



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: FIR 滤波器的 DSP 实现

班 级: 9151040G02

姓 名: 傅 超

学 号: 9151040G0216

指导老师: 李彧晟

2018 年 11 月 27 日

目 录

目 录.....	- 1 -
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 实验流程.....	1
4.2 程序流程.....	2
4.3 FIR 滤波器设计.....	3
4.4 DSP 算法实现.....	3
4.5 算法实时性测试.....	4
5 实验步骤.....	4
6 实验结果.....	6
6.1 系数设计.....	6
6.2 算法功能验证.....	7
6.3 算法实时性验证.....	8
6.4 实际的幅频特性曲线.....	9
6.5 数码管轮流显示学号.....	11
7 实验感悟.....	12
7.1 实验中遇到的问题与解决方案.....	12
7.1.1 信号通过滤波器后有失真.....	12
7.1.2 C 语言效率相较汇编效率低.....	12
7.1.3 MATLAB 滤波器系数产生不理解.....	13
7.1.4 算法验证正确，但无法得到波形.....	13
7.2 实验的收获与感受.....	13

1 实验目的

- 1、巩固数字 FIR 滤波器的概念
- 2、理解定点 DSP 中数的定标、有限字长、溢出等概念
- 3、理解算法实现中实时的概念
- 4、掌握 DSP 开发过程以及基本调试方法
- 5、理解汇编以及高级语言开发 DSP 实现算法的区别

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

针对 FIR 算法, 设计滤波器系数, 完成数据的定标, 查看滤波器特性曲线。

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 FIR 滤波器算法, 通过信号源、示波器理解滤波器特性, 验证实现与理论设计的一致性。

4 实验准备

4.1 实验流程

实验之前首先必须对 FIR 滤波器的设计、实现算法有所了解, 必要时通过计算机算法仿真, 理解 FIR 滤波器特性。由于计算机仿真属于浮点运算, 而 TMS320F2812 是定点 DSP, 因此需要针对所设计的 FIR 滤波器系数进行定标, 随后对定标后的数据再次进行仿真, 以验证定点实现的性能是否满足系统指标。

根据 FIR 滤波器算法，编写 C 源程序或者汇编程序，实现算法功能。并验证 DSP 实现时算法的正确性以及精度的要求。这种算法功能上的仿真可以利用 CCS 集成开发环境中数据 IO 来模拟信号的输入，完成验证算法精度与功能的正确。

验证了算法的功能正确之后，可以将程序下载到 DSP 上运行，观察现象。更为重要的是，在硬件平台上验证系统的实时性，以及评估资源的使用情况。若满足实时性要求，则测试各项指标，应该与原理设计相吻合。如果实现与理论不一致，则首先检查算法的实时性，以及资源使用是否冲突等原因，对程序进行优化后再次编译链接，重新验证直至正确。算法的优化有时会贯穿于整个设计之中。

4.2 程序流程

FIR 滤波器算法属于典型的数据流处理方式，每到达一个新数据，就必须进行一次计算，更新输出。因此，当一次采样完成之后，就可以进行 FIR 核心算法，并将计算结果输出给 DAC。

因此，和 DSP 的数据采集实验类似，用 DSP 实现实时的 FIR 信号处理算法必须依赖于 ADC、DSP 以及 DAC 三大基本部件。充分利用 DSP 片上 ADC 外设，实现模拟信号的采样，并由 DSP 完成 FIR 核心算法，由 DSP2000 实验箱中 DAC1（AD768）来完成数字到模拟的还原。在数据采集实验基础上，我们对程序流程稍加改动，就可实现完整数字 FIR 滤波器功能。

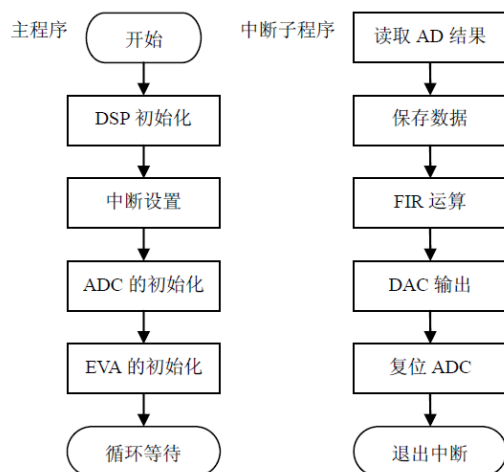


图 1：FIR 滤波器程序流程

4.3 FIR 滤波器设计

数字滤波器用于完成信号的滤波处理功能，是用有限精度算法实现的离散时间非时变系统。用 DSP 实现数字 FIR 滤波算法，具有稳定性强、精度高、实时性好、灵活性大、实现简单等优点。

有限长的单位冲击响应滤波器（FIR）差分方程可表示为：

$$y[n] = \sum_{k=0}^N h[k]x[n-k]$$

其中， h 是滤波器系数， x 为输入的数字信号， y 为 FIR 滤波器计算输出。 N 为滤波器阶数。由此可得，一个 N 阶的滤波器计算，需要 $N+1$ 个滤波器系数， $N+1$ 个数字输入，每得到一个 y 值，需要 $N+1$ 次乘法以及 N 次加法。另外， N 阶滤波器需要保存当前的 $N+1$ 个输入信号数值，以及事先设计的 $N+1$ 个滤波器系数。

4.4 DSP 算法实现

TMS320F2812 是定点型 DSP，存储器字长 16bit，可进行 32bit 的运算。而仿真计算中得到的数据大多是浮点型，因此将算法用定点 DSP 实现时，必须进行数据格式的定标。比如对 FIR 滤波器系数的定标可以参照“实验 10 任意信号发生器”中介绍的方法来完成。对系数定标后，还要进行仿真以验证性能。

另外，由于 TMS320F2812 的数据字长只有 32bit，DAC1 接受的字长为 16bit，因此有限字长带来精度的损失。更为重要的是当加法的结果超过 16 位表示范围时，数据产生了溢出，这是有限字长带来的第二个问题。再者，数据的计算结果存放在 32bit 的寄存器中，但 DAC 却是 16bit，取高位输出还是低位输出，还是取一个合适的范围，这是数据截取的问题。

因此在编写程序时，必须考虑定点数据的运算效应，由数据的动态范围来确定截取、定标等问题。定点 DSP 内部一般有溢出保护功能，可以查看溢出标志位及时发现溢出现象，其次用溢出模式位来使 ACC 结果控制在最大值范围之内，以达到防止溢出引起精度严重恶化的目的。

具体的实现编程，可以采用 C 语言或者汇编语言。C 语言描述算法较为简单，在此不作详细叙述。

4.5 算法实时性测试

算法的实时性测试主要指该算法能否在规定的时间内完成 FIR 运算，规定时间在此是指采样周期。FIR 的运算必须在两次采样间隔内完成，否则会造成数据的丢失。这是数据流处理的特点，数据的运算速度必须大于数据的更新速度。

在实验平台中，我们可以利用 GPIO 管脚来实测采样周期和算法执行时间。添加程序，当程序进入断点时，将 GPIO 的某一个引脚输出置高，完成算法退出中断时再置低。由此当全速执行程序时，测量该 GPIO 引脚上的周期，便是采样周期，高电平持续时间即为 FIR 滤波器算法执行时间，由此判断计算法实现是否实时。

5 实验步骤

1、系数设计

利用 MATLAB 设计滤波器系数，并对系数进行数据定标，完成浮点到定点的转换。分别作出两组系数的幅频、相频特性曲线，看是否满足设计要求。

2、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。

3、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。

4、建立工程

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件。

在上次实验程序的基础上加以修改，在中断程序内添加 FIR 算法模块，完成 FIR 算法程序。设置 GPIO 的输出控制，使之能够完成算法执行时间测量的工作。

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、加载程序

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

6、算法功能验证

在中断服务子程序恰当的地方，设置探针点，利用文件 IO 的方式，输入 x 数据；在 FIR 算法结束处设置断点，验证 FIR 滤波器算法的正确性。输入的数据可具有一定的特殊性，进行特例验证。

若计算结果不正确，则进行程序的调试。调试关键在于现象重现、错误定位，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，根据数值以及实验原理，查找错误原因，修改程序直至正确为止。

7、算法实时性验证

去除程序中的探针点以及断点重新全速运行程序，测量采样频率以及 FIR 算法的核心执行时间，判断该系统是否实时。

若非实时，则优化程序，甚至修改滤波器系数，直至满足实时要求。

8、连接外部电路

打开信号源，产生一个合适频率 (ADC 的采样频率必须满足奈奎斯特采样定律) 的正弦信号，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。

全速运行程序，调节信号源 (正弦信号) 的输出频率 (正弦信号)，记录各频点的示波器上输出幅度。描点作图，与理论幅频特性曲线比较，分析是否满足设计要求。

6 实验结果

6.1 系数设计

根据题目要求，需要设计一个低通 FIR 滤波器，Matlab 中自带 FIR 滤波器设计函数，在 Matlab 界面利用 `fir1` 命令来设计示波器系数，`fir1` 的完整命令如下：

$$h = \text{fir1}(n, Wn, 'ftype')$$

其中， n 为滤波器阶数， Wn 为归一化截止频率（这里的归一化指与采样频率一半进行归一化）， Wn 对应了在幅频曲线上 -6dB 点的频率数值，`ftype` 为滤波器类型，可以是低通、带通、高通、带阻等形式。

在生成 50 阶滤波器系数后，由于 DSP 为定点运算，所以需要根据 Q15 标准将滤波器系数进行定标转换，转换后，为了验证转换前后的滤波器幅频特性几乎不变，所以执行 `freqz()` 指令，用来绘制两个系数的幅频特性，对比图如下：

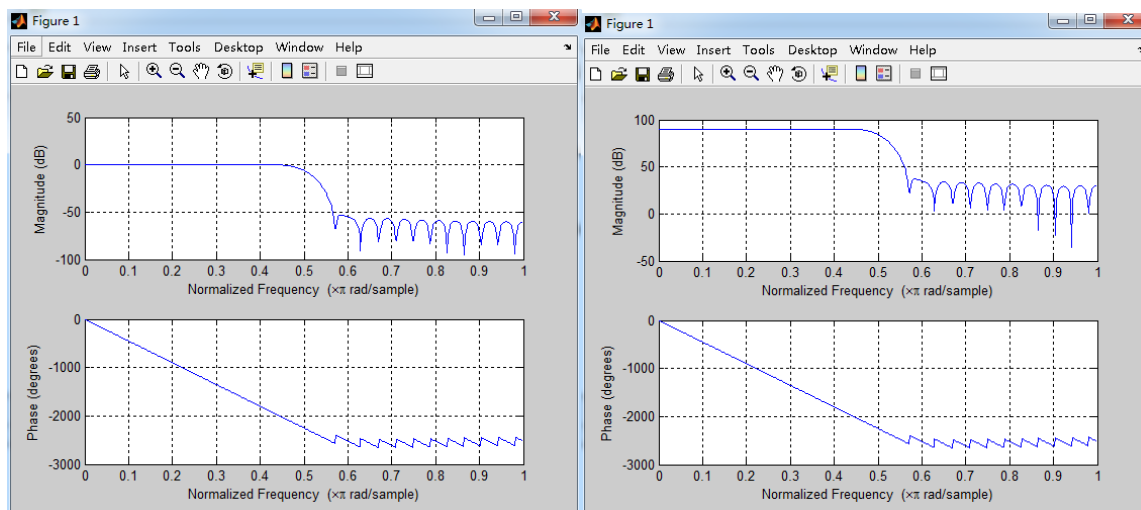


图 2：定标前后幅频特性对比图（左为定标前，右为定标后）

从图中对比可以发现，定标转换前后滤波器的幅频特性几乎一致，这说明定标转换后的滤波器系数可以使用。

最终得到的滤波器的系数为 $h = \{33, 0, -43, 0, 68, 0, -112, 0, 180, 0, -276, 0, 410, 0, -593, 0, 852, 0, -1240, 0, 1901, 0, -3361, 0, 10382, 16368, 10382, 0, -3361, 0, 1901, 0, -1240, 0, 852, 0, -593, 0, 410, 0, -276, 0, 180, 0, -112, 0, 68, 0, -43, 0, 33\}$

本实验中所使用到的所有 Matlab 指令如下：


```

-h=fir1(50,0.5,'low')
-hq=h*2^15
-roundhq=round(hq)
-freqz(roundhq)
-freqz(h)

```

图 3: Matlab 设计滤波器指令截图

6.2 算法功能验证

在得到低通滤波器的系数后，我们设计完成了低通 FIR 滤波器，算法的设计思路主要是，对于新采样得到的点，将其存入到 **data** 数组中，不断覆盖原有数组中的值，从而让数据得到更新。

```

Uint16 lowFIR(int data[FIR_ORDER+1], const int fir_coff[FIR_ORDER+1], int index){
    int i;
    long int x;
    long int h;
    long int sum = 0;
    for (i=0; i<=FIR_ORDER; ++i, ++index) {
        if (index > FIR_ORDER) index -= FIR_ORDER+1;
        x = data[index];
        h = fir_coff[i];
        sum += x*h/0x8000;
    }
    return (Uint16)(sum+0x8000);
}

```

图 4: FIR 低通滤波器代码截图

根据 FIR 原理可以发现，采样得到的数据与滤波器的系数是倒序相乘再相加，所以在写代码时，我将每次新采样得到的数据存入 **data** 数组的末尾，这样在相乘相加时就不需要再额外变动了。

完整的中断程序代码如下：

```

interrupt void adc_isr(void){
    static int i = FIR_ORDER;
    Uint16 fir_result;
    LedDisplay("04090213");
    #ifdef CHECK
        *DAOUT = 0xFFFF;
    #endif
    EALLOW;
    GpioDataRegs.GPASET.all = 0xFFFF;
    data[i] = (int)(AdcRegs.ADCRESULT0-0x8000);
    fir_result = lowFIR(data, fir_coff, i);
    #ifndef CHECK
        *DAOUT = fir_result;
    #endif
    if (--i<0) i = FIR_ORDER;
    //重新初始化ADC采样序列
    AdcRegs.ADCCTL2.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;
    EDIS;
    #ifdef CHECK
        *DAOUT = 0x0000;
    #endif
}

```

图 5: 中断程序代码截图

为了验证我写的 FIR 滤波器算法的正确性，我需要对其进行验证，根据课件中的说法，我需要输入一份数据文件，然后将输出的文件进行图形绘制，进而来验证算法的合理性与正确性，由于不熟悉数据文件的产生方法，所以在本实验中，我直接将信号源的接入到输入端，然后通过 CCS 自带的绘图工具，将波形进行绘制，在最终的绘图结果中可以发现，当我信号源给了 1kHz 的输入信号时，图形可以输出正弦信号，根据计算，为 0.99997kHz 的信号，这说明我写的算法没有问题，可以进行接下来的步骤。

6.3 算法实时性验证

由于 AD 采样需要时间，FIR 滤波器算法计算也需要时间，而本实验中又对采样频率做了一定的限制，这意味着我们设计的 FIR 滤波器算法速度必须满足一定的要求，这就是算法实时性验证。

具体的验证方法在第三个实验已经使用过，即当程序进入中断时，将*DAOUT 置为高电平，当执行完中断中的 FIR 算法后，将*DAOUT 置为低电平，这样输出的就是有高低电平的波形，具体波形如下：

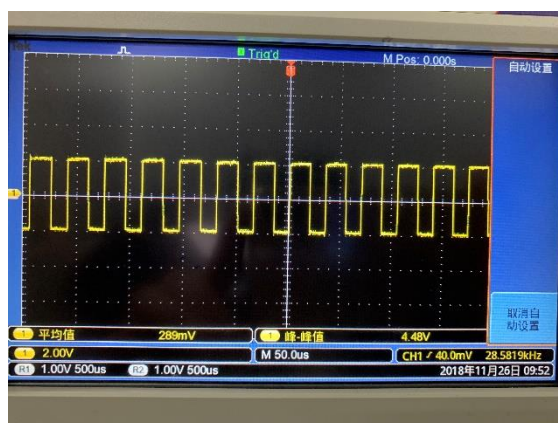


图 6：算法实时性验证图

从图中可以看出，该波形的频率为 28.5819kHz，这和我们之前设定的采样频率一致，波形中可以看出高低电平，低电平宽度适宜，这说明我们设计的算法，在该采样频率下，执行完 FIR 滤波器算法还有时间空余，这就避免了数据流速度大于算法执行速度导致的死机情况出现。

在后期完善中, 为了更详细地了解 FIR 算法的执行时间相对于采样周期的时间关系, 我们对这两个参数进行了测量, 具体操作如下图所示:

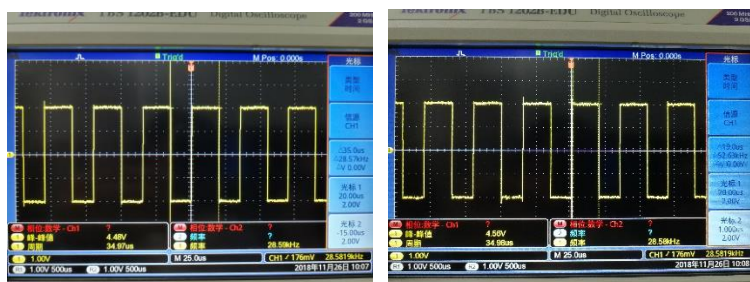


图 7: 算法实时性验证图

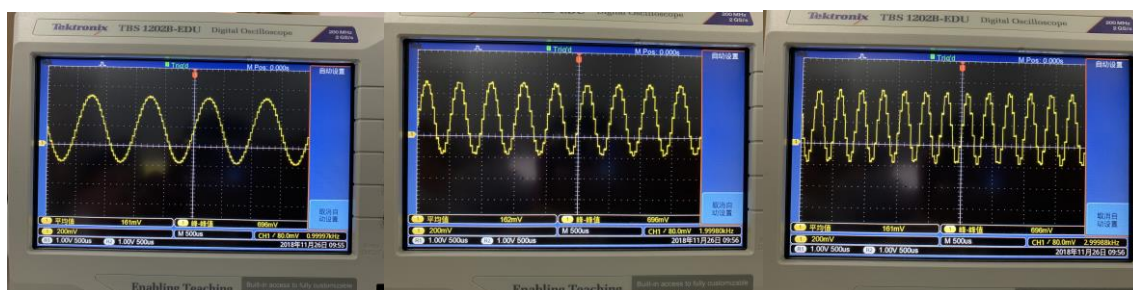
从图中可以看到, 一个采样周期的时间为 $35\mu\text{s}$, 而我的 FIR 滤波器算法执行时间为 $20\mu\text{s}$, 大约占用了 57% 的采样周期, 这个情况符合预期。

6.4 实际的幅频特性曲线

为了得到我们设计的 FIR 低通滤波器的幅频特性曲线, 我们通过不断调整输入信号的频率, 观察输出信号的幅值, 最终绘制成一张完整的图。

因为我们设计的采样频率为 28.5819kHz , 所以最高可输入频率则为 14.2910kHz , 由于我滤波器设置的归一化截止频率为 0.5, 所以该滤波器的理论截止频率应为 7.1455kHz 。

我们将信号源的频率从 1kHz 开始向上调整, 最终得到了不同频率下测得波形的峰峰值。调整过程中的波形如图所示:



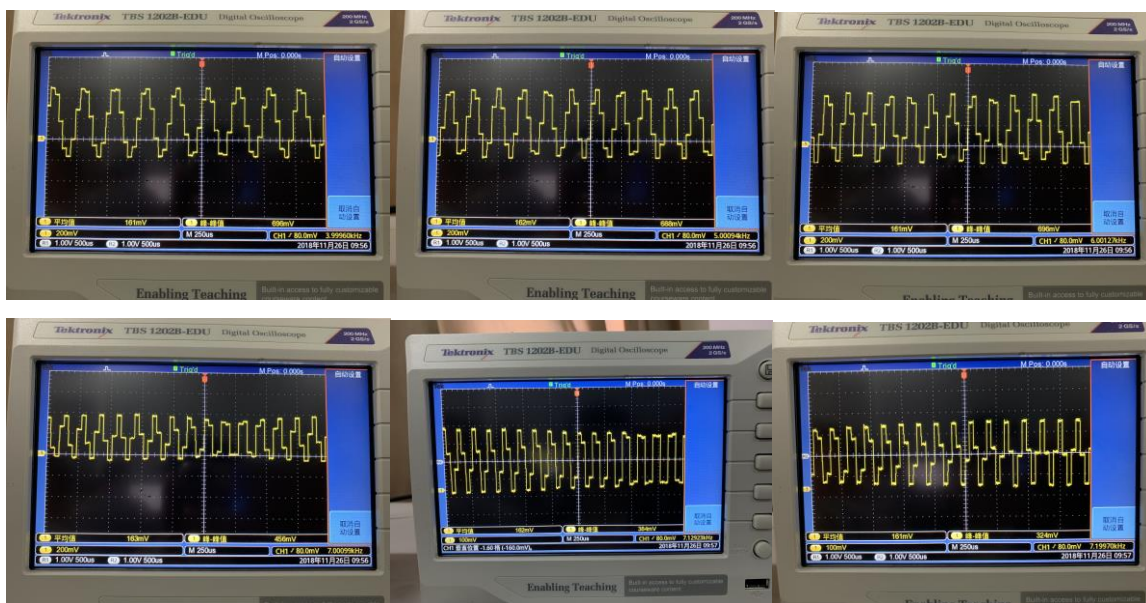


图 8：调整频率过程中输出波形

最终将频率与幅值的对应关系整理成一张表格如下：

FIR 低通滤波器频率与幅值对应关系表						
频率 (kHz)	0.99997	1.9998	2.99988	3.9996	5.00094	6.00127
峰峰值 (mv)	696	696	696	696	688	696
频率 (kHz)	7.00099	7.12923	7.1997	7.29906	7.40023	7.49951
峰峰值 (mv)	456	384	324	260	208	156

表 1：FIR 低通滤波器频率与幅值对应关系表

将上述数据绘制成图如下：

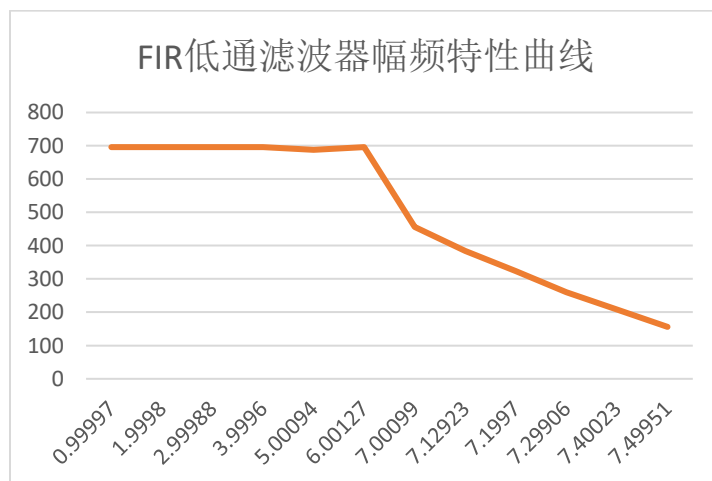


图 9：FIR 低通滤波器幅频特性曲线

从图中可以看出幅频特性曲线在峰峰值下降 3dB 左右，即峰峰值为 400mV 左

右，对应的频率为 7.14kHz 左右，这与我们预先设计的截止频率符合，这说明这个实验整体流程操作下来，完全没有问题，符合实验要求。

6.5 数码管轮流显示学号

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。

```
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){  
            zxzLED();  
        }  
        else{  
            ztLED();  
        }  
    }  
}
```

图 10: 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。

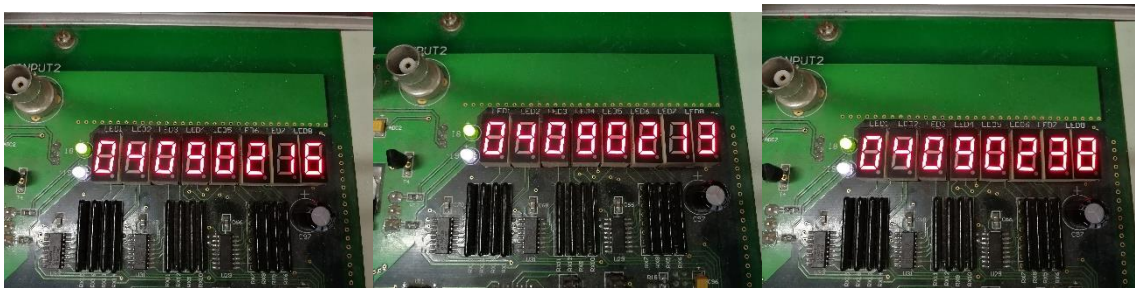


图 11: 学号显示效果图

7 实验感悟

7.1 实验中遇到的问题与解决方案

7.1.1 信号通过滤波器后有失真

由于 DSP 芯片中的 DAC 的接收字长为 16 位，所以输入信号在采样存储计算，最终输出的过程中肯定存在一定的数据损失，最重要的是为了让滤波器成功实现功能，需要将经过 FIR 滤波器的信号进行移位输出，移位不同对于信号的失真也有很大的关系。

7.1.2 C 语言效率相较汇编效率低

本实验给的例程是汇编语言编写的，加载实验所给的汇编语言，发现实现相同的功能，DSP 处理数据的速度大大提高。所以在之前也有粗略地了解过汇编语言的执行效率极高，但是没想到实际操作时还是高了近一个量级，在网上查阅相关文献发现，1. 编译器将 C 语言自动翻译成机器码；编译器再智能、再强大，还是会产生冗余；这些冗余就是造成比汇编慢的主要原因 2. 对于汇编来说，由汇编产生的机器码，几乎没有冗余，因为汇编指令和机器码是严格的一一对应的，这就是汇编比 C 快的主要原因。所以直接使用汇编语言编写每一个操作都是针对算法的要求采用最优的指令和存储空间，是软件和硬件的最大利用。

7.1.3 Matlab 滤波器系数产生不理解

Matlab 自带 `fir1` 函数用于产生滤波器系数，但是该函数中只有归一化频率，这和实验要求的低通滤波器的截止频率不一致，后通过上网查阅资料发现，归一化频率即为截止频率/最高频率，而最高频率，根据奈奎斯特采样定律则为采样频率的一半。

7.1.4 算法验证正确，但无法得到波形

在软件调试阶段可以确保软件正确无误，但是在实际利用信号源与示波器调整时发现一直没有输出应有的波形，最后发现信号源的输出开关没有打开，这真的十分让人崩溃，也浪费了很多时间。

7.2 实验的收获与感受

本次实验是总共四次实验中，难度最大的一次实验，但也是给人成就感最大的一次实验，这个实验不同于前几个实验，都有现成的例程可以直接运行，该实验只给了汇编语言，虽然可以直接编译运行，但是无法根据我们想要的内容进行修改，所以只能在前几次的实验程序的基础上进行修改。修改程序的前提是要读懂这个程序，所以在本次实验开始操作前，我花了大量的实践去阅读之前的程序，了解其设置每一个参数的作用，然后再结合自己的想法，最终编写完成了整个 FIR 滤波器的代码。

但是我并没有一次成功，再第一次编译下载后，示波器上输出的是锯齿波，我们小组成员在讨论之后，发现问题出在了中断上面。我们换了一种方式去进行中断，在每次中断后，我们都取最新的一个采样点去与相应的系数相乘，即用移位的方法来实现。

前前后后也已经做了 3 个小实验和一个较大的实验，这段时间的实践学习，让我对于整个数字信号处理的系统有了一个大致的认识，能够根据自己的想法设计想要的功能，并通过程序将其再硬件上实现，我觉得这是我通过这个实验最大的收获。

在这个实验中给我印象最深的还是小数的定标，其实程序部分由于我之前有一定的基础，所以写起来也不是很吃力，但是定点 DSP 的定标对于我来说真的十分陌生，虽然在上课的时候也已经学习过，在课后作业也已经做过，但是真正在实验中需要用到时，我还是感受手足无措，在实验中也是花费了大把的时间去完成小数的定标，但是通过这个自主学习的过程，我对于这个知识点算是透彻地了解了，明白了小数定标的原因，以及怎么定标等一些方法，这个是单纯的理论课程所不能带给我的。

在实验的过程中，李老师一致强调要我们在实验的过程中多遇到问题，我去想为什么，而不是一位地追求结果，我觉得这个确实是让我受益匪浅的地方，我记得我大一大二做各种实验时，实验课程成绩的评判依据有一项是实验速度，这就迫使我们为了追求那所谓的速度，而对于一些知识点囫圇吞枣，在实验过程中遇到的问题也不加思考就直接跳过，最终一个实验下来，几乎没有收获什么，但是在这几次的 DSP 实验当中，我真的学习到了很多，学到了一个全新的硬件开发环境，学会了对于一个程序或者猜想，分别进行软件验证和硬件验证，学会了如何结合 Matlab 等我们熟悉的软件产生我们想要的数数据，学会了遇到问题如何冷静下来，使用“二分法”一点点地进行排查调试。我觉得从一个或者几次实验中，我们要学习的除了知识点之外，更多的是要去学习实验的方法，就像我们进行本科生科研训练，目的不是让我们发表多厉害的论文专利，而是让我们能够对于科研有一个提前的了解，对于科研的过程有一个大致的概念，培养我们独立思考、查阅文献等多方面的能力。

所以，要透过现象看本质，把握好每一件事背后的核心和目的，我们在做这件事的时候才会有更清晰的目标与更丰硕的收获。

感谢李老师在 5 次实验课中的指导！