

高等学校电子与通信类专业“十一五”规划教材

# DSP 原理与应用实验

主 编 姜 阳 周锡青

副主编 张启东 崔旭晶

主 审 张雪英

西安电子科技大学出版社

2008

## 内 容 简 介

本书根据高等院校工科本科生“DSP 原理与应用”和“DSP 技术”等课程的基本要求编写,主要介绍了以美国 TI(Texas Instruments)公司 TMS320VC55x 系列芯片为核心的 DSP 实验技术。本书从 CCS 入门实验开始,安排了软件仿真实验(如正弦波产生、C 语言和汇编语言混合编程、FIR、IIRFFT 等)、硬件仿真实验(如硬件仿真设置、定时器及硬件中断和外围高性能立体声音频编解码芯片实验等)以及实时操作系统 DSP/BIOS 和数字图像处理仿真实验。书中各实验所使用的程序都是精心编写并经过认真调试运行的。本书最后增加了一款 TMS320VC5501 芯片组成的最小系统硬件应用实例,并在附录中给出了完整的硬件电路原理图和部分软件程序,使读者能够初步了解和掌握 C55xx 芯片 DSP 技术的设计、开发和仿真调试过程,具有一定的实用价值。

本书适用于高等学校电子信息工程、通信工程、电子科学与技术及自动化等电信类专业的本科生,同时也可供相关技术人员参考。

★本书配有电子教案,需要者可登录出版社网站,免费下载。

### 图书在版编目(CIP)数据

DSP 原理与应用实验 / 姜阳, 周锡青主编. —西安: 西安电子科技大学出版社, 2008.9

高等学校电子与通信类专业“十一五”规划教材

ISBN 978-7-5606-2109-8

I. D… II. ①姜… ②周… III. 数字信号—信号处理—高等学校—教材 IV. TN911.72

中国版本图书馆 CIP 数据核字(2008)第 118273 号

策 划 曹 昶

责任编辑 曹 昶

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfb001@163.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2008 年 9 月第 1 版 2008 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 12.625

字 数 289 千字

印 数 1~4000 册

定 价 18.00 元

ISBN 978-7-5606-2109-8/TN·0455

**XDUP 2401001-1**

\*\*\*如有印装问题可调换\*\*\*

本社图书封面为激光防伪覆膜,谨防盗版。

# 前 言

DSP(Digital Signal Processor)芯片是一种高性能的微处理器,其技术发展大大地推动了数字信号处理技术的发展与应用,目前从工业系统到家电产品,从军事装备到生物医学仪器,无不融入了 DSP 技术。可以预料,随着 DSP 芯片向着运算速度更高、功能更强、制造成本更低、体积更小、重量更轻的方向发展,以 DSP 技术为核心的新产品和新技术将更加广泛和普及。

目前,DSP 芯片主要供应商有美国 TI(Texas Instruments)公司、AD 公司、AT&T 公司和 Motorola 公司等,其中 TI 公司约占世界 DSP 芯片销售总量的 50%,其技术支持雄厚,在我国被广泛采用。TI 公司 TMS320C5000 系列中 C55xx 芯片是性价比最高的产品之一,其主要特点是高性能、超低功耗和低价位。它与 C54xx 芯片相比,相同价格的芯片其内核性能可提高五倍,而功耗仅为 C54xx 芯片的六分之一。

在我国, TI 公司虽然提供了很多用户手册及其技术支持,但是读懂用户手册并与 TI 公司进行技术交流的技术人员却为数不多,而熟练掌握 DSP 技术的人员则更少,这极大地限制了 DSP 技术在我国的应用与发展。目前,在已出版的高校 DSP 技术教材中,大多数只停留在基础理论、算法和硬件结构特点上,很少有适用的实验教材与之配套。在 TMS320C5000 系列中,介绍 C54xx 芯片方面的较多,而 C55xx 芯片方面的却很少。为了加快我国信息化发展速度,为培养信息产业专业人才奠定良好的应用基础,我们编写了这本 C55xx 芯片方面的实验教材。

本书的目的是让读者尽快地了解 and 掌握 C55xx 芯片 DSP 的设计、仿真与开发过程,特点是由浅入深、图文并茂及方便快捷。书中使用 C 语言和汇编语言编程,软件仿真和硬件仿真兼备,内容全面充实。该书作为“DSP 原理与应用”或“DSP 技术”课程理论教学的配套实验教材,要求理论课讲授 C 语言、TMS320VC55x 的硬件结构、外围接口及其汇编语言等。全书共 11 章,约需 24 学时实验,各校可根据理论课内容作适当选用。前 10 章中的每个实验都结合参考程序给出了具体实验方法和步骤,这些程序都是经过精心编写并认真调试和运行才得出的结果。最后一章是应用一款 TMS320VC5501 芯片组成最小系统的设计实例,且在附录中给出了系统的部分软件程序和完整的硬件电路原理图。

本书由姜阳负责组织全书的编写、统稿和完善工作,并编写了第 1 章、第 3 章和第 9 章;周锡青负责全书的程序编制与调试工作,并编写了第 2 章、第 4 章、第 5 章和第 6 章;张启东编写了第 7 章和第 10 章;崔旭晶编写了第 8 章,并完成全书文字校对工作;杨子江编写了第 11 章和附录。

本书在编写过程中得到了沈阳理工大学信息科学与工程学院领导的大力支持,在此对他们表示衷心感谢。主审张雪英教授为本书提出了宝贵的修改意见,使本书更加完善,在此表示衷心感谢。此外,还要感谢 TI 公司中国技术支持部的 Shine Zhang 先生和 TI 公司软

件部的 Amaresk Kumar 先生，他们多次解答了调试硬件和程序中遇到的各种问题；感谢贾书涛先生无偿地帮助焊接实验主板上的贴片元件。

由于编者水平有限，加之时间仓促，书中难免有不当之处，敬请广大读者批评指正。

编 者

2008 年 6 月于沈阳理工大学

# 目 录

第 1 章 CCS 集成环境的认识与使用.....	1
1.1 CCS 集成开发环境的认识.....	1
1.1.1 CCS 集成开发环境的构成.....	1
1.1.2 CCS 集成开发环境的主要功能.....	1
1.1.3 工具栏常用图标.....	2
1.2 CCS 集成开发环境的安装配置及使用实验.....	3
1.2.1 实验目的.....	3
1.2.2 实验设备.....	3
1.2.3 实验原理.....	3
1.2.4 CCS 应用程序的安装和配置.....	3
1.2.5 CCS 环境中工程文件的使用.....	5
1.2.6 编译、链接和运行目标文件.....	8
1.2.7 程序书写时应该注意的问题及程序示范.....	11
第 2 章 DSP 硬件开发系统的使用.....	13
2.1 硬件仿真器驱动软件的安装实验.....	13
2.1.1 实验目的.....	13
2.1.2 实验设备.....	13
2.1.3 实验要求.....	13
2.1.4 实验原理.....	13
2.1.5 仿真器驱动程序的安装.....	15
2.2 CCS v3.1 硬件仿真设置实验.....	17
2.2.1 实验目的.....	17
2.2.2 实验设备.....	18
2.2.3 实验要求.....	18
2.2.4 实验步骤.....	18
2.2.5 参考程序清单.....	22
第 3 章 正弦波信号产生.....	25
3.1 单频正弦波信号实验.....	25
3.1.1 实验目的.....	25
3.1.2 实验设备.....	25
3.1.3 实验要求.....	25

3.1.4	实验原理.....	25
3.1.5	实验步骤.....	28
3.1.6	参考程序清单.....	31
3.2	两种频率的正弦波叠加信号实验.....	36
3.2.1	实验目的.....	36
3.2.2	实验设备.....	36
3.2.3	实验要求.....	36
3.2.4	实验原理.....	36
3.2.5	实验步骤.....	36
3.2.6	参考程序清单.....	39
<b>第 4 章</b>	<b>定时器及硬件中断.....</b>	<b>45</b>
4.1	定时器硬件结构原理.....	45
4.2	定时器及硬件中断实验.....	49
4.2.1	实验目的.....	49
4.2.2	实验设备.....	49
4.2.3	实验要求.....	49
4.2.4	实验步骤.....	50
4.2.5	参考程序清单.....	51
<b>第 5 章</b>	<b>立体声芯片 TLV320AIC23 的应用.....</b>	<b>54</b>
5.1	TLV320AIC23 芯片介绍.....	54
5.1.1	TLV320AIC23 特点介绍.....	54
5.1.2	TLV320AIC23 芯片内部结构框图和引脚安排.....	55
5.1.3	TLV320AIC23 与 C5509 的硬件连接.....	57
5.2	TLV320AIC23 应用实验.....	58
5.2.1	实验目的.....	58
5.2.2	实验设备.....	58
5.2.3	实验要求.....	58
5.2.4	实验步骤.....	59
5.2.5	参考程序清单.....	61
<b>第 6 章</b>	<b>C 语言和汇编语言混合编程.....</b>	<b>72</b>
6.1	混合编程中参数传递和寄存器使用.....	72
6.1.1	C 语言和汇编语言之间名称转换.....	72
6.1.2	变量定义及编译模式.....	73
6.1.3	参数传递及返回值.....	73
6.1.4	寄存器的使用和保存.....	74
6.2	C55x DSP C 语言和汇编语言混合编程实验.....	75
6.2.1	实验目的.....	75
6.2.2	实验设备.....	75
6.2.3	实验内容.....	75

6.2.4	实验步骤	75
6.2.5	参考程序清单	76
<b>第 7 章</b>	<b>数字滤波器设计与实现</b>	<b>80</b>
7.1	有限冲激响应 FIR 滤波器实验	80
7.1.1	实验目的	80
7.1.2	实验设备	80
7.1.3	实验内容	80
7.1.4	实验原理	80
7.1.5	FIR 滤波器的设计方法	81
7.1.6	实验步骤	86
7.1.7	参考程序清单	88
7.2	无限冲激响应 IIR 滤波器实验	91
7.2.1	实验目的	91
7.2.2	实验设备	92
7.2.3	实验内容	92
7.2.4	实验原理	92
7.2.5	IIR 滤波器设计	92
7.2.6	实验步骤	94
7.2.7	参考程序清单	96
<b>第 8 章</b>	<b>FFT 的 DSP 实现</b>	<b>100</b>
8.1	利用标准 C 语言实现 FFT 实验	100
8.1.1	实验目的	100
8.1.2	实验设备	100
8.1.3	实验要求	100
8.1.4	实验原理	100
8.1.5	实验步骤	103
8.1.6	FFT 的标准 C 语言参考程序清单	105
8.2	利用 DSP 库函数实现 FFT 实验	109
8.2.1	实验目的	109
8.2.2	实验设备	109
8.2.3	实验要求	109
8.2.4	实验原理	109
8.2.5	实验步骤	110
8.2.6	参考程序清单	112
<b>第 9 章</b>	<b>DSP/BIOS 开发工具的简单应用</b>	<b>116</b>
9.1	DSP/BIOS 配置工具认识实验	116
9.1.1	实验目的	116
9.1.2	实验设备	116
9.1.3	实验原理	116

9.1.4	实验步骤.....	121
9.1.5	参考程序清单及说明.....	123
9.2	DSP/BIOS 实时分析工具实验.....	123
9.2.1	实验目的.....	123
9.2.2	实验设备.....	124
9.2.3	实验原理.....	124
9.2.4	实验步骤.....	124
9.2.5	参考程序清单及说明.....	129
<b>第 10 章</b>	<b>数字图像处理初步实验.....</b>	<b>132</b>
10.1	数字图像直方图统计.....	132
10.1.1	实验目的.....	132
10.1.2	实验设备.....	132
10.1.3	实验内容.....	132
10.1.4	实验步骤.....	133
10.1.5	参考程序清单.....	135
10.2	数字图像直方图均衡化增强.....	137
10.2.1	实验目的.....	137
10.2.2	实验设备.....	137
10.2.3	实验内容.....	138
10.2.4	实验原理.....	138
10.2.5	实验步骤.....	140
10.2.6	参考程序清单.....	144
10.3	数字图像边缘检测.....	147
10.3.1	实验目的.....	147
10.3.2	实验设备.....	147
10.3.3	实验内容.....	147
10.3.4	实验原理.....	147
10.3.5	实验步骤.....	151
10.3.6	参考程序清单.....	153
<b>第 11 章</b>	<b>简易音频播放与处理系统的硬件设计实例.....</b>	<b>157</b>
11.1	系统硬件结构概述.....	157
11.1.1	系统硬件结构概述.....	157
11.1.2	SD/MMC 存储卡.....	159
11.1.3	S6B1713 点阵 LCD 显示器.....	161
11.1.4	用户图形界面的实现.....	162
11.2	TMS320VC5501 与 C8051F340 的连接及通信协议.....	163
11.2.1	TMS320C5501 的主要特点及内部结构.....	163
11.2.2	TMS320C5501 的主机接口(HPI).....	164
11.3	音频编解码器 TLV320AIC32.....	166

11.3.1 TLV320AIC32 纵览 .....	166
11.3.2 TLV320AIC32 芯片封装及引脚及其定义 .....	168
附录 A 简易音频播放与处理系统单片机部分的程序清单 .....	171
A.1 DSP&HPI 主机接口程序 .....	171
A.2 TASKS 多任务操作程序 .....	177
附录 B 简易音频播放与处理系统的硬件原理图 .....	183
参考文献 .....	189

## 第 1 章 CCS 集成环境的认识与使用

CCS(Code Composer Studio)是 TI 公司推出的用于开发 DSP 芯片的集成环境,它集编辑、编译、链接、软件仿真、硬件调试以及实时跟踪等功能于一体。对于 TI 公司早期版本的 CCS 只适用于一种系列的 DSP 芯片,例如 C5000 CCS 只适用于 5000 系列(如 C54xx 和 C55xx)的 DSP 芯片,而 Code Composer Studio IDE v3.1 版本(以下称 CCS v3.1)支持多种型号的 DSP 芯片及 ARM 单片机,其中包括 C24x、C24xx、C27xx、C28xx、C54xx、C55xx、C62xx、C64xx、C67xx、OMAP、ARM7、ARM9 和 ARM11 等。本章首先简单介绍 CCS 的组件构成和主要功能,然后介绍 CCS v3.1 的安装和配置,最后介绍如何使用 CCS 进行软件仿真。

### 1.1 CCS 集成开发环境的认识

#### 1.1.1 CCS 集成开发环境的构成

CCS 集成开发环境由以下几个组件构成:

- 集成代码产生工具。
- CCS 集成开发环境。
- DSP/BIOS 实时内核插件及其应用程序接口 API。
- 实时数据交换的 RTDX 插件以及相应的程序接口 API。
- 由 TI 公司以外第三方提供的各种应用模块插件。

这些组件用来对由 C 语言、汇编语言或混合语言编写的 DSP 源程序进行编译、汇编,并链接成可执行的 DSP 程序,主要包括汇编器、链接器、C/C++编译器和建库工具等。

#### 1.1.2 CCS 集成开发环境的主要功能










CCS 集成开发环境的功能十分强大,它集成了代码编辑、编译、链接、软件仿真、硬件调试和实时跟踪等诸多功能,包括编辑工具、工程管理工具和调试工具等,而且支持 C/C++和汇编的混合编程。其主要功能如下:





- 具有集成可视化代码编辑界面。用户可通过其界面直接编写 C 语言、汇编语言和.cmd 命令文件等。

- 含有集成代码生成工具。包括汇编器、优化 C 编译器和链接器等，将代码的编辑、编译、链接和调试等诸多功能集成到一个软件环境中。
- 含有高性能编辑器。支持汇编文件的动态语法加亮显示，使用户很容易阅读代码，发现语法错误。
- 工程项目管理工具可对用户程序实行项目管理。在生成目标程序和程序库的过程中，建立不同程序的跟踪信息，通过跟踪信息对不同的程序进行分类管理。
- 含有基本调试工具。此工具具有装入执行代码、查看寄存器、存储器和反汇编、变量窗口等功能，并支持 C 源代码级调试。
- 含有断点工具。能在调试程序的过程中，完成硬件断点、软件断点和条件断点的设置。
- 含有探测点工具。可用于算法的仿真、数据的实时监控等。
- 含有分析工具。包括模拟器和仿真器分析，可用于模拟和监视硬件的功能、评价代码执行的时效。
- 含有数据的图形显示工具。可以将运算结果用图形显示，包括显示时域/频域波形、眼图、星座图和图像等，并能进行自动刷新。
- 提供 GEL 工具。利用 GEL 扩展语言，用户可以编写自己的控制面板/菜单，设置 GEL 菜单选项，方便直观地修改变量和配置参数等。
- 支持多 DSP 调试。
- 支持 RTDX 技术。可在不中断目标系统运行的情况下，实现 DSP 与其他应用程序的数据交换。
- 提供 DSP/BIOS 工具。此工具可增强对代码的实时分析能力。

### 1.1.3 工具栏常用图标

CCS 集成开发环境为程序调试提供了方便快捷的方式，除了使用主菜单中的命令外，还在工具栏上提供了快捷图标，非常直观。以下介绍了常用图标(或称工具条按钮)的功能，熟悉这些图标对运行和调试程序非常有帮助，其他图标可查阅参考文献[1]。

-  (Rebuilds the Project): 编译整个工程并链接生成可执行文件，等同于 Project→Rebuild All 命令。
-  (Incremental Build): 重新编译修改过的文件，再链接生成可执行文件，等同于按 F7 功能键或 Project→Build 命令。
-  (Run): 运行程序，等同于按 F5 功能键或 Debug→Run 命令。
-  (Animate): 动画运行，等同于按 Alt+F5 功能键或 Debug→Animate 命令。
-  (Halt): 终止程序运行，等同于按 Shift+F5 功能键或 Debug→Halt 命令。
-  (Step Into): 单步跟踪，等同于按 F11 功能键。
-  (Step Over): 单步执行，等同于按 F10 功能键。
-  (Step Out): 跳出函数或子程序，等同于按 Shift+F11 功能键或 Debug→Step Out 命令。如果当前正处于子程序中，执行该命令可以完成子程序的执行。
-  (Run to Cursor): 运行到光标处，等同于按 Ctrl+F10 功能键。

-  (Toggle Breakpoint): 在当前位置设置/取消断点, 等同于按 F9 功能键或 Debug→Breakpoints 命令。
-  (Remove all Breakpoints): 取消所有断点。
-  (Toggle Probe Point): 在当前位置设置/取消剖析点(也称探测点)。
-  (Remove all Probe Points): 取消所有剖析点(或探测点)。

## 1.2 CCS 集成开发环境的安装配置及使用实验

### 1.2.1 实验目的

- 掌握 CCS v3.1 安装和软件仿真配置过程。
- 学习创建工程和管理工程的方法。
- 初步了解 CCS 软件集成环境中的基本编译和调试方法。

### 1.2.2 实验设备

PC 兼容机一台; 操作系统为 Windows 2000(或 WindowsNT 4.0、Windows 98、Windows X P, 以下均假定操作系统为 Windows2000)。

### 1.2.3 实验原理

开发 TMS320C55xx 应用系统, 一般需要先进行系统的软件开发和调试, 这也是硬件调试的辅助手段, 可以使用 CCS 集成开发环境进行软件仿真调试, 完成系统的软件开发。CCS 集成开发环境提供了一整套程序编制、编译、维护和调试环境, 能将汇编语言和 C 语言程序编译链接, 生成 COFF 公共目标文件格式的可执行文件, 并能将程序下载到目标 DSP 上运行调试。用户系统的软件部分由 CCS 建立的工程文件(projectname.pjt 或\*.pjt)进行管理。工程文件一般包含以下几种文件:

- 源程序文件: C 语言文件(filename.c 或\*.c)或汇编语言文件(filename.asm 或\*.asm)。
- 头文件和包含文件(filename.h 或\*.h)。
- 链接命令文件(filename.cmd 或\*.cmd)。
- 库文件(filename.lib 或\*.lib)。
- COFF 公共目标文件(filename.obj 或\*.obj)。
- 可执行文件(filename.out 或\*.out)。

### 1.2.4 CCS 应用程序的安装和配置

#### 1. CCS 的安装

首先将安装光盘插入 DC-ROM 驱动器中, 找到光盘根目录下的安装程序 setup.exe(假设安装路径是 E:\TI\Code.Composer.Studio.Ver.3.1\setup.exe), 双击图标运行安装程序, 然后按照提示进行安装(或先将安装程序 Code.Composer.Studio.Ver.3.1\ Setup.exe 从光盘上复制到

PC 机的硬盘上再进行安装)。安装完成后，屏幕桌面上将出现如图 1-1 所示的(a)和(b)两个图标。



图 1-1 CCS 的安装图标

## 2. CCS 的配置

双击桌面上图 1-1(a)中的“Setup CCStudio v3.1”图标，运行 CCS 设置程序，将出现一个“Code Composer Studio Setup”系统配置对话框，如图 1-2 所示。本章只介绍软件仿真设置，有两种设置方法。其一是用标准配置文件设置系统配置，即在 Factory Boards 中添加其设置；其二是用导入方法来实现系统配置。

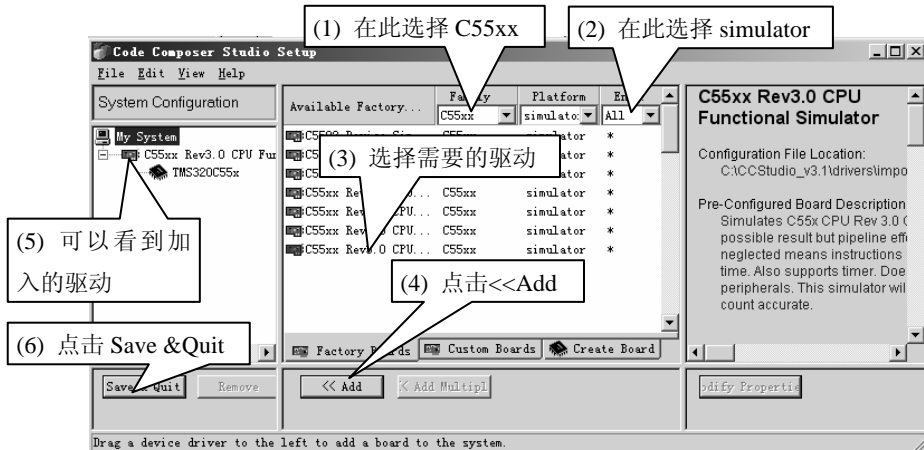


图 1-2 用标准配置文件设置系统配置

首先介绍在 Factory Boards 中添加设置，其步骤如下(见图 1-2):

- (1) 在 Family 下选择 C55xx，将看到所有 C55xx 的仿真驱动，包括软件仿真和硬件仿真。
- (2) 在 Platform 下选择 Simulator。在 Available Factory Boards 中只显示软件仿真驱动，选中相应的驱动。
- (3) 在此我们选择 C55xx Rev3.0 CPU Functional Simulator，图 1-2 中右边窗口会显示相应的说明。
- (4) 点击<<Add，将所选的驱动加入 System Configuration 中。
- (5) 可以在 System Configuration 中 My System 下看到所加入的驱动，见图 1-2 中标注的步骤(5)。
- (6) 点击 Save & Quit。保存设置并退出 Code Composer Studio Setup 程序，此时将弹出一个如图 1-3 所示的对话框。点击 Yes，将保存设置、退出 Code Composer Studio Setup 程序并启动运行 CCS；点击 No，将保存设置、退出 Code Composer Studio Setup 程序而不启动运行 CCS；点击 Cancel，将保存设置、不退出 Code Composer Studio Setup 程序。

配置好后，不必每次实验都进行设置，下次实验如果不更改系统配置，就可直接双击屏幕桌面上图 1-1(b)所示图标，运行 CCS。

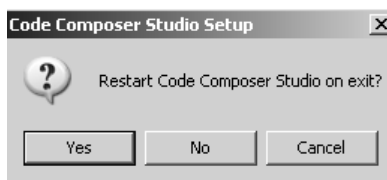


图 1-3 启动运行 CCS 对话框

下面介绍用导入方法来实现设置。运行 Code Composer Studio Setup，从菜单 File→Import 进行导入设置，如图 1-4 所示。点击 Browse 进入图 1-5 所示界面，在 CCS 3.1 目录下的 drivers\import 中找到 c55xfuncsim\_rev30.ccs，选中并打开即可。在导入设置时，系统将清除原有的设置。



图 1-4 导入方法步骤(一)

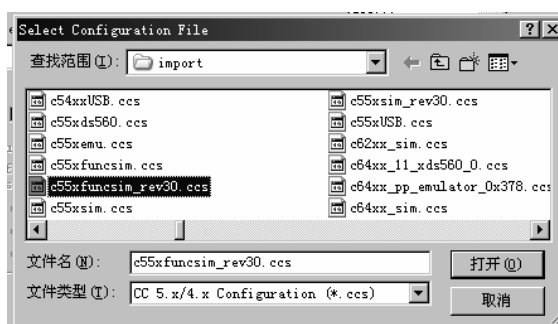


图 1-5 导入方法步骤(二)

## 1.2.5 CCS 环境中工程文件的使用

在 CCS 集成环境下开发汇编程序或者 C/C++程序，首先要建立一个工程项目文件 (\*.pj1)，再向工程项目文件中添加汇编程序源文件 (\*.asm)、C/C++源文件 (\*.c)和链接命令文件 (\*.cmd)，并设置工程项目选项。使用 CCS 开发应用程序的一般步骤如下：

(1) 创建或打开一个工程项目文件 (\*.pj1)，编辑各类文件。可以使用 CCS 提供的集成编辑环境，对链接命令文件 (\*.cmd 文件)和源程序 (\*.c, \*.asm 文件)进行编辑。

(2) 对工程项目进行编译。在编译过程中如果出现语法错误，将在编译链接信息视窗 (Build)窗口中显示错误信息，用户可以根据显示的信息找到错误位置，更改错误。

(3) 对结果和数据进行分析和算法评估。用户可以利用 CCS 提供的探测点、图形显示和性能评价等工具，对运行结果及输出数据进行分析，评估算法性能。

下面是使用 CCS 的具体步骤，掌握这些方法很重要，在以后的章节中都要用到。

### 1. 建立一个新的工程项目文件

新建一个工程文件的步骤如下：

(1) 在主菜单中选择 **Project**→**New** 命令,将在屏幕上弹出一个 **Project Creation** 创建工程项目文件对话框,如图 1-6 所示。

(2) 例如,将 **CCS** 安装在 **C:\CCStudio\_v3.1** 中,则在 **C:\CCStudio\_v3.1** 路径下有一个 **MyProjects** 文件夹,可以将所创建的工程文件保存在这个文件夹中。在 **Location** 域中选择路径 **C:\CCStudio\_v3.1\MyProjects**;在 **Project Name** 工程文件名域中键入工程名称,如果工程文件名为 **dsp\_prog**(由用户命名),系统将在 **MyProjects** 文件夹下新建一个 **dsp\_prog** 文件夹,并把工程 **dsp\_prog.pjt** 放在 **dsp\_prog** 文件夹下。

特别提示:千万要记住这个路径,这对初学者尤为重要。因为本次的工程在调试过程中生成的目标文件和可执行的输出文件都会自动地保存在该路径下名为“**dsp\_prog.pjt**”的工程文件夹中。下一次当你要打开“**dsp\_prog.pjt**”这个工程文件时,就要按照这个路径去找。

(3) 在 **Project Type** 工程类型域中,选 **Executable (.out)**可执行文件。

(4) 在 **Target** 目标类型域中,**TMS320C55xx** 仿真目标类型是自动出现的,想要改变这一选项,必须回到前面图 1-2 所示的界面重新进行设置。

(5) 单击 **Finish** 按钮,系统就会创建一个名为 **dsp\_prog.pjt** 的工程项目文件,如图 1-6 所示。



图 1-6 创建新的工程项目文件对话框

### 2. 打开和关闭工程

在主菜单中选择 **Project**→**Open** 命令,可以按照一定路径找到已存在的工程文件名。打开这个工程文件,工程包含的信息也将被载入。

在主菜单中选择 **Project**→**Close** 命令,可关闭当前工程文件。

### 3. 创建新文件

创建新文件的步骤如下：

(1) 在主菜单中选择 **File**→**New**→**Souce File** 命令,新建一个空白文件,在文件中输入源代码。

(2) 在主菜单中选择 **File**→**Save** 命令或选择 **File**→**Save As** 命令,出现保存文件对话框,如图 1-7 所示,将文件保存在与工程目录相同的文件夹下,以便于查找。在“文件名(N)”域中输入文件名和扩展名(扩展名为 **.asm**)。

(3) 单击保存(Save)按钮，将源程序保存。

(4) 在主菜单中选择 File→New→Souce File 命令，新建一个空白文件，在其中输入命令文件内容。

(5) 在主菜单中选择 File→Save 命令或选择 File→Save As 命令，同样会出现保存文件对话框，如图 1-7 所示，将文件保存在与工程目录相同的文件夹下，以便于查找。在“文件名(N)”域中输入文件名和扩展名(扩展名为 .cmd)。

(6) 单击保存(Save)按钮，将命令文件保存。

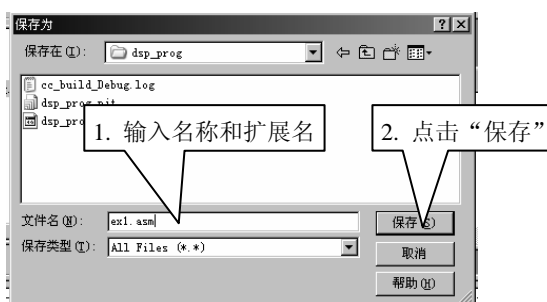


图 1-7 保存文件对话框

特别提示：保存文件时一定要记住它放在了什么地方，在这里，当前文件所在路径是：“C:\CCStudio\_v3.1\ MyProjects\dsp\_prog”。我们新建的其他文件最好也都保存在名为“dsp\_prog.pjt”的工程文件夹中，这样做是为了便于管理，在下一步把“dsp\_prog.asm”源文件添加到这个工程时，按照这个路径能够容易找到。

#### 4. 向工程项目中添加文件

(1) 给工程添加汇编源文件。在主菜单中选择 Project→Add Files to Project 命令，或者在工程视窗中选中 dsp\_prog.pjt 工程项目文件中的 Source 源文件，单击鼠标右键，从快捷菜单中选择 Add Files to Project 命令，系统会弹出如图 1-8 所示的对话框，在“文件类型(T)”中选择 Asm Source Files (\*.a;\*.s)，将可以看到所有 .asm 文件，选中所需要的文件，点击打开按钮或双击文件名，即可添加文件到工程项目中。

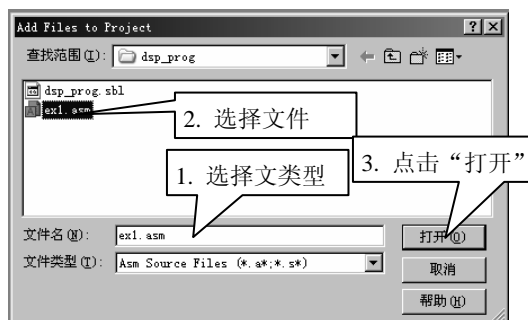


图 1-8 向工程项目中添加文件对话框

(2) 给工程添加命令文件。方法和添加汇编源文件相似，在主菜单中选择 Project→Add Files to Project 命令，系统同样会弹出如图 1-8 所示对话框，在“文件类型(T)”域中选择 Linker

Command Files(\*.cmd, \*.lcf), 将可以看到所有 .cmd 文件, 选中所需要的文件, 点击打开按钮或双击文件名, 即可添加文件到工程项目中。(注意, 这里与前面添加的方法只是扩展名不同。)


### 5. 从工程项目中删除文件

在工程管理视窗中右击某文件, 从关联菜单中选择 Remove from project 命令, 可以从工程项目中删除此文件。



## 1.2.6 编译、链接和运行目标文件

编好程序后就可以对工程进行编译、汇编和链接, 具体操作方法如下。

### 1. 对程序进行编译

在主菜单中选择 Project→Compile 命令(或者选择工具条中的  按钮), 就可以对当前的汇编程序进行编译并生成 .obj 文件。如果程序存在语法错误, 那么就会在左下方的编译链接信息视窗中显示错误信息, 根据错误提示, 可以对程序进行修改。

### 2. 对程序进行编译链接

在主菜单中选择 Project→Rebuild All 命令(或者选择工具条中的  按钮), 会对当前工程文件中的所有项目文件同时进行编译、汇编和链接并生成 .out 文件。在编译、汇编和链接过程中, 如有错误, 则会在编译链接信息框中显示错误信息, 对错误进行修改无误后, 在主菜单中选择 Project→Build 命令(或者选择工具条中的  按钮), 就会生成与工程项目同名的可执行 dsp\_prog.out 文件, 如图 1-9 所示。

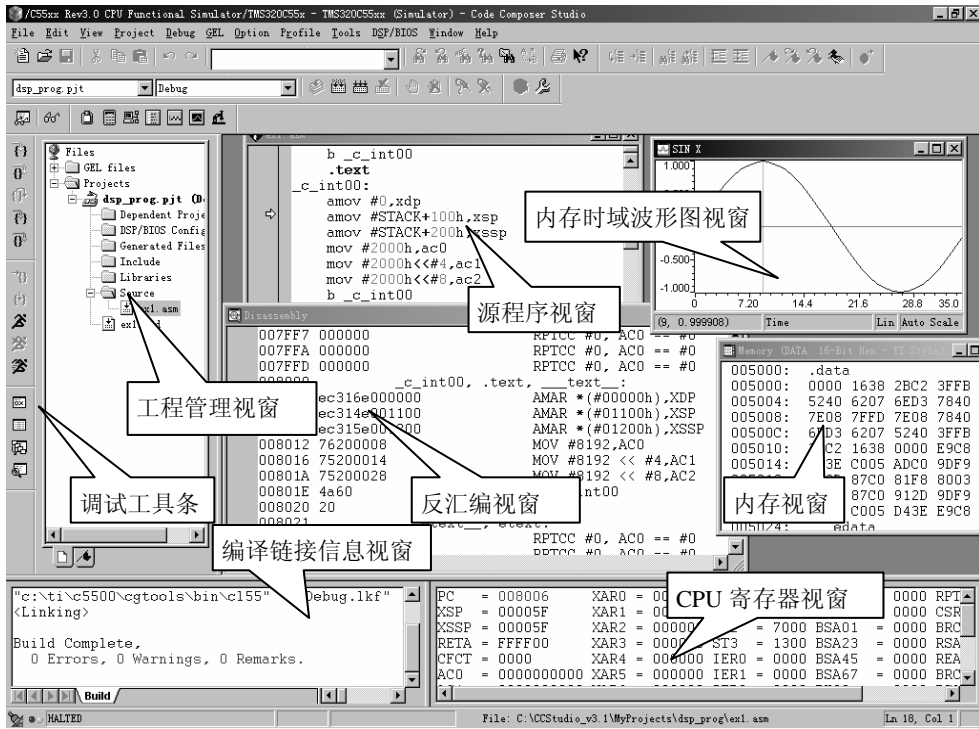


图 1-9 用 CCS 调试程序时出现的几种视窗工程界面