南京理工大学

NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

DPS 应用技术实验

任意信号发生器

学　　院：电子工程与光电技术学院

专　　业：电子信息工程

组　　别：第二组 B3

姓　　名：刘孝雨

学　　号：9161040G0424

指导老师：李彧晟

2019 年 11 月 21 日

目 录

1 实验目的.....1
2 实验仪器.....1
3 实验步骤.....1
3.1 程序流程.....1
3.1.1 数据的定标.....2
3.1.2 相关实验硬件资源.....3
3.2 设计方法.....4
3.3 实验效果.....5
3.4 实验要求回答.....9
4 实验问题与解决方案.....13
4.1.1 通过改变步进改变频率出现杂波.....13
5 实验总结.....13

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

计算机，C2000 DSP 教学实验箱，XDS510 USB 仿真器，示波器

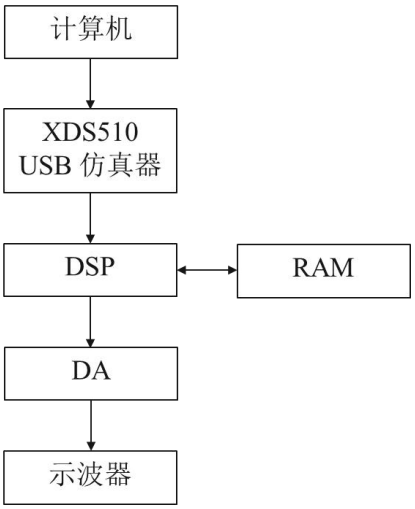

图 2. 1 硬件连接示意图

3 实验步骤

3.1 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 3.1 所示。

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。

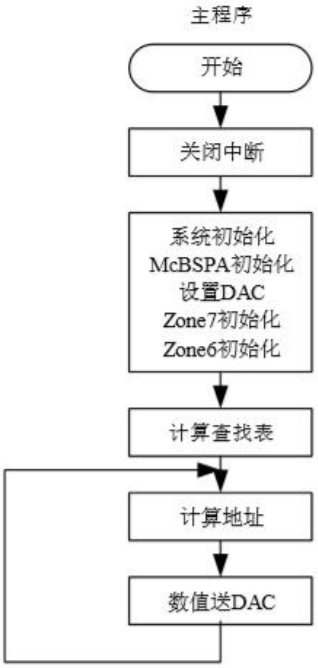

图 3.1 任意信号发生程序流程图

3.1.1 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： 定点数 $X_D = \lfloor X_F \times 2^Q \rfloor$

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。

3.1.2 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。

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 3.1 所示。

地址范围	存储器	等待时间	备注
0x08,0000 ~ 0x08,0FFF	双端口 RAM	至少 2 等待	占 ZONE2
0x10,0000 ~ 0x13,FFFF	SRAM	至少 2 等待	占 ZONE6
0x14,0000 ~ 0x14,FFFF	FIFO	至少 4 等待	占 ZONE6

表格 1 外扩存储器地址映射

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 4.2 所示。

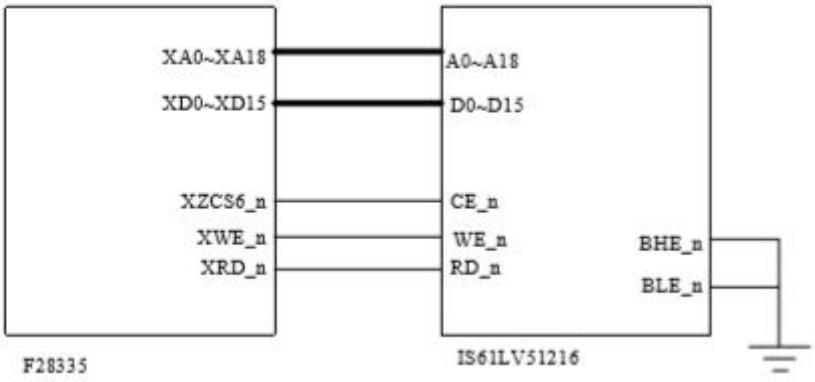

图 3.2 SRAM 与 DSP 连接示意图

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。

3.2 设计方法

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：

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

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。

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。

3.3 实验效果

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。

通过将主程序和各个子程序的名称加入到变量窗口中可以看到主程序和不同的子程序对应的地址，其中主程序的地址为 0x00009AFB。

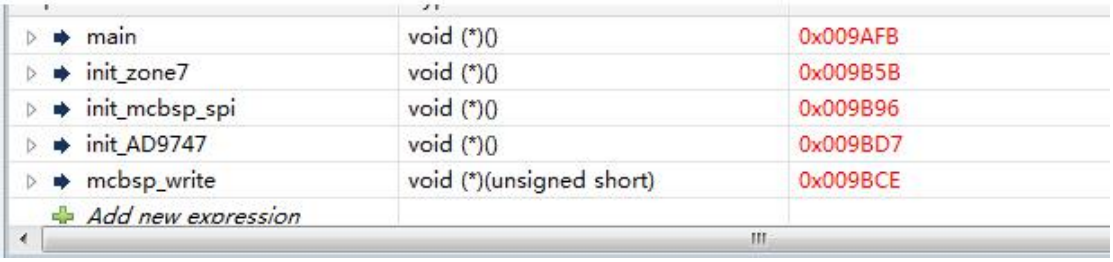

图 3.3 各子程序入口地址

通过查看.map 的文件内容，RAML1 起始地址为 0x00009000，长度为 0x00003000，可以看到以上程序的地址均在 PAGE0 的 RAML1。

```
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  00001333  00001ccd  RWIX
19   ZONE7A               00200000  0000fc00  00000000  0000fc00  RWIX
20   CSM_RSVD             0033ff80  00000076  00000000  00000076  RWIX
```

图 3.4 存储器起始地址

```
.text      0    00009000    00001335
                  00009000     000003f4     DSP2833x_DMA.obj (.text)
                  000093f4     00000323     DSP2833x_DefaultIsr.obj (.text:retain)
                  00009717     00000274     DSP2833x_Lcd.obj (.text)
                  0000998b     00000170     DSP2833x_Mcbsp.obj (.text)
                  00009afb     00000150     main.obj (.text)
                  00009c4b     00000130     DSP2833x_ECan.obj (.text)
                  00009d7b     000000fa     DSP2833x_Xintf.obj (.text)
                  00009e75     000000f3     DSP2833x_SysCtrl.obj (.text)
```

图 3.5 函数所在段

观察.cmd 文件同样也可以验证。

```
117
118 SECTIONS
119 {
120    /* Setup for "boot to SARAM" mode:
121       The codestart section (found in DSP28_CodeStartBranch.asm)
122       re-directs execution to the start of user code.  */
123    codestart        : > BEGIN,     PAGE = 0
124    ramfuncs         : > RAML0,     PAGE = 0
125    .text            : > RAML1,     PAGE = 0
126    .cinit           : > RAML0,     PAGE = 0
127    .pinit           : > RAML0,     PAGE = 0
128    .switch          : > RAML0,     PAGE = 0
```

图 3.6 .cmd 文件

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。

从图中可以看到波形的数据存储地址为 0x0000C024，存储有 1024 个整型数据。

Da_out	unsigned int *	0x00200400	0x0000C028@Data
RamAddr	int *	0x0000F000	0x0000C024@Data

图 3.7 RamAddr 地址截图

线性调频信号产生代码如下所示：

```
for(i=0;i<1024;i++)

        //*(RamAddr+i) = (int)((cos(39062*Pi*(i/(delta_t))*(i/(delta_t)))*8192));
//    *(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128+(0.0256/1024)*i)*(-0.0128+(0.0256/1024)*i))*4096));
    *(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128+(0.0256/1024)*i)*(-0.0128+(0.0256/1024)*i))*2048));//线性调频
    //*(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128)*(-0.0128))*4096));
```

图 3.8 线性调频信号产生代码截图

利用图形工具，将图形的起始地址设为 RamAddr，即波形数据的起始存储地址，在图中显示 1024 个点的线性调频信号波形，图形如下

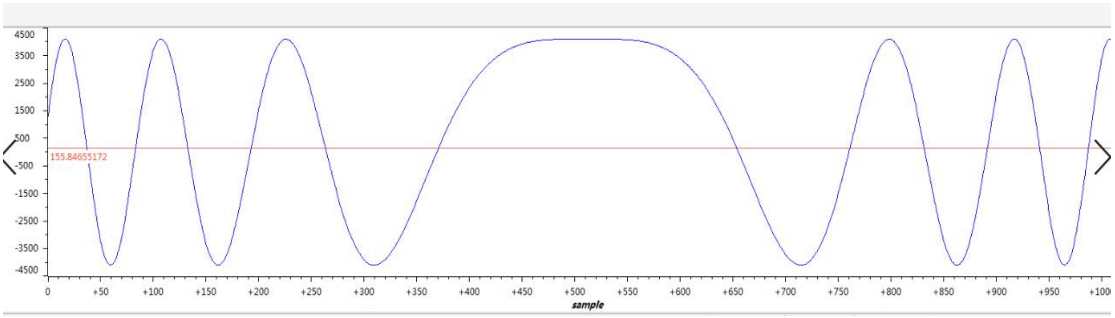

图 3.9 产生的线性调频信号仿真图

在示波器中可以看到该线性调频信号，图形如下所示：

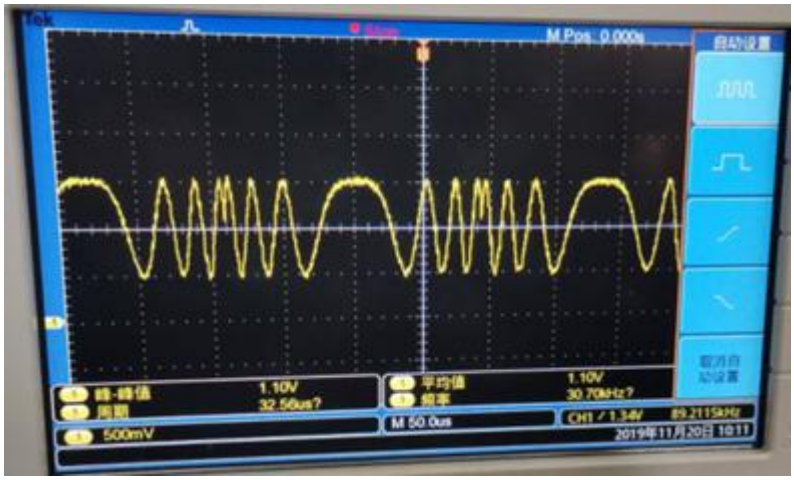

图 3.10 产生的线性调频信号实际输出图

3.改变正弦信号频率编程实现，在示波器上验证，要求记录改变参数以及实测频率。

正弦信号产生代码为，产生正弦信号周期为 192us

```
for(i=0;i<1024;i++)

            //*(RamAddr+i) = (int)((cos(39062*Pi*(i/(delta_t))*(i/(delta_t)))*81
//    *(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128+(0.0256/1024)*i)*(-0.0128+(0.025
  *(RamAddr+i) = (int)(sin(2*Pi/1024*i)*2048);
    //*(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128)*(-0.0128))*4096));
```

图 3.11 正弦信号产生代码

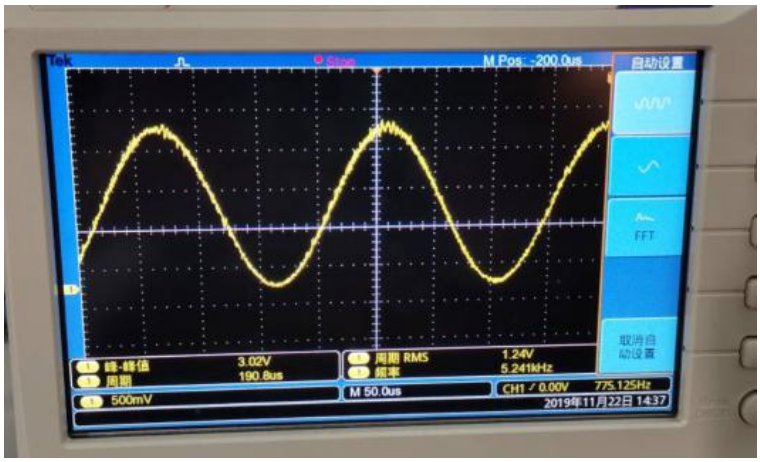

图 3.12 正弦信号 192us

修改源程序可增大频率，将频率增大一倍，周期缩小为 96us

```
//减低幅度
    for(i=0;i<1024;i++)

                //*(RamAddr+i) = (int)((cos(39062*Pi*(i/(delta_t))*(i/(delta_t)))*8192));
//    *(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128+(0.0256/1024)*i)*(-0.0128+(0.0256/1024)
    *(RamAddr+i) = (int)(sin(2*2*Pi/1024*i)*2048);
        //*(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128)*(-0.0128))*4096));

```

图 3.13 正弦信号 96us

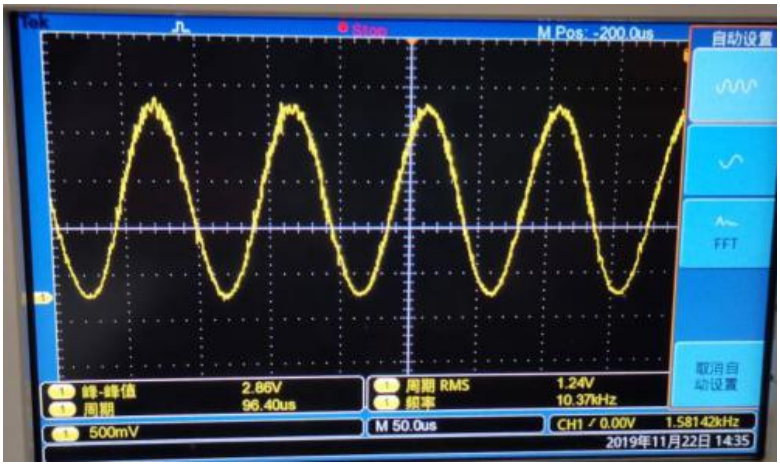

图 3.14 正弦信号 96us

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。

3.4 实验要求回答

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。

思考 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

如图 3.15 所示，将 RAML1 的地址 0x00009000 修改为 0x00009001，程序正常执行，由图 3.16 可以发现 main 函数首地址同样增加 0x00000001。

```
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 t
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 3. 15 RAML1 地址修改

Expression	Type	Value
main	void (*)()	0x009AFC
init_zone7	void (*)()	0x009B5C
init_mcbsp_spi	void (*)()	0x009B97
init_AD9747	void (*)()	0x009BD8
mcbsp_write	void (*)(unsigned short)	0x009BCF
RamAddr	int *	0x0000F000

图 3. 16 main 函数首地址变化

多次修改.cmd 文件中段的地址，发现当段地址与其他段地址重合时，会出现错误。如图 3.17 所示，修改 RAML1 地址和 RAML0 地址相同，出现错误。

```
83 PAGE 0 :
84    /* BEGIN is used for the "boot to SARAM" bootloader mode
85
86    BEGIN      : origin = 0x000000, length = 0x000002     /* Bo
87    RAMM0      : origin = 0x000050, length = 0x0003B0
88    RAML0      : origin = 0x008000, length = 0x001000
89    RAML1      : origin = 0x008000, length = 0x003000
90    //RAML2      : origin = 0x00A000, length = 0x001000
91   // RAML3      : origin = 0x008000, length = 0x001000
92    ZONE7A     : origin = 0x200000, length = 0x00FC00    /* XI
```

图 3. 17 段地址重合

思考 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，调整正弦信号的输出周期。

正弦信号产生代码如图 3.18 所示，产生正弦信号周期为 192us。

```
    for(i=0;i<1024;i++)

                //*(RamAddr+i) = (int)((cos(39062*Pi*(i/(delta_t))*(i/(delta_t)))*81
    //    *(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128+(0.0256/1024)*i)*(-0.0128+(0.025
      *(RamAddr+i) = (int)(sin(2*Pi/1024*i)*2048);
        //*(RamAddr+i) = (int)((cos(39062*Pi*(-0.0128)*(-0.0128))*4096));

     while(1)
     {
         for(i=0;i<1024;i++)
         {
             //*Da_out = (unsigned int)((*(RamAddr+1*i))<<3) + 0x8000;//左移4位
             //*Da_out = (unsigned int)((*(RamAddr+1*i))<<3);
             //*Da_out = (unsigned int)((*(RamAddr+1*i)))+ 0x8000;
             *Da_out = (unsigned int)((*(RamAddr+2*i))<<4) + 0x8000;
         }
     }
```

图 3. 18 正弦信号产生程序

图 3.19 正弦信号 192us

通过修改步进的方式可以修改频率，将步进改为 2，周期变为原来 1/2，96us。

```
    //*zone7Addr = 66;
    while(1)
    {
        for(i=0;i<512;i++)
        {
            //*Da_out = (unsigned int)((*(RamAddr+1*i))<<3) + 0x8000;//左移4位
            //*Da_out = (unsigned int)((*(RamAddr+1*i))<<3);
            //*Da_out = (unsigned int)((*(RamAddr+1*i)))+ 0x8000;
            *Da_out = (unsigned int)((*(RamAddr+2*i))<<4) + 0x8000;
        }
    }

}
```

图 3.20 修改步进为 2 的程序

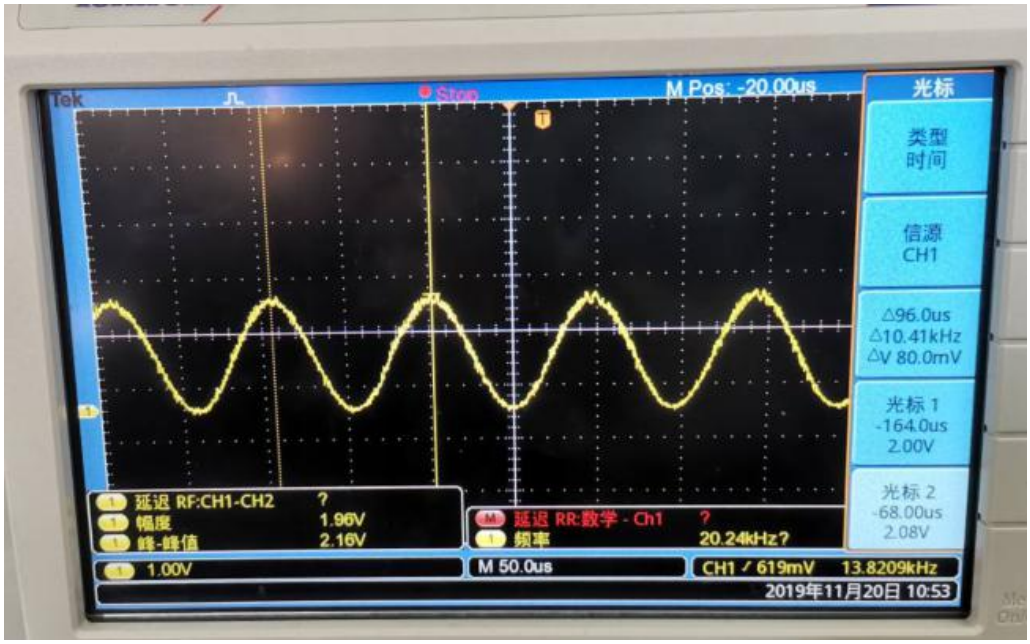

图 3.21 修改步进为 2，周期为 96us

将步进改为 4，周期变为原来 1/4，48us。

```
while(1)
{
    for(i=0;i<256;i++)
    {
        //*Da_out = (unsigned int)((*(RamAddr+1*i))<<3) + 0x8000;//左移4位
        //*Da_out = (unsigned int)((*(RamAddr+1*i))<<3);
        //*Da_out = (unsigned int)((*(RamAddr+1*i)))+ 0x8000;
        *Da_out = (unsigned int)((*(RamAddr+4*i))<<4) + 0x8000;
    }
}
```

图 3.22 修改步进为 4 的程序

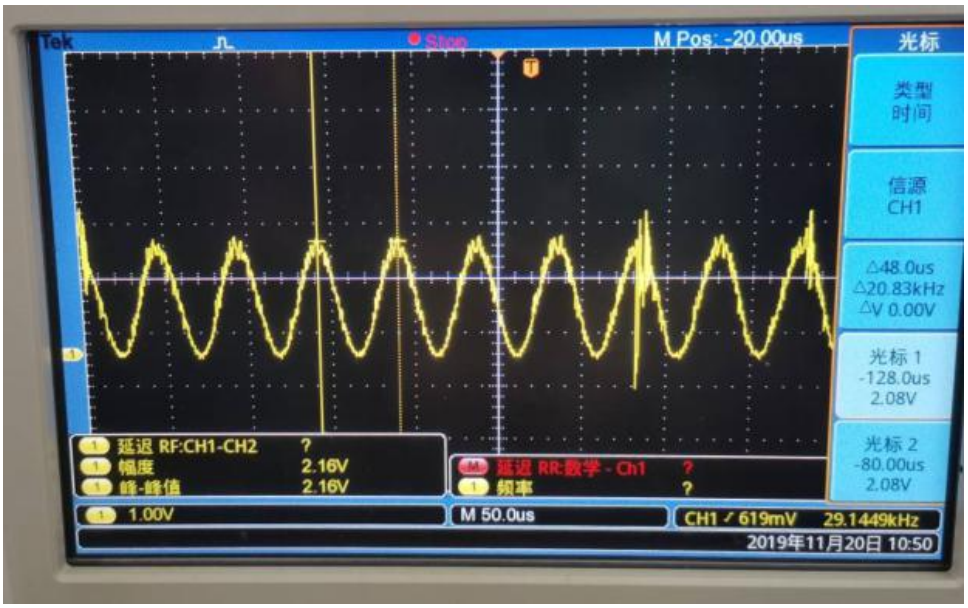

图 3.23 修改步进为 4，周期为 48us

4 实验问题与解决方案

4.1.1 通过改变步进改变频率出现杂波

本次实验在修改正弦波频率时，最开始只修改了步进，没有修改循环次数，由于正弦表中只存在 1024 个数据，循环次数为 1024，步进改为 2 之后，无法采到 1024 之后的点，导致输出波形后半个周期出现杂波。同时修改循环次数及步进后解决了这个问题。

5 实验总结

经过了第一次实验，我对于整个实验箱的操作以及 CCS 软件的操作有了大致的了解，所以在进行本次实验的时候我能够快速地了解程序并能够快速地运行。通过此次实验，进一步了解了 DSP 硬件中 DA 的配置以及相应的如何使用，除此之外，由于 DDS 实验在之前的实验中有所接触，所以对于其原理还是有所了解的，但是由于当时的基础知识不够扎

实，所以做实验的时候也没有很透彻地理解原理，在 本次实验中，我在上次实验的基础上对于频率控制字有了更进一步的了解，并且能够在现有的例程中添加频率控制字，从而实现能够改变该变量来改变最后输出波形。