



南京理工大学

NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

电子工程与光电技术学院

实验报告

课程名称: DSP 应用技术

实验名称: DSP 数据采集

班 级: 9151040G02

姓 名: 傅 超

学 号: 9151040G0216

指导老师: 李彧晟

2018 年 11 月 25 日

目 录

1 实验目的	1
2 实验仪器	1
3 实验内容	1
4 实验准备	1
4.1 程序流程	1
4.2 DSP 初始化	2
4.3 模数转换器（ADC）	3
4.4 事件管理器	3
5 实验步骤	5
6 实验结果	6
7 实验感悟	13
7.1 实验中遇到的问题与解决方案	13
7.1.1 计算理论采样频率时总是和实际得到的采样频率有差距	13
7.1.2 输入信号与示波器显示信号、存储器中存储波形信号幅度存在差异	13
7.2 实验的收获与感受	14

1 实验目的

- 1、熟悉 DSP 的软硬件开发平台
- 2、掌握 TMS320F2812 的 ADC 外设的使用
- 3、熟悉 TMS320F2812 的中断的设置
- 4、掌握代码调试的基本方法

2 实验仪器

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

3 实验内容

建立工程，编写 DSP 的主程序，并对工程进行编译、链接，利用现有 DSP 平台实现数据的采集、存储以及模拟还原，通过图表以及示波器观察结果。

4 实验准备

4.1 程序流程

为实现 DSP 的数据采集存储以及模拟的还原，必须依赖于 ADC、DSP 以及 DAC 三大基本部件，而 TMS320F2812 芯片上集成了外设 ADC，因此实现该功能较为简单，数据采集的工作可以由 DSP 单独完成，只需要对相关外设进行配置。模拟还原由 DSP2000 实验箱中 DAC1（AD768）来完成。TMS320F2812 中的 ADC 外设与 DSP 的通信可以通过查询方式或中断方式，在此，我们采用 ADC 的中断功能实现数据的交换。TMS320F2812 中 ADC 的转换频率和采样频率可以独立设置，分别位于 ADC 外设模块和事件管理器模块中，因此要使 ADC 工作，必须掌握 ADC 外

设和事件管理器外设中的相关设置。

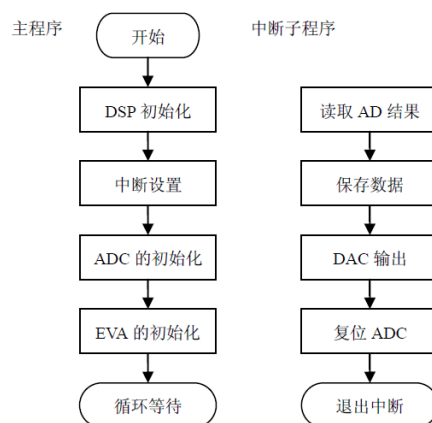


图 1：程序流程图

4.2 DSP 初始化

一般而言，DSP 要正常工作，必须首先设置时钟，时钟确定了 DSP 工作主频。

TMS320F2812 中时钟设置大致分为三个主要寄存器，它们分别是锁相环控制寄存器（PLLCR）、外设时钟使能控制寄存器（PCLKCR）和外设时钟预定标设置寄存器（HISPCP、LOSPCP）。

1. PLLCR 寄存器（地址@0x7021）

PLLCR 寄存器用于改变 PLL 的锁相环倍频值，输出 CLKIN 用于 DSP 内部的主频，控制 DSP 指令执行周期以及外设输入时钟。

2. PCLKCR 寄存器（地址@0x701C）

外设时钟使能控制器用于控制片上各种外设时钟的工作状态，禁止外设时钟能够有效降低 DSP 功耗。若程序中使用某外设功能，则必须使能该外设时钟。当该位置 1 时，相应的外设时钟被使能。

3. 外设时钟预定标设置寄存器

HISPCP（地址@0x701A）和 LOSPCP（地址@0x701B）寄存器分别用来配置高速和低速外设时钟。

4.3 模数转换器（ADC）

TMS320F2812 内部有一个 16 通道、采样精度为 12bit 的 ADC 模块，分别为事件管理器 A 和事件管理器 B 服务。这 16 通道可配置两个独立的 8 通道模块，具有同步采样和顺序采样模式，模拟输入范围 0~3V，最快转换时间为 80ns，具有多个触发源用于启动 AD 的转换，采用灵活的中断控制。

4.4 事件管理器

TMS320F2812 片内集成了两个事件管理器 EVA 和 EVB，他们具有完全相同的结构和功能，内部包含通用定时器、全比较/PWM 单元、捕获单元以及正交编码脉冲（QEP）电路。

每个事件管理器模块有两个通用定时器（GP Timer），在 GPTCONA/B 寄存器中可以定义 ADC 的启动触发信号由通用定时器的时间来产生，比如这些事件可以是下溢、比较匹配或周期匹配，下溢是指定时器计数器（TxCNT）的数值为 0，比较匹配是指 TxCNT 与比较寄存器（TxCMPR）中的数值相等，周期匹配是指 TxCNT 与周期寄存器（TxPR）中的数值相等。这一特性允许在没有 CPU 干涉的情况下，实现通用定时器事件和模数转换启动操作的同步。

每个通用定时器有 4 种可选的操作模式：

- 停止/保持模式；
- 连续递增计数模式；
- 定向的增/减计数模式；
- 连续增/减计数模式。

定时器控制寄存器 TxCON 中相应的模式位决定了通用定时器的计数模式。定时器的使能位为 TxCON[6]，可以使能或禁止定时器的计数操作。当定时器被禁止时，定时器的计数器操作将停止。当定时器被使能时，定时器将按照寄存器 TxCON 中的相应位（TxCON[12~11]）设定的计数模式并开始计数。

1.4.4 TMS320F281x 中断系统.

TMS320F281x 的外设中断扩展 (PIE) 单元通过少量中断输入信号的复用来扩展大量的中断源, PIE 单元支持多达 96 个独立的中断, 这些中断以 8 个为一组进行分类, 每组中的所有中断共用一个 CPU 级中断 (INT1~INT12)。96 个中断对应的中断向量表存储在专用 RAM 区域中。PIE 向量表用来存储系统中每个中断服务程序 (ISR) 的入口地址。一般来说, 在设备初始化时就要设置 PIE 向量表, 并可在程序执行期间根据需要对其进行更新。

在实验中, 当我们设置 $VMAP=1$ (ST1 寄存器的 bit3), $ENPIE=1$ (PIECTRL 寄存器的 bit0) 后, TMS320F2812 的中断向量表地址范围 0x000D00~0x000DFF。例如外设模块 ADC 使用的中断 ADCINT 向量地址为 0x000D4A。

TMS320F281x 的中断分为三个级别: 外设级, PIE 级和 CPU 级。每个片内外设的各个中断信号都具有自己的中断标志寄存器和中断使能寄存器, 例如 ADC 中的 INT ENA SEQ1 位就是 ADC 的中断使能位, 其标志为在 ADCST 中。PIE 单元将中断分为 12 组, 每组 8 个中断。一旦片内外设向 PIE 发出中断请求, 对应的外设中断标志寄存器 $PIEIFRx.y$ 就会被置位, 如果外设中断使能寄存器 $PIEIERx.y$ 为 1 (即被使能), 当前外设中断应答寄存 $PIEACKx.y$ 为 0, PIE 就会向 CPU 级发出中断请求。CPU 级中断接受到 PIE 的中断, 会立即置位 CPU 中断标志寄存器 IFR 相应位, 并判断 CPU 中断使能寄存器 IER 中相应位是否被使能, 以及全局中断 IMTN 是否被允许, 若满足条件, 则对应的中断向量表中的地址被加载到程序计数器 PC 中。

在退出中断服务程序时, 为确保下次中断服务能够被可靠的执行, 务必人工清除相应的中断标志位, 其标志位也分为三个级别, 分别是外设级中断应答标志, PIE 级中断应答标志 PIEACK 以及 CPU 级中断标志 IFR。

由此可见, 要想正确使用中断, 首先应该合理设置中断向量表, 在对应地址填入中断服务子程序的入口地址。其次, 必须对上述三个级别的中断作出正确的设置。比如实验中, 要想实现 CPU 利用中断方式读取 ADC 的采样数据, 必须使能 ADC 外设的中断, 其次使能外

设使能寄存器 $PIEIER1.6$, 保证中断发生时 $PIEACK1.6$ 位清零, 最后使能 CPU 中断使能寄存器 IER 中的 INT1, 以及全局中断使能位 INTM。这些工作必须在系统初始化时完成。退出中断服务程序前, 清除 ADCST 中的 INT SEQ1 以及相应的 $PIEACKx$ 。

5 实验步骤

1、设备检查

检查仿真器、C2000 DSP 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 CCS 2 (C2000) 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。在源程序中，通过对中断、ADC 外设以及事件管理通用时钟的设置，利用中断方式读取 ADC 的采样结果，并用 DAC 实现模拟信号的还原。在程序中，开辟一段数据空间，用于保存 ADC 的采样结果，要求保存 1024 点数据，且该空间的数据不断刷新。

4、建立工程

建立工程，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、加载程序

在主菜单下，选择“File → Load Program”，将程序下载到 DSP 内部。

6、连接外部电路

打开信号源，产生一个合适的频率，信号幅度控制在 $\pm 0.5V$ 以内，验证后将信号通过 INPUT1 接口输入到 DSP 中。

打开示波器，将 C2000 实验箱中的 OUT3 接口输出到示波器上，并正确设置。

7、调试程序

首先验证中断设置是否正确。可以在 ADC 中断服务程序的入口地址处添加断点，全速或者动画运行程序，检查程序计数器 PC 能否间隔性的停留在中断服务入口地址处。

8、运行程序

连接 C2000 实验箱 OUT3 输出口至示波器，调节示波器，观察信号的输出。可以实时的改变信号源的输入信号，示波器上显示的波形亦会随之变化。数据直通通道就是最简单的实时信号处理电路。

6 实验结果

1、根据范例程序，给出 ADC 的采样频率计算公式，修改 ADC 的采样频率并验证。

通过阅读例程，综合相关资料，可以得到 ADC 采样频率的计算公式如下。

$$f_s = \frac{150\text{MHz}}{2\text{TPS} \times 63}$$

改变信号源输入信号的频率，来观察示波器的输出。

当信号源输入频率为 500Hz 的正弦波时，示波器显示频率为 500.026Hz

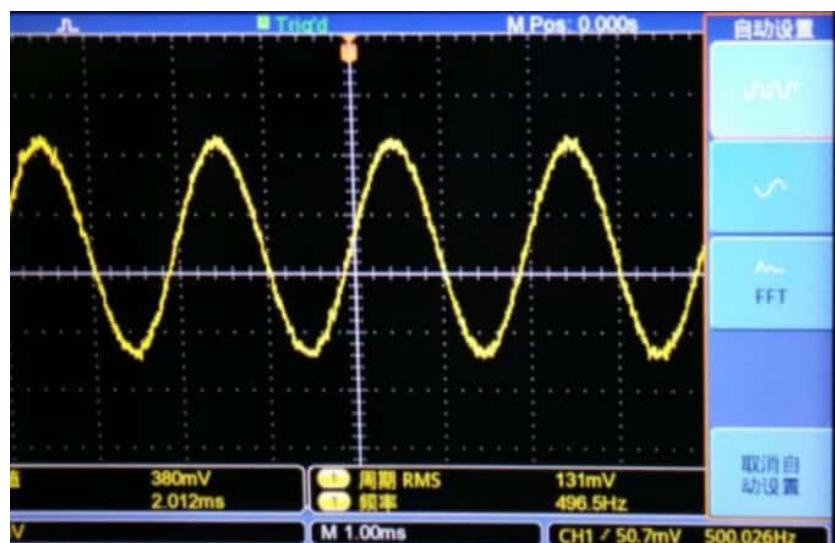


图 2：信号源频率为 500Hz 时示波器显示波形

当信号源输入频率为 1KHz 的正弦波时，示波器显示频率为 0.99KHz

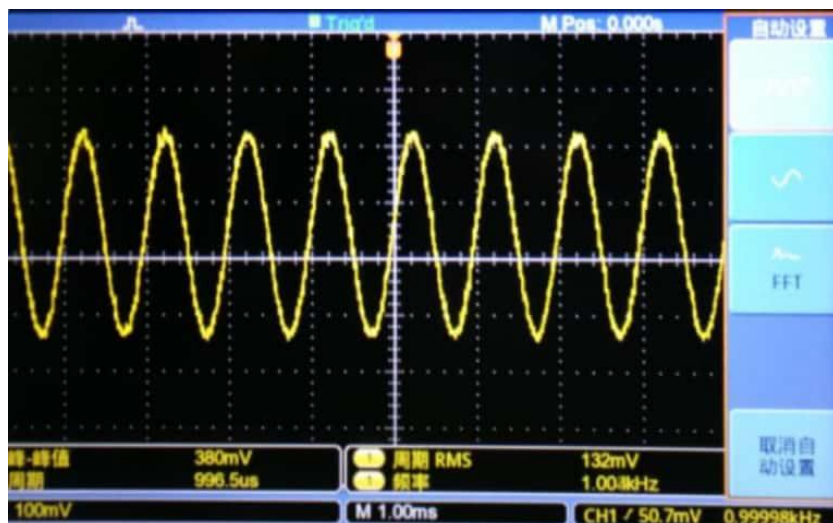


图 3：信号源频率为 1KHz 时示波器显示波形

当信号源输入频率为 2KHz 的正弦波时，示波器显示频率为 1.99KHz

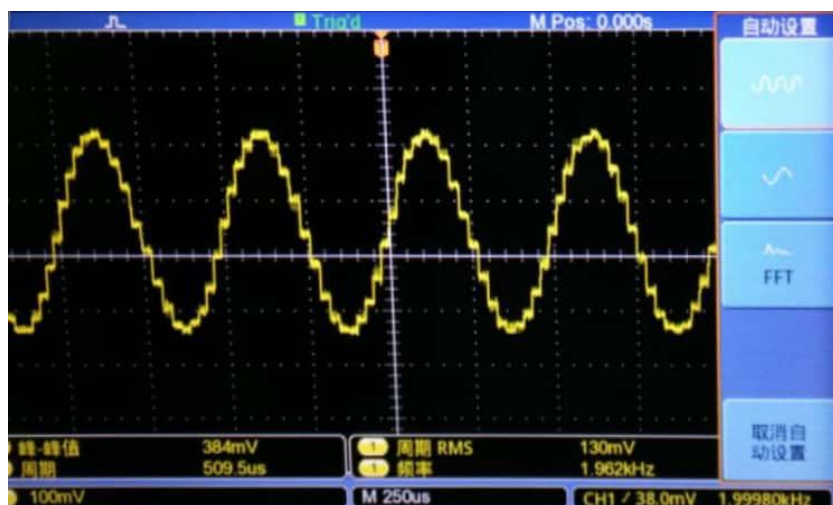


图 4：信号源频率为 2KHz 时示波器显示波形

当信号源输入频率为 2.5KHz 的正弦波时，示波器显示频率为 2.49KHz

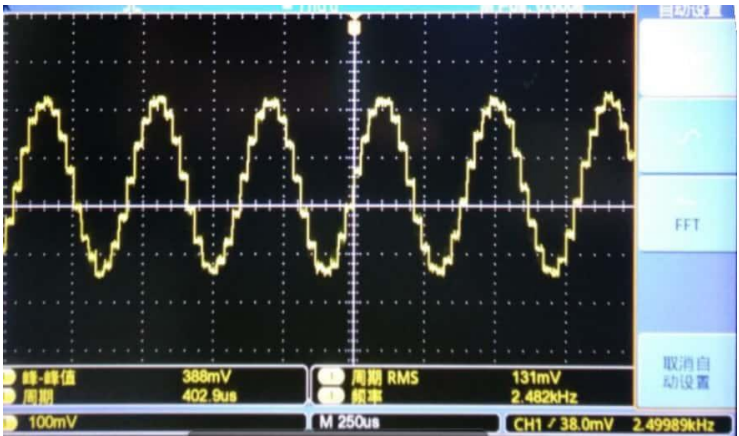


图 5：信号源频率为 2.5KHz 时示波器显示波形

可见信号经过 AD 和 DA 后，输出的信号频率与输入信号的频率很接近。在信号源频率大于 2KHz 以后，波形已经出现阶梯状失真，这是由于缺少滤波器进行滤波所致。

2、指出波形数据保存的空间地址，并以图形方式显示采集的信号波形，并保存，附在实验报告中。在实际操作时，将 ADC 采样得到的数据存储在 `adcResult` 变量中，通过查看该变量地址，具体的地址如下。

Name	Value	Type	Radix
DAOUT	0x00002900	unsigned int *	hex
(*DAOUT)	65535	unsigned int	unsigned
adcResult	0x003F9040	unsigned int...	hex

Watch Locals Watch 1

图 6：adcResult 地址截图

将其设为图形工具的数据起始地址，即可得到预想的波形。

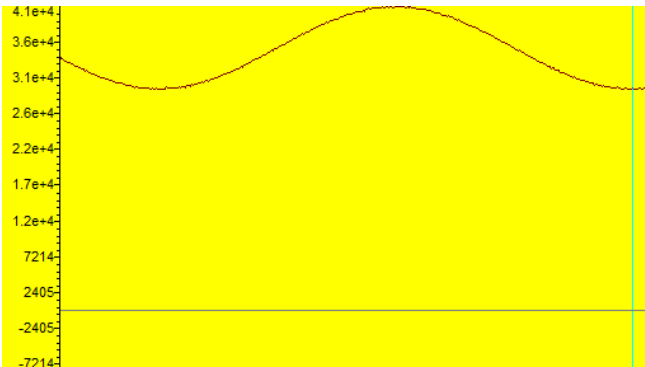


图 7：CCS 模拟仿真图

3、利用上述图形，给出验证采样频率的方法，以此验证数据采集程序的正确性。

验证采样频率的方法有多种，在本次实验中我们总共使用了两种方法来计算我们的采样频率。

方法一 硬件验证

硬件验证的方法就是在 DSP 的实际运行中，通过示波器等外设进行采样频率的验证。具体操作方法为在中断服务程序刚开始时，给 DAC 输出一个高电平，在中断服务结束时，给 DAC 输出一个低电平，在一系列的采样后可以在示波器上看到一系列脉冲，通过方法波形可以看到，脉冲信号就是极窄的方波信号，高电平为 AD 采样程序所消耗的时间，而相邻两个脉冲上升沿则是采样时间间隔。具体的程序实现如下。

```
interrupt void adc_isr(void)
{
    EALLOW;
    * DAOUT= 0x0000;
    GpioDataRegs.GPASET.all = 0xFFFF;

    //重新初始化ADC采样序列
    AdcRegs.ADCTRL2.bit.RST_SEQ1=1; //复位SEQ1
    AdcRegs.ADCST.bit.INT_SEQ1_CLR=1; //清除中断位INT SEQ1
    PieCtrlRegs.PIEACK.all=PIEACK_GROUP1; //清除PIE1的中断响应位
    GpioDataRegs.GPACLEAR.all = 0xFFFF;
    * DAOUT= 0xffff;
    EDIS;
}
```

图 8：硬件验证程序代码截图

通过改变定时器时钟频率可以看到示波器的输出波形频率发生改变。

当改变 TPS 的值为 7 时，示波器的输出显示如下，示波器测得的采样频率为 9.30048kHz。

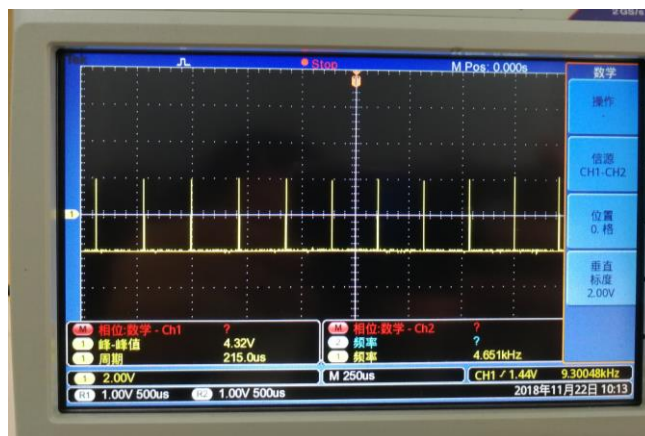


图 9：TPS=7 时示波器显示

当改变 TPS 的值为 6 时，示波器的输出显示如下，示波器测得的采样频率为 18.6009kHz。

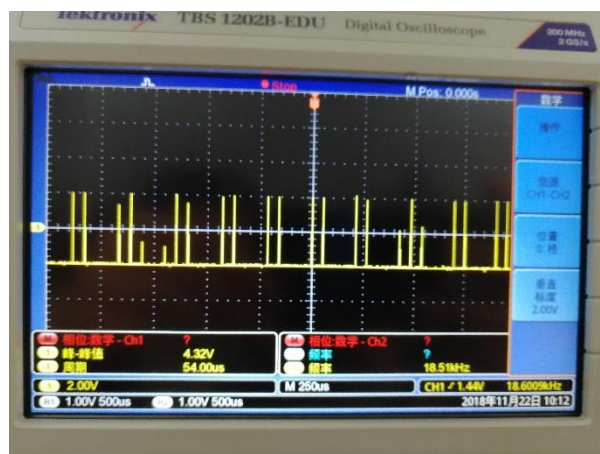


图 10：TPS=6 时示波器显示

当改变 TPS 的值为 5 时，示波器的输出显示如下，示波器测得的采样频率为 37.2019kHz。

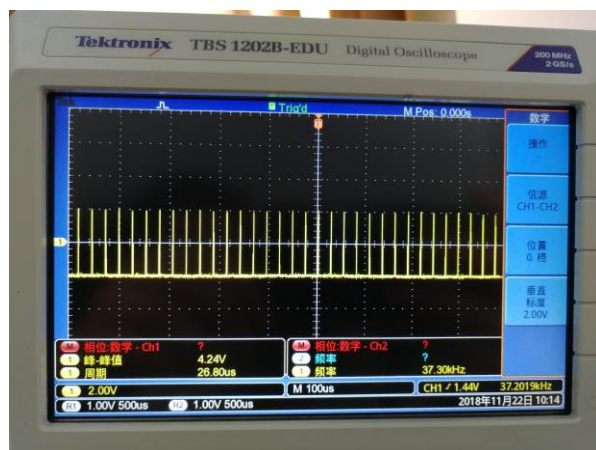


图 11: TPS=5 时示波器显示

方法二 软件验证

为了让采样频率更有说服力,我还通过 CCS 软件进行了软件验证,具体是同样通过改变定时器的时钟频率,再通过绘制信号波形图,计数一个周期的点数,将该点数乘以信号源输入频率便可得到采样频率。我将输入信号源频率改为 50Hz,将 TPS 的值设置成 6,通过 CCS 自带绘图软件可以看到如下图。

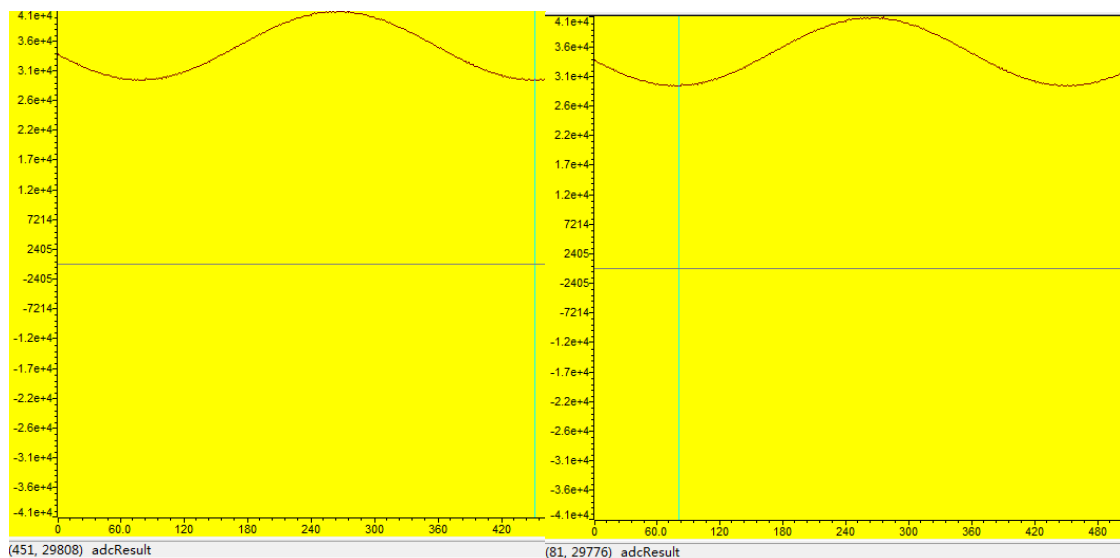


图 12: TPS=6 时软件验证波形图对比

从图中可以看到,一个周期内总共有 370 个点,乘上输入信号的频率,即可得到采样频率 18500Hz。

同理,将 TPS 设置成 7 时,可以绘制出存储器中的波形。

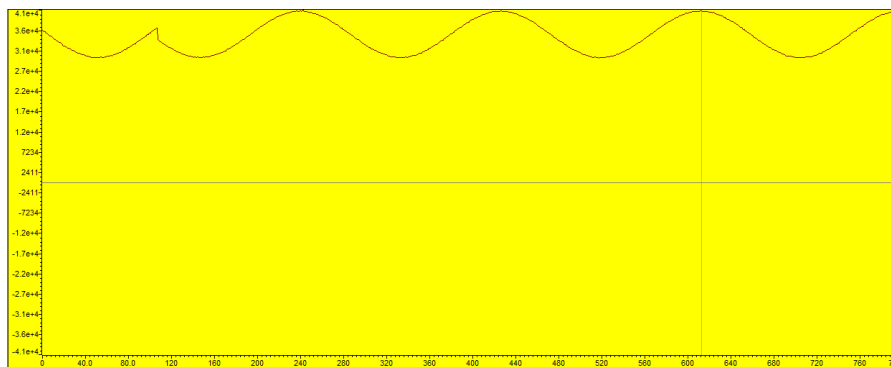


图 13: TPS=7 时软件验证波形图

通过对比软硬件验证的结果可以发现，相同条件下通过软硬件验证得到的结果都是几乎一致的，所以这都很好地验证了 ADC 地采样频率。

4、利用数码显示管，添加语句或者在中断服务子程序中编写程序，使之能够交替显示实验者的学号。为了保证能够看清实验者学号，所以需要让学号能够显示一段时间，显示学号的方法有很多种，在本次实验中，我定义了三个 LED 显示程序，分别用来显示三个实验者的学号，然后通过判断三种状态，从而让数码管可以轮流显示。具体的代码实现如下。

```
interrupt void adc_isr(void)
{
    EALLOW;
    GpioDataRegs.GPASET.all = 0xFFFF;
    adcResult[i]=AdcRegs.ADCRESULT0;
    * DAOUT= adcResult[i];
    ++i;

    if(i>1024){
        i=0;
        ++n;
        if(n>20){
            n=0;
            ++k;
            if(k>3) k=0;
            if(k==0){
                fcLED();
            }
            else if(k==1){
                zxxLED();
            }
            else{
                ztLED();
            }
        }
    }
}
```

图 14: 学号显示程序核心代码

从图中可以看到，我通过计算采样点数来轮流判定三种状态，即当采样点数到达 1024 时，变量 n 加 1，当 n 累加到 20 时，程序判定 k 再加 1，而 k 代表着 LED 进入另一种状态，从整体上看就是当采够 20000+个点的时候，数码管的显示会更换一个人的学号，图中 `fcLED()`、`zxzLED()`、`ztLED()` 分别表示每个实验者学号的 LED 显示程序。最终实现的效果图如下。

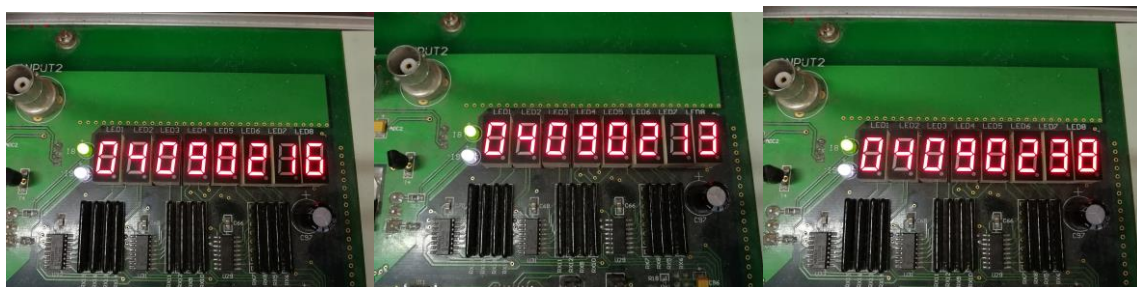


图 15：学号显示效果图

7 实验感悟

7.1 实验中遇到的问题与解决方案

7.1.1 计算理论采样频率时总是和实际得到的采样频率有差距

通过研究发现，在计算定时器周期寄存器时一直少算了一个周期数，即实际计算时的周期为定时器周期+1，因为定时器周期寄存器在计数时当等于 0 时还会计数一个周期，所以在最终计算采样频率时，也要给予考虑。

7.1.2 输入信号与示波器显示信号、存储器中存储波形信号幅度存在差异

造成这种差异主要有两个原因：第一，整个实验箱电路中有各种元器件会分压，信号通过电路会有能量的损耗，所以最终导致存储器中的波形幅值小于输入信号幅值；第二，DSP 芯片是以 4 位 16 进制数的形式存储采样数据的，所以最终导致输出到示波器上的波形幅值又有一定程度上的衰减。

7.2 实验的收获与感受

这次是第三次实验，让我对于整个 DSP 芯片有了更深层次的理解，在上一次实验 DA 输出的基础上，本次实验让我对于 AD 采样有了进一步认识，在第一次接触 AD 采样还是在信号与系统的专业课程中，当时的学习只是单纯的理论学习，很难说对于实际的数字采样到底能有什么了解，但是通过一年多时间的沉淀，以及本次实验，可以说，我能够基本根据自己的想法运用 ADC 和 DAC 了。

在本次实验中，一开始对于采样频率的计算还不是很清楚，最后通过上网查阅相关资料成功计算出了采样频率，而在硬件方面每次进入中断程序就是一次采样，所以可以通过进入中断程序的频率等效为采样频率，而示波器输出的高电平脉冲宽度则代表了每次中断程序的运行时间，即每次采样的时间。

在预习下次实验时我发现，下一次实验完全是在这一次实验的基础上完成的，唯一的区别就是本次实验只是单纯地将采样得到的输入信号输出到 DAC 中，中间没有进行任何运算，但是在下一次实验中，需要将采样得到的信号进行处理再输出，这就要考虑到信号实时性的问题了，在本次实验中，通过硬件验证的方法也同样可以观察到信号的实时性，示波器中显示的都是脉冲，这就说明采样时间占采样周期的时间很短，满足信号实时性的要求。

其实最近在做实验时都有一种兴奋感与成就感，之前学习到的理论定理和知识都可以在这些实践中得到验证。比如奈奎斯特采样定律，在本次实验中都可以得到验证，当看到频率接近超过二分之一采样频率时输出波形不正常，这种成就感真的很棒，这种感觉能够鼓励我进行接下来的实验。

在这三次的实验中，每一次做软件仿真时总是忘记调整数据长度，多次踩进同一个坑能够让我加深印象，现在在每一次使用绘图工具时，我都会思考数据长度，其实实验就是这样，在实验过程中发现自己的不足，在下一次实验中不断进步，这才是我们要从实验中学习到的东西，而不是单纯地追求一个结果，那真的是毫无意义的。