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：	周鹏	学　　号：	9181040G0740
同 组 人：	杨霄宇	学　　号：	9161040G0736
同 组 人：	许昕荣	学　　号：	9161040G1038
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 B6		
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		

2021 年 4 月

目录

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

2 实验仪器

2.1 实验仪器清单

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

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。

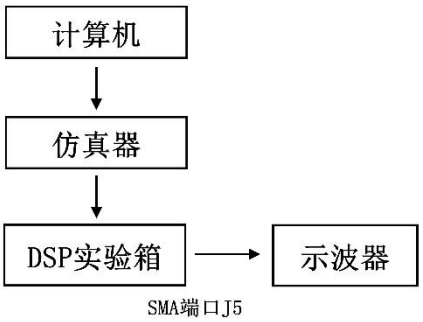


图 2.1 硬件连接示意图

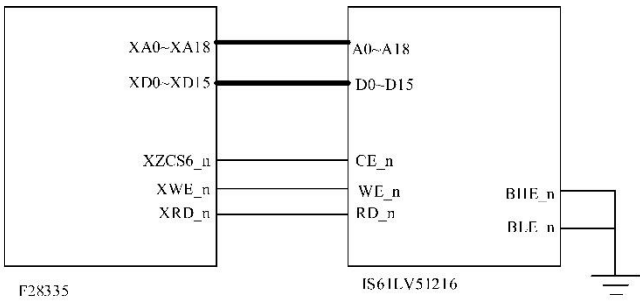


图 2.2 SRAM 与 DSP 连接示意图

3 实验步骤及现象

3.1 程序流程图

在 TMS320F32028335 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。

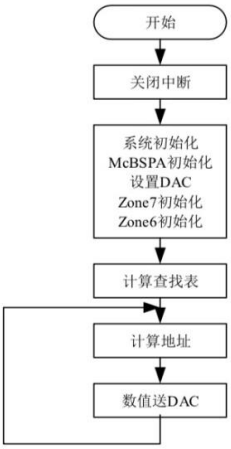


图 3.1 任意波形发生器程序流程图

3.2 检查设备并启动开发环境

1.将 TMS320F28335 教学实验箱连接至计算机，将 J5 输出口接至示波器。

2.点击桌面 CCS5 快捷方式，启动 CCS 集成开发环境。

3.添加工程文件“LAB 10”至目录，完成各项设置后运行程序，

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：

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

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 XF 与定点数 XD 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：

```
64    int i;
65
66    for( i=-N/2;i<N/2;i++)
67    {
68        *(RamAddr+i+N/2) = (cos(Pi*K*i*i*T/N*T/N)*2048);
69    }
70
71    //*zone7Addr = 66;
72
73    while(1)
74    {
75        for(i=0;i<N/1;i++)
76        {
77            *Da_out = (unsigned int)((*(RamAddr+1*i))<<4)+ 0x8000;//±ä4±
78            //*Da_out = (unsigned int)((*(RamAddr+1*i))<<3);
79        }
80    }
81
82}
```

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。

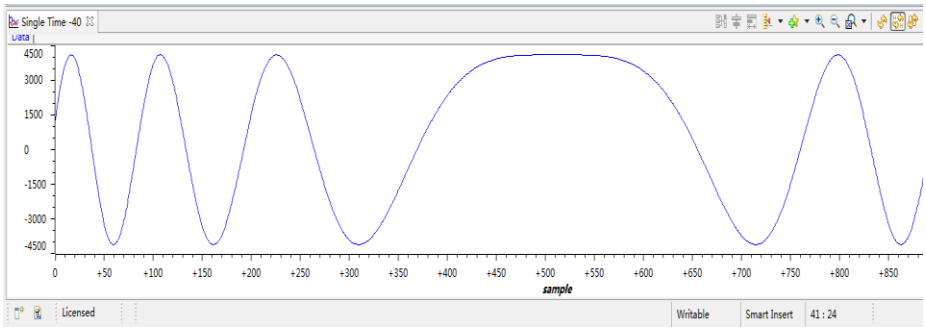


图 3.2 存储空间内的线性调频信号

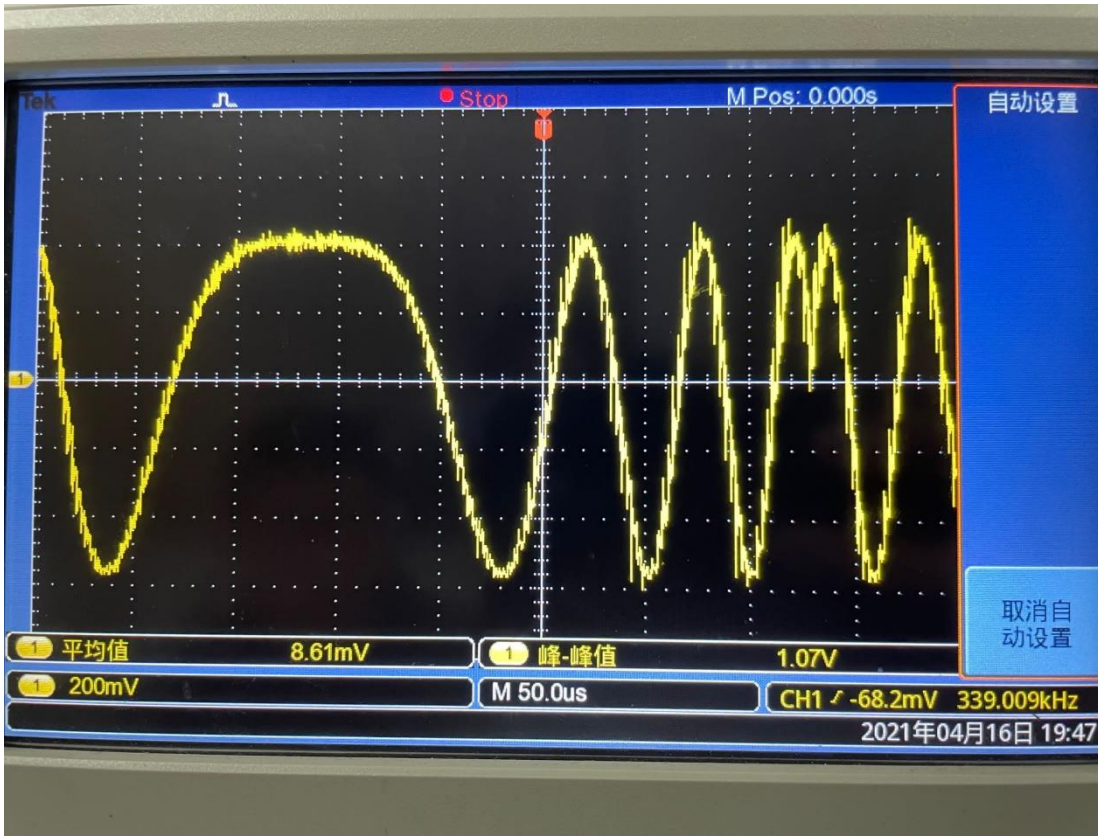


图 3.3 示波器上的线性调频信号

4 实验结果及思考题回答

4.1 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。

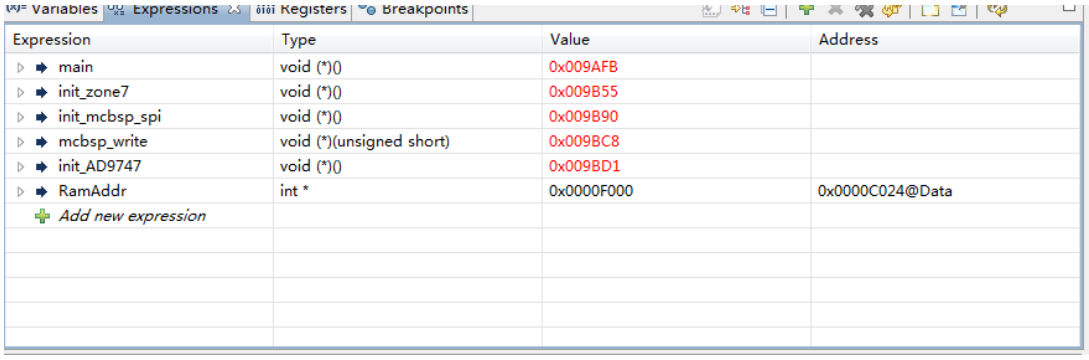


图 4.1 各子程序入口地址

表 4.1 各子程序入口实际地址

名称	地址
main	0x009AFB
init_AD9747	0x009BCB
mcbsp_write	0x009BC2
init_zone7	0x009B4F

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。

```
MEMORY CONFIGURATION

      name            origin    length      used     unused   attr    fill
---------------------- --------  ---------  --------  --------  ----  --------
PAGE 0:
  BEGIN               00000000  00000002  00000002  00000000  RWIX
  RAMM0               00000050  000003b0  00000000  000003b0  RWIX
  RAML0               00008000  00001000  000000af  00000f51  RWIX
  RAML1               00009000  00003000  00002a2e  000005d2  RWIX
  ZONE7A              00200000  0000fc00  00000000  0000fc00  RWIX
  CSM_RSVD            0033ff80  00000076  00000000  00000076  RWIX
  CSM_PWL             0033fff8  00000008  00000000  00000008  RWIX
  ADC_CAL             00380080  00000009  00000007  00000002  RWIX
  IQTABLES            003fe000  00000b50  00000000  00000b50  RWIX
  IQTABLES2           003feb50  0000008c  00000000  0000008c  RWIX
  FPUTABLES           003febdc  000006a0  00000000  000006a0  RWIX
  BOOTROM             003ff27c  00000d44  00000000  00000d44  RWIX
  RESET               003fffc0  00000002  00000000  00000002  RWIX
```

图 4.2 .map 文件部分存储器设置的相关代码

4.2 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 0x000F000。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 比较波形数据保存不同存储空间区域（DPS 内部 RAM 和外扩 SRAM），对系统实现的影响。

通过改变波形数据存储地址 0x0000F000 至 0x00100000，将存储空间从 RAM 改为 SRAM，发现由于存储器读取的速度不同导致示波器上显示波形的频率不同。

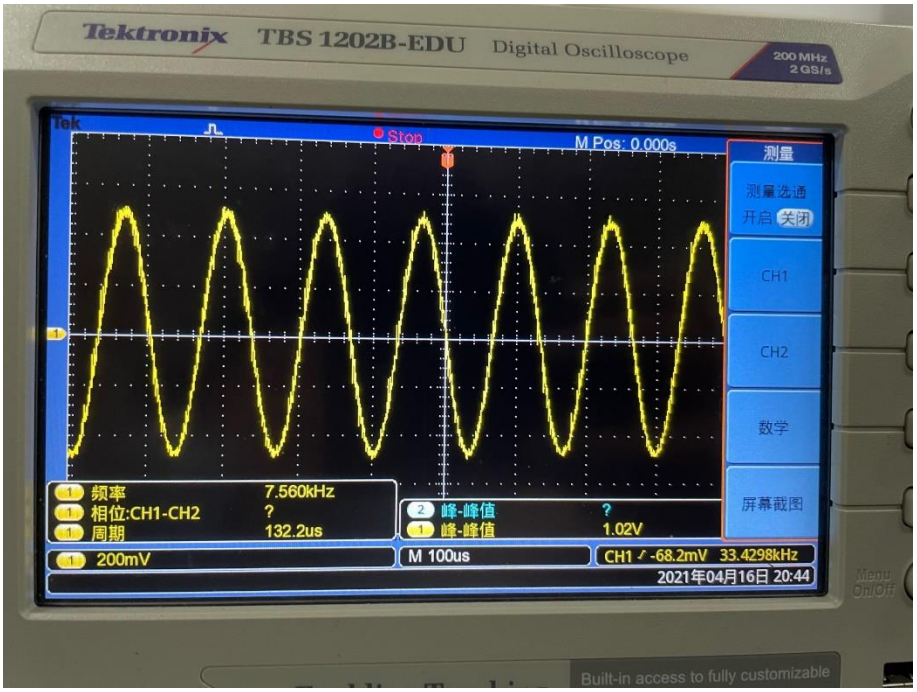


图 4. 3 RAM 地址下的显示波形

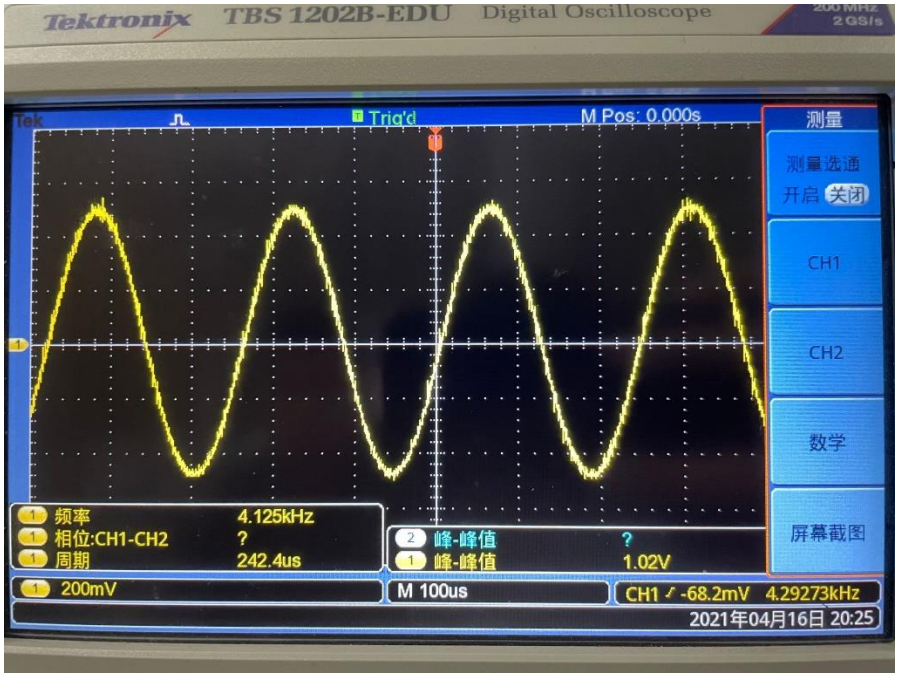
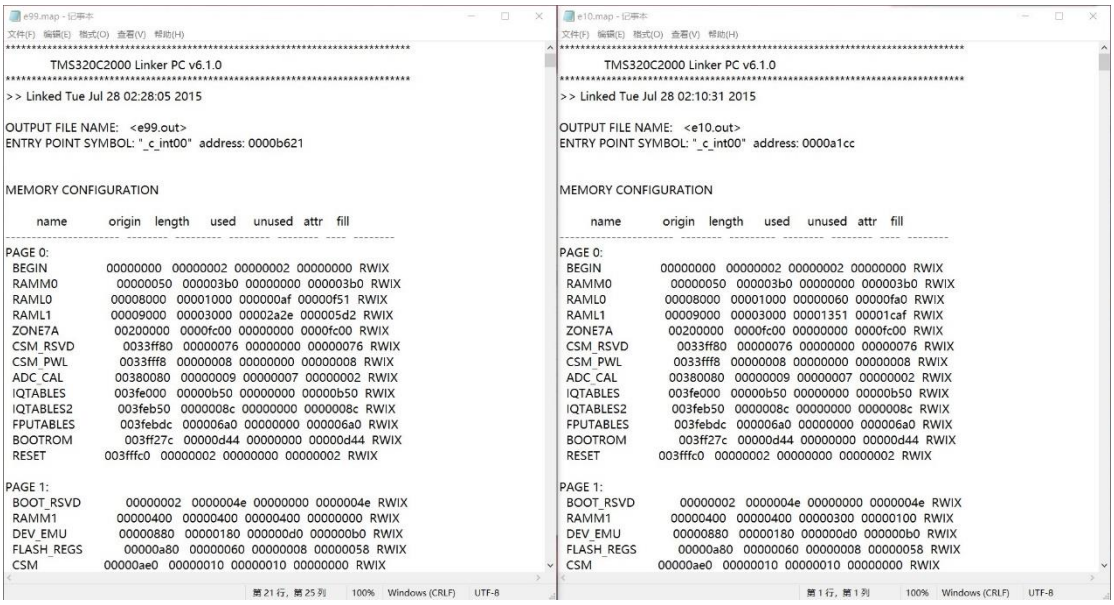


图 4. 4 SRAM 地址下的显示波形

4.4 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



e99.map - 记事本

```
        TMS320C2000 Linker PC v6.1.0
>> Linked Tue Jul 28 02:28:05 2015

OUTPUT FILE NAME:   <e99.out>
ENTRY POINT SYMBOL: "_c_int00"  address: 0000b621


MEMORY CONFIGURATION

         name        origin   length     used    unused   attr    fill
----------------- --------  ---------  --------  --------  ----  --------
PAGE 0:
  BEGIN           00000000  00000002  00000002  00000000  RWIX
  RAMM0           00000050  000003b0  00000000  000003b0  RWIX
  RAML0           00008000  00001000  000000af  00000f51  RWIX
  RAML1           00009000  00003000  00002a2e  000005d2  RWIX
  ZONE7A          00200000  0000fc00  00000000  0000fc00  RWIX
  CSM_RSVD        0033ff80  00000076  00000000  00000076  RWIX
  CSM_PWL         0033fff8  00000008  00000000  00000008  RWIX
  ADC_CAL         00380080  00000009  00000007  00000002  RWIX
  IQTABLES        003fe000  00000b50  00000000  00000b50  RWIX
  IQTABLES2       003feb50  0000008c  00000000  0000008c  RWIX
  FPUTABLES       003febdc  000006a0  00000000  000006a0  RWIX
  BOOTROM         003ff27c  00000d44  00000000  00000d44  RWIX
  RESET           003fffc0  00000002  00000000  00000002  RWIX

PAGE 1:
  BOOT_RSVD       00000002  0000004e  00000000  0000004e  RWIX
  RAMM1           00000400  00000400  00000400  00000000  RWIX
  DEV_EMU         00000880  00000180  000000d0  000000b0  RWIX
  FLASH_REGS      00000a80  00000060  00000008  00000058  RWIX
  CSM             00000ae0  00000010  00000010  00000000  RWIX
```

e10.map - 记事本

```
        TMS320C2000 Linker PC v6.1.0
>> Linked Tue Jul 28 02:10:31 2015

OUTPUT FILE NAME:   <e10.out>
ENTRY POINT SYMBOL: "_c_int00"  address: 0000a1cc


MEMORY CONFIGURATION

         name        origin   length     used    unused   attr    fill
----------------- --------  ---------  --------  --------  ----  --------
PAGE 0:
  BEGIN           00000000  00000002  00000002  00000000  RWIX
  RAMM0           00000050  000003b0  00000000  000003b0  RWIX
  RAML0           00008000  00001000  00000060  00000fa0  RWIX
  RAML1           00009000  00003000  00001351  00001caf  RWIX
  ZONE7A          00200000  0000fc00  00000000  0000fc00  RWIX
  CSM_RSVD        0033ff80  00000076  00000000  00000076  RWIX
  CSM_PWL         0033fff8  00000008  00000000  00000008  RWIX
  ADC_CAL         00380080  00000009  00000007  00000002  RWIX
  IQTABLES        003fe000  00000b50  00000000  00000b50  RWIX
  IQTABLES2       003feb50  0000008c  00000000  0000008c  RWIX
  FPUTABLES       003febdc  000006a0  00000000  000006a0  RWIX
  BOOTROM         003ff27c  00000d44  00000000  00000d44  RWIX
  RESET           003fffc0  00000002  00000000  00000002  RWIX

PAGE 1:
  BOOT_RSVD       00000002  0000004e  00000000  0000004e  RWIX
  RAMM1           00000400  00000400  00000300  00000100  RWIX
  DEV_EMU         00000880  00000180  000000d0  000000b0  RWIX
  FLASH_REGS      00000a80  00000060  00000008  00000058  RWIX
  CSM             00000ae0  00000010  00000010  00000000  RWIX
```

图 4. 5 实验 9 与实验 10.map 文件对比

如图 4.5，两实验.map 文件在 RAML0　RAML1　RAMM1　FLASH_REGS 寄存器上地址不一样。

4.5 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. 到**错误!未找到引用源。**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。

```
MEMORY
{
PAGE 0 :
   /* BEGIN is used for the "boot to SARAM" bootloader mode      */

   BEGIN       : origin = 0x000000, length = 0x000002     /* Boot to M0 will go here
   RAMM0       : origin = 0x000050, length = 0x0003B0
   RAML0       : origin = 0x008000, length = 0x001000
   RAML1       : origin = 0x009001, length = 0x003000
   //RAML2       : origin = 0x00A000, length = 0x001000
  // RAML3       : origin = 0x00B000, length = 0x001000
```

图 4. 6 修改.cmd 文件

4.6 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：

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

在存储空间内的波形如图 4. 所示。

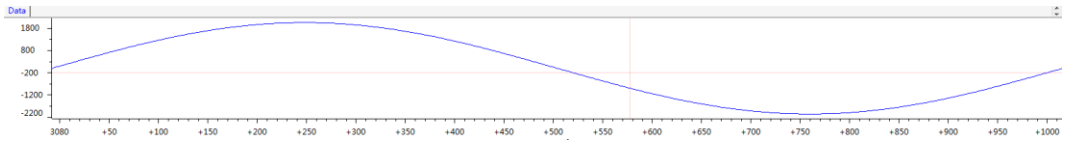


图 4.7 存储空间中的正弦波形

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. 到图 4. 10 所示。

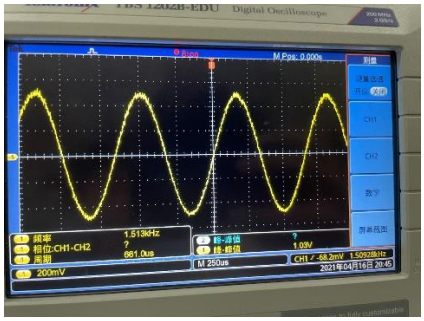


图 4.8 频率控制字为 1 时的示波器波形

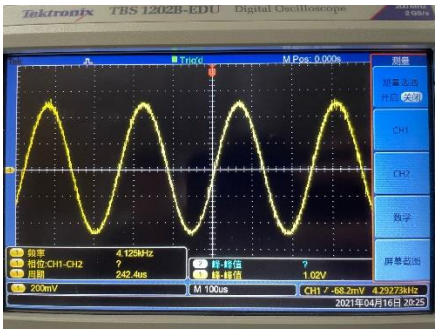


图 4.9 频率控制字为 2 时的示波器波形

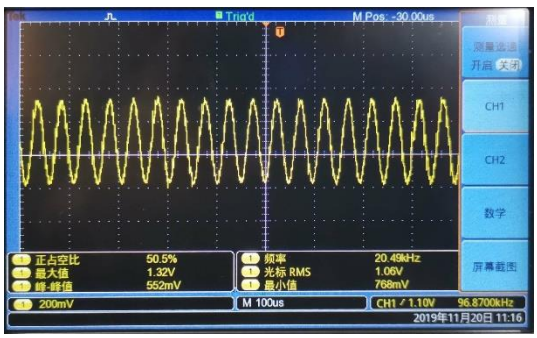


图 4.10 频率控制字为 10 时的示波器波形

5 实验总结

5.1 实验中遇到的问题及解决方法

1.编译错误

电脑中自带的实验 10 范例一直出现编译错误并显示无法连接，在尝试了各种办法后均无法解决，最后换了仿真器才得以成功，猜测是由于仿真器硬件损坏导致连接失败并不断报错。

2.仿真后无法实现功能

发现给的示例程序有问题，问题出现在左移位数出了问题，应该左移 2 位，示例程序为左移 3 位，导致整体数值超出存储范围。

3.波形锯齿比较严重

可能由于频率较高放大倍数较大所以波形锯齿状严重，在修改采样点为 2048 个之后锯齿变小得以更好的观察。

4.改变频率控制字时输出波形不对

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 个值，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。

5. graph 图形工具绘制波形错误

在实验一中，graph 工具的地址是断点地址，在实验步骤中给出了具体的一步步引导，但实验二并没有给出，因此出现了不知道地址怎么设的问题。一开始时选择了 0x00000000 作为地址，发现波形完全不对，仔细理解程序后发现应为 RamAddr 的地址 0x00F000，最终出现正确波形。

5.2 实验心得体会

本次实验的目的在于实现任意信号的生成与输出，根本上在于对 DSP 实验箱中 DA 部分的配置。有关于 DDS 的内容在上学期的数字系统综合设计中已经有过学习，因此实验的原理和例程代码都很容易理解，需要的只是将平台从 FPGA 搬移到 DSP 上实现。

但是虽然对频率控制字和采样都能够理解，我们仍花了一定的时间才弄清了频率控制字对波形的控制的重要意义，同时明白了频率控制字对循环周期的重要影响，在使用 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。

最后，通过这次实验，对 DSP 芯片的内部结构有了更深的了解，比如我们通过实践，证实了我们对外部存储器读取速度比片内存储器慢的猜想，熟悉了 DSP 芯片存储器配置的具体方法，收获很多。