TI 公司 DSP 器件系列丛书

TMS320C54x DSP 应用程序设计与开发

刘益成　编著

北京航空航天大学出版社
http://www.buaapress.com.cn

TI 公司 DSP 器件系列丛书

TMS320C54x DSP 应用程序设计与开发

刘益成 编著

北京航空航天大学出版社
http://www.buaapress.com.cn

内容简介

本书详细说明了 TMS320C54x 系列 DSP 应用程序的设计与开发，重点论述了其汇编语言和高级 C/C++语言应用程序的设计和调试方法，对其各种开发工具和最新的集成开发环境 Code Composer Studio(CCS)软件的使用方法进行了详尽的描述，并给出了应用实例。

应用程序的设计与开发是 DSP 应用的核心技术。本书强调先进性与实用性，全书采用 Texas Instruments 公司的最新资料编写而成。各章对各种开发工具的使用方法都列举了大量的程序例子，并专门用一章针对数字信号处理应用，说明了 DSP 应用程序的设计与开发过程，具有很强的实用性。

本书可作为大专院校电子信息、通信、自动控制、仪器仪表类专业本科生和研究生学习 DSP 的教材和参考书，也可供从事 DSP 开发与应用的广大工程技术人员参考。

图书在版编目(CIP)数据

TMS320C54x DSP 应用程序设计与开发/刘益成编著.
--北京：北京航空航天大学出版社，2002.5
ISBN 7-81077-166-3

Ⅰ.T… Ⅱ.刘… Ⅲ.数字信号 信号处理 程序设计 Ⅳ.TN911.72

中国版本图书馆 CIP 数据核字(2002)第 024489 号

TMS320C54x DSP 应用程序设计与开发

刘益成 编著

责任编辑 刘晓明

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话：(010)82317024 传真：(010)82328026

http://www.buaapress.com.cn

E-mail:pressell@publica.bj.cninfo.net

河北省涿州市新华印刷厂印装 各地书店经销

*

开本：787×1092 1/16 印张：26.25 字数：672 千字

2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷 印数：5 000 册

ISBN 7-81077-166-3/TP·092 定价：39.00 元

前　言

数字信号处理器(Digital Signal Processor,简称为DSP)是针对数字信号处理需要而设计的一种可编程的单片机,是现代电子技术、计算机技术和信号处理技术相结合的产物。随着信息处理技术的飞速发展,数字信号处理器在电子信息、通信、软件无线电、自动控制、仪器仪表、信息家电等高科技领域获得了越来越广泛的应用。

数字信号处理器由于运算速度快,具有可编程特性及接口灵活,使得它在许多电子信息产品的研制、开发与应用中,发挥着越来越重要的作用;采用DSP器件来实现数字信号处理系统更是成了当前的发展趋势。与此同时,如何以最短的开发周期,开发出能充分发挥DSP潜能的高质量的应用软件,已经成了广大DSP工程技术人员共同关心的问题。据估计,在DSP应用系统的开发中,特别是对于比较复杂的或对时间要求十分严格的应用系统,绝大部分的开发时间用于软件的设计与调试。DSP软件的调试离不开DSP的开发工具,因此熟悉并掌握DSP的开发工具是开发出高质量DSP软件的必备条件。正是出于这种目的,本书针对美国德洲仪器(简称TI)公司推出的新一代16位定点TMS320C54x DSP进行了介绍,该系列DSP是目前使用最为广泛的数字信号处理器之一。

本书从TMS320C54x DSP软件的开发过程出发,详细地说明了基于汇编语言和高级C/C++语言的程序设计方法及各种开发工具的使用。实际上,由于信号处理系统是硬件和软件相结合的产物,特别是像信号处理器这样的单片机系统,硬件和软件是分不开的;但限于篇幅,本书只讨论了基于该系列芯片的信号处理软件的设计、开发与调试。为了对软件设计与开发有良好的理解,读者应对TMS320C54x的硬件结构以及在片的各种资源有较好的了解,熟悉TMS320C54x的指令系统。同时,还要求读者会使用标准C/C++语言进行程序设计。

信号处理是一个涉及面很广的学科,包括信号处理的理论与应用

系统的实现两个方面。本书不涉及信号处理的理论问题，并在有关的程序例子中假定读者对所涉及的信号处理的理论有基本的了解。有关这方面已有很多文献和书籍。

全书正文共10章，第1章对DSP系统的开发方法、TMS320C54x芯片的性能和TMS320C54x的软件开发过程进行了简要介绍。其余各章分为4大部分：第1部分为汇编语言程序设计及有关的开发工具，包括第2～6章，介绍了汇编语言程序设计的基础和汇编程序的开发工具及其使用方法，这部分是高级语言程序设计的基础；第2部分包括第7章和第8章，介绍TMS320C54x的高级C/C++语言程序设计及其开发工具TMS320C54的C/C++编译器、集成开发环境CCS的使用；第3部分为第9章，举例具体说明应用程序的开发过程；第4部分为第10章，介绍如何将调试好的目标程序烧制进EPROM。为了便于查阅，在附录中列出了TMS320C54x的汇编助记符指令集、汇编伪指令、TMS320C54 C/C++编译器选项以及EPROM编程器的目标格式。

本书第7章由刘建国编写，第9章由朱正平编写，附录1由吴莉莉编写，其余部分由刘益成教授编写，并对全书进行了审校。杜红副教授审阅了部分书稿，提出了不少宝贵意见，在此表示衷心的感谢。

由于作者的水平所限，书中的缺点和错误恳请广大读者批评指正。

编　者

2002年2月

《单片机与嵌入式系统应用》月刊

- 北京航空航天大学出版社承办
- 何立民教授主编
- 中央级科技期刊
- 单片机与嵌入式系统专业期刊
- 特色：专业期刊、专家办刊、着眼世界、面向国内、应用为主、读者第一
- 已开辟有 8 个栏目：业界论坛、专题论述、技术纵横、新器件新技术、应用天地、经验交流、学习园地、编读往来
- 杂志社网址：http://www.microcontroller.com.cn，http://www.dpj.com.cn

诚挚欢迎业界人士向本刊投稿，欢迎广大读者订阅本刊。

联系方式

通信或汇款地址：北京市海淀区学院路 37 号《单片机与嵌入式系统应用》杂志社
邮编：100083　　电话：(010)82317029,82313656　　传真：010-82317043
E-mail：mcu@publica.bj.cninfo.net(投稿亦请用此 E-mail 地址)
广告业务联系人：王金萍　魏洪亮　　开户行：北京市商业银行学院路支行
户名：《单片机与嵌入式系统应用》杂志社　　账号：010903391001201110299-36

订阅方式

国内统一刊号：CN 11-4530/V　　国际标准刊号：ISSN 1009-623X
每期定价：8 元，全年定价：96 元，每月 1 日出版。
第一种订阅方式：通过邮局订阅，邮发代号：2-765。
第二种订阅方式：直接向《单片机与嵌入式系统应用》杂志社订阅，另加 15%邮资。

2002 年《单片机与嵌入式系统应用》月刊订阅单
(复印有效)

姓名/单位				收件人	
通信地址				邮　编	
单　价	8.00 元	起订月份		订阅期数	
订阅份数		金额大写			

2002 年《单片机与嵌入式系统应用》月刊订阅单
(复印有效)

姓名/单位				收件人	
通信地址				邮　编	
单　价	8.00 元	起订月份		订阅期数	
订阅份数		金额大写			

目　录

第 1 章　TMS320C54x 应用程序开发过程

1.1　DSP 应用系统开发方法 …… (1)
1.2　定点 DSP 的数据格式 …… (5)
1.3　TMS320C54x 系列数字信号处理器简介 …… (8)
1.4　TMS320C54x 应用软件开发流图与开发工具 …… (11)

第 2 章　公共目标文件格式简介

2.1　COFF 文件的基本单元——段 …… (14)
2.2　汇编器对段的处理 …… (15)
2.3　链接器对段的处理 …… (19)
2.4　重新定位 …… (21)
2.5　程序装入 …… (22)
2.6　COFF 文件中的符号 …… (23)

第 3 章　TMS320C54x 汇编语言程序设计

3.1　汇编语言源程序格式 …… (24)
3.2　汇编器及其调用 …… (27)
3.3　汇编语言中的常数与字符串 …… (30)
3.4　汇编源程序中的符号 …… (33)
3.5　汇编源程序中的表达式 …… (37)
3.6　汇编器的内部函数 …… (40)
3.7　汇编器伪指令 …… (41)
3.8　源清单文件 …… (52)
3.9　交叉引用清单文件 …… (56)

第 4 章　宏语言

4.1　宏的使用 …… (59)
4.2　定义宏 …… (59)
4.3　宏参数和替代符号 …… (61)
4.4　宏　库 …… (66)
4.5　在宏中使用条件汇编 …… (67)
4.6　在宏中使用标号 …… (69)

4.7 在宏中产生的信息 …… (70)
4.8 格式化输出清单文件 …… (71)
4.9 使用递归和嵌套宏 …… (72)
4.10 宏伪指令小结 …… (73)

第 5 章 链接器及其使用

5.1 链接器的调用 …… (75)
5.2 链接器的选项 …… (76)
5.3 链接器命令文件的编写与使用 …… (84)
5.4 目标库 …… (86)
5.5 MEMORY 伪指令及其使用 …… (87)
5.6 SECTIONS 伪指令及其使用 …… (90)
5.7 指定段运行时间的地址 …… (98)
5.8 默认的定位算法 …… (100)
5.9 使用 UNION 和 GROUP 说明语句 …… (101)
5.10 覆盖页面 …… (105)
5.11 特殊段类型 …… (108)
5.12 在链接时间给符号赋值 …… (109)
5.13 产生和填充空洞 …… (111)
5.14 部分链接或增加性链接 …… (114)
5.15 链接 C/C++代码 …… (115)
5.16 使用链接器的例子 …… (116)
5.17 TMS320C54x 汇编程序设计的一些考虑 …… (119)

第 6 章 其他汇编程序开发工具

6.1 存档器及其使用 …… (123)
6.2 绝对列表器及其使用 …… (125)
6.3 交叉引用列表器及其使用 …… (130)
6.4 助记符指令到代数指令转换器 …… (132)

第 7 章 TMS320C54x 高级 C 语言程序设计

7.1 TMS320C54x C/C++ 语言 …… (137)
7.2 TMS320C54x C/C++ 编译器综述 …… (149)
7.3 TMS320C54x C/C++ 编译器的使用 …… (153)
7.4 C/C++代码优化 …… (171)
7.5 链接 C/C++代码 …… (182)
7.6 运行时间环境 …… (186)
7.7 汇编语言与 C/C++混合编程 …… (194)

7.8 数值计算 …… (202)
7.9 系统初始化 …… (204)
7.10 运行时间支持函数 …… (207)
7.11 建库工具 …… (216)
7.12 C/C++语言编程的注意事项 …… (218)

第 8 章 集成开发环境 CCS 及其使用

8.1 C5000 Code Composer Studio 简介 …… (230)
8.2 CCS 编辑器 …… (235)
8.3 CCS 的基本操作 …… (244)
8.4 CCS 工程管理工具 …… (256)
8.5 CCS 的调试工具 …… (264)
8.6 CCS 优化工具——评价器 …… (284)
8.7 存储器映射 …… (289)
8.8 通用扩展语言 …… (291)
8.9 DSP/BIOS 实时内核插件 …… (301)
8.10 实时数据交换插件 …… (306)

第 9 章 应用程序开发实例

9.1 硬件系统简介 …… (312)
9.2 系统初始化 …… (313)
9.3 数字计算程序 …… (320)
9.4 FIR 数字滤波程序 …… (327)
9.5 TMS320C54x 的 FFT 程序实现 …… (338)
9.6 DSP 独立系统的实现 …… (348)

第 10 章 Hex 转换工具

10.1 调用 Hex 转换工具 …… (352)
10.2 编写命令文件 …… (354)
10.3 存储器宽度与字的宽度 …… (355)
10.4 ROMS 伪指令 …… (359)
10.5 SECTIONS 伪指令 …… (362)
10.6 输出文件名 …… (363)
10.7 映像模式和－fill 选项 …… (365)
10.8 由在片引导装入器构造引导表 …… (366)
10.9 控制 ROM 器件的地址 …… (368)
10.10 使用 Hex 转换工具的例子 …… (371)
10.11 Hex 码转换工具的输出信息 …… (380)

附 录

附录 1 TMS320C54x 助记符汇编指令集 …… (381)
附录 2 汇编伪指令分类列表 …… (396)
附录 3 TMS320C54x C/C++编译器选项分类列表 …… (400)
附录 4 EPROM 编程器目标格式说明 …… (405)

参考文献

第1章　TMS320C54x应用程序开发过程

1.1　DSP应用系统开发方法

1. 数字信号处理的特点

信号处理是信息科学的核心技术之一，数字信号处理则是利用计算机或专用的数字设备对信号进行分析、合成、变换、估计、辨识等加工处理，以便提取有用的信息并进行有效的传输与应用。如何有效地实现数字信号处理一直是信号处理工作者研究的目标。数字信号处理包括两方面的内容。其一是算法研究，即研究如何以最小的运算量和存储器使用量完成指定的任务。一个最著名的例子是20世纪60年代出现的离散富氏变换(DFT)的快速算法——快速富氏变换(FFT)。它使得数字信号处理技术发生了革命性的变化。其二是数字信号处理的系统实现，除了有关的输入/输出部分外，其中最核心的部分就是算法的实现，即用硬件、软件或软硬件相结合的方法来实现各种算法，例如FFT算法的实现。早期的数字信号处理算法的实现一般有以下2种方法：

① 利用通用的计算机或微机通过软件的方法实现；

② 利用专用的数字设备实现。

通过软件和通用的计算机或微机来实现，是早期的主要办法。在20世纪70～80年代，国内外相继出现了多个数字信号处理软件包。但由于数字信号处理算法通常要求完成大量的数值运算，特别是乘加运算，采用通用计算机速度较慢，所以这种方法多用于教学和研究，或后处理，不适合于须进行大量数据运算的现场处理，特别是进行实时现场处理。专用的数字设备主要问题是结构复杂，体积大，成本高，耗电量大，可靠性差，而且也不灵活。例如早期利用位片式微处理器加上快速乘法器来实现的系统就存在这样的问题，这种系统已逐渐被淘汰。

仔细分析数字信号处理的各种算法，都是以乘加为基础的。数字信号处理中最常用的算法如下。

卷积运算：设 $x(n)$，$h(n)$为输入数字序列和滤波器脉冲响应序列，长度分别为 M 和 N；$y(n)$为卷积输出序列，则

$$y(n)=\sum_{i=0}^{N}x(n-i)h(i)\qquad n=0,\cdots,M+N-2 \tag{1-1}$$

FFT(DFT)运算：设 $x(n)$为长度为 N 的输入数据序列，$X(k)$为 $x(n)$的DFT，则

$$X(k)=\sum_{n=0}^{N}x(n)W_N^{kn}\qquad k=0,\cdots,N-1 \tag{1-2}$$

矩阵乘法：设 $a(i,j)$，$b(i,j)$分别为 $A(M\times N$ 阶)、$B(N\times K$ 阶)矩阵的元素，$c(i,j)$为 $C(M\times K$ 阶)乘积矩阵的元素，则

$$c(i,j)=\sum_{s=0}^{N}a(i,s)b(s,j)\qquad i=1,\cdots,M;\quad j=1,\cdots,K \tag{1-3}$$

可以发现，它们都可化为 $A_k=\sum_i B_iC_i$ 形式，即乘积和形式，而且适合于进行并行计算。除此

以外，有些算法具有自己的特殊要求。例如 FFT，按照所使用的算法，要求对输入或输出的数据进行倒码。统计表明，采用高级语言，例如，Fortran 或 C 语言，这种倒码运算占所有 FFT 运算量的 15%～25%，十分费时。而实时的滤波过程(即卷积运算)则要求不断地修改输入缓冲区的数据，用新数据替代老数据。这将大大地增加数据编排的时间。随着电子技术、大规模集成电路和计算机技术的发展，一种新型的专用于数字信号处理的单片机——数字信号处理器就应运而生，极大地推动了信号处理学科的发展。

随着信息技术的数字化和高速发展，DSP 的应用越来越广泛。目前，DSP 的年增长率已达 30%以上，是集成电路平均增长速度的 2 倍。由于它具有很高的处理速度和可编程特性，很多行业纷纷转而采用 DSP 芯片来制造具有更高性能，体积、耗电更小的便携式产品，例如移动通信产品、网关、软件无线电等。近年来，由于半导体技术的进步，DSP 芯片价格也大大降低，现在 TI 公司的第 1～5 代产品，每片均低于 10 美元，与普通的单片机相当，这就更进一步推进了 DSP 应用的发展。

2. 数字信号处理系统的设计过程

一个数字信号处理系统是电子技术、信号处理技术与计算机技术相结合的产物。系统设计通常分为信号处理部分和非信号处理部分。信号处理部分包括系统的输入和输出、数据的编排和处理、各种算法的实现、数据显示和传输等；非信号处理部分则包括电源、结构、成本、体积、可靠性与可维护性等。一个应用系统的设计过程大致可分为以下 7 个部分：

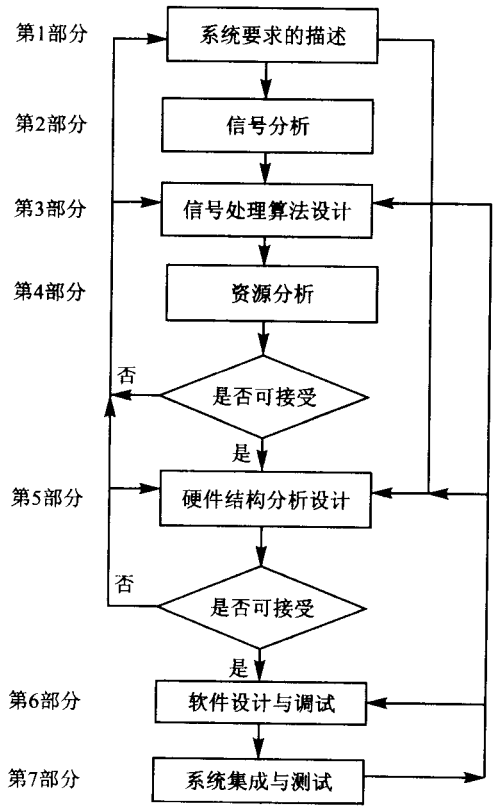


图 1-1 信号处理系统设计过程

① 系统要求的描述；
② 信号分析；
③ 信号处理算法的设计；
④ 资源分析；
⑤ 硬件结构分析与设计；
⑥ 软件设计与调试；
⑦ 系统集成与测试。

这 7 个部分的相互关系如图 1-1 所示，下面予以简单的说明。

(1) 系统要求的描述

在这一部分，根据用户对应用系统的要求，提出一组系统级的技术要求和相关说明。这些要求包括人机接口、信号类型与特征、处理的项目和方式、处理系统的所有性能指标要求(含系统非信号处理的性能)以及系统的测试和验证方式等。由此形成相应的文档，作为以后设计的依据。

(2) 信号分析

信号分析部分定义输入/输出信号的类型，例如，分析随机或确定信号，模拟或数字信号是一维信号还是多维信号等，确定描述输入信号的模型。分析信号的频率范围和系

统的带宽，估计信号的最大和最小电平以及信号噪声比(SNR)，是否需要进行预处理等。确定输出信号使用的方式、数据的吞吐率和对实时性的要求。信号分析的结果是第 3 部分信号处理算法设计的基础。

(3) 信号处理算法设计

本部分是信号处理系统的核心，其任务是根据第 1 部分提出的要求和第 2 部分对信号分析的结果，对不同类型的信号和所要求的处理方式确定相应的算法。这部分要求设计者对信号处理的各种算法有较好的了解。首先是要保证算法的正确性，必要时，可利用现成的软件包，例如用 MATLAB 等工具进行仿真计算。其次是算法的有效性，当硬件给定后，算法的有效性决定了系统的处理能力或吞吐能力。算法设计的主要目标是对于一个特定的任务，获得运算量最小和使用资源最少的算法。有时，这二者是矛盾的，这就需要找出最好的折中方案。下面的例子说明了算法设计的重要性。

例 1-1　在一个多抽样率信号处理系统中，希望将数字信号输入的采样率从 $f_0=256$ kHz降低到 $f=1\ 000$ Hz，抽取比 $M=f_0/f=256$。为了保证信号经再抽样后不产生混叠失真，在再抽样前需对信号进行数字滤波处理。同时，为了保证滤波后的信号不产生相位失真，滤波器一般采用 FIR 滤波器。直接进行滤波处理的公式为

$$y(n)=\sum_{i=0}^{N-1}x(nR-i)h(i) \tag{1-4}$$

其中：$x(n)$为输入的高采样率的信号，$h(n)$为 FIR 滤波器系数，N 为滤波器的节数；$y(n)$为滤波重采样后的输出序列。完成上述运算的框图如图 1-2 所示。

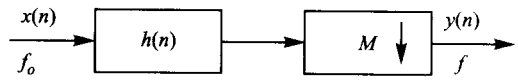


图 1-2　直接滤波抽取

式(1-4)的实现看起来十分简单，但若直接用该式来完成所述的抽取过程几乎是不可能的。一般需采用多级抽取的方法，如图 1-3 所示。

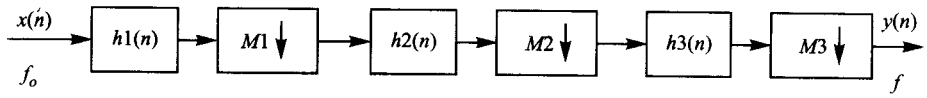


图 1-3　多级抽取

图 1-3 中 $M=M1\times M2\times M3$，$h1(n)$，$h2(n)$，$h3(n)$分别为分 3 级抽取的第 1 级，第 2 级和第 3 级滤波器的系数。表 1-1 列出了单级抽取与 3 级抽取的滤波器参数及所需的运算速率。设计中第 1 级抽取滤波器为梳状滤波器，其他各滤波器的通带和阻带波纹都假定为小于 120 dB，采用 kaiser 窗函数法设计。有关多级抽取滤波器设计的详细内容见参考文献 15。

由表 1-1 可见，采用单级抽取要求的滤波器节数已经大到了不能实现的地步，而且运算量要比 3 级抽取的总运算量大许多倍。多级滤波抽取运算的公式虽然仍然为式(1-4)，但因为多级抽取中滤波器的节数大大降低，因此运算量也就大大减少了。

当然，采用多级抽取的实现方式，系统硬件结构与单级实现也大不相同。此例说明，算法的选择对于信号处理系统来说是致关重要的。它对系统的软件和硬件的设计、系统的性能都起着决定性的作用。

(4) 资源分析

系统资源分成 3 大类：数据吞吐率、存储器容量和输入/输出带宽。这 3 大资源主要决定

于系统所使用的核心处理器或有关硬件，例如 DSP 器件的运算速率、串并口数据传送的能力以及存储器空间的大小。一个信号处理系统通常可分成如图 1-4 所示的 4 个大的功能块：系统控制、信号处理、数据存储、数据通信。资源分析的目的就是将前述的 3 大资源在功能块之间进行合理地分配，以取得最佳的性能。

表 1-1 单级抽取与 3 级抽取的滤波器参数和运算速率

滤波器参数	单级抽取	3 级抽取		
		第 1 级	第 2 级	第 3 级
输入采样率/kHz	256	256	32	4
输出采样率/kHz	1	32	4	1
抽取比	256	8	8	4
通带上边频/kHz	0.4	梳状滤波器	0.5	0.4
阻带下边频/kHz	0.5	梳状滤波器	3.5	0.5
滤波器节数	19 977	29	85	313
乘加次数/样点	39 963	57	169	625
要求最小运算速率	39 953 000	1 824 000	676 000	625 000
(运算次数/s)		3 级总计运算速率：3 125 000		

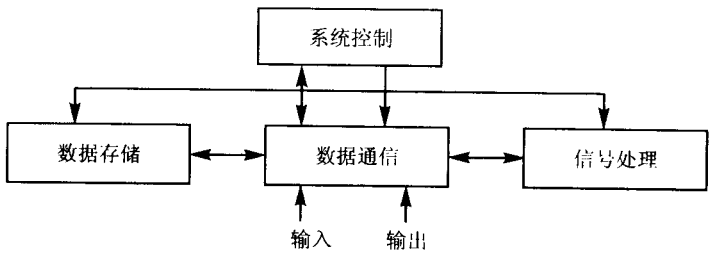


图 1-4 信号处理系统的功能块

(5) 硬件结构分析与设计

本部分根据第 3 和第 4 部分的结果进行硬件的结构分析与设计。这包括 DSP 芯片的选择、存储器配置和输入/输出通道的设计、控制和显示电路设计、电源电路以及其他相关硬件电路的设计。

以 DSP 器件为核心的信号处理系统，大致可分成 2 种。一种是主从式系统，这时 DSP 的功能为主机的一种加速处理器，数据由主机传给 DSP 器件，DSP 器件完成数据处理后再将数据返回到主机，所有工作都在主机的控制下进行。另一种是 DSP 器件自成体系的系统，这时所有的控制、处理工作都由 DSP 器件完成。这样的 DSP 系统一般都要带有自举引导功能。

对于最终产品而言，系统的主要成本由硬件决定；软件成本主要是开发成本。因此硬件设计要求设计者除熟悉 DSP 器件外，还必须对其他各种芯片，例如 FPGA 等有很好的了解。在进行电路设计的同时，还必须考虑系统的结构包括机械结构设计，使之坚固耐用并便于维护。硬件设计是软件设计的基础；而反过来，使用的算法和软件也决定了硬件的结构，因此这部分的设计必须与算法分析和软件设计结合进行。

(6) 软件设计与调试

基于第 3 部分所设计的算法和第 5 部分的硬件设计，本部分的任务是完成系统的所有软件设计与调试。软件设计包括系统软件和信号处理软件。系统软件包括人机接口界面、系统的控制软件、输入/输出管理、显示以及如何与主机的操作系统(包括嵌入式操作系统)接口等。信号处理软件主要是用程序语言来实现在第 3 部分确定的信号处理算法，以完成特定的处理功能。

对主从式系统，系统软件大都采用高级语言编写，例如 VB，VC 等。信号处理软件则既可采用高级语言，也可采用汇编语言；但对于一些关键的核心代码最好采用汇编语言编写，这样可获得最佳的性能。对 DSP 自成体系的系统，则要根据具体硬件来定，对于一些小的单一功能的系统，大都采用汇编语言完成。

软件设计的重要工作之一是开发工具和开发环境的选择。目前，大多数 DSP 器件都提供了完善的开发工具和开发环境，例如本书要介绍的 TMS320C54x 的各种开发工具。基于 DSP 器件的信号处理程序的调试只有在相应的开发工具支持下才能完成。

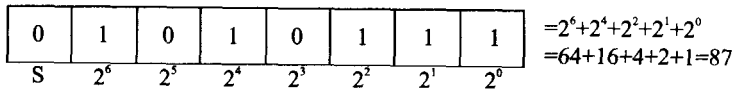
(7) 系统集成与测试

当所有的硬件和软件设计完成后，最后是将系统的各个部分集成为一个整体，进行实际的运行测试。由于信号处理系统是一个相当复杂的系统，牵涉到电子电路、硬件、软件各个方面，设计出来的应用系统是否能满足最初提出的要求，必须通过实际的测试；如果不能满足要求，必须再对硬件和软件进行相应的修改。

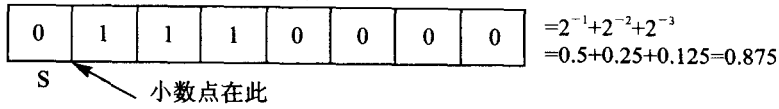
1.2　定点 DSP 的数据格式

在用 DSP 进行数字信号处理时，首先遇到的问题是数的表示方法。DSP 分为 2 种：一种是定点 DSP，另一种是浮点 DSP。在浮点 DSP 中，数据既可以表示成整数，也可以表示成浮点数。浮点数在运算中，表示的数的范围由于其指数可自动调节，因此可避免数的规格化和溢出等问题。但浮点 DSP 一般比定点 DSP 复杂，成本也较高。本书要介绍的 TMS320C54x 是定点 DSP。在定点处理器中，数据采用定点表示方法。它有 2 种基本的表示方法：整数表示方法和小数表示方法。整数表示方法主要用于控制操作、地址计算和其他非信号处理的应用，而小数表示方法则主要用于数字和各种信号处理算法的计算中。很清楚，定点表示并不意味着就一定是整数表示。下面是对于一个 8 位字长的数据用定点数表示的 2 种基本方法。对带符号的整数，其中最高位 S 为符号位，0 代表正数，1 代表负数。除符号位外，其他的各位采用 2 补码表示，表示数的范围为 $-2^n \sim 2^n-1$，这里 n 为数的字长，单位为位。

整数表示



小数表示



1. 定点数的 Q 表示法

定点数最常用的是 Q 表示法，或 Q*m*.*n* 表示法。它可将整数和小数表示方法统一起来。其中，*m* 表示数的 2 补码的整数部分；*n* 表示数的 2 补码的小数部分；1 位符号位；数的总字长为 $m+n+1$ 位。表示数的整数范围为$(-2^m, 2^m)$，小数的最小分辨率为 2^{-n}。下面举例说明几种常用的 Q 表示法的格式。

(1) Q15.0 格式

Q15.0 格式的字长为 16 位，其每位的具体表示如下：

位　数	15	14	13	12	11	10	9	…	0
值	S	I14	I13	I12	I11	I10	I9	…	I0

最高位为符号位 S，接下来的 Ix 为 15 位 2 补码的整数，高位在前，无小数位。这实际就是定点数的整数形式。Q15.0 格式表示数的范围为$(-2^{15}, 2^{15}-1)$，最小的分辨率为 1。

(2) Q3.12 格式

Q3.12 格式的字长为 16 位，其每位的具体表示如下：

位　数	15	14	13	12	11	10	9	…	0
值	S	I3	I2	I1	Q11	Q10	Q9	…	Q0

最高位为符号位 S，接下来的 3 位为 2 补码的整数位，高位在前，后面的 12 位为 2 补码小数位。Q3.12 格式表示数的大致范围为(−8,8)，小数的最小分辨率为 2^{-12}。

(3) Q0.15 (或 Q.15)格式

Q.15 格式的字长为 16 位，其每位的具体表示如下：

位　数	15	14	13	12	11	10	9	…	0
值	S	Q14	Q13	Q12	Q11	Q10	Q9	…	Q0

最高位为符号位 S，接下来的为 2 补码的 15 位小数位，小数点紧接着符号位，无整数位。Q.15 格式表示数的范围为(−1,1)，小数的最小分辨率为 2^{-15}。对于 16 位的定点处理器 TMS320C54x 来说，Q15 是在程序设计中最常用的格式。例如，TI 公司提供的数字信号处理应用程序库 DSPLIB 就主要采用这种数据格式。

(4) Q0.31(或 Q.31)格式

Q.31 格式的字长为 32 位，需要 2 个 16 位的存储器字来表示。它实际上是 Q.15 格式的扩展表示。其低 16 位的具体表示如下：

位　数	15	14	13	12	11	10	9	…	0
值	Q15	Q14	Q13	Q12	Q11	Q10	Q9	…	Q0

高 16 位表示为：

位　数	31	30	29	28	27	26	25	…	16
值	S	Q30	Q29	Q28	Q27	Q26	Q25	…	Q16

高 16 位的最高位为符号位 S，接下来的为 2 补码的 31 位小数位，小数点紧接着符号位，无整数位。Q.31 格式表示数的范围为(－1,1)，小数的最小分辨率为 2^{-31}。

2. 定点数格式的选择

由前面对几种 Q 格式的介绍可见，定点格式表示数的范围和数据的精度是确定的。表示数的范围越大，数据精度就越低，也就是说，数的范围与精度是一对矛盾。对 16 位的数据来说，动态范围最大的格式为整数 Q15.0，精度(或分辨率)最高的格式为 Q.15。若动态范围或精度超过了 16 位的范围，则只有增加字长，例如，采用 32 位(2 个 16 位字)来表示。因此，数据格式的选择实际上就是如何根据具体应用问题，在它们之间寻求最好的折中。一旦格式选定后，就必须保证在整个运算过程中，数据不会溢出。例如，对 Q.15 格式，数据的范围在(－1,1)之间，这样就必须保证在所有运算中其结果都不能超过这个范围，否则，芯片将结果取其极大值－1 或 1，而不管其真实结果为多少。为了保证不会出现溢出，在数据参加运算前，程序员应估计数据及其结果的动态范围，选择合适的格式对数据进行规格化。例如，假设有 100 个 0.5 相加，采用 Q.15 格式进行运算，其结果将等于 1。为了保证结果正确，可先将 0.5 规格化为 0.005 后进行运算，然后将所得结果反规格化；或者采用 Q6.9 格式进行计算。因此，从另一个角度说，定点格式的选择实际上就是要根据 Q*m*.*n* 表示方法确定数据的小数点位置。

3. 定点格式数据的转换

同一个用二进制表示的定点数，当采用不同的 Q*m*.*n* 表示方法时，其代表的十进制数是不同的。例如，由 5000H 表示的 16 位字长的数，当采用 Q.15 格式时，其值等于 0.625；当采用 Q3.12 格式时，其值等于 5；当采用 Q15.0 格式时，其值等于 20 480。当 2 个不同 Q 格式的数进行加减运算时，通常必须将动态范围较小的格式的数转换为动态范围较大的格式的数。有 2 种转换的方法，下面予以说明。

① 若十进制数没有表示成任何形式的二进制数，则要表示成 Q*m*.*n* 格式。此时，先将数乘以 2^n 变成整数，然后再将整数转换成相应的 Q*m*.*n* 格式。例如设 $y=-0.625$，若要表示成 Q.15 格式，先将－0.625 乘以 2^{15} 得到－20 480，再将－20 480 表示成 2 的补码数为 B000H，这也就是－0.625 的 Q.15 格式表示；若要将－0.625 表示成 Q3.12 格式，则将－0.625 乘以 2^{12} 得到－2 560，表示成 2 的补码数为 F600H，这也就是－0.625 的 Q3.12 格式表示。

② 若数已经是某种动态范围较小的 Q*m*.*n* 格式，为了与动态范围较大的 Q*m*.*n* 格式数进行运算，则可根据运算结果的动态范围，直接将数据右移，将数据转换成结果所需的 Q*m*.*n* 格式，这时原来格式的最低位将被移出，高位则进行符号位扩展。这在某些情况下会损失动态范围较小的格式的数据的精度。例如，若 5.625＋0.625＝6.25，5 和结果 6.25 需要采用 Q3.12 格式才能保证其动态范围，若 0.625 原来用 Q.15 格式表示，则需先要将它表示成 Q3.12 后，再进行运算，自然，最后的结果也为 Q3.12。下面分几种情况具体说明带符号数据的转换过程。

情况 1　若 x，y 为正数，$x=5.625$，则采用 Q3.12 格式表示的二进制码为 5A00H；$y=0.625$；采用 Q.15 格式表示的二进制码为 5000H，求 $x+y$。

由于 Q3.12 格式与 Q.15 格式的整数位相差 3 位，所以将 y 的 Q.15 格式表示的二进制码 5000H 右移 3 位；由于 5000H 为正数，将整数部分补零，得到用 Q3.12 格式表示的 0.625 为 0A00H。将 5A00H 加上 0A00H 得到 6400H，该数的格式为 Q3.12，其值等于 $x+y=6.25$。

情况 2　若 x 为正数，y 为负数，$y=-0.625$，则采用 Q.15 格式表示的二进制码为