

DSP应用丛书

# DSP集成开发环境

——CCS及DSP/BIOS的原理与应用

彭启琮 管庆 等编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

DSP 应用丛书

# DSP集成开发环境

——CCS及DSP/BIOS的原理与应用

彭启琮 管 庆 等编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

CCS 是 TI 开发的一个完整的 DSP 集成开发环境,也是目前使用得最为广泛的 DSP 开发软件之一。

本书详细地介绍了 CCS 中各种开发工具的使用,特别是对 DSP/BIOS 的应用做了较为详细的介绍。本书前半部分从 CCS 的基本使用开始,由浅入深地阐述了使用 CCS 开发环境完成 DSP 项目的建立、编辑、编译并最终完成调试的过程;接着,对 DSP/BIOS 的应用做了详细说明;本书的最后部分介绍了片级支持库(CSL)的使用。为了便于读者尽快掌握 CCS 以及 DSP/BIOS 的应用,本书还提供了大量的应用例子及其配套的完整程度代码。这些例子程序都具有相当的代表性,并通过实际运行验证。

本书可以作为广大 DSP 应用工程师、高年级本科生和研究生的学习教材,也可以作为 DSP 开发人员的重要参考手册。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

### 图书在版编目(CIP)数据

DSP 集成开发环境: CCS 及 DSP/BIOS 的原理与应用/彭启琮,管庆等编著. —北京:电子工业出版社,2004.7  
(DSP 应用丛书)

ISBN 7-121-00064-4

I. D… II. ①彭… ②管… III. 软件工具, CCS IV. TP311.56

中国版本图书馆 CIP 数据核字(2004)第 062789 号

责任编辑:雷洪勤

印 刷:北京东光印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:25.5 字数:653 千字

印 次:2004 年 7 月第 1 次印刷

印 数:5000 册 定价:39.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。  
联系电话:(010)68279077。质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

# 前 言

随着 DSP 应用技术的迅速发展, 学习 DSP 课程的学生和利用 DSP 进行开发的工程师越来越多。

以 DSP 为核心的系统包括硬件系统和软件系统, 随着 DSP 应用技术的深入发展, 系统也日益复杂, 对开发环境的依赖也就越来越大。它迫切需要高效而方便的开发环境, 也包括阅读方便、对开发工作具有指导意义的文档与指南。这就是我们编写本书的目的和动力。

CCS (Code Composer Studio) 是 TI 开发的一个完整的 DSP 集成开发环境, 也是目前使用最为广泛的 DSP 开发软件之一。现在, 所有的 TI DSP 都可以使用该软件工具来进行开发。在 CCS 中, 不仅集成了常规的开发工具, 如源程序编辑器、代码生成工具 (编译、连接器) 及调试环境, 还提供了 DSP/BIOS™ 开发工具。DSP/BIOS™ 是一个简易的嵌入式操作系统, 它大大方便了用户编写多任务应用程序。使用 DSP/BIOS™ 后, 还能增强对代码执行效率的监控。目前, CCS 已经成为 DSP 开发过程中不可缺少的工具。

参加本书编著的几位老师, 这几年来一直都在从事各种 DSP 应用项目的开发, 对 CCS 的使用有比较深刻的理解。本书围绕一二个具体的工程应用实例, 由浅入深、从简到繁、详细地介绍了 CCS 中各种工具的使用, 以及我们自己在实践中体会到的各种使用技巧。相信在读完本书后, 读者能够运用 CCS 轻松地完成 DSP 项目的开发。同时, 由于针对 TI DSP 各系列的 CCS 有所差别, 本书力求抛开具体的系列与型号, 更多地讨论共性的东西。但是, 为了讨论的方便, 仍以 CCS FOR C5000 V2.1 为例来介绍 CCS 的使用。这样, 既不失一般性, 也具有一定的针对性, 同时满足各个系列 DSP 的 CCS 使用需要。

全书分为 4 个部分, 共 7 章。第一部分包括前言和第 1 章, 主要讲述 CCS 的基本特点和操作技巧, 包括安装、配置、操作界面、菜单以及各种帮助文档的使用等。

第二部分包括第 2, 3, 4 章, 主要讲述 CCS 提供的常规开发工具的使用, 包括应用程序的建立、编译、连接、调试以及代码效率分析等。

第三部分包括第 5 章和第 6 章, 主要讲述 DSP/BIOS 工具的使用, 包括如何利用 DSP/BIOS 提供的 API 函数编写应用程序。

第四部分是本书的最后一章, 即第 7 章, 主要讲述对 DSP 外设资源进行编程控制的工具, 介绍 CSL 函数库的使用。

本书的读者对象是学习 DSP 技术课程的研究生和高年级本科生, 以及从事 DSP 项目开发的工程技术人员。希望他们通过阅读本书, 并通过实验和工程实践, 能够很好地掌握 CCS 的使用。另外, 我们将书中使用的所有例子程序的完整代码都放在银杏科技的网站 (<http://www.dpsolution.com>) 上, 供读者下载使用。

本书第 1, 2 章由徐胜编写, 第 3, 4 章由向超编写, 第 5, 6 章由管庆编写, 第 7 章由胡全编写。电子科技大学得州仪器 (TI) DSP 技术/培训中心的研究生霍杰、涂灏、张健、卓为参加了部分资料的翻译工作。

在本书的编写过程中, 得到得州仪器 (中国) 公司的大力支持, 在此一并表示衷心的感谢。

由于作者水平有限, 书中错误之处在所难免, 望广大读者批评指正。

编著者于电子科技大学

2004 年 6 月

# 目 录

第 1 章 CCS 的功能特点与安装	(1)
1.1 CCS 功能简介	(1)
1.2 为 CCS 安装设备驱动程序	(2)
1.2.1 安装 C5409 软仿真型设备	(2)
1.2.2 仿真设备的快速选择	(3)
1.2.3 系统配置的导入和导出	(3)
1.2.4 删除已安装的仿真设备	(4)
1.2.5 安装第三方公司提供的仿真设备	(4)
第 2 章 基于 CCS 开发 DSP 软件	(8)
2.1 CCS 的基本操作——开发一个简单的程序	(8)
2.1.1 准备工作	(9)
2.1.2 新建工程文件	(9)
2.1.3 向工程添加各类型文件	(10)
2.1.4 查阅代码	(11)
2.1.5 建造和运行程序	(12)
2.1.6 改变建造选项和修改语法错误	(13)
2.1.7 使用断点与观察窗	(14)
2.2 工程的高级管理	(16)
2.2.1 准备工作	(16)
2.2.2 生成库工程	(18)
2.2.3 生成可执行工程	(18)
2.2.4 建造文件与工程的各种选项	(20)
2.2.5 初始建造步骤和最终建造步骤	(21)
2.2.6 工程配置	(23)
2.2.7 设置连接顺序	(24)
2.2.8 编辑工程文件	(25)
2.3 高级编辑技术	(27)
2.3.1 文件书签	(27)
2.3.2 列编辑	(29)
2.3.3 使用外部编辑器	(30)
2.3.4 扩展高亮显示的关键字	(30)
2.3.5 CodeMaestro 工具	(31)
2.3.6 选择边缘的使用	(33)

2.4 GEL 语言的运用 .....	( 34 )
2.4.1 GEL 语法 .....	( 35 )
2.4.2 编辑一个简单的 GEL 应用程序 .....	( 41 )
2.4.3 定义 GEL 的局部变量 .....	( 42 )
2.4.4 使用 GEL 实现批处理任务 .....	( 42 )
2.4.5 使用 GEL 控制 DSP 变量 .....	( 44 )
2.5 GEL 函数介绍 .....	( 46 )
<b>第 3 章 利用 CCS 调试应用程序 .....</b>	<b>( 69 )</b>
3.1 装入并运行应用程序 .....	( 69 )
3.2 存储器/变量的查看与修改 .....	( 72 )
3.2.1 寄存器的查看与修改 .....	( 73 )
3.2.2 存储器的查看与修改 .....	( 74 )
3.3 断点工具的使用 .....	( 78 )
3.3.1 软件断点 .....	( 78 )
3.3.2 硬件断点的使用 .....	( 80 )
3.3.3 存储器访问断点 .....	( 80 )
3.4 探针点工具的使用 .....	( 80 )
3.5 图形工具的使用 .....	( 83 )
3.5.1 时域波形与频谱 .....	( 84 )
3.5.2 眼图 .....	( 90 )
3.5.3 星座图 .....	( 91 )
3.5.4 图像显示 .....	( 95 )
3.6 程序剖析工具 .....	( 97 )
3.6.1 剖析时钟及其设置 .....	( 97 )
3.6.2 剖析设置 .....	( 98 )
3.6.3 剖析函数 .....	( 99 )
3.6.4 剖析范围 .....	( 100 )
3.6.5 剖析不连续范围 .....	( 101 )
3.6.6 剖析时钟的精确性 .....	( 102 )
3.6.7 剖析需要注意的问题 .....	( 102 )
3.6.8 剖析策略 .....	( 103 )
<b>第 4 章 CCS 高级工具的使用 .....</b>	<b>( 104 )</b>
4.1 数据转换器件支持插件 .....	( 104 )
4.1.1 支持的器件 .....	( 104 )
4.1.2 TLV320AIC10 的说明 .....	( 105 )
4.1.3 使用数据转换器件支持插件 .....	( 105 )
4.1.4 生成的文件 .....	( 110 )
4.1.5 使用生成的文件 .....	( 139 )
4.1.6 结论 .....	( 140 )

4.2	硬件仿真器分析工具	(140)
4.2.1	分析模块的主要功能	(140)
4.2.2	使用分析模块	(142)
4.3	符号浏览器	(147)
4.4	可视化连接器的使用	(147)
4.4.1	简介	(148)
4.4.2	使用可视化连接器	(148)
4.4.3	结论	(157)
4.5	端口与引脚仿真	(157)
4.5.1	端口仿真	(157)
4.5.2	引脚仿真	(158)
4.6	命令窗口	(159)
4.7	算法标准模板代码生成器	(160)
4.7.1	算法标准简介	(160)
4.7.2	模板代码生成器	(162)
4.8	实时数据交换 (RTDX)	(163)
4.8.1	RTDX 简介	(163)
4.8.2	在目标应用程序中使用 RTDX	(165)
4.8.3	主机客户程序	(168)
4.8.4	RTDX 配置	(173)
4.8.5	多处理器 RTDX	(174)
4.8.6	结论	(175)
<b>第 5 章</b>	<b>DSP/BIOS™ 的功能</b>	<b>(176)</b>
5.1	DSP/BIOS 功能介绍	(176)
5.2	建立 DSP/BIOS 的配置文件	(177)
5.3	用 DSP/BIOS 工具创建应用程序	(182)
5.4	DSP/BIOS 系统工具的使用	(185)
5.4.1	DSP/BIOS 工具控制面板	(185)
5.4.2	内核/模块查看窗口	(186)
5.4.3	CPU 负荷图	(187)
5.4.4	程序模块执行状态图	(187)
5.4.5	主机通道控制	(189)
5.4.6	信息显示窗口	(190)
5.4.7	状态统计窗口	(191)
5.5	基础类的 DSP/BIOS API 调用	(192)
5.5.1	基础类的模块: 时钟管理 CLK	(192)
5.5.2	基础类的模块: 周期函数 PRD	(200)
5.5.3	基础类的模块: 软件中断管理 SWI	(205)
5.5.4	其他基础类的模块: 信息输出管理 LOG	(217)



5.5.5	其他基础类的模块：存储器管理 MEM	(223)
5.5.6	其他基础类的模块：全局设置 GBL	(232)
5.5.7	其他基础类的模块：C5000 系列专用模块 C54x	(234)
5.5.8	其他基础类的模块：C6000 系列专用模块 C62x 和 C64x	(236)
5.5.9	通信类模块：实时数据交换 (RTDX)	(237)
5.5.10	统计类模块：STS 模块	(243)
5.5.11	统计类模块：TRC 模块	(248)
5.6	DSP/BIOS 的任务管理	(248)
5.6.1	任务管理：TSK 任务管理模块	(249)
5.6.2	任务管理：HOOK 钩子函数管理模块	(264)
5.7	DSP/BIOS 的后台管理：IDL 模块	(266)
5.7.1	模块描述	(266)
5.7.2	后台 IDL 模块属性参数	(267)
5.7.3	IDL 对象的属性参数	(267)
5.7.4	IDL 模块提供的 API 函数	(268)
5.8	DSP/BIOS 的应用例子	(268)
<b>第 6 章</b>	<b>DSP/BIOS 的高级应用</b>	<b>(274)</b>
6.1	DSP/BIOS 中的线程	(274)
6.1.1	选择线程类型	(274)
6.1.2	线程选择的一些原则	(276)
6.1.3	SWI 和 PRD 线程使用的例子	(279)
6.1.4	任务线程使用的例子	(289)
6.2	任务的通信和同步	(293)
6.2.1	MBX 邮箱管理模块	(293)
6.2.2	SEM 旗语管理模块	(296)
6.2.3	QUE 队列管理模块	(300)
6.2.4	任务之间的共享资源与同步	(306)
6.3	硬件中断管理	(317)
6.3.1	配置中断管理模块	(317)
6.3.2	在 DSP/BIOS 中使用中断的实例	(325)
6.4	DSP/BIOS 中的数据交换	(331)
6.4.1	数据管道管理 (PIP 模块)	(331)
6.4.2	PIP 管道模块的属性参数	(336)
6.4.3	PIP 模块的 API 函数	(337)
6.4.4	使用 PIPE 的实例	(343)
6.4.5	HST 主机通道管理 (HST 模块)	(347)
6.4.6	使用 HST 主机通道管理的实例	(351)
6.5	对 DSP/BIOS 内核的评估	(354)
6.5.1	估算 DSP/BIOS 内核的开销	(354)

6.5.2	DSP/BIOS 内核中的主要 API 函数调用的开销评估	( 355 )
6.5.3	DSP/BIOS 内核中的主要 API 函数开销表	( 360 )
6.5.4	测量中断的延迟	( 360 )
6.5.5	DSP/BIOS 监控工具对软件实时性的影响	( 361 )
6.5.6	设置 DSP 目标系统与 CCS 之间的数据传输频率	( 362 )
6.6	利用配置工具对 DSP/BIOS 进行优化	( 362 )
6.6.1	对 DSP/BIOS 后台 IDL 循环的优化	( 363 )
6.6.2	关闭 DSP/BIOS 的任务管理功能	( 363 )
6.6.3	禁止使用动态堆	( 364 )
6.6.4	禁止 CLK 时钟管理	( 364 )
6.6.5	禁止实时数据交换 ( RTDX ) 功能	( 364 )
6.6.6	关闭实时分析功能	( 365 )
6.6.7	去掉 CSL 片级支持库	( 365 )
6.6.8	去除系统处理函数	( 366 )
6.6.9	最小化数据存储器	( 367 )
6.6.10	选择静态或动态对象创建	( 367 )
<b>第 7 章</b>	<b>基于 CCS 的 DSP 片级支持库</b>	<b>( 368 )</b>
7.1	CSL 概要	( 368 )
7.1.1	CSL 框架结构	( 368 )
7.1.2	CSL 命名规则	( 370 )
7.1.3	CSL 数据类型和符号常量	( 370 )
7.1.4	CSL 函数和宏	( 371 )
7.2	CSL 的使用	( 372 )
7.2.1	CSL 的 GUI 调用	( 372 )
7.2.2	CSL 库的直接调用	( 378 )
7.3	CSL 模块功能	( 385 )
7.3.1	CHIP 模块	( 385 )
7.3.2	DAT 模块	( 385 )
7.3.3	DMA 模块	( 386 )
7.3.4	EBUS 模块	( 388 )
7.3.5	GPIO 模块	( 389 )
7.3.6	IRQ 模块	( 389 )
7.3.7	MCBSP 模块	( 391 )
7.3.8	PLL 模块	( 392 )
7.3.9	PWR 模块	( 393 )
7.3.10	TIMER 模块	( 394 )
7.3.11	UART 模块	( 394 )
7.3.12	WDTIM 模块	( 395 )
<b>参考文献</b>		<b>( 397 )</b>

# 第 1 章 CCS 的功能特点与安装

## 1.1 CCS 功能简介

Texas Instruments 公司的 DSP 集成开发环境 CCS (Code Composer Studio), 是一个基于 Windows 的 DSP 开发平台, 可以加速和提高程序员创建和测试实时嵌入式信号处理系统的开发过程, 从而缩短将产品推向市场所需要的时间。

CCS 是一个完整的 DSP 集成开发环境, 也是目前最优秀、最流行的 DSP 开发软件之一。CCS 最早是由 GO DSP 公司为 TI 的 C6000 系列开发的, 后来 TI 收购了 GO DSP, 并将 CCS 扩展到其他系列。现在所有的 TI DSP 都可以使用该软件工具进行开发, 并为 C2000 (版本 2.2 以上)、C5000 和 C6000 系列 DSP 提供 DSP/BIOS 功能, 而在 C3X 中是没有 DSP/BIOS 功能的。所以有时也将用于 C3X 开发的集成开发环境称为 CC (Code Composer), 以示区别。

CCS 主要包含了以下功能:

- 集成可视化代码编辑界面, 可直接编写 C、汇编、.H 文件、.cmd 文件等。
- 集成代码生成工具, 包括汇编器、优化 C 编译器、连接器等。
- 基本调试工具, 如装入执行代码 (.out 文件), 查看寄存器、存储器、反汇编、变量窗口等, 支持 C 源代码级调试。
- 支持多 DSP 调试。
- 断点工具, 包括硬件断点、数据空间读/写断点, 条件断点 (使用 GEL 编写表达式) 等。
- 探针工具 (Probe Points), 可用于算法仿真、数据监视等。
- 分析工具 (Profile Points), 可用于评估代码执行的时钟数。
- 数据的图形显示工具, 可绘制时域/频域波形、眼图、星座图、图像等, 并可自动刷新 (使用 Animate 命令运行)。
- 提供 GEL 工具, 用户可以编写自己的控制面板/菜单, 方便直观地修改变量, 配置参数等。
- 支持 RTDX (Real Time Data Exchange) 技术, 可在不中断目标系统运行的情况下, 实现 DSP 与其他应用程序 (OLE) 的数据交换。
- 开放式的 Plug-in 技术, 支持其他第三方的 ActiveX 插件, 支持包括软仿真在内的各种仿真器 (只需安装相应的驱动程序)。
- 提供 DSP/BIOS 工具, 增强对代码的实时分析能力 (如分析代码执行的效率)、调度程序执行的优先级、方便管理或使用系统资源 (代码/数据占用空间、中断服务程序的调用、定时器使用等), 从而减少了开发人员对硬件资源熟悉程度的依赖性。

可见, CCS 具有实时、多任务、可视化的软件开发特点, 已经成为 TI DSP 家族的程序设计、制作、调试、优化的利器。本书对 CCS 的基本概念、多处理功能、断点和探测点、

文件输入/输出功能、可视化窗口、存储器图、观察窗口、集成编辑器、项目环境、代码执行评估、通用扩展语言 (GEL)、DSP/BIOS 技术和 RTDX 技术做了详细分析。使读者能够快速地使用 CCS 开发出优秀的 TI DSP 代码。

## 1.2 为 CCS 安装设备驱动程序

在 Windows 操作系统中成功安装完成 CCS 后，桌面上会出现两个快捷方式图标，如图 1-2-1 所示。

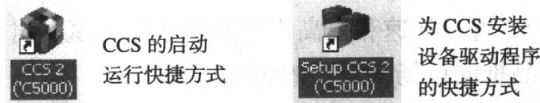


图 1-2-1 CCS 的两个图标

在使用 CCS 进行 DSP 软件开发之前，读者需要建立一个概念，即 CCS 是运行在一系列仿真设备之上的一个集成的开发环境，这一系列仿真设备包括软仿真器 (Simulator)、各种硬仿真器 (Emulator)、TI 或第三方公司提供的 DSP 入门套件 (DSK)、TI 或第三方公司提供的 DSP 评估板 (EVM) 等。任一种仿真设备都可形象地将其看做计算机主板上的扩展设备，在它们正常工作之前都需要在操作系统中为其安装驱动程序。下面将介绍为 CCS 的仿真设备安装驱动程序的方式，读者可按要求任意选择。

### 1.2.1 安装 C5409 软仿真型设备

双击如图 1-2-1 所示的快捷方式“Setup CCS 2 (C5000)”，运行仿真设备的安装程序，界面如图 1-2-2 所示。

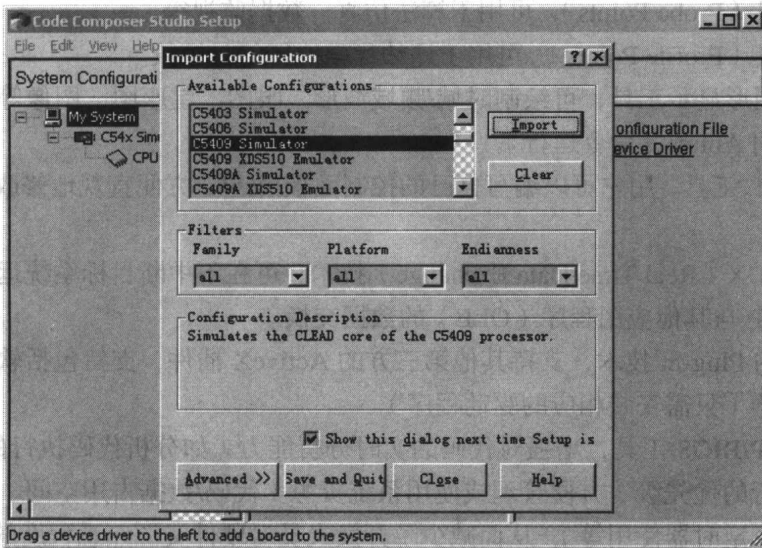


图 1-2-2 仿真设备的安装程序界面

CCS 安装完成之后，会在系统中默认安装 C55x 的软仿真设备 (C55x Simulator)。在

CCS 弹出的“Import Configuration”对话框中，用户可在“Available Configurations”区域中单击“Clear”按钮，即可清除 CCS 的默认配置，然后滑动滚动条选择所需要的仿真设备，如 C5409 软仿真设备（C5409 Simulator），单击“Import”按钮，即可将其添加到系统配置中（System Configuration）。单击“Save and Quit”按钮，安装程序就会提示是否退出后启动 CCS，如图 1-2-3 所示，单击“是”按钮，即可成功启动 CCS 在 C5409 软仿真设备下工作。

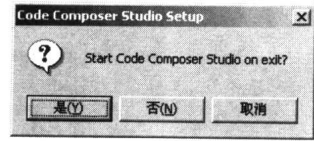


图 1-2-3 安装程序启动 CCS 的提示

### 1.2.2 仿真设备的快速选择

对 CCS 进行软件升级后，安装程序可能会在“Import Configuration”对话框中提供大量的仿真设备，而不易找到我们需要的仿真设备型号，这时可利用该对话框中“Filters”区域所提供的功能来快速选择。

例如要安装 TI 的 TMS320VC5402 DSK 的驱动程序，单击“Family”下拉菜单，选择“c54x”，再单击“Platform”选择“DSK”，在“Available Configurations”区域就只会出现该 DSK 的驱动程序，单击“Import”按钮即可，如图 1-2-4 所示。单击“Save and Quit”按钮，安装程序提示是否退出后启动 CCS，如图 1-2-3 所示，单击“是”按钮，启动 CCS 在 TMS320 VC5402 DSK 设备下工作。

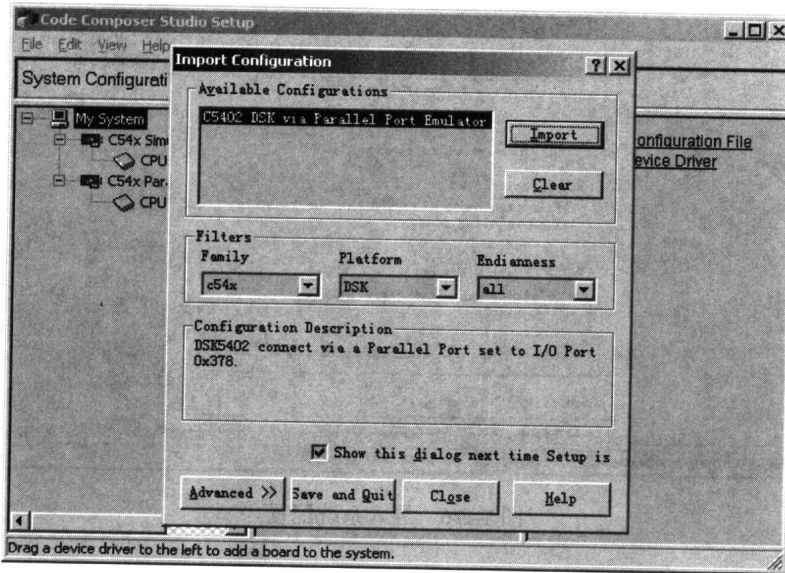


图 1-2-4 快速选择仿真设备的驱动程序

### 1.2.3 系统配置的导入和导出

当我们为系统添加某种仿真器设备 A 后，而在其他项目开发中可能又会使用其他类型的仿真器 B。当我们再次需要使用仿真器设备 A 时，使用系统配置导入和导出功能，就不用再重新设置。

例如，在 1.2.1 小节和 1.2.2 小节中，我们添加了 C5409 软仿真设备和 TMS320VC5402

DSK 仿真设备，可在主界面的“System Configuration”栏中看到上述步骤添加的两种仿真设备。选择“File”|“Export”命令，单击“保存”按钮，将当前配置导出在 CCS 安装目录“\ti\drivers\import”下，存储为“My\_setup.ccs”，如图 1-2-5 所示。

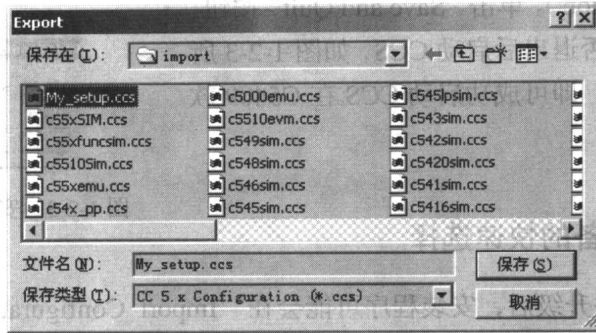


图 1-2-5 导出系统配置

如果需要恢复以前的系统配置，选择“File”|“Import”命令，打开“Import Configuration”对话框，单击“Advanced”按钮，弹出“Import”对话框，单击“Browse”按钮，选择曾经保存的“My\_setup.ccs”文件即可恢复，如图 1-2-6 所示。

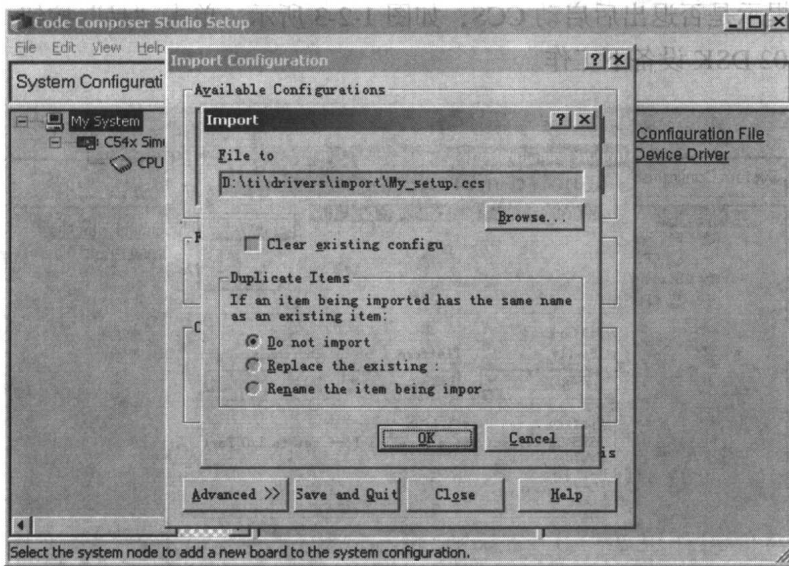


图 1-2-6 导入系统配置

## 1.2.4 删除已安装的仿真设备

如果需要删除已安装的仿真设备，在仿真设备名上单击鼠标右键，选择“Remove”命令即可，如图 1-2-7 所示。

## 1.2.5 安装第三方公司提供的仿真设备

前几节均是添加 CCS 提供的标准仿真设备，当使用第三方公司提供的仿真设备时，如

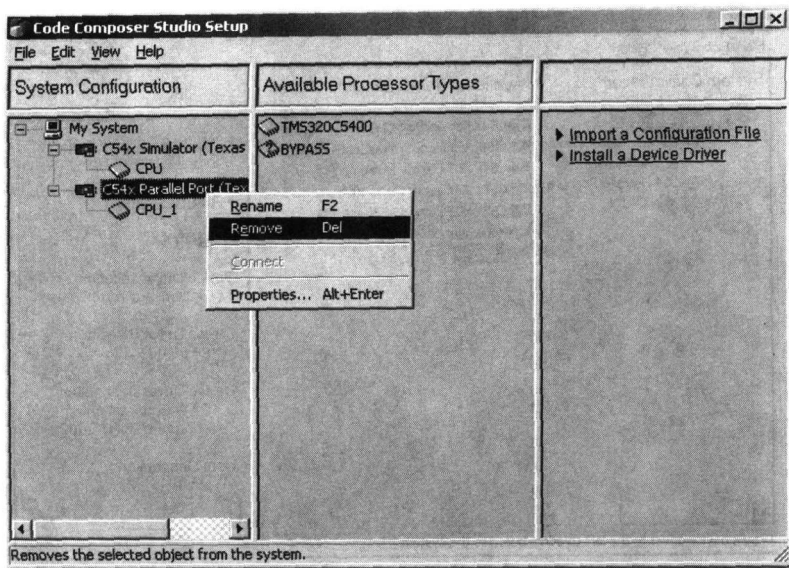


图 1-2-7 删除已安装的仿真设备

使用 Spectrum Digital 公司型号为“XDS510PP PLUS”并口型硬仿真器调试 DSP 时, 将该公司提供的驱动程序安装在 CCS 的安装目录之后, 还需要在 CCS 的仿真设备安装程序中添加“有效的板卡/仿真器类型 ( Available Board/Simulator Types)”, 具体步骤如下。

(1) 在仿真设备安装程序的最右栏单击“Install a Device Driver”, 弹出“Select Device Driver File”对话框, 在 CCS 的安装目录下的 drivers 文件夹中找到 sdgo5xx.dvr 文件, 将其打开即可, 如图 1-2-8 所示。

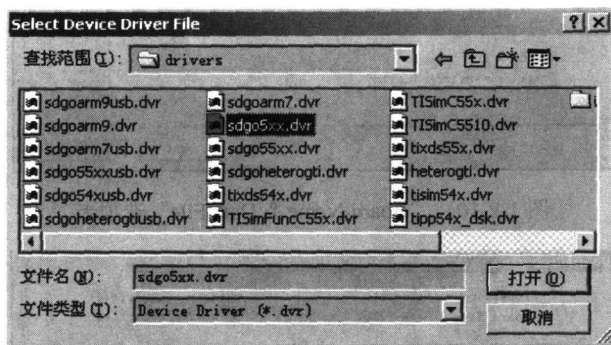


图 1-2-8 安装“XDS510PP PLUS”硬仿真器驱动程序

(2) 在“Available Board/Simulator Types”栏中可看到刚添加的板卡“sdgo5xx”, 单击该板卡, 可在“Install a Device Driver”栏中看到其简介, 然后单击“Add To System”, 如图 1-2-9 所示。

(3) 在弹出的“Board Properties”对话框中, 在“Board Name & Data File”选项卡中, 在“Board”文本框中输入“My sdgo5xx”, 如图 1-2-10 所示。

(4) 单击“Next>”按钮, 弹出“Board Properties”标签, 将“I/O Port”的值改为计算机并口地址“0x378”, 如图 1-2-11 所示。注意该选项因使用的仿真器不同可能会有所区别。用户可以参考所使用的仿真器的安装说明书。

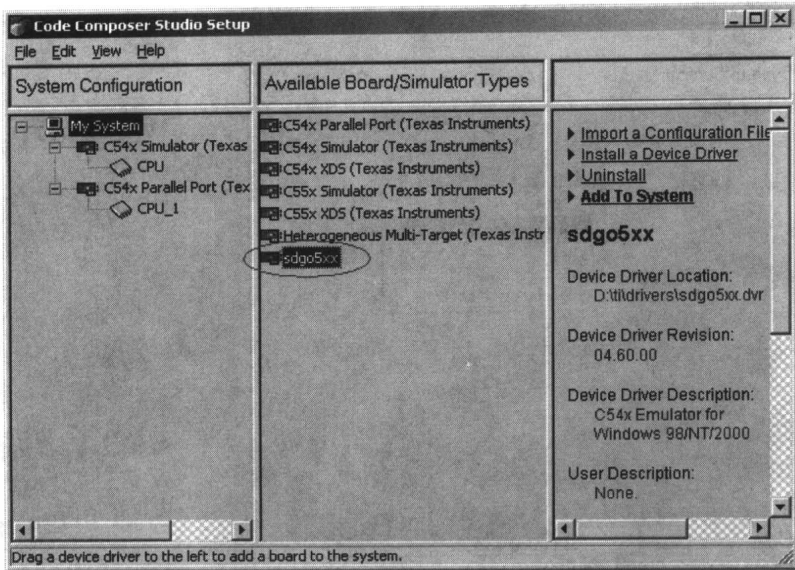


图 1-2-9 添加“XDS510PP PLUS”硬仿真器驱动程序到系统

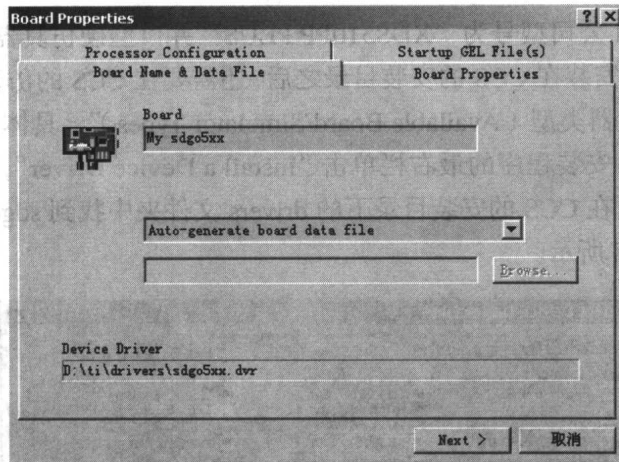


图 1-2-10 “Board Name & Data File”选项卡

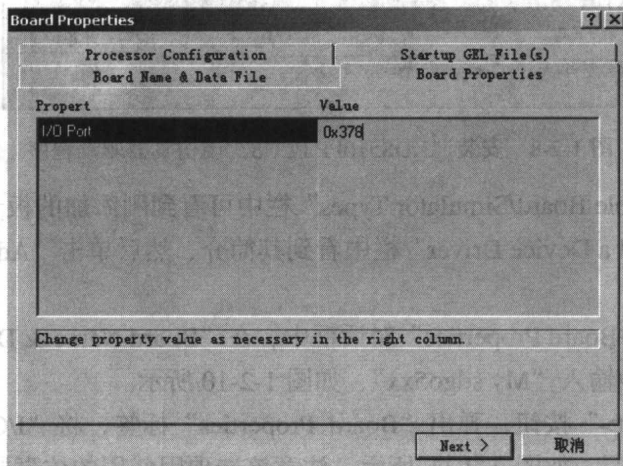


图 1-2-11 “Board Properties”选项卡



(5) 单击“Next>”按钮，弹出“Processor Configuration”选项卡，单击正中间的“Add Single”按钮，结果如图 1-2-12 所示。

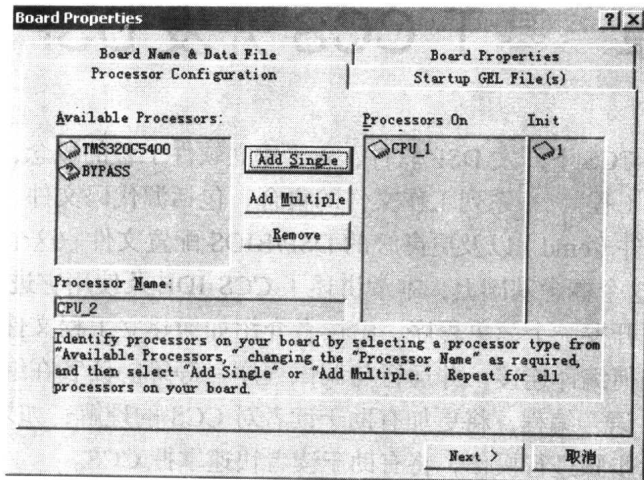


图 1-2-12 “Processor Configuration”选项卡

(6) 单击“Next>”按钮，弹出“Startup GEL File(s)”选项卡。GEL 即通用扩展语言 (General Extension Language)，它是一种解释性语言，类似于 C 语言，读者可以使用 GEL 来编写函数，从而扩展 CCS 的功能。有关 GEL 语言的详细情况我们将在后面介绍。这里应该根据所调试或仿真的具体 DSP 型号，选择相应的 GEL 启动配置文件，以便在 CCS 进入后控制 DSP 的各个状态。例如，我们要调试的目标板使用的是 VC5416，则单击浏览按钮，选择所需要的 GEL 文件 c5416.gel，结果如图 1-2-13 所示。

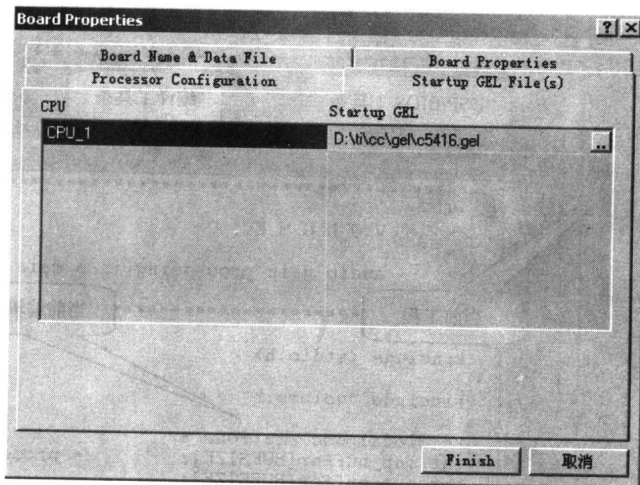


图 1-2-13 “Startup GEL File(s)”对话框

(7) 单击“Finish”按钮，完成板卡属性配置，选择“File”|“Save”命令，存储当前配置。至此，使用 CCS 进行 DSP 软件开发之前的准备工作，即告一个段落。