



南京理工大学

NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

DSP 应用技术实验

任意信号发生器实验报告

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 : 第二组 B4
题 目 : DSP 应用技术实验
任意信号发生器实验报告
指 导 者 : 李彧晟

2019 年 11 月

目录

1 实验目的.....	1
2 实验仪器.....	1
2.1 实验仪器清单.....	1
2.2 硬件连接示意图.....	1
3 实验步骤及现象	1
3.1 程序流程图.....	1
3.2 检查设备并启动开发环境.....	2
3.3 编写线性调频信号产生代码.....	2
3.3.1 函数参数求解.....	2
3.3.2 数据定标.....	2
3.3.3 线性调频信号查找表产生代码.....	2
3.4 建立工程并运行、调试程序.....	2
4 实验结果及思考题回答	3
4.1 子程序与主程序入口实际地址.....	3
4.2 波形数据空间地址及图形.....	4
4.3 改变段地址分配.....	4
4.4 调整线性调频信号输出周期.....	4
5 实验总结.....	5
5.1 实验中遇到的问题及解决方法.....	5
5.2 实验心得体会.....	6

1 实验目的

1. 熟悉 DSP 硬件开发平台；
2. 熟悉 TI DSP 软件集成开发环境；
3. 学习 DSP 程序的编程开发；
4. 熟悉工程代码产生方法；
5. 熟悉 DSP 代码调试基本方法。

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. 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,SRAM 与 DSP 连接示意图如图 2.2 所示。

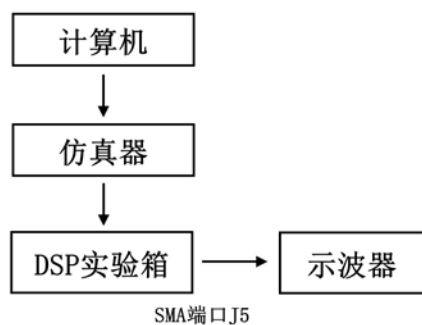


图 2.1 硬件连接示意图

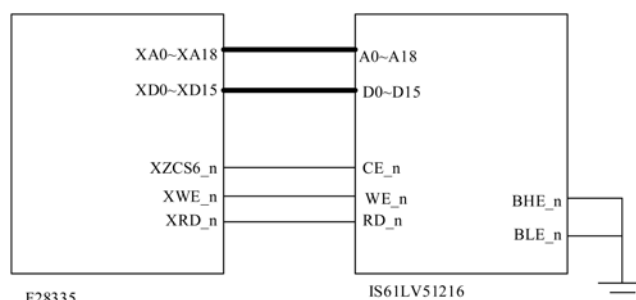


图 2.2 SRAM 与 DSP 连接示意图

3 实验步骤及现象

3.1 程序流程图

结合实验要求，程序流程大致为：通过 DSP 实时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。于是，程序流程图如图 3.1 所示。

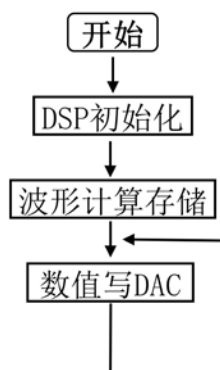


图 3.1 任意波形发生器程序流程图

3.2 检查设备并启动开发环境

检查仿真器、F28335 DSP 教学实验箱、计算机之间的连接是否正确。确认无误后开启电源，并在计算机上启动开发环境。

3.3 编写线性调频信号产生代码

3.3.1 函数参数求解

要求产生的线性调频信号为：

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

其中调制斜率 $K=39062$ ， 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。相当于将 $t \in [-0.0128, 0.0128]$ 映射到 $N \in [1, 1024]$ 。构建 $AN + B = t$ 形式的二元一次方程组如下：

$$\begin{cases} A + B = -0.0128 \\ 1024A + B = 0.0128 \end{cases}$$

解得：

$$\begin{cases} A = \frac{0.0256}{1024} \\ B = -0.0128 \end{cases}$$

于是，产生线性调频信号代码的主体部分为：

$$s(N) = \cos\left(K\pi\left(-0.0128 + \frac{0.0256}{1024}N\right)^2\right)$$

3.3.2 数据定标

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

浮点数 X_F 与定点数 X_D 的转换关系可表示为：

$$\text{定点数 } X_D = [X_F \times 2^Q]$$

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

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 $|max|$ ，尽量使下式成立：

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

即， $Q = 15 - n$ 。由于正余弦函数的值域为 $[-1, 1]$ ，则 Q 至多取 15。实验中，我们在产生查找表时进行了幅度增大，在传送给 DA 使又进行了一步幅度增大。

3.3.3 线性调频信号查找表产生代码

综合以上内容，线性调频信号查找表的产生代码如下：

```
67 for(i=0;i<1024;i++)
68 {
69     *(RamAddr+i) = (int)((cos(Pi*39062*(-0.0128+(0.0256/1024)*i)*(-0.0128+(0.0256/1024)*i))*4096));
70     /**(RamAddr+i) = (int)((sin(2*Pi*i/1024))*2048);*/
71 }
```

3.4 建立工程并运行、调试程序

编译链接工程。查看存储空间中的时域波形是否正确，如图 3.3 所示，正确后运行程序，连接教学实验箱 SMA 输出端口 J5 至示波器，查看输出的线性调频

信号，如图 3.3 所示。

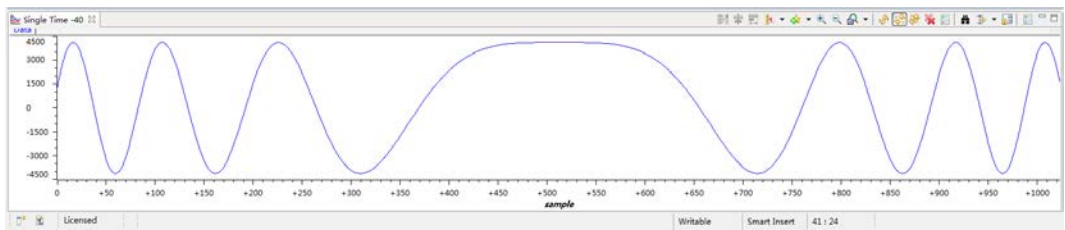


图 3.2 存储空间内的线性调频信号

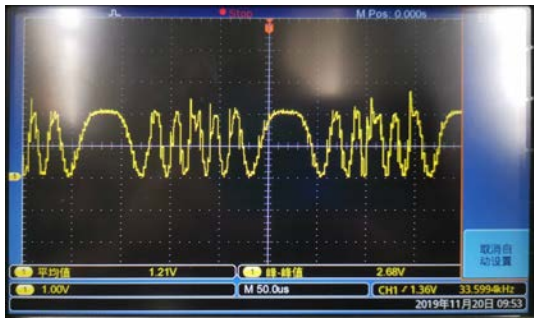


图 3.3 示波器上的线性调频信号

4 实验结果及思考题回答

4.1 子程序与主程序入口实际地址

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。

如图 4.1 所示，可汇总得到表 4.1。

Expression	Type	Value	Address
▷ RamAddr	int *	0x0000F000	0x0000C024@Data
▷ main	void (*)0	0x009AFB	
▷ init_AD9747	void (*)0	0x009BD9	
▷ mcbbsp_write	void (*)(unsigned short)	0x009BD0	
▷ init_zone7	void (*)0	0x009B5D	
✚ Add new expression			

图 4.1 各子程序入口地址

表 4.1 各子程序入口实际地址

名称	地址
main	0x009AFB
init_AD9747	0x009BD9
mcbbsp_write	0x009BD0
init_zone7	0x009B5D

通过查看.map 文件的相关内容，如图 4.2 所示，可知以上程序均在 RAML1 中。

10	MEMORY CONFIGURATION							
11								
12		name	origin	length	used	unused	attr	fill
13		-----	-----	-----	-----	-----	-----	-----
14	PAGE 0:							
15	BEGIN		00000000	00000002	00000002	00000000	RWIX	
16	RAMM0		00000050	000003b0	00000000	000003b0	RWIX	
17	RAML0		00008000	00001000	00000060	00000fa0	RWIX	
18	RAML1		00009000	00003000	000013dd	00001c23	RWIX	

图 4.2 .map 文件部分存储器设置的相关代码

4.2 波形数据空间地址及图形

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。

如图 4.1 所示，波形数据存储地址为 0x0000F000。RamAddr 中保存的波形如图 3.2 所示，示波器上的波形如图 3.3 所示。

4.3 改变段地址分配

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。

与实验9中的情况类似，如图4.3到图4.4所示，修改.cmd文件配置命令后，对应的段地址发生改变。而当不同段之间发生重合时，将会报错。

```

81 MEMORY
82 {
83     PAGE 0 :
84         /* BEGIN is used for the "boot to SARAM" bootloader mode */
85
86         BEGIN      : origin = 0x000000, length = 0x000002 /* Boot to M0 will go here
87         RAMM0      : origin = 0x000050, length = 0x0003B0
88         RAML0      : origin = 0x008000, length = 0x001000
89         RAML1      : origin = 0x009001, length = 0x003000
90         //RAML2    : origin = 0x00A000, length = 0x001000
91         //RAML3    : origin = 0x00B000, length = 0x001000

```

图 4.3 修改.cmd 文件

```

110 .text      0      00009001  000013dd
111           00009001  000003f4  DSP2833x_DMA.obj (.text)
112           000093f5  00000323  DSP2833x_DefaultIsr.obj (.text:retain)
113           00009718  00000274  DSP2833x_Lcd.obj (.text)
114           0000998c  00000170  DSP2833x_Mcbsp.obj (.text)
115           00009afc  0000014e  main.obj (.text)
116           00009c4a  00000130  DSP2833x_ECan.obj (.text)
117           00009d7a  000000fa  DSP2833x_Xintf.obj (.text)

```

图 4.4 .map 文件的变化情况

4.4 调整线性调频信号输出周期

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。

产生正弦信号的代码如下：

```

67     for(i=0;i<1024;i++)
68     {
69         /*(RamAddr+i) = (int)((cos(Pi*39062*(-0.0128+(0.0256/1024)*i)*(-0.0128+(0.0256/1024)*i))*4096));
70         *(RamAddr+i) = (int)((sin(2*Pi*i/1024))*2048));
71     }

```

在存储空间内的波形如图 4.5 所示。

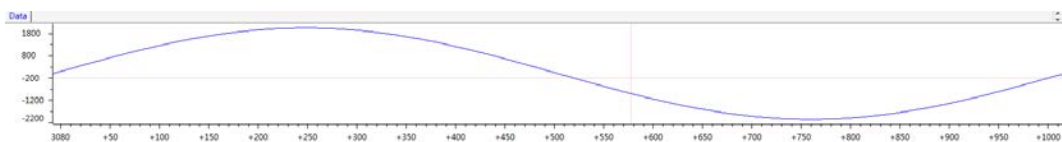


图 4.5 存储空间中的正弦波形

而要求不改变波形数值表中的数据，则构造数据存储的程序不需要改变，而在最后从波形数据表中传输到 DAC 时，设定一个频率控制字，每隔几个点输出，实现改变输出正弦信号的频率。对应的代码如下：

```

74      while(1)
75      {int freqK=1;
76        for(i=0;i<1024/freqK;i++)
77        {
78          /*Da_out = (unsigned int)((*(RamAddr+freqK*i)<<4)+ 0x8000; //左移4位
79          /*Da_out = (unsigned int)((*(RamAddr+freqK*i)<<3)+ 0x8000; //左移3位
80          *Da_out = (unsigned int)((*(RamAddr+freqK*i)<<2)+ 0x8000); //左移2位
81          /*Da_out = (unsigned int)((*(RamAddr+freqK*i)<<3);
82        }
83      }
84    }
85  }
```

在示波器上可以看到波形如图 4.6 到图 4.8 所示。

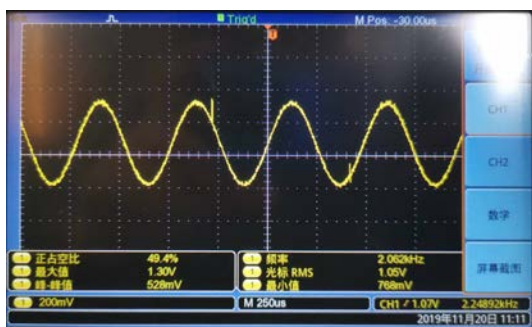


图 4.6 频率控制字为 1 时的示波器波形

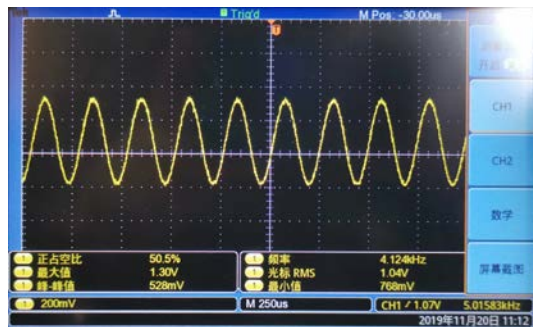


图 4.7 频率控制字为 2 时的示波器波形

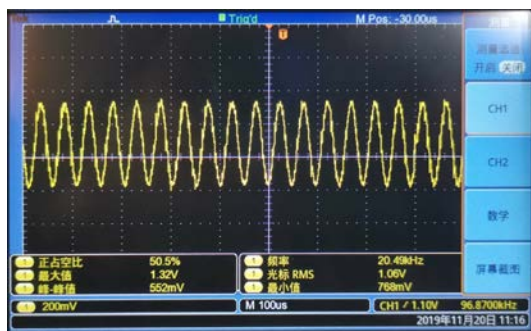


图 4.8 频率控制字为 10 时的示波器波形

5 实验总结

5.1 实验中遇到的问题及解决方法

1. 改变频率控制字时输出波形不正确

在修改正弦波频率控制字时，输出的波形不正确。在反复阅读程序后，发现是没有修改循环的个数。由于查找表中，只有 1024 各值，修改频率控制字而又不修改循环终止个数，会导致输出的波形后段结果不正确。修改循环终止个数为

1024/频率控制字 后，可以实现正常。

2.graph 图形工具无法绘制波形

在使用 graph 工具绘图时，输入属性参数后无法显示图形（但示波器上有输出），查看输入的参数并没有问题。后来在同学的提示下，重启 CCS 多次后，可以绘制图形。推测是 CCS 5 开发环境不稳定的原因。

5.2 实验心得体会

有了第一次实验的基础，我对于实验箱的操作以及 CCS 开发环境的操作大致有了了解。一些基本的操作，比如地址的查看等等，可以比较快速的完成。

而这次的实验，是对 DSP 实验箱中 DA 的部分进行配置。有关 DDS 的内容我之前的在 EDA 实验中有所接触，但之前的实验更多的是原理性的理解，对于实际用 DDS 原理来修改代码实现操作上的波形改变还是首次。在本次实验中，通过频率控制字来实现波形频率的改变。

在实验中，我们修改频率控制字并编译链接后，使用 graph 工具查看 RamAddr 内的波形，发现仍然是单个周期的正弦波，而示波器上的频率发生变化，说明查找表中的内容没有发生改变，是通过修改抽取数据的间隔来实现频率改变的。这种方法，当频率控制字过大时，会因为周期数据不多而出现失真的现象。

同时，本次实验也让我明白有时软硬件设备是存在“不稳定”情况的，例如在编译链接后，调试运行时，有时示波器上没有输出波形。退出调试并重启实验箱后，用同样的代码可以在示波器上看到波形。因此，当出现与理论情况不符的现象时，也要学会考虑偶然误差的影响。