

**DSP应用技术实验**

**DSP开发基础实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 作者 | ： | 许晓明 | 学号 | ： | 9161040G0734 |
| 同组人 | ： | 李玥 | 学号 | ： | 9161040G0703 |
| 同组人 | ： | 陈锦涛 | 学号 | ： | 9161040G0614 |
| 学院 | ： | 电子工程与光电技术学院 | | | |
| 专业 | ： | 电子信息工程 | | | |
| 班级 | ： | 电信3班 | | | |
| 组号 | ： | B4 | | | |
| 题目 | ： | DSP应用技术实验 | | | |
|  |  | DSP开发基础实验报告 | | | |
| 指导者 | ： | 李彧晟 | | | |

2019 年 11 月

目录

[1 实验目的 1](#_Toc25312292)

[2 实验仪器 1](#_Toc25312293)

[2.1 实验仪器清单 1](#_Toc25312294)

[2.2 硬件连接示意图 1](#_Toc25312295)

[3 实验步骤及现象 1](#_Toc25312296)

[3.1 实验箱测试 1](#_Toc25312297)

[3.2 C程序基础调试 2](#_Toc25312298)

[4 实验结果汇总及问题回答 8](#_Toc25312299)

[4.1 子程序入口地址与结构体存储地址 8](#_Toc25312300)

[4.2 显示缓冲存储器中的波形 9](#_Toc25312301)

[4.3 比较不同单步方式的区别 9](#_Toc25312302)

[4.4 查看.map文件信息 9](#_Toc25312303)

[4.5 查看及修改.cmd文件 9](#_Toc25312304)

[5 实验总结 10](#_Toc25312305)

[5.1实验中遇到的问题及解决方法 10](#_Toc25312306)

[5.2实验心得体会 11](#_Toc25312307)

# 1 实验目的

1. 了解DSP硬件开发平台基本配置；

2. 熟悉TI DSP软件集成开发环境；

3. 学习DSP软件开发流程；

4. 掌握工程代码产生方法；

5. 学习DSP软件调试方法。

# 2 实验仪器

## 2.1 实验仪器清单

|  |  |  |
| --- | --- | --- |
| 1. | DSP仿真平台（仿真器、DSP实验箱、计算机） | 一套 |
| 2. | 信号发生器 | 一台 |
| 3. | 示波器 | 一台 |

## 2.2 硬件连接示意图

实验硬件连接大致如图2. 1所示。其中，测试完实验箱后，实际上信号发生器与示波器可以不在与实验箱连接。

|  |
| --- |
| 图2. 1 硬件连接示意图 |

# 3 实验步骤及现象

## 3.1 实验箱测试

1.开启示波器、信号发生器，调节信号发生器输出幅度在0-1V以内。

2.连接实验设备，确认无误后开启实验箱电源，此时可观察到实验箱电源指示灯亮起。

3.在计算机上通过CCS 5创建工程并导入LAB11工程。

4. 对项目工程进行编译、链接（Build Project），进入调试（Debug）界面并运行程序（Resume）。

5.可观察到示波器波形与信号发生器一致。改变信号发生器波形及参数，示波器上波形也随之改变，如图3. 1所示，说明实验箱正常。

|  |
| --- |
| 图3. 1 测试实验箱是否正常 |

## 3.2 C程序基础调试

1.通过CCS 5新建工程并设置工程参数信息。

2.添加LAB9相关工程文件，为工程添加搜索路径、库文件。

3.对项目工程进行编译、链接（Build Project），进入调试（Debug）界面并运行程序（Resume），此时观察到屏幕上出现“SineWave example started”字样。

4.添加结构体变量currentBuffer到变量观察窗口，如图3. 2到图3. 3所示，可观察到**currentBuffer.input和currentBuffer.output所在存储器地址分别为0x0000C1C0和0x0000C240**。

5. 添加子程序dataIO()、子程序processing()到变量观察窗口，如图3. 4所示，可观察到**dataIO()入口地址为0x00B6DA；processing()入口地址为0x00B6BD**。

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

7.重新运行程序，如图3. 5所示，可观察到存储空间currentBuffer.input和currentBuffer.output中的数值发生变化。

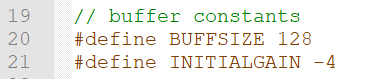
|  |
| --- |
| 图3. 2 currentBuffer.input地址及部分数值 |

|  |
| --- |
| 图3. 3 currentBuffer.output地址及部分数值 |

|  |
| --- |
| 图3. 4 子程序入口地址 |

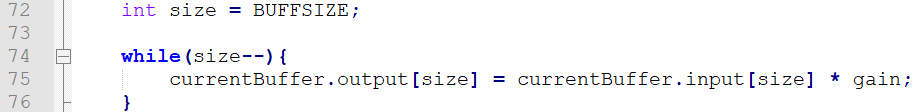
|  |
| --- |
| 图3. 5 设置断点并关联文件后的程序运行结果 |

8.通过图形显示功能，查看存储空间currentBuffer.input和currentBuffer.output的时域波形，如图3. 6到图3. 7所示，可知通过增益子程序processing()后，**output的波形幅度比input增大4倍**。同时**观察到二者的波形反相**，查看头文件sine.h ，可发现以下代码：



结合主程序main.c中的如下代码：





可知：这是**由于processing()对input中的数据乘上了负增益**而产生的结果。

|  |
| --- |
| 图3. 6 设置断点并关联文件后currentBuffer.input的时域波形 |

|  |
| --- |
| 图3. 7 设置断点并关联文件后currentBuffer.output的时域波形 |

9.在processing()子程序中设置断点，分别执行 “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。

10. 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。.map文件的部分内容如图3. 8所示。

11. 查看.cmd 命令文件，部分内容如图3. 9所示，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。

|  |
| --- |
| 图3. 8 .map文件部分内容 |

|  |
| --- |
| 图3. 9 .cmd文件部分内容 |

# 4 实验结果汇总及问题回答

## 4.1 子程序入口地址与结构体存储地址

如图3. 2到图3. 4所示，可汇总得到表4. 1。

表4. 1 子程序入口地址与结构体存储地址

|  |  |
| --- | --- |
| 名称 | 地址 |
| dataIO()子程序 | 0x00B6DA |
| Procession()子程序 | 0x00B6BD |
| currentBuffer.input | 0x0000C1C0 |
| currentBuffer.output | 0x0000C240 |

## 4.2 显示缓冲存储器中的波形

如图3. 6到图3. 7所示，由于processing()对input中的数据乘上了负增益，二者的波形反相，output的波形幅度比input增大4倍。

## 4.3 比较不同单步方式的区别

通过设置断点并单步调试可知：Step into单步执行遇到子程序将进入并且继续单步执行；Step over单步执行时，遇到子程序时不会进入子程序单步执行，而是将整个子程序执行完毕后再停止，即将子程序整个作为一步。

## 4.4 查看.map文件信息

如图3. 8所示，.map文件的MEMORY CONFIGURATION中，给出了各个存储器空间的首地址、总长度、已用空间和未用空间等信息；在SECTION ALLOCATION MAP中，给出了各段的首地址、长度等信息。查阅资料文件，可得到表4. 2。

表4. 2 .map文件各段信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 段名称 | page | 首地址 | 长度 | 作用 | 所在位置 |
| .pinit | 0 | 0x00008000 | 0x00000000 |  | RAML0 |
| .cinit | 0 | 0x00008000 | 0x00000090 | 存放程序中的变量初值和常量 | RAML0 |
| .text | 0 | 0x00009000 | 0x00002a2e | 存放程序代码 | RAML1 |
| .reset | 0 | 0x003fffc0 | 0x00000002 |  | RESET |
| .sysmem | 1 | 0x00000400 | 0x00000400 | 为动态存储分配保留的空间 | RAMM1 |
| .ebss | 1 | 0x0000c000 | 0x00000363 | 为程序中的全局和静态变量保留存储空间 | RAML4 |
| .econst | 1 | 0x0000d000 | 0x0000025a | 存放常量 | RAML5 |
| .stack | 1 | 0x0000e000 | 0x00000300 | 为程序系统堆栈保留存储空间 | RAML6 |

同时，可根据变量的存储地址及程序的入口地址推测它们所在的段，如currentBuffer结构体的input和output在.ebss段；dataIO()子程序和Procession()子程序在.text段。

## 4.5 查看及修改.cmd文件

如图3. 9所示，每个段映射得到的存储器首地址与.map文件中的地址相同。如.text段映射在RAML1，PAGE 0中定义RAML1的首地址为0x009000，长度为0x003000。则在.map文件中，.text段的首地址为0x00009000，且长度0x00002a2e与MEMORY CONFIGURATION中RAML1的使用空间一致。

修改.cmd文件中RAML1的首地址为0x009001，同时将长度修改为0x002FFFF，重新编译、链接后，.map文件中.text段的相关信息如图4. 1所示，其中.text段长度和原来不同，推测是每次编译、链接后本来长度便会不一（由C语言转换到汇编语言可能存在多种转换方式）。

|  |
| --- |
| 图4. 1 修改.cmd后重新编译链接的.map文件部分内容 |

# 5 实验总结

## 5.1实验中遇到的问题及解决方法

1.找不到“小锤子”编译链接选项。

在一开始测试实验箱时，找不到“小锤子”编译链接（Build）选项。经过摸索后发现是已经进入了调试（Debug）模式（但至于为什么一开始便进入调试模式，这个问题我还无法回答），回到编辑（Edit）模式后可以进行编译、链接。

2.无法在工程中添加“28335\_RAM\_lnk.cmd”文件。

添加“28335\_RAM\_lnk.cmd”文件，报错无法添加，修改添加方式（link或copy）都无法解决，后来发现是工程文件中已经存在了28335\_RAM\_lnk.cmd，将其删除后可以正常添加。

3.编译时报极多错误。

编译链接时发现错误极多，查看报错信息后发现文件路径乱码，推测是是添加工程文件时因为选择了“link”方式，而该些文件在桌面中文目录下。后来将工程文件创建在英文目录下，同时添加文件方式选择“copy”。再进行编译时，错误变少，但仍然有错，错误提示为找不到XXX文件。反复核对后发现是添加工程文件时，由于讲义中的要求是“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、DSP2833x\_Headers\_nonBIOS.cmd，添加到当前工程中”，于是我们只添加了这些文件，事实上该目录下还有“header”等文件夹，将一系列文件全部添加后，编译链接可以通过。

4.graph图形工具绘制波形杂乱

在使用graph工具绘图时，得到的波形杂乱无章。摸索相关选项后发现，在使用graph工具时输入的属性参数并没有在graph界面得到体现，即点击图5. 1中标出的选项，弹出的属性对话框中的参数仍然是初始值。在对话框中修改为正确的参数并确认，可以得到正确的波形。（但事实上，第二次实验时，使用graph工具时输入的正确属性参数却可以直接产生正确波形，而不需要点击图5. 1中的选项再次修改了，推测是因为第一次实验使用graph工具时CCS 5运行了较长的时间，CCS 5中的缓存等相关内容影响了graph工具的使用）

|  |
| --- |
| 图5. 1 graph工具部分选项 |

## 5.2实验心得体会

事实上在本学期的《电子信息工程课程设计》实验中，也接触过DSP实验箱和CCS，但那时更多的是修改C语言程序并烧录，而不是像此次实验一样深入学习DSP调试。这使得我对DSP的调试有了更深刻的认识。

CCS 5软件的界面是全英文，这使得我们在摸索一些功能时出现了一些困难，例如由于不小心关闭了“Project Explorer”栏，在添加工程文件这一步，又需要在“窗口”中找到显示“Project Explorer”的选项；在调试界面下，变量观察窗口没有显示，需要调出变量观察窗口；设置断点动作等等。这些卡壳的地方解决起来并不难、也知道解决的方法，只是在英文界面下查找起来不是十分的方便。这就要求我们对使用的开发环境比较熟悉，才能更快地“对症下药”。

在使用各种工具过程中，如果出现结果和预想不一致的情况，往往查找问题会很花费时间，而问题出现的位置又常常出乎意料。例如在使用graph工具时，开始输入的参数对照实验讲义是无误的，但始终出不来正确的波形。经过不断摸索该工具下的各个选项，才发现需要进一步修改属性参数。而同样是使用graph工具，第二次实验中，一开始直接输入参数，又能够正常显示波形，不需要再次修改参数了。这种软件中“迷”的地方，常常让人不知所措。

在本次使用中，进行了许多调试步骤，如查看变量地址及内容、查看子程序地址、设置断点动作等等，结合工程.map文件及.cmd文件，使得我对DSP的程序运行原理有了更深刻的了解，对DSP开发有了更进一步的认识，也为接下来DSP编程打下了基础，希望自己能够在之后的实验中合理运用本实验中学到的内容。