

9. DSP 开发基础报告

9.1 实验目的

1. 了解 DSP 开发系统的基本配置
2. 熟悉 DSP 集成开发环境（CCS）
3. 掌握 C 语言开发的基本流程
4. 熟悉代码调试的基本方法

9.2 实验仪器

计算机, TMS320F28335 DSP 教学实验箱, XDS510 USB 仿真器

9.3 实验内容

建立工程, 对工程进行编译、链接, 载入可执行程序, 在 DSP 硬件平台上进行实时调试, 利用代码调试工具, 查看程序运行结果。

9.4 实验准备

TI 的 CCS 5 集成开发环境, 不仅支持汇编的编译、链接, 还支持对 C/C++ 汇编、编译、链接以及优化。同时强大的 IDE 开发环境也为代码的调试提供了强大的功能支持, 已经成为 TI 各 DSP 系列的程序设计、制作、调试、优化的主流工具。

TMS320F283x 软件开发流程如图 9.1 所示。(图 9.1 重新绘制, 按照 P28 图 2-1)

下面简单介绍各主要模块功能。

- C/C++ Compiler (C/C++ 编译器)

C/C++ 编译器把 C/C++ 程序代码编译为基于 DSP 汇编指令集的汇编代码。这种转换并非一一对应, 甚至会产生冗余的汇编代码, 在某些场合需要使用优化器 (Optimizer) 来提高转换的效率, 使得汇编代码长度尽可能的短小, 程序所使用的资源尽可能的少。优化器是编译器的一部分。编程效率与编译器直接相关。

- Assembler (汇编器)

汇编器负责将汇编语言代码转换为符合公共目标格式 (COFF) 的机器语言, 这种转换是一一对应的, 每一条汇编指令都对应了唯一的机器代码。源文件中还包括汇编指令、伪指令和宏指令。这里的汇编代码包括了由 C/C++ 编译器生成的汇编代码和直接编写的汇编代码。

- Linker (链接器)

链接器负责把可重定位的多个目标文件和目标库文件转换为一个 DSP 可执行程序, 其中包含程序的机器代码、数据以及其他用来链接和加载程序所需要的信息。链接器必须依赖配置命令文件 (CMD) 的指令, 实现对目标文件中各段的定位。

- Run-time-support library (运行支持库)

对于用 C/C++ 语言中编写 DSP 程序中的某些功能 (例如存储器的寻址定位、字符串转换等) 并不属于 C/C++ 语言所能描述对象, 包含在 C/C++ 编译器中的运行支持库却可以很好的支持这些算法的标准 ANSI/ISO C 函数描述。函数运行支持库包含有 ANSI/ISO C 的标准运行支持库函数、编译器功能函数、浮点算术函数和系统初始化子程序 (这些函数都集成在汇编源文件 `rts.src` 中)。当对 C/C++ 编写的 DSP 程序进行链接时, 必须根据不同型号的 DSP 芯片添加相应的运行支持库到工程中。除此之外, 在使用运行支持库中的函数时, 必须在程序起始处用 `include` 语句包含相应的头文件 (如使用数学运算 `sin`、`cos` 时, 必须包

含 math.h)。而采用汇编语言编写程序时，却不需要这个运行支持库。因此 C 语言编写的 DSP 程序链接后，会产生大量的“冗余”汇编程序。

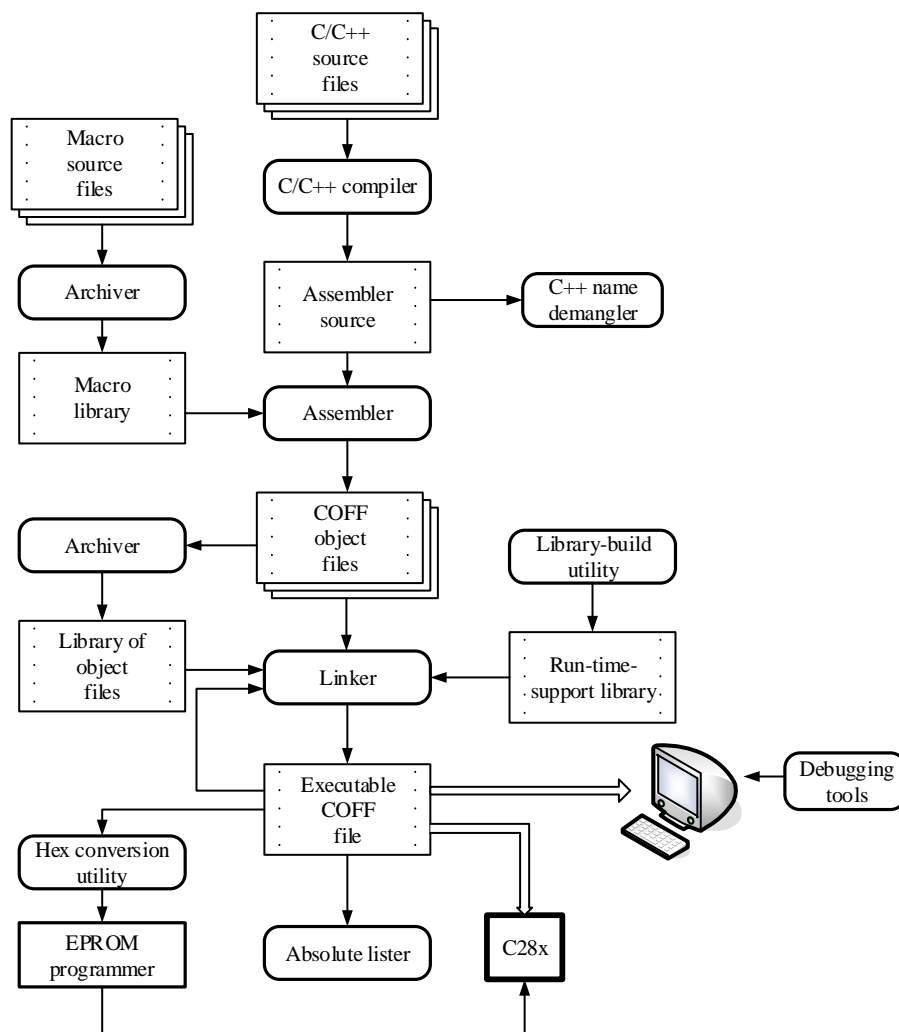


图9.1 TMS320F283x软件开发流程

由此可见，用 C/C++语言来开发 DSP 程序，一般在工程中必须包含以下文件：

- .c 或者 .cpp：C 或 C++程序，是主程序或函数，用于描述用户特定的算法功能；
- .cmd：配置命令文件，用于对编译生成的 COFF 格式目标文件（.obj）定位，安排各段的物理存储空间；
- .lib：运行支持库文件，不同芯片有不同的运行支持库，必须根据具体芯片加以选择，例如 TMS320F283x 的运行支持库文件名为 rts2800_fpu32.lib。（后缀 fpu32 含义是支持 32 位浮点运算）。

至于头文件（.h），只有当使用了运行支持库中相应的函数时，才需要在 C 文件的主程序中用 include 语句指定相应的头文件（math.h、stdlib.h、float.h 等）。具体内容参见 TI 公司的 TMS320C28x Optimizing C/C++ Compiler User's Guide。

其次用户自定义函数、寄存器地址、常量定义等信息也可以编制到头文件中，使用时也同样需要在 C 主程序中指定。

例如本实验中，需要的文件：

- main.c：C 语言主程序。
- 28335_RAM_lnk.cmd, DSP2833x_Headers_nonBIOS.cmd：配置命令文件。

- rts2800_fpu32.lib: 运行支持库。
- Sine.h: 常量定义头文件。
- FPU.h: 浮点运算库头文件。
- sine.dat: 实验中需要的数据文件。

对于使用 CCS 以工程为单位进行 DSP 程序的项目开发时，一般为每个工程建立一个独立的目录，将项目中所需要的文件都存放在该工程目录下，便于程序的管理。

rts2800_fpu32.lib 在 TI 的安装目录...\TI\c2000\cgtools\lib 中可以找到。

9.5 实验步骤

1. 设备检查

检查仿真器、F28335 DSP 教学实验箱、计算机之间的连接是否正确，打开计算机和实验箱电源。

2. 启动集成开发环境

点击桌面 CCS5 快捷方式，启动 CCS 集成开发环境。

3. 新建工程

新建一个工程“Project → New CCS Project”命令，弹出“CCS Project”对话框。在第一项 Project Name 中输入新建的工程名称，第二项 Project Type 中选择输出文件格式

“Executable (.out)”，在第三项 Location 中选择工程所在目录，在第四项 Device 中选择与当前 DSP 芯片吻合的“2833x Delfino → TMS320F28335”，在 Connection 中选择仿真器型号“SEED XDS510PLUS Emulator”。在“Project templates and examples”中选择

“Empty Project”，单击“完成”按钮确定。则在工程指定的目录中，建立了一个以工程命名的工程文件，它会存储有关该工程的所有设置。

4. 添加工程文件

选中工程文件后右键选择“Add Files to Project”命令，在弹出的对话框中依次选择当前工程目录下 main.c、source 目录夹下所有的文件、以及 28335_RAM_lnk.cmd（原工程产生的 cmd 文件内存分配不够会报错，需要修改，将它替换成改动过的 cmd 文件）、DSP2833x-Headers_nonBIOS.cmd，添加到当前工程中。在工程浏览窗口中，展开工程文件列表，可看到刚刚所添加的文件。

如果错误的添加了文件，可以在工程浏览窗口中的文件名中单击鼠标右键，在弹出的菜单中选择“Delete”。

当然，CCS 也支持文件编辑功能，可以在主菜单选择“File → New”新建一个文件，编辑完成保存为所需要相应格式的 C 语言程序、汇编程序、cmd 配置命令或者头文件，然后添加到工程中。

在添加完文件后，需要为工程添加搜索路径。右击工程标题，在弹出的对话框选“Properties”，进入工程配置对话框，选中左侧的“include Options 选项卡”，在右侧的“Add dir to #include path”中点击该框右上侧的“+”，选择“Workspace”，在新建的工程的目录下选择“header”文件夹，点击“Ok”，完成搜索路径的添加。

为工程添加库文件，在工程浏览窗口中的文件名中单击鼠标右键，在弹出的菜单中选择“Properties”，进入工程配置对话框，选中左侧的“General”选项卡，在“Runtime support library”选项中通过下拉框选择“rts2800_fpu32.lib”后点击“OK”完成库文件的添加。

5. 查阅代码

在 build 工程之前，先阅读一下源代码，明白各文件的内容：在“Project Explorer”栏展开“LAB_9”工程，双击“main.c”文件，即可在 CCS 的编辑窗口看到 c 程序的源代码，代码中有以下四个部分：

- 系统初始化函数 InitSysCtrl();
- 在主函数输出消息“SineWave example started”之后，进入一个无限循环，在循环体内调用了两个函数 dataIO() 和 processing()。
- 函数 dataIO() 在本实验中，DSP 不作任何实际操作而直接返回。
- 函数 processing() 对输入缓冲区的每个数据进行增益控制，并将结果存入输出缓冲区中。

6. 建立工程（Build 工程）

建立工程（build）是指对 asm、c 源程序文件进行编译（Compile）、汇编（Assemble），并结合配置命令文件对工程进行链接（Link），输出可执行程序（.out）。在主菜单选择“Project → Build Project”命令进行编译链接，生成的可执行.out 程序位于工程目录的 debug 子目录下。

对工程文件中的语法或是链接错误，CCS 会终止当前的 build，在底部消息窗口指示出程序包含的编译链接错误，或是警告信息。根据错误提示修改源程序文件或者配置命令文件，直至编译链接正确。

以上的工作称为目标代码生成。

7. 调试程序

当工程被正确建立以后，只有将程序通过仿真器下载到 DSP 芯片上，才能够进行实时的代码调试。

在“LAB_9”工程中双击“TMS320F28335.ccxml”，在弹出的“Basic”界面中“connection”选项中选择“SEED XDS510PLUS Emulator”，在“Board or Device”选项选择“TMS320F28335”后，点击右侧“Save Configuration”下的“Save”保存设置。

打开实验箱电源，在主菜单下选择“Run → Debug”，若仿真器正确连接后，进入“CCS Debug”界面。

8. 程序的运行

在 CCS Debug 环境界面的主菜单中选择“Run → Resume”可以让 DSP 从 main 函数的第一条语句开始执行程序。由于 DSP 程序输出并不具备 GUI 界面，由此执行结果只有依赖外部硬件或者查看寄存器、存储器的数值加以验证。在主菜单中选择“Run → Suspend”，可以暂停程序的执行。DSP 指令的执行严格按照指令流的顺序。

当想再次运行程序，可以执行菜单命令“Run → Restart”，使程序指针 PC 重新指向 _c_int00，也可以重新加载程序（“Run → Load → Reload Program”）。当执行菜单命令“Run → Reset”时，DSP 复位，内部寄存器恢复默认值，程序指针 PC 指向中断矢量表的复位向量处。

9. 程序的调试

在程序的开发与测试过程中，常常需要检查某个变量、或者是存储器的数值在程序运行过程中变化情况，这就需要暂停程序执行，用断点与观察窗口等方式来验证数值的正确性。这就是 DSP 目标代码的调试。

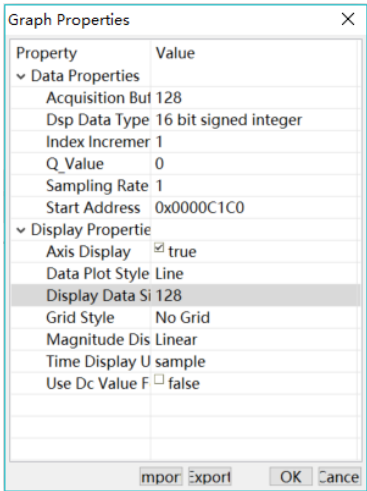
添加结构体变量 currentBuffer 到变量观察窗口（Add Watch Expression），观察 currentBuffer.output 和 currentBuffer.input 的地址以及数值。添加 dataIO() 到变量窗口，查看该子程序的入口地址。

在 dataIO() 处设立断点，在断点属性中关联输入文件 sine.dat，设置数据加载的起始地址为 currentBuffer.input，长度为 128。

Identity	Name	Condition	Count	Action
<input checked="" type="checkbox"/>	main.c, line 55 (CSL1)	S/W Breakpoint	0 (0)	Remain Halted

鼠标移动到断点所在行，右键选择“Breakpoint Properties”，在“Action”选项中选择“Read Data from file”，在“File”选项中选择工程文件夹中的“sine.dat”文件，勾选“Wrap Around”选项为“true”，起始地址“Start Address”为 currentBuffer.input 的起始地址，数据长度为 128，点击“OK”。

打开图形显示功能，在主菜单的“Tools → Graph → single time”查看存储空间 currentBuffer.input 和 currentBuffer.output 的时域波形。



查看存储空间的数值在程序相关语句执行前后的变化。
 在 processing() 子程序中设置断点，分别执行主菜单命令“Run → Step into”和“Run → Step over”单步执行程序，查看并比较这些单步执行方式的区别。

9.6 实验要求

1. 独立完成项目编译、链接、调试的全过程。
2. 记录 dataIO()、processing() 子程序的入口地址，记录 currentBuffer.input 和 currentBuffer.output 所在存储器地址。
3. 记录增益控制处理后，以图形方式显示数据空间 currentBuffer.input 和 currentBuffer.output 缓冲存储器中的波形。
4. 打开工程的.map 文件，查看所有的段在存储空间的地址、长度和含义，指出分别位于 TMS320F28335 的什么存储空间以及物理存储块名称，主程序中所用的变量分别属于什么段？
5. 查看.cmd 命令文件，比较其与上述.map 中的映射关系。试图修改.cmd 文件，再次编译链接，查看配置命令与各段的映射关系。

9.7 注意事项

在 CCS 集成开发软件中调试程序时，务必确保 DSP 实验箱电源加载正常。

10. 任意信号发生器

10.1 实验目的

1. 熟悉 DSP 硬件开发平台
2. 熟悉 DSP 集成开发环境（CCS）
3. 掌握 TMS320F28335 的存储器配置表
4. 学习 TMS320F28335 的编程开发
5. 熟悉代码调试的基本方法

10.2 实验仪器

计算机，TMS320F28335 DSP 教学实验箱，XDS510 USB 仿真器，示波器

10.3 实验内容

建立工程，编写 DSP 的主程序，并对工程进行编译、链接，利用现有 DSP 平台实现任意波的产生，通过示波器观察结果。

10.4 实验准备

10.4.1 程序流程

在 TMS320F28335 DSP 教学实验箱平台上实现任意波形的产生，可通过 DSP 实时运算得到相应波形的数据，随后通过 DAC 完成模拟输出。在该实验中，我们利用 DSP 的运算能力，首先计算出波形的数值信息，存储到相应的数据空间中，通过查表的方式读取该波形的数值并写入到 DAC 端口，实现任意波形的生成。由此可得程序流程如图 10.1 所示。

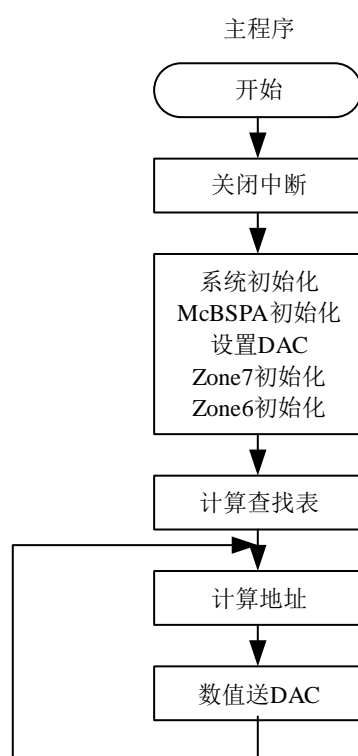


图 10.1 任意信号发生程序流程图

该实现方式属于查表法，类似于直接数字频率合成 DDS 的数字产生部分原理，可以改变相位控制字来改变输出信号的频率。

实现程序可参照工程 Lab10。

10.4.2 数据的定标

TMS320F283xx 是浮点 DSP 芯片，可以采用浮点或定点数进行数值的运算。然而板载 DAC 器件 AD9747 是 16 位定点格式，因此存在数据的定标问题。

数据最大表示范围取决于 DSP 芯片给定的字长，字长越长，所能表示数的范围就越大，精度也越高。数据以 2 进制补码格式表征，最高位是符号位，其余 15 位表示数值的大小。

而在实际中，数值的大小、数据的运算都会带来小数，用定点数格式表示小数，确定小数点的位置，称之为数据的定标。数据的定标一般有 Q 表示法，即 Q15 表示在定点数格式中有 15 位小数。由此 16 位定点数有 16 种 Q 表示形式，对应了 16 种十进制数据范围。例如 16 位定点的 Q0 表示没有小数，数据范围[-32768, 32767]；Q4 表示有 4 位小数，数据范围[-2048, 2047.9375]；Q15 表示有 15 位小数，数据范围[-1, 0.9999695]。可见，不同 Q 所表示的数据范围和精度都有所不同，精度与范围是一对矛盾，在实际定点算法中，为了达到最佳性能，必须对数据进行合理的定标。

浮点数 X_F 与定点数 X_D 的转换关系可表示为：

$$\text{定点数 } X_D = \lfloor X_F \times 2^Q \rfloor$$

$$\text{浮点数 } X_F = X_D \times 2^{-Q}$$

在程序中，根据数据的动态范围来确定 Q 值，分析程序中的数据可能的绝对值最大值 $|\max|$ ，使下式成立：

$$2^{n-1} < |\max| < 2^n$$

则 $Q=15-n$ 。
例如某变量的值在 -1 到 $+1$ 之间，即 $|\max|<1$ ，因此 $n=0$ ， $Q=15$ 。

10.4.3 相关实验硬件资源

TMS320F28335 内部采用哈佛结构总线，内部的程序空间、数据空间采用统一的编址方式。其 memory map 参见附录 C 中图 C3.3。
除了 TMS320F28335 片上集成的存储器，在实验箱上还扩展了外扩控制/状态寄存器，SRAM 和 CPLD 等资源供实验者使用。其地址分配如下表 10.1 所示。

表 10.1 外扩存储器地址映射		
地址范围	存储体	备注
0x00 4000 ~ 0x00 4FFF	外扩控制/状态寄存器	占 ZONE0
0x10 0000 ~ 0x10 FFFF	外扩 SRAM	占 ZONE6
0x20 0000 ~ 0x2F FFFF	外扩 CPLD	占 ZONE7

TMS320F28335 的外部接口模块（XINTF）负责完成对外扩设备的连接管理。
TMS320F28335 的 XINTF 映射到 3 个独立的存储空间，分别是 ZONE0、ZONE6 和 ZONE7。每一个空间都有一个内部的片选信号，并可以通过编程来独立的配置访问建立时间（lead）、有效时间（active）和跟踪时间（trail），以实现 TMS320F2828335 与各种外部存储器或设备的无缝连接。
实验平台中外扩 SRAM 占用 ZONE6 区域，该 SRAM 型号为 IS61LV51216-12，16 位数据线，19 位地址线，与 DSP 的连接原理图如图 10.2 所示。具体内容参见“作业 XINTF 模块设置”。

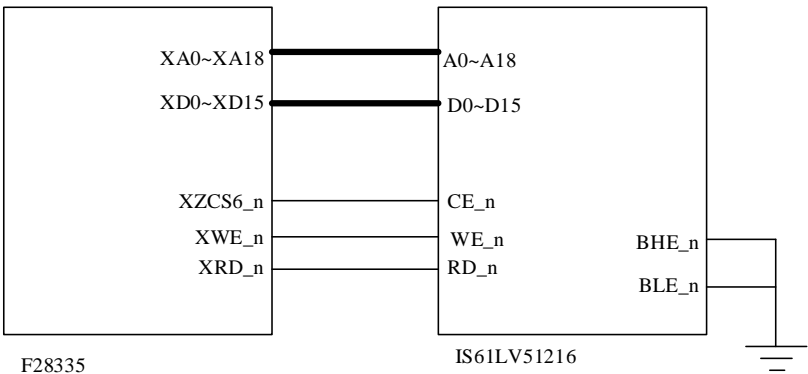


图 10.2 SRAM 与 DSP 连接示意图

实验箱上的 DAC 采用的是 AD9747，位宽 16bit，数据以无符号数表示，转换速度 4ns，通过 SMA 端口 J5 输出，在 TMS320F28335 的地址映射为 0x200400（只写）。即 DSP 只要将数字信号写到该端口，DAC 自动完成模拟的转换。

10.5 实验步骤

- 1. 设备检查
检查仿真器、F28335 DSP 教学实验箱、计算机之间的连接是否正确，打开计算机和实验箱电源。
- 2. 启动集成开发环境
点击桌面 CCS5 快捷方式，进入集成开发环境 CCS。
- 3. 新建工程

新建一个 DSP 工程，编辑源程序、配置命令等相关文件，并在工程中添加这些程序文件。

要求产生一个线性调频信号，其数学表达式如式 10.1 所示：

$$s(t) = \cos(\pi K t^2) \quad (10.1)$$

其中调制斜率 K 为 39062，t 为持续时间是[-0.0128, 0.0128]，在采样时间内共 1024 个采样点，即有 1024 个离散数值。

源程序的编写可参照范例工程 LAB10 中的相关内容。

4. 建立工程 (Build)

建立工程 (build)，若出错，则根据错误提示，修改源程序文件或者配置命令文件，直至编译链接正确，生成可执行的.out 文件。

5. 调试程序

在工程中合理配置 ccxml 文件，打开实验箱电源，在主菜单下选择 “Run → Debug”，若仿真器正确连接后，进入 “CCS Debug” 调试界面。

在程序中的 “波形数值计算” 子模块后设置断点，运行程序后 PC 指针会停留在此处，打开图形显示功能，查看存储空间中保存的时域波形，是否为线性调频信号。如果不是，则重新修改程序，直至正确为止。

程序调试时，可以利用各种调试手段，比如打开寄存器窗口、变量窗口等辅助手段，查看数值计算是否满足要求。

6. 运行程序

若第 5 步正确，可去掉断点，重新全速运行程序。

连接 F28335 DSP 教学实验箱 SMA 输出端口 J5 至示波器，调节示波器，观察线性调频信号的输出。

10.6 实验要求

1. 独立完成项目编译、链接、调试的全过程。
2. 记录实验中个子程序包括主程序的入口实际地址，与 memory 比较，指出分别位于什么类型的存储器中。
3. 指出波形数据保存的空间地址，并以图形方式显示线性调频信号的波形，并保存，附在实验报告中。
4. 比较波形数据保存的不同存储空间区域 (DPS 内部 RAM 和外扩 SRAM)，对系统实现的影响。

10.7 注意事项

在 CCS 集成开发软件中调试程序时，务必确保 DSP 实验箱电源加载正常。

10.8 实验思考

1. 打开工程的.map 文件，与实验 9 比较，指出编译产生的段有哪些区别。
2. 在保持源文件功能正确的前提下，仅修改.cmd 配置命令文件，改变段的地址分配，链接工程后，执行程序，如果出现错误，思考原因。
3. 在不修改波形数值计算子模块前提下，即保持波形数值表中的数据，依照 DDS 原理，修改程序，调整线性调频信号的输出周期。

11. DSP 数据采集

11.1 实验目的

1. 熟悉 DSP 的软硬件开发平台
2. 掌握 TMS320F28335 的 ePWM 中时间基准子模块和事件触发子模块的基本使用方法
3. 熟悉 TMS320F28335 的中断的设置
4. 掌握 TMS320F28335 的 ADC 模块的基本使用方法
5. 掌握代码调试的基本方法

11.2 实验仪器

计算机，TMS320F28335 DSP 教学实验箱，XDS510 USB 仿真器，示波器，信号源

11.3 实验内容

建立工程，编写 DSP 的主程序，对工程进行编译、链接，利用现有 DSP 平台实现数据的采集、存储以及模拟还原，并采取多种方法予以验证。

11.4 实验准备

11.4.1 程序流程

为实现 DSP 的数据采集存储以及模拟的还原，必须依赖于 ADC、DSP 以及 DAC 三大基本部件，而 TMS320F28335 芯片上集成了 ADC 模块，因此实现该功能较为简单，数据采集的工作可以由 DSP 单独完成，只需要对相关外设模块进行合理配置。模拟还原由实验箱中 DAC（AD9747）来完成。TMS320F28335 中的 ADC 模块与 DSP 内核之间的通信可以通过查询方式或中断方式，在此，我们采用 ADC 的中断功能实现数据的交换。

TMS320F28335 中 ADC 的转换频率和采样频率可以独立设置，分别位于 ADC 模块和 ePWM 的时间基准子模块中，因此要使 ADC 工作，必须掌握 ADC 模块和 ePWM 模块中的相关设置。

由此可得程序流程如图 11.1 所示。

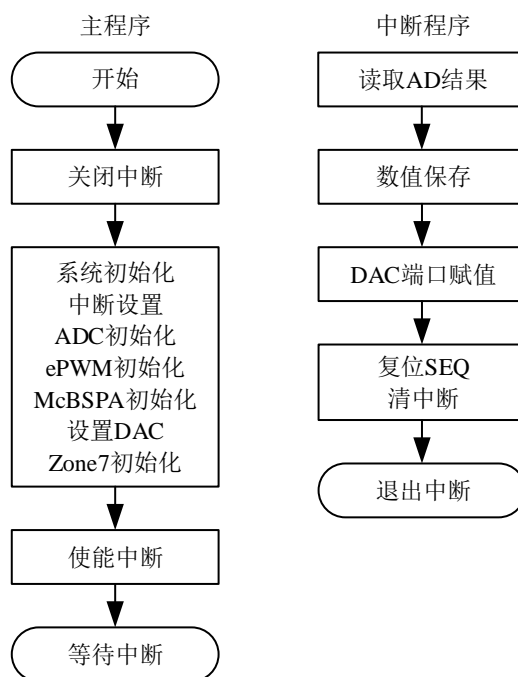


图 11.1 数据采集实验程序流程图

范例程序可参照工程 LAB11。

11.4.2 DSP 初始化

一般而言，DSP 要正常工作，必须首先设置时钟，时钟确定了 DSP 工作主频。TMS320F28335 中时钟设置大致分为三个主要寄存器，它们分别是锁相环控制寄存器（PLLCR）、外设时钟使能控制寄存器（PCLKCR0, PCLKCR1, PCLKCR2）和外设时钟预定标设置寄存器（HISPCP、LOSPCP）。

F28335 的时钟设置具体内容参照课件 2.4 等相关资料。

11.4.3 模数转换模块（ADC）

TMS320F28335 内部有一个 16 通道、采样精度为 12bit 的 ADC 模块。这 16 通道可配置两个独立的 8 通道模块，具有同步采样和顺序采样模式，模拟输入范围 0~3V，最快转换时间为 80ns，具有多个触发源用于启动 AD 的转换，采用灵活的中断控制。

ADC 模块具体工作原理以及设置可参见课件 2.8 等相关资料。

11.4.3 ePWM 模块

TMS320F28335 中 ePWM 模块的事件可产生 ADC 转换启动脉冲信号 SOC，本次实验采用时间基准子模块的产生周期事件，通过事件触发子模块的设置来产生 ADC 转换启动脉冲信号 SOC。

ePWM 模块具体工作原理以及设置可参见课件 2.7 等相关资料。

11.4.4 TMS320F28335 中断系统.

TMS320F283x 的外设中断扩展（PIE）单元通过少量中断输入信号的复用来扩展大量的中断源，PIE 单元支持多达 96 个独立的中断，这些中断以 8 个为一组进行分类，每组中的所有中断共用一个 CPU 级中断（INT1~INT12）。96 个中断对应的中断向量表存储在专用 RAM 区域中。PIE 向量表用来存储系统中每个中断服务程序（ISR）的入口地址。一般来说，在设备初始化时就要设置 PIE 向量表，并可在程序执行期间根据需要对其进行更新。

在实验中，当我们设置 $VMAP=1$ （ST1 寄存器的 bit3）， $ENPIE=1$ （PIECTRL 寄存器的 bit0）后，TMS320F28335 的中断向量地址范围 $0x000D00\sim 0x000DFF$ 。例如 ADC 外设模块 SEQ1INT 中断向量地址是 $0x000D40$ ，SEQ2INT 中断向量地址是 $0x000D42$ ，ADCINT 中断向量地址为 $0x000D4A$ （ADCINT 是 SEQ1INT 和 SEQ2INT 的逻辑或）。

要想正确使用中断，首先应该合理设置中断向量表，在对应地址填入中断服务子程序的入口地址。其次，必须对上述三个级别的中断作出正确的设置。比如实验中，要想实现 CPU 利用中断方式读取 ADC 的采样数据，可以使能 ADC 模块的中断 SEQ1INT，其次使能外设使能寄存器 PIEIER1.1，保证中断发生时 PIEACK1.1 位清零，最后使能 CPU 中断使能寄存器 IER 中的 INT1，以及全局中断使能位 INTM。这些工作必须在系统初始化时完成。退出中断服务程序前，清除 ADCST 中的 INT SEQ1 以及相应的 PIEACK_x。

关于 CPU 中断设置的详细内容参见课件 2.5 等相关资料。

11.5 实验步骤

1. 设备检查

检查仿真器、F28335 DSP 教学实验箱、计算机之间的连接是否正确，打开计算机和实验箱电源。

2. 启动集成开发环境

点击桌面 CCS 5 快捷方式，进入集成开发环境 CCS。

3. 新建工程

新建一个 DSP 工程，编辑源程序、配置命令等相关文件，并在工程中添加这些程序文件。

在源程序中，通过对中断、ADC 外设以及事件管理通用时钟的设置，利用中断方式读取 ADC 的采样结果，并用 DAC 实现模拟信号的还原。在程序中，开辟一段数据空间，用于保存 ADC 的采样结果，要求保存 1024 点数据，且该空间的数据不断刷新。

源程序的编写可参照工程 LAB11 中的相关内容。

4. 建立工程（Build）

建立工程（build），若出错，则根据错误提示，修改源程序文件或者配置命令文件，直至编译链接正确，生成可执行的.out 文件。

5. 连接外部电路

打开信号源，产生一个合适的频率（ADC 的采样频率必须满足奈奎斯特采样定律），信号幅度控制在 0-3V 以内，验证后将信号通过接口输入到 DSP 中。

打开示波器，将实验箱中的 SMA 接口 J5 输出到示波器上，并正确设置。

6. 调试程序

在工程中合理配置 ccxml 文件，打开实验箱电源，在主菜单下选择“Run → Debug”，若仿真器正确连接后，进入“CCS Debug”调试界面。

首先验证中断设置是否正确。可以在 ADC 中断服务程序的入口地址处添加断点，全速或者动画运行程序，检查程序计数器 PC 能否间隔性的停留在中断服务入口地址处。若能，说明中断设置基本正确。

若以上步骤正确，其次，验证数据采集的正确性。程序连续运行一段时间后，暂停程序执行，打开图形显示功能，查看存储空间中保存的时域波形，是否为信号源输出的信号波形。

若上述步骤正确，则调节示波器，观察信号波形，是否为信号源的输入波形。若是，则实验调试结束。

以上步骤如果出错，则可以利用各种调试手段，比如打开寄存器窗口、变量窗口等辅助手段，根据数值以及实验原理，查找错误原因，重新修改程序，直至正确为止。

7. 运行程序

若第 6 步正确，可去掉断点，重新全速运行程序。

连接实验箱 SMA 输出口 J5 至示波器，调节示波器，观察信号的输出。可以实时的改变信号源的输入信号（注意信号幅度不要随意修改，超出输入范围易烧毁实验电路），示波器上显示的波形亦会随之变化。

数据直通通道就是最简单的实时信号处理电路。

11.6 实验要求

1. 独立完成项目编译、链接、调试的全过程。
2. 根据范例程序，给出 ADC 的采样频率计算公式，修改 ADC 的采样频率，并验证。
3. 指出波形数据保存的空间地址，并以图形方式显示采集的信号波形，并保存，附在实验报告中。
4. 利用上述图形，给出采样频率的验证方法，以此检验数据采集程序的正确性。

11.7 注意事项

1. 运行 CCS 集成开发软件后，务必确保 DSP 实验箱电源加载正常。
2. 信号源在连接实验箱前，务必保证信号幅度控制在 0-3V 以内。

11.8 实验思考

1. 观察输入信号与示波器显示信号、存储器中存储波形信号幅度的差异，解释差异产生的原因。
2. 除了上述粗略验证 ADC 采样频率以外，思考其他测试采样频率的方法和手段。
3. 除了中断方式，DSP 内核还可以采用查询方式获取 ADC 外设的采样数据。如果采样查询方式，则需要查询哪些标志位，给出程序流程并编程实现。
4. 如何将存储的采样数据保存到数据文件中，并利用动态有效位 ENOB 测试方法分析实验平台数据采集的性能。

12. FIR 滤波器的 DSP 实现

12.1 实验目的

1. 巩固数字 FIR 滤波器的概念
2. 理解定点 DSP 中数的定标、有限字长、溢出等概念
3. 理解算法实现中实时的概念
4. 掌握 DSP 开发过程以及基本调试方法

12.2 实验仪器

计算机, TMS320F28335 DSP 教学实验箱, XDS510 USB 仿真器, 示波器, 信号源

12.3 实验内容

针对 FIR 算法, 设计滤波器系数, 完成数据的定标, 查看滤波器特性曲线。

建立工程, 编写 DSP 的主程序, 并对工程进行编译、链接, 利用现有 DSP 平台实现 FIR 滤波器算法, 通过信号源、示波器理解滤波器特性, 验证实现与理论设计的一致性。

12.4 实验准备

12.4.1 实验流程

实验之前首先必须对 FIR 滤波器的设计、实现算法有所了解, 必要时通过计算机算法仿真, 理解 FIR 滤波器特性。

根据 FIR 滤波器算法, 编写 C 源程序, 实现算法功能。并验证 DSP 实现时算法的正确性以及精度的要求。这种算法功能上的仿真可以利用 CCS 集成开发环境中数据 IO 来模拟信号的输入, 完成验证算法精度与功能的正确。

验证了算法的功能正确之后, 可以将程序下载到 DSP 上运行, 观察现象。更为重要的是, 在硬件平台上验证系统的实时性, 以及评估资源的使用情况。若满足实时性要求, 则测试各项指标, 应该与原理设计相吻合。如果实现与理论不一致, 则首先检查算法的实时性, 以及资源使用是否冲突等原因, 对程序进行优化后再次编译链接, 重新验证直至正确。算法的优化有时会贯穿于整个设计之中。

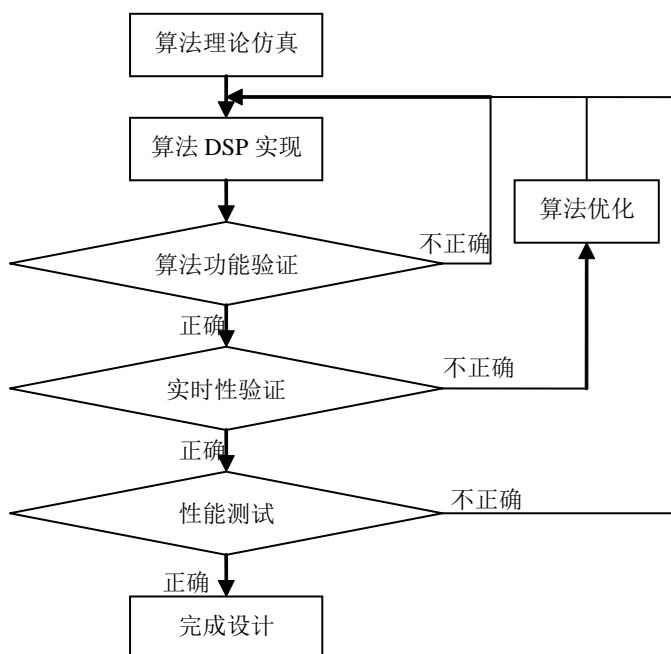


图 12.1 算法实现流程

12.4.2 程序流程

FIR 滤波器算法属于典型的数据流处理方式, 每到达一个新数据, 就必须进行一次计算, 更新输出。因此, 当一次采样完成之后, 就可以进行 FIR 核心算法, 并将计算结果输出给 DAC。

因此，和 DSP 的数据采集实验类似，用 DSP 实现实时的 FIR 信号处理算法必须依赖于 ADC、DSP 以及 DAC 三大基本部件。充分利用 DSP 片上 ADC 外设，实现模拟信号的采样，并由 DSP 完成 FIR 核心算法，由实验箱中 DAC（AD9747）来完成数字到模拟的还原。在数据采集实验基础上，我们对程序流程稍加改动，就可实现完整数字 FIR 滤波器功能。程序流程如图 12.2 所示。

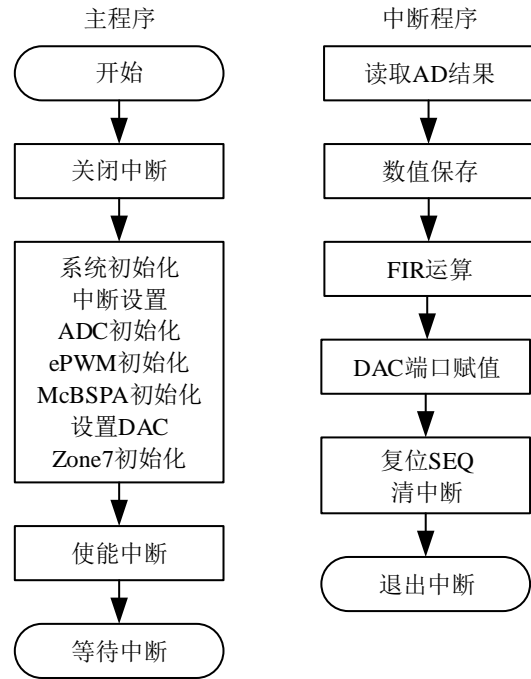


图 12.2 FIR 滤波器程序流程

12.4.3 FIR 滤波器设计

数字滤波器用于完成信号的滤波处理功能，是用有限精度算法实现的离散时间非时变系统。用 DSP 实现数字 FIR 滤波算法，具有稳定性强、精度高、实时性好、灵活性大、实现简单等优点。

有限长的单位冲击响应滤波器（FIR）差分方程可表示为：

$$y[n] = \sum_{k=0}^N h[k]x[n-k] \quad (12.1)$$

其中， h 是滤波器系数， x 为输入的数字信号， y 为 FIR 滤波器计算输出。 N 为滤波器阶数。由此可得，一个 N 阶的滤波器计算，需要 $N+1$ 个滤波器系数， $N+1$ 个数字输入，每得到一个 y 值，需要 $N+1$ 次乘法以及 N 次加法。另外， N 阶滤波器需要保存当前的 $N+1$ 个输入信号数值，以及事先设计的 $N+1$ 个滤波器系数。

滤波器系数的设计有很多方式，这里我们采用 MATLAB 软件来对 FIR 滤波器算法进行仿真并验证性能。

在 MATLAB 界面中，利用 `fir1` 命令来设计滤波器系数。`fir1` 的完整命令如下：

$$h = \text{fir1}(n, Wn, 'ftype', window) \quad (12.2)$$

其中, n 为滤波器阶数, W_n 为归一化截止频率 (这里的归一化指与采样频率一半进行归一化), W_n 对应了在幅频曲线上 -6dB 点的频率数值, f_{type} 为滤波器类型, 可以是低通、带通、高通、带阻等形式。window 是使用的窗函数, 可以是 hamming、hanning、chebwin 等形式。h 为产生的滤波器系数。详细说明可在 MATLAB 中输入 help fir1 或 doc fir1 查看。

对产生的滤波器系数可以用 freqz 命令查看幅频、相频特性曲线。具体命令如下:

$$\text{freqz}(h) \quad (12.3)$$

其中 h 为设计的滤波器系数。当然也可以使用 fvtool(h) 命令, 验证滤波器设计是否满足系统指标要求, 例如通带频带、阻带衰减、过渡带宽度等。

12.4.4 DSP 的算法实现

TMS320F28335 是浮点型 DSP, 存储器字长 16bit, 可进行 32bit 的运算。在 DSP 编程时, 若采用浮点运算, 则需要对采样的结果进行定点到浮点的转换, 输出时则需要进行浮点到定点的转换。若采用定点运算, 必须进行数据格式的定标。比如对 FIR 滤波器系数的定标可以参照“实验 10 任意信号发生器”中介绍的方法来完成。对系数定标后, 还要进行仿真以验证性能。

另外, 由于 TMS320F28335 的数据字长只有 32bit, DAC 接受的字长为 16bit, 因此有限字长带来精度的损失。更为重要的是当加法的结果超过 16 位表示范围时, 数据产生了溢出, 这是有限字长带来的第二个问题。再者, 数据的计算结果存放在 32bit 的寄存器中, 但 DAC 却是 16bit, 取高位输出还是低位输出, 还是取一个合适的范围, 这是数据截取的问题。因此在编写程序时, 必须考虑定点数据的运算效应, 由数据的动态范围来确定截取、定标等问题。定点 DSP 内部一般有溢出保护功能, 可以查看溢出标志位及时发现溢出现象, 其次用溢出模式位来使 ACC 结果控制在最大值范围之内, 以达到防止溢出引起精度严重恶化的目的。

12.4.5 算法实时性测试

算法的实时性测试主要指该算法能否在规定的时间内完成 FIR 运算, 规定时间在此是指采样周期。FIR 的运算必须在两次采样间隔内完成, 否则会造成数据的丢失。这是数据流处理的特点, 数据的运算速度必须大于数据的更新速度。

在实验平台中, 我们可以利用 GPIO 管脚来实测采样周期和算法执行时间。添加程序, 当程序进入断点时, 将 GPIO 的某一个引脚输出置高, 完成算法退出中断时再置低。由此当全速执行程序时, 测量该 GPIO 引脚上的周期, 便是采样周期, 高电平持续时间即为 FIR 滤波器算法执行时间, 由此判断计算法实现是否实时。

12.5 实验步骤

1. 系数设计

利用 MATLAB 设计滤波器系数, 并对系数进行数据定标, 完成浮点到定点的转换。分别作出两组系数的幅频、相频特性曲线, 看是否满足设计要求。

2. 设备检查

检查仿真器、F28335 DSP 教学实验箱、计算机之间的连接是否正确, 打开计算机和实验箱电源。

3. 启动集成开发环境

点击桌面 CCS 5 快捷方式, 进入集成开发环境 CCS。

4. 建立工程

新建一个 DSP 工程, 编辑源程序、配置命令等相关文件, 并在工程中添加这些程序文件。

在上次实验程序的基础上加以修改，在中断程序内添加 FIR 算法模块，完成 FIR 算法程序。设置 GPIO 的输出控制，使之能够完成算法执行时间测量的工作。

建立工程（build），若出错，则根据错误提示，修改源程序文件或者配置命令文件，直至编译链接正确，生成可执行的.out 文件。

5. 调试程序

在工程中合理配置 ccxml 文件，打开实验箱电源，在主菜单下选择“Run → Debug”，若仿真器正确连接后，进入“CCS Debug”调试界面。

6. 算法功能验证

在中断服务子程序恰当的地方，设置探针点，利用文件 IO 的方式，输入 x 数据（具体方法参见“实验 9 DSP 开发基础”）；在 FIR 算法结束处设置断点，验证 FIR 滤波器算法的正确性。输入的数据可具有一定的特殊性，进行特例验证。

若计算结果不正确，则进行程序的调试。调试关键在于现象重现、错误定位，可以利用各种调试手段，比如打开寄存器窗口、变量窗口等辅助手段，根据数值以及实验原理，查找错误原因，修改程序直至正确为止。

解决错误后，再重新恢复原错误程序，观察错误现象是否重现，以确定错误的唯一性。

7. 算法实时性验证

去除程序中的探针点以及断点重新全速运行程序，测量采样频率以及 FIR 算法的核心执行时间，判断该系统是否实时。

若非实时，则优化程序，甚至修改算法，直至满足实时要求。

8. 系统测试

打开信号源，产生一个合适频率（ADC 的采样频率必须满足奈奎斯特采样定律）的正弦信号，信号幅度控制在 0-3V 以内，验证后将信号通过 SMA 接口输入到 DSP 中。

打开示波器，将实验箱中的 SMA 接口 J5 输出到示波器上，并正确设置。

全速运行程序，调节信号源输出正弦信号的频率，记录各频点的示波器上输出幅度。描点作图，与理论幅频特性曲线比较，分析是否满足设计要求。

12.6 实验要求

1. 独立完成项目编译、链接、调试的全过程。
2. 当输入信号为正弦信号时，改变正弦信号频率，观察示波器，记录各频点对应的幅度，并描点作图，与理论设计的幅频曲线比对，做误差分析。实际测量幅频曲线与理论曲线均需附在实验报告中，指出 FIR 滤波器系数的设计参数指标。
3. 记录 FIR 核心算法程序执行时间，以及采样时间，判断该系统是否实时。

12.7 注意事项

1. 运行 CCS 集成开发软件的调试模式，务必确保 DSP 实验箱电源加载正常。
2. 信号源在连接实验箱前，务必保证信号幅度控制在 0-3V 以内。

12.8 实验思考

1. 观察各种输入信号通过数字滤波器系统之后的输出波形，解释信号失真原因。
2. 加载由汇编语言编写的 FIR 滤波器程序，测量运算时间，比较分析 C 语言效率低的原因。
3. 以该 FIR 滤波器系统为例，总结分析系统实时性的取决因素。

