

**DSP应用技术实验**

**DSP开发基础实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 作者 | ： | 龚天韵 | 学号 | ： | 9181040G0620 |
| 同组人 | ： | 沈伟 | 学号 | ： | 9181040G0726 |
| 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2班 | | | |
| 组号 | ： | A2 | | | |
| 题目 | ： | DSP应用技术实验 | | | |
|  |  | DSP开发基础实验报告 | | | |
| 指导者 | ： | 李彧晟 | | | |

2021 年 4 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。

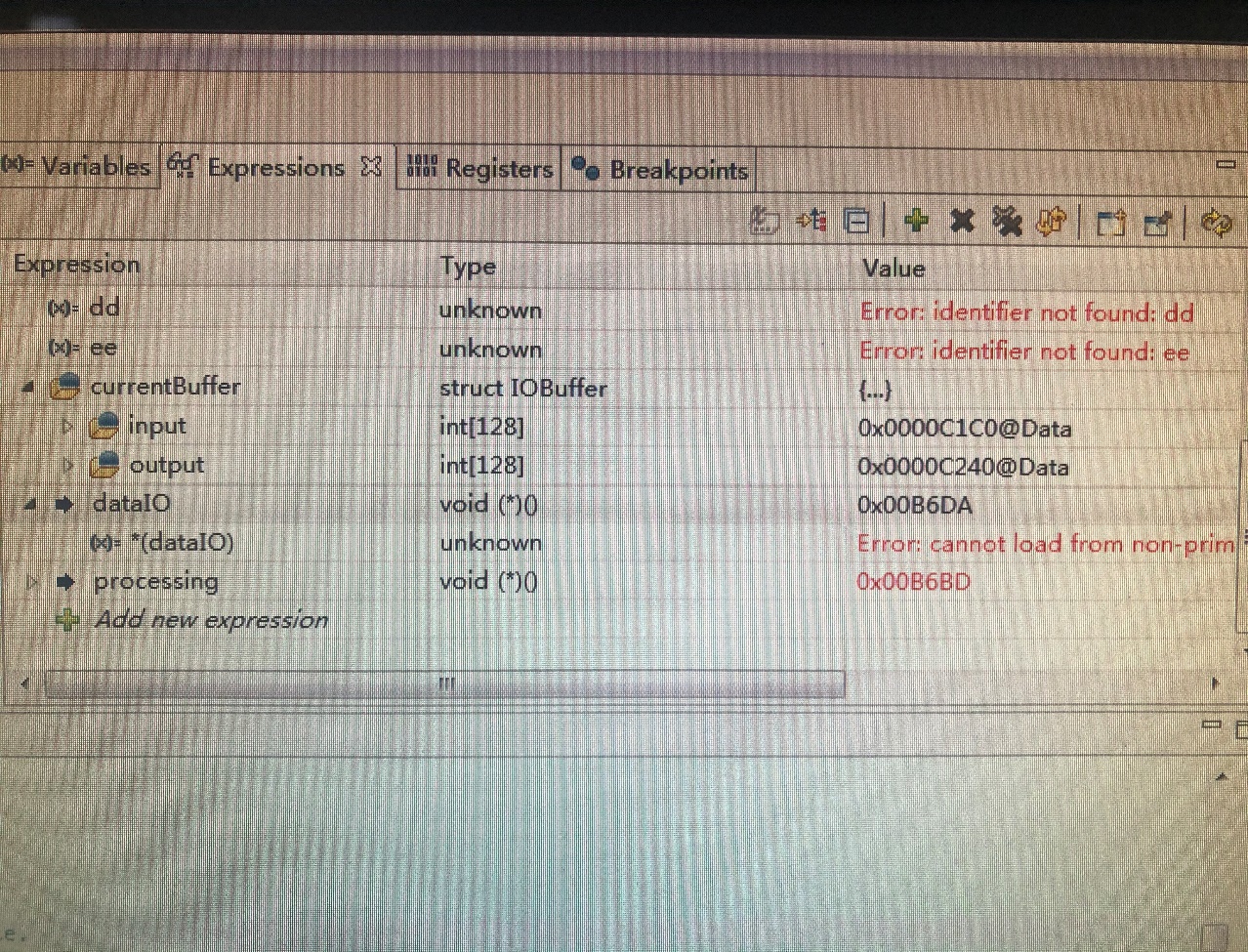
4. 对项目工程进行编译、链接（Build Project），进入调试（Debug）界面并运行程序（Resume）。

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，说明实验箱正常。

|  |
| --- |
|  |

## 3.2 C程序基础调试

1. 添加结构体变量至观察窗口： 选中变量currentBuffer.input，点击右键，在 下 拉 菜 单 中 点 击 “ Add Watch Expression…”即可添加变量至观察窗口来观察变量的类型、内容和地址。将变量 currentBuffer.output添加至观察窗口，同时将 dataIO( )到变量窗口，查看该子程序 的入口地址。观察窗口截图如图1所示：

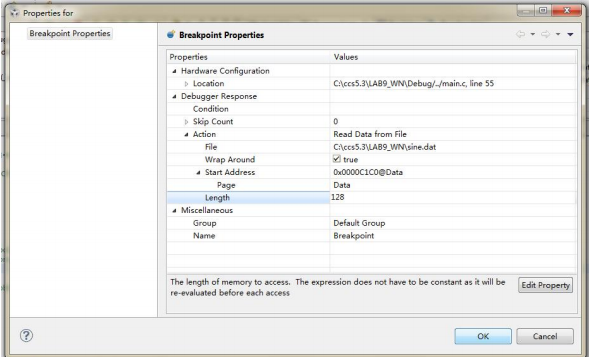


**图 1 变量观察窗口**

通 过 此 操 作 可 以 观 察 到 ， 变 量 currentBuffer.input 所 在 存 储 器 地 址 为 0x0000C1C0@Data，变量 currentBuffer.output 所在存储器地址为 0x0000C240@Data，子程 序 dataIO( )的地址为 0x00B6DA。记录下两个变量和子程序的地址，以供后续操作查看数据 绘制图像使用。

2. 设置断点，关联输入文件：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（0x0000C1C0@Data），数据长度为 128，点击“OK”，如图2所示：

|  |
| --- |
|  |



|  |
| --- |
| 图2.断点及关联数据设置 |

3. 图形显示数据空间：打开图形显示功能，在主菜单的点击“Tools→Graph→single time”。设置各项参数，设置起始地址为变量 currentBuffer.input 的起始地址 0x0000C1C0@Data，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，图4所示：

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

|  |
| --- |
| **图4 设置断点并关联文件后currentBuffer.output的时域波形** |

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。

|  |
| --- |
|  |

# 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

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

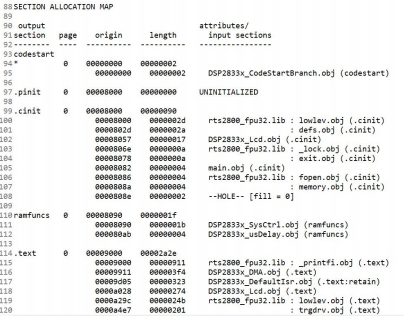
**表****1.子程序入口地址与结构体存储地址**

## 4.2 显示缓冲存储器中的波形

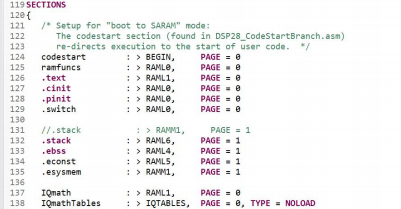
如图3，图4所示

## 4.3 查看.map文件信息

MAP 文件大概分为文件头、内存配置、段映射、全局符号四部分。打开 MAP 文件至 section location map 部分，即可知道该部分程序中所有的段实际映射的起始地址与实际长度。如 图 5 所示。联系 cmd 文件中 section 指令即可了解该段所在地址。如图 6 所示：



**图5.段地址部分截图**



**图6.sections指令内容**

根据程序内容得出表2：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

**表2.map文件各段信息**

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()子程序在.text段。

## 4.4 查看及修改.cmd文件

MAP 文件大概分为文件头、内存配置、段映射、全局符号四部分。内存配置与 CMD 文件 中的 MEMORY 指令关联，在 CMD 文件中定义的程序与数据区间定义，在该部分均可以找到对 应，与 CMD 文件不同的时，在 MAP 文件中加入了一个实际使用的区间，即在程序中实际用到 的空间长度。段映射部分与 CMD 文件中的 SECTION 指令关联，在该部分程序中所有的段实际 映射的起始地址与实际长度均有详细说明。可以具体到程序中#pragma 指定的段和各个单独 文件产生的 OBJ 文件。 修改 cmd 配置文件，在进行断点求地址操作，发现地址产生变化。映射也发生了改变。 例如将 ebss 段命令志向 RAML5 空间，该段的起始地址将变为 0x00D000。

# 5 实验总结

## 5.1实验中遇到的问题及解决方法

1.编译时显示无法连接，且出现大量编译错误。

编译链接时发现错误极多，且显示无法与仿真器连接，首先我们推测是由于没有关掉之前默认存在的工程项目导致工作环境被占用，在关掉前面同学打开的工程项目并重启软件后还是显示如上问题。在查看报错信息后，我们发现所有错误均出现在header文件中，且所有header文件内工程文件都显示无法连接。我们检查了头文件的关键路径后发现关键路径重复，推测是在给工程项目赋予关键路径前由于错误操作已经先给header文件赋予了关键路径导致路径重复，由于路径重叠导致连接错误，在删去原header文件重新建立并赋予关键路径后编译成功。

2.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，检查sine.data 文件，观察其中数据确是正弦波样点的数据，仔细检查操作步骤后发现，是在 debug之后忘记进行 Resume 操作，程序没有完全执行，改正之后就得到了正确波形。

|  |
| --- |
| **图7.graph工具部分选项**  3.重新编译时graph选项变灰无法绘图  在使用graph工具绘图时，发现在重新编译之后graph图标变灰无法再次点击绘图，在上网查询了有关资料后发现是由于工作环境被占用导致，根据网上的解决办法删掉了workspace文件夹和其它几个与CCS相关的文件夹，然后再通过file->switch workspace重新设定原来的文件夹signal time，之后重新进入软件便可使用，在使用过程中发现关闭程序重新打开graph图标也会恢复，推测是释放了工作环境导致。 |

## 5.2实验心得体会

在本次实验之前，由于专选课程和科技竞赛等其他方面的需求我对CCS软件有过少量的接触和了解，但应用到DSP中还是第一次。由于我一直不大习惯CCS软件的工作环境，加上版本比较古老，第一次实验中还是遇到了各种各样奇奇怪怪的问题，即使是重新熟悉界面也花了我一番功夫。此外，由于要使用之前没有使用的过的很多新鲜功能，我们的操作上也出现了相当多不必要的失误，这直接导致我们的实验进度推进缓慢。但好在在同学和助教老师的帮助下，以及上网查阅了相关资料的情况下，我们最终完成了第一个实验，了解了CCS在DSP上的实际应用，并掌握了一部分入门知识，通过第一次实验我们对DSP的调式和CCS的使用有了更为深入的了解，在之后的学习中也将继续加深这种了解的程度。