

DSP

DSP YINGYONG XITONG
KAIFA SHILI

李黎 魏伟 编著

应用系统开发实例 ——基于TMS320F281x和C语言

第 ❶ 篇 开发基础篇，重点介绍了DSP系统

开发的基础知识和集成开发环境

第 ❷ 篇 模块实例篇，通过8个设计实例，详

细介绍了TMS320F281x DSP基础模块各种开

发技术和使用技巧，每个实例基础实用、易学

易懂

第 ❸ 篇 综合应用篇，是本书的重点，精选

了10个DSP系统综合应用实例，并给出了完

整的设计过程



化学工业出版社



应用系统开发实例

——基于TMS320F281x和C语言

李黎 魏伟 编著



化学工业出版社

·北京·

本书针对目前通用流行的 TMS320F281x DSP 芯片, 通过大量实例详细介绍了 DSP 基础模块与综合系统设计的方法及技巧。全书共分 3 篇 13 章。第 1 篇为开发基础篇, 重点介绍了 DSP 系统开发的基础知识和集成开发环境。第 2 篇为模块实例篇, 通过 8 个设计实例, 详细介绍了 TMS320F281x DSP 基础模块各种开发技术和使用技巧, 每个实例基础实用、易学易懂。第 3 篇为综合应用篇, 是本书的重点, 精选了 10 个 DSP 系统综合应用实例, 并给出了完整的设计过程。

本书语言简洁, 层次分明, 精选的每个实例都对它的实例功能、设计思路、工作原理、硬件电路、软件设计、参考程序做了详细的描述和注释, 为读者提供了一套完整的 TMS320F281x DSP 芯片开发设计手册。

本书可作为从事 DSP 芯片开发的工程技术人员的一本实用的参考书, 也适合高校计算机、自动化、电子及通信等相关专业的师生使用。

图书在版编目 (CIP) 数据

DSP 应用系统开发实例: 基于 TMS320F281x 和 C 语言/

李黎, 魏伟编著. —北京: 化学工业出版社, 2018. 1

ISBN 978-7-122-31153-5

I. ①D… II. ①李…②魏… III. ①数字信号处理

IV. ①TN911.72

中国版本图书馆 CIP 数据核字 (2017) 第 300811 号

责任编辑: 李军亮 万忻欣

文字编辑: 吴开亮

责任校对: 边 涛

装帧设计: 刘丽华

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 刷: 三河市航远印刷有限公司

装 订: 三河市瞰发装订厂

787mm×1092mm 1/16 印张 21 字数 551 千字 2018 年 4 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 88.00 元

版权所有 违者必究

前言

Preface

DSP 系统开发是硬件、软件相结合的过程。要完成 DSP 系统的开发,不仅要掌握编程技术,而且还要针对实际应用选择合理的 DSP 芯片和外围电路,并以此为基础,设计硬件电路。本书针对目前通用流行的 TMS320F281x DSP 芯片,重点介绍 DSP 系统开发实例的完整过程,以 DSP 系统开发的应用为主,介绍 DSP 系统开发的设计和实现方法,使读者通过本书的学习,掌握 TMS320F281x DSP 芯片综合设计方法与技巧。

本书具有以下特点:

① 本书是一本专门介绍 TMS320F281x DSP 芯片应用实例的书,并以“由浅入深”“相互贯穿”“重点突出”“文字叙述与典型实例相结合”为原则,向读者全面介绍 DSP 芯片开发的完整设计过程。

② 本书突破了传统的软硬件截然割裂的方法,使读者对 DSP 芯片实际工程应用技术能够独立进行 DSP 芯片的软硬件开发。可节省读者进入 DSP 芯片开发领域的时间,同时能更清楚认识 DSP 芯片相关开发工具的使用及应用技巧。

③ 本书从应用的角度出发,结合了作者多年教学、科研实践的经验,系统、全面地以 DSP 芯片应用为例介绍系统开发的完整过程,是一本重在实际应用的实用手册。

④ 实例多。本书提供了多个典型实例,覆盖领域较广,代表性强,通过大量的 DSP 芯片应用实例阐述了基本设计过程,读者在学习的过程中可较为容易地掌握 DSP 芯片开发的完整过程。

⑤ 本书在内容的选择和安排上,着重突出了“应用”和“实用”两个原则。给出的实例是作者多年 DSP 芯片开发项目精选出来的,也是经验的归纳与总结。程序代码部分做了较为详细的注释,有利于读者举一反三,快速应用与提高。

本书内容系统全面,论述深入浅出,循序渐进,硬件设计和软件设计相结合。本书是从事 DSP 嵌入式系统开发应用与产品开发的工程技术人员的一本实用的参考书,也可以作为电子信息工程、通信工程、自动化等相关专业的高年级本科生和研究生的参考书。

限于作者水平,书中难免存在不足之处,恳请读者批评指正。

编著者

目录

Contents

第1篇

开发基础篇

1

第1章 ▶ DSP 应用系统开发基础	1
1.1 DSP 应用系统开发流程	1
1.1.1 DSP 总体方案设计	1
1.1.2 DSP 芯片选型	2
1.1.3 硬件电路设计	4
1.1.4 软件程序设计	5
1.1.5 DSP 系统集成	6
1.2 DSP 应用系统开发工具	7
1.2.1 软件开发工具	7
1.2.2 硬件开发工具	7
1.3 实例：基于 TMS320F2812A DSP 的最小系统设计	8
第2章 ▶ DSP 集成开发环境	10
2.1 CCS 简介	10
2.1.1 CCS 概述	10
2.1.2 代码生成工具	10
2.2 CCS3.3 的基本应用	12
2.2.1 开发 TMS320C28xx 应用系统环境	12
2.2.2 CCS3.3 安装	12
2.2.3 CCS3.3 设置	12
2.2.4 启动 CCS3.3 仿真	15
2.3 实例：用 CSS3.3 开发一个音频信号采集、处理输出的程序	19
2.3.1 实例目的	19
2.3.2 实例原理	19
2.3.3 实例步骤	20
2.3.4 实例结果	25

第2篇

模块实例篇

26

第3章 ▶ 数字输入/输出模块	26
3.1 数字 I/O 端口概述	26

3.2	数字 I/O 端口寄存器	26
3.2.1	I/O 复用寄存器	27
3.2.2	I/O 数据寄存器	27
3.2.3	GPIO 多路复用器的寄存器	29
3.2.4	GPIO 寄存器基本功能	32
3.3	实例: I/O 端口应用——键盘接口设计	33
3.3.1	键盘接口的硬件设计	33
3.3.2	键盘接口的软件设计	34
3.4	实例: I/O 端口应用——LCD 接口设计	35
3.4.1	LCD 显示接口的硬件设计	35
3.4.2	LCD 显示接口的软件设计	36
第 4 章	事件管理器模块	47
4.1	事件管理器模块概述	47
4.1.1	事件管理器结构框图	47
4.1.2	事件管理器寄存器地址列表	48
4.1.3	事件管理器中断	50
4.2	通用定时器	55
4.2.1	通用定时器概述	55
4.2.2	通用定时器功能模块	55
4.2.3	通用定时器的计数操作	61
4.3	PWM 电路	64
4.3.1	有比较单元的 PWM 电路	64
4.3.2	PWM 信号的产生	67
4.3.3	空间向量 PWM	70
4.4	实例: 事件管理器应用——产生 PWM 波	73
4.5	实例: 事件管理器——捕获 PWM 波	75
第 5 章	模数转换模块	78
5.1	ADC 概述	78
5.2	自动排序器的工作原理	79
5.2.1	连续的自动排序模式	80
5.2.2	排序器的启动/停止模式	80
5.2.3	输入触发源	80
5.3	ADC 时钟预定标	81
5.4	低功耗方式	81
5.5	功耗上升顺序	82
5.6	ADC 模块的寄存器	82
5.7	ADC 转换时钟周期	89
5.8	实例: ADC 的应用	90
第 6 章	串行外设接口模块 (SPI)	93
6.1	串行外设接口概述	93

6.2	串行外设接口操作	94
6.2.1	操作介绍	94
6.2.2	串行外设接口模块的主动和从动方式	94
6.2.3	串行外设接口中断	95
6.2.4	数据格式	95
6.2.5	串行外设接口波特率设置和时钟方式	96
6.2.6	串行外设接口的初始化	98
6.3	串行外设接口控制寄存器	98
6.4	实例: SPI 端口输出 DAC 串行数据	103

第 7 章 ▶ 串行通信接口模块 (SCI) 106

7.1	串行通信接口概述	106
7.2	多处理器 (多机) 异步通信模式	107
7.2.1	串行通信接口可编程的数据格式	107
7.2.2	串行通信接口的多处理器通信	108
7.2.3	串行通信接口通信格式	108
7.2.4	串行通信接口中断	110
7.2.5	串行通信接口波特率计算	110
7.2.6	串行通信接口增强特征	111
7.3	串行通信接口控制寄存器	113
7.4	实例: SCI 的应用——串行通信接口与 RS-232 串行口的异步通信	119
7.4.1	串行通信硬件电路设计	119
7.4.2	串行通信软件设计	120

第 8 章 ▶ eCAN 控制器模块 123

8.1	eCAN 控制器模块概述	123
8.1.1	eCAN 技术简介	123
8.1.2	F2812 eCAN 控制器概述	124
8.2	邮箱	125
8.2.1	eCAN 信息包格式说明	125
8.2.2	eCAN 邮箱寄存器	125
8.3	eCAN 控制寄存器	127
8.4	eCAN 控制器的操作	134
8.4.1	初始化 eCAN 控制器	135
8.4.2	信息的发送	136
8.4.3	信息的接收	136
8.4.4	远程帧	137
8.4.5	中断	138
8.5	实例: 采用 eCAN 控制器发送和接收消息	142

第9章 ▶ DSP 在电力系统中的应用	147
9.1 实例：光伏并网逆变器设计	147
9.1.1 实例功能	147
9.1.2 设计思路	147
9.1.3 工作原理	148
9.1.4 硬件电路	150
9.1.5 软件设计	154
9.1.6 参考程序	157
9.2 实例：风力发电并网逆变器设计	168
9.2.1 实例功能	168
9.2.2 设计思路	168
9.2.3 工作原理	169
9.2.4 硬件电路	170
9.2.5 软件设计	174
9.2.6 参考程序	176
第10章 ▶ DSP 在开关电源中的应用	182
10.1 实例：直流斩波电源的设计	182
10.1.1 实例功能	182
10.1.2 工作原理	182
10.1.3 硬件电路	183
10.1.4 软件设计	186
10.1.5 参考程序	187
10.2 实例：三相高精度逆变电源的设计	193
10.2.1 实例功能	193
10.2.2 工作原理	193
10.2.3 硬件电路	196
10.2.4 软件设计	199
10.2.5 参考程序	200
第11章 ▶ DSP 在电机控制系统中的应用	232
11.1 实例：异步电动机矢量控制设计	232
11.1.1 实例功能	232
11.1.2 工作原理	232
11.1.3 硬件电路	234
11.1.4 软件设计	237
11.1.5 参考程序	242
11.2 实例：感应电动机软启动器设计	246
11.2.1 实例功能	247
11.2.2 工作原理	247

11.2.3	硬件电路	248
11.2.4	软件设计	250
11.2.5	参考程序	253
第 12 章	DSP 在检测及控制系统中的应用	259
12.1	实例: 三相交流参数测试仪的设计	259
12.1.1	实例功能	259
12.1.2	工作原理	259
12.1.3	硬件电路	264
12.1.4	软件设计	266
12.1.5	参考程序	270
12.2	实例: 新型多电平混合级联逆变器设计	277
12.2.1	实例功能	277
12.2.2	工作原理	278
12.2.3	硬件电路	281
12.2.4	软件设计	283
12.2.5	参考程序	286
第 13 章	DSP 在电力电子装置中的应用	287
13.1	实例: 并联混合有源滤波器的设计	287
13.1.1	实例功能	287
13.1.2	设计思路	288
13.1.3	工作原理	289
13.1.4	硬件电路	298
13.1.5	软件设计	301
13.1.6	参考程序	303
13.2	实例: 新型谐振阻抗型混合有源滤波器的设计	314
13.2.1	实例功能	314
13.2.2	工作原理	314
13.2.3	硬件电路	318
13.2.4	软件设计	322
13.2.5	参考程序	323
参考文献		328

第①篇 开发基础篇

第 1 章 ▶ DSP 应用系统开发基础

本章主要讲述 DSP 应用系统开发的基础知识，共分三部分。第一部分是关于 DSP 应用系统开发的基础知识，主要包括 DSP 总体方案设计、DSP 芯片选型、硬件电路设计、软件程序设计、DSP 系统集成；第二部分是关于 DSP 应用系统开发工具的知识，分别介绍了常用 DSP 的软件和硬件开发工具；第三部分简单介绍了 1 个 DSP 应用实例，即基于 TMS320F2812 DSP 的最小系统设计的基本步骤和方法。

在设计一个 DSP 应用系统时，不仅要熟悉芯片的硬件结构、指令系统等，还要熟悉 DSP 开发、调试工具的使用，从而使后续各章的学习目标更加明确。充分理解这些知识对于后续各章的学习具有非常重要的作用。

1.1 DSP 应用系统开发流程

1.1.1 DSP 总体方案设计

利用 DSP 芯片设计一个 DSP 系统的大致步骤如图 1.1 所示。现对图 1.1 所列各步骤作简要说明。

在进行 DSP 系统设计之前，首先要明确设计任务，给出设计任务书。在设计任务书中，应该将系统要达到的功能描述准确、清楚。描述的方式可以是人工语言，也可以是流程图或算法描述。在此之后应该把设计任务书转化为量化的技术指标。结合 DSP 系统的设计，这些技术指标主要包括：

- ① 由信号的频率决定的系统采样频率。
- ② 由采样频率完成任务书最复杂的算法所需最大时间及系统对实时程度的要求判断系统能否完成工作。
- ③ 由数据量及程序的长短决定片内 RAM 的容量，是否需要扩展片外 RAM 及片外 RAM 容量。

④ 由系统所要求的精度决定是 16 位还是 32 位，是定点还是浮点运算。

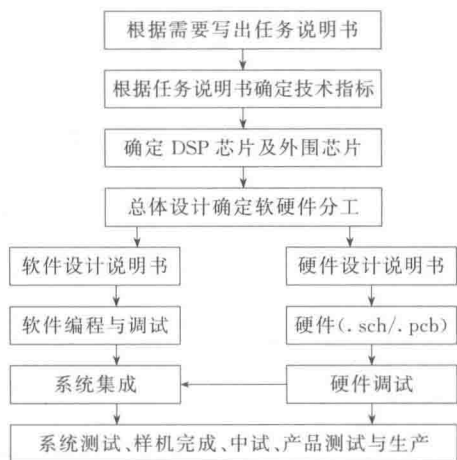


图 1.1 DSP 总体设计框图

⑤ 根据系统是计算用还是控制用来决定对输入输出端口的要求。在一些特殊的控制场合还有一些专门的芯片可供选用。如电机控制领域很适合用 TMS320C2812 系列，因为它上面集成了 2 路 A/D 输入、6 路 PWM 输出及强大的人机接口。

由上述的一些技术指标，大致可以确定应该选用的 DSP 芯片的型号。根据选用的 DSP 芯片及上述技术指标，还可以初步确定 A/D、D/A、RAM 的性能指标及可供选择的产品。当然在产品选型时，还须考虑：成本、供货能力、技术支持（资料、第三方部门）、开发系统（开发系统可能很贵，这时还要考虑成本）、体积、功耗、工作环境温度（这在一些场合是非常重要的）。

在确定 DSP 芯片选型之后，应当先进行系统的总体设计。首先采用高级语言或 MATLAB 等对算法进行仿真，确定最佳算法并初步确定参数，对系统中的哪些功能用软件来实现、哪些功能用硬件实现进行初步的分工，如 FFT、FIR 等是否需要用专用芯片来实现等。

1.1.2 DSP 芯片选型

在设计 DSP 应用系统时，选择 DSP 芯片是非常重要的一个环节。只有选定了 DSP 芯片才能进一步设计其外围电路及系统的其他电路。总的来说，DSP 芯片的选择应根据实际的应用系统需要而确定。随着应用场合和设计目标的不同，DSP 选择的依据重点也不同，通常需要考虑以下因素：

（1）DSP 芯片的运算速度

运算速度是 DSP 芯片一个最重要的性能指标，也是选择 DSP 芯片时所需要考虑的一个主要因素。设计者先由输入信号的频率范围确定系统的最高采样频率，再根据算法的运算量和实时处理限定的完成时间确定 DSP 运算速度的下限。DSP 芯片的运算速度可以用以下几种指标来衡量。

① 指令周期：即执行一条指令所需的时间，通常以纳秒（ns）为单位。如 TMS320F2812A 在主频为 150MHz 时的指令周期为 7ns。

② MAC 时间：即一次乘法加上一次加法的时间。大部分 DSP 芯片可在一个指令周期内完成一次乘法和加法操作，如 TMS320F2812A 的 MAC 时间就是 7ns。

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

④ MIPS：即每秒执行百万条指令。如 TMS320F2812A 的处理能力为 150MIPS，即每秒可执行 1.5 亿条指令。

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

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

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

(2) DSP 芯片的运算精度

由系统所需要的精度确定是采用定点运算还是浮点运算。

参加运算的数据字长越长精度越高,目前,除少数 DSP 处理器采用 20 位、24 位或 32 位的格式外,绝大多数定点 DSP 都采用 16 位数据格式。由于其功耗小和价格低廉,实际应用的 DSP 处理器绝大多数是定点处理器。

为了保证底数的精度,浮点 DSP 的数据格式基本上都做成 32 位,其数据总线、寄存器、存储器等的宽度也相应是 32 位。在实时性要求很高的场合,往往考虑使用浮点 DSP 处理器。与定点 DSP 处理器相比,浮点 DSP 处理器的速度更快,但价格比较高,开发难度也更大一些。

(3) 片内硬件资源

由系统数据量的大小确定所使用的片内 RAM 及需要扩展的 RAM 的大小;根据系统是作计算用还是控制用来确定 I/O 端口的需求。

不同的 DSP 芯片所提供的硬件资源是不相同的,如片内 RAM、ROM 的数量,外部可扩展的程序和数据空间,总线接口、I/O 接口等。即使是同一系列的 DSP 芯片(如 TI 的 TMS320C54x 系列),系列中不同 DSP 芯片也具有不同的内部硬件资源,以适应不同的需要。在一些特殊的控制场合有一些专门的芯片可供选用,如 TMS320C281x 系列自身带有 2 路 A/D 输入和 6 路 PWM 输出及强大的人机接口,特别适合于电动机控制场合。

(4) DSP 芯片的功耗

在某些 DSP 应用场合,功耗也是一个很重要的问题。功耗的大小意味着发热的大小和能耗的多少。如便携式的 DSP 设备、手持设备(手机)和野外应用的 DSP 设备,对功耗都有特殊的要求。

(5) DSP 芯片的开发工具

快捷、方便的开发工具和完善的软件支持是开发大型复杂 DSP 系统必备的条件,有强大的开发工具支持,就会大大缩短系统开发时间。现在的 DSP 芯片都有较完善的软件和硬件开发工具,其中包括 Simulator 软件仿真器、Emulator 在线仿真器和 C 编译器等。如 TI 公司的 CCS 集成开发环境、XDSP 实时软件技术等,为用户快速开发实时高效的应用系统提供了巨大帮助。

(6) DSP 芯片的价格

在选择 DSP 芯片时一定要考虑其性能价格比。如价格过高,即使其性能较高,在应用中也会受到一定的限制,如应用于民用品或批量生产的产品中就需要较低廉的价格。另外,DSP 芯片发展迅速,价格下降也很快。因此在开发阶段可选择性能高、价格稍贵的 DSP 芯片,等开发完成后,会具有较高的性价比。

(7) 其他因素

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

上述各因素中,确定 DSP 应用系统的运算量是非常重要的,它是选用不同处理能力的 DSP 芯片的基础,运算量小则可以选用处理能力不是很强的 DSP 芯片,从而可以降低系统成本。相反,运算量大的 DSP 系统则必须选用处理能力强的 DSP 芯片,如果 DSP 芯片的处

理能力达不到系统要求，则必须用多个 DSP 芯片并行处理。如何确定 DSP 系统的运算量并选择 DSP 芯片，主要考虑以下两种情况。

① 按样点处理 所谓按样点处理，就是 DSP 算法对每一个输入样点循环一次。数字滤波就是这种情况，在数字滤波器中，通常需要对每一个输入样点计算一次。

例如，一个采用 LMS 算法的 256 抽头的自适应 FIR 滤波器，假定每个抽头的计算需要 3 个 MAC 周期，则 256 抽头计算需要

$$256 \times 3 = 768 \text{ 个 MAC 周期}$$

如果采样频率为 8kHz，即样点之间的间隔为 125μs，DSP 芯片的 MAC 周期为 200ns，则 768 个 MAC 周期需要

$$768 \times 200\text{ns} = 153.6\mu\text{s}$$

由于计算 1 个样点所需的时间 153.6μs 大于样点之间的间隔 125μs，显然无法实时处理，需要选用速度更高的 DSP 芯片。

若选 DSP 芯片的 MAC 周期为 100ns，则 768 个 MAC 周期需要

$$768 \times 100\text{ns} = 76.8\mu\text{s}$$

由于计算 1 个样点所需的时间 76.8μs 小于样点之间的间隔 125μs，可实现实时处理。

② 按帧处理 有些数字信号处理算法不是每个输入样点循环一次，而是每隔一定的时间间隔（通常称为帧）循环一次。中低速语音编码算法通常以 10ms 或 20ms 为一帧，每隔 10ms 或 20ms 语音编码算法循环一次。所以，选择 DSP 芯片时应该比较一帧内 DSP 芯片的处理能力和 DSP 算法的运算量。

例如，假设 DSP 芯片的指令周期为 p ，一帧的时间为 Δt ，则该 DSP 芯片在一帧内所能提供的最大运算量为

$$\text{最大运算量} = \Delta t / p \text{ 条指令}$$

例如 TMS320VC5402-100 的指令周期为 10ns，设帧长为 20ms，则一帧内 TMS320VC5402-100 所能提供的最大运算量为

$$\text{最大运算量} = 20\text{ms} / 10\text{ns} = 200 \text{ 万条指令}$$

因此，只要语音编码算法的运算量不超过 200 万条指令（单周期指令），就可以在 TMS320VC5402-100 上实时运行。

1.1.3 硬件电路设计

DSP 硬件系统可能由一个 DSP 及外围总线组成，也可能由多个 DSP 组成，这完全取决于 DSP 处理的要求。DSP 硬件系统的主要任务是将前向通道输出的信号按照一定的算法进行处理，然后将处理的结果以数据流的形式输出给后向通道。后向通道主要由 D/A、f/V、平滑滤波器及功率放大器等部分组成。DSP 硬件系统设计阶段一般分为以下几步进行。

第一步：设计硬件实现方案。

所谓硬件实现方案是指根据性能指标、工期、成本等，确定最优硬件实现方案（考虑到实际的工作情况，最理想的方案不一定是最优的方案），并画出其硬件系统框图，如图 1.2 所示。这时对于具体器件的要求应该已经比较明确。

第二步：进行器件的选型。

一般系统中常用 A/D、D/A、内存、电源、逻辑控制、通信、人机接口、总线等基本部件。下面将大致介绍它们的确定

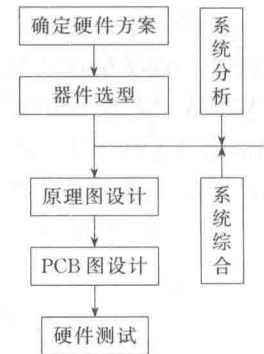


图 1.2 硬件设计系统框图

原则。

A/D: 根据采样频率、精度来确定 A/D 型号, 是否要求芯片自带采样保护、多路器、基准电源等。

D/A: 信号频率、精度是否要求自带基准电源、多路器、输出运放等。

内存: 包括 SRAM、EPROM (或 EEPROM、FLASH MEMORY), 在 TMS320C6000 等一些产品中还有 SDRAM、SBSRAM。所有这些的选型主要考虑工作频率、内存容量位长 (8 位/16 位/32 位)、接口方式 (串行还是并行)、工作电压是 5V 还是 3.3V 或其他。

逻辑控制: 首先是确定用 PLD、CPLD 还是 FPGA。其次根据自己的特长和公司芯片的特点决定采用哪家公司的哪一系列的产品。最后还须根据 DSP 的频率决定芯片的工作频率以确定使用的芯片。

通信: 通信的要求一般系统都是需要的。首先需要根据通信的速率决定采用的通信方式。一般采用串口只能到达 19.2kbps (RS-232), 而并口则可达到 1Mbps 以上。如果还有更高的要求则应考虑通过总线进行通信。

总线: 一般有 PCI、ISA、现场总线如 CAN, 3×bus 等。采用哪一种总线主要看使用的场合、数据传输速率的高低 (总线宽度、频率高低、同步方式等)。

人机接口: 有键盘、显示器等, 它们可以通过与 80C196 等单片机的通信来构成, 也可以在 DSP 的基础上直接构成, 视情况而定。

电源: 主要是电压的高低以及电流的大小。电压要匹配, 电流容量要足够。

上述这些部件的选择可能会相互有些影响。同时, 在选型时还必须充分考虑到供货能力、性能价格比、技术支持、使用经验等因素。

第三步: 进行原理图设计。

在这一步之前的工作基本上是分析工作。而从这一步起, 则开始综合的工作, 逐步开始系统的集成。在所有的综合工作中, 原理图的设计是关键的一步。在原理图的设计时必须清楚了解器件的使用和系统的开发, 对于一些关键的环节有必要做一定的仿真。随着大规模集成芯片和可编程逻辑芯片的发展, 硬件原理设计的难度得以降低, 但它依然是 DSP 系统集成中关键的一步。原理图设计的成功与否是 DSP 系统能否正常工作的最重要的一个因素。

第四步: PCB 图设计。

PCB 图的设计要求 DSP 系统的设计人员既要熟悉系统工作原理, 还要清楚布线工艺和系统结构设计。

第五步: 硬件调试。

对 DSP 系统整体的硬件方案的确定、各种器件的选型、原理的设计和各種硬件电路进行详细分析与调试。

1.1.4 软件程序设计

在成功的 DSP 技术应用中, 好的编程技术扮演了重要的角色。开始开发时, 首先要确定编程的结构方法和好的证明方法。在写任何程序步骤之前, 花一些时间制定信号处理任务的综合计划有好处。这个计划应该考虑到需要的存储器大小、强加在处理器上程序长度方面的限制、执行时间等。图 1.3 所示为 DSP 软件开发的流程图。该图详细指出了—个典型工程所需要的步骤。这里的描述比较完整, 在流程图的中间阶段, 所需文件的输入与输出类型也给出来了。

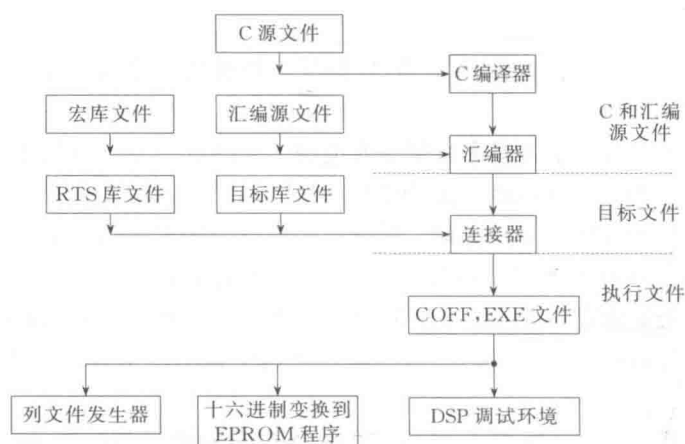


图 1.3 DSP 软件开发的流程图

(1) DSP 软件编程的特点

在此仅对 DSP 系统软件开发流程做简单的介绍。

① 与计算机的汇编语言比起来，由于 TI 公司汇编语言的指令系统比计算机汇编语言的指令系统要简单一些，而且由于有许多专门为数字信号处理器而设计的指令，因此比较容易掌握并运用于数字信号处理的编程中。

② 与高级语言比起来，使用 DSP 汇编语言的用户一定要熟悉 DSP 芯片内部结构和指令系统。尤其是在多 DSP 并行处理的场合，或在便携电话、磁盘驱动器等编程空间很小的场合，这对偏重高效的 DSP 软件是非常重要的。

③ 高级语言（如 C 语言）的开发工具不断完善，随着 TI 公司 C 语言编译器、优化器的不断改进，以及一些第三部门的不断努力，C 语言的编译效率已经得到了很大的提高。在 C3X 中，其编译效率大约为汇编语言的 1/10，而到了 C6X 系列，其编译效率提高了 3 倍。

④ 在实时要求高的场合或实时要求高的算法中，用汇编语言开发；实时要求低的场合用 C 语言编程。将两者结合起来，既能保持算法的实时性，又能做到程序结构的清晰明了。

(2) 软件编程的步骤

① 用汇编语言、C 语言或汇编语言和 C 语言的混编来编写程序，然后把它们分别转化成 TMS320 的汇编语言并送到汇编语言编译器进行编译，生成目标文件。

② 将目标文件送入连接器进行连接，得到可执行文件。

③ 将可执行文件调入到调试器（包括软件仿真、软件开发系统、评测模块、系统仿真器，一般在系统调试中，系统仿真器是最常用的）进行调试，检查运行结果是否正确。如果正确进入第④步；如果不正确，则返回第一步。

④ 进行代码转换，将代码写入 FLASH ROM，并脱离仿真器运行程序，检查结果是否正确。如果不正确，返回第③步；如果正确，进入下一步。

⑤ 软件测试。如果测试结果合格，软件调试完毕；如果不合格，返回第一步。

1.1.5 DSP 系统集成

在完成系统的软硬件设计之后，将进行 DSP 系统集成。所谓系统集成是将软硬件结合起来，组装成一台样机，并在实际 DSP 系统中运行，进行 DSP 系统测试。如果 DSP 系统测试结果符合指标，则样机的设计完毕。但由于在软硬件调试阶段调试的环境是模拟的，因此在 DSP 系统测试中往往可能会出现一些问题，如精度不够、稳定性不好等。出现问题时，

一般采用修改软件的方法。如果软件修改无法解决问题,则必须调整硬件,这时问题就较为严重了。

1.2 DSP 应用系统开发工具

对于 DSP 开发工程师来说,除必须了解和熟悉 DSP 本身的结构和技术指标外,大量的时间和精力要花费在熟悉和掌握其开发工具和环境上。此外,通常情况下开发一个嵌入式系统,80%的复杂程度取决于软件。所以,设计人员在为实时系统选择处理器时,都极为看重先进的、易于使用的开发环境与工具。

因此,各 DSP 生产厂商以及许多第三方公司做了极大的努力,为 DSP 系统集成和硬件的开发提供了大量有用的工具,使其成为 DSP 发展过程中最为活跃的领域之一,随着 DSP 技术本身的发展而不断地发展与完善。

1.2.1 软件开发工具

DSP 软件可以使用汇编语言或 C 语言编写源程序,通过编译、连接工具产生 DSP 的执行代码。在调试阶段,可以利用软仿真(Simulator)在计算机上仿真运行;也可以利用硬件调试工具(如 XDS510)将代码下载到 DSP 中,并通过计算机监控、调试运行该程序。当调试完成后,可以将该程序代码固化到 EPROM 中,以便 DSP 目标系统脱离计算机单独运行。

下面简要介绍几种常用的软件开发工具。

(1) 代码生成工具

代码生成工具包括编译器、连接器、优化 C 编译器、转换工具等。可以使用汇编语言或 C 语言(最新版的 CCS 中带的代码生成工具可以支持 C++)编写的源程序代码。编写完成后,使用代码生成工具进行编译、连接,最终形成机器代码。

(2) 软仿真器

软仿真器(Simulator)是一个软件程序,使用主机的处理器和存储器来仿真 TMS320 DSP 的微处理器和微计算机模式,从而进行软件开发和非实时的程序验证。可以在没有目标硬件的情况下作 DSP 软件的开发和调试。在 PC 上,典型的软仿真速度是每秒几百条指令。早期的软仿真器软件与其他开发工具(如代码生成工具)是分离的,使用起来不太方便。现在,软仿真器作为 CCS 的一个标准插件已经被广泛应用于 DSP 的开发中。

(3) 集成开发环境 CCS

CCS(Code Composer Studio)是一个完整的 DSP 集成开发环境,包括了编辑、编译、汇编、连接、软件模拟、调试等几乎所有需要的软件,是目前使用最为广泛的 DSP 开发软件之一。它有两种工作模式:一是软件仿真器,即脱离 DSP 芯片,在 PC 上模拟 DSP 指令集与工作机制,主要用于前期算法和调试;二是硬件开发板结合在线编程,即实时运行在 DSP 芯片上,可以在线编制和调试应用程序。

1.2.2 硬件开发工具

下面简要介绍几种常用的硬件开发工具。

(1) 硬仿真器(Emulator)

硬仿真器(Emulator)由插在 PC 内 PCI 卡或接在 USB 口上的仿真器和目标板组成。

C54x 硬件扫描仿真口通过仿真头（JTAG）将 PC 中的用户程序代码下载到目标板的存储器中，并在目标板内实时运行。

TMS320 扩展开发系统 XDS（eXtended Development System）是功能强大的全速仿真器，用于系统级的集成与调试。扫描式仿真（Scan-Based Emulator）是一种独特的、非插入式的系统仿真与集成调试方法。程序可以从片外或片内的目标存储器实时执行，在任何时钟速度下都不会引入额外的等待状态。

XDS510/XDS510WS 仿真器用户界面友好，是以 PC 或 SUN 工作站为基础的开发系统，对 C2000、C5000、C6000、C8x 系列的各片种实施全速扫描式仿真。因此，可以用来开发软件和硬件，并将它们集成到目标系统中。XDS510 适用于 PC，XDS510WS 适用于 SPARC 工作站。

(2) DSK 系列评估工具及标准评估模块

DSP 入门套件 DSK（DSP Starter Kit）、评估模块 EVM（Evaluation Module）是 TI 或 TI 的第三方为 TMS320 DSP 的使用者设计和生产的一种评估平台，目前可以为 C2000、C3x、C5000、C6000 等系列片种提供该平台。DSK 或 EVM 除了提供一个完整的 DSP 硬件系统外（包括 A/D&D/A、外部程序/数据存储器和外部接口等），还提供有完整的代码生成工具及调试工具。用户可以使用 DSK 或 EVM 来做 DSP 的实验，进行诸如控制系统、语音处理等应用；也可以用来编写和运行实时源代码，并对其进行评估；还可以用来调试用户自己的系统。

在 DSP 应用系统开发过程中，需要开发工具支持的情况如表 1.1 所示。

表 1.1 DSP 应用系统开发工具支持

开发步骤	开发内容	开发工具支持	
		硬件支持	软件支持
1	算法模拟	计算机	C 语言, MATLAB 语言等
2	DSP 软件编程	计算机	编辑器(如 Edit 等)
3	DSP 软件调试	计算机、DSP 仿真器等	DSP 代码生成工具(包括 C 编辑器、汇编器、连接器等)DSP 代码调试工具(软仿真器 Simulator、CCS 等)
4	DSP 硬件设计	计算机	电路设计软件(如 Protel、DXP 等)、其他相关软件(如 EDA 软件等)
5	DSP 硬件调试	计算机、DSP 仿真器、信号发生器、逻辑分析仪等	相关支持软件
6	系统集成	计算机、DSP 仿真器、示波器、信号发生器、逻辑分析仪等	相关支持软件

1.3 实例：基于 TMS320F2812A DSP 的最小系统设计

一个 DSP 硬件系统可以分为最小硬件系统设计和外围接口设计两个部分。一个 DSP 最小硬件系统包括电源、复位电路、时钟电路、总线接口和仿真接口等部分，缺一不可。给出最小系统原理框图如图 1.4 所示。

实例：搭建一个基于 TMS320F2812A DSP 的最小系统。

下面给出一个完整的 TMS320F2812A 芯片的最小系统的原理图，它是一个独立的最小系统。整个设计包括 4 张图，分别为最小系统原理框图（图 1.4）、DSP 外围扩展电路图（图 1.5）、JTAG 仿真口端子图（图 1.6）、DSP 与 JTAG 仿真口连接图（图 1.7）。