

**DSP应用技术实验**

**FIR滤波器实现实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | | | |
|  |  | FIR滤波器实现实验报告 | | | |
| 指导者 | ： | 李彧晟 | | | |

2019 年 11 月

目录

[1 实验目的 1](#_Toc25868323)

[2 实验仪器 1](#_Toc25868324)

[2.1 实验仪器清单 1](#_Toc25868325)

[2.2 硬件连接示意图 1](#_Toc25868326)

[3 实验步骤及现象 1](#_Toc25868327)

[3.1 算法实现流程图 1](#_Toc25868328)

[3.2 程序流程图 2](#_Toc25868329)

[3.3 系统设计 3](#_Toc25868330)

[3.3.1 利用设计滤波器系数 3](#_Toc25868331)

[3.3.2 数据定标 4](#_Toc25868332)

[3.4 设备检查并启动集成开发环境 5](#_Toc25868333)

[3.5 编写FIR算法模块 5](#_Toc25868334)

[3.5.1 FIR滤波器编写原理 5](#_Toc25868335)

[3.5.2 DSP芯片存储器字长转换关系验证 5](#_Toc25868336)

[3.5.3 输入信号的更新 6](#_Toc25868337)

[3.5.4 FIR滤波器代码 6](#_Toc25868338)

[3.6 建立工程并运行、调试程序 7](#_Toc25868339)

[3.7 算法实时性验证 7](#_Toc25868340)

[4 实验结果及思考题回答 7](#_Toc25868341)

[4.1 设计FIR滤波器并仿真 7](#_Toc25868342)

[4.2 算法功能验证 9](#_Toc25868343)

[4.3 实际幅频特性曲线 12](#_Toc25868344)

[4.4 验证算法实时性 13](#_Toc25868345)

[5 实验总结 15](#_Toc25868346)

[5.1实验中遇到的问题及解决方法 15](#_Toc25868347)

[5.2实验心得体会 15](#_Toc25868348)

# 1 实验目的

1. 巩固数字FIR滤波器的概念；

2. 了解DSP运算特点；

3. 理解算法实时性含义；

4. 熟练掌握DSP软件开发过程；

5. 熟练掌握DSP软件调试方法。

# 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 |
| 3. | 信号发生器 | 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。

|  |
| --- |
| 图2. 1 硬件连接示意图 |

# 3 实验步骤及现象

## 3.1 算法实现流程图

结合实验要求，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。

根据FIR滤波器算法，编写C源程序，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，对程序进行用是否冲突等原因，对程序进行用是否冲突等原因，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。于是，程序流程图如图3. 1所示。

|  |
| --- |
| 图3. 1 算法实现流程图 |

## 3.2 程序流程图

FIR滤波器处理数据时，每到达一个新数据，就必须进行一次计算，更新输出。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，由实验箱中DAC（AD9747）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。程序流程如图3. 2所示。

|  |
| --- |
| 图3. 2 程序流程图 |

## 3.3 系统设计

### 3.3.1 利用设计滤波器系数

利用MATLAB 设计滤波器系数，设计的滤波器为48阶，其幅频、相频特性曲线如图3. 3所示，系数矩阵如表3. 1所示。

|  |
| --- |
| 图3. 3 MATLAB设计的滤波器的幅频相频特性曲线 |

表3. 1 MATLAB设计的滤波器的h系数

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| h | 值 | h | 值 | h | 值 | h | 值 |
| 0 | -0.00042 | 13 | 0.016914 | 26 | 0.061816 | 39 | 0.008389 |
| 1 | 0.000179 | 14 | 0 | 27 | 0.015797 | 40 | 0.002434 |
| 2 | 0.000546 | 15 | -0.02467 | 28 | -0.02946 | 41 | -0.00075 |
| 3 | 0.000528 | 16 | -0.05121 | 29 | -0.06258 | 42 | -0.0015 |
| 4 | 0 | 17 | -0.07129 | 30 | -0.07673 | 43 | -0.00089 |
| 5 | -0.00089 | 18 | -0.07673 | 31 | -0.07129 | 44 | 0 |
| 6 | -0.0015 | 19 | -0.06258 | 32 | -0.05121 | 45 | 0.000528 |
| 7 | -0.00075 | 20 | -0.02946 | 33 | -0.02467 | 46 | 0.000546 |
| 8 | 0.002434 | 21 | 0.015797 | 34 | 0 | 47 | 0.000179 |
| 9 | 0.008389 | 22 | 0.061816 | 35 | 0.016914 | 48 | -0.00042 |
| 10 | 0.015997 | 23 | 0.096004 | 36 | 0.023942 |  |  |
| 11 | 0.022467 | 24 | 0.10862 | 37 | 0.022467 |  |  |
| 12 | 0.023942 | 25 | 0.096004 | 38 | 0.015997 |  |  |

### 3.3.2 数据定标

h系数中最大的值为0.108619773528739，将它映射到16位有符号数的最大值32767，即乘301666并取整，其余的值也做此操作，得到表3. 2的系数，完成浮点到定点的转换。作出定标后系数的幅频相频特性曲线如图3. 4所示，发现满足设计要求。

表3. 2 数据定标后的滤波器系数

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| h | 值 | h | 值 | h | 值 | h | 值 |
| 0 | -126 | 13 | 5102 | 26 | 18648 | 39 | 2531 |
| 1 | 54 | 14 | 0 | 27 | 4765 | 40 | 734 |
| 2 | 165 | 15 | -7442 | 28 | -8889 | 41 | -227 |
| 3 | 159 | 16 | -15448 | 29 | -18877 | 42 | -452 |
| 4 | 0 | 17 | -21505 | 30 | -23146 | 43 | -268 |
| 5 | -268 | 18 | -23146 | 31 | -21505 | 44 | 0 |
| 6 | -452 | 19 | -18877 | 32 | -15448 | 45 | 159 |
| 7 | -227 | 20 | -8889 | 33 | -7442 | 46 | 165 |
| 8 | 734 | 21 | 4765 | 34 | 0 | 47 | 54 |
| 9 | 2531 | 22 | 18648 | 35 | 5102 | 48 | -126 |
| 10 | 4826 | 23 | 28961 | 36 | 7223 |  |  |
| 11 | 6778 | 24 | 32767 | 37 | 6778 |  |  |
| 12 | 7223 | 25 | 28961 | 38 | 4826 |  |  |

|  |
| --- |
| 图3. 4 数据定标后的滤波器的幅频相频特性曲线 |

## 3.4 设备检查并启动集成开发环境

检查仿真器、C2000 DSP 实验箱、计算机之间的连接是否正确，确认无误后，打开计算机和实验箱电源，并进入集成开发环境 CCS。

## 3.5 编写FIR算法模块

### 3.5.1 FIR滤波器编写原理

有限长的单位冲击响应滤波器（FIR）差分方程可表示为：

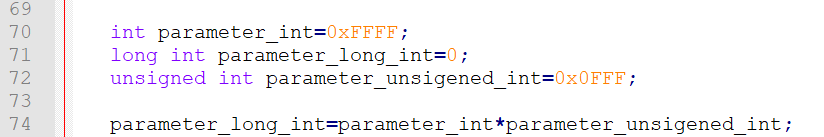
其中，是滤波器系数，为输入的数字信号，为FIR滤波器计算输出。为滤波器阶数。由此可得，一个N阶的滤波器计算，需要N+1个滤波器系数，N+1个数字输入，每得到一个值，需要N+1次乘法以及N次加法。另外，N阶滤波器需要保存当前的N+1个输入信号数值，以及事先设计的N+1个滤波器系数。

### 3.5.2 DSP芯片存储器字长转换关系验证

在程序中，涉及到16位有符号、无符号数相乘等运算，涉及到字长的变化，需要先对DSP芯片中字长变化的关系进行验证，主要分3部分：

**1.验证有符号长整型变量能否直接接受有符号数与无符号数相乘的结果**

在主程序中编写验证代码如下：

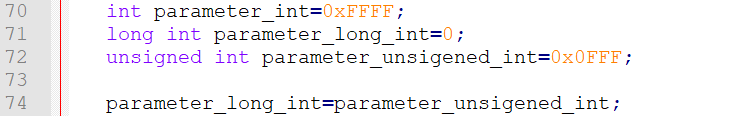


运行后，查看变量结果如图3. 5所示。则可知，这种运算方式与期望值（-4095）不一致，不可以使用这种方式来进行字长转换。

|  |
| --- |
| 图3. 5 验证有符号长整型变量能否直接接受有符号数与无符号数相乘的结果 |

**2. 验证无符号整型变量能否直接赋值给有符号长整型变量**

在主程序中编写验证代码如下：

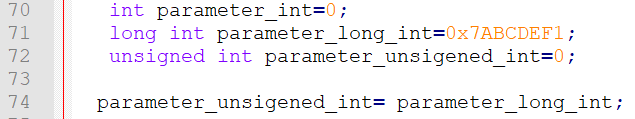


运行后，查看变量结果如图3. 6所示。则可知，无符号整型变量直接赋值给有符号长整型变量后，值不变，可以利用这种方式实现字长转换。

|  |
| --- |
| 图3. 6 验证无符号整型变量能否直接赋值给有符号长整型变量 |

**3.验证有符号长整型变量赋值给无符号整型变量的结果**

在主程序中编写验证代码如下：

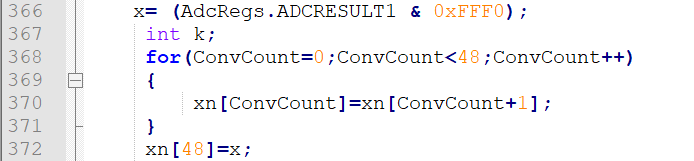


运行后，查看变量结果如图3. 7所示。parameter\_unsigened\_int=0xEDF1。则可知，有符号长整型变量赋值给无符号整型变量后，低16位有效。

|  |
| --- |
| 图3. 7 验证有符号长整型变量赋值给无符号整型变量的结果 |

### 3.5.3 输入信号的更新

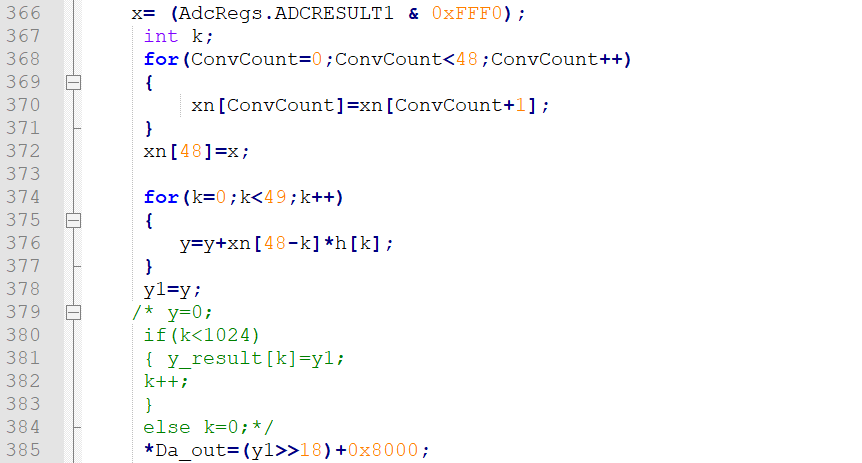
为实现滤波器输出的实时更新，每一个采样信号到来时，需要得知这个信号之前的48个信号，共49个信号。因此需要对输入信号进行前移，并将最新采样的数据存到最后，使得49个数据始终为按时间顺序排列的最新数据，实现的代码如下：



通过这段代码，前48个数据循环前移，并将最新的数据放入数组的最后一个位置。

### 3.5.4 FIR滤波器代码

综合以上内容，FIR滤波器代码如下：



## 3.6 建立工程并运行、调试程序

连接信号发生器至教学实验箱SMA输入端口J2、教学实验箱SMA输出端口J5至示波器，编译链接工程并进入调试调试界面，运行程序后，查看输出是否正确，验证算法正确性。

## 3.7 算法实时性验证

测量采样频率与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算法，直至满足实时要求。

# 4 实验结果及思考题回答

## 4.1 设计FIR滤波器并仿真

***完成FIR滤波器系数的设计并仿真。***

设计的FIR滤波器系数见表3. 2，幅频相频特性见图3. 4。

利用MATLAB对设计的FIR滤波器进行仿真验证，得到的仿真结果，如图4. 1到图4. 2所示，说明可以衰减1kHz到2kHz之外的信号。

|  |
| --- |
| 图4. 1 通过fir滤波器前的信号 |

|  |
| --- |
| 图4. 2通过fir滤波器后的信号 |

验证系统的实时性，测试采样周期以及计算时间

## 4.2 算法功能验证

***调试程序，实现FIR功能，利用硬件验证。***

通过信号发生器输入不同频率的正弦波，观察示波器上结果，如图4. 3到图4. 24所示，可观察到在1kHz到2kHz之外的信号衰减，而1kHz到2kHz之内的信号可以通过。

|  |  |
| --- | --- |
| 图4. 3 600Hz时的输出波形 | 图4. 4 700Hz时的输出波形 |
| 图4. 5 800Hz时的输出波形 | 图4. 6 900Hz时的输出波形 |
| 图4. 7 1000Hz时的输出波形 | 图4. 8 1100Hz时的输出波形 |
| 图4. 9 1200Hz时的输出波形 | 图4. 10 1300Hz时的输出波形 |
| 图4. 11 1400Hz时的输出波形 | 图4. 12 1500Hz时的输出波形 |
| 图4. 13 1600Hz时的输出波形 | 图4. 14 1700Hz时的输出波形 |
| 图4. 15 1800Hz时的输出波形 | 图4. 16 1900Hz时的输出波形 |
| 图4. 17 2000Hz时的输出波形 | 图4. 18 2100Hz时的输出波形 |
| 图4. 19 2200Hz时的输出波形 | 图4. 20 2300Hz时的输出波形 |
| 图4. 21 2400Hz时的输出波形 | 图4. 22 2500Hz时的输出波形 |
| 图4. 23 2600Hz时的输出波形 | 图4. 24 2700Hz时的输出波形 |

## 4.3 实际幅频特性曲线

***改变输入正弦信号频率，记录对应的幅度，描点作图，与理论幅频曲线比较***

如图4. 3到图4. 24所示，整理可得到表4. 1，绘图，得到实际的幅频特性曲线如图4. 25所示。

表4. 1 实际幅频情况记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 频率(Hz) | 幅度(mv) | 频率(Hz) | 幅度(mv) |
| 600 | 120 | 1700 | 800 |
| 700 | 200 | 1800 | 680 |
| 800 | 280 | 1900 | 560 |
| 900 | 360 | 2000 | 480 |
| 1000 | 440 | 2100 | 280 |
| 1100 | 560 | 2200 | 280 |
| 1200 | 680 | 2300 | 200 |
| 1300 | 720 | 2400 | 200 |
| 1400 | 800 | 2500 | 120 |
| 1500 | 840 | 2600 | 120 |
| 1600 | 800 | 2700 | 120 |

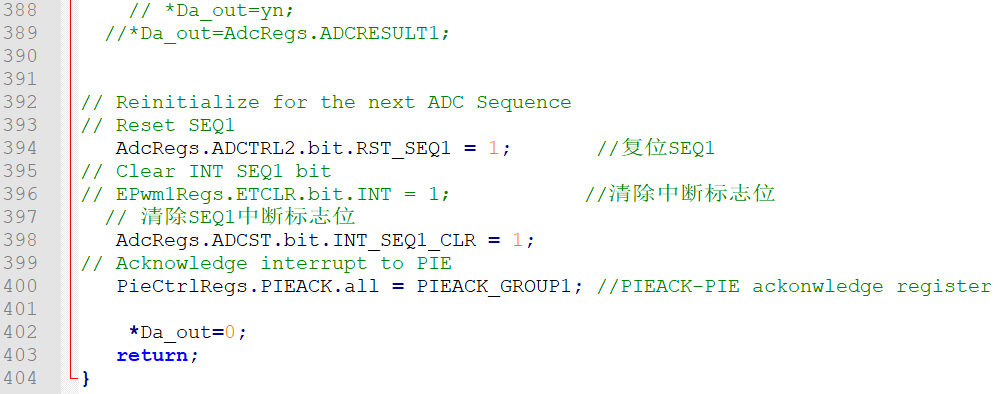
|  |
| --- |
| 图4. 25 实际幅频特性曲线 |

## 4.4 验证算法实时性

***修改代码，测量算法运行时间与采样时间的关系，验证算法实时性。***

在进入中断时给DA高电平，中断结束时给DA低电平，代码如下：





可得到图4. 26，可知算法可以实时性。进一步用光标测量各时间，如图4. 27到图4. 28所示，可知FIR运行所需时间为20微秒，采样时间为50微秒，算法时间小于采样时间且有一定裕余，满足实时性要求。

|  |
| --- |
| 图4. 26 验证算法实时性 |

|  |
| --- |
| 图4. 27 测算采样时间 |

|  |
| --- |
| 图4. 28 测算程序运行时间 |

# 5 实验总结

## 5.1实验中遇到的问题及解决方法

***1.程序运行后示波器无输出***

在第一次编写代码时，运行后发现示波器上无输出。屏蔽各段代码分别调试后发现是采样信号前移部分的代码出错。进一步检查后发现，是没有设置ConvCount变量自增，使得循环无法停止造成的，修改代码后，可以正确运行。

***2. 当输入在1kHz到2kHz之间时，示波器上的输出波形不正确***

在修改程序代码运行后，发现在某个区间内的示波器输出波形不正确，如图5. 1、图5. 2所示。经过老师的指点，得知是因为数据溢出，将输出给DA的数据多右移2位后，波形正常。

|  |  |
| --- | --- |
| 图5. 1 错误波形1 | 图5. 2 错误波形2 |

## 5.2实验心得体会

这次实验中，没有现成的范例程序，需要根据需求，在之前的实验代码上独立完成FIR滤波器的编写。

在这次实验中，我首次深刻感受到“实际是检验真理的唯一标准”的准确性。在第四次实验课上时，我始终纠结于无符号数与有符号数计算的结果，因为无符号数的最高位不表示正负，最大值是65535，但有符号数的最高位表示正负，最大值是32767。由于在滤波器的程序中，对滤波器系数定标后，系数最大值是32767，又要考虑相乘后给DA多少位等等问题。经过思考，在第五次实验课程中，我决定直接在系统上验证，看看字长转换该用什么方式进行。这比“凭空思考”有效率得多，而且能得到正确的结论。

在验证功能时，老师指出我们的滤波器某个区间内溢出。但事实上，我们将相乘的值给长整型变量并右移16位后给DA，理论上应该是所有数据均被接收，仍然出现溢出，说明是32位的长整型也不够存储相乘之后的结果，于是我将该存储变量类型修改为long long int，并右移18位，此时在示波器上出现了正确的波形，而波形的幅度较小。

在这次实验中，由于既要验证采样频率、又要验证算法实时性，因此采用了与上次实验不同的验证方案，即在进入中断程序的一开始给DA高电平，中断中的程序照常执行、但不赋值给DA，中断程序的最后一条语句给DA低电平。在这种方案下，产生的方波周期就是采样频率的周期；而通过查看高电平的持续时间是否足够短，可以验证系统实时性。