

CCS

DSP集成开发环境

CCS

开发指南

尹 勇 欧光军 关荣锋 编著



北京航空航天大学出版社

# DSP 集成开发环境 CCS 开发指南

尹 勇 欧光军 关荣锋 编著

北京航空航天大学出版社

## 内容简介

本书详细介绍了如何使用 TI 公司的 DSP 软件开发集成环境 CCS2 开发和调试 C 语言及 DSP 汇编语言程序, 尽量达到帮助读者熟练使用 CCS2 软件的目的。本书不是介绍 DSP 软件和硬件方面的知识, 而是采用实例方式循序渐进地介绍如何操作 CCS2 软件。读者只要按照实例步骤实践, 就能在最短的时间内熟练使用 CCS2。

本书可供从事 DSP 应用系统开发的技术人员、工程师以及高等院校工科电子类专业师生学习参考。

## 图书在版编目(CIP)数据

DSP 集成开发环境 CCS 开发指南/尹勇等编著. —北京:  
北京航空航天大学出版社, 2003. 11

ISBN 7-81077-363-1

I. D… II. 尹… III. 软件工具, CCS 2.0—指南

IV. TP311.56-62

中国版本图书馆 CIP 数据核字(2003)第 094310 号

## DSP 集成开发环境 CCS 开发指南

尹 勇 欧光军 关荣锋 编著

责任编辑 孔祥燮

\*

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(邮编 100083) 发行部电话:(010)82317024 传真:(010)82328026

<http://www.buaapress.com.cn> E-mail: bhp@buaa.net

北京市松源印刷有限公司印装 各地书店经销

\*

开本: 787×1 092 1/16 印张: 18.25 字数: 467 千字

2003 年 11 月第 1 版 2004 年 7 月第 2 次印刷 印数: 5 001~10 000 册

ISBN 7-81077-363-1 定价: 28.00 元

## 前 言

数字信号处理器 DSP(Digital Signal Processor)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。可编程 DSP 芯片的开发与应用是当前电子领域的热点,CCS 的推出是 DSP 软件开发的一次革命性突破。CCS2(Code Composer Studio Version 2.0)代码调试器是一种针对标准 TMS320 调试接口的 DSP 芯片集成开发环境 IDE(Integrated Development Environment),由 TI 公司在 1999 年推出。CCS 目前有 CCS1.1、CCS1.2、CCS2.0 和 CCS2.2 等几个版本,有 CCS2000(针对 C2XX)、CCS5000(针对 C54XX、C55XX)、CCS6000(针对 C6X)等几个不同的型号。

本书讨论了基于 TMS320C54X 系列芯片的 CCS2 集成开发环境的使用,尽量达到帮助读者熟练使用 CCS2.0 软件的目的。读者应了解 TMS320C54X 的硬件结构以及芯片的各种资源,熟悉 TMS320C54X 的指令系统。同时,还要求读者会使用 DSP 汇编语言和标准 C/C++ 语言进行程序设计。

本书不涉及 DSP 的硬件和软件开发设计以及信号处理的理论问题,在有关的程序例子中假定读者对所涉及的 DSP 硬件和软件开发设计以及信号处理的理论有基本的了解。

全书正文共 9 章。

第 1 章介绍 CCS2 的安装与配置;

第 2 章简单介绍 CCS2;

第 3 章深入介绍 CCS 集成开发环境;

第 4 章介绍 CCS 的 C 语言调试实例;

第 5 章介绍 CCS 的汇编语言调试实例;

第 6~9 章介绍 CCS2 高级使用——使用文件 I/O、DSP/BIOS 原理与应用、RTDX 原理与应用和使用 GEL。为了便于查阅,在附录中列出了 TMS320C54X 的汇编助记符指令集和汇编伪指令。

本书的第 1 章、第 2 章、第 3 章、第 5 章由华中科技大学尹勇博士执笔,第 4 章由华中科技大学欧光军博士执笔,第 6 章由华中科技大学关荣锋博士执笔,第 7 章由华中科技大学李红杰博士执笔,第 8 章、第 9 章和附录由华中科技大学张超勇、朱传军、李林凌博士和将闻悦同学执笔。全书由尹勇统编。本书在编写过程中受到中南财经政法大学项铭硕士的鼎力帮助,特表感谢。书的出版同时得到北京航空航天大学大力支持和鼓励,在此深表敬意。

由于作者水平有限,书中出现的错误和不妥之处,恳请广大读者批评指正!

作 者

2003 年 6 月于喻园

# 目 录

|                               |    |
|-------------------------------|----|
| 第 1 章 CCS2 的安装与配置 .....       | 1  |
| 1.1 DSP 芯片的开发工具介绍 .....       | 1  |
| 1.2 CCS2 的主要特性 .....          | 1  |
| 1.3 CCS2 的安装 .....            | 3  |
| 1.3.1 系统需求 .....              | 3  |
| 1.3.2 安装 CCS2 .....           | 3  |
| 1.4 CCS2 的系统配置 .....          | 4  |
| 第 2 章 初识 CCS2 .....           | 8  |
| 2.1 CCS2 的组成 .....            | 8  |
| 2.2 CCS2 代码生成工具 .....         | 8  |
| 2.3 CCS2 集成开发环境功能介绍 .....     | 10 |
| 2.3.1 强大的源代码编辑器 .....         | 10 |
| 2.3.2 方便的应用程序生成特性 .....       | 11 |
| 2.3.3 方便的应用程序调试特性 .....       | 11 |
| 2.4 DSP/BIOS 插件 .....         | 12 |
| 2.4.1 DSP/BIOS 设置 .....       | 13 |
| 2.4.2 DSP/BIOS 的 API 模块 ..... | 14 |
| 2.5 硬件仿真和实时数据交换 .....         | 15 |
| 2.5.1 硬件仿真 .....              | 15 |
| 2.5.2 实时数据交换 .....            | 15 |
| 第 3 章 深入 CCS 集成开发环境 .....     | 18 |
| 3.1 CCS 集成开发环境的特性 .....       | 18 |
| 3.2 菜单栏 .....                 | 19 |
| 3.2.1 File 菜单 .....           | 19 |
| 3.2.2 Edit 菜单 .....           | 21 |
| 3.2.3 View 菜单 .....           | 23 |
| 3.2.4 Project 菜单 .....        | 25 |
| 3.2.5 Debug 菜单 .....          | 26 |
| 3.2.6 Profiler 菜单 .....       | 27 |
| 3.2.7 Option 菜单 .....         | 28 |
| 3.2.8 GEL 菜单 .....            | 31 |
| 3.2.9 Tool 菜单 .....           | 31 |
| 3.2.10 DSP/BIOS 菜单 .....      | 34 |

|                                   |           |
|-----------------------------------|-----------|
| 3.3 工具栏.....                      | 34        |
| 3.3.1 标准工具栏.....                  | 34        |
| 3.3.2 工程工具栏.....                  | 35        |
| 3.3.3 调试工具栏.....                  | 35        |
| 3.3.4 编辑工具栏.....                  | 36        |
| 3.3.5 剖析工具栏.....                  | 36        |
| 3.4 工程管理.....                     | 37        |
| 3.4.1 建立、打开和关闭工程 .....            | 37        |
| 3.4.2 向工程中添加或删除文件.....            | 38        |
| 3.4.3 查看文件关联性.....                | 39        |
| 3.4.4 工程文件剖析.....                 | 40        |
| 3.5 源文件管理.....                    | 41        |
| 3.5.1 创建新的源文件.....                | 41        |
| 3.5.2 打开文件.....                   | 41        |
| 3.5.3 保存文件.....                   | 41        |
| 3.6 文件编辑.....                     | 42        |
| 3.6.1 设置编辑属性.....                 | 42        |
| 3.6.2 拷贝、剪切和粘贴文本 .....            | 43        |
| 3.6.3 编辑整列.....                   | 43        |
| 3.6.4 跳到指定行.....                  | 43        |
| 3.6.5 查找和替换文本.....                | 45        |
| 3.6.6 利用书签.....                   | 46        |
| <b>第 4 章 CCS 的 C 语言调试实例 .....</b> | <b>48</b> |
| 4.1 创建一个新工程.....                  | 48        |
| 4.2 向工程中添加文件.....                 | 49        |
| 4.3 代码浏览.....                     | 50        |
| 4.4 编译、链接和运行程序 .....              | 52        |
| 4.5 改变程序设置并查找语法错误.....            | 54        |
| 4.6 使用断点和观察窗口.....                | 55        |
| 4.7 观察结构体的值.....                  | 56        |
| 4.8 测试代码执行统计.....                 | 57        |
| <b>第 5 章 CCS 的汇编语言调试实例 .....</b>  | <b>59</b> |
| 5.1 载入可执行程序.....                  | 61        |
| 5.2 使用反汇编工具.....                  | 61        |
| 5.3 利用断点调试程序.....                 | 62        |
| 5.3.1 断点的设置和取消.....               | 62        |
| 5.3.2 程序的执行.....                  | 63        |

|                                        |           |
|----------------------------------------|-----------|
| 5.3.3 流水线冲突的解决                         | 64        |
| 5.3.4 查看 CPU 寄存器的值                     | 66        |
| 5.3.5 查看内存数据                           | 72        |
| 5.3.6 查看变量的值                           | 75        |
| 5.4 剖析点的调试                             | 76        |
| <b>第 6 章 CCS2 高级使用——使用文件 I/O</b>       | <b>80</b> |
| 6.1 探测点与文件 I/O                         | 80        |
| 6.2 利用探测点观察寄存器的值                       | 80        |
| 6.2.1 探测点的设置与删除                        | 80        |
| 6.2.2 观察寄存器的值                          | 81        |
| 6.3 利用文件 I/O                           | 82        |
| 6.3.1 I/O 文件格式                         | 82        |
| 6.3.2 打开工程文件                           | 83        |
| 6.3.3 阅读源代码                            | 84        |
| 6.3.4 设置 PC 数据文件与探测点关联                 | 86        |
| 6.3.5 设置图形显示窗口                         | 89        |
| 6.3.6 程序的动画执行                          | 90        |
| 6.3.7 增益的调节                            | 92        |
| 6.3.8 查看变量属性和值                         | 93        |
| 6.3.9 从文件读入数据到内存                       | 95        |
| <b>第 7 章 CCS2 高级使用——DSP/BIOS 原理与应用</b> | <b>98</b> |
| 7.1 DSP/BIOS 介绍                        | 98        |
| 7.2 DSP/BIOS 组件                        | 99        |
| 7.2.1 实时库与 API 函数                      | 100       |
| 7.2.2 DSP/BIOS 配置工具                    | 100       |
| 7.2.3 DSP/BIOS 插件                      | 102       |
| 7.3 DSP/BIOS 命名规则                      | 103       |
| 7.3.1 头文件命名                            | 103       |
| 7.3.2 对象命名                             | 103       |
| 7.3.3 函数命名                             | 103       |
| 7.3.4 数据类型名                            | 104       |
| 7.3.5 存储器段命名                           | 104       |
| 7.4 DSP/BIOS 程序生成过程                    | 105       |
| 7.4.1 使用配置工具                           | 105       |
| 7.4.2 创建 DSP/BIOS 程序所使用的文件             | 108       |
| 7.4.3 编译和链接 DSP/BIOS 应用程序              | 109       |
| 7.4.4 DSP/BIOS 应用程序执行顺序                | 112       |

|                                           |            |
|-------------------------------------------|------------|
| 7.5 DSP/BIOS 仪表 .....                     | 124        |
| 7.5.1 实时分析 .....                          | 124        |
| 7.5.2 软件仪表与硬件仪表 .....                     | 125        |
| 7.5.3 仪表性能 .....                          | 125        |
| 7.5.4 仪表 API .....                        | 126        |
| 7.6 创建一个 DSP/BIOS 程序 .....                | 126        |
| 7.6.1 打开存在的工程 .....                       | 126        |
| 7.6.2 剖析 stdio.h 的执行时间 .....              | 128        |
| 7.6.3 使用 HELLO2 文件 .....                  | 129        |
| 7.6.4 创建一个配置文件 .....                      | 130        |
| 7.6.5 添加 DSP/BIOS 文件到工程 .....             | 131        |
| 7.6.6 用 CCS2 测试 .....                     | 132        |
| 7.6.7 剖析 DSP/BIOS 代码执行时间 .....            | 133        |
| 7.7 调试 DSP/BIOS 程序 .....                  | 134        |
| 7.7.1 打开并检查工程 .....                       | 134        |
| 7.7.2 观察源代码 .....                         | 135        |
| 7.7.3 修改配置文件 .....                        | 137        |
| 7.7.4 使用执行图观察线程执行 .....                   | 139        |
| 7.7.5 更改和观察 Load 函数的执行 .....              | 141        |
| 7.7.6 分析线程统计 .....                        | 142        |
| 7.7.7 添加显式 STS 仪表 .....                   | 143        |
| 7.7.8 观察显式仪表 .....                        | 145        |
| <b>第 8 章 CCS2 高级使用——RTDX 的原理与应用 .....</b> | <b>148</b> |
| 8.1 RTDX 介绍 .....                         | 148        |
| 8.2 使用 RTDX 插件 .....                      | 149        |
| 8.2.1 RTDX 配置控制窗口 .....                   | 149        |
| 8.2.2 RTDX 通道控制窗口 .....                   | 150        |
| 8.2.3 RTDX 诊断控制窗口 .....                   | 151        |
| 8.3 配置 RTDX .....                         | 152        |
| 8.3.1 修改 DSP 目标缓冲区大小 .....                | 152        |
| 8.3.2 修改主机缓冲区大小 .....                     | 152        |
| 8.3.3 RTDX 工作模式 .....                     | 152        |
| 8.3.4 RTDX 目标中断屏蔽 .....                   | 153        |
| 8.4 实时通信程序的设计方法 .....                     | 153        |
| 8.4.1 编写目标 DSP 应用程序 .....                 | 154        |
| 8.4.2 编写 RTDX OLE 自动化客户程序 .....           | 155        |
| 8.4.3 在 CCS 中使能 RTDX .....                | 156        |
| 8.4.4 运行 OLE 自动化客户程序 .....                | 156        |



|                                      |            |
|--------------------------------------|------------|
| 8.5 RTDX 实例一 .....                   | 156        |
| 8.5.1 从 DSP 目标系统传输一个整数到 PC 主机 .....  | 156        |
| 8.5.2 从 PC 主机传输一个整数到 DSP 目标系统 .....  | 159        |
| 8.6 RTDX 实例二 .....                   | 161        |
| 8.6.1 从 DSP 目标系统传输一个整数到 PC 主机 .....  | 161        |
| 8.6.2 从 PC 主机传输一个整数到 DSP 目标系统 .....  | 164        |
| <b>第 9 章 CCS2 高级使用——使用 GEL</b> ..... | <b>166</b> |
| 9.1 GEL 语言简介 .....                   | 166        |
| 9.2 GEL 语法 .....                     | 166        |
| 9.3 GEL 函数 .....                     | 169        |
| 9.3.1 GEL 函数定义 .....                 | 169        |
| 9.3.2 GEL 函数参数 .....                 | 169        |
| 9.3.3 调用 GEL 函数 .....                | 171        |
| 9.3.4 加载/卸载 GEL 函数 .....             | 171        |
| 9.4 用关键词将 GEL 函数添加到菜单中 .....         | 171        |
| 9.5 输出窗口函数 .....                     | 174        |
| 9.6 在 CCS2 启动时自动执行 GEL 函数 .....      | 175        |
| 9.7 嵌入式 GEL 函数 .....                 | 176        |
| 9.8 GEL 使用实例 .....                   | 183        |
| 9.8.1 一个简单的 GEL 函数 .....             | 183        |
| 9.8.2 定义局部变量 .....                   | 183        |
| 9.8.3 GEL 函数的自动执行 .....              | 184        |
| 9.8.4 用 GEL 控制 DSP 变量 .....          | 186        |
| <b>附 录 指令详解</b> .....                | <b>188</b> |
| <b>参考文献</b> .....                    | <b>279</b> |

# 第 1 章 CCS2 的安装与配置

## 1.1 DSP 芯片的开发工具介绍

DSP 芯片的开发需要一套完整的软、硬件开发工具。DSP 芯片的开发工具可以分为代码生成工具和代码调试工具两类。

代码生成工具的作用是将 C 语言、汇编语言或两者的混合语言编写的 DSP 源代码程序编译、汇编并链接成可执行的 DSP 代码。C 编译器、汇编器和链接器是 DSP 代码生成工具所必须的。

- C 编译器(C compiler)将 C 语言源代码程序自动地编译成 C54X 的汇编语言源代码程序。
- 汇编器(assembler)将汇编语言源代码文件汇编成机器语言 COFF 目标文件,在源文件中包含了汇编指令、宏命令及指令等。
- 链接器(linker)把汇编生成的可重定位的 COFF 目标模块组合成一个可执行的 COFF 目标模块。它能调整并解决外部符号参考。链接器的输入是可重定位的 COFF 目标文件和目标库文件,它也可以接受来自文档管理器中的目标文件以及链接以前运行时所产生的输出模块。

代码调试工具的作用是对 DSP 程序及目标系统进行调试,使之能够达到设计目标。TMS320 系列 DSP 芯片的系统集成和调试工具主要有:C 语言源代码调试器、汇编语言源代码调试器、初学者工具 DSK(Designer's Starter Kit)、软件仿真器(simulator)、评估模块 EVM(Evaluation Module)、仿真器 XDS(eXtended Developing System)和软件开发系统 SWDS(Software Developing System)等。

## 1.2 CCS2 的主要特性

CCS2(Code Composer Studio Version 2.0)代码调试器是一种针对标准 TMS320 调试接口的集成开发环境 IDE(Integrated Development Environment),由 TI 公司在 1999 年推出。CCS 目前有 CCS1.1、CCS1.2、CCS2.0 和 CCS2.2 等几个版本,有 CCS2000(针对 C2XX)、CCS5000(针对 C54XX、C55XX)和 CCS6000(针对 C6X)等几个不同的型号。各个不同的版本和型号之间的差别并不大。在 TI 公司的官方网站(<http://www.ti.com>)上,可以下载免费使用期限为 30 天的试用版本。

CCS2 包含源代码编辑工具、代码调试工具、可执行代码生成工具和实时分析工具,并支持设计和开发的整个流程,如图 1.1 所示。

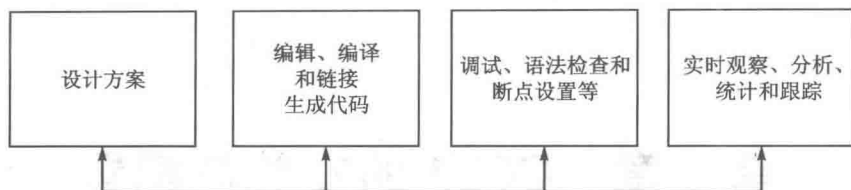


图 1.1 CCS2 的开发流程

CCS2 除了包含 HLL 调试器的主要特性外,还包括下列特性:

- 完全集成的开发环境 CCS2 将 TI 公司的汇编器、编译器、链接器和调试器等都集成到它的开发环境中,用户可以从菜单栏中选择 TI 公司的各种工具,并且可以直接观察到流水线输出到窗口的编译结果。同时,出错信息加亮显示,只要双击出错信息便可以打开源文件,光标停留在出错的地方。在 Windows 环境中,用户可以很方便地同时编辑、调试和编译源程序。代码编译器可以跟踪一个项目中的所有文件及其相关内容。用户可以选择编译单个的文件,或者将所有文件包括在一个项目中,或者逐步建立项目。在编辑器、编译器、链接器和调试器选项中有方便的对话框。
- 高度集成的源代码编辑器 能动态提示 C 语言和 DSP 汇编语言源代码,能很容易地阅读和理解源代码,及时发现和定位语法错误。CCS2 是一个完全集成的包含 TI 编译器的开发环境。CCS2 目标管理系统、内建编辑器和所有的调试分析能力都集成在 Windows 环境中。
- 支持编辑和调试的后台编辑 用户在编译和调试程序时,不必退出系统而回到 DOS 系统中,CCS2 会自动将这些工具交互式地装载到它的环境中。在代码调试窗口中,只要双击错误,就可以直接显示源代码的出错处。
- 对 C 语言源文件和 DSP 汇编语言文件的目标管理 编辑器能跟踪所有文件及其相关内容。这样,编译器只对最近一次编译中改变过的文件进行编译,节省了编译时间。CCS2 在 Windows 98 和 Windows 2000 中支持多线程处理,并行管理调试器(PDM),允许将命令传播给所有的或所选择的处理器。
- 文件探针在算法中通过文件提取或加入信号和数据 CCS2 允许用户从 PC 机中的文件直接读取或写入信号流,而不是实时地读取输入信号。这样,用户可以使用已有的文件来仿真算法。
- 可以在后台执行 DOS 程序 可以执行后台 DOS 命令,并将其输出通过流水线的方式输出,允许用户将应用集成到 CCS2。
- 图形分析功能 具有强大的图形分析功能。
- 方便的代数分解窗口 可以选择查看 C 语言格式的代数表达式,以便容易读懂操作码。
- 有在任何算法点观察信号的图形窗口探针 图形显示窗口使用户能够观察时域或者频域内的信号。对于频域图,FFT 在 PC 机上执行,以便观察所感兴趣的部分而不需要改变它的 DSP 代码。图形显示也可以同探针相连接,当前显示窗口被更新时,探针被确定;当代码执行到这一点时,可以迅速地观察到信号。
- 有状态观察窗口 CCS2 的可视窗口允许用户直接进入 C 表达式及相关变量,结构、数

组和指针等都能很简单的进行增加和减少,以便进入复杂结构。

## 1.3 CCS2 的安装

### 1.3.1 系统需求

下面给出了 CCS2 对 PC 机的最低配置要求。

软件需求:Microsoft Windows 95、Microsoft Windows 98、Microsoft Windows 2000 或 Windows NT4.0。

硬件需求:至少 32 MB 的 RAM,100 MB 的剩余磁盘空间,Pentium100 以上的处理器和 SVGA 显示器(分辨率 800×600 以上)。

### 1.3.2 安装 CCS2

将 CCS2 安装光盘插入 CD-ROM 驱动器中,此时启动光盘自动运行程序,提示用户是否要安装 CCS2。也可以运行光盘根目录下的 setup.exe,按照安装提示,一步步完成安装(系统的默认安装目录是 c:\ti)。同时,建议安装 Acrobat,这是为方便用户阅读 CCS2 自带的帮助文档。安装成功后,安装程序将自动在桌面上创建 CCS2studio 和 Setup CCS2studio 2 个快捷图标。

成功地安装完 CCS2 后,将在安装文件夹中产生下列目录和文件:

- C5400\bios TI 代码产生工具目录;
- C5400\examples 源代码例子目录;
- C5400\rt dx RTDX(实时数据交换)文件目录;
- C5400\tutorial CCS2 使用帮助目录;
- CC\bin CCS2 环境配置目录;
- CC\gel CCS2 中使用的 GEL(通用扩展语言)目录;
- Bin 应用程序目录;
- Docs CCS2 的相关文档和手册;
- Myprojects CCS2 存放用户设计工作目录。

在 CCS2 中,常用的文件类型如下:

- .mak CCS2 使用的工程文件;
- .c C 语言程序源代码文件;
- .asm DSP 汇编语言程序源代码文件;
- .h C 语言程序和 DSP/BIOS API 中的头文件;
- .lib 库文件;
- .cmd 链接命令文件;
- .obj 汇编或编译后产生的目标文件;
- .out 完成汇编、链接和编译所产生的可执行程序文件;
- .wks 保存环境设置的工作文件;
- .cdb 调用 DSP/BIOS API 所必须的配置数据库文件。

如果使用了 DSP/BIOS,当保存配置文件的同时,还将产生下列类型的文件:

- \* cfg. cmd 由配置工具所产生的链接命令文件,在链接生成可执行文件时使用。此文件定义了 DSP/BIOS 应用程序的链接选项、目标名和 DSP 程序的通用数据块,如.txt、.bss 和 .data 等。注意这样的文件名在扩展名前面有 cfg 三个字母。
- \* cfg. h54 配置工具所产生的头文件。此头文件由 \* cfg. s54 所包含。
- \* cfg. s54 配置工具所产生的汇编源代码。
- \* cfg. 054 配置工具产生的源代码文件所生成的目标文件。

CCS2 安装完成后,系统将自动设置以下环境变量:

- C54X\_A\_DIR 汇编器查找 DSP/BIOS 和 RTDX 及代码生成工具所需要的库文件和头文件的搜索路径。
- C54X\_C\_DIR 编译器和链接器查找 DSP/BIOS 和 RTDX 及代码生成工具所需要的库文件和头文件的搜索路径。
- PATH CCS2 安装完成后,在安装目录中将\cgtools\bin 和\bin 添加到路径中。比如,CCS2 安装到 D:\programs,则把 D:\programs\c5400\cgtools\bin 和 D:\programs\c5400\bin 添加到路径中。

如果使用的是 Windows 95 操作系统,则应该增加 DOS 运行环境的内存容量,以支持运行 CCS2 所需要的环境变量。把下面一行程序添加到系统文件 config. sys 中,重新启动计算机。

```
shell=c:\windows\command.com /e:4096 /p
```

## 1.4 CCS2 的系统配置

在第一次运行 CCS2 软件之前,需要运行 CCS2 软件设置程序。CCS2 软件设置程序是建立 CCS2 集成开发环境与 DSP 目标系统或者 Simulator 之间的通信接口。CCS2 软件集成了 TI 公司的 Simulator 和 Emulator 的驱动程序,用户可以直接使用 TI 的仿真器进行开发和调试。如果使用的仿真器不是 TI 公司的,则需要安装相应的仿真器的驱动程序。

每个标准的配置或用户自定义的配置都放在一个特殊格式的文件中。当前系统的配置记录在系统的注册表中,这个配置可以被 CCS2 的配置工具修改更新。当 CCS2 的配置工具建立了一个新的配置时,这个新的配置就记录到注册表中,取代了原先的配置。如果在工程中需要使用几种仿真工具时,就需要给每个仿真工具做相应的配置,并且在使用时需要在这几个配置中来回切换。

双击桌面上的 Setup CCS2studio 图标,或者在开始菜单中运行 Setup Code Composer Studio 程序,可以激活 CCS2 的配置工具,配置界面如图 1.2 所示。

配置界面分为三部分:

图 1.2 左边一栏中的 My System 表示系统配置;C55X Simulator/C54X Simulator 表示当前的仿真器目标板;CPU 表示相应的处理器。这个界面中列举了已经安装的可供选择的配置。改变当前的工作配置时,需要保存改变的配置才能生效。中间一栏表示已经安装的仿真器类型。图中是已经安装的 CCS2 自带的 TI 公司的 C54X 和 C55X 系列仿真器。右边一栏是配置命令/信息选项,单击 Import a Configuration File,便出现仿真器导入配置对话框。对话

框中显示的是已经安装了驱动程序的可以导入的仿真器类型,如图 1.3 所示。如果不改变当前的配置,单击 Close 按钮。

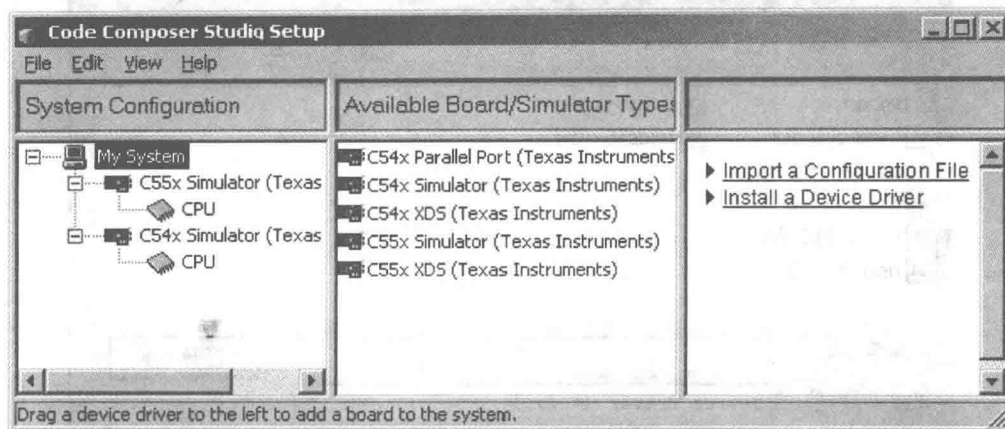


图 1.2 CCS2 配置界面

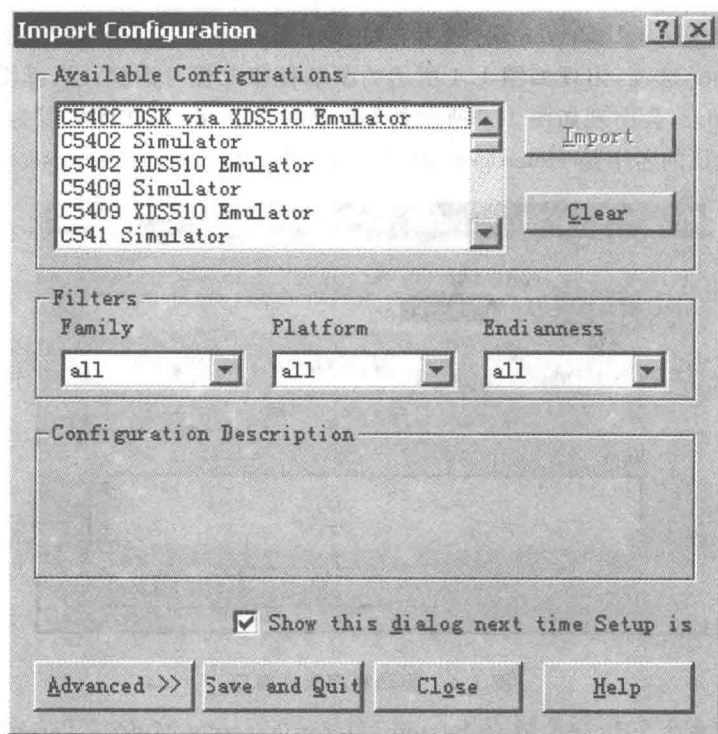


图 1.3 导入配置对话框

单击 Install a Device Driver 选项,便出现驱动程序配置对话框,如图 1.4 所示。

下面举例说明 CCS2 的配置步骤,假设已经将驱动程序安装在 D:\programs\CCS2 目录下。

(1) 双击桌面上的 Setup CCS2studio 图标,运行 CCS2 的配置程序,出现如图 1.2 所示的

配置界面。

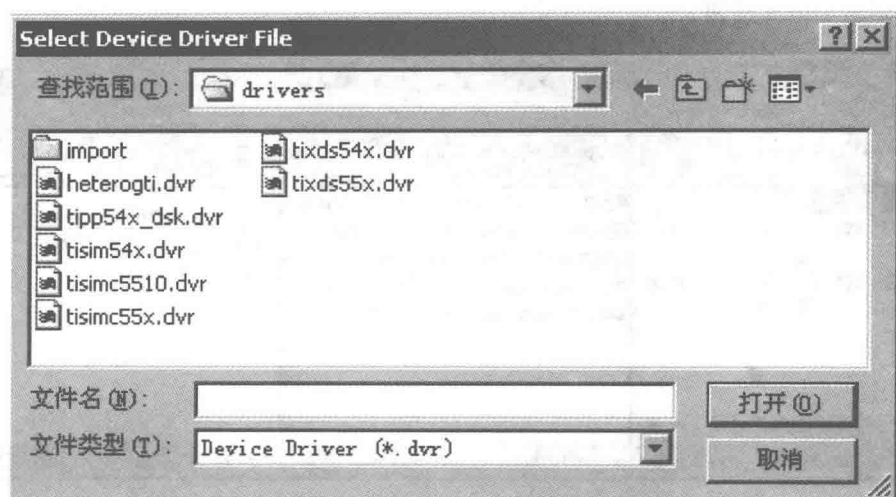


图 1.4 驱动程序配置对话框

(2) 选择 Edit→Install Driver 项,或者在如图 1.2 所示的配置界面的右边一栏单击 Install a Device Driver 选项,出现如图 1.4 所示的驱动程序配置对话框。选择需要的驱动程序,这里选 heterogti.dvr,会出现如图 1.5 所示的驱动程序属性窗口,在此可以为驱动程序重新命名。该驱动程序默认的名字是 Heterogeneous Multi - Target (Texas Instruments),单击 OK。

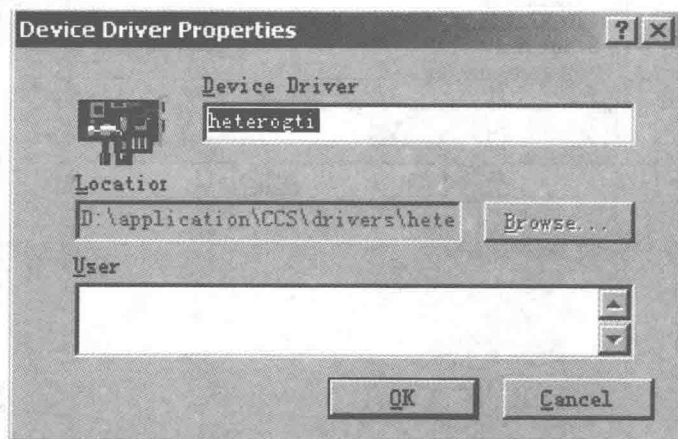


图 1.5 驱动程序属性窗口

(3) 执行菜单命令 Edit→Add to System, Heterogeneous Multi - Target (Texas Instruments)便出现在 CCS2 配置界面的中间一栏,如图 1.6 所示(与图 1.2 对照,中间一栏多了 Heterogeneous Multi - Target (Texas Instruments)一项)。

如果是配置 Simulator,在这一步就可以使用 CCS2 的 Simulator 功能。如果要配置 Emulator,则还需要进行步骤(4)~(6)的配置。

(4) 双击 Heterogeneous Multi - Target (Texas Instruments),出现如图 1.7 所示的属性设置对话框,单击 Board Properties 标签,设置 I/O 端口,这里为 0x240。

(5) 单击图 1.7 中的 Processor Configuration 标签,单击 Add Single 按钮将 CPU-1 加入配置,单击 Finish 按钮结束。

(6) 执行菜单命令 File→Save,保存设置,使配置生效。

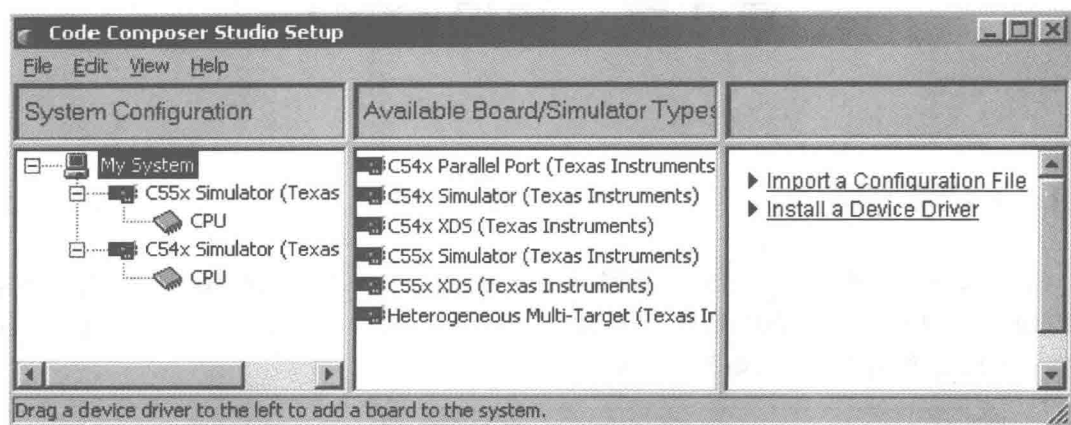


图 1.6 配置后的界面

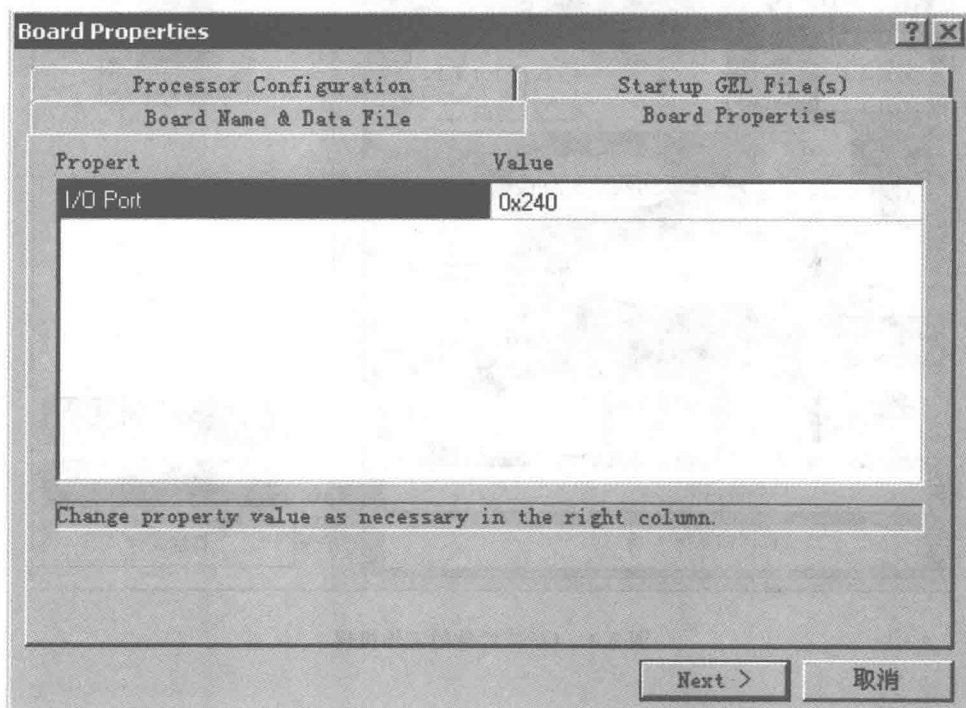


图 1.7 属性设置对话框



## 第 2 章 初识 CCS2

### 2.1 CCS2 的组成

CCS2 集成了下列组件: TMS320 系列 DSP 可执行代码生成工具、源代码集成开发环境、DSP/BIOS 插件和 API、RTDX 插件、主机接口和 API。所有这些组件在 CCS2 开发包中协调工作。CCS2 组件的工作机理如图 2.1 所示。

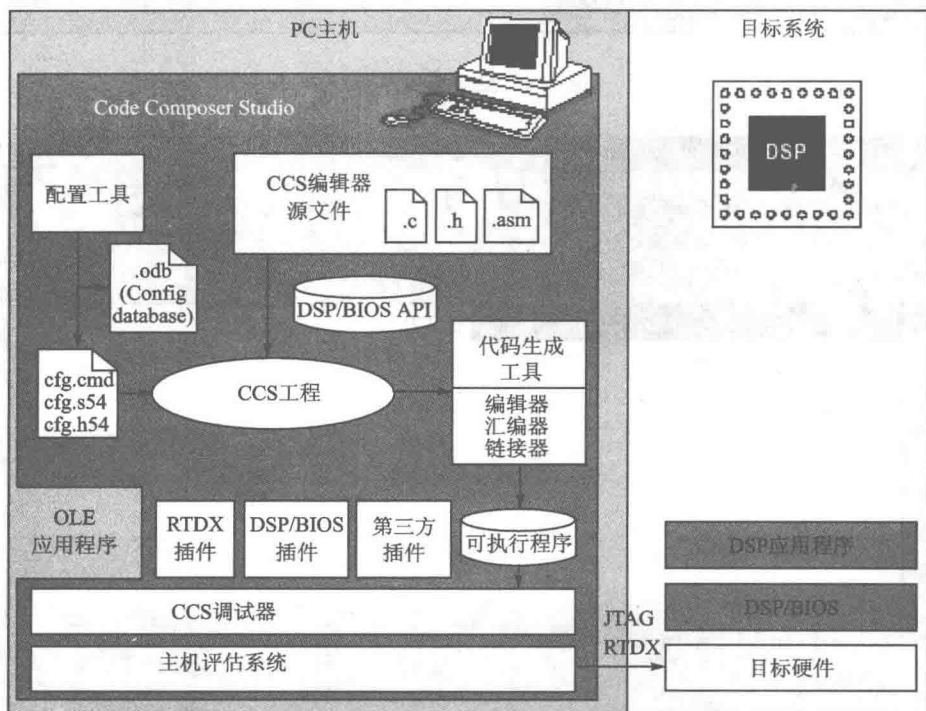


图 2.1 CCS2 组件的工作机理

### 2.2 CCS2 代码生成工具

代码生成工具是 CCS2 集成开发环境的基础，它的作用是将 C 语言、汇编语言或两者的混合语言编写的 DSP 源代码程序编译、汇编并链接成可执行的 DSP 代码。代码生成工具的工作流程如图 2.2 所示，其中 C 语言程序的最常用的开发路径用阴影显示，其他部分是加强开发过程中的附加功能。