

TI DSP 系列中文手册

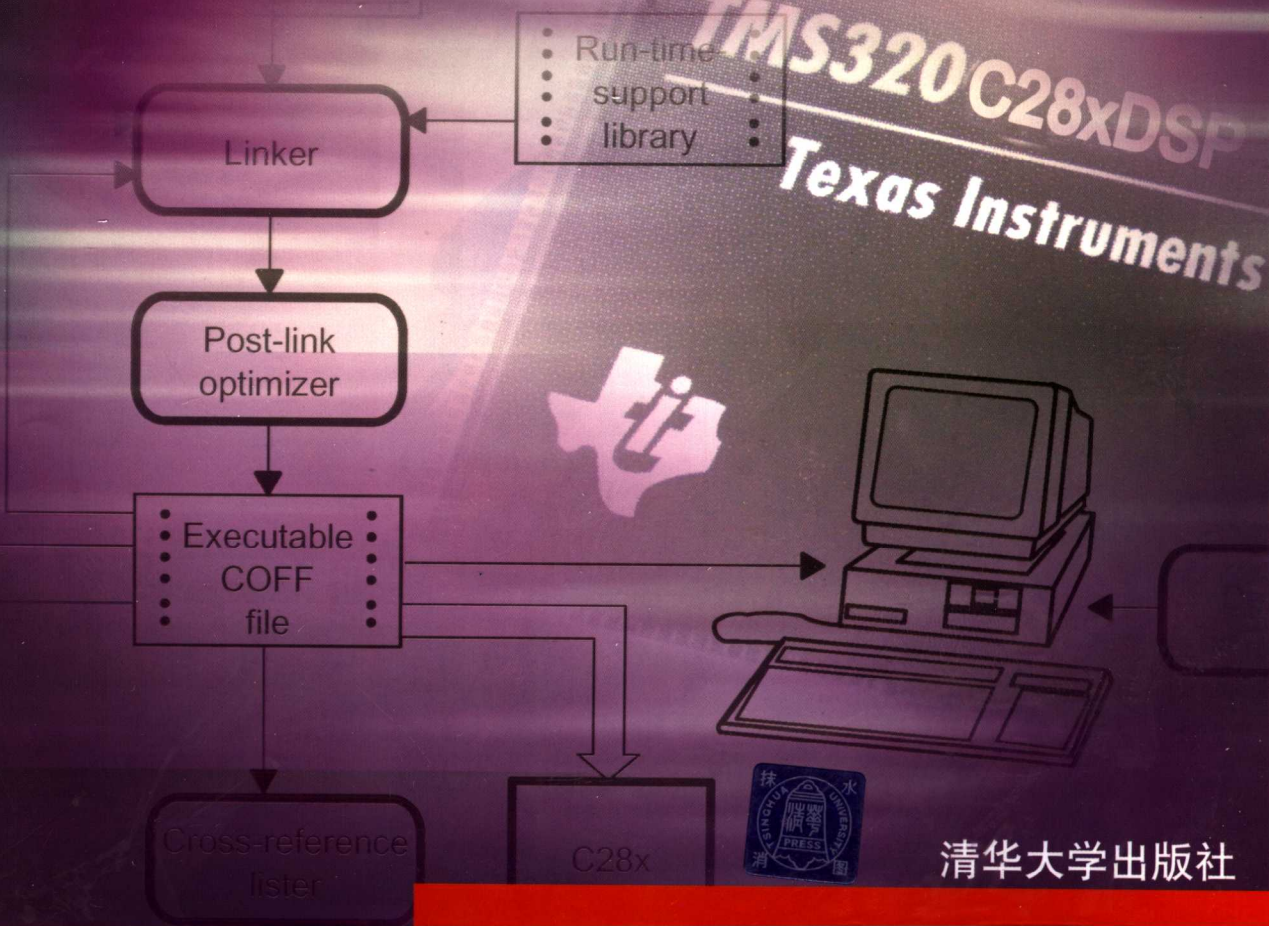
TMS320C28x 系列 DSP

指令和编程指南

[美] Texas Instruments Incorporated 著

刘和平 张卫宁 刘 林 编译
周有为 邓 力 江 渝

Run-time
support
library



清华大学出版社

TMS320C28x 系列 DSP

指令和编程指南

[美] Texas Instruments Incorporated

著

刘和平 张卫宁 刘 林

编译

周有为 邓 力 江 渝

清华大学出版社

北 京

内 容 简 介

本书由 TI 公司的两个文献编译而成, 编号为 SPRU513 的文献介绍了如何使用汇编语言工具: 汇编器、归档器、目标代码链接器、交叉引用列表程序、绝对地址列表程序、十六进制转换应用程序。编号为 SPRU430B 的文献中的一部分介绍了 C28x 汇编语言指令集。由于这两部分内容紧密相关, 故将其放在一起, 以便读者查阅。

本书主要针对从事 TI 公司 2000 系列 DSP 开发应用的工程技术人员, 也可以作为在校研究生的参考用书。

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目 (CIP) 数据

TMS 320 C28X 系列 DSP 指令和编程指南/TI 公司著; 刘和平等编译. —北京: 清华大学出版社, 2005. 3
(TI DSP 系列中文手册)

ISBN 7-302-10438-7

I. T… II. ①T… ②刘… III. 数字信号—信号处理—数字通信系统, TMS 320 C28X DSP—程序设计—指南 IV. TN911. 72-62

中国版本图书馆 CIP 数据核字 (2005) 第 009219 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

责任编辑: 曾 刚

封面设计: 刘春敏

版式设计: 郑轶文

印 刷 者: 北京密云胶印厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 31.5 字数: 693 千字

版 次: 2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷

书 号: ISBN 7-302-10438-7/TP·7090

印 数: 1~5000

定 价: 46.00 元

TI DSP 系列中文手册编译委员会

(按汉语拼音排序)

主任委员:

胡广书	教授	清华大学
彭启琮	教授	电子科技大学
沈洁	经理	TI 中国大学计划

委员:

陈健	教授	上海交通大学
戴逸民	教授	中国科学技术大学
何佩琨	教授	北京理工大学
刘和平	教授	重庆大学
潘亚涛	工程师	TI 中国大学计划
桑恩方	教授	哈尔滨工程大学
王军宁	副教授	西安电子科技大学
张旭东	副教授	清华大学
曾刚	编辑	清华大学出版社

序

经过全体编译老师和编译委员会近一年的努力,“TI DSP 系列中文手册”终于陆续和广大读者见面了。

数字信号处理器(Digital Signal Processing, DSP)是对信号和图像实现实时处理的一类高性能的 CPU。所谓“实时(Real-Time)实现”,是指一个实际的系统能在人们听觉、视觉或按任务要求所允许的时间范围内实现对输入信号的处理并将其输出。目前, DSP 已广泛应用于通信、家电、航空航天、工业测量、控制、生物医学工程及军事等许许多多需要实时实现的领域。

美国德州仪器(Texas Instruments, TI)公司是全球 DSP 研发和生产的领先者。自 1982 年推出第一块 DSP 芯片以来,到 20 世纪 90 年代中期, TI 先后推出了 C10、C20、C30、C40、C50 及 C80 等 6 代 TMS320 系列的 DSP 产品。紧接着又推出了 C2000 系列、C5000 系列和 C6000 系列三大主流产品,并推出了将 DSP 和 ARM 合为一体的 OMAP 系列。这些产品无论在国外还是在国内都得到了广泛的应用。例如,“TI 中国大学计划”在 2003 年举办的“TI DSP 设计比赛”中,国内高校就有约 90 个队参加,足见 DSP 在我国已经得到普遍的重视。

凡是从事过含有 CPU 的系统设计(单片机或 DSP)的设计人员都知道,为了顺利地实现设计任务,一本或几本好的手册是必不可少的,其中包括该 CPU 的结构手册、指令和汇编语言手册以及开发手册等。

由于 TI 的 DSP 发展迅速,产品更新快,因此其手册自然也非常多。由于手册需要更新和补充,因此,彼此之间难免会出现重复和种类繁多的现象。使用过 TI DSP 文档的用户都感觉到,其手册在使用上是有相当难度的。另外, TI DSP 文档都是用英文写成,这也给部分工程技术人员带来一定的困难。

鉴于此, TI 中国主管提出委托国内的高校老师对其文档进行编译,并授权清华大学出版社正式出版。在“TI 中国大学计划”的建议下,2003 年 6 月通过推荐和报名方式成立了编译委员会。

通过认真讨论,编译委员会首先确定了文档编译的原则,然后确定了编译的书目,最后确定了每一本书的编译者。

关于编译的原则,我们提出了如下两点:

(1) 本文档的定位为“手册”。也就是说,每一位文档的编译者应全面了解和掌握所编译书目的所有英文文档,并了解各个文档之间的关系,在保证文档完整的基础上,选择最新的文档,并去除其中的重复内容和已经淘汰的内容。

(2) 要尽可能地按照 TI 英文文档的“本意”来形成中文,以保证手册的准确性。允

许作者按自己的经验有所发挥，以便于难点的理解。

这次编译的书目包含三大部分：一是各个系列的共用部分，如 CCS、DSP/BIOS、算法标准、C 语言编译器及开发工具等各个手册；二是按 C2000、C5000 和 C6000 三大系列分别编译它们的 CPU 结构及指令手册；三是分别编译它们的应用。

编译计划在“TI 中国大学计划”的相关会议上提出后，得到了国内高校许多老师的热情支持，很快便将要编译的书目一一落实。这些老师都有着从事 DSP 教学和科研的丰富经验，正是由于他们的大力支持，才使这一庞大的工作计划能够付诸实施。在此，谨向参加本系列手册编译工作的全体老师表示衷心的感谢！

“TI 中国大学计划”在本系列手册的编译过程中给予了多方面的大力支持，在此向他们表示衷心的感谢！

由于本系列手册的编译工作量大、时间紧，因此，尽管编译的老师和编译委员会都尽了最大的努力，但也难免有不妥，甚至错误之处，编译委员会全体老师恳切地希望广大读者给以批判指正。

清华大学生物医学工程系

胡广书 教授

2004 年 3 月

前 言

TMS320C28x 芯片作为 TI 公司 DSP 系列的新成员,是 TMS320C2000TM 平台下的一款最新推出的 DSP 芯片。C28x 芯片为功能强大的 TMS320TM DSP 结构设计提供了低成本、低功耗和高性能的处理能力,特别适合于电机的数字式运动控制。C28x 具备现有 C2000 DSP 控制器芯片的所有功能且得到了很大的加强,芯片的处理性能更快(150MIPS)、外设集成度更高、程序存储器更大(128 千字)、A/D 转换速度更快、精度更高(12 位)、开发软件集成度更高、开发效率更高、支持 C/C++ 语言等,是电机数字化控制的升级产品。在电机控制、工业控制和电力系统继电保护等方面将会得到大量的应用。

为了满足广大用户的需求,我们将 TI 公司的文献进行选择归类,将相关的部分放在一起,以便读者使用。本译文将两部分文档集成在一起,一部分为设计汇编指令和汇编工具的使用资料,其中使用了编号为 SPRU513 文献,这一部分将涉及如何使用汇编语言工具:汇编器、归档器(档案库存储器)(大容量外存储器)、目标代码链接器、交叉引用列表程序、绝对地址列表器、十六进制转换应用程序;另一部分编号为 SPRU430B 文献将涉及 C28x 汇编语言指令集。

参加本书成书工作的还有尉冰娟、何懋渝、崔晶、吴俊、冼成瑜、杨立勇、张伟、杨利辉、刘钊、李纯、王军生、高苏州等,他们参加了部分资料的翻译、校对、录入等大量工作,在此表示十分的感谢。

本书的成书过程中还得到了重庆大学-美国德州仪器数字信号处理方案实验室、重庆大学电气工程学院电力电子与电力传动系郑群英等许多老师的大力帮助和支持,在此表示衷心的感谢。

在这里还要感谢美国德州仪器公司大学计划所给予的大力支持。

限于编译者的水平,书中难免存在错误和翻译不妥之处,恳请读者批评指正。

作 者

(2004 年 8 月于重庆大学)

目 录

第 1 章	软件开发工具.....	1
1.1	软件开发工具概况.....	2
1.2	软件开发工具介绍.....	2
第 2 章	通用目标文件格式介绍.....	4
2.1	段.....	4
2.2	汇编器如何处理段.....	5
2.2.1	未初始化段.....	6
2.2.2	已初始化段.....	6
2.2.3	已命名段.....	7
2.2.4	子段.....	8
2.2.5	段程序计数器.....	8
2.2.6	使用段伪指令实例.....	8
2.3	链接器如何处理段.....	10
2.3.1	默认内存分配.....	11
2.3.2	在存储器映像中存放段.....	12
2.4	重定位.....	12
	运行中的重定位.....	13
2.5	装载程序.....	14
2.6	COFF 文件中的符号.....	14
2.6.1	外部符号.....	14
2.6.2	符号表.....	15
第 3 章	汇编器.....	16
3.1	汇编器功能.....	16
3.2	在软件开发过程中汇编器的作用.....	17
3.3	运行汇编器.....	17
3.4	为汇编器输入的替换目录命名.....	19
3.4.1	使用-i 汇编器选项.....	20
3.4.2	使用 C2000_A_DIR 或 A_DIR 环境变量.....	20
3.5	源程序语句格式.....	20
3.5.1	标号域.....	21
3.5.2	助记符域.....	22
3.5.3	操作数域.....	22
3.5.4	注释域.....	22

3.6	常量	22
3.6.1	二进制整数	23
3.6.2	八进制整数	23
3.6.3	十进制整数	23
3.6.4	十六进制整数	24
3.6.5	字符常量	24
3.6.6	汇编编译过程使用的 (Assembly-Time) 常量	24
3.6.7	浮点型常量	25
3.7	字符串	25
3.8	符号	25
3.8.1	标号	26
3.8.2	局部标号	26
3.8.3	符号常量	28
3.8.4	定义符号常量 (-d 选项)	28
3.8.5	预定义符号常量	29
3.8.6	置换符号	30
3.9	表达式	31
3.9.1	运算符	31
3.9.2	表达式的上溢和下溢	32
3.9.3	定义明确的表达式	32
3.9.4	条件表达式	32
3.9.5	合法的表达式	33
3.10	内嵌函数	34
3.11	源程序列表	35
3.12	交叉引用列表	36
3.13	灵巧的编码	37
3.14	汇编变量的 C 类型符号调试	38
3.15	TMS320C28x 汇编器的模式	40
3.15.1	C27x 目标模式	40
3.15.2	C28x 目标模式	41
3.15.3	C28x 目标——兼容 C27x 的语法模式	41
3.15.4	C28x 目标——兼容 C2xlp 的语法模式	41
第 4 章	汇编伪指令	44
4.1	伪指令简介	44
4.2	与 TMS320C1x/C2x/C2xx/C5x 汇编伪指令的兼容性	47
4.3	定义段的伪指令	48
4.4	常数初始化伪指令	50
4.5	调准段程序计数器伪指令	52

4.6	输出列表格式伪指令	53
4.7	引用其他文件的伪指令	54
4.8	条件汇编伪指令	54
4.9	汇编过程使用的符号的伪指令	55
4.10	汇编器模式伪指令	56
4.11	其他伪指令	56
4.12	伪指令索引表	57
第 5 章	宏语言	99
5.1	宏的使用	99
5.2	定义宏	100
5.3	宏参数/置换符号	101
5.3.1	定义置换符号的伪指令	102
5.3.2	内置置换符号函数	103
5.3.3	递归的置换符号	104
5.3.4	强制置换	104
5.3.5	访问下标置换符号的单个字符	105
5.3.6	在宏中作为局部变量的置换符号	106
5.4	宏库	107
5.5	在宏中使用条件汇编	107
5.6	在宏中使用标号	109
5.7	在宏中产生信息	110
5.8	用伪指令格式化输出列表	111
5.9	递归和嵌套宏的使用	111
5.10	宏伪指令汇总	113
第 6 章	归档器	115
6.1	归档器概述	115
6.2	软件开发流程中归档器的作用	116
6.3	调用归档器	116
6.4	归档器实例	117
第 7 章	链接器	120
7.1	链接器概述	120
7.2	在软件开发流程中链接器的作用	121
7.3	调用链接器	121
7.4	链接器选项	123
7.4.1	重定位 (-a 和 -r 选项)	124
7.4.2	禁止合并符号调试信息	125
7.4.3	C 语言程序选项 (-c 和 -cr 选项)	125

7.4.4	定义入口 (-e 选项)	125
7.4.5	设置默认的填充值 (-f fill_value 选项)	126
7.4.6	全局化符号 (-g symbol 选项)	126
7.4.7	静态化所有的全局符号 (-h 选项)	126
7.4.8	定义堆的大小 (-heap size 选项)	126
7.4.9	改变库的搜索路径 (-l 选项、-i 选项和 C_DIR 环境变量)	127
7.4.10	取消条件链接	128
7.4.11	忽略分配 (-k 选项)	129
7.4.12	创建映像文件 (-m filename 选项)	129
7.4.13	命名输出文件 (-o filename 选项)	130
7.4.14	隐藏运行信息 (-q 选项)	130
7.4.15	删除符号信息 (-s 选项)	130
7.4.16	定义堆栈的大小 (-stack size 选项)	130
7.4.17	引入未定义的符号 (-u symbol 选项)	131
7.4.18	当创建未定义的段时显示信息 (-w 选项)	131
7.4.19	穷举读库 (-x 选项)	131
7.5	链接器命令文件	132
7.5.1	链接器命令文件中的保留名	133
7.5.2	链接器命令文件中的常量	134
7.6	目标库	134
7.7	MEMORY 伪指令	135
7.7.1	默认的存储器模式	135
7.7.2	MEMORY 伪指令格式	135
7.8	SECTIONS 伪指令	138
7.8.1	SECTIONS 伪指令的格式	138
7.8.2	地址分配	140
7.8.3	规定输入段	143
7.9	指定一个段的运行地址	145
7.9.1	指定装载和运行地址	145
7.9.2	未初始化的段	146
7.9.3	用 label 伪指令访问装载地址	146
7.10	UNION 和 GROUP 语句的使用	147
7.10.1	用 UNION 语句重叠段	147
7.10.2	把输出段在一起分组	149
7.11	重叠页	149
7.11.1	用 MEMORY 伪指令定义重叠页	150
7.11.2	重叠页实例	150
7.11.3	在 SECTIONS 伪指令中使用重叠页	150

7.11.4 对重叠页的存储器分配	151
7.12 特殊段类型	152
7.13 默认分配	152
7.13.1 输出段的形成	153
7.13.2 默认分配算法	153
7.14 链接时给符号赋值	154
7.14.1 赋值语句的格式	154
7.14.2 将 SPC 赋值到一个符号	154
7.14.3 赋值表达式	155
7.14.4 链接器定义的符号	156
7.15 创建和填充空位	156
7.15.1 初始化段和未初始化段	156
7.15.2 创建空位	157
7.15.3 填充空位	158
7.15.4 未初始化段的显式初始化	159
7.16 部分(增量)链接	159
7.17 链接 C 代码	161
7.17.1 运行中的初始化	161
7.17.2 目标库和运行时的支持	161
7.17.3 设置堆栈和堆段的大小	162
7.17.4 运行中变量的自动初始化	162
7.17.5 装载时变量的自动初始化	162
7.17.6 -c 和-cr 链接器选项	163
7.18 链接器举例	163
第 8 章 绝对列表程序	167
8.1 生成绝对列表	167
8.2 调用绝对列表程序	168
8.3 绝对列表程序实例	169
第 9 章 交叉引用列表程序	173
9.1 生成交叉引用列表	173
9.2 调用交叉引用列表程序	174
9.3 交叉引用列表程序举例	175
第 10 章 十六进制转换应用程序	177
10.1 十六进制转换应用程序在软件开发 流程中的作用	178
10.2 调用十六进制转换应用程序	178
10.2.1 从命令行调用十六进制转换应用程序	179
10.2.2 用命令文件调用十六进制转换应用程序	180

10.3	存储器宽度	181
10.3.1	目标宽度	181
10.3.2	存储器宽度	181
10.3.3	数据划分到输出文件	182
10.3.4	规定输出字的字次序	184
10.4	ROMS 伪指令	184
10.4.1	何时使用 ROMS 伪指令	186
10.4.2	ROMS 伪指令举例	186
10.5	SECTIONS 伪指令	188
10.6	分配输出文件名	189
10.7	映像模式和填充 (-fill) 选项	190
10.7.1	生成一个存储器映像	190
10.7.2	填充值	191
10.7.3	在映像模式下所遵循的步骤	191
10.8	控制 ROM 器件地址	191
10.9	目标格式	192
10.9.1	ASCII-Hex 目标格式 (-a 选项)	192
10.9.2	Intel MCS-86 目标格式 (-i 选项)	193
10.9.3	Motorola-S 目标格式 (-m 选项)	194
10.9.4	TI-Tagged SDSMAC 目标格式 (-t 选项)	194
10.9.5	扩展 Tektronix 目标格式 (-x 选项)	195
10.10	十六进制转换应用程序错误信息	196
第 11 章	C28x 汇编语言指令集	197
11.1	指令概述 (按功能分类)	197
11.2	寄存器操作	199
附录 A	通用目标文件格式	428
附录 B	符号调试伪指令	444
附录 C	汇编器错误信息	453
附录 D	链接器错误信息	465
附录 E	术语表	477



第 1 章 软件开发工具

TMS320C28xTM 由一系列软件开发工具支持，包括 C/C++ 优化编译器、汇编器、链接器和各种应用程序。本章将简要说明这些工具在软件开发流程中是如何协调工作的，并对各种工具作简要介绍。

TMS320C28x 汇编语言开发工具有：

- ☐ 汇编器
- ☐ 归档器
- ☐ 目标代码链接器
- ☐ 交叉引用列表程序
- ☐ 绝对地址列表程序
- ☐ 十六进制转换应用程序

1.1 软件开发工具概况

图 1.1 所示为 TMS320C28x 软件开发流程。阴影部分强调最常用的开发路径；其他部分是可选的，它们用于增强开发能力的外围功能。

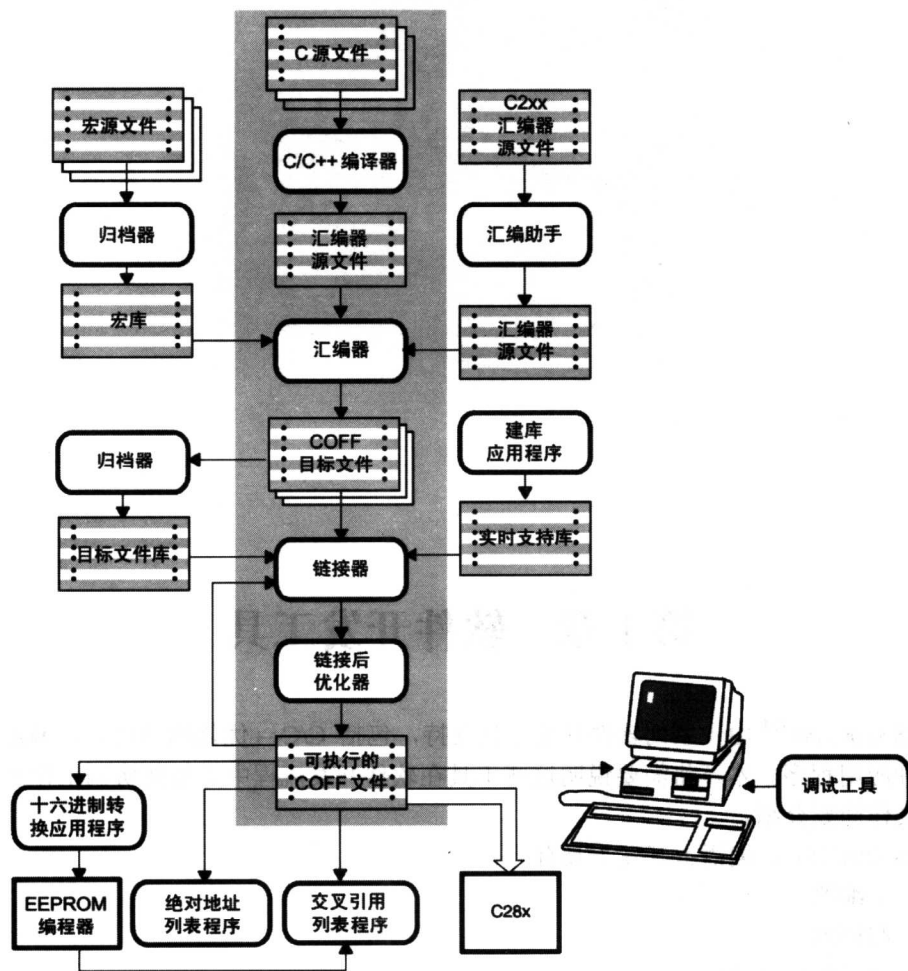


图 1.1 TMS320C28x 软件开发流程图

1.2 软件开发工具介绍

下面将简要介绍图 1.1 中所列出的工具：

- **C/C++编译器** 接收 C 和 C++源代码并生成 TMS320C28x 汇编语言源代码。编译

程序包中包括集成 (shell) 程序、优化器、交叉引用列表程序:

- 集成程序使用户能够一步完成编译、汇编和链接过程
- 优化器优化代码以提高 C 语言程序的效率
- 交叉引用列表程序使汇编语言输出和 C 源语句进行交互访问, 使其能与经编译器编译后的代码相关联

要想获取更多信息, 请查阅 TMS320C28x C/C++ 优化编译器用户手册。

- **汇编器** 将汇编语言源文件转换成机器语言 COFF 目标文件。源文件中包含指令、汇编伪指令和宏伪指令。用户可以使用汇编伪指令控制汇编过程的各个方面, 如, 源程序列表格式、数据排列以及段内容。可以从第 3 章汇编器介绍到第 5 章宏语言章节中了解到更多信息。若想获得汇编语言指令集的详细内容, 可以参见第 11 章的内容。
- **链接器** 将目标文件组合成一个可执行 COFF 目标模块。在它创建可执行模块的同时, 进行重定位和解决外部引用。链接器接收可重新定位的 COFF 目标文件 (由汇编器创建) 作为输入。它也接收上一次链接器运行所创建的档案库成员和输出模块。链接器伪指令允许用户组合目标文件段, 把段或符号约定在存储器的某些地址范围内, 并定义和重新定义全局符号。要了解更详细的内容, 请参阅第 7 章链接器介绍。
- **归档器** 允许用户把一组文件收集到单个档案文件, 称为库。例如, 用户可以收集几个宏放入宏库中。汇编器搜索库并使用被源文件称作宏的成员。用户也可以收集一组目标文件放入目标库。链接时链接器将确定的外部引用包含到库中。归档器允许用户以删除、替换、提取、增加成员的方式修改库。阅读第 6 章归档器介绍, 可以获得更多信息。
- 用户可以使用 **建库应用程序** 来建立用户自己的实时支持库。要想获取更多信息, 请参阅 TMS320C28x C/C++ 优化编译器用户手册。
- **绝对地址列表程序** 接收目标文件作为输入, 创建 .abs 文件作为输出。用户可以汇编 .abs 文件生成包含绝对地址而不是相对地址的一个列表。如果没有绝对地址列表程序, 生成这种列表的工作将是冗长乏味的, 可能需要许多手工操作。
- **十六进制转换应用程序** 将 COFF 目标文件转换成 TI-Tagged, ASCII-hex, Intel, motorola-S 或 Tektronix 目标格式。这种转换后的文件可以用编程器下载到 EPROM。详情请阅读第 10 章十六进制转换应用程序介绍。
- **交叉引用列表程序** 使用目标文件来生成交叉列表, 显示符号、符号的定义及它们在已链接的源文件中的引用情况。详情请阅读第 9 章交叉引用列表程序介绍。
- 开发过程的主要产物是可以被 TMS320C28x 器件执行的程序模块。
- 用户可以使用下列几种调试工具中的一种来精简和纠正代码。可利用的产品包括:
 - 软件仿真器
 - XDS 仿真器
 - 评估板 (EVM)

要获取有关调试工具的信息, 请阅读 TMS320C28x 代码设计人员工作室用户手册。



第 2 章 通用目标文件格式介绍

汇编器和链接器可以创建 TMS320C28x™ 器件可执行的目标文件。这些目标文件的格式称为通用目标文件格式（COFF）。

COFF 使得程序的模块化编程更容易。因为它鼓励用户在编写汇编语言程序时用代码块和数据块，这些块结构称为段。汇编器和链接器均提供伪指令允许用户创建和生成段。这一章主要叙述汇编语言程序中段的概念及其使用方法。查阅附录 A 通用目标文件格式，可以获取有关 COFF 目标文件格式的更多信息。

2.1 段

目标文件的最小单元称为段。段是代码或数据块的组合，它最终将在存储器映像中占据一个连续的空间。目标文件的每一个段都是各自独立的。COFF 目标文件必须包含 3 个默认段：