

摘 要

目前矢量水听器信号处理技术的研究已经成为一个热门课题，倍受水声、鱼雷和水雷界的关注。本论文的研究内容为矢量阵接收系统的硬件设计。该系统是针对矢量水听器设计的一套数据采集装置，主要功能是信息的获取、存储，系统研制的主要目的是利用获取的数据检验各种算法的性能，从而进一步优化算法。因此要求矢量阵接收系统能够提供高质量的水声数据。

论文的主要工作为：针对该实验装置的任务和需求进行了分析，提出了由水下分机和水上分机两分部组成数据采集系统的实验装置解决方案。本装置共有 24 个数据采集通道，可以同时接收来自 6 个矢量水听器的数据，并且可以通过光纤对每个水听器单独进行远程控制，配置其接收增益和滤波器带宽。

水下分机由预处理电路、采集控制、存储控制和电源管理电路组成，通过电池供电，各模块采用低功耗设计并配置有独立的电源开关。采集控制和存储控制，以数字信号处理器（DSP）和现场可编程门阵列（FPGA）为核心，简化了电路的设计难度并为今后的功能扩展提供了方便。

水上分机包括数据接收卡和综合显控程序两部分，完成数据储存、状态监控、命令发送等功能。

本系统已经调试成功，并在附录中提供了本系统的主要性能指标。

关键词：数据采集；矢量水听器；DSP；FPGA；

ABSTRACT

Vector sensor array is a hotspot technique in the field of underwater acoustic torpedo and underwater torpedo. This dissertation aims at the design of receiver system of vector sensor array. This system is a data collection device correspond with vector sensor array, its main task is the information acquisition and storage. In order to verify vector algorithm using the data collected, we need a receiver system of vector sensor array to gather pure underwater acoustic.

In terms of the system's task and requirement, an experimental resolving project is put forward, which includes a buoy system and a station system. The experiment equipment has 24 gathering data channels and can sample the signals from six transducers simultaneously. Every channel can be controlled by optical fiber in distance, through which can alter the plus and filter band of the receiver.

The buoy system consists of signal conditional circuit, sample control board, data storage board and power supply board. Because the buoy system is powered by battery, its power dissipation is a serious problem and every module have been tried to reduce power dissipation in the paper. Digital Signal Processor (DSP) and Field Program Gate Array (FPGA) have been used in the circuit, which made the hardware smaller, decreased the power consumption and enhanced the system reliability.

The station system comprises signal receiver board and monitor program, its task is to storage data, control status and send instruction.

The device has been debugged successfully and its main performances have been provided in the appendix.

Key Word : data collection, vector sensor array, DSP, FPGA

哈尔滨工程大学

学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出，并与参考文献相对应。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出贡献的个人和集体，均已在文中以明确的方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）：杨鹤

日 期：2004年 3 月 2 日

第 1 章 绪论

1.1 引言

在现代社会中，信号处理已经渗透到人们生活的各个领域，数字信号处理作为信号处理的主要方法得到了长足的发展。如今数字信号处理已经广泛应用到雷达、声呐、图像处理、通信等领域中。采样定理是数字信号处理理论的基础，正是因为依据采样定理将模拟信号数字化后，可以不丢失其信息，使信号处理数字化成为可能。所以模拟信号的采集、量化在整个数字信号处理系统中就显得尤为重要，数据采集器的指标直接影响到信号处理系统的性能。

在我们所熟悉的水声领域中，数字信号处理已经成为了必不可少的重要手段，并逐渐发展成为一门专门的学科——水声信号处理，信号采集也具有至关重要的作用，水声信号数据采集系统形成了它自身的特点。一般来说，声呐均采用多元阵，基阵输出信号的路数较多，这就要求采用多通道采集系统。一般水声信号的频率较低，但通道多采集时间长，这就要求有较大容量的数据储存器。水听器输出的信号十分微弱，在采集量化前要对信号进行一些处理工作，如放大、滤波等。在放大、滤波的过程中信号很容易受到电噪声、数字电路的影响，使信号相位发生变化。为了保证各通道幅度和相位的一致性，对模拟接收机提出了很高的要求。

1.2 矢量阵接收系统

1.2.1 矢量阵接收系统简介

“十五”海军装备预研项目“潜用拖曳线列阵声呐信号处理技术研究”

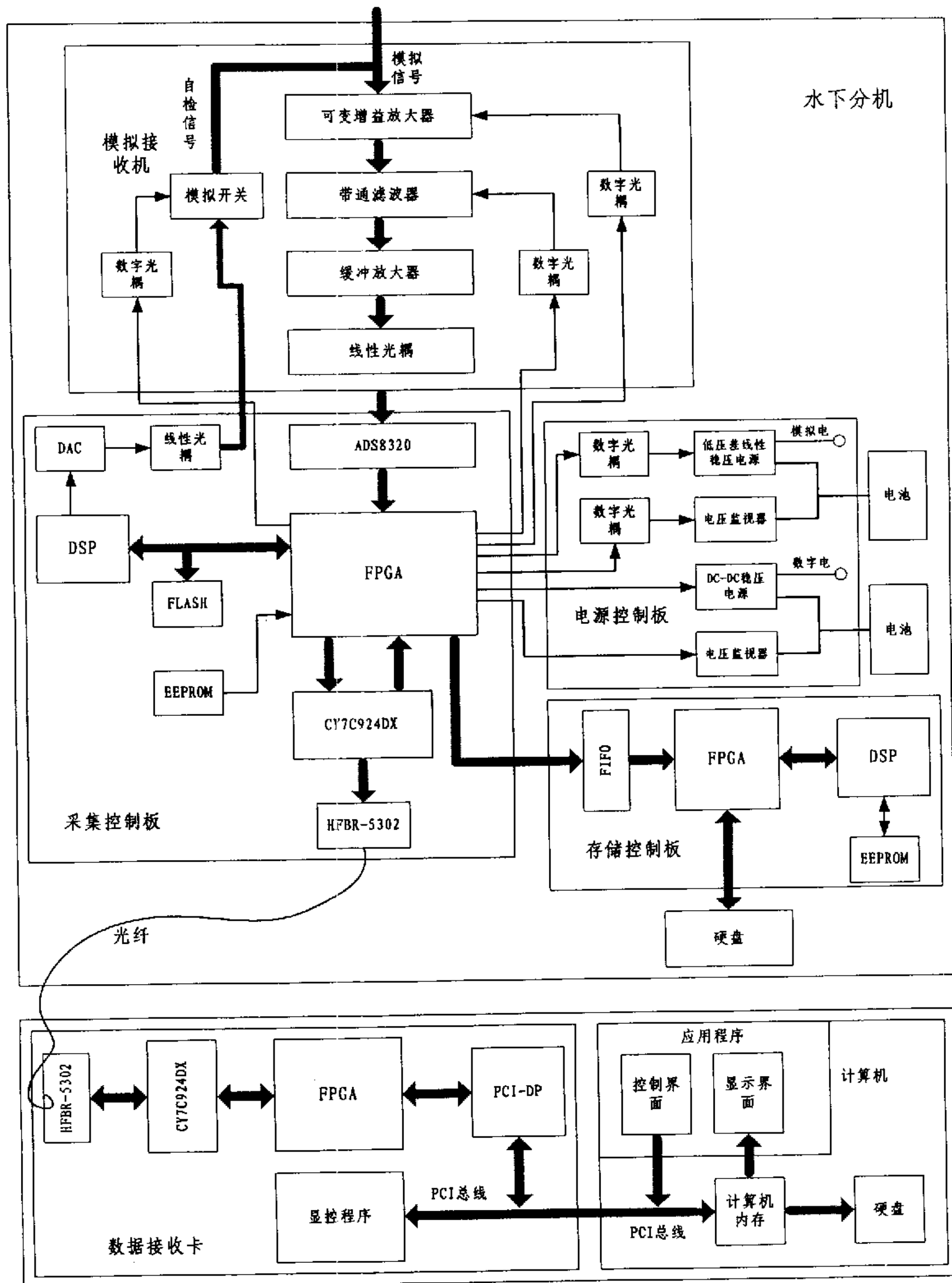


图 1.1 矢量阵接收系统原理框图

的主要内容为矢量水听器处理技术的研究。目前矢量水听器信号处理技术的研究已经成为一个热门课题，倍受水声、鱼雷和水雷界的关注。矢量水听器可在声场空间一个点接收到两个物理量：标量（声压）、矢量（质点振速在三个垂直坐标上的投影），加之单个矢量水听器在空间具有偶极子指向性，这样单个矢量水听器获取的信息量要比单个水听器获取的信息量丰富得多。因此研究矢量水听器阵处理的优势是显而易见的。

该项目研究内容主要分为二部分，一部分为矢量水听器阵的理论及计算机仿真研究，另一部分为实验装置的硬件设计。本论文的研究内容就是实验装置的硬件系统设计。实验装置的主要功能是信息的获取、存储，系统研制的主要目的是利用获取的数据检验各种算法的性能，从而进一步优化算法。因此要求矢量阵接收系统能够提供高质量的水声数据。

1.2.2 系统主要功能

矢量阵接收系统分为水下接收分系统和计算机分系统两部分。水下接收分系统完成模拟信号的放大、滤波、增益控制、采集量化和高速储存；计算机分系统，负责实现信号采集控制、采集数据的储存和显示监控以及后续处理。

1.2.2.1 水下接收分系统的主要功能

水下接收分系统的电子设备位于电子仓中，包括接收放大、滤波、控制、内记和电源管理等，电源由电池组供给。其主要设计思想如下：

- 选用低功耗、单电源器件
- 模拟电路与数字电路具有隔离电路
- 接收系统的放大量应使接上换能器后电噪声级位于 16bit 的 3~4bit
- 带有自动增益控制功能，每个矢量传感器可单独控制
- 接收通道为 24 路，即最多扩展为 6 元阵。其中每 4 路为一组，对应一个矢量水听器
- 滤波器频带可调，总工作频段为 50Hz~25KHz
- 具有自检功能
- 电源状态可控

- 传输为双工方式
- 使用光纤传输数据，以保证数据传输的可靠性

1.2.2.2 计算机分系统的主要功能

计算机分系统是测量船上或基站上的主要设备，包括数据接收卡和综合显控程序两部分，完成数据储存、对接收分系统的控制。计算机分系统具有如下功能：

- 实时储存水下传输的数据
- 标准光纤输入（采用 cypress 公司的 Hotlink 接口标准）
- 具有监视数据、发送命令等功能

1.3 论文主要研究内容

本论文的主要研究内容是面向矢量水听器阵列（6 元阵），设计并实现一套多通道数据采集系统（共 24 通道）。包括水下分机，负责对基阵输出的模拟信号放大、滤波、增益控制、采集量化和储存；水上分机，负责实现信号采集控制、采集数据的储存和显示以及后续处理；光纤传输系统，负责水下分机和水上分机的通信工作。

论文第一章讲述了设计矢量阵接收系统的意义，并给出了采集系统的主要技术指标和本论文的主要研究内容。论文第二章论述了水下分机的设计与实现的要点，确定了 24 通道数据采集系统的主要结构、数据储存和电源管理的实现方案。着重讲述了 24 路串行 AD 的采集控制和数据的串并转换问题，并讨论了电路中数字电路和模拟电路的隔离以及信号增益控制的方法。为实现采集系统的内记功能，论文提出了使用 DSP 和 FPGA 实现硬盘接口和文件管理的解决方案。同时对数据的光纤传输系统的设计进行了说明。论文第三章主要针对水上分机使用的数据接收卡和显控程序进行设计，论述了基于 PCI 总线的数据接收卡和设备驱动程序以及应用程序在设计中应注意的问题。

第 2 章 水下接收系统设计

2.1 概述

水下接收系统由五部分组成：模拟信号接收板、采集控制板、存储控制板、电源管理板和底板。模拟信号接收板完成对模拟信号的放大、滤波；采集控制板实现数据采集、打包、与上位机通讯和控制电子仓中各设备等复杂功能；储存控制板完成对采集数据的接收和存盘；电源管理板对模拟和数字部分供电并实时监控电池状态。

2.2 预处理电路

在采集系统中，通常将输入模拟信号先通过预处理电路（Signal Conditioning Circuit）然后再进行采样和量化。信号预处理的目的是完成缓冲、放大、衰减、隔离和线性化传感器信号等操作^[16]。广义的说，人们常将数据采集系统中 ADC 之前的电路统称为预处理电路。本系统信号预处理电路由衰减器、放大器、滤波器、光耦隔离和模拟开关组成。系统开始工作时首先由采集控制板输出自检信号，经过模拟开关 MAX4678（平时为低，即 SWITCH OFF），在模拟输入端输入作为系统自检，自检完成后由上位机配置衰减器和滤波器，系统开始工作。

其框图如下：

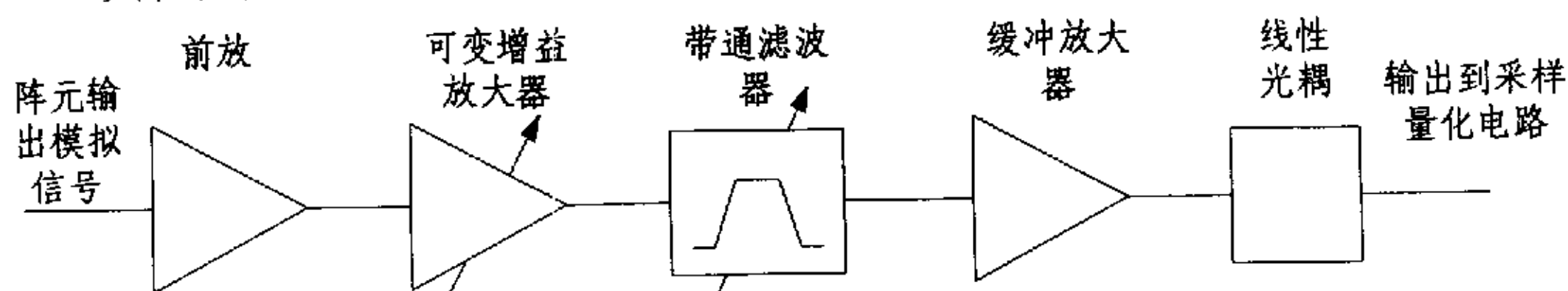


图 2.1 模拟接收机原理框图

模拟接收机的性能及指标：

- 滤波器通带频率（0Hz~25kHz）可调
 - 高通滤波器分为4级（0Hz、50Hz、1kHz、10kHz）
 - 低通滤波器任意可调（500Hz~25kHz）
- 固定放大增益 46dB
- 接收机电压输出 0~5Vp-p

2.2.1 滤波与衰减器

带通滤波器由一个低通滤波器和一个高通滤波器构建。系统最宽通带 0Hz~25kHz，可调滤波器由低通和高通滤波器组成，低通滤波器的最大截止频率 25kHz，可调到 500Hz；高通滤波器的最大截止频率 10kHz，可调至 0Hz。

其框图如下：

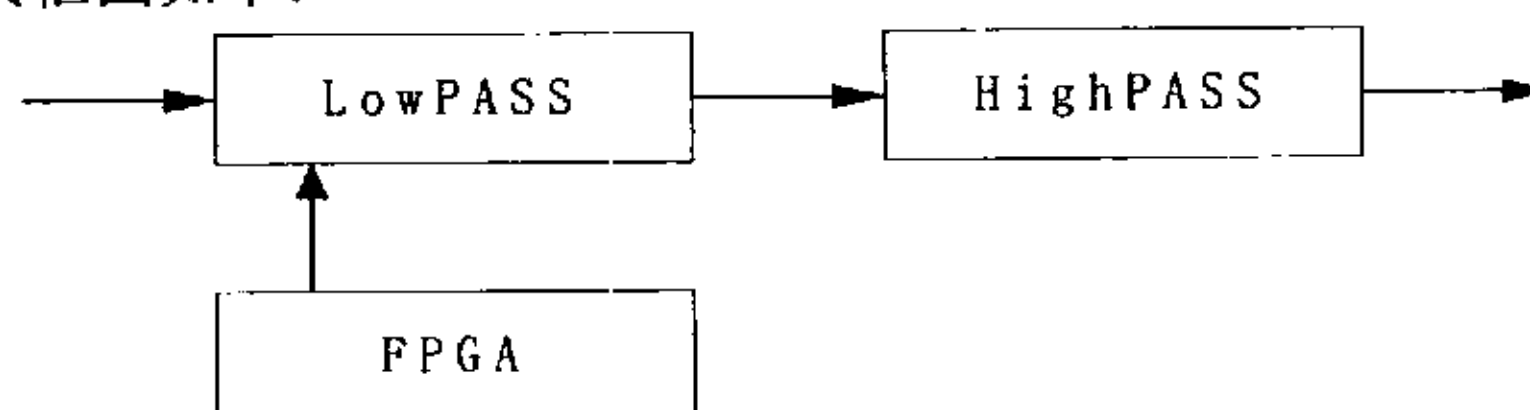


图 2.2 滤波器结构框图

高通滤波器配置电阻通过 Jumper 形式调整截止频率，截止频率分为 4 档(0Hz、50Hz、1kHz、10kHz)。

低通滤波器由数字电位器（50Ω~100kΩ）调整，截止频率 500Hz~25kHz 可调。数字电位器选用 AD8400AR100，其主要性能如下：

$$R_{WB}(D_x) = (D_x)/256 \times R_{BA} + R_w \quad R_w = 50\Omega$$

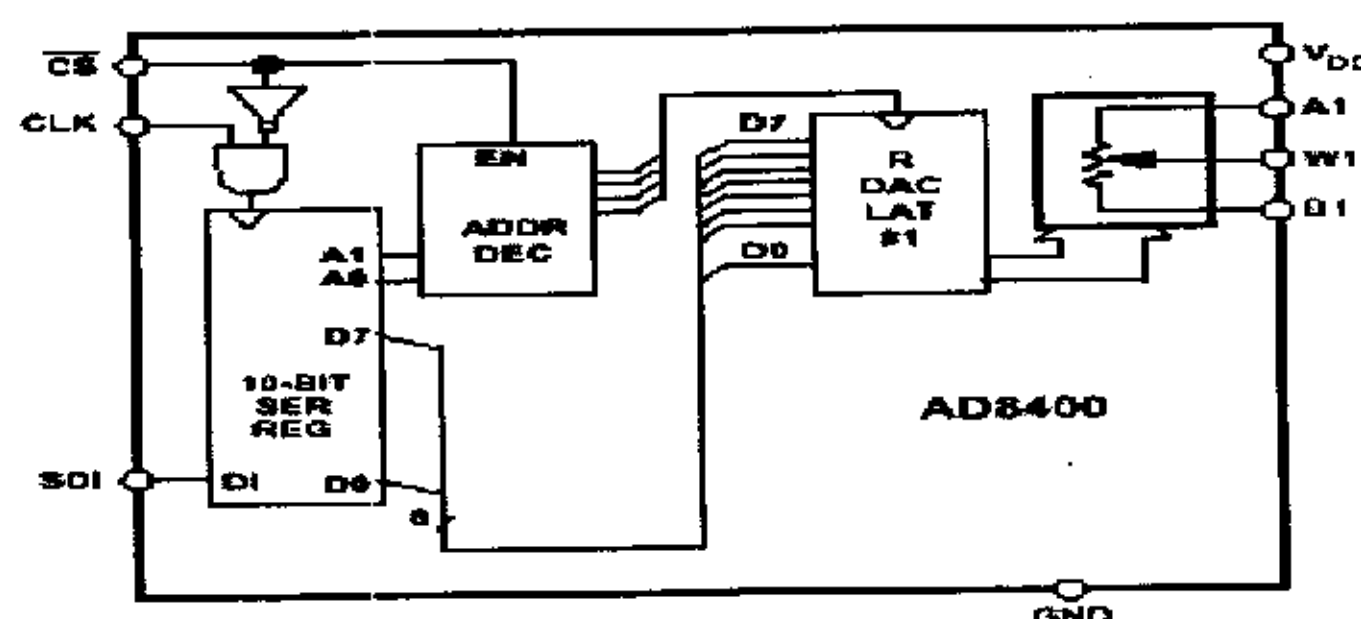


图 2.3 数字电位器结构框图

NOTE:

CLK: 输入最大频率 50MHz

SDI 输入数据格式:

ADDR		DATA							
B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0	0	D7	D6	D5	D4	D3	D2	D1	D0

调整放大量的工作逻辑与滤波器配置类似, 每 4 个通道单独控制, 由一组 CS、CLK、SDI 串行控制线控制。通过调整第一级放大器的放大量来控制整个接收机的增益, 其中数字电位器也选用 AD8400AR100。

其框图如下:

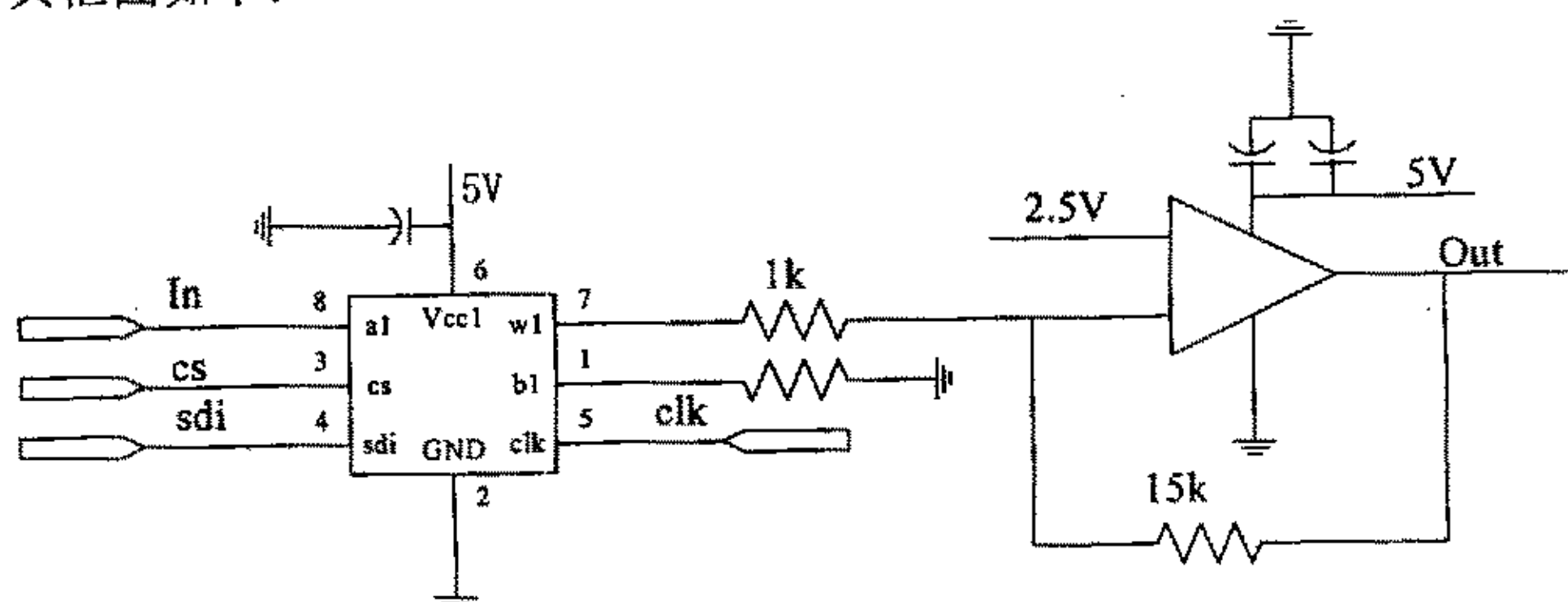


图 2.4 增益可控放大电路原理图

2.2.2 光耦隔离

为了提高接收机抗干扰能力, 避免模拟信号与数字信号相互干扰, 选用光电隔离芯片将模拟部分与数字部分隔离。

在接收板上数字电位器的 CLK、CS 和 SDI 以及模拟开关的 CS、输入的自检信号和接收机的输出, 都会受到数字电路产生的高频干扰的影响, 从而影响整个系统的性能。隔离接收板上的数字信号选用 AGILENT 公司的 HCPL-0708, 最高转换率达 15MBd, 它兼容 CMOS 电平, 输出的电流足以驱动数字电位器, 不用加线驱动器。

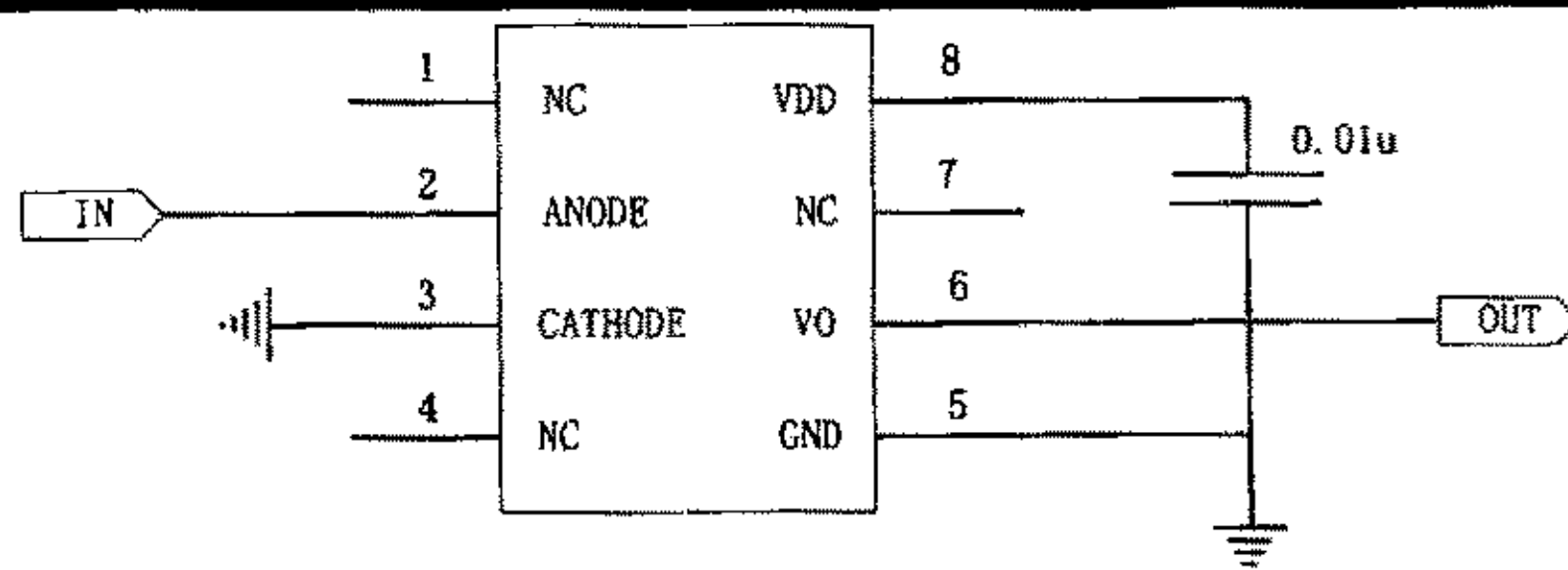


图 2.5 数字光耦原理图

模拟信号经滤波、防大后，通过线性光耦送给 A/D 器件。线性光电耦合器件由一个发光二极管 LED 和两个光敏二极管组成，输入信号激发 LED 发光，输入光敏二极管监控和稳定 LED 的光源，消除 LED 中的非线性和漂移，输出光敏二极管感应产生光敏电流，从而在输入和输出进行了有效的隔离。线性光耦采用 SIEMENS 公司的 IL388^[30]，其信号失真度小于-80dB 满足系统的要求。下面给出一个简单的原理图：

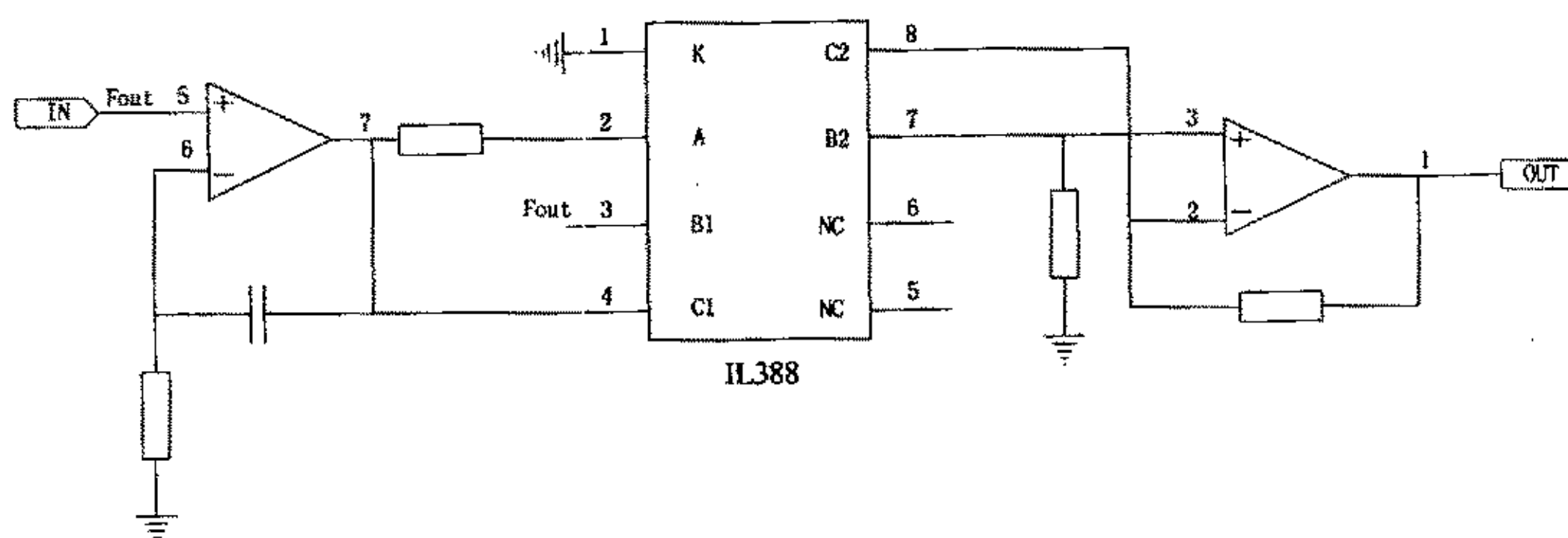


图 2.6 线性光耦隔离电路原理图

2.3 采集控制的设计与实现

采集控制板是水下部分的中枢，通过接收上位机命令完成对系统各个部分的控制，实现系统自检、信号采集、数据存储等功能。为降低系统功耗，采集器有自检、工作和待机三种模式。当工作在待机模式时，由采集控制板值班，通过光纤接收显控台下传的命令，依次将系统各部分唤醒并开始工作。

2.3.1 器件选型

2.3.1.1 DAC 简介

系统中 DAC 选用的是 BURR-BROWN 公司的 DAC7611, 它是 12 位串行输入数模转换器。固定的 5V 电源, 而功耗只有 2.5mW。它内部包括有串并转换移位寄存器、DAC 寄存器、2.435V 片内参考电压、12 位的 DAC 和高速的轨到轨 (rail-to-rail) 输出放大器等几部分, 详见图 2.8。

图中, 大方框之外的“O”代表 DAC7611 的 8 个管脚。其中, 时钟 CLK, 串行数据输入 SDI 和装载选通 $\overline{LD}$ 三者组成串行接口; 片选 $\overline{CS}$ 和异步清零 $\overline{CLR}$ 是两个控制管脚; V_{OUT} 为模拟电压输出脚。12 位数字输入 000h~0FFFh 对应 0V~4.095V 的模拟输出。

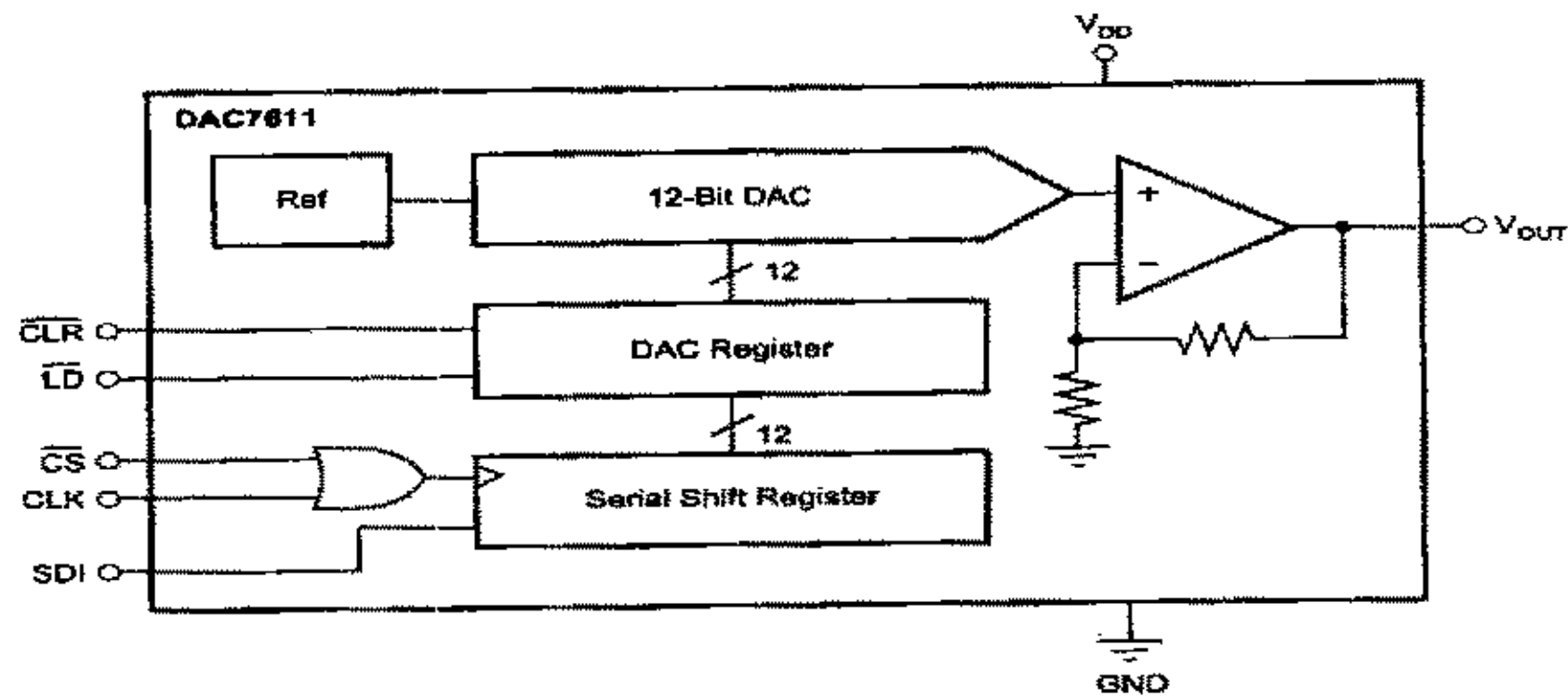


图 2.7 DAC7611 结构框图

进入 DAC7611 的数据是双缓冲的。SDI 脚上的数据在 CLK 的同步下, 从高位 (MSB) 开始移入 12 位的串行移位寄存器, 而 $\overline{LD}$ 由高到低就把数据转送到 DAC 寄存器, 其代表的模拟值立刻在 V_{OUT} 脚反映出来。这一过程由图 2.9 可以清楚地看出来。

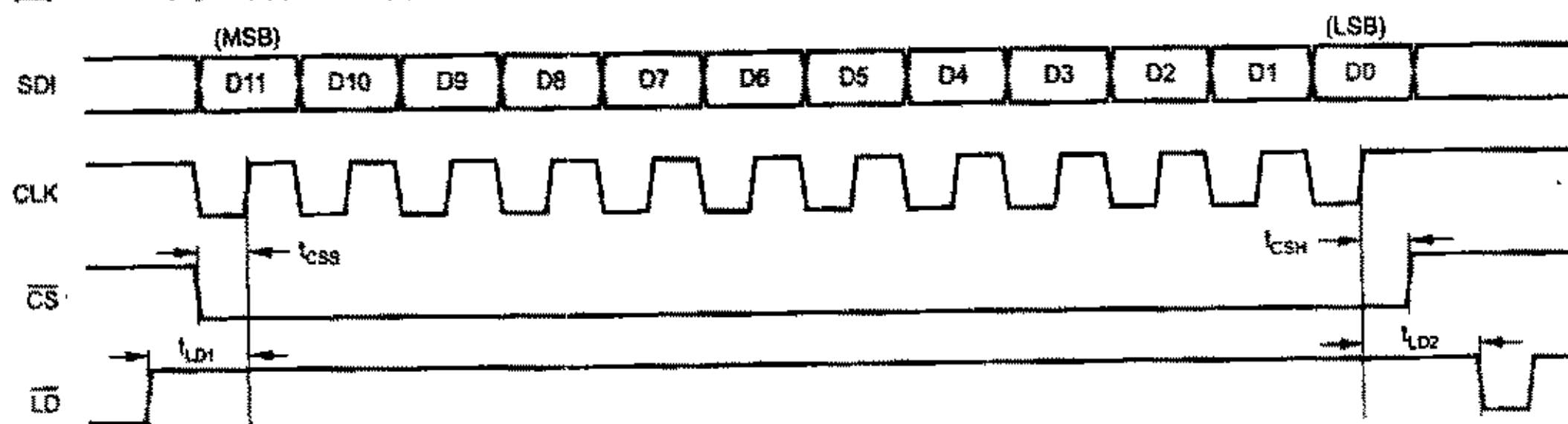


图 2.8 DAC7611 时序图

2.3.1.2 ADC 简介

系统中 ADC 选用的是 BURR-BROWN 公司的 ADS8320, 它是 16 位、高速、低功耗模数 (A/D) 转换器, 电源为 2.7V~5.25V 可选。当工作在 2.7V 电源和 100kHz 的最高采样率下, 它的功耗只有 1.8mW。它提供一个同步串行接口 (与 SPI/SSI 兼容), 其参考电压的范围为: 0.5V~V_{CC}。

外部时钟通过 DCLOCK 脚送入 ADS8320, 其频率范围为: 24kHz~2.4MHz, 而采样周期即一次转换时间为:

$$T = 24 / f_{DCLOCK} \quad ()$$

式中: f_{DCLOCK} —— 移位时钟频率, Hz。

因子“24”的存在是因为假设一次完整的采样周期为 24 个 DCLOCK 时钟周期, 参见图。两个输入脚 +In 和 -In 允许差分输入。采样后的结果 (数字输出) 在 DCLOCK 输出的同步下, 由高位开始串行移出 D_{out} 脚, 输出 0000h~0FFFFh, 代表电压 0V~V_{CC}⁽¹⁾。

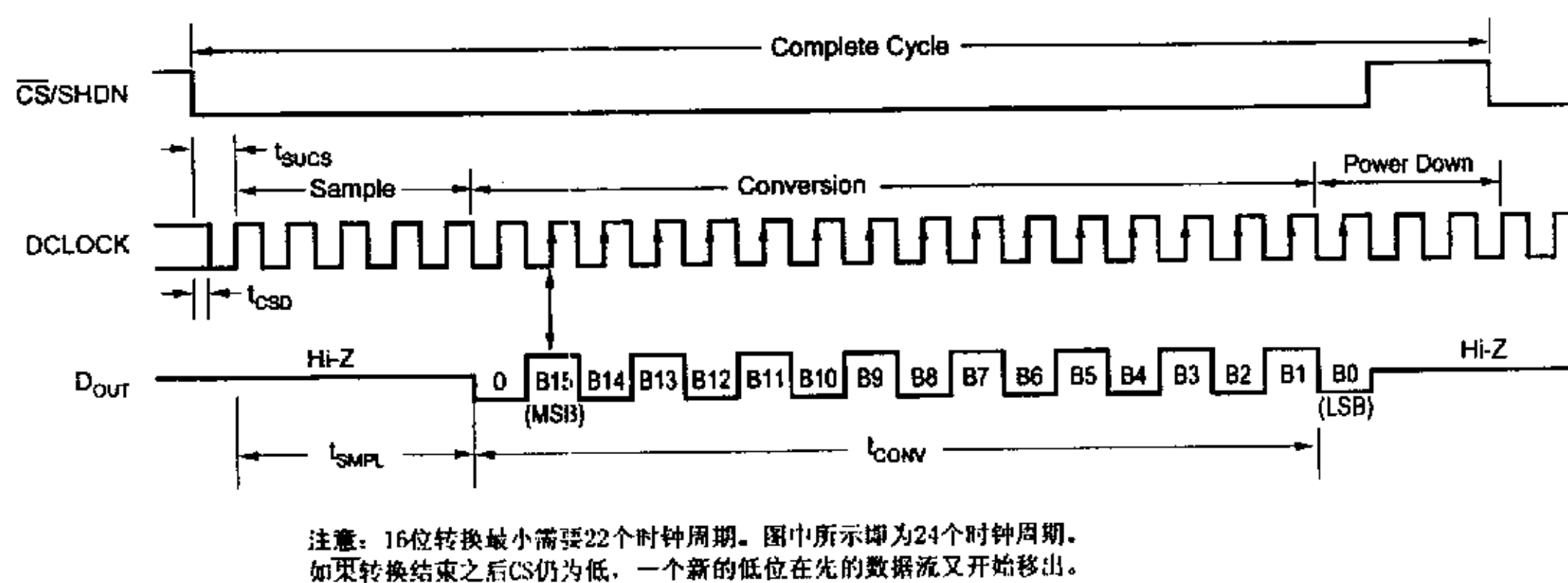


图 2.9 ADS8320 基本时序图

参见上图, CS 由高到低指示一次转换和数据传输的开始, 前 5 个 DCLOCK 周期为采样转换时间, 第 6 个周期 D_{out} 输出的为有效的转换数据, 数据是在 DCLOCK 的上升沿采样, 在下降沿传输的。这样 CS 的低状态至少保持 22 个 DCLOCK 周期, 然后 CS 变高进入省电模式。CS 再次由高到低指示下一次转换和数据传输的开始。

2.3.1.3 光纤接口芯片简介

采集系统位于水下, 数据要传输到水上需经过较长距离, 因而只能采用

串行数据传输方式。由于光纤数据传输具有传输速率高、抗干扰能力强以及接口比较简单,成本较低的优点,因而决定采用光纤进行数据传输。

本采集系统的数据传输速率最大为 40Mbytes/s,综合考察了几种常见的光纤接口器件,决定采用 Cypress 公司的第二代 HOTLink 系列串行数据传输芯片^[26]进行设计。本系统中采用的是 HOTLink 系列中的 CY7C924 数据发送/接收器^[25] HOTLink 系列器件是专门用于进行高速串行数据通信的通信接口芯片,可以用于通信速率为 50-200Mbits/s 的通信系统中。该芯片的主要特点为:

- 具有 8B/10B 编解码功能
- 8bit 或 10bit 编码传输
- 10bit 或 12bit 非编码传输
- 内嵌 FIFO (256×8bit) 并可被旁路
- 具有地址选择功能
- 信号传输速率 50~200Mbaud

本设计选用标准工作模式,宽度为 8 bits 的用户数据或命令送入发射模块中被编码,然后在移位寄存器中形成串行数据流,最后送往 PECL 电平驱动器中输出。在接收端,串行比特流进入差分线路接收器,利用其片内的锁相环时钟同步器恢复出位时钟,根据这个位时钟重建数据,进行串并转换、译码,最终恢复出发送端的并行数据来。

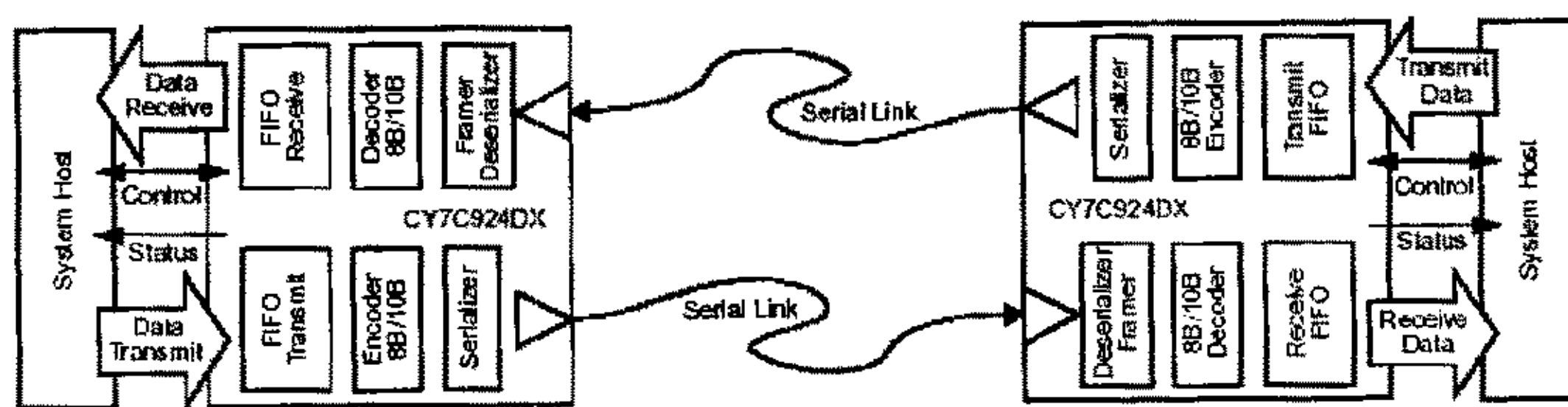


图 2.10 CY7C924 在数据传输系统中的连接方式

图 2.10 是 CY7C924 接口器件在数据传输系统中的连接方式示意图,在本系统中,数据是一帧一帧向外发送的,所以使用了 CY7C924 内部 FIFO 作为缓存。输出数据流接到光纤发送模块,本系统中采用的是 HP 公司的 HFBR5302 收发合置模块^[27],从其光器件输出编码的光信号进入光纤传输。

CY7C924 具有地址选择功能, 当地址匹配信号 AM 为低电平时将总线上的数据作为地址信息, 这使多片 CY7C924 可以共享一条总线。在使用地址选择功能时, 每片 CY7C924 均被分配一个地址, 地址值保存在地址寄存器中, 地址寄存器 8 位有效也可以扩展到 10 位。CY7C924 有两种寻址方式, 域地址(Multicast)和精确地址(Unicast)。域地址用 bit 描述, 只要有一位匹配即认为地址有效, 所以共有 8 个 (或 10 个) 域地址。使用精确地址时, 必须每一位都匹配才认为地址有效, 目的地址共有 256 个 (或 1024 个)。一片 924 可以拥有多个域地址或一个精确地址。在接收数据的过程中, 只有当总线上的地址与地址寄存器中的地址匹配时, 才将接收的数据存入接收 FIFO 中, 当地址不匹配时接收工作终止。选择 Multicast 模式时, 如果地址寄存器的值为 FFh 或 3FFh, 则将忽略地址信息所有接收到的数据都被存入 FIFO。

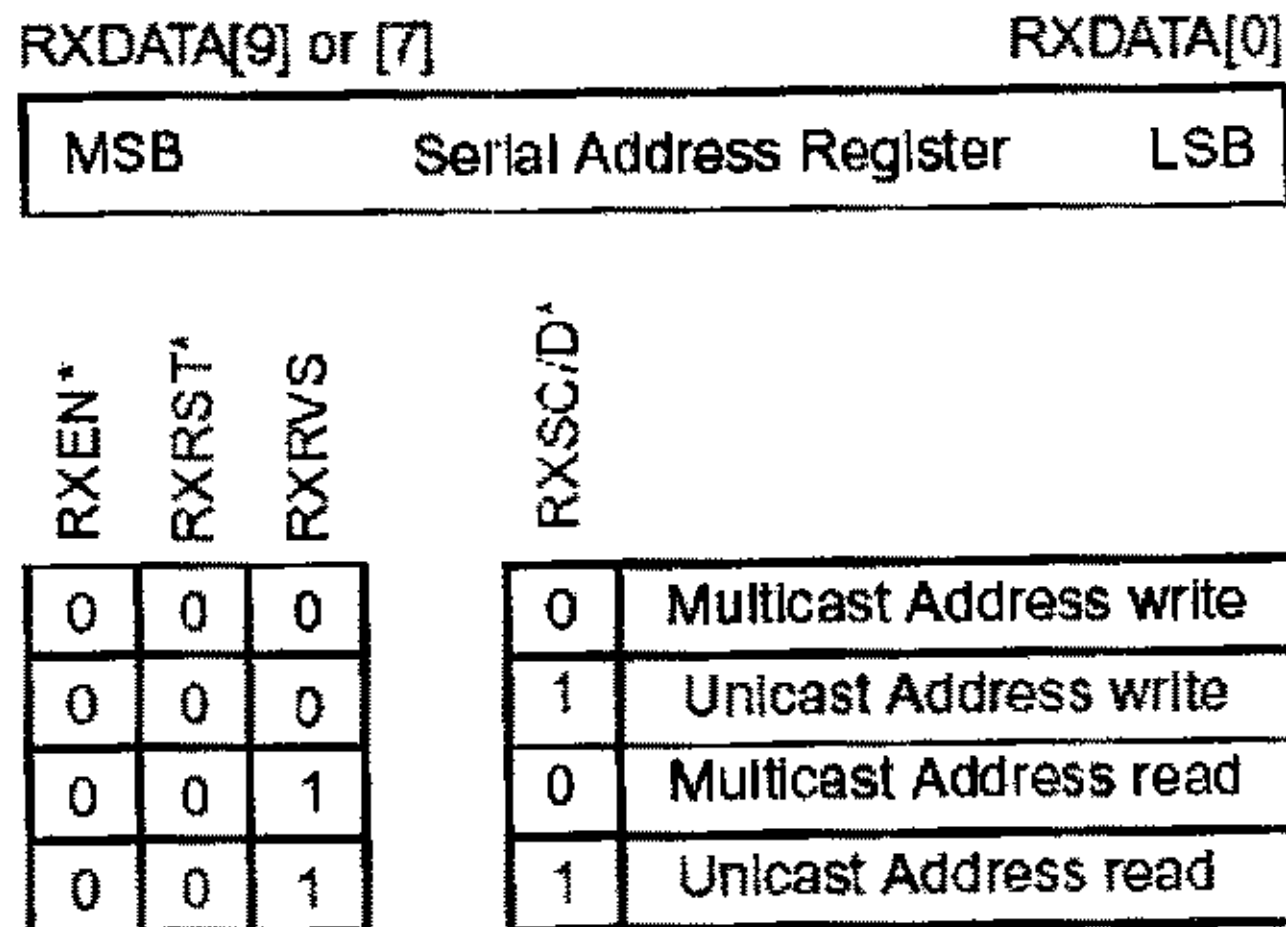


图 2.11 串行地址寄存器

使用 CY7C924 地址功能时, 其接收 FIFO 不能被旁路 (FIFOBYP 管脚设置为高), 通过 RXDATA 端口访问地址寄存器。读写寄存器时设置 AM 信号和 RXRST 信号为低电平, RXRVS 信号作为读写选择, RXSC/D 用做选择地址匹配模式。

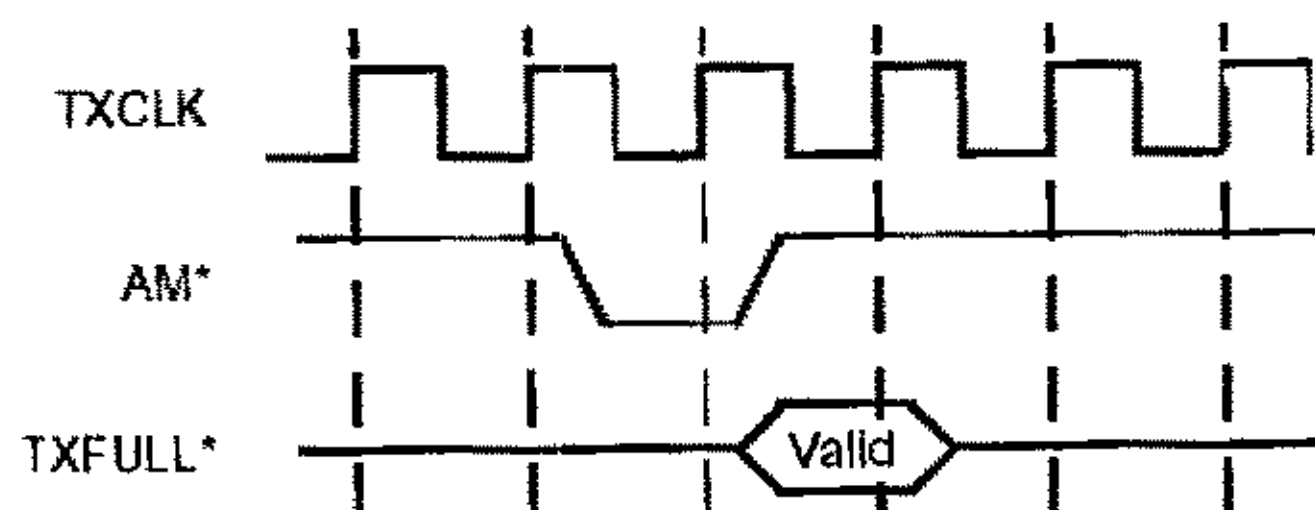


图 2.12 发送端口寻址时序图

CY7C924 在发送数据前应首先发送接收端的地址，在发送 FIFO 没有被旁路时使用 TXCLK 上升沿对 AM 采样，发送 FIFO 被旁路时使用 REFCLK 上升沿对 AM 采样，当 AM 为低时发送地址。此时，FIFO 的满标志 TXFULL 作为指示信号使用，TXFULL 跳变时操作结束。接收数据时使用 RXCLK 上升沿采样 AM，RXEMPTY 跳变时表明地址已接收。

在 CY7C924 复位的过程中 FIFO 的满标志 TXFULL 和空标志 RXEMPTY 作为复位指示信号使用，当其翻转时表示 CY7C924 已复位。图为发送 FIFO 复位时序

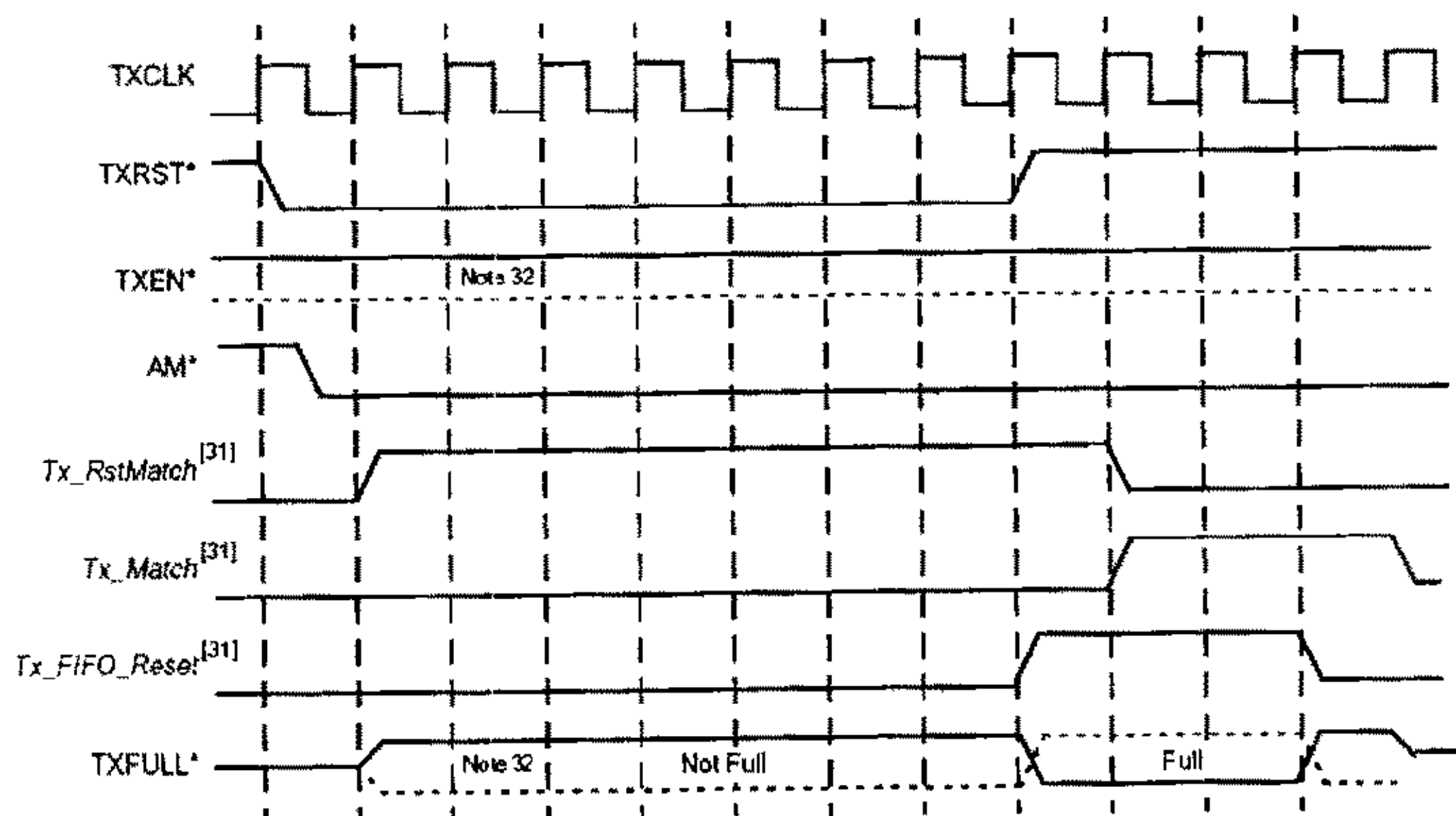


图 2.13 发送 FIFO 复位时序图

工作过程中光纤接口由 FPGA 控制，当 FPGA 将一帧数据写入发送 FIFO 后，使能发送控制信号将 FIFO 中的数据送出。本设计采用 20M 参考时钟，光纤传输速率为 200M 波特，足以满足系统要求。HOTLink 器件在光纤链路

没有数据传输时，会自动填充空字符（K28.5）以保持接收锁相环的同步，在接收数据的过程中 CY7C924 会自动识别 K28.5，只将有效数据存入接收 FIFO。

2.3.2 DSP 程序和接口设计

采集控制电路中 DSP 完成产生自检信号和接收机增益控制的工作。系统开始工作时首先进行系统自检，由 DSP 产生线性调频信号作为自检信号。自检信号经 DAC、线性光耦和模拟开关加载到接收机输入端，通过对自检信号的采样来监控接收机增益和相位一致性以及滤波器的通带特性。采集数据时 DSP 自动调节接收机增益，使接收机输出信号的幅度保持稳定。

2.3.2.1 系统自检

系统自检时，24 通道同时自检，由 AC_COM 信号统一控制模拟开关。自检信号从 C5402 的串口 1 通道（McBSP1）发送至 DAC，光耦隔离后经过模拟开关送给 24 通道。。在这种连接中，McBSP1 是主设备，而 DAC7611 是从设备。它们之间的简单连接方式如图所示。

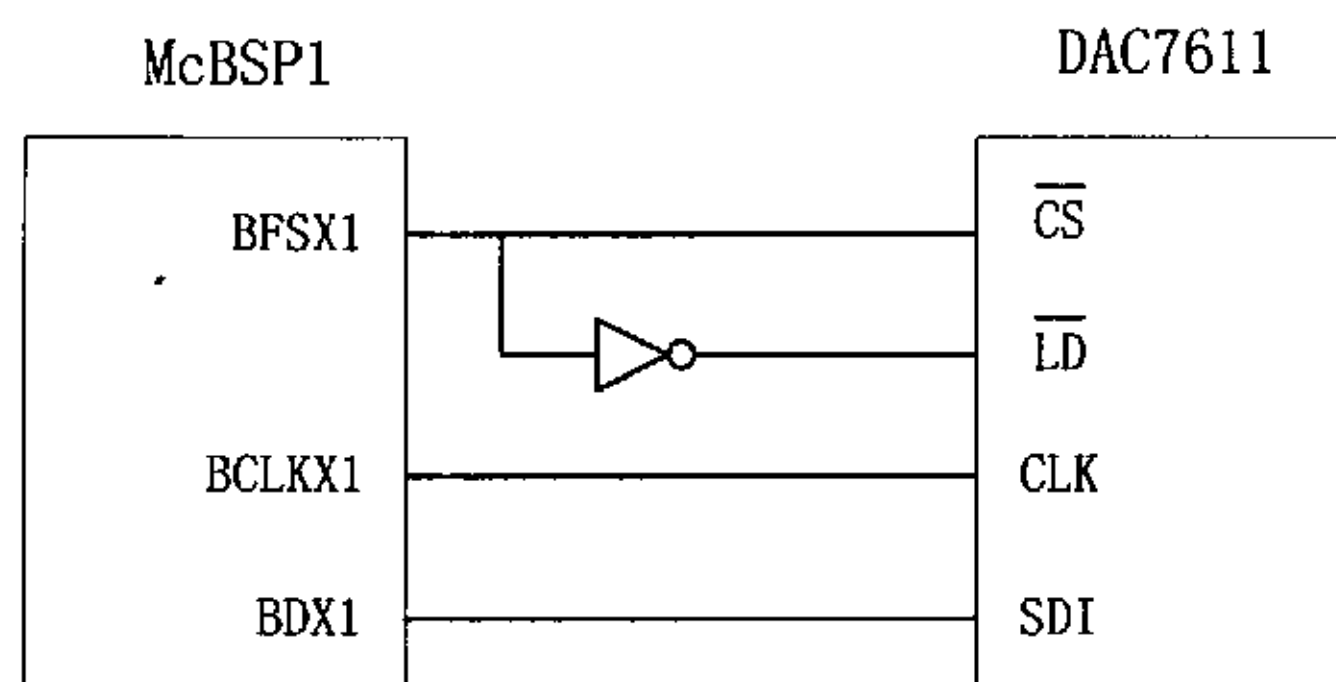


图 2.14 McBSP1 与 DAC7611 之间的连接

时钟 BCLKX1 为 10MHz，使用 8 点的正弦表可产生最高频率为 100KHz 的正弦信号。从图 2.10 中我们可以看出， $\overline{CS}$ 与 $\overline{LD}$ 互为反相关系。 $\overline{LD}$ 为高应至少保持 12 个 CLK 周期，若多于 12 个周期，则高位数据被覆盖掉，串行移位寄存器只保留最后移入的 12 位数据。 $\overline{LD}$ 为低至少保持 1 个 CLK 周期。另外，数据在 CLK（BCLKX1）的上升沿移入串行移位寄存器。

CPU 首先将要发送的数据写入 DXR[1,2]，然后把 DXR[1,2] 中的数据拷贝

到 XSR[1,2]中。在帧信号的同步下, XSR[1,2]中的发送数据开始移出 DX 脚。若数据全部移出, 则 McBSP 向 CPU 产生中断, CPU 可以向 DXR[1,2]写入下一数据。结合时序图, 可以确定出 DXR[1,2]中待转换数据的存放格式(表中“X”表示不影响输出的取值):

表 2.1 DXR[1,2]中的数据格式

DXR21															
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DXR11															
X	X	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	X	X

2.3.2.2 接收机增益控制

为了避免接收机饱和或过载并且保证输入信号有足够的动态范围, 在矢量阵接收系统的模拟接收机中采用了自动增益控制系统。其主要作用是当无信号时进行噪声增益控制, 即保证接收机具有一定的放大量; 当有信号时进行信号增益控制避免接收机饱和或过载, 使其输出稳定在一定范围内。

由于矢量水听器有声压和三维质点振速共 4 路输出, 所以在 24 通道接收机中每 4 个通道为一组单独控制, 即可以控制 6 个水听器的接收增益。本设计采用的是闭环结构, 用软件方法实现。开始工作时首先由显控台下传接收机参数, 通过 FPGA 配置给接收机, 并且 FPGA 向 DSP 发出中断信号, 使 DSP 接收到接收机初始增益。在数据采集模式下接收机的输出完成 A/D 转换后, FPGA 采样并抽取 1、5、9、13、17 和 21 通道数据, 将数据打包发送给 DSP。DSP 使用 McBSP0 接收数据, 并使用 DMA 通道 4 将数据从串口数据接收寄存器搬移到内部 RAM 进行后续处理。DSP 程序流程如图, 每接收到一个采样点的数据都要经过一次计算, 并得出下一次采样时接收机的增益。

程序流程如图:

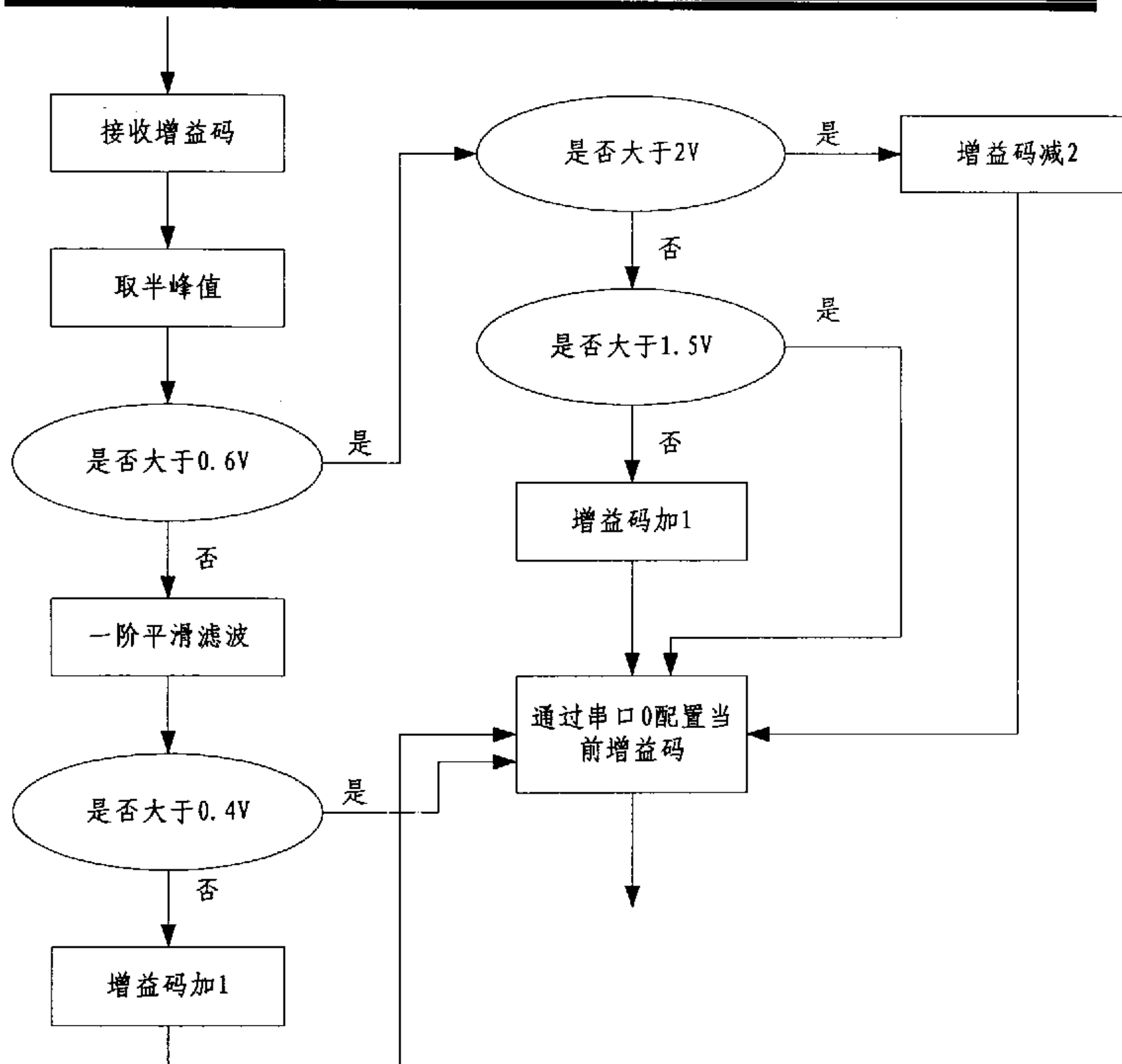


图 2.15 自动增益控制程序流程图

DSP 使用 DMA 通道 5 和 McBSP0 发送下一个采样周期的接收机增益到 FPGA，数据格式为每帧 12bit 共 6 帧，高 4 位为通道号，低 8 位为增益码。接收机增益的配置工作最后由 FPGA 内 configure 模块完成，通过识别每帧数据中的通道号对各通道依次配置，例如通道号为 5 则将增益码配置给通道 5、6、7、8（每次控制 4 个通道）。A/D 采样及数据抽取在数据转换控制步内完成，DSP 通过串口接收数据、计算接收机增益和最后完成增益码的配置都要在采样间歇控制步内完成，时间大约需要 5us。故当采样率大于 66kHz 时采样间歇时间要小于 5us，对接收机增益的控制会延迟一个采样周期，采样率小于 66kHz 时不会产生延迟。

接收机输出信号的峰峰值范围为 0~5V，本设计将噪声控制门限定为 0.4~0.6V（半峰值），将输出信号控制在 1.5~2V（半峰值）。

2.3.4 采集控制逻辑设计

采集控制板上使用一片 FPGA 实现采集、控制以及各种接口逻辑。芯片采用 Altera 公司的 FLEX10K30A，它是第一代嵌入式可编程逻辑器件。它不仅极大地提高了系统设计效率，而且使设计者摆脱了大量的辅助性工作，将精力集中于创造性的方案与概念的构思上。FPGA 还可以起到总线隔离的作用，光纤接口芯片和 AD 是 5V 的 TTL 电平，而 FPGA 的 I/O 电压为 3.3V，容忍 5V 的输入 TTL 电平。这在一定程度上降低了本系统电路的复杂性。

采集的数据以每帧 64Byte 的格式向外发送，一个 4Byte32 位的计数器表示该数据包的序号；24 通道的数据，每通道 2Byte，共 48byte；6Byte 的增益码，每个管 4 路；6Byte 的姿态码。

表 2.2 数据采集格式

数据格式	Byte offset
帧头	0~3 Byte
数据（1~24 通道）	4~51 Byte
增益码	51~57 Byte
水听器姿态	57~63 Byte

2.3.4.1 总体设计思想

FPGA 程序应给上层控制系统提供一个尽可能简单清晰、可操作性强、易于编程的接口，以简化上层程序的设计并提高其执行效率。一个简单清晰的程序可以避免产生一些不可预知的错误。同时，在设计 FPGA 程序时还要考虑到如何充分利用其内部资源，避免不必要的浪费。本系统 FPGA 程序的设计思想如下：

- 各逻辑单元均采用层次化、模块化设计，根据功能分为采集控制、数据转换、光纤接口、存储控制接口、电源管理、增益码配置、时钟分频七个模块。各模块间信号尽可能独立，互不影响，采用模块化设计给程序的开发和维护带来很大方便。

- 完成多通道串并转换电路时会消耗大量的资源。本设计充分利用 FPGA 内部的 EAB 资源，用 RAM 作为串并转换电路的缓存区，降低了片内资源的消耗。
- 根据 ADS8320 的特点，FPGA 内核逻辑依据采样保持、数据转换、采样间歇三个控制步设计。由一个采集控制模块给出采样保持、数据转换、采样间歇三个控制信号作为其它各单元的使能信号。

2.3.4.2 各功能模块的设计实现

➤ 数据转换模块

ADS8320 的 CS 和 CLK 由时钟分频模块分频产生,CLK 值固定为 500ns,显控台下传采样频率到分频模块产生 AD 片选信号,实现对采样频率的控制。采集控制模块将 24 通道的 AD 数据 $24 \times 16\text{bit}$ (串行) 转换成 $48 \times 8\text{bit}$ 的数据包。如果直接将串行数据转换成并行数据,则每个串并转换模块需占用 35 个 LC (逻辑核),完成 24 通道的串并转换需占用 FPGA 内大量的资源。

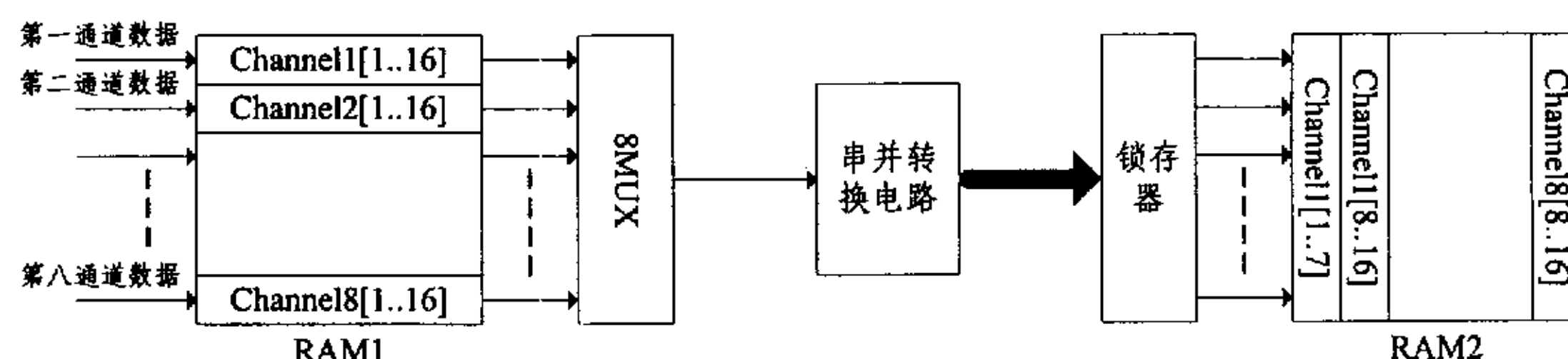


图 2.16 串并转换电路原理框图

本论文采用 3 个转换模块完成 24 通道的串并转换工作,每个模块由 2 块 RAM ($16 \times 8\text{bit}$)、一个 8 选 1 电路、一个串并转换模块、地址发生器和锁存器组成。在数据转换控制步内将串行数据直接写入 RAM1,在采样间歇控制步内通过 8 选 1 电路分别选通各路 AD 数据,使用一个串并转换电路依次将各路串行数据转为并行并存入 RAM2 (见图 2.16)。在采样保持控制步中完成数据打包和发送工作。

➤ 时钟分频模块

在本设计中主时钟为 40M,其它各模块所需时钟、DSP 串口的帧信号和 AD 的 CS 等,都可以当作占空比不同的时钟信号处理。所以设计一个可变参数为时钟周期和占空比的通用分频模块,用于产生各种时钟信号。

➤ 增益码配置模块

增益码配置模块与 DSP 的 McBSP0 的发送端接口。从 DSP 串口输出的配置码每帧为 12bit，高 4 位用作通道选择，低 8 位是数据。首先使用一个串并转换电路将配置码变为并行并用锁存器锁存。然后通过高 4 位数据判断这一帧配置码是滤波器带宽还是某一通道的接收增益，其中各通道的滤波器带宽统一配置，接收增益码 CS 有 6 路每路管 4 个通道。配置过程在采样间歇控制步内完成。

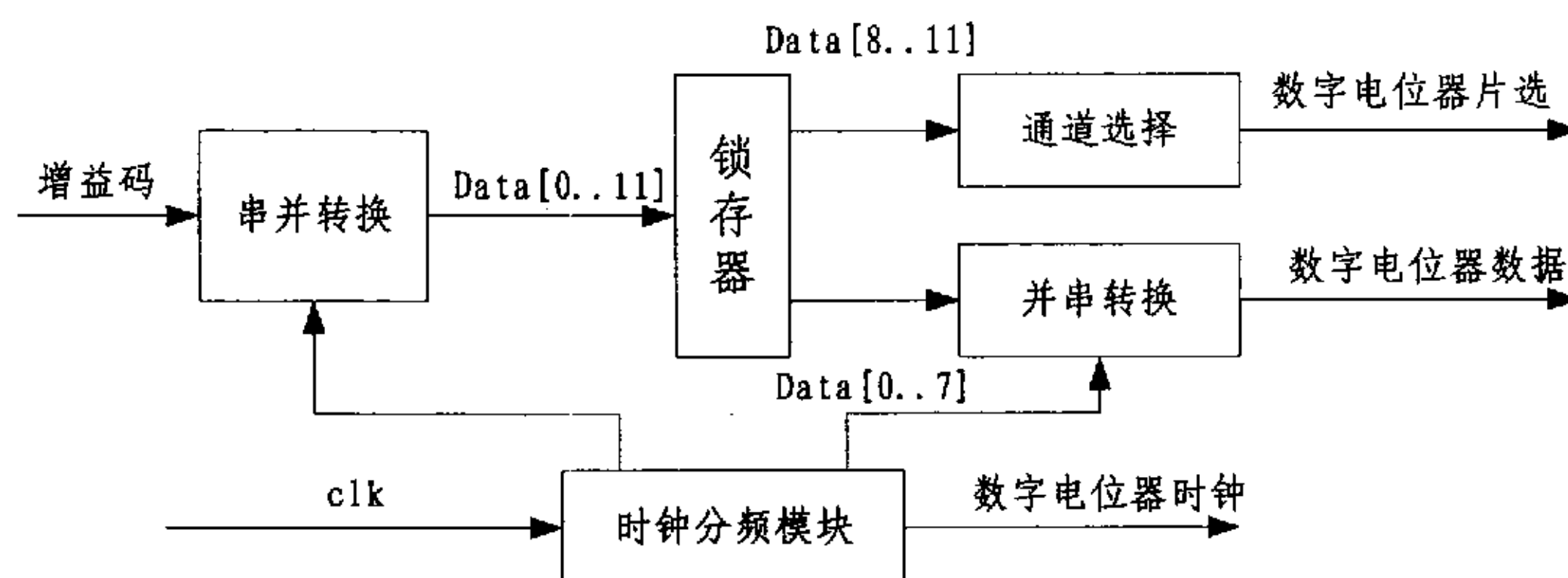


图 2.17 增益码配置模块原理框图

➤ 光纤接口模块

光纤接口芯片采用的是 Cypress 公司的第二代 HOTLink 系列串行数据传输芯片 CY7C924（见光纤接口简介），它按工作方式可分为同步、异步、标准和级连 4 组模式，本设计采用的是使用内部 FIFO 作为缓存区的标准异步工作模式。接口模块分为两个单元，一部分为 FIFO 控制，另一部分为光纤发送控制。要注意的是 FIFO 的写时钟 CLKW 和光纤发送时钟 REFCLK 一定要同步。本系统光纤为点对点传输，所以没有使用地址功能，AM 信号被置为高。

➤ 存储控制模块

存储控制模块和存储板 FIFO 接口。系统初始化时首先将 FIFO 复位，然后等待存盘命令。存盘命令由光纤下传，当控制命令字是 03H 时将打包后的数据写入 FIFO。当存储控制板 FPGA 查询到 FIFO 的状态非空时，立即启动存盘程序将数据写入硬盘。本设计中从 FIFO 中读取数据的速度大于写入速度，所以在没有接收到终止命令时，存储控制模块将打包后的数据持续写入 FIFO 而不受其它信号的影响。当已经存储 40M 数据时，存储控制板 FPGA

发出终止信号，控制模块接收到该信号后停止向 FIFO 传输数据，并将 FIFO 复位等待下一次传输。

2.4 存储控制设计与实现

2.4.1 硬盘控制逻辑及接口设计

数据存储是数据采集过程中的一个重要环节。为了使系统具有内记功能，并能及时存储采集到的数据，设计了存储控制电路完成这一任务，选用计算机硬盘作为数据储存载体。在以往的内记系统中，往往采用内置工控机的方法完成数据保存任务。这种方案的缺点是系统功耗大，硬件成本高。采用直接控制硬盘的方法可将系统功耗降至最低，并可降低硬件成本。本设计采用 FPGA 和 DSP 直接控制硬盘，并采用一片 FIFO 作为数据缓存。

2.4.1.1 ATA 接口协议概述

我们一般习惯称硬盘接口为 IDE 接口，IDE 所代表的含义是集成磁盘电路设备（Integrated Disk Electronics）。实际上，IDE 接口的正式名字是 AT-Attachment（ATA）。IDE 接口于 1984 年开始了它的发展进程，并且在得克萨斯州的计算机制造商康柏（Compaq）公司的努力下，得到了长足的发展。这个思想被应用于 IBM AT 兼容机的硬盘控制器的磁盘驱动机构上。1985 年，磁盘制造商 Imprimis（CDC）在他们的硬盘驱动器中集成了这种控制器。这样，第一部 IDE 的磁盘驱动器就被生产出来，并且还被安装到了康柏的计算机系统中。紧接着，在 1988 年 10 月，ANSI 协会的 X3T9.2 工作组的一个委员会开始着手处理这个问题。在此次会议中，通用存取方法协会（CAM）提出了对 IDE 接口进行标准化处理的建议，并以此作为它的第一个议程。然后就出现了 IDE 接口的新名字也就是 ATA。这个标准化后的规范也就是现在的 X3.221-1994。在 1995 年的秋天 ATA-2 标准被制定并得到批准。该标准提出了更高的数据传输速率，并制定了一些新的命令。在那以后 ATA 协议开始了它的长足发展。本系统是基于 ATA-4 标准设计的，使用 IO2 数据传输方式，

稳定存储速率为 6.3Mbytes/S，满足最高采样率时每秒 5Mbytes 的存盘要求。

2.4.1.2 硬盘接口概述

ATA 接口一般为 40 脚，用扁平电缆互连，其长度一般不超过 46cm。以下为各管脚功能的简要描述。IN 表示进入硬盘，OUT 表示从硬盘中出来，I/O 表示双向传输。

表 2.3 硬盘接口

来源	信号	管脚	管脚	信号	来源
IN	RESET	1	2	GND	
I/O	DD7	3	4	DD8	I/O
I/O	DD6	5	6	DD9	I/O
I/O	DD5	7	8	DD10	I/O
I/O	DD4	9	10	DD11	I/O
I/O	DD3	11	12	DD12	I/O
I/O	DD2	13	14	DD13	I/O
I/O	DD1	15	16	DD14	I/O
I/O	DD0	17	18	DD15	I/O
	GND	19	20	N.C.	
OUT	DMARQ	21	22	GND	
IN	DIOW	23	24	GND	
IN	DIOR	25	26	GND	
OUT	IORDY	27	28	CSEL	IN
IN	DMACK	29	30	GND	
OUT	INTRQ	31	32	RSVD	
IN	DA1	33	34	PDIAG	I/O
IN	DA0	35	36	DA2	IN
IN	CS0	37	38	CS1	IN
OUT	DASP	39	40	GND	

- CS0—、CS1—、DA0、DA1、DA2 为硬盘地址线，用来选通硬盘内部寄存器，CS0、CS1 低电平有效。
- DMARQ、DMACK 一是进行 DMA 操作时的一对握手信号。
- INTRQ 是硬盘产生的中断请求信号。
- CSEL (cable select) 接 GND 默认硬盘为设备 0，接 VCC 默认硬盘为设备 1。通常一个 IDE 接口可以连接两块硬盘，以 80 线电缆为例，在电缆

中间的接口连接的是设备 0，电缆末端的接口连接的是设备 1。

- PDIAG—上电启动后当设备 1 自检结束时给出 PDIAG 通知设备 0，低电平有效。当系统 reset 后、或执行完命令 identify device 或 identify packet device 后主机可对该管脚采样，如为低电平则使用的是 80 线电缆可以进行高速数据传输，否则使用的是 40 线电缆不能使用 DMA 方式 3 及以上工作方式。
- IORDY 该管脚被 IORDY：DDMARDY—：DSTROBE 复用。
IORDY 在 I/O 传输方式 3、方式 4 和进行 DMA 传输时主机一定要响应 IORDY 如果使用 I/O 传输方式 0、方式 1、方式 2 传输数据时可选择使硬盘不产生这个信号。
DDMARDY—，在以 UDMA 方式向硬盘传输数据时，该信号为低表示设备已经可以接收数据，当该信号变高则主机应暂停数据传输。
DSTROBE，在以 UDMA 方式从硬盘读取数据时，该信号作为读时钟，上升沿和下降沿都有效。
- DIOR 该管脚被 DIOR—：HDMARDY—：HSTROBE 复用。
DIOR—是 I/O 和 Register 传输时的读信号。
HDMARDY—，在以 UDMA 方式从硬盘读取数据时，主机使用该信号通知设备已经可以接收数据，如果该信号未被激活，则设备应暂停数据传输。
HSTROBE，在以 UDMA 方式向硬盘传输数据时，该信号作为写时钟，上升沿和下降沿都有效。
- DIOW 该管脚被 DIOW—和 STOP 复用。
DIOW—是 I/O 和 Register 传输时的写信号。
STOP，在以 UDMA 方式传输数据时应首先激活该信号，在 UDMA 初始化过程中将其拉低，当主机再次激活该信号时则终止 UDMA 传输。
- DASP—该信号用来表示硬盘的工作状态，当 DASP 为低电平时表示硬盘正在工作。
RSVD (reserved) 可将其悬空。

2.4.1.3 硬盘内部寄存器及控制命令

对硬盘的操作如读写扇区，发送命令等都是通过读写寄存器来完成的，向硬盘存储或读出数据也是通过读写寄存器完成的。每向硬盘发送一组命令后都要读取状态寄存器中的信息，通过状态寄存器的值来判断硬盘是否已经执行完所发送的命令。除 data register 和 data port 外，每个寄存器字长均为 8bit，用硬盘控制线中的 CS0,CS1,DA0,DA1,DA2 来选通寄存器，通过数据总线中 DD0~DD7 向寄存器写入或从中读取数值。

一般每硬盘发送一次控制命令，都要对一组寄存器进行写操作。下面以向硬盘发送写扇区命令为例，对这一过程详细论述。

写扇区命令 (WRITE SECTOR, 命令码 30h) 是一个 IO 传输方式的控制命令，其功能是将数据寄存器中的数据储存在指定的扇区上。首先，将要存储数据的数据量，即数据占用的扇区数写入扇区计数寄存器，每扇区容量为 512 字节。硬盘存储数据时以扇区为基本单位，当要存储的数据不足 512 字节时也会占用一个扇区的空间，剩余空间用 0 填充。从硬盘读取数据时同理，一次至少要读取 512 字节即一扇区数据。如某扇区上有 400 字节数据，则在接收到 400 字节数据后需继续从数据寄存器读取 112 字节数据 0，否则将产生错误状态。

其次，需将接收数据的扇区地址写入地址寄存器。对扇区寻址共有两种方式，物理寻址和逻辑寻址。物理地址即使用柱面、磁头和扇区编号唯一的确定一个扇区。逻辑地址是物理地址的线性映射。也就是说，扇区是从柱面 0、磁头 0、扇区 0 开始连续地排列下去。在这个磁道之后，就是同一个柱面的磁头 1，这样下去直到整个柱面，再接下去是下一个柱面的 0 磁头。在逻辑寻址方式（以下称 LBA）中，使用 28 个数据位来表示逻辑区块的地址信息，这样可以寻址 2^{28} 个区段，拥有 127 个 GB。在 ATA 标准中，从物理结构到逻辑块编号的映射是按照如下的方式进行的：

$$LBA = (\text{柱面编号} \times \text{磁头数} + \text{磁头编号}) \times \text{扇区数} + \text{扇区编号} - 1$$

这种映射方式假设从 LBA_N 到 LBA_{N+1} 的访问时间要比从 LBA_N 到 LBA_{N+2} 用得少。换一种说法就是，逻辑扇区在访问时间上也是按顺序排列的，这意味着对于逻辑上连续存放的数据，其读写的时间是最短的。本设计采用逻辑寻址方式，需在 4 个寄存器中写入扇区对应的逻辑地址，并置 Device register

的第 7 位为 1，表示采用逻辑寻址方式。Device register 的第 5 位为设备识别位，置 0 表示对主盘操作，置 1 表示对从盘操作。

最后，将写操作的命令码 30h 写入 Command register，完成整个操作。

当使用 IO 工作方式时，硬盘通过 data register 接收发送数据；使用 DMA 工作方式时，通过 data port 接收发送数据。WRITE SECTOR 为 IO 控制命令，所以应使用 data register，只有在硬盘处于数据接收准备好状态时才能将数据写入 data register。我们通过访问 Status register 来获得硬盘状态。当 Status register 的 DRDY 和 DRQ 为 1 时，表示硬盘处于数据接收准备好状态，可以向 data register 写入数据。当 Status register 的 ERR 位为 1 时，表示命令执行过程中产生了错误。

WRITE SECTOR

命令码：30h

表 2.4 磁盘命令输入格式

Input:

Register	7	6	5	4	3	2	1	0
Sector count register	要写入硬盘的扇区个数							
LBA High register	逻辑地址 23~16 位							
LBA Mid register	逻辑地址 15~7 位							
LBA Low register	逻辑地址 7~0 位							
Device register	obs	LBA	obs	DEV	逻辑地址 27~24 位			
Command register	30h							

LBA 被置 1 表示使用逻辑扇区地址寻址，被置 0 表示使用柱面、磁头和扇道号寻址。

DEV 被置 1 表示选择设备 1，被置 0 表示选择设备 0。

Normal output:

Register	7	6	5	4	3	2	1	0
Device register	na	na	na	DEV	na	na	na	na
Status register	BSY	DRDY	DF	na	DRQ	na	na	ERR

DEV 的值应与被选择的设备相同。

BSY 被置 0，表明命令已经执行完毕。

DRDY 被置 1。

DF 被置 0。

DRQ 被置 1，表示设备已经可以接收数据。

ERR 被置 0，没有错误产生。

Na 表示该位被忽略。

Error output:

Register	7	6	5	4	3	2	1	0
Error register	na	WP	MC	IDNF	MCR	ABRT	NM	na
Status register	BSY	DRDY	DF	Na	DRQ	na	na	ERR

WP 被置 1 表示可移动驱动器被写保护。

MC 被置 1 表示在执行最后一个命令时可移动驱动器被改变。

IDNF 被置 1 表示地址寄存器中的地址超出硬盘的逻辑地址范围。

MCR 该信号在可移动驱动器检测到请求改变磁盘的命令时被置 1，只有 GET MEDIA STATUS 命令和插入磁盘命令才可将其置 0。

ABRT 被置 1 表示命令错误，在 IDNF 被置 1 时 ABRT 也会被置 1。在进行 Ultra DMA 操作时如出现冗余校验错误该信号也会被置 1。

NM 被置 1 表示可移动驱动器中没有磁盘

2.4.1.4 操作控制流程

➤ 硬盘启动操作

硬盘加电后应先将硬盘复位。硬盘复位有三种方式，使用 IDE 管脚的 RESET 信号、将 Device register 中 SRST 位置 1 或使用 RESET COMMAND 命令。

本设计采用的是将 SRST 置 1，当 SRST 被置 1 后硬盘立即复位，这时应至少等待 2ms 才读取 Status register。如果 Status register 的值为 80h，则表明硬盘已完成复位。

硬盘启动流程图：

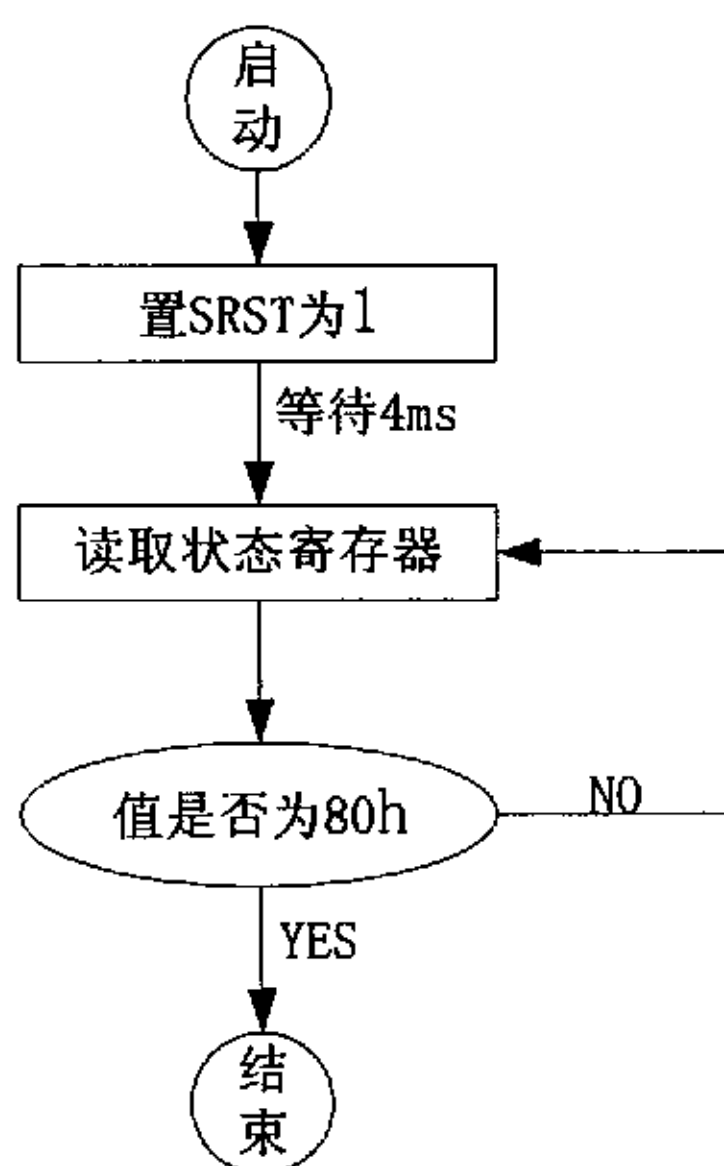


图 2.18 硬盘启动程序流程图

➤ 扇区读写操作流程

首先应向硬盘发送命令参数，将要读写的扇区数目写入 Sector count register，将扇区的逻辑地址写入 LBA High register、LBA Mid register 和 LBA Low register，将要接收命令的设备号写入 Device register。

将命令 READ SECTOR 或 WRITE SECTOR 写入 Command register 后，在 400ns 内 Status register 的值并不可靠，所以需等待 400ns 才能读取 Status register，这段等待时间可以通过读取 Alternate Status register 和 Error register 度过，读取的数值应被丢弃。当状态寄存器的值为 58h 时，表示硬盘已经可以从 Data register 接收或传输数据。

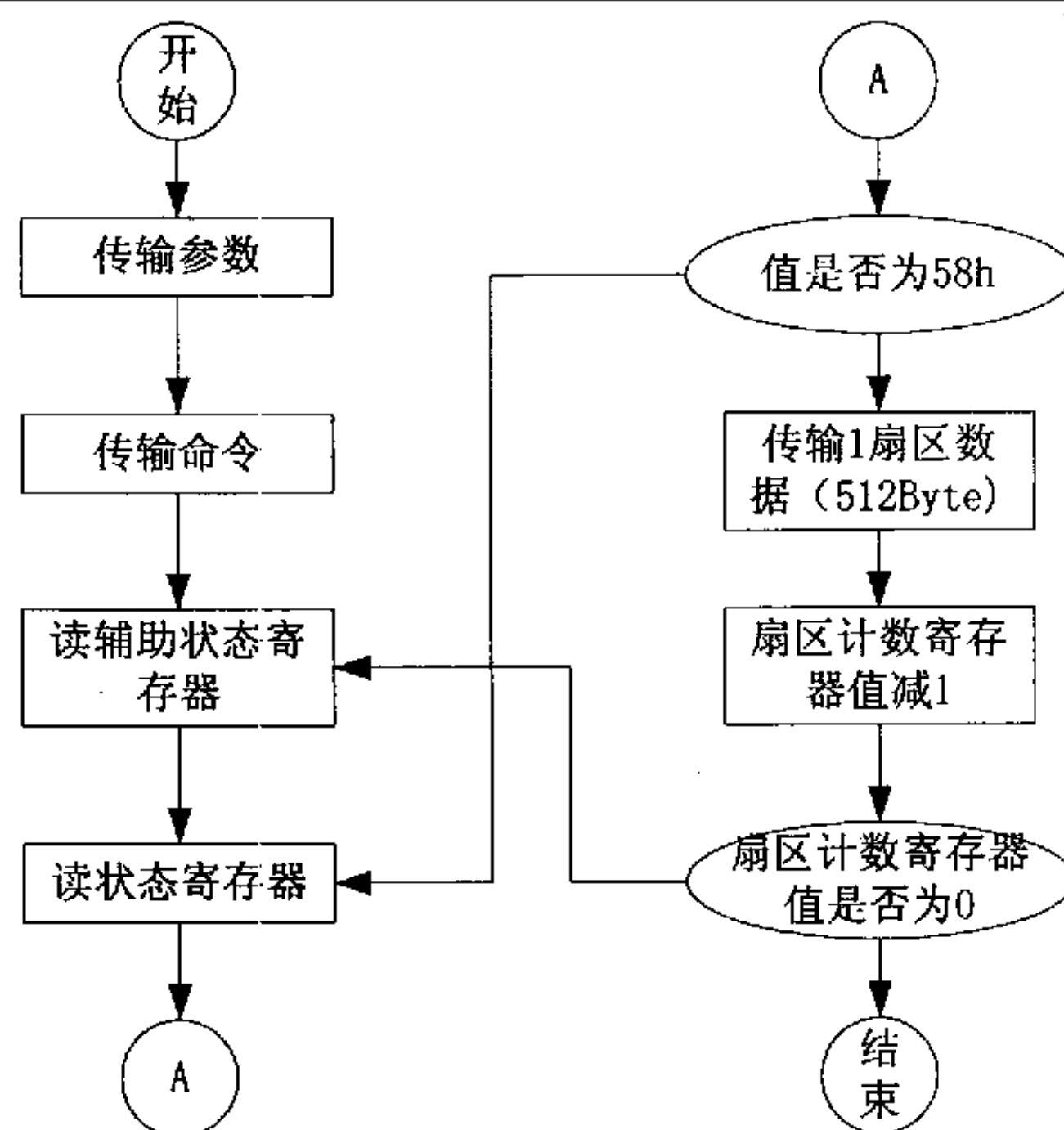


图 2.19 扇区读写操作流程

➤ DMA 操作流程

DMA 操作与 I/O 操作相似，首先应向硬盘发送命令参数，将要读写的扇区数目写入 Sector count register，将扇区的逻辑地址写入 LBA High register、LBA Mid register 和 LBA Low register，将要接收命令的设备号写入 Device register。将命令 READ DMA 或 WRITE DMA 写入 Command register 后，在 400ns 内 Status register 的值并不可靠，所以需等待 400ns 才能读取 Status register，这段等待时间可以通过读取 Alternate Status register 和 Error register 度过，读取的数值应被丢弃。当状态寄存器值为 58h 时主机应给出 DMA 初始化时序开始传输数据。

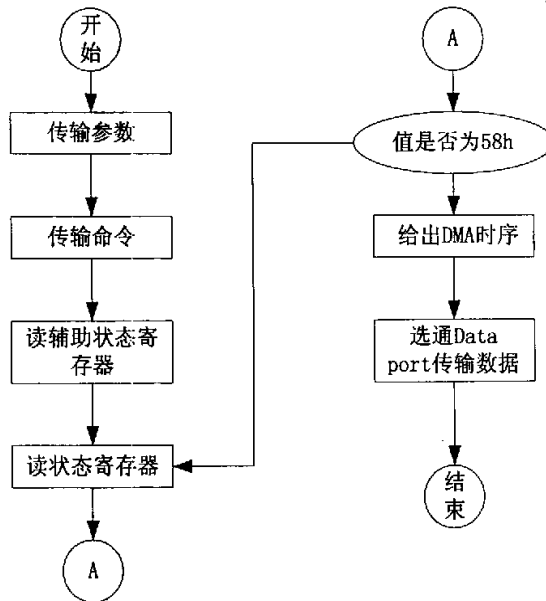


图 2.20 DMA 操作流程

2.4.2 文件管理

2.4.2.1 磁盘分区格式

当硬盘被格式化时，硬盘空间被分为 4 个部分：硬盘分区表（MBR）、硬盘 BIOS 表（BPB）、文件分配表（FAT）、数据空间。

当格式化时将硬盘分为多个磁盘时（如分为 C、D、E、F），硬盘的空间划分如下表所示。

MBR0	隐含扇区	BPB 0	隐含扇区	FAT	数据空间	MBR1
隐含扇区	BPB1	隐含扇区	FAT	数据空间	MBR2	... 数据空间

硬盘分区表（MBR0）如下图所示，1C7h 处的值 3Fh 为 BPB0 逻辑地址，1CAh 处的值 3E821Dh 为当前分区的磁盘容量，即分区包含的扇区个数。1D7h 处的值 3E823Fh 是 MBR1 的逻辑地址，MBR0 的逻辑地址是 0h，1DAh 处的值 2241C07h 是硬盘除第一个分区以外的容量（包含的扇区个数）

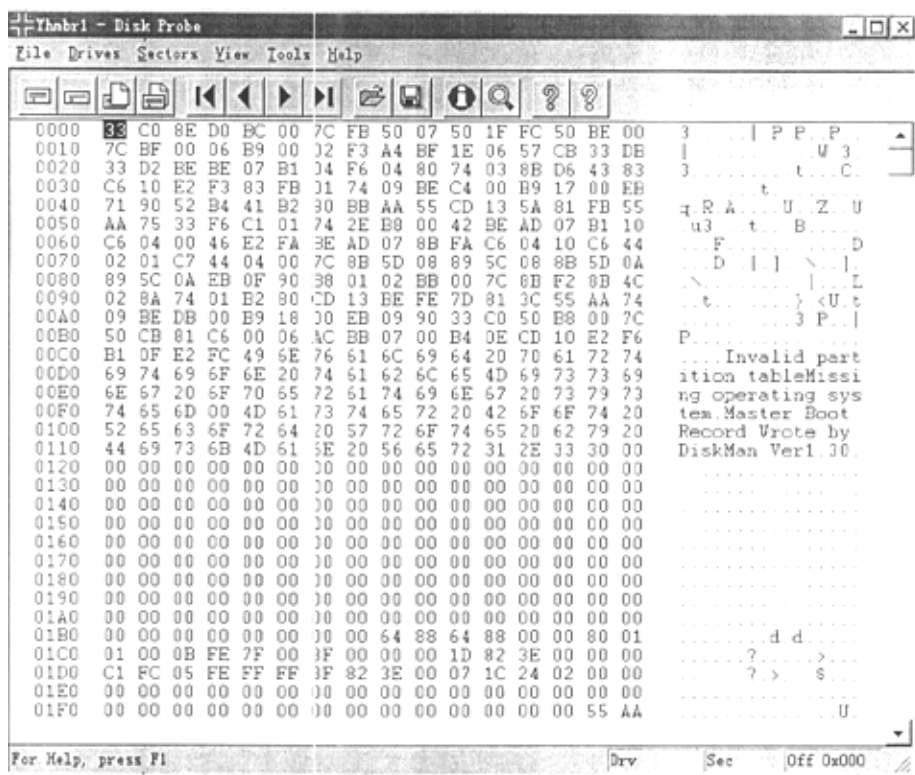


图 2.21 硬盘分区表 (MBR0)

硬盘分区表 MBR1 如下图所示。1C7h 处的值 3Fh 为 BPB1 的地址，1CAh 处的值 9C64BFh 为当前分区的磁盘容量，即分区包含的扇区个数。1D7h 处的值 9C64FEh 是 MBR2 的逻辑地址偏移量，1DAh 处的值 E2BF5Dh 是下一个分区的容量（包含的扇区个数）。

如果 1D2h 处为 05h 则应按照下列规则计算 MBR 逻辑地址：

MBR0 逻辑地址=00h

MBR1 逻辑地址=MBR0 中 1D7h 处的值

MBR2 逻辑地址=MBR1 逻辑地址+MBR1 中 1D7h 处的值

MBR3 逻辑地址=MBR1 逻辑地址+MBR2 中 1D7h 处的值

BPB 逻辑地址=MBR 逻辑地址+MBR 中 1C7h 处的值

当 MBR 中 1D2h 处值为 0h 时，表明当前分区为最后一个磁盘分区。

MBR0 起始值为 C033XXD0h, MBR 和 BPB 均为特殊扇区, 最后两字节都为 AA55h。

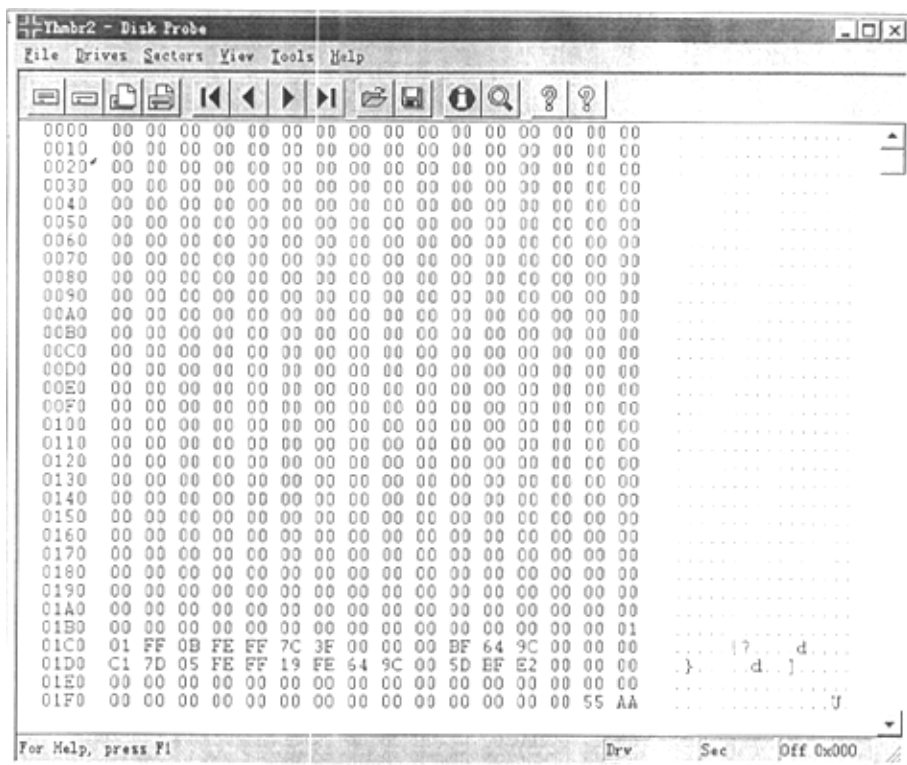


图 2.22 硬盘分区表 (MBR1)

2.4.2.2 FAT32 文件系统

➤ 引导扇区及文件分配表概述

引导扇区即 BPB 表, 记录了文件分配表的地址、个数、大小、簇的大小等信息。BPB 表内容如下:

表 2.5 硬盘 BIOS 结构信息

名称	偏移量	长度 (Byte)	
BS_jmpBoot	0	3	起始标志, 有两种形式 JmpBoot [0]=0xEB, jmpBoot [1]=0x?, jmpBoot [2]=0x90

			Gumboot [0]=0xE9, jmpBoot [1]=0x?, jmpBoot [2]=0x??
BS_OEMName	3	8	格式化硬盘的系统的标志, 一般为“MSWIN4.1”
BPB_BytsPerSec	11	2	每扇区的字节数, 一般为 512、1024、2048 或 4096
BPB_SecPerClus	13	1	每簇的扇区个数
BPB_RsvdSecCnt	14	2	保留扇区数, 应是一个非零数
BPB_NumFATs	16	1	文件分配表个数, 一般为 2
BPB_RootEntCnt	17	2	FAT32 文件系统中这两个字节应为 0
BPB_TotSec16	19	2	FAT32 文件系统中这两个字节应为 0
BPB_Media	21	1	如果设备是硬盘则该字节应为 0xF8, 如果设备是可移动磁盘 (如软盘) 则该字节通常为 0xF0
BPB_FATSz16	22	2	FAT32 文件系统中这两个字节应为 0
BPB_SecPerTrk	24	2	每条磁道含有的扇区个数
BPB_NumHeads	26	2	设备的磁头个数
BPB_HiddSec	28	4	隐藏扇区数
BPB_TotSec32	32	4	磁盘容量
BPB_FATSz32	36	4	文件分配表大小
BPB_ExtFlags	40	2	Bit0~3: 正在读写的文件分配表编号, 只有在对文件分配表操作失败时有效 Bit7: 0 表示将操作映射到所有的 FAT 表 1 表示只有 bit0~3 表示的那个 FAT 表有效 其余各位是保留位
BPB_FSVer	42	2	文件分配表的版本号
BPB_RootClus	44	4	指向数据区起始段的簇号, 通常为 2
BPB_FSInfo	48	2	FSInfo Sector 的地址, 通常是 1
BPB_BkBookSec	50	2	BPB 表备份的地址, 通常是 6
BPB_Reserved	52	12	保留
BPB_DrvNum	64	1	软盘为 0x00、硬盘为 0x80
BS_Reserved1	65	1	保留 (被 Windows NT 使用), 格式化硬盘的系统代码, 一般为 0
BS_BootSig	66	1	扩展引导扇区标志 (0x29),
BS_VolID	67	4	磁盘的 ID 号
BS_VolLab	71	11	磁盘的名称, 如没有名称则为 “NO NAME”
BS_FilSysType	82	8	文件系统格式, FAT16 或 FAT32

下图为硬盘 BPB 表:

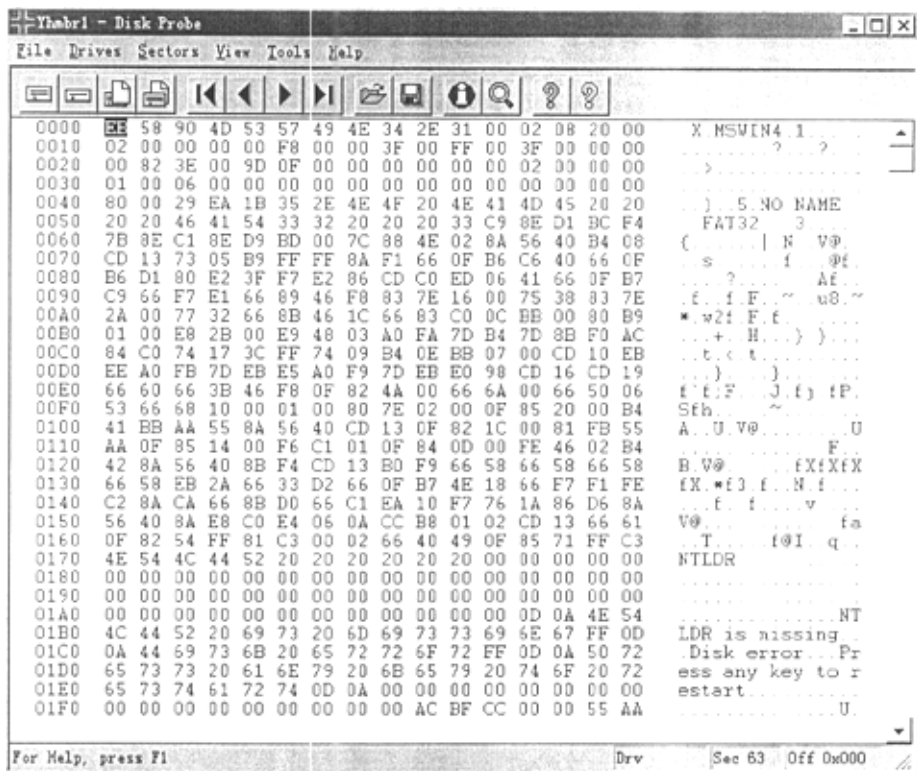


图 2.23 硬盘 BIOS

BytsPerSec=0x200 表示 512 字节每扇区; SecPerClus=0x8 表示每 8 个扇区为一簇; RsvdSecCn=0x20 表示保留扇区数为 32, 即文件分配表地址=BPB 地址+32; Media=0xF8 表明磁盘为硬盘; RootClus=0x2 表示第 0 簇、1 簇保留, 第 2 簇指向数据区起始段; BkBookSec=0x6 指明了引导区备份的地址, 备份的地址=引导区地址-6; TotSec32=0x3E8200 表明磁盘可用空间大小为 1.95G; FATSz32=0xF9D 表明每个文件分配表占 3997 个扇区; NumFATs=0x2 表明共有两个 FAT 表。

FSInfo Sector 的地址=BPB 地址+FSInfo

FSInfo 的值通常是 1。

表 2.6 FSI 扇区结构

名称	偏移	长度	
FSI_LeadSig	0	4	标志位, 值为 0x4161 5252
FSI_Reserved1	4	480	保留, 值为 0
FSI_StrucSig	484	4	标志位, 值为 0x6141 7272
FSI_Free_Count	488	4	记录系统所知道的第 1 个空簇的簇号, 如果值为 0xFFFF FFFF 则存储文件时需重新搜索空簇
FSI_Nxt_Free	492	4	指明系统在开始搜索空簇时的起始簇号
FSI_Reserved2	496	12	保留, 值为 0
FSI_TrailSig	508	4	结束标志位, 值为 0xAA55 0000

➤ 簇

在 FAT32 文件系统中储存文件的单位是以簇来划分的。例如, 硬盘每扇区容量为 512Bytes 每簇 8 个扇区, FAT32 用 4 个字节表示一个簇号, 即文件分配中每扇区可存储 128 个簇号。每个簇唯一对应 8 个扇区, 即文件分配中每扇区可表示数据空间中 512K 的内容。

可用簇	0000,0000h
坏簇	F7FF, FFFFh
已用簇	0XXX, XXXXh
文件最后一簇	0FFF, FFFFh

FAT32 在存储文件时首先在 FAT 表中找到第一个可用簇, 如果一簇的空间可以保存这个文件, 则在该簇写上结束标志表明文件只占用了一簇的空间。如果一簇不够储存这个文件, 则查找下一个可用簇并在所使用的上一簇的位置上记录所使用的下一簇的簇号, 形成一条簇号链, 所使用的最后一簇内容为结束标志。

下图是 FAT 表起始扇区内容:

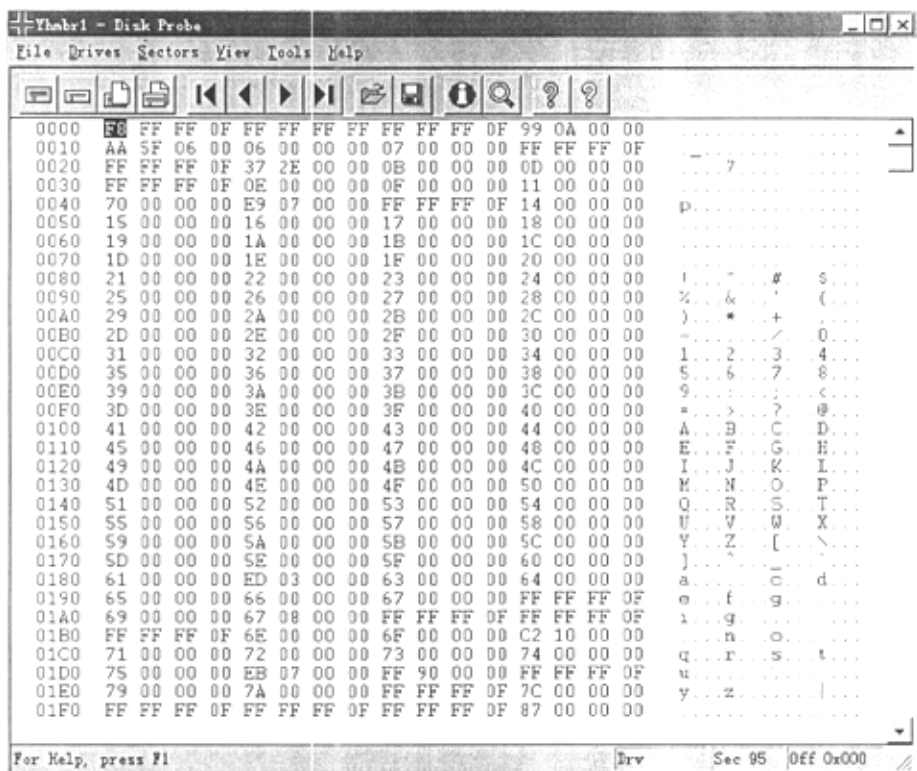


图 2.24 文件分配表起始扇区内容

第 0、1、2 簇为在硬盘格式化时自动生成的固定值，第 2 簇 0x0FFF FFFF 指向硬盘数据区起始段存储着文件的根目录。从上图可以看出第 5 簇值为 0x6、第 6 簇值为 0x7、第 7 簇值为 0x0FFF FFFF，表明有一个文件占用了 3 个簇的空间。

➤ 文件逻辑地址的计算方法

文件逻辑地址 = 数据段起始地址 + (簇号 - RootClus) × SecPerClus

数据段起始地址 = BPB 地址 + RsvdSecCn + FATSz32 × NumFATs

BPB 地址在 MBR 中获得

➤ 文件目录

Windows 既支持短文件名也长文件名，但长文件名的文件系统过于复杂，故本设计采用短文件名。短文件名文件系统每个文件名和相关信息占用 32

字节，具体内容如下：

表 2.7 短文件名格式

名称	偏移	长度	
DIR_Name	0	11	短文件名，前 8 字节是文件名，后 3 字节是扩展名
DIR_Attr	11	1	文件属性 Attr_Read_Only 0x01 Attr_Hidden 0x02 Attr_System 0x04 Attr_Volume_ID 0x08 Attr_Directory 0x10 Attr_Archive 0x20
DIR_NTRes	12	1	保留 (for Windows NT)
DIR_CrtTimeTenth	13	1	文件创建时间 (单位：毫秒)，计数间隔 2 秒 故其值域为 0~199
DIR_CrtTime	14	2	文件创建时间
DIR_CrtDate	16	2	文件创建日期
DIR_LstAccDate	18	2	最后一次读写文件的时间
DIR_FstCluHI	20	2	文件起始簇号高两字节
DIR_WrtTime	22	2	文件被更改的时间
DIR_WrtDate	24	2	文件被更改的日期
DIR_FstCluLO	26	2	文件起始簇号低两字节
DIR_FileSize	28	4	文件大小

2.4.2.3 存储控制程序流程

系统采用 DSP 完成文件管理工作，利用 FPGA 实现 DSP 于硬盘的接口。当采样率为 80k 时，数据流为 5Mbyte/S，所以选用 IO2 工作方式存盘速率设计为 7.8Mbyte/S。程序启动后，首先将硬盘 Device register 中的 SRST 置 1 使硬盘复位，在硬盘复位后读取 MBR0 和 BPB0，在 FAT 中找到 Special Sector 并检查是否有标志位 0x55FF，如果没有则说明硬盘是第 1 次被使用，如果有

则从中读取参数准备传输数据。

- 为了在每次启动程序后能迅速查找到应使用的文件名、下一个空簇的位置、正在使用的磁盘等参数，将硬盘第 3 簇中第 1 扇区标志为 Special Sector，用于储存这些参数。
- 每个磁盘被划分出 200M 空间为保留区，当发现磁盘空间不足 200M 时即显示磁盘已满，不在用当前磁盘存储数据而是使用下一个磁盘。
- 每个磁盘第 2 簇为目录区，因使用短文件名无子目录，所以当每簇为 16 扇区时目录区共可储存 256 个文件名。
- 数据文件大小固定为 40M。

如果硬盘是第一次被使用，则主机顺次读取 MBR0、BPB0、MBR1、BPB1 直到查询到最后一个硬盘分区，并将相关参数如每个分区的大小、分区的逻辑地址、每个分区数据空间起始段的逻辑地址、FAT 表逻辑地址等信息写入 Special Sector，并做一个文件的目录(32 字节)将第一个文件命名为“000.dat”。

当 FAT 中第三簇被标志为坏簇时，表明已经在硬盘上划分 Special Sector 并已存入硬盘参数。主机从 Special Sector 中读取下一个可用空簇，并计算这一簇在 FAT 中的位置并将 FAT 读入 C54 的 RAM 中，读取 Special Sector 中的文件名及相关参数并将其写入目录区。然后查询 BIO，当 BIO 为低电平时开始传输数据（参见储存板接口），传输数据时 FPGA 启动元件 trans_data，由主机发出 XF 作为启动信号。FPGA 从 FIFO 中每读出 2Bytes 向硬盘写一次，连续写 256 次，即向硬盘传输 512 字节数据。主机每发出一个 XF 下降沿 FPGA 就从 FIFO 中读取 512 字节数据并存入硬盘。

传输 40M 数据结束后，主机将根据传输的扇区数填写 FAT 表，判断硬盘空间是否小于 200M、已储存的文件数是否大于 256，如果为真则再次存储数据时将使用下一个磁盘分区。

2.4.3 存储控制接口设计

当需要将数据存入内记系统硬盘时，由采集控制板 FPGA 将数据以每帧 64 字节的格式发送至存储控制系统。硬盘接口设计存盘速度为 7.8M 字节每秒，显然采集速率与存盘速率并不匹配，因此采用一片 FIFO 作为数据缓冲

区。由于对硬盘操作过程中,需不断查询其状态,而硬盘状态改变时间又不确定,所以要选用存储深度较大的 FIFO 作为缓存。IDT 公司的 72V2101^[33] 满足设计要求,它是一款高速同步 FIFO,宽度 8bit,深度 256k。而且,72V2101 除了具备一般 FIFO 常有的写满、读空、半满等状态信号外,还具有可编程的状态信号 (Almost Full/Empty),即用户可以编程设定读出/写入多少个数据时产生这些信号,因而使用起来很灵活。另外,72V2101 还可以级联以扩展 FIFO 深度。

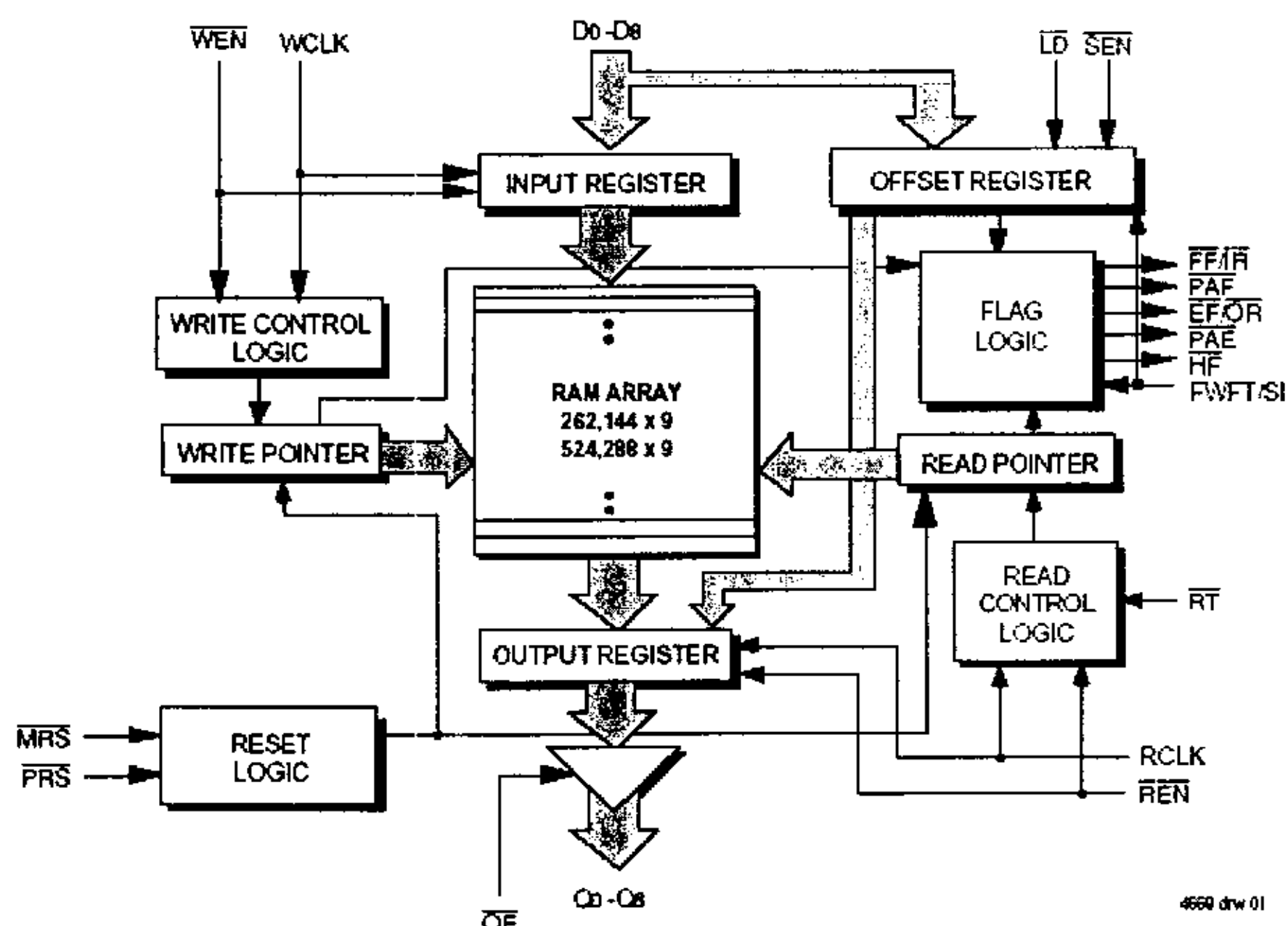


图 2.25 72V2101 结构框图

工作前首先由采集控制板 FPGA 将 FIFO 复位,需要储存数据时采集控制板 FPGA 直接将数据写入 FIFO,当存储控制板 FPGA 查询到 FIFO 的状态为非空时就输出 BIO 信号通知 DSP 启动存盘程序。DSP 先访问硬盘确定所需参数,如下一个文件应使用的文件名,保存文件的逻辑扇区号等。然后 DSP 向硬盘发出存盘命令,当硬盘响应控制命令后,DSP 通知 FPGA 可以向硬盘传输数据。存盘时 FPGA 从 FIFO 中读出数据并将数据打包,由 8bit 变为 16bit,并写入硬盘数据寄存器。当存满 40M 时,DSP 通知采集控制系统已写满一个文件,采集控制系统将 FIFO 复位完成一次存盘操作。

2.5 电源管理

浮标电源的功能是，将浮标电池组提供的系统电源通过程序的控制向浮标内各个部分提供所需要的电源；能够监测系统电池组能源消耗的情况，在电池组的输出低于标称值的时候给出报警信号；浮标出现漏水是给出报警信号。

2.5.1 电源技术指标

电源在系统的控制下应能提供七路电源输出，总体给出的电源指标如下：

- A. 控制板电源：+5V / 1A 不受控
- B. 存储板电源：+5V / 500mA 程控
- C. 硬盘电源： +5V / 2A 程控
- D. 硬盘电源： +12V / 2 A 程控
- E. 接收板模拟电源：+5V / 500mA 程控；与数字电源实现隔离

由于系统电池族是采用 1.2V 的镍锌电池串并构成的，电池组的输出如果降低到低于 1.0V 的过放会造成电池的损坏，这就要求对系统电池组进行监测。根据镍锌电池的充放电特性选择两个监测点进行监测，第一监测点对应单节电池输出降到 1.1V，第二监测点对应单节电池输出降到 1.0V。

2.5.2 电源电路原理与设计

系统要求程控各电源模块，除了水声信号模拟接收板必须使用线性稳压电源之外，其余都可以采用直流变换。

2.5.2.1 直流变换

浮标电源重大部分电源采用直流变换实现。DC-DC 直流变换器效率要比线性电源的效率，这对提高系电池组的利用率是有利的。浮标电源 DC-DC 变换采用的是美信公司的 MAX1684/5^[28]，其典型的应用电路如图一所示：

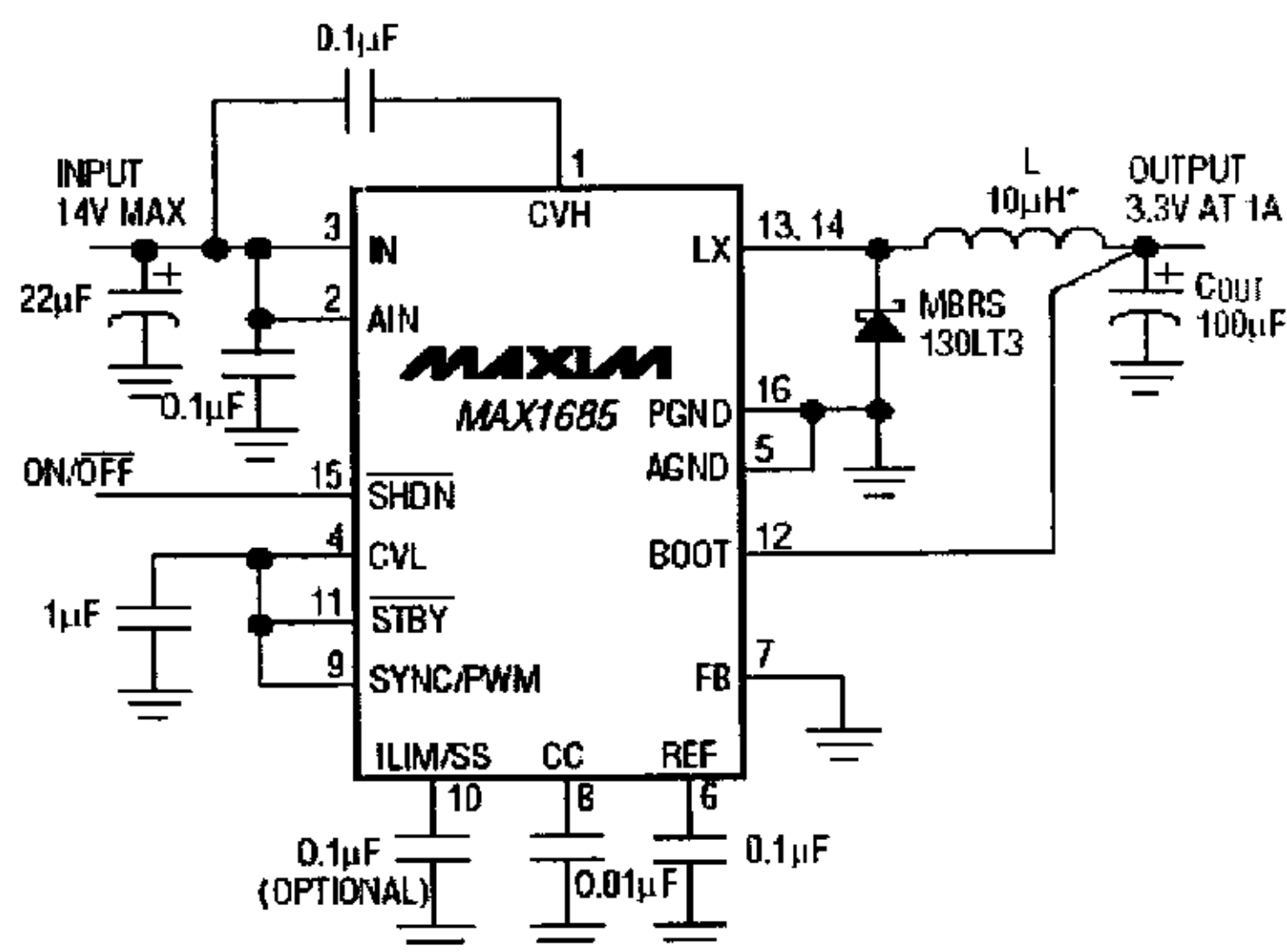


图 2.26 MAX1684/5 典型应用电路

MAX1684/5 是属于 PWM step-down 型的 DC-DC 变换器，具有片内低导通阻抗功率开关管，效率可达到 96%，而且具有四种不同的工作方式：固定频率方式，常规方式，低功率方式和关闭方式。固定频率的 PWM 模式具有良好的噪声特性；常规模式在满负荷的情况下具有最大效率；低功率模式适用于待机情况；休眠模式是彻底关闭电源使器件的功耗减少到最低的限度。MAX1684 与 MAX1685 除了频率不同之外，各项功能和技术指标完全一样。

电池组在充完电后的输出可达 1.2V，采用六节串联输出最高可达到 7.2V，MAX1684 具有较宽的输入电压范围 (2.7V-14V)，完全适应电池供电的要求。同时 MAX1684 是输出可控的，当输出控制引脚为高电平时器件电压输出，低电平器件处于休眠状态无电压输出。MAX1684 的最大输出电流为 1A，输出电压可固定也可调节。在图一中器件的输出电压的采样输入引脚 (FB) 直接接地，输出电压为 3.3V。根据需要选择输出电压可采用图二所示的方法：

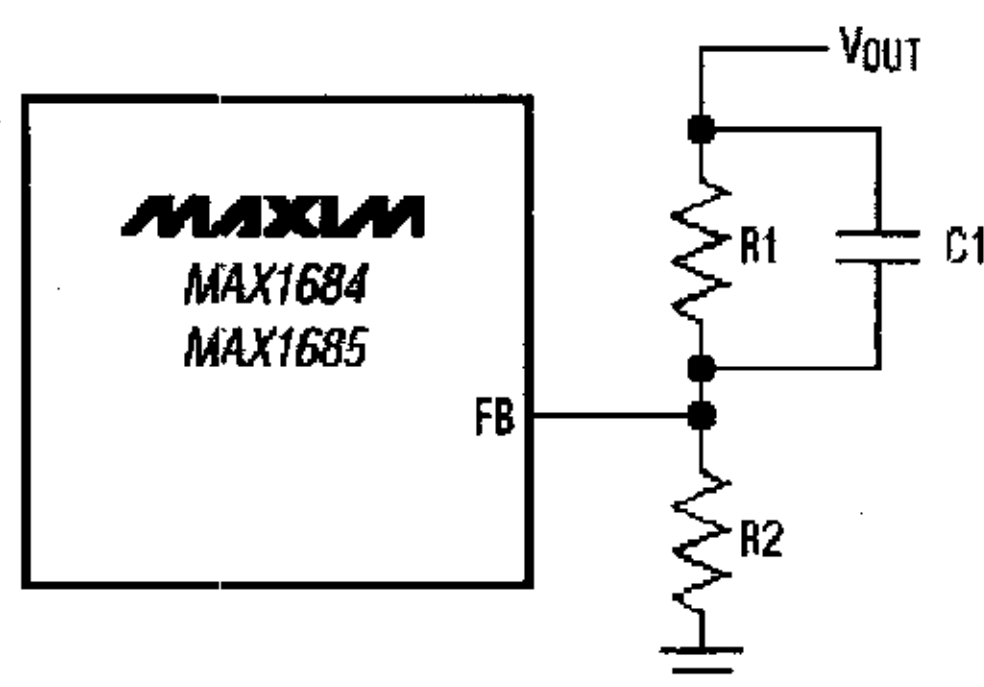


图 2.27 输出电压设定

改变电阻 R_1 、 R_2 的比例可以改变器件的输出电压值。输出电压的计算公式为： $V_{OUT}=(R_1/R_2+1)V_{FB}$ 。

其中 V 为器件内部给出的参考电压值 1.25V

在电源的输入端需加入滤波网络衰减高频噪声，常见的滤波器有穿心电容、L 型滤波器、PI 型滤波器、T 滤波器型。选用什么形式的滤波器取决于滤波器接入端阻抗的大小，其中 PI 型滤波器对高频噪声有较高的衰减，并且适用于两端阻抗较大的场合，符合本设计的要求。

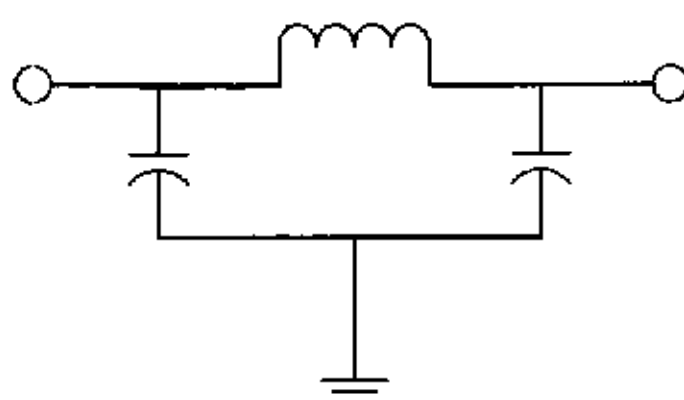


图 2.28 PI 型滤波器原理图

硬盘要求 12V 供电，额定的工作电流为 750mA。考虑到电池组的供电方便和系统的简化，我们采用升压型供电，整个数字部分可以采用一个电池组供电。这里 DC-DC 变换器采用的是 MAX1771^[29]，典型的应用电路如图所示：

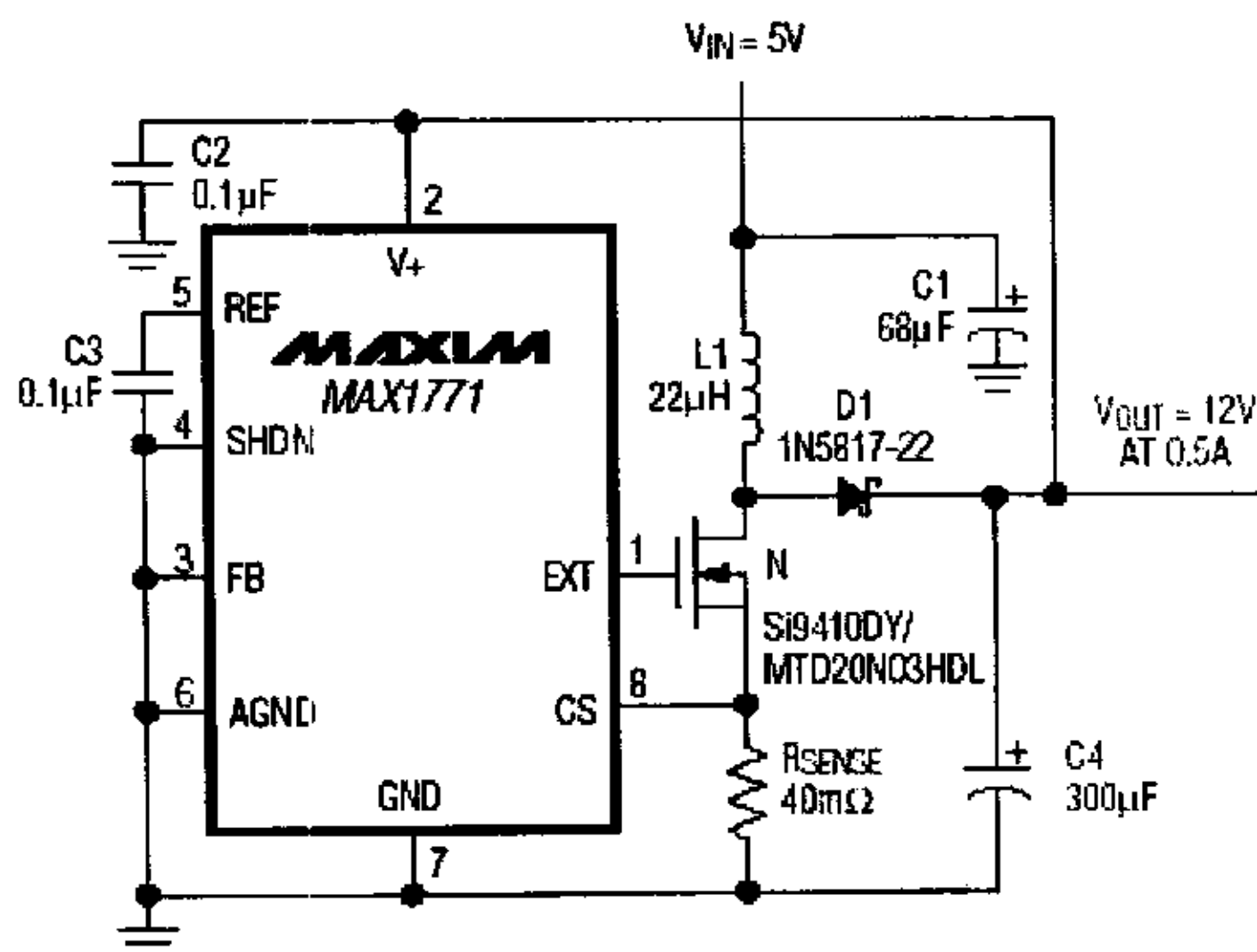


图 2.29 MAX1771 典型应用电路图

MAX1771 是 PFM step-up 型的 DC-DC 变换器, 采用 PFM 控制模式在满载的情况下可获得 92% 的效率, 最大功率可达 24W, 可满足硬盘的要求。输入电压范围为 2V-16.5V, 输出电压同样可以受控, 控制引脚为低电平时电压输出, 高电平时器件休眠并无电压输出。MAX1771 的电压可固定也可以调节, 输出电压采样输入引脚 (FB) 接地, 对应的输出电压为 12V。MAX1771 靠外围所接 MOSFET 提供功率

2.5.2.2 低压差线性稳压电源

水声信号模拟接收机对矢量水听器输出的小信号进行放大和滤波处理, 如果采用 DC-DC 直流变换器的方法供电, 由于 DC-DC 变换器产生的脉冲是属于功率型脉冲, 电源输出含有丰富的高频噪声, 而且很难滤除干净, 这样势必会对弱小的水声模拟信号产生影响, 所以必须采用线性稳压电源方法供电。

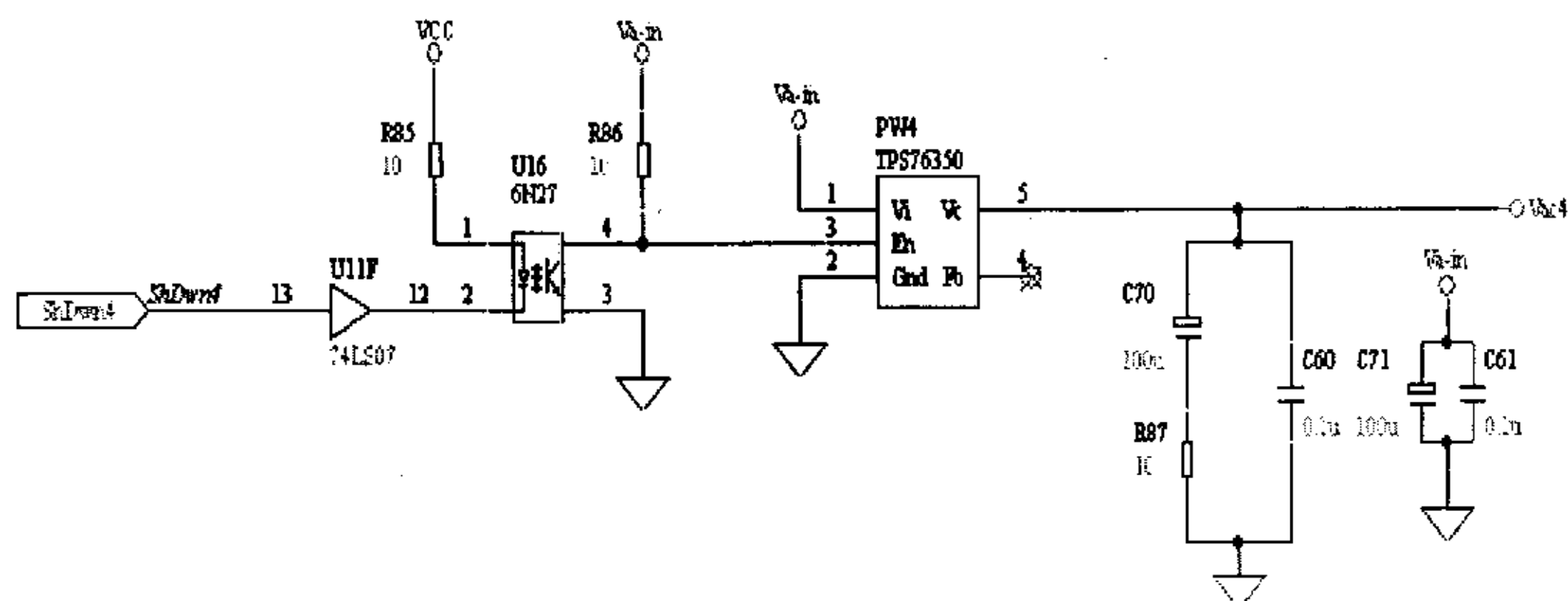


图 2.30 线性稳压电源原理框图

TPS76350^[31]是 TI 公司生产的低压差线性稳压电源 (LDO)。该器件可以最大提供 150mA 的输出电流，且最大输入电压为 10V，可满足系统电池组最大输出的要求。由于模拟电源与数字电源是两组不同的电池组，而且不共地。程序控制模拟电源输出，这就要求将供电使能信号和模拟电源隔离。光电耦合器 6N27 的作用就是将模拟地与数字地进行隔离，避免数字信号对模拟信号的干扰。

2.5.3 系统电池组监测电路原理与设计

2.5.3.1 电池能量监测原理

浮标在投放之后，系统电池组就开始放电。电池组的能源消耗必须进行实时监测，在能源消耗至下限是需要向系统提供报警，电池组的电压监测方法是很简单的，可以采用电压比较器、DA 变换器的方法。前者电路简单、监测电路功耗低；后者监测精度高、监测门限可程序设定。在浮标系统中并不要求高精度的监测，而更需要低的监测电路功耗。所以系统电池组监测电路采用比较器电路更为合适。对于电池组的监测关键是如何确定监测门限。监测门限基本上是由可复充电电池的充放特性来决定。一般来讲，镍锌可复充电电池的标称值为 1.2V，电池在充满电之后电压可高达 1.25V 左右；在放电时，电池的能源大约放掉 5% 左右电池的电压便降到标称值；在电池放电的中间阶段 80% 的能量的释放都能维持在标称值输出；在电池放电的最后阶段，

电池两端的电压已经低于标称值，释放能量约 5%左右。如果电池电压下降到 1.0V 就不能继续放电了，否则的话就会损坏电池。从而看出电池能量最小下限监测点电压值为 1.0V。次下限监测点电压值为 1.1V。在六节电池串联的条件下最小下限监测值为 6V，次下限监测值为 6.6V。这两个监测点足以满足对电池组能量消耗的监视。

2.5.3.2 电池监测电路设计

电池能量监测用电压比较器件的种类很多，有高速型、高精度型、通用型和低功耗型等。在浮标系统中，由于系统能源是采用电池供电的方式，能源是有限的，那么就要求电源不受控的电路部分的功耗要尽量低，所以电池监测电路首先要考虑的是要功耗低。一般高速型、高精度型的电压比较器功耗都比较大，不适合浮标系统的器件要求。MAX8214 是美国 Maxim 公司生产专用的电池监测用电压比较器。MAX8214 的最大电流为 33mA，功耗是很低的；在芯片内部集成了四个电压比较器；另外芯片内部还提供了一个 1.25V 参考电源。

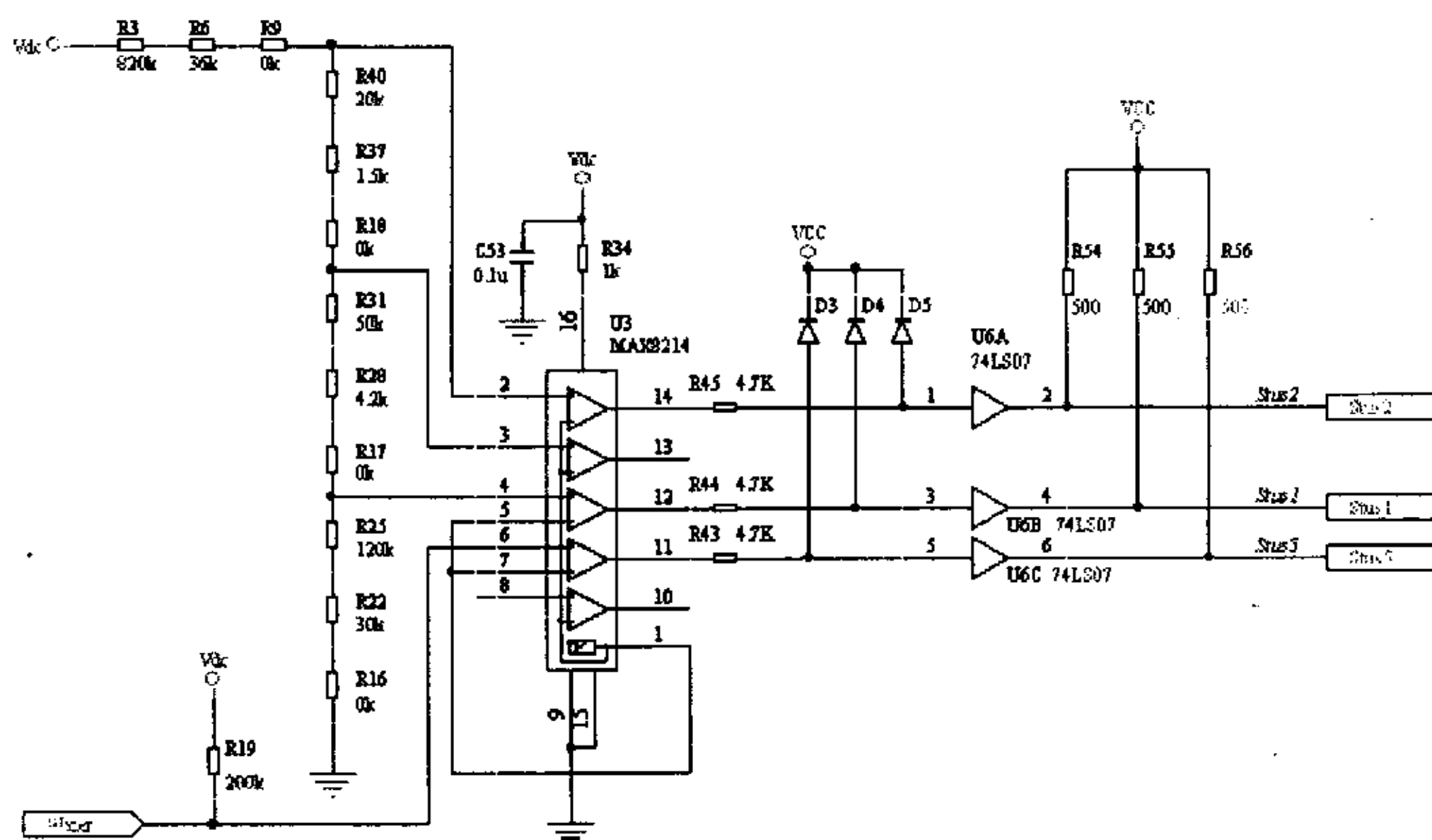


图 2.31 电压监测电路原理框图

图给出了 MAX8214 的电池电压监测的典型应用电路，在图中可以看出利用电阻串联分压网络给出了电池正常、电池电压次下限、电池电压最低下限三点电池电压的采样值，并且给出对应的三路监测结果输出。可以看出采

用 MAX8214 器件作为浮标系统电池监测用器件是比较理想的。

MAX8214 输出为 CMOS 电平，所以用二级管实现电平转换以保护驱动门。对模拟电池的监测电路与上图大致相同，区别在于输出的监测信号需用光耦隔离。

2.6 本章小结

本章详细介绍了矢量阵接收系统水下分机的结构，以及各模块的功能和设计思路。在 2.2 小节中介绍了信号调理电路的设计思路，给出了放大增益可变滤波器带宽可调的信号调理电路解决方案，并给出了通带频率特性曲线等参数。2.3 小节介绍了采集控制电路的设计，它是水下系统的核心。该部分主要为 DSP 程序和 FPGA 内核逻辑的设计，并利用 FPGA 的 EAB 资源对程序进行了优化，FPGA 采用 10K30A 时使用了 92% 的内部资源。2.4 小节讲述了内记系统的设计，详细介绍了硬盘接口协议，并给出了使用 DSP 和 FPGA 直接控制硬盘的方法。2.5 小节介绍了电源管理电路的设计，有一个低噪声的电源对于一个采集系统十分重要，因为对于交流信号来说 VCC 就相当于 GND。本节论述了数字电源和模拟电源混合设计方案，并采用光耦将控制信号隔离，避免其对模拟电源的影响。

本章是论文工作的重点，目前水下分机已经调试成功，达到设计指标的要求。

第 3 章 计算机分系统

3.1 概述

当矢量阵接收系统处于测量船工作方式时,水下设备被放置在电子仓中,利用光纤与测量船上主控计算机进行通信。测量船上主控计算机中数据接收卡和应用程序两部分组成计算机分系统。数据接收卡配置有光纤接口和 PCI 总线接口,单工工作方式。应用程序是整个系统的控制平台,用户通过应用程序监控水下设备的工作状态,如发送采集控制命令、配置接收机初始增益码和滤波器带宽、监视接收机各通道相位和幅度特性等。

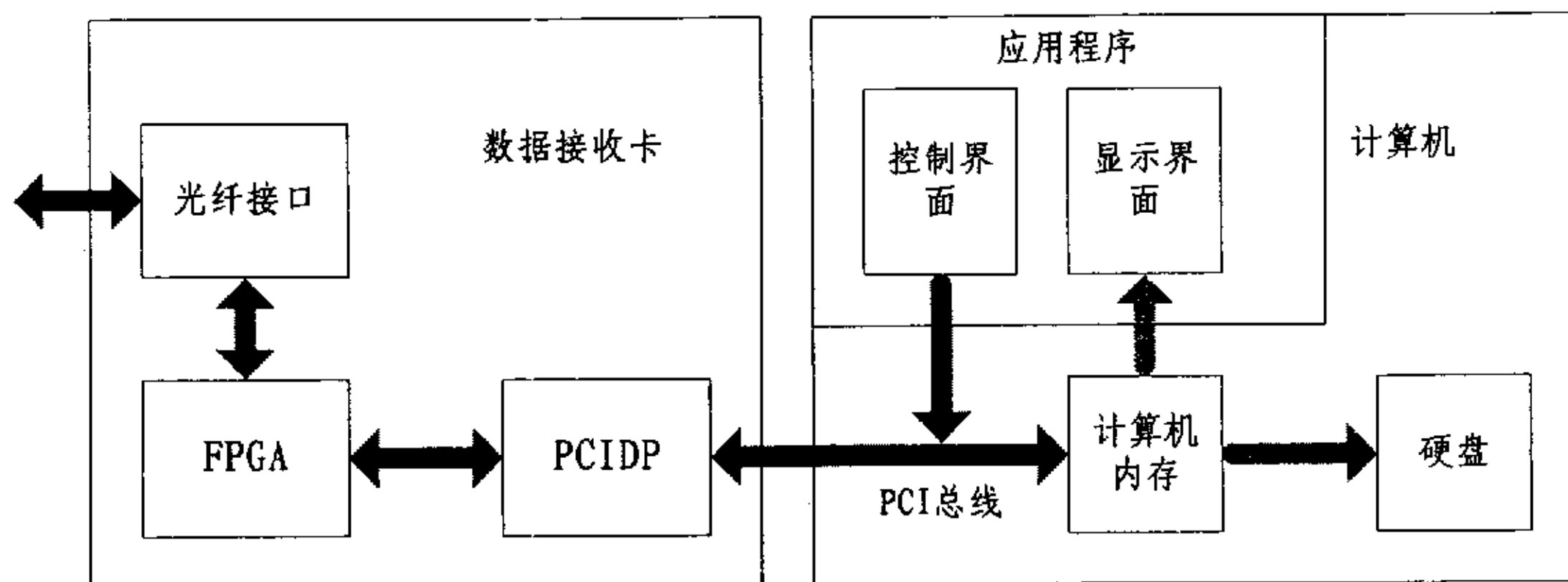


图 3.1 计算机分系统原理框图

计算机分系统由数据接收卡和应用程序两部分组成,其中应用程序分为设备驱动程序和综合显控程序两部分。整个数据接收、显示和控制的过程是这三部分相互配合的过程。数据接收卡是光纤和 PCI 总线的接口;设备驱动程序是数据接收卡和综合显控程序相互通信的媒介;综合显控程序是显示和控制的平台,用户操纵整个系统的工具。由于每一项简单操作都是这三部分配合完成的,所以作者通过这一节给出相对清晰的工作流程便于读者理解,各部分的详细内容见后续章节。

当用户打开综合显控程序时,显控程序首先访问 PCIBIOS 得到操作系统

分配给数据接收卡的资源信息。在数据接收卡上, PCI 接口芯片 PCIDP 的主要资源是 16KByte 的共享双口 RAM, 我们将这 16KRAM 映射到计算机的内存空间。数据接收卡是主机的一个 PCI 设备, 操作系统分配给数据接收卡的资源实际上就是分配给 PCI 接口芯片 PCIDP 的, 这些资源包括中断号、内存空间的物理地址和 IO 空间的物理地址, PCIDP 中双口 RAM 的地址就可表示为分配给 PCIDP 的内存空间物理地址加上 16KByte 的偏移量 (见 PCIDP 简介)。但是 Windows 工作在 32 位保护模式下, 应用程序只能依据线性地址访问内存空间, 所以我们要将物理地址转换为线性地址, 这一工作由设备驱动程序完成。于是需要将一些资源参数在综合显控程序和设备驱动程序之间相互传递, 这些参数包括中断号、消息句柄、内存空间的物理地址和线性地址和 IO 空间的物理地址和线性地址。我们将这些参数封装在一个名为 param 的结构体中, 并在一个头文件中说明这个结构。在显控程序访问 PCIBIOS 后, 调用驱动程序并通过 DeviceIoControl 函数将结构体 param 传递给驱动程序, 由驱动程序将物理地址转换为线性地址后返回给显控程序 (见参数传递小节)。在初始化过程中, 驱动程序通过内存空间的线性地址访问 PCIDP 内部寄存器对其进行设置, 并使用 VDMABuffer 类在计算机内存里开辟 2Mbyte 的缓存区, 分为大小为两块 1M 的乒乓 RAM 使用。

在显控程序发出采集命令前, 用户应先设置接收机参数, 否则将使用默认参数。此时, 首先执行 DeviceIoControl 函数, 由其下传采集命令和接收机参数; 然后显控程序创建一个线程用于存储上传的数据。在驱动程序中 OnW32DeviceIoControl 函数对 DeviceIoControl 函数产生响应, 这个函数的结构是一个分发例程。当下传的是采集命令时, 驱动程序通过 PCIDP 向 FPGA 发出一个中断 (见系统中断小节), 这个中断被系统当作一个控制信号使用, 在下传停止命令时才将其清除, 同时将接收机参数写入 PCIDP 上的双口 RAM 中。FPGA 以 PCIDP 发出的中断为程序启动的控制信号。当接收到这个中断时先从 PCIDP 的双口 RAM 中读出接收机参数并通过光纤将参数下传, 然后从光纤接收的数据写入 PCIDP 的双口 RAM 中 (见总线控制)。这 16KRAM 被分为大小为 8K 的两块乒乓 RAM, 在其中一块已满时 FPGA 向 PCIDP 发出一个中断。设备驱动程序接收到这个中断后先清除这个中断, 再设置 PCIDP 的 DMA 控制寄存器, 由 PCIDP 将双口 RAM 中的数据写入主机内存缓存区

中。当主机内存缓存区中乒乓 RAM 已满时，驱动程序向显控程序发送一个 Ring0 事件（见综合显控程序），通知显控程序读取数据。显控程序在它创建的线程中等待这个事件，并使用一个指针指向缓存区的乒乓 RAM，通过这个指针进行下一步操作。

显控程序发出采集停止命令时，驱动程序清除主机发出的中断，FPGA 停止从光纤接收数据。

3.2 计算机分系统硬件设计与实现

3.2.1 PCI 接口设计

3.2.1.1 PCIDP 简介

CY7C09449PV-AC（以下简称为 PCI-DP）是 CYPRESS 公司 PCI-DP 家族的 PCI 接口控制器之一，它提供了一个和众多流行的微处理器和 DSP 直接连接的 PCI 主/从接口。其内部的 128k-Bit 的双口静态 RAM 用作局部总线和 PCI 总线的共享 Memory。PCI-DP 允许设计者简单、低价位地完成 PCI 的接口应用设计。其主要特点如下：

- 128kBit 的共享双口 RAM
- 满足 PCI 规范 2.2 的主从接口
- 内嵌主桥能力
- 和众多微处理器直接连接
- I₂O 消息传送单元
- 单一 3.3V 电源供电，和 3.3V、5V PCI 信号环境兼容

PCI-DP 结构如下图所示：

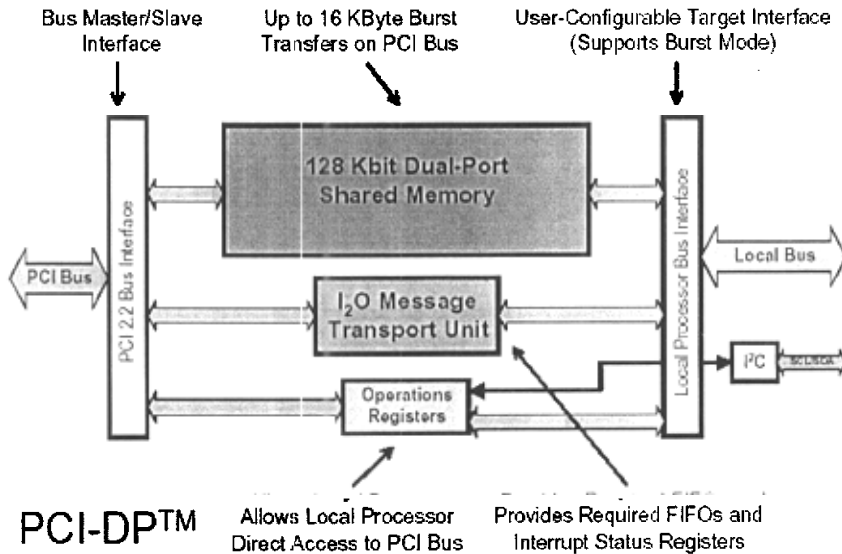


图 3.2 PCI-DP 结构框图

PCI-DP 内的最基本的资源就是 128K Bits 的双口 RAM，该 RAM 是两端分别与 PCI 总线和微处理器总线（局部总线）相连。在同一时刻，PCI 总线和微处理器总线都能访问该 RAM 以完成数据的交换。PCI-DP 满足 PCI V2.2 规范的主/从传输标准，因此，PCI-DP 能作为 PCI 总线的 Master 设备，通过 DMA（Direct Memory Access）即可将双口 RAM 的数据送入至 PC 机的物理内存中或将 PC 机的物理内存中数据送至双口 RAM 中，该过程无须 CPU 的干涉，通过 DMA Master PCI-DP 就能充分发挥 PCI 总线的 Busrt 功能，以达到最快的数据传输速率。

表 3.1 PCIDP 片上资源分配表

Memory contents	Address [14..0] Byte Offset	Size
I ₂ O Specific Registers	0x0000~0x03FF	1KB
Operation Registers	0x0400~0x07FF	1KB
Reserved	0x8000~0x1FFF	2KB
Direct Access to PCI Bus	0x2000~0x3FFF	8KB
Shared Memory	0x4000~0x7FFF	16KB

PCI-DP 的另一资源是 4 个先进先出(FIFO)存储单元, PCI 总线和微处理器总线也都能访问这 4 个 FIFO 来完成不同的传输协议。

像其它的 PCI 接口芯片一样, PCI-DP 也提供了一个 I²C 总线, 用作和串行 EEPROM 连接的接口, 通过串行 EEPROM 可加载一些特定的配置信息(见下一节)。

PCI-DP 提供的双口 RAM 是一种同步 RAM, 同步 RAM 和异步 RAM 的主要区别就是同步 RAM 有一个时钟标准来完成各个不同的时序操作。图 3.3 给出了双口 RAM32bit 接口模式的单周期读写时序:

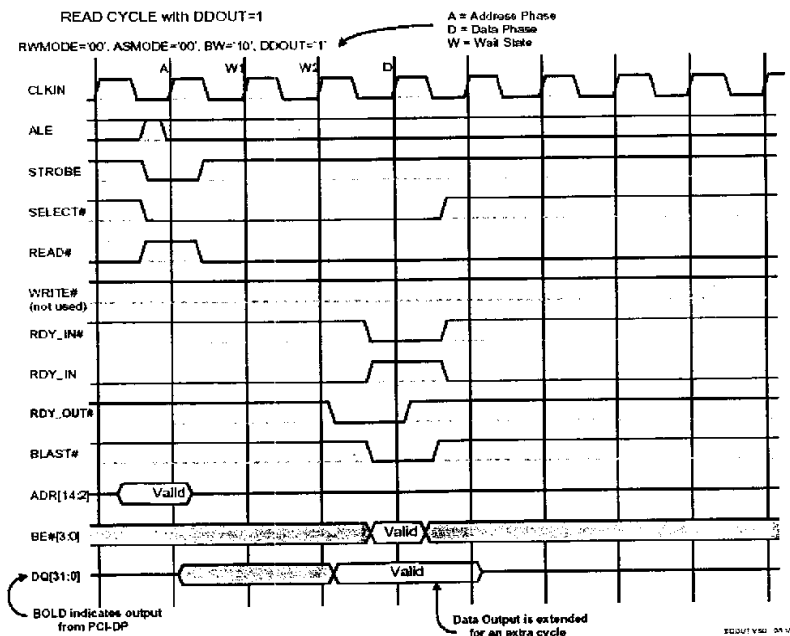


图 3.3 PCI-DP 读写时序图

从图中可看出读写时序都是以 CLKIN 为标准的(最大频率为 50MHz)。基本的局部总线操作由一个地址周期和一个数据周期组成, 多个数据周期将在下面讨论。地址周期的信号主要有 ALE、/STOBE、/SELECT、/READ、/WRITE 和地址总线 ADR[14:2], 数据周期信号主要有 /RDY_IN、RDY_IN、RDY_OUT、/BALST、/BE[3:0]和数据总线 DQ[31:0]。在地址周期里, 用 CLKIN 的上升沿采样 /STOBE、/SELECT 是否有效, 如有效, 同时也采样 /READ、

/WRITE 判断是读还是写,此时地址 ADR[14:2]也在 CLKIN 的上升沿被锁存。地址周期后就是等待周期和数据周期,当在 CLKIN 的上升沿时/RDY_IN、RDY_IN 和 RDY_OUT 都有效,进入数据周期,反之,当在 CLKIN 的上升沿时/RDY_IN、RDY_IN 和 RDY_OUT 任何一个无效时,就进入等待周期,直到它们全有效,则进入数据周期。

上面介绍了 PCI-DP 单周期的读写时序图,下图给出了双口 RAM16-bit 接口模式的数据 Burst 操作的时序图,也就是基本的局部总线操作由一个地址周期和多个数据周期组成:

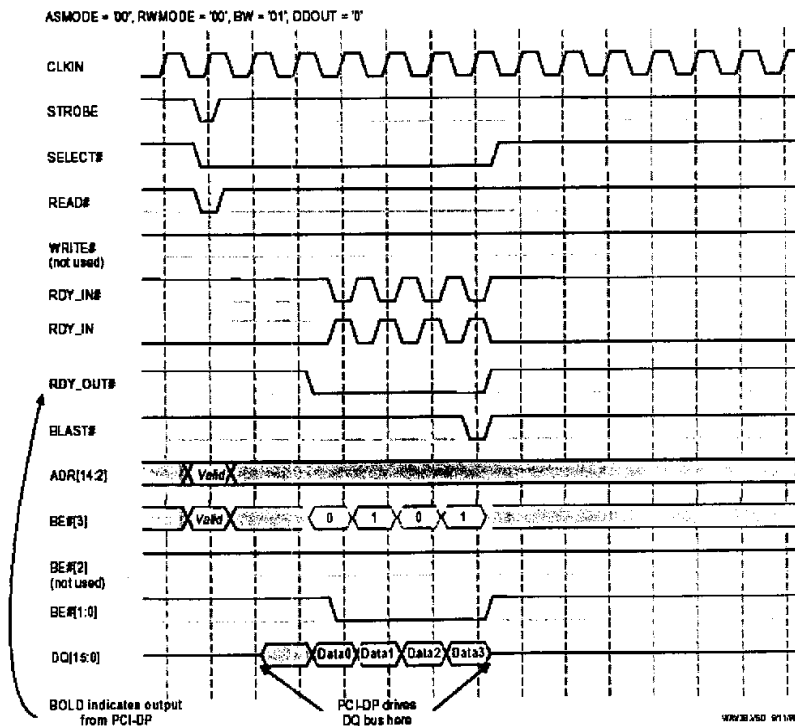


图 3.4 Burst 操作的时序图

从图中可看出,利用数据 Burst 操作,可比单周期操作更快的传输数据,数据 Burst 操作省了 3 个地址周期(图为 4 个数据的 Burst 操作)。数据 Burst 操作的地址周期和单周期操作的地址周期一样,在这就不重复说明了。其数据周期也是从 CLKIN 的上升沿采样/RDY_IN、RDY_IN 和 RDY_OUT 信号开始,如都有效则进入数据周期,反之,进入等待周期。通过判断/BLAST

信号是否有效，Burst 操作决定是否在下一个 CLKIN 周期结束。

因为 PCI-DP 的局部总线能和大多数通用的微处理器连接，因此，PCI-DP 提供不同的配置方式：8bit、16bit、32bit 的微处理器接口。PCI 总线是 32bit 的，为了充分利用该总线宽度，当和 8bit 和 16bit 的微处理器相连时，通过 BE[3:0]和局部总线配置寄存器(LBUSCFG)就可把 4 个 8bit 数据或 2 个 16bit 数据拼成一个 PCI 32bit 的数据。

3.2.1.2 PCIDP 配置

I ² C Serial Port Device 0x0 Memory Map for Auto-Configuration				
Byte 3	Byte 2	Byte 1	Byte 0	Internal Address, Byte Offset
don't care	don't care	don't care	don't care	0x00 ... 0x3F
CY7C09449PV Signature 0x48	CY7C09449PV Signature 0x37	reserved	reserved	0x40
Device ID high byte	Device ID low byte	Vendor ID high byte	Vendor ID low byte	0x44
Class Code, base class high byte	Class Code, sub-class middle byte	Class Code, programming intf. low byte	Revision ID	0x48
Subsystem Device ID high byte	Subsystem Device ID low byte	Subsystem Vendor ID high byte	Subsystem Vendor ID low byte	0x4C
MAX_LAT	MIN_GNT	Interrupt Pin, Master Enable ^[2]	don't care	0x50
Cardbus CIS Pointer high byte	Cardbus CIS Pointer low byte	Cardbus CIS Pointer high byte	Cardbus CIS Pointer low byte	0x54
reserved	reserved	reserved	reserved	0x58
reserved	reserved	reserved	reserved	0x5C
reserved	reserved	reserved	reserved	0x60
reserved	reserved	reserved	reserved	0x64
reserved	reserved	reserved	reserved	0x68
reserved	Local Bus Configuration high byte	Local Bus Configuration middle byte	Local Bus Configuration low byte	0x6C
reserved	reserved	reserved	reserved	0x70
Host Control bits [31:24]	Host Control bits [23:16]	Host Control bits [15:8]	Host Control bits [7:0]	0x74
don't care	don't care	don't care	don't care	0xFF ... 0x78

以上为 PCIDP 的初始化配置信息，这些信息保存在 E²PROM (24LC02) 里，在初始化的时候通过 I²C 总线调用。下面说以下几个要注意的地方：

偏移量 0x42-0x43：必须为 48 37，此处为 PCIDP 的标识符。

偏移量 0x44-0x45 **Vendor ID**：缺省值为 0x12be。PCI 标准化组织给 cypress 公司的代码。驱动程序的相应位置也应为此值。

偏移量 0x46-0x47 **Device ID**：缺省值为 0x3042。PCI 标准化组织给 cypress 公司的代码。驱动程序的相应位置也应为此值。

偏移量0x48 **Revision ID**: 缺省值为0x02。PCI标准化组织给cypress公司的代码。驱动程序的相应位置也应为此值。

偏移量0x49 **Class Code, programming interface.**: 0x01。表示the programming interface that supports system interrupt capability (详见PCI规范)

偏移量0x4a **Class Code, sub class**: 0x80。表示Other communications device (详见PCI规范)

偏移量0x4b **Class Code, base class**: 0x7。表示为Simple communication controllers (详见PCI规范)

偏移量0x51 **Interrupt Pin, Address**: 0x51, bits 2, 1, 0:

Master Enable, Address: 0x51, bit 7:

0x81表示允许PCI Master方式传输, 使用中断INTA#。

偏移量 0x6c-0x6e **Local Bus Configuration**: 提供初始化的 Local Bus Configuration 值。

初始化时 Local Bus Configuration 值被加载到 PCIDP 内部寄存器 Local Bus Configuration 中, 它确定了 PCIDP 局部总线的管脚工作模式, 如规定时钟上升沿采样 RDY_IN 信号、READ 信号无效、WRITE=R_W#等。

3.2.2 FPGA 程序流程

3.2.2.1 总线控制

FPGA 分为 5 个模块。光纤接收模块负责接收光纤数据, 拼成 32bit 的数据传输给数据上行传输模块。数据上行传输模块负责把数据传输给 PCIDP 和 SHARC。数据下行传输模块负责控制其它的模块, 从主机接收命令和配置信息, 并下传给接收机。光纤发送模块负责发送配置信息和命令。接口模块负责 Local Bus 接口的选通工作。

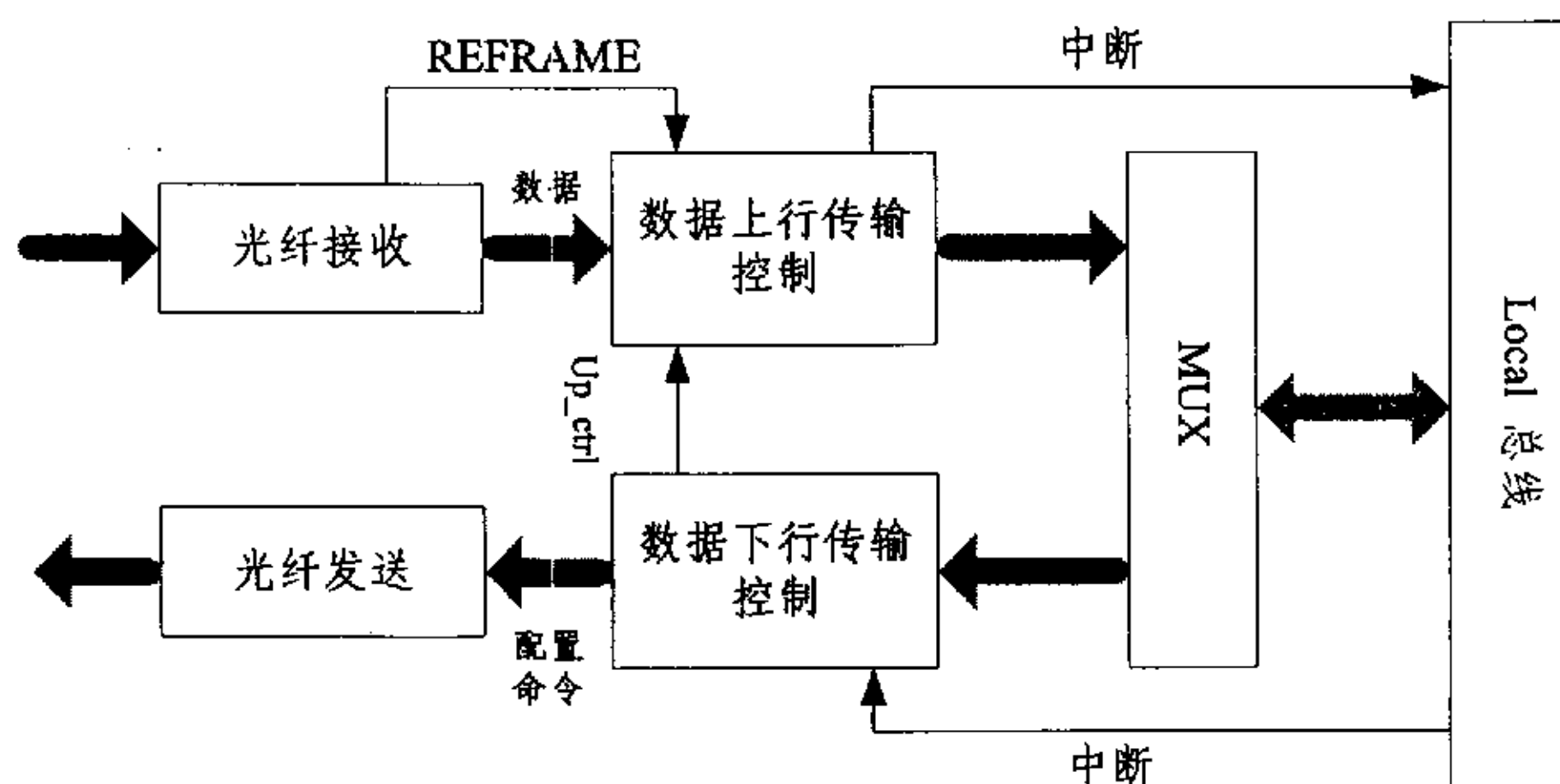


图 3.5 控制电路原理框图

当 FPGA 处于空闲时，等待主机的中断。当主机要求传输数据时，先将接收机参数存到 PCIDP 的内存中，然后触发 PCIDP 的中断。PCIDP 的中断触发时，FPGA 以这个主机中断为控制信号通知上行传输模块接管 Local 总线控制权，先从 PCIDP 的内存中读取参数，将参数下传到接收机，接收机接到参数后开始工作。当参数下传后，下行控制模块输出 up_ctrl 信号给上行传输控制模块，由上行传输控制模块接管 Local 总线控制权，接收接收机传来的数据，并传送到 PCIDP。当 PCIDP 的中断被清除时，FPGA 停止传送数据，并向接收机发送停止命令。

数据包为 64Byte，接收机每包为连续发送，发送时钟 20MByte，数据率最高为 6.4MByte。空闲时间光纤自动填充 K28.5 以保持内部锁相环同步，接收模块通过 CY7C924 的输出信号 RDY、RVS 和 SCD 判断是否有数据上传并输出 dat_signal 信号。FPGA 通过这个信号判断为新的一帧的开始，输出 REFRAME 信号到上传模块，上传模块开始新的一帧的传输。

3.2.2.2 系统中断

系统共有两个中断：一个是FPGA通过PCIDP给主机的，另一个是主机通过PCIDP给FPGA的。PCIDP中共有16Kbyte共享双口RAM，分为两块乒乓RAM每块8Kbyte。FPGA每向双口RAM中传输一帧数据（64Byte）就将计数器加1，当PCIDP中的乒乓RAM满时（计数器值为128时），发给主机一个中断。主机接到中断后，设置PCIDP的DMA控制器，将乒乓RAM中的数据传输

到主机DMA缓存区中。当用户下达开始命令时，主机发给FPGA一个中断，并且不清这个中断，使其一直有效。FPGA收到中断后，开始数据传输。当中断有效时一直传输数据。当用户下达结束命令时，主机清中断。FPGA发现中断无效时，停止传输数据。

Local to Host中断由FPGA发出，当向PCIDP的IRQ_IN管脚输出一个低脉冲时，PCIDP就向主机发出一个中断。

Host Interrupt Control and Status Register – HINT0x04E4

31	25 24	16 15	10 9	0
Interrupt Enable			Interrupt Status	

Bit	Description
25:16 Interrupt Enable	Interrupt Enables 000000000 = no interrupts are enabled (default) xxxxxxxx1 = I ₂ O Local FIFO overflow interrupt enabled xxxxxxxx1x = I ₂ O PCI FIFO overflow interrupt enabled xxxxxxx1xx = reserved - always read as 0 xxxxxx1xxx = Local to host, mailbox interrupt enabled xxxxx1xxxx = Local to host, external signal interrupt enabled xxx1xxxxx = DMA complete interrupt enabled xxx1xxxxxx = I ₂ O inbound post FIFO not empty interrupt enabled xx1xxxxxx = I ₂ O outbound post FIFO not empty interrupt enabled x1xxxxxxx = PCI target abort interrupt enabled 1xxxxxxx = PCI master abort interrupt enabled Note: All enable bits are initially cleared.
9:0 Interrupt Status	Interrupt Event Status 000000000 = no events active xxxxxxxx1 = I ₂ O Local FIFO overflow xxxxxxxx1x = I ₂ O PCI FIFO overflow xxxxxxx1xx = reserved - always read as 0 xxxxxx1xxx = Local to host mailbox xxxxx1xxxx = Local to host external signal interrupt xxx1xxxxx = DMA complete xxx1xxxxxx = I ₂ O inbound post FIFO not empty (mirror of I2OLISR[3] - read only at this address) xx1xxxxxx = I ₂ O outbound post FIFO not empty (mirror of I2OHLISR[3] - read only at this address) x1xxxxxxx = PCI target abort 1xxxxxxx = PCI master abort Note: When an event status bit is active, writing a '1' to that bit location will clear the bit except for bits 6 and 7. All event status bits are initially cleared.

上图为PCIDP的主机中断配置寄存器，主机通过向Local to host external signal interrupt位写1来清除这个中断。

Host to Local中断是通过主机写PCIDP的mailbox寄存器实现的。

Local Interrupt Control and Status Register – LINT0x04F4

31	25 24	18 15	10 9	0
Interrupt Enable			Interrupt Status	

Bit	Description
25:16 Interrupt Enable	Interrupt Enables 00 0000 0000 = no interrupts are enabled (default) xxxxxx xxx1 = I₂O Local FIFO overflow interrupt enabled xx xxxx xx1x = I₂O PCI FIFO overflow interrupt enabled xx xxxx x1xx = reserved - always read as 0 xx xxxx 1xxx = Host to Local mailbox interrupt enabled xx xxx1 xxxx = reserved - always read as 0 xx xx1x xxxx = DMA complete interrupt enabled xx x1xx xxxx = I₂O inbound post FIFO not empty interrupt enabled xx 1xxx xxxx = I₂O outbound post FIFO not empty interrupt enabled x1 xxxx xxxx = PCI target abort interrupt enabled 1x xxxx xxxx = PCI master abort interrupt enabled Note: All enable bits are initially cleared.
9:0 Interrupt Status	Interrupt Event Status 00 0000 0000 = no events active xx xxxx xxx1 = I₂O Local FIFO overflow xx xxxx xx1x = I₂O PCI FIFO overflow xx xxxx x1xx = reserved - always read as 0 xx xxxx 1xxx = Host to Local mailbox xx xxx1 xxxx = reserved - always read as 0 xx xx1x xxxx = DMA operation complete xx x1xx xxxx = I₂O inbound post FIFO not empty (mirror of I2OLISR[3] - read only at this address) xx 1xxx xxxx = I₂O outbound post FIFO not empty (mirror of I2OHISR[3] - read only at this address) x1 xxxx xxxx = PCI target abort 1x xxxx xxxx = PCI master abort Note: When an event status bit is active, writing a '1' to that bit location will clear the bit except for bits 6 and 7. All event status bits are initially cleared.

Host to Local Data Mailbox – HLDATA0x04E8

31	25 24 23	16 15	8 7	0
		Byte 1	Byte 0	

Bit	Description
24 -- 1	Interrupt to Local This bit enables the host to send an interrupt to the Local. When it is set to 1 by the host, it triggers a mailbox interrupt to the local processor. The interrupt remains active until it is cleared by writing to the Local Interrupt Control and Status Register - LINT. 0 = inactive 1 = active This bit is write only.
15:8 -- Byte 1 7:0 -- Byte 0	Data byte Two bytes of data that can be written by the host and read by the local processor.

首先，将设备中断控制和状态寄存器中mailbox interrupt enable位置1，使能这个中断。当将mailbox寄存器中interrupt to local位置1时，就向设备发出了这个中断。对中断控制和状态寄存器相应位写1来清中断。

3.2.2.3 DMA 配置

主机与 PCI 卡的数据通信都是利用 DMA 方式来完成的。DMA 传输时，PCIDP 为 master 模式，利用 PCIDP 的 DMA 控制器将数据从 PCIDP 中的 RAM

中传输到主机的 DMA 缓冲区中。因为 DMA 传输的特性，需要在主机内存中开一块物理地址连续的空间，并且可以知道它的物理地址和线性地址，用一般的内存分配指令无法满足这一要求，所以我们用 VToolsD 中的一个类 VDMABuffer 来分配 DMA 缓冲区。在程序初始化的时候创建类进行分配。（详见后续章节）

3.3 计算机分系统软件设计

3.3.1 设备驱动程序设计

在设计和使用 PCI 设备时，经常要在 PC 机的软件中访问和控制硬件设备，但 Windows 操作系统（包括 Windows95/98、Windows NT、Windows 2000）为了保证系统的安全性、稳定性和可移植性，对应用程序访问硬件资源加以限制，这就要求设计设备驱动程序^[35]以实现 PC 机的软件对 PCI 设备的访问。所以本设计中要想利用 PC 机的应用程序人机界面完成对整个采集系统的监控工作，就要通过调用驱动程序来完成。因此计算机分系统的另一个重要部分是驱动程序的开发，本设计驱动程序用 VtoolsD 开发，适用于 Windows95/98 操作系统。

Windows 下的驱动程序不仅仅包括物理设备的驱动程序，也包括为文件系统等非物理设备编写的虚拟设备驱动程序。为了简化问题，下面只讨论硬件物理设备的驱动程序。

3.3.1.1 读写内存

Windows 工作在 32 位保护模式下，保护模式与实模式的根本区别在于 CPU 寻址方式上的不同，这也是 Windows 驱动程序设计中需要着重解决的问题。Windows 采用了分段、分页机制这样使应用程序产生一种错觉，好像程序中可以使用非常大的物理存储空间。这样做最大的好处就是一个程序可以很容易地在物理内存容量不一样的、配置范围差别很大的计算机上运行，编程人员使用虚拟存储器可以写出比任何实际配置的物理存储器都大得多的程序。通

过分段机制，系统由虚拟地址产生线性地址。再通过分页机制，由线性地址产生物理地址。线性地址被分割成页目录(Page Directory)、页表(Page Table)和页偏移(Offset)三个部分。当建立一个新的Win32进程时，操作系统会为它分配一块内存，并建立它自己的页目录、页表，页目录的地址也同时放入进程的现场信息中。当计算一个地址时，系统首先从CPU控制器CR3中读出页目录所在的地址，然后根据页目录得到页表所在的地址，再根据页表得到实际代码/数据页的页帧，最后再根据页偏移访问特定的单元。硬件设备读写的是物理内存，但应用程序读写的是虚拟地址，所以存在着将物理内存地址映射到用户程序线性地址的问题。

从物理地址到线性地址的转换工作也是由驱动程序来完成的。在VtoolsD中这个函数的定义如下：

```
PVOID MapPhysToLinear (CONST VOID * PhysAddr, DWORD nBytes,
    DWORD Flags);
```

其中第一个参数PhysAddr就是要映射的内存的物理地址的起始位置，而nBytes是内存区域的长度，Flags必须设置为0。这个函数返回的就是这段物理地址映射的线性内存地址。如果指定的内存不能存取，函数将返回FFFFFFFFH。

比如要映射物理内存ED000000H开始的4096个字节，可以这样做：

```
(PCHAR
*)PointerToPage=(PCHAR)MapPhysToLinear ((PVOID) 0xED000000,
    4096, 0);
```

而将PointerToPage传递给调用驱动的用户程序，在用户程序中使用

```
DWORD *pFIF0BodyBase=(DWORD*)PointerToPage;
```

而这个 PFIF0BodyBase 指针就可以象普通的指针一样进行读写操作，而通过对这个指针的操作就可以实现对物理内存 ED000000H 进行读写。

3.3.1.2 参数传递

应用层和设备驱动程序打交道主要是通过 CreateFile、ReadFile、WriteFile 和 DeviceIoControl 等 Win32 API 来进行的。其中 ReadFile、WriteFile 对应固定的命令代码，而 DeviceIoControl 携带的命令代码可由

程序员设定，使用时灵活方便，所以本设计使用 DeviceIoControl 函数实现应用程序和驱动程序的消息传递。“命令代码”我们可以看作为一种消息，由 Win32 API 传递到驱动程序，驱动程序根据它的值进行相应的处理，执行这些代码所对应的具体的操作。

```

BOOL DeviceIoControl(
    HANDLE hDevice
    DWORD dwIoControlCode // 应用程序调用 VxD 的命令代码
    LPVOID lpInBuffer // 应用程序传递给 VxD 的数据缓存地址
    DWORD nInBufferSize // 应用程序传递给 VxD 的数据缓存字节
    LPOVID lpOutBuffer // VxD 的返回数据所存放的缓存地址
    DWORD nOutBufferSize // VxD 的返回数据所存放的缓存字节数
    LPDWORD lpBytesReturned // VxD 实际返回数据的字节数
    LPOVERLAPPED lpOverlapped // 同步时置为 NULL
);
    
```

在应用程序调用设备驱动程序时，驱动程序要从应用程序获得设备所用的中断号，端口地址的范围、存储器的物理地址等信息。这些信息一般被封装在一个结构体中，在设备初始化时由 DeviceIoControl 函数传递给驱动程序，lpInBuffer 为结构体的首地址，nInBufferSize 为结构体长度。

在 VtoolsD 中，与 DeviceIoControl 函数相对应的是 OnW32DeviceIoControl 函数，其参数 pDIOCParams 为指向 IOCTLPARAMS 结构的指针。

```

typedef struct tagIOCTLParams{ PCLIENT_STRUCT dioc_pers; VMHANDLE
dioc_hvm;   DWORD   dioc_VxdDDB;   DWORD   dioc_IOCTLCode;   PVOID
dioc_InBuf;   DWORD   dioc_cbInBuf;   PVOID   dioc_OutBuf;   DWORD
dioc_cbOutBuf;   PDWORD dioc_bytesret;   OVERLAPPED* dioc_ovrlp;   DWORD
dioc_hDevice;   DWORD dioc_ppdb;} IOCTLPARAMS, *PIOCTLPARAMS;
    
```

其中 dioc_IOCTLCode 为应用程序调用 VtoolsD 的命令代码，VtoolsD 根据它的值进行相应处理。后面的六个参数分别与前面介绍的 DeviceIoControl() 函数中的参数相对应。

在 VtoolsD 中 OnW32DeviceIoControl() 函数的形式为一分发历程，其格式如下：

```

DWORD pcidpDevice::OnW32DeviceIoControl(PIOCTL_PARAMS pDIOCPParams)
{
    switch(pDIOCPParams->dioc_IOCTLCode)
    {
        case DEVICEIOC_POST_PCIDPRES: //get related parameter from ring3
            .....
    }
}

```

其中 `DEVICEIOC_POST_PCIDPRES` 为自定义的命令代码，由应用层程序中 `DeviceIoControl` 函数传递。应注意的是，在 Win98 系统中命令代码值 0~100 为系统保留，在 Win2000 中系统保留的命令代码值为 0~800，自定义的命令代码值应大于系统保留值。

3.3.1.3 系统 DMA

VtoolsD 提供类 `VDMABuffer` 和 `VDMACHannel` 支持系统 DMA。类 `VDMABuffer` 针对 DMA 缓存区而设计；类 `VDMACHannel` 实现对某个 DMA 通道的虚拟化。

本设计设备驱动程序创建一个叫 `pcidpbuf` 的类，他由 `VDMABuffer` 的类派生而来。初始化后在主机的主存里开辟一个 2MByte 的，物理地址连续的缓存区，并将缓冲区分为两个大小为 1MByte 的乒乓 RAM。定义一个 `pcidpbuf` 类的对象，然后调用该类的构造函数开辟缓存区。构造函数的形式如下：

```

VDMABuffer::VDMABuffer(
    PVOID regionAddress,    // 缓存区的线性地址
    DWORD regionSize);     // 缓存区的长度

```

类返回缓冲区的基地址，设备驱动程序将它转换为线性地址传回给应用程序。应用程序利用线性地址来访问缓冲区，而 DMA 则通过物理地址访问。

3.3.1.4 中断的设置与响应

对中断的设置、响应与调用应该在驱动程序中完成。PCI 设备驱动程序应当从 PCI 配置寄存器的中断寄存器 (INTLN) 和中断引脚寄存器 (INTPIN) 中获得有关中断的信息。如在 Windows 98 中，VPICD 服务用来管理所有硬件中断事件。PC 机的硬件中断需要确定硬件中断的 IRQ，对一个特定的 IRQ

中断源, VPICD 或者提供缺省的中断处理函数, 或者允许其它 VtoolsD 重载中断处理函数。

VtoolsD 提供两种硬件中断类: VHardwareInt 类和 VSharedHardwareInt 类。类 VHardwareInt 实现对某个 IRQ 端口的虚拟化, 并处理该 IRQ 端口上的硬件中断, 对其进行实时处理。一个 VxD 一旦控制了某个 IRQ, 其他 VxD 便不能在对其虚拟化。类 VSharedHardwareInt 是类 VHardwareInt 的派生类。不同之处在于 VSharedHardwareInt 允许一个 IRQ 被多个 VxD 虚拟化, 即多个 VxD 共享一个硬件中断。另一个不同点是, 类 VSharedHardwareInt 的中断通知事件处理函数 OnSharedHardwareInt() 不是 VOID 型, 而是 BOOL 型函数。

本设计驱动程序中断相应函数从类 VSharedHardwareInt 派生。进入中断响应函数后查询 PCIDP 中主机中断状态寄存器, 如果是 FPGA 发出的 Local to Host 中断, 则首先清除该中断, 然后配置 PCIDP 的 DMA 控制器, 将 PCIDP 中双口 RAM 内的数据写入在主机内存开辟的缓存区中。当有 1M 字节数据存入主机内存时, 就向主机发出一个 Win32 事件通知主机将内存中数据读出。

3.3.2 综合显控程序

系统选用 VC++ 开发了一个采集监控程序, 我们称之为综合显控台。其主要功能为:

- 发送采集控制命令
- 配置接收机初始增益码和滤波器带宽
- 监视任意通道状态
- 监控任意两通道的幅度和相位差
- 完成系统自检

显控程序基于 Cview^[34]类编写, 在视窗内绘制两个显示窗口, 显示采集数据的波形, 并在工具栏内添加若干按钮用于采集控制。应用程序首先访问 PCIBIOS 来获得程序初始化 PCI 板所需要的各种信息, 如操作系统分配给数据接收卡的 IO 物理地址、内存空间的物理地址和中断资源等。然后将这些信息传输给设备驱动程序, 由设备驱动程序来初始化 PCIDP 内各寄存器, 清除中断和创建 DMA 缓冲区, 并将物理地址转换为线性地址回传给应用程序。

有两种方法适用于 VxD 对 Win32 应用程序的通信，它们是：一、使用 APC (asynchronous procedure call) 异步过程调用；二、使用 WIN32 事件。二者均依赖 VxD “唤醒” 一个 Win32 应用线程。异步过程调用比较简单，使用 WIN32 事件相对复杂一点，但它的效率要高于异步过程调用。

本程序使用 WIN32 事件在 VxD 和应用程序之间通信。在 VxD 中使用 VWIN32_SetWin32Event(hDMAEventR0) 函数发送一个事件，在应用程序中用 WaitForSingleObject(hDMAEventR3, INFINITE) 函数响应这个事件。首先，创建一个接收数据的线程，在线程函数中使用 WaitForSingleObject(hDMAEventR3, INFINITE) 函数等待驱动程序发送的 R0 事件。每当有 1Mbyte 数据存入主机 DMA 缓存区时，驱动程序就发送一个 R0 事件给应用层程序，当应用层程序接收到这个事件时，就从缓存区中将数据读出。

3.4 本章小结

在这一章节中，作者介绍了基于 PCI 总线的数据接收卡的具体解决方案。在本章概述中论述了接口芯片、FPGA、设备驱动程序和应用层程序是如何相互配合工作的。在 3.2 小节中介绍了接口芯片 PCIDP 的内部资源和工作方式，并给出了其详细的配置信息。PCIDP 只能作为从设备，本小节给出了使用 FPGA 控制 PCIDP 的设计思路。在设备驱动程序一节给出了使用 VtoolsD 开发 VxD 驱动程序的方法和应注意的问题，并给出了一个具体程序的主要框架。综合显控程序一节中介绍了如何使用显控程序监控整个系统。

由于时间关系，本设计只开发了应用于 Win98 的虚拟设备驱动程序，应用于 Win2000 的 WDM 驱动程序没有完成，这有待于在今后的工作中继续努力，以完善这一系统。

结 论

本论文的主要研究内容是针对矢量水听器的程控数据采集系统硬件设计。该系统由水下分机和水上分机两部分组成，水下分机和水上分机通过光纤进行远距离高速数据通信。论文给出了采集系统硬件的详细解决方案。

其中水下分机由预处理电路、采集控制、存储控制和电源管理电路四部分组成。为了降低电噪声的影响，系统采用电池供电，各模块均采用低功耗设计并有独立的电源开关。数字部分和模拟部分使用光耦电路隔离，将数字电路对模拟电路的干扰减至最低。采集控制和存储控制以数字信号处理器（DSP）为控制器，使用可编程逻辑阵列（FPGA）实现接口电路和时序电路的设计。现代微电子技术和 EDA 技术的发展使得可编程 ASIC 的性能和易用性飞速提高，本文较详细地阐述了以 FPGA 为代表的可编程 ASIC 在本系统中的应用，使用 FPGA 可以将对电子线路的设计转化为对应用程序的设计，这样不仅极大地提高了系统设计效率，而且使设计者摆脱了大量的辅助性工作，将精力集中于创造性的方案与概念的构思上，并为将来对系统功能进行扩展提供了方便。

水上分机由数据接收卡和综合显控程序两部分组成。数据接收卡配置有光纤接口和 PCI 总线接口，单工工作方式。应用程序是整个系统的控制平台，用户通过应用程序监控水下设备的工作状态，如发送采集控制命令、配置接收机初始增益码和滤波器带宽、监视接收机各通道相位和幅度特性等。

本系统接收通道为 24 路，即最多扩展为 6 元阵，其中每 4 路为一组，对应一个矢量水听器；滤波器频带可调，总工作频段为 50Hz~25KHz。本论文所设计的采集系统，达到了预期的指标，满足应用的要求。

在设计时各电路板接口未预留足够的保留管脚，对扩展系统功能产生了障碍。显控程序界面还比较简单，功能也不够完善。内记系统每次固定存储 40M 数据，还不能实时存储任意长度的数据流。这些不足都有待于在今后的工作中完成，使之成为一个更加实用、可靠、完善的数据接收系统。

参考文献

- [1] A.V.奥本海姆, R.W.谢弗著, 董士嘉译. 数字信号处理. 科学出版社, 1981
- [2] 曾繁泰, 陈美金 编著. VHDL 程序设计. 清华大学出版社, 2001
- [3] 林敏, 方颖立 编著. VHDL 数字系统设计与高层次综合. 2002
- [4] 宋新见. 水下目标跟踪硬件设计. 哈尔滨工程大学硕士学位论文, 2001
- [5] 周敏东. 水下目标模拟器设计研究. 哈尔滨工程大学硕士学位论文, 2002
- [6] 袁孝康 编著. 自动增益控制与对数放大器. 国防工业出版社, 1987
- [7] 德 FRIEDHELM SCHMIDT 著, 韩永彬, 袁潮著. SCSI 总线 和 IDE 接口: 协议、应用和编程. 中国电力出版社, 2001
- [8] 宋万杰, 罗丰, 吴顺君. CPLD 技术及其应用. 西安电子科技大学出版社, 2000
- [9] 赵曙光, 郭万有, 杨颂华. 可编程器件原理、开发与应用. 西安电子科技大学出版社, 2000
- [10] ADS8320 ANALOG-TO-DIGITAL CONVERTER Datasheet. 1999
- [11] DAC7611 DIFITAL-TO-ANALOG CONVERTER Datasheet. 1999
- [12] SST39VF040 Multi-Purpose Flash Datasheet. 1999
- [13] Evaluating Power for Altera Devices. 1999
- [14] Flex 10k Embedded Programmable Logic Family. June 1999
- [15] TMS320C54xDSP Enhanced Peripherals Reference Set. 1999
- [16] TMS320C54xDSP Mnemonic Instruction Set Reference Set. 1999
- [17] TMS320C54xDSP Applications Guide Reference Set. 1999
- [18] TMS320C54xDSP Assembly Language Tools User's Guide. 2000
- [19] TMS320C54xDSP DSKplus User's Guide. 1999
- [20] TMS320C54xDSP CPU and Peripherals Reference Set. 1999

- [21] TMS320VC5402 and TMS320UC5402 Bootloader Datasheet. 1999
- [22] 童诗白. 模拟电子技术. 上海: 人民教育出版社. 1983
- [23] 柯南. Protel 98 电脑辅助电路图设计. 北京: 中国铁道出版社. 1999
- [24] 伍丁红. 增强并行口 EPP 以及高速并行口 A/D 转换器的设计. 电子技术应用. 1998
- [25] 张义和 Protel for Windows 实践技巧 科学出版社 1997
- [26] 阎石 数字电子技术基础 北京: 清华大学电子学教研组 1981
- [27] CY7C924DX 200-Mbaud HotLink Transceiver Datasheet. 2001
- [28] MAX1685 Step-Down Converters Datasheet. 2001
- [29] MAX1771 Step-Up DC-DC Controller Datasheet. 2002
- [30] IL388 High Performance Liner Optocoupler Datasheet. 2001
- [31] TPS76350 LOW-DROPOUT LINER REGULATORS Datasheet. 2001
- [32] OPA2345 OPERATIONAL AMPLIFIERS Datasheet. 2000
- [33] IDT72V2111 SUPERSYNC FIFO Datasheet. 2001
- [34] (美) Richard C. Leinecker & Tom Archer 著. Visual C++6 宝典. 电子工业出版社. 1999
- [35] (美) Art Baker & Jerry Lozano 著. Windows 设备驱动程序设计指南. 机械工业出版社. 2001

攻读硕士学位期间发表的论文和取得的科研成果

致 谢

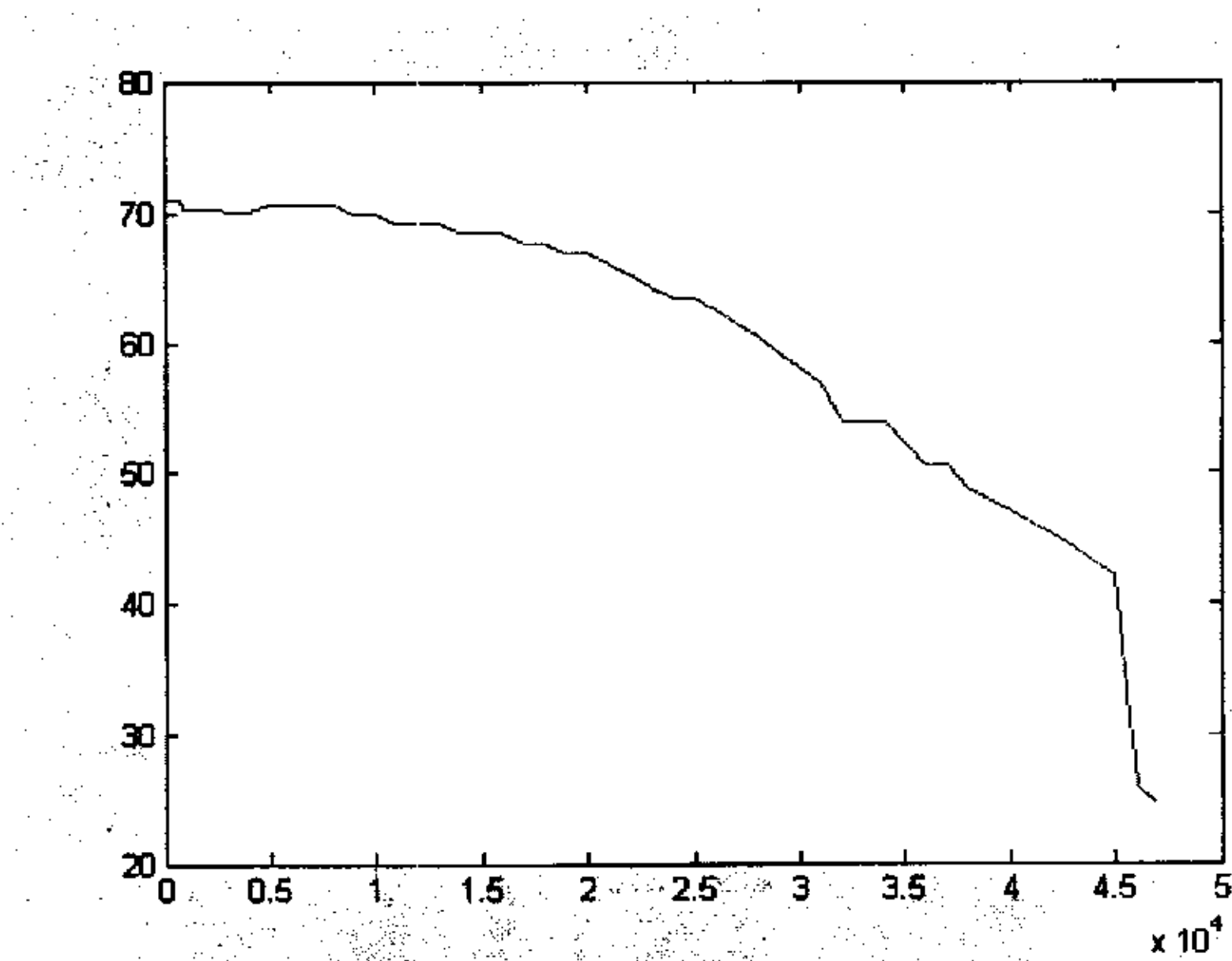
本论文是在蔡平教授的悉心指导下完成的。蔡老师特别注重工作方法、工作方向的指导，使我的工作能力有很大的提高。蔡老师谦逊的品格、和蔼的态度和严谨的工作作风对我产生了深远的影响，令我终身受益。在此作者衷心感谢蔡平老师在两年半之中对我的细心指导和辛勤培养。

本论文的研究内容并不是完全由作者完成的，而是一个集体努力的结晶。作者在工作过程中得到了王逸林讲师、周敏东同学、马俊同学、刘海波同学和王静宇同学的帮助，论文中也凝结了他们的心血和汗水，在此向他们表达我诚挚的谢意。

感谢多年来默默支持我的家人，他们对我的关怀和希望永远是我前进的动力。也感谢我的母校—哈尔滨工程大学，在这里留下了我人生最珍贵的一段记忆。

附 录

A 预处理电路滤波器幅频特性曲线



参考通道幅频特性曲线，信号频率 20Hz~41kHz。

B 预处理电路各频段相位和幅度特性参考表

信号频率 20Hz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	-6.2425	5.8024	0.0518	6.2943
2	-3.1158	1.6297	-0.7726	2.3432	14	2.2755	0.9854	-1.4102	3.6857
3	-2.6190	1.2083	-0.4200	2.1990	15	-2.0529	0.8427	1.2699	3.3227
4	-5.4340	2.3748	-0.7242	4.7097	16	-3.9942	4.8491	-2.3099	1.6844

5	-0.3327	1.2764	0.4685	0.8011	17	-0.6465	0.2619	0.2501	0.8966
6	-1.6151	0.9992	-0.0836	1.5315	18	-2.2233	1.6063	-2.3803	0.1570
7	-0.2086	0.1772	-0.4156	0.2070	19	-0.2213	0.7833	0.3116	0.5330
8	-3.4806	1.4886	-0.8712	2.6093	20	-4.5474	2.4710	-2.5759	1.9715
9	-3.0740	1.6459	2.0996	5.1736	21	-2.5258	2.0289	-0.6648	1.8610
10	-4.4578	5.2202	-2.3551	2.1027	22	-2.4234	2.2025	-2.7420	0.3186
11	-6.2526	3.9994	0.2611	6.5137	23	-2.1625	0.8492	-0.5634	1.5991
12	-4.9741	4.5502	-2.2438	2.7303	24	-3.4827	2.7938	-2.5931	0.8896

信号频率 30Hz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	-4.3829	1.1599	-0.5144	3.8685
2	-2.6316	0.3287	-0.6477	1.9839	14	1.9789	0.2507	-1.4305	3.4094
3	-2.3629	0.2326	-0.3502	2.0127	15	-2.1957	0.3504	1.5730	3.7687
4	-4.6578	0.4772	-0.3819	4.2758	16	-2.3129	1.1013	-2.7449	0.4320
5	-0.8415	0.4424	0.8610	1.7025	17	-0.6561	0.1904	0.5400	1.1961
6	-2.5288	1.1624	0.5619	3.0906	18	-1.6022	0.3245	-2.2005	0.5984
7	-0.6223	0.1184	-0.2834	0.3388	19	-0.3709	0.2983	0.7311	1.1021
8	-3.0818	0.3741	-0.5162	2.5655	20	-3.2398	0.4928	-2.4608	0.7791
9	-3.4815	0.6775	2.6819	6.1633	21	-1.7679	0.4243	-0.6389	1.1290
10	-2.7419	1.1649	-2.7559	0.2144	22	-1.7892	0.4907	-2.6808	0.8916
11	-5.2135	0.6495	-0.0084	5.2050	23	-1.7618	0.2090	-0.3186	1.4432
12	-2.8338	0.7973	-2.4293	0.4045	24	-2.4690	0.5462	-2.5848	0.2155

信号频率 40Hz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	-3.6587	0.3139	-0.6584	3.0003
2	-2.1298	0.0559	-0.5296	1.6002	14	1.5255	0.1586	-1.5909	3.1164
3	-1.9732	0.0658	-0.2479	1.7253	15	-1.8359	0.1259	1.7793	3.6152
4	-3.7428	0.1016	-0.2218	3.5210	16	-1.9411	0.2662	-3.0046	1.0635
5	-0.6641	0.1312	1.0759	1.7401	17	-0.5203	0.1054	0.5511	1.0714
6	-1.1444	0.5143	1.1419	2.2863	18	-1.2543	0.0806	-2.1897	0.9354
7	-0.5949	0.0509	-0.1858	0.4091	19	-0.2763	0.1287	0.7806	1.0569
8	-2.3022	0.1076	-0.3950	1.9072	20	-2.4310	0.1032	-2.4587	0.1977
9	-2.9148	0.1948	3.0298	5.9446	21	-1.4436	0.1175	-0.7071	0.7365
10	-2.3696	0.2836	-2.9925	0.6229	22	-1.4831	0.1316	-2.7029	1.2198
11	-4.2128	0.2071	-0.0594	4.1535	23	-1.3858	0.0944	-0.3106	1.0752
12	-2.6087	0.2841	-2.7598	0.2526	24	-2.0534	0.1712	-2.6284	0.5749

信号频率 50Hz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	-2.5981	0.9225	0.5039	3.1020
2	-0.0865	0.2164	0.5425	0.6290	14	0.0984	0.1473	-0.5646	0.6630
3	-0.4426	0.0408	0.0326	0.4753	15	-0.4158	0.6469	1.6365	2.0523
4	-1.6648	0.2867	0.3644	2.0292	16	-2.4424	0.5582	-0.7146	1.7278
5	0.8800	0.5734	0.7193	0.1607	17	-0.9891	0.1693	0.7672	1.7562
6	0.1530	0.6467	0.7903	0.6372	18	1.5366	0.2719	0.0088	1.5279
7	0.4078	0.2568	0.1346	0.2732	19	0.7044	0.1359	0.6152	0.1022
8	-0.2445	0.4956	1.4489	1.6934	20	-0.1900	0.0412	-0.1074	0.0912

9	0.6788	0.4710	1.5438	0.8650	21	-0.9652	0.1482	0.7911	1.7562
10	-3.0203	0.9150	-0.9617	2.0586	22	1.2641	0.1981	-0.0008	1.2650
11	-2.8108	0.5515	0.6586	3.4693	23	-0.7835	0.1517	0.5865	1.3700
12	-2.4552	0.6139	-0.7339	1.7213	24	1.3395	0.7487	0.6198	0.7198

信号频率 100Hz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	-1.6589	0.7737	0.3718	2.0308
2	-0.7007	0.1632	0.5906	1.2913	14	0.0341	0.0993	-0.5268	0.5609
3	-0.4188	0.1179	0.0017	0.4205	15	-0.7398	0.5771	1.7751	2.5149
4	-0.9015	0.2505	0.5101	1.4116	16	-1.5935	0.4158	-0.7252	0.8683
5	0.0442	0.6113	0.7403	0.6962	17	-0.7490	0.2343	0.8769	1.6259
6	0.4405	0.5362	1.0456	0.6051	18	0.8732	0.1779	-0.1501	1.0233
7	-0.2038	0.2751	0.1788	0.3827	19	0.1133	0.1849	0.5159	0.4027
8	-0.1504	0.3905	1.4946	1.6450	20	0.1860	0.2446	-0.3222	0.5082
9	0.0032	0.4098	1.5682	1.5650	21	-0.3817	0.1480	0.7328	1.1146
10	-1.8946	0.6517	-0.9532	0.9414	22	0.8142	0.1687	-0.0267	0.8409
11	-1.7363	0.3493	0.7937	2.5300	23	-0.5612	0.1658	0.6543	1.2155
12	-1.4241	0.3652	-0.7204	0.7037	24	0.3691	0.4068	0.3750	2.5394

信号频率 200Hz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	-0.3499	1.1514	-1.7009	0.8901
2	-0.5350	0.2501	0.9096	1.4132	14	0.0620	1.1737	-0.7312	0.7877
3	-0.2813	0.2018	0.0272	0.3132	15	-0.6885	1.0643	3.5194	3.8997

4	-0.5813	0.2819	1.0836	1.5635	16	-0.5580	0.6139	-1.6842	0.9250
5	0.4092	0.6325	2.5922	2.6595	17	-0.4114	0.6354	1.5186	1.7869
6	-0.0443	0.0711	2.4925	2.2202	18	0.2856	0.4722	-0.2433	0.5485
7	-0.3671	0.0732	1.0855	1.2640	19	-0.0558	0.8632	0.8802	0.7935
8	-0.3983	1.5006	2.8498	2.9560	20	0.0238	0.1077	-0.6426	0.6008
9	-0.3665	0.3421	2.9490	3.0949	21	-0.0637	0.9234	0.4219	0.4871
10	-0.5005	0.4988	-2.6359	1.7943	22	0.2421	0.2115	-0.2173	0.4650
11	-0.7156	0.1355	0.4562	1.3085	23	-0.3629	1.1690	1.1496	1.3846
12	-0.4312	0.0940	-1.5583	0.9754	24	0.0405	0.2551	0.2243	0.1751

信号频率 500Hz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	-0.0641	0.3513	-1.3756	0.8977
2	-0.1191	0.1921	0.9858	1.0405	14	0.1358	0.3387	-0.7644	0.8450
3	-0.1335	0.1213	0.0638	0.1767	15	-0.2683	0.3545	3.2596	3.2853
4	-0.1205	0.0671	1.0819	1.0771	16	-0.0648	0.2520	-1.5790	1.3219
5	-0.1320	0.2900	2.3074	2.2056	17	0.0974	0.1865	1.4284	1.2258
6	0.1143	0.0538	2.1946	1.7488	18	0.1360	0.1159	-0.2417	0.3719
7	-0.1900	0.0660	0.9984	1.0270	19	0.3174	0.3124	0.8628	0.5441
8	-0.0186	0.5511	2.6246	2.4168	20	0.0118	0.0505	-0.5942	0.5427
9	-0.1708	0.0781	2.7649	2.7273	21	0.2891	0.2838	0.4456	0.1988
10	-0.1437	0.1757	-2.4172	1.9791	22	0.0490	0.1586	-0.0917	0.1728
11	-0.2944	0.0978	0.5400	0.8718	23	0.1561	0.4610	1.0538	0.8346
12	-0.0241	0.0792	-1.5836	1.4144	24	0.0006	0.1257	0.3585	0.3580

信号频率 1KHz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	-0.1188	0.2772	-0.9945	1.0222
2	0.3362	0.2191	0.9488	0.6630	14	0.0363	0.2199	-0.7721	0.7693
3	-0.1729	0.1062	0.0406	0.1966	15	-0.1441	0.2416	2.9523	3.0772
4	0.3106	0.1511	0.9782	0.6997	16	-0.0153	0.1828	-1.4517	1.4761
5	0.0513	0.1521	2.0213	2.0211	17	0.4896	0.1405	1.2918	0.8377
6	0.5740	0.0908	1.8895	1.3701	18	0.1745	0.1369	-0.2439	0.5992
7	-0.0772	0.0979	0.8507	0.9595	19	0.5201	0.2008	0.8133	0.3562
8	0.3467	0.3418	2.4134	1.9964	20	0.0120	0.1790	-0.5555	0.5919
9	-0.2049	0.1249	2.5132	2.7096	21	0.5555	0.1865	0.4847	0.1265
10	-0.2729	0.2102	-2.1557	2.0419	22	0.1050	0.1395	-0.0984	0.2603
11	-0.1665	0.0887	0.4365	0.5657	23	0.4953	0.3009	0.9568	0.4975
12	0.0563	0.1101	-1.4287	1.5083	24	0.0754	0.0802	-0.0913	0.2483

信号频率 2KHz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	0.3149	0.3021	0.1740	0.2612
2	0.7958	0.3450	0.7812	0.0570	14	-0.1064	0.3450	-0.5839	0.4346
3	-0.0565	0.3580	0.0481	0.1252	15	-0.3643	0.4535	1.9750	2.2667
4	0.7893	0.1776	0.5825	0.3630	16	0.2026	0.2215	-0.8508	1.0975
5	-0.1926	0.4195	0.9873	0.9643	17	1.0379	0.2183	0.9963	0.0988
6	0.9972	0.2037	1.0417	0.2208	18	-0.1289	0.2649	-0.1676	0.5153
7	-0.1978	0.1131	0.3644	0.3753	19	0.8383	0.3044	0.7770	0.2207
8	0.7309	0.4464	1.7289	0.8784	20	0.1002	0.4419	-0.2874	0.5066

9	-0.7530	0.2031	1.7806	2.3629	21	1.3574	0.3926	0.7358	0.3532
10	-0.0340	0.2417	-1.1743	0.9213	22	0.1634	0.4125	0.1930	0.1197
11	0.0023	0.2116	0.6936	0.8132	23	0.8348	0.3858	0.7622	0.0782
12	-0.0209	0.3234	-0.8161	1.0815	24	0.2657	0.1563	-0.0292	0.2623

信号频率 5KHz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	0.0258	0.3268	0.3315	0.3057
2	2.8908	0.3194	0.9838	1.9069	14	-0.2423	0.3185	-0.6937	0.4514
3	-0.3327	0.1580	0.0233	0.3560	15	-0.3242	0.1365	2.0040	2.3282
4	2.6848	0.3409	0.9070	1.7778	16	0.0546	0.1946	-0.6812	0.7358
5	0.3719	0.2963	1.2018	0.8299	17	2.9834	0.2377	1.0882	1.8952
6	3.3532	0.3561	2.2356	1.1176	18	0.2079	0.2187	-0.0723	0.2803
7	-0.2460	0.2092	0.3454	0.5915	19	2.5459	0.3671	0.5306	2.0153
8	2.7027	0.2317	2.0736	0.6290	20	0.2428	0.1000	-0.1202	0.3630
9	-1.0944	0.2598	1.6704	2.7648	21	3.2512	0.2281	0.8865	2.3647
10	-1.0903	0.4320	-1.0058	0.1452	22	0.2348	0.1357	-0.2811	0.5159
11	-0.0198	0.1644	0.7441	0.7638	23	2.9244	0.2426	0.4161	2.5083
12	0.0785	0.3506	-1.0568	1.1353	24	1.2505	0.2095	2.0882	0.8378

信号频率 10KHz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	0.4460	0.2722	-0.6466	1.0926
2	0.4592	0.2644	-0.0826	0.5418	14	-1.0832	0.3014	-1.4487	0.3818
3	-0.3690	0.2795	0.2370	0.6060	15	-1.0891	0.2506	2.6371	3.7262

4	-0.1131	0.2715	0.5579	0.6710	16	-1.1754	0.4053	-2.9006	1.7252
5	-0.0810	0.4337	1.3295	1.4105	17	0.0408	0.3137	0.8786	0.8378
6	0.0737	2.6678	4.2196	4.1459	18	0.5592	0.8520	-1.9048	2.4640
7	-0.6761	0.2142	0.1611	0.8372	19	-0.5632	0.2148	1.3882	1.9513
8	0.6803	0.4059	0.1892	0.4911	20	-0.0268	0.4676	-2.1915	2.1647
9	-1.9952	0.3413	4.3995	6.3948	21	-0.2773	0.4856	-0.4389	0.3099
10	-2.6359	0.4066	-2.6167	0.5653	22	-0.2965	0.8414	-2.4164	2.1199
11	-0.7376	0.3375	0.2559	0.9935	23	-0.2072	0.3822	0.2163	0.4821
12	-0.5444	0.3074	-2.2318	1.7374	24	0.3383	0.5936	-2.0812	2.4195

信号频率 15KHz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	0.8430	0.0886	-2.0492	2.8922
2	1.2936	0.1194	0.1088	1.1848	14	-1.2016	0.2028	-1.4035	0.2802
3	0.0502	0.1325	0.6059	0.5557	15	-1.1103	0.1258	3.8978	5.0081
4	0.1650	0.1206	1.2201	1.0551	16	-1.1636	0.1403	-3.9101	2.7464
5	0.1615	0.0779	2.1506	1.9891	17	0.2659	0.1543	1.1649	0.8990
6	3.0169	0.9054	4.7876	1.8000	18	1.3040	0.2848	-1.8172	3.1212
7	-0.8983	0.1118	0.7588	1.6571	19	-0.4401	0.1527	2.2814	2.7215
8	1.8869	0.2764	1.0454	1.0880	20	0.7646	0.1086	-2.0783	2.8429
9	-2.5920	0.1960	6.7296	9.3216	21	0.3348	0.1652	-0.6890	1.0237
10	-2.9888	0.1387	-3.3748	0.8111	22	0.5357	0.2391	-2.3043	2.8400
11	-1.6385	1.3638	-0.2855	1.3530	23	0.1006	0.1855	0.8306	0.7300
12	-0.5389	0.1884	-3.0928	2.5539	24	1.3864	0.2906	-1.7027	3.0891

信号频率 20KHz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	0.9157	0.2838	-2.5674	3.4831
2	2.1432	0.3001	-0.4206	2.5638	14	-1.7547	0.4338	-1.4272	0.3842
3	0.5177	0.3147	0.7617	0.2687	15	-1.2045	0.3275	4.0788	5.2833
4	0.4249	0.2599	0.9335	0.5292	16	-1.6512	0.6007	-4.2435	2.5923
5	-0.2080	0.1825	2.0299	2.2379	17	0.0042	0.2166	0.8957	0.8915
6	5.5266	1.0924	5.1310	1.5053	18	1.2364	0.5783	-2.6837	3.9201
7	-0.9282	0.2823	0.9546	1.8828	19	-0.5251	0.2609	2.3228	2.8478
8	3.2611	0.3733	0.8153	2.4458	20	0.9097	0.4214	-2.7562	3.6659
9	-2.7431	0.2992	7.1852	9.9283	21	0.3133	0.4833	-1.1580	1.4712
10	-3.5358	0.5172	-3.4099	0.7111	22	1.2379	0.8071	-2.7923	4.0303
11	-3.7832	3.0529	-0.6411	3.1421	23	0.0948	0.3827	0.6723	0.5775
12	-1.1573	0.4978	-3.1604	2.0031	24	2.3959	0.8934	-2.6801	5.0760

信号频率 25KHz

通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)	通道	相位均值(度)	相位均方差(度)	幅度差均值(%)	幅度均方差(%)
1	0	0	0	0	13	1.3304	0.1132	-1.7447	3.0752
2	2.2663	0.1645	-1.0192	3.2855	14	-1.8967	0.1197	-1.1021	0.7946
3	0.3866	0.1477	0.4053	0.1400	15	-1.4390	0.0632	3.3327	4.7718
4	0.5890	0.0604	0.4171	0.4273	16	-1.3656	0.1437	-3.3895	2.0239
5	-0.7900	0.0649	1.4110	2.2010	17	-0.1220	0.0962	0.7310	0.8530
6	9.9140	1.1225	1.3970	8.5170	18	1.3097	0.3532	-2.6701	3.9798
7	-1.3768	0.1194	0.5397	1.9165	19	-0.5770	0.0709	1.8461	2.4231
8	3.6463	0.3312	0.1097	3.5366	20	1.7151	0.1530	-2.8398	4.5549

哈尔滨工程大学硕士学位论文

9	-3.6076	0.1525	5.9188	9.5264	21	0.0784	0.2775	-1.0577	1.13
10	-4.2706	0.2516	-2.8122	1.4584	22	1.5971	0.2614	-3.1375	4.73
11	-2.6103	0.6916	0.1839	2.7942	23	0.2575	0.0980	0.3106	0.32
12	-0.8970	0.1236	-2.4926	1.5956	24	2.8843	0.3874	-3.0900	5.97