

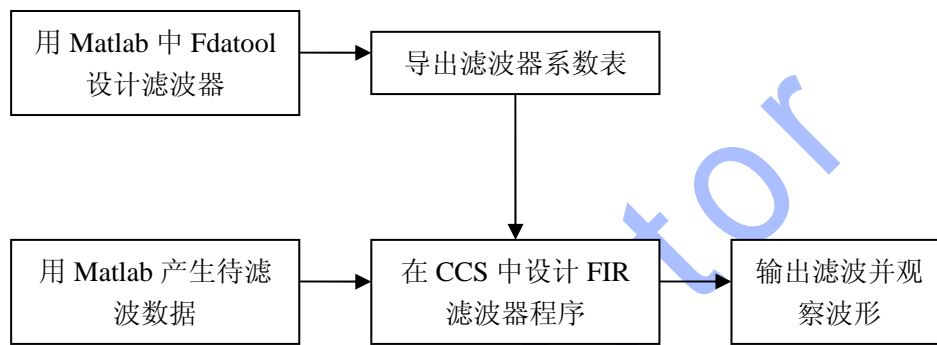
FIR 滤波器的设计

——使用 Matlab 和 CCS

摘要:

本文通过使用 Matlab 软来辅助 CCS 设计 FIR 滤波器, 图文并茂, 讲解详细, 思路清晰。

设计流程如下图



1、使用 Matlab 中的 Fdatool 设计滤波器（本文以 FIR 带通滤波器为例）

1.1、在 Matlab 的 Start 菜单中选择 Toolboxes -> Filter Design -> Filter Design & Analysis Tools(fdatool)，或者在命令行中输入 fdatool 来启动滤波器设计分析器。启动成功后界面如图 1-1 所示。

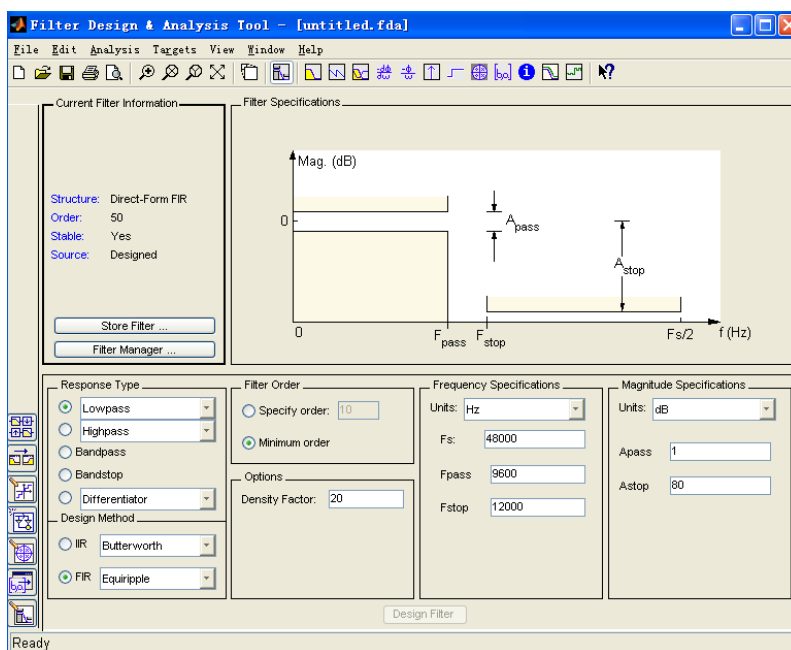


图 1-1

1.2、在选项中选择或输入滤波器参数，然后点击“Design Filter”按钮，完成滤波器的设计。具体参数及设计成功后的结果如图 1-2

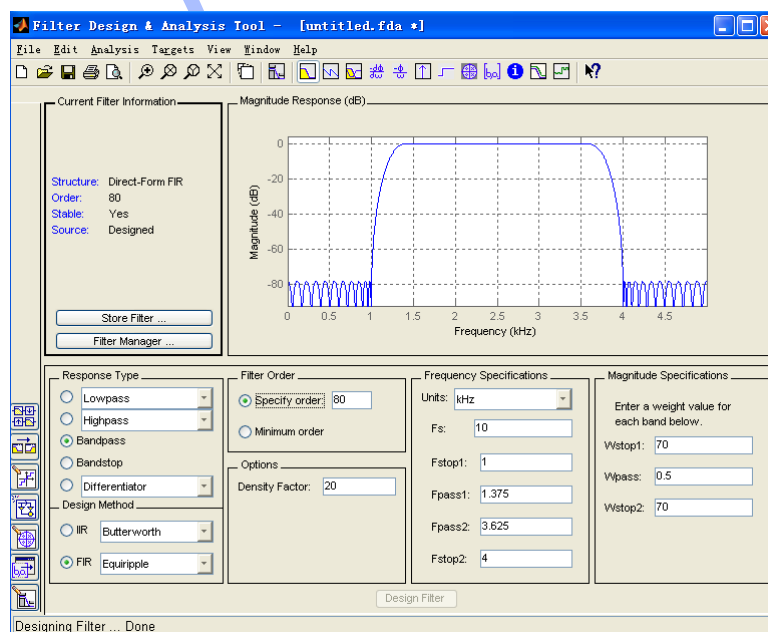


图 1-2

1.3、从 Matlab 中导出 FIR 滤波器系数。

a.在 Fdatool 中，选择 Targets -> Code Composer Studio (tm) IDE，如图 1-3。

b.在出现的对话框中选择输出文件类型为 C header file，输出系数类型为 signed 16-bit integer，如图 1-4 所示。

c.点击 Generate 按钮，选择路径，即可输出前一步设计出的 FIR 滤波器的系数表。（假设生成的系数表文件为 `fdacoefs.h`）

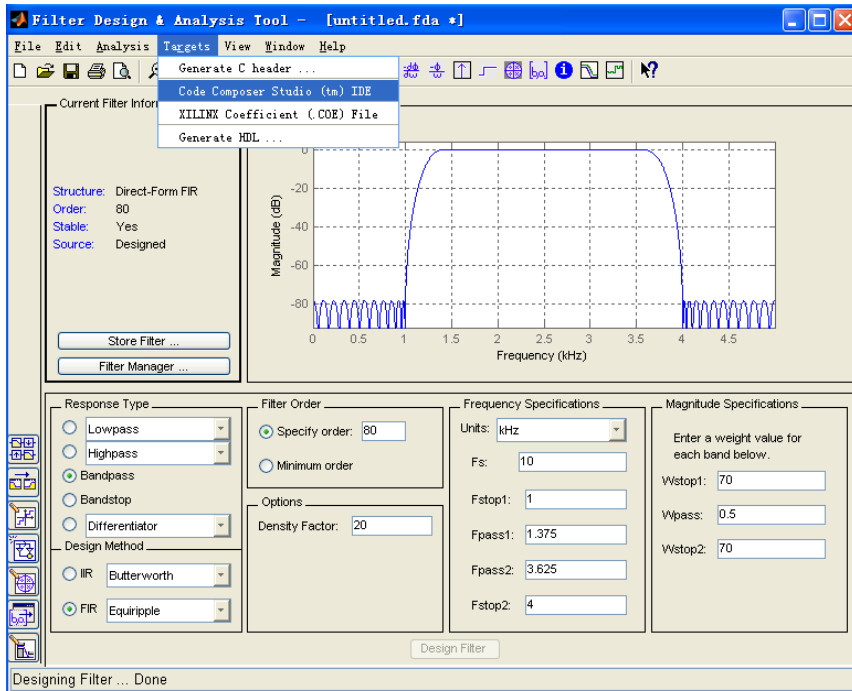


图 1-3

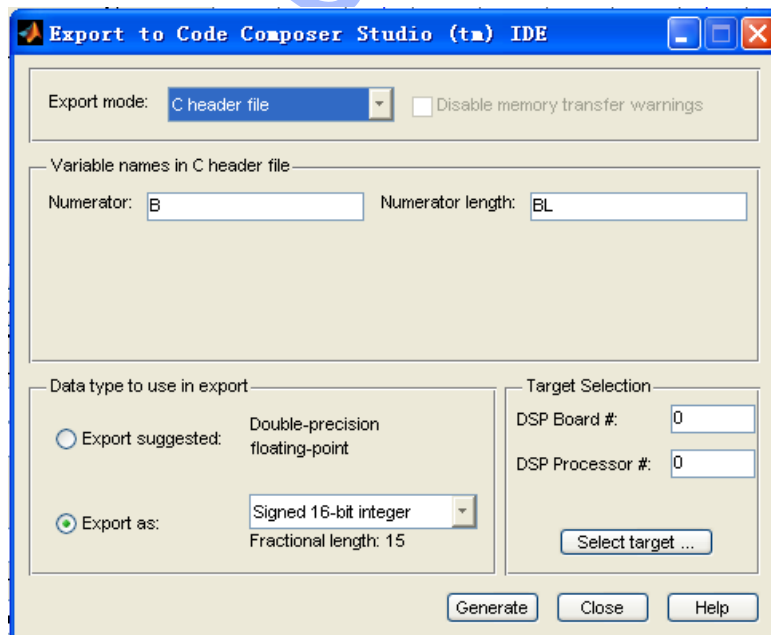


图 1-4

2、利用 Matlab 产生噪声信号用于滤波器测试

将下面代码另存为 M 文件，在 Matlab 中运行后将会生成 input.dat 文件。该数据文件中含有 500Hz、3000Hz、8000Hz 三种频率的信号，用于滤波器滤波效果测试。信号的时域图和频谱分别图 2-1、图 2-2 所示。

```
f11=500;    %/Hz
f12=3000;   %/Hz
f13=8000;   %/Hz
fs=10000;   %/采样 Hz
N=1000      %数据个数
T=1/fs;     %采样周期
n=0:N;

x11=sin(2*pi*f11*n*T);
x12=0.7*sin(2*pi*f12*n*T);
x13=0.5*sin(2*pi*f13*n*T);
x_base=(x11+x12+x13);

%待滤波信号波形
figure(1)
plot(x_base)

%待滤波信号频谱
figure(2)
yff=abs(fft(x_base))
df=n*(fs/N)
plot(df,yff)

xout=x_base/max(x_base);%归一化

xto_ccs=round(32767*xout)

fid=fopen('input.dat','w');%打开文件
fprintf(fid,'1651 1 0 0 0\n');%输出文件头
fprintf(fid,'%d\n',xto_ccs);%输出
fclose(fid);
```

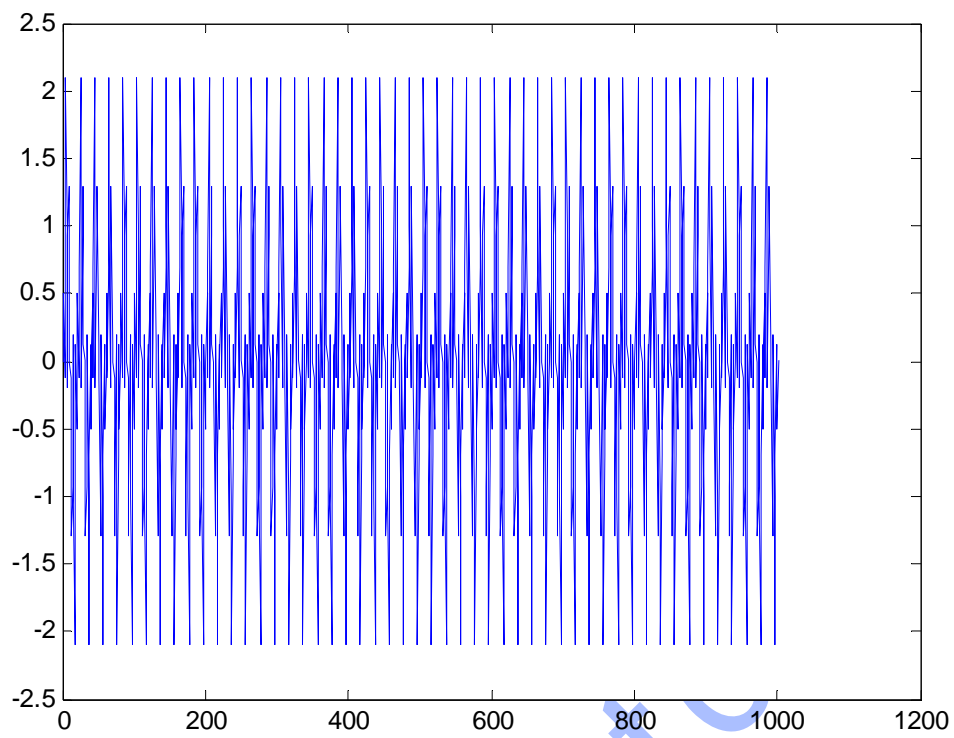


图 2-1

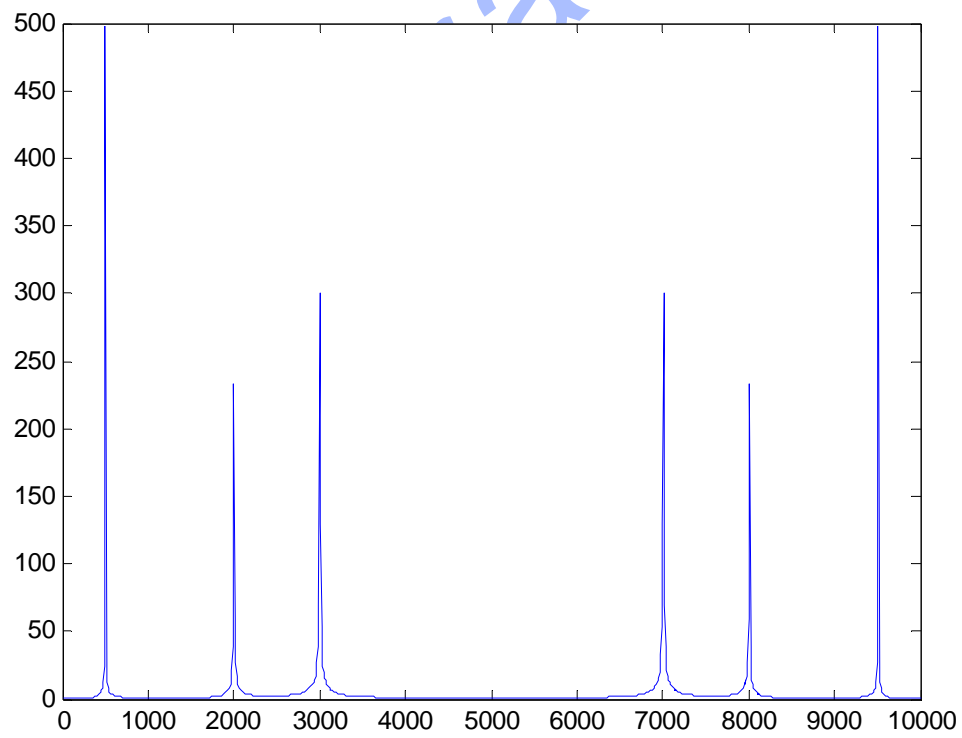


图 2-2

3、在 CCS 中编写 FIR 滤波器程序（本文以 C 语言为例）

3.1、新建工程，作者是以 C54x Simulator 为例，新建工程的过程就不再赘述。

3.2、编写 C 语言源代码并导入工程，如下：

```
#include "stdio.h"
#include "fdacoefs.h"
//fdacoefs.h 为 Matlab 生成的系数表头文件
//如运行不通过，请修改 fdacoefs.h 中的代码，将"#include"这行修改为如下：
//"#include "d:\MATLAB7\extern\include\tmwtypes.h"
//也就是自己机器上的 Matlab 安装的绝对路径
#define N 81 //FIR 滤波器的级数+1，本例中滤波器级数为 80
#define LEN 200 //待滤波的数据长度

long yn;
int input[LEN]; //输入缓冲，在仿真时将从内存载入
int output[LEN]; //输出缓冲，直接存放在内存中

void main()
{
    int i,j;
    int *x;
    for(j=0;j<LEN-1;j++)
    {
        x=&input[j];
        yn = 0;
        for(i=0; i<N-1; i++)
            yn += B[i]*(*x++);
        output[j]=yn>>15;
    }
    while(1);
}
```

3.3、cmd 文件如下，其实是从 CCS 软件中的例子里复制过来的

```
MEMORY
{
    PAGE 0: EPROG:      origin = 0x1400,      len = 0x7c00
                  VECT:      origin = 0xff80,      len = 0x80

    PAGE 1: USERREGS:  origin = 0x60,      len = 0x1c
                  BIOSREGS:  origin = 0x7c,      len = 0x4
                  IDATA:      origin = 0x80,      len = 0x1380
}
```

```
EDATA:      origin = 0x1400,      len = 0x8000
EDATA1:     origin = 0x9400,      len = 0x4c00
}

SECTIONS
{
    .vectors: {} > VECT PAGE 0
    .sysregs: {} > BIOSREGS PAGE 1
    .trcinit: {} > EPROG PAGE 0
    .gblinit: {} > EPROG PAGE 0
    .frt:     {} > EPROG PAGE 0
    .text:    {} > EPROG PAGE 0
    .cinit:   {} > EPROG PAGE 0
    .pinit:   {} > EPROG PAGE 0
    .sysinit: {} > EPROG PAGE 0
    .bss:     {} > IDATA PAGE 1
    .far:     {} > IDATA PAGE 1
    .const:   {} > IDATA PAGE 1
    .switch:  {} > IDATA PAGE 1
    .systemem: {} > IDATA PAGE 1
    .cio:     {} > IDATA PAGE 1
    .MEM$obj: {} > IDATA PAGE 1
    .sysheap: {} > IDATA PAGE 1
    .stack:   {} > IDATA PAGE 1
}
```

3.4、另外还要导入 rts.lib 文件，在...\ti\c5400\cgtools\lib 文件夹下。

4、滤波器仿真测试

4.1、在第 3 部分编译成功后会在”<工程所以目录>/debug”文件夹下产生*.out 文件，在 CCS 软件的 File->Load Program 里打开这个.out 文件（图 4-1）

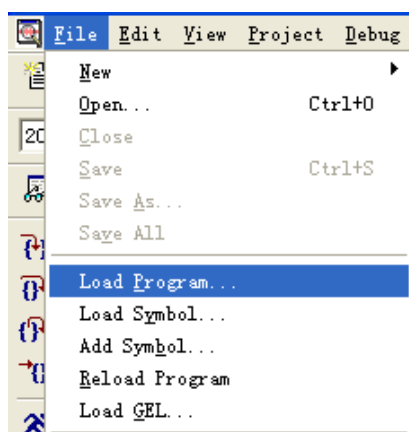


图 4-1

4.2、将滤波器设计文件载入到内存中

4.2.1、选择 File->Data->Load... 打开之前 Matlab 生成的 input.dat 文件（图 4-2）

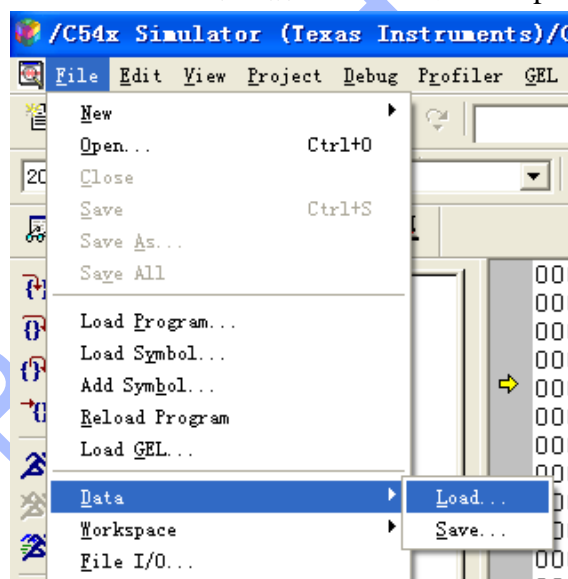


图 4-2

4.2.2、将 Address 设置为 input，Length 设置为 200，Page 设置为 Data（图 4-3）

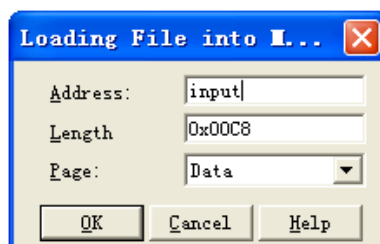



图 4-3

4.3、运行程序，点击  按钮，程序即开始运行

4.4、查看滤波器滤波效果

4.4.1、打开 View->Graph->Time/Frequency (图 4-4)

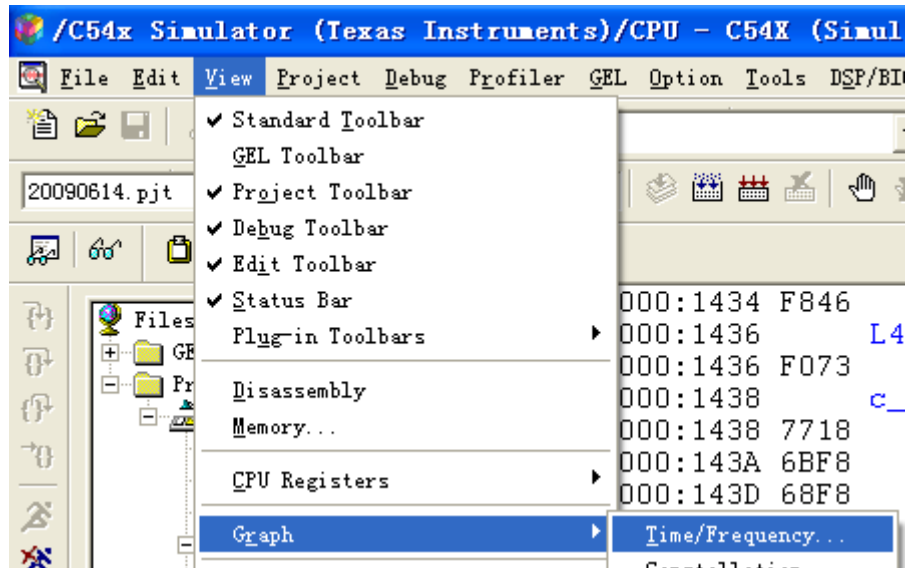


图 4-4

4.4.2、在上一步出现的对话框中，按如图 4-5 设置：

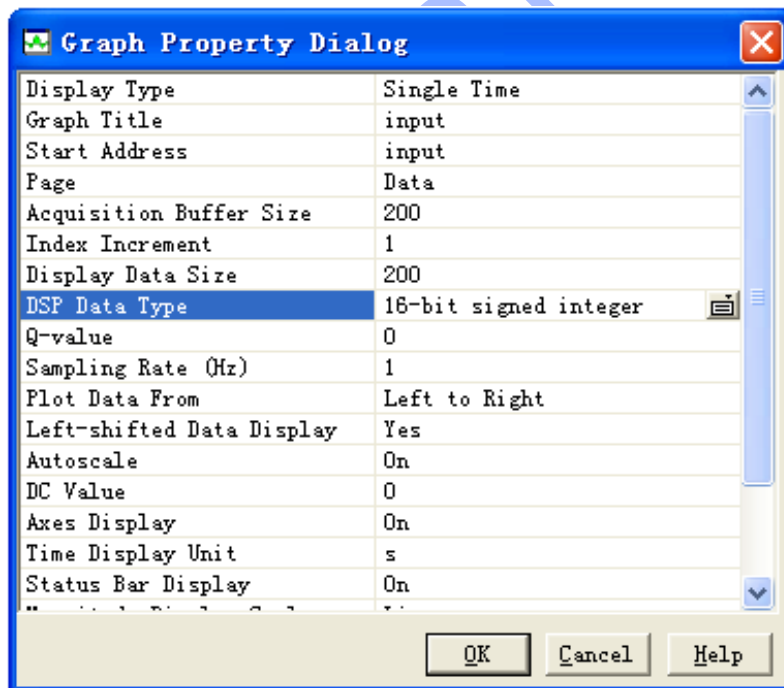


图 4-5

4.4.3、如果出现的波形图太大，在图形上点右键，将”Allow Docking”、”Float in main window”之前的勾去掉，即会变成如图 4-6 的波形：

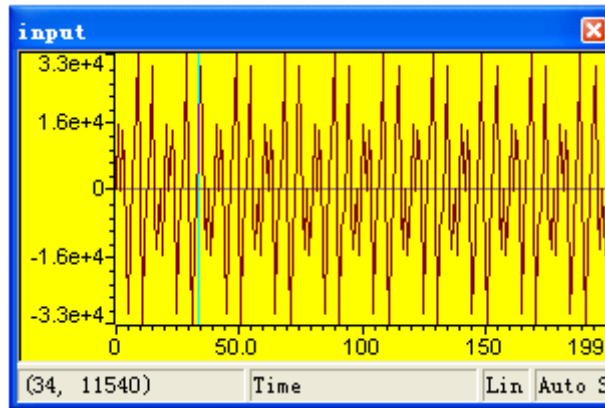


图 4-6

4.4.4、重复前三个步骤，只改变图形选项中的 Display Type、Graph Title、Start Address，使之最后出现如下的图形：

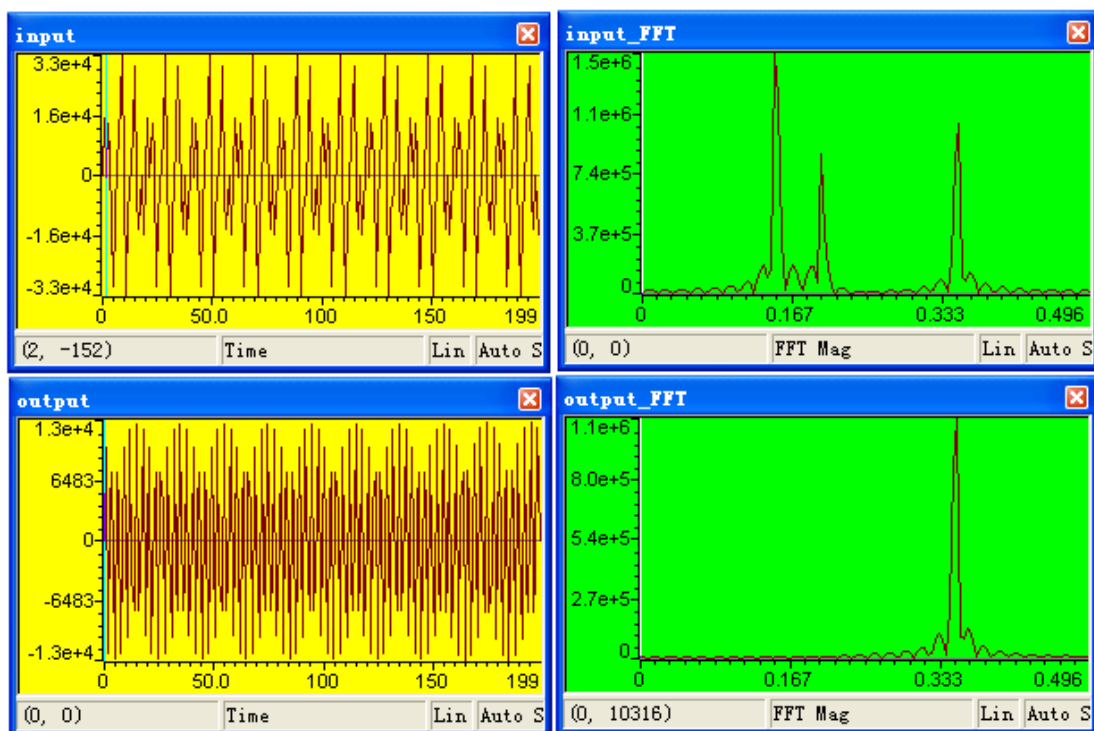


图 4-7

- 左上角：输入数据时域图(Start Address : input)
- 右上角：输入数据频谱 (Display Type : FFT Magnitude)
- 左下角：输出数据时域图(Start Address : output)
- 右下角：输出数据频谱 (Display Type : FFT Magnitude)

5、后记

本文的灵感来自于作者的 DSP 课程设计。刚开始学 DSP 的时候，疯狂地在网上找资料，也的确找到不少有价值的资料，但唯一遗憾的就是都不是非常完美（或者说不完整）。有些看似简单却很重要的步骤经常会被作者省略，可能是作者水平较高，不能体会到初学者的苦衷吧。所以本人在这次 DSP 课程设计之后，在原来的设计报告的基础上做了一定修改和加工，故成此文，希望能给 DSP 初学者一定的帮助吧！如有不足之处，还望读者批评指正！

By Craftor @ 2009-6-26

E-mail: craftor@126.com

QQ: 363901060

By Craftor