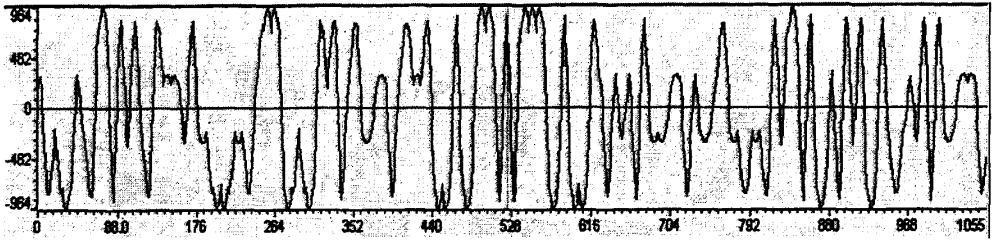


图 3.13 卷积法成形滤波后的基带输出波形

3.3 DMR 基带信号的波形存储法产生

实现4FSK调制基带信号产生的方法有两种：第一种是卷积滤波法，第二种是波形存储法。卷积滤波法已在3.2.3节中作过介绍。本节介绍波形存储法的原理^[17]，然后在CCS环境下做仿真研究与分析。

3.3.1 波形存储法原理

假设 $\{a_n\}$ 为输入符号序列，DMR 下为四电平，即取值为+1,-1,+3,-3，则对应的基带信号 $d(t)$ 为间隔为 T_s ，强度由 a_n 决定的单位冲击序列组成：

$$d(t) = \sum_{n=-\infty}^{\infty} a_n \delta(t - nT_s) \quad \text{式(3-13)}$$

a_n 取值为+1,-1,+3,-3。信号经过发送滤波器输出为：

$$s(t) = d(t) * g_T(t) = \sum_{n=-\infty}^{\infty} a_n g_T(t - nT_s) \quad \text{式(3-14)}$$

$g_T(t)$ 为单个冲击脉冲 δ 经过发送滤波器作用下形成的发送基本波形。如果考虑左右相邻两码元的影响，则第 k 个码元间隔内滤波器响应波形是由前后共5个码元信息的响应组合而成^[18]，此时：

$$s(t + kT_s) = \sum_{n=-\infty}^{\infty} a_n g_T[t - (n - k)T_s] \approx \sum_{n=k-2}^{k+2} a_n g_T[t - (n - k)T_s] \quad \text{式(3-15)}$$

DMR 规定每个符号抽样8个样值，当前码元输出受五个码元的影响，因此要设计的基带成形滤波器阶数为40。要存储的波形为四种，分别为符号+1,-1,+3,-3对应的脉冲经过40阶的根升余弦滤波器的输出波形。如图3.14示，为符号1通过滤波器的输出波形f1，存储五段波形值f1_1、f1_2、f1_3、f1_4、f1_5，每段包含8个样点值。依照此种方法，可分别存储符号-1,+3,-3通过滤波器的输出波形值。

图3.15为五个连续码元1时当前输出波形结果(虚线内所示)。由图可以看出，当前码元实际输出波形为前二个码元第五段波形、前一个码元第四段波形，当前码元第三段波形、后一个码元第二段波形和后两个码元第一段波形叠加而成。

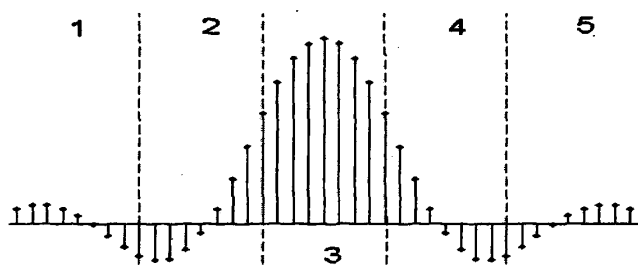


图 3.14 符号 1 需存储的五段波形

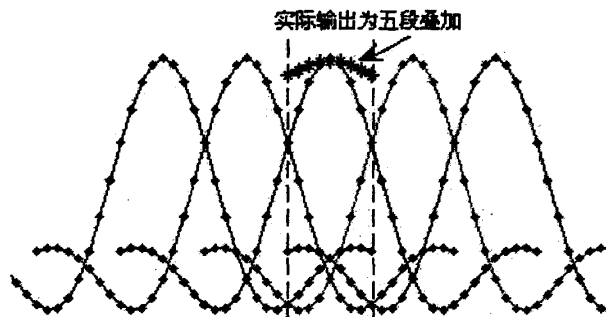


图 3.15 波形存储法原理

波形存储法等效于图 3.2 中的内插和滤波运算, 是以空间复杂度换取时间复杂度的数据处理方式, 因为没有乘法运算, 在实际 DSP 系统中可以大大节省运算时间, 有利于信号的实时处理。波形存储法在实际处理中有两种实现方式: 波形累加求和法和查表法。

波形累加求和法原理是, 分别存储四个码元的输出波形的五段。根据当前第 k 个码元与第 $k-2$ 个、第 $k-1$ 个、第 $k+1$ 个、第 $k+2$ 个码元, 查找对应段叠加, 得到 k 码元的输出结果。需要注意的是输入码元的最前面两个和最后两个, 第一个输入码元仅受后两个码元影响, 第二个输入码元仅受前一个和后两个的影响。同理可知最后两个码元的输出影响。

查表法将每个码元对应的所有可能输出值全部存储, 根据输入的五個码元作为索引生成地址值, 查表直接输出当前码元结果。DMR 一帧数据由 132 个符号组成, 考虑前后两个码元的影响, 每个码元间隔内共有 $4^5=1024$ 种可能的组合波形。此外, 第一个码元和最后一个码元各有 4^3 种可能, 第二个码元和倒数第二个各有 4^4 种可能。因此要制的表有五个。

在制表时, 首先要将码元映射。由于输入电平为四进制, 因此将码元+1、+3、-1、-3 依次映射为 0、1、2、3。将码元完成映射后, 在一个 5×1024 的矩阵里, 第一行按照 0、1、2、3 的顺序依次四个码元的第一段, 每一段重复存储 256 次。第二行存储四个码元的第二段, 每个码元重复存储 64 次之后, 以四个码元一组重复存储。第三行存储四个码元的第三段, 每个码元重复存储 16 次之后, 以四个码元一组重复存储。如此类推, 将 5×1024 的矩阵存满。具体存储结构如下图 3.16 所示。

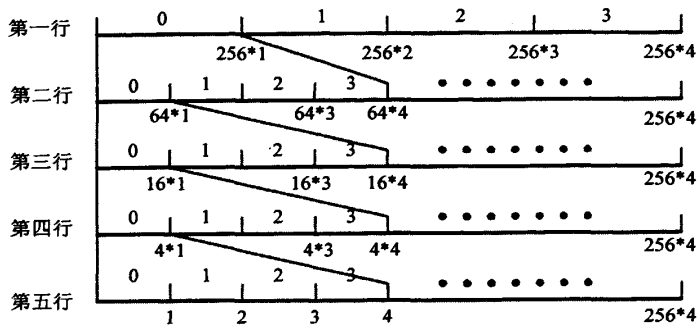


图 3.16 查表法存储结构

将上图所得的矩阵转置，将每一行对应的存储波值相加，得到存储的波形值。按照此规则可制得一张 1024×8 大小的存储表。假设输入五个码元为 abcde(已经完成映射)，则查找表的索引 key 为 $key = a \times 256 + b \times 64 + c \times 16 + d \times 4 + e$ 。即对应存储表的第 key 行输出的 8 个点为当前码元 c 的输出波形值。

按照此原理可分别制得 2 张 64×8 和 2 张 256×8 的表用于前面两个和倒数两个码元的波形存储表。对于 64×8 的表，输入的三个码元为 abc，则查找表的索引 key1 为 $key1 = a \times 16 + b \times 4 + c$ 。对于 256×8 的表，输入四个码元为 abcd，查找表的索引 key2 为 $key2 = a \times 64 + b \times 16 + c \times 4 + d$ 。

3.3.2 波形存储法的 CCS 仿真

一、仿真条件

在 CCS 的 TMS320C5510 的 Device Simulator 下，由于 DMR 四电平输入数据为 ± 3 范围内，用 Q13 定点数表示，波形存储滤波法存储的波形也采用 Q13 定点数表示。输入为 132 个 DMR 四进制序列，分别用卷积法、波形累加求和法和查表法在 Device Simulator 下建立工程，对同一帧数据进行滤波处理，再设定断点用 view 下提供的 Graph 工具观察仿真的结果。

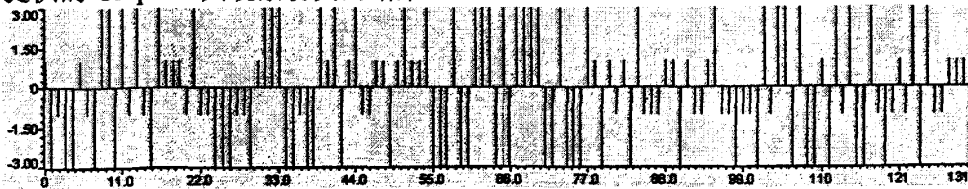


图 3.17 输入四进制序列

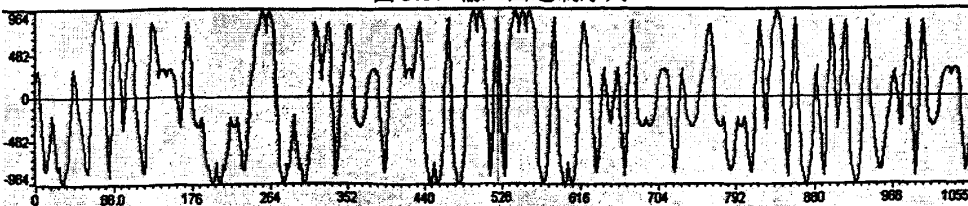


图 3.18 卷积滤波(16 阶)

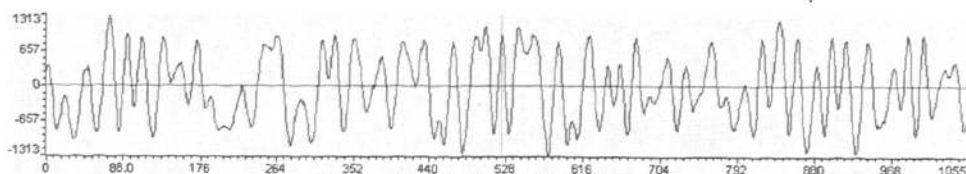


图 3.19 波形累加求和法滤波(40 阶)

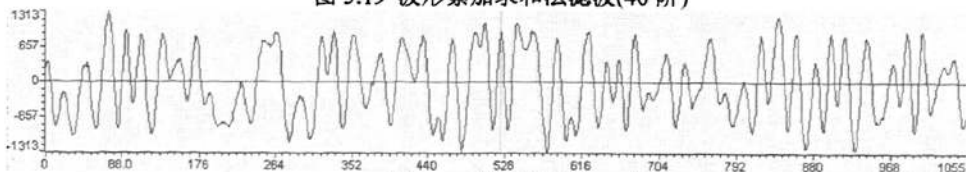


图 3.20 查表法滤波(40 阶)

将 CCS 下得到的发送端的发送数据，导入到 matlab 下，编写程序画出发端眼图比较。

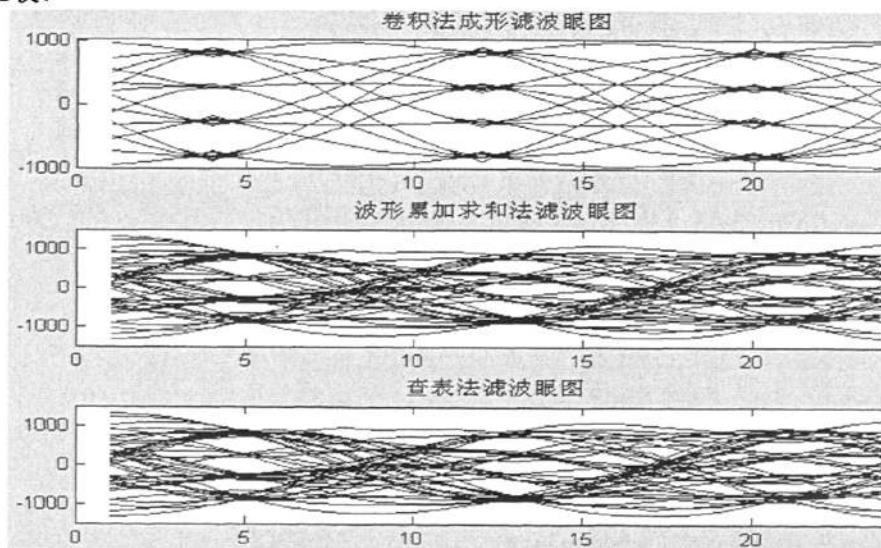


图 3.21 三种方法滤波眼图

二、仿真结果分析

由图 3.21 可见三种方式都能对序列进行正确的滤波。波形累加求和法和查表法是波形存储法滤波的两种不同实现形式，因此滤波效果完全一样。仿真发现，从眼图效果来看，通过 16 阶的成形滤波器卷积滤波的效果优于波形存储法。

通过 CCS 集成环境的剖析工具（profile 工具）发现，卷积滤波方式与波形存储滤波方式相比极耗时间。卷积滤波程序中滤波器阶数为 16，即滤波器长度为 17，完成滤波用了 341968 个指令周期，而用第二种方式完成滤波用了 92131 个指令周期，第三种方式完成滤波仅用了 50793 个指令周期。相比之下，查表法耗时最少，波形累加求和法次之，卷积滤波耗时最长。所以就算法的实时性来说，查表法最好。

三种方式建立工程，可以发现.map 文件中，发现卷积法所用的存储器空间为 12672 个字节，波形累加求和法为 8850 个字节，查表法为 38360 个字节。因此从

空间复杂度角度来看, 波形累加求和法是最优选择。

本系统从最小串扰角度出发, 选择发端成形滤波眼图效果最好的方法来实现四电平基带信号的产生, 因此, 在实际 DSP 实现中最终选择 16 阶的成形滤波器卷积滤波。

3.4 DMR 端机接收基带信号处理

在接收端, 基带信号经过与发送成形滤波器相同的匹配滤波器滤去噪声分量, 再经过抽样判决器恢复 DMR 四进制码元 $s(n)$ (取值为 $\pm 1, \pm 3$), 再经过反射射恢复出二进制 0、1 比特。

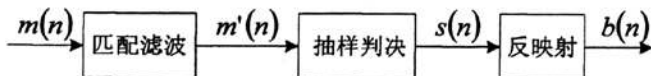


图 3.22 DMR 基带信号接收端处理

3.4.1 DMR 端机接收基带信号处理的 DSP 实现

DMR 一帧数据为 132 个四进制码元, 对每个码元采样 10 个值, 为 1320 个数据, 为保证数据完整性, 多采了 100 个样点。因此本项目得到的基带 4FSK 解调的一帧数据为 1420 个 A/D 采样数据。经过大量实验发现, 接收采用与发送端对应的 24 的根升余弦滤波器进行匹配滤波眼图效果最好。收端每个符号取 10 个样点, 由于同步码为 24 个符号, 本地存储的是 240 个数据长的同步码波形。将本地存储波形与匹配滤波后的数据帧做滑动相关运算, 得到相关峰, 然后再进行抽判、反射射等一系列操作。

下面就无线传输中接收到的一个数据帧为例, 在 CCS 环境下, 说明 DMR 端机接收端基带信号处理的 DSP 实现过程。基带传输接收端的信号处理过程与之完全相同, 不同的是, 基带传输的接收信号效果更好更利于处理。

(1) 采样接收

DSP 让 MCU 在合适的时候对发送端的数据突发进行接收。为了保证有效数据段采样的完整性, 每个时隙采样 1420 个数据。图 3.23 为 CCS 下观察到的原始采样数据图。

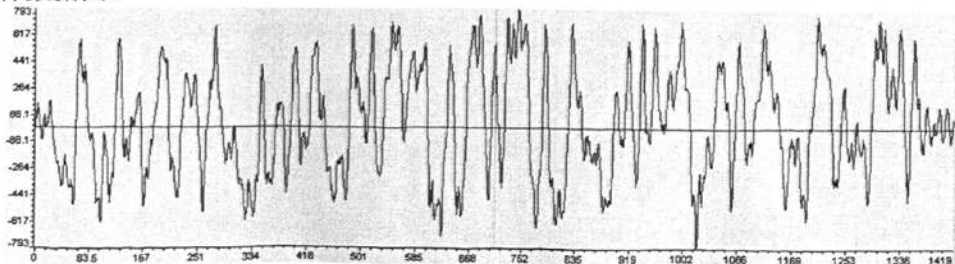


图 3.23 接收的一帧原始采样数据图

(2) 匹配滤波

将每个时隙得到的 1420 个采样数据经过 24 阶的匹配滤波器, 滤波后得到较为平滑的输出波形。

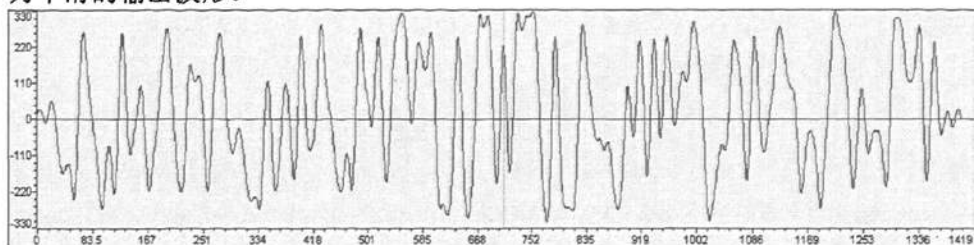


图 3.24 接收数据匹配滤波后波形

(3) 相关运算

将匹配滤波后的波形送入相关器, 进行相关运算。由于本地存储了长度为 240 的同步码波形, 将此段波形与数据进行 240 长度窗的相乘累加运算, 得到相关运算结果。图 3.25 显示为 DSP 中存储的本地同步码波形; 图 3.26 是同步码从数据帧头滑动到帧尾进行相关运算得到的结果, 能看到尖锐的单峰特性, 返回的单峰位置为同步码波形末端对应的数据帧位置。

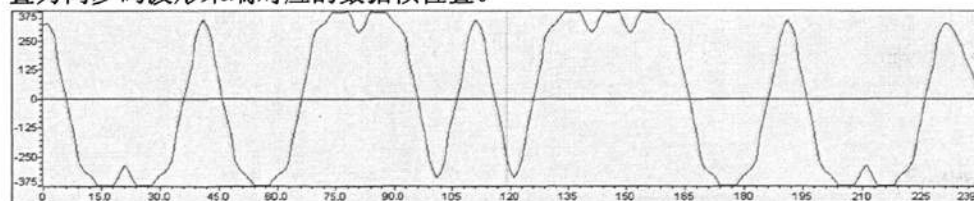


图 3.25 本地存储的同步码波形

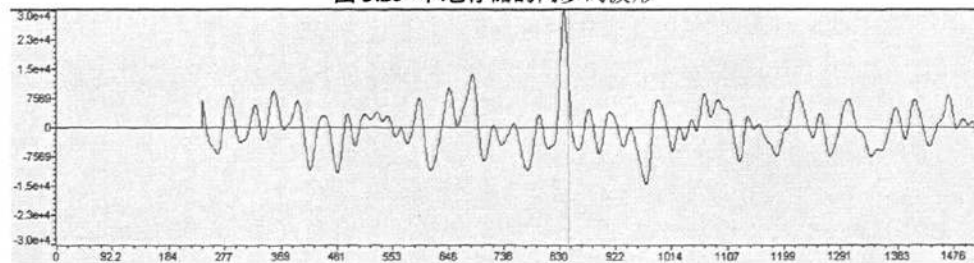


图 3.26 相关运算结果

理论上, 一帧数据应刚好在一个突发的正中央, 即数据中央 710 刚好对应了同步码的中央位置, 因此对数据帧进行相关运算时相关峰出现的理论位置为 $710+120=830$ 。在实际处理中, 由于滑动相关运算在 DSP 中非常耗费运算时间, 为减少运算量, 可将滑动搜索窗口定义在 $[800, 860]$ 之间。并且设置一门限为 15000, 当相关运算的值大于 15000 时, 寻找最大值并返回该点位置。且当累加值再次小于 15000 时结束该过程。如果从 800—860 进行滑动相关运算均没有超过 15000 的相关值, 则认为没有找到同步码。

(4) 抽样判决

将相关峰返回位置减去 $(54+24)*10$ ，即得到第一个抽判位置。以后每 10 个样点抽取一次，即可抽样得到 132 个数据点。图 3.27 为 CCS 下对图 3.25 的数据抽样得到的 132 个样点。

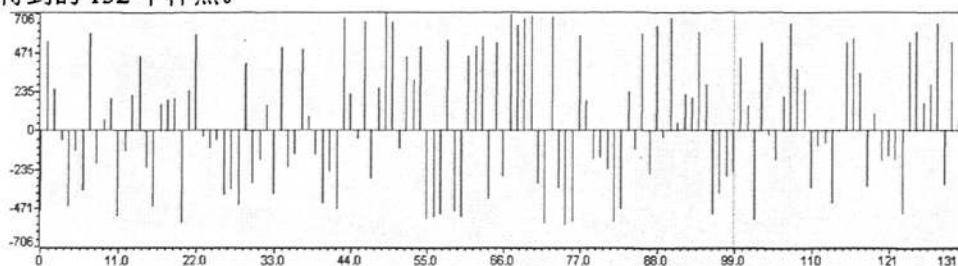


图 3.27 抽取得到的 132 个样点图

得到的 132 个抽样值根据三个门限：Threshold1、Threshold2、Threshold3 进行判决。当抽样值大于 Threshold1 时判为+3，介于 Threshold1 和 Threshold2 之间的判为+1，介于 Threshold2 和 Threshold3 之间的判为-1，小于 Threshold3 的判为-3。三个门限值可以根据输入的 1420 个数据在 MATLAB 下画眼图直观观察得到。下图 3.28 为 MATLAB 下画出基带一帧接收数据匹配滤波眼图，图 3.29 为根据眼图观察的三个门限 100、0 和-100。

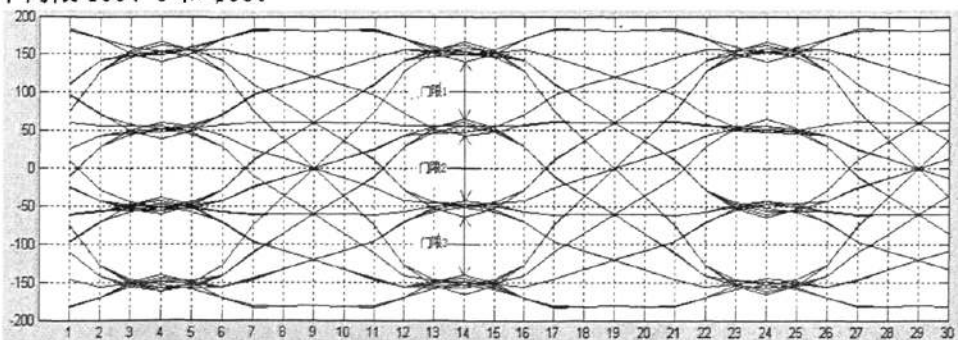


图 3.28 基带接收的一帧数据匹配滤波眼图

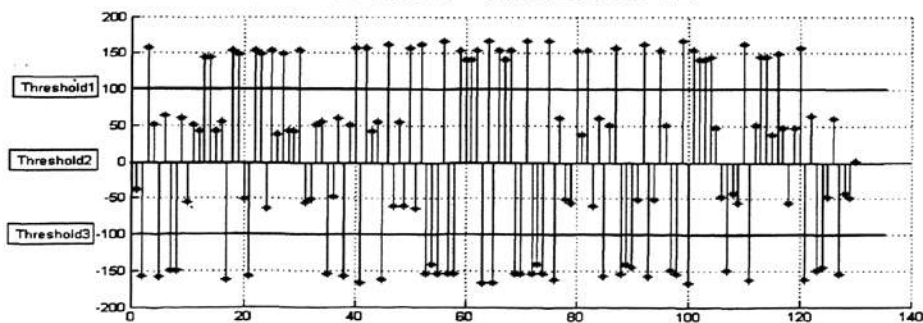


图 3.29 由眼图直观观察的判决门限

因为码元的最佳抽判位置对应于眼图张开度最大的位置，而门限则是在“眼

睛”张开最大中央所对应的纵坐标的值,如图 3.28 所示。根据此原理,笔者提出一种基于眼图的 DMR 系统 4FSK 基带解调自适应门限判别方案。对应于图 3.28,给出具体算法如下:

- 1) 解调输入的一帧 A/D 采样数据个数为 N , 由于收端每 10 个点抽判一次。以 30 个点作为横轴周期, 每个点作为一个数组, 每个数组大小 M 为 $[N/30]$ 。
- 2) 将 N 个值存入 30 个数组 $A_1, A_2 \dots A_i \dots A_{30}$ 中。第 i 个数组 A_i 对应横轴为 i 的点, 依次将 N 个采样点的第 i 个、第 $i+30$ 个、.....第 $i+(M-1)*30$ 个存入 A_i 。
- 3) 将每个横轴点 i 对应的数组 A_i 的 M 个值进行冒泡法排序^[19], 最大值放在 $A_i[M-1]$ 点, 最小值放在 $A_i[0]$ 点, 最大值减最小值为 d 。
- 4) 对于每个 A_i , 依次作差 $cha=A_i[n]-A_i[n-1]$ (n 从 $M-1$ 到 1)。若差值 cha 在纵轴 $A_i[0]+d/2$ 处仍没有大于 $d/4$, 该点 i 放弃; 若有大于 $d/4$, 计数器 $count$ (原始为 0) 加一, 且记下相应的 $A_i[n]$ 和 $A_i[n-1]$ 。
- 5) 对每个 A_i , 依次作差完成后, $count$ 若等于 3, 将记下的三组 $A_i[n]$ 和 $A_i[n-1]$ 按大小顺序依次赋值给 $pan0$ 、 $pan1$ 、 $pan2$ 、 $pan3$ 、 $pan4$ 和 $pan5$ 。若 $pan1$ 与 $pan2$ 的差值、 $pan3$ 与 $pan4$ 的差值均小于 $d/12$, 则门限 1 为 $Threshold1=(pan0+pan1)/2$, 门限 2 为 $Threshold2=(pan2+pan3)/2$, 门限 3 为 $Threshold3=(pan4+pan5)/2$, 一个码元对应的最佳抽判位置 $zuijia=[i/10]$ 。
- 6) 对每个 A_i , 依次作差完成后, $count$ 若大于 3, 将 i 加 10, 按步骤 3)、4)、5) 所述算法运算。

编写相应的程序, 将上述 CCS 下的经过匹配滤波的数据作为输入, 程序执行得到三个门限 $Threshold1$ 、 $Threshold2$ 、 $Threshold3$ 分别为 356、98 和 -260。根据这三个门限对图 3.27 中的抽取值进行判断, 得到 132 个判决值。

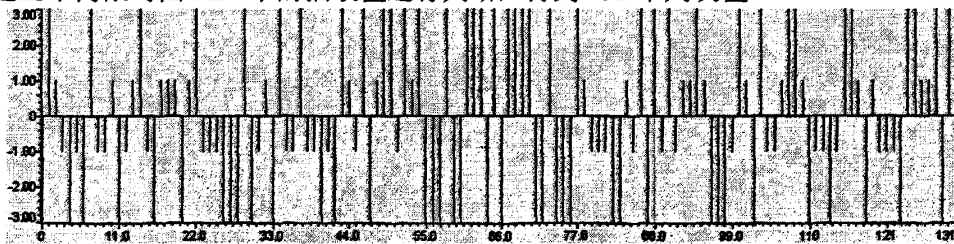


图 3.30 判决恢复的四进制电平序列

上述算法中所用的 $d/4$ 、 $d/12$ 均为经验值, 在做大量实验的基础上总结出的。算法的局限性是在眼图效果不好时不能得到正确的抽判位置和门限, 直接观察眼图法得到的抽判门限更加简便快速。对图 3.30 得到的四进制符号序列再进行按表 3.1 反映射, 即可得到二进制比特流, 如图 3.31 所示。

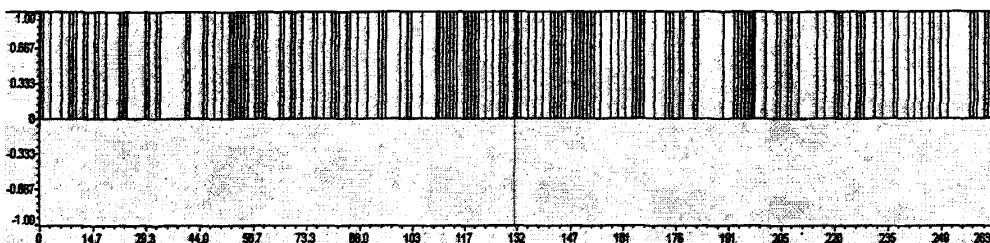


图 3.31 反映射得到的二进制比特

3.5 DMR 信号 4FSK 间接法实现的仿真

本项目采用的 4FSK 调制解调方式是直接法调频与非相干法解调。本节中, 将基带产生的 DMR 信号作为输入, 用间接法实现 4FSK 调制解调的 MATLAB 仿真研究, 所采用的间接法为数字正交合成调制与正交差分解调。

3.5.1 数字正交合成调制

对式(2-27)经离散化(用 nT_s 代替 t), 除了离散化外, 还要涉及到积分运算转化为数值计算方法, 以便于数字处理。复化积分就是数值积分的一种, 它具有精度高、运算量小、多样点求值方便等优点。复化积分的基本过程如下^[12]:

将积分空间 $[a, b]$ 划分为 n 等分, 步长为 $h = \frac{b-a}{n}$, 各分界点为 $x_k = a + kh, k = 0, 1, \dots, n$ 。先求各区间上的积分值 I_k , 然后再求和 $\sum_{k=0}^{n-1} I_k$, 作为所求积分的近似值。各区间上的积分值 I_k 可以用梯形计算公式 $I_k = \frac{T_s}{2} (f(x_k) + f(x_{k+1}))$ 求得, 因此采用复化求积后的 FM 离散数学表达式如下:

$$S(nT_s) = A \cos \left[2\pi f_c nT_s + 2\pi K_f T_s \sum_{i=1}^n \frac{m(iT_s) + m[(i-1)T_s]}{2} \right] \quad \text{式(3-16)}$$

三角展开后可以写为如下形式:

$$\begin{aligned} S(nT_s) &= A \cos(2\pi f_c nT_s) \cos \left[2\pi K_f T_s \sum_{i=1}^n m(iT_s) \right] - A \sin(2\pi f_c nT_s) \sin \left[2\pi K_f T_s \sum_{i=1}^n m(iT_s) \right] \\ &= A \cos(2\pi f_c nT_s) I(nT_s) - A \sin(2\pi f_c nT_s) Q(nT_s) \end{aligned} \quad \text{式(3-17)}$$

其中采样间隔时间 $T_s = 1/f_s$, f_s 为采样频率。将输入信号按 I、Q 两个支路计算信号后再累加, 可以得到数字化的 FM 信号, 具体实现过程如图 3.32 所示。

仿真条件:

如图 3.32 示, 在 MATLAB 下, DMR 基带四进制输入符号率为 4.8k, 每个符号周期采样 8 个值, 经过成形滤波之后的数据率为 38.4k($f_s = 38.4k$), 该信号作为

输入(A 点)。调频系数 K_f 设为 648, 信号经过累加求和和求余弦运算后, 得到 I 路基带信号(B 点), 求正弦运算后得到 Q 路基带信号(C 点)。假设载波频率 f_c 为 96kHz, 载波数据率为 $96 \times 4 = 384\text{k}$ ($f_s = 384\text{k}$), 则内插滤波器的内插因子刚好是 10, 为整数倍内插, 用五级 CIC 滤波器实现。内插滤波后的 I 路波形如 D 点所示, Q 路波形如 E 点所示, 经过累加器, 最后得到的数字化调频信号输出如 F 点所示。图 3.32 为数字正交合成调制的框图, 图 3.33 到图 3.38 为 MATLAB 下仿真的各点的观测波形, 图 3.39 为调频输出信号的功率谱, 可以看出, 信号的载波频率正是 96k。

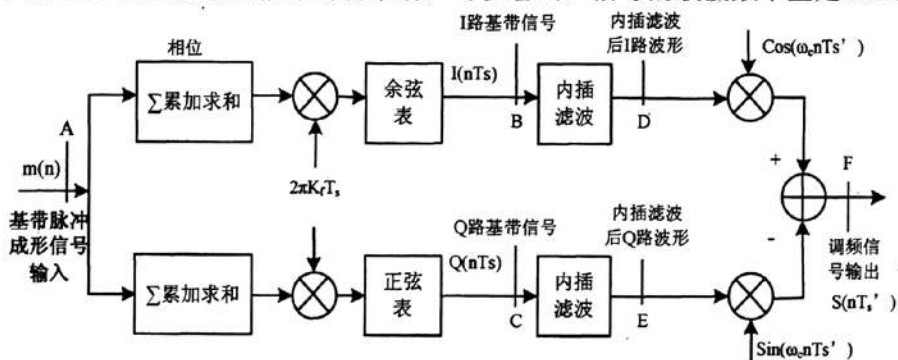


图 3.32 数字正交合成调制框图

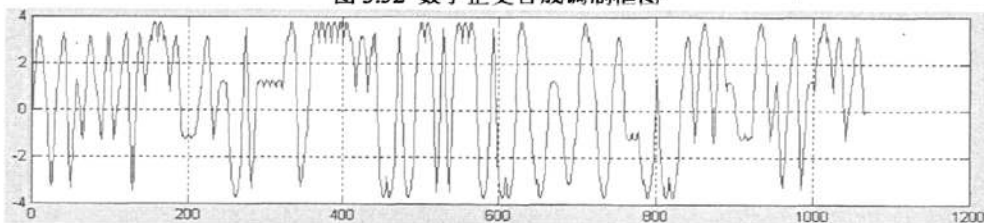


图 3.33 A 点基带输入波形

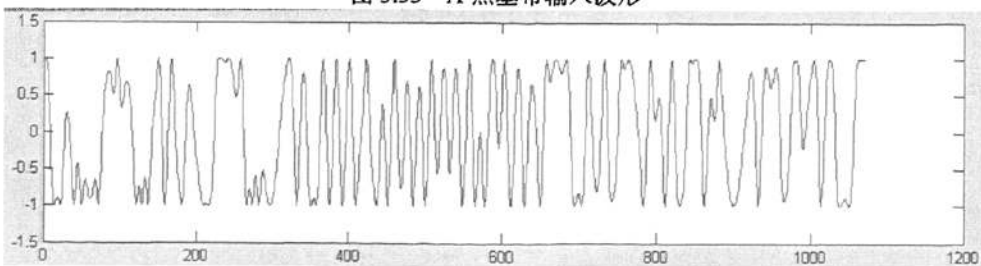


图 3.34 B 点 I 路基带波形

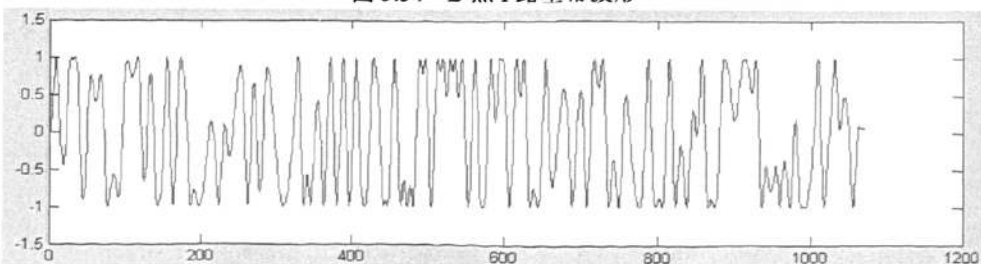


图 3.35 C 点 Q 路基带波形

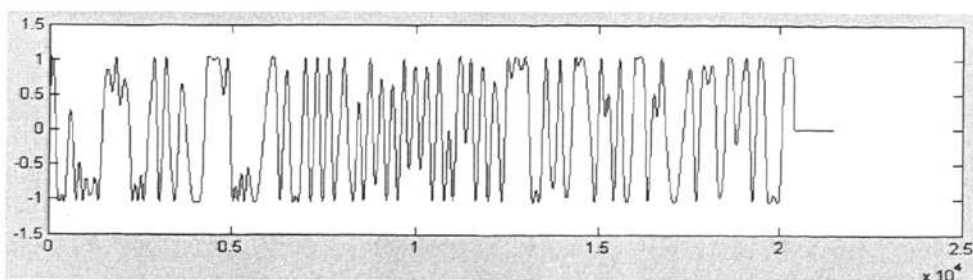


图 3.36 D 点插值滤波后的 I 路波形

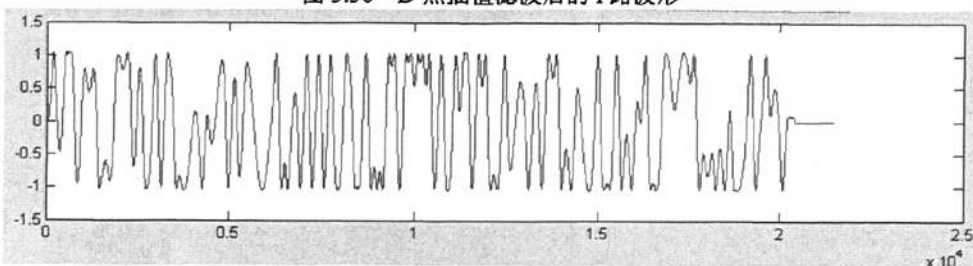


图 3.37 E 点插值滤波后的 Q 路波形

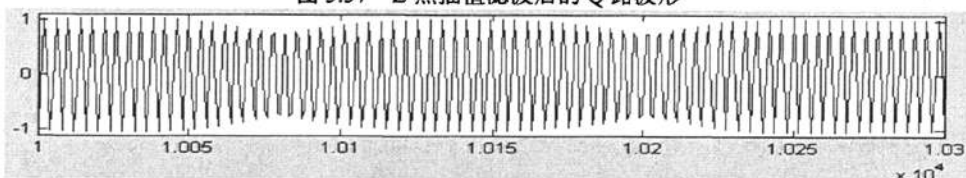


图 3.38 F 点调频信号输出波形

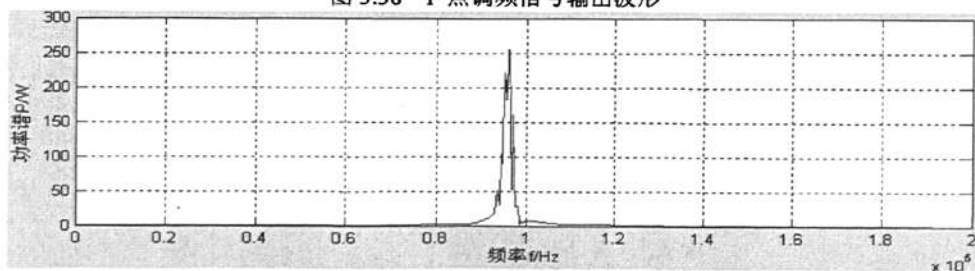


图 3.39 4FSK 调频信号功率谱

3.5.2 数字正交差分解调

在接收端, 对式(2-30)经离散化表达式为(经过低通滤波和抽取之后, 高频分量被去掉, 数据量减少):

$$I(nT_s) = \frac{A}{2} \cos \left[2\pi K_f T_s \sum_{i=1}^n m(iT_s) \right] \quad \text{式 (3-18)}$$

$$Q(nT_s) = \frac{A}{2} \sin \left[2\pi K_f T_s \sum_{i=1}^n m(iT_s) \right] \quad \text{式 (3-19)}$$

对正交与同相分量做反正切运算, 得相位:

$$\phi(n) = \arctan\left[\frac{Q(n)}{I(n)}\right] = 2\pi K_f T_s \sum_{i=1}^n m(iT_s) \quad \text{式(3-20)}$$

然后对相位进行差分运算, 得瞬时相位值:

$$\phi_d(n) = \phi(n) - \phi(n-1) = 2\pi K_f \cdot T_s \cdot m(n) \quad \text{式(3-21)}$$

将瞬时相位值再除以常数 $2\pi K_f T_s$ 即可得到调制信号 $m(n)$ 。

但是, 在实际的应用中, $\phi_d(n)$ 会出现相位跳变, 必须对其进行校正。由于相位 $\phi(n)$ 是做反正切运算的结果, 而正切函数的周期是 π , 因此具体的校正规则如下:

$$\begin{cases} \phi_d'(n) = \phi_d(n) - \pi, & \phi_d(n) > -\pi/2 \text{ 时} \\ \phi_d'(n) = \phi_d(n) + \pi, & \phi_d(n) < -\pi/2 \text{ 时} \end{cases} \quad \begin{matrix} \text{式(3-22)} \\ \text{式(3-23)} \end{matrix}$$

仿真条件:

在 MATLAB 下, 将 3.5.1 节中的调频信号作为输入 (F 点), 经过与本地的余弦载波相乘 ($f_s = 384\text{k}$, $f_c = 96\text{k}$), 再经过与发送端对应的 CIC 滤波器进行 10 倍速低通抽取滤波, 得到 I 路抽取滤波波形 (G 点)。与本地正弦载波相乘, 进行 CIC 低通抽取滤波, 得到 Q 路抽取滤波波形 (H 点)。按式 (3-21) 进行差分鉴频, 得到输出波形 (J 点), 再按式 (3-22)、式 (3-23) 将 J 点波形进行相位校正, 最后乘以常数, 得到解调输出波形 (K 点)。正交差分解调的实现框图如图 3.40 所示, 图 3.41 到图 3.44 为 MATLAB 下仿真的各点的观测波形。

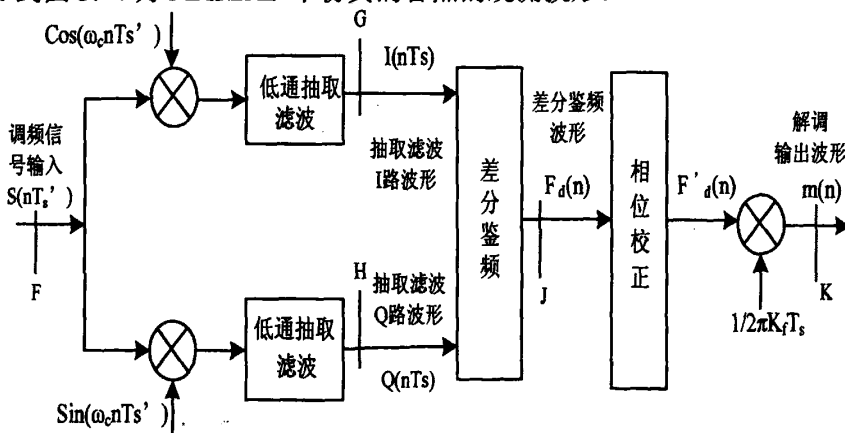


图 3.40 数字正交差分解调框图

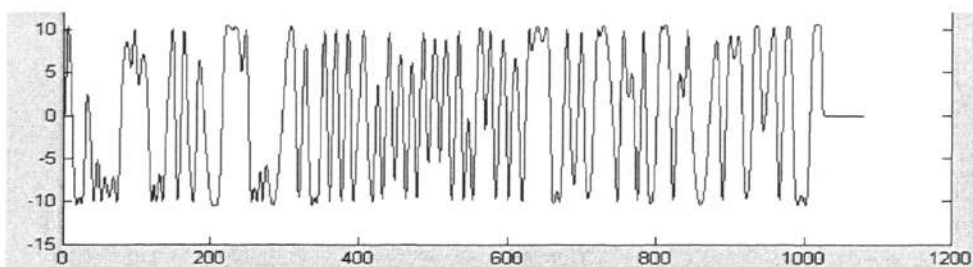


图 3.41 G 点 I 路抽取滤波波形

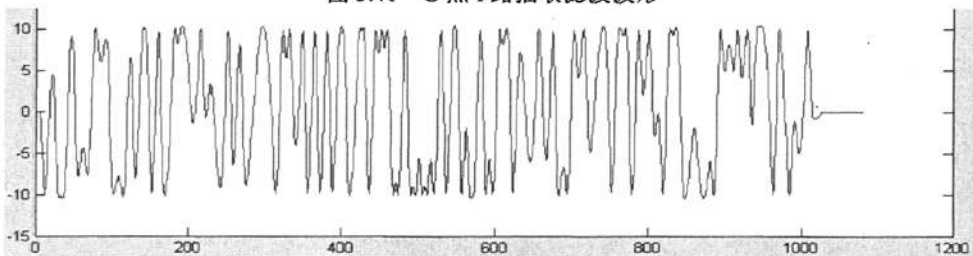


图 3.42 H 点 Q 路抽取滤波波形

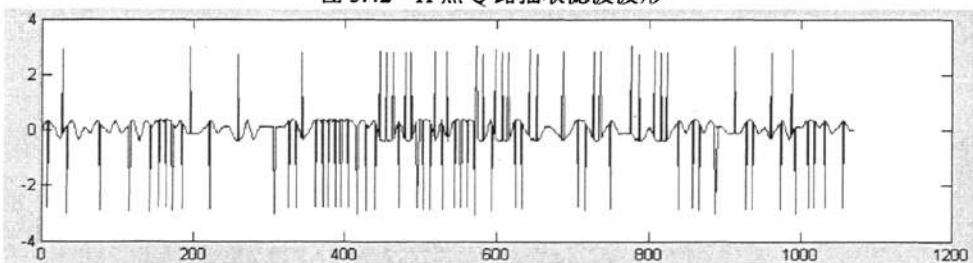


图 3.43 J 点差分鉴频波形

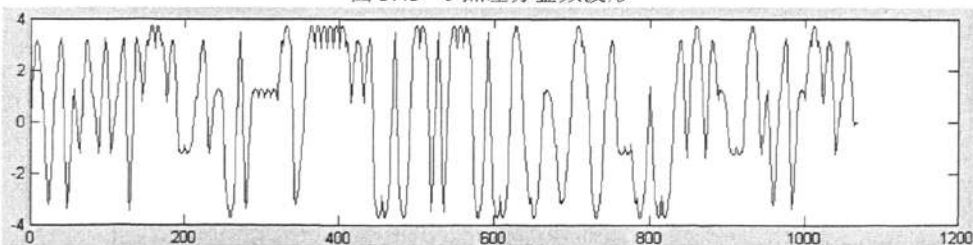


图 3.44 K 点相位校正后的解调波形

3.5.3 DMR 信号的正交调制解调仿真

一、仿真条件

在 MATLAB 下, 随机产生长度为 216 比特的 0、1 序列, 加上 64 位的 DMR 同步码, 得到 DMR 规定的一帧数据输入。将该输入映射、内插、基带成形滤波后, 采用 3.5.1 节所述方法正交调制, 得到的调频信号通过高斯白噪声信道后, 采用 3.5.2 节所述方法进行差分解调, 将解调后的数据匹配滤波, 采用 3.4 节所述方法进行解调处理。先进行相关运算, 找出第一个抽判点, 然后每隔八个点对匹配滤波输出

波形进行抽取，再根据固定门限 2、0、-2 对抽取的值进行判决、反射射、去同步，得到二进制比特输出。将输出信号与原 216 位二进制输入比较，进行误码率统计。下图为仿真模型。

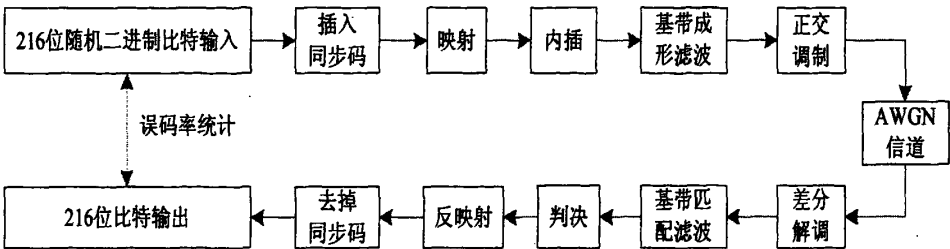


图 3.45 DMR 信号的正交调制解调仿真模型

将信噪比在 0 到 100dB 之间取步长 0.05，每个信噪比下统计 50 帧 DMR 二进制输入比特，得到的误码率仿真结果如下：

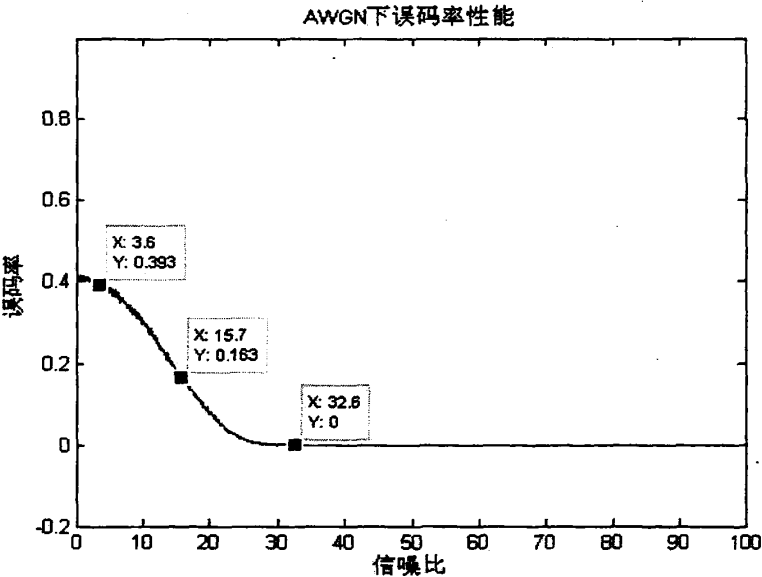


图 3.46 误码性能仿真图

二、仿真结果分析

由图 3.46 看出，采用正交调制解调的误码率随着信噪比的提高而减小。在信噪比达到 32dB 左右时，误码率达到 0。

第四章 DMR 端机基带处理的 DSP 实现

本章在前期做的仿真实现的基础上,在 DSP 上实现单工通信与双工通信的 C 语言程序方案,并在 DMR 端机硬件平台上实现。

4.1 TMS320VC5510 简介

TMS320VC5510^[13]是TI推出的一款低功耗定点型DSP芯片,主要面向低功耗、手持设备、无线终端应用。如图4.1示,由于5510的CPU地址总线是24位的,所以C5510的寻址空间为16MB。其中,片内存储器包括32KB的ROM、256KB的单存取RAM(SARAM, Single Access RAM)和64KB的双存取RAM(DARAM, Dual Access RAM),使得程序和数据能在片内进行高速传递,实时完成各种任务。SARAM在一个周期内只能访问一次,DARAM则能访问两次。显然DARAM的速率比SARAM快,因此在实际的应用中,常把程序代码放在SARAM中,而把访问较频繁的变量、堆栈等放在DARAM中。

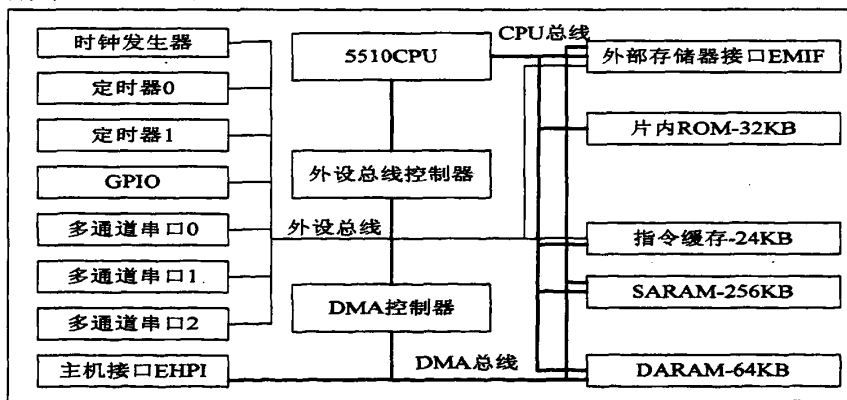


图4.1 VC5510结构框图

TMS3205510 DSP 有两种类型的存储器:程序器和数据存储器,其存储空间包括统一的数据/程序空间和 I/O 空间。其存储器映射关系如图 4.2 所示,C5510 的寻址空间为 16MB,当 CPU 从程序空间读取程序代码时,使用 24 位地址;当访问数据空间时,使用 23 位地址。但是在访问数据空间时将 23 位地址左移一位,并将地址总线上的最低有效位(LSB)置 0,使得在对数据空间或程序空间寻址时,地址总线都传送 24 位地址。

DSP 的数据空间被分为 128 个主数据页(第 0 页~第 127 页),每个主数据页的大小为 64KB,指令通过 7 位的主数据页值和 16 位的偏移值共同确定数据空间的任何一个地址。在第 0 主数据页中,前 96 个地址(000000h~00005Fh)为存储映射寄存

器(MMR)保留, 相对应在程序空间有 192 个地址(000000h~0000BFh), 建议不要使用这段存储区。

在 DSP 的程序存储器中包含的是可执行的程序代码, 而数据存储器中则是一些程序所需的外部变量、静态变量和系统堆栈。DSP 生产厂商及第三方为 DSP 软件开发提供了 C 编译器, 使得利用高级语言实现 DSP 程序的开发成为可能。C 编译器共创建 4 个已初始化块(.text 块、.cinit 块、.const 块和.switch 块)和 3 个未初始化块(.bss 块、.stack 块、.system 块)^[20]。其中,.text 块包含可执行代码和字符串;.cinit 块包含初始化变量和常数表;.const 块包含字符串和 switch 表;.switch 块包含为.switch 语句建立的表格。未初始化块用于保留存储器空间, 程序利用这些空间在运行时创建和存储变量。其中,.bss 块用于保留全局和静态变量空间, 在程序开始运行时, C 初始化 Boot 程序将数据从.cinit 块中拷贝到.bss 块;.stack 块为系统堆栈分配存储器, 用于将变量传递到函数和分配局部变量;.stack 块为系统堆栈分配存储器, 用于将变量传递到函数和分配局部变量。

本文基于 TMSVC5510 DSP 的程序开发都是用 C 语言实现^[21]。由于涉及到片内外设的配置, 使用到了 DSP 开发软件 CCS3.1 提供的 DSP/BIOS 的芯片支持库函数(CSL)。CSL 提供 C 语言可调用的 DSP 外围接口库函数, 其中包括 DMA 模块、McBSP 模块、TIMER 模块等。应用这些库函数可大大提高程序可读性, 缩短软件开发周期。

4.2 DMR 端机平台介绍

DMR 端机的硬件平台主要包含两个部分^[22], 一是射频模块, 另一个是基带模块。基带模块的组成结构如图 4.2 所示, 包含了数字信号处理器(DSP)、微处理器(MCU)、声码器(AMBE2000)、flash 闪存、显示输出等。

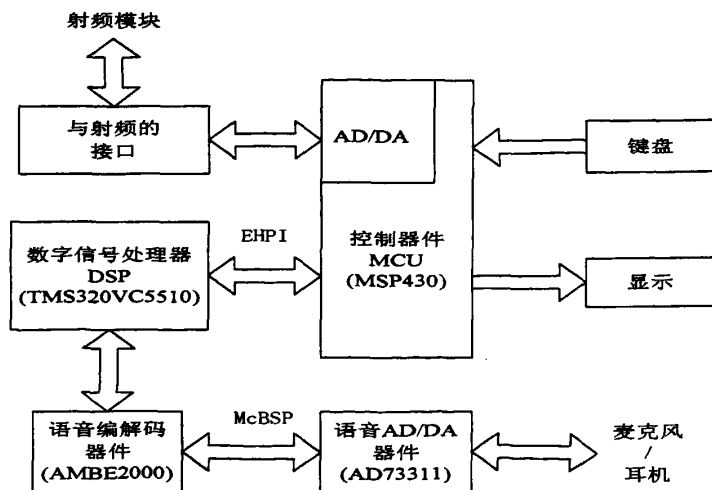


图 4.2 DMR 端机基带平台

MCU 采用 TI 公司的 MSP430FG4619, 用于完成网络接入层和部分数据链路层的处理, 倾向于系统控制。声码器采用 AMBE-2000, 完成语音编码解码功能。flash 存储器采用 AMD 的 AM29LV800BT, 用于开发完成后存放程序代码。TMS320VC5510 型定点芯片 DSP, 用于完成物理层的处理, 倾向于数字基带信号的处理。

4.2.1 DMR 端机中不同类型数据的基带处理

图 4.3 为 DMR 端机基带数据流图, 由图可见 DSP 基带处理包括对两种类型数据帧的处理: 语音帧和信令帧。如图 4.3 示, 在发送端, 语音帧的基带处理过程为: 语音处理芯片 AMBE-2000 以一定速率通过 A/D 采样模拟语音, 将得到的数字信号进行语音编码, 然后每 20ms 一个数据包传送至 DSP, DSP 对接收的语音帧组帧, 加同步码, 调制处理之后, 通过 MCU 的 D/A 送至射频模块调频发送。在接收端, 语音帧基带处理过程为: 射频模块接收鉴频的信号, 由 MCU 的 A/D 采样得到一帧数字信号, DSP 对该帧解调完成后, 进行拆帧处理, 以 20ms 一个数据包的速率传送至 AMBE 语音解码, 最后通过扬声器得到输出的模拟语音。

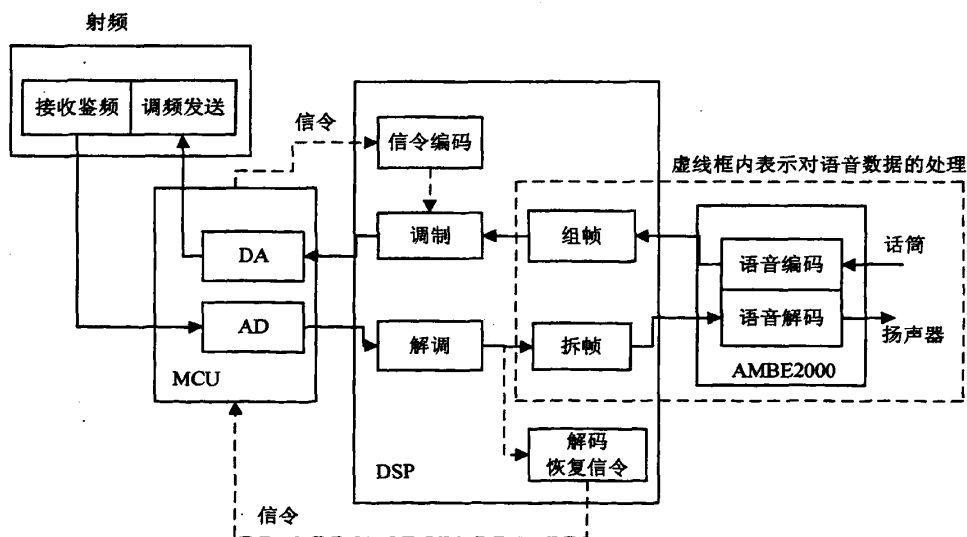


图 4.3 基带数据处理流程图

DMR 端机中, 用到了四种信令进行呼叫控制^[22]。主呼首先要向被呼发送呼叫请求信令 (TX-CSBK), 被呼方收到后, 若是空闲状态, 给主叫发送呼叫应答信令 (UU-Ans), 主呼收到后会发送语音控制启动信令 (LC-Header), 被呼收到这个信令后, 接通 AMBE2000, 即可进行语音通话了, 主呼如果想结束通话, 要向被呼方发送语音结束信令 (LC-Terminator) 从而结束通话。TX-CSBK 和 UU-Ans 都是长度为 80bit, LC-Header 和 LC-Terminator 长度为 72bit。

DMR 端机中由 MCU 完成呼叫控制, 信令帧不需要进行图 4.3 矩形虚线框所包含的处理。首先由 MCU 产生原始的信令码, 然后送给 DSP, 在 DSP 中对其进行一系列的编码交织等差错控制处理, 加信令同步码、调制, 再由 MCU 的 D/A 送至射频模块调频发送。接收端解调完成后, 再进行一系列的解码操作, 恢复原始信令, MCU 判断信令类型后作出相应的控制操作。

因此可以看出, DMR 物理层的基带处理技术, 包括同步定时, 语音帧的组帧拆帧, 调制解调, 信令帧的编解码等几乎都在 DSP 上实现, 如何正确有效的将各个功能模块组织起来是构建基带处理系统要解决的问题。由图 4.3 可以看出, DSP 主要和 MCU、AMBE-2000 实时交换数据。以下将分别介绍 DSP 与 MCU、AMBE-2000 的数据传输方式以及 DSP 与外部存储器的接口。

4.2.2 DSP 与 MCU 数据传输方式介绍

在本项目中, MCU 与 DSP 是通过 16 位的增强型主机接口 EHPI 连接, 使 DSP 系统受控于外部主机。采用 VC5510 EHPI 的复用模式, 通过专门的 EHPI 通道与 DSP 内的 DMA 控制器相连, 使主机能访问 DSP 的片内存储空间和部分外部存储空间^[14]。因此 DSP 要传送给 MCU 的数据并不用直接送到 MCU, 只需存放在 DSP 中指定的内存区, 给出相应标志或中断, 通知 MCU 来读取; 同理, MCU 给 DSP 的数据不需要 DSP 去接收, MCU 直接通过 EHPI 将数据写到 DSP 指定的内存区, 完成后给 DSP 中断或标志即可。

表 4.1 为 DSP 中定义的 DSP 和 MCU 都能访问的公共数据区, 主要存放两者之间通信的标志和呼叫控制所用的信令。此外, DSP 处理好的将要由 MCU 发送的数据, 以及 MCU 接收的 A/D 采样数据, 都在 DSP 中开辟了相应的空间存放。

表 4.1 DSP 中开辟的与 MCU 共用的存储区

存储区	作用	DSP 内存地址 (字地址)
FLAG D2M	DSP 发 MCU 收的标志	0x6000
发信令区	MCU 发往 DSP 的信令 (80bit)	0x6001~0x6050
调制数据区	待发送往射频的调制数据 (1056)	0x6051~0x6470
接收数据存储首地址	指明首地址, 说明 MCU 采样数据存储 在哪个区	0x6471
接收数据区 (共两个区域)	MCU 接收采样得到数据 (各 1420 个)	0x6472~0x6A07 0x6A08~0x6F9D
收信令区	DSP 解调给 MCU 的信令 (80bit)	0x6F9E~0x6FED
FLAG M2D	MCU 发 DSP 收的标志	0x6FEE

其中 FLAG_D2M 和 FLAG_M2D 是 DSP 与 MCU 之间进行通信的标志。FLAG_D2M 是 DSP 给 MCU 的标志, DSP 写 MCU 读。

4.2.3 DSP 与 AMBE-2000 数据传输方式介绍

VC5510和AMBE-2000之间数据信息的交换是通过标准的多通道缓冲串口(McBSP, Multi-channel Buffered Serial Port)来实现的。由于基带系统中语音部分是不允许被打扰的,其收发的中断优先级也应该是比较高的。硬件选用串口0(McBSP0)与AMBE-2000相连。EPR信号是AMBE-2000发送与VC5510的数据准备好信号,每20ms一次高电平突发,因此可以作为AMBE-2000的数据准备好中断信号来中断VC5510,提示其进行接收或者传送。VC5510可以有6个外部中断源(INT0~INT5),本系统中EPR引脚与VC5510的INT3连接。

AMBE-2000 编码速率为 3.6Kbps, 帧长 20ms, 每帧承载有效数据 72bit。其每 20ms 编码产生一个数据包, 每个数据包包含 24 个 16 位的字。相反, AMBE-2000 以此速率进行解码, DSP 往 AMBE-2000 发送的数据也是 20ms 一次, 每次发送 24 个 16 位的字。将 Mcbsp0 配置为一个字一帧, 接收来自 AMBE-2000 的数据时, 帧同步信号 FSX 由外部器件 AMBE-2000 提供; Mcbsp0 发送时, 发送帧同步信号 FSR 则由 Mcbsp0 自己产生。

4.2.4 DSP 与外部存储器的接口

TMS320VC5510 DSP 的外部存储器接口 EMIF 有 22 位的地址总线, 32 位的数据总线, 4 个片选输出和其它多种控制信号。它能支持多种不同类型的外部存储器件, 除了对异步存储器的支持以外, 还提供对同步突发静态存储器(SBSRAM)和同步动态存储器(SDRAM)的支持。异步存储器可以是静态随机存储(SRAM)、闪存等存储设备, 也可以是 A/D 转换器、并行显示设备等, 只是使用这些设备需要增加一些外部逻辑器件来保证设备的正常使用。

在本项目中, DSP 通过 EMIF 与闪存 AM29LV800BT 相连, 闪存对应 EMIF 接口的片选空间 CE1。当整个系统软件调试开发完成后, 编写在线烧写程序将 DSP 程序通过 EMIF 烧写到 AM29LV800BT 中, 之后即可脱离仿真器和 CCS 仿真环境运行。

4.3 DMR 时隙的建立

DMR 采用时分双工(TDD)的工作方式。根据 DMR 的帧结构我们知道, 一个帧长为 60ms, 包含两个时隙, 每个时隙长 30ms。而 DMR 协议规定每一个突发总长度为 30ms: 27.5ms 用于 264 比特内容, 余下 2.5ms 为保护时间, 一个突发正

好在一个时隙中进行传输。

DSP 有两个定时器, Timer0 和 Timer1, 可以用来定时或计数^[22]。定时器的计数器会在每个时钟周期减 1, 减到 0 时就产生一个输出信号。该输出可用于中断 CPU 或触发 DMA 传输, 这种方式非常有利于对数字通信中的帧、时隙等进行处理。本项目采用 DSP 的定时器 1 定时 30ms 作为时隙建立的基础。在定时器配置好后, 它的输出信号可用于中断 CPU, 指示时隙的开始和结束。

根据功能把时隙性质定为两种: 收时隙(RX)和发时隙(TX)。由于要实现的系统是一个实时系统, 发时隙要对数据帧进行发送, 那么对这些数据帧的处理必须要在发时隙的上一个收时隙中完成。同理, 收时隙若要对数据帧进行接收, 那么下一个发时隙中必须要完成对这些接收数据帧的处理。下图 4.4 为 DMR 单工的收发时隙功能图。

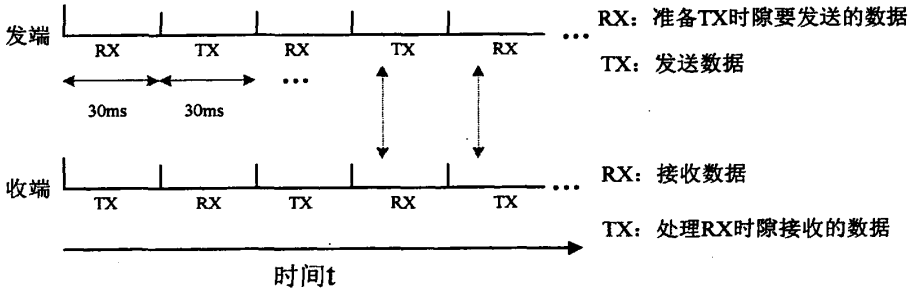


图 4.4 DMR 单工的收发时隙功能图

DMR直通模式下, 源端的MS建立定时参考。目的移动台发送反向信道信令去源, 并且同步于前向路径, 前向路径定时作为反向信道定时的参考基准。一旦源移动台停止传输, 任何一个希望传输的移动台将发送异步消息并且建立一个新的定时参考^[3]。也就是说, 发端只要按下相应的功能键后, 就能建立起收发时隙交替的定时结构, 收端要按发端的定时参考建立自己的定时。怎样建立起这种收发端时隙对准的结构, 这就是DMR初同步的过程, 具体会在4.5节中介绍。

4.4 DSP 中断在 DMR 端机中的应用

DSP中断^[14]是由硬件或软件驱动信号产生的, 它使DSP暂停当前程序的执行而去执行中断服务程序(ISR)。在DSP应用系统中, 中断是完成数据传递、实时处理等的重要手段, 因而用C语言完成对DSP中断设置是DSP开发的重要内容。本项目作为一个实时系统, 主要的数据传输和握手是通过中断来实现的。

5510中有中断使能寄存器(IER)和中断向量寄存器(IFR), 包含所有可屏蔽中断的标志位。当一个可屏蔽中断请求发送给CPU时, CPU把IFR中相应的标志位设置为1, 表明中断被挂起或等待CPU的响应。通过读IFR来识别挂起中断, 向IFR写入0

来清除挂起中断。所有挂起中断都可以通过将当前 IFR 的内容回写 IFR 或响应硬件中断请求来清除 IFR 中相应的标志位。将 IER0 或 IER1 中相应的使能位设置为 1 时, 允许一个可屏蔽中断; 将相应的使能位清 0 则禁止一个屏蔽中断。在复位时, 所有的 IER 位被设置为 0, 此时禁止所有的可屏蔽中断。

DSP 按 4 个主要步骤处理中断:

- 1) 接受中断请求, 挂起当前程序。
- 2) 响应中断请求, 若中断是可屏蔽的, 则在响应时必须满足某些条件; 对非屏蔽中断, 则立即响应。
- 3) 准备中断服务程序, CPU 执行的主要任务是: 完成当前指令的执行, 清除流水线中未译码的指令; 将相关寄存器自动保存到数据堆栈和系统堆栈中; 从中断矢量表取出中断矢量, 中断矢量指针 (IVPD, IVPH) 指向中断服务程序。
- 4) 执行中断服务程序 (ISR), ISR 中包含了中断返回指令, 当中断返回时, 自动恢复前面保存的寄存器。

VC5510 的中断之间遵循如下规则: 高优先级的中断可以打断低优先级中断, 而低优先级的中断不能打断高优先级的中断。同时, 当低优先级和高优先级两种中断同时发生的时候, 只响应高优先级中断。

4.4.1 中断优先级安排

VC5510 不同中断源的中断优先级是固定的^[24], 下表 4.2 给出本项目中断源及与该中断相关联的中断服务子程序。

表 4.2 中断安排

中断源	中断优先级	中断服务程序	功能
Timer0	6	timer0Isr_cnt()	初同步期间的计数
McBSP0	7	rcv_isr()	语音的组帧拆帧
HPI	14	ehpi_isr()	主机中断
EPR	15	AMBE_isr(), EPR_isr_1()	AMBE 引起的外部中断
Timer1	23	timer1Isr()	时隙的定时

本系统中, 选用 Timer1 中断 (TINT1) 作为时隙定时功能, 中断优先级为 23。MCU 引起的主机中断为 DSPINT, 中断优先级为 14; AMBE-2000 引起的外部中断可以通过 EPR 信号与 DSP 中断管脚的连接灵活配置, 因为该中断要比 Timer1 中断优先级高, 又要比主机中断低, 所以可选择优先级为 15 的外部中断 3 (INT3)。另外还用到了 Timer0 的中断 (TINT0) 和 McBSP0 的中断 (RINT0), 中断优先级分别为 6 和 7。Timer0 中断用于时隙同步期间的计数, 与其他中断在时间上没有冲突;

McBSP0 的中断用于接收来自 AMBE-2000 的数据,高优先级令其不被其他中断干扰,保证了语音数据的可靠接收。

本项目在程序的.cmd 文件配置中,设置中断向量表 VECS 在第 0 主数据页,起始地址为 0000200h,长度为 00100h。以下在 DSP 中断相关设置的 C 语言实现格式:

```
IRQ_setVecs((Uint32)(&VECSTART)); //设置中断向量
old_intm = IRQ_globalDisable( ); //关闭所有的可屏蔽中断
/* 打开定时器 0 和定时器 1*/
mhTimer1 = TIMER_open(TIMER_DEV1, TIMER_OPEN_RESET);
mhTimer0 = TIMER_open(TIMER_DEV0, TIMER_OPEN_RESET);
/*取得定时事件 ID */
eventId1 = TIMER_getEventId(mhTimer1);
eventId0 = TIMER_getEventId(mhTimer0);
/*对 IFR 寄存器进行操作,中断标志位清零*/
    IRQ_clear(eventId1);
    IRQ_clear(eventId0);
    IRQ_clear(IRQ_EVT_DSPINT);
    IRQ_clear(IRQ_EVT_RINT0);
    IRQ_clear(IRQ_EVT_INT3);
/*将中断函数与中断事件相关联*/
    IRQ_plug(IRQ_EVT_RINT0,&rcv_Isr);
    IRQ_plug(eventId1,&timer1Isr);
    IRQ_plug(eventId0,&timer0Isr_cnt);
    IRQ_plug(IRQ_EVT_INT3,&EPR_Isr_1);
    IRQ_plug(IRQ_EVT_DSPINT,&ehpi_Isr);
/*对 IER 寄存器进行操作,使能中断*/
    IRQ_enable(IRQ_EVT_DSPINT);
    IRQ_enable(IRQ_EVT_INT3);
    IRQ_enable(eventId1);
    IRQ_plug(IRQ_EVT_RINT0);
/*对 CSR 中 GIE 标志位进行配置,标志位置 1,使能所有可屏蔽中断*/
    IRQ_globalEnable();
```

4.4.2 中断在语音组帧拆帧中的应用

DSP 对 AMBE-2000 数据包的收发为语音帧的拆帧组帧过程。由 4.2.3 节知, AMBE 每 20ms 产生一个数据包, 每个数据包包含 24 个 16 位的字, EPR 信号是 AMBE-2000 发送与 VC5510 的数据准备好信号, 每 20ms 一次高电平突发, EPR 与 DSP 的外部中断源 INT3 连接, 用于语音帧组帧开始或拆帧开始的触发。

单工语音帧组帧时, EPR 触发中断函数 `EPR_Isr1()`, 在该中断函数中, 使能串口 0 (Mcbsp0) 的接收中断 RINT0, 该中断由 AMBE-2000 产生, 它的中断服务程序为 `rcv_Isr()`。AMBE 每传送产生的一个字执行一次 `rcv_Isr()`, 如果 24 个字接收正确, 中断服务程序 `rcv_Isr()` 执行 24 次。这样 EPR 每触发一次, DSP 完成 20ms 语音数据的接收。这样接收三个 20ms 的数据包后, 将其组帧形成 216bit 的语音数据, 然后置语音组帧完成标志 `voice_in_flag=1`。如图 4.5 为语音组帧的中断函数流程图。

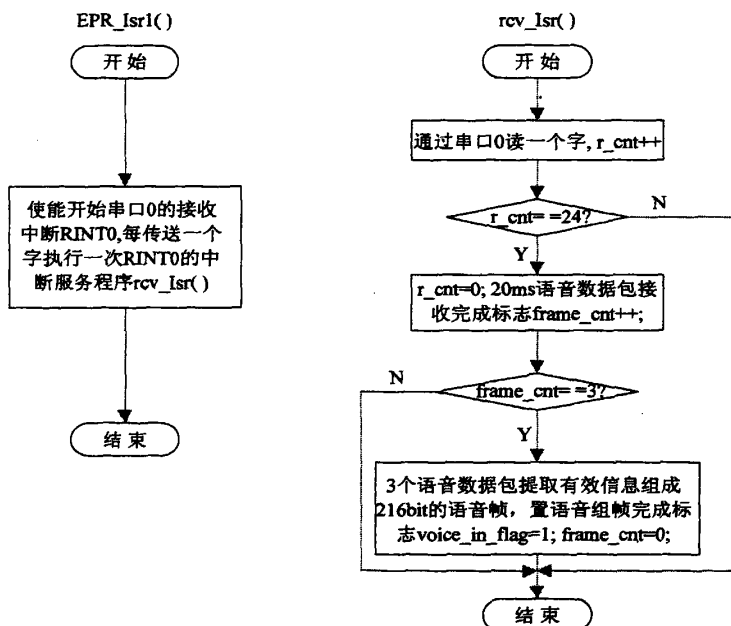


图 4.5 语音组帧的中断函数流程图

语音拆帧时, 在同样由 EPR 触发中断函数 `AMBE_Isr()` 中, 检测是否有解调好的语音帧数据, 通过 `voice_out_flag` 标志是否为 1 来判断, 若有, 则语音数据转换成符合 AMBE-2000 解码的数据包。由 4.2.3 节知, 即一帧解调好的语音帧要拆分成三个 20ms 的数据包, 且每个数据包包含 24 个 16 位的字, DSP 以查询的方式把这 24 个字送至 AMBE。AMBE_Isr() 执行三次就能把一个语音帧拆分成三个数据包发送完成。如图 4.6 为语音拆帧的中断函数流程图。

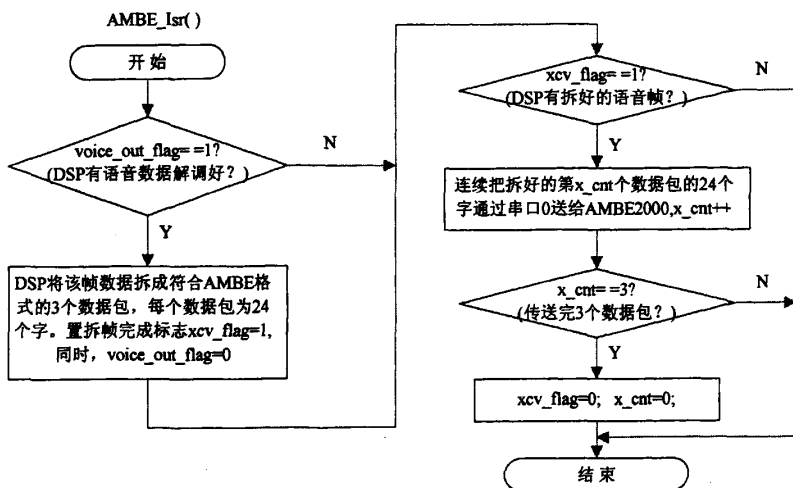


图 4.6 语音拆帧的中断函数流程图

4.4.3 中断在 DSP 与 MCU 之间通信的应用

DMR 端机中 MCU 实现呼叫控制功能, 呼叫控制是通过信令实现的。DSP 对数据进行解调处理时, 若判断出是信令, 则需要给 MCU 中断, 将信令送往 MCU 由 MCU 判断信令类型, 并完成相应操作。此外, 由图 4.3 和图 4.4 看出, DSP 在时隙中发送数据和接收数据, 只需在时隙初给 MCU 中断, 由 MCU 的 D/A 和 A/D 完成, DSP 不需作任何操作。

因此总结出, DSP 给 MCU 的中断有三种情况: (1) DSP 解调发现是信令需 MCU 判断信令类型; (2) DSP 有数据调制完成, 需要 MCU 进行 D/A 发送; (3) DSP 需要 MCU 进行 A/D 接收数据。此部分的软件是在 MCU 中实现的, MCU 可以通过表 4.4 中 FLAG_D2M 标志判断进行相关操作, 本文不作介绍。

MCU 给 DSP 的中断(主机中断)也有三种情况: (1) MCU 有信令产生(标志位 `command_in=1`)需要 DSP 调制处理; (2) DSP 解调数据后判断该数据为信令(标志位 `command_in=0`), MCU 判断出了信令类型, 此时 DSP 要作出相应操作开始或结束语音通信; (3) MCU 指示 DSP 进行收发端的切换(仅单工时产生)。此部分的软件是在 DSP 中实现的, 以下给出介绍。

在 MCU 给 DSP 中断的中断服务程序 `ehpi_Isr()` 中, 若为单工通信, 先要判断是否要进行收发端的切换。单工收发两端外部中断 INT3 的中断服务程序不同: 主呼端有按键按下, 有相应信令产生标志位 `command_in=1`, 再将其余标志位置位后, 判断信令类型, 若为 LC 头, 开 AMBE2000, 配置语音帧组帧程序 `EPR_Isr1()`; 若为语音结束信令, 则进行系列操作停止语音传输。被呼端标志位 `command_in=0`, 对 DSP 解调出的信令判断类型, 若为 LC 头, 则开 AMBE2000 配置语音帧拆帧程

序 `EPR_Isr2()`，使能中断；若为语音结束信令，进行系列操作停止语音传输。在双工通信中，主呼和被呼两端的中断服务程序一样，不需要分别配置。下图 4.7 为 MCU 给 DSP 中断的中断函数流程图，单工为虚线加实线部分，双工为实线部分。

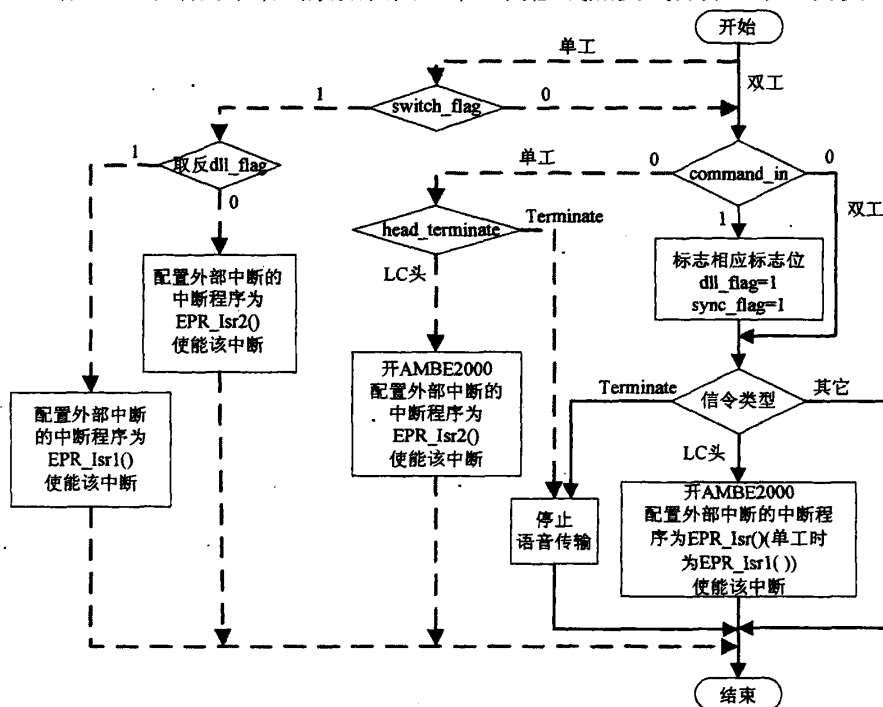


图 4.7 MCU 给 DSP 中断的中断函数流程图

4.5 DMR 基带处理的 DSP 实现

4.5.1 DSP 程序的初始化问题

在执行程序之前，DSP 有一系列的初始化工作。首先对工作时钟初始化。于 DSP 采用的输入时钟频率为 25MHz，DSP 的 CPU 及其外围电路所需的工作时钟为 200MHz。对 5510 的时钟发生器的模式控制寄存器 CLKMD 编程配置，即得到相应的倍频输出。之后，配置定时器 0 和定时器 1，对 Timer1 使用外部时钟 25MHz 作为工作时钟。Timer0 用于初同步期间的模拟 MCU 采样时钟 0.0208ms，采用内部时钟 200MHz 作为工作时钟。对定时器设定为自动重装模式，计数到 0 时，定时器又会自动配置，继续定时。McBSP0 用于 AMBE 发送和接收，配置为一个字一帧。

由于中断函数不能返回值。因此，本文设计在 DSP 内部不同中断函数之间进行通信时，用到全局变量。DSP 使用的主要全局变量标志含义如下表示。

表 4.3 DSP 中主要全局变量

标志	含义
sync_flag	同步标志。主呼端为 1,被呼端初同步完成后置为 1
rt_flag	时隙标志, 置 1 为收时隙, 置 0 为发时隙
dll_flag	主呼端标志, 置 1 为主呼端, 置 0 为被叫端
30ms_flag	置 1, 2, 3, 4 分别表示主呼端发时隙, 主呼端收时隙, 被叫端发时隙, 被叫端收时隙。初始时置为 0
voice_in_flag	语音帧组帧完成标志
voice_out_flag	语音帧拆帧完成标志

dll_flag 为主呼端标志, 置 1 为主呼端, 置 0 为被叫端。sync_flag 为同步建立好标志, 主呼端在响应主机中断函数时将其置 1。被呼端则要在初同步完成后才能置 1。rt_flag 用来区分时隙, 同步建立好后, 收发时隙交替。voice_in_flag 和 voice_out_flag 为语音帧组、拆完成标志。

表 4.4 FLAG D2M 各位定义

15--4	3	2	1	0
保留	Tx_Anounce	Rx_Anounce	xmt_flag	command_out

第 0 位: command_out 信令解调标志, 1 有, 0 无。

第 1 位: xmt_flag 调制数据标志, 1 有, 0 无。

第 2 位: Rx_Anounce 置 1 表示 DSP 告之 MCU 接收数据

第 3 位: Tx_Anounce 置 1 表示 DSP 告之 MCU 发送数据

DSP 解调出来的信息如果是信令的话, 会将信令写到收信令区(0x6F9E~0x6FED), 同时将 command_out 置 1, 并给 MCU 中断信号, 以便 DSP 能及时处理信令并作出相应反应。当 DSP 调制完成时, 会将要发送的 1056 个数据写到调制数据区(0x6051~0x6470), 同时将 xmt_flag 置 1。Rx_Anounce 与 Tx_Anounce 则是在每个 30ms 时隙开始时, DSP 根据时隙功能来相应作出更改, 指示 MCU 是接收还是发送。

FLAG_M2D 是 MCU 给 DSP 的标志, MCU 写 DSP 读。定义如下:

表 4.5 FLAG_M2D 各位定义

15--5	4	3	2	1	0
保留	switch_flag	head_terminate	rqst_lc_flag	rcv_flag	command_in

第 0 位: command_in 有无信令标志, 1 有, 0 无。

第 1 位: rcv_flag 接收数据完成标志, 1 是, 0 否。

第 2 位: rqst_lc_flag 信令长度标志, 1 代表 80bit, 0 代表 72bit。

第 3 位: head_terminate 区分语音头和结束的标志, 1: head, 0: terminate。

第 4 位: swith_flag 收发切换标志(仅单工通信时用), 1 切换, 0 否。

当 MCU 产生信令时, 会把信令写到发信令区(0x6001~0x6050), 同时根据信令的内容修改 rqst_lc_flag 和 head_terminate, 将 command_in 置 1, 给 DSP 中断信号。如果 MCU 检测到 DSP 送给它的信令是语音头或语音结束标志, 则会置 head_terminate 为相应值, 并把 command_in 置 0, 给 DSP 中断信号。MCU 在接收时隙接收满 1420 个数据时, 会将数据写到接收数据区(0x6472~0x6A07 或 0x6A08~0x6F9D, 这两个数据区交替使用), 并把本次存放的数据区的首地址(0x6472 或 0x6A08)写到 0x6471, 然后把 rcv_flag 置 1。swith_flag 用于单工通信时, MCU 如果检测到键盘的切换键被按下, 则将该位置 1, 并给 DSP 中断信号, 以便 DSP 作出相应变换。

由于 AMBE 的 EPR 语音中断为 20ms 一次, DMR 定时隙所用的 Timer1 的中断为 30ms 一次, 因此可能导致这两个中断冲突。由于 EPR 的中断优先级更高, 以 EPR 作为基准, 在程序初始运行时, 定时 7ms 之后再开定时器 1 和使能定时器 1 的中断, 这样就可以保证两者不会冲突。

4.5.2 DMR 的初同步

由于主呼端建立独立的定时参考, 被呼端在初始时未建立与主呼端对准的同步关系, 即 30ms 的数据突发可能没有完整地落在 30ms 的收时隙内, 如图 4.8 示。因此, 被呼端在每一个时隙都接收数据, 同时不断在检测数据的有效性。当检测到有效数据时, 连续两个时隙接收数据, 再关闭定时器 1 以及定时器 1 的接收中断。由于两个时隙长度为 60ms, 数据突发长度为 30ms, 因此两个时隙一定能保证能完整接收一帧数据。在第三个时隙里, 将数据与本地存储的同步码波形进行相关运算, 返回相关峰的位置 sync_location。若相关峰返回位置大于一帧数据的采样点数 N, 将 sync_location 减去 N。理论上, 若时隙完全对准, 相关峰的位置距一帧数据完成之间的时间间隔是 12.5ms。用定时器 0 配置 MCU 的采样时间间隔为 0.026ms, 即每 0.026ms 进行一次定时器 0 的中断函数 timer0Isr_cnt(), 在该中断函数中设定计数器, 当计数器从 0 计数至 sync_location 时, 立即配置定时器 1 定时 12.5ms, 之后打开定时器 1 的时隙中断定时功能 timer1Isr()。这样就保证了数据突发在 30ms 时隙内完整接收, 初同步建立完成, 置被呼端同步标志 sync_flag 置为 1。

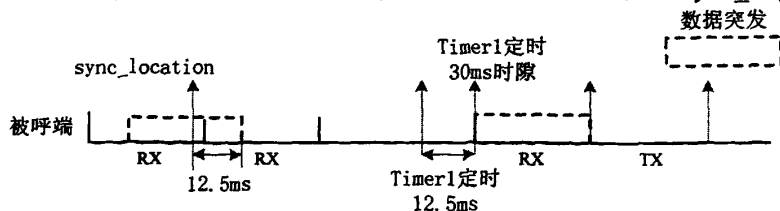


图 4.8 初同步示意图

4.5.3 DMR 单工通信的 DSP 实现

DMR 单工通信主要完成图 4.4 所示的功能,在前面所述各函数功能模块的基础上,本节给出最主要的两个函数的流程图,这两个函数分别为单工通信的 DSP 主函数,以及用于 DMR 时隙定时的中断函数 `timer1Isr()`。图 4.9 为 DMR 单工通信的 DSP 主函数流程图,图 4.10 为定时器 1 定时 30ms 的中断函数 `timer1Isr()` 流程图,相应标志位的含义已在 4.5.1 节中给出。前一版在时隙定时的 30ms 中断函数 `timer1Isr()` 中完成调制解调等一系列处理,导致中断函数执行的时间过长。在笔者设计的 `timer1Isr()` 中,除了作时隙调整外,仅给出相关标志的设置,执行时间很短。在设计 DSP 主函数中,完成调制解调等系列处理,下图 4.9 为 DSP 主函数流程图。

两个 DMR 端机之间通信,当任何一方都没有按键按下时,如图 4.10 示,单工 30ms 中断函数中,DSP 告之 MCU 每个时隙都接收数据。DSP 主函数处于一个不断查询状态。当有一方按下键盘上的主呼键成为主呼端时,MCU 会给 DSP 中断,`dll_flag` 和 `sync_flag` 都置 1。主呼端收时隙时,在 30ms 中断程序中将 `30ms_flag` 置为 2。主函数判断出其值后,再判断 MCU 给其的标志是否为信令。若为信令,判断信令类型,再根据信令的不同加上不同的信令编码和同步码,调制信令。若为语音,则不加编码,直接将语音数据加上语音同步码调制。操作完成后,置调制准备好标志 `xmt_flag` 为 1。成为主呼端 `30ms_flag` 为 1 时为发时隙,此时隙中,通过 `xmt_flag` 判断是否有调制好的数据,若有,DSP 给 MCU 中断告之发送。

被呼端的处理较主呼端复杂很多。首先,按上节 4.5.2 所述的方法完成初同步的建立,`sync_flag=1`。发时隙中,DSP 通过 MCU 给的标志判断是否有接收好的一帧数据。若有,则判断该帧数据是否有效。对有效数据进行解调处理,若解调发现数据为信令,则对其进行去同步解码等操作,置 `command_out_flag` 为 1,并给 MCU 中断告之。如果数据为语音,则只需去同步,置语音解调好标志 `voice_out_flag` 为 1。在这个过程中,若数据无效,或者解调数据发现没有找到同步,会使无效帧计数器计数加 1,一旦累计到有两帧无效时,同步标志置 0,需返回再重新同步。

对被呼端而言,因为长时间的通信可能导致定时有漂移。本文通过解调时相关器返回的同步位置与理论精确同步好同步位置比较,若二者之间的偏移值在 3 个点范围内,则不需要调整,反之则调整,然后置时隙是否需要调整标志。

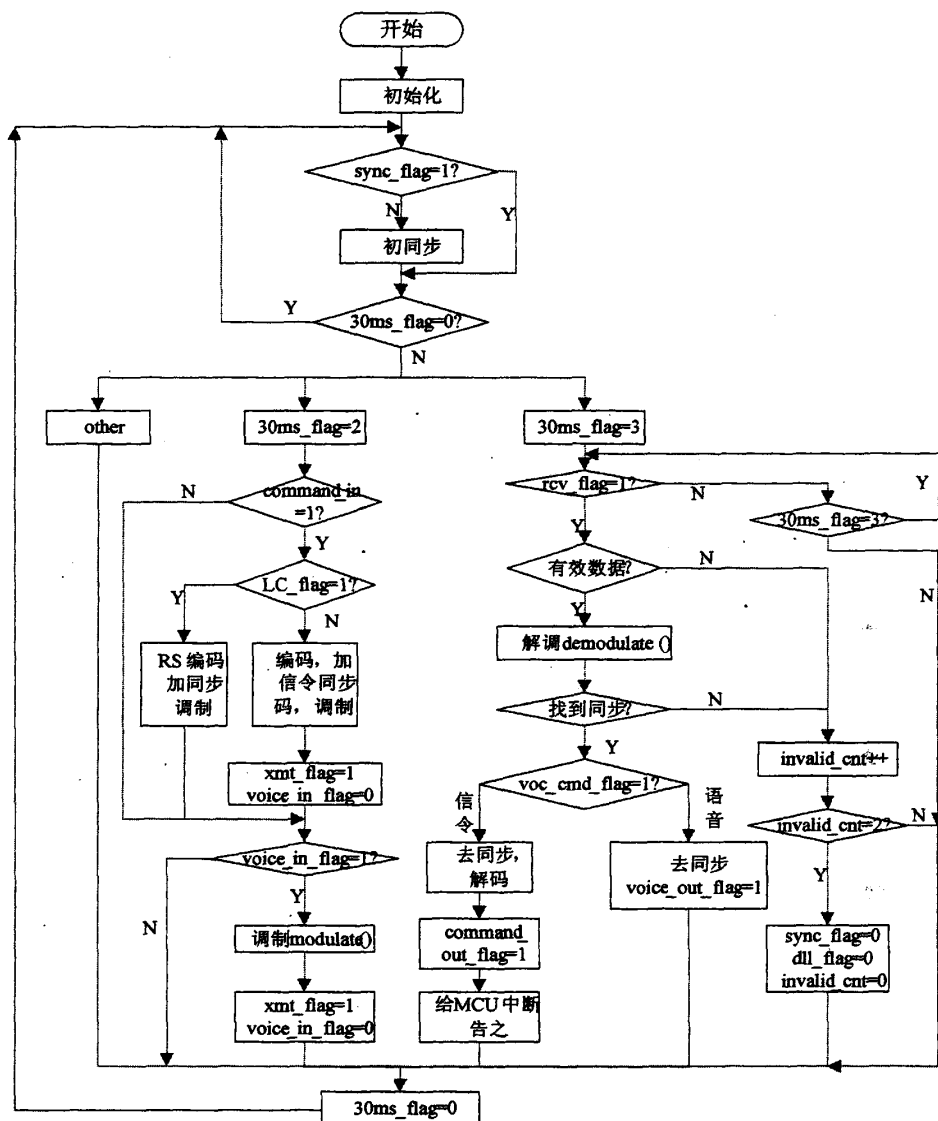


图 4.9 单工主函数流程图

如图 4.10 所示的 30ms 时隙定时中断函数中, 同步建立好之前(sync_flag=0), DSP 在每个时隙初给 MCU 中断, 告之每个时隙都接收数据。同步建立好后(sync_flag=1), 若为主呼端, 在 30ms_flag 为 1 时判断若有数据调制准备好, 则给 MCU 中断告之发送; 若为被呼端, 需要判断是否调整时隙, 若需要, 配置定时器 1 的定时长度来实现调整, 在 30ms_flag 为 4 时, DSP 给 MCU 中断告之接收。此外, 主呼端的收时隙、被呼端的发时隙的 30ms_flag 标志被分别置为 2、3。

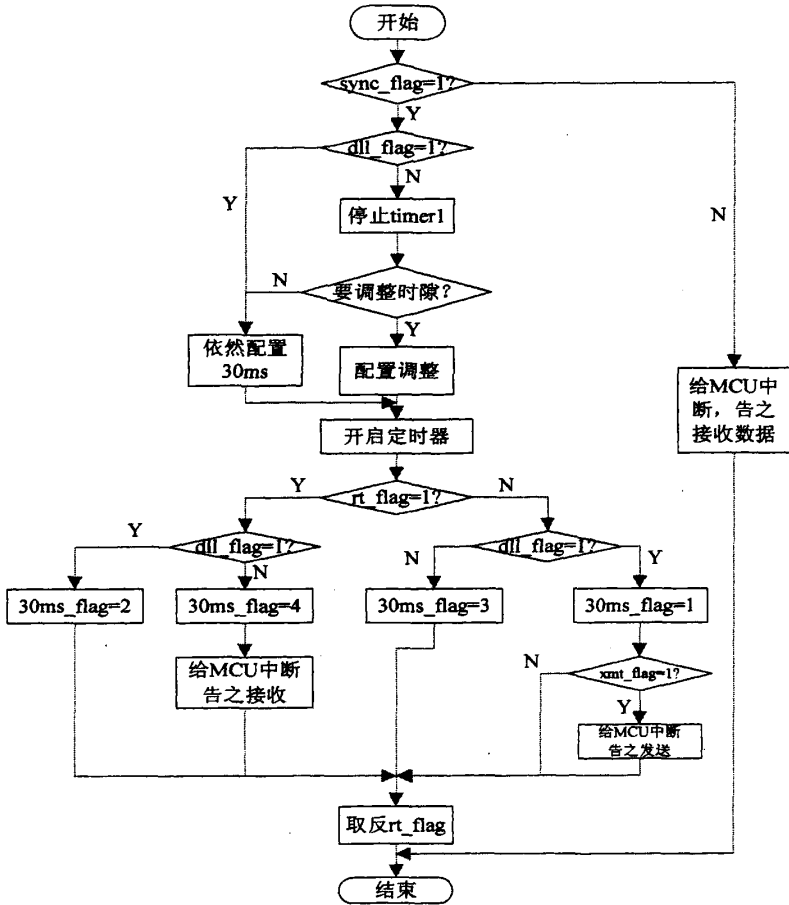


图 4.10 单工 30ms 时隙中断函数流程

4.5.4 DMR 双工通信的 DSP 实现

双工通信时，主呼端和被呼端都是数据发送端和数据接收端，时隙功能与单工相比更多。两端的收时隙（RX）除了准备发时隙（TX）要发送的数据外，也要接收数据，发时隙除了要发送数据外，也要处理接收的数据。在同步建立好后，两端是对称的，只区分时隙功能，如图 4.11 示。

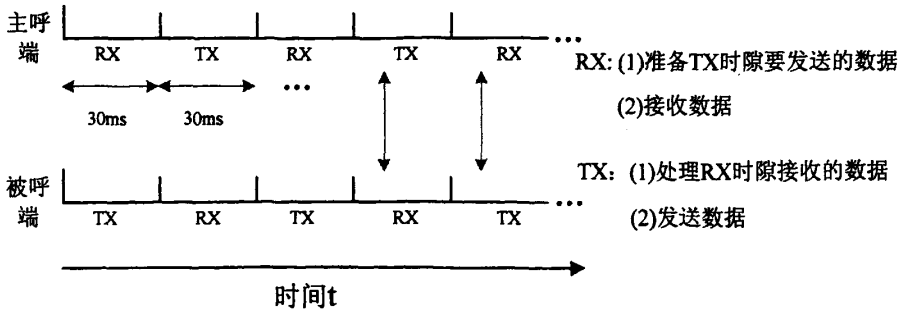


图 4.11 DMR 双工通信的时隙功能图

双工通信中用 DMA 控制器来接收 AMBE 传送的数据,初始化除了与单工相同的部分外,还包括了对 DMA 的初始化。DMA 控制器允许不受 CPU 控制而直接在内部存储器、外部存储器、片内外设和主机端口(HPI)之间传输数据^[25]。它有 4 个标准端口分别对应 4 个数据源:内部 DARAM、SARAM、外部存储器和外设。还有一个辅助端口用于 HPI 和存储器之间的数据传输,它们之间的数据传输不使用 DMA 中的通道。DMA 具有 6 个通道,可以设置每个通道的优先级,每个通道的传输可以由选定事件触发。当操作完成后, DMA 控制器可向 CPU 发出中断。

本项目中,使用 DMA 的通道 4,中断优先级为 21,配置 DMA 的源端口为外设,目的端口为 DARM,由于 AMBE 发送的是 24 个 16 位的字,因此 DMA 数据类型为 16bit。此外,配置 DMA 为自动初始化使能,同步事件为串口 0 接收事件 REVT0,每帧包含 24 个数据单元,每个数据块包含 3 帧,一帧传输结束时发中断。因此, DMA 中断事件也是 20ms 发生一次,配置 DMA 接收中断函数为 dmaRcvIsr()。此中断函数完成的功能,相当于单工时语音的拆帧和组帧。dmaRcvIsr()中判断当有数据解调完成(voice_out_flag=1)时,拆成 3 帧,每帧 24 个字。拆帧完成后,置拆帧好标志 ready_flag=1,xcnt=0,然后以查询方式,由串口向外送 24 个字,每个 20ms 送一帧,三帧送完后置 ready_flag=0。组帧时,定义一个计数器 frame_cnt, DMA 每接收 AMBE 一帧 20ms 的数据, frame_cnt 加 1,加到 3 时,将接收语音数据放入相应缓冲区内,并置语音准备好标志 voice_in_flag 为 1。所以 DMA 接收中断函数 dmaRcvIsr()包含了图 4.5 所示的 rcv_Isr()函数的组帧功能以及 AMBE_Isr()的全部功能。以下给出 DMA 中断函数配置的 C 语言实现:

```
hDmaRcv = DMA_open(DMA_CHA4,DMA_OPEN_RESET); /* 打开通道 4 */
rcvEventId = DMA_getEventId(hDmaRcv); /*取得 DMA 定时事件 ID */
IRQ_clear(rcvEventId);
IRQ_plug(rcvEventId,&dmaRcvIsr);
IRQ_enable(rcvEventId);
```

双工主函数流程与单工主函数(图 4.9 示)一样,不同的是引入语音通话开始标志 sound_start_flag。主呼端在语音开始传输时将该标志置 1,被呼端在初同步建立好,第一帧语音传输时将该标志置 1。此后,主呼端和被呼端的功能完全对称,只需要区分时隙的功能。在收时隙,除了象单工发端一样要准备下一时隙的发送数据外,还要完成数据的接收。发时隙除了要发送数据外,若有数据接收完成,应对数据完成解调等一系列处理。

图 4.11 为双工 30ms 时隙中断函数流程,当语音开始传输标志 sound_start_flag 未置 1 时,双工 30ms 时隙中断函数与单工时完全一致。sound_start_flag 置 1 之后,在发时隙中(rt_flag=0),若有数据调制准备好(xmt_flag=1),DSP 告之 MCU 发送数据,之后将 30ms_flag 置为 3,主函数中根据 30ms_flag 为 3,判断是否有数据来需

第五章 DMR 端机传输性能测试

本章在前面所做研究与实现的基础上，在 DMR 端机硬件平台上测试。在 CCS 下设置硬件仿真环境 C5510 v2.x XDS510 Emulator，将工程编译链接完成后，通过 DSP 仿真器下载到硬件芯片，配合 MCU 的程序进行一系列调试测试工作。

5.1 DMR 端机基带传输测试

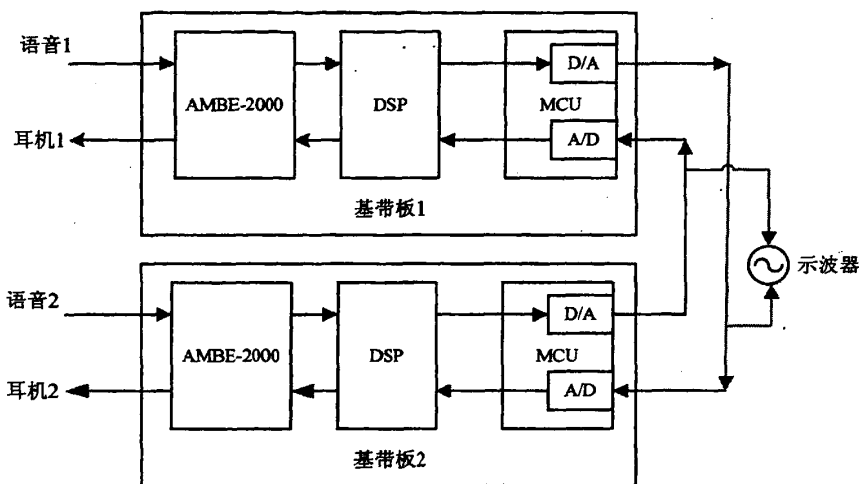


图 5.1 DMR 端机基带传输测试平台

基带传输测试平台采用两个基带板用导线对接，如图 5.1 所示，中间连示波器观察实验结果。

5.1.1 基带单工语音测试

按下主呼端基带板连接键盘的呼叫键，此时可在示波器上看到主呼端发出呼叫请求信令，在另一通道上可观察到被呼端的应答信令，这一过程很快，示波器上的波形一闪而过。在信令正确接收，呼叫信道建立后马上进入语音阶段。图 5.2 为基带板 1 作为主呼的单工语音测试的示波器截图，图中信号 1 代表主呼端，信号 2 代表被呼端。主呼端的 DSP 从 AMBE 接收数据，经过组帧、速率转换、成形滤波等发送端相关处理后在 30ms 的发时隙内由 MCU 的 D/A 发送。因此从示波器上能看到图 5.2 中信号 1 有数据发送。被呼端的 DSP 在发时隙处理接收数据，包括抽样、判决、反映射、拆帧等，然后将语音送至收端 AMBE 解码输出，通过示波器观察被呼端没有数据发送（图 5.2 中信号 2）。在基带板 2 端连接耳机可听到基带板 1 发送过来的语音，语音清晰流畅。将基带板 2 作为主呼，基带板 1 作为

被呼, 效果一样, 如 5.3 图所示。

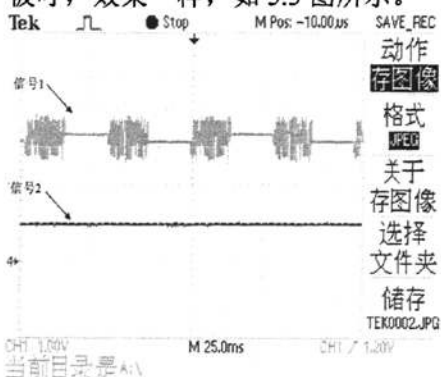


图 5.2 板 1 作为主呼的单工语音测试图

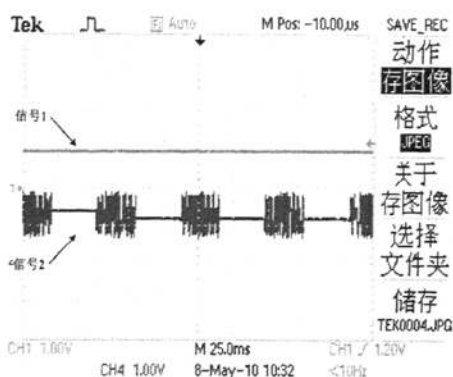


图 5.3 板 2 作为主呼的单工语音测试图

5.1.2 基带双工语音测试

基带板 1 和基带板 2 两边都通过麦克风连接语音。基带板 1 作为主呼端, 呼叫连接建立后, 在示波器上观察到板 1 和板 2 都有数据发送, 且两端时隙对准。如图 5.4 为双工语音通信的示波器截图。基带板 1 的耳机可听到基带板 2 发送过来的语音, 基带板 2 的耳机可听到基带板 1 发送过来的语音, 语音清晰流畅。

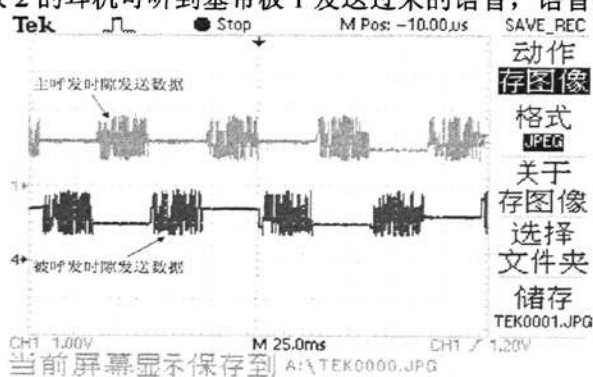


图 5.4 基带双工语音测试

5.1.3 基带误码性能测试

误码测试的信源采用的是伪随机序列, 是由级数 $n=10$ 和反馈系数 2443 的 m 序列产生器产生。主呼端在当 3 个 AMBE 数据包接收完成需要进行组帧形成 216bit 帧的时候, 将组帧函数替换成随机码生成函数, 由该函数产生 200 比特的二进制码元, 剩下的 16 比特作为帧号区填充帧号。为了防止帧号区传输出错, 采用 5 比特进行编号, 重复将帧号填充 3 次, 余下一个比特添 0。如下所示:

表 5.1 测试帧结构

200bit 伪随机码	5 bit 帧号	5 bit 帧号	5 bit 帧号	1bit
-------------	----------	----------	----------	------

接收端进行解调去同步之后,如果检测到三个帧号中有两个相同,即将其作为接收帧号。本地会同步生成与发送端相同的帧,并生成本地帧号,如果接收帧号与本地帧号相同,则将本地产生的帧与接收帧逐位比较前 200 位统计误比特数。测试结果见表 5.2 示。

表 5.2 基带误码率测试结果

对比帧数	误比特总数	单帧最大误比特数	误码率
1000	0	0	0
2000	0	0	0
5000	0	0	0
5000	1	1	0.000001
5000	0	0	0
5000	0	0	0
10000	0	0	0
10000	0	0	0
10000	0	0	0

可见通过基带连接的系统误码性能较好,最长测量到 10000 帧数据(时长 10 分钟)没有出现误码。以上测试说明 DMR 端机基带传输系统性能良好稳定。

5.2 DMR 端机无线传输测试

将基带板与厂家提供的射频板连接,DSP 完成相关的数字信号处理后,由基带板 MCU 的 D/A 输出数据送往射频板 1 的调频模块完成 4FSK 调制,经过天线发射出去。信号通过无线信道传输后,由射频板 2 的接收天线接收,完成 4FSK 信号的鉴频输出,基带板 MCU 的 A/D 完成数据的采样后,DSP 再完成相关的数字信号处理。在此平台上,进行 DMR 端机无线传输的系列测试。

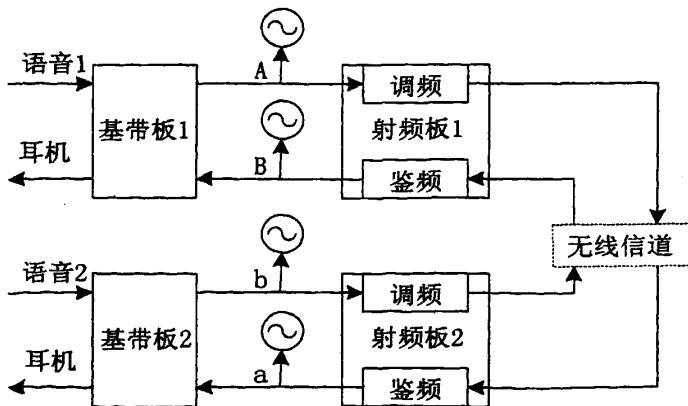


图 5.5 DMR 端机无线传输测试平台

5.2.1 无线单工语音测试

将板 1 作为主呼，板 2 作为被呼。主呼端发送语音，图 5.6 为近距离无线单工语音测试的示波器双通道截图，图中上方对应于通道 1，下方对应于通道 2。通道 1 接图 5.5 中 A 点处，为板 1 的基带发送信号，通道 2 接图 5.5 中 a 点处，为板 2 的接收鉴频信号。图 5.7 为图 5.6 改变示波器时基的局部放大图。

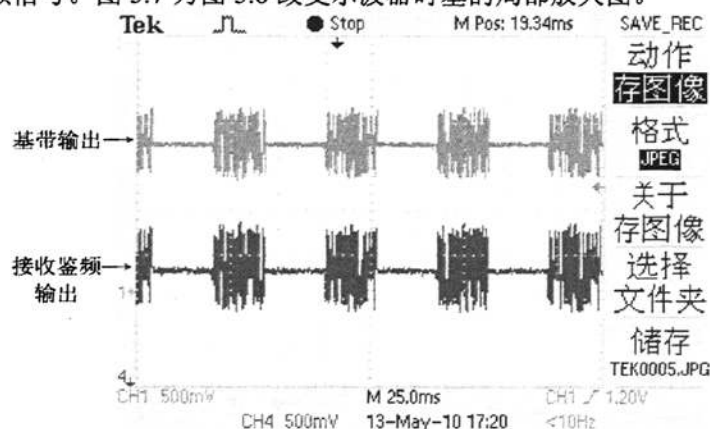


图 5.6 无线单工语音测试图

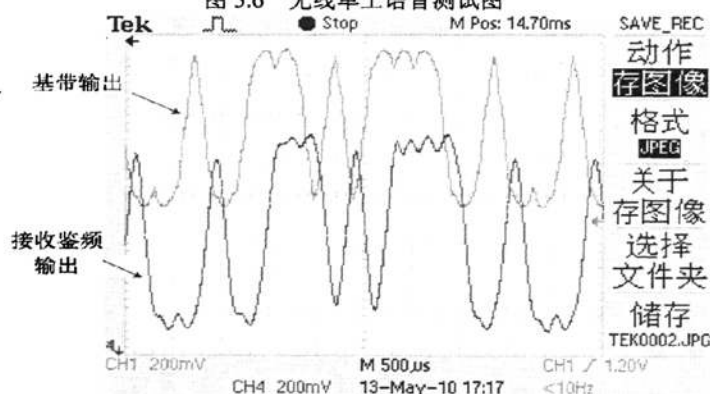


图 5.7 无线单工语音测试局部放大图

图 5.6 中可以看出，主呼被呼的时隙正确建立。图 5.7 可以看出，鉴频输出的信号能较好保持基带发送信号的波形，仅在时间上有一定延时。在被呼端 2 连接耳机，可听到主呼端 1 发过来清晰流畅的语音。将基带板 2 和射频板 2 组成的板 2 作为主呼，板 1 作为被呼，效果一样

5.2.2 无线单工误码性能测试

主呼端发送 264bit 的 DMR 固定帧，被呼端将接收解调的数据与本地存储的固定帧相比较，使用 3.4.1 所述的自适应门限法进行判决，统计误码个数，给出误码性能分析。

表 5.3 无线单工误码率测试结果

测试帧数	误比特总数	单帧最大误比特数	误码率
500	3	1	0.0000227
500	2	1	0.0000152
1000	5	1	0.0000189
1000	9	2	0.0000341
5000	72	2	0.0000545
5000	66	2	0.0000500
5000	45	2	0.0000341
5000	81	2	0.0000614

由表 5.3 可以看出, 基带加射频的无线传输误码率在 10^{-5} 数量级, 说明无线传输的性能较好。

5.2.3 远距离无线传输语音测试

将板 1(包括基带板 1 与射频板 1)作为主呼端发送语音, 板 2 作为被呼端接收语音。板 1 放置在六楼室内, 板 2 从六楼不断移动至一楼, 仍可听到板 1 发过来的清晰流畅语音。说明远距离无线单工传输的性能良好。

5.2.4 无线双工传输测试

在图 5.5 所示的无线传输平台上, 板 1 和板 2 都发送数据, 进行无线双工传输测试。图 5.8 为无线双工测试的四通道示波器截图, 图中自上而下, 第一个波形为板 1 的基带输出 (A 点), 第二个波形为板 2 的鉴频输出 (a 点), 第三个波形为板 2 的基带输出 (b 点), 第四个波形为板 1 的鉴频输出 (B 点)。

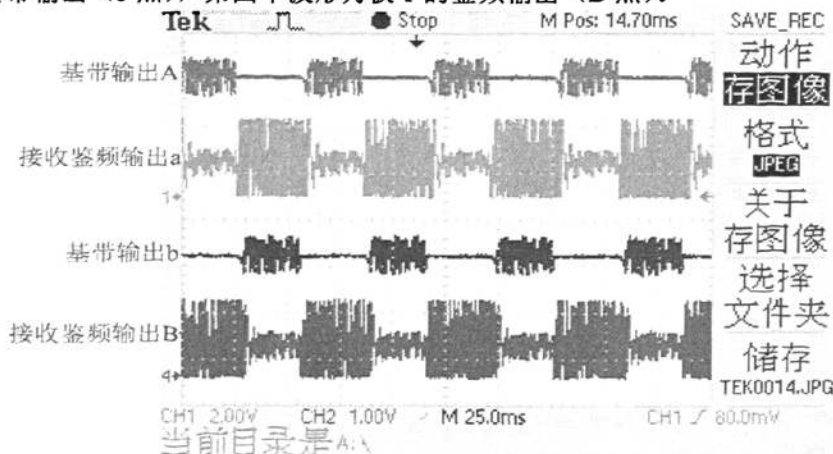


图 5.8 无线传输双工测试图

改变示波器时基, 查看通道一和通道二的放大图, 如图 5.9 所示, 图中上方代表发端基带输出波形 (A 点), 下方代表接收端鉴频输出波形 (a 点)。

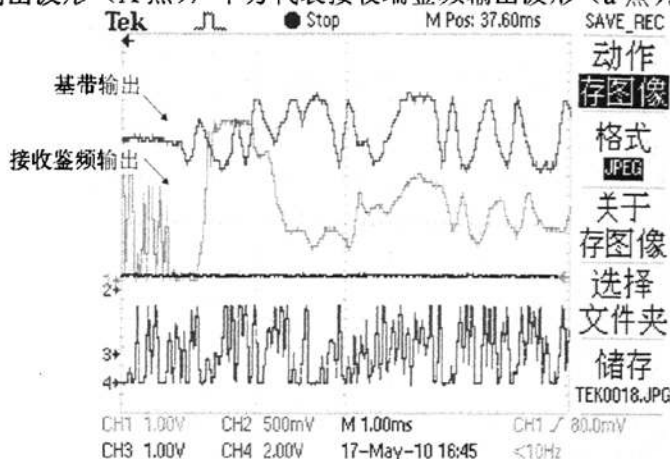


图 5.9 改变时基无线传输双工测试图

由图 5.8 可以看出, 收发两端时隙对准, 且射频能够正确进行收发切换。由图 5.9 看出, 鉴频输出的波形大致保持了基带发送波形, 但在数据开始时有失真。

主呼端发送 264bit 的 DMR 固定帧, 被呼端将接收解调的数据与本地存储的固定帧相比较的方式来统计误码个数, 统计结果如表 5.4 所示。

表 5.4 无线双工误码率测试结果

测试帧数	误比特总数	单帧最大误比特数	误码率
100	2197	26	0.083
100	2084	30	0.079
100	2302	27	0.087
500	10297	33	0.078
500	10268	28	0.078
500	10459	30	0.079
1000	20729	37	0.078
1000	20278	44	0.077
1000	21405	37	0.081

通过表 5.4 可以看出, 无线双工传输测试的误码率在 10^{-2} 数量级。比较表 5.4 和表 5.3 发现, 无线双工误码测试结果与无线单工误码测试结果相差较大。分析原因如下: 该系统为时分双工系统, 因此在射频上需要根据收发时隙来不断的关闭和打开发射机和接收机, 但是经过测试, 射频板的发射机的打开时功率稳定需要 3ms 多, 本系统中在接收数据 29ms 后关闭接收机, 打开发射机, 发射数据有 1.25ms 的数据冗余 (抗干扰用的), 即使这样还会有不到 1ms 的数据会在接收端发生失真。

图 5.10 为无线双工传输时 MCU 接收的一帧数据在 CCS 下显示的波形，由图 5.9 和图 5.10 可以看出，在时隙接收数据的开始会有数据失真。这是引起无线双工误码较大的主要原因。

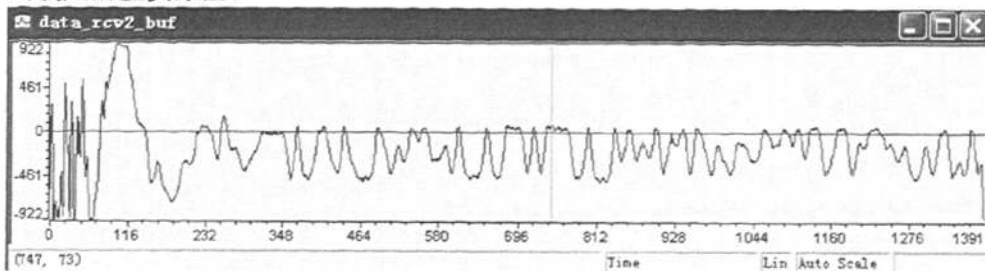


图 5.10 无线双工传输时 DSP 接收数据在 CCS 下观察图

在双工传输的两端连接语音进行语音传输测试，发现两端声音都可以听清，效果没有无线单工语音好，原因主要是误码大。

第六章 结束语

DMR (Digital Mobile Radio) 数字专用无线通信系统做为国际 ETSI 组织正在完善的最新数字专业通信标准, 具有产品成本低、技术简单的优点, 并且支持从模拟到数字的过渡, 因此有巨大的研究开发价值。

本课题来源于西安电子科技大学信息科学研究所与某企业的合作项目: 开发新一代 DMR 数字集群终端, 即基于 DMR 协议的新型数字对讲机。本人主要工作是在 DMR 端机的 DSP 上开发数字基带传输程序, 完成 DSP 整体程序结构调整, 以及在基带板对基带板、基带板加射频板的硬件平台上完成 DSP 程序调试及相关性能测试工作。

现阶段基带单双工语音和误码测试性能很好, 基本能做到无误码传输。无线单工传输误码率在 10^{-5} 数量级, 且语音通话质量良好, 将发送样机放置在六楼室内, 接收样机从六楼不断移动至一楼, 仍可听到发端发过来清晰流畅的语音。无线双工测试中, 由于射频硬件模块的收发切换原因, 测试的误码率在 10^{-2} 数量级, 语音能听清但效果不如无线单工。

针对本项目, 本人提出改进建议和进一步研究的方向:

- (1) 射频模块的硬件需要改进, 应保证收发的正确切换。
- (2) DSP 现阶段程序注重于系统功能的实现, 没有作相关的优化处理, 可以进一步作 DSP 程序优化方面的研究。
- (3) 现阶段语音帧采用每一帧都加同步的方式, 而 DMR 协议规定采用超帧结构, 如何实现采用语音超帧正确实现 DMR 通信是下一阶段需完成的工作。

由于个人能力和经验有限, 工作中难免会出现错误或者这样那样的问题, 恳请各位专家、同仁指正。

致谢

在论文完成之际，回顾以往的点点滴滴，我深切的感受到我所取得的每一步成功都离不开老师和同学的帮助。

首先，深深感谢我的导师杨家玮教授。在几年的硕士学习期间，杨老师一直在我的学业上严格要求、研究工作方向上认真把关。在平常的工作学习中，杨老师给予了认真和耐心的指导。在论文研究工作中，从课题的选题、课题的进展到论文撰写，都倾注了杨老师的心血，他花费了大量的时间和精力，给我们创造了一个适合学生发展的工作和学习环境。杨老师严谨的治学态度、孜孜不倦的钻研精神、敏锐的学术洞察力、平易近人的品格和要求我们不断创新的作风给我留下了深刻的印象，是我永远学习的楷模。在此感谢杨老师并希望他身体健康，合家欢乐，桃李满天下！

同时，衷心感谢丁婧、杨懋、李伟、程万灵、罗继芳、谭谨博等师兄师姐一直以来对我的帮助，有前期的基础才有现在的成果，我取得的每一次进步都离不开你们的耐心指导。

感谢同项目组的王保坤、丁春丽、李梦、李远亮对我的信任和支持，与你们共同探讨、共同学习、共同进步的时光是我一生的宝贵财富。

感谢其他同室的高玉霞、赵媛媛、吴环宇、范勇、赵允平、陈宏亮、陈磊、周俊、曾环宇、杨迪、张惠娟、田琳、马瑞、赵侃，与你们共同的学习和工作非常愉快，这段日子令人难忘！

感谢西安电子科技大学信息科学研究所的李建东老师、李维英老师、陈彦辉老师、刘勤老师、张文柱老师、黄鹏宇老师、冯军老师、盛敏老师、赵林靖老师、安凤英老师！感谢他们为我提供良好的学习环境和试验条件及平时给予的热心指导和帮助。

深深感谢我的家人对我学习的鼓励以及生活的照顾，你们的支持是我不断前进的动力！

感谢所有关心和帮助过我的人。

最后，对所有参加论文评审和对本文提出宝贵意见的各位专家、教授以及老师表示衷心的感谢！

参考文献

- [1] 席永利, 全球数字集群市场发展概况, Telecommunication Technology, 2005 年第 8 期, p4~p6
- [2] ETSI TS 102 361-3: "Electromagnetic compatibility and Radio spectrum Matters (ERM); Digital Mobile Radio (DMR) Systems; Part 3: DMR Data protocol".
- [3] ETSI TS 102 361-1: "Electromagnetic compatibility and Radio spectrum Matters (ERM); Digital Mobile Radio (DMR) Systems; Part 1: DMR Air Interface (AI) protocol".
- [4] 孙丽华, 谢仲华, 陈荣伶等, 《信息论与纠错编码》, 北京: 电子工业出版社, 2005. p153-p154
- [5] 啜钢, 王文博, 常永宇, 《移动通信原理与系统》, 北京: 北京邮电大学出版社, 2005. p108~p110
- [6] (英)Alister Burr, 《Modulation and Coding for Wireless Communications》, 北京: 电子工业出版社, 2003. p27~p35
- [7] 张辉, 曹丽娜, 《现代通信原理与技术》, 西安: 西安电子科技大学出版社, 2002. p116~p117, p100~p104, p217~p219
- [8] 凌云志, 基于DSP实现根升余弦滤波器的研究, 国外电子测量技术, 2006.1, Vol.25, No.1, p8~p9
- [9] 丁玉美, 高西全, 《数字信号处理》第 2 版, 西安: 西安电子科技大学出版社, 2000. p195~p201
- [10] (英)/贝特曼 (Bateman, A)等著, 陈健等译, 《DSP 算法、应用与设计》, 北京: 机械工业出版社, 2003. p264~p269
- [11] (美)Rodger E. Ziemer著, 袁东风, 江铭炎译, 《通信原理—系统、调制与噪声》第5版, 高等教育出版社, p133~p135
- [12] 陈亮, 杨吉斌, 张雄伟, 《信号处理算法的实时 DSP 实现》, 北京: 电子工业出版社, 2008. p261~p263, p19~p30
- [13] Kamilo Feher, "Modulation/Microwave Integrated Digital Wireless Developments", IEEE Transactions on Microwave Theory and Techniques, 1995, VOL. 43, No.7
- [14] 申敏, 邓矣兵, 郑建宏 等, 《DSP 原理及其在移动通信中的应用》第 1 版, 北京: 人民邮电出版社, 2001. p185, p124~p126, p127~p129, p140, p37
- [15] 李建新, 刘乃安, 《现代通信系统分析与仿真: MATLAB通信工具箱》, 西安:

- 西安电子科技大学出版社, 2000. p105~p110, p123~p124
- [16] 严城, 基带脉冲成形数字滤波器的MATLAB设计与DSP上的实现, 电子科技, 2006.9, p14~p16
- [17] 李建东, 陈家模, 平滑四电平调频的实现及其性能分析, 通信学报, 1990.1, p84~p85
- [18] 王锋, 廉保旺, 王永生, 一种高斯预滤波函数的DSP实现, 无线电通信技术, 2003, Vol.29, No.5, p28~p29
- [19] 谭浩强, 《C程序设计》第2版, 北京: 清华大学出版社, 1999. p124
- [20] 戴逸民, 梁晓雯, 裴小平, 《基于 DSP 的现代电子系统设计》, 北京: 电子工业出版社, 2002. p181~p183
- [21] 邬可军, 朱铭镭, 《DSP实时多任务操作系统设计与实现》, 北京: 电子工业出版社, 2005. p220, p248
- [22] 杨懋, 杨家玮, 程万灵, DMR 通信协议与数字对讲机基带模块设计, 单片机及嵌入式系统应用, 2008 年第 7 期
- [23] 丁春丽, DMR系统差错控制技术的研究, 西安电子科技大学硕士论文, 2009, p57
- [24] Texas Instruments Data Manual, TMS320VC5510/5510A Fixed-Point Digital Signal Processors, 2006, Literature Number: SPRS076N
- [25] 胡庆钟, 李小刚, 吴钰淳 等, 《TMS320C55x DSP原理、应用和设计》第1版, 北京: 机械工业出版社, 2006. p42, p76, p109