

## ABSTRACT

Distribution Network Reconfiguration (DNR) is a hot topic in the research on distribution network automation system which can improve the network running structure, reduce active loss, and increase network reliability by changing the combination states of section switches and contact switches. Consequently, it is meaningful to design a DNR experiment device on which we can practise the DNR algorithm research and simulate the whole process of reconfiguration.

The test bench is composed of four function modules: distribution network circuit simulation module, embedded system module, data acquisition module, and reconfiguration execution module. Among these modules the embedded system module, which is composed of S3C2410X microprocessor based on ARM920T core and Linux embedded operating system, is the center control module. Its main function is to receive the parameters from data acquisition module, run the DNR algorithm, and then output its operation result. The distribution network circuit simulation module is a low voltage distribution network based on IEEE-RBTS Bus2 and Bus 4. The data acquisition module centers about a STC MCU, and its main function is to get the parameters from the distribution network circuit simulation module and then send these parameters to the embedded system module. The reconfiguration execution module chooses a ALTERA FPGA as its processor to implement the order of reconfiguration. Then this paper introduces and analyzes the four function modules about their principle, function, the keys and difficulties about their design, and also develops a DNR algorithm and successfully realizes it on this test bench.

Finally, this paper shows the results of the experiment and the points to be improved and developed.

**KEY WORDS:** Distribution Network Reconfiguration, Embedded System, ARM, Linux

## 原创性声明

本人声明，所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了论文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中南大学或其他单位的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献均已在论文中作了明确的说明。

作者签名: 董怡 日期: 2008年5月28日

## 学位论文版权使用授权书

本人了解中南大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并根据国家或湖南省有关部门规定送交学位论文，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用复印、缩印或其它手段保存学位论文。同时授权中国科学技术信息研究所将本学位论文收录到《中国学位论文全文数据库》，并通过网络向社会公众提供信息服务。

作者签名: 董怡 导师签名: 王东 日期: 2008年5月28日

# 第一章 绪论

## 1.1 课题研究背景

随着我国工业的蓬勃发展以及随之而来的电器产品越来越多地走入并影响人们的生活，电力在人们的生活中扮演了更加重要的角色。那么，如何以最经济的方式为用户连续提供高质量的电能呢？电力系统自动化作为改善供电质量以及提高供电可靠性的一种主要途径得到了广大电力人的关注<sup>[1]</sup>。

由于电力系统的复杂性，通常将电力系统分为发电系统、输电系统和配电系统三个子系统进行分析和研究。其中配电系统是电力系统中直接与用户相连的部分，它的可靠性及安全性直接影响用户用电的可靠性及安全性<sup>[2]</sup>。多供电途径的网格状配电网的运行方式非常灵活，仅凭借值班员的经验来调度和管理配电网将越来越困难，迫切需要发展智能配电软件来协助操作员甚至代替操作员更科学、更及时地做出最佳的调度决策。

配电网重构<sup>[3]</sup>（Distribution Network Reconfiguration）就是在保证配网呈辐射状、满足馈线热容、电压降落要求和变压器容量等的前提下，改变分段开关、联络开关的组合状态，使配网某些指标（如：配电网线损、负荷均衡或供电电压质量等）最佳的配网运行方式。

### 1.1.1 配电网重构研究的目的和意义

配电网重构的研究起源于 80 年代后期，因其在降低配电网网损和改善系统的安全方面的重要作用而受到不少学者的关注<sup>[4-10]</sup>。早期的配电网重构主要是研究通过怎样的供电路径给新用户供电可以使总的费用最小，即研究配网规划阶段的配电网重构问题。随着对配电网重构认识的逐步加深，学者们开始研究配电自动化系统中加入网络重构的可行性，研究结果表明配电网重构不仅在经济和技术上可行，而且可以极大地优化配电系统的运行。目前，国内外学者所作的配电网重构研究大多集中于降低线损<sup>[11-13]</sup>、负荷均衡化<sup>[14]</sup>及最佳恢复受故障影响的健全区域供电<sup>[15]</sup>等方面。

配电网重构的意义主要体现在以下三个方面<sup>[16]</sup>：

#### 1、均衡负荷，消除过载，提高供电电压质量

在配网中，每条馈线均有不同类型的负荷，如商业类、民用类和工业类。由于不同类型负荷的日负荷曲线是不同的，在变电所的变压器及每条馈线上峰值负荷出现的时间也是不同的，导致实际当中配电网中的负荷分布是不均匀的、有时甚至是极不均衡，这严重降低了配电线路和设备的利用率，同时也增加了系统的线损。通过配电网重构，可以将负荷从重负载或是过负载馈线(或变压器)转移

到轻负载馈线(或变压器)上,这种转移不仅调节了运行馈线的负荷水平,消除馈线过载,还能改善电压质量。

## 2、降低配电网线路损耗,提高系统经济性

降低电力系统的线损,提高电力系统运行经济效益,是电力系统面临的一项长期课题。在电能的输送和分配过程中,电网中的变压器线路元件都要消耗一定的电能,配电网的网损占了整个电网网损相当大的比例。据统计,西方主要工业国家的线损率大致在5%-8%,我国为9%左右,与发达国家相比尚有差距。

而35-110KV配网线损是地区线损的重要组成部分,降低配网线损是降损工作关键问题之一。目前,可以通过多种途径来降低配电网的线损,如配电网网络重构、安装补偿电容器、提高配电网的电压等级和更换导线的措施。但对于一个给定网络结构的配电网,通过配电网网络重构可以在不需要额外投资的情况下,只是改变电网联络开关和分段开关的状态从而改变配电网的运行方式,就可以达到降低配电网网络损耗的目的。

## 3、提高供电可靠性

提高供电可靠性有两层含义,一是在配电系统发生故障时,通过分段开关运行状态的重构,把故障线路上的部分或全部负荷转移到其他线路上去,从而达到快速隔离故障和恢复供电、尽量减少用户停电时间的目的。二是在系统已经建成,各元件的可靠性和系统的结构都不易改变的情况下,通过网络重构的方法,改变用户的供电路径,使重要用户和多数用户从更为可靠的路径获得供电,提高整个配电系统的可靠性。

### 1.1.2 配电网网络重构工程实用化的条件

实现配电网网络重构工程实用化必须满足下列要求<sup>[17]</sup>:

#### 1、算法上的要求

要求算法可靠、稳定,并且计算速度快。

#### 2、硬件设施的要求

配电网的硬件设施是重构实用化的物质基础。配电网建立起了SCADA系统,具备了配电调度中心、通道和执行端装置,才能对配电网中的开关设备进行远动操作。当这些条件还不能满足时,可以只考虑远动开关,通过这些远动开关实现网络的部分重构;或者实时操作无法实现,也可先实现季节性重构和节假日前后重构,这种重构方式不要求必须是远动开关。坚持下来,获得的经济效益同样是非常可观的。

#### 3、配电网负荷数据的要求

实现配电网的实时优化控制,需要系统能获得各个负荷点在任一时刻的负荷数据。

#### 4、软件技术的要求

重构软件如何合理的加入已有的配电管理系统中,或者如何对已有的重构软件进行升级,成为重构能否实现工程实用化的另一个关键因素。

##### 1.1.3 配电网重构启动的条件

配电网重构的启动需要考虑以下几个方面的因素<sup>[17]</sup>:

1、综合考虑历史、现在和未来的运行情况。配电网重构控制不能仅依赖某个时间断面的数据,而应当综合考虑历史、现在和预测的未来的负荷情况,并对它们进行加权处理。

2、设置配电网重构控制效果阈值。设置降损效果阈值,当重构后的降损效果超过该阈值时,才考虑配电网重构控制的启动。

3、设置配电网的节点电压阈值。计算出重构后各个节点的电压,若均在最高节点电压阈值  $U_{\max-set}$  和最低节点电压阈值  $U_{\min-set}$  之间,才考虑配电网重构控制的启动。

4、网络重构的最终执行,一般要经过调度员的确认。

## 1.2 嵌入式系统及其应用现状

### 1.2.1 现代计算机技术发展

关于嵌入式技术在现代计算机技术的发展历程中,因其与通用计算机系统不同的应用特点所迈上的两种不同发展道路,探矿工作室在其著名的《嵌入式系统开发圣经》一书中有如下描述<sup>[18]</sup>:

实用化的数字计算机诞生于 1946 年,在其后漫长的历史进程中,计算机始终是供养在特殊的机房中实现数值计算的大型昂贵设备。直到 20 世纪 70 年代,微处理器的出现,计算机才出现了历史性的变化。以微处理器为核心的微型计算机以其小型、价廉、高可靠性等特点迅速走出机房。

基于高速数值计算能力的微型机表现出的智能化水平引起了控制专业人士的兴趣,要求将微型机嵌入到一个对象体系中,实现对象体系的智能化控制。例如将微型计算机经电气加固、机械加固并配置各种外围接口电路,安装到大型舰船中构成自动驾驶仪或轮机状态监测系统,这样一来计算机便失去了原来的形态与通用的计算机功能。为了区别于原有的通用计算机系统,把嵌入到对象体系中实现对象体系智能化控制的计算机称作嵌入式计算机系统。

嵌入式系统的嵌入性本质是将一个计算机嵌入到一个对象体系中去,这些是理解嵌入式系统的基本出发点。由于嵌入式计算机系统要嵌入到对象体系中,实现的是对对象的智能化控制。因此,它有着与通用计算机系统完全不同的技术要求与技术发展方向。

通用计算机系统的技术要求是高速、海量的数值计算技术,发展方向是总线

速度的无限提升,存储容量的无限扩大;而嵌入式计算机系统的技术要求则是对对象的智能化控制能力,技术发展方向是与对象系统密切相关的嵌入性能控制能力与控制的可靠性。

早期,人们勉为其难地将通用计算机系统进行改装,在大型设备中实现嵌入式应用。然而,对于众多的对象系统(如家用电器、仪器仪表、工控单元……),无法嵌入通用计算机系统,况且嵌入式系统与通用计算机系统的技术发展方向完全不同。因此,必须独立地发展通用计算机系统与嵌入式计算机系统,这就形成了现代计算机技术发展的两大分支。

如果说微型机的出现,使计算机进入到现代计算机发展阶段,那么嵌入式计算机系统的诞生,则标志了计算机进入了通用计算机系统与嵌入式计算机系统两大分支并行发展时代,从而共同促进了20世纪末计算机行业的高速发展。

通用计算机系统与嵌入式计算机系统的专业化分工发展,计算机专业领域集中精力发展通用计算机系统的软硬件技术,不必兼顾嵌入式应用要求。通用微处理器迅速从286、386、486到奔腾系列;操作系统则迅速扩张计算机基于高速海量的数据文件处理能力,使通用计算机系统进入到尽善尽美阶段。

嵌入式计算机系统则走上了一条完全不同的道路,这条独立发展的道路就是单芯片化道路,它动员了原有的传统电子系统领域的厂家与专业人士,接过起源于计算机领域的嵌入式系统,承担起发展与普及嵌入式系统的历史任务,迅速地将传统的电子系统发展到智能化的现代电子系统时代。

因此,现代计算机技术发展的两大分支的里程碑意义在于它不仅形成了计算机发展的专业化分工,而且将发展计算机技术的任务扩展到传统的电子系统领域,使计算机成为进入人类社会全面智能化时代的有力工具。

### 1.2.2 嵌入式微处理器 ARM

ARM<sup>[19-21]</sup>(Advanced RISC Machine)既可以认为是一个公司的名字,也可以认为是对一类微处理器的通称,还可以认为是一种技术的名字。1991年ARM公司成立于英国剑桥,主要出售芯片设计技术的授权。目前,采用ARM技术知识产权IP核的微处理器,即通常所说的ARM微处理器,已遍及工业控制、消费类电子产品、通信系统、网络系统、无线系统等各类产品市场,基于ARM技术的微处理器应用约占据了32位RISC微处理器75%以上的市场份额,ARM技术正在逐步渗入到我们生活的各个方面。

ARM公司是专门从事基于RISC技术芯片设计开发的公司,作为知识产权供应商,本身不直接从事芯片生产,靠转让设计许可由合作公司生产各具特色的芯片,世界各大半导体生产商从ARM公司购买其设计的ARM微处理器核,根据各自不同的应用领域,加入适当的外围电路,从而形成自己的ARM微处理器芯片。

进入市场。

目前，全世界有几十家著名的半导体公司都使用ARM公司的授权，其中包括摩托罗拉、IBM、Intel、ATMEL、SONY、NEC、三星、LG等。

ARM微处理器核技术广泛用于便携式通信产品、手持运算、多媒体和嵌入式解决方案等领域，已成为RISC标准。ARM处理器核是系统中的引擎，它从存储器读取ARM或Thumb指令并执行这些指令。

目前基于ARM核的处理器主要有以下几类：ARM7、ARM9、ARM9E、ARM10E、ARM11、SecurCore、MPCore、StrongARM/XScale等。每一个系列的ARM微处理器都有各自的特点和应用领域。其中，ARM7、ARM9、ARM9E、ARM10E以及ARM11为通用处理器系列，每一个系列提供一套相对独特的性能来满足不同应用领域的需求。SecurCore系列专门为安全要求较高的应用而设计。StrongARM和XScale则是Intel公司基于ARM体系的解决方案。

以下来了解一下将要使用的ARM9系列处理器：

ARM9系列微处理器在高性能和低功耗特性方面提供最佳的性能。具有以下特点：

- 1、5级整数流水线，指令执行效率更高。
- 2、提供1.1MIPS/MHz的哈佛结构。
- 3、支持32位ARM指令集和16位Thumb指令集。
- 4、支持32位的高速AMBA总线接口。
- 5、全性能的MMU，支持Windows CE、Linux、Palm OS等多种主流嵌入式操作系统。
- 6、MPU支持实时操作系统。
- 7、支持数据Cache和指令Cache，具有更高的指令和数据处理能力。

### 1.2.3 常用的嵌入式操作系统

#### 1、Linux<sup>[22-26]</sup>

在所有的操作系统中，Linux是发展最快、应用最广泛的。Linux本身的种种特性使其成为嵌入式开发的首选。在进入市场的前两年中，嵌入式Linux的设计通过广泛应用而取得巨大成功。随着嵌入式Linux技术的成熟，定制需要的尺寸尤为方便，同时支持更多硬件平台，并从早期的试用阶段迈进到成为嵌入式市场的主流。

Linux发展如此之快的主要原因是产品的成本。在激烈的市场竞争中，只拥有先进的技术远远不够，如何减少产品的投入也是需要重点考虑的问题。免费的

Linux无疑为厂商节约了一大笔开支。

此外，嵌入式Linux协会（ELC）在2002年底发布了ELC产品规范版本1.0（ELCPS），它可提供的好处包括：

- (1) 嵌入式Linux发布可以很清楚地显示自身的功能特性；
- (2) 在很少或没有额外接口的条件下，第三方类库和中间程序就可以运行不同的嵌入式Linux配置；
- (3) 软件开发人员可以将程序段写入一个现成的API中，并相信这一代码段可为其他的嵌入式Linux环境提供接口。

另一种常用的嵌入式Linux是μCLinux，μCLinux是一个符合GNU/GPL公约的开放源代码项目，是由LINEO公司在Linux 2.0的基础上裁减得到的。在μCLinux这个英文单词中，μ表示Micro，即小的意思；C表示Control，即控制。所以μCLinux就是Micro-Control-Linux，字面上的理解就是“针对微控制领域而设计的Linux系统”。μCLinux去掉了Linux中的内存管理单元，主要用于没有内存管理单元的处理器（比如三星公司的S3C44B0X处理器），该系统直接访问内存，所有程序中访问的地址都是物理地址。

## 2、μC/OS-II<sup>[27-29]</sup>

在嵌入式操作系统领域，由Jean J.Labrosse开发的μC/OS，由于开放源代码和强大而稳定的功能，曾经一度在嵌入式系统领域引起强烈反应。μC/OS-II是μC/OS的第二版，目前已经应用在很多领域，比如医疗设备、引擎控制、网络适配器、ATM机和工业机器人等。

μC/OS-II最大的特点就是它的源代码开放，这是其它商业实时内核无法比拟的。它是针对嵌入式应用而设计的，而且在设计之初就充分考虑到了它的可移植性，它的大部分源代码都是用高可移植性的ANSI C编写的。μC/OS-II可以移植到很多的处理器上，它能运行在大部分的8位，16位，32位，甚至64位的微处理器和DSP上。

μC/OS-II是一个完全抢占式的内核，它总是运行最高优先级的就绪任务，但不支持轮转法任务调度。每个任务被赋予唯一的优先级，使用自己独立的堆栈。μC/OS-II提供了许多系统服务，如邮箱机制、队列机制和信号量机制、固定大小的内存分区以及时间相关的函数等等。

## 3、VxWorks

VxWorks是WindRiver公司专门为实时嵌入式系统设计开发的操作系统软件，它为程序员提供了高效的实时任务调度、中断管理、实时的系统资源以及实时的任务间通信。

VxWorks自1983年问世以来，已成功应用在航天、航空、舰船、通信、医疗

等关键领域。不久前引人关注的美国“勇气”号火星探测器就采用了VxWorks操作系统。目前，VxWorks已经成为事实上的工业标准和军用标准，大量软硬件厂家都提供VxWorks的扩展组件。

因此，VxWorks在各种CPU硬件平台上可以提供统一的接口和一致的运行特征，应用程序不用任何改动就可以运行在各种CPU上，为程序员提供了一致的开发和运行环境，减少了重复的劳动。

#### 4、WinCE<sup>[30]</sup>

WinCE(Windows CE)是Microsoft公司1997年发布的应用在PDA等设备上的嵌入式操作系统。1998年，装有中文版WinCE操作系统的手持设备面世。这几年中，WinCE在PDA操作系统市场中份额逐年上升。

WinCE是为各种嵌入式系统和产品设计的一种紧密的、高效的、可升级的操作系统。它是从整体上为有限资源的平台设计的多线程、完整优先权、多任务的操作系统。WinCE采用模块化设计，初始设备开发商(OEM)或者嵌入式操作系统开发商可以从适合目标平台的操作系统中自行选择系统模块和组件。选择的系统模块和组件的多少决定了所需内存的大小。其操作系统的基本内核至少需要200KB的ROM。

WinCE作为一个高度模块化并且具有Internet服务、强大的实时性支持、广泛的通信功能的平台，使得开发人员能获得更大的灵活性，让他们可以充分发挥自己的想象力，构建高度定制化的设备来满足用户的需求。

目前为止，WinCE可以作为许多嵌入式系统的平台，应用于包括工业控制器、机器人、网络集线器、办公室设备、零售设备、摄像机、电话、机顶盒、Internet访问设备和家庭娱乐设备等方面。

#### 5、Palm OS

Palm OS是由Palm公司开发、应用在PDA上的操作系统。自最早版本1.0在1996年3月随Palm公司的PDA产品Pilot 1000面市以来，Palm OS不断推出新版本，在PDA操作系统上市场占有率位居前列。

Palm OS 提供可用性很强的 API，使得任何人都可以开发基于 Palm OS 的应用程序，在 Internet 上也发布了很多免费的应用程序，丰富的可用资源是 Palm OS 风靡全球的一个重要原因。它运行在一个抢占式的多任务内核之上，同一时刻用户界面仅仅允许一个应用程序被打开，与同步软件 HotSync 结合可以使 PDA 与 PC 上的信息实现同步，把 PC 的功能扩展到掌上电脑上。同其他嵌入式操作系统相比，Palm 具有更大的灵活性和移动性。

#### 6、Sysmbian

Sysmbian 成立于 1998 年 6 月，其在移动计算设备和手机软件技术开发领域

拥有近二十年的丰富经验,是一家由各主要无线通信技术公司持股的软件许可证授权公司,为支持数据分组传输的手机提供高级、开放式、标准化的操作系统Sysmbian OS<sup>TM</sup>。

### 1.3 研制配电网络重构实验装置的意义

由 1.1.1 节对于配网重构意义的论述可以知道,配电网络重构对于提高配电网络运行的可靠性、经济性以及供电质量的改善都具有十分重要的作用。但是,对网络拓扑结构的改变同时也伴随着一定的风险,如节点负荷转移对母线的影响,开关不可靠工作对节点供电的影响等,都可能引发供电事故。因此,配网重构运行中系统工作的可靠性就尤其重要。

目前,目前广大院校对于配电网络重构的研究多集中在优化算法层面,基于各种提高供电质量或可靠性指标的目标函数对配电网络拓扑结构进行最快、最优的求解过程。这部分研究对于 1.1.2 节提出的配电网络重构工程实用化条件中第一点中算法所要求的可靠、稳定及快速性的提高具有非常重要的意义。除算法外,重构中各相关设备本身工作的可靠性也十分重要。现在国内外各大公司也都展开了关于配电网络重构装置的研制,但多数由于在可靠性方面的缺陷使得相关方面的产品尚无法真正得到应用。

因此,研制这么一套配电网络重构实验装置,对于配电网络重构的研究具有十分积极的意义:

1、为配电网络重构算法提供一套可供实验的平台。相比于在计算机上仿真,本装置可以让学生们更清晰地了解配电网络重构的工作原理及工作过程;

2、为配电网络重构走向真正应用打下了基础。本装置完全采用模块化设计,为今后各模块功能的升级提供了方便,也使其他功能模块的加入变得简单可行。正如 1.1.2 节第二点所述,在各种自动化装置尚不健全的的状况下,根据历史数据实现季节性重构和节假日前后重构,也能产生可观的效果;

3、为配电网络重构从实验室研究走向真正应用铺陈了道路。

### 1.4 本文的主要工作

1、深入研究配电网结构和原理,详细解析了配电网络重构技术在走向应用过程中所需开展的研究工作以及注意事项。

2、深入学习研究了 UP-NetARM2410-S 实验仪器平台,包括其核心处理器 S3C2410X 的各项特性、资源及其应用;以及嵌入式 Linux 操作系统的开发环境、开发工具、其驱动程序开发和应用程序的开发等。

3、设计开发了以 STC 单片机 STC12C5410AD 为核心、外加模拟开关组以及锁相分频电路构成的数据采集模块,同时结合傅立叶算法实现其对多路模拟电

压输入信号电压峰值、相位以及频率的采集，并将这些数据通过 RS232 上传至嵌入式系统模块。

4、设计开发了以 ALTERA 公司 CYCLONE III 系列 FPGA EP3C25E144 为核心的重构执行模块，系统程序以 Verilog HDL 硬件描述语言设计实现，其主要内容包括 RS232 串口通信程序及重构指令的解析程序等。

5、研究开发了基于克隆遗传算法的配电网络重构算法，并结合所开发的实验装置进行联合实验，证实了该实验装置的有效性及可靠性。

## 第二章 配网重构实验装置总体设计

### 2.1 系统方案论证

配电网络重构实验装置主要用于实验模拟配电网络重构算法，同时也可以实验模拟配电网络重构的整个过程。

模拟配电网络重构算法时，除了算法上的准备，硬件上主要是提供一个可供算法运行的平台以及对运算结果的显示。而在模拟整个配电网络重构过程时，其主要在于模拟配电网络与中央处理核心的互动，即中央处理核心对模拟配电网络参数的采集以及根据这些参数得出的对模拟配电网络拓扑重构结果的反馈。可见，在实现两种功能的过程中，对装置配置要求的着重点也有所不同。

那么，可不可以搭建一个系统，能够同时很好地满足上述两种实验要求呢？我们为此制定了以下两种实现方案：

方案一：以一款强大的处理器为核心，为其搭建庞大的模拟配电网络，并为其配置各节点的电压采集电路，各分段开关远程控制接口等，然后围绕处理器还需要有电压参数采集的相关电路、远程开关控制的相关电路、LCD 显示模块用于显示重构结果参数等。

该方案的优点在于只用一款控制器实现对整个系统的控制功能，系统看起来简单明了，同时实现成本较低。但是该方案的缺点也相当明显：其一，强弱电不分家，核心控制部分与现场不分家，使得系统的工作环境相当恶劣；其二，整个系统将众多复杂的控制功能集中于一块处理器，软件实现难度大；其三，系统升级不便，甚至需要将整个系统推倒重建。

方案二：采用模块化设计理念，除了选择以一款功能强大的处理器用于重构算法的计算这一核心功能模块以外，还将与模拟配电网络相关的控制部分分别设计为数据采集模块和重构命令执行模块，通过这两个模块，将核心功能模块从现场复杂的电磁环境中解放出来。同时，还可设计一个以模拟配电网络中开关网络为模板的 LED 显示模块，以 LED 的亮灭来显示网络中开关的开合，这样，只有在模拟演示配电网络重构过程时，才需要用到结构庞大的模拟配电网络，在进行算法相关研究时，通过 LED 显示模块与核心功能模块所带的 LCD 屏，重构结果及参数就可以一目了然地显示出来。

相对于方案一，该系统的优点在于：其一，系统结构灵活，可以是用于实验算法时由包括 LED 模块在内的几个模块构成的“小”平台，也可以是演示重构过程时由包括模拟配电网络在内的“大”装置；其二，分不同功能块实现，结构清晰，方便系统问题的查找；第三，系统维护、升级方便。其主要缺点在于设计

复杂以及价格相对较高。

通过对上述两种方案的比较讨论,本文最终选用了第二种方案来实现该配电网重构装置。同时,根据不同的实验要求设计了不同等级的两个系统,其所包含的功能模块稍有区别:第一等级是为了实验模拟配网重构实验算法的实验装置,其主要构成包括嵌入式系统模块、重构执行模块以及 LED 显示模块,嵌入式系统模块运行重构算法,LED 显示模块显示重构结果;第二等级则能实验模拟在实际应用中配电网重构的完整过程,其主要由数据采集模块、嵌入式系统模块、重构执行模块以及模拟配电网模块构成。

其中,嵌入式系统模块选用的是 ARM+Linux 的嵌入式平台,相比于计算功能强大的 DSP 芯片,ARM+Linux 嵌入式平台的优点在于其丰富的软件支持,更好的通用性以及可移植性。因此,计算能力上的小小损失相对来说是可以接受的。

数据采集模块的核心选用的是内置 10 位高速 A/D 通道的 STC12C5410AD,其优点不仅在于超高的性价比以及便捷的 IAP 开发环境,更在于其超强的抗干扰能力和保密性能。尤其是其抗干扰能力,在配电网现场复杂的电磁环境下显得尤其重要。此外,模块还配置了应用傅立叶算法对节点电流的峰值、相位以及频率进行计算采集所需的相关电路。

重构执行模块的核心是一块 ALTERA 的 CYCLONE III 系列 FPGA,在这里选用 FPGA 来接受嵌入式系统模块的重构结果,实现计算结果的输出,是基于模块化设计的需求。通过将该部分功能模块化,就可以在系统中其他模块升级后继续实现无缝连接以及正常工作。而这里之所以选择了一款性能超出系统要求不少的芯片,另外一个重要因素在于我们将该款性价比很高的 CYCLONE 最新一代 FPGA 设计成最小开发板模式,通过该最小开发板搭配外围电路实现模块功能,同时这个最小开发板还可以提供一些其他的 FPGA 应用实验。

与其他模块设计时考量的标准不同,模拟配电网模块电路模型的选取是对配电网重构算法适应度和可靠度进行检验时的重要因素之一,因此模拟配电网模型的权威度就显得相当重要。本设计参考了 IEEE 配电网可靠性测试系统 (IEEE Reliability Test System)<sup>[31-32]</sup>,并选取其中等同于配电网拓扑结构的 Bus2 和 Bus4 为模板,构建本次的模拟配电网电路。其不仅具有相当的权威度,也很好地契合了本项目主要基于可靠性来进行配网重构的目标。

## 2.2 系统总体结构

根据上面的方案论证,系统根据实验要求被设计成两个等级:

第一等级系统没有模拟配电网模块,也不考虑数据的采集部分,网络的起始潮流参数根据其标准网络参数自行生成,经过嵌入式系统模块中的重构算法,计算得出理想的重构模式,然后输出至 FPGA 重构执行模块执行。执行结果也不

以改变开关状态来展示，而是增加了一块同样基于 IEEE-RBTS Bus2 和 Bus 4 模型的 LED 显示模块，以 LED 的亮灭来表示模拟配电网中开关的开合情况，使得算法的执行时间及执行结果清晰明了。其所对应的结构框图如图 2-1 所示。

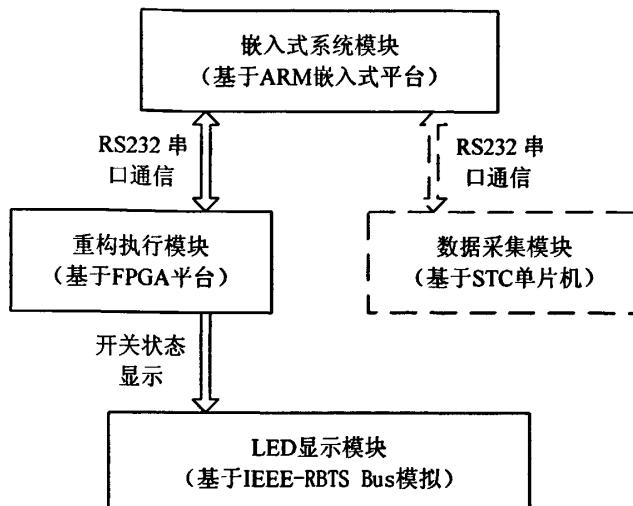


图 2-1 第一等级系统总体结构框图

第二等级系统中，主要包括四个功能模块：模拟配电网模块、嵌入式系统模块、数据采集模块以及重构执行模块。首先由以 STC 单片机为核心的数据采集模块对模拟配电网模块中各节点的潮流数据进行采集，并将这些初始潮流参数通过 RS232 串口发送到嵌入式系统模块保存，待系统中所有节点数据采集并发送过一遍后，嵌入式系统模块运行相应的配电网智能重构算法，再根据运算结果将重构结果指令通过 RS232 串口传送至 FPGA 重构执行模块，重构执行模块执行相关的开关状态重组指令，实现模拟配电网结构的重构实验。其各模块对应的结构关系如图 2-2 所示。

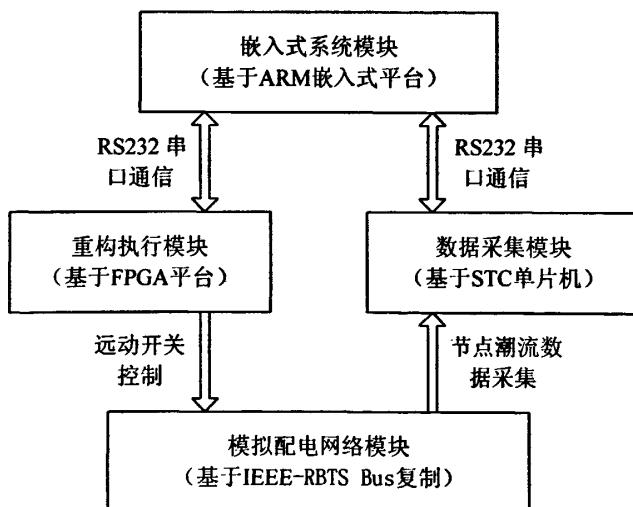


图 2-2 第二等级系统总体结构框图

## 2.3 系统各功能模块设计

### 2.3.1 嵌入式系统模块

嵌入式系统模块是整个配电网重构实验装置的大脑,是接收模拟配电网潮流参数、执行重构算法、发布重构指令的核心中枢。

为了缩减整个系统的风险和投入,现阶段,嵌入式系统模块以博创公司开发的 UP-NETARM2410-S 实验箱为平台,搭载嵌入式 Linux 组成一个 ARM+Linux 的嵌入式系统。UP-NetARM2410-S 是北京博创公司基于三星 S3C2410X 研制的一款多功能仪器,集教学、科研等众多功能于一身,是目前国内软、硬件配置非常完善的嵌入式开发平台,国内 70%以上的高校嵌入式实验室用户均选择采用这款平台。

### 2.3.2 数据采集模块

数据采集模块主要功能是实现对模拟配电网中多达 43 个节点的电流互感器数据进行采集,并将此电流数据经 A/D 转换后通过 RS232 串口发送至嵌入式系统模块,作为配电网重构的初始潮流参数。

根据该模块要实现的功能,可以发现其最大的特点在于需要采集的数据量众多,如果为每个节点配备一个 A/D 通道,显然会使系统变得庞大而难于控制,成本也十分巨大。

因此,本文选用 74HC155 解码芯片来控制四块 16 选一模拟开关 CD4067B,通过 74HC155 编码器控制 CD4067B 的使能管脚,单片机只需要 7 个 I/O 控制管脚就可实现最多  $8 \times 16$  个通道的选通功能。

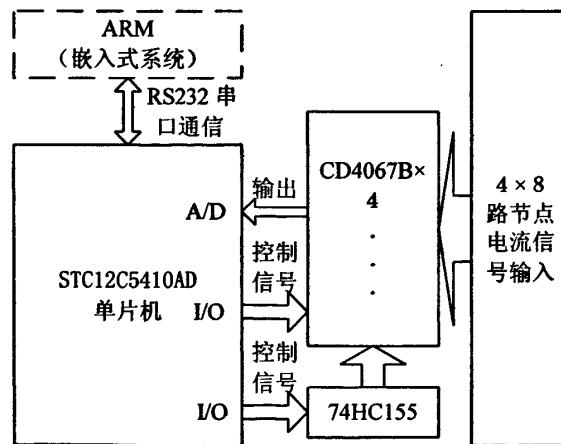


图 2-3 数据采集模块原理框图

控制核心选用了 STC12C5410AD,其有方便的 IAP 编程环境及超强的抗干扰能力。此外, STC12C5410AD 还内置 8 路 10 位高速 A/D 转换器,采集速度可达 100KHZ,完全可以满足本系统的采集精度及速度要求。

MCU 每次采集完一个节点参数后, 系统保存数据, 同时改变采集通道, 并将上一组采集的数据通过 RS232 传送至嵌入式系统保存, 然后进入下一个节点的采集。当系统中所有节点都采集完一个周期后, 嵌入式系统开始运行新一轮的配电网络重构算法, 待计算完后输出结果至重构执行模块, 同时发送命令给数据采集模块开始下一轮采集过程。该模块的结构框图如图 2-3 所示。

### 2.3.3 重构执行模块

重构执行模块主要功能在于接收从嵌入式系统的 RS232 发送过来配电网络重构控制信号, 并将其转换为 50 个继电器开关的控制信号, 执行改变配电网络拓扑结构的重构任务。

本设计选用了 ALTERA 旗下高性能低价格的 CYCLONE III 代 FPGA 为控制核心。CYCLONE III 是 ALTERA 在 2007 年底新上市的产品, 被称为功耗、功能及成本完美的结合品, 非常适合对成本敏感的大批量应用。同时, ALTERA 还提供大量的 IP 核共享, 大大缩短了研发人员的时间及经济成本。

系统选用了 144 管脚封装的 EP3C25E144, EP3C25 系列芯片有超过两万四千个逻辑门, 近 600KBit 存储空间, 66 个可用 I/O 口以及支持多种 I/O 口电压标准。此外, 本系统设计支持 JTAG 及 AS 配置两种调试模式, 采样 EPCS4 作为 AS 配置芯片。所有 I/O 口全部配置为 3.3VTTL 模式。

信号从 FPGA 输出后, 首先经过光电隔离 TLP521, 其所起的作用主要有两点: 第一, 隔离外部噪音, 提高系统工作的可靠性。重构执行模块是将重构结果直接输出的环节, 所以该部分工作的可靠性也格外重要, 如果某个隔离开关因为噪音发生误动, 就直接改变了整个网络的拓扑结构甚至引发供电故障, 影响巨大。第二, 电平转换。通过 TLP521, 可以顺利将输入为 3.3V 的信号转换为 12V 的开关量输出以直接驱动继电器。重构执行模块的原理框图如图 2-4 所示。

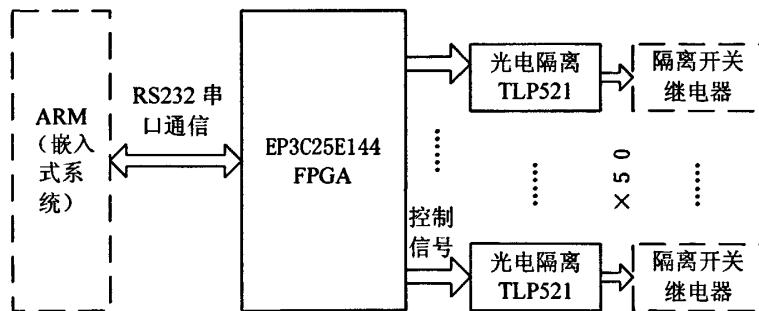


图 2-4 重构执行模块原理框图

### 2.3.4 模拟配电网络模块

模拟配电网络模块以 IEEE 配电网可靠性测试系统 IEEE-RBTS Bus2 和 IEEE-RBTS Bus 4 为样本。

IEEE-RBTS Bus 2 系统有 22 个负荷点、16 个开关, 其中 S15 和 S16 为联络

开关。

IEEE-RBTS Bus 4 系统有 38 个负荷点、33 个开关，其中 S46、S47、S48、S49 为联络开关。

此外，为了串联 BUS 2 和 BUS 4 以及增加网络复杂程度，更好地符合中国国情的配电网络拓扑结构，还在 BUS 2 和 BUS 4 之间增加了一个联络开关 S50。

模拟配电网络模块电路具体网络结构见图 2-5。网络具体参数见附表 1。

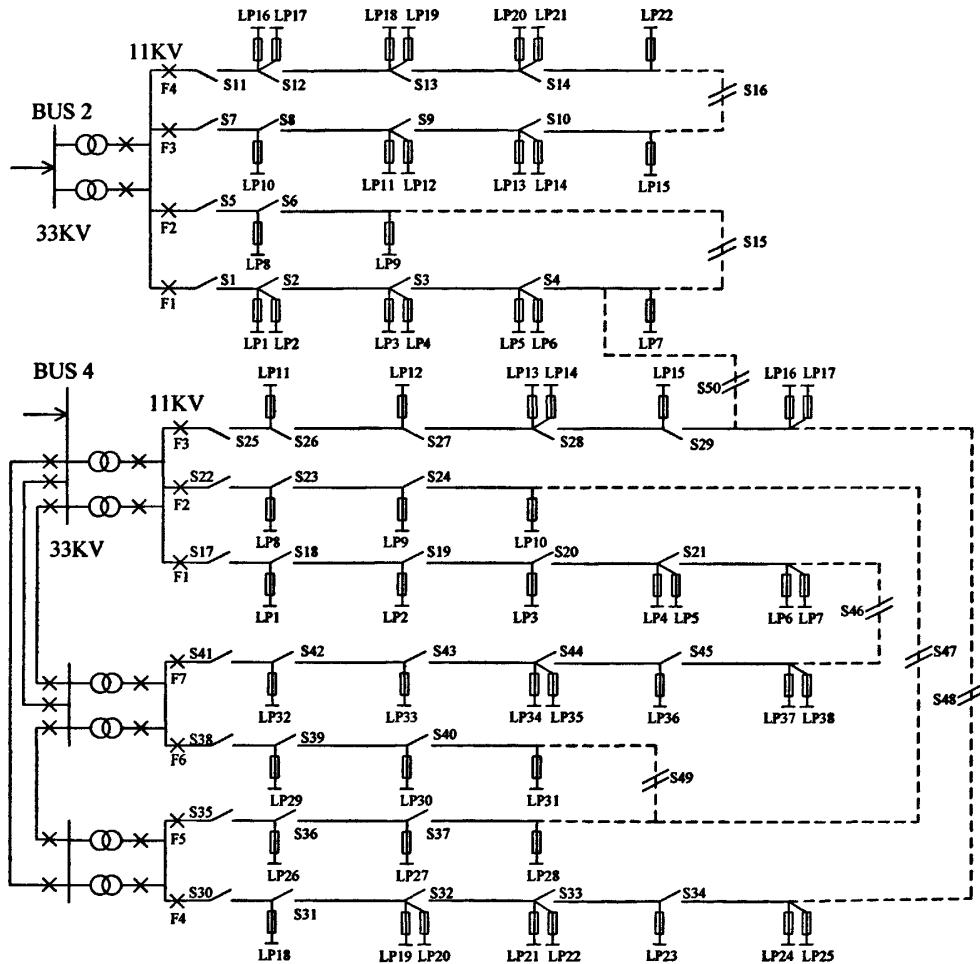


图 2-5 参考 IEEE RBTS Bus 2 和 Bus 4 的配电网络结构图

## 2.4 本章小结

本章首先对整个系统的设计方案进行了论证，并最终选择了以模块化方式搭建组成重构实验装置，根据不同的应用需求，选择不同的功能模块搭建不同等级的两种实验平台。

然后，文章对系统的各个组成模块的功能、实现原理进行了具体的分析介绍。

## 第三章 嵌入式系统平台搭建

本课题中，嵌入式系统平台的搭建将按照 ARM9+Linux 开发模式进行。

为了充分结合实验室现有实验条件以及降低系统开发的风险，硬件平台上采用的是北京博创公司开发的 UP-NETARM2410-S 实验箱，该实验箱以三星公司的 S3C2410X 为核心处理器，配备了由 AX88796 构成的双网卡，两个 RS232 串口，还增加了一个 RS485 串口，S3C2410X 的主 USB 口也被扩展至 4 个，等等……板载资源十分丰富<sup>[33]</sup>。

嵌入式操作系统方面，本方案采用的是嵌入式 Linux，其优点在于源码完全开放、价格低廉、功能强大以及易于移植。

### 3.1 硬件平台

#### 3.1.1 S3C2410X 处理器

平台核心的S3C2410X<sup>[34][35]</sup>是韩国三星电子公司一款基于ARM920T内核的16/32位RISC嵌入式微处理器，主要面向手持式设备以及高性价比、低功耗的应用。S3C2410X内部集成了大量的功能单元，包括：

- 1、内部1.8V，存储器3.3V，外部I/O口3.3V，16KB 数据Cache，16KB 指令Cache，MMU；
- 2、内置外部存储器控制器(SDRAM 控制和芯片选择逻辑)；
- 3、LCD控制器(最高4K色STN和256K彩色TFT)，一个LCD专用DMA；
- 4、4路带外部请求线的DMA；
- 5、三个通用异步串行端口(IrDA1.0,16-Byte Tx FIFO, and 16-Byte Rx FIFO), 2通道SPI；
- 6、一个多主IIC总线，一个IIS总线控制器；
- 7、SD主接口版本1.0和多媒体卡协议版本2.11兼容；
- 8、2个USB HOST，一个USB DEVICE(VER1.1)；
- 9、4个PWM定时器和一个内部定时器；
- 10、看门狗定时器；
- 11、117个通用IO；
- 12、24个外部中断；
- 13、电源控制模式：标准、慢速、休眠、掉电；
- 14、8通道10位ADC和触摸屏接口；
- 15、带日历功能的实时时钟；
- 16、芯片内置PLL；

17、设计用于手持设备和通用嵌入式系统；

18、16/32位RISC体系结构，使用ARM920T CPU核的强大指令集；

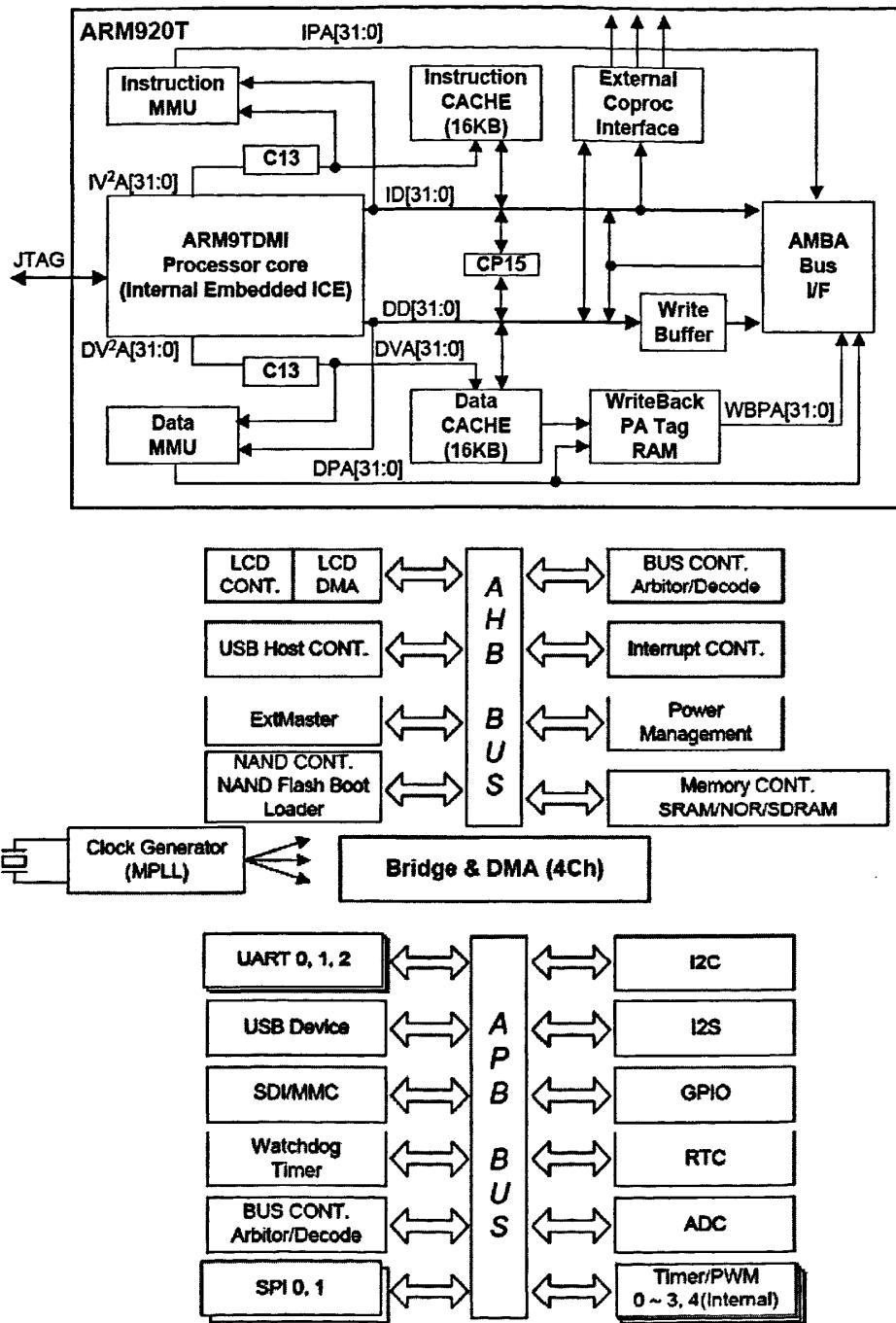


图3-1 S3C2410 ARM MPU体系结构框图

19、ARM带MMU的先进的体系结构支持WINCE、EPOC32、LINUX；

20、指令缓存(cache)、数据缓存、写缓冲和物理地址TAG RAM，减小了对

主存储器带宽和性能的影响;

- 21、ARM920T CPU核支持ARM调试的体系结构;
- 22、内部先进的位控制器总线(AMBA2.0, AHB/APB).  
其结构图如图3-1所示。

S3C2410X在时钟方面有其突出的特点, 该芯片集成了一个具有日历功能的RTC和具有PLL (MPLL和UPLL) 的芯片时钟发生器。MPLL产生主时钟, 能够使处理器工作频率最高达到203MHz。这个工作频率能够使处理器轻松运行于Windows CE, Linux等操作系统以及进行较为复杂的信息处理。UPLL产生实现主从USB功能的时钟。

S3C2410X将系统的存储空间分为8组 (Bank), 每组的大小是128MB, 共1GB。Bank0到Bank5的开始地址是固定的, 用于ROM或SRAM。Bank6和Bank7用于ROM、SRAM或SDRAM, 这两个组可编程且大小相同。Bank7的开始地址是Bank6的结束地址, 灵活可变。所有内存块的访问周期都可编程。S3C2410X采用nGCS[7:0]8个通用片选信号选择这些组。

S3C2410X支持从NAND Flash启动, NAND Flash具有容量大、比NOR Flash价格低等特点。系统采用NAND Flash与SDRAM组合, 可以获得非常高的性价比。S3C2410X具有三种启动方式, 可通过OM[1:0]管脚进行选择:

- 1、OM[1:0]=00时处理器从NAND Flash启动;
- 2、OM[1:0]=01时处理器从16位宽的ROM启动;
- 3、OM[1:0]=10时处理器从32位宽的ROM启动;

用户可以将引导代码和操作系统镜像存放在外部的NAND Flash中, 并从NAND Flash启动。当处理器在这种模式下上电复位时, 内置的NAND Flash将访问控制接口, 并将引导代码自动加载到内部的SRAM (此时该SRAM定位于起始地址空间0x00000000, 容量为4KB) 并且运行。之后, SRAM中的引导程序将操作系统镜像加载到SDRAM中, 操作系统就能够运行在SDRAM中。启动完毕后, 4KB的启动SRAM就可以用于其他用途。如果从其他方式启动, 启动ROM就要定位于内存的起始地址0x00000000, 处理器直接在ROM上运行启动程序, 而4KB启动SRAM被定位于内存地址的0x40000000处。

S3C2410X对于片内的各个部件采用了独立的电源供给方式:

- 1、内核采用1.8V供电;
- 2、存储单元采用3.3V独立供电, 对于一般SDRAM可以采用3.3V, 对于移动SDRAM可以采用V<sub>DD</sub>等于1.8/2.5V;
- 3、V<sub>DDQ</sub>等于3.0/3.3V;
- 4、I/O采用独立3.3V供电。

### 3.1.2 博创 UP-NETARM2410-S 实验箱简介

UP-NetARM2410-S是集教学、科研于一身的强大实验平台，是被中国电子学会嵌入式专委会选定为“嵌入式（助理）工程师培训认证”的指定产品<sup>[36]</sup>。其硬件结构框图如图3-2所示。

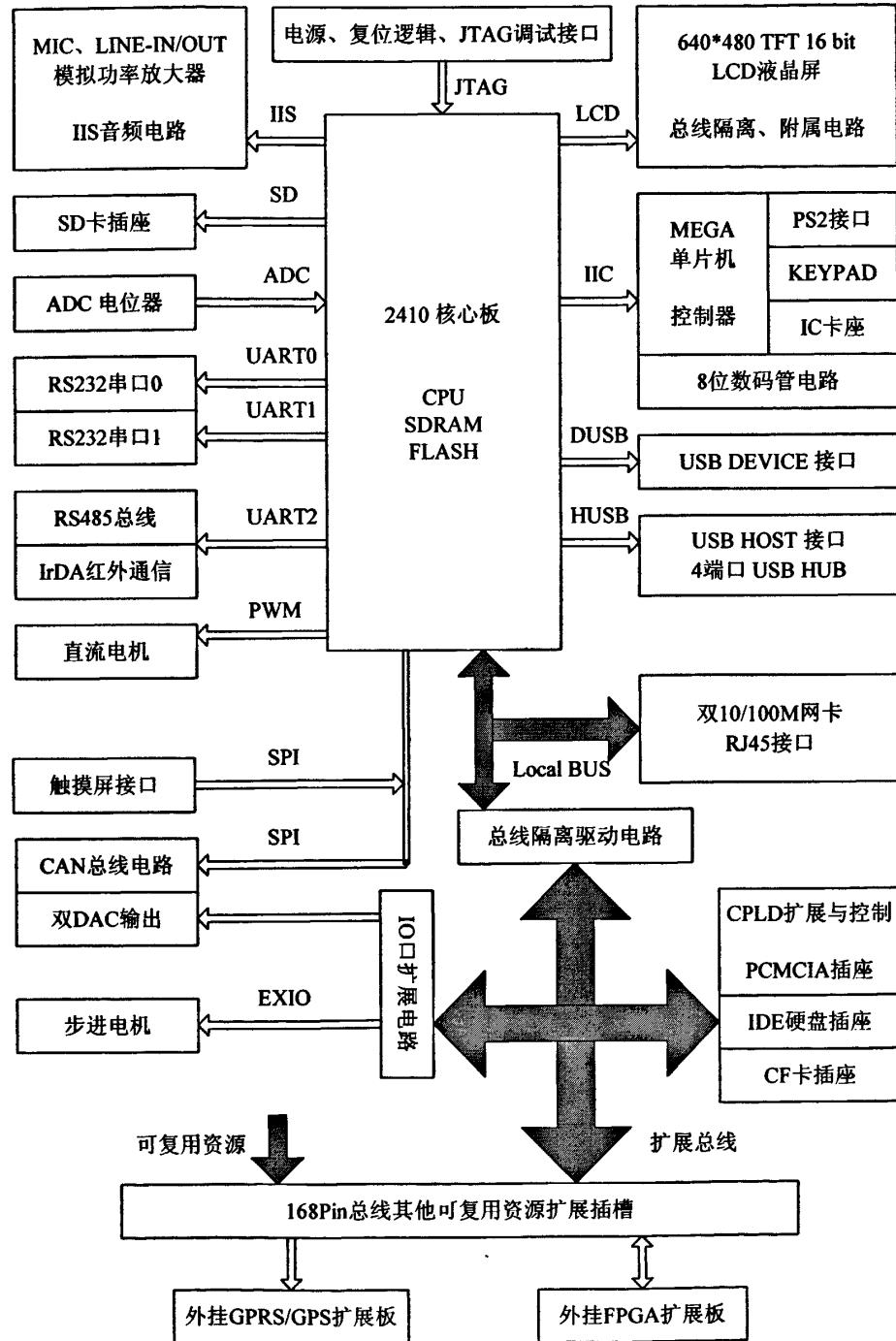


图3-2 UP-NetARM2410-S硬件架构框图

其详细的外围硬件配置如表3-1所示：

表3-1 UP-NetARM2410-S 硬件配置表

配置名称	型号	说明
CPU	ARM920T 结构芯片三星 S3c2410X	工作频率203MHz
FLASH	SAMSUNG K9F1208	64M NAND
SDRAM	HY57V561620AT-H	32M×2=64M
Ethernet网卡	AX88796 两片	10/100M 自适应
LCD	LQ080V3DG01	8寸16bit TFT
触摸屏	SX-080-W4R-FB	FM7843 驱动
LED	ZLG7290	四个共阴极LED
USB接口	4个HOST/1个DEVICE	由AT43301构成USB HUB
UART/IrDA	2个RS232, 1个RS485, 1个IrDA	从处理器的UART2引出
A/D	由S3C2410X芯片引出	3个电位器控制输入
AUDIO	IIS总线, UDA1341芯片	44.1KHz 音频
扩展卡插槽	168Pin EXPORT	总线直接扩展
GPS_GPRS扩展板	SIMCOM的SIM100-E模块	支持双道语音通信
IDE/CF卡插座	笔记本硬盘, CF卡	
PCMCIA和SD卡插座	PCMCIA型号为DWL-650	
PS2	PC键盘和鼠标	由ATMEGA8单片机 控制
IC卡座	AT24CXX系列	由ATMEGA8单片机 控制
DC/STEP电机	DC由PWM控制, STEP由74HC573 控制	
CAN BUS	由MCP2510和TJA1050构成	
Double DA	MAX504	两个10位DAC端口
调试接口	JTAG	14针、20针

由图3-2以及表3-1可以看出, UP-NETARM2410-S在S3C2410X周围, 配置了丰富了实验接口以及外围电路, 从而满足更多基于嵌入式系统平台的开发实验和研究需求。

## 3.2 嵌入式 Linux

### 3.2.1 嵌入式 Linux 系统

近几年得到蓬勃发展的Linux操作系统，已广泛应用于各类计算应用，不仅包括IBM的微型Linux腕表、手持设备(PDA和蜂窝电话)、因特网装置、客户机、防火墙、工业机器人和电话基础设施设备，甚至还包括了基于集群的超级计算机。Linux在高端服务器的优越表现及其天生具有的突出特点，就注定它必将在低端嵌入式系统中再次给人们以惊喜，而基于嵌入式Linux操作系统的应用，必定给我们未来的工作和生活带来翻天覆地的变化。

Linux正在嵌入式开发领域稳步发展<sup>[37]</sup>。Linux使用GPL，所有对特定开发板、PDA、掌上机、可携带设备等使用嵌入式Linux感兴趣的人都可以从因特网上免费下载其内核和应用程序，并开始移植和开发。许多Linux改良品种迎合了嵌入式市场，它们包括 RTLinux（实时Linux）、μclinux（用于非MMU设备的Linux）、Montavista Linux（用于ARM、MIPS、PPC的Linux分发版）、ARM-Linux（ARM上的Linux）和其它Linux系统。

嵌入式Linux的发展比较迅速。NEC、索尼已经在销售个人视频录像机等基于Linux的消费类电子产品，摩托罗拉则计划在其未来的大多数手机上使用Linux，IBM也制定了在手持机上运行Linux的计划。

数年来，“Linux标准库”组织一直在从事对在服务器上运行的Linux进行标准化的工作，现在，嵌入式计算领域也开始了这一工作。嵌入式Linux标准吸引了“Linux标准库”以及Unix组织中有益的元素。Linux的安装和管理比Unix更加简单灵活，同时Linux中有许多命令和编程接口同传统的Unix一样。但是对于习惯于Windows操作系统的人来说，需要记忆大量的命令行参数却是一个缺点。随着Linux社团的不断努力，Linux的人机界面开发环境正在不断完善<sup>[38]</sup>。

典型的Linux系统经过打包，在拥有硬盘和大容量内存的PC机上运行，嵌入式系统不需要这么高的配置。一个功能完备的Linux内核要求大约1MB内存。而Linux微内核只占用其中很小一部分内存，包括虚拟内存和所有核心的操作系统功能在内，只需占用系统约100KB内存。只要有500KB的内存，一个有网络栈和基本实用程序的完整的Linux系统就可以在一台8位总线（SX）的Intel386微处理器上运行的很好了。由于内存要求常常是需要的应用所决定的，比如Web服务器或者SNMP代理，Linux系统甚至可以仅使用256KB ROM和512KB RAM进行工作。因此它是一个瞄准嵌入式市场的轻量级操作系统。

与传统的实时操作系统相比（RTOS），采用像嵌入式Linux这样的开放源码的操作系统的另外一个好处是Linux开发团体看来会比RTOS的供应商更快地支持新的IP协议和其它协议。例如，用于Linux的设备驱动程序要比用于商业操作

系统的设备驱动程序多，如网络接口卡（NIC）驱动程序以及并口和串口驱动程序。核心Linux操作系统本身的微内核体系结构相当简单，网络和文件系统以模块形式置于微内核的上层，驱动程序和其它部件可在运行时作为可加载模块编译到或者是添加到内核，这为构造定制的可嵌入系统提供了高度模块化的构件方法，而在典型情况下该系统需结合定制的驱动程序和应用程序以提供附加功能。

嵌入式系统也常常要求通用的功能，为了避免重复劳动，这些功能的实现运用了许多现成的程序和驱动程序，它们可以用于公共外设和应用。Linux可以在外设范围广泛的多数微处理器上运行，并早已经有了现成的应用库。Linux用于嵌入式的因特网设备也是很合适的，原因是它支持多处理器系统，该特性使Linux具有了伸缩性。因而设计人员可以选择在双处理器系统上运行实时应用，提高整体的处理能力<sup>[39]</sup>。例如，您可以在一个处理器运行GUI，同时在另一个处理器上运行Linux系统。在嵌入式系统上运行Linux的一个缺点是Linux体系提供实时性能需要添加实时软件模块。而这些模块运行的内核空间正是操作系统实现调度策略、硬件中断异常和执行程序的部分。由于这些实时软件模块是在内核空间运行的，因此代码错误可能会破坏操作系统从而影响整个系统的可靠性，这对于实时应用将是一个非常严重的弱点。已经有许多嵌入式Linux系统的示例，可以有把握地说，某种形式的Linux能在几乎任何一台执行代码的计算机上运行。

### 3.2.2 嵌入式 Linux 的技术架构

嵌入式系统的软件平台结构可以分为四个层次：设备驱动层，操作系统内核层，应用中间层和应用系统层<sup>[40][41]</sup>。如图3-3所示：

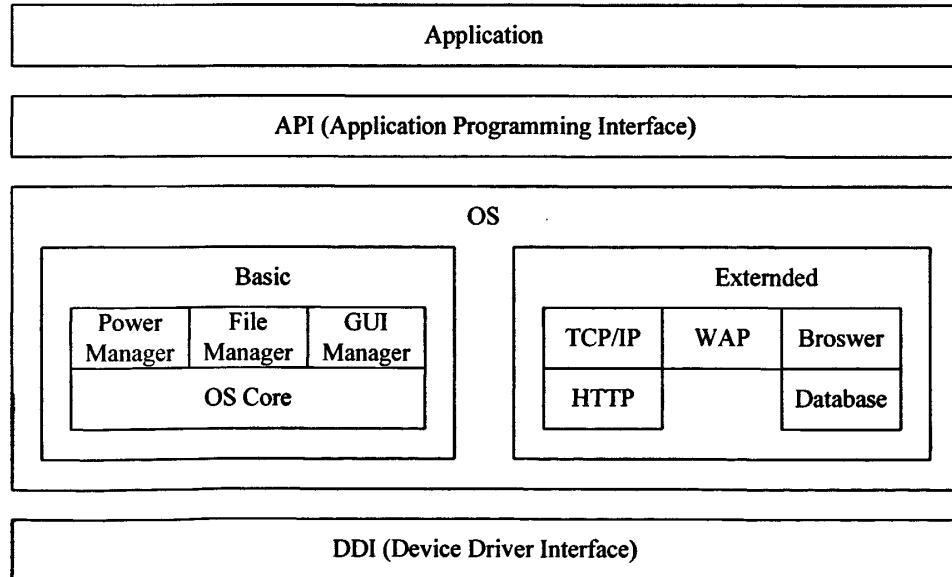


图3-3 Linux软件平台系统架构图

设备驱动接口DDI：它负责嵌入式系统与外部设备的信息交互；

**操作系统RTOS:** 该系统分成基本和扩展两部分。前者是操作系统的核，负责整个系统的任务调度、存储分配、时钟管理和中断管理，并提供文件、GUI等基本服务；后者是为用户提供操作系统的扩展功能，包括网络、数据库等；

**中间件:** 该接口是为编制应用程序提供的各种编程接口库(Lib)。它可以针对不同应用领域、不同安全要求分别构建，从而减轻应用开发者的负担；

**应用系统层:** 即特定功能的应用程序。

### 3.3 基于 ARM+Linux 的嵌入式系统的开发

嵌入式操作系统是ARM CPU重要的软件基础。从8位/16位单片机发展到以ARM CPU核为代表的32位嵌入式处理器，嵌入式操作系统将替代传统的由手工编制的监控程序或调度程序，成为重要的基础组件。嵌入式操作系统可以提供ARM CPU核和外设的设备管理，如核的初始化、MMU(内存管理单元)、定时器、中断、串口、SPI, GPIO、以太网、图形接口等，嵌入式操作系统提供任务调度、内存管理、中断处理、同步互斥机制、文件系统和网络功能，更重要的一点是嵌入式操作系统对应用程序可以起到屏蔽的作用，让应用程序员面向操作系统级开发应用软件，并易于在不同的基于ARM核的嵌入式处理器上移植<sup>[42][43]</sup>。

嵌入式系统通常为一个资源受限的系统。直接在嵌入式系统的硬件平台上编写软件比较困难，有时甚至是不可能的。目前，一般采用的办法是，先在通用计算机上编写程序，然后，通过交叉编译，生成目标平台上可运行的二进制代码格式，最后下载到目标平台上的特定位置上运行，具体步骤如下<sup>[44-48]</sup>：

第一步，建立嵌入式Linux交叉开发环境。目前，常用的交叉开发环境主要有开放和商业两种类型。开放的交叉开发环境的典型代表是GNU工具链，目前已经能够支持x86, ARM, MIPS, PowerPC等多种处理器。商业的交叉开发环境主要有Metrowerks CodeWarrior, ARM Software Development, Toolkit, SDS Cross compiler, WindRiver Tornado, Microsoft Embedded Visual C++等。交叉开发环境是指编译、链接和调试嵌入式应用软件的环境。它与运行嵌入式应用软件的环境有所不同，通常采用宿主机/目标机模式。

第二步，交叉编译和链接。在完成嵌入式软件的编码之后，就是进行编译和链接，以生成可执行代码。由于开发过程大多是在Intel公司x86系列CPU的通用计算机上进行的，而目标环境的处理器芯片却大多为ARM, MIPS, PowerPC, DragonBall等系列的微处理器，这就要求在建立好的交叉开发环境中进行交叉编译和链接。

基于ARM体系结构的gcc交叉开发环境中，arm-linux-gcc是交叉编译器，arm-linux-ld是交叉链接器。通常情况下，并不是每一种体系结构的嵌入式微处理器都只对应于一种交叉编译器和交叉链接器。编写好的嵌入式软件经过交叉编译

和交叉链接后，通常会生成两种类型的可执行文件：用于调试的可执行文件和用于固化的可执行文件。交叉编译和链接过程如图3-4所示：

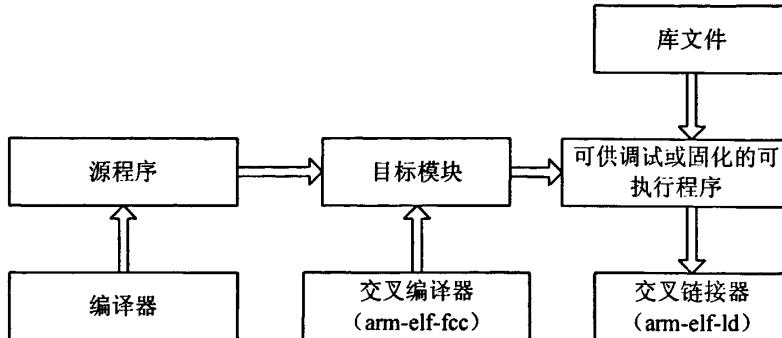


图3-4 嵌入式Linux交叉编译链接过程框图

### 第三步，交叉调试。

硬件调试。如果不采用在线仿真器，可以让CPU直接在其内部实现调试功能，并通过在开发板上引出的调试端口，发送调试命令和接收调试信息，完成调试过程。ARM公司提供的开发板上使用的则是JTAG调试端口。使用合适的软件工具与这些调试端口进行连接，可以获得与ICE类似的调试效果。

软件调试。在嵌入式Linux系统中，Linux系统内核调试，可以先在Linux内核中设置一个调试桩（debug stub），用作调试过程中和宿主机之间的通信服务器。然后，可以在宿主机中通过调试器的串口与调试桩进行通信，并通过调试器控制目标机上Linux内核的运行。

嵌入式上层应用软件的调试可以使用本地调试和远程调试两种方法。如果采用的是本地调试，首先要将所需的调试器移植到目标系统中，然后就可以直接在目标机上运行调试器来调试应用程序了；如果采用的是远程调试，则需要移植一个调试服务器到目标系统中，并通过它与宿主机上的调试器共同完成应用程序的调试。在嵌入式Linux系统的开发中，远程调试时目标机上使用的调试服务器通常是gdbserver，而宿主机上使用的调试器则是gdb。两者相互配合共同完成调试过程。

第四步，系统测试。整个软件系统编译过程，嵌入式系统的硬件一般采用专门的测试仪器进行测试，而软件则需要有相关的测试技术和测试工具的支持，并要采用特定的测试策略。测试技术指的是软件测试的专门途径，以及能够更加有效地运用这些途径的特定方法。在嵌入式软件测试中，常常要在基于目标机的测试和基于宿主机的测试之间做出折衷。基于目标机的测试需要消耗较多的时间和经费，而基于宿主机的测试虽然代价较小，但毕竟是在仿真环境中进行的，因此难以完全反映软件运行时的实际情况。这两种环境下的测试可以发现不同的软件缺陷，关键是要对目标机环境和宿主机环境下的测试内容进行合理取舍。嵌入式

软件测试中经常用到的测试工具主要有：内存分析工具、性能分析工具、覆盖分析工具、缺陷跟踪工具等。

### 3.4 本章小结

本章首先从硬件部分着手，介绍了 S3C2410X 三星 ARM 处理芯片的内核及其外部资源，以及博创 UP-NETARM2410-S 实验箱的结构框图及其硬件配置。

软件部分，从嵌入式 Linux 近年来的飞速发展入手，点出了其优点和存在的不足，此外，文章还就嵌入式 Linux 的技术架构进行了分析。

最后，本章就由ARM加嵌入式Linux构建的嵌入式系统开发设计的步骤及其要点进行分析介绍。

## 第四章 系统硬件设计

如第二章所述，整个系统采用模块化设计方式，根据系统中各部分应实现的功能将整个系统划分为四大功能模块。其中，嵌入式系统模块在之前的第三章已经作了详细阐述，在此不再赘言。

下面，将具体给出系统其他功能模块的详细设计方案。

### 4.1 重构执行模块硬件设计

重构执行模块以 ALTERA 公司高性价比的 CYCLONE 系列最新的III代 FPGA EP3C25E144 为核心，外加电源电路、配置电路以及 RS232 通信接口电路等组成。同时，芯片所有的 I/O 口以及时钟输入管脚都通过插接口引出，设计为最小开发板的模式，以方便进行其他功能的扩展。此外，在本系统中，通过这些插接口的引出，重构结果通过插接板上的 TLP521 即可对模拟配电网络模块的继电器进行控制。而为了更清楚地描述其工作原理，TLP521 部分的电路将在模拟配电网络模块中一并介绍，下面先就该模块其他功能部分的电路进行介绍：

FPGA 设计中比较关键的几个环节，第一是供电，CYCLONE III<sup>[49]</sup>的供电分为三个部分：内核供电、PLL 锁相环供电、I/O 口供电。现在芯片都采用低功耗的方式，内核电压一般比较低，这款芯片的内核电压为 1.2V。所以功耗小，发热量少，可靠性也就更高。此外，其 PLL 锁相环的供电电压为 2.5V，由于锁相环是模拟电路，所以对电源和地要求都比较高，要通过电感或 0 欧电阻与数字地隔离开来。另外，其 I/O 口的供电可以以同 BANK 为单位在 1.2V 到 3.3V 之间可选。我们这里统一采用 3.3VTTL 供电方式，所以只要把每个 VCCIO 都连接到 3.3V 电源处即可。

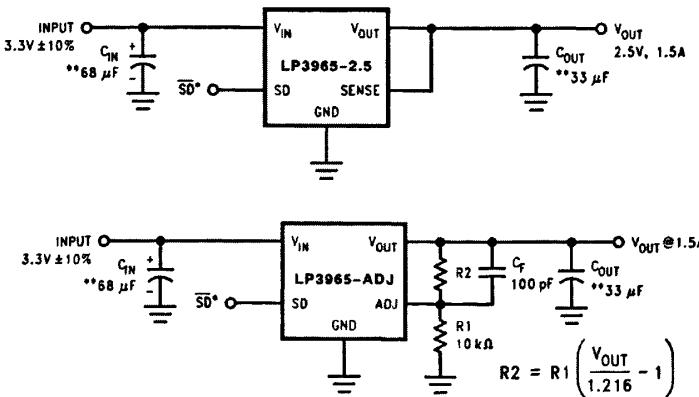


图 4-1 LP3965 电源稳压电路

该设计的供电方案是这样的，输入电压定为 5±0.5V，考虑到 I/O 口众多，

且都使用 3.3V 供电，电流可能较大，所以，在此选用供电电流可达 3A 的 LT1085CM 直接从输入电压稳定出 3.3V 以供使用；对于 2.5V 及 1.2V 电源，由于其输出电压较低，为了降低因压降过大在稳压块上产生的能耗，先将输入的 5V 左右电压经过两个 S3A 的二极管降压后提供给其稳压块，由于供电电流都不是很大，分别选用了最大可提供 1.5A 电流 LP3965-2.5 及 LP3965-ADJ 以提供稳定的 2.5V 及 1.2V 电压源。LP3965 详细电路可参见图 4-1 所示，其它详细资料可参见相关芯片文档。

除了供电，FPGA 的全局时钟口也是在设计时要多加注意的地方，不用的时钟口一定要直接接地或者通过电阻接地。本方案将所有的 16 个全局时钟口统一使用 330 欧姆电阻连接至地。这样，既不会对输入时钟产生影响，也可以防止未使用时钟口被悬空引入干扰。重构执行模块的 FPGA EP3C25E144 电路原理见图 4-2 所示。

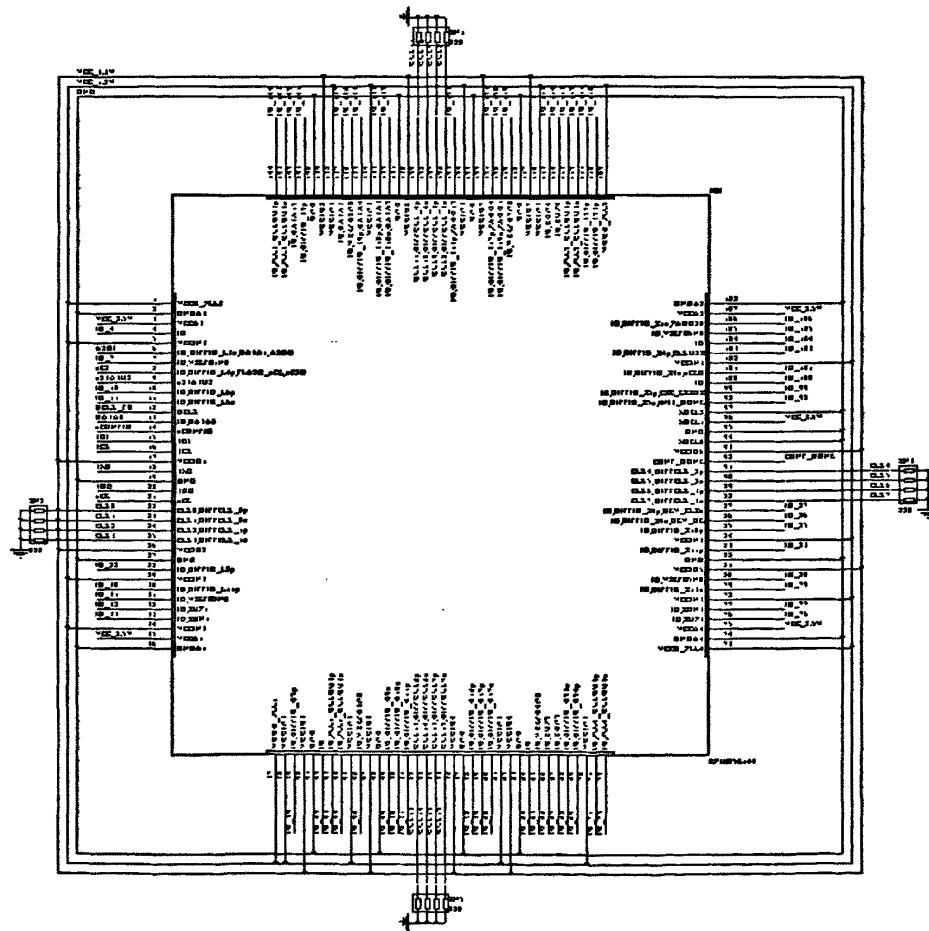


图 4-2 重构执行模块 FPGA 电路原理图

FPGA 由于内部采用 SRAM 工艺，它的配置数据存储在 SRAM 中。由于

SRAM 的易失性，每次系统上电时，必须重新配置数据，即 ICR(In-Circuit Reconfigurability)，只有在数据配置正确的情况下系统才能正常工作。在线配置方式一般有两类：一是通过下载电缆由计算机直接对其进行配置，二是通过配置芯片对其进行配置。Altera 公司为 CYCLONE III 提供了 AS (Active Serial)、PS (Passive Serial)、FPP (Fast Passive Parallel)、AP (Active Parallel)、JTAG (Joint Test Action Group) 等五种配置方式，EP3C25E144 则支持其中的 AS、PS、JTAG 三种配置模式。

FPGA 设计中第三个比较关键的环节就在于配置模式，本设计中，系统选用了 JTAG 加 AS 的配置模式，使用 Altera 公司提供的专用 AS 下载芯片 EPICS4 对 EP3C25E144 进行配置。虽然 Altera 官方资料介绍<sup>[50]</sup>的针对 EP3C25 系列 FPGA 最小应选取 EPICS16 的芯片，但通过相关人员介绍 EPICS4 已经完全可以满足要求了，我们也通过实验证实了这一点，其价格只有 EPICS16 的约 1/4。其配置电路如图 4-3 所示。

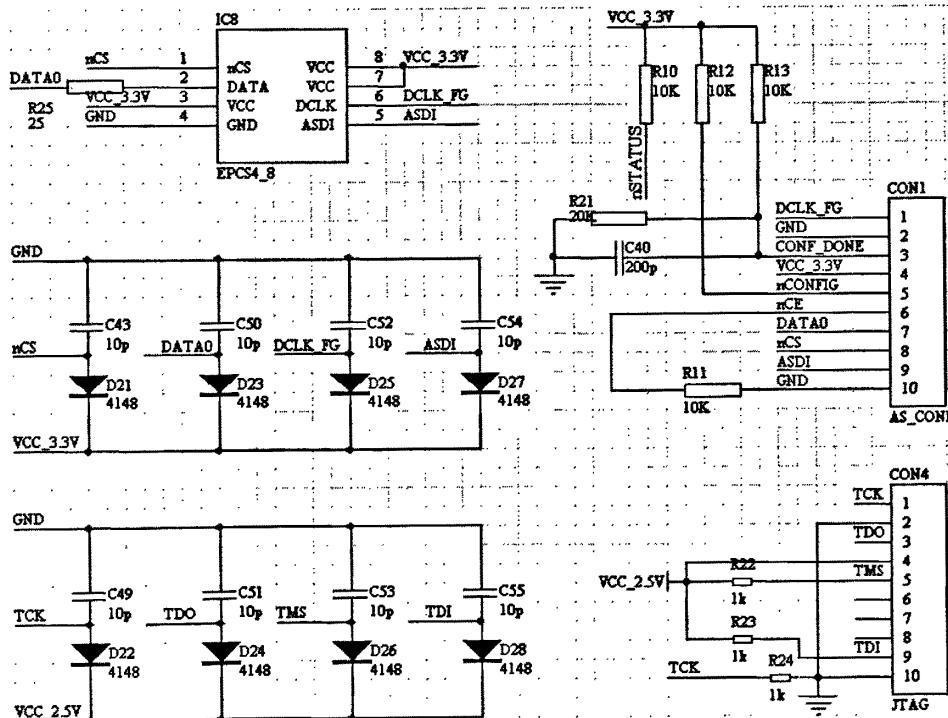


图 4-3 CYCLONE III 系列 FPGA 的 AS 加 JTAG 配置电路

此外，嵌入式系统计算出网络重构结果后，通过 RS232 串口将重构指令发送至重构执行模块，由于接口电平匹配的原因，需要为 FPGA 配置专用的电平转换电路。

在该设计中，RS232 电平转换选用了可支持 3.0-5.5V 宽电压输入的 MAX3232E 芯片，其详细电路如图 4-4 所示。

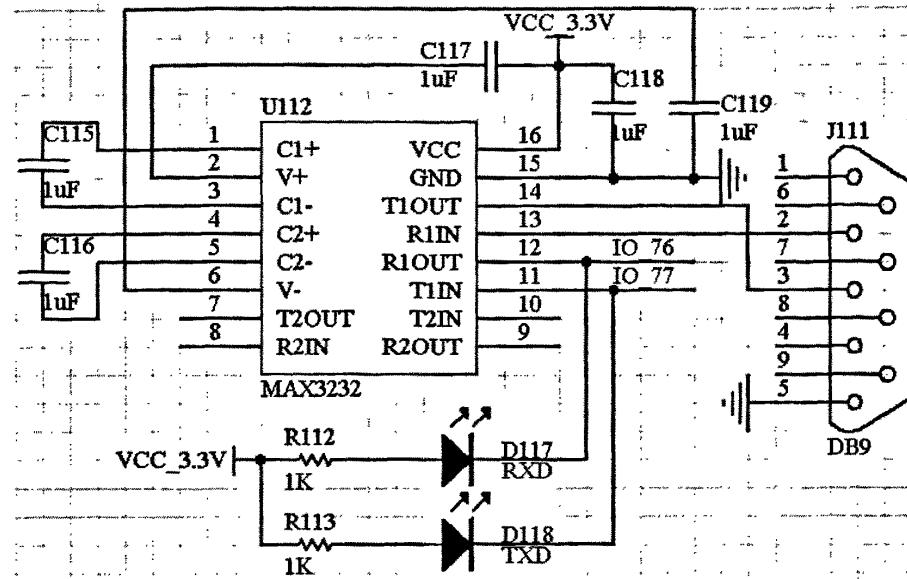


图 4-4 RS232 电路原理图

## 4.2 数据采集模块硬件设计

数据采集模块是以宏晶科技 STC12C5410AD 系列单片机为核心，对超过 43 路的支路电流信号通过模拟开关选择后再依次锁相、采样保持后送入单片机进行 A/D 采集，再根据傅立叶算法计算得出采集到每个支路的电流峰值、相位、频率，并将这些参数通过 RS232 接口发送至嵌入式系统模块，为配电网络重构算法提供相关的初始参数信息。

STC12C5410AD<sup>[51]</sup>系列单片机是宏晶公司的新一代增强型、低功耗 8051 单片机，其运行速度为一般标准 51 系列单片机的 8-12 倍。其加密性强，无法解密，有超强的抗干扰能力，内部集成 MAX810 专用复位电路，2KB 非易失性 E2PROM，8 路的 10 位 A/D 口，并具有标准的 SPI 和 RS232 串行口，机器周期缩减至 1 个时钟周期，且在系统可编程，大大方便了系统的研究与开发。所以，STC12C5410AD 完美地匹配了本系统在硬软件方面的要求。

从图 4-5 可以发现, 本设计为 STC12C5410AD 配备了一块 11.0592MHz 的有源晶振, 同时配备了手动复位电路和自制的 IAP 接口。

在输入信号通道方面，通过 74HC155 解码器的输出信号选中待检测输入信号所在的 CD4067B。CD4067B 是一款采用了 CMOS 工艺的 16 选 1 模拟多路开关(Multiplexer)，在这里将 CD4067B 的输出口连接到一起，是由于当 CD4067B 的/INHIBIT 口为高电平输入时，其输出口设定为三态，而 74HC155 解码的结果是只有一位为低电平，即意味着只有一块 CD4067B 的输出口有效。74HC155 可通过功能引脚配置成双通道的 2-4 解码器，也可以配置成一个通道的 3-8 路解码器。在本设计中，为了今后的功能扩展，我们将其配置为后一种的 3-8 解码器模

式，但是由于设计中只设计了 4 块 CD4067B，所以我们只用到了 8 路解码输出中的低四位。

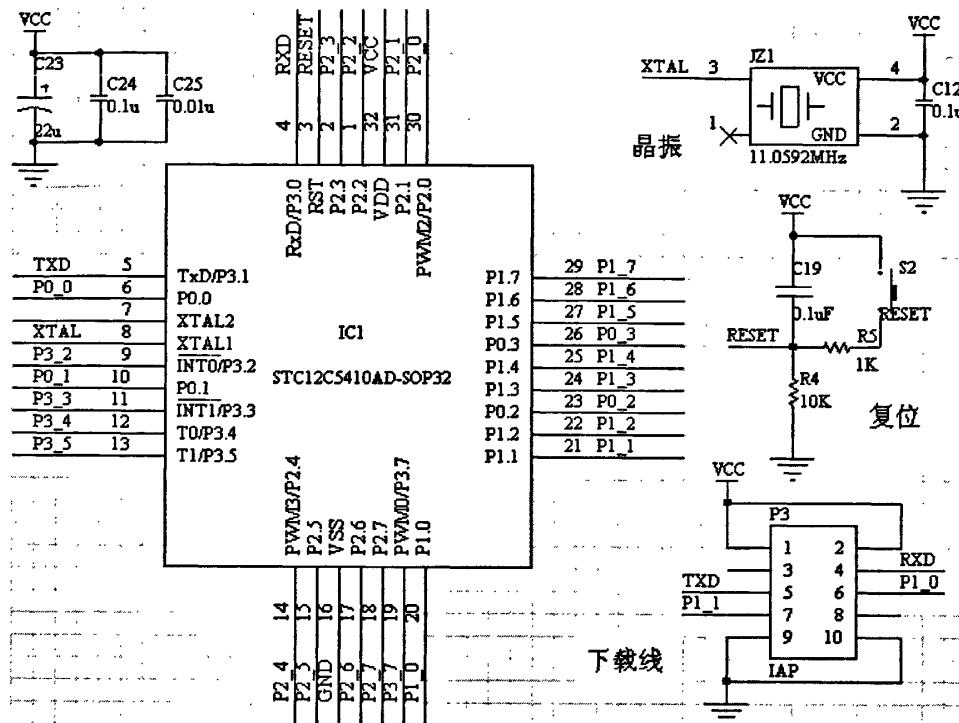


图 4-5 STC12C5410AD 电路原理图

CD4067B 除了有芯片使能输入/INHIBIT 外，还有四个控制信号输入端 A、B、C、D，即当 A、B、C、D 分别配置为 0、0、0、0 时，管脚“0”与管脚“OUT”接通，配置为 0、0、0、1 时，管脚“1”与管脚“OUT”接通，依此类推……

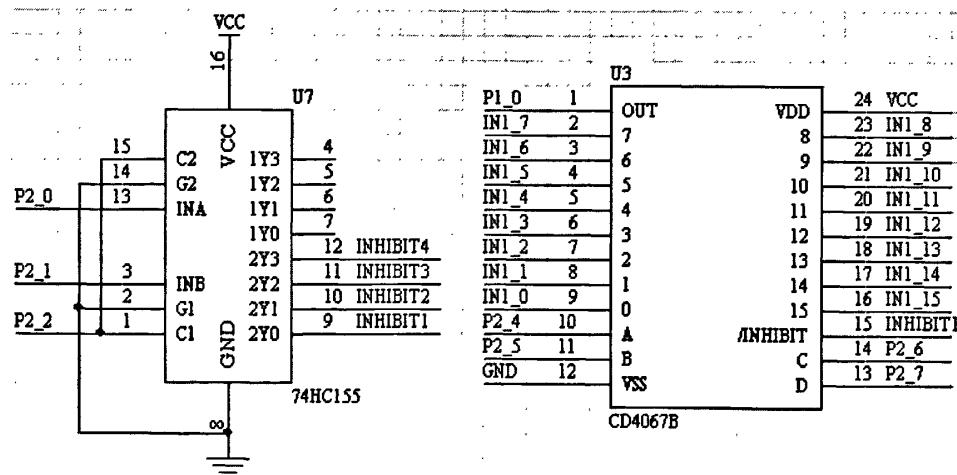


图 4-6 模拟开关切换电路

因此，通过控制 CD4067B 的这四个控制信号输入端，即可选中待测输入信

号所在通道通过模拟信号开关直通输出端“OUT”。其电路原理如图 4-6 所示，限于篇幅，图中只列出了四块 CD4067B 中的一块，其他三块电路原理相同，不同点只在于选通信号/INHIBIT 以及 16 路模拟输入信号。

信号通过模拟开关后，还要经过锁相、采样保持后，才能送到单片机 A/D 口，进行 A/D 采集并经过傅立叶算法计算出相关输入信号的相关参数，然后将其通过 RS232 串口送至核心控制模块。

电流互感器输出信号被转化为-5V 至+5V 的输出信号后，首先要经过如图 4-7 所示电路处理，即输入信号  $U_{IN0}$  通过一级负反馈放大器后，经二极管 D2 将 0V 以下的输入电平钳制在 0V，即输入的正选波在这里输出后将变成只有正半波电平输出，负的半波信号变为 0V。电路再经过一个相对 0V 的比较器将输入 0V 以上的电平全部输出为 5V，即通过这个电路后，-5V 至+5V 的正弦输入信号  $U_{IN0}$  转换为以过零点为分界的 0V 和 5V 方波信号。

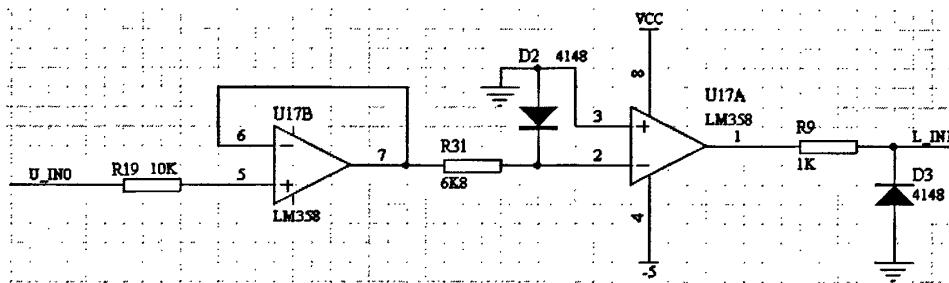


图 4-7 锁相电路原理 1

锁相环电路以 CD4046B 和 CD4020 为核心，其电路原理如图 4-8 所示。CD4046B 是一款低功耗的 CMOS 锁相环芯片，芯片包含一个低功耗的线性压控振荡器 (VCO)、有同一个公共输入端的两个不同的相位比较器以及一个 5.2V 的齐纳二极管。通过控制其 VCO 要求的一个外部电容及两个外部电阻，即可调节其 VCO 频率范围。CD4020 则是一款 14 等级的二进制波形计数器，其功能在于将输入的时钟 CLK 以不同的二进制单位进行计数，比如  $Q_4$  表示以时钟的 2 的 4 次方倍计数，即  $Q_4$  的周期将是 CLK 的 16 倍，依次类推，其最大输出周期  $Q_{14}$  可至 CLK 的 2 的 14 次方倍。

电路中，将上一环节锁相电路中的输出信号连接至 CD4046B 相位比较器的输入 A 口，将 CD4046B 的输出信号经 CD4020 分频后连接至 CD4046B 相位比较器的输入 B 口，通过这样一个反馈的比较后，我们就可以得到经锁相环倍频的理想频率的输出信号，以得到进行傅立叶算法所需的各个相位点的电压数据。比如在电路中我们将 CD4020 输出的  $Q_4$  连接至  $BIN_G$ ，设采集电压的  $L_{IN1}$  为工频 50Hz，两者比较后的输出  $VCOUT_G$  则是 800Hz。即可以在每个电压全波周期内得到 16 个采集点。

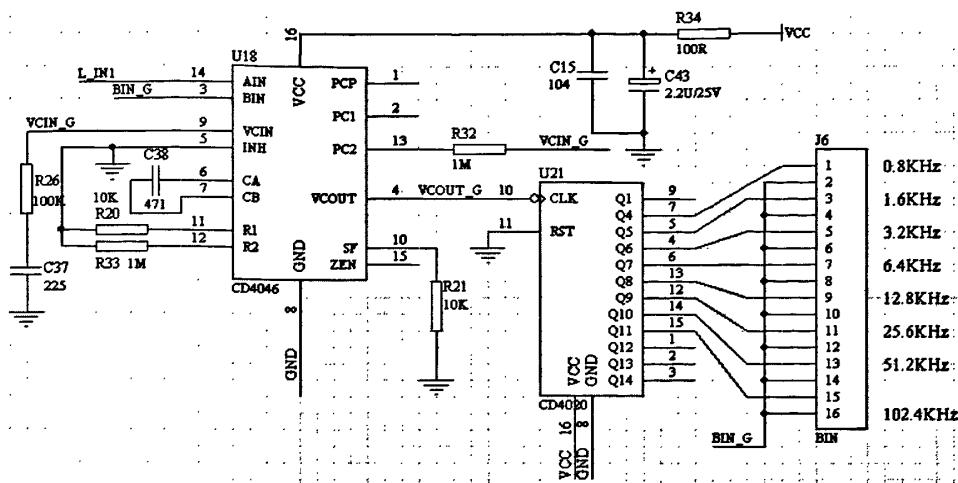


图 4-8 锁相环电路原理图

在 A/D 转换器进行采样期间，保持被转换输入信号不变的电路称为采样保持电路；A/D 转换器完成一次转换所需要的时间称为转换时间；对于不同 A/D 转换芯片，其转换时间各异，对于连续变化较快的模拟信号如果不采取采样保持措施，将会引起转换误差，当然对于慢速变化的模拟信号，在 A/D 转换系统中，完全可以不必采用采样保持电路，而且并不会影响 A/D 转换的精度。但是对于采集像我们这样的傅立叶算法参数，其一是采样频率较高，其二则是对采样精度较敏感，所以我们在这里还是为其配置了采用保持电路。

采样保持电路是指在逻辑电平的控制下处于“采样”或“保持”两种工作状态的电路，在采样状态下，电路的输出跟踪输入模拟信号，在保持状态下，电路的输出保持着前一次采样结束时刻的瞬时输入模拟信号，直到进入下一次采样状态为止。

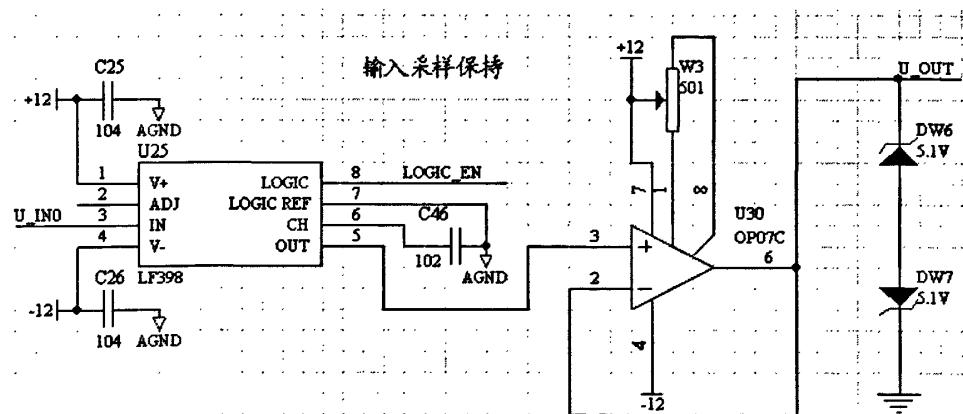


图 4-9 输入采样保持电路原理

如图 4-9 所示，在这里我们选用了通用的采样/保持器芯片 LF398，通过调节 LOGIC\_EN 管脚电平的高低来控制输出 U\_OUT 是对输入 U\_IN0 是进行采样还

是保持, 即当 LOGIC\_EN 为高电平时, U\_OUT 输出对 U\_IN0 的采样值, LOGIC\_EN 为低电平时, U\_OUT 则保持着前一次采样结束时刻的瞬时 U\_IN0 的输入模拟信号, 直到进入下一次采样状态为止。

### 4.3 模拟配电网络模块硬件设计

#### 4.3.1 模拟配电网络模块

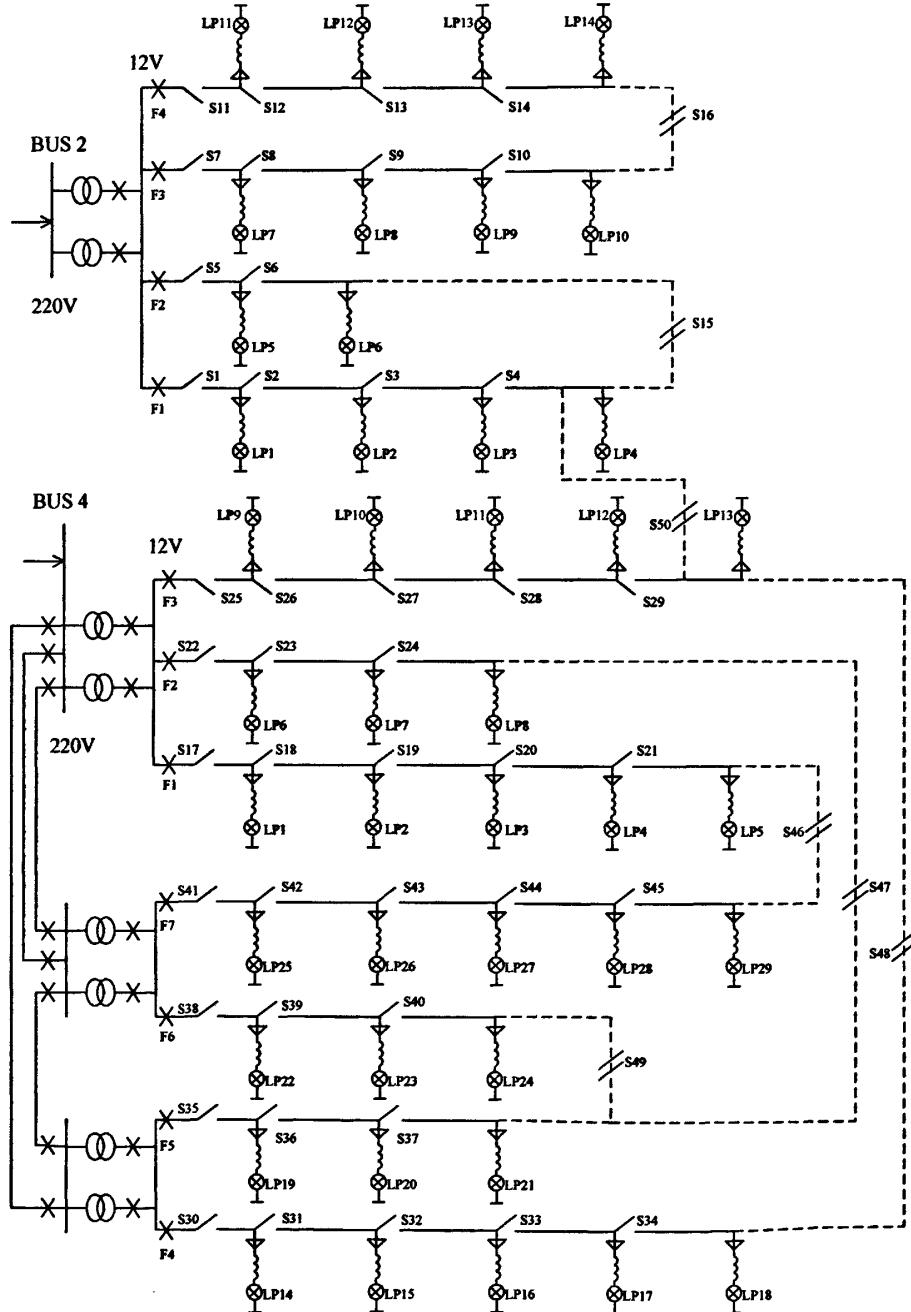


图 4-10 配电网重构实验装置模拟配电网络电路结构图

由于本文所要设计的是实验装置，同时出于安全及成本各方面的考虑，不能完全按照实际配电网来制作实验装置。所以根据实验装置对系统的要求，在设计模拟配电网模块硬件的过程中把 IEEE 中的测试系统及其参数做了以下相应简化、修改，最后得出的模拟配电网实验电路图如图 4-10 所示。

1、IEEE 中各 BUS 是以 33KV 为上级电压，通过两个一组的配电变压器降压后再发往各节点，而实验装置只能以 220V 为上级供电电压，并通过变压器组变为低压（12V），然后再进行配网实验，这样就可以选用额定电压低，功率小的元件（额定<12V，功率<10W），从而减少制作成本，并有利于操作安全。

2、各馈线断路器和分段开关都使用直流微型继电器代替，一共有 50 个继电器。同时为了简化电路，将图中同一节点上的不同负载（如 BUS2 的 F1 支路中 LP1, LP2）合并为一个节点负载。因为这并不会影响整个网络的拓扑结构与负载分配。这样整个实验装置一共有 43 个节点。

3、在各馈线每个节点负载的进线处安装了一个电流互感器（用三角形表示），以随时测量负载电流，并可计算得出馈线各段的线路电流，从而实现对整个电网电流的实时监控，方便实验数据的获取。

4、IEEE 图中的负载为下级变压器，而在本实验装置中，各节点都是由电阻和电感组成的终端用户负载。具体数据见附表 2。考虑到工作电流会比较大，所以负载是由小灯泡和 2W 大功率电阻、绕线电感（额定电流达到 3A）组成。

本实验装置选用的电流互感器为西安横山微型互感器研究所最新研制生产的 HCT215 测量用电流互感器，它体积小，额定工作频率为 50HZ，额定电流比为 5A/5mA。应用电路为图 4-11 所示。LM741 是单片高性能内补偿运算放大器。

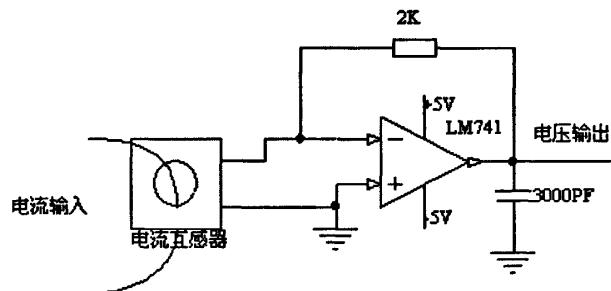


图 4-11 HCT215 实际应用电路图

此外，重构执行模块的命令下来后，首先经过光电隔离 TLP521 将 FPGA 的小电流信号转换为 12V 的继电器线圈开关信号，继电器选用的是 GIANT 的 JQD-14FC-2Z，可开断在 250V 交流电压下最大 5A 的电流。其应用电路如图 4-12 所示。

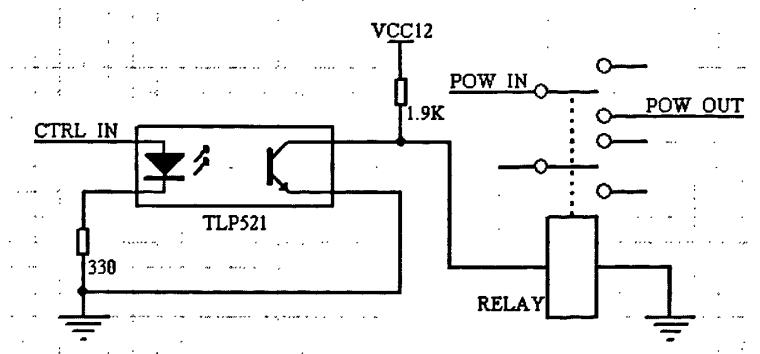


图 4-12 光隔及继电器控制电路原理图

由于本装置模拟配电网络电路线路不是很复杂，元器件相对简单且集成化程度低。同时最重要的是因为其模拟配电网络工作电流很大，所以不适宜制作电子线路板，除电流互感器二次侧电路制作线路板外，模拟配电网络电路各个元件都根据其电流选用不同导线连接，直接焊接在万能板上。

#### 4.3.2 LED 显示模块

在以实验配电网络重构算法为目标的第一等级系统中，我们使用 LED 来代表配电网络中所有开关，以 LED 的亮灭来显示开关的开合，设计制作了该 LED 显示模块。这样做有如下优点：其一是简化了系统结构，减少了在数据采集和计算上耗费的时间；其二是通过 LED 的显示来表现开关的开合状态，更加直接鲜明；其三是避免了大电流的开关过程中可能引起的安全问题，提高了系统运行可靠性。

LED 显示模块同样以图 4-10 所示 IEEE 配电网可靠性测试系统 IEEE-RBTS Bus2 和 IEEE-RBTS Bus 4 联络结构为样本，只是将其中的 50 个分段开关简化为 50 个 LED。因其电路原理较为简单，在此就不再详细介绍，其 PCB 板图可参见附 3.1 所示。

### 4.4 本章小结

本章介绍了各个系统功能模块的硬件设计，从以 ALTERA 公司 CYCLONE III 系列 FPGA 为中央控制单元的重构执行模块电路，到以 STC 单片机 STC12C5410AD 为核心的数据采集模块及其各个功能部分的电路原理及实现过程，以及根据第一等级系统需求设计的 LED 显示模块和根据第二等级系统设计的模拟配电网络模块等。

## 第五章 系统软件设计

### 5.1 数据采集模块软件设计

数据采集模块的处理器是 STC 的单片机，所以该部分的软件设计在于单片机的软件编程。根据该模块所需实现的功能，其软件部分可以设计为三个功能块：A/D 功能块、串口通信功能块以及电压的傅立叶计算功能块。

#### 5.1.1 A/D 功能块

A/D 功能块程序包括了两个部分，第一部分是 A/D 通道选择程序，第二部分则为 A/D 转换程序。

A/D 通道选择程序首先是通过 P2.0、P2.1、P2.2 三个 I/O 口的输出控制 74HC155，选通待输入模拟输入信号所在的 CD4067B，再通过控制 P2.4、P2.5、P2.6、P2.7 选通该输入信号在该 CD4067B 的位置。比如：要选通第二个 CD4067B 上的第 9 路信号为 A/D 输入，首先就要将 P2.2、P2.1、P2.0 三个 I/O 输出为“001”来选通第二块 CD4067B，然后将 P2.7、P2.6、P2.5、P2.4 输出为“1001”选定为第 9 通道。

A/D 转换程序重点在于理清 STC12C5410AD 中与 A/D 转换相关的寄存器。其主要的相关寄存器包括以下几个：

1、P1M0 & P1M1：用来对 P1.X 的 IO 口模式进行设定，当需要将其做 A/D 使用时，应将其 IO 口模式设定为开漏模式或高阻输入；

2、ADC\_DATA & ADC\_LOW2：A/D 转换结果特殊功能寄存器。由于 STC12C5410AD 是 10 位精度的 A/D 转换器，因此，在转换结束后，其结果的高八位将存储在 ADC\_DATA 中，其转换结果的低两位将存储在 ADC\_LOW2 的低两位里。A/D 转换结果的计算公式为：结果(ADC\_DATA[7:0], ADC\_LOW2[1:0]) =  $1024 \times V_{IN}/V_{CC}$ 。其中  $V_{IN}$  为模拟输入通道输入电压， $V_{CC}$  为单片机实际工作电压。

3、ADC\_CONTR：A/D 转换控制特殊功能寄存器位对应功能如表 5-1 所示。

表 5-1 ADC\_CONTR 寄存器位结构

位	7	6	5	4	3	2	1	0
	ADC_POWER	SPEED1	SPEED0	ADC_FLAG	ADC_START	CHS2	CHS1	CHS0

CHS2/CHS1/CHS0：模拟输入通道选择；

ADC\_START：模数转换器（ADC）转换启动控制位，设置为“1”时，转换启动，转换结束后为 0；

ADC\_FLAG：模数转换器（ADC）转换结束标志位，当 A/D 转换完成后，

ADC\_FLAG=1; 要由软件清 0;

SPEED1/ SPEED0: 模数转换器转换速度控制位。

ADC\_POWER: ADC 电源控制位。“0”表关闭 ADC 电源, “1”则是打开其电源; 启动 A/D 转换前一定要确认已打开 ADC 电源, A/D 转换结束后可关闭该电源以降低功耗; 初次打开该电源时, 应适当延时, 等内部模拟电源稳定后, 再启动 A/D 转换, 且尽量在转换过程中不要改变所有 I/O 口状态, 以提高 A/D 转换的精度。

A/D 转换的参考程序如下:

```
*****STC12C5410AD*****
*****A/D 转换程序*****
*****设置 P1.0 为模拟输入口*****
.....
ADC_CONTR |= 0X80;      //打开 ADC 电源;
Delay_ms(1);           //延时 1ms, 等内部模拟电源稳定;
P1M0 |= 0X01;
P1M1 |= 0X01;           //设置 P1.0 所在 AD 通道为开漏模式;
ADC_CONTR = 0XE0;       //选定 P1.0 对应的模拟输入通道, 将转换速度设为
                        //最高的 210 个时钟周期转换一次;
ADC_CONTR |= 0X08;       //开始转换;
While((ADC_CONTR & 0X10)==0); //等待转换完成;
RESULT0 = ADC_DATA;
RESULT1 = ADC_LOW2;     //保存转换结果;
.....
```

### 5.1.2 串口通信功能块

以定时器 1 作为波特率发生器为例, 串口通信模块的基本程序可以参考如下:

```
*****STC12C5410AD*****
*****串口通信程序*****
*****设置 T1 为波特率发生器*****
.....
*****RS232 初始化程序*****
void rs232_init(void)
{
    ES = 0;           //禁止串口中断;
```

```

TMOD = 0X20;          //设置 T1 为波特率发生器;
SCON = 0X50;          //8 位数据, 无奇偶校验位;
TH1  = 0XFB;
TL1  = 0XFB;          //18.432MHz 晶振, 波特率=9600bps;
TR1  = 1;              //启动定时器 1;
}
.....
*****RS232 发送程序*****
void rs232_send(void)
{
    TI = 0;
    SBUF = SEND_DATA;
    while(TI==0);
    TI = 0;
}
.....

```

### 5.1.3 电压的傅立叶算法

首先来了解一下傅立叶算法的原理<sup>[52]</sup>, 我们知道关于标准的正弦电压主要有三个参数: 电压峰值、周期以及相位。

在这里设输入电压为峰值  $U$ 、角频率  $\omega$  的正弦电压, 即

$$\begin{aligned}
 u(t) &= U \times \sin(\omega t + \varphi) \\
 &= U \times \sin(\omega_0 t + \psi)
 \end{aligned} \tag{5-1}$$

式中,  $\psi = (\omega - \omega_0)t + \varphi = \Delta\omega t + \varphi = 2\pi\Delta f t + \varphi$

$u(t)$  可用矢量  $\bar{U}$  的虚部表示

$$\begin{aligned}
 \bar{U} &= A e^{j\psi} e^{j\omega_0 t} \\
 &= A [\cos \psi + j \sin \psi] [\cos \omega_0 t + j \sin \omega_0 t] \\
 &= A \cos \psi \cos \omega_0 t - A \sin \psi \sin \omega_0 t + j [A \sin \psi \cos \omega_0 t + A \cos \psi \sin \omega_0 t]
 \end{aligned} \tag{5-2}$$

若将  $A e^{j\psi}$  看作是  $\bar{U}$  的复数振幅  $u_m$

$$u_m = A e^{j\psi} = A \cos \psi + j A \sin \psi \tag{5-3}$$

对  $u(t)$  信号每周期采样  $N$  (如本设计中  $N=16$ ) 次产生采样序列  $\{u_k\}$

$$u_k = A \sin(2\pi f_0 k \frac{T_0}{N} + \psi) = A \sin\left(\frac{2\pi}{N} k + \psi\right) \tag{5-4}$$

对  $\{u_k\}$  进行离散傅立叶变换, 得到基波分量的频谱系数

$$\begin{aligned}
 u_k(k) &= \frac{2}{N} \sum_{k=0}^{N-1} u_k e^{-j \frac{2\pi k}{N}} \\
 &= \frac{2}{N} \sum_{k=0}^{N-1} u_k \cos \frac{2\pi}{N} k - j \frac{2}{N} \sum_{k=0}^{N-1} u_k \sin \frac{2\pi}{N} k \\
 &= U_I - j U_R
 \end{aligned} \tag{5-5}$$

式中

$$U_I = \frac{2}{N} \sum_{k=0}^{N-1} u_k \cos \frac{2\pi}{N} k \tag{5-6}$$

$$U_R = \frac{2}{N} \sum_{k=0}^{N-1} u_k \sin \frac{2\pi}{N} k \tag{5-7}$$

即假如每个周期里在特定的相位对电压均匀采样 16 次, 可得到各采样点的参数对应表如表 5-2 所示。

表 5-2 (1) 每周期 16 次采样对应傅立叶参数表

N	0	1	2	3	4	5	6	7
Angle	0	$\pi/8$	$\pi/4$	$3\pi/8$	$\pi/2$	$5\pi/8$	$3\pi/4$	$7\pi/8$
Sin	0	0.383	0.707	0.924	1	0.924	0.707	0.383
Cos	1	0.924	0.707	0.383	0	-0.383	-0.707	-0.924

表 5-2 (2) 每周期 16 次采样对应傅立叶参数表

N	8	9	10	11	12	13	14	15
Angle	$\pi$	$9\pi/8$	$5\pi/4$	$11\pi/8$	$3\pi/2$	$13\pi/8$	$7\pi/4$	$15\pi/8$
Sin	0	-0.383	-0.707	-0.924	-1	-0.924	-0.707	-0.383
Cos	-1	-0.924	-0.707	-0.383	0	0.383	0.707	0.924

根据式 5-6、5-7, 可以得出  $U_I$ 、 $U_R$  的计算公式

$$\begin{aligned}
 U_I &= [U_0 - U_8 + 0.707 \times (U_2 - U_6 - U_{10} + U_{14}) + 0.383 \times (U_3 - U_5 - U_{11} + U_{13}) \\
 &\quad + 0.924 \times (U_1 - U_7 - U_9 + U_{15})] \div 8
 \end{aligned} \tag{5-8}$$

$$\begin{aligned}
 U_R &= [U_4 - U_{12} + 0.707 \times (U_2 + U_6 - U_{10} - U_{14}) + 0.383 \times (U_1 + U_7 - U_9 - U_{15}) \\
 &\quad + 0.924 \times (U_3 + U_5 - U_{11} - U_{13})] \div 8
 \end{aligned} \tag{5-9}$$

该式只需进行 11 次加减与四次乘法, 运算量不大。又因为对于正弦输入信号, 有

$$U_I = U \sin \psi \tag{5-10}$$

$$U_R = U \cos \psi \tag{5-11}$$

即可以根据该  $U_I$ 、 $U_R$  的值求出采样电压峰值  $U$  及其偏差角度  $\psi$ 。

在计算  $U$  的时候, 由于  $U = \sqrt{U_I^2 + U_R^2}$  中涉及的计算需要开平方, 为节约计算时间, 也可使用  $A = L + S/3$  的近似算法, 其中  $L$  和  $S$  分别是两个数的绝对值中

较大 (L) 和较小 (S) 的那个。

计算  $\psi$  时, 可由  $\tan\psi = U_I/U_R$  得出角度的正切值, 然后通过查表或利用 C 语言库函数中的求反正切函数, 或者参考文献[53]中所示的将求反正切函数转移为求反正弦函数的方法。

连续取样两个周期, 即一次性采取 32 个点后, 求得  $\psi_1$ 、 $\psi_2$ , 并设  $f_0$  为 50HZ, 可得  $\Delta f$  的计算公式:  $\Delta f = (\psi_1 - \psi_2) \cdot f_0 / 2\pi$ , 则频率  $f = f_0 + \Delta f$ 。

再以相同的方法计算一次只是将此次的  $f_0$  改为上式中计算所得的频率  $f$ , 使得频率的计算结果可自适应地调整, 实现高精度的频率测量。

傅立叶算法的部分程序如下:

```
*****STC12C5410AD*****
*****电压傅立叶采样程序*****
*****设置每周期采样 16 个点*****
.....
*****计算出各个点的正弦、余弦值*****
void sc_cal(void)
{
    for(i=0;i<16;i++)
    {
        cosij[i] = cos(i*PI/16);
        sinij[i] = sin(i*PI/16);
    }
}

*****采样数据计算部分*****
void calculate(void)
{
    Uchar i;
    //float a1,a2,b;
    IE = 0x80;           //开总中断
    LOGIC = 1;
    while (samplepoint<16); //16 点采样完成?
    IE = 0x00;           //关中断
    LOGIC = 0;
    samplepoint = 0;      //采样点标志置零
```

```

for(i=0;i<16;i++)
{
    if(i<7)                                //判断正负
        addata[i]= bakdata[i] *5.0/256.0;    //正
    else
        addata[i]=- bakdata[i] *5.0/256.0;  //负
}
ur = 0.0;
ui = 0.0;
for(i=0;i<16;i++)
{
    ur+=addata[i] *cosij[i]/8/sqrt(2.);  //计算实部电压
    ui+=addata[i] *sinij[i]/8/sqrt(2.);  //计算虚部电压
}
angu = atan2(ui,ur);                      //相位
realdata =  sqrt(ur*ur+ui*ui);            //电压峰值
}

```

## 5.2 重构执行模块软件设计

在重构执行模块的软件设计中，最主要的工作是在 FPGA 上实现 RS232 的功能。即为其在 FPGA 的系统程序中为其添加一个 UART 的通信模块。

首先，在硬件描述语言方面，本方案选择了 Verilog HDL。Verilog HDL 是在 1983 年由 GDA (Gate Way Design Automation) 公司的 Phil Moorby 首创的。Phil Moorby 后来成为 Verilog-XL 的主要设计者和 Cadence 公司的第一个合伙人。随着 Verilog-XL 算法的成功，Verilog HDL 语言得到迅速发展。IEEE 于 1995 年制定了 Verilog HDL 的 IEEE 标准，即 Verilog HDL 1364-1995。

相比于 VHDL，Verilog HDL 在系统的抽象方面比 VHDL 强一些，Verilog HDL 较为适合算法级 (Alogrithem)，寄存器传输级 (RTL)，逻辑级 (Logic)，门级 (Gate) 设计，而 VHDL 更为适合特大型的系统级 (System) 设计。

根据串口的功能，将整个系统划分为 4 个模块：顶层模块、波特率发生模块、接收模块以及发送模块。下面具体就后四个模块作进一步介绍：

### 1、波特率发生模块

波特率发生器其实就是一个简单的分配器，根据给定的系统频率(晶振频率)和要求的波特率计算出波特率的分频因子。

### 2、接收模块

接收模块在监测到 RS-232 总线上的数据，顺序读取串行数据并且将其输出给 CPU 的过程。当信号监测器监测到新的数据（RS-232 输入逻辑变为 0，即 RS-232 传输协议的起始位）就会触发接收过程，接收模块数据程序流程图如图 5-1 所示。

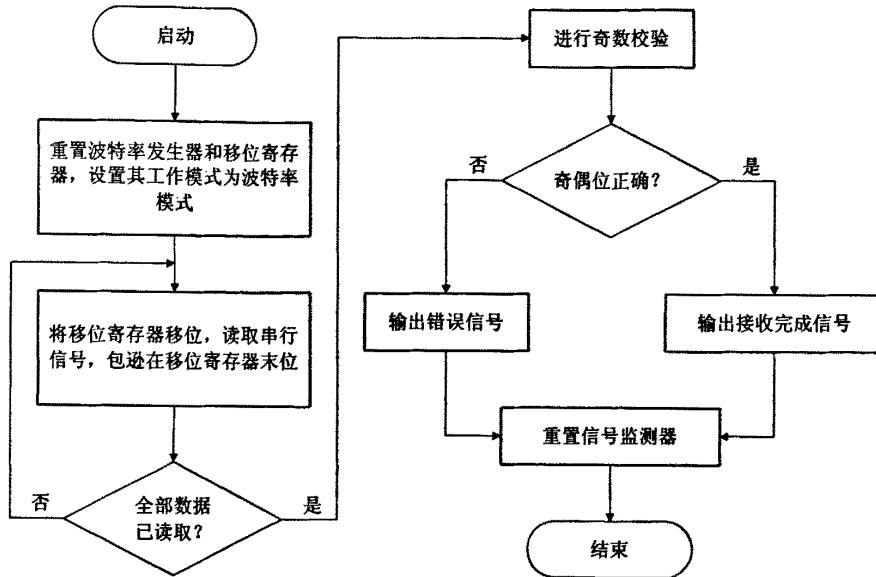


图 5-1 UART 接收数据流程图

首先 UART 内核会重置波特率发生器和移位寄存器，并且设置移位寄存器的工作模式为波特率模式，以准备接收数据。其次，移位寄存器在波特率时钟的驱动下工作，不断读取 RS-232 串行总线的输入数据，并且将数据保存在内部的寄存器内。接收完成后，UART 内核会对已接收的数据进行奇偶校验并且输出校验结果。最后，UART 内核会重置信号监测器，以准备进行下一次数据接收。

接收模块 Verilog HDL 程序见附录 4.1 所示，程序中，`rst` 是系统复位信号，`clk16x` 是波特率发生器产生的分频时钟，`rx` 是来自其他设备的串行数据，`data_ready` 表示数据已准备好，在 `rdn` 为 1 的情况下，并行数据转换完毕就可以输出，此时将 `data_ready` 置 0，在 `RDN` 不为 0 的情况下，串并转换完成后将此 `data_ready` 置 1，以表示数据已经转换完毕，可以输出了，`dout[7:0]` 是接收模块将收到的串行数据转换成的并行数据。

### 3、发送模块

发送过程是由加载和发送两个步骤组成，加载步骤是 UART 内核按 RS-232 串行发送的顺序将起始位、数据位、奇偶校验位和停止位加载到移位寄存器内，这个过程工作在系统时钟下，相对于 RS-232 的传输速度来说非常快。完成加载步骤后，UART 内核会重置波特率发生器，并且设置移位寄存器工作在波特率模式下，于是移位寄存器便在波特率时钟的驱动下依次将加载的数据发送到

RS-232 的发送端 Txd, 这样便产生了 RS-232 的数据发送时序。其发送数据流程图如图 5-2 所示。

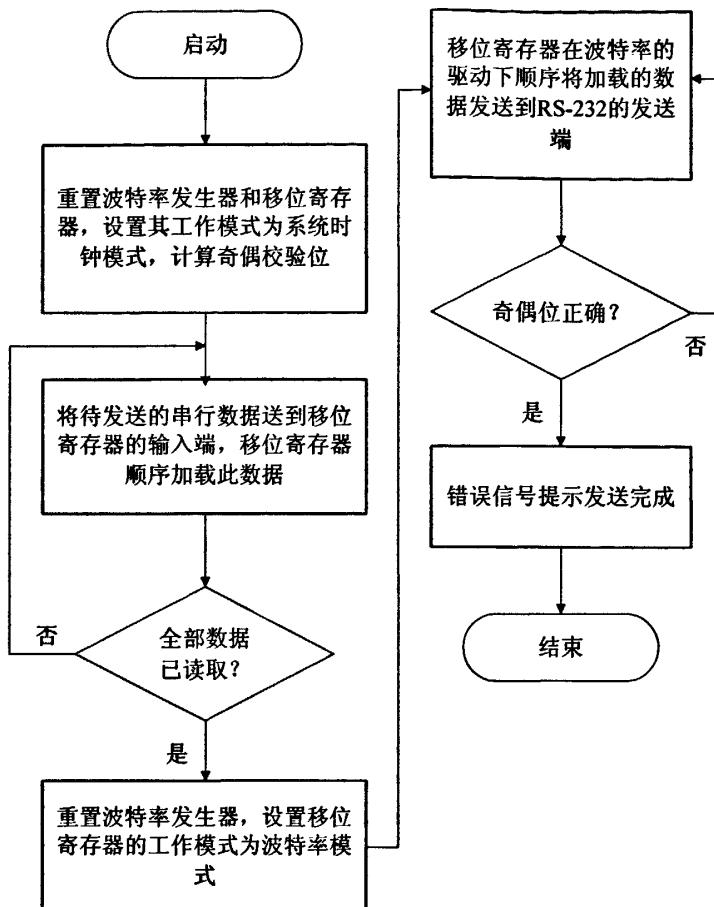


图 5-2 UART 发送数据流程图

发送模块 Verilog HDL 程序见附录 4.2 所示, 其中 `rst`, `clk16x` 信号同接收模块, `din[7:0]` 是即将发送出去的并行数据, `wrn` 是数据发送的启动信号, `txd` 是发送出去的串行数据信号。该程序每隔 16 个 `clk16x` 周期从 `txd` 发送一位数据, 次序依次是 1 个起始位、8 个数据位、1 个停止位。

### 5.3 嵌入式 Linux 下设备应用开发

在本系统的嵌入式 Linux 设备应用程序的开发中, 最重要的就是串行通信的应用程序开发, 其在配电网重构的应用程序中起着十分重要的作用。

串口通信包括异步通信和同步通信两种通信方式, 但最常用的是异步通信。异步通信可以随时进行数据通信, 需要通信双方知道如何判别数据的起始和终止, 所以发送的数据帧由起始位、数据位、校验位和停止位构成, 各部分的单位是位(bit)。起始位总是 space(0)。

Linux 下的串口编程主要包括四个部分，依次是打开串口、设置串口、串口通信（发送或接收）、关闭串口退出。

### 1、打开串口

程序通过 `open` 系统调用来打开要使用的串口，以设备名为 `ttyS1` 的系统第二串口为例，第二个串口对应的设备名为 `/dev/ttyS1`。在 Linux 下串口作为系统的串行终端来使用。所以在 Linux 下，串口设备的主设备号为终端设备的主设备号 4，次设备号默认是从 64 开始。也就是说第一个串口对应的设备文件为 `ttyS0`，主设备号是 4，次设备号是 64，第二个串口对应的设备文件为 `ttyS1`，主设备号是 4，次设备号是 65，以此类推。

`Open` 的第二个参数是串口设备打开的标志，一般为可读写，能发送和接收数据 `O_RDWR`，以及非控制台 `O_NOCTTY`，不响应 `Ctrl-C` 等特殊字符。具体取值可参见 `open` 系统调用说明。

打开串口是通过标准的文件打开函数来实现的，程序如下：

```
int fd;
fd = open( "/dev/ttyS0", O_RDWR); /*以读写方式打开串口*/
if (-1 == fd){ /* 不能打开串口 */
    perror(" 提示错误！ ");
}
```

### 2、设置串口

在串口设备打开成功后，需要对串口进行设置，这也是串口编程的主要内容。在 Linux 下，每个串口对应于一个结构 `termios`，具体定义在 `include/asm/termbits.h` 中，如下：

```
Struct termios{
    tcflag_t c_iflag;          /* 输入模式标志 */
    tcflag_t c_oflag;          /* 输出模式标志 */
    tcflag_t c_cflag;          /* 控制模式标志 */
    tcflag_t c_lflag;          /* 本地模式标志 */
    cc_t c_line;               /* 行规程 */
    cc_t c_cc[NCCS];           /* 控制字符 */
}
```

这个结构中的成员的设置是比较复杂的，关于其具体取值对应的含义可参考相关资料，我这里就只考虑常见的一些设置：

#### (1) 波特率设置

下面是修改波特率的代码：

```
{  
    struct termios Opt;  
    tcgetattr(fd, &Opt);  
    cfsetispeed(&Opt,B19200); /*设置为 19200Bps*/  
    cfsetospeed(&Opt,B19200);  
    tcsetattr(fd,TCANOW,&Opt);  
}
```

## (2) 校验位和停止位的设置

无效验 8 位

```
{  
    Option.c_cflag &= ~PARENB;  
    Option.c_cflag &= ~CSTOPB;  
    Option.c_cflag &= ~CSIZE;  
    Option.c_cflag |= ~CS8;  
}
```

奇效验(Odd) 7 位

```
{  
    Option.c_cflag |= ~PARENB;  
    Option.c_cflag &= ~PARODD;  
    Option.c_cflag &= ~CSTOPB;  
    Option.c_cflag &= ~CSIZE;  
    Option.c_cflag |= ~CS7;  
}
```

偶效验(Even) 7 位

```
{  
    Option.c_cflag &= ~PARENB;  
    Option.c_cflag |= ~PARODD;  
    Option.c_cflag &= ~CSTOPB;  
    Option.c_cflag &= ~CSIZE;  
    Option.c_cflag |= ~CS7;  
}
```

Space 效验 7 位

```
{  
    Option.c_cflag &= ~PARENB;
```

```
Option.c_cflag &= ~CSTOPB;  
Option.c_cflag &= &~CSIZE;  
Option.c_cflag |= CS8;  
}
```

### (3) 设置停止位

1 停止位:

```
options.c_cflag &= ~CSTOPB;
```

2 停止位:

```
options.c_cflag |= CSTOPB;
```

### 3、串口通信

串口通信实际上就是对串口设备文件进行读写调用，从串口读数据使用 `read` 系统调用，往串口发送数据则使用 `write` 系统调用。

#### (1) 发送数据

```
{  
    char buffer[1024];  
    int Length=1024;  
    int nByte;  
    nByte = write(fd, buffer ,Length);  
}
```

#### (2) 读取串口数据

如果设置为原始模式(Raw Mode)传输数据，那么 `read` 函数返回的字符数是实际串口收到的字符数。可以使用操作文件的函数来实现异步读取，如 `fcntl`，或者 `select` 等来操作。

```
char buff[1024];  
int Len=1024;  
int readByte = read(fd, buff, Len);
```

### 4、关闭串口

在关闭串口之前，需要通过

```
tcsetattr(fd,TCSANOW,&oldtio);
```

将串口设置恢复为原来的设置，最后通过 `close` 系统调用来关闭串口。

```
close(fd);
```

串口通信过程也就此结束。

## 5.4 本章小结

本章从软件的层面详细解析了系统各个功能模块的设计，包括数据采集模块

中 A/D 处理、串口以及电压傅立叶算法的单片机程序设计；重构执行模块中核心的 UART 通信模块的 Verilog HDL 程序设计；嵌入式 Linux 系统中串口应用程序的开发等。

## 第六章 网络重构算法实现

通过前几章的描述，一个可用于模拟配电网络重构的实验装置平台已经被搭建了起来。下面，我们将就在这个平台上模拟实现配电网络重构算法进行进一步的研究。

配电网络重构程序运行于嵌入式系统中，以应用程序的方式运行。其主要组成模块除了 5.2 节所介绍的串口通信程序外，还包括与配电网络重构相关的拓扑识别与潮流计算子程序、优化算法程序等。各程序的体系结构及调用关系如图 6-1 所示。

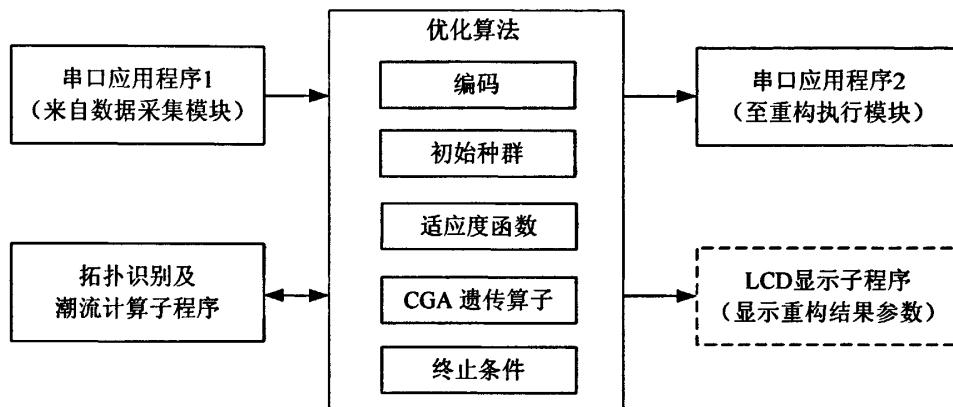


图 6-1 程序结构框图

首先，由来自数据采集模块的命令启动优化算法程序，优化算法采用的是克隆遗传算法 (CGA)，在优化算法计算得出新一代种群后，系统调用拓扑识别与潮流计算子程序，计算得出其每个新生个体的适应度函数。通过一定代数的遗传进化后，即可得到适应度函数的最大值以及其所对应的染色体，该染色体对应的编码含义即配电网络所对应的最佳重构拓扑结构。最后，系统将重构指令通过串口发送至重构执行模块，并可将相关的重构结果参数如网络损耗、最低节点电压、停电户数等显示在其 LCD 屏上。

下面，将研究的重点集中在拓扑识别及潮流计算以及优化算法这两方面：

### 6.1 拓扑识别及潮流计算

网络损耗是配电网络重构中常用的目标函数之一，要计算配电网络的损耗，就需要得到网络中各节点和支路的电压、电流等数据。此外，为了重构后网络中各节点电压质量的保证以及各支路电流不致太大，也要对重构方案中各节点及支路的电压、电流数据事先进行计算。因此，网络的潮流计算是重构算法中十分重要的一个环节。

本文采用前推回代法计算配电网络潮流，前推回代法的基本原理是：根据网络各节点的已知负荷及设定的各节点电压，根据基尔霍夫电流定律 KCL 由网络末端向首端计算各支路电流，称为前推；根据已知根节点的电压、各支路阻抗及已求得的支路电流，按基尔霍夫电压定律 KVL 由网络首端向末端计算各节点电压及电压损耗，称为回代。

前推回代潮流计算时要求从配电网末梢一级负荷节点往前依次计算各节点和支路的电压、电流等数据，所以计算前要确定配电网层次关系和拓扑结构表。对于配电网重构，每产生一个新的开关组合状态，所对应的配电网结构各有不同，即配电网拓扑结构各不相同。因此潮流计算前，首先要确定配电网拓扑结构和层次关系。

为了配合算法和避免复杂的网络编号，本文建立了以下原始数据输入结构。使用这一套数据结构，不用形成节点导纳矩阵，就可以自动搜索节点关系，确定网络的拓扑结构。

节点结构体：

{节点号 节点有功功率 节点无功功率}

支路结构体：

{支路首端节点号 支路末端节点号 支路电阻 支路电抗}

根据线路首末节点就可以确定每个节点连接的节点及其关系，从而可以形成整体呈树状的关系结构。下面以图 6-2 所代表的一个简单的 7 节点配电网络来说明。

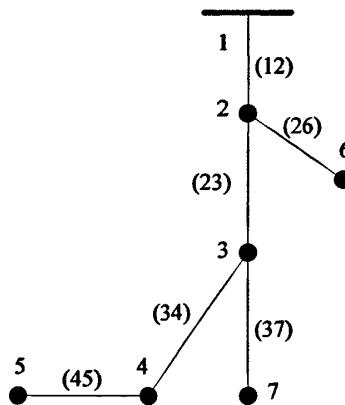


图 6-2 7 节点配电网络结构

首先要确定拓扑结构表，了解所有节点的连接关系。以图 6-2 中的节点 2 来说，搜索所有线路的首末节点，发现它是线路 12 的末节点，同时它又是线路 26 和线路 23 的首节点。那么，可以确定节点 1 是节点 2 的父节点，节点 2 是节点 1 的子节点，同时，节点 2 又是节点 6 和 5 的父节点。由此可以找到所有

节点的连接节点及其连接关系, 如表 6-1 所示。

表 6-1 图 6-2 配电网拓扑结构表

节点号	连接节点数	连接节点号	连接关系
			(S 为子节点, F 为父节点)
1	1	2	S
2	3	1, 3, 6	F, S, S
3	3	2, 4, 7	F, S, S
4	2	3, 5	F, S
5	1	4	F
6	1	2	F
7	1	3	F

有了表 6-1 的拓扑结构, 还需经过一次按层遍历的广度优先搜索, 形成层次关系, 确定前推回代潮流算法的节点计算顺序。结合图 6-2 配电系统, 介绍具体搜索步骤如下:

1、搜索末梢节点作为第一层次节点, 研究拓扑结构表, 找到节点 5、6、7 都只有一个连接节点, 而且连接节点是父节点, 则这 3 个节点是末梢节点, 放在第一层。

2、搜索末梢节点的父节点作为第二层节点, 可知节点 5、6、7 的父节点分别为节点 4、2、3, 那么节点 4、2、3 为第二层节点。

3、继续搜索第二层节点的父节点作为第三层节点, 这样一层层地搜索下去, 直到搜索到某层节点的父节点全部是根节点 1 时, 停止搜索。形成以下的层次结构:

第一层节点: 5、6、7

第二层节点: 4、2、3

第三层节点: 3、1、2

第四层节点: 2、1

第五层节点: 1

4、上面的层次中有重复节点, 需要删除在后面层次中有重复的前面层次中的节点, 形成真正的层次关系, 确定潮流计算的节点顺序。如第二层中的节点 2、3 在第三层中存在, 则应该删除第二层中的节点 2 和 3, 同样应该删除第三层中的节点 1、2 和第四层中的 1, 最后得到真正层次如下:

第一层节点: 5、6、7

第二层节点: 4

第三层节点: 3

第四层节点: 2

第五层节点: 1

有了配电网层次关系之后, 就可以用前推回代法进行配电网潮流计算了。

下面介绍前推回代法迭代过程:

前推回代法迭代步骤如下<sup>[54][55]</sup>:

步骤①: 初始化数据, 给负荷节点电压赋初值  $\dot{U}_i^{(0)}$  ( $i=1,2,3\dots n-1$ ), 一般在额定电压附近。计算潮流时设电源点(根结点)为松弛节点, 其电压大小已知, 电压相角为零; 设全网其余节点电压均为额定值。

步骤②: 从末梢一级负荷节点开始, 计算流入该节点的支路电流  $\dot{I}_{i,j}^{(0)}$

$$\dot{I}_{i,j}^{(0)} = \left( \frac{S_j}{\dot{U}_j^{(0)}} \right)^* = \frac{P_j - jQ_j}{\dot{U}_j^{(0)*}} \quad (6-1)$$

式中,  $j$  为最末一级负荷节点号;  $i$  为相应支路的首端节点号。

步骤③: 对于非末梢节点, 根据基尔霍夫电流定律应等于式(6-1)计算结果与流出电流之和, 前推求支路电流

$$\dot{I}_{i,j}^{(0)} = \left( \frac{S_j}{\dot{U}_j^{(0)}} \right)^* + \sum \dot{I}_{j,k}^{(0)} \quad (6-2)$$

式中,  $j$  为非末梢负荷节点号,  $i$  为该节点的父节点;  $k$  为子节点号; 即

$$\dot{I}_{i,j} = \sum_{k \in d} \frac{P_k - jQ_k}{\dot{U}_k^*} \quad (6-3)$$

式中,  $d$  为流过支路  $b_{i,j}$  的所有负荷节点的集合。假定集合  $d$  中的节点电压均近似等于支路  $b_{i,j}$  末节点电压  $\dot{U}_j$ , 则式(6-3)可以简化为

$$\dot{I}_{i,j} = \frac{\sum_{k \in d} (P_k - jQ_k)}{\dot{U}_j^*} \quad (6-4)$$

步骤④: 由步骤②和③可求出所有支路的支路电流, 再利用根节点电压, 则可从前向后顺次求得各个负荷节点的电压, 回代求节点电压

$$\dot{U}_j = \dot{U}_i - \dot{I}_{i,j} (R_{i,j} + jX_{i,j}) \quad (6-5)$$

式中  $R_{i,j} + jX_{i,j}$  为该支路的等效阻抗, 如图 6-3 所示。

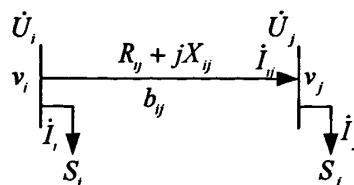


图 6-3 支路等效电路

步骤⑤：计算各个负荷节点的电压幅值修正量  $\Delta U_j^{(1)}$

$$\Delta U_j^{(1)} = |\dot{U}_j^{(1)} - \dot{U}_j^{(0)}| \quad (6-6)$$

步骤⑥：计算节点电压修正量的最大值  $\max(\Delta U_j^{(1)})$

步骤⑦：判别收敛条件

$$\max(\Delta U_j^{(1)}) < \varepsilon \quad (6-7)$$

其中上标1表示第1次迭代， $\varepsilon$  为电压修正量阈值。如果满足收敛条件，则跳出循环；否则转步骤②，进行下一次迭代。

这样，在已知根节点电压  $\dot{U}_0$  和各负荷节点的取用负荷  $\dot{S}_i$  ( $i=1,2,3\dots n-1$ ) 的情况下，就求得了每条线路的功率潮流、线路电流、各个负荷节点的电压  $\dot{U}_i$  ( $i=1,2,3\dots n-1$ )。

前推回代法潮流计算流程如图 6-4 所示。

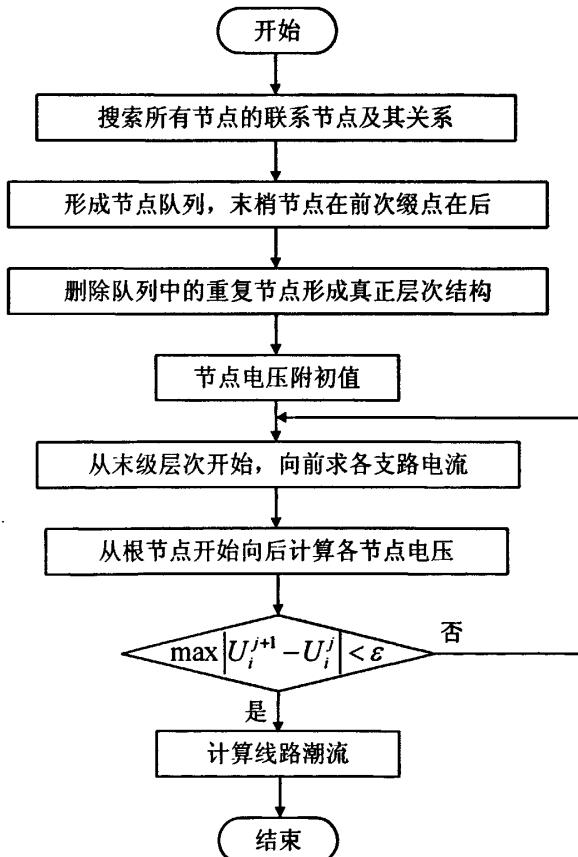


图 6-4 前推回代法流程图

## 6.2 配网重构优化算法

一个全局极小化问题可以化为一个( $S$ ,  $f$ )对的形式, 其中  $S$  是  $n$  维欧氏空间的某一区域,  $f$  是  $n$  维实值函数。所要求解的问题就是要找一点  $x_{\min} \in S$ , 使得  $f(x_{\min})$  是  $S$  上的全局极小值, 即满足

$$\forall x \in S : f(x_{\min}) \leq f(x) \quad (6-8)$$

遗传算法是一种基于自然选择和群体遗传机理的搜索算法, 它模拟了自然选择和自然遗传过程中发生的繁殖、杂交和突变现象。在利用遗传算法求解问题时, 问题的每个可能的解都被编码成一个“染色体”, 即个体。几个个体构成了群体(所有可能解)。在遗传算法开始时, 总是随机的产生一些个体(即初始解), 根据预定的目标函数对每个个体进行评价, 给出了一个适应度值, 基于此适应度值, 选择个体用来复制下一代。选择操作体现了“适者生存”原理, “好”的个体被选择用来复制, 而“坏”的个体则被淘汰。然后选择出来的“好”的个体经过交叉和变异算子进行再组合生成新一代。这一群新个体由于继承了上一代的一些优良性状, 因而在性能上要优于上一代, 这样逐步朝着更优解的方向进化<sup>[56-58]</sup>。

遗传算法求解函数  $f(x)$  极小值(极大值可以通过改变符号按同样的方式找到)的主要过程是: 将  $S$  上的点  $x$  编码为数字串(称为染色体), 染色体的适应函数  $F(x)$  则表示成与目标函数  $f(x)$  有关的函数, 在最初随机产生  $N$  个染色体  $x(1,0) \cdots x(N,0)$  的基础上, 根据各染色体的适应函数执行复制、交叉和变异等遗传操作, 产生下一代染色体  $x(1,k) \cdots x(N,k)$  ( $k$  表示代数, 初始代  $k=0$ )。

交叉和变异操作是分别以概率  $p_c$  交换两个父代染色体之间的对应位和以概率  $p_m$  改变染色体上的某些位而产生更加优秀的染色体。随着遗传次数的增加, 将会出现一批适应函数值很高的染色体, 把这些染色体解码还原就可以得到原问题的解。从理论上讲, 当  $N$  足够大和  $k$  足够多时, 遗传算法可以给出原问题的最优解。

本系统采用的是克隆遗传算法(CGA), 也叫单亲遗传算法, 其基本思想是取消传统遗传算法的交叉算子, 全部遗传操作都只在一个个体上进行, 通过基因重组算子和基因突变算子实现个体的进化。其优点在于遗传操作简单, 计算效率高, 并且不要求初始群体中个体的多样性, 也不存在“早熟收敛”问题<sup>[59][60]</sup>。

传统遗传算法的遗传算子主要是选择(复制)、交叉(交换)和变异(突变), 此外还有显性、倒位等高级遗传算子。除了交叉算子以外, 传统遗传算法所使用的遗传算子都可以作为克隆遗传算法的遗传算子, 还可以根据克隆遗传算法自身的特点构造一些遗传算子, 特别是可以针对具体问题构造克隆遗传算法的遗传算子。近年来, 已经有学者将克隆遗传算法用于配电网重构<sup>[61]</sup>, 并取得了一

定的成果。

克隆遗传算法的遗传算子主要有基因重组算子和基因突变算子两大类。基因重组算子又包括基因换位算子、基因移位算子和基因倒位算子等<sup>[62]</sup>。

1、CGA 的基因换位算子是按一定的概率交换一条染色体中某些位置上的基因的过程，被交换基因的位置是随机的。

2、CGA 的基因移位操作是按一定的概率把一条染色体中的一个(些)子串中的基因依次后移，并把该子串的最后一个基因移到最前面的位置，在一条染色体中进行基因移位的子串及其长度是随机的。CGA 的基因移位操作也可使子串中的基因依次向前移动。

3、CGA 的基因倒位算子是按一定的概率把一条染色体中的某个子串中的基因依次首尾倒置的过程，被倒位的子串及其长度是随机选取的。

4、CGA 的基因突变算子是按一定的概率让一条染色体中的某个位置上的基因取其同序基因集中的其它值的过程，被改变基因值的位置是随机选取的。

对克隆遗传算法，相关的编码策略、种群初始化、适应度函数、停止条件等都对算法的最终效果起着十分关键的作用。

### 6.2.1 编码策略

遗传算法的一个显著特征是它交替地在编码空间和解空间中工作，它在编码空间对染色体进行遗传运算，而在解空间对解进行评估和选择。本文的克隆遗传算法同样存在这一特征。

利用克隆遗传算法进行问题求解时，首先确定问题的目标函数和变量，然后对变量进行编码。这样做主要是因为在克隆遗传算法中，问题的解是用数字串表示的，而且遗传算子也是直接对串进行操作。

编码方案的优劣是影响智能优化算法效率的主要因素。编码方式一般采用二进制编码。二进制编码方式有如下优点：

1、它与计算机码制相一致，适于计算机应用。对于串码的每一位只有 1 和 0 两个码值，在交叉和变异等操作中原理清晰，操作简单；

2、可表示的变量范围大。

但是在配电网络重构系统中，如果以二进制开关量表示染色体基因，染色体长度为网络开关数总和。这样，染色体长度会很长。且这样的编码存在大量的不可行解，大大减弱了重构搜索效率。

与二进制编码方案相比，序号编码方案具有实现简便、有效编码比例高等优点<sup>[63]</sup>。首先将全网的联络开关以自然数编号，并在每个联络开关所确定的环网内将开关单独编号(从 1 到小环里的开关总数)。编码以联络开关确定的小环网为基因位，该位上的值是该环网里打开的开关号，基因位上的取值是受限制

的，必须是正整数，并且从 1 到小环里的开关总数的任一个。染色体长度等于联络开关数(环网数)。

以图 6-5 所示的 IEEE-RBTS Bus2 系统为例说明，将图中开关任意编号。图中的联络开关是 S15、S16。

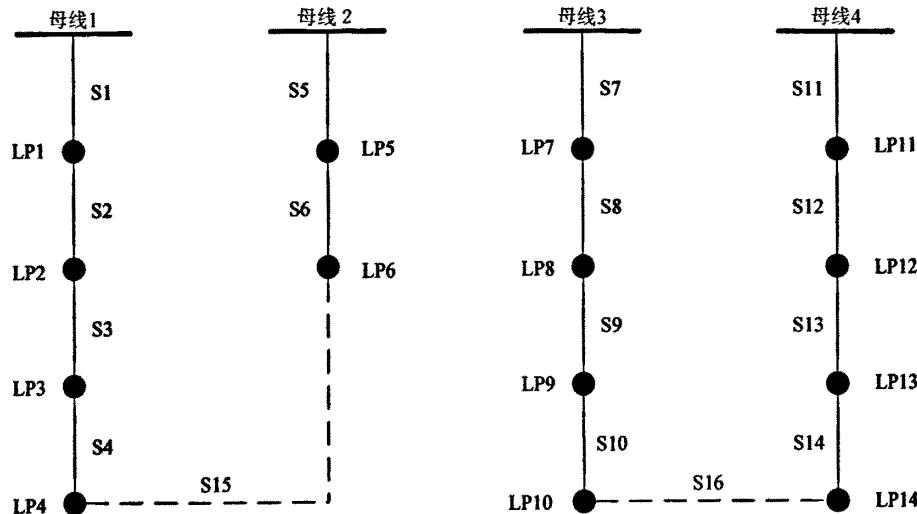


图 6-5 支路编号后的 IEEE-RBTS Bus2 系统

图 6-5 中有两个环网，将联络开关 S15 确定的环网定义为 1 号环网，由 S16 确定的环网定义为 2 号环网。然后分别对环网中的所有开关重新统一编号，联络开关本身定义为该环网中的最后一号。这样，由图 6-5 中各开关就可得到表 6-2 所示编码：

表 6-2 图 6-5 开关编码

环网	环内编号	1	2	3	4	5	6	7	8	9
1 号环网		S1	S2	S3	S4	S5	S6	S15		
2 号环网		S7	S8	S9	S10	S11	S12	S13	S14	S16

这样该配网染色体长度为 2，如果图 6-5 中联络开关全部打开，其余开关合上，则该网络对应的编码是(7, 9)。这样的编码规则覆盖了所有的开关组合状态，并且没有不可行解。满足了染色体和解之间唯一映射的要求。图 6-5 的系统用这种数字编码规则共有  $7 \times 9 = 63$  个解，而如果按二进制的编码规则，有  $2^{16} = 65536$  个解，不可行解占 99. 9%，很多计算都是无效的。

上述编码规则对单环网非常有效，但对于环路间含有公共开关的复杂配电网存在一些问题，如图 4-10 中 Bus4 里面由联络开关 S47 和 S49 所构成的环网中就共同拥有 S35、S36、S37 三个开关，当这三个开关中的两个分别被两个环网所开断时，就形成了不可行解，但是该部分编码在整个系统编码中只是一小

部分，本系统中，我们采用排除不可行解的方法将其剔除，即构建不可行解列表，当新生个体与不可行解列表中个体相同时，放弃该个体。

### 6.2.2 种群初始化

克隆遗传算法不依赖于初始值，因此可以随机产生  $x$  个长度为  $l$  ( $x$  为种群大小， $l$  为配电网中参与重构的环网数量的总和) 的序号 (自然数) 编码作为初始种群。 $x$  不能取太大或太小，太小不能让初始种群遍及整个解空间，影响收敛速度，太大则使运算时间增加。一般根据  $l$  的大小来确定种群大小。此外，初始种群的选择同样应保证在可行解的范围之内。

一般采取的方法是首先提炼出配电网中各环网的支路 (开关) 总数  $n(i)$ ，如图 6-5 的 1 号环网，则是  $n(1)=7$ 。然后随机地在  $(1, n(i))$  间产生一个自然数，将它作为对应环网  $i$  的编码，即该环网中所打开的联络开关号码。依此类推，选择剩余所有环网中被打开的开关号，直到打开的开关数等于环网数  $l$ ，这些被选中的号码构成的数组就生成了一条染色体。最后对该染色体的可行性进行判别，若可行，则保留，否则舍去，再重新生成染色体。

同理，生成种群规模数  $x$  个染色体，就得到了初始种群  $group(0)$ 。

此外，还需注意的是在模拟短路故障下的重构时，首先要求将向该节点供电的所有支路开关断开，并将因此被断开的环网剔除出来，针对剩余的环网进行种群初始化。

### 6.2.3 适应度函数

本文以可靠性指标是 FIC (故障停电用户数) 最少为目标函数进行重构实验，目标函数可以表示为

$$FIC = \sum \text{停电用户数} \quad (6-9)$$

式中，各节点所包含的用户数可参考附表 1。

同时，重构中各节点还要满足下列约束条件

$$U_{i_{\min}} \leq U_i \leq U_{i_{\max}} \quad (6-10)$$

$$I_i \leq I_{i_{\max}} \quad (6-11)$$

式中， $V_{i_{\min}}$ 、 $V_i$ 、 $V_{i_{\max}}$  分别为节点  $i$  电压的下限值、计算值、上限值； $I_i$ 、 $I_{i_{\max}}$  分别为支路  $i$  流过的电流及其最大容许值，需要调用潮流计算模块的计算结果。

FIC 最少为目标函数的适应度函数定义为

$$F(x) = f_{\max} - f(x) \quad (6-12)$$

式中， $f_{\max}$  为一大于初始目标函数的任意常数。

### 6.2.4 停止条件

每代群体中适应度最大的染色体即群体中最优良的染色体，通过优化算法

不断对这些个体进行进化以得到适应度最大及系统全局最优解。在实现过程中可以通过监控每代群体中出现的最大适应度值，当该最大适应度值连续  $N$  代都没有发生变化时，即可认为算法收敛并已经找到全局最优解，满足停止条件。

### 6.2.5 克隆遗传算法操作步骤

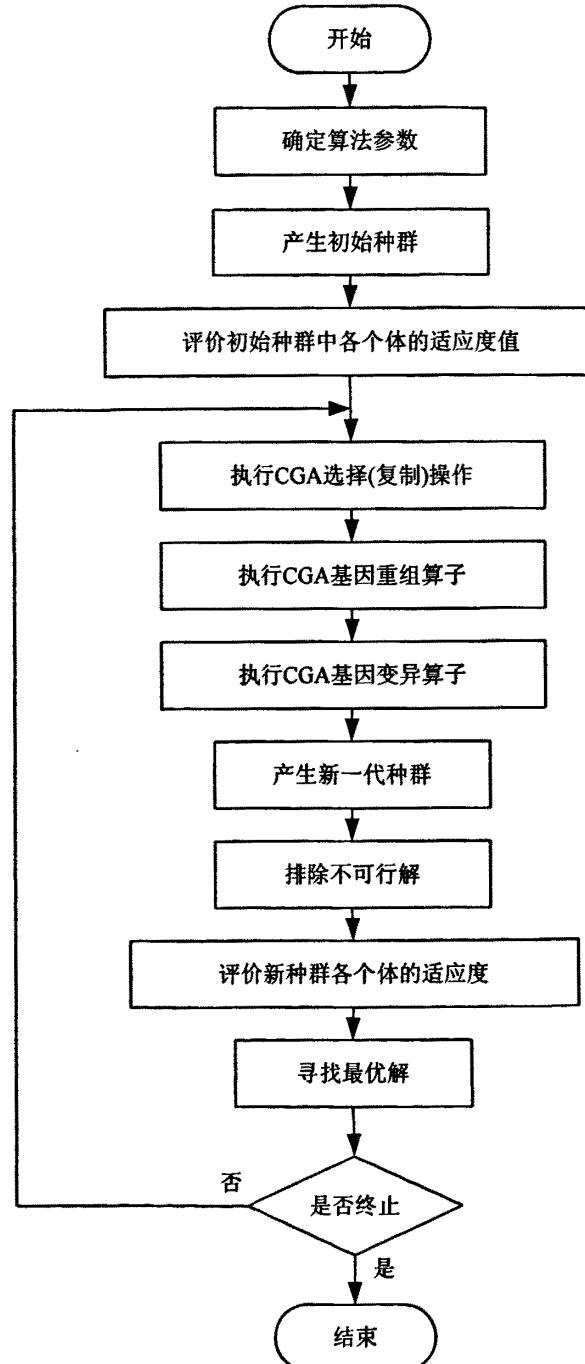


图 6-6 克隆遗传算法操作流程图

第一步：确定算法参数，包括群体规模  $x$ ，染色体长度  $l$  等。

第二步：产生初始群体  $group(0)$ 。

第三步：评价初始种群各个体的适应度值。

第四步：执行CGSA的选择(复制)操作以及以一定概率执行基因重组和基因突变操作，产生新一代种群  $group(i)$ 。

第五步：根据约束规则排除不可行解，如网络中某节点出现多条供电途径，即形成环网，或节点电压低于设定最小值  $V_{min}$  等。

第六步：评价当前种群各个体的适应度值。

第七步：寻找该种群中适应度值最高的个体，其染色体编码所对应的网络即为其中的最优网络结构。

第八步：判断是否达到终止条件，达到则则终止计算，否则返回第四步继续。

克隆遗传退火算法计算流程图如图 6-6 所示。

### 6.3 重构实验

首先，按照第一等级的实验方案搭建实验平台，由外部接口或数据采集模块发出故障节点信息，嵌入式系统平台收到信号后，马上启动 6.1.1 及 6.1.2 节所介绍的以故障停电用户数(FIC)最少为目标函数的配电网可靠性重构算法，模拟其在节点出现短路故障情况下的网络重构，并将重构结果在 LED 模块上及 LCD 屏上显示出来。

实验分别模拟了在 Bus2 的 LP4、Bus4 的 LP11、以及 Bus4 的 LP13 三个节点分别出现短路故障的情况下，以 FIC 最少为目标函数进行网络重构。其重构结果参数如表 6-3 所示。

表 6-3 基于 FIC 最少的配电网重构结果

	重构前的停 电用户数	最后打开的 开关	重构后的停 电用户数	最低电压点(p.u.)
Bus2 的 LP4	980	4、15、50、 16、46、47、 48、49	10	0.9178 在 Bus4 的 LP18 处
Bus4 的 LP11	640	27、28、15、 16、46、47、 48、49	420	0.9021 在 Bus4 的 LP12 处
Bus4 的 LP13	980	29、48、50、 15、16、46、 47、49	20	0.9178 在 Bus4 的 LP18 处

从其中的参数可以发现，重构结果不仅很好地降低了停电时户数，使得重

构后的停电用户数相比重构前大量降低，也保证了系统中各节点的电压质量，线路中电压最低节点也保持在起始点电压（额定电压）标幺值的 0.9 以上。

#### 6.4 本章小结

本章结合本文所开发的嵌入式系统为平台，研究了以故障停电用户数最少为目标函数、结合前推后判网络潮流算法的基于克隆遗传优化算法的配电网络重构程序开发，就程序中各要点环节及其实现方法原理进行了分析研究，并给出了相关实验结果。

## 第七章 结论与展望

本文介绍了基于嵌入式系统的配电网重构实验装置的开发设计，主要内容包括配电网重构实现过程中的一些问题、以 ARM 和嵌入式 Linux 为基础搭建的嵌入式系统的实现、该配电网重构实验装置各个功能模块硬软件方面的设计实现，以及通过 RS232 将各个模块联接成一个以 ARM+Linux 嵌入式系统为核心的配网重构实验平台等。

本文的创新之处在于：其一，以实验装置代替电脑仿真，为配电网重构算法研究提供了可供实验的平台，为配网重构走向应用打下了基础；其二，采用了基于 ARM+Linux 的嵌入式系统平台，为重构算法提供了丰富的库文件以及开发工具支持；其三，采用模块化设计方法，保证了系统的可靠性以及可升级性。

### 7.1 结论

配电网重构是一个复杂的课题，国内外很多学者对此付出了辛勤努力和不懈探索，已经取得了令人瞩目的进展，并提出了各种求解方案和数学模型，通过各种算法来提高配电网重构算法的效能。

本文正是致力于为这些学者的研究提供一个可供实验的平台，让他们可以将自己研究的实验算法只需经过简单的打包下载，就可以做为嵌入式系统的应用软件在该平台上演示出来，以检验算法的有效性、收敛性以及时效性。同时，还可以通过该实验平台为相关学生或研究者提供关于配电网重构系统的结构及其工作过程的演示实验。

本文采用模块化设计的方法，根据不同的应用需求，围绕以 ARM+Linux 构建的嵌入式系统模块，在其周围选择添加不同的功能模块搭建组成两种不同等级的实验装置：对于以算法实验为应用目标的第一等级系统，除用于运行重构算法的嵌入式系统外，还配置了以 CYCLONE III FPGA 为核心的重构执行模块以及以 IEEE 配电网可靠性测试系统 Bus2 和 Bus 4 为网络模本设计的 LED 模块，使得重构算法的执行结果可以很清晰地得到展示；对于以模拟演示配电网重构过程为目标设计的第二等级系统，除了与第一等级系统共有的嵌入式系统模块以及重构执行模块以外，还以 IEEE 配电网可靠性测试系统中模拟配电网结构的 Bus2 和 Bus 4 为网络模本复制的一个低压网络模型组成模拟配电网模块，以及一个以 STC 单片机为核心的数据采集模块来对模拟配电网中的潮流数据进行采集，以便为嵌入式系统模块中的重构算法提供初始潮流参数。

此外，本文还研究了基于克隆遗传算法的配电网络重构优化算法，并结合所设计实验装置进行了联合调试，实验结果证明了该系统工作的有效性。

## 7.2 展望

在设计之初，就考虑到为了方便今后的升级，系统各个部分采用模块化设计，使用通用的 RS232 进行数据的传输。

就硬件系统而言，由于目前实验条件方面的限制，嵌入式系统模块选择了以 ARM9 为核心的实验箱来搭建，ARM9 处理器在数据运算方面的能力在应付优化算法方面稍显不足，尤其是在浮点型运算方面。当配电网络结构越来越复杂、网络节点越来越多、重构算法计算量越来越大的时候，更换一款功能更加强大的处理器是可供改进选项。

在模拟配电网络方面，为各个节点加入基于硬件的节点过电流保护电路也可以是下一步设计中可改进的地方。基于软件的保护功能由于几十个节点电流采集一轮下来的周期较长，因此在进行基于可靠性的重构实验时，模拟节点短路故障就会出现短路电流难以即时开断的问题。

此外，系统可以改进的地方还有通信接口方面，RS232 在通信距离上存在一定的缺陷，将其替换为现在工业现场广为应用的现场总线（如 CAN 总线）是可升级的另一方面。

就软件方面而言，对于嵌入式 Linux 内核的剪裁与优化是本文研究中较少涉及的地方，而这些工作对于提升系统的效能及可靠性也有着相当重要的作用，是下一步研究中可供探索的地方。

由于水平有限，文中其它不足的地方恳请各位专家和老师批评指正。

## 参考文献

- [1] 李先彬. 电力系统自动化(第五版). 北京: 中国电力出版社, 2007, 1~5
- [2] 许克明, 熊炜. 配电网自动化系统. 重庆: 重庆大学出版社, 2007, 2~9
- [3] Merlin A, Back H. Search for a Minimal-Loss Operating Spanning Tree for an Urban Power Distribution System. Proceedings of Fifth Power System Computation Conference. Cambridge. 1975: 2~6
- [4] 张大海, 江世芳, 赵建国. 配电网重构研究的现状与展望. 电力自动化设备, 2002, 22(2): 75~82
- [5] 刘栋, 陈允平, 沈广, 等. 负荷随机性对网络计算和配电网重构的影响. 电力系统自动化, 2006, (30): 25~28
- [6] 刘蔚, 韩祯祥. 基于时间区间的配电网重构. 电力系统自动化, 2006, (30): 33~38
- [7] Fan Ji Yuan, Zhang Lan, McDonald J D. Distribution Network Reconfiguration:Single Loop Optimization. IEEE Trans on Power Systems, 1996, 11(3): 1643~1647
- [8] Shirmohammadi D, Wayne Hong H. Reconfiguration of Electric Distribution Networks for Resistive Line Losses Reduction. IEEE Trans on Power Delivery, 1989, 4(2): 1492~1498
- [9] 韩学军, 陈鹏, 国新凤, 等. 基于潮流计算的配电网重构. 电网技术, 2007, (31): 60~63
- [10] 车仁飞, 李仁俊. 一种大规模配电网实时重构新方法. 电力系统自动化, 2004, (28): 59~63
- [11] Civanlar S, Grainger J J, Yin H, et al. Distribution Feeder Reconfiguration for Loss Reduction. IEEE Trans on Power Delivery, 1988, 3(3): 1217~1223
- [12] 赵登福, 刘云, 周文华等. 以能量损耗最小为目标函数的网络重构. 西安交通大学学报, 1999, 33(8): 16~19
- [13] GOSWAMI S K, BASU S K. A New Algorithm for the Reconfiguration of Distribution Feeders for Loss Reduction. IEEE Trans on Power Delivery, 1992, 7(3): 1484~1491
- [14] 毕鹏翔, 刘健, 张文元. 以提高供电电压质量为目标的配网重构. 电网技术, 2002, 26(2): 41~43
- [15] 李海锋, 张尧, 钱国基, 等. 配电网故障恢复重构算法研究. 电力系统自动

- 化, 2001, (08): 34~37
- [16] 贾立新. 配电网故障恢复与网络重构方法研究: [博士学位论文]. 西安: 西安交通大学, 2003
- [17] 电力小站. 配电网网络重构.  
[http://zxh20040410.home.sunbo.net/show\\_hdr.php?xname=4G1I701&dbname=TEDEAUO&xpos=4](http://zxh20040410.home.sunbo.net/show_hdr.php?xname=4G1I701&dbname=TEDEAUO&xpos=4)
- [18] 探矿工作室. 嵌入式系统开发圣经(第二版). 北京: 中国铁道出版社, 2003, 1~17
- [19] 周立功等. ARM 与嵌入式系统基础教程. 北京: 北京航空航天大学出版社, 2005, 3~167
- [20] ARM Limited. ARM Architecture Reference Manual. ARM Limited, 2000
- [21] ARM Architecture Reference Manual. ARM Corp, 2000
- [22] 陈俊宏. Embedded Linux 嵌入式系统原理与实务. 北京: 清华大学出版社, 2004, 24~112
- [23] RUBINIA, CORBETJ. Linux Device drivers. 2th ed. California, USA: O'Reilly Press, 2001
- [24] 王卓, 包洁. 嵌入式 Linux 系统及其应用前景. 单片机及嵌入式系统应用, 2004, (5): 5~9
- [25] 邹荣士, 郭立红, 司玉美, 等. 多通道串行通信设备的 Linux 驱动程序实现. 吉林大学学报(工学版), 2007, 37(1): 164~167
- [26] 魏永明, 耿岳等. Linux 编程宝典. 北京: 电子工业出版社, 2000
- [27] Jean.Labrosse 著, 邵贝贝等译. 嵌入式实时操作系统 uC/OS-II(第 2 版). 北京: 北京航空航天大学出版社, 2003. 1~7
- [28] 李纪奎, 向怀坤, 胡泓, 等. 基于 uCOS-II 的视频动态交通信息采集系统研究. 哈尔滨工业大学学报, 2007, 39(2): 306~309
- [29] Labrosse J J. MicroC/OS-II—The Real-time-Kernel(Second Edition). USA: CMP Book, 2002
- [30] 罗毅, 谭增显, 吕跃刚, 等. 基于 Win CE 的控制系统优化整定仪的设计与开发. 仪器仪表学报, 2003, 24(4): 519~524
- [31] Billinton Roy, Jonnavithula Satish. Test system for teaching overall power system reliability assessment. IEEE Transactions on Power Systems, 1996, 11(4): 1670~1676
- [32] Allan R N, Billinton R, Sjariel I, Goel L, So K S. A reliability test system for educational purposes--Basic distribution system data and results. IEEE

- Transactions on Power Systems, 1991, 6(2): 813~820
- [33] 博创科技. UP-NETARM2410-S 开发平台硬件说明书, 2006
- [34] S3C2410A User's Manual. Revision1.0. Samsung Electronics Corp, 2004
- [35] S3C2410A Application Note Revision1. Samsung Electronics Corp, 2004
- [36] UP-NETARM2410-S 实验指导书. 博创科技, 2006
- [37] 孙天泽, 袁文菊, 张海峰. 嵌入式设计及 Linux 驱动开发指南-基于 ARM9 处理器. 北京: 电子工业出版社, 2005. 23~152
- [38] [美]Arnold Berger. 嵌入式系统设计. 电子工业出版社, 2002. 11~135
- [39] 慕春棣. 嵌入式系统的构建. 北京: 清华大学出版社, 2004. 3~125
- [40] Daniel P.Bovet, Marco cesati. Understanding Linux kernel, 3rd Edition. USA: O'Reilly, 2005
- [41] 梁丁, 李迅波, 蔡勇. 应用中的嵌入式 Linux 实时优化. 计算机工程, 2007, 33(1): 77~79
- [42] 孙天泽, 袁文菊, 张海峰. 嵌入式设计及 Linux 驱动开发指南—基 ARM9 处理器. 电子工业出版社, 2005. 3~175
- [43] 杜明芳. ARM&Linux 架构下智能门禁控制器以太网接口开发. 计算机工程, 2007, 33(16): 234~236
- [44] 张金平, 张定会, 谢光伟, 等. 基于 Linux 下的 ARM 与 DSP 之间数据通信. 仪器仪表学报, 2006, 27(6): 2056~2059
- [45] 李顶根, 陈军, 吴朝辉. 基于 ARM-Linux 的车载信息平台的研制. 浙江大学学报(工学版), 2006, 40 (9):1531~1535
- [46] 张金平, 张定会, 谢光伟, 等. 基于 Linux 下的 ARM 与 DSP 之间数据通信. 仪器仪表学报, 2006, 27(6): 2056~2059
- [47] 刘昌举, 戴基智, 龙再川, 等. 基于嵌入式 Linux 和 S3C2410 的远程 CCD 图像采集系统. 光电技术应用, 2006, 27(6): 777~783
- [48] 马忠梅, 李善平, 康慨, 叶楠. ARM & Linux 嵌入式系统教程. 北京航空航天大学出版社, 2004. 115~176
- [49] Cyclone III Device Handbook. [www.altera.com](http://www.altera.com), 2007
- [50] Configuration Handbook. [www.altera.com](http://www.altera.com), 2007
- [51] STC12C5410AD 系列单片机器件手册. [www.MCU-Memory.com](http://www.MCU-Memory.com), 2007
- [52] 李振然. 利用富里叶变换实现微机自动准同期. 继电器, 1994(4): 35~40
- [53] 陈小桥, 周水斌, 王先培, 等. 一种新的相位(差)算法及其在自动准同期中的应用. 武汉大学学报(工学版), 2003, 6(36): 96~99
- [54] Sun Jian, Jiang Dao-zhuo, Liu Zhi-hua. Modified power flow algorithm of

- distribution network based on forward/backward sweep method. *Electric Power Automation Equipment*, 2004, 24(3): 81~84
- [55] 戴雯霞, 吴捷. 基于支路电流的配网潮流前推后代法. *继电器*, 2002, 30(5): 6~8
- [56] 李金鹏, 韩英仕, 李基波. 遗传算法原理及在结构优化设计中的应用. *辽宁工学院学报*, 2004, 24(3): 56~60
- [57] Sivanagaraju S, Srikanth Y, Jagadish Babu E. An efficient genetic algorithm for loss minimum distribution system reconfiguration. *Electric Power Components and Systems*, 2006, 34(3): 249~258
- [58] Prasad K, Ranjan R, Sahoo Nirod Chandra, Chaturvedi A. Optimal reconfiguration of radial distribution systems using a fuzzy mutated genetic algorithm. *IEEE Transactions on Power Delivery*, 2005, 20(2): 1211~1213
- [59] 李茂军, 童调生. 单亲遗传算法及其全局收敛性分析. *自动化学报*, 1999, 25(1): 68~72
- [60] 李茂军, 邱丽芳, 童调生. 单亲遗传算法的计算效率分析. *长沙电力学院学报(自然科学版)*, 1999, 14(1): 48~50
- [61] Wang Ji, Luo An, Li Mao jun. The Improved Clonal Genetic Algorithm & Its Application in Reconfiguration of Distribution Networks. *IEEE Power Engineering Society. PSCE 2004 - Power Systems Conference and Exposition*. New York, USA. 2004. 559~604.
- [62] 李茂军, 罗日成, 童调生. 单亲遗传算法的遗传算子分析. *系统工程与电子技术*, 2001, 23(8): 84~87
- [63] 麻秀范, 张粒子. 基于十进制编码的配网重构遗传算法. *电工技术学报*, 2004, 19(10): 65~69

## 附录

### 附 1 IEEE RBTS BUS2、BUS4 原始数据表

附表 1.1 IEEE RBTS BUS2 原始数据表

节点	用户类型	平均视在功率(MVA)	最大视在功率(MVA)	用户数
LP1-LP2	residential	1.07	1.7336	420
LP3-LP4	residential + govt	1.101	1.7835	211
LP5-LP6	govt + commercial	1.02	1.6667	11
LP7	commercial	0.454	0.75	10
LP8	small user	1	1.6279	1
LP9	small user	1.15	1.8721	1
LP10	residential	0.535	0.8668	210
LP11-LP12	residential	0.985	1.5959	410
LP13-LP14	govt	1.132	1.8334	2
LP15	commercial	0.454	0.75	10
LP16-LP17	commercial + resident	0.904	1.4791	210
LP18-LP19	residential	0.9	1.4582	400
LP20	govt	0.566	0.9167	1
LP21-LP22	govt + commercial	1.02	1.6667	11
总计		12.291	20.0006	1908

附表 1.2 IEEE RBTS BUS4 原始数据表

节点	用户类型	平均视在功率(MVA)	最大视在功率(MVA)	用户数
LP1	residential	0.545	0.8869	220
LP2	residential	0.545	0.8869	220
LP3	residential	0.545	0.8869	220
LP4-LP5	residential	1.045	1.7006	420
LP6-LP7	commercial	0.83	1.3428	20
LP8	small user	1	1.63	1
LP9	small user	1.5	2.445	1
LP10	small user	1	1.63	1
LP11	residential	0.545	0.8869	220
LP12	residential	0.545	0.8869	220
LP13-LP14	residential	1.045	1.7006	200
LP15	residential	0.5	0.8137	420
LP16-LP17	commercial	0.83	1.3428	20
LP18	residential	0.545	0.8869	220

LP19-LP20	residential	1.09	1.7738	440
LP21-LP22	residential	1.045	1.7006	420
LP23	residential	0.5	0.8137	200
LP24-LP25	commercial	0.83	1.3428	20
LP26	small user	1	1.63	1
LP27	small user	1	1.63	1
LP28	small user	1	1.63	1
LP29	small user	1	1.63	1
LP30	small user	1	1.63	1
LP31	small user	1.5	2.445	1
LP32	residential	0.545	0.8869	220
LP33	residential	0.545	0.8869	220
LP34-LP35	residential	1.09	1.7738	440
LP36	residential	0.5	0.8137	200
LP37-LP38	resident + commercial	0.915	1.4851	210
总计		24.58	39.9992	4779

## 附 2 实验装置 BUS2、BUS4 各节点电阻、电感值表

附表 2.1 实验装置 BUS2 各节点电阻、电感值

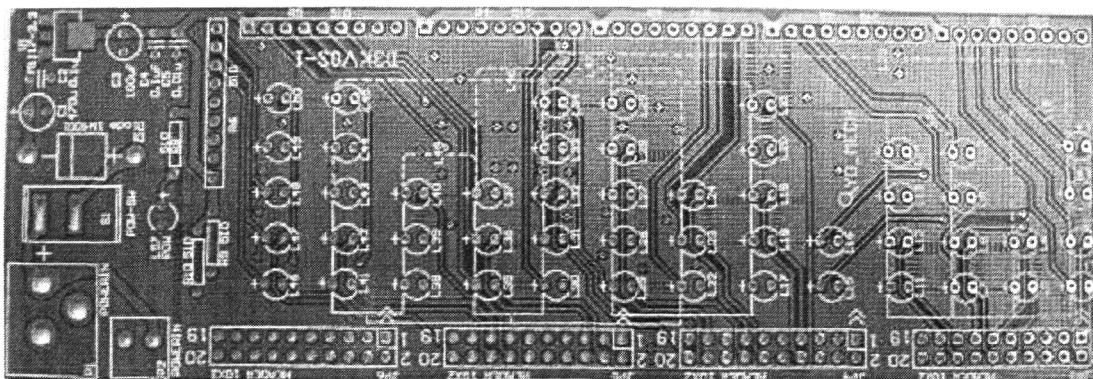
节点号	旧节点	电阻(欧姆)	电感(mH)
节点 1	LP1-LP2	17.5795	18.4013
节点 2	LP3-LP4	17.0836	17.8824
节点 3	LP5-LP6	18.4412	19.3038
节点 4	LP7	41.4296	43.3788
节点 5	LP8	18.8059	19.7097
节点 6	LP9	16.3564	17.1219
节点 7	LP10	35.1542	36.8047
节点 8	LP11-LP12	19.0952	19.9893
节点 9	LP13-LP14	16.6163	17.3951
节点 10	LP15	41.4296	43.3788
节点 11	LP16-LP17	20.8071	21.7822
节点 12	LP18-LP19	20.3240	20.5513
节点 13	LP20	33.2345	34.7820
节点 14	LP21-LP22	18.4412	19.3038

附表 2.2 实验装置 BUS4 各节点电阻、电感值

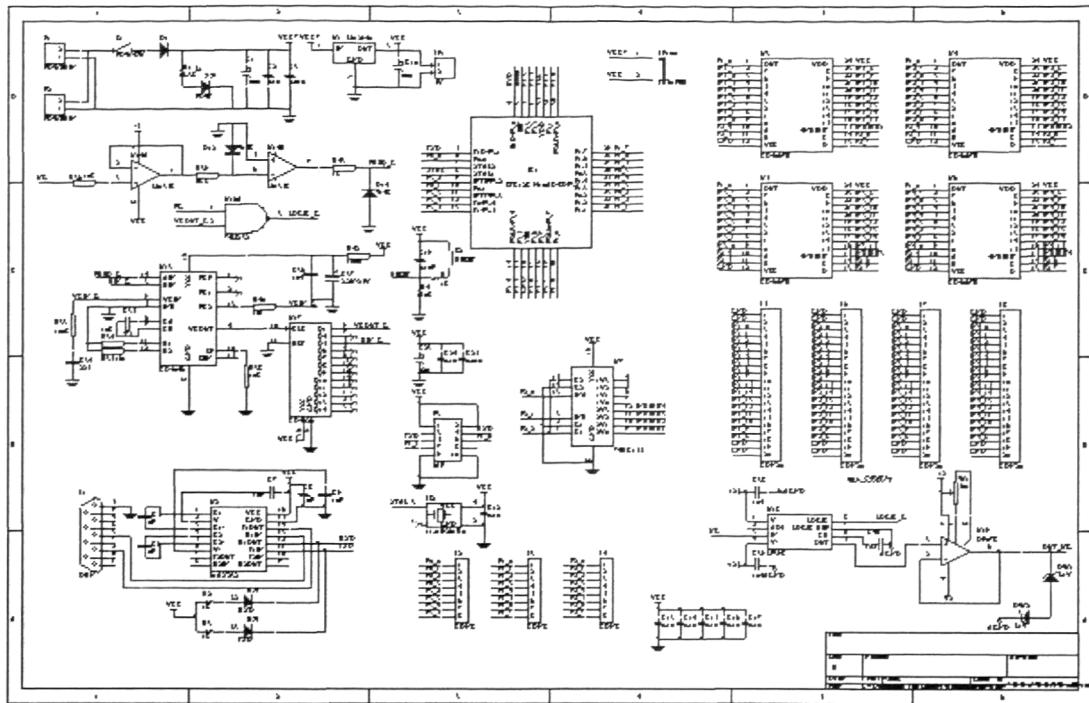
节点号	旧节点	电阻(欧)	电感(mH)
节点 1	LP1	34.5102	36.1256
节点 2	LP2	34.5102	36.1256
节点 3	LP3	34.5102	36.1256

节点 4	LP4-LP5	17.9993	18.8400
节点 5	LP6-LP7	22.6622	23.7249
节点 6	LP8	18.8094	19.6921
节点 7	LP9	12.5398	13.1270
节点 8	LP10	18.8094	19.6921
节点 9	LP11	34.5102	36.1256
节点 10	LP12	34.5102	36.1256
节点 11	LP13	17.9993	18.8400
节点 12	LP14-LP15	37.6212	39.3743
节点 13	LP16-LP17	22.6622	23.7249
节点 14	LP18	34.5102	36.1256
节点 15	LP19-LP20	17.2565	18.0660
节点 16	LP21-LP22	17.9993	18.8400
节点 17	LP23	37.6212	39.3743
节点 18	LP24-LP25	22.6622	23.7249
节点 19	LP26	18.8094	19.6921
节点 20	LP27	18.8094	19.6921
节点 21	LP28	18.8094	19.6921
节点 22	LP29	18.8094	19.6921
节点 23	LP30	18.8094	19.6921
节点 24	LP31	12.5398	13.1270
节点 25	LP32	34.5102	36.1256
节点 26	LP33	34.5102	36.1256
节点 27	LP34-LP35	17.2565	18.0660
节点 28	LP36	37.6212	39.3743
节点 29	LP37-LP38	20.5566	21.5160

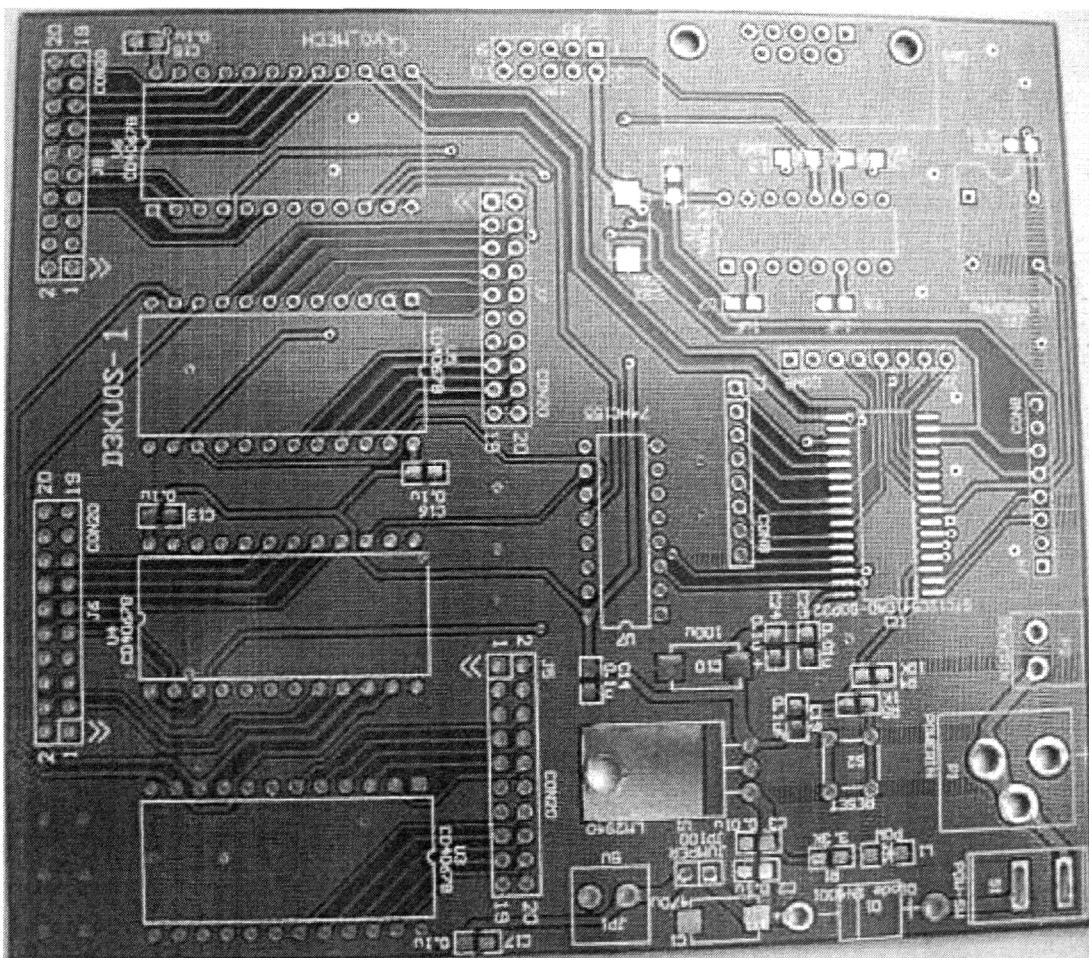
### 附 3 各部分电路原理及 PCB 版图



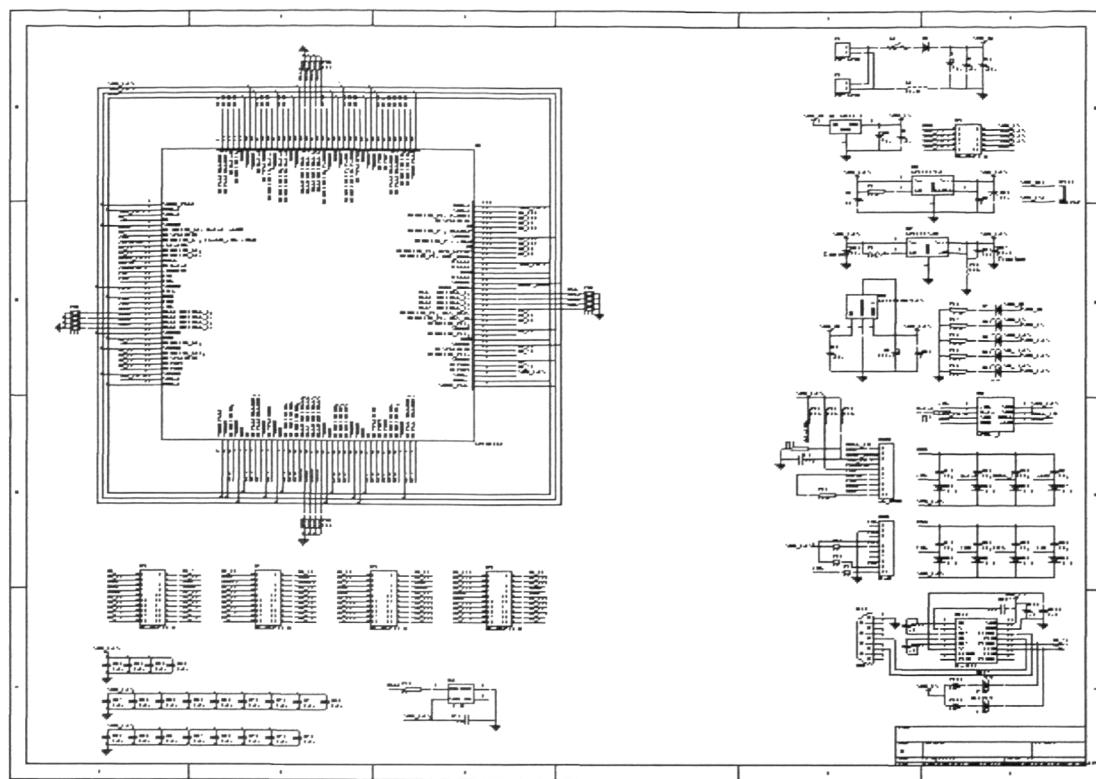
附图 3.1 LED 模块 PCB 版图



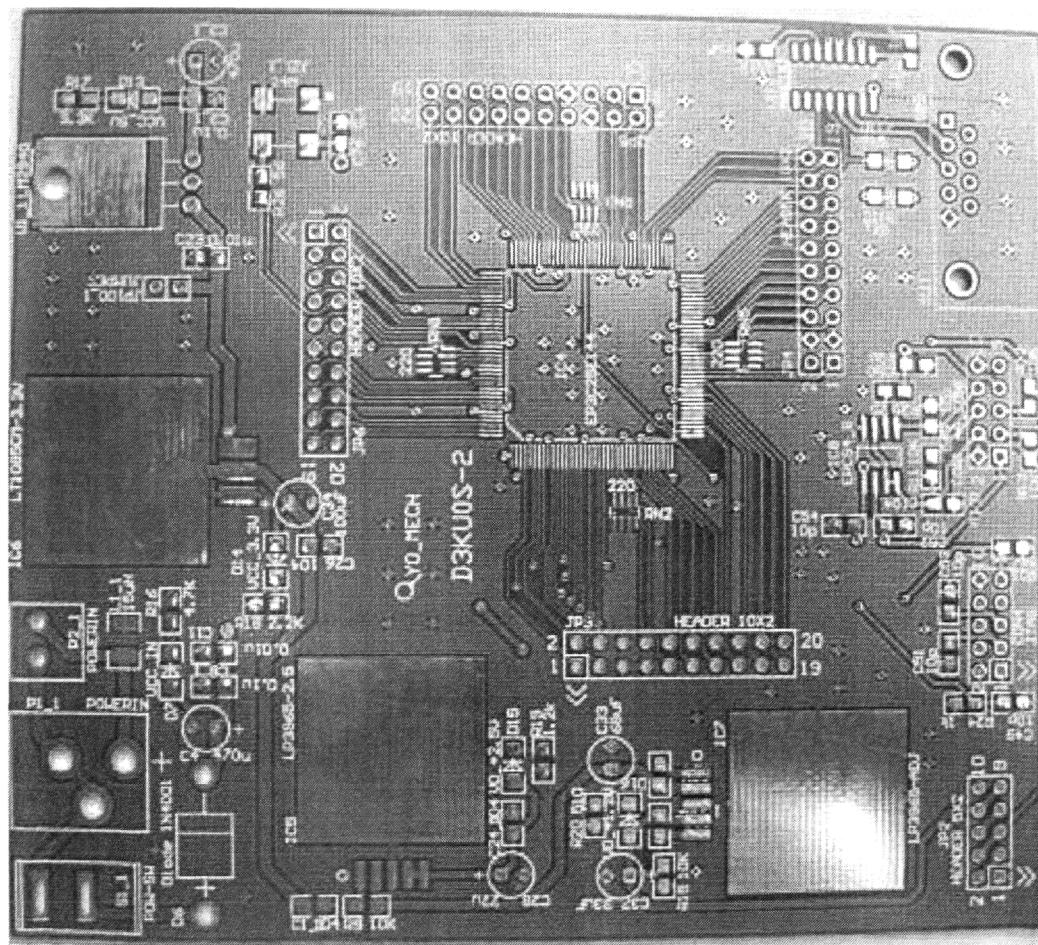
附图 3.2 数据采集模块电路原理图



附图 3.3 数据采集模块 PCB 版图



附图 3.4 重构执行模块电路原理图



附图 3.5 重构执行模块 PCB 版图

## 4、FPGA 的 UART 通信的 Verilog HDL 程序

### 4.1 接收模块程序

```
module receive(dout,data_ready,framing_error,parity_error,rx,clk16x,rst,rdn);
input rx,clk16x,rst,rdn;
output[7:0] dout;
output data_ready,framing_error,parity_error;

reg data_ready,framing_error,parity_error,clk1x_enable,parity;
reg rx1,rx2;
reg[3:0] rbr,clkdiv,no_bits_rcvd;
reg[7:0] rsr;
wire clk1x;

assign dout=!rdn?rbr:8'bzz;
always@(posedge clk16x or posedge rst)
begin
  if(rst)
    begin
      rx1<=1'b1;
      rx2<=1'b1;
    end
  else
    begin
      rx1<=rx;
      rx2<=rx1;
    end
end

always@(posedge clk16x or posedge rst)
begin
  if(rst) clk1x_enable<=1'b0;
  else if(!rx1&&rx2) clk1x_enable<=1'b1;
  else if(no_bits_rcvd==4b'1100) clk1x_enable<=1'b0;
end
```

```
always@(posedge clk16x or posedge rst or negedge rdn)
begin
    if(rst) data_ready=1'b0;
    else if(!rdn) data_ready=1'b0;
        else if(no_bits_rcvd==4'b1011) data_ready=1'b1;
end

always@(posedge clk16x or posedge rst)
begin
    if(rst) clkdiv= 4'b0000;
    else if(clk1x_enable) clkdiv= clkdiv+1;
end

assign clk1x=clkdiv[3];

always@(posedge clk1x or posedge rst)
if(rst)
begin
    rsr <= 8'b0;
    rbr <= 8'b0;
    parity <= 1'b1;
    framing_error=1'b0;
    parity_error=1'b0;
end
else
begin
    if(no_bits_rcvd==4'b0001&&no_bits_rcvd==4'b1001)
        begin
            rsr[0]<=rxd2;
            rsr[7:1]<=rsr[6:0];
            parity <= Parity^rsr[7];
        end
    else if(no_bits_rcvd==4'b1010) rbr <= rsr;
    else if(!parity) parity_error=1'b1;
```

```

    else if((no_bits_rcvd==4'b1011)&&(rx2!=1'b1))
        framing_error=1'b1;
    else framing_error=1'b0;
end

always@(posedge clk1x or posedge rst or negedge clk1x_enable)
if(rst) no_bits_rcvd=4'b0000;
else if(!clk1x_enable) no_bits_rcvd=4'b0000;
else no_bits_rcvd=no_bits_rcvd+1;

endmodule

```

#### 4.2 发送模块程序

```

module trans(din,tbre,tsre,rst,clk16x,wrn,txd);
output tbre,tsre,txd;
input[7:0] din;
input rst,clk16x,wrn;

reg tbre,tsre,txd,parity;
reg [7:0]tbr,tsr;
reg [3:0] clkdiv;
reg clk1x_enable;
wire clk1x;
reg [3:0] no_bits_sent;
reg wrn1,wrn2;

always@(posedge clk16x or posedge rst)
begin
if(rst)
begin
    wrn1<=1'b1;
    wrn2<=1'b1;
end
else
begin

```

```
wrn1<=wrn;
wrn2<=wrn1;
end
end

always@(posedge clk16x or posedge rst)
begin
  if(rst)
    begin
      tbre<=1'b0;
      clk1x_enable<=1'b0;
    end
  else if(!wrn1&&wrn2)
    begin
      clk1x_enable<=1'b1;
      tbre<=1'b1;
    end
  else if(no_bits_sent==4'b0010) tbre<=1'b1;
  else if(no_bits_sent==4'b1101)
    begin
      clk1x_enable<=1'b0;
      tbre<=1'b0;
    end
end

always@(posedge wrn or posedge rst)
begin
  if(rst) tbr=8'b0;
  else tbr=din;
end

always@(posedge clk16x or posedge rst)
begin
  if(rst) clkdiv=4'b0;
```

```
else if(clk1x_enable) clkdiv=clkdiv+1;  
end  
  
assign clk1x=clkdiv[3];  
always@(negedge clk1x or posedge rst)  
if(rst)  
begin  
    txd<=1'b1;  
    tsre<=1'b1;  
    parity<=1'b1;  
    tsr<=8'b0;  
end  
else  
begin  
    if(no_bits_sent==4'b0001)  
    begin  
        tsr<=tbr  
        tsre<=1'b0;  
    end  
    else if(no_bits_sent==4'b0010) txd<1'b0;  
    else if((no_bits_sent==4'b0011)&&(no_bits_sent==4'b1010))  
    begin  
        tsr[7:1]<=tsr[6:0];  
        tsr[0]<=1'b0;  
        txd<=tsr[7];  
        parity<=parity^tsr[7];  
    end  
    else if(no_bits_sent==4'b1011) txd<=parity;  
    else if(no_bits_sent==4'b1100)  
    begin  
        txd<=1'b1;  
        tsre<=1'b1;  
    end  
end
```

```
always@(posedge clk1x or posedge rst or negedge clk1x_enable)
if(rst) no_bits_sent=4'b0000;
else if (!clk1x_enable) no_bit_sent=4'b0000;
else no_bits_sent=no_bits_sent+1;

endmodul
```

## 致 谢

在中南大学这座美丽的校园里，我度过了近三年美好的学习时光。在这段可贵的人生经历中，我得到了很多人的关心和帮助，谨在此向他们表示我最真挚的谢意！

首先，我要衷心感谢我的导师王击副教授。这三年来，王老师从各个方面都给予了我许多帮助，无论是在做学问上，还是做人上。王老师知识渊博、经验丰富、精力充沛、平易近人，让人钦佩，使我受益匪浅。其对技术前沿的敏锐观察力使我有机会来完成这样一个富有挑战性和创造乐趣的课题。在毕业论文即将完成之际，谨向王老师致以最真挚的谢意！

同时，感谢我的师兄弟们，周辉、覃业梅、李学军、边志伟、游佳、马肃、徐留杰、邹凤姣等同学在各方面给予了我真诚的帮助，信息科学与工程学院的其他老师、同学和朋友也都给予过我无私的指导和帮助，在此一并谢过！

此外，我还要特别感谢我的父母、家人，是他们给予了我工作上、经济上和精神上的大力支持，使我得以能够顺利完成学业，希望在今后的人生中，我将不辜负他们对我的期望。

黄 怡  
二〇〇八年四月于民主楼

## 攻读硕士学位期间的主要研究成果

### 1、作者攻读硕士学位期间参加的科研项目情况

- [1] 2006 年“飞思卡尔”杯全国大学生智能车邀请赛，主持设计制作了以飞思卡尔 16 位单片机 MC9SDG128B 为控制器的智能寻迹小车
- [2] 基于 FPGA 的 SDI 视频键控混合放大器，与湖南双菱电子合作研究开发 SDI 键控混合放大器，并已成功面向市场，被不少电视台选用
- [3] 基于嵌入式系统的配电网络重构实验装置开发

### 2、作者攻读硕士学位期间已经发表的论文（两篇）

- [4] 黄怡，王击，覃业梅. 基于 P-FUSSY-PI 控制的寻迹机器人设计. 微计算机信息. (已录稿)
- [5] 黄怡，王击，覃业梅. Design of Line-Tracking Robot Based on P-FUSSY-PI Control. Microcomputer Information. (已录稿)