

# 四川大学

## 《DSP 原理与技术》期末设计报告



题	目：	DSP 实时音频信号处理系统的设计与实现
学	院：	电子信息学院
专	业：	信号与信息处理
姓	名：	崔 敖、李尚文
学	号：	2014222050111、2014222050113
任 课 教 师：		何培宇
成	绩：	
评	语：	

# 目 录

第 1 章 实验目的及方案设计 .....	3
1.1 实验目的 .....	3
1.2 方案设计 .....	3
1.2.1 总体方案设计 .....	4
1.2.2 TMS320C54x 的自举加载及 HPI 接口 .....	4
1.2.3 TLV320AIC23 音频编解码器及 McBSP 接口 .....	9
第 2 章 电路原理及程序设计 .....	17
2.1 电路原理设计 .....	17
2.1.1 系统供电设计 .....	17
2.1.2 DSP 基本控制及中断、调试接口设计 .....	19
2.1.3 MCU、编码/解码器电路设计 .....	20
2.1.4 接口与外部连接件 .....	22
2.1.5 PCB .....	22
2.2 程序设计 .....	24
2.2.1 程序开发流程 .....	24
2.2.2 MCU 向 DSP 自举程序 .....	24
2.2.3 DSP 信号处理程序 .....	25
2.2.4 上位机程序 .....	26
第 3 章 实验结果及总结 .....	28
参考文献 .....	29
附录 A 电路原理图 .....	30

附录 B	PCB 设计 .....	36
附录 C	程序代码.....	38

## 第1章 实验目的及方案设计

本课程设计的目的是以 DSP 为核心构建一套完整的音频信号处理系统。本章对该系统的功能要求进行了详细的说明，并提出了具体设计方案。最后，本章对方案中所设计的相关技术、接口、通信协议等进行了详细的说明。

### 1.1 实验目的

《DSP 原理与技术》是一门理论与实践并重的课程。本实验的目的在于将所学知识与数字信号处理技术结合，以 DSP 芯片为核心设计一个音频处理系统，并在其上实现简单滤波算法。

本设计功能要求如下：

- 1、音频处理系统上电后，无需用户干预，默认将输入语音信号直通输出；
- 2、用户可使用上位机软件通过 USB 接口对音频处理系统进行控制，实现预置的 FIR 低通、带通、高通滤波器及其他功能；
- 3、支持功能扩展、支持在线调试、用户中断输入、外部程序分支控制、麦克风输入、耳机\扬声器输出、线路输入\输出等功能。

### 1.2 方案设计

本文采用自顶向下的方式进行方案设计，即首先给出一个切实可行的整体方案，并将方案划分为各个模块。最后再分别对各个模块进行更细化的设计。

### 1.2.1 总体方案设计

本设计的总体方案如图 1-1 所示。待处理音频信号首先输入音频编解码芯片，经芯片中 AD 模块采样后，再通过 McBSP 接口将音频数据传入 DSP 中进行处理。处理后的数据再次通过 McBSP 接口传入音频编解码芯片，最终利用芯片中 DA 模块将数据转换为模拟信号后输出。图中 MCU 负责通过 HPI 接口向 DSP 加载程序，并利用 I/O 口模拟 SPI 通信协议控制音频编解码芯片的工作模式。为了实现使用者对系统的控制，本设计中需要编写 PC 端上位机程序，通过串行口与 MCU 进行通信。MCU 根据从 PC 端传来的不同控制指令，向 DSP 加载不同的程序，从而实现了系统功能的多样化。

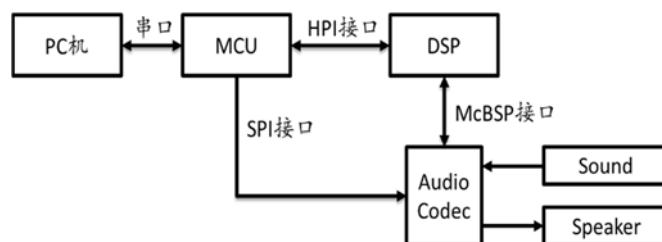


图 1-1 系统总体方案框图

本设计中各模块所用芯片选型如表 1-1 所示。

描 述	型 号	概 览
DSP	TMS320VC5402	100 MIPS, 16K×16-Bit DARAM, Timer, McBSP, HPI ...
MCU	C8051F320	25 MIPS, 16kB Flash, On-chip USB, 25 I/O port, SPI
Audio Codec	TLV320AIC23	High performance ADC/DAC, support SPI/McBSP

表 1-1 芯片选型

### 1.2.2 TMS320C54x 的自举加载及 HPI 接口

TMS320VC5402 DSP 芯片（以下简称为 5402）内部具有 4K 字的掩膜式 ROM，其中内容如表 1-2 所示。由于这种 ROM 一经出厂是不可更改的，因此在实际开发中，往往将 DSP 程序放置在片外的存储器

中,并在复位时与 DSP 内部的 Bootloader 配合完成 5402 程序的加载,这种加载方式叫做自举加载。

地址范围	描 述
0xF000h – 0xF7FFh	保留 (Reserved)
0xF800h – 0xFBFFh	自举加载程序 (Bootloader)
0xFC00h – 0xFCFFh	256 字 $\mu$ 律压扩查找表 ( $\mu$ -law expansion table)
0xFD00h – 0xFDFFh	256 字 A 律压扩查找表 (A-law expansion table)
0xFD00h – 0xFDFFh	256 字正弦函数查找表 (Sine look-up table)
0xFE00h – 0xFEFFh	保留 (Reserved)
0xFF00h – 0xFF7Fh	中断向量表 (Interrupt vector table)

表 1-2 TMS320VC5402 ROM 中内容

在硬件复位期间,如果 C54X 系列 DSP 的  $MP/\overline{MC}$  引脚为高电平,表示 DSP 设置为微处理器工作方式,则从外部程序存储器 0FF80h 起执行用户程序;若  $MP/\overline{MC}$  为低电平,表示 DSP 设置为微型计算机工作方式,则从片内 ROM 的 0xF800h 起执行 Bootloader 程序。其工作过程如图 1-2 所示。

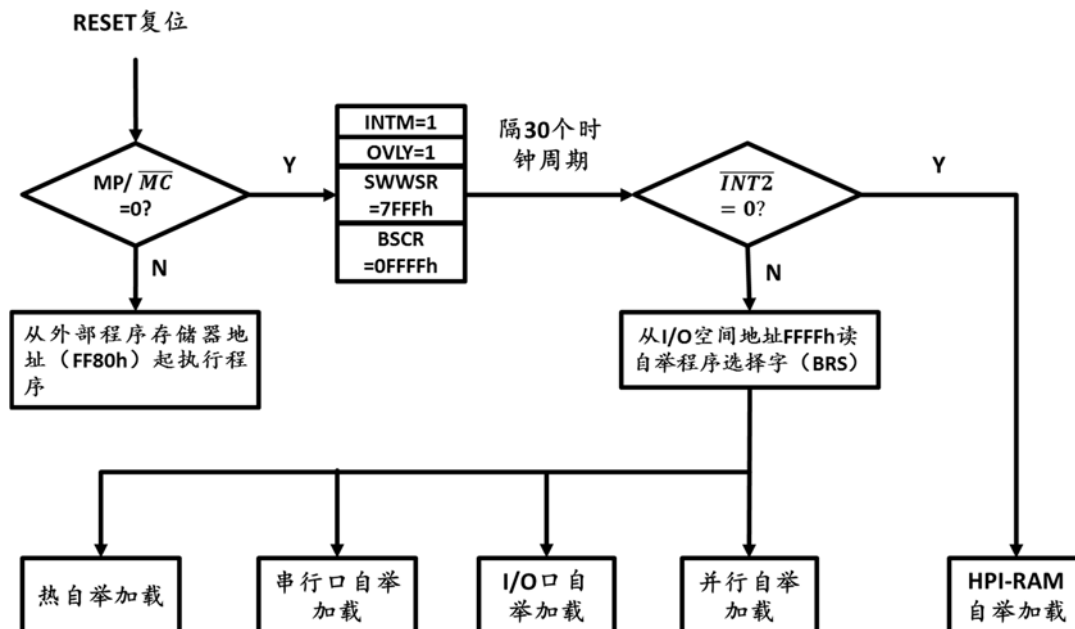


图 1-2 自举加载流程图

从图 1-2 可以出, DSP 支持 HPI、并口、I/O 口、串行口和热自举

共 5 种自举加载模式。由于 HPI 自举模式使用较为灵活，速度较快，并且能够和 MCU 的控制功能结合起来，因此本实验中选择使用 HPI 模式对 DSP 进行自举加载。HPI 自举加载流程如图 1-3 所示。

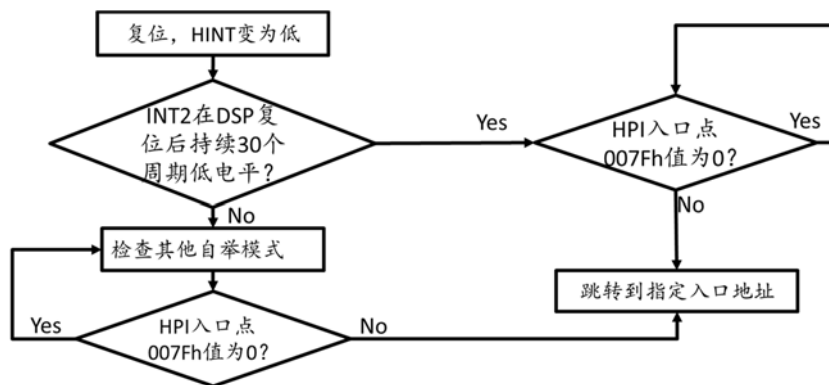


图 1-3 HPI 自举加载流程图

若要进入 HPI 自举方式，则需要将 INT2 引脚在 DSP 复位后保持 30 个时钟周期为低电平。进入 HPI 自举加载模式后，Bootloader 程序将自动检测程序加载情况，此时主机通过 HPI 接口直接访问 DSP 的在片 RAM，向 DSP 写入程序。若 HPI 入口点地址 0x007Fh 中的值为 0，表明 HPI 自举加载尚未完成，需要继续等待判断。若 0x007Fh 中的值不为 0，则表明 HPI 自举加载完成。此时 Bootloader 将 0x007Fh 中地址加载到 PC 指针，从对应地址开始执行所加载的 DSP 程序。

C54x 系列 DSP 片内有一个 HPI 接口。大多数 HPI 是一个 8 位并行接口，用来将主机与 DSP 相连。主机是 DSP 的主控者，它可以通过 HPI 直接访问 DSP 的在片 RAM，包括存储器映像寄存器。HPI 的接口框图如图 1-4 所示。

由图 1-4 可见，HPI 主要由 4 个部分组成，即：

1、HPI 地址寄存器（HPIA）。它只能由主机对其直接访问。寄存

器中存放着当前寻址 HPI 存储单元的地址。

2、HPI 数据寄存器 (HPID)。它只能由主机对其进行访问。如果当前进行的是读操作,则 HPID 中存放的是要从 DSP 在片 RAM 中读出的数据;如果当前进行的是写操作,则 HPID 中存放的是将要写入到 DSP 在片 RAM 的数据。

3、HPI 控制寄存器 (HPIC)。DSP 和主机都能对它直接访问,它在 C54x 系列 DSP 中的映像地址为 0x002Ch。

4、HPI 控制逻辑。用于接收和处理 HPI 与主机之间的接口信号。

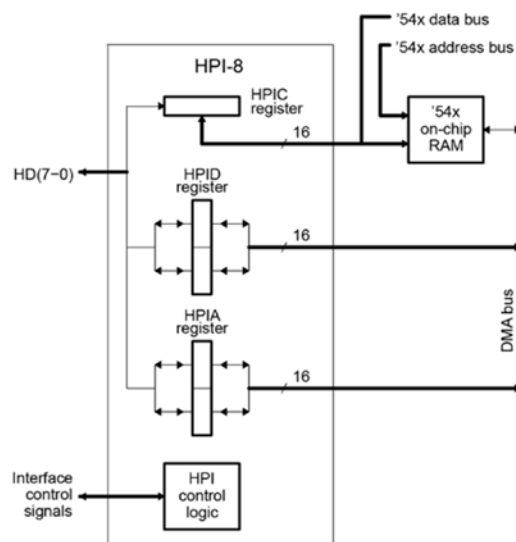


图 1-4 HPI 接口内部框图

当 C54x 系列 DSP 与主机（或者主设备）交换信息时, HPI 在逻辑上相当于主机的一个外围设备。HPI 的外部数据线是 8 根(HD0~HD7), 在与主机传数据时,HPI 能自动地将 HD0~HD7 传来的每个字节(8 位), 按高低顺序组合成一个字(16 位)后传送给 DSP。

C54x 系列 DSP 的 HPI 接口与主机的连接框图如图 1-5 所示。其中 HD0~HD7 为数据总线, 其余为控制信号线。DSP 通过 HPI 与主机



相连时，除了 8 位 HPI 数据总线以及控制信号线以外，基本上不需要其它的逻辑电路。通过 HPI 数据线与控制线的相互配合，可以完成主机与 DSP 之间的高速数据传输。

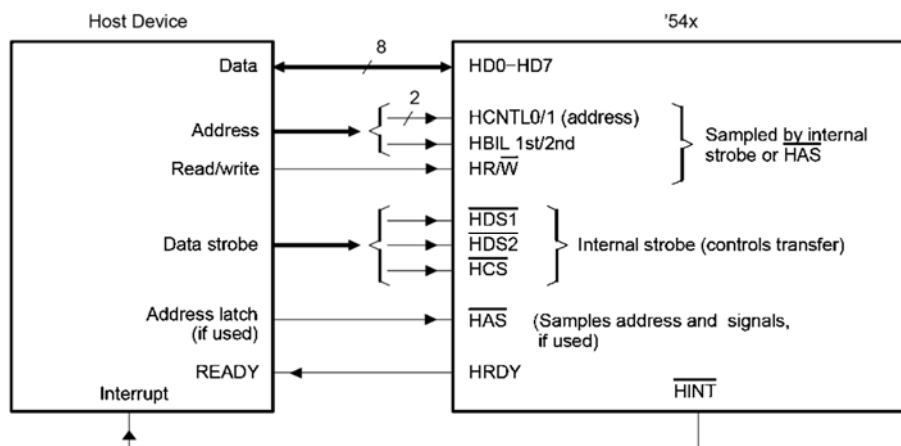


图 1-5 C54x HPI 与主机的连接框图

图 1-5 中，HCNTL0/1 引脚用于使能当前主机能够访问的 HPI 寄存器；HBIL 引脚用于区分当前主机向 DSP 所发送的是第一个或第二个字节； $\overline{HR}/\overline{W}$  引脚用于告之 DSP 当前为读/写数据； $\overline{HCS}$ 、 $\overline{HDS1}$ 、 $\overline{HDS2}$  引脚用于数据选通，互相配合可实现一个主机对多个 DSP 的 HPI 访问； $\overline{HAS}$  为地址锁存信号；HRDY 为 HPI 就绪引脚，用于通知主机 DSP 的 HPI 接口已准备好进行数据传输与； $\overline{HINT}$  为 HPI 中断信号，即 DSP 可向主机发出中断请求。

在本设计电路中 HPI 接口与主机的连接如图 1-6 所示。

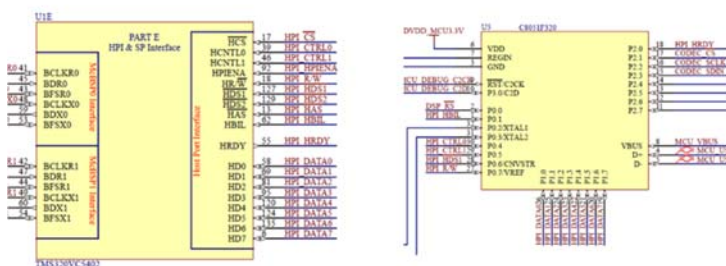


图 1-6 本设计中 HPI 接口与主机连接示意图

HPI 的具体操作由 HPI 控制寄存器 (HPIC) 中各状态位的控制。各状态位在 HPIC 中的位置 and 其所对应的功能分别如图 1-7 和表 1-3 所示。

15-13	12	11	10	9	8	7-5	4	3	2	1	0
X	XHPIA†	HINT	DSPINT	X	BOB	X	XHPIA†	HINT	DSPINT	X	BOB

图 1-7 HPIC 寄存器中各状态位对应位置

位	功 能
HPIA	扩展地址使能，用于自动增量寻址
HINT	DSP->主机中断，用于描述 HINT 引脚状态。
DSPINT	主机->DSP 中断，用于主机向 DSP 产生 HPI 中断
BOB	HPI 传输字节顺序，0 时先高后低，1 时先低后高

表 1-3 寄存器中各位所对应功能

其中，HPIA 位用于扩展地址使能，用于 C5410 芯片的自动增量寻址（本设计中未使用）；HINT 位为 DSP->主机的中断标志位，用于描述  $\overline{HINT}$  的引脚状态；DSPINT 为主机->DSP 中断标志位，即主机可通过将此位置位向 DSP 发出中断请求，通常用于主机和 DSP 之间建立通信前的握手；BOB 为 HPI 传输字节顺序控制位，0 时传输数据先高字节后低字节，1 时传输数据先低字节后高字节。

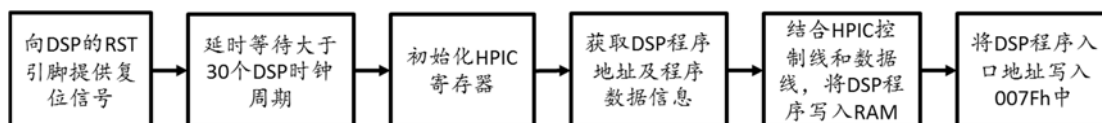


图 1-8 主机 HPI 加载 DSP 程序流程

本设计中主机向 DSP 自举加载程序的流程如图 1-8 所示。

### 1.2.3 TLV320AIC23 音频编解码器及 McBSP 接口

TLV320AIC23（以下简称为 AIC23）是 TI 推出的一款高性能的立体声音频编解码芯片，内置耳机输出放大器，支持 MIC 和 LINE IN 两种

输入方式（二选一），且对输入和输出都具有可编程增益调节。

AIC23 的模数转换（ADCs）和数模转换（DACs）部件高度集成在芯片内部，采用了先进的 $\Sigma-\Delta$ 过采样技术，可以在 8K 到 96K 的频率范围内提供 16bit、20bit、24bit 和 32bit 的采样。

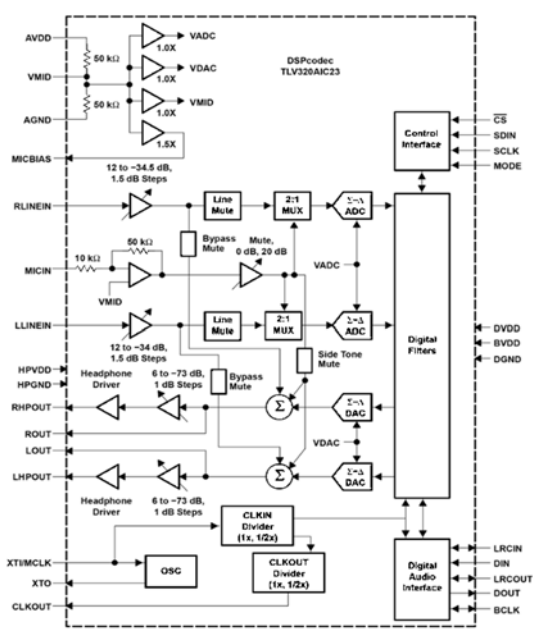


图 1-9 AIC23 管脚和内部框图

AIC23 管脚及内部框图如图 1-9 所示。其引脚功能描述如表 1-4 所示。

引 脚	描 述
BCLK	I2S 时钟，主模式与从模式不同
CLKOUT	时钟输出
CS	片选，3 线/2 线控制模式不同
DIN	I2S 格式串行数据输入(DAC)
DOUT	I2S 格式串行数据输出(ADC)
LHPOUT	左声道耳机输出
LLINEIN	左声道线路输入
LOUT	左声道线路输出
LRCIN	I2S 字同步时钟(DAC，主从不同)
LRCOUT	I2S 字同步时钟(ADC，主从不同)
MICBIAS	麦克风直流偏置
MICIN	麦克风输入
MODE	串口模式选择

RHPOUT	右声道耳机输出
RLINEIN	右声道线路输入
ROUT	右声道线路输出
SCLK	控制端口串口数据时钟
SDIN	控制端口串口数据输入
XTI/MCLK	外部时钟输入
XTO	晶振输出(主模式)

表 1-4 AIC23 引脚功能描述

在本设计中，AIC23 相关电路如图 1-10 和图 1-11 所示。

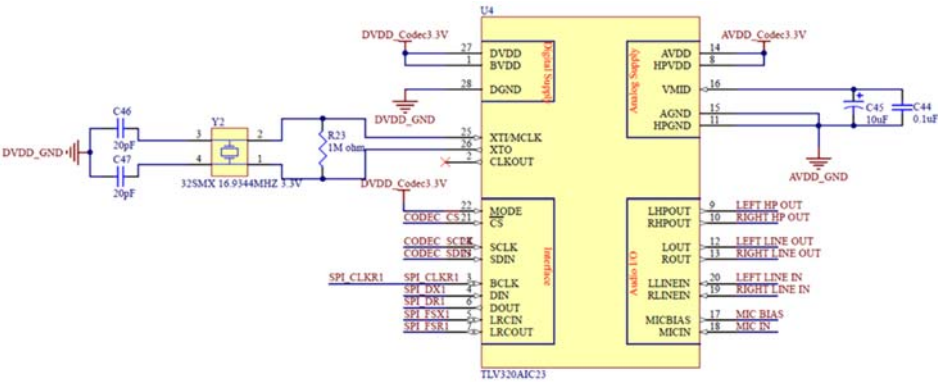


图 1-10 AIC23 外围电路图

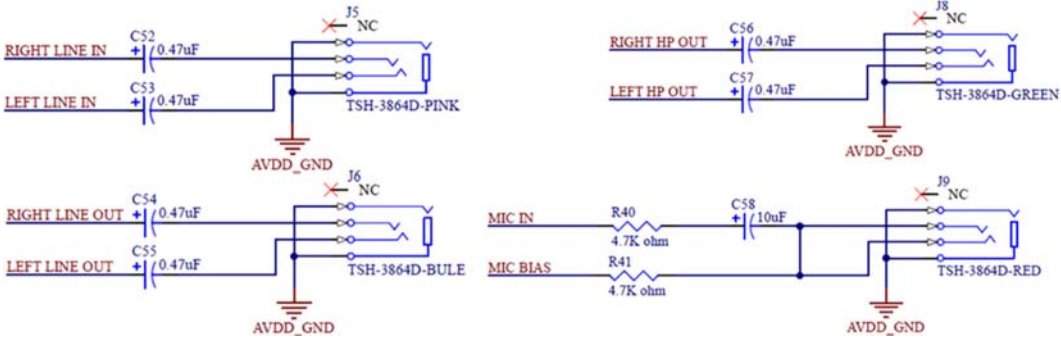


图 1-11 TLV320AIC23 音频外部接口电路图

本设计中 AIC23 芯片的工作模式由 MCU 进行控制。MCU 的 I/O 与 AIC23 的  $\overline{CS}$ 、SCLK、SDIN 引脚相连。MCU 利用 I/O 模拟 SPI 总线通信协议向 AIC23 写入控制指令，实现对 AIC23 工作模式的控制，这种控制模式为 SPI (三线) 控制模式。SPI 总线工作时序如图 1-12 所示。

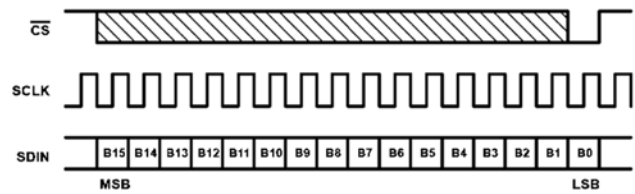


图 1-12 SPI 总线工作时序图

在 SPI 控制模式中，在每次控制字数据传输时， $\overline{CS}$  置低电平，SCLK 为 SPI 发送时钟（由 MCU 提供），SDIN 用于控制字的串行传输，AIC23 在 SCLK 的每个上升沿锁存当前 SDIN 的输入。控制字共 16 位，每次传输都从最高位开始到最低位结束， $\overline{CS}$  置高表明传输完成。

地 址	寄 存 器
0000000	左声道线路输入音量控制
0000001	右声道线路输入音量控制
0000010	左声道耳机音量控制
0000011	右声道耳机音量控制
0000100	模拟音频路径控制
0000101	数字音频路径控制
0000110	电源关闭控制
0000111	数字音频接口格式
0001000	采样率控制
0001001	数字接口激活
0001111	复位寄存器

表 1-5 AIC23 各寄存器地址

AIC23 各寄存器地址如表 1-5 所示。本设计中，通过对 AIC23 各寄存器进行设置，设定 AIC23 工作模式如下：

- 采样率：44.1KHz
- 工作模式：主模式
- 数字音频接口模式：DSP 模式
- MIC 输入增益：0dB
- 耳机输出音量：+6dB

本设计中，AIC23 通过 McBSP 接口与 DSP 芯片进行数据传输。

多通道缓冲串行口（McBSP）是在标准串口基础上发展而来，其具有全双工通信、多达 128 个通道、数据选择范围宽、可进行压缩通信等优点。McBSP 接口内部框图如图 1-13 所示。

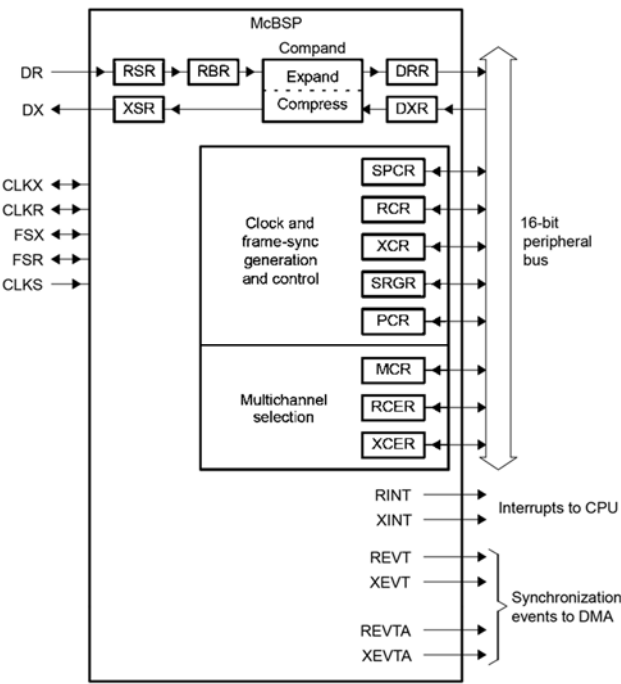


图 1-13 McBSP 接口内部框图

从图 1-13 可以看出，McBSP 通过 7 个引脚与外部相连进行数据传输。各引脚功能描述如表 1-6 所示。

引 脚	描 述
CLKR	接收时钟信号引脚
CLKX	发送时钟信号引脚
CLKS	外部时钟信号引脚
DR	串行接收数据引脚
DX	串行发送数据引脚
FSR	接收帧同步信号引脚
FSX	发送帧同步信号引脚

表 1-6 McBSP 各引脚功能描述

为保证 DSP 可顺利的通过 McBSP 口与外部进行通信，需要根据

需求设定 McBSP 接口的工作模式。这些需要通过对 McBSP 控制相关的各寄存器的设定来达到。其各寄存器描述如表 1-7 所示。

寄 存 器	描 述
RSR	移位寄存器
RBR	发送时钟信号引脚
DRR	接收缓冲寄存器
XSR	发送移位寄存器
DXR	发送移位寄存器
SPCR	串口控制寄存器
RCR	接收控制寄存器
XCR	发送控制寄存器
SRGR	采样率发生寄存器
PCR	引脚控制寄存器
MCR	多通道寄存器
RCER	接收通道使能寄存器
XCER	发送通道使能寄存器

表 1-7 McBSP 各寄存器描述

AIC23 的 I<sup>2</sup>S 串口，可以与 DSP 的多通道缓冲串口(McBSP)相联。需要注意的是，当 AIC23 工作在主模式时，McBSP 的发送时钟和接收时钟都由 AIC23 提供。AIC23 芯片与 DSP 连接示意图如图 1-14 所示。其中，BCLK 为 McBSP 传输时钟，由 AIC23 提供给 DSP 的发送/接收时钟 (CLKX/CLKR)；FSX-IRCIN/FSR-IRCOUT 分别为 DSP 向 AIC23 的发送帧同步和 DSP 从 AIC23 的接收帧同步；DX-DIN 和 DR-DOUT 分别为 DSP 向 AIC23 的数据发送和 DSP 从 AIC23 的数据接收。

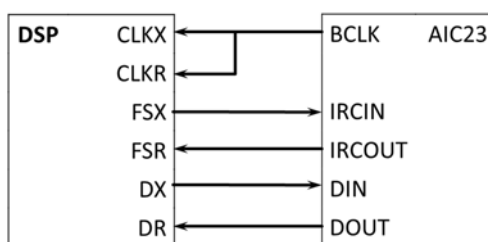


图 1-14 AIC23 与 McBSP 接口连接示意图

本设计中 AIC23 工作在主模式，DSP 工作在从模式。工作时时序图如图 1-15 所示：

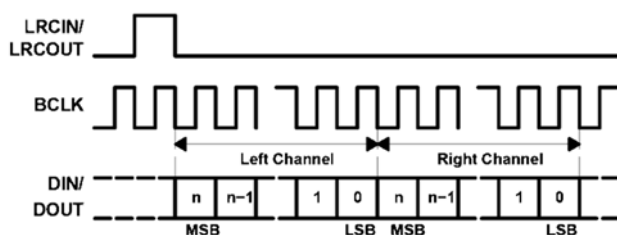


图 1-15 McBSP 接口工作时序图

在每次的数据传输中，LRCIN/LRCOUT 的脉冲用于标记接收/发送帧的开始，在 LRCIN/LRCOUT 置低后，BCLK 的每个上升沿发送/接收当前传输数据的每一位。每次传输遵从先左声道后右声道的原则，并按先高位后低位的顺序传输。

AIC23 与 DSP 连接的电路原理图如图 1-16 所示。其中 AIC23 的 MODE 引脚置高，选择为 SPI 控制模式； $\overline{CS}$  引脚由 MCU 的 IO 口进行控制，SCLK 和 SDIN 连接 MCU 的串行接口；BCLK 与 DSP 的 1 号 McBSP 发送、接收时钟信号连接；DIN、DOUT、LRCIN、LRCOUT 分别与 DSP 的 1 号 McBSP 的对应引脚连接。

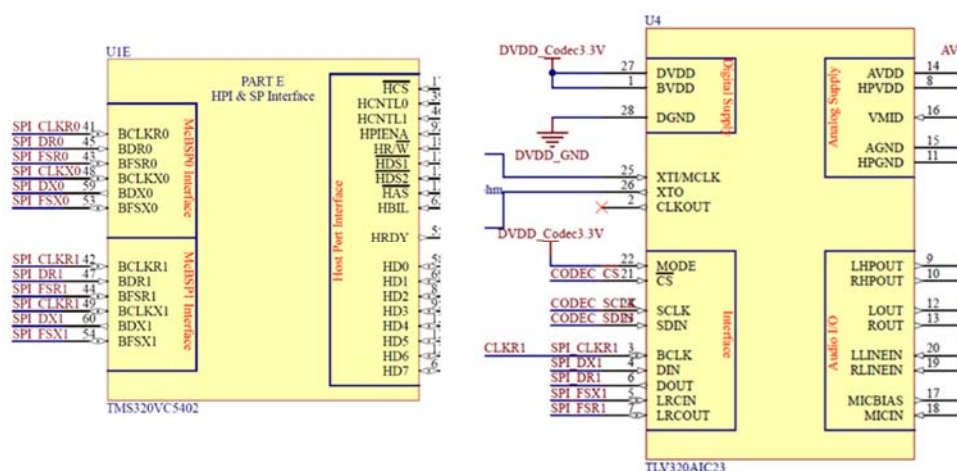


图 1-16 McBSP 与 AIC23 电路连接图



DSP 使用 McBSP 与 AIC23 进行通信的程序流程如图 1-17 所示。

DSP 在完成时钟和 McBSP 初始化后等待 McBSP 接收中断，当中断发生后接收到新数据并对其进行处理，并在收到下次接收中断时发送出本次处理的结果，从而对接收到的音频信号完成信号处理算法。

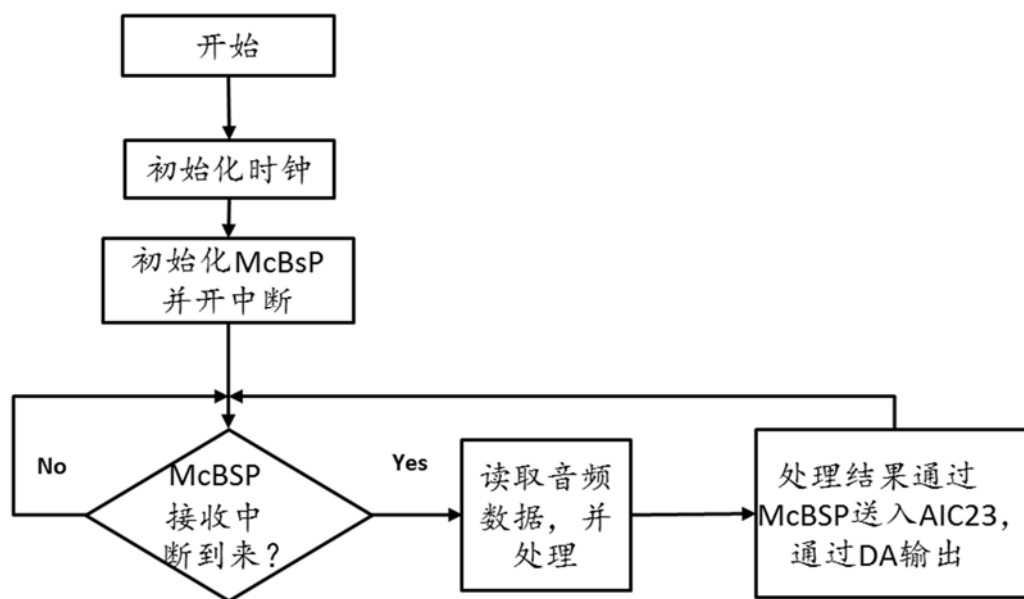


图 1-17 DSP 与 AIC23 通信程序流程图

## 第2章 电路原理及程序设计

第 1 章中在系统构架和各模块技术原理层面对本设计进行了阐述。本章将在具体电路设计及原理和具体程序设计上对其进行描述。

### 2.1 电路原理设计<sup>1</sup>

#### 2.1.1 系统供电设计

本设计中，所用主要芯片供电电压、电流如表 2-1 所示。

表 2-1 所用主要芯片供电电压电流列表

芯片	电 压	电 流
TMS320VC5402	核心数字 1.8V	45mA
	引脚数字 3.3V	30mA
C8051F320	数字 3.3V	11mA
TLV320AIC23	数字 3.3V 及模拟 3.3V	23mA
MAX6835	数字 3.3V	20mA

表 2-1 本设计中所用主要芯片的供电电压及电流

估计总功耗为：

$$P_{Total} = P_{Chips} + P_{Est} \approx 400mW$$

设计中总电源为直流 5V 供电，考虑到设计扩展并提高带载能力，采用双 USB 接口设计，J1 为供电和数据传输复用 USB 接口，其 2、3 引脚为 USB 数据差分信号，正相由 R33 上拉至 3.3V，分别由 27 欧电阻匹配；J3 为独立供电 USB 接口；S9 为供电开关，电路如图 2-1 所示。

<sup>1</sup> 本节只给出电路设计中的关键细节，详细电路设计请见附录 A

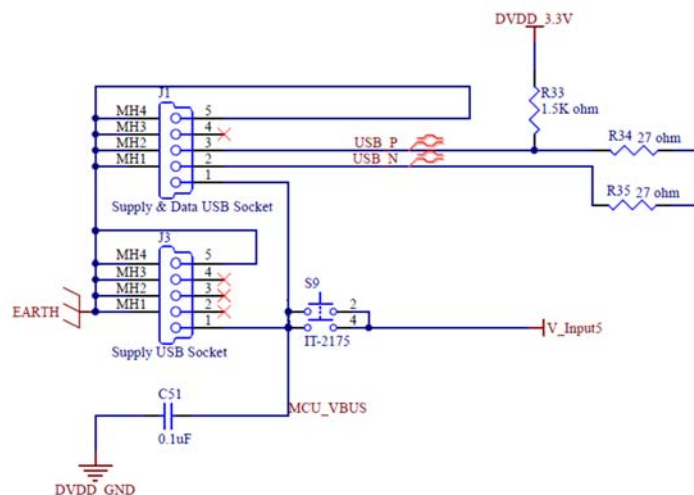


图 2-1 供电输入和数据传输复用 USB 接口

采用 LM1117 稳压芯片将输入 5V 分别转换为 3.3V 和 1.8V。考虑到电路中既需要模拟供电，又需要数字供电，为了减小供电中产生的干扰，因此将 3.3V 电源分为数字与模拟两路并使用 0 欧电阻做数模隔离。

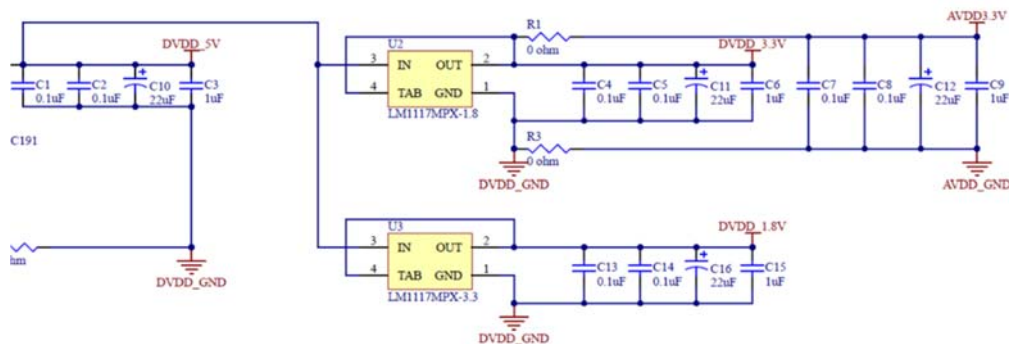


图 2-2 供电电路设计

并且，如图 2-3 所示，为进一步减小供电之间的互相影响，在不同芯片供电电源之间也使用 0 欧电阻隔离，并使用去耦电容减小供电波纹。

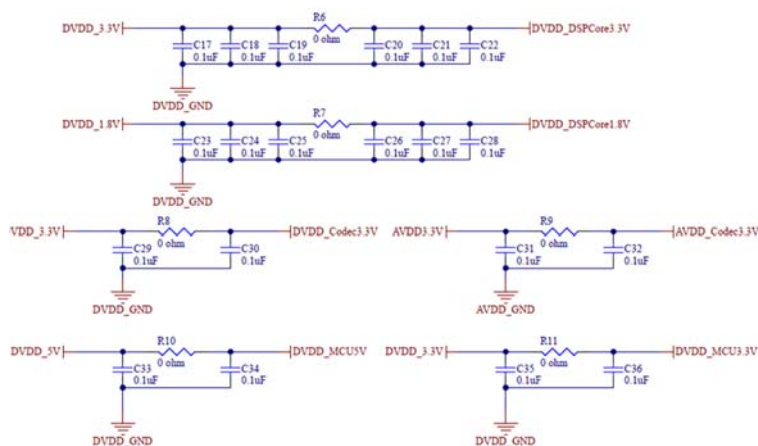


图 2-3 供电隔离与去耦电路

### 2.1.2 DSP 基本控制及中断、调试接口设计

本设计中 DSP 的复位信号由 MCU 提供，时钟振荡信号由外接 20MHz 的晶振提供。并且使用拨码开关 S1 控制 DSP 内部锁相环的倍频系数，实现外部对 DSP 运行速度的控制，电路如图 2-4 所示。

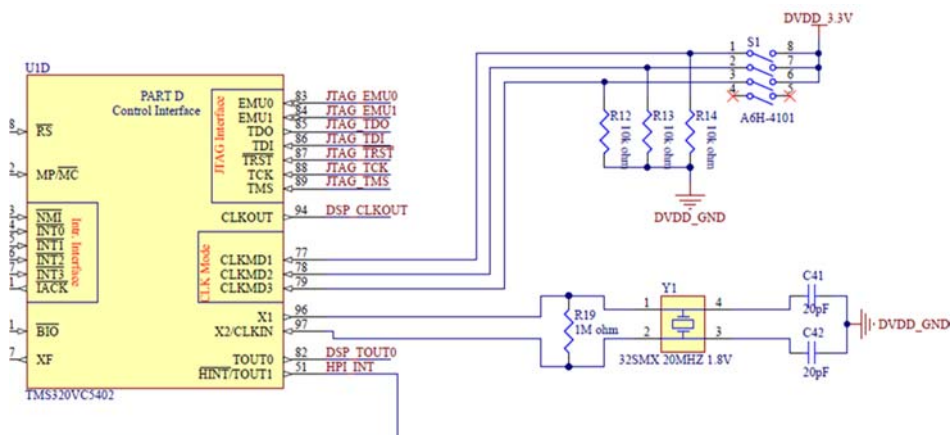


图 2-4 时钟信号输入及 CLKMD 模式选择电路

本设计中 DSP 的  $MP/\overline{MC}$  引脚使用跳线设置；将 HPI 中断引脚与外部中断 2 引脚相连，使 DSP 能够自动进入 HPI 自举模式；中断信号可由矩阵键盘给出；程序分支转移信号可由 S6 开关给出；XF 引脚和  $\overline{IACK}$  外接发光二极管，可与程序配合观察到 DSP 的工作状态；电路如图 2-5 所示，JTAG 调试接口如图 2-6 所示。

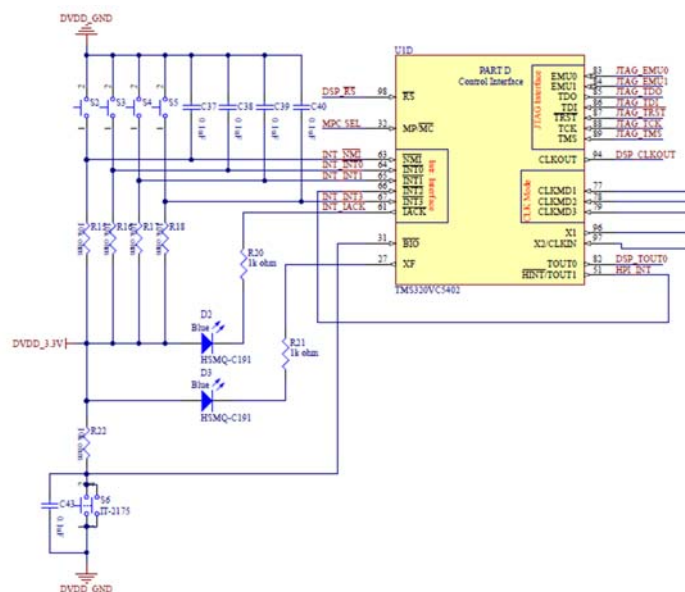


图 2-5 中断、BIO、MP/MC 模式选择等电路

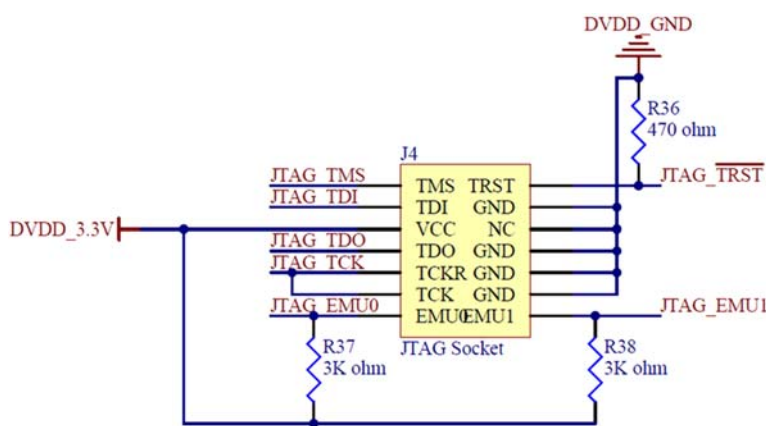


图 2-6 JTAG 调试接口

### 2.1.3 MCU、编码/解码器电路设计

本设计中 MCU (8051) 内核使用 3.3V 电源, 并在其内部集成了 5V 的稳压芯片, 因此需要 3.3V 和 5V 数字供电; 使用 20MHz 晶振作为 MCU 的时钟源; 使用 MAX6835 芯片监测电源及开关 S7 的输入为 MCU 提供复位信号; 使用拨码开关 S8 对 8051 的工作模式进行设置, 电路如图 2-7 所示。

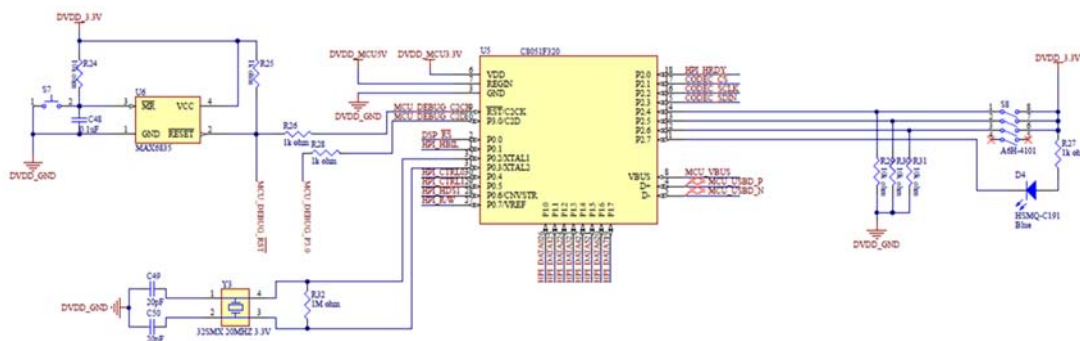


图 2-7 8051 外部电路设计

8051 的调试接口电路如图 2-8 所示。

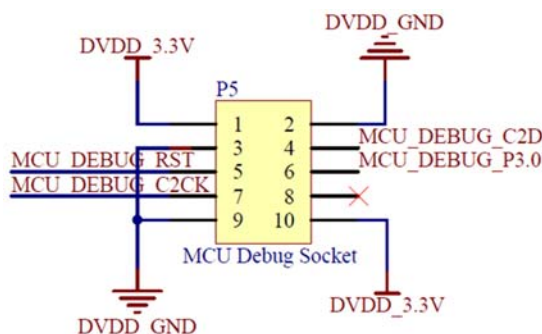


图 2-8 8051 调试接口

本设计中 AIC23 采用数字 3.3V 和模拟 3.3V 供电，模拟 VMID 引脚用于 AIC23 内部的运放供电，因此需要接电容去耦；使用 SPI 控制模式，因此 MODE 引脚上拉；晶振选用 16.9344MHz 为 AIC23 提供频率源，电路如图 2-9 所示

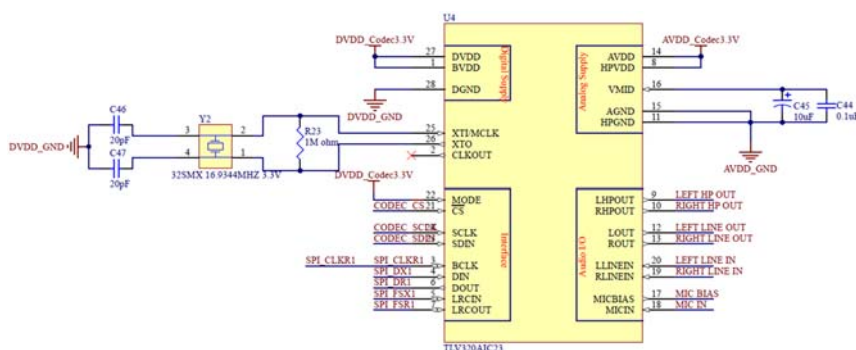


图 2-9 AIC23 外部电路设计

AIC23 的音频接口参考 TI 公司所提供的 EVM 板设计，电路如图

2-10 所示。

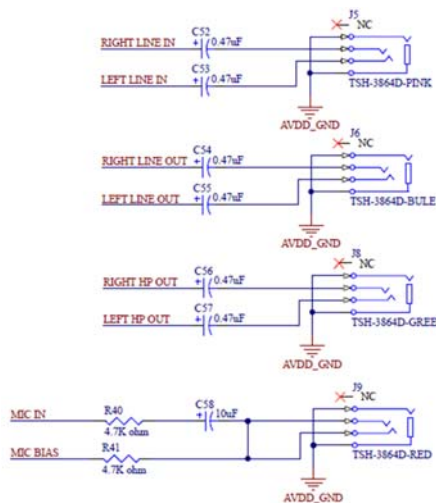


图 2-10 AIC23 音频接口设计

#### 2.1.4 接口与外部连接件

本设计中 USB 接口、JTAG 调试接口、8051 调试接口、AIC23 音频接口分别见图 2-1、图 2-6、图 2-8、图 2-10，其他接口（HPI 设置跳线、MP/MC 模式选择跳线、地址总线接口、数据总线接口、总线控制信号接口、HPI 接口、时钟及定时器输出接口、中断接口、McBSP0 及 McBSP1 接口）设计请见。

#### 2.1.5 PCB

本设计的 PCB 采用双层设计，大小为 162.55mm（长）×67.05mm（高），其线路连接（未覆铜）如图 2-11 所示。考虑到优化地平面，增强供电性能和信号传输性能，在 PCB 表面左上角 USB 接口处的覆铜与外部电源输入地相连；PCB 右侧 AIC23 芯片及音频接口处的覆铜与模拟地相连；其余部分及 PCB 背面全部覆铜与数字地相连。PCB 覆铜后正面与背面分别如图 2-12、图 2-13 所示，三维俯视图如图 2-14 所示。



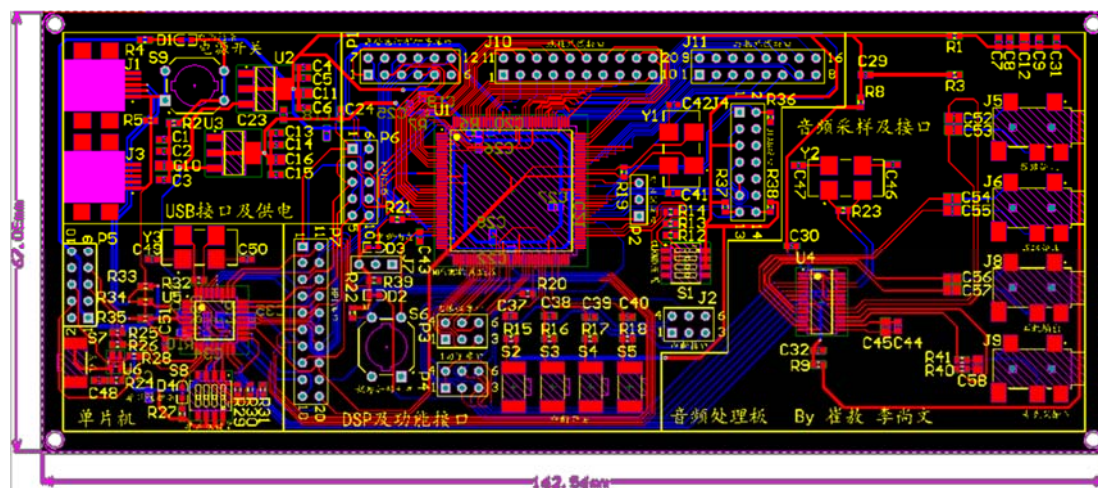


图 2-11 PCB 设计（正面，未覆铜）

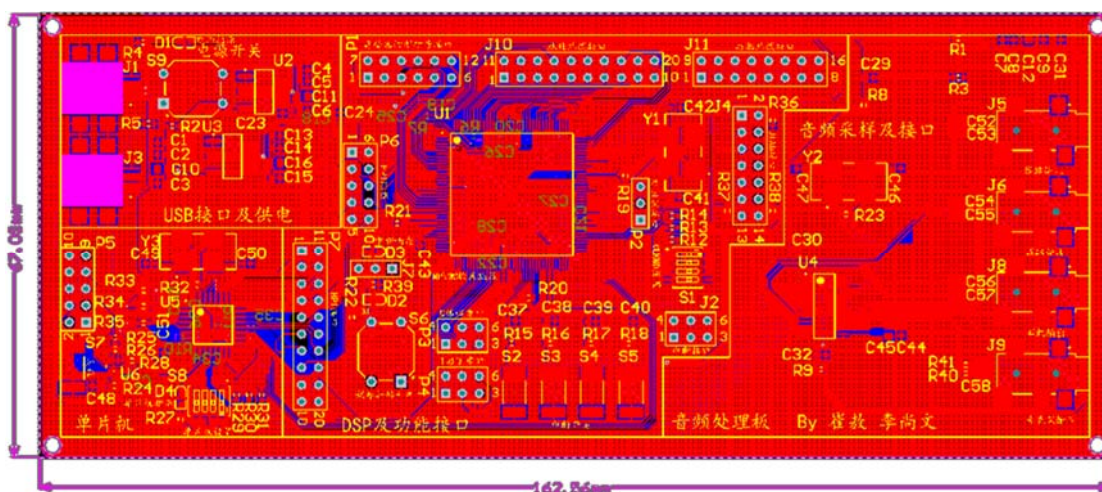


图 2-12 PCB 设计，覆铜（正面）

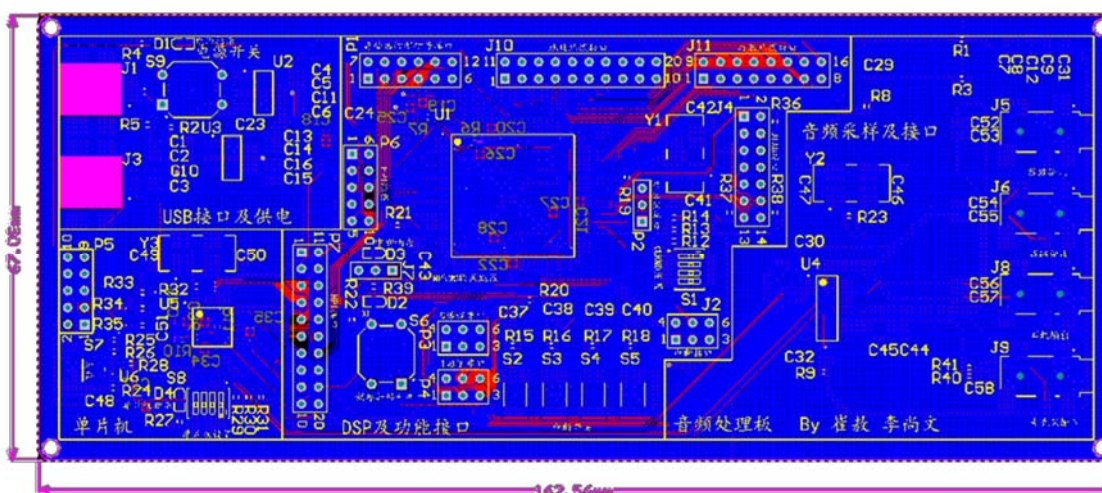


图 2-13 PCB 设计，覆铜（背面）



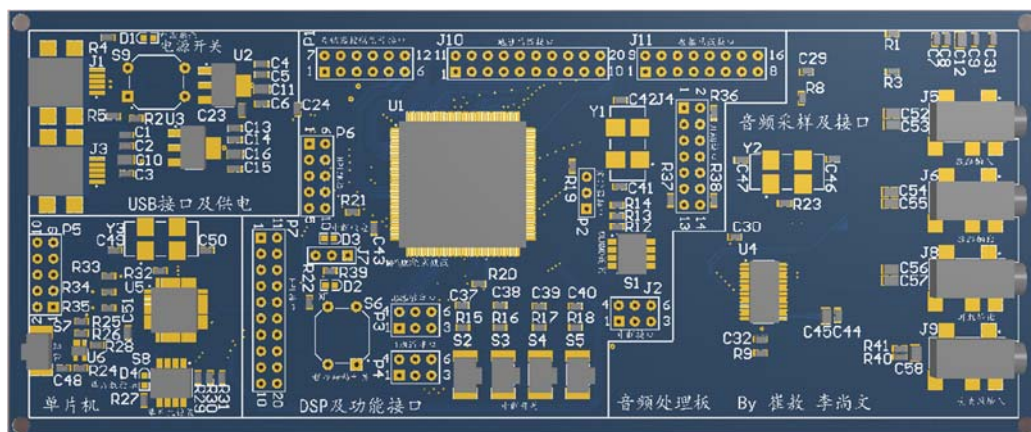


图 2-14 PCB 三维俯视图

## 2.2 程序设计<sup>2</sup>

### 2.2.1 程序开发流程

为保证系统开发的顺利进行，在开展具体的工作之初，需要对整个开发工作的流程进行设计。本文中软件开发流程如图 2-15 所示。

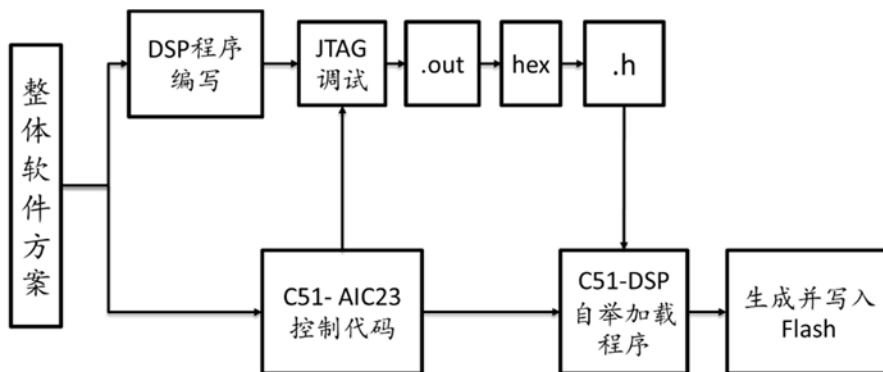


图 2-15 软件开发流程图

### 2.2.2 MCU 向 DSP 自举程序

本设计中，MCU 主要用于接收 PC 端串行通信指令，并根据指令向 DSP 加载不同的程序。MCU 程序流程如图 2-16 所示。

<sup>2</sup> 本节主要介绍软件算法，详细代码请见附录 C

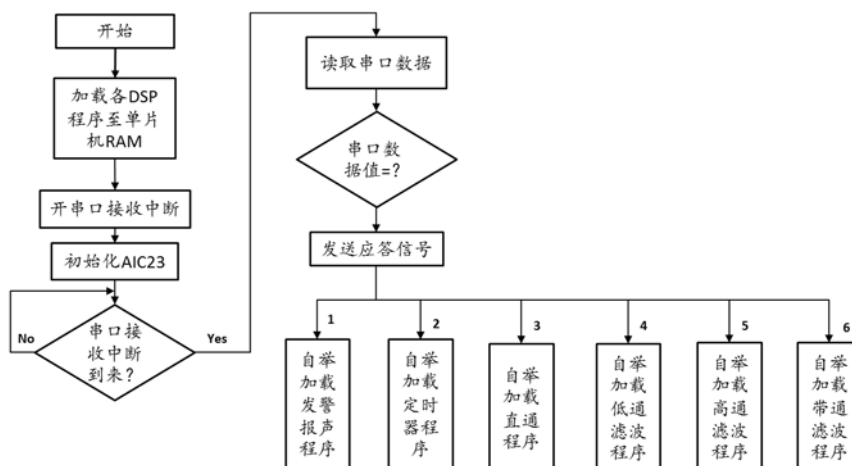


图 2-16 MCU 程序流程图

### 2.2.3 DSP 信号处理程序

DSP 信号处理程序流程图如图 2-17 所示。

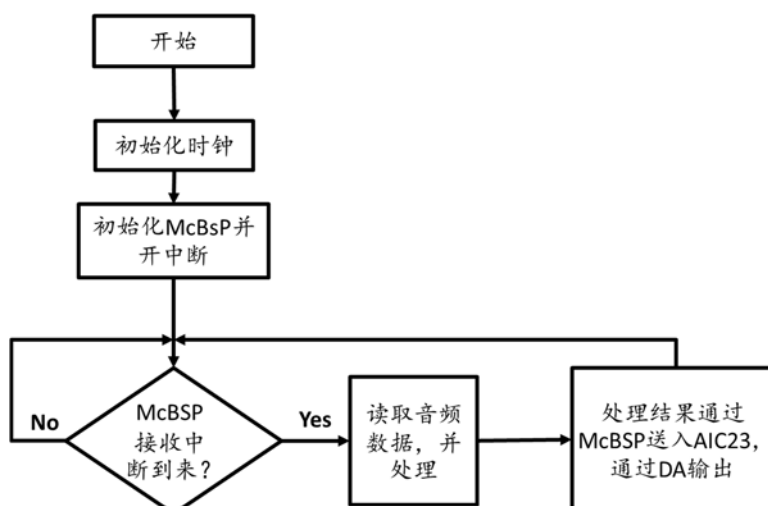


图 2-17 DSP 信号处理程序流程图

本文中实现了利用带通滤波器和低通滤波器对信号的处理。设计中所用到的滤波器的频响特性及相关指标如下所示：

(1) 设计中所用低通滤波器，其频响特性如图 2-18 所示。

- 采样率：44.1kHz
- 阶数：N=249
- 通带截止频率：500Hz

- 阻带边界频率：1500Hz

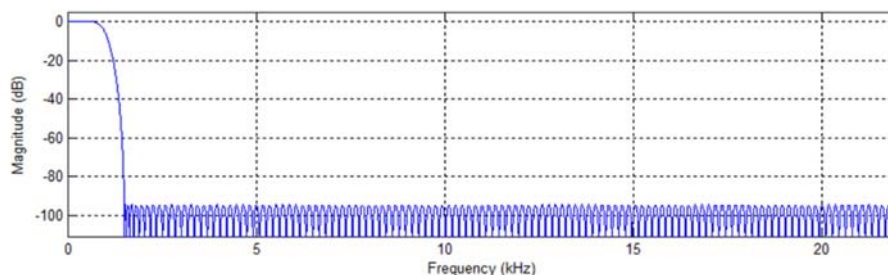


图 2-18 低通滤波器频响特性

(2) 设计中所用带通滤波器，其频响特性如图 2-19 所示。

- 采样率：44.1kHz
- 阶数：N=249
- 下通带截止频率：2000Hz
- 下阻带边界频率：1000Hz
- 上通带截止频率：3000Hz
- 上阻带边界频率：4000Hz

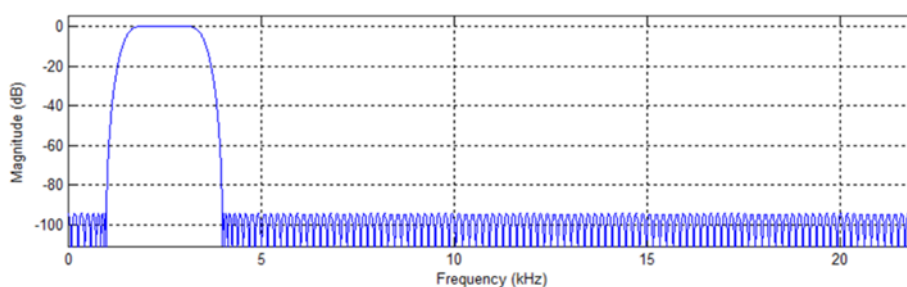


图 2-19 带通滤波器频响特性

#### 2.2.4 上位机程序

本设计中利用 Microsoft VB 6.0 开发 PC 端上位机程序，以实现 DSP 功能的控制。其软件界面如图 2-20 所示。



图 2-20 PC 端上位机控制平台软件界面图

该软件流程图如图 2-21 所示

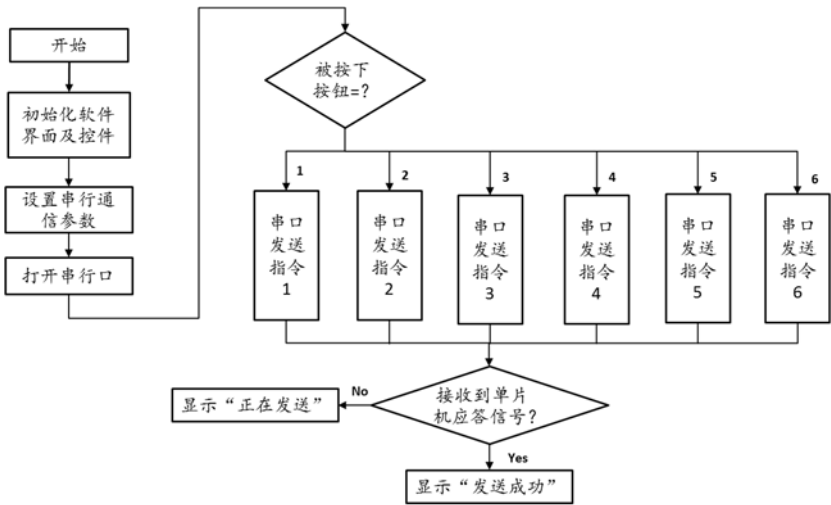


图 2-21 上位机软件流程图

### 第3章 实验结果及总结

系统设计开发完成后，对系统各个功能进行测试。将实验结果及分析总结如下：

- 1、PC 端上位机软件可实现与 DSP 系统的可靠通信；
- 2、单片机可顺利向 DSP 自举加载程序，且加载速度快；
- 3、DSP 可成功生成警报声信号；
- 4、DSP 能够定时产生稳定输出，使发光二极管定时闪烁；
- 5、在采样率为 44.1K 时，DSP 可实时对音频信号进行低通和带通滤波处理，DSP 在进行滤波算法时效率非常高。

在本系统的设计与软件实现过程中，我们收获到许多东西，总结如下：

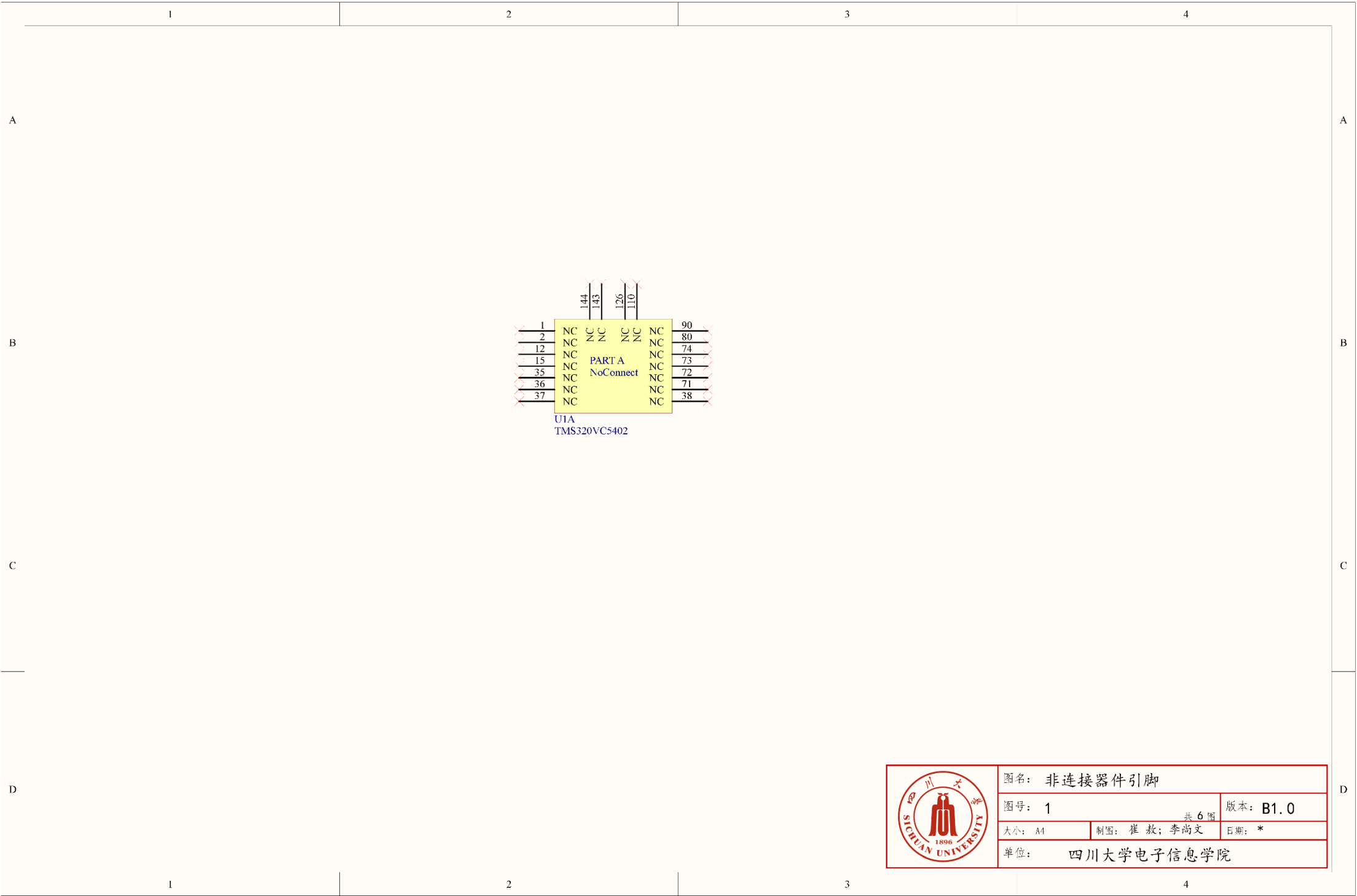
- 1、学习并掌握了 C54x 系列 DSP 的原理及汇编编程方法；
- 2、掌握并实现常用信号处理算法的 DSP 实现；
- 3、学习了 C54x 系列 DSP 定时器的原理及使用；
- 4、学习了 PC 端软件开发技能；
- 5、在一次次的争论、分歧和问题解决过程中，梳理了基础知识，提高了解决问题的能力，增强了团队协作能力。

由于时间和成本有限，我们并没有将所设计的 PCB 制作成实物，只是将本文设计所涉及到的程序在所购买的开发板上加以验证。希望在以后能够完完整整实现一个基于 DSP 的信号处理硬件系统。

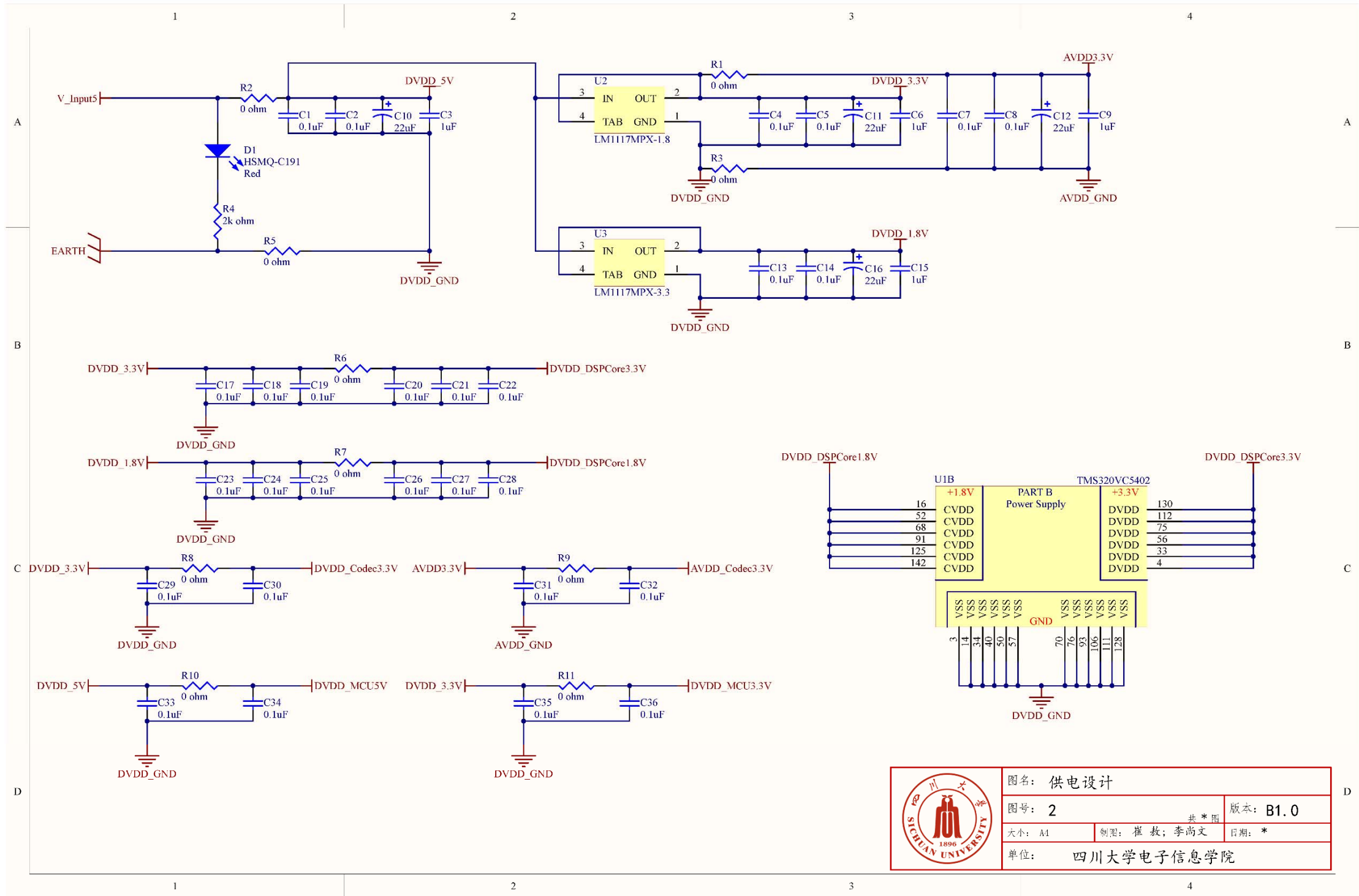
## 参考文献

- [1] INSTRUMENTS T. Data Sheet: TMS320VC5402 Fixed-point Digital Signal Processor [M/OL]. 1998
- [2] INSTRUMENTS T. TMS320C54x DSP Reference Set Volume 1: CPU and Peripherals [M/OL]. 2001
- [3] INSTRUMENTS T. TMS320C54x DSP Reference Set Volume 2: Mnemonic Instruction Set [M/OL]. 2001
- [4] INSTRUMENTS T. TMS320C54x DSP Reference Set Volume 3: Algebraic Instruction Set [M/OL]. 2001
- [5] INSTRUMENTS T. TMS320C54x DSP Reference Set Volume 4: Applications Guide [M/OL]. 2001
- [6] INSTRUMENTS T. TMS320C54x DSP Reference Set Volume 5: Enhanced Peripherals [M/OL]. 2001
- [7] TATER S. Application Report: Bootloading the TMS320VC5402 in HPI Mode [M/OL]. 2002
- [8] PEREZ R D. Application Report: TMS320VC5402 and TMS320UC5402 Bootloader [M/OL]. 2004
- [9] INSTRUMENTS T. Data Sheet: TLV320AIC23 Stereo Audio Codec [M/OL]. 2002
- [10] INSTRUMENTS T. User's Guide: TLV320AIC23 EVM [M/OL]. 2001
- [11] INSTRUMENTS T. User's Guide: TLV320AIC23 EVM2 [M/OL]. 2002
- [12] FANG W X, MILLER P. Application Report: Interfacing the TLV320AIC10/11 Codec to the TMS320C5402 DSP [M/OL]. 2000
- [13] LABORATORIES S. Data Sheet: C8051F320/1 Full Speed USB, 16k ISP FLASH MCU [M/OL]. 2009
- [14] LABORATORIES S. User's Guide: C8051F32x Development Kit [M/OL]. 2014

附录A 电路原理图

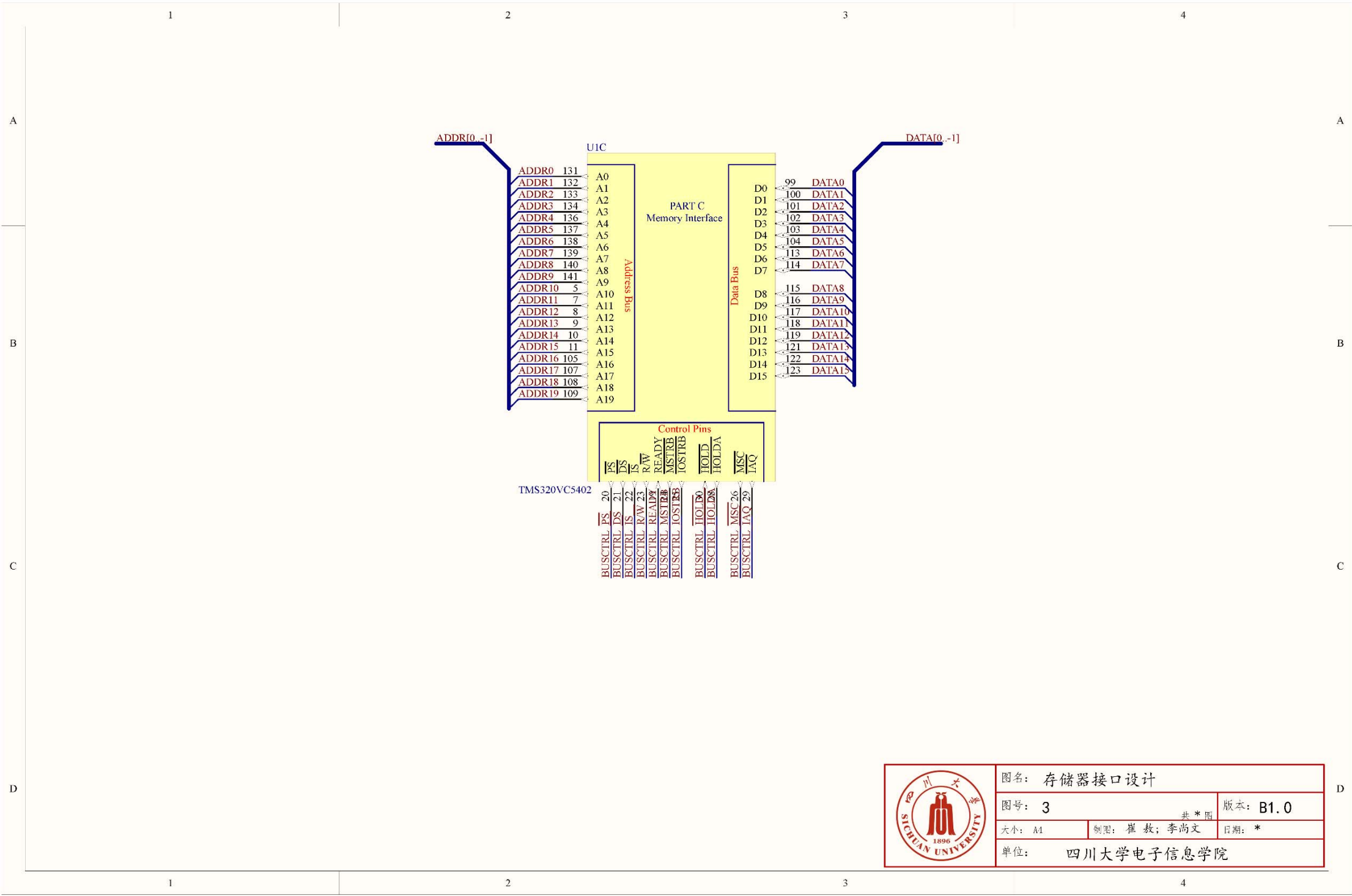


附图 A-1 非连接器件电路图

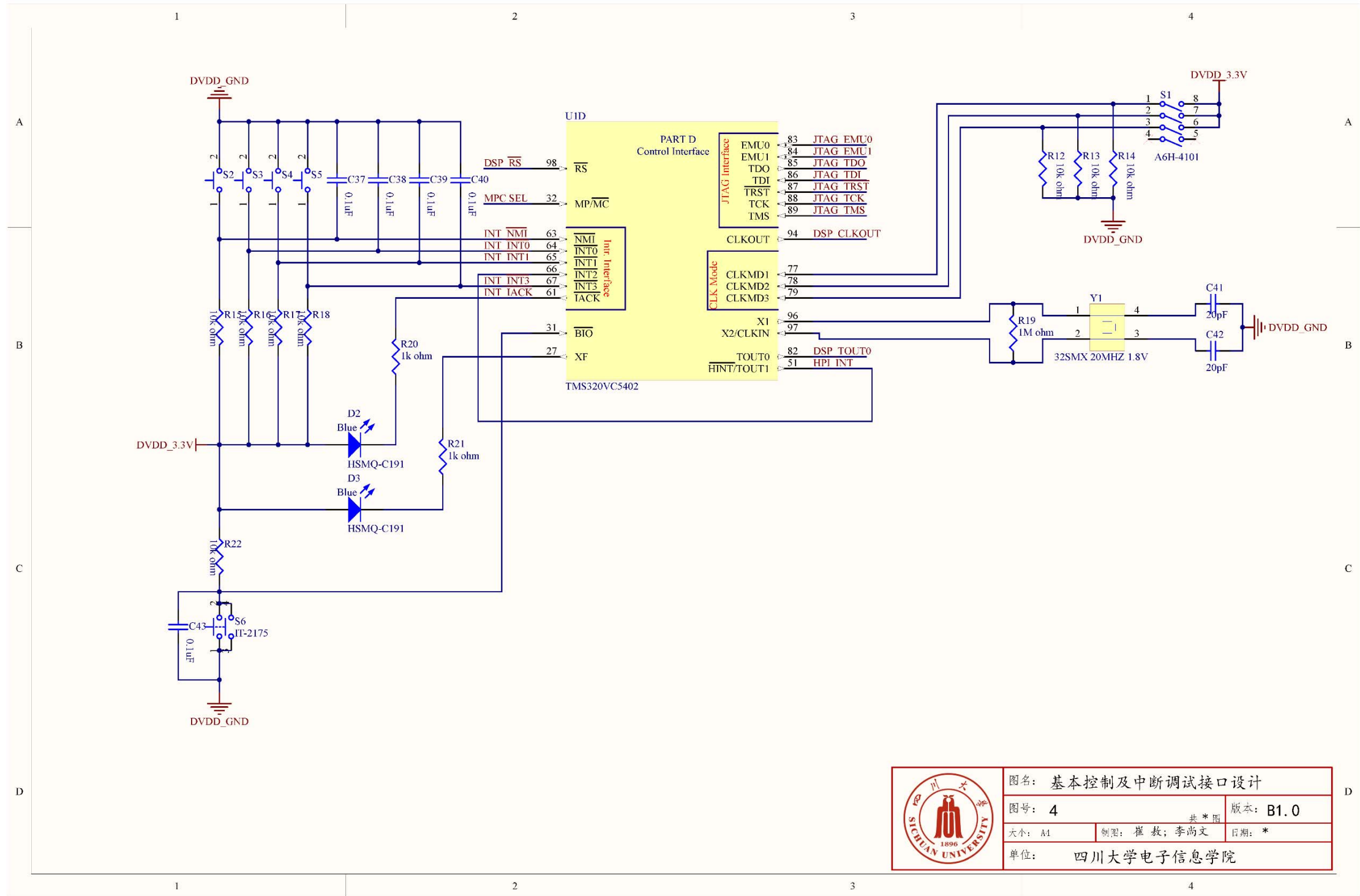


附图 A-2 供电设计电路图

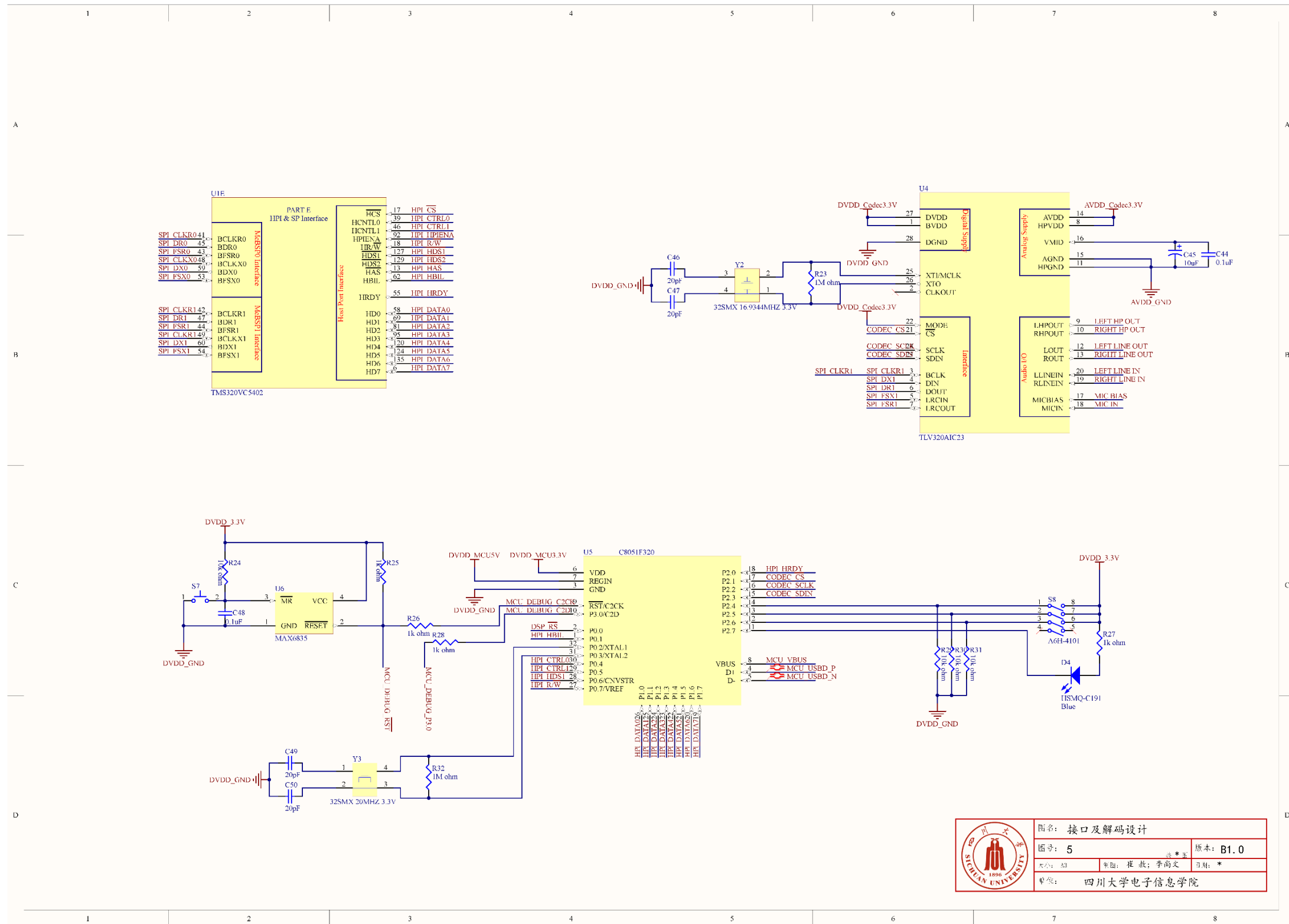




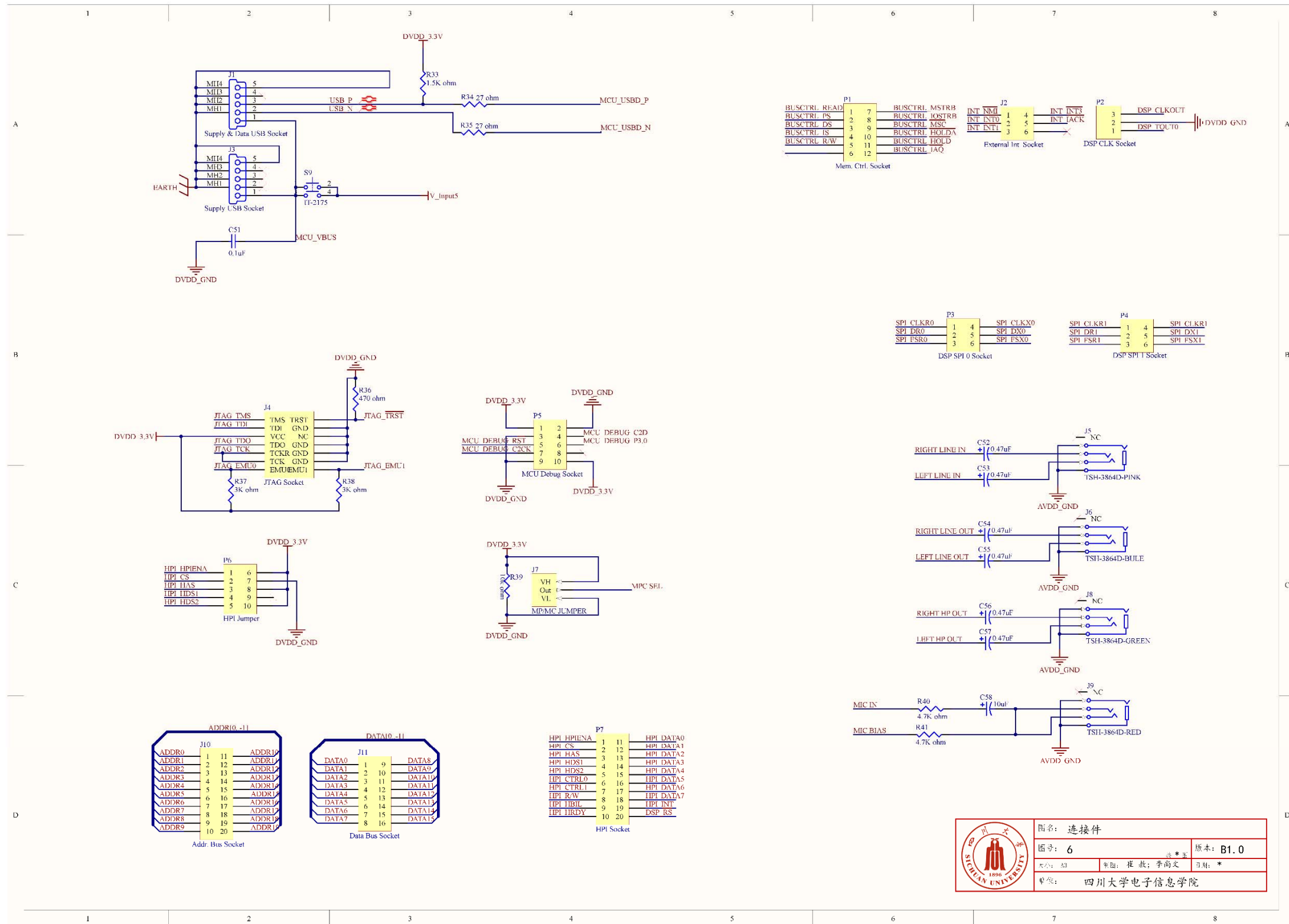
附图 A-3 存储器接口设计



附图 A-4 DSP 基本控制、中断、调试电路设计

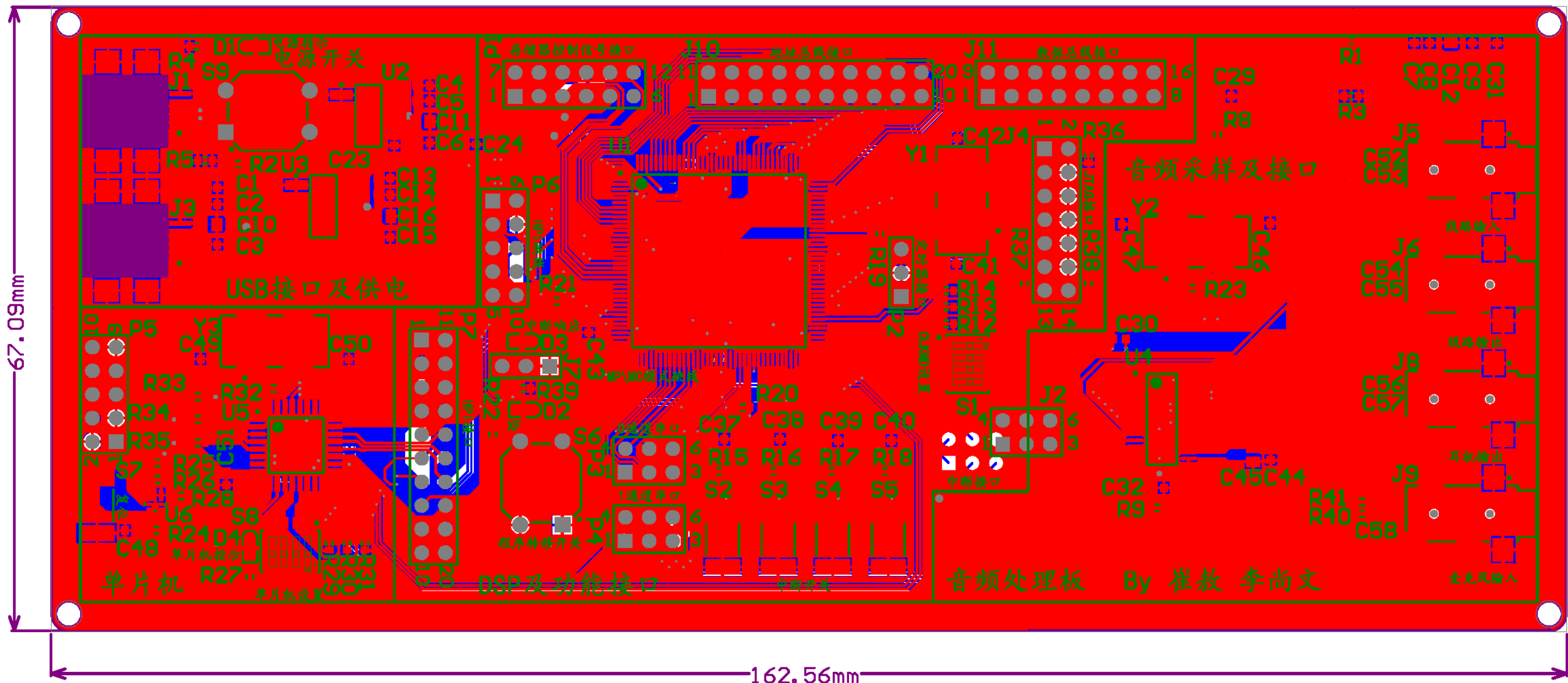


附图 A-5 DSP HPI、McBSP 接口与 MCU、音频编解码芯片电路图



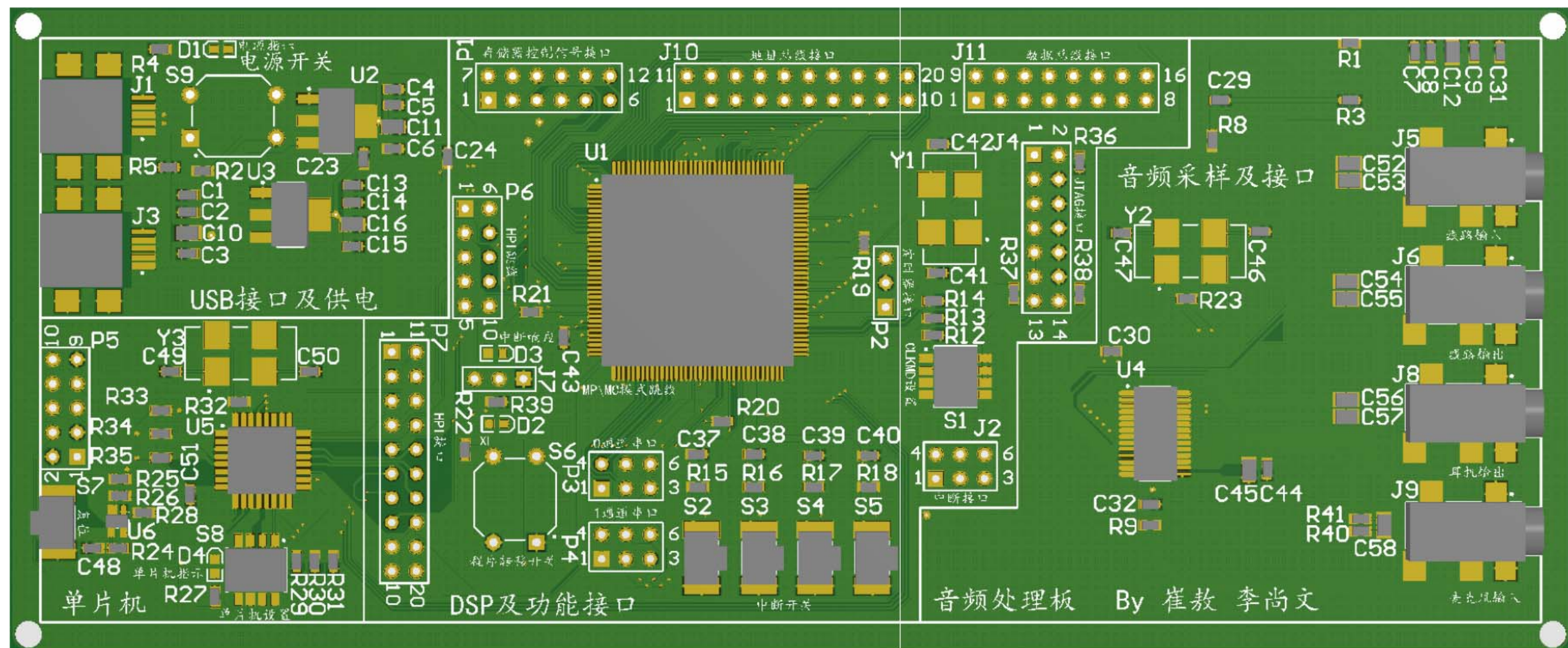
附图 A-6 连接件电路图

附录B PCB 设计



附图 B-1 PCB 设计





附图 B-2 PCB 三维俯视图

## 附录C 程序代码

## DeNoisemMain.asm

```

1; Code for Audio FIR low pass filtering
2; Fs=44100Hz, Fpass=500Hz, Fstop=1500Hz
3; File Name: DeNoiseMain.asm
4; Date: 1st July, 2015
5; Author: Cui Ao, Li Shangwen
6          .title          "DeNoiseMain.asm"
7
8          .mmregs
9
10         .def            _c_int00
11
12         .def            Addr_SPSA0
13         .def            Addr_SPSD0
14         .def            Addr_SPCR1
15         .def            Addr_SPCR2
16         .def            Addr_RCR1
17         .def            Addr_RCR2
18         .def            Addr_XCR1
19         .def            Addr_XCR2
20         .def            Addr_PCR
21         .def            Addr_DRR10
22         .def            Addr_DXR10
23
24         .def            Value_SPCR10
25         .def            Value_SPCR20
26         .def            Value_RCR10
27         .def            Value_RCR20
28         .def            Value_XCR10
29         .def            Value_XCR20
30         .def            Value_PCR0
31
32         .def            CurrentInput
33         .def            CurrentOutput
34
35 FIR_Length          .set            250
36 MusicLength          .set            26
37 SampleRate          .set            16000
38
39 Addr_SPSA0          .set            38h          ;Address of McBSP0 sub-bank address
40 reg.
41 Addr_SPSD0          .set            39h          ;Address of McBSP0 sub-bank data
42 reg.
43 Addr_SPCR1          .set            00h          ;Address of SP control reg 1.
44 Addr_SPCR2          .set            01h          ;Address of SP control reg 2.
45 Addr_RCR1          .set            02h          ;Address of receive control reg. 1
46 Addr_RCR2          .set            03h          ;Address of receive control reg. 2
47 Addr_XCR1          .set            04h          ;Address of transmit control reg. 1
48 Addr_XCR2          .set            05h          ;Address of transmit control reg. 2
49 Addr_PCR            .set            0eh          ;Address of pin control reg.
50 reg. 1
51 Addr_DRR10          .set            21h          ;Address of McBSP0 data receive
52 reg. 1
53 Addr_DXR10          .set            23h          ;Address of McBSP0 transmit reg. 1
54
55 Value_SPCR10        .set            00h
56 Value_SPCR20        .set            0200h
57 Value_RCR10         .set            0140h
58 Value_RCR20         .set            04h
59 Value_XCR10         .set            0140h
60 Value_XCR20         .set            04h
61 Value_PCR0          .set            01h

```

代码 C-1 DSP 代码（以 FIR 低通滤波器为例，不含中断服务代码）

页 1/4



```

59 LPFCOEFF      .word      0xffff,0xffff,0xffff,0xffff,0xffff
60               .word      0xffff,0xffff,0xffff,0xffff,0xffff
61               .word      0xffff,0xffff,0xffff,0xffff,0xffff
62               .word      0xffff,0x0000,0x0000,0x0001,0x0001
63               .word      0x0002,0x0002,0x0003,0x0003,0x0004
64               .word      0x0005,0x0005,0x0006,0x0006,0x0006
65               .word      0x0006,0x0006,0x0005,0x0005,0x0004
66               .word      0x0002,0x0001,0xffff,0xfffd,0xfffb
67               .word      0xfffb,0xfffd,0xfffd,0xfffd,0xfffe
68               .word      0xffeb,0xffe9,0xffe7,0xffe5,0xffe4
69               .word      0xffe3,0xffe3,0xffe4,0xffe6,0xffe8
70               .word      0xffec,0xfffd,0xfffd,0xfffb,0x0002
71               .word      0x000a,0x0012,0x001a,0x0023,0x002c
72               .word      0x0035,0x003d,0x0045,0x004c,0x0052
73               .word      0x0056,0x0059,0x0059,0x0058,0x0054
74               .word      0x004e,0x0046,0x003b,0x002d,0x001d
75               .word      0x000b,0xfffd,0xfffd,0xffc9,0xffb0
76               .word      0xff97,0xff7e,0xff66,0xff4f,0xff3a
77               .word      0xff27,0xff17,0xff0c,0xff05,0xff03
78               .word      0xff07,0xff11,0xff22,0xff39,0xff58
79               .word      0xff7e,0xffac,0xffe1,0x001c,0x005f
80               .word      0x00a7,0x00f5,0x0148,0x019f,0x01f9
81               .word      0x0255,0x02b2,0x030f,0x036a,0x03c3
82               .word      0x0418,0x0469,0x04b4,0x04f7,0x0534
83               .word      0x0567,0x0591,0x05b1,0x05c7,0x05d2
84               .word      0x05d2,0x05c7,0x05b1,0x0591,0x0567
85               .word      0x0534,0x04f7,0x04b4,0x0469,0x0418
86               .word      0x03c3,0x036a,0x030f,0x02b2,0x0255
87               .word      0x01f9,0x019f,0x0148,0x00f5,0x00a7
88               .word      0x005f,0x001c,0xffe1,0xffac,0xff7e
89               .word      0xff58,0xff39,0xff22,0xff11,0xff07
90               .word      0xff03,0xff05,0xff0c,0xff17,0xff27
91               .word      0xff3a,0xff4f,0xff66,0xff7e,0xff97
92               .word      0xffb0,0xffc9,0xffe0,0xfffd,0x000b
93               .word      0x001d,0x002d,0x003b,0x0046,0x004e
94               .word      0x0054,0x0058,0x0059,0x0059,0x0056
95               .word      0x0052,0x004c,0x0045,0x003d,0x0035
96               .word      0x002c,0x0023,0x001a,0x0012,0x000a
97               .word      0x0002,0xfffb,0xfffd,0xfffd,0xffec
98               .word      0xffe8,0xffe6,0xffe4,0xffe3,0xffe3
99               .word      0xffe4,0xffe5,0xffe7,0xffe9,0xffeb
100              .word      0xffee,0xfffd,0xfffd,0xfffd,0xfffd
101              .word      0xfffb,0xfffd,0xfffd,0x0001,0x0002
102              .word      0x0004,0x0005,0x0005,0x0006,0x0006
103              .word      0x0006,0x0006,0x0006,0x0005,0x0005
104              .word      0x0004,0x0003,0x0003,0x0002,0x0002
105              .word      0x0001,0x0001,0x0000,0x0000,0xffff
106              .word      0xffff,0xffff,0xffff,0xffff,0xffff
107              .word      0xffff,0xffff,0xffff,0xffff,0xffff
108              .word      0xffff,0xffff,0xffff,0xffff,0xffff
109
110
111
112              .data
113
114              .bss      CurrentInput,1
115              .bss      CurrentOutput,1
116
117
118 STACK          .usect      "STACK",10h
119

```

代码 C-2 DSP 代码（以 FIR 低通滤波器为例，不含中断服务代码）

页 2/4



```

120 Buffer_Old      .usect      "Buffer_Old",FIR_Length/2
121 Buffer_New      .usect      "Buffer_New",FIR_Length/2
122
123                .text
124
125 LPF:             MVKD        *(CurrentInput),*AR2
126                 RPTZ        B,#(FIR_Length/2-1)
127                 FIRS        *AR2+0%,*AR3+0%,LPFCOEFF
128                 STH         B,*(CurrentOutput)
129                 MAR         *+AR2(1)%
130                 MVDD        *AR2,*AR3+0%
131
132                 STM         #1,AR5
133
134                 RET
135
136 System_Init:     STM         #0000h,CLKMD
137                 RPT         #100h
138                 NOP
139                 STM         #40c7h,CLKMD
140
141                 STM         #0FFFFh,IFR
142                 STM         #0000h,IMR
143                 ST          #0,*(CurrentInput)
144                 ST          #0,*(CurrentOutput)
145                 STM         #Buffer_Old,AR0
146                 RPT         #(FIR_Length/2-1)
147                 ST          #0,*AR0+
148                 STM         #Buffer_New,AR0
149                 RPT         #(FIR_Length/2-1)
150                 ST          #0,*AR0+
151
152                 STM         #1,AR5
153
154                 RET
155
156 McBSP_Init:      STM         Addr_SPCR1,Addr_SPSA0
157                 STM         #0,Addr_SPSD0
158
159                 STM         Addr_SPCR2,Addr_SPSA0
160                 STM         #0,Addr_SPSD0
161
162                 STM         Addr_SPCR1,Addr_SPSA0
163                 STM         #Value_SPCR10,Addr_SPSD0
164
165                 STM         Addr_SPCR2,Addr_SPSA0
166                 STM         #Value_SPCR20,Addr_SPSD0
167
168                 STM         Addr_PCR,Addr_SPSA0
169                 STM         #Value_PCR0,Addr_SPSD0
170
171                 STM         Addr_RCR1,Addr_SPSA0
172                 STM         #Value_RCR10,Addr_SPSD0
173
174                 STM         Addr_RCR2,Addr_SPSA0
175                 STM         #Value_RCR20,Addr_SPSD0
176
177                 STM         Addr_XCR1,Addr_SPSA0
178                 STM         #Value_XCR10,Addr_SPSD0
179
180                 STM         Addr_XCR2,Addr_SPSA0

```

代码 C-3 DSP 代码（以 FIR 低通滤波器为例，不含中断服务代码）

页 3/4

```

181          STM          #Value_XCR20,Addr_SPSD0
182
183          RPT          #10h
184          NOP
185
186          STM          Addr_SPCR1,Addr_SPSA0
187          STM          #0001h,Addr_SPSD0
188
189          STM          Addr_SPCR2,Addr_SPSA0
190          STM          #0201h,Addr_SPSD0
191
192          RPT          #10h
193          NOP
194
195          STM          #0010h,IMR
196
197          RET
198
199
200
201 _c_int00:      STM          #STACK+10h,SP
202               STM          #00E0h,PMST
203
204               SSBX          INTM
205
206               CALL         System_Init
207               CALL         McBSP_Init
208
209
210
211               SSBX          FRCT
212               STM          #Buffer_New,AR2
213               STM          #(Buffer_Old+FIR_Length/2-1),AR3
214               STM          #(FIR_Length/2),BK
215               STM          #-1,AR0
216
217               RSBX          INTM
218
219
220
221 LOOP:         BANZ        LOOP,*AR5
222               CALL         LPF
223               B            LOOP
224
225               .end
226

```

代码 C-4 DSP 代码（以 FIR 低通滤波器为例，不含中断服务代码）

页 4/4

```

1; Code for Audio FIR low pass filtering
2; Fs=44100Hz, Fpass=500Hz, Fstop=1500Hz
3; File Name: IntSer.asm
4; Date: 1st July, 2015
5; Author: Cui Ao, Li Shangwen
6
7             .title      "IntSer.asm"
8             .mmregs
9
10            .def         _c_int01_NMI
11            .def         _c_int02_SINT17
12            .def         _c_int03_SINT18
13            .def         _c_int04_SINT19
14            .def         _c_int05_SINT20
15            .def         _c_int06_SINT21
16            .def         _c_int07_SINT22
17            .def         _c_int08_SINT23
18            .def         _c_int09_SINT24
19            .def         _c_int10_SINT25
20            .def         _c_int11_SINT26
21            .def         _c_int12_SINT27
22            .def         _c_int13_SINT28
23            .def         _c_int14_SINT29
24            .def         _c_int15_SINT30
25            .def         _c_int16_INT0
26            .def         _c_int17_INT1
27            .def         _c_int18_INT2
28            .def         _c_int19_TINT0
29            .def         _c_int20_BRINT0
30            .def         _c_int21_BXINT0
31            .def         _c_int22_DMAC0
32            .def         _c_int23_TINT1
33            .def         _c_int24_INT3
34            .def         _c_int25_HPINT
35            .def         _c_int26_BRINT1
36            .def         _c_int27_BXINT1
37            .def         _c_int28_DMAC4
38            .def         _c_int29_DMAC5
39
40            .ref         Addr_SPSA0
41            .ref         Addr_SPSD0
42            .ref         Addr_SPCR1
43            .ref         Addr_SPCR2
44            .ref         Addr_RCR1
45            .ref         Addr_RCR2
46            .ref         Addr_XCR1
47            .ref         Addr_XCR2
48            .ref         Addr_PCR
49            .ref         Addr_DRR10
50            .ref         Addr_DXR10
51
52            .ref         Value_SPCR10
53            .ref         Value_SPCR20
54            .ref         Value_RCR10
55            .ref         Value_RCR20
56            .ref         Value_XCR10
57            .ref         Value_XCR20
58            .ref         Value_PCR0
59
60            .ref         CurrentInput
61            .ref         CurrentOutput

```

代码 C-5 DSP 中断服务程序

页 1/3

```
62
63
64      .text
65
66 _c_int01_NMI:    NOP
67                  RETE
68
69 _c_int02_SINT17:  NOP
70                  RETE
71
72 _c_int03_SINT18:  NOP
73                  RETE
74
75 _c_int04_SINT19:  NOP
76                  RETE
77
78 _c_int05_SINT20:  NOP
79                  RETE
80
81 _c_int06_SINT21:  NOP
82                  RETE
83
84 _c_int07_SINT22:  NOP
85                  RETE
86
87 _c_int08_SINT23:  NOP
88                  RETE
89
90 _c_int09_SINT24:  NOP
91                  RETE
92
93 _c_int10_SINT25:  NOP
94                  RETE
95
96 _c_int11_SINT26:  NOP
97                  RETE
98
99 _c_int12_SINT27:  NOP
100                 RETE
101
102 _c_int13_SINT28:  NOP
103                 RETE
104
105 _c_int14_SINT29:  NOP
106                 RETE
107
108 _c_int15_SINT30:  NOP
109                 RETE
110
111 _c_int16_INT0:    NOP
112                 RETE
113
114 _c_int17_INT1:    NOP
115                 RETE
116
117 _c_int18_INT2:    NOP
118                 RETE
119
120 _c_int19_TINT0:   NOP
121                 NOP
122                 RETE
```

代码 C-6 DSP 中断服务程序

页 2/3

```

123
124 _c_int20_BRINT0:
125         MVMD      Addr_DRR10,*(CurrentInput)
126         MVDM      *(CurrentOutput),Addr_DXR10
127         STM       #0,AR5
128         RETE
129
130 _c_int21_BXINT0:  NOP
131         RETE
132
133 _c_int22_DMAC0:   NOP
134         RETE
135
136 _c_int23_TINT1:  NOP
137         RETE
138
139 _c_int24_INT3:   NOP
140         RETE
141
142 _c_int25_HPINT:  NOP
143         RETE
144
145 _c_int26_BRINT1: NOP
146         RETE
147
148 _c_int27_BXINT1: NOP
149         RETE
150
151 _c_int28_DMAC4:  NOP
152         RETE
153
154 _c_int29_DMAC5:  NOP
155         RETE
156
157         .end
158

```

代码 C-7 DSP 中断服务程序

页 3/3

```

// Code for 8051 & 5402 HPI Boot and Initialing AIC23
//File Name: boot.c
//Date: 1st July, 2015
//Author: Cui Ao, Li Shangwen
#include <Reg52.h>
#include <absacc.h>
#include <stdio.h>
#include <intrins.h>
#include <LED Flash.h>
#include <Music LPF.h>
#include <AIC23 Through.h>
#include <HPF.h>
#include <BPF.h>
#include <Alarm.h>
/*定义宏*/
#define uint unsigned int
#define uchar unsigned char

#define L LINE VOLUME      0x00
#define R LINE VOLUME      0x01
#define L HEADPHONE VOLUME 0x02
#define R HEADPHONE VOLUME 0x03
#define A AUDIO PATH       0x04
#define D AUDIO PATH       0x05
#define POWER CON          0x06
#define D AUDIO INTERFACE  0x07
#define SAMPLE RATE        0x08
#define D INTERFACE ACT    0x09
#define RESET              0x0f

/*定义寄存器端口地址*/
#define HPICL XBYTE[0xf000]
#define HPICH XBYTE[0xf100]
#define WR_HPIL0_D XBYTE[0xf200]
#define WR_HPIH0_D XBYTE[0xf300]
#define WR_HPIL_A XBYTE[0xf400]
#define WR_HPIH_A XBYTE[0xf500]
#define WR_HPIL1_D XBYTE[0xf600]
#define WR_HPIH1_D XBYTE[0xf700]

#define RD_HPIL0_D XBYTE[0xfa00]
#define RD_HPIH0_D XBYTE[0xfb00]
#define RD_HPIL1_D XBYTE[0xfe00]
#define RD_HPIH1_D XBYTE[0xff00]

/*定义IO端口*/
sbit SDIN =P1^2; //aic23数据输入端口
sbit SCLK =P1^1; //aic23数据 strobe
sbit CS =P1^0; //aic23片选端口

/*定义IO端口*/
sbit NRST =P2^4;
sbit LED =P1^3;

bit ok;
unsigned char ms;
unsigned char volume;
int Msg data=0;
unsigned char Flag=0;

void write_AIC23(uchar addr,uint dat);
void init_AIC23();

void wr_dspscode(unsigned char Mode);

```

代码 C-8 8051 HPI 加载 DSP 及 AIC23 初始化代码

页 1/4

```

void LOADDSP();
void Nrst();
void delay(uint wu);

void delay1(unsigned char k);
void Mscmm_init()
{
    EA=1;           //开总中断
    ES=1;           //串口中断

    TMOD=0X21;
    SCON=0X50;
    PCON=0X00;
    TH1=0xfd;
    TL1=0xfd;
    TR1=1;
    REN=1;
}

void Send_Record(unsigned char Msg_data) //EEPROM
{
    ES=0;

    SBUF=Msg_data;
    while(TI==0);
    TI=0;

    ES=1;
}

unsigned char getDSP();

main()
{
    Mscmm_init();
    init_AIC23();
    delay(10000);

    LED = 1;

    delay(10000);
    LED = 1;
    delay(20000);
    LED = 0;
    volume=100;
    write_AIC23(L_HEADPHONE_VOLUME, 0x0180+volume);
    LED = 0;
    while(1)
    {
        LOADDSP();
    }
}

void delay(uint wu)
{
    for(wu;wu>0;wu--);
}

void Nrst()
{
    NRST=0;
    delay(1000);
    NRST=1;
    delay(1000);
}

```

代码 C-9 8051 HPI 加载 DSP 及 AIC23 初始化代码

页 2/4

```

void LOADDSP()
{
    if(Flag)
    {
        Nrst();
        wr dspcode(Msg_data);
        Flag=0;
    }
}

void wr_dspcode(unsigned char Mode)
{
    uchar code *dspptr,*temp;
    uchar len;
    char i;
    uint addr;

    HPICL=0x08; //clear HPIint,least is first
    HPICH=0x08;
    switch(Mode)
    {
        case 1:dspptr=Alarm;break;
        case 2:dspptr=LED_Flash;break;
        case 3:dspptr=AIC23_Through;break;
        case 4:dspptr=Music_LPF;break;
        case 5:dspptr=HPF;break;
        case 6:dspptr=BPF;break;
    }

    while(1)
    {
        len=*dspptr++;
        if(len==0) break;
        else
        {
            addr=((*dspptr++)*256)+(*dspptr++);
            addr--;
            temp=dspptr;

again:
            WR HPIL A=addr/256;
            WR HPIH A=addr%256;
            for(i=0;i<len/2;i++)
            {
                WR HPIL0 D=*dspptr++;
                WR HPIH0_D=*dspptr++;
            }
            dspptr=temp;
            addr++;
            WR HPIL A=addr/256;
            WR HPIH A=addr%256;
            for(i=0;i<len/2;i++)
            {
                if(RD HPIL0 D!=*dspptr++) goto again; //不相等，再写
                if(RD HPIH0_D!=*dspptr++) goto again;
            }
        }
    }
    WR HPIL A=0x00;
    WR HPIH A=0x7f;
    WR HPIL1 D=0x00; //THE START ADDRESS OF RUN
    WR HPIH1_D=0x80;
}

void init_AIC23()

```

代码 C-10 8051 HPI 加载 DSP 及 AIC23 初始化代码

页 3/4



```

{
    write_AIC23(RESET,0);
    write_AIC23(D_INTERFACE_ACT,0x001); //Active interface
    write_AIC23(POWER_CON,0); // 开启所有模块power
    write_AIC23(SAMPLE_RATE,0x022); //44.1k
    // write_AIC23(SAMPLE_RATE,0x062); //22.05k
    //write_AIC23(SAMPLE_RATE,0x02e); //8.021k

    write_AIC23(L_LINE_VOLUME,0x0117);
    write_AIC23(L_LINE_VOLUME,0x0117); //线输入音量+12dB

    volume=100;
    write_AIC23(L_HEADPHONE_VOLUME,0x0180+volume);
    write_AIC23(R_HEADPHONE_VOLUME,0x0180+volume); //音量控制

    // write_AIC23(A_AUDIO_PATH,0x009); //bypass DAC OFF
    // write_AIC23(A_AUDIO_PATH,0x014); //MIC ADC DAC ON 0DB
    // write_AIC23(A_AUDIO_PATH,0x1fc); //bypass MIC ADC DAC ON 0DB sidetone
    //write_AIC23(A_AUDIO_PATH,0x07d); //bypass MIC ADC DAC ON 20DB sidetone(-6db)
    // write_AIC23(A_AUDIO_PATH,0x011); //LINE ADC DAC ON

    write_AIC23(D_AUDIO_PATH,0x04);
    write_AIC23(D_AUDIO_INTERFACE,0x043); //master dsp mode
}

void write_AIC23(uchar addr,uint dat)
{
    uchar i;
    dat|=addr<<9;
    CS=0;
    for(i=0;i<16;i++)
    {
        SCLK=0;
        if((dat<<i)&0x8000)
            SDIN=1;
        else
            SDIN=0;
        SCLK=1;
    }
    SCLK=0;
    CS=1;
}

void SCON_1(void) interrupt 4 //
{
    Msg_data=SBUF;
    RI=0; //
    TI=0; //
    Send_Record(Msg_data);
    Flag=1;
}

```

代码 C-11 8051 HPI 加载 DSP 及 AIC23 初始化代码

页 4/4