

# 浙江大学实验报告

专业：信息工程

姓名：李坤林

学号：3200101135

日期：2022/11/05

地点：教 4-309

课程名称：数字信号处理

指导老师：潘翔

成绩：

实验名称：基 4-FFT 算法编程

实验类型：验证

同组学生姓名：——

## 一、实验目的和要求

FFT 是快速计算 DFT 的一类算法的总称。通过序列分解，用短序列的 DFT 代替长序列的 DFT，使计算量大大下降。基 4-FFT 是混合基 FFT 的一个特例。

通过编写基 4-FFT 算法程序，加深对 FFT 思路、算法结构的理解。

## 二、实验内容和步骤

编写 16 点基 4-FFT 算法的 MATLAB 程序（studentname.m 文件）。

产生 16 点输入序列  $x$ ，用自己的学号作为前 10 点的抽样值，后面补 6 个零值抽样。算出 16 点频谱序列  $X$ ，用 stem( $X$ )显示频谱图形。

撰写实验报告，存为“学号姓名实验三.doc”文件，ftp 上传。

## 三、主要仪器设备

用 MATLAB。

## 四、操作方法和实验步骤

（参见“二、实验内容和步骤”）

## 五、实验数据记录和处理

### 5.1 基 4-FFT 算法思路、流图结构简述如下

#### 1) 算法过程

令序列  $x(n)$  的  $N$  点 DFT 的结果为  $X(k)$ ，且有  $N = 4^m$ ，现按  $((n))_4$  的结果对序列进行分组，得：

$$\begin{aligned}x^{(0)}(n) &= x(4n) \\x^{(1)}(n) &= x(4n + 1) \\x^{(2)}(n) &= x(4n + 2) \\x^{(3)}(n) &= x(4n + 3)\end{aligned} \quad 0 \leq n \leq \frac{N}{4} - 1$$

进而有：

$$\begin{aligned}
X(k) &= \sum_{l=0}^{4^{m-1}-1} x(4l)W_N^{4k} + \sum_{l=0}^{4^{m-1}-1} x(4l+1)W_N^{(4+1)k} + \sum_{l=0}^{4^{m-1}-1} x(4l+2)W_N^{(4l+2)k} \\
&+ \sum_{l=0}^{4^{m-1}-1} x(4l+3)W_N^{(4l+3)k} = \sum_{l=0}^{4^{m-1}-1} x^{(0)}(l)W_{4^{m-1}}^{kk-1} + W_N^k \sum_{l=0}^{4^{m-1}-1} x^{(1)}(l)W_{4^{m-1}}^{k-1} \\
&+ W_N^{2k} \sum_{l=0}^{4^{m-1}-1} x^{(2)}(l)W_{4^{m-1}}^k + W_N^{3k} \sum_{l=0}^{4^{m-1}-1} x^{(3)}(l)W_{4^{m-1}}^k = X^{(0)}((k))_{4^{m-1}} \\
&+ W_N^k X^{(1)}((k))_{4^{m-1}} + W_N^{2k} X^{(2)}((k))_{4^{m-1}} + W_N^{3k} X^{(3)}((k))_{4^{m-1}} \\
0 \leq k \leq N-1 &= 4^m - 1
\end{aligned}$$

其中：

$$X^{(0)}(k) = \text{DFT}_{4^{m-1}} \{x^{(0)}(n)\}$$

$$X^{(1)}(k) = \text{DFT}_{4^{m-1}} \{x^{(1)}(n)\}$$

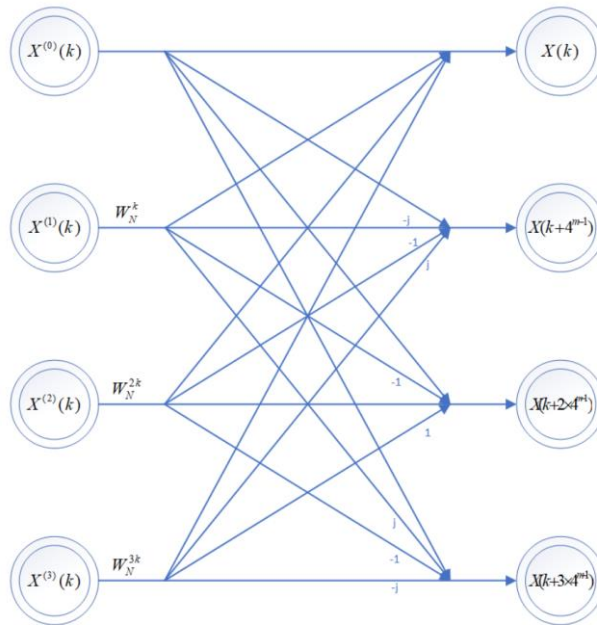
$$X^{(2)}(k) = \text{DFT}_{4^{m-1}} \{x^{(2)}(n)\}$$

$$X^{(3)}(k) = \text{DFT}_{4^{m-1}} \{x^{(3)}(n)\}$$

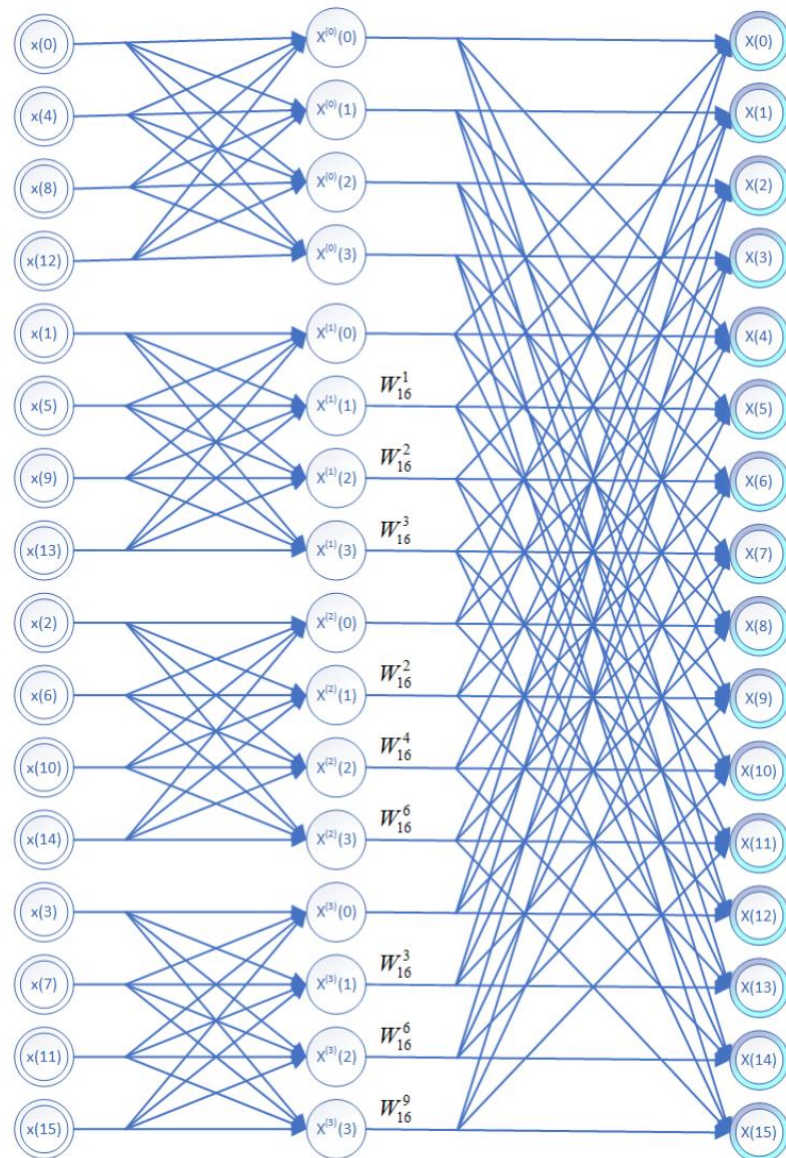
限制  $k$  的取值范围，则上式可以写成：

$$\begin{cases}
X(k) = X^{(0)}(k) + W_N^k X^{(1)}(k) + W_N^{2k} X^{(2)}(k) + W_N^{3k} X^{(3)}(k) \\
X(k + 4^{m-1}) = X^{(0)}(k) - jW_N^k X^{(1)}(k) - W_N^{2k} X^{(2)}(k) + jW_N^{3k} X^{(3)}(k) \\
X(k + 2 \times 4^{m-1}) = X^{(0)}(k) - W_N^k X^{(1)}(k) + W_N^{2k} X^{(2)}(k) - W_N^{3k} X^{(3)}(k) \\
X(k + 3 \times 4^{m-1}) = X^{(0)}(k) + jW_N^k X^{(1)}(k) - W_N^{2k} X^{(2)}(k) - jW_N^{3k} X^{(3)}(k)
\end{cases}$$

## 2) 流图结构



5.2 16 点基 4-FFT 算法的流图绘出如下



5.3 16 点基 4-FFT 算法的 MATLAB 程序（studentname.m）列出如下

%输入学号并补全

```
x = [3 2 0 0 1 0 1 1 3 5, zeros(1, 6)];
```

%基 4 蝶式运算的系数

```
t = [1 1 1 1 ;
```

```
1 -1j -1 1j ;
```

```
1 -1 1 -1 ;
```

```
1 1j -1 -1j];
```

```
x1 = [x(1:4:16) ; x(2:4:16) ; x(3:4:16) ; x(4:4:16)];
```

%每组进行 4 点 FFT

```
X1 = x1*t;
```

%二级运算的旋转因子矩阵

```
w = exp(-1j*pi/8);
```

```
W = [1 1 1 1 ; 1 w w^2 w^3 ; 1 w^2 w^4 w^6 ; 1 w^3 w^6 w^9];
```

%二级 FFT 变换

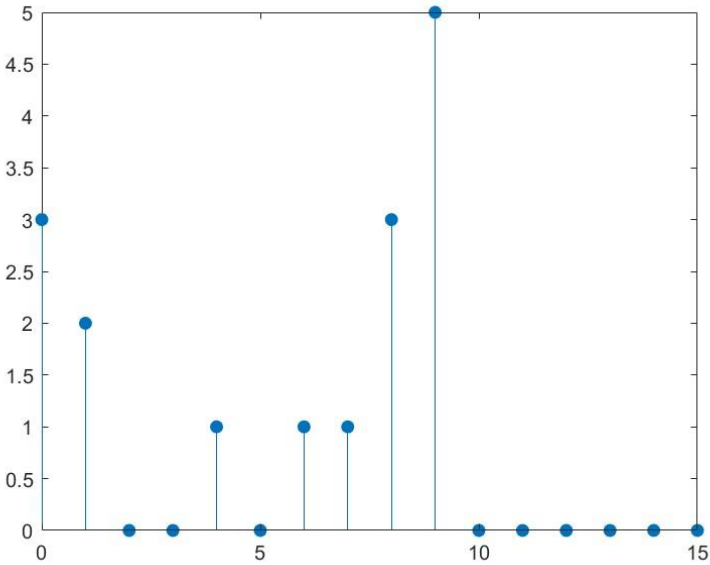
```
X2 = ((X1.*W).'*t').';
X = [X2(1 , 1:4) , X2(2 , 1:4) , X2(3 , 1:4) , X2(4 , 1:4)];
%绘图
figure(1)
n = 0:1:15;
stem(n,x, 'filled');
figure(2)
stem(n,abs(X), "filled");
figure(3)
stem(n,abs(fft(x)), 'filled');
fft(x);
```

5.4 用自己的学号构成的输入序列为（列出数值，插入图形）

1) 数值

```
x = [3 2 0 0 1 0 1 1 3 5 0 0 0 0 0]
```

2) 图形

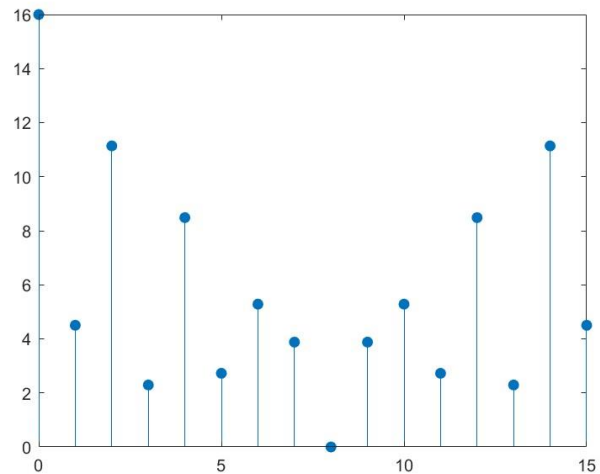


5.5 对应的输出频谱序列为（列出数值，插入图形）

1) 数值

1	2	3	4	5	6	7	8
16.0000+0.0000i	-4.4026-0.9417i	10.6569-3.2426i	-0.8236+2.1407i	6.0000-6.0000i	2.2378+1.5549i	-0.6569-5.2426i	-0.6569-5.2426i
9	10	11	12	13	14	15	16
0.0000 + 0.0000i	2.9884-2.4725i	-0.6569+5.2426i	2.2378-1.5549i	6.0000+6.0000i	-0.8236-2.1407i	10.6569+3.2426i	-4.4026+0.9417i

2) 图形



## 六、实验结果与分析

（观察结果，进行分析讨论）

与系统自带得 `fft` 函数结果进行比较后，可以看出结果具有正确性。

同时，同基 2 时域抽选法相比，基 4 所需的复乘次数降低，而复加次数则有所上升，因为复乘有着较大的运算开销，因此复乘次数的减少有利于改进 FFT 的计算效率。

（注：

A、黑色部分不要改动。

B、蓝色部分根据学生本人情况填写。

C、“五、实验数据记录和处理”和“六、实验结果与分析”根据要求（见红色部分），逐条撰写。

D、从第二页起，在每页头部填写实验名称、姓名、学号，标上页码。不够时自行加页。

E、上传本报告的电子版）

