

天津大学
硕士学位论文
嵌入式动态信号采集分析便携系统的研究
姓名：薛国光
申请学位级别：硕士
专业：机械电子
指导教师：王太勇
20030101

# 中文摘要

先进故障诊断技术离不开先进的诊断仪器。目前,故障诊断技术正朝着分布式、网络化、高精度、高速度发展。而作为设备状态检测和故障诊断系统前端的数据采集分析装置也在朝着便携式、低功耗、网络化方向发展。针对国内相关诊断仪器的空白,我们采用了先进的电子仪器设计技术,结合数字信号处理技术、嵌入式 CPU 设计技术和实时操作系统任务设计技术,模块化的构建一个广泛适用于工厂基层设备的便携式数据采集分析系统。

论文首先考察了故障诊断技术的发展现状和发展趋势,并在详细分析了现有产品的基础上提出了“DSP+嵌入式 CPU”的设计思路,并论证了系统的可行性。

其次,论文阐述了基于 DSP 的数据采集分析模块的设计原理。在充分考察 DSP 技术的原理、设计方法、DSP 芯片选择方法之后,讨论了数据采集分析模块的实现细节。解决了若干原来仪器设计当中未能解决的问题,如:可编程抗混滤波器设计、加速度传感器和 ICP 传感器的信号调理、DSP 数据采集分析模块和主模块的数据通讯等问题。

然后,论文重点阐述了主控模块的设计原理。在充分熟悉了 MC68332 微处理器的基础上,设计了广泛适用于多种接口的主控模块。并结合实时操作系统的特点和 MC68332 微处理器自身的体系结构,讨论了  $\mu$ COSII 实时操作系统在 MC68332 微处理器上的移植问题,为进一步的开发奠定基础。

论文最后阐述了本数据采集分析平台的软件设计问题。论述了在实时操作系统基础上的任务设计方法,并模块化的编写了从底层驱动程序到上层应用程序的相关代码。最后,论文重点讨论了网络化诊断需要解决的网络协议实现问题,使系统具有了分布式诊断的能力。

关键词: 嵌入式、实时操作系统、数据采集、故障诊断、DSP、MCU

# ABSTRACT

Advanced fault diagnostic technologies cannot get away from advanced diagnostic instruments. Nowadays, fault diagnostic technologies are developing towards distribution, interconnection, high precision and high speed. Being the front-end of equipment status monitoring and their related fault diagnosis, data acquisition and analysis systems are developing towards portable, low power consumption and networking. In this thesis, we build a portable data acquisition and analytical instrument widely used in the factories' manufacturing equipments based on advanced electronical design technologies, digital signal processing technologies, embeded CPU application technologies and real time operation system (RTOS) tasks design technologies aiming at filling the blank of related diagnostic instruments.

First of all, the thesis investigates the present condition and the trend of fault diagnostic technologies, and presents a design architecture using the structure of DSP plus embedded CPU based on a detail analysis of related products on hand.

Then, the paper illustrates the design principles of data acquisition module based on digital signal processor (DSP). After a fully inspection of the principles of DSP and its design methodologies and the selection method of DSP chip, we discuss the realization of a data acquisition and analysis system in detail. We solve many problems such as a programmable anti-aliasing filters design, the signal conditioning circuit of accelerative sensors and ICP sensors and the data transfer communication problems between DSP data acquisition and analysis system and the main control module.

Then, the paper focuses on the design principles of main control module. We make a module given flexible interface widely used in control system based on the familiarity of MC68332 microcontroller. Also we compare features of ucos RTOS and computing architecture and port ucos RTOS to MC68332 which make a solid foundation to further development.

The paper eventually discusses the software design problems based on the data acquisition and analysis platform. The tasks division method based on RTOS and how to modularize the software architecture from low-level drivers and upper application software are also discussed. In the end, the paper focus on the solution of network

diagnosis and the related protocol realization, which enable the system, diagnose on the Internet.

**Keywords:** Embedded system, RTOS, data acquisition, fault diagnosis, DSP, MCU

## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，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 天津大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：薛国光 签字日期：2002年12月15日

## 学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：薛国光

导师签名：王树勇

签字日期：2002年12月15日

签字日期：2002年12月15日

## 第一章 绪论

### 1.1 故障诊断科学概述

设备故障诊断是利用设备在运行中的状态信息,通过对所测信号的处理和分析,结合诊断对象的历史状态来定量识别设备及其零部件的运行状态,并预知有关异常、故障,从而确定必要对策的技术。故障诊断技术最早于 70 年代初成形于英国,由于其实用性以及为社会和企业带来的效益,日益受到政府和企业主管部门的重视。由于计算机技术的迅速发展和普及,它已经成为一门较为完善的边缘综合工程学科。该学科以设备的管理、状态监测和故障诊断为研究内容,以建立新的维修体制为目标,目前是国际上的一门热门学科。

故障诊断技术依据设备在运行过程当中伴有的振动、噪声、发热、应变、压力等物理参数的变化,通过一定的方式来判断和识别设备运行的状态和故障,对故障的危害进行早期的预报和识别,防止设备在故障状态下连续运行而造成的设备失效和相应的附加损失,保证设备安全、正常、长周期的、满负荷优质运行。避免“过剩维修”所造成的不经济、不合理的现象<sup>[1]</sup>。

### 1.2 在线状态监测故障诊断系统的发展概况

从 80 年代初开始,以信号分析为基础的先进分析方法和仪器以及离线/在线状态监测和故障诊断系统在促进生产自动化和提高劳动生产率方面取得了显著的成效<sup>[2]</sup>。在线方式主要用于对大型设备或关键机组的监测和诊断。这类系统一般一次安装,采用先进的计算机技术、传感器技术和网络技术,可实现对机组的连续状态监测和诊断,并且能够保存设备运行的历史信息。但是由于工况环境、监测和诊断的实时性要求、被监测设备的多样性等苛刻的技术要求,在线监测和故障诊断系统一般体积也比较庞大并且价格昂贵。为了保证实时监测和诊断的要求,有时候不得不牺牲系统的计算能力,一般只能实现平稳信号的采集和处理。而且不能对所有的设备安装次类系统,应用上受到限制。

离线监测和诊断系统对系统的需求没有在线系统那么苛刻。该类系统能够对离线数据进行分析 and 预测,并得出诊断结论。该类系统可以远离工况,对实时性能没有什么特别的要求,因此可以采用先进的分析和诊断算法。除了能够对系统进行平稳信号特征提取外,还可以利用先进的信号处理技术,如 Wigner 分布、



短时傅利叶变换 (STFT), 小波变换 (WT) 等平稳和非平稳信号处理的先进信号分析方法, 结合分形几何、神经网络、专家系统和模糊辨识等近代先进的识别手段, 与传统的监测和诊断方法结合, 能够充分的对设备状态进行监测、诊断和预测。连接离线分析系统和现场设备运行情况的关键, 便是设备运行状况的测试仪器。测试仪器完成对现场工业环境的巡回监测, 记录或监测设备运行状况的特征参数和振动波形。目前广泛使用的是便携式数据采集器, 便携式振动分析仪以及带监测功能的便携式状态监测仪<sup>[3][4][5]</sup>。

(1) 便携式振动数据采集器: 主要完成振动信号的数据采集, 将信号特征如振幅、频率、相位或完整的波形存储到存储介质内。丹麦的 B&K 公司的 2526 型和德国的 SCHENCK 公司的 S60 系列产品是该类产品的典型。该类系统的特点是具备在线采集和离线采集功能、液晶显示、以 PCMCIA 标准存储卡作存储介质、模块化结构。可用互组态, 用户可选择的模块功能大多数以固件 (Firmware) 方式提供。

(2) 便携式振动分析仪: 主要完成振动信号的波形、频谱、相位的测量和相关的运行状态量的测量, 具有实时频谱分析功能。英国 DI 公司的 PL202 系列产品和 ENTEK 公司的 DATAPAC 系列产品就属于这一类仪器。该类仪器不但具有数据采集和存储功能, 还能够实时对信号进行 FFT 等运算, 并且具有和计算机的通讯功能, 但是系统一般比较昂贵。国内也有类似产品, 性能上和国外产品还有一定的差距, 成本比较高, 操作比较繁琐和复杂, 有些功能并不能满足用户需求, 所以一般用于实验室内, 很难在国内基层工况环境推广使用。

(3) 便携式状态监测仪: 用于对设备的巡检, 具有初步的监测和诊断功能。典型产品有 SCHENCK 公司的 PORT5000 系列、CSI 公司的 3200 系统。ENTEK 公司的 DATAPAC 系列也具备这一功能。这类产品虽然是便携式系统, 但是体积庞大笨重并且昂贵, 在国内使用较少。

下面我们通过对 ENTEK 公司的 DATAPAC 产品的分析, 来大致的了解这类产品的特点和性能要求<sup>[6]</sup>。ENTEK 是 ROCKWELL 公司的一个子公司, 主要是集成状态检测解决方案的提供商。状态检测产品覆盖了振动分析、在线监测、保护系统、远程监控等领域。国内有数家厂家在使用其相关产品。如天津天钢集团、广西玉柴等大型企业。

DATAPAC 系统是 ENTEK 推出的掌上型监测仪器, 双通道输入, 能够做单平面或双平面的动平衡。基于 WINDOWS CE 嵌入式操作系统, 操作方便。在过程设备当中作为振动监测、轴承检测方面具有一定的技术优势。相关性能参考如下。

- ◆ 操作系统: WINDOWS CE 嵌入式操作系统;
- ◆ 主处理器: MIPS RISC 嵌入式处理器;
- ◆ DSP 采用 MOTOROLA DSP56303, 运算能力 60MIPS 到 80MIPS;

- ◆ 操作系统存储容量 8M FLASH, 4M 文件系统, 16M 系统内存;
- ◆ 可充电 L-ION 电池, 可以不用外接电源;
- ◆ 支持 I 类、II 类 PCMCIA 存储卡, 或者是 SRAM、ATA FLASH 等存储介质;
- ◆ 和 PC 能够进行串口通讯和红外接口 IrDA
- ◆ 240x160 或 240x320 点阵式液晶屏幕, 背光可调;
- ◆ 工作温度-10 度到+50 (60) 度
- ◆ 双通道输入可以是 ICP、AC、DC
- ◆ 自动量程设置
- ◆ 动态范围>90dB

我们看出, DATAPAC 系列的产品目前采用的是主从结构的模块化的硬件系统, 在嵌入式操作系统的控制下, 每个模块实现自己的功能, 在 MIPS CPU 的控制下, 完成整个系统的功能。并且, 系统的功能只要采用组态的先进思想, 就可以实现一个产品的系列化, 从技术上说可以专心的实现系统更多、性能要求更高的功能, 从价格上来讲由于采用了模块化的设计技术, 降低了成本, 缩短了产品的上市时间, 提高了产品的市场竞争力。

故障检测和诊断方法是一个不断在发展的学科, 诊断对象也是越来越复杂, 诊断的手段也越来越高。如何缩短与世界在这个方向上的差距, 是我们每一位工程人员应尽的职责。因此, 抓住机遇, 在硬件和软件上采用先进的技术和方法的基础上, 研究适用于企业各种设备工况的工业级嵌入式便携数采分析单元, 并模块化的构建广泛用于设备智能诊断与状态预测的通用模块和系统, 对基于状态监测的设备预知维护和科学管理具有重要意义。

### 1.3 状态监测系统的发展趋势

随着信息技术的飞速发展, 计算机技术和数字信号处理技术以及片上系统设计工具、软件的日益成熟状态监测系统向着如下几个方向发展 [7][8][9][36][37][38][39][40]。

- 1、状态监测系统和仪器正朝着网络化、智能化和开放式方向发展。系统的总体结构以先进的计算机技术为依托, 以虚拟仪器的架构逐步替代掉了传统的诊断仪器。最具代表性的是美国的国家仪器公司 NI (National Instrument) 推出的 S8000 系列在线监测故障诊断系统。
- 2、状态监测和故障诊断系统正逐步发展成为现代化的管理系统。它不仅包括了现代企业设备的状态监测、故障诊断和预报等已有的功能, 而且正在扩大到以设备全过程经济管理的庞大的软件包, 使得状态监测和故障诊断系统成为现代企业管理的一个重要的组成部分。



- 3、运用网络通讯技术实现数据的远程共享和诊断，实现分布式意义上的故障诊断系统。Internet 网络上的专家可以随时随地地参加设备的会诊。
- 4、状态监测的前端数据采集模块在新型的通讯总线标准和日益强大的单片集成 (SOC) 技术的支持下，不断地向小型化、单片化、便携式、方便安装方便组态、高度集成和通用化方向发展。

## 1.4 课题任务的提出

本课题是在天津市自然科学基金的申请过程中，结合实验室的发展要求，在分析现有的测试硬件及相应的信号分析和故障诊断软件以及中国机床网和设备诊断网的网络架构基础上提出来的。如下是我们提出的设计指标：

具有实时 FFT 功能，具有 FFT 的实时性，可以集成慢变信号的采集和处理，可以实现多种工业现场信号的采集分析；具有在线数据管理和在线监测故障诊断功能；有较高的传输速率和上位机交换数据，可以通过适当的软硬件扩展和 PC 组成一套完善的故障诊断分析系统；具有控制网络功能，能够提供现场总线组成分布式的测试系统；能够进行数据的远程传输，提供诊断网络系统需要的原始数据。具体的设计指标如下：

- 双通道同步数据采集，每通道最高 50KHz 采集速率；
- 每通道输入方式有通常的电压输入、电荷传感器、ICP 传感器输入方式；
- 具有和 PC 的串口通讯能力；
- 人机界面友好；
- 支持网络功能；

## 1.5 技术评价和技术选型

技术的进步使得系统开发变得简易和快速。但是，技术的发展有时候还是会以成本作为代价的，选择成熟、可靠而又成本低廉的开发方式，始终会成为每一个产品研发者首要考虑的问题。当然，还必须要考虑到产品的生命周期，技术领先的时间和空间等因素。

传统仪器设计在需求分析的基础上选择电子元件、布线、制版、调试。现代微电子技术的飞速发展，采用微控制器控制系统的各个部件的技术已经日益成熟。很多高性能的微处理器和微控制器成本低廉，并且开发方式也从过去的一个大程序调用若干个小程序前后台程序设计方式逐步转变成为在微处理器基础上加上嵌入式 RTOS(实时操作系统)，根据任务的划分来进行应用程序开发的设计模式。嵌入式计算机系统是目前计算机应用领域的一个研究热点。

由于计算速度上的限制,数字信号处理技术仅作为一种理论研究的工具并在通用计算机上实现。20 世纪六、七十年代是数字信号处理的理论研究阶段。到了 20 世纪 80 年代初,微电子技术的飞速发展,在快速算法上的理论突破,出现了 DSP 器件。基于信号处理技术的 DSP 器件的具有极快的指令执行速度、专为信号处理设计的指令系统、较大的总线带宽和哈佛结构以及强大的外设控制能力(甚至有些 DSP 还有丰富的片上外设资源)、极短的频谱分析时间(一般的 256 点的 FFT 只需要 2 到 3ms),这些特点十分适合实时信号处理。故障诊断的核心技术就是数据采集和处理,目前的数据处理有很大一部分需要在频域中完成,因此,只有做到了频谱分析的实时性,才能做到真正的在线监测。

在 DSP 和 MPU 集成度和速度越来越高的今天,分立的元器件还在广泛使用着。同时,由于采用了 DSP 和 MPU 的系统设计,导致整个系统的控制复杂。如果还是采用原来的分立元件作为系统的控制单元,会导致如下问题:时钟速度的提高会导致系统的高频特性不好;分立元件功能单一,在复杂逻辑实现中会导致元件数量增多。

CPLD/FPGA 是近年来解决上述问题的一个较好的解决方案,CPLD 是复杂可编程逻辑器件的缩写,FPGA 是现场可编程逻辑阵列的缩写。利用 CPLD/FPGA 的逻辑可编程特性实现原来由分立元件实现的逻辑转换功能,能够做到多(CPLD/FPGA 的逻辑门从几十门到百万门)、快(集成的 CPLD/FPGA 的门延迟都能够控制在 ns 级)、好(主要指价格便宜,相对于同样要买那么多逻辑器件来说)、省(全部的逻辑控制都可以集中在一片 CPLD/FPGA 里面)。

鉴于目前电子设计的这些技术方案,我们决定在本系统中选择 DSP+MPU 的设计模式,由 CPLD 来实现 DSP 和 MPU 的控制逻辑,以提高产品性能和附加值,方便系统扩充,延长产品周期。

## 1.6 系统结构

在这个系统里面,我们采用 Main-Daughters(主从)结构来设计系统。采用我们自行设计的基于 MOTOROLA 公司的 32 位微控制芯片 MC68332 的 CPU 主板,在外扩的局部总线基础上加入自己设计的通讯、人机接口板、数据采集板等子模块,模块化的构建该系统。这样做的好处是单个模块可以单独调试,到最后把功能不断的完善,像写源代码一样,采用 TOP-DOWN 的方法,把每一个部分分隔开来单独调试,全部通过之后再联合调试。

系统结构如图 1.1 所示。CPU 主板主要完成整个系统的整合和模块间的协调。在嵌入式实时操作系统的控制下,读取键盘、串口等的用户请求和信息,将 DSP 板采集来的数据通过液晶界面显示出来。另外,当需要进行网络传输的时候,CPU

主板负责整个传输协议的实现。

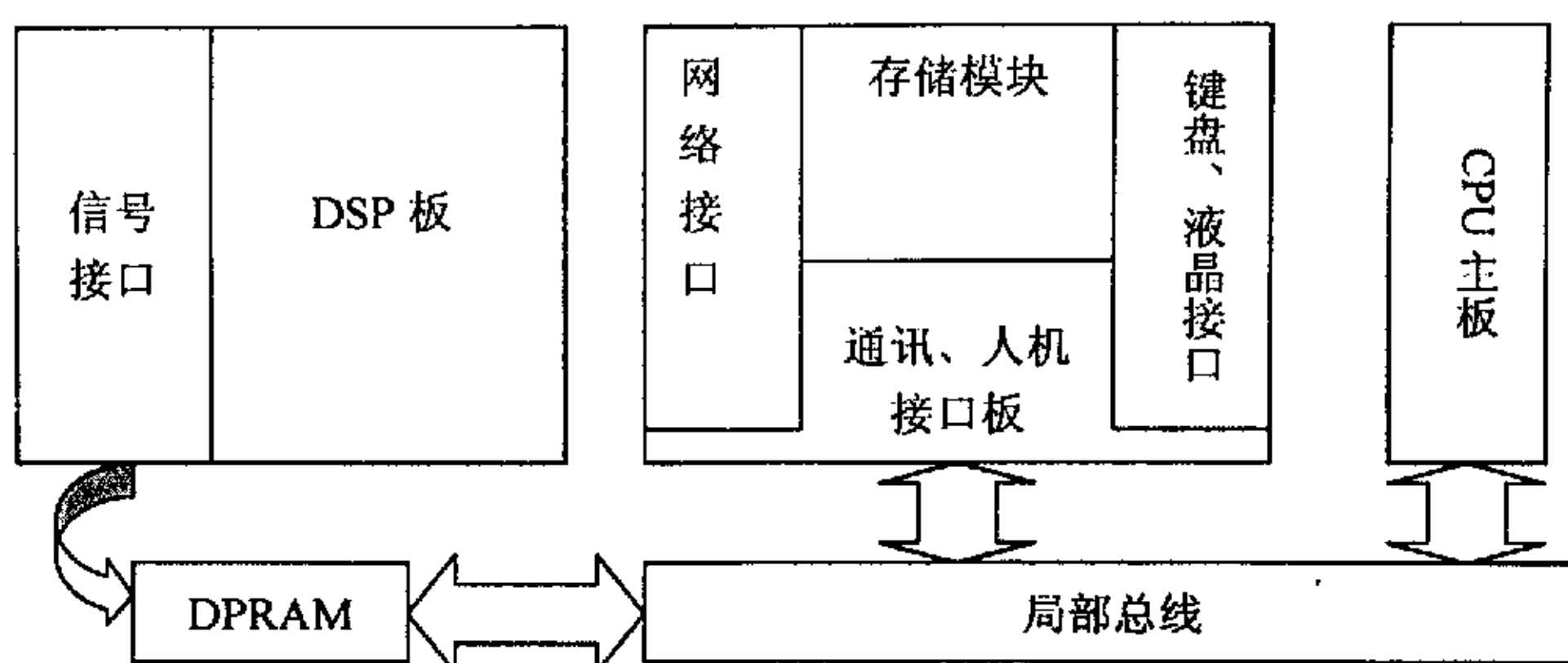


图 1.1 便携式诊断仪的系统结构

在这个系统里，DSP 板主要完成信号的采集分析等功能，通过 DPRAM（双口 RAM）把数据传输给主 CPU 板进行进一步的处理。DSP 板是调试难度最大却是最核心的一个硬件模块。

## 第二章 数据采集分析模块的实现

### 2.1 数据采集技术简介

#### 2.1.1 数据采集的基本原理

数字信息源提供一组离散的可能存在的信息，模拟信号则提供可定义在连续区间上的信息。只有对模拟信号进行采样后，才能通过数字电路进行处理。采样是通过采样脉冲对模拟信号进行抽取的过程，所得到的数据是采样数据。

假设现在对一个模拟信号  $x(t)$  每隔  $\Delta t$  时间采样一次，时间间隔  $\Delta t$  被称为采样间隔或者采样周期，它的倒数  $1/\Delta t$  被称为采样频率，单位是采样数/每秒。 $t=0, \Delta t, 2\Delta t, 3\Delta t, \dots$  等等， $x(t)$  的数值就被称为采样值。所有  $x(0), x(\Delta t), x(2\Delta t)$  都是采样值。这样信号  $x(t)$  可以用一组分散的采样值来表示：

$$\{x(0), x(\Delta t), x(2\Delta t), x(3\Delta t), \dots, x(k\Delta t), \dots\}.$$

下图显示了一个模拟信号和它采样后的采样值。采样间隔是  $\Delta t$ ，注意，采样点在时域上是分散的。

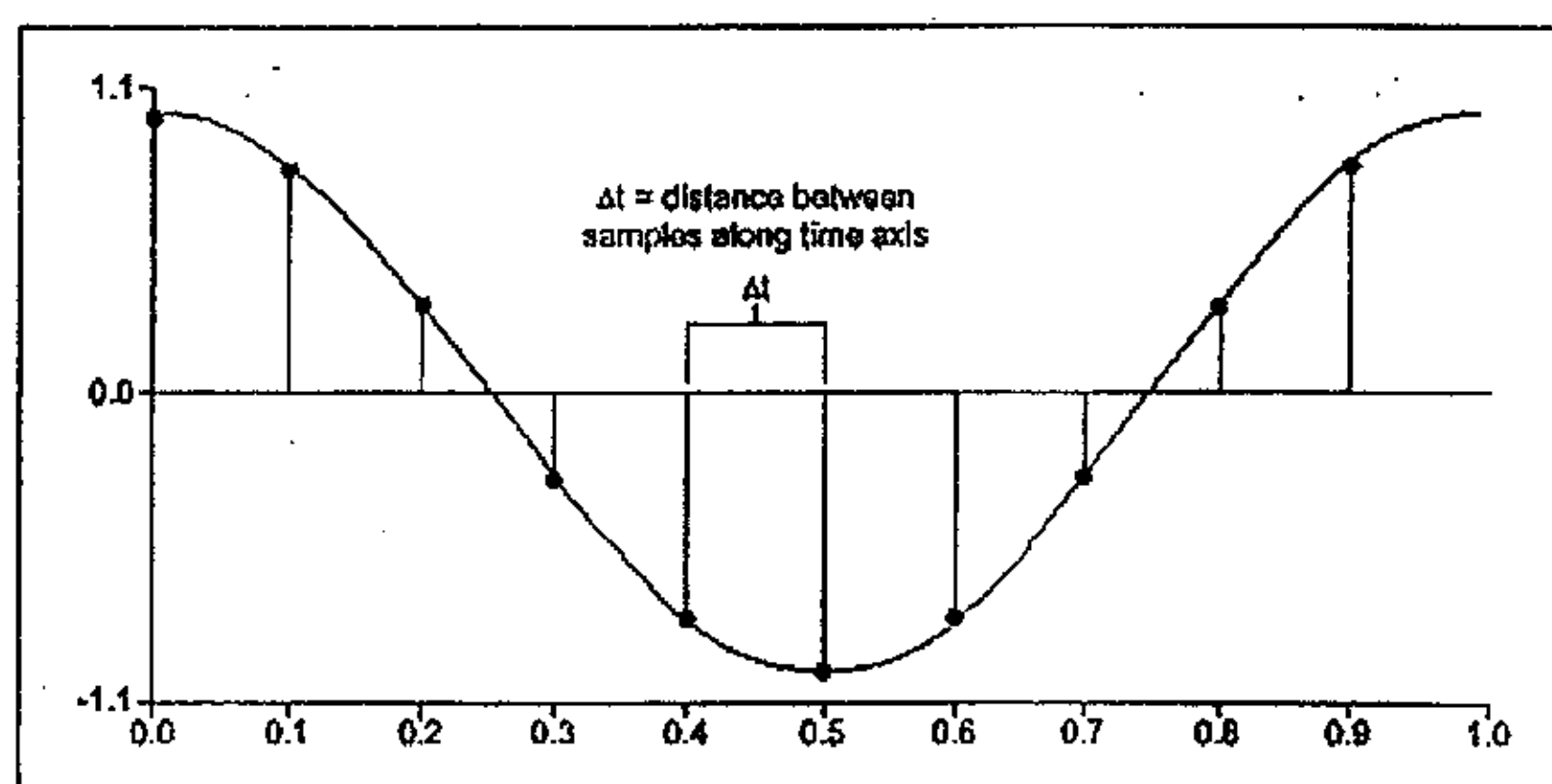


图 2.1 模拟信号和采样显示

如果对信号  $x(t)$  采集  $N$  个采样点，那么  $x(t)$  就可以用下面这个数列表示：

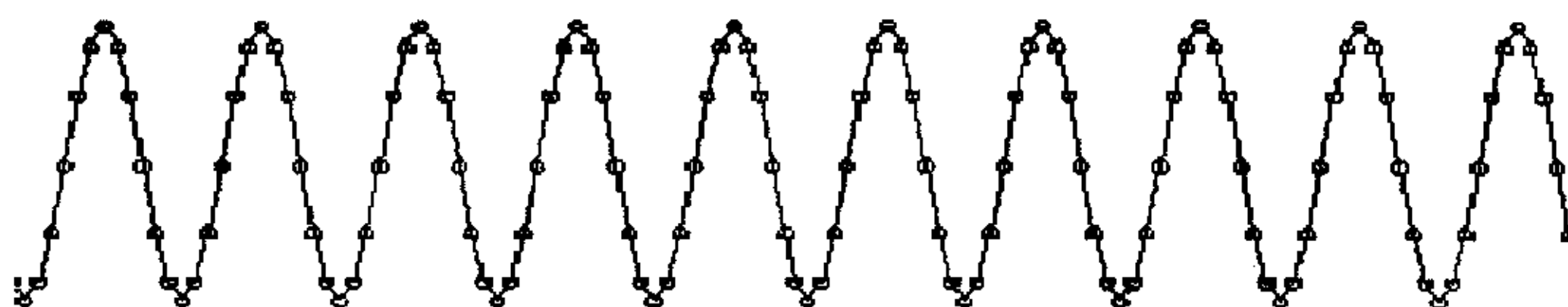
$$X = \{x[0], x[1], x[2], x[3], \dots, x[N-1]\}$$

这个数列被称为信号  $x(t)$  的数字化序列或者采样序列。注意这个数列中仅仅用

下标变量编制索引，而不含有任何关于采样率（或  $\Delta t$ ）的信息。所以如果只知道该信号的采样值，并不能知道它的采样率，缺少了时间尺度，也不可能知道信号  $x(t)$  的频率。

采样定理指出，被采样信号必须是带限信号，采样频率必须是信号带宽的两倍以上。反过来说，如果给定了采样频率，那么能够正确重构信号而不发生畸变的最大频率叫做奈奎斯特频率，它是采样频率的一半。如果信号中包含频率高于奈奎斯特频率的成分，信号将在直流和奈奎斯特频率之间畸变。图 2.2 显示了一个信号分别用合适的采样率和过低的采样率进行采样的结果。

采样率过低的结果是重构后的信号与原始信号的差异。由于采样频率过低而导致的信号畸变叫做混叠（alias）。出现的混频偏差（alias frequency）是输入信号的频率和最靠近的采样率整数倍的差的绝对值。



a) 足够的采样率下的采样结果



b) 过低采样率下的采样结果

图 2.2 不同采样率的采样结果

为了避免频谱混叠的发生，通常在信号被采集（A/D）之前，经过一个低通滤波器，将信号中高于奈奎斯特频率的信号成分滤去。这个滤波器称为抗混叠滤波器。

### 2.1.2 数据采集系统的现状

随着集成电路技术的飞速发展，集成电路不但在集成度上遵循着摩尔定律飞速发展，集成电路的速度也是以很快的速度逐步提高。对于数据采集系统的核心——A/D 转换芯片来说，集成度能够影响 A/D 转换器件的精度。在集成度和设计



方法水平提高后, A/D 转换的精度能够以较低成本取得较大幅度的提高。集成电路的速度却能够影响 A/D 转换的速率。从低速的 TTL、CMOS 器件的广泛应用, 到目前的高速 ECL 器件在电子电路设计中的广泛实现, 都让数据采集模块能够随着器件水平的提高进入高速化、高精度化发展。

单一的数据采集设备仅仅能够采集现场数据, 而目前随着数字信号处理技术的发展, 诊断技术和算法的不断完善, 需要能够对采集到的信号进行实时处理的采集分析系统。集成电路技术的进步也让这一切变成现实。高速 DSP 技术和高速的 CPU 设计技术让整个系统的设计变得简洁。

高性能 PC 技术让虚拟仪器变成现实。在一台普通的 PC 上安装各种总线上的板卡, 加上在 GUI 界面下编写的应用软件, 就能够实现原来的专用仪器的功能。在硬件模块标准化以后, 软件模块也逐渐的向着标准化、可组态发展。NI 公司的 LABVIEW 产品就是一个典型的例子。用户只要安装有 NI 公司的采集卡, 加上 LABVIEW 编程环境, 就可以在一个虚拟的环境下面完成自己的虚拟仪器的开发, 组态灵活, 已经发展到了“傻瓜”式开发的阶段。

随着高速器件的应用, 总线速度就会成为整个采集系统的瓶颈。为了满足高速总线传输的需要, 高速总线也不断的被开发出来。目前可以选用的有两大类: 通用总线和专用仪器总线。

通用总线广泛用于 PC 上的外设, 如 PC/AT、ISA、PCI、USB、IEEE1394 等总线。其中, USB2.0 和 IEEE1394 允许实现设备的级连, 使用非常方便。总线速率也不相同, ISA 总线只能达到 8MHz 总线速率, PCI 能够工作在 33MHz 的总线时钟下。USB1.1 总线时钟支持 1.5MHz 和 12MHz 两种速率, 开发手段较为成熟, USB2.0 总线时钟定义在 480MHz, 但目前还未提供成熟产品。另外, 总线数据总线宽度也不一样, ISA 总线支持 8 位和 16 位数据传输; PCI 总线支持 32 位和 64 位总线传输; 而 USB 和 IEEE1394 总线对于数据的传输却是象网络数据传输一样, 数据是按照数据包来传输的。

专用总线能够提供标准的仪器总线接口, 如 VXI、PXI、GPIB 等仪器总线接口, 目前也在各类采集仪器当中广泛应用。

## 2.2 DSP 数字信号处理系统原理

### 2.2.1 DSP 技术概述

数字信号处理 (Digital Signal Processing, 简称 DSP) 是一门涉及许多学科而又广泛应用于许多领域的新兴学科。20 世纪 60 年代以来, 随着计算机和



信息技术的飞速发展,数字信号处理技术应运而生并得到迅速的发展。在过去的二十多年时间里,数字信号处理已经在通信等领域得到极为广泛的应用<sup>[10]</sup>。

数字信号处理是利用计算机或专用处理设备,以数字形式对信号进行采集、变换、滤波、估值、增强、压缩、识别等处理,以得到符合人们需要的信号形式。数字信号处理是围绕着数字信号处理的理论、实现和应用等几个方面发展起来的。数字信号处理在理论上的发展推动了数字信号处理应用的发展。反过来,数字信号处理的应用又促进了数字信号处理理论的提高。而数字信号处理的实现则是理论和应用之间的桥梁。

数字信号处理是以众多学科为理论基础的,它所涉及的范围极其广泛。例如,在数学领域,微积分、概率统计、随机过程、数值分析等都是数字信号处理的基本工具,与网络理论、信号与系统、控制论、通信理论、故障诊断等也密切相关。近来新兴的一些学科,如人工智能、模式识别、神经网络等,都与数字信号处理密不可分。可以说,数字信号处理是把许多经典的理论体系作为自己的理论基础,同时又使自己成为一系列新兴学科的理论基础。

数字信号处理的实现方法一般有以下几种:

- (1) 在通用的计算机(如 PC 机)上用软件(如 Fortran、C 语言)实现;
- (2) 在通用计算机系统中加上专用的加速处理机实现;
- (3) 用通用的单片机(如 MCS-51、96 系列等)实现,这种方法可用于一些不太复杂的数字信号处理,如数字控制等;
- (4) 用通用的可编程 DSP 芯片实现。与单片机相比,DSP 芯片具有更加适合于数字信号处理的软件和硬件资源,可用于复杂的数字信号处理算法;
- (5) 用专用的 DSP 芯片实现。在一些特殊的场合,要求的信号处理速度极高,用通用 DSP 芯片很难实现,例如专用于 FFT、数字滤波、卷积、相关等算法的 DSP 芯片,这种芯片将相应的信号处理算法在芯片内部用硬件实现,无需进行编程。

在上述几种方法中,第 1 种方法的缺点是速度较慢,一般可用于 DSP 算法的模拟;第 2 种和第 5 种方法专用性强,应用受到很大的限制,第 2 种方法也不便于系统的独立运行;第 3 种方法只适用于实现简单的 DSP 算法;只有第 4 种方法才使数字信号处理的应用打开了新的局面。

虽然数字信号处理的理论发展迅速,但在 20 世纪 80 年代以前,由于实现方法的限制,数字信号处理的理论还得不到广泛的应用。直到 20 世纪 70 年代末 80 年代初世界上第一片单片可编程 DSP 芯片的诞生,才将理论研究结果广泛应用到低成本的实际系统中,并且推动了新的理论和应用领域的发展。可以毫不夸张地说,DSP 芯片的诞生及发展对近 20 年来通信、计算机、控制等领域的技术发展起到十分重要的作用。

### 2.2.2 DSP系统构成

典型的 DSP 系统结构由前向通道和后向通道组成。如下图所示。

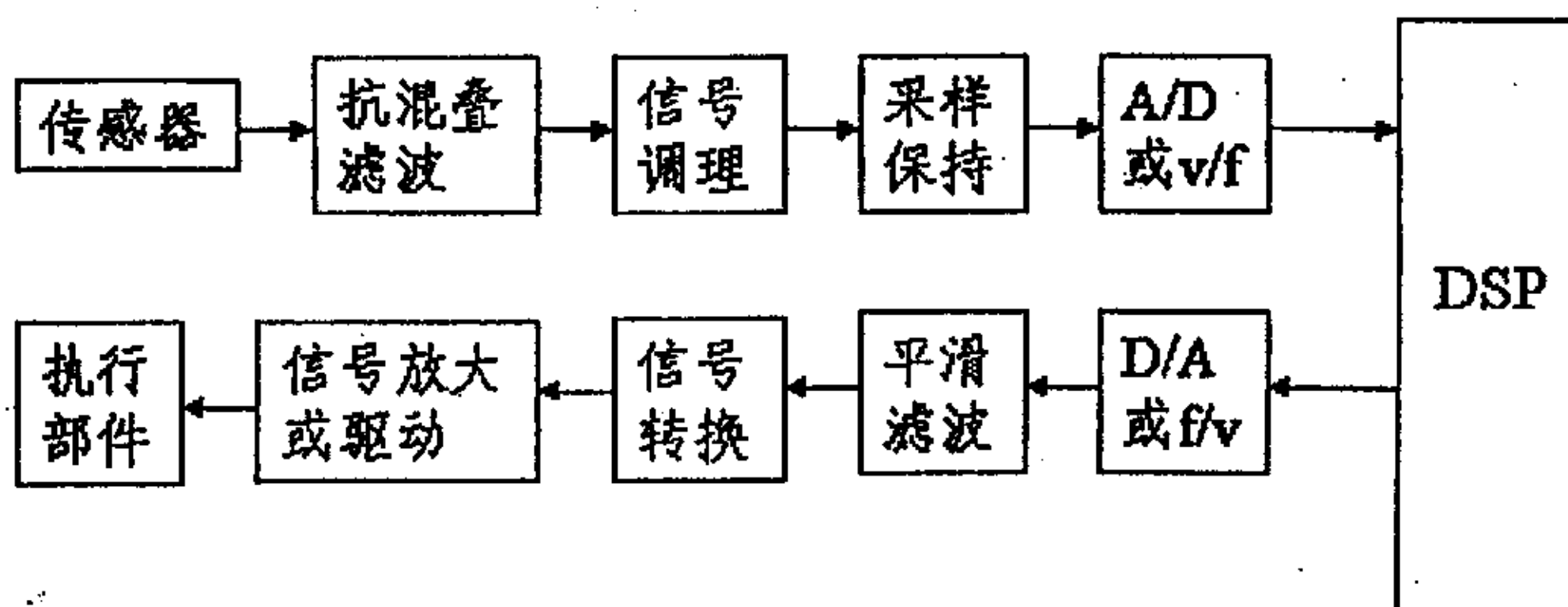


图 2.3 典型的 DSP 系统

输入信号首先进行抗混滤波和信号调理，然后进行 A/D (Analog to Digital) 变换将信号变换成数字比特流。根据前述奈奎斯特抽样定理，为保证信息不丢失，抽样频率至少必须是输入带限信号最高频率的 2 倍。

DSP 芯片的输入是 A/D 变换后得到的以抽样形式表示的数字信号，DSP 芯片对输入的数字信号进行某种形式的处理，如进行一系列的乘累加操作 (MAC)。数字处理是 DSP 的关键，这与其他系统（如电话交换系统）有很大的不同，在交换系统中，处理器的作用是进行路由选择，它并不对输入数据进行修改。因此虽然两者都是实时系统，但两者的实时约束条件却有很大的不同。最后，经过处理后的数字样值再经 D/A (Digital to Analog) 变换转换为模拟样值，之后再进行内插和平滑滤波就可得到连续的模拟波形。

必须指出的是，上面给出的 DSP 系统模型是一个典型模型，但并不是所有的 DSP 系统都必须具有模型中的所有部件。如语音识别系统在输出端并不是连续的波形，而是识别结果，如数字、文字等；有些输入信号本身就是数字信号（如 CD, Compact Disk），因此就不必进行模数变换了。

### 2.2.3 DSP系统的特点

数字信号处理系统是以数字信号处理为基础，因此具有数字处理的全部优点：

- (1) 接口方便：DSP 系统与其他以现代数字技术为基础的系统或设备都是相互兼容的，与这样的系统接口以实现某种功能要比模拟系统与这些系统接口要容易得多；

(2) 编程方便: DSP 系统中的可编程 DSP 芯片可使设计人员在开发过程中灵活方便地对软件进行修改和升级;

(3) 稳定性好: DSP 系统以数字处理为基础, 受环境温度以及噪声的影响较小, 可靠性高;

(4) 精度高: 16 位数字系统可以达到  $10^{-5}$  的精度;

(5) 可重复性好: 模拟系统的性能受元器件参数性能变化比较大, 而数字系统基本不受影响, 因此数字系统便于测试、调试和大规模生产;

(6) 集成方便: DSP 系统中的数字部件有高度的规范性, 便于大规模集成。

当然, 数字信号处理也存在一定的缺点。例如, 对于简单的信号处理任务, 如与模拟交换线的电话接口, 若采用 DSP 则使成本增加。DSP 系统中的高速时钟可能带来高频干扰和电磁泄漏等问题, 而且 DSP 系统消耗的功率也较大。此外, DSP 技术更新的速度快, 数学知识要求多, 开发和调试工具还不尽完善。

虽然 DSP 系统存在着一些缺点, 但其突出的优点已经使之在通信、语音、图像、雷达、生物医学、工业控制、仪器仪表等许多领域得到越来越广泛的应用。

#### 2.2.4 DSP 系统的设计过程

总的来说, DSP 系统的设计还没有非常好的正规设计方法。图 1.2 所示是 DSP 系统设计的一般过程。

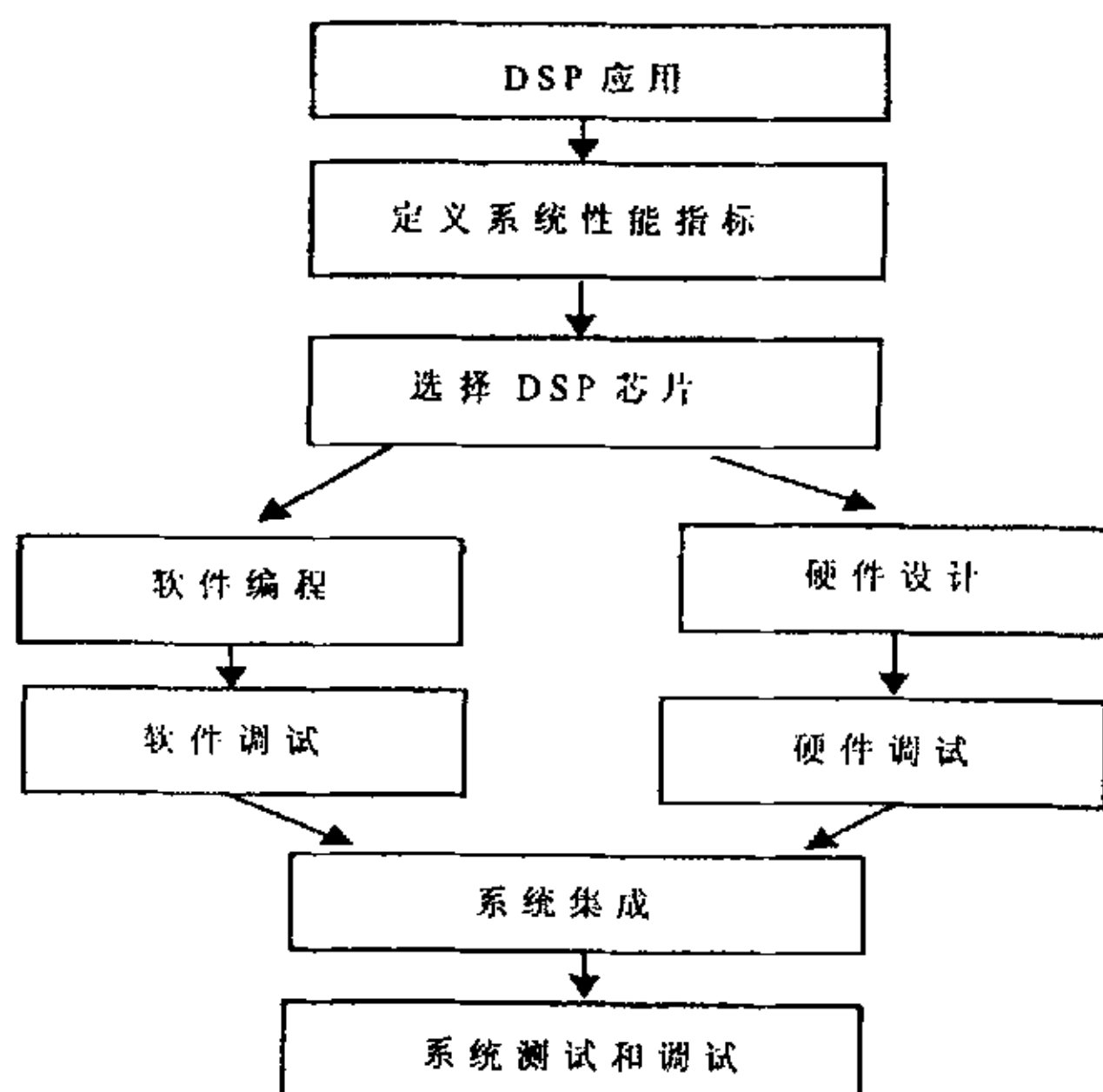


图 2.4 DSP 系统的设计流程

在设计 DSP 系统之前, 首先必须根据应用系统的目标确定系统的性能指标、

信号处理的要求,通常可用数据流程图、数学运算序列、正式的符号或自然语言来描述。

第二步是根据系统的要求进行高级语言的模拟。一般来说,为了实现系统的最终目标,需要对输入的信号进行适当的处理,而处理方法的不同会导致不同的系统性能,要得到最佳的系统性能,就必须在这一步确定最佳的处理方法,即数字信号处理的算法(Algorithm),因此这一步也称算法模拟阶段。例如,语音压缩编码算法就是要在确定的压缩比条件下,获得最佳的合成语音。算法模拟所用的输入数据是实际信号经采集而获得的,通常以计算机文件的形式存储为数据文件。如语音压缩编码算法模拟时所用的语音信号就是实际采集而获得并存储为计算机文件形式的语音数据文件。有些算法模拟时所用的输入数据并不一定要是实际采集的信号数据,只要能够验证算法的可行性,输入假设的数据也是可以的。

在完成第二步之后,接下来就可以设计实时 DSP 系统,实时 DSP 系统的设计包括硬件设计和软件设计两个方面。硬件设计首先要根据系统运算量的大小、对运算精度的要求、系统成本限制以及体积、功耗等要求选择合适的 DSP 芯片。然后设计 DSP 芯片的外围电路及其它电路。软件设计和编程主要根据系统要求和所选的 DSP 芯片编写相应的 DSP 汇编程序,若系统运算量不大且有高级语言编译器支持,也可用高级语言(如 C 语言)编程。由于现有的高级语言编译器的效率还比不上手工编写汇编语言的效率,因此在实际应用系统中常常采用高级语言和汇编语言的混合编程方法,即在算法运算量大的地方,用手工编写的方法编写汇编语言,而运算量不大的地方则采用高级语言。采用这种方法,既可缩短软件开发的周期,提高程序的可读性和可移植性,又能满足系统实时运算的要求。

DSP 硬件和软件设计完成后,就需要进行硬件和软件的调试。软件的调试一般借助于 DSP 开发工具,如软件模拟器、DSP 开发系统或仿真器等。调试 DSP 算法时一般采用比较实时结果与模拟结果的方法,如果实时程序和模拟程序的输入相同,则两者的输出应该一致。应用系统的其它软件可以根据实际情况进行调试。硬件调试一般采用硬件仿真器进行调试,如果没有相应的硬件仿真器,且硬件系统不是十分复杂,也可以借助于一般的工具进行调试。

系统的软件和硬件分别调试完成后,就可以将软件脱离开发系统而直接在实际应用系统上运行。当然,DSP 系统的开发,特别是软件开发是一个需要反复进行的过程,虽然通过算法模拟基本上可以知道实时系统的性能,但实际上模拟环境不可能做到与实时系统环境完全一致,而且将模拟算法移植到实时系统时必须考虑算法是否能够实时运行的问题。如果算法运算量太大不能在硬件上实时运行,则必须重新修改或简化算法。

### 2.2.5 可编程 DSP 芯片



DSP 芯片, 也称数字信号处理器, 是一种特别适合于进行数字信号处理运算的微处理器, 其主要应用是实时快速地实现各种数字信号处理算法。根据数字信号处理的要求, DSP 芯片一般具有如下主要特点:

- (1) 在一个指令周期内可完成一次乘法和一次加法;
- (2) 程序和数据空间分开, 可以同时访问指令和数据;
- (3) 片内具有快速 RAM, 通常可通过独立的数据总线在两块中同时访问;
- (4) 具有低开销或无开销循环及跳转的硬件支持;
- (5) 快速的中断处理和硬件 I/O 支持;
- (6) 具有在单周期内操作的多个硬件地址产生器;
- (7) 可以并行执行多个操作;
- (8) 支持流水线操作, 使取指、译码和执行等操作可以重叠执行。

当然, 与通用微处理器相比, DSP 芯片的其他通用功能相对较弱些。

### 2.2.5.1 DSP 芯片的发展<sup>[11]</sup>

世界上第一个单片 DSP 芯片应当是 1978 年 AMI 公司发布的 S2811, 1979 年美国 Intel 公司发布的商用可编程器件 2920 是 DSP 芯片的一个主要里程碑。这两种芯片内部都没有现代 DSP 芯片所必须有的单周期乘法器。1980 年, 日本 NEC 公司推出的  $\mu$ PD7720 是第一个具有乘法器的商用 DSP 芯片。

在这之后, 最成功的 DSP 芯片当数美国德州仪器公司 (Texas Instruments, 简称 TI) 的一系列产品。TI 公司在 1982 年成功推出其第一代 DSP 芯片 TMS32010 及其系列产品 TMS32011、TMS320C10/C14/C15/C16/C17 等, 之后相继推出了第二代 DSP 芯片 TMS32020、TMS320C25/C26/C28, 第三代 DSP 芯片 TMS320C30/C31/C32, 第四代 DSP 芯片 TMS320C40/C44, 第五代 DSP 芯片 TMS320C5X/C54X, 第二代 DSP 芯片的改进型 TMS320C2XX, 集多片 DSP 芯片于一体的高性能 DSP 芯片 TMS320C8X 以及目前速度最快的第六代 DSP 芯片 TMS320C62X/C67X 等。TI 将常用的 DSP 芯片归纳为三大系列, 即: TMS320C2000 系列 (包括 TMS320C2X/C2XX)、TMS320C5000 系列 (包括 TMS320C5X/C54X/C55X)、TMS320C6000 系列 (TMS320C62X/C67X)。如今, TI 公司的一系列 DSP 产品已经成为当今世界上最有影响的 DSP 芯片。TI 公司也成为世界上最大的 DSP 芯片供应商, 其 DSP 市场份额占全世界份额近 50%。

第一个采用 CMOS 工艺生产浮点 DSP 芯片的是日本的 Hitachi 公司, 它于 1982 年推出了浮点 DSP 芯片。1983 年日本 Fujitsu 公司推出的 MB8764, 其指令周期为 120ns, 且具有双内部总线, 从而使处理吞吐量发生了一个大的飞跃。而第一个高性能浮点 DSP 芯片应是 AT&T 公司于 1984 年推出的 DSP32。

与其它公司相比, Motorola 公司在推出 DSP 芯片方面相对较晚。1986 年,

该公司推出了定点处理器 MC56001。1990 年, 推出了与 IEEE 浮点格式兼容的浮点 DSP 芯片 MC96002。

美国模拟器件公司 (Analog Devices, 简称 AD) 在 DSP 芯片市场上也占有一定的份额, 相继推出了一系列具有自己特点的 DSP 芯片, 其定点 DSP 芯片有 ADSP2101/2103/2105、ASDP2111/2115、ADSP2161/2162/2164 以及 ADSP2171/2181, 浮点 DSP 芯片有 ADSP21000/21020、ADSP21060/21062 等。

自 1980 年以来, DSP 芯片得到了突飞猛进的发展, DSP 芯片的应用越来越广泛。从运算速度来看, MAC (一次乘法和一次加法) 时间已经从 20 世纪 80 年代初的 400ns (如 TMS32010) 降低到 10ns 以下 (如 TMS320C54X、TMS320C62X/67X 等), 处理能力提高了几十倍。DSP 芯片内部关键的乘法器部件从 1980 年的占模片区 (die area) 的 40% 左右下降到 5% 以下, 片内 RAM 数量增加一个数量级以上。从制造工艺来看, 1980 年采用  $4\mu\text{m}$  的 N 沟道 MOS (NMOS) 工艺, 而现在则普遍采用亚微米 (Micron) CMOS 工艺。DSP 芯片的引脚数量从 1980 年的最多 64 个增加到现在的 200 个以上, 引脚数量的增加, 意味着结构灵活性的增加, 如外部存储器的扩展和处理器间的通信等。此外, DSP 芯片的发展使 DSP 系统的成本、体积、重量和功耗都有很大程度的下降。表 1.1 是 TI 公司 DSP 芯片 1982 年、1992 年、1999 年的比较表。

表 2.1 TI DSP 芯片发展比较表 (典型值)

年份	1982 年	1992 年	1999 年
制造工艺	$4\mu\text{m}$ NMOS	$0.8\mu\text{m}$ CMOS	$0.3\mu\text{m}$ CMOS
MIPS	5MIPS	40MIPS	100MIPS
MHz	20MHz	80MHz	100MHz
内部 RAM	144 字	1K 字	32K 字
内部 ROM	1.5K 字	4K 字	16K 字
价格	\$150.00	\$15.00	\$5.00~\$25.00
功耗	250mW/MIPS	12.5mW/MIPS	0.45mW/MIPS
集成晶体管数	50K	500K	

#### 2.2.5.2 DSP 芯片的选择

设计 DSP 应用系统, 选择 DSP 芯片是非常重要的一个环节。只有选定了 DSP 芯片, 才能进一步设计其外围电路及系统的其他电路。总的来说, DSP 芯片的选择应根据实际的应用系统需要而确定。不同的 DSP 应用系统由于应用场合、应用目的等不尽相同, 对 DSP 芯片的选择也是不同的。一般来说, 选择 DSP 芯片时应考虑到如下诸多因素。



1. DSP 芯片的运算速度。运算速度是 DSP 芯片的一个最重要的性能指标,也是选择 DSP 芯片时所需要考虑的一个主要因素。DSP 芯片的运算速度可以用以下几种性能指标来衡量:

(1) 指令周期:即执行一条指令所需的时间,通常以 ns(纳秒)为单位。如 TMS320LC549-80 在主频为 80MHz 时的指令周期为 12.5ns;

(2) MAC 时间:即一次乘法加上一次加法的时间。大部分 DSP 芯片可在一个指令周期内完成一次乘法和加法操作,如 TMS320LC549-80 的 MAC 时间就是 12.5ns;

(3) FFT 执行时间:即运行一个 N 点 FFT 程序所需的时间。由于 FFT 运算涉及的运算在数字信号处理中很有代表性,因此 FFT 运算时间常作为衡量 DSP 芯片运算能力的一个指标;

(4) MIPS:即每秒执行百万条指令。如 TMS320LC549-80 的处理能力为 80 MIPS,即每秒可执行八千万条指令;

(5) MOPS:即每秒执行百万次操作。如 TMS320C40 的运算能力为 275 MOPS;

(6) MFLOPS:即每秒执行百万次浮点操作。如 TMS320C31 在主频为 40MHz 时的处理能力为 40 MFLOPS;

(7) BOPS:即每秒执行十亿次操作。如 TMS320C80 的处理能力为 2 BOPS。

2. DSP 芯片的价格。DSP 芯片的价格也是选择 DSP 芯片所需考虑的一个重要因素。如果采用价格昂贵的 DSP 芯片,即使性能再高,其应用范围肯定会受到一定的限制,尤其是民用产品。因此根据实际系统的应用情况,需确定一个价格适中的 DSP 芯片。当然,由于 DSP 芯片发展迅速,DSP 芯片的价格往往下降较快,因此在开发阶段选用某种价格稍贵的 DSP 芯片,等到系统开发完毕,其价格可能已经下降一半甚至更多。

3. DSP 芯片的硬件资源。不同的 DSP 芯片所提供的硬件资源是不相同的,如片内 RAM、ROM 的数量,外部可扩展的程序和数据空间,总线接口,I/O 接口等。即使是同一系列的 DSP 芯片(如 TI 的 TMS320C54X 系列),系列中不同 DSP 芯片也具有不同的内部硬件资源,可以适应不同的需要。

4. DSP 芯片的运算精度。一般的定点 DSP 芯片的字长为 16 位,如 TMS320 系列。但有的公司的定点芯片为 24 位,如 Motorola 公司的 MC56001 等。浮点芯片的字长一般为 32 位,累加器为 40 位。

5. DSP 芯片的开发工具。在 DSP 系统的开发过程中,开发工具是必不可少的。如果没有开发工具的支持,要想开发一个复杂的 DSP 系统几乎是不可能的。如果有功能强大的开发工具的支持,如 C 语言支持,则开发的时间就会大大缩短。所以,在选择 DSP 芯片的同时必须注意其开发工具的支持情况,包括软件和硬件的开发工具。

6. DSP 芯片的功耗。在某些 DSP 应用场合, 功耗也是一个需要特别注意的问题。如便携式的 DSP 设备、手持设备、野外应用的 DSP 设备等都对功耗有特殊的要求。目前, 3.3V 供电的低功耗高速 DSP 芯片已大量使用。

7. 其他。除了上述因素外, 选择 DSP 芯片还应考虑到封装的形式、质量标准、供货情况、生命周期等。有的 DSP 芯片可能有 DIP、PGA、PLCC、PQFP 等多种封装形式。有些 DSP 系统可能最终要求的是工业级或军用级标准, 在选择时就需要注意到所选的芯片是否有工业级或军用级的同类产品。如果所设计的 DSP 系统不仅仅是一个实验系统, 而是需要批量生产并可能有几年甚至十几年的生命周期, 那么需要考虑所选的 DSP 芯片供货情况如何, 是否也有同样甚至更长的生命周期等。

在上述诸多因素中, 一般而言, 定点 DSP 芯片的价格较便宜, 功耗较低, 但运算精度稍低。而浮点 DSP 芯片的优点是运算精度高, 且 C 语言编程调试方便, 但价格稍贵, 功耗也较大。例如 TI 的 TMS320C2XX/C54X 系列属于定点 DSP 芯片, 低功耗和低成本是其主要的特点。而 TMS320C3X/C4X/C67X 属于浮点 DSP 芯片, 运算精度高, 用 C 语言编程方便, 开发周期短, 但同时其价格和功耗也相对较高。

DSP 应用系统的运算量是确定选用处理能力为多大的 DSP 芯片的基础。运算量小则可以选用处理能力不是很强的 DSP 芯片, 从而可以降低系统成本。相反, 运算量大的 DSP 系统则必须选用处理能力强的 DSP 芯片, 如果 DSP 芯片的处理能力达不到系统要求, 则必须用多个 DSP 芯片并行处理。那么如何确定 DSP 系统的运算量以选择 DSP 芯片呢? 下面我们来考虑两种情况。

### 1. 按样点处理

所谓按样点处理就是 DSP 算法对每一个输入样点循环一次。数字滤波就是这种情况。在数字滤波器中, 通常需要对每一个输入样点计算一次。例如, 一个采用 LMS 算法的 256 抽头的自适应 FIR 滤波器, 假定每个抽头的计算需要 3 个 MAC 周期, 则 256 抽头计算需要  $256 \times 3 = 768$  个 MAC 周期。如果采样频率为 8kHz, 即样点之间的间隔为  $125\mu\text{s}$ , DSP 芯片的 MAC 周期为 200ns, 则 768 个 MAC 周期需要  $153.6\mu\text{s}$  的时间, 显然无法实时处理, 需要选用速度更高的 DSP 芯片。表 1.3 示出了两种信号带宽对三种 DSP 芯片的处理要求, 三种 DSP 芯片的 MAC 周期分别为 200ns、50ns 和 25ns。从表中可以看出, 对话带的应用, 后两种 DSP 芯片可以实时实现, 对声频应用, 只有第三种 DSP 芯片能够实时处理。当然, 在这个例子中, 没有考虑其它的运算量。

### 2. 按帧处理

有些数字信号处理算法不是每个输入样点循环一次, 而是每隔一定的时间间隔 (通常称为帧) 循环一次。例如, 中低速语音编码算法通常以 10ms 或 20ms

为一帧，每隔 10ms 或 20ms 语音编码算法循环一次。所以，选择 DSP 芯片时应该比较一帧内 DSP 芯片的处理能力和 DSP 算法的运算量。假设 DSP 芯片的指令周期为  $p$  (ns)，一帧的时间为  $\Delta\tau$  (ns)，则该 DSP 芯片在一帧内所能提供的最大运算量为  $\Delta\tau/p$  条指令。例如 TMS320LC549-80 的指令周期为 12.5ns，设帧长为 20ms，则一帧内 TMS320LC549-80 所能提供的最大运算量为 160 万条指令。

表2.2 用DSP芯片实现数字滤波

应用领域	采样率 (KHz)	采样周期 ( $\mu$ s)	256 抽头 LMS 滤波运算量 (MAC 数)	每样点允许 MAC 指令数 (200ns)	每样点允许 MAC 指令数 (50ns)	每样点允许 MAC 指令数 (25ns)
语音	8	125	768	625	2500	5000
声频	44.1	22.7	768	113	453	907

因此，只要语音编码算法的运算量不超过 160 万条指令，就可以在 TMS320LC549-80 上实时运行。

### 2.2.5.3 DSP芯片的基本结构

为了快速地实现数字信号处理运算，DSP 芯片一般都采用特殊的软硬件结构。下面以 TMS320 系列为例介绍 DSP 芯片的基本结构。

TMS320 系列 DSP 芯片的基本结构包括：(1) 哈佛结构；(2) 流水线操作；(3) 专用的硬件乘法器；(4) 特殊的 DSP 指令；(5) 快速的指令周期。

这些特点使得 TMS320 系列 DSP 芯片可以实现快速的 DSP 运算，并使大部分运算（例如乘法）能够在指令周期内完成。由于 TMS320 系列 DSP 芯片是软件可编程器件，因此具有通用微处理器具有的方便灵活的特点。下面分别介绍这些特点是如何在 TMS320 系列 DSP 芯片中应用并使得芯片的功能得到加强的。

#### 哈佛结构

哈佛结构是不同于传统的冯·诺曼 (Von Neuman) 结构的并行体系结构，其主要特点是将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个相互独立的存储器，每个存储器独立编址，独立访问。与两个存储器相对应的是系统中设置了程序总线 and 数据总线两条总线，从而使数据的吞吐率提高了一倍。而冯·诺曼结构则是将指令、数据、地址存储在同一存储器中，统一编址，依靠指令计数器提供的地址来区分是指令、数据还是地址。取指令和取数据都访问同一存储器，数据吞吐率低。

在哈佛结构中,由于程序和数据存储器在两个分开的空间中,因此取指和执行能完全重叠运行。为了进一步提高运行速度和灵活性,TMS320 系列 DSP 芯片在基本哈佛结构的基础上作了改进,一是允许数据存放在程序存储器中,并被算术运算指令直接使用,增强了芯片的灵活性;二是指令存储在高速缓冲器(Cache)中,当执行此指令时,不需要再从存储器中读取指令,节约了一个指令周期的时间。如 TMS320C30 具有 64 个字的 Cache。

### 流水线

与哈佛结构相关,DSP 芯片广泛采用流水线以减少指令执行时间,从而增强了处理器的处理能力。TMS320 系列处理器的流水线深度从 2~6 级不等。第一代 TMS320 处理器采用二级流水线,第二代采用三级流水线,而第三代则采用四级流水线。也就是说,处理器可以并行处理 2~6 条指令,每条指令处于流水线上的不同阶段。图 2.5 所示为一个三级流水线操作的例子。

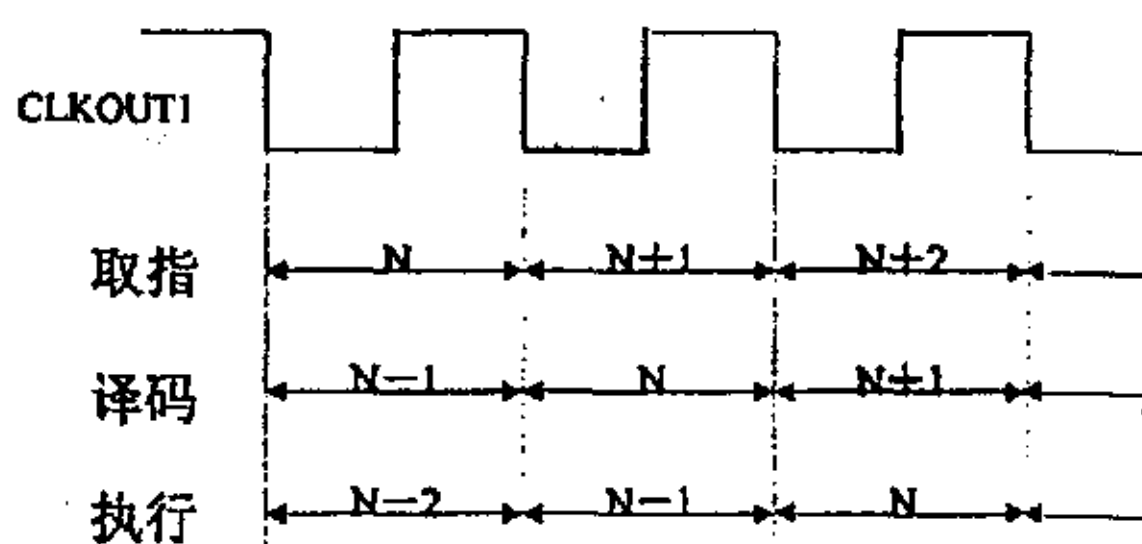


图 2.5 三级流水线操作

在三级流水线操作中,取指、译码和执行操作可以独立地处理,这可使指令执行能完全重叠。在每个指令周期内,三个不同的指令处于激活状态,每个指令处于不同的阶段。例如,在第  $N$  个指令取指时,前一个指令即第  $N-1$  个指令正在译码,而第  $N-2$  个指令则正在执行。一般来说,流水线对用户是透明的。

### 专用的硬件乘法器

在一般形式的 FIR 滤波器中,乘法是 DSP 的重要组成部分。对每个滤波器抽头,必须做一次乘法和一次加法。乘法速度越快,DSP 处理器的性能就越高。在通用的微处理器中,乘法指令是由一系列加法来实现的,故需许多个指令周期来完成。相比而言,DSP 芯片的特征就是有一个专用的硬件乘法器。在 TMS320 系列中,由于具有专用的硬件乘法器,乘法可在一个指令周期内完成。从最早的 TMS32010 实现 FIR 的每个抽头算法可以看出,滤波器每个抽头需要一条乘法指令 MPY:

LT                   ; 装乘数到 T 寄存器



DMOV : 在存储器中移动数据以实现延迟

MPY : 相乘

APAC : 将乘法结果加到 ACC 中

其他三条指令用来将乘数装入到乘法器电路 (LT), 移动数据 (DMOV) 以及将乘法结果 (存在乘积寄存器 P 中) 加到 ACC 中 (APAC)。因此, 若采用 256 抽头的 FIR 滤波器, 这四条指令必须重复执行 256 次, 且 256 次乘法必须在一个抽样间隔内完成。在典型的通用微处理器中, 每个抽头需要 30 ~ 40 个指令周期, 而 TMS32010 只需 4 条指令。如果采用特殊的 DSP 指令或采用 TMS320C54X 等新一代的 DSP 芯片, 可进一步降低 FIR 抽头的计算时间。

### 特殊的DSP指令

DSP 芯片的另一个特征是采用特殊的指令。2.2.3 节中介绍的 DMOV 就是一个特殊的 DSP 指令, 它完成数据移位功能。在数字信号处理中, 延迟操作非常重要, 这个延迟就是由 DMOV 来实现的。TMS32010 中的另一个特殊指令是 LTD, 它在一个指令周期内完成 LT、DMOV 和 APAC 三条指令。LTD 和 MPY 指令可以将 FIR 滤波器抽头计算从 4 条指令降为 2 条指令。在第二代处理器中, 如 TMS320C25, 增加了 2 条更特殊的指令, 即 RPT 和 MACD 指令, 采用这 2 条特殊指令, 可以进一步将每个抽头的运算指令数从 2 条降为 1 条:

RPTK 255 : 重复执行下条指令 256 次

MACD : LT, DMOV, MPY 及 APAC

### 快速的指令周期

哈佛结构、流水线操作、专用的硬件乘法器、特殊的 DSP 指令再加上集成电路的优化设计, 可使 DSP 芯片的指令周期在 200ns 以下。TMS320 系列处理器的指令周期已经从第一代的 200ns 降低至现在的 20ns 以下。快速的指令周期使得 DSP 芯片能够实时实现许多 DSP 应用。

## 2.3 数据采集分析系统设计

数据采集分析系统完成模拟信号的调理、采集、分析功能, 并把分析的结果传输给主控模块。我们设计的采集分析系统模块采用了 TMS320LF2407 作为该系统的主控 DSP, 并在该平台基础上建立了和主控模块的数据命令连结。该模块主要完成数据采集、根据主控模块的要求进行的参数设置进行数字信号处理、处理后的数据放在通讯双口 RAM 当中传输给主控模块进行显示和保存, 并把诊断结果输出。如下图 2.6 是整个采集分析系统的模块连接图, 其中, 主要画出了信号调

理和抗混滤波的主体部分。和主控模块的通讯部分在后边进行说明。

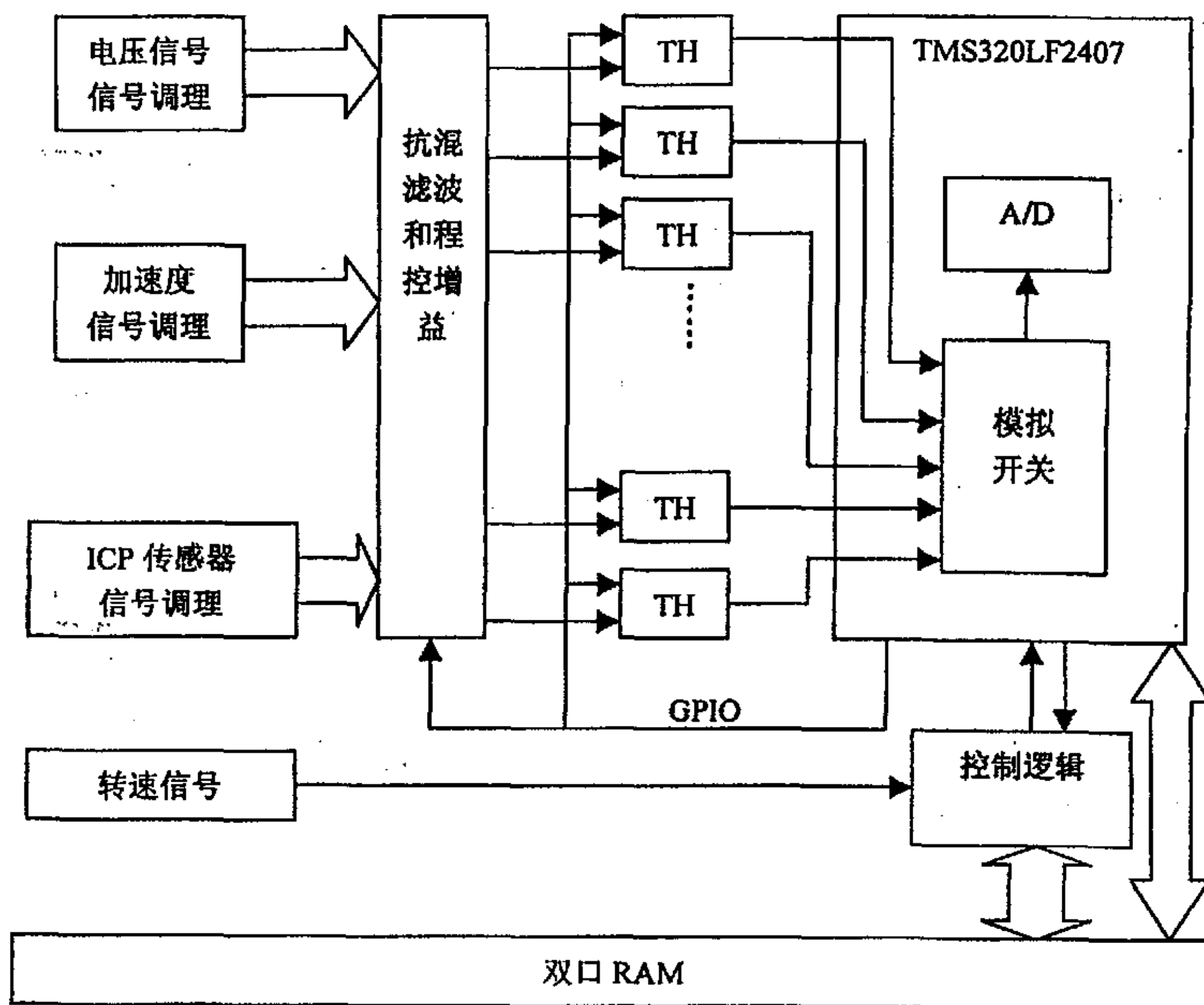


图 2.6 数据采集分析系统结构

### 2.3.1 抗混滤波器设计

信号在被采集之前一定要经过抗混滤波器。传统的抗混滤波器设计采用的都是无源元件，如广泛使用的 RC 滤波器。这一类滤波器设计上有很多缺点<sup>[10]</sup>：

- 截止频率不可调，特别是对于通用的数据采集系统，采集频率可以人为设定也要求截止频率能够人为设定，而通常的做法是把截止频率固定为系统最高的采集频率乘以一个大于二的系数作为最终的截止频率。
- 高阶滤波器设计复杂，采用元件较多，参数调试不方便。
- 有源器件和无源器件等元件之间的分布参数复杂，滤波器性能很难保证。
- RC 无源滤波器的截止频率非线性，依靠调节 RC 元件的参数调节很难去适应这个非线性，灵活性差。

正是由于应用系统对滤波器的特殊要求，刺激了集成有源滤波器、可编程有源滤波器的发展。有源滤波器的发展，一方面提供了高精度、高稳定度、高性



能的集成有源滤波器,另一方面提供了通用的有源滤波器。可编程有源滤波器的出现是开关电容技术发展的结果,正是由于开关电容技术的成熟使得可编程有源滤波器的实现成为可能。

许多公司都提供了基于开关电容技术的可编程有源滤波器产品,最突出的是 Maxim 公司的可编程有源滤波器。这类产品分为三类:时钟可编程低通滤波器产品、引脚可编程的带通滤波器以及能用于低通、高通、带通、带阻等场合的通用可编程有源滤波器。

在这个系统当中,必须能够对信号进行从直流到 50kHz 的交流信号进行抗混滤波,因此,我们选用了 MAX262 通用可编程有源滤波器。

MAX262 是 MAXIM 公司采用 CMOS 技术的双二阶通用开关电容有源滤波器,可以方便的和 DSP、CPU 接口。无需任何附加元件可通过编程实现各类低通、高通、带通、陷波和全通滤波器配置。处理器可编程的参数有中心频率  $f_0$ 、品质因数  $Q$ 、滤波器工作域等。 $f_0$  和  $Q$  可编程且相互独立,互相解耦。输入时钟频率由 SP 输入、输入时钟和 6 位  $f_0$  可编程输入代码共同决定滤波器的中心频率和截止频率。

### 2.3.2 程控增益放大器

有些时候,对于不同的信号,需要把信号放大到不同的倍数,以满足不同动态范围的信号的需要。因此,在仪器仪表、高性能 A/D 采集系统,程控增益放大器广泛使用。可编程放大器是指可通过软件设置对增益进行编程的运算放大器。A/D 公司、BB 公司都提供满足各种要求的可编程放大器产品,如 AD 公司的 AD526、BB 公司的 PGA102/103/202/203/204/205 等。可编程运放的增益编程方式有的是二进制的,如 1 倍、2 倍、4 倍、8 倍,以此类推;有的是十进制的,如 1 倍、10 倍、100 倍、1000 倍,以此类推。

#### PGA203

PGA203 是 BB 公司的可编程增益运放。通过数字电平直接选择的编程增益为  $X1$ 、 $X10$ 、 $X100$ 。它是一种高速器件,具有良好的动态响应。在增益为 100 时,带宽可以达到 250kHz。该器件广泛应用在高速数据采集系统、多增益模拟放大系统和医用仪器中。器件采用 DIP8 和 SO-8 封装。

主要性能指标为增益误差不大于  $\pm 0.05\%$  ( $G=10$ );失调电压偏移  $2\mu V/^\circ C$ ;非线性度不大于  $\pm 0.001\%$ ;噪声电压峰值不大于  $0.6\mu V$ ;数字电平兼容 CMOS 和 TTL 电平。

### 2.3.3 信号调理

#### 2.3.3.1 加速度传感器信号调理

电压信号调理主要对电压信号进行输入调理。输入的电压信号如果是一般的电压信号，只需要进行限幅就可以了。加速度信号通常采用的是压电加速度传感器，输出是电荷信号，需要进行电荷到电压的转化。

压电加速度传感器的信号调理如图 2.7，压电加速度传感器输出电荷信号经电荷放大器转化为电压信号，此信号再经过高通滤波器输出，产生与振动加速度成正比的模拟信号  $a$ 。此模拟信号  $a$  经过有源积分器和放大之后，产生与振动速度成正比的信号  $v$ 。模拟信号  $v$  再经过有源积分器和放大之后，产生与振动位移成正比的信号  $s$ ，送入抗混滤波和程控增益单元。

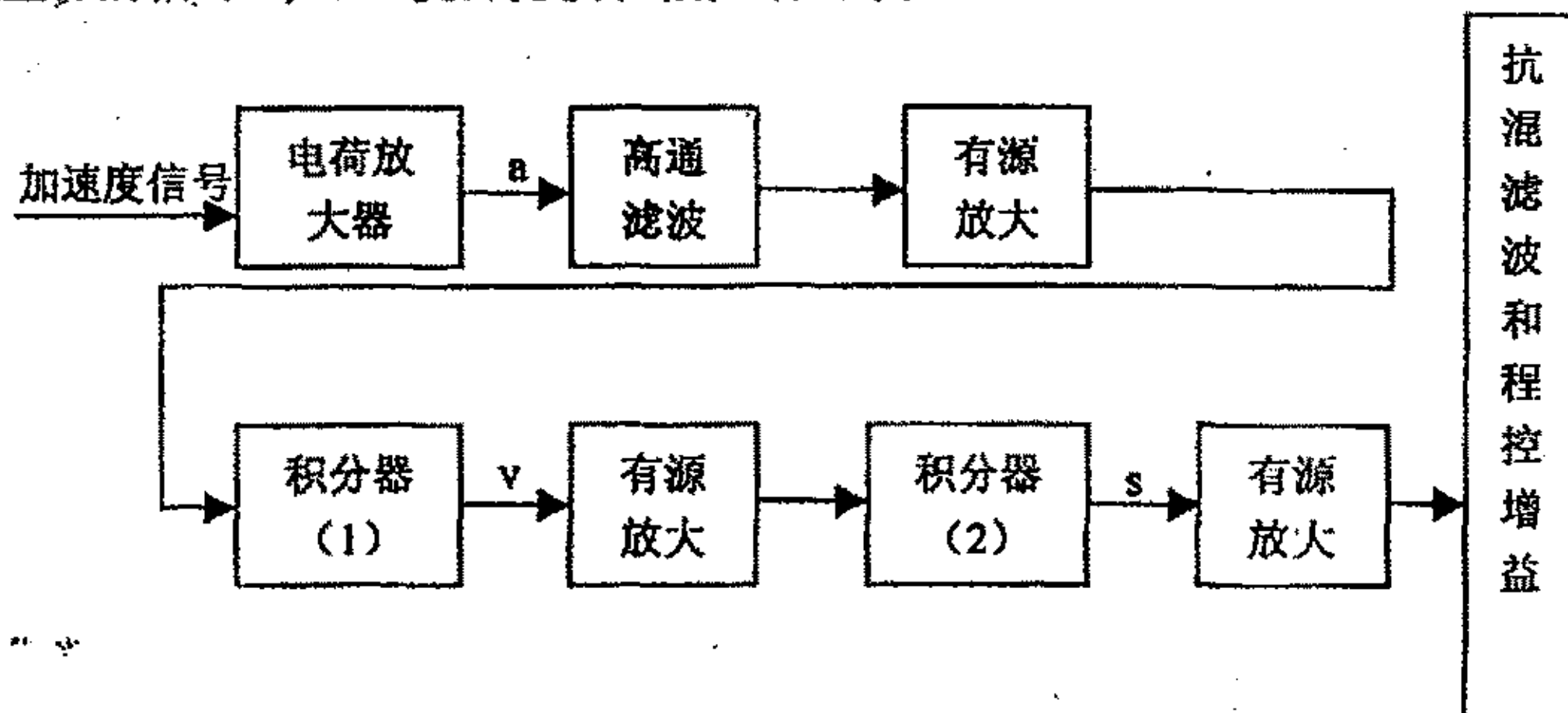


图 2.7 加速度信号调理原理

### 2.3.3.2 ICP 传感器信号调理

由 ICP 传感器的原理我们可以把 ICP 传感器等效为一个可变电阻。伴随着振动位移的变化，ICP 传感器的电阻值发生变化。因此，只要将这个电阻值的变化变换成为电压信号，就能够被采集系统进行提取作进一步的分析。如图 2.8 是其原理图。

我们看出，ICP 传感器的变换电路的核心是一个电流恒定的恒流源电路。这样输出电压才能跟随传感器电阻的变化。 $U=AI\Delta R$ ，所以，恒流源电路的设计便成为该系统的关键，恒流源的电流稳定性便成为影响变换精度的一个关键。为此，我们选用了 SGS-THOMSON 公司的 LM334 可调电流源作为恒流源的关键芯片。

LM334 是一个电流可调的三端元件。具有很宽的电流动态范围：80dB。电流大小由一个外接电阻阻值决定。在 25℃ 时， $I=67.7\text{mV}/R$ ，这里  $R$  是外接的电阻阻值。

在使用 LM334 的时候最重要的一点就是温度变化。半导体器件总会随着温度的变化而导致自身特性的改变。同样，虽然 LM334 内部电路把温度的影响减到最小，可还是会因为一些别的原因导致温度变化。主要归纳为：①芯片工作时候的内部热量②环境温度的改变。因此，在电路连接和布线的时候要考虑温度的影响，

以获得高精度的电流输出。

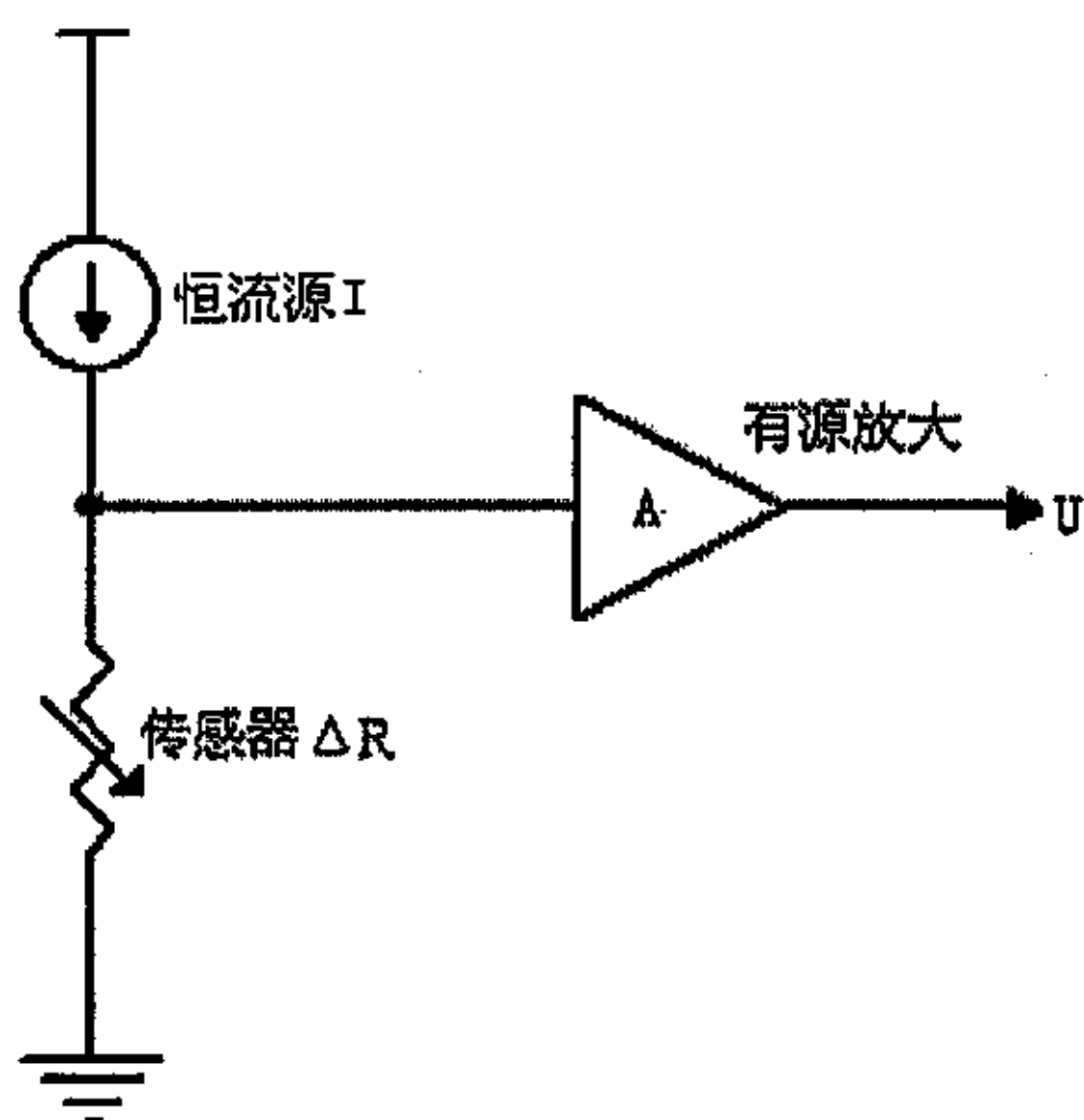


图 2.8 ICP 传感器信号调理原理

另外，如果消除温度影响的电路比较复杂或元件缺乏，也可以采用如下公式对数值进行修正： $I = I_0 (T/T_0)$ ，这里我们已经知道  $25^{\circ}\text{C}$  时候的  $I$  值，根据公式可以计算出  $I_0$  的值，然后代入公式可以计算出任意温度下的  $I$  值，注意  $T$  是开氏温标。

如下是我们采用的简单接法，这样必须在后续处理当中消除温度影响。消除方法如上。

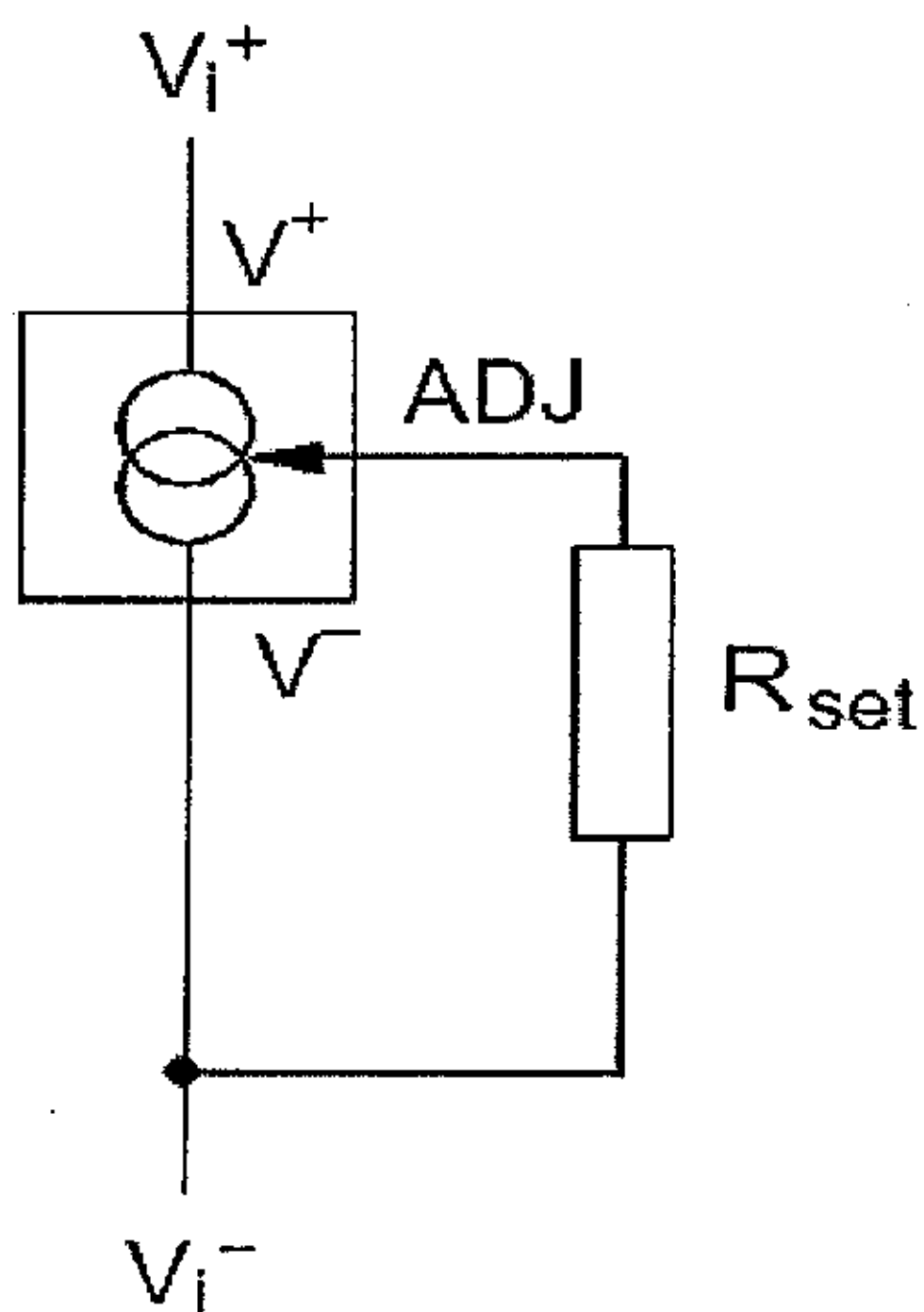


图 2.9 恒流源接口电路

### 2.3.4 TMS320LF2407 系统设计

本系统的设计我们采用了 TI 公司的 TMS320C24 系列的 TMS320LF2407A 定点数字信号处理芯片, 下面我们重点对 TMS320LF2407A 芯片进行介绍。

TMS320LF2407A 是 TMS320C2000 系列的一个产品, 主要用于工业控制和数据采集领域, 具有如下特点<sup>[11][12][13]</sup>:

- 采用高性能静态 CMOS 技术, 供电电压降到了 3.3V, 减小了控制器的功耗; 30MIPS 的指令执行速度使得指令周期只有 33ns(30MHz), 提高了 DSP 的实时控制能力;
- 基于 TMS320C2xx DSP 的 CPU 核, 保证了 TMS320LF2407 和 TMS320C2xx 的代码兼容性, 原来的算法代码可以重复使用;
- 片内具有高达 32K 字的 FLASH 程序存储器, 高达 1.5K 字的数据/程序 RAM, 544 字的双口 RAM(DARAM)和 2K 字的单口 RAM(SARAM);
- 两个事件管理器模块(EVA 和 EVB), 每个包括两个 16 位通用定时器、8 个 16 位的脉宽调制(PWM)通道、16 路的 A/D 转换;
- 可扩展的外部存储器总共有 192K 字空间, 分别为 64K 字的程序存储空间、64K 字的数据存储器空间、64K 字的 I/O 空间;
- 看门狗定时器模块(WDT);
- 10 位 A/D 转换器最小转换时间为 500ns, 可以选择两个事件管理器来触发两个 8 通道的输入或一个 16 通道的 A/D 转换器, 采样频率能够达到 166kHz;
- 控制器局域网 CAN2.0B 接口模块;
- 串行通信接口 SCI;
- 16 位串行外设接口模块(SPI);
- 基于锁相环的时钟发生器, 系统的工作时钟可编程;
- 40 个可单读编程或复用的通用输入/输出引脚(GPIO);
- 5 个外部中断;
- 电源管理模式使系统进入三种低功耗模式;

从以上特点我们可以看出, 采用 TMS320LF2407 非常适合作为数据采集和控制类应用的场合, 事实上 TMS320LF2407 自从推出以来就在数据采集控制等工业领域广泛应用。如下是 TMS320LF2407 的框图。



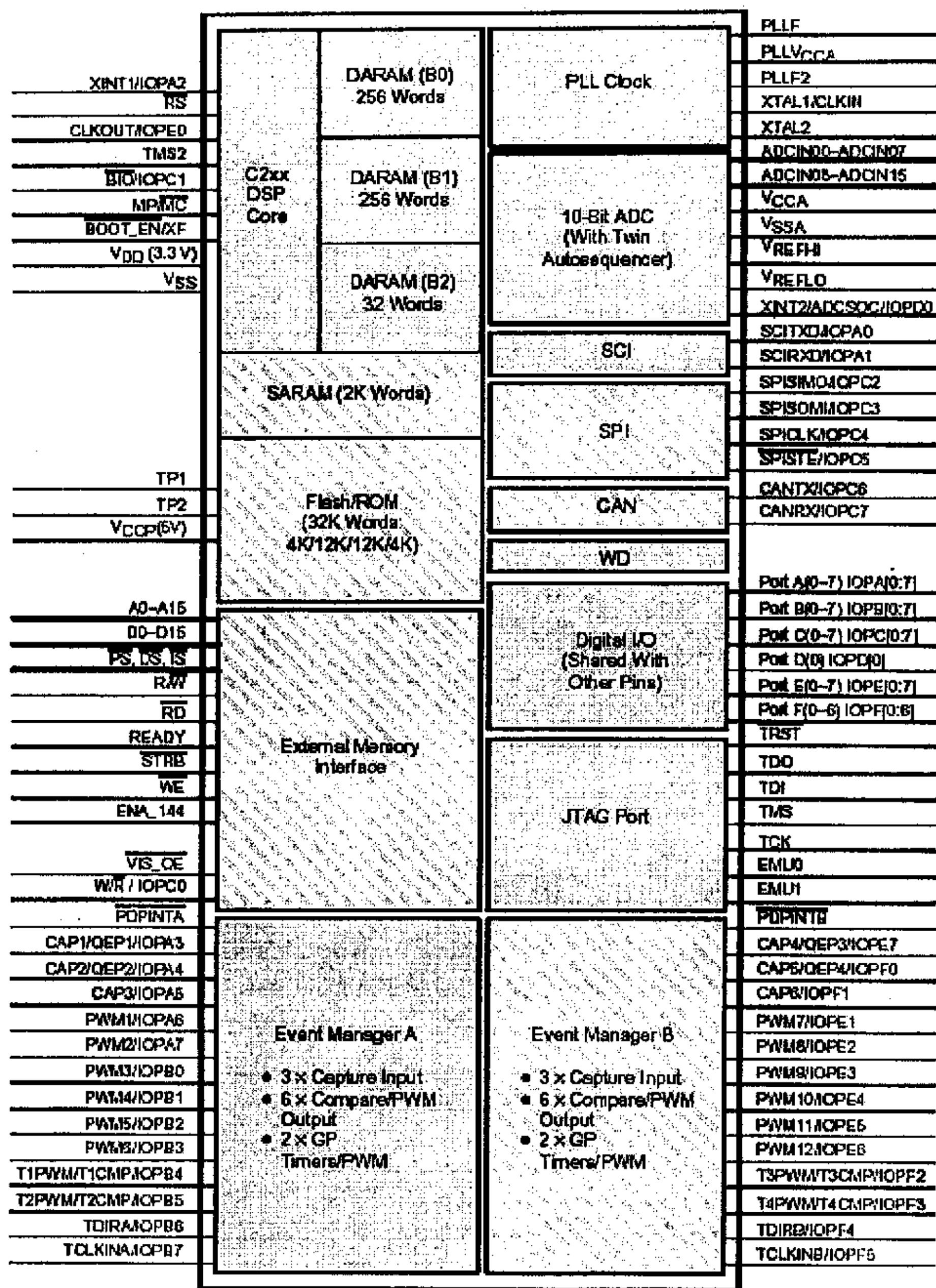


图 2.9 TMS320LF2407 结构图

### 2.3.4.1 存储器设计

存储器是 DSP 系统当中很重要的部件。从使用的功能角度来说,存储器可以分为 ROM 和 RAM。而根据每一种 ROM 和 RAM 的工艺条件来说,又有很多分法。对于 ROM 来说,随着半导体技术的飞速发展,目前可以使用的 ROM 有好几种,如一般的掩膜 ROM,紫外线擦写的 PROM,电可擦写的 EEPROM,可在线擦写多次的 FLASH ROM,还有目前正在逐步广泛使用的 NVROM 等。从 RAM 来说,也是随着半导体工艺技术的发展出现了很多易失性存储器。RAM 大体上分为静态 RAM(SRAM)、动态 RAM(DRAM)。SRAM 是基于触发器原理的,读写速度快,集成度低,成本高,功耗

大。由于 DSP 本身的速度很高, 因此 SRAM 在 DSP 系统中应用比较广泛。但是随着技术的进步, DRAM 成为 RAM 的发展核心。DRAM 读写速度较慢 (目前 PC 上的 DRAM 读写速度最快可以达到 5ns), 容量大, 成本低, 唯一的缺点就是控制复杂, DRAM 的电容特性需要一个附加逻辑对 DRAM 进行刷新。所以, 目前的 DRAM 发展途径就是把注意力集中在存储器芯片的片外附加逻辑上, 试图在片外组织连续的数据流来提高单位时间内的数据流量, 增加存储器带宽。

在这个系统当中, 我们采用了两片 ISSI 公司的 SRAM, 每一片存储容量为 64K 大小, 两片共同组合成为 16 位的存储器访问逻辑, 共 64K 字。下图是两片 RAM 的接口。

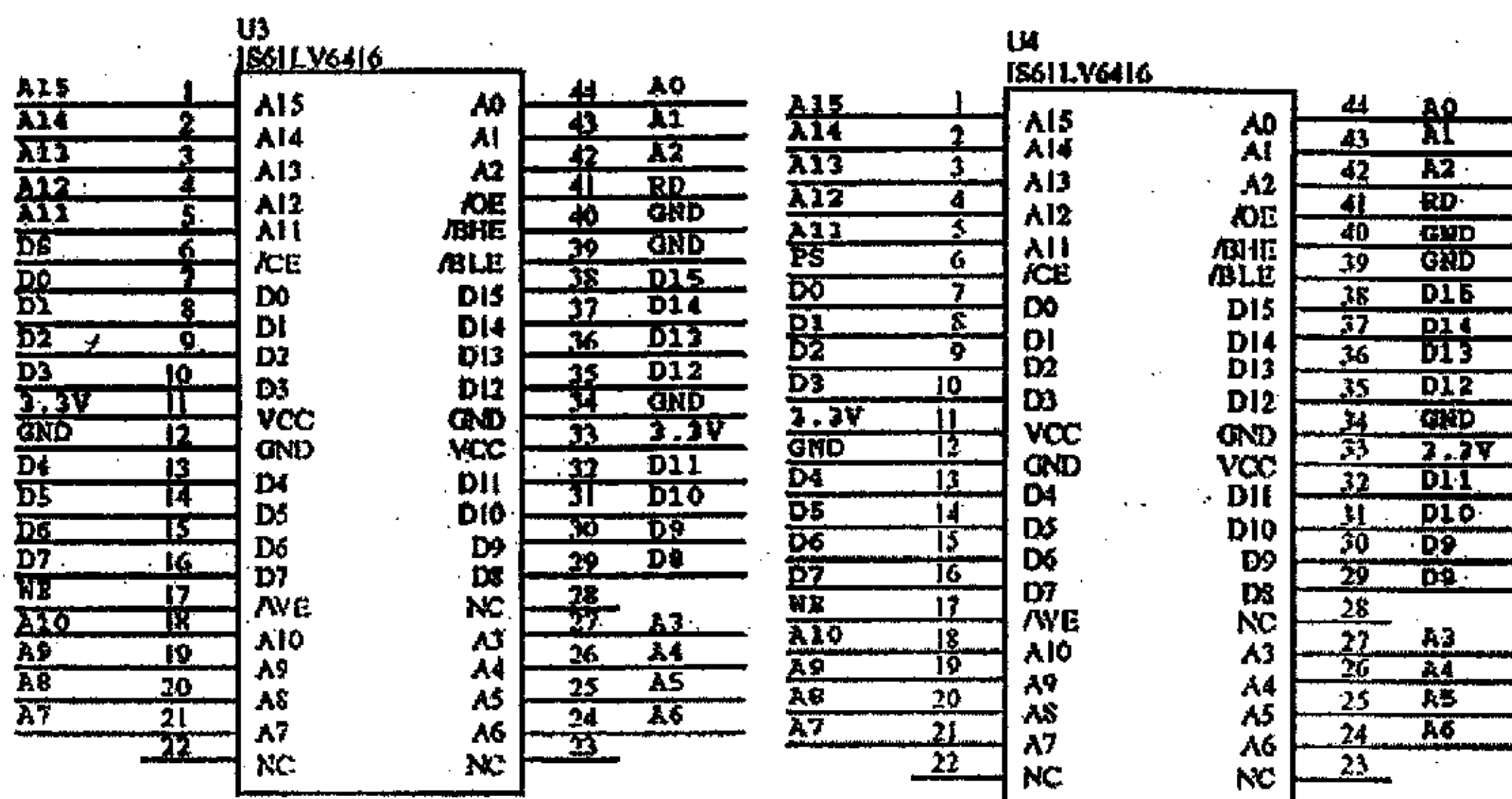


图 2.10 RAM 接口原理图

#### 2.3.4.2 双口 RAM 接口设计

为了提高数据的吞吐量, 现代采集系统一般都采用缓冲技术。缓冲技术的目的是利用一个中间存储部件, 发送数据者把批量数据放入这个中间存储部件当中, 数据接收者在一定条件下批量接收中间存储部件中的数据。在本系统中, 双口 RAM 就是作为数据缓冲作用的。存储在双口 RAM 当中的数据有两种, 依赖于主系统对数据的需求。如果主系统需要对数据进行傅利叶 (FFT) 变换, 则双口 RAM 中存储的数据是采集到的原始数据变换后的数据; 如果主系统仅需要原始数据, 则双口 RAM 中存储的数据就是原始数据。另外, 双口 RAM 中的前面几个字节还必须作命令和状态通讯端口。这样, 主 CPU 向这几个端口写如命令和配置数据, DSP 模块读取这部分命令和配置数据, 然后按照命令要求和配置数据的需要工作, 并把当前状态写入双口 RAM 当中, 主 CPU 就可以通过状态端口知道当前 DSP 的工



作状态。下图说明了双口 RAM 和主 CPU 的接口原理图。

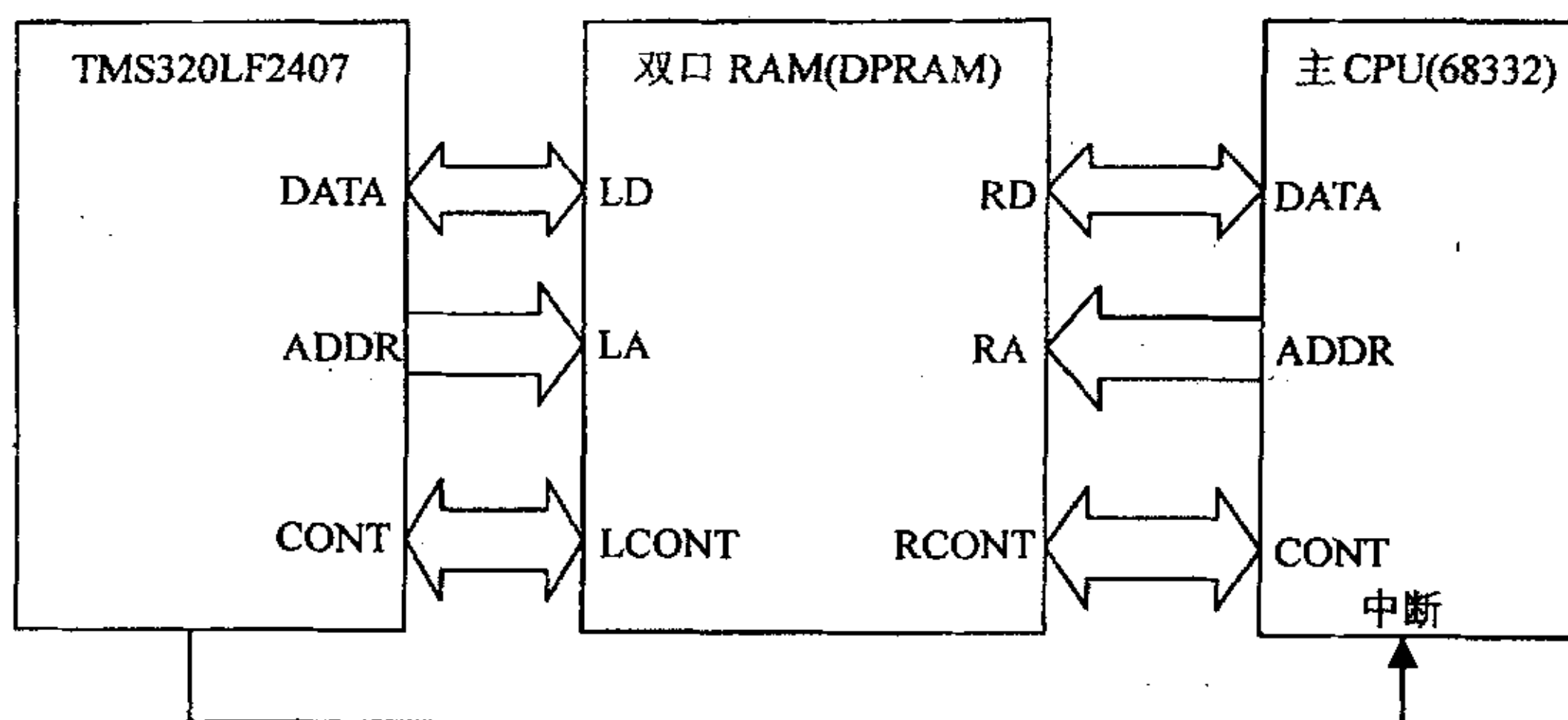


图 2.10 双口 RAM 和主 CPU 的接口原理图

DSP 将采集到或分析后的数据通过双口 RAM 的左边总线部分写入双口 RAM 中，当写到一定的大小之后（一般是双口 RAM 存储容量的一半），由 IO 发出中断给主 CPU，主 CPU 响应中断并向双口 RAM 中响应的端口写入数据表示响应中断，DSP 撤销中断。然后主 CPU 可以通过右边的总线读取数据。

## 第三章 接口板功能设计

### 3.1 网络接口模块的设计

为了能够和远程诊断中心的服务器相连接,本系统设计成为能够和远程诊断服务器连接,从而能够实现远程网络监测和故障诊断的一个前端子系统。

为了能够和 Internet 互连,需要支持 TCP/IP 协议。在这里,接口部分完成物理层和数据链路层的协议数据要求,而上层的网络(IP)层和传输层(TCP、UDP)还有应用层则必须通过软件实现。关于上层的软件实现部分是主控模块实现的内容,本节讨论物理层和数据链路层的设计<sup>[14][15][16]</sup>。

#### 3.1.1 RTL8019 (AS) 概述

物理层和数据链路层是通过 RTL8019 (AS) 以太网控制器芯片来完成的。它的主要性能如下:

- (1) 适应于 Ethernet II、IEEE802.3 、10Base5、10Base2、10BaseT;
- (2) 与 NE2000 兼容,支持 8 位、16 位数据总线;
- (3) 全双工,收发可同时达到 10Mbps 的速率,具有睡眠模式,以降低功耗;
- (4) 内置 16KB 的 SRAM,用于收发缓冲,降低对主处理器的速度要求;
- (5) 可连接同轴电缆和双绞线,并可自动检测所连接的介质;
- (6) 100 脚的 TQFP 封装,缩小 PCB 尺寸。

RTL8019AS 实现以太网媒介访问层(MAC)和物理层(PHY)的功能,包括 MAC 数据帧的组装/拆分与收发、地址识别、CRC 编码/校验、曼彻斯特编解码、接收噪声抑制、输出脉冲成形、超时重传、链路完整性测试、信号极性检测与纠正等。主处理器需要做的只是在 RTL8019AS 的外部总线上读写 MAC 帧。

按数据链路的不同,可以将 RTL8019AS 内部划分为远程 DMA(remote DMA)通道和本地 DMA(local DMA)通道两个部分。本地 DMA 完成控制器与网线的的数据交换;主处理器收发数据只需对远程 DMA 操作。当主处理器要向网上发送数据时,先将一帧数据通过远程 DMA 通道送到 RTL8019AS 中的发送缓存区,然后发出传送命令。RTL8019AS 在完成了上一帧的发送后,再进行此帧的发送。RTL8019AS 接收到的数据通过 MAC 比较、CRC 校验后,由 FIFO 存到接收缓冲区,收满一帧后,以中断或寄存器标志的方式通知主处理器,主处理器再通过远

程 DMA 通道读取这一帧数据。RTL8019AS 原理框图如图虚线框内部所示。

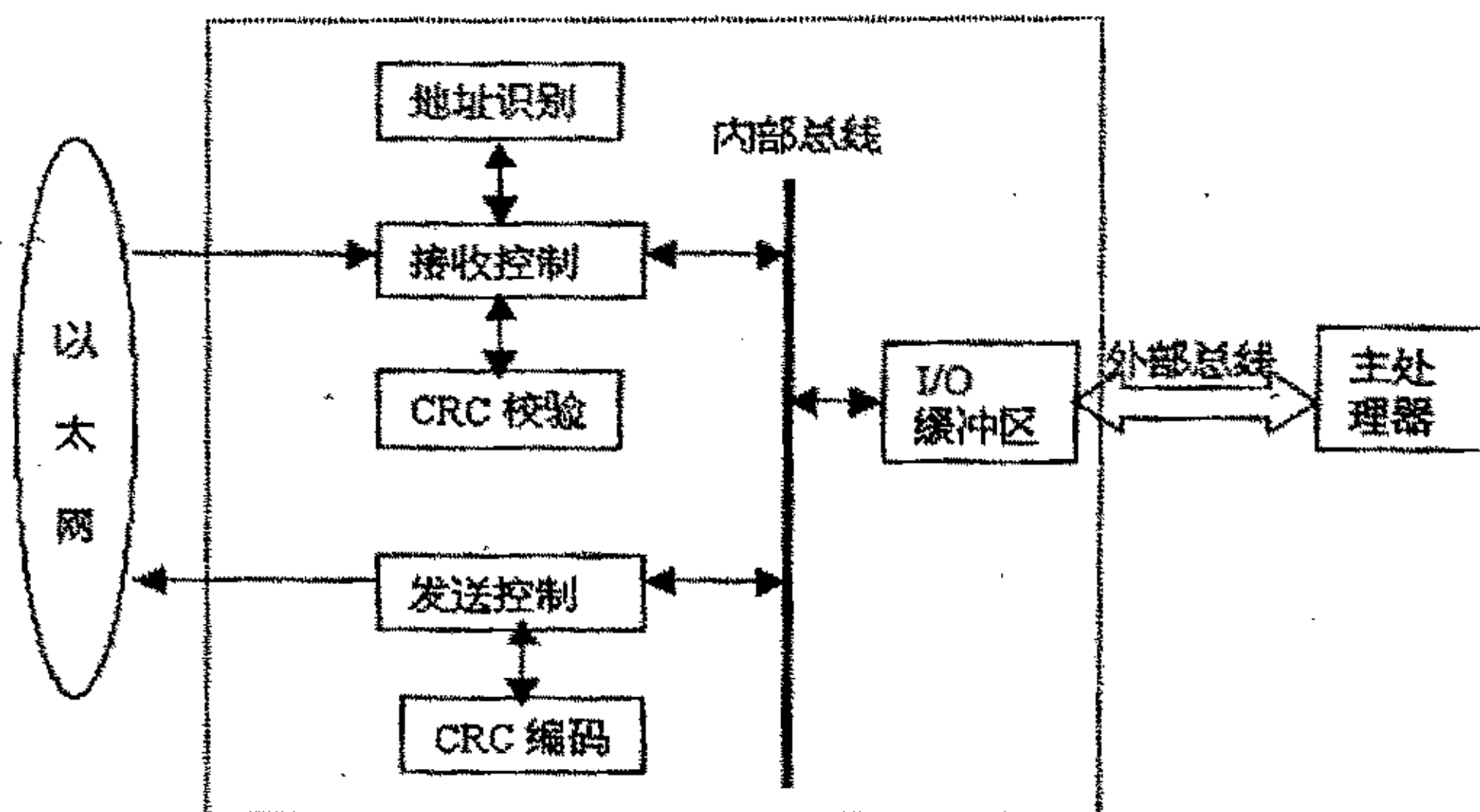


图 3.1 RTL8019AS 原理框图

### 3.1.2 接口电路设计

RTL8019 (AS) 的外部总线符合 ISA 标准，因此，如运用于嵌入式设备中，则在硬件和软件的设计上应有一些特殊性。嵌入式设备的主处理器可通过其映射到 16 个 I/O 地址上的寄存器来完成对 RTL8019AS 的操作。

RTL8019 的引脚信号如下表所示。为了满足系统扩充的需要，我们对接口电路中的信号线进行简化。

表 3.1 RTL8019(AS)的信号线定义

信号线	信号方向	引脚
数据信号线	I/O	SD0~15
地址信号线	I	SA0~19
地址允许线	I	AEN
I/O 通道准备好信号线	O	IOCHRDY
复位信号线	I	RESETDRV
系统存储器读信号	I	SMEMRB
系统存储器写信号	I	SMEMWB
I/O 读信号	I	IOR
I/O 写信号	I	IOW
数据总线宽度选择	I(复位时检测)	IOCS16
中断信号线	O	INT0~7

(1) RTL8019AS 需要 20 根地址线主要是为了对 Boot ROM 进行读写。Boot ROM 用于无盘工作站，它使得没有硬盘的电脑就可以从网卡进行引导，而不是从本地

硬盘等引导系统。由于我们不会用到 Boot ROM 功能,而 RTL8019AD 的 I/O 地址只有 32 个,所以与总线相连的地址线可减至 5 根;SMEMRB 和 SMEMWB 两根信号线也可以不用,直接接高电平。

(2)ISA 为了兼容的需要,设置了 IOCS16 信号线。由于我们使用的是 8 位总线,不进行 8/16 位数据读写的转换,故此信号接地。下表是 RTL8019 和 CPU 主板的连接关系表。

表 3.2 BUS 与 RTL8019AS 接口所需引脚

引脚	连接	
	RTL8019	BUS
外部总线	SA0~4	XA0~XA4
	SD7~0	XD7~XD0
读写信号	IOR/IOW	XWR/XRD
中断申请信号	INT0	XIRQ2

### 3.1.3 RTL8019 (AS) 芯片的编程和芯片初始化

RTL8019AS 是与 NE2000 兼容的以太网控制器。NE2000 是 NOVEL 公司生产的 16 位 ISA 总线以太网卡,它已成为 ISA 总线以太网控制器的标准。RTL8019 (AS) 的编程是通过 32 个 I/O 地址上的寄存器来完成的。其寄存器地址如表 3.3 所列。表中的地址为寄存器相对于 RTL8019AS 的 I/O 基地址的相对地址。

- 00H—0FH 为配置和状态寄存器,其中,"-"表示保留,用斜体标识的寄存器为 RTL8019AS 专用,其它寄存器为 NE2000 兼容寄存器。为了增强设计的通用性,我们应尽量避免使用专用寄存器。
- 10H—17H 共 8 个地址,为远程 DMA 地址,用于 RTL8019AS 与主机之间的数据交换。这 8 个地址都可以用来做 DMA 端口,只要用其中的一个就可以了。
- 18H—1FH 共 8 个地址,为复位端口,每个地址的功能都是一样的,向其中任何一个端口读或者写入任何数,都引起网卡的复位。由于有些以太网控制器不支持 19H,1BH,1DH 等奇数地址的复位,为了增强设计的通用性,我们只使用 18H,1AH,1CH,1EH 这几个复位端口中的一个就可以了。

#### 1、复位与初始化

RTL8019AS 可以通过引脚“冷”复位,也可以通过软件“热”复位。复位引脚为 RSTDRV(33脚),给这个引脚提供大于 800ns 的高电平信号即可启动复位过程。

复位的过程将执行一些操作,根据相关跳线引脚电平来决定 I/O 基地址及中断,检测以太网媒介类型,将内部寄存器初始化等。这些至少需要 2 毫秒的时间,所以推荐等待更久的时间之后才对 RTL8019AS 操作,比如 100 毫秒之后才对它操



表 3.3 RTL8019 (AS) 寄存器地址

No.(Hex)	Page0		Page1	Page2	Page3	
	[R]	[W]	[R/W]	[R]	[R]	[W]
00	CR	CR	CR	CR	CR	CR
01	CLDA0	PSTART	PAR0	PSTART	9346CR	9346CR
02	CLDA1	PSTOP	PAR1	PSTOP	BPAGE	BPAGE
03	BNRY	BNRY	PAR2	-	CONFIG0	-
04	TSR	TPSR	PAR3	TPSR	CONFIG1	CONFIG1
05	NCR	TBCR0	PAR4	-	CONFIG2	CONFIG2
06	FIFO	TBCR1	PAR5	-	CONFIG3	CONFIG3
07	ISR	ISR	CURR	-	-	TEST
08	CRDA0	RSAR0	MAR0	-	CSNSAV	-
09	CRDA1	RSAR1	MAR1	-	-	HLTCLK
0A	8019ID0	RBCR0	MAR2	-	-	-
0B	8019ID1	RBCR1	MAR3	-	INTR	-
0C	RSR	RCR	MAR4	RCR	-	FMWP
0D	CNTR0	TCR	MAR5	TCR	CONFIG4	-
0E	CNTR1	DCR	MAR6	DCR	-	-
0F	CNTR2	IMR	MAR7	IMR	-	-
10-17	Remote DMA Port (远程DMA端口)					
18-1F	Reset Port (复位端口)					

作, 以确保完全复位。

对复位端口(18H—1FH)的偶数地址的读, 或者写入任何数, 都引起网卡的复位。

中断状态寄存器ISR中有一个跟复位有关的标志位RST(第7位), 当RTL8019AS处于复位状态时被置位, 当CR寄存器中出现一个开始命令时被复位。可检查该位以确认复位正确。

## 2、初始化

完成复位之后, 还要对网卡的工作参数进行设置, 以使网卡开始工作。

先介绍一个选择寄存器页的子函数的伪代码如下

```
void PageSelect(unsigned char ucPageNumber)
{
    unsigned char ucTemp=reg00&0x3b; //reg00就是CR寄存器
    //PS1 PS0 RD2 RD1 RD0 TXP STA STP
    //0   0   1   1   1   0   1   1
    ucPageNumber=ucPageNumber<<6; //把页数移到最高两位
    reg00=ucTemp|ucPageNumber; //写入页位置
}
```

TXP清0是因为, 如果在发送数据包的过程中使用该函数, TXP会保持1直到数据包传送完成, 数据包还没发送完就重新写入1会导致重新向外发送数据包。而将ucTemp中的TXP对应位清0再写入reg00不会起任何作用, 所以不会导致正在传送的数据包重发。

下面介绍网卡的初始化子程序, 这里仅说明初始化的过程, 具体参数应根据

不同的应用而定。

```
void NE2000Init()
```

```
{
```

```
    //选择页0的寄存器, 网卡停止运行, 因为还没有初始化。
```

```
    reg00=0x21; //PS1 PS0 RD2 RD1 RD0 TXP STA STP
```

```
                //0 0 1 0 0 0 0 1
```

```
    reg01=0x4C; // PSTART(接收缓冲区起始地址)
```

```
    reg02=0x80; //PSTOP(接收缓冲区终止地址, 该页不用于接收)
```

```
    reg03=0x4C; //BNRY(从接收缓冲区读出数据的边界,读指针)
```

```
    reg04=0x40; //TPSR(传输数据包的页起始地址)
```

```
    //RCR(接收配置寄存器),只接收与本站地址匹配的正确包
```

```
    reg0c=0xC0; //- - MON PRO AM AB AR SEP
```

```
                //1 1 0 0 0 0 0 0
```

```
    //TCR(传输配置寄存器), 工作在正常模式, 需要CRC校验
```

```
    reg0d=0xE0; //- - - OFST ATD LB1 LB0 CRC
```

```
                //1 1 1 0 0 0 0 0
```

```
    //DCR(数据配置寄存器), FIFO阈值为8字节, 远程DMA由主机控制,
```

```
    //16位数据总线
```

```
    reg0e=0xC9; //- FT1 FT0 AR ML S LAS BOS WTS
```

```
                //1 1 0 0 1 0 0 1
```

```
    reg0f=0xFF; //IMR(中断屏蔽寄存器), 开放所有中断
```

```
    PageSelect(1); //选择页1的寄存器
```

```
    //以下写入网卡物理地址, 不同的卡应写入不同的地址
```

```
    reg01=00;
```

```
    reg02=0xE0;
```

```
    reg03=0x18;
```

```
    reg04=0x1C;
```

```
    reg05=0xC4;
```

```
    reg06=0x08;
```

```
    reg07=0x4C; // CURR(即将接收的数据包的存放页地址, 写指针)
```

```
    //初始化完成, 开始工作
```

```
    reg00=0x22; //PS1 PS0 RD2 RD1 RD0 TXP STA STP
```

```
                //0 0 1 0 0 0 1 0
```

```
}
```

### 3.2 键盘接口设计

在这个系统中，键盘接受用户的输入，是使用者与设备之间的唯一的交流渠道。在人机交互日益重要的今天，可靠的键盘设计是系统设计是否成功的重要因素。因此，在键盘设计上，我们需要考虑如下一些问题。

按键数量涉及到系统是否方便实用。多余的键盘对于现场工作人员来说是个累赘，而按键数量不足会导致操作的复杂性增加。在这个系统里面，参考了很多国外产品的设计思想，充分考虑到人机交互的简易性，如下是键盘的布局。

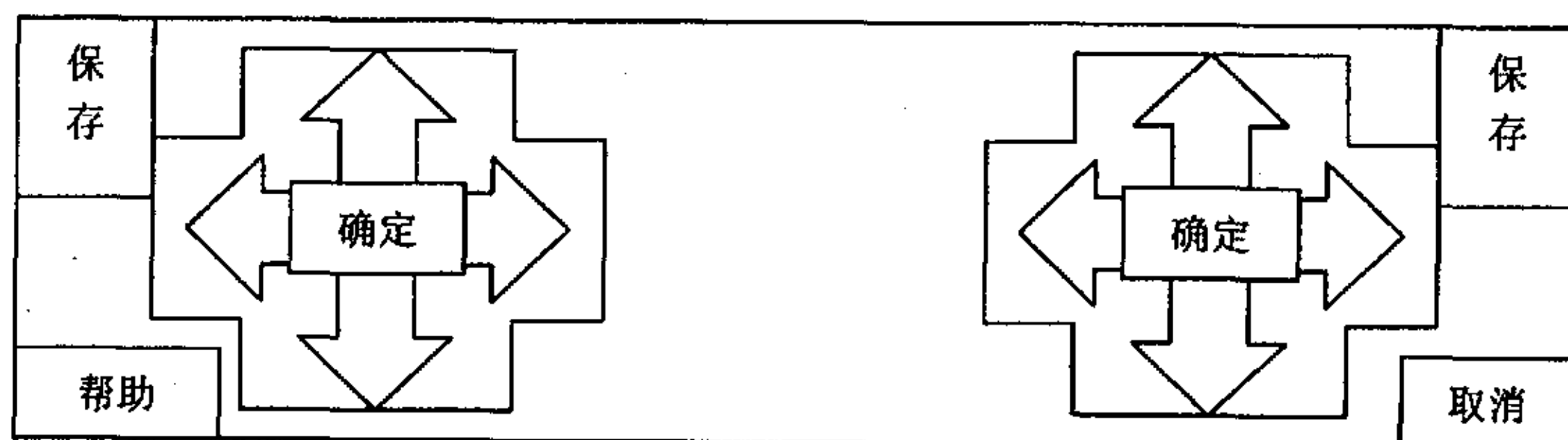


图 3.2 键盘布局结构图

共使用了 8 个按键。图 3.3 是键盘扫描电路的原理图。

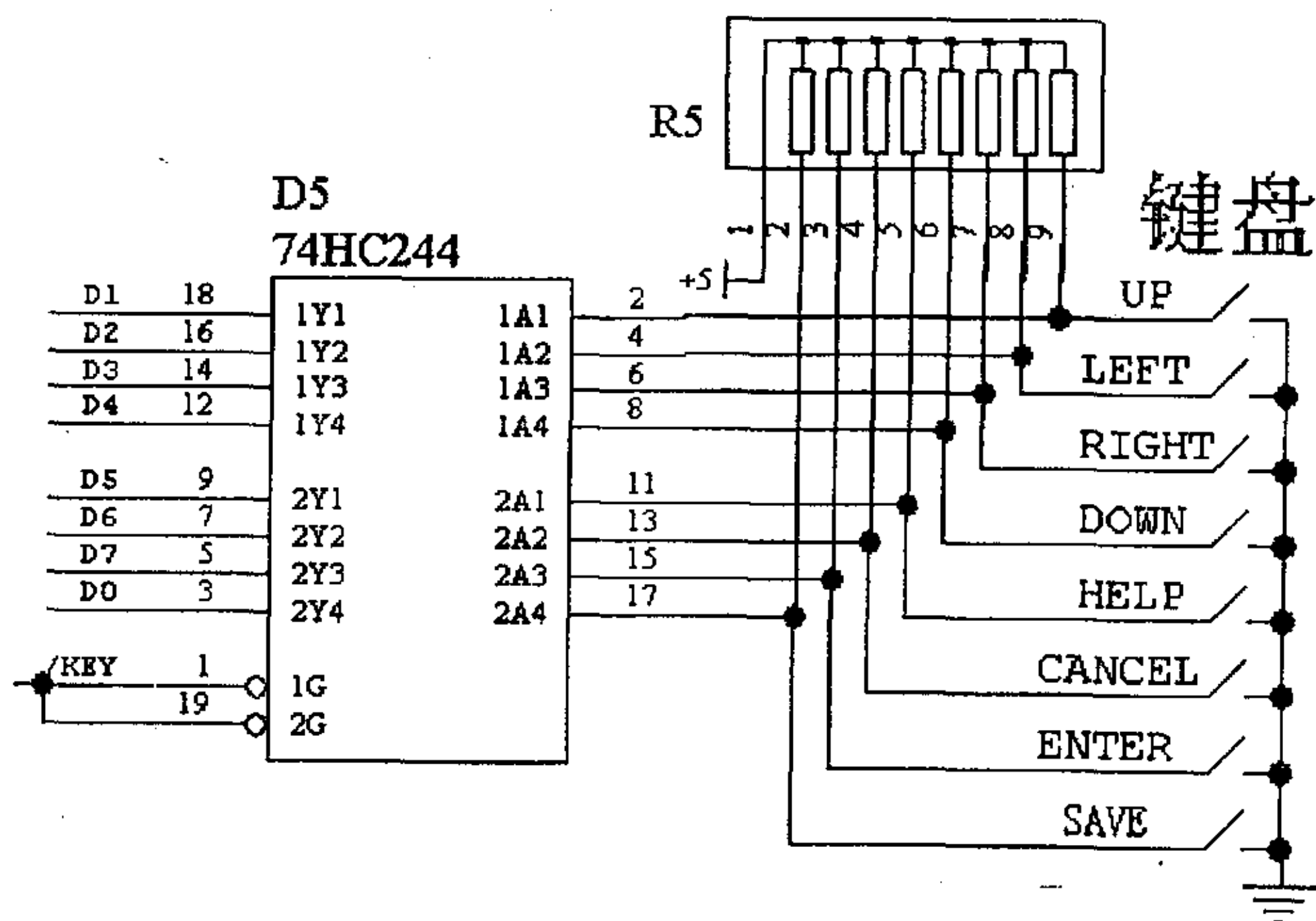


图 3.3 是键盘扫描电路的原理图

该键盘设计方案简单，但是需要做很多的额外处理。特别需要注意的就是键盘的去抖动。由于键盘数量不多，需要花的系统时间也很少。并且从系统运行效果来

说,能够满足系统设计要求。

### 3.3 液晶显示模块的设计

液晶显示是嵌入式系统最常用的一种输出方式,不但接口简单方便,还能满足低功耗设计的要求。在没有背光的情况下,液晶显示器能够低功耗的完成显示任务。因此,在这个系统里面,为了满足便携式功耗要求低的特点,我们选用了5英寸的DMF50081单色液晶显示屏,320×240点阵。采用EPSON公司的SED1335液晶显示控制器设计了液晶控制板。

#### 3.3.1 基于 SED1335 液晶显示控制器设计

SED1335液晶控制器是SED1330的增强型控制芯片。具有较强的IO缓冲功能,指令丰富,最大驱动能力640×256点阵,支持图形方式和文本方式的混合显示方式。SED1335具有很强的功能,具有如下几个特点:

- 有较强功能的I/O缓冲器;
- 指令功能丰富;
- 四位数据并行发送,最大驱动能力为640×256点阵;
- 图形和文本方式混合显示;
- 具有垂直、水平和滚动功能;
- 图形方式下具有三重屏幕显示;
- 内含有160个点阵字符的字符发生器;
- 与80及68系列MPU可直接相连;
- 低功耗:工作电流5mA,休眠电流0.05mA(VDD=5V)。
- SED1335只需要使用两个I/O端口和MPU进行数据交换,接口方便。

该芯片由MPU接口、内部控制器和液晶驱动部分组成。MPU接口部分和MC68332相连,在内部时序的控制下把CPU传来的指令和数据送入内部缓冲器当中,在内部控制器的译码下完成系统的功能。SED1335是在显示存储器的方式下工作的,CPU只要对相应的显示缓存写入相应的数据就能够显示出相应的点阵图形来,使用起来非常方便。下图是SED1335的接口电路。

左边部分是和处理器MC68332的接口,只要通过相应的地址空间的读写,就能够控制SED1335芯片的行为。右边是和显示存储器的接口,显示存储器的大小视显示的点阵和显示平面的多少而定。图3.4是和液晶显示模块的接口部分,能够直接驱动液晶显示器。



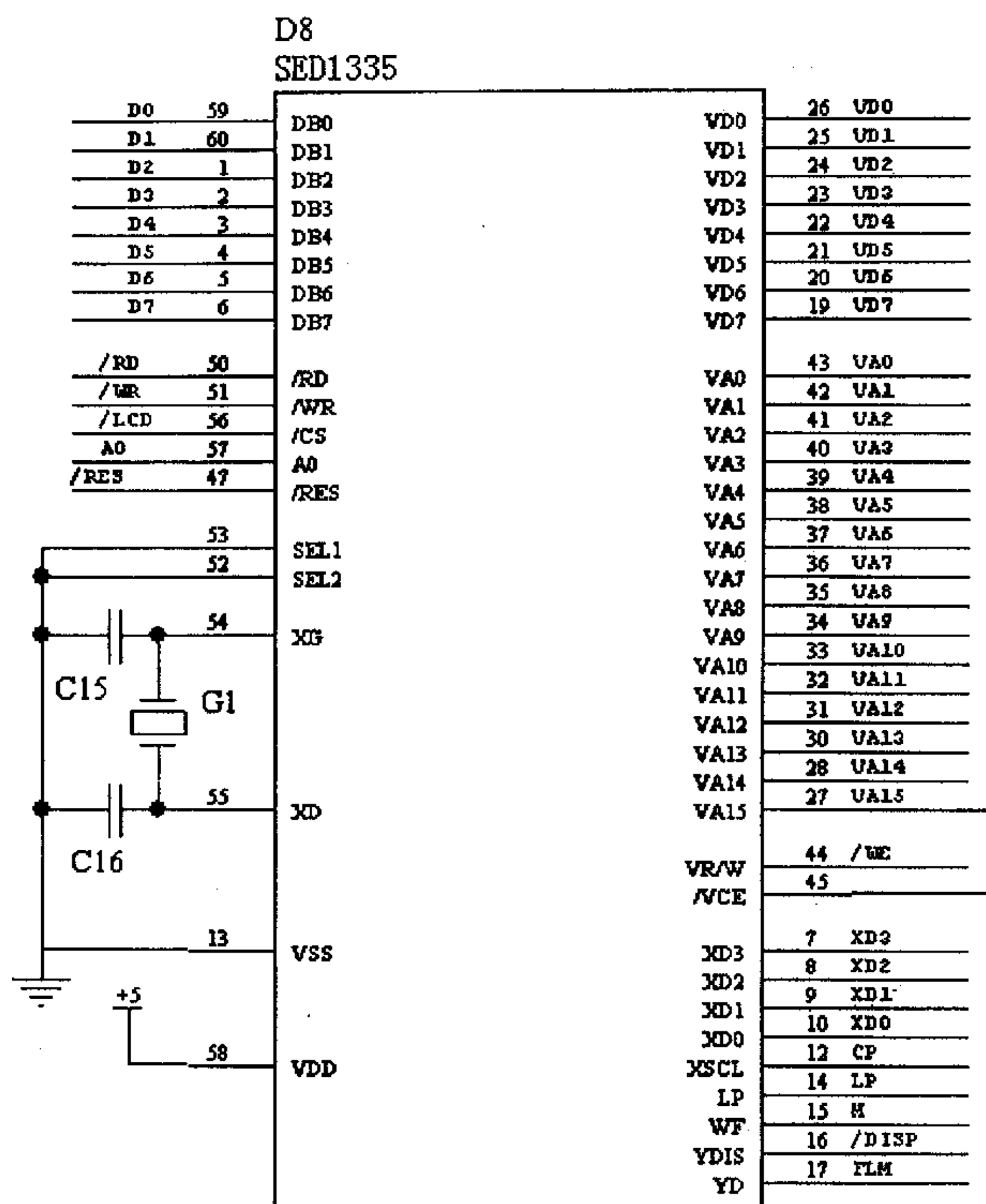


图 3.4 SED1335 和 MC68332 的接口电路

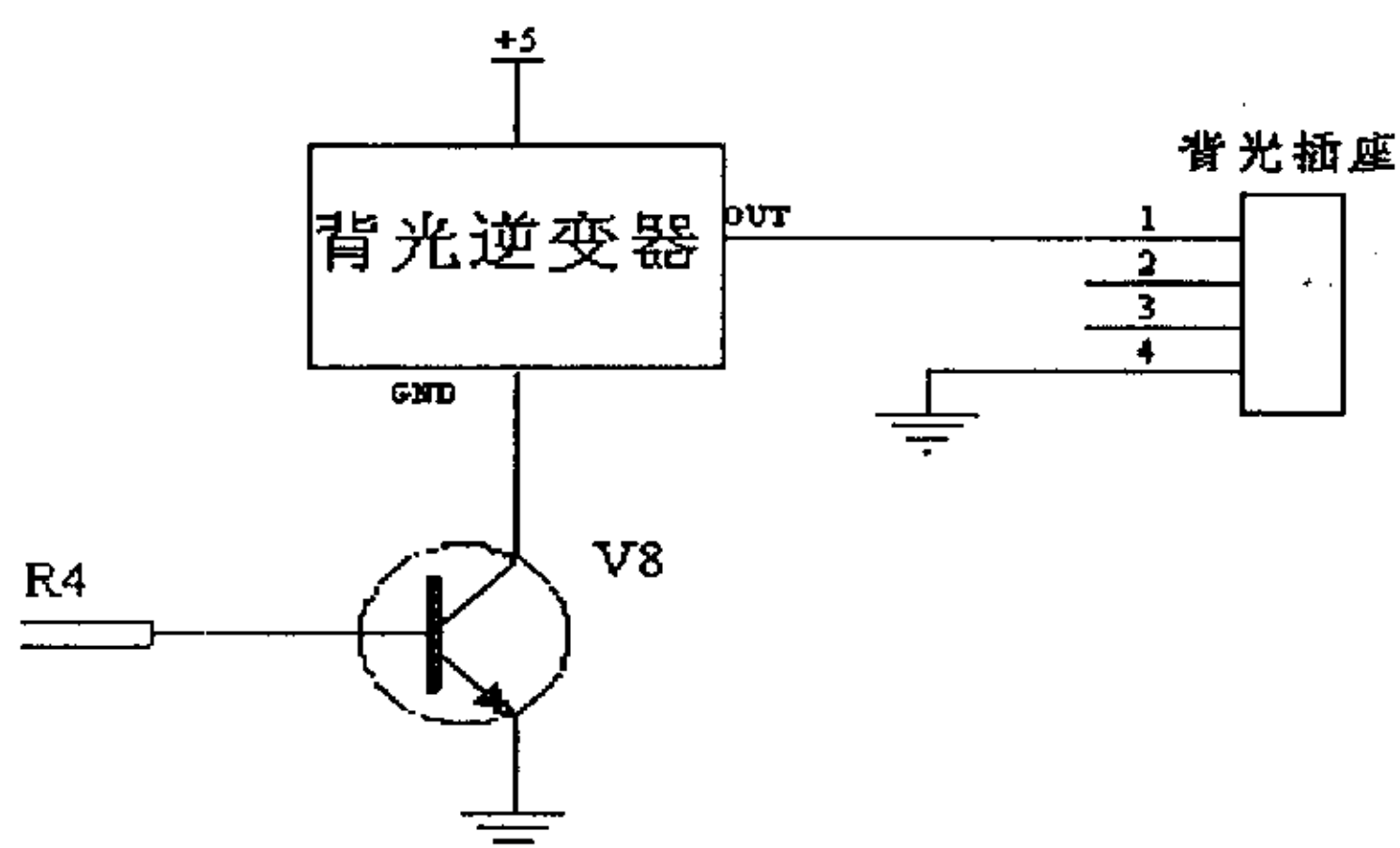


图 3.5 液晶背光逆变器的控制电路

另外，液晶显示器还有一些附件需要连接上去的，一个就是背光部分，为系统提供液晶显示的背光电源，背光控制如图 3.5 所示，通过一个三级管的开关来

控制背光的亮灭。另外一个就是字符对比度的调节，这是通过一个负电压来实现的。通过一个开关电源产生负压，经过可调电阻就能够实现对比度的调节功能。

## 第四章 主控模块的设计

主控模块是整个系统的控制核心。采用 Motorola 的 MC68332 为控制芯片，我们为系统设计了主控模块。该主控模块在 MC68332 的控制下，实现整个系统的功能整合，并协调各模块的通信与协作。

在主控模块的基础上，我们采用了  $\mu$ COS-II 实时操作系统实现整个系统的控制，并在操作系统的基础上实现键盘扫描、采集数据读取、参数设置、图形显示、网络协议的实现等功能。

### 4.1 主控制板的设计

MC68332 芯片是 32 位的微控制器，是 MOTOROLA 公司在 1989 年推出的 32 位单片机，它具有较高的执行速度、较完善的系统保护功能、较强的 I/O 功能和数据处理能力。MC68332 是较早进入应用的 32 位单片机，现在已经被应用于许多高技术领域，如首先被应用到汽车仪表和自动控制、数据采集和控制系统、现代测试分析设备、计算机和外部设备控制器、高级通讯设备，以及航空、航天、机器人等领域，并且其应用范围还在不断的扩大。在许多方面，MC68332 代表了今后单片机发展的一个方向。

#### 4.1.1 MC68332 处理器概述

##### 4.1.1.1 MC68332 的主要特性

MC68332 的主要特性如下<sup>[17][18][19]</sup>：

- ◆ 低功耗：最大功耗约 600mW，后备方式仅 500 $\mu$ W 左右。
- ◆ 工作频率：5V 电源时为 16.78MHz，内部锁相环时钟可软件编程。
- ◆ 集成度：422 000 个晶体管。
- ◆ 封装：132 个引脚，塑料方扁平组件（PQFP）。
- ◆ CPU：32 位，与 68000 系列软件全兼容。
- ◆ 智能型的 16 位定时处理器（TPU）。
- ◆ 两个串行 I/O 子系统（QSPI 和 SCI 组成 QSM 模块）。
- ◆ 片上 RAM：2K 字节的静态高速 SRAM。
- ◆ 可编程片选逻辑：12 条可编程片选逻辑，省去了多级译码的麻烦，自由选择外部存储空间和 IO 空间。

- 多达 31 个 I/O 引脚。
- 系统出错保护功能：
  - 类 M68HC11 看门狗定时器；
  - 类 MCHC11 周期中断定时器；
  - 类 M68000 伪中断、HALT 监视器和总线定时输出监视器；

#### 4.1.1.2 MC68332 的组成模块

MC68332 采用模块化设计的芯片，几个不同功能的模块通过内部总线组成了片上系统（System On Chip）<sup>[17]</sup>。这些模块主要有：中央处理单元 CPU32，系统集成模块 SIM，队列串行模块 QSM，片上静态存储器 RAM，定时处理器 TPU 等。

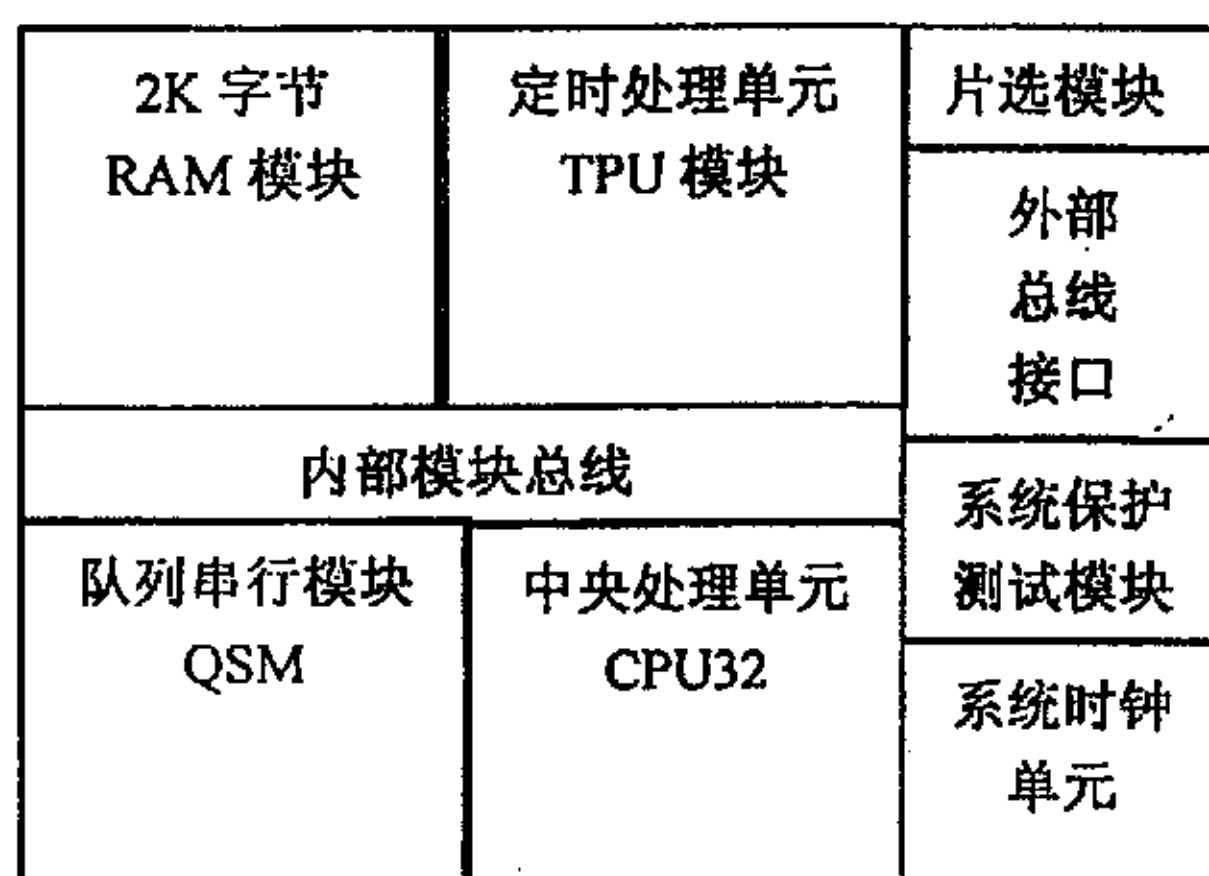


图 4.1 MC68332 的组成结构图

CPU32：32 位的处理能力，与 TPU 并行工作，可以使系统的处理能力大大提高。采用高速互补金属氧化物技术，保证 CPU32 在正常工作时具有较低的功耗，并且具有专用的低功耗执行指令，保证在不丢失数据信息的情况下进入休眠状态。CPU32 是芯片的控制核心，采用冯诺伊曼结构的三级流水线，可在指令执行前进行预取和译码，使 CPU 能够并发工作，在执行一串指令时，总的执行时间减少，满足数控系统高速大计算量的性能要求。

CPU32 的特点总结如下：

- 1) 与 M68K 系列 CPU 基本指令系统兼容；
- 2) 32 位的内部数据地址总线；
- 3) 外部 16 位数据总线和 24 位的地址总线；
- 4) 8 个通用数据寄存器，7 个通用地址寄存器，32 位的程序计数器，两个分开的 32 位堆栈指针（分别用于用户级和管理级）和若干专用控制寄存器；



- 5) 具有动态数据总线端口的高速总线接口 (EBI);
- 6) 支持虚拟存储;
- 7) 改进的异常处理功能 (这两项是现代操作系统的要求);
- 8) 丰富的指令系统: (适合于要求高速运算, 如 DSP 的场合)
  - ◆ 高速、高精度乘除和移位指令
  - ◆ 条件码陷阱 (用于条件判断并跳转)
  - ◆ 增强的断点指令
  - ◆ 循环方式执行指令
  - ◆ 查表和插值指令
  - ◆ 低功率 STOP 指令
- 9) 跟踪程序流变化;
- 10) 硬件断点信号 (支持 BDM 背景调试模式);
- 11) 16.78MHz 工作频率;
- 12) 全静态设计;

系统集成模块 SIM: 系统集成模块是 MOTOROLA 芯片的一个比较有特点的模块。该模块确定复位后或初始化期间 MC68332 的状态。它主要由几个控制系统操作的子模块组成, 控制着系统的初始化、软硬件配置和外部总线片选逻辑的产生等, 如图 4.2 所示。

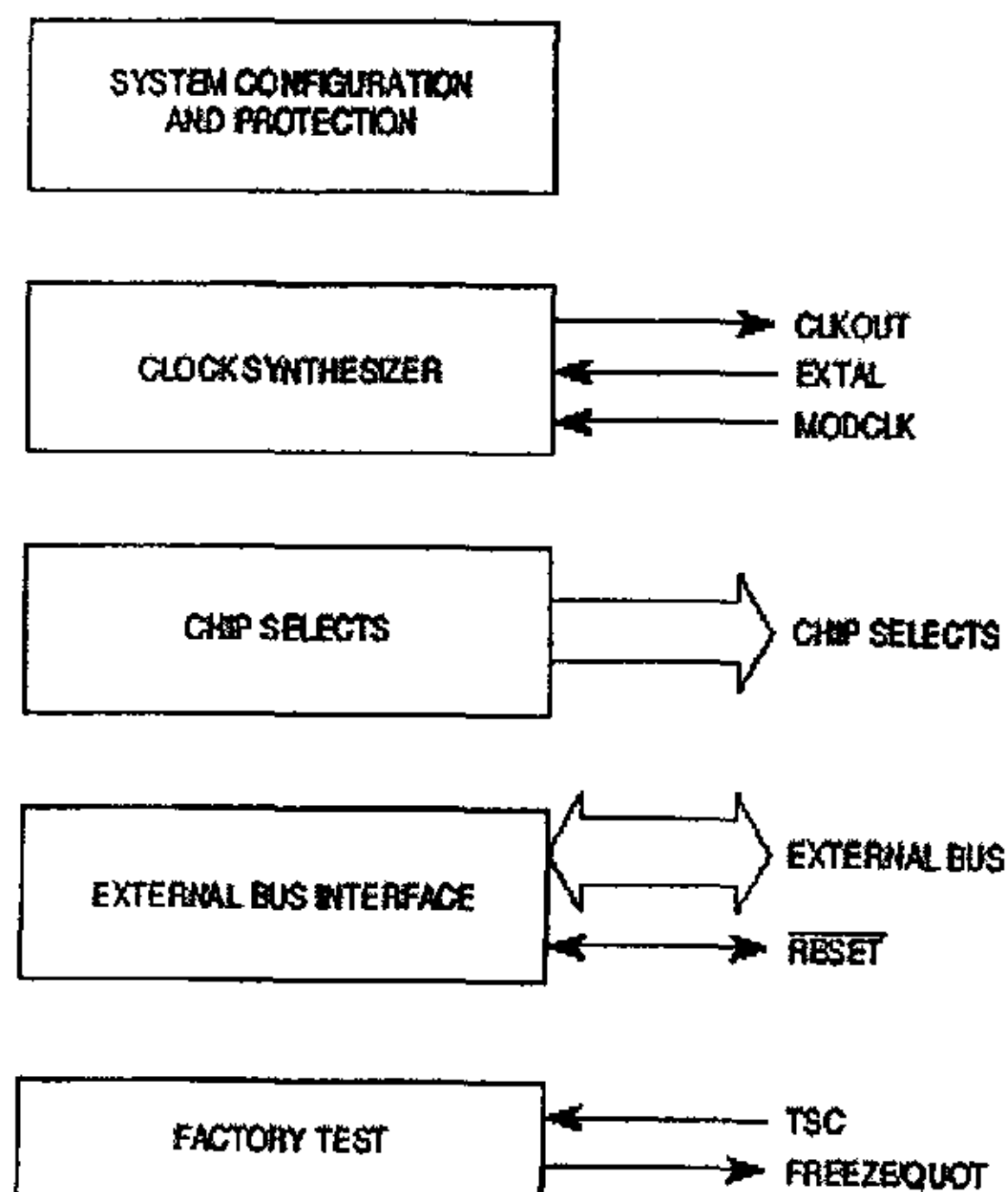


图 4.2 SIM 的组成结构图

- 1) 系统配置和保护子模块: 控制系统配置并提供总线监视器和软件看门狗以实

- 现系统保护，此外，它提供一个周期中断，满足周期性事件的控制要求；
- 2) 时钟合成器：锁相产生整个系统运行所需要的时钟；
  - 3) 外部总线接口：处理信息在内部模块和外部地址空间之间的传输，并复用 GPIO（通用 IO 子模块）功能；
  - 4) 片选逻辑：提供 12 个片选逻辑，可编程实现外部地址空间的片选，简化系统设计功能；
  - 5) 系统测试：包含测试芯片所有模块的功能，仅用于工厂测试之用；

串行队列输出模块（QSM）：串行队列输出模块主要完成系统与外部的串行通信接口。串行通信方式分为同步传输和异步传输方式。QSM 的 SCI 部分完成和外界的异步串行通讯的接口功能，QSM 的 QSPI 完成与外部世界的同步串行通讯的接口功能，如图 4.3 所示：

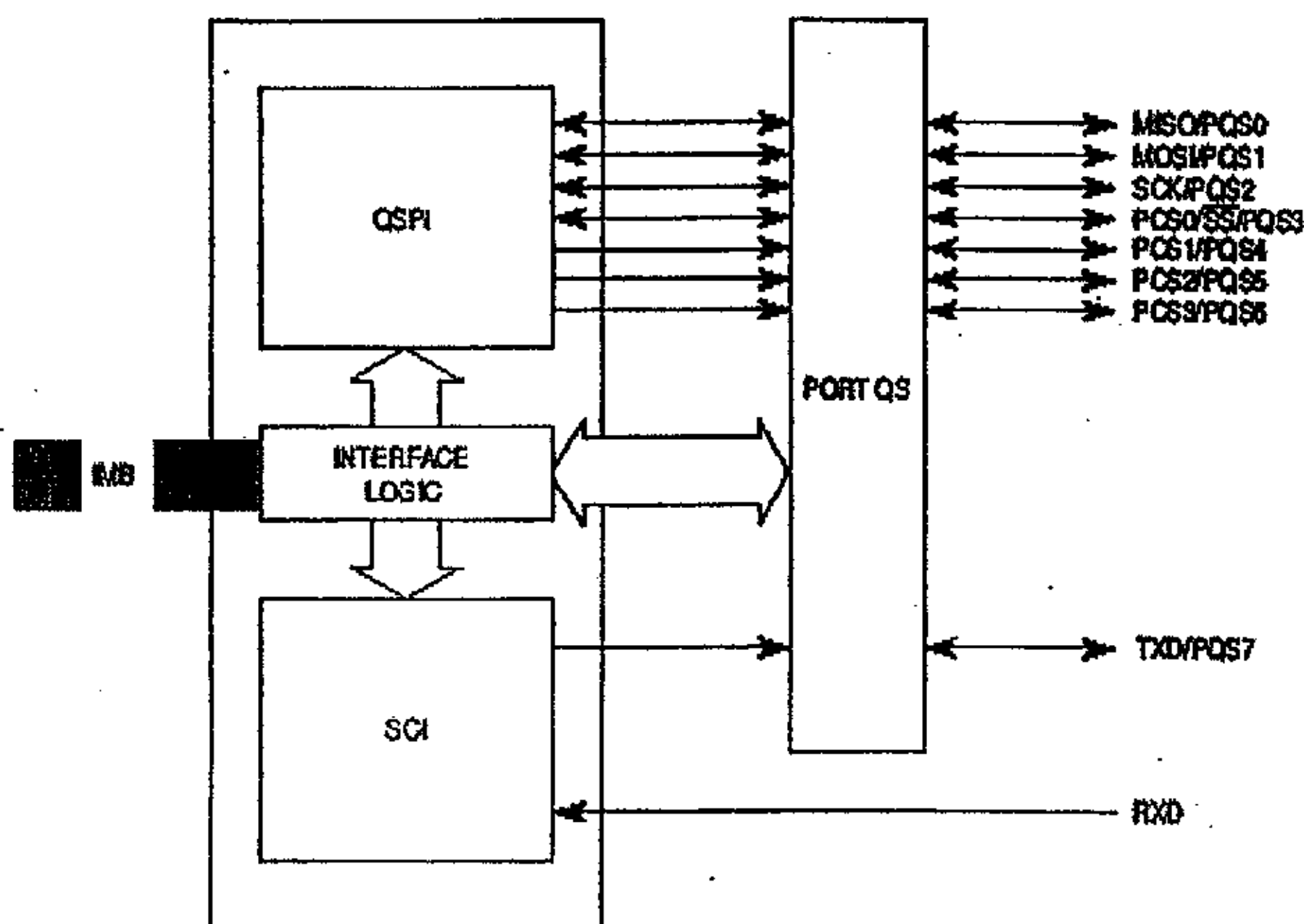


图 4.3 QSM 的组成结构图

两个子模块能够独立的完成传输操作。SCI 是全双工通用异步接受发送器（UART）串行接口。带有先进的错误监测线路，能够捕捉只有 1/16 位传输时间的噪声信号，通讯数据位可编程，能够实现 64 到 524K 波特率的通信速率，并且具有唤醒功能允许 CPU 不间断运行。QSPI 是外部设备和其他串行设备进行同步通讯的全双工串行串口，提供方便的外部扩展能力和全双工、同步的数据输入、数据输出、串行时钟等通信功能。QSPI 的 4 颗可编程外部片选线使其最多可以寻址 16 个外部设备。通过附加队列，使 QSPI 可以在没有 CPU 干预的条件下独立完成最多 16 个串行传输或 256 位的数据流传输。

定时处理器模块 (TPU): TPU 模块是整个控制系统的核心, 完成电机的所有控制功能。TPU 是一个智能型的微控制器, 在系统中专门处理定时任务。不需要 CPU 干预, 能够独立的执行与时间有关的操作。它可完成从简单到复杂的定时处理任务, 是目前最先进的定时器系统。输入捕捉和输出比较功能是 TPU 的基本功能。TPU 模块一共有 16 个通道, 每个通道对应一个 I/O 端口, 能够独立的执行任何一种定时功能。

### 4.1.2 电路模块设计

#### 4.1.2.1 系统时钟电路设计

MC68332 采用外部晶振经过内部锁相电路 (锁相频率可编程) 来产生系统定时所需要的基准时钟。在本系统的设计当中, 采用了外部无源晶振, 经过时钟合成器得到系统的工作时钟。G1 是 32.768KHz 的无源晶振, C41、C42 作滤波。

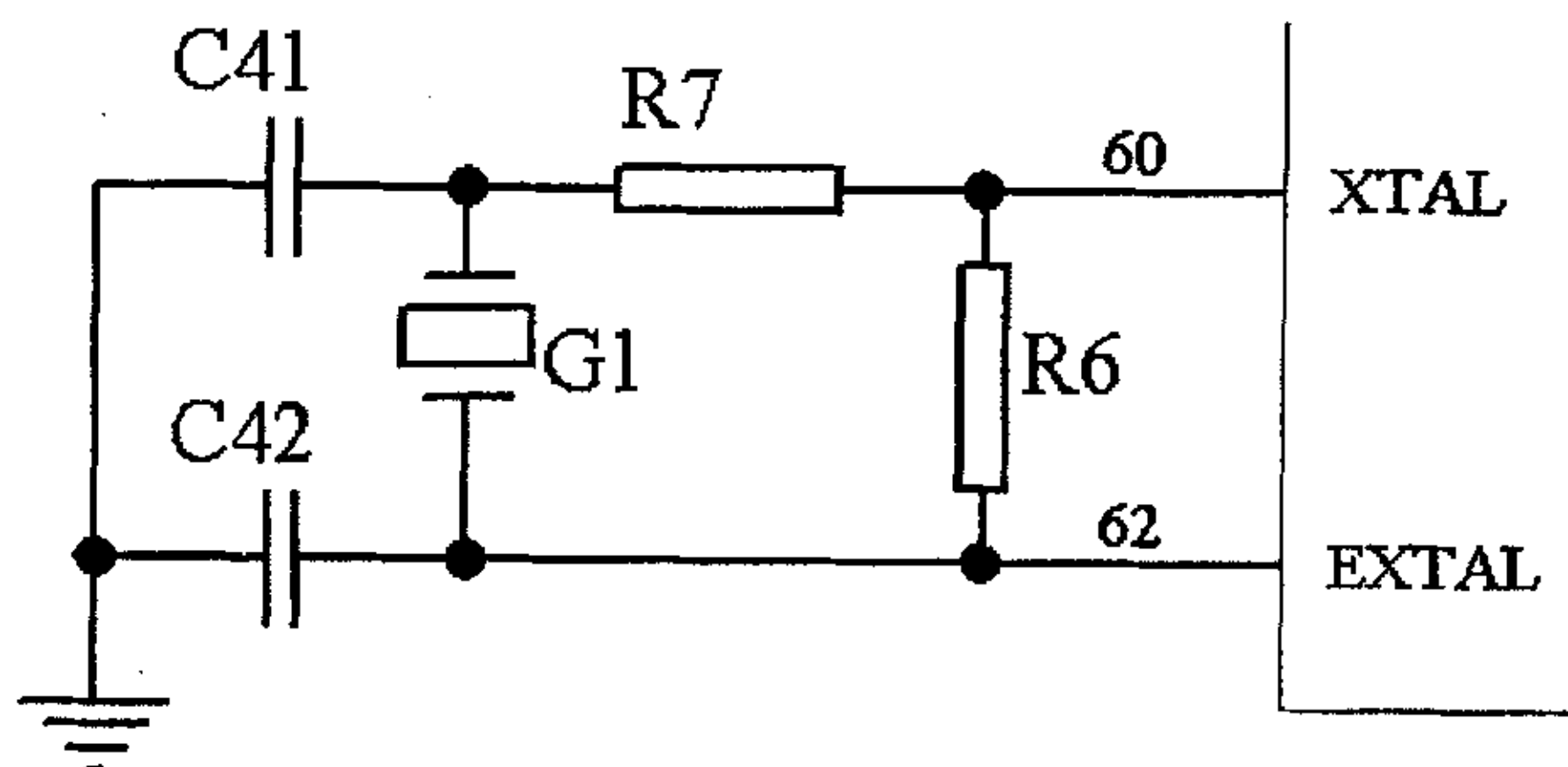


图 4.4 时钟电路原理图

#### 4.1.2.2 系统复位与低压保护电路的设计

MC68332 系统本身提供看门狗模块, 系统需要一个定时器, 所以该模块另作它用。系统需要稳定可靠的工作, 需要外接一个能够让系统自动复位的电路, 提供在系统运行进入不正常的情况下让系统快速复位, 重新进入正常状况。这个功能一般是采用看门狗电路来实现的。如图 4.5 所示, 采用了 MAXIM 公司的 MAX818 复位芯片, 加上外部电池切换, 能够提供电源保护和电源切换的功能, 非常方便。

系统工作原理如下: MC68332 必须定时的 (每隔 1 秒种) 给 WDI 管脚输入定时脉冲, 否则 MAX818 就会产生 RST 信号让系统整体复位。这样的话, 一旦

程序运行出现故障或“跑飞”，WDI 定时脉冲输入就得不到满足，系统复位，让系统重新进入正常工作。另外，系统提供电源切换功能。当 VCC 端出现电压过低的情况，系统将电源切换到 BATT 端，由 G3 电池供电。这样的话，系统的日历

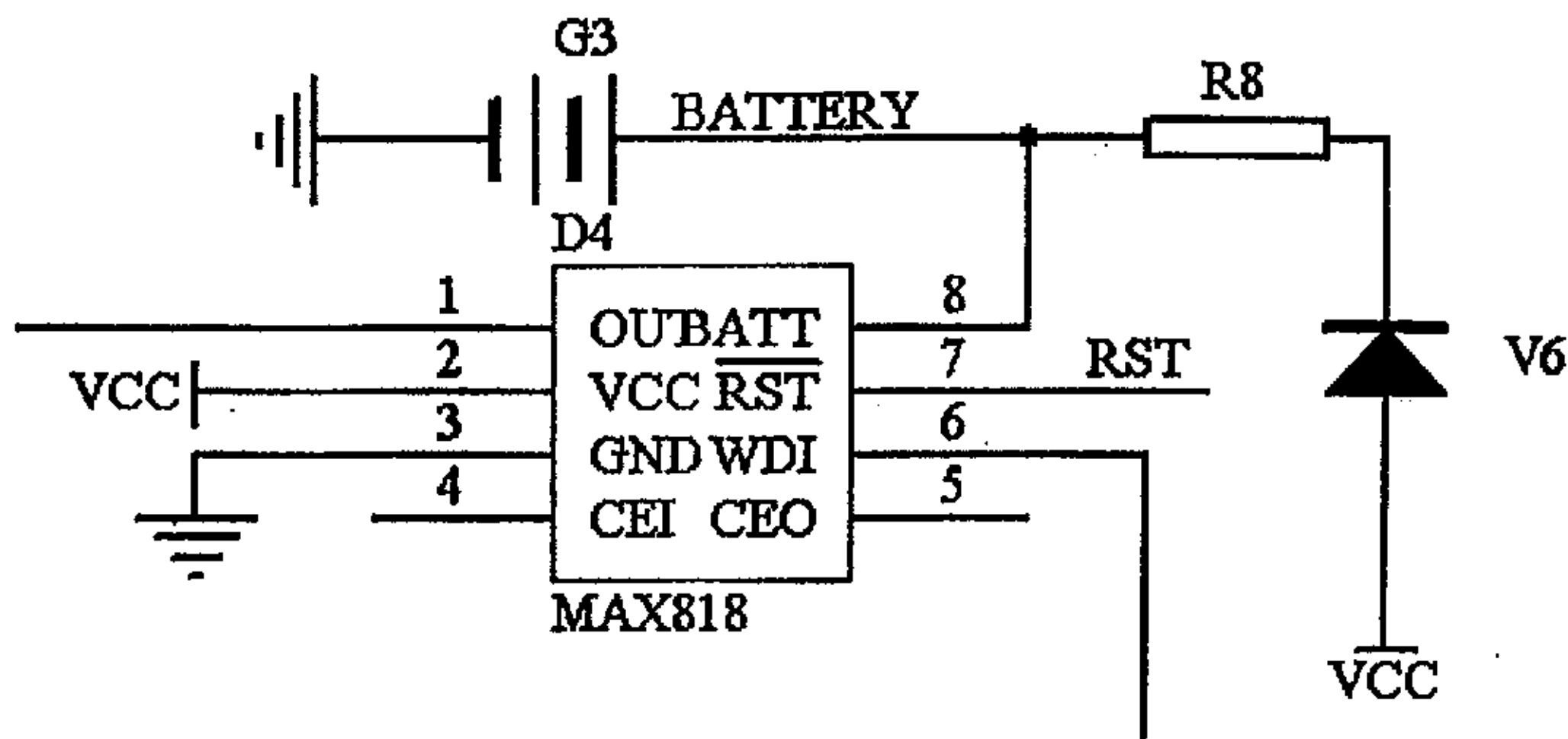


图 4.5 掉点保护与看门狗电路原理图

芯片就能够在系统掉电以后，一样能够正常工作，保证了时间计数的延续性。另外，V6 二极管和 R8 组成了 G3 的充电电路，系统能够在工作的时候充电，不断的让系统正常工作下去。

#### 4.1.2.3 存储器扩展电路

存储器系统是一个嵌入式系统设计的重要部分。在这个系统里面，一方面系统功能较强，如需要具有管理功能、各种运算、控制，还需要系统支持网络协议的要求。在 RTOS 的基础上进行开发，还需要为 RTOS 留出不小的存储空间（看系统大小而定，估计需要 20K 的存储容量）。另外，采集数据必须能够保存，能够为用户重新使用。这样就要求系统的存储系统具有可在线擦写功能，调电之后数据能够存储下来。在此基础上，系统的存储部件采用了 SRAM 和 FLASH 结合的办法，两部分各为 512K，满足了大容量的存储容量的要求。

MC68332 的动态数据宽度的特性，要求 8 位外设的数据总线接到 MC68332 的 D7-15 上面去，这是处理器本身的要求。FLASH 选用了 ATMEL 公司的 AT29C040A 芯片，512K 的存储容量，可在线擦写。SRAM 选用了 TOSHIBA 公司的 TC554161 存储芯片，存储容量也是 512K，16 位宽。

AT29C040A FLASH 和 TC554161 SRAM 和处理器的接口电路分别参考图 4.6 和图 4.7。



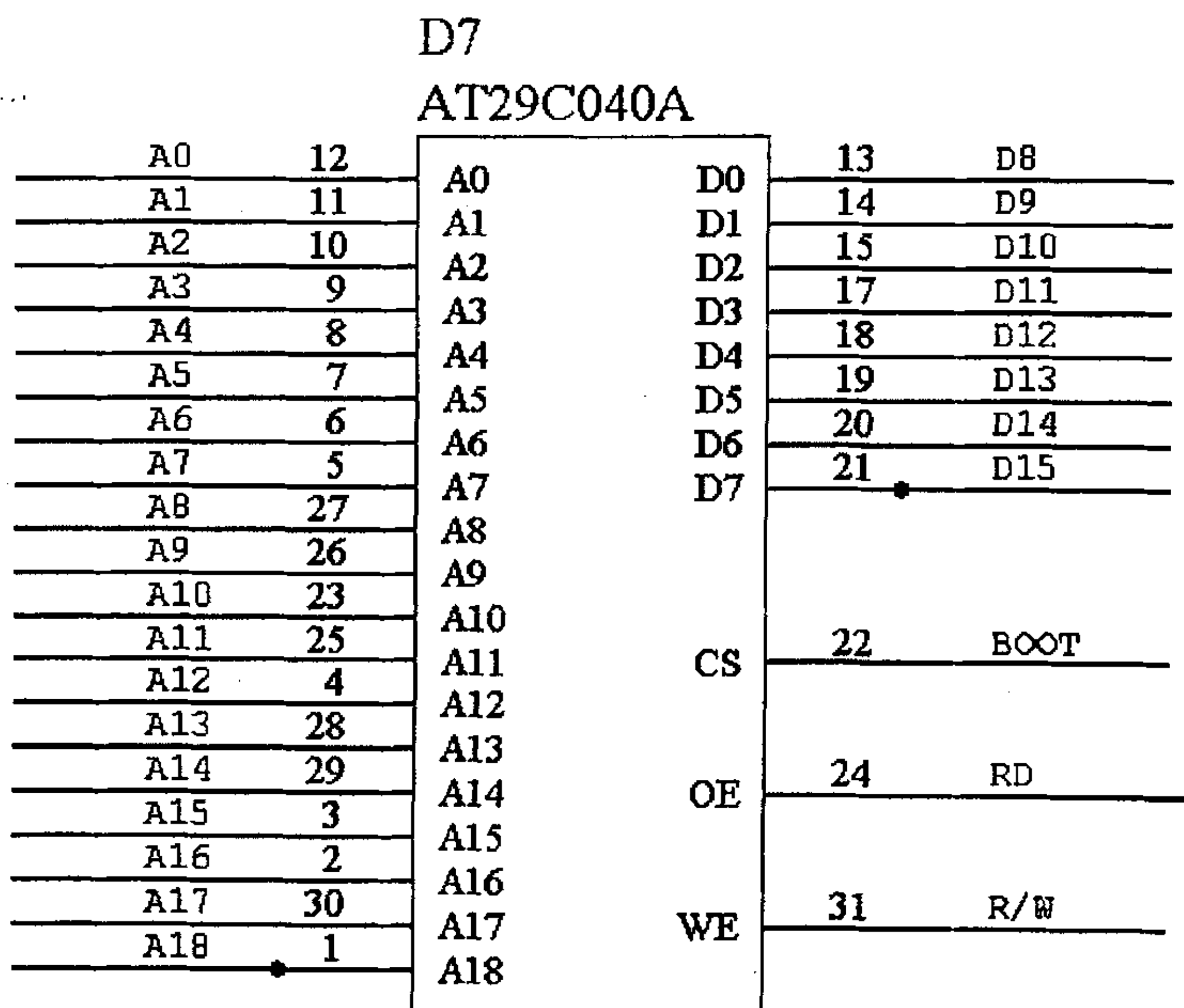


图 4.6 FLASH 的存储接口图

#### 4.1.2.4 MC68332 和 PC 串口的接口电路

采集系统需要和 PC 机进行串行通讯, 这个通讯方式一般是通过 RS232 接口实现的。目前 PC 上实现的 232 接口标准是 RS-232C, RS-323C 标准是美国 EIA(电子工业联合会) 与 BELL 等公司一起开发的 1969 年公布的通信协议。它适合于数据传输速率在 0~20000b/s 范围内的通信。这个标准对串行通信接口的有关问题, 如信号线功能、电器特性都作了明确规定。由于通行设备厂商都生产与 RS-232C 制式兼容的通信设备, 因此, 它作为一种标准, 目前已在微机通信接口中广泛采用。EIA-RS-232C 对电器特性、逻辑电平和各种信号线功能都作了规定。

- ◆ 在 TxD 和 RxD 上: 逻辑 1(MARK)=-3V~-15V
  - ◆ 逻辑 0(SPACE)=+3~+15V
- ◆ 在 RTS、CTS、DSR、DTR 和 DCD 等控制线上:
  - ◆ 信号有效(接通, ON 状态, 正电压)=+3V~+15V
  - ◆ 信号无效(断开, OFF 状态, 负电压)=-3V~-15V

从上面可以看出 RS-232C 的电平标准和 CMOS 电平是不一样的, 因此必须

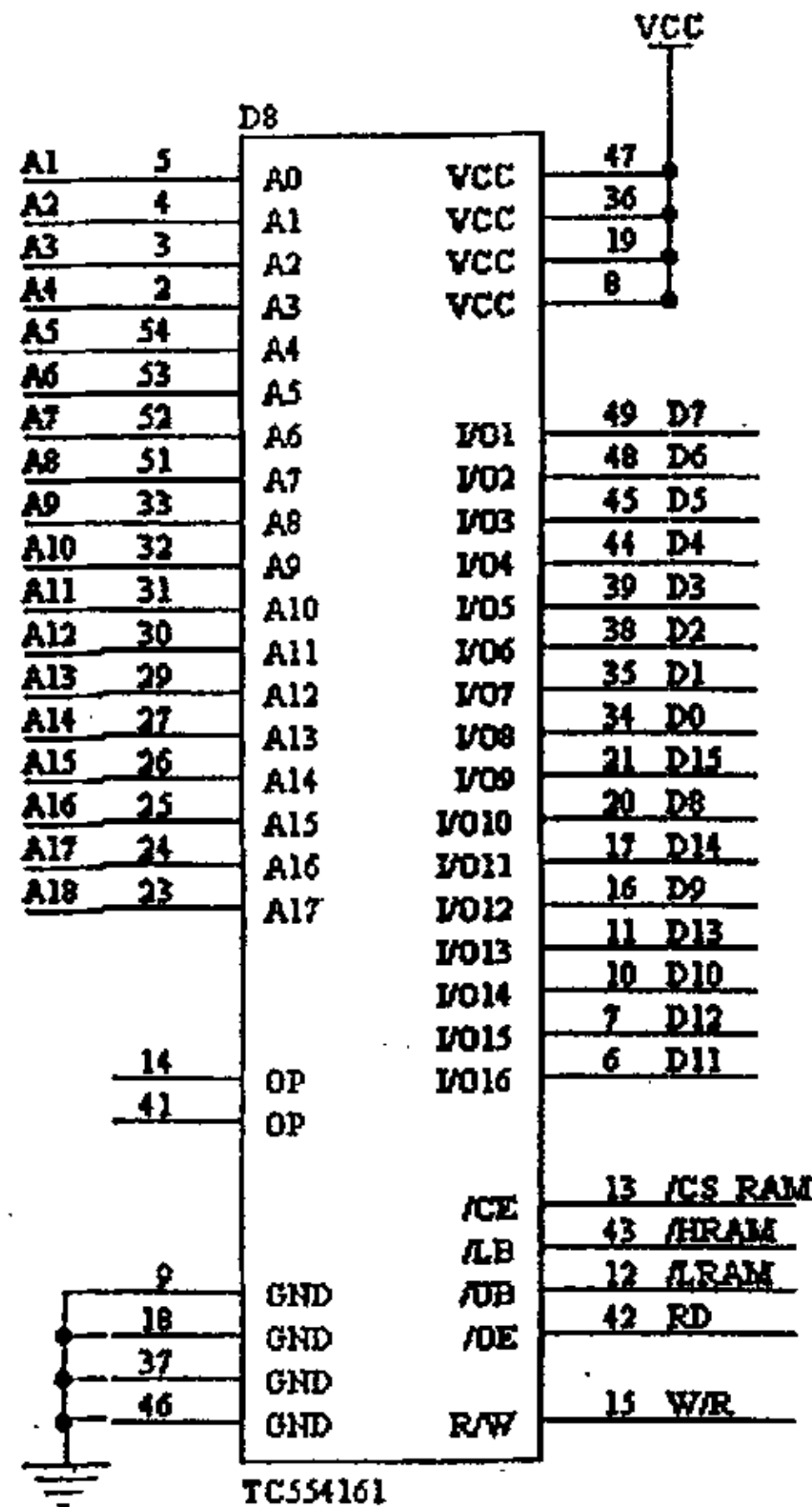


图 4.7 SRAM 的存储接口图

有一个接口芯片来完成这样的功能。我们选用了 MAXIM 公司的 MAX202 芯片实现这样的电平转换。下图是 232 的接口电路图。

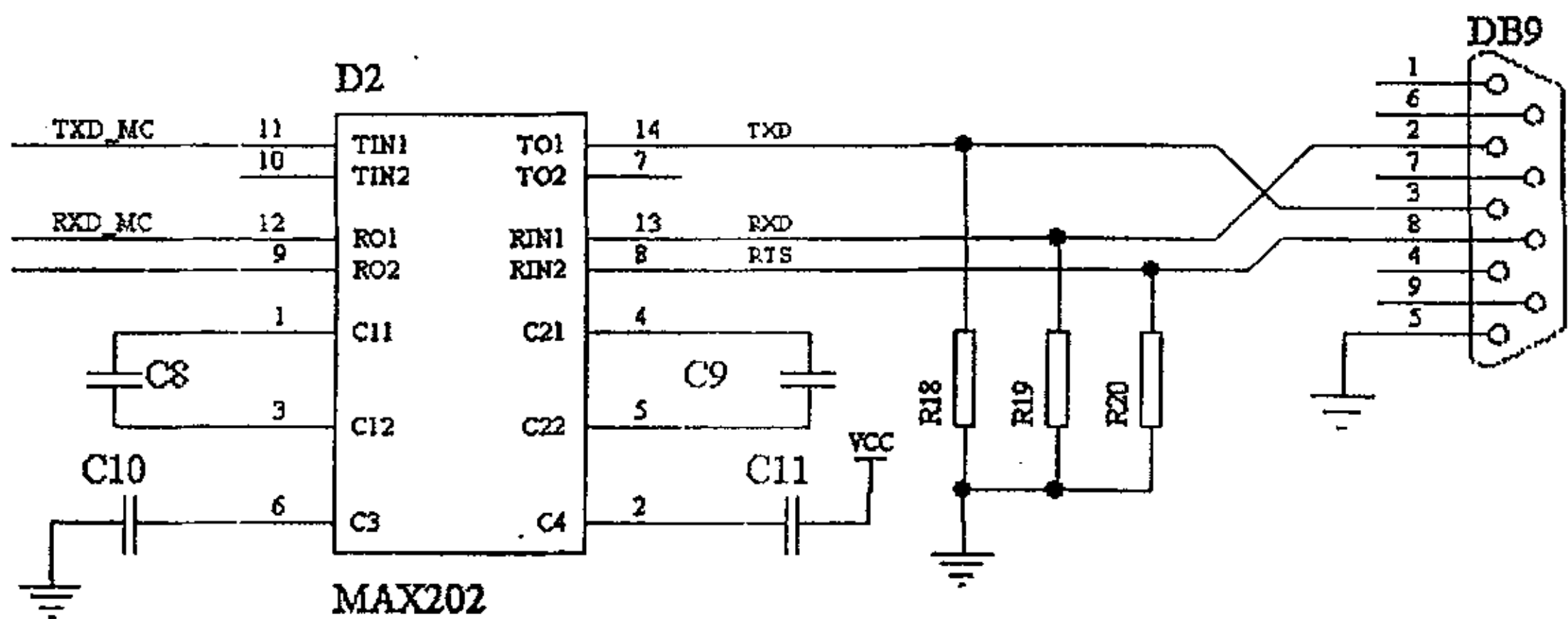


图 4.7 RS232 接口电路图

## 4.2 $\mu$ COS-III 嵌入式实时操作系统及其在 MC68332 上的移植

### 4.2.1 实时多任务操作系统概述

嵌入式系统是近年来计算机应用领域的一个研究热点。嵌入式系统是嵌入到目标设备,辅助目标设备完成相应功能的计算机系统。嵌入式操作系统是运行在嵌入式计算机系统上的一种系统软件,是嵌入式计算机系统的一部分<sup>[20]</sup>。

随着通讯、网络技术的飞速发展,国内外由专家提出 Web Anywhere 的概念,网络已经延伸到生产生活的每一个角落。从机顶盒、DVD、到维纳斯计划、女蜗计划等的报道,无不说明嵌入式的发展领域将会多么广阔。

实时系统是对外来事件在限定的时间里面做出反应的系统。实时系统的正确性不仅依赖于系统计算的正确性,还依赖于产生这个结果的时间。实时系统的基本特征就是实时操作模式。实时操作模式指的是,在计算机内部,用于处理从外部到达的程序总处于就绪状态,而这些程序的运行结果只在确定的时间里面有效。在本系统里面,连续的数据采集过程就要求一个确定的响应时间,否则数据点就有可能丢失,这一点必须由整个系统的硬件软件系统来保证。

实时计算也越来越渗透到越来越多的应用领域,包括过程控制、核电站、智能车辆公路系统、航空航天、飞行控制、自动控制、办公自动化、计算机外设、多媒体应用、通信、医疗、军事等领域,都在广泛的使用实时操作系统。

随着后 PC 时代的到来,计算机在生产生活中的应用也日益广泛。而操作系统作为各种应用程序的底层平台,有着十分重要的作用。自 60 年代操作系统问世以来,有三种操作系统一直在实际当中广泛应用<sup>[21]</sup>:分时操作系统、多道批处理操作系统和实时操作系统。在嵌入式计算机系统内运行的实时操作系统称为嵌入式实时操作系统。

目前,通用操作系统一直是微软的天下,桌面操作系统一直为微软等少数大公司垄断着,而嵌入式系统却完全不一样。嵌入式系统和具体的应用领域结合紧密,因此,设计出一个满足所有功能要求的操作系统是不现实的;另外,嵌入式操作系统资源有限,特别是嵌入式实时操作系统,对代码要求苛刻,也不可能由一个通用的操作系统来完成,因此,有专家指出,嵌入式系统是我国信息产业发展的新的生机。

嵌入式操作系统有实时的,也有非实时的,如 PDA 应用等。目前流行的商业嵌入式操作系统主要有 Wind River System 公司的实时操作系统 VxWorks、pSOSystem, Mentor Graphics 公司的实时操作系统 VRTX、Microsoft 公司的支持 Win32 API 的实时的 WinCE、实时的 Embedded NT 等。另外,在自由软件风靡

的今天,也有很多开放的嵌入式操作系统,如在Linux基础上为没有内存管理单元(MMU)设计的非实时uClinux、在Linux基础上为了满足实时性能要求的实时操作系统RTLinux等。另外,由Jean Labrosse先生写的 $\mu$ COS-II抢先式实时操作系统在国内推广也越来越多。

在实时系统中,所有的重要组件所使用的资源都需要有时间限制,为此就需要新的任务调度策略。没有时间限制的进程和任务调度方法早已被人们广泛研究,比如一般的计算机系统所采用的先进先出(FIFO)、短任务优先或时间片轮转调度等技术,然而,这些方法对实时系统都不太适宜。因为这些调度策略都是试图减少系统的平均响应时间,没有涉及时间限制,而实时系统的调度策略必须保证在所有情况下任务都能满足其限定时间的要求。

实时调度算法可以分为两类<sup>[22]</sup>:可抢先调度算法和非抢先调度算法。可抢先调度算法允许进程在运行期间被中断;而非抢先调度算法不允许一个处于运行期间的进程被中断,但在当前运行进程所分配的时间片用完后,可以进行新的调度。二者各有其优越性和不足。

#### (一) 可抢先调度算法

可抢先调度算法处理允许在执行期间被中断的任务,适用于上下文切换开销小的系统。在可抢先调度算法中,按照对任务派遣方式的区别,可以分为优先级任务派遣算法和时间驱动任务派遣算法两类。

##### 优先级任务派遣

在优先级任务派遣算法中,每一个任务被赋予一个优先级,CPU执行等待任务中优先级最高的任务。调度算法则研究如何赋予任务优先级,以及用一个或若干个条件来判断任务是否可被调度。

##### 时间驱动任务派遣

采用时间驱动任务派遣的可抢先调度算法大多数同非抢先调度算法类似,只是在这种可抢先调度算法中任务可以被分成更小的子任务来进行调度,从而能够实现抢先。

#### (二) 非抢先调度算法

非抢先调度算法适用于抢先带来的上下文切换开销过大的系统,不使用中断还能够增强系统的确定性。几乎所有的非抢先调度算法都采用时间驱动的时间片分配方式完成实时任务的调度。

一个实时操作系统可能由若干部分组成,有些部分,如文件系统、网络系统可以根据系统要求进行剪裁。所有的操作系统都必须有一个核心,就是内核。内核的质量和性能对整个实时操作系统来说是必不可少的,其设计体系结构决定了操作系统内核和其它系统组件的关系。



## 4.2.2 $\mu$ COS-II 实时操作系统

$\mu$ COS-II 是一个源代码公开, 几乎完全免费的实时操作系统。由美国人 Jean Labrosse 先生最初为了一个基于 Intel 80C188 的嵌入式产品的开发自行编写出来, 后经过很多产品的严格测试, 并移植到了很多不同架构的微处理器上, 已经证明比较适合中小型的嵌入式系统的开发<sup>[23][24][25]</sup>。目前的最高版本是 2.5。

$\mu$ COS-II 不是一个非常完善的操作系统, 它不提供很多大型操作系统如 pSOS、VxWorks 的完整服务, 如编译环境、文件系统、网络系统、多处理器支持等,  $\mu$ COS-II 仅仅是一个内核, 主要的功能是完成任务调度。因此, 这个内核能够很快的移植到新的处理器架构上去, 并能够在此基础上进行各种功能的开发。

### 1、可移植性

$\mu$ COS-II 的代码使用可移植性很强的 ANSI C 写的, 而和微处理器等硬件相关的部分使用汇编语言来写的, 仅占整个系统的很小的一部分, 因此,  $\mu$ COS-II 很容易就可以移植到许多新的微处理器上面。只要该微处理器具有堆栈指针, 有 CPU 内部寄存器入栈、出栈指令; 使用的 C 编译器支持内嵌汇编或者 C 语言可扩展、可链接汇编模块, 使得开中断、关中断能够在 C 语言中实现。目前,  $\mu$ COS-II 已经在各种 8 位、16 位、32 位、64 位的微处理器、微控制器、数字信号处理器 (DSP) 上运行。

### 2、可固化

$\mu$ COS-II 是为嵌入式系统设计的, 只要提供下载手段,  $\mu$ COS-II 操作系统可以和用户代码一起固化到嵌入式系统的 ROM、FLASH 里面去, 成为系统的固件 (Firmware)。

### 3、可剪裁

可剪裁性能够满足嵌入式系统资源有限的限制。用户需要的功能可以通过一个定义头文件进行预先的定义, 而用户不需要的功能也可以在这个头文件中把相应的功能给屏蔽掉。可剪裁性是一个嵌入式系统的特殊要求。

### 4、抢先式

$\mu$ COS-II 是一个完全的抢先式实时操作系统。 $\mu$ COS-II 内核总是调度当前处于就绪条件下的优先级最高的任务。图 4.8 是  $\mu$ COS-II 的任务转换图。我们可以看出抢先式实时操作系统在任务切换过程中, 总去调度优先级最高的就绪任务。

### 5、多任务

$\mu$ COS-II 可以管理最多 64 个任务, 其中, 8 个任务保留给系统。每个任务都有自己的优先级, 根据优先级来进行任务调度。 $\mu$ COS-II 不支持时间片轮转

法调度（如一般的分时系统）。

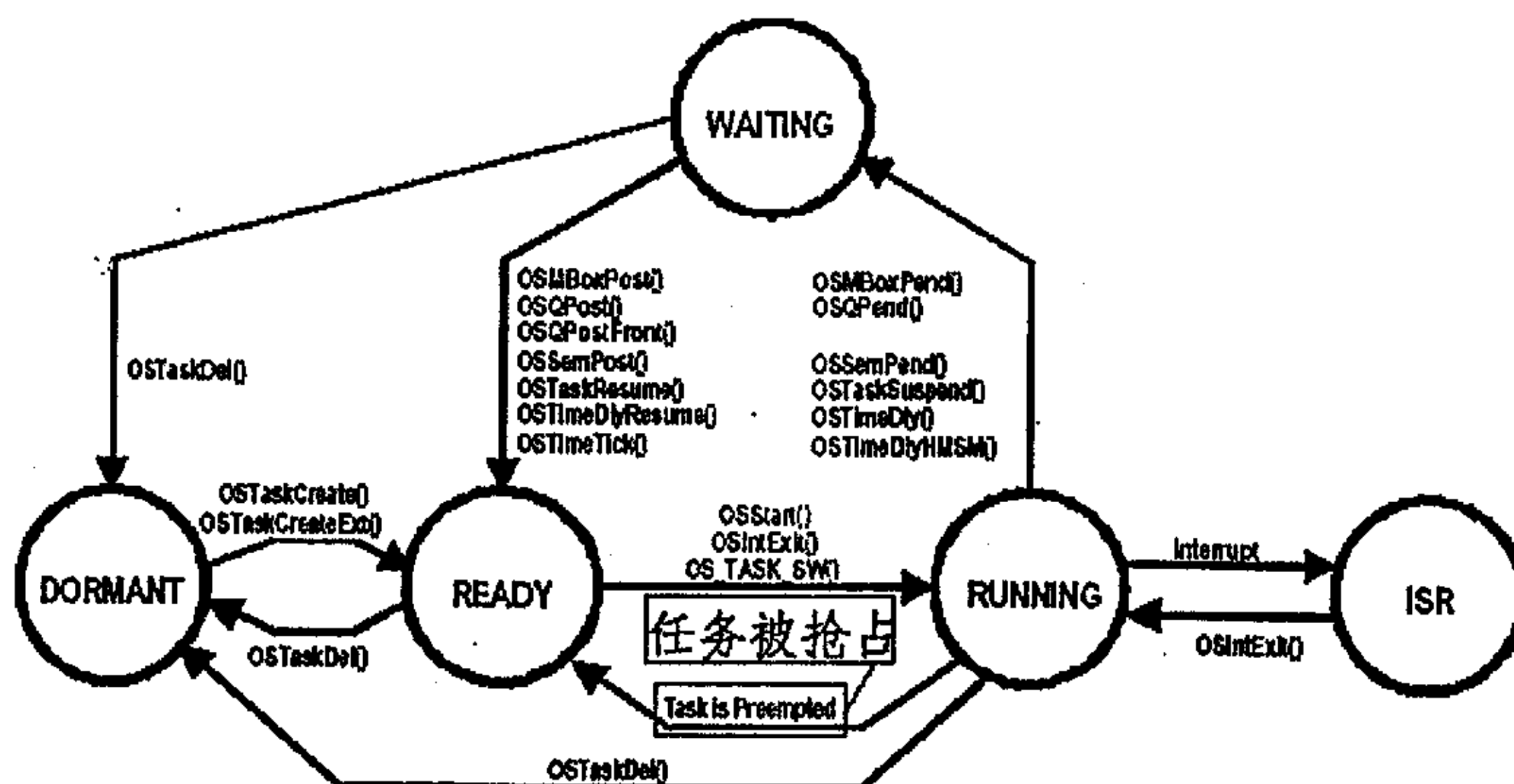


图 4.8  $\mu$  COS-II 的任务转换图

## 6、可确定性

$\mu$  COS-II 的函数调用执行时间具有确定性。 $\mu$  COS-II 提供函数的运行测量时间函数，能够测量单个函数运行时间的多少。 $\mu$  COS-II 的函数调用时间和服务执行时间是可知的。 $\mu$  COS-II 服务的执行时间不依赖于应用程序任务的多少。

## 7、任务栈

每个任务都有自己的堆栈，每个任务的栈空间大小、位置都是独立的。

## 8、系统服务

$\mu$  COS-II 提供很多实时多任务操作系统需要的系统服务，如邮箱、消息队列、信号量、块大小固定的内存申请和释放（如 malloc() 函数）、时间处理等。

## 9、中断管理

中断使得正在执行的任务暂时挂起。高优先级的任务如果被唤醒的话，高优先级的任务会在中断结束后进入运行状态。中断可嵌套，共 255 级。

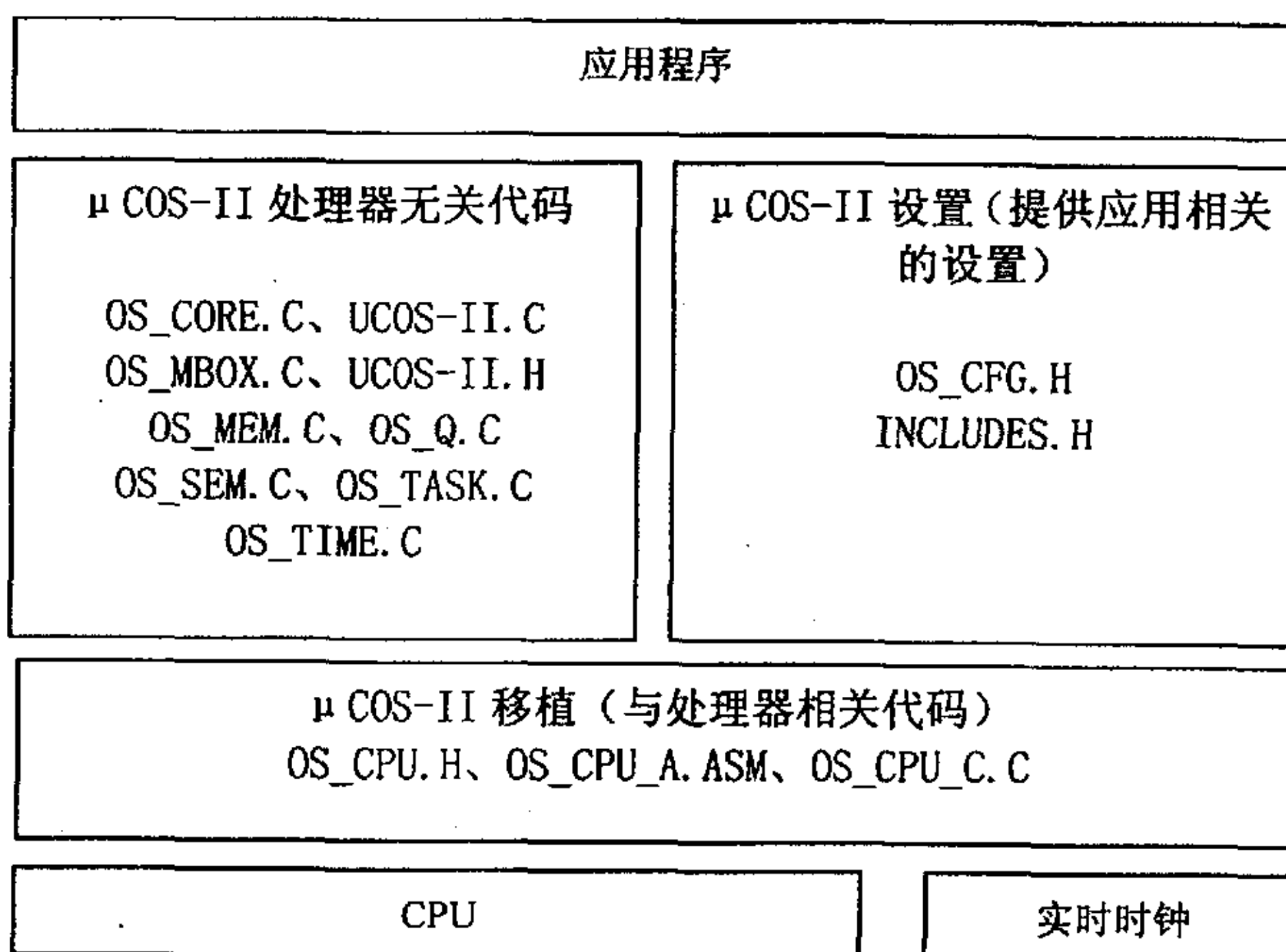
## 10、可靠性、稳定性

在  $\mu$  COS-II 的网站上提供了很多成功的商业应用的例子。实践证明， $\mu$  COS-II 在运行的时候还是相当可靠的，可以满足大多数的设计要求。

在我们的便携式故障诊断系统中，我们将  $\mu$  COS-II 移植到了 MC68332 微处理器上，并且实现了整个系统的资源管理。包括键盘的读取、数据的采集和参数设置、网络协议、串口通讯、存储系统的控制等功能。

4.3  $\mu$ COS-II 到 MC68332 的移植

移植指的是使一个实时操作系统能够在某个微处理器或位控制器上运行的过程。 $\mu$ COS-II 的大部分代码是用 C 语言来写的, 只要采用相应的 C 语言编译器就可以了, 还有一部分代码是用户必须自己编写的, 这部分代码是和硬件直接相关的。如图 4.9, 是  $\mu$ COS-II 的软硬件体系结构。

图 4.9  $\mu$ COS-II 的软硬件体系结构

从图上可看出, 我们需要做得移植工作主要集中在三个文件当中: OS\_CPU.H、OS\_CPU\_A.ASM、OS\_CPU\_C.C, 现在, 我们逐个文件的设计, 完成到 MC68332 处理器的移植。另外, 我们是在假定我们使用的是 GNU 的开放的交叉编译器 m68k-elf-gcc.exe。

## 1、OS\_CPU.H

OS\_CPU.H 主要是设置一些常量, 并声明十个数据类型和三个宏定义。下边是它的源代码。

```

typedef unsigned char  BOOLEAN;
typedef unsigned char  INT8U;
typedef signed   char  INT8S;
typedef unsigned short INT16U;
typedef signed   short INT16S;
typedef unsigned int   INT32U;
typedef signed   int   INT32S;
typedef float         FP32;

```

```

typedef double      FP64;
#define BYTE        INT8S
#define UBYTE       INT8U
#define WORD        INT16S
#define UWORD       INT16U
#define LONG        INT32S
#define ULONG       INT32U
typedef unsigned short OS_STK;

```

之后，在  $\mu$ COS-II 中的所有代码都用上边定义的那些宏（INT8U 等），这样整个系统就不要做很多的修改了。OS\_STK 定义了栈的组织结构，在 MC68332 当中，这个栈类型定义为无符号的短整型双字节栈结构。

然后，我们实现两个宏定义 OS\_ENTER\_CRITICAL() 和 OS\_EXIT\_CRITICAL()，这两个宏是实现临界区的关键函数。临界区能够实现在对临界区内的资源进行操作的时候，不会因为可屏蔽中断或任务切换导致资源被修改而引起的不确定性。两个宏定义都是通过对 MC68332 的标志寄存器当中中断标志位的置位和复位来完成的，代码如下：

```

#define OS_ENTER_CRITICAL()  asm("MOVE SR, -(A7)\n ORI #$0700, SR\n")
#define OS_EXIT_CRITICAL()   asm(" MOVE (A7)+, SR\n")

```

除此之外，还必须定义处理器的堆栈增长方向。MC68332 的堆栈增长方向是通过 A7 寄存器来控制的，可以向上增长（数据入栈的时候，堆栈地址增加）或者向下增长（数据入栈的时候，堆栈地址减少）。在这里，我们定义堆栈增长方向是向下增长的，如对 SR 的堆栈操作：MOVE SR, -(A7)，A7 指针向下减少。

## 2、OS\_CPU\_A.ASM

OS\_CPU\_A.ASM 主要要求编写四个简单的函数：

```

xdef  _OSCtxSw——OSCtxSw()
xdef  _OSIntCtxSw——OSIntCtxSw()
xdef  _OSIntExit68K——OSIntExit68K()
xdef  _OSStartHighRdy——OSStartHighRdy()
xdef  _OSTickISR——OSTickISR()

```

四个函数用于任务切换中，对堆栈的操作，右边是在别的函数中调用该汇编代码的时候相应的 C 语言函数调用原型。

## 3、OS\_CPU\_C.C

OS\_CPU\_C.C 中用户必须实现一个关键的函数 OSTaskStkInit()，该函数要完成任务堆栈初始化的结构，让任务在第一次运行的时候堆栈状况就像刚发生过中断并将所有的寄存器保存到堆栈中的情形一样。这样做的目的是任务第一次被调度的时候，由于通常的任务调度是在时钟中断之后被开始的，只要有高优先级的任务处于就绪状态，就会被调度运行。这部分代码由 OSCtxSw() 来完成。因此，

任务第一次被创建并进入就绪状态的时候,就必须让操作系统认为任务是由于被定时中断调度到堆栈当中保存起来,并处于随时准备运行的就绪状态。这样的工作就必须初始化任务堆栈的结构。



## 第五章 便携式故障诊断仪的软件设计

### 5.1 软件结构

便携式故障诊断仪的软件结构由如下几个部分组成：

#### 1、硬件驱动程序

硬件驱动程序完成对系统硬件资源的抽象，为上层软件提供函数调用。通过抽象底层硬件的物理行为，能够为软件设计建立一个优良的接口，使上层软件在底层标准接口定义下工作；将来硬件性能提高了，上层软件不用作过多的修改，只需要修改底层驱动就可以了。另外，硬件驱动程序的提出也能够方便项目开发，软硬件协同进行设计。

驱动程序主要包括：采集分析数据的读取、键盘状态读取、液晶显示模块的操作、网络接口以太网数据帧的获取。

#### 2、实时操作系统

我们采用了  $\mu$ COS-II 实时操作系统作为我们的开发平台，让其能够满足连续数据采集和实时在线诊断的要求。

#### 3、应用任务

多个应用任务协同完成数据的读取、键盘状态的读取、数据的显示、网络协议的实现等功能。图 5.1 是整个系统的软件模块的框图。

### 5.2 软件设计

#### 5.2.1 驱动程序设计

如图 5.1 所示，键盘、液晶显示部分底层驱动程序需要对键盘状态进行读取、对液晶显示控制芯片的命令口和数据口进行读写；DSP 数据通讯驱动主要完成对双口 RAM 的通讯数据存储空间的读写，完成和 DSP 的数据通讯；采集分析程序 DPRAM 半满中断处理程序在 DPRAM 半满时候读取相应的采集数据或分析数据，并存入主板缓冲区当中，为数据显示提供数据；定时中断完成定时器(TPU)的定时任务；串口发送接口中断完成数据通过串口的发送、接收，与上位 PC 进行数据通讯；RTL8019 芯片的驱动程序主要完成以太网数据帧的发送和接收。由于除了通讯程序之外，其它驱动程序仅涉及到内存空间的读写，代码比较简单。我们

重点讨论两个通讯程序设计的问题。

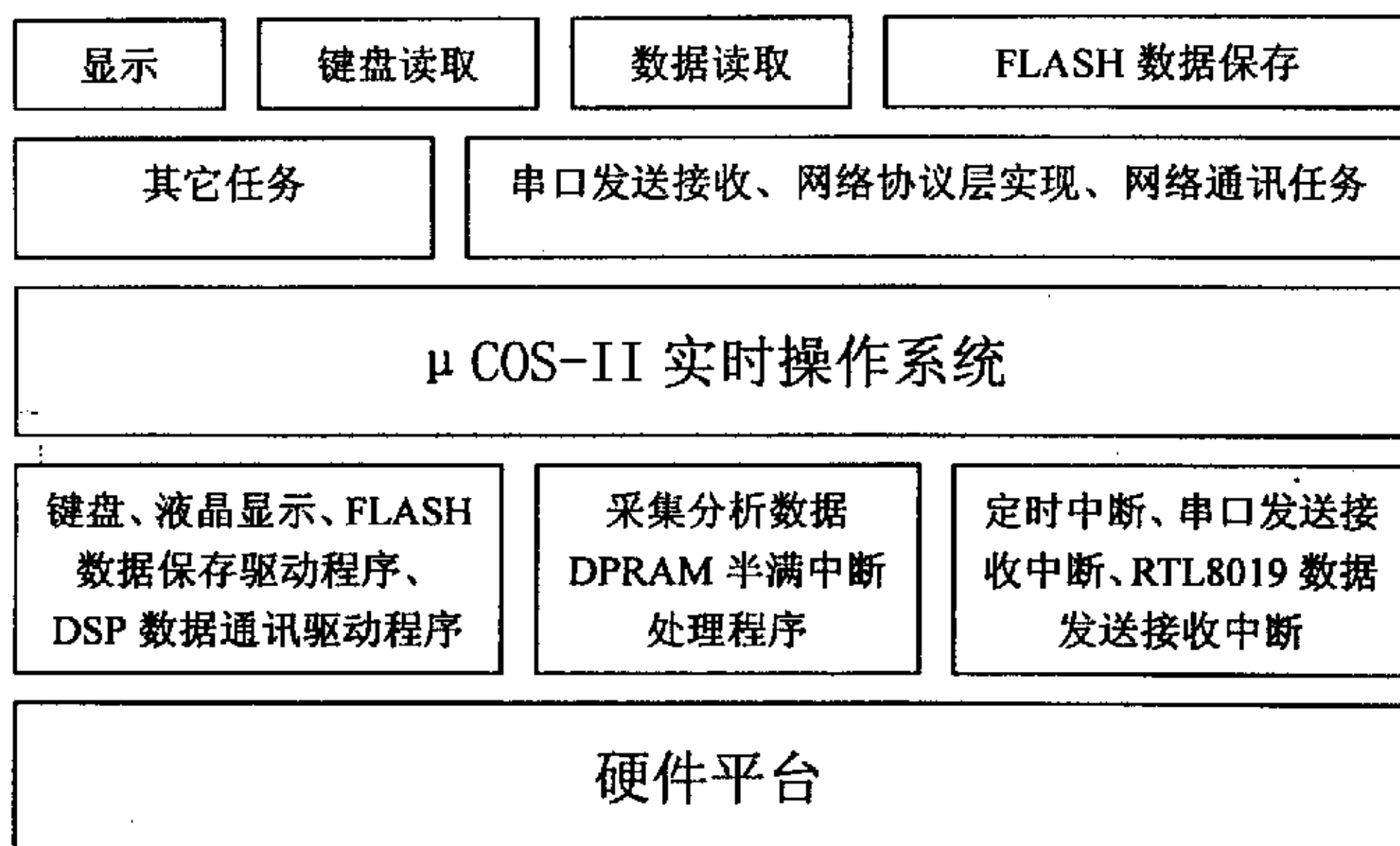


图 5.1 系统软件模块框图

### 5.2.1.1 串口通讯程序设计

MC68332 的串口 SCI 是双缓冲功能的异步串口，同时 SCI 是 QSM 模块的一部分。对串口的操作需要在主模块里面对串口进行初始化和发送接收数据的缓冲区的初始化。

#### A) 串口初始化

包括中断仲裁号、中断级别、中断向量、波特率的设定。

#### B) 发送数据缓冲区的初始化

#### C) 接收数据缓冲区的初始化

在串口中断处理程序里面，对中断的性质作简单判断后，进行进一步的处理，下边是中断处理程序的伪代码。

```

/* SCI 模块中断处理程序 */
#pragma interrupt()
void INT_SCI_Com(void)
{
    Read the SCI status register;
    If the register indicate a transfer interrupt occurred
    Send the date in transfer buffers to SCI transfer buffer until the
    data be completely sent;
    Disable the transfer end interrupt and transfer interrupt bits;
    Enable the receive interrupt;
}
  
```

```

Else if the register indicate a receive interrupt occurred
    Receive the first byte indicate a frame begin;
    Receive the second byte indicate a frame numbers;
    Receive the data and save to a receive buffer until the frame number
    reach;
    Send a signal to the task there is a data in receive buffer;
}

```

### 5.2.1.2 网络驱动程序的设计

网络驱动程序负责把以太网上的数据进行收发,并判断是否是自己需要的数据,发给上层协议作进一步处理。对 RTL8019AS 的数据收发有查询和中断两种方式。

在查询方式下,主程序通过 CURR 和 BNRV 两个寄存器的值来判断是否收到一帧数据。当 BNRV 与 CURR 不等时,说明接收缓冲区接收到了新的帧,主程序读取数据后,以读取帧的第二个字节(下一帧的页地址)更新 BNRV,主程序循环跟踪 CURR 和 BNRV 达到数据的接收目的。主程序在发送一帧数据时,先要查发送状态寄存器(TSR)判断上一帧是否发送完毕,如果已经发送完毕则开始下一帧的发送。

在采用了多任务的环境,一般采用中断方式来处理 RTL8019AS 的收发。下图是数据收发的中断处理程序的流程。当中断处理程序响应 RTL8019AS 的中断时,根据读取的中断状态寄存器(ISR)的值来确定程序的走向。在本系统中,采用中断处理的办法实现数据的发送和接收,而数据的收发是通过远程 DMA 来控制的。

对于主控制器来说 RTL8019AS 相当于一个向以太网收发数据的 DMA 控制器。RTL8019AS 内部将这些操作分为本地 DMA(Local DMA)和远程 DMA(Remote DMA)。与 DMA 操作相关的寄存器如图 5.2 所示。

#### 1、DMA

本地DMA完成RTL8019AS与以太网的数据交换。发送时,数据包从缓冲区移到FIFO然后从网口送出,如果发生冲突,RTL8019AS会自动重发而不需要主处理器的干预;接收时,RTL8019AS从网口接收符合要求的数据(地址匹配,无帧错和校验错等),然后从FIFO移到缓冲区。

远程DMA完成与主控制器的数据交换。发送时,主处理器将数据送入 RTL8019AS 内存,并告诉 RTL8019AS 待发送的数据包的起始地址(TPSR)和长度(TBCR0、TBCR 1),然后由 RTL8019AS 完成发送;接收时,主处理器从 RTL8019AS 的接收缓冲区中将已接收到的数据包读出。

需要指明的一点是,RTL8019AS 的 DMA 与平时所说的 DMA 有点不同。

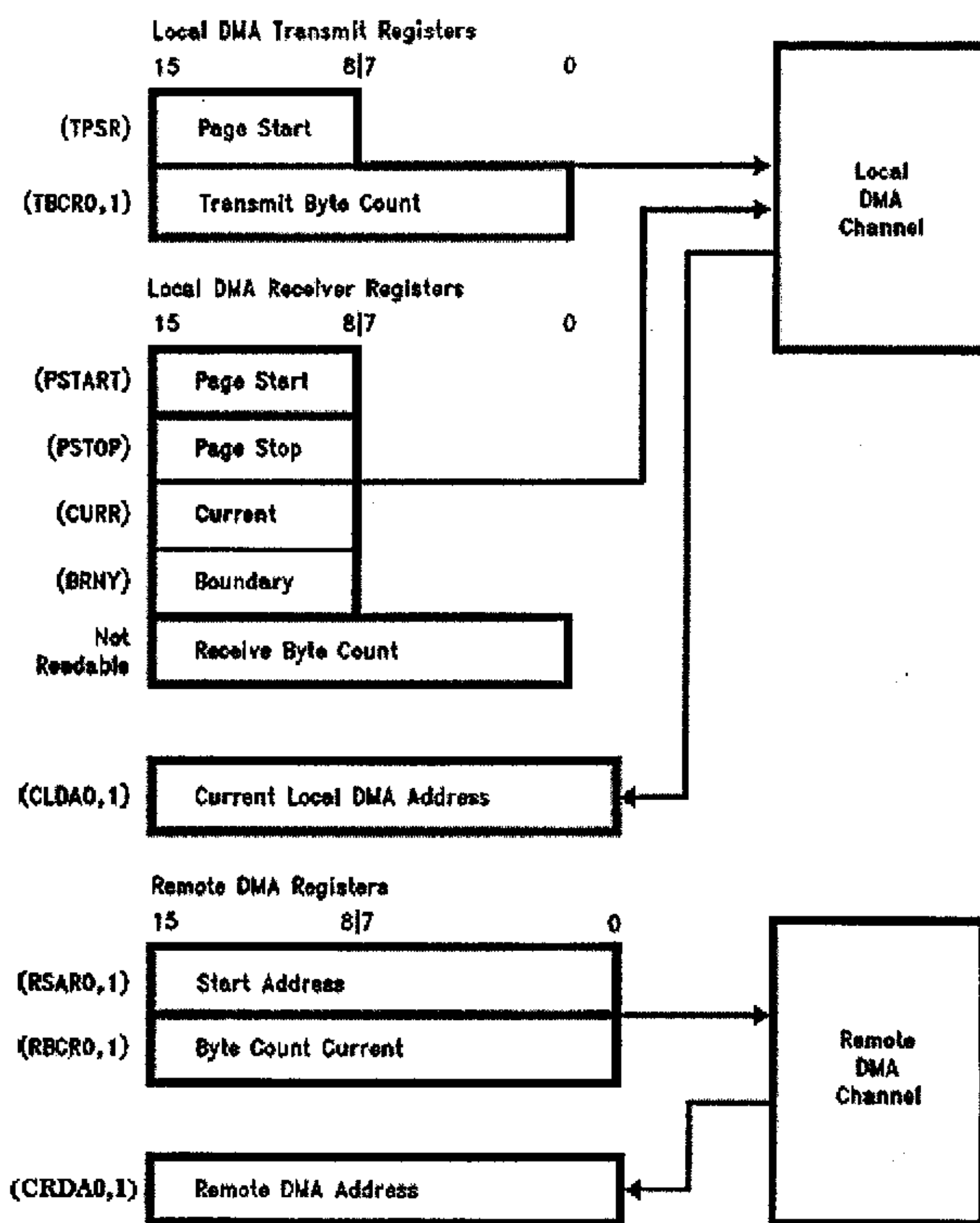


图 5.2 与 DMA 操作相关的寄存器

RTL8019AS的Local DMA操作是由控制器本身完成的,而其Remote DMA 并不是在无主处理器的参与下,数据能自动移到主处理器的内存中,它的操作机制是这样的:主处理器先赋值于Remote DMA 的起始地址寄存器RSAR0、RSAR1和字节计数器RBCR0、RBCR1,然后在RTL8019AS的DMA I/O地址上读写数据,每读写一个数据RTL8019AS会将字节计数器减小,当字节计数器减到0时,远程DMA操作完成,主处理器不应再在DMA端口地址上读写数据。

可以通过读取CRDA0-1和CLDA0-1来获得当前DMA操作的地址。

## 2、接收数据

接收数据的基本过程是,本地DMA从网口读入数据存储在RTL8019AS的内存中,远程DMA将RTL8019AS内存中的数据送往主处理器。

RTL8019AS的接收缓冲区是一个环形缓冲区,如图5.3所示。接收缓冲区在内

存中的位置由页起始寄存器(PSTART)和页终止寄存器(PSTOP)指出。环形缓冲区的当前读写位置则由当前页寄存器(CURR)和边界页寄存器(BNRY)指出。

CURR是RTL8019AS写内存的指针,那么初始化它就应该指向PSTART(这里是0x4C)。网卡写完接收缓冲区一页,就将这个页地址加一,  $CURR = CURR + 1$ , 这是网卡自动加的。当加到最后的空页(这里是0x80, PSTOP)时,将CURR置为接收缓冲区的第一页(这里是0x4C, PSTART),这也是网卡自动完成的。当  $CURR + 1 = BNRY$  时,表示缓冲区全部被存满,数据没有被用户读走,这时网卡将停止往内存写数据,新收到的数据包将被丢弃不要,而不覆盖旧的数据。此时实际上出现了内存溢出,中断状态寄存器中的OVW被置位。

BNRR是数据缓冲区的读指针,要由主处理器来操作,初始时也指向PSTART。主处理器从RTL8019AS读走一页数据,要将BNRY加一,当BNRY加到最后的空页(0x80, PSTOP)时,同样要将BNRY变成第一个接收页(这里是0x4C, PSTART)。当  $BNRY = CURR$  时表示缓冲区已空,无数据可读了。

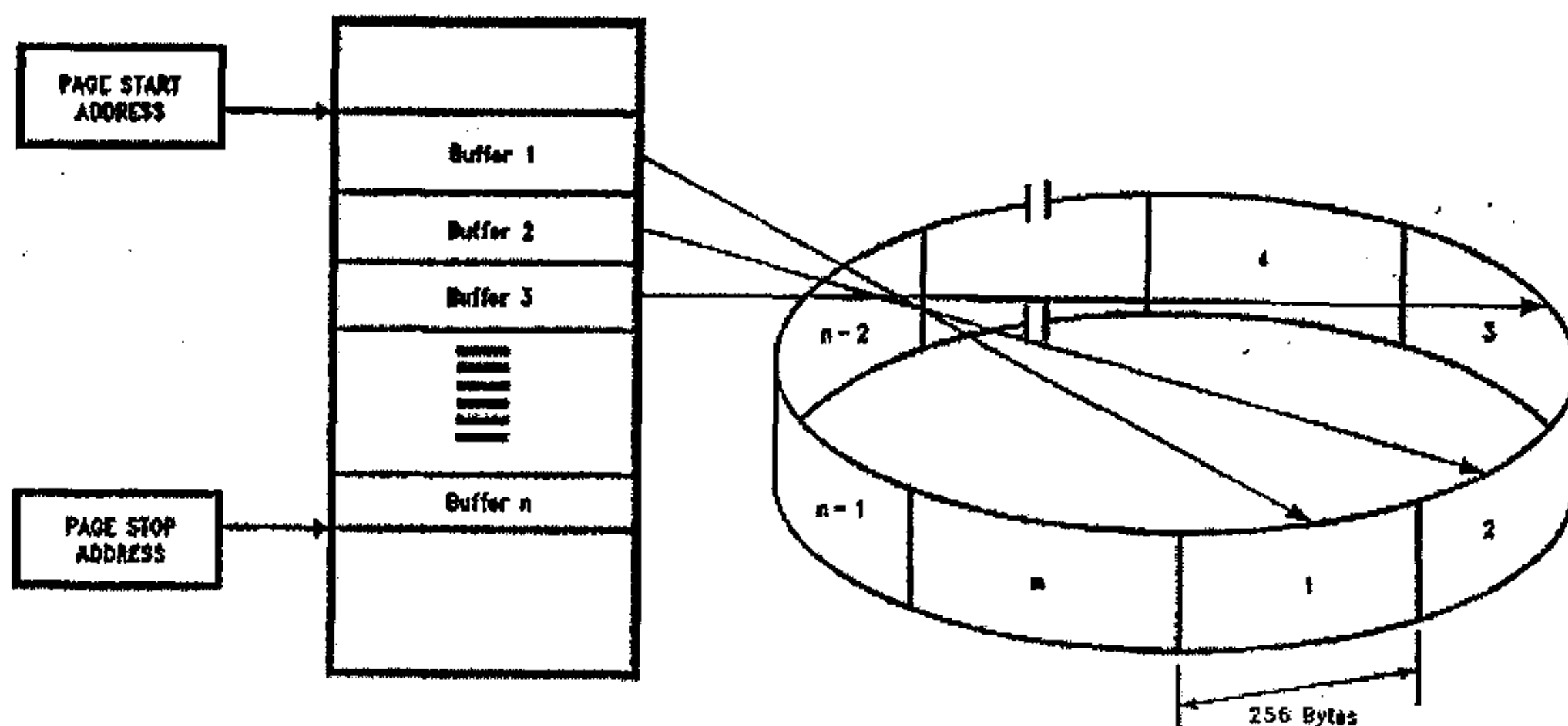


图 5.3 环形接收缓冲区

下面总结一下这个环形缓冲区的读写规则:

- 初始化:  $CURR = BNRY = PSTART$
- 本地DMA向缓冲区写数据:

if(缓冲区未满)

{

    向CURR页写数据;

$CURR = CURR + 1$ ;

    if( $CURR == PSTOP$ )

$CURR = PSTART$ ;

    if( $CURR == BNRY$ )

    {



```

缓冲区满;
产生中断信号;

```

```

}
}

```

♦ 远程DMA从缓冲区读数据:

```

if(BNRY==CURR)
缓冲区空;
else //有数据
{
    从BNRY读数据;
    BNRY=BNRY+1;
    If(BNRY==PSTOP)
    BNRY=PSTART;
}

```

### 3、发送数据

发送数据的基本过程是, 远程DMA将数据写入RTL8019AS的内存中, 本地DMA将内存中的数据从网口送出。

有三个寄存器用于控制发送数据, 它们是传输页起始地址寄存器(TPSR)和传输字节寄存器(TBCR0,1)。传输开始前, 主处理器将初始化这三个寄存器, 然后将命令寄存器(CR)中的 TXP 置位、STP 复位, 即向 CR 写入 xxxxx110(二进制, 其中 x 表示不改变寄存器的原有值), 这时 RTL8019AS 将从内存中把数据读入 FIFO 并逐位送出, RTL8019AS 还会自动为要发送的数据包添加包头和 CRC 校验序列而无需主处理器费心。

图 5.4 是整个网卡驱动程序的流程图。

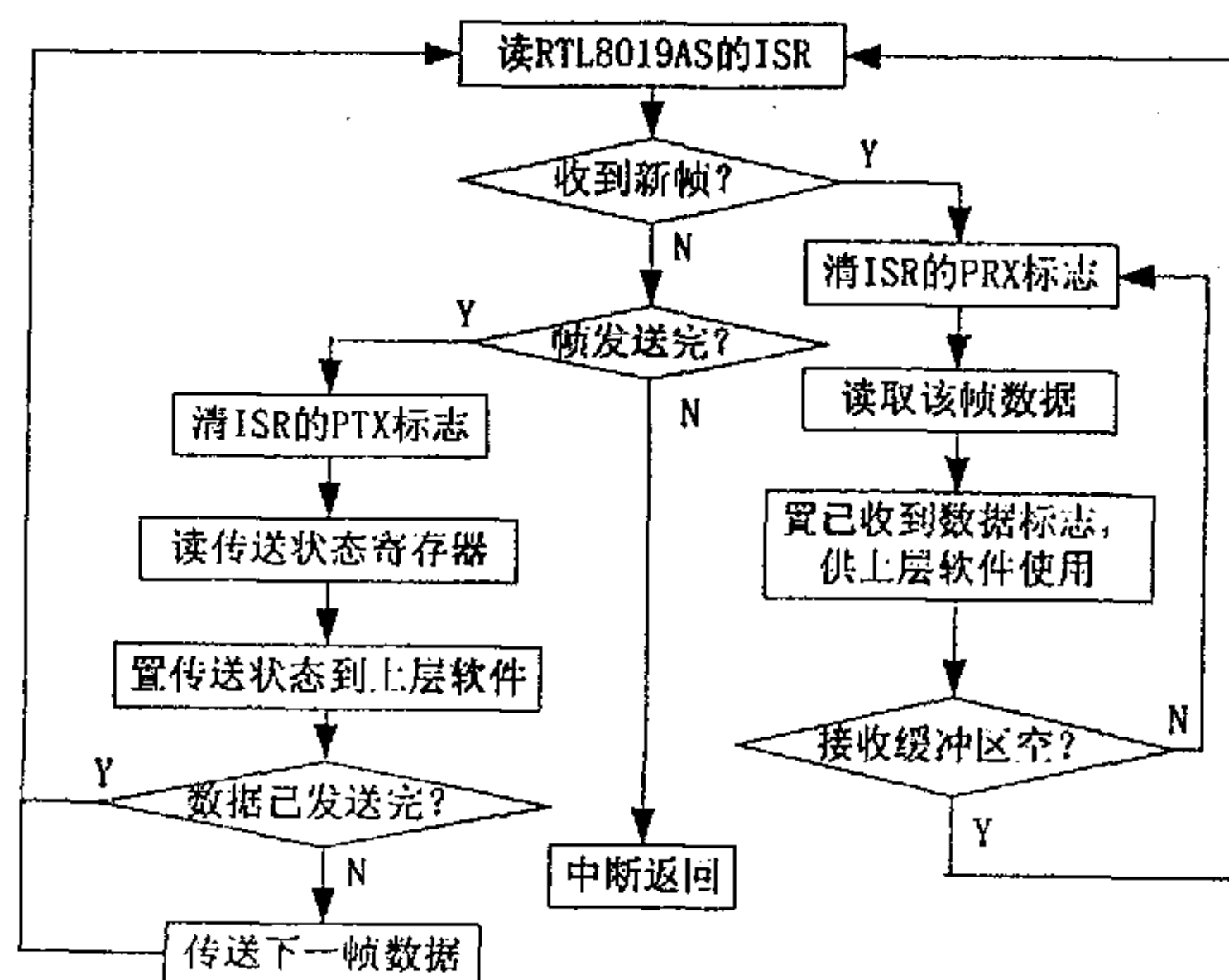


图 5.4 网卡驱动程序的流程图

### 5.2.2 任务设计

实时操作系统下的应用程序设计和原来的前后台应用程序设计的方法不一样。前后台应用程序设计的方法是把系统作为一个主函数 `main()`，主函数管理所有的模块的调用和协调，然后单独写出中断处理程序<sup>[23][24][26]</sup>。下图是原来的前后台应用程序设计的典型运行流程。

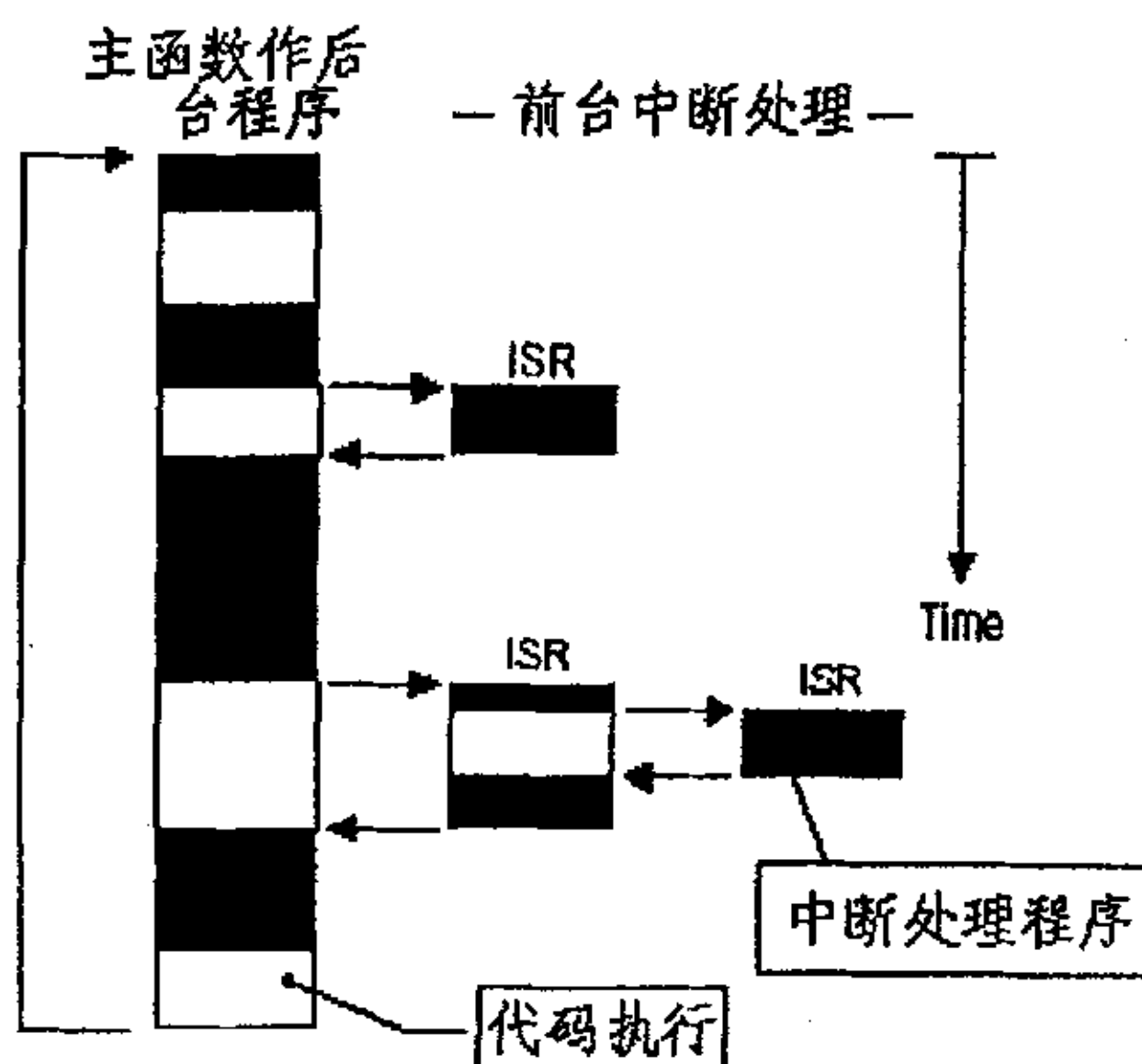


图 5.5 前后台程序典型运行流程

实时多任务的程序设计过程和前后台程序设计的过程是完全不一样的。实时多任务操作系统的程序设计是在实时多任务操作系统的基础上进行开发的，因此，重要的设计过程就是要对系统进行任务划分，根据每一个任务的重要程度划分优先级。

#### 5.2.2.1 嵌入式实时软件的设计特点

嵌入式实时软件不单纯是运行在嵌入式 CPU 或者 DSP 上面的一个软件，嵌入式实时软件设计需要注意的问题比原来的单纯 PC 下的软件设计要多得多，比如，实时性、并发性、任务的生命周期、接口性能、和基于异质平台的设计<sup>[27][28][29]</sup>。

实时性能是大多数嵌入式软件设计的根本要求，程序或者相关的任务必须在指定的时限内对外部时间做出响应。

并发性指的是在同一时刻可能会有若干个事件发生，如何正确的安排这个优先级以及若干任务并行运行如何解决任务之间的通讯等问题需要得到考虑。很少有系统只对一个物理过程进行监控的，系统软件必须同时对来自网络和传感器的多个输入激励进行响应，并且在指定的时间内对外部执行器发出命令。

在一个嵌入式软件里面并不是每一个任务都在运行，任务有自己的生命周

期,如前面 $\mu$ COS-II 的任务转换我们可以看出,一旦任务完成了相应的任务,就必须退出运行态,甚至是不再运行的睡眠状态。

良好的接口性能要求容易使用的函数调用 API 和软件复用。目前,面向对象的软件工程的设计、分析、实现方法已经在嵌入式软件设计当中广泛使用。写出的代码能够在不同的平台下面进行移植,否则产品会很快淘汰,代码没有重用会导致原先研发投资的浪费。

异质性一直都是嵌入式软件的本质特点。首先,是平台的异质,嵌入式系统一般都是由不同的计算模型和实现技术实现的,有可能嵌入式系统的 CPU 不一样,算法实现上有可能是纯软件实现,也有可能用 SOC 或者专用集成电路(ASIC)来实现。其次,是输入激励和输出响应的异质性。系统的事件有可能是周期性的,也有可能是非周期性的,这些都要软件合理的安排任务来响应这些不同类型的输入;系统的输出方式多种多样,可以是单纯的显示,也可能是需要系统去驱动一个执行器,这些也需要软件来实现不同的输出方式。最后,嵌入式系统的编程一般不是单纯的采用高级语言来编写,很多时候是汇编语言和高级语言一起编程实现。如何合理的安排这样的软件结构,使得系统的实现难度减到最小,也是一个嵌入式软件应该注意的地方。

### 5.2.2.2 嵌入式软件工程方法

从上边的分析我们可以看出来,嵌入式实时软件的设计需要一套行之有效的分析设计方法。底层驱动软件的编写仅仅需要知道底层硬件的相关知识,然后采用 C 语言或汇编语言就可以完成对硬件的操作了,但是嵌入式实时软件的任务设计就复杂多了,下面我们介绍一些我们在软件设计当中用到的嵌入式软件的分析 and 设计方法。

在软件系统设计过程中,主要用到了如下的一些方法:语言描述和数学分析方法、流程图、结构图、伪代码和程序设计语言、有限状态机和 Petri 网理论。

语言描述和数学分析:在系统分析和设计当中,语言描述是不可缺少的,但是其功能也是对其他分析设计方法的补充。在系统分析当中,也可以采用数学分析方法对系统的功能进行描述。数学分析是一种相当精确的方法,在系统优化当中可以对系统的数学模型进行优化来获得。

流程图:流程图是最早出现的软件结构模型,而且目前也在广泛使用着,因此,对于简单的系统,可以采用流程图来设计,简单明了。流程图适合于指令不超过 5000 到 10000 条的系统。在多任务系统当中,流程图可以分别对每一个任务进行描述,可是任务之间的通讯以及瞬时表现就无法表现出来了。

结构图:结构图是一种广泛采用的、用以描述系统的软件模块的组成的方法。结构图从左边到右边表示软件执行的顺序,从上边到下边代表软件模块的细化程

度。

伪代码和程序设计语言：伪代码和程序设计语言的区别不大，都是采用一种与硬件系统底层的环境分离开的一种抽象语言来描述系统行为的方法。它以一种类似于编写代码的方式来描述系统。如 Ada 语言。采用伪代码和程序设计语言的好处就是比直接用程序设计语言来实现系统更抽象，却比别的描述方法更接近代码系统。

有限状态机：有限状态机是实时系统设计当中的一种数学模型，是一种重要的、易于描述的、应用广泛的、以描述控制特性为主的建模方法，可以应用于系统分析和系统设计的所有阶段。有限状态机由如下几个要素组成：

- 1) 一个有限状态集合  $Q$
- 2) 一个有限输入状态集合  $I$
- 3) 一个变迁函数： $\delta: Q \times I \rightarrow Q$ ，变迁函数是一个状态函数，在某一状态下，给定输入之后，根据函数转入一个新状态。 $\delta$  是一个部分函数，其定义域内某些数值可以是无定义的。有限状态机可以用图来表示，节点表示状态，如果在输入  $i$  的情况下  $Q1$  状态转变成为  $Q2$ ，则自  $Q1$  画出一条指向  $Q2$  的弧线，弧线上标注转变条件  $i$ ，表示  $\delta(Q1, i) = Q2$ 。

图 5.6 是一个简单的有限状态机。

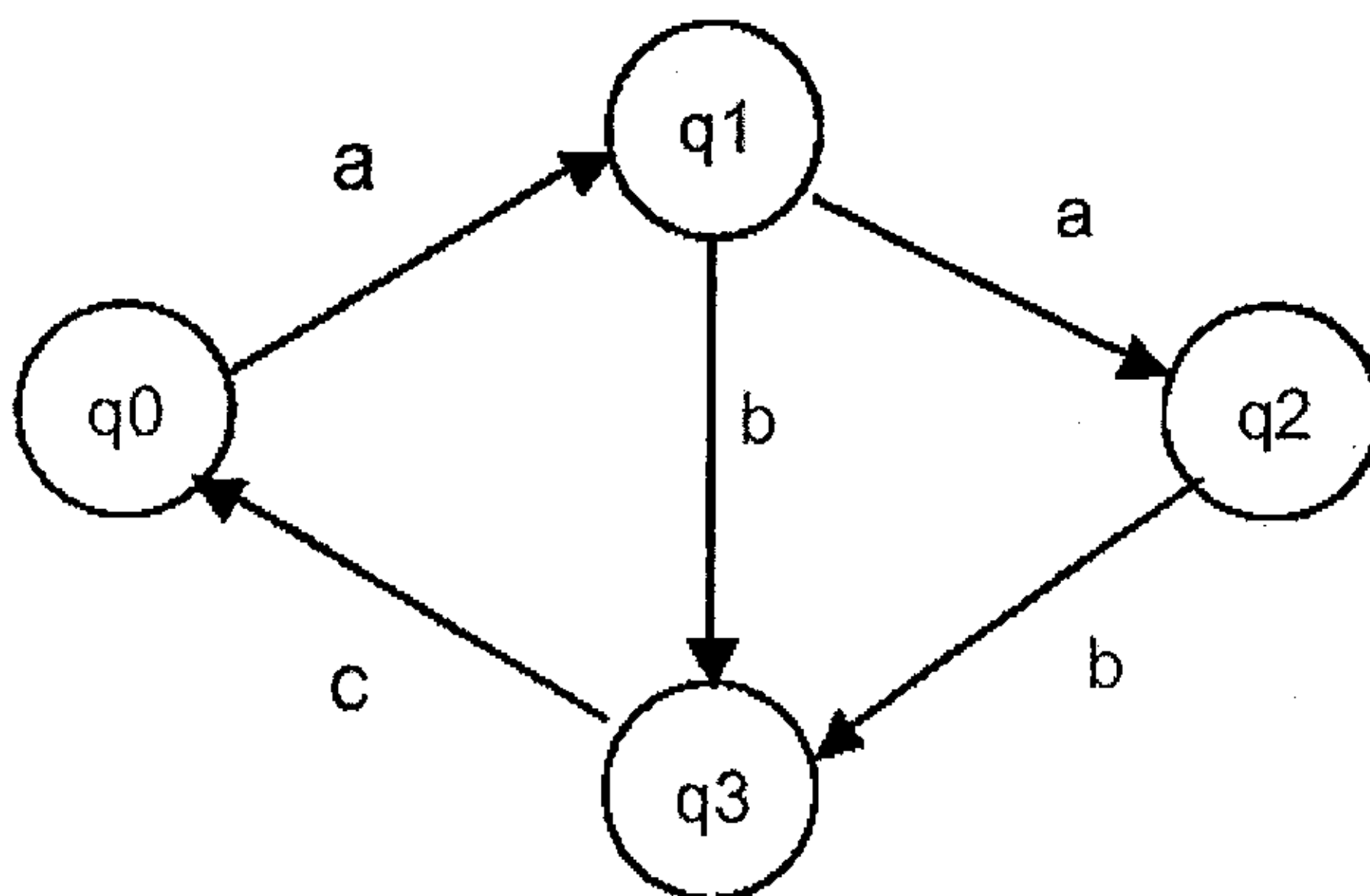


图 5.6 一个简单的有限状态机

该有限状态机的  $Q = \{q0, q1, q2, q3\}$ ，输入  $I = \{a, b, c\}$ ，状态转换的关系如图所示。有限状态机非常适合描述这样的系统：系统具有有限个状态，不同事件的发生可以用不同的状态之间的转换来模拟。

有限状态机的优点就是简单易用，状态之间的转换关系能够很容易表示

出来。但是在实际的实时系统设计当中,却有一定的局限性。其中,比较突出的就是系统在指定的时刻仅有一个状态,无法表达出系统的并发性出来。另外,在系统部件增加的时候,系统的状态数量也会增加,导致系统的复杂性显著增长。解决这些问题的新方法就是 Petri 网络理论。

**Petri 网:** Petri 网是一种使用图形方式对系统进行需求规格说明的技术,用来定义多线程、多任务系统的数学模型,易于描述系统的并发、竞争、同步等特征,并可以用于评价和改进系统。如今, Petri 网已经大量应用于包括硬件、软件和社会领域等各种系统的模型化。类似于有限状态机, Petri 网也是通过一些定义好的状态来描述事物的抽象的虚拟机。但是与有限状态机的不同, Petri 网不仅能够描述同步模型,更适合与相互独立、协同操作的处理系统。Petri 网的组成成分包括:

- (1) 一个有限的库所(place)集合,表示系统中的状态。
- (2) 一个有限的变迁(transition)集合,表示系统中的事件。
- (3) 一个有限的连续库所到变迁或者反向的有向箭头的集合,又分为输入(input)和输出(output)。Input 表示变迁/事件的前提, output 表示变迁/时间的结果。

图 5.7 是 Petri 网的一个简单实例。库所由圆圈表示,事件由矩形表示。Petri 网具有如下特性:

- (1) 状态:通过标记 Petri 网的库所来给出其状态,标记 Petri 网的库所在图形中表现为对库所茬不同数目的令牌。
- (2) 状态变化规则:一个变迁可以有多个输入或输出库所,如果一条有向箭头从库所到变迁,则该库所是改变前的一个输入库所,反之则为输出库所。如果一个变迁中每一个输入库所都至少有一个令牌,则该变迁是一个使能变迁。
- (3) 点燃:一个使能变迁可以被点燃,即从该变迁的每一个输入库所中移走一个令牌,在该变迁的每一个输出库所中增加一个令牌。

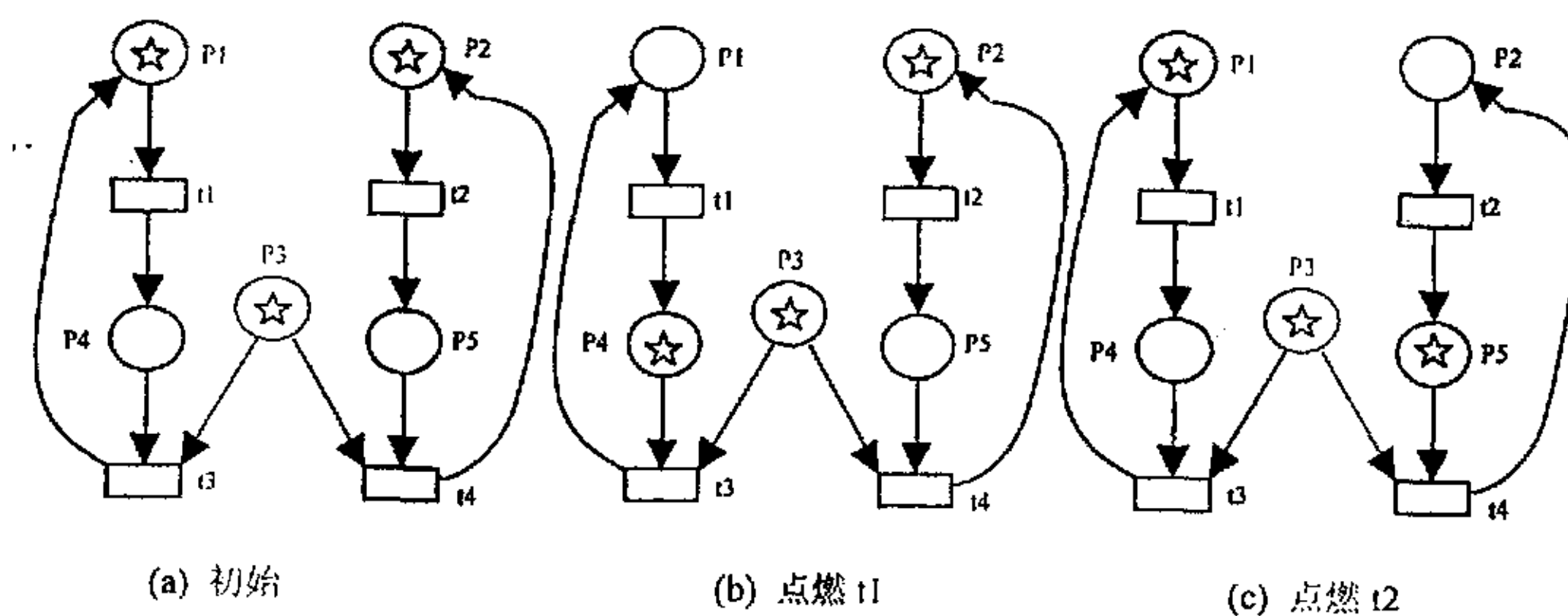


图 5.7 Petri 网的一个例子



如图 5.7 所示, 库所中的星号代表令牌。由(a)图可以看出, 变迁  $t_1$  和  $t_2$  都是使能的, (b)表示了点燃  $t_1$  的过程, (c)表示点燃  $t_2$  的过程。从这个模型可以看出, 在给定初始令牌后, Petri 网的发展过程可能是不同的。在 Petri 网中, 经常利用变迁来模拟一个事件, 而点燃则用来表示事件的发生。这样, 如果一个变迁发生的条件是满足的, 那么这个变迁就是使能的, 库所中的星号标记则说明某个条件是满足的。图中的 Petri 网可以由两部分构成:  $t_1$ 、 $t_3$  和  $t_2$ 、 $t_4$ , 它们相当于两个独立的工作流。并且共享了资源  $p_3$ 。开始的时候, 双方可以互不干涉的、异步的进行, 因为变迁  $t_1$  和  $t_2$  双方互不妨碍。点燃  $t_1$  后,  $t_3$  处于使能状态, 点燃  $t_2$  后,  $t_4$  处于使能状态。在两个变迁都点燃以后, 两个工作流都有新的变迁处于使能状态, 但是这两个变迁的点燃处于竞争状态, 两个工作流中只能有一个获得  $p_3$  的资源从而可以继续变迁。如果资源满足的情况可以解决这种冲突, 就可以设  $p_3$  的令牌有两个, 两个工作流就互不干扰, 可以并发执行了。

**数据流图:** 数据流图主要是作为软件系统建模中的结构化分析工具。建立数据流图从分析系统的功能需求开始, 分析系统中的数据流并且确定主要的函数。数据流图需要细分系统和子系统, 从而在图中对数据流有清楚的表现。控制流图是一种在系统中实现控制信号流的数据流图。这些控制信号能够描述离散的从开关或感应器得到的信号数据, 同时也能包含从离散时钟得到离散的定时信息。数据流图和控制流图结合使用, 可以提供更具体的控制信息。总体来说, 数据流图的特点是: 强调数据的流动, 不太强调控制信号的流动, 对于确定并发性很有用。实际上, 如果缺少了并发性的特点, 数据流图就是有限状态机。数据流图最主要的缺点是描述同步很困难。数据流图容易理解, 应用广泛, 通常是和一些其它方法结合使用来产生连贯的软件需求文档。

### 5.2.2.3 系统任务设计

采用现代软件工程的设计方法, 我们设计了系统的软件模块, 并且在硬件平台上进行调试, 实践证明, 基于软件工程的程序设计, 能够最大效率的提高代码的编写、调试效率, 保证系统可靠、稳定的运行。

图 5.8 是整个程序的设计框图, 任务之间通过进程通讯机制进行通讯, 任务之间的数据交换通过邮箱进行, 保证了系统的实时性。

### 5.2.2.4 网络数据传输任务设计

图 5.8 四个图是整个系统运行过程当中的主要任务。还有一些辅助任务需要辅助上面任务的运行。除了网络数据的发送和接收之外, 任务实现都是比较简单的。我们只详述网络数据的发送和接收任务的实现原理。

要实现数据的网络传输, 必须能够提供基于 TCP/IP 协议的数据包。但是在

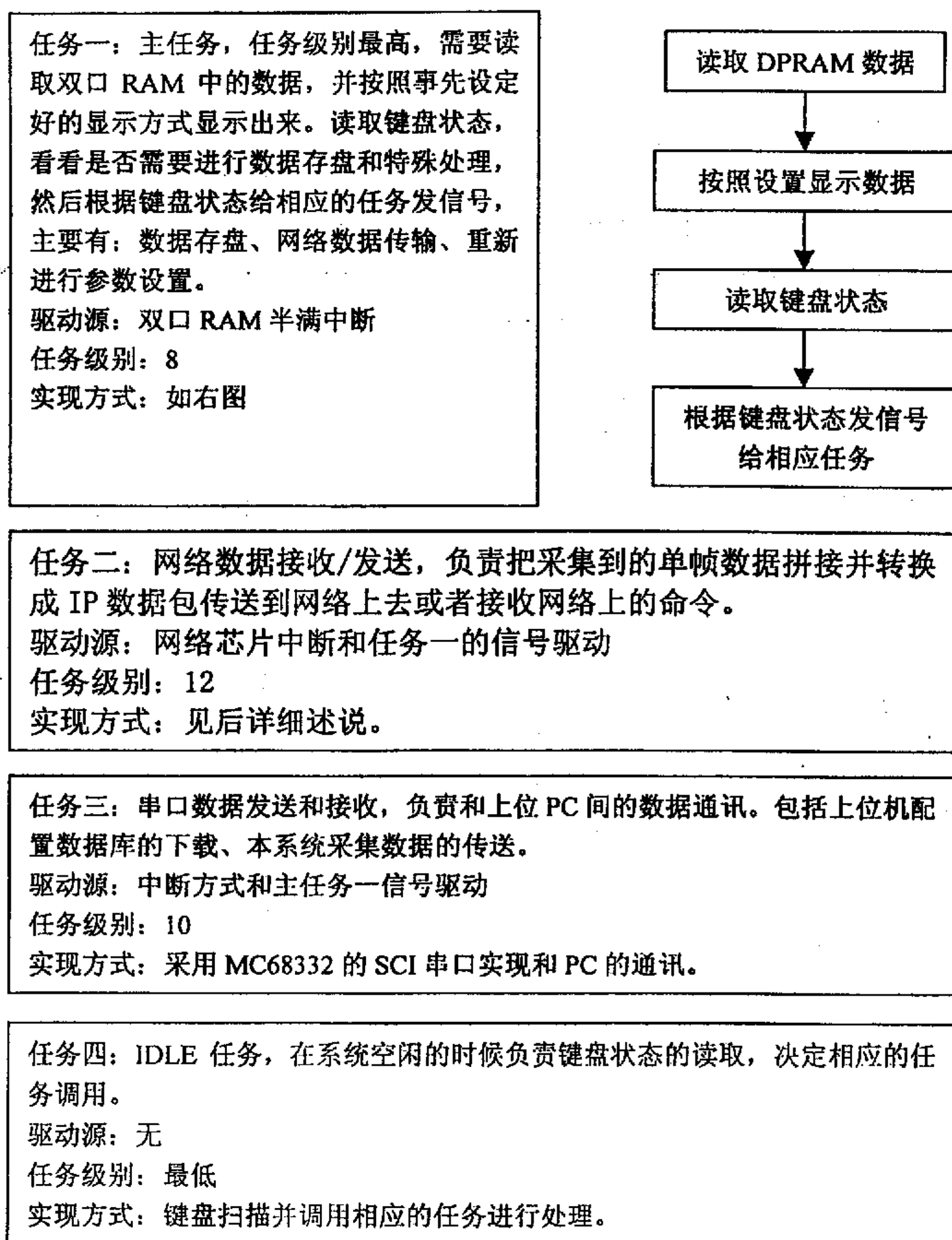


图 5.8 任务设计图

嵌入式系统当中，不但内存资源有限，并且系统的时限还相当严格，对于性能远远低于 PC 上的 PENTIUM 处理器的嵌入式处理器来说，要实现庞大的 TCP/IP 协议是不可能的，只能采用精简的 TCP/IP 来满足嵌入式系统的设计要求，因此，我们需要重新设计 TCP/IP 协议，去掉不需要的功能，如很多 TCP/IP 协议栈上提供的 SOCKET API 接口等。关于 TCP/IP 网络协议栈的详细资料请参考相应的教材 [31][32][33][34]。

在嵌入式实时操作系统  $\mu$ COS-II 上实现 TCP/IP 需要作如下几个工作：①底

层驱动的编写（如上，该过程已经完成）②上层网络协议栈的软件实现。

发送和接收数据的过程是一样的，只不过过程不一样，发送是接收过程的逆过程。因此，为了书写方便，这里只写出了数据接收的原理。数据发送只要把相应的操作方向换过来就可以了。

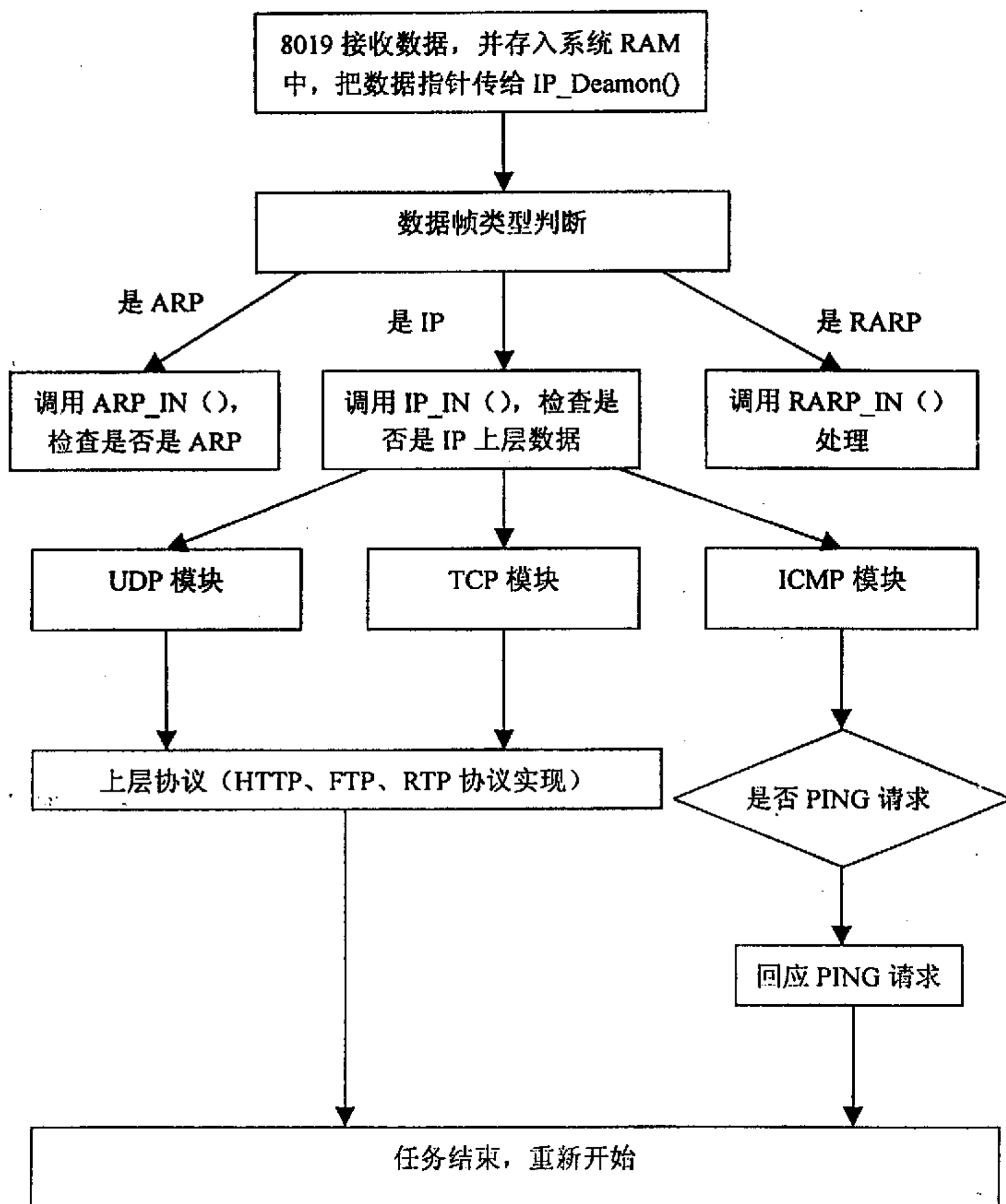


图 5.9 ARP、RARP、IP 及上层协议实现流程

### 1、底层发送接收函数的实现

上边驱动部分描述了 RTL8019 (AS) 的初始化和数据发送接收原理。在此基础上，实现三个函数：

```
void 8019_input(u8_t *, u16_t, u16_t);
```

```
void 8019_output(volatile u8_t *, u16_t, u16_t);
```

```
void ethernet_init(void);
```

初始化和数据发送接收通过这四个函数就可以实现。

## 2、ARP、RARP、IP 及上层协议实现

ARP、RARP、IP 及上层协议实现可以用图 5.9 表示。IP\_Deamon() 判断数据帧类型,然后将数据抛给上层的相应模块作进一步处理。如果仅仅是 ARP 或 RARP 请求,直接回应就可以了。如果需要进行数据包的接收,则转入 TCP 或 UDP 模块进行数据包的接收。如果需要实现上层协议,则要进一步的编写上层模块,实现命令、原始数据的读取和转发。

… 依照 TCP/IP 协议的多层模型来完成程序,每一层都实现该层的功能,这样的程序移植性较好。

## 第六章 系统调试与优化

系统设计完成后需要对系统进行调试和分析,并在此基础上进行整个系统的性能优化,从降低成本、可靠性、性能的提高、功耗、系统升级、软件代码的兼容性等方面对系统进行改进<sup>[35]</sup>。

### 6.1 信号完整性分析

信号的完整性(SI)指的是信号线上信号的质量。信号波形的破损往往不是由单一因素导致的,而是在板级设计中多种因素共同引起的。信号完整性主要包括反射(reflection)、串扰(crosstalk)、对地反射(Power/GND)等。

反射是由于信号的源端与负载端阻抗不匹配而引起的,负载会将一部分电压反射回源端。如果负载阻抗小于源端阻抗,则反射电压为负;反之,如果负载阻抗大于源端阻抗,则反射电压为正。布线的几何形状、不正确的线端接、经过连接器的传输及电源平面的不连续等因素的变化均会导致此类反射。

串扰是两条信号线之间的耦合问题,信号线之间的互电感和互电容导致了传输线上的噪声。PCB 板层参数、信号线间距、驱动端和接收端的电气特性和线端接方式对串扰都能有一定的影响。

当电路上有大的电流时会引起对地的反弹。如:大量芯片的输出级同时开启,此时有一个较大的瞬态电流在芯片与板的电源平面流过,芯片封装与电源平面间的寄生电感和电阻就会引发噪声,从而在真正的地参考电上产生电压的波动和变化,这个噪声往往会影响其它的元器件。

减小和消除反射的方法是根据传输线的特性阻抗在其发送端和接收端进行终端阻抗匹配,从而使源反射系数或负载反射系数为零。

对于线间耦合过大引起的较大的串扰,通常改用上升时间较缓的驱动源;在被动接收端端接,重新布线或检查地平面的设计。

### 6.2 电路的调试与优化

决定一个电路调试便利程度的关键在于系统各个部分的“能见度”。我们可以利用多种手段获得一个系统在不同层次上的“能见度”,包括利用仿真器,通过 JTAG 接口检验 IC,设置信号探测点检测信号波形与时序等。

JTAG 是基于 IEEE1149.1 标准的一种边界扫描测试方式。美国德州仪器公司



的大部分 DSP 产品都提供 JTAG 接口支持。结合配套的仿真软件，可以访问 DSP 的所有资源，包括片内寄存器以及所有的存储器，从而提供了一个实时的硬件仿真与调试环境，便于开发人员进行系统软件调试。

调试过程当中主要考虑点包括：

#### 1、信号探测点

为了验证板子上各个元件间的通信是否正常，应当在关键信号线上设置探测点，或者借助标准的插座将信号引出。探测点可以放在需要经常需要观察的信号线上，以及电源和地线上，这样便于采用仪器来进行测试。

#### 2、手工复位

电路中添加一个按键作为开关为系统提供硬件复位信号。这是因为在早期的设计调试过程中，系统中的多个器件可能会出现一些非法工作状态，这时，手工硬件复位是最简单的方法，将系统重新初始化，使各个部分进入缺省状态。

### 6.3 电路优化

本系统是一个简单的便携式数据采集分析系统的雏形。还有很多地方值得改进和修改的地方。下面，我们把若干设计中的问题提出来，为后续的设计提供借鉴。

#### 1、存储模块

在设计这个系统的时候，也曾经考虑过在数据采集当中的无限大容量数据存储问题。但是由于一些条件的限制未能实现，所以，只在主控模块的 FLASH 存储部件上实现了一个小容量的存储模式。

目前，有很多外存解决方案，如 PCMCIA 接口的存储卡、IDE 接口的硬盘等。这些大容量数据存储部件的接口设计比较复杂，但是使用上还是比较方便的，数据存储容量也比较大。

采用了大容量数据存储部件之后，在实时操作系统基础上进行数据文件的管理也是一个重要的方面。通常采用的方式就是在实时操作系统的基础上实现文件系统(File System)，根据文件系统对数据文件进行管理。

另外，在巡检检测当中，可以根据上位机（如 PC）的厂房机组测点数据库进行数据存盘，这个方式在 DATAPAC 中实现了。因此，如果能够实现这种自动数据存盘功能，将会大大的提高系统的灵活性。

#### 2、网络传输

目前的网络传输部分只能实现简单的功能，将来在大容量存储部件的基础上，能够实现采集数据的直接获取访问(FTP)，这样就可大大提高数据传输的速度。

#### 3、CAN 总线的应用

TMS320LF2407 内部自带了一个 CAN 总线控制器。CAN 总线能够在工厂构成低层的分布式数据采集系统，能够做到车间检测系统的网络化，随时提供上层数据获取的需要，因此，可以充分利用 DSP 内部的 CAN 总线控制器，构成分布式测控网络。

#### 4、良好的电源模块

本系统的电源采用的还是 5V 电源，其它芯片需要的电源是通过开关电源逆变实现的，因此，严格意义上来说还不能做到便携式。如果系统需要扩充的话，必须充分考虑到电源的设计。在系统的功耗降低下来以后，采用低功耗、可重复充电的电源模块，能够让系统实现真正的“便携式”。

## 第七章 数据采集分析技术相关应用实例

故障诊断系统的广泛应用使得诊断系统的硬件基础：数据采集分析部分的研究显得尤为重要。目前，故障诊断系统也象消费类电子产品的发展模式一样，要求向小型化、低功耗、方便携带等方向发展，而制约故障诊断系统的采集精度、采集速度、数据存储容量等方面的问题也在逐步得到解决。高精度的 A/D 采集芯片的出现，低功耗设计和集成度的提高，使得高速、高精度、低功耗采集成为可能。高速、大容量的存储介质也在逐步的发展，使得大容量的数据采集成为可能。

本系统是在研究生两年以来经历过的一些项目经验中积累出来的。数据采集和信号分析技术是综合了电子工程、信号分析、随机数学和人工智能等学科的原理和技术的边缘学科。同样，本系统也不是单一技术的成果。文章中采用的技术曾经在若干项目中使用过。

### 1、IDPM 综合故障诊断系统的设计

IDPM 系统是实验室自行开发的一套集数据采集、数据库文件管理、信号分析、故障诊断为一体的综合的状态监测和故障诊断系统。如图所示。如果需要

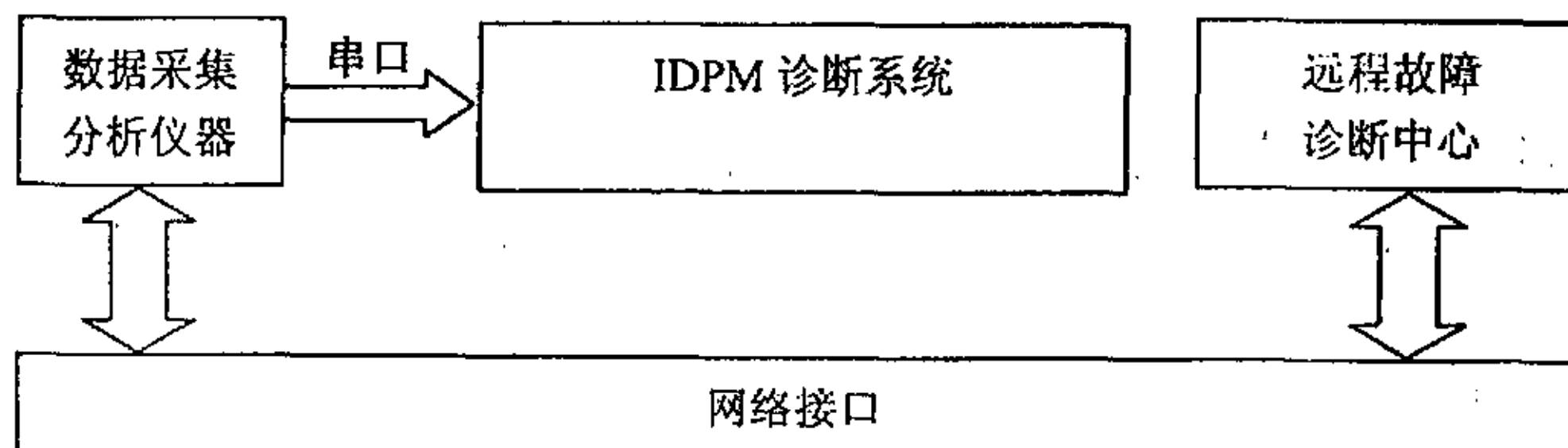


图 7.1 IDPM 系统结构图

现场数据的话，需要使用数据采集分析仪器采集现场设备的运行数据。我们设计的嵌入式动态数据采集分析仪器能够与 IDPM 系统、远程诊断中心服务器集成在一起。即能够满足强大的数据分析功能，又能够使用远程专家知识库和网络环境的优势进行远程诊断。目前，IDPM 和远程诊断中心都已经付诸使用。并且在实际的若干项目中起了关键作用。本文的前端数据采集分析系统能够替代原来数据采集依靠 ENTEK 公司的采集系统，真正实现采集、分析诊断的系统集成。

### 2、本实验室的 SD150 信号分析系统当中的采集卡的研究。性能如下：

- 方式：智能采集，板上 CPU 为 80C196
- 采集通道数：8 通道

- 样速度: 100KSPS
- 采集精度: 12 位或 16 位
- 采集综合误差: 小于  $\pm 0.5\text{LSB}$  或小于  $\pm 3\text{LSB}$
- 信号收入阻抗: 10M 欧
- 输入信号: 电压或电流
- 独立增益通道: 4 通道
- 程控放大器放大倍数: 1、2、4、8 或 1、10、100、1000
- A/D 芯片输入信号量程:  $\pm 5.0\text{V}$ , 0-10V

目前, 该套系统已经在若干高校和研究部门产生效益, 在教学、研究、生产等方面经过了严格的验证, 系统也在不断的改进。实践证明, 该套系统很适合作为动态信号测试和模态分析教学和实验的采集分析系统。

### 3、无缝钢管检测系统的研究

该系统的数据采集分析部分要求采集钢管经过传感器后的漏磁信号。为了检测出钢管的轴向(纵向)和周向(横向)方向的缺陷, 系统由横向(64 路励磁线圈传感器)和纵向(32 路励磁线圈传感器)两套检测装置组成。这样传感器路数多, 系统实时性要求高, 因此对硬件 A/D 采集和后续处理电路要求高。

本系统横向和纵向探伤检测装置共有 96 路线圈, 对此, 我们采用多路转换为一路的时分复用技术设计了基于双口 RAM 的双通道 PCI 高速采集卡(如图 7.3 所示), 满足了实时检测和高速存储采集信号功能, 卡上设有一定的存储空间, 通过端口直接读取卡上的存储器的数据, 满足高速采集, 对信号分析而言, A/D 板设计后, 提供 windows 底层驱动和相应的接口函数(VXD)供应用程序调用。

该项目去年 5 月份已经结题, 采集系统能够实现漏磁信号的监测功能。如图 7.3 是我们采集到的漏磁信号和同步时钟的关系图。

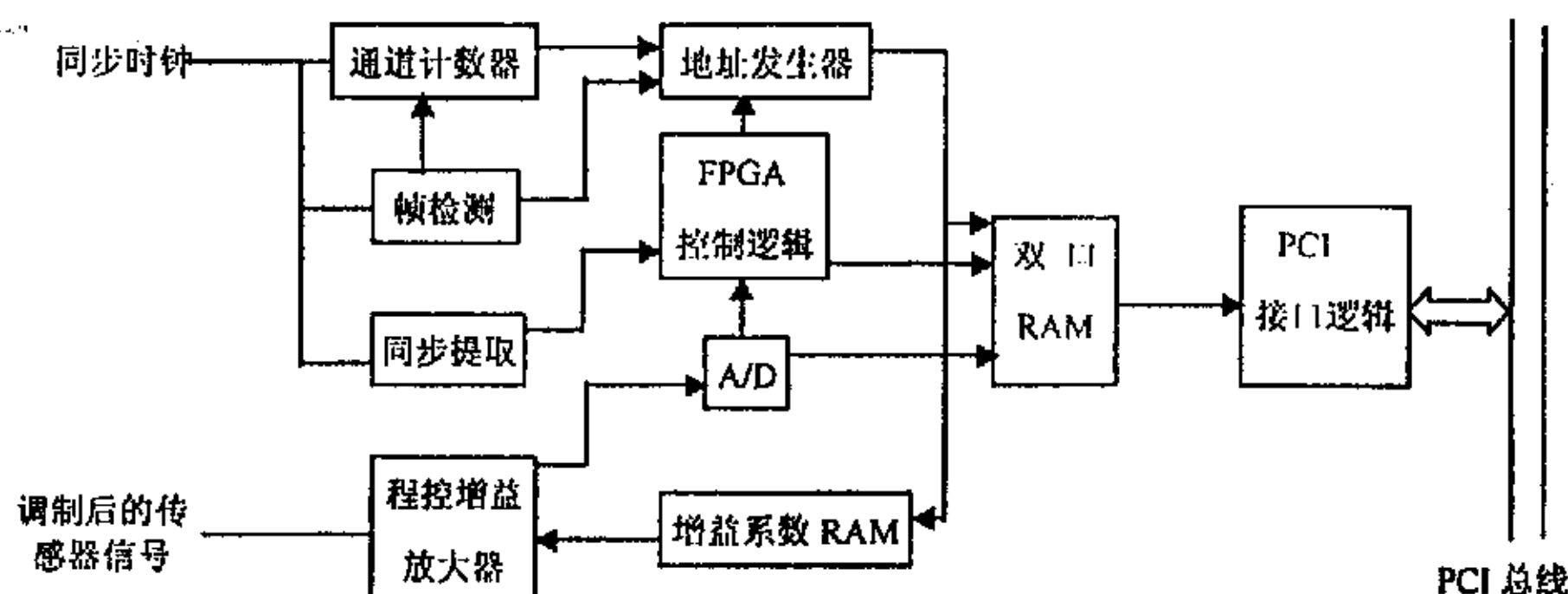


图 7.2 采集系统原理示意

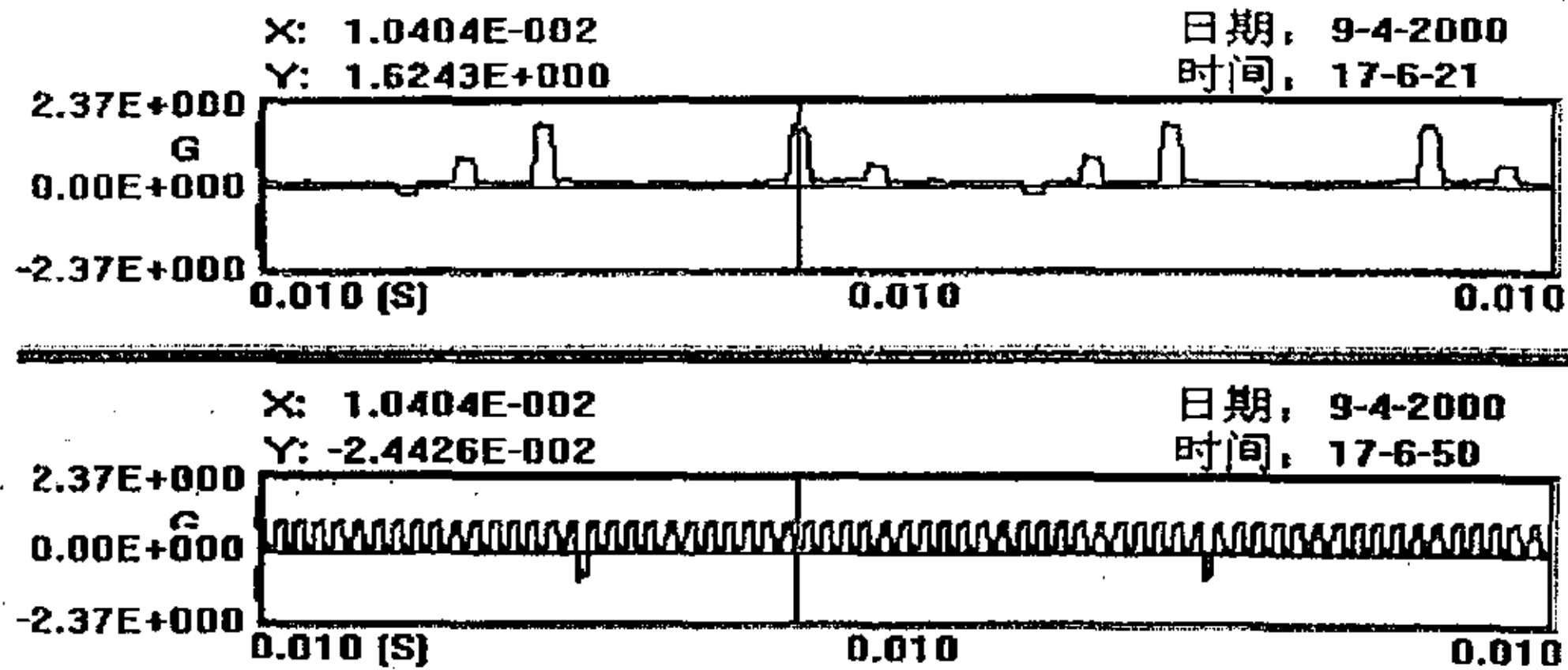


图 7.3 漏磁信号和同步时钟的关系图

#### 4、海底石油管道漏磁检测系统的研究

该系统是用于海底石油管道的管缝检测，采用的原理也是漏磁检测技术。该系统采用 PC104 工控计算机和 FPGA 采集技术结合起来，完成整个系统的设计。简单的工作流程如下，当 2MM 脉冲到来后，进行数据采集，采样的结果经过 FPGA 放到 RAM0 中，(RAM0, RAM1 是两个 1M 容量的静态存储器)，当 RAM0 中的数据存满以后，由 FPGA 控制，向 PC104 发出中断请求信号，当 PC104 接

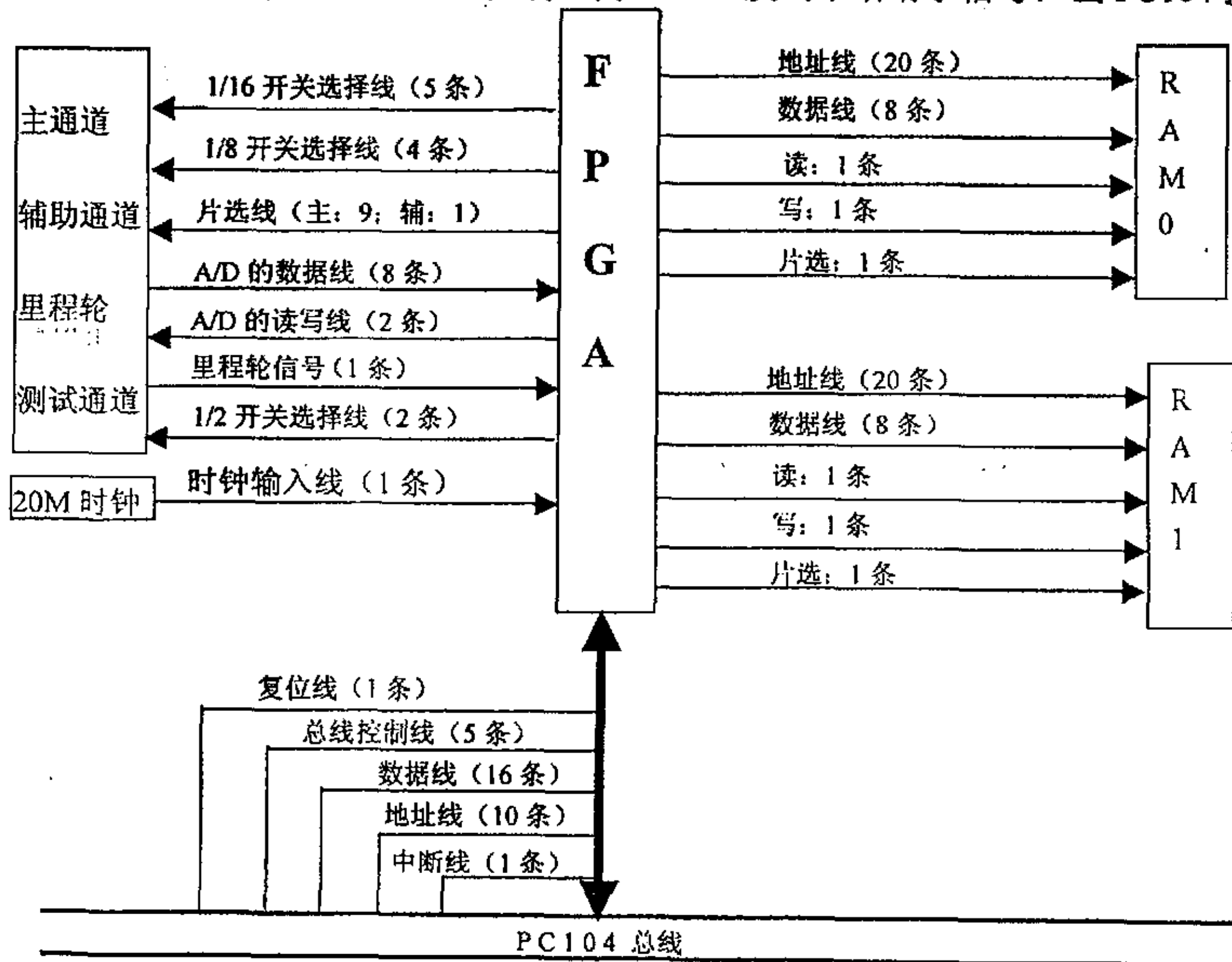


图 7.4 海底石油管道漏磁检测系统



接到中断信号以后，马上执行中断响应程序，由中断响应程序把 RAM0 中的数据搬到 PC104 的硬盘中，并且产生一个数据文件；同时，当下一个 2MM 脉冲到来后，采样的数据开始往 RAM1 中放置，因为此时 RAM0 和 PC104 正在通讯，这个切换的逻辑同样由 FPGA 来控制。当 RAM1 中的数据存满以后，又做类似的切换。所以，FPGA 是上述硬件框图的控制核心，是整个电路控制的关键。

该系统目前正在塘沽基地进行实验数据的采集，经验证，符合数据采集的和存盘、网络传输的要求。

## 结 论

以数字信号处理原理和技术为基础的先进分析方法和仪器以及离线/在线状态监测和故障诊断系统在促进生产自动化和提高劳动生产率方面取得了显著的成效。本文对采集分析系统的一种前端模块：数据采集分析仪的相关技术进行了说明。综合来说，本文是两年以来数据采集和数字信号处理技术在实际当中的应用的一个总结。结论如下：

- ①在分析了当前故障诊断学科的现状与发展趋势的基础上，提出了对故障诊断系统的关键部分：数据采集和分析系统的需求，结合当代电子信息技术的飞速发展，提出了满足便携式需求、具有动态信号采集和分析、能够实现数据存盘的系统原理框图，并在此基础上进行了技术的可行性分析。
- ②根据第一部分的需求分析设计了数据采集分析系统。首先分析了基于 DSP 数字信号处理器的设计特点以及设计方法，并在此基础上对采集部分的信号调理、数据采集、DSP 的存储模块、和主处理器的数据接口等方面进行了介绍。给出了原理说明和原理图。
- ③根据系统的需要设计了接口板。系统需要的额外功能基本都可以在接口板上实现。文中实现了基于 RTL8019(AS)的 10M 网卡与主处理器的接口设计、键盘接口设计、液晶显示接口设计等模块。
- ④详细说明了主模块的硬件、软件设计。在选择了 Motorola 的微控制器芯片 MC68332 基础上，展开了整个系统的设计。在主模块硬件基础上，移植了  $\mu$ COS-II 实时操作系统并在此基础上进行了硬件驱动程序和若干任务的开发。详细给出了嵌入式 TCP/IP 协议的实现方法，并在本系统上实现了一个小型的 TCP/IP 协议，满足了网络数据传输的要求。
- ⑤概要说明了为了满足信号完整性，在电路设计当中应该注意的问题以及若干调试的时候应该注意的问题。在现有系统基础上，提出若干值得优化的方面，为将来扩充系统提出若干设想。

## 参考文献

- [1] 屈梁生, 何正嘉, 机械故障诊断学, 上海: 上海科学技术出版社, 1987.3, 70-75
- [2] 屈梁生, 孟建, 机械故障诊断技术与当代前沿科学 (一、二、三), 设备管理和维护, 1995 年第 12 期, 1996 年第 1 期, 1996 年第 2 期
- [3] 赵新民, 刘纪元, 便携式机械设备现场动态分析系统研究, 西安交通大学学报, 1997, 031(001), 0076-0081
- [4] 贾平民, 刘玉春, 便携式数据采集与工况监测分析系统, 东南大学学报, 1997, 027(002), 0099-0102
- [5] 曹健, 哈思东, 机械加工自动线嵌入式故障诊断研究, 华中理工大学学报, 1998, 26(2), 14-16
- [6] [WWW.ENTEK.COM](http://WWW.ENTEK.COM)
- [7] 何岭松, 王峻峰, 杨叔子等, 基于因特网的设备故障远程协作诊断技术, 中国机械工程, 1999.3: 336-338
- [8] 史铁林, Internet 与现代故障诊断, 面向 21 世纪的中国振动工程研究, 1999, 8: 90-94
- [9] 黎洪生, 何岭松, 史铁林, 基于因特网远程故障诊断系统架构, 华中科技大学学报, 2000.3: 34-39
- [10] 王念旭等, DSP 基础与应用系统设计, 北京航空航天大学出版社, 2001, 200-240
- [11] 刘和平, 严利平, 张学峰等, TMS320LF2407 DSP 结构、原理及应用, 北京航空航天大学出版社, 2002 年, 1-310
- [12] Texas Instrument, TMS320F243/F241/C242 DSP Controllers Reference Guide, Chip Struture, January 2000, 10-230
- [13] Texas Instrument, TMS320F/C240 DSP Controllers Peripheral Library and Specific Device Reference Guide, January 2000
- [14] 李农, 因特网技术在嵌入式系统中的应用, 测控技术, 2000, 19 (4), 15-16
- [15] 梁庆合, 关于 Internet 网和 TCP/IP 协议的实用技术, 电子产品世界, 2000.7, 57-58
- [16] 吕京建, 嵌入式因特网技术的兴起与前景, 今日电子, 2000 (增刊)
- [17] 张宁, MC68332 单片机结构与应用, 北京: 北京航空航天大学出版社, 1996, 10-35
- [18] 齐秋群, 刚寒冰, Motorola 32 位单片机 68300 系列原理与应用, 北京理工大学出版社, 1996, 20-80

- [19] 赵永瑞, MC68300 系列微控制器的特点和应用, 电子产品世界, 2000.4: 78-80
- [20] Savitzky, Stephen R., Real-Time Microprocessor System, VanNostrand Reinhold, 1985, 50-54
- [21] Andrews Tanenbauw, 操作系统: 设计与实现(上, 下册), 电子工业出版社, 1999, 19-23
- [22] Allworth, Steven, Introduction To Real-Time Software Design, Springer-Verlag, 1981, 1-12
- [23] Jean J. Labrosse,  $\mu$ C/OS The Real-Time Kernel, R&D, 1992, 40-65
- [24] 邵贝贝译,  $\mu$ C/OS-II 源代码公开的实时嵌入式操作系统, 中国电力出版社, 2001, 44-65
- [25] 王田苗, 嵌入式系统设计与实例开发—基于 ARM 微处理器与  $\mu$ C/OS-II 实时操作系统, 北京: 清华大学出版社, 21-36
- [26] Jean J. Labross, Embeded System Building Blocks, R&D, 1992, 49-59
- [27] 蒋嗣荣, 金伯寿等, 实时多任务环境下的监控系统的实现, 工业控制计算机, 1995.6: 77-81
- [28] 陈丽蓉, 熊光泽, 雷航等, 嵌入式软件系统的实时性设计, 单片机与嵌入式系统应用, 2000.3, 18-22
- [29] 郭兵, 熊光泽, 嵌入式应用软件开发环境的构造, 计算机应用, 2000, 20(7), 7-9
- [30] Michael Barr, C/C++嵌入式系统编程(于志红译), 北京: 中国电力出版社, 2001.3, 9-18
- [31] Andrew S. Tanenbaum, 计算机网络, 清华大学出版社, 1998, 200-400
- [32] Dauglas E. Comer, David L. Stevens, 用 TCP/IP 进行网际互连, 第一卷: 原理、协议和体系结构, 电子工业出版社, 1998
- [33] Dauglas E. Comer, David L. Stevens, 用 TCP/IP 进行网际互连, 第二卷: 设计、实现和内部构成, 电子工业出版社, 1998
- [34] Dauglas E. Comer, David L. Stevens, 用 TCP/IP 进行网际互连, 第三卷: 客户机-服务器编程和应用, 电子工业出版社, 1998
- [35] 李挺, 朱金刚, 机电一体化系统的抗干扰措施, 机电工程, 2000.6: 5-7
- [36] Dalpiaz G, Rivola A. Condition monitoring and diagnostics in automatic machines: comparison of vibration analysis techniques. Mechanical Systems and Signal processing, 1997, 11(1): 53-73
- [37] Vinoski S. CORBA Integrating Diverse Applications within Distributed Heterogeneous Environments. IEEE Communication Magazine. 1997. 35(2): 45-55
- [38] JAY Lee. Modern Computer-aided Maintenance of Manufacturing Equipment

- and Systems: Review and Perspective. Computers ind. Engng. Vol.28, No.4, 1995, 793-811
- [39] J.Tarn,M.Tomizuka,On-line monitoring of tool and cutting condition in milling , Journal of Engineering for Industry III (1989) 206-212.
- [40] Y.S.Tarn,On-line detection of tool breakage in millling, Mechanical Systems and Signal Processing 5 (5) (1991) 389-401.



## 攻读硕士期间发表论文和参加科研情况

### 发表论文:

1. 王太勇, 薛国光, 蒋奇, 李国威, 钢管表面缺陷在线计算机检测, 噪声与振动控制 (已录用, 安排在 2002 年 12 月第 6 期)
2. 王太勇, 蒋奇, 薛国光, 钢管漏磁在线检测技术的研究, 计量学报 (已录用, 安排在 2002 年 10 月第 4 期)
3. 李国威, 王太勇, 薛国光等, 设备远程检测诊断系统设计与实现, 天津理工大学学报 (已录用, 安排在 2003 年第 1 期)
4. 李宏伟, 王太勇, 薛国光, 李波, 林建波, 基于嵌入式微控制器核心和层次化网络监控机制的开放结构数控系统的构建, 振动与冲击 (已录用, 安排在 2003 年第一期)

### 参加科研情况:

1. 参与 SD150 数据采集分析系统 ISA、PCI、USB 采集平台的开发。
2. 参与无缝钢管探伤理论及在线诊断与智能预测技术项目, 天津市科委资助的天津市重点基金项目, 项目编号: 993802411
3. 参与美国 NDT 检测设备计算机改造项目, 资助单位: 天津钢管公司。
4. 参与管道智能检测设备及智能数据分析系统的研究项目 (国家 863 科技攻关项目), 资助单位: 深圳巨涛公司
5. 参与研究天津市科委重大项目攻关“基于网络 and 状态监测的层次化嵌入式开放结构数控系统”。

## 致 谢

本论文是在王太勇导师的亲切关怀和悉心指导下完成的，论文的每一步进展都凝聚着导师的心血。导师渊博的知识、严谨的治学态度、求实的工作作风、平易近人的待人方式使我终生受益，必将激励我在科学的道路上奋力拼搏，在此向王教授表示衷心的感谢！

感谢动态测控实验室的每一位成员，能够在这样的环境中工作，感觉终身难忘。特别感谢刘增强博士、倪海洋硕士，本课题的很多工作他们都作了很好的铺垫。感谢邓学欣博士、李波硕士、陈海鹏硕士，本文提到的很多工作都是他们帮助的结果。感谢李宏伟博士、商同博士、蒋奇博士、尚志武博士、文松博士、秦旭达博士后、王国峰博士后，李国威硕士，能够和你们共事是我终身的荣幸。

特别感谢实验室和课题组的其他工作人员。

我还要感谢我的父母，是他们的鼓励和教导让我不断学习，不断进步。感谢我的女友，多谢你的笑容让我能够全身心的投入。感谢所有支持、帮助过我的朋友，并向他们致以最诚挚的谢意。

感谢各位专家教授在百忙之中对我的论文进行指导。

薛国光

2002.12 于天津大学