FFT 的 C 语言编程

一、上机目的

掌握基 2 时域抽取 FFT 算法及其 C 语言编程。

二、上机内容

1．根据已给参考程序和说明，自行编写蝶形计算部分的程序；

2．用 FFT 计算下面 8 点复数信号的离散傅利叶变换；

x(0)=1+5i；x(1)=2+2i；x(2)=5+2i；x(3)=3+7i；

x(4)=7+6i；x(5)=5+3i；x(6)=9+1i；x(7)=3+8i；

3．分别计算下面两个实信号的 FFT：

16 点信号：

x(0)=0；x(1)=0.7071；x(2)=1；x(3)=0.7071；x(4)=0；x(5)=-0.7071；x(6)=-1；

x(7)=-0.7071；x(8)=0；x(9)=0.7071；x(10)=1；x(11)=0.7071；x(12)=0；

x(13)=-0.7071；x(14)=-1；x(15)=-0.7071；

32 点信号：

x(0)=0；x(1)=1.0898；x(2)=1.7071；x(3)=1.6310；x(4)=1；x(5)=0.2168；

x(6)=-0.2929；x(7)=-0.3244；x(8)=0；x(9)=0.3244；x(10)=0.2929；x(11)=-0.2168；

x(12)=-1；x(13)=-1.631；x(14)=-1.7071；x(15)=-1.0898；x(16)=0；x(17)=1.0898；

x(18)=1.7071；x(19)=1.631；x(20)=1；x(21)=0.2168；x(22)=-0.2929；x(23)=-0.3244；

x(24)=0；x(25)=0.3244；x(26)=0.2929；x(27)=-0.2168；x(28)=-1；x(29)=-1.631；

x(30)=-1.7071；x(31)=-1.0898；

将 FFT 的输出数据代入 Matlab，当 fs=16Hz 时，分别画出上述两个实信号的频谱图(f- $|X(e^{jw})|$)，并给出两个信号中所包含的频率值。

三、上机步骤

1．安装 WinTc 软件；

2．运行 WinTc，输入参考程序并完成剩余的蝶形计算程序；

3．运行程序后进入输入界面，完成上机内容 2

```
8  3        /*信号个数 N=8，蝶形运算级数 M=3
1  5        /*输入 x(0)，实部与虚部用空格隔开
2  2        /*输入 x(1)
……
3  8        /*输入 x(7)
```

8 点 FFT 的参考输出如下：

```
35.000000       34.000000
-8.5355340      5.121320
-15.999999      7.000000
-6.292893       -2.878680
9.000001        -5.999998
-1.464467       0.878680
3.999999        9.000000
-7.707107       -7.121320
```

（右键点击 DOS 窗口的标题栏，可实现输出内容的复制与粘贴；可在属性中修改布局选项中的窗口大小）

4．重复上述步骤完成上机内容 3；

5．撰写上机实验报告。

四、参考程序

```
#include <stdio.h>
#include <math.h>
#define re 0          /*  re=0，用 re 表示实部   */
#define im 1          /*  im=1，用 im 表示虚部   */
main()
{
   float x[128][2],w[2],temp[2];
/*
x[128][2]：  复数变量；
x[i][re]：   第 i 个复数变量的实部；
x[i][im]：   第 i 个复数变量的虚部；
w[2]：       存储旋转因子 W_N^P，w[re]、w[im]分别代表旋转因子的实部和虚部；
temp[2]：    蝶形计算中的临时变量，temp[re]、temp[im]分别代表其实部和虚部；
*/
float arg,wreal,wimag;
/*  arg 存储旋转因子指数 p(数值上相差-2π/N)。wreal 存储 cos(arg)，wimag 存储
-sin(arg) */
float tem,tr,ti;
   int L,M,B,j,i,k, N,N2;
   char c='i';
   scanf("%d   %d",&N,&M);          /*  输入复数信号长度 N，蝶形运算级数 M */
   N2=N>>1;

   for(j=0;j<N;j++)                 /*  输入复数信号实部和虚部  */
   {
    scanf("%f   %f",&x[j][re],&x[j][im]);
   }
   printf("\n");

/*输入倒序*/
for(j=0,i=1;i<N-1;i++)
{
      k=N2;
```

```
    while(k<=j)
    {
        j=j-k;
        k=k>>1;
     }
    j=j+k;
    if(i<j)
    {
        tr=x[j][re];
        ti=x[j][im];
        x[j][re]=x[i][re];
        x[j][im]=x[i][im];
        x[i][re]=tr;
        x[i][im]=ti;
     }
 }

/*FFT 三重循环模块*/
for(L=1; L<=M; L++)          /*   逐级进行计算共 M 级   */
{
    B=1<<L-1;                /*   第 L 级共有 B=2^(L-1) 个不同的旋转因子   */

    arg=-acos(-1)/B;         /*   旋转因子初始化  注释见结尾处  */
    w[re]=cos(arg);
    w[im]=-sin(arg);

    for(j=0; j<B; j++)       /*   j 代表第 L 级不同旋转因子的个数  */
   {
     /*  旋转因子*/
     arg=acos(-1)/B;         /* arg=π/B  */
     wreal=cos(arg);
     wimag= -sin(arg);
     tem=w[re]*wreal-w[im]*wimag;
     w[im]=w[re]*wimag+w[im]*wreal;
     w[re]=tem;
```

```
        for(k=j; k<N; k+=2*B)
          /*第 L 级具有相同旋转因子蝶形计算，每个蝶形相距 2^L=2B 个点*/
         {


              /*   编写蝶形运算程序                                    */
              /*   第 L 级每个蝶形计算的输入节点距离为 B               */
              /*   蝶形运算 { x[i] = x[i] + W_N^P x[i+B]
                            { x[i+B] = x[i] - W_N^P x[i+B]              */
              /*   利用临时存储变量 temp[2]计算 W_N^P x[i+B]           */
              /*   复数运算：(a+bj)(c+dj)=(ac-bd)+(bc+ad)j             */
              /*   temp[re]= ac-bd，temp[im]= bc+ad                    */


           }
       }
  }
    for(j=0;j<N;j++)                    /*输出*/
   {
       printf("%f          %c%f\n",x[j][re],c,x[j][im]);
   }
getch();
}


/*
计算旋转因子说明
P = j*2^(M-L)、 B = 2^(L-1)、 N = 2^M
有 W_N^P = e^(-i(2π/N)P) = e^(-i(jπ/B)) = e^(-iθ) = cosθ - i sinθ
其中 θ = jπ/B
下一个旋转因子旋转角度为 arg
e^(-iθ) * e^(-i(arg))
= (cosθ*cos(arg) - sinθ*sin(arg)) - i(cosθ*sin(arg) + sinθ*cos(arg))
*/
```