按时间抽取的基2FFT算法分析及MATLAB实现

张登奇 李宏民 李 丹
(湖南理工学院 信息与通信工程学院)

摘 要：DFT是一种应用广泛的数学变换工具，MATLAB是一款功能强大的科学计算语言。MATLAB提供的FFT函数解决了DFT的快速计算问题，但由于它是内建函数而不能了解到软件实现的过程。文章以按时间抽取的基2FFT算法为例，根据快速傅里叶变换的原理和规律，绘出了算法实现的程序框图，列出了MATLAB环境下软件实现的程序，建立了从算法理论到程序实现的完整概念。

关键词：数字信号处理；离散傅里叶变换；快速傅里叶变换；按时间抽取；MATLAB

Analysis of Decimation-in-time Radix 2 FFT Algorithm and MATLAB Based Realization

Zhang Dengqi Li Hongmin Li Dan
(College of Information & Communication Engineering, Hunan Institute of Science and Technology)

Abstract: DFT is a widely used mathematical transform tool, and MATLAB is a powerful scientific computing language. The FFT function of MATLAB can solve the problems of fast calculation of DFT, but it can't understand the process of software realization because it is a built-in function. Taking the algorithm of the radix-2DIT FFT as the example and based on the principle and rules of FFT, this paper draws the block diagrams of the algorithm realization, lists the procedures of the software realization under MATLAB environment, and also establishes the complete concept from algorithm theories to program realization.

Key words: digital signal processing (DSP); discrete Fourier transform (DFT); fast Fourier transform (FFT); decimation in time (DIT); MATLAB

0 引言

离散傅里变换(DFT)是数字信号处理中重要的数学变换工具，自1965年库利和图基提出基2FFT算法[1]以来，现已有多种快速算法，且还在不断地进行研究和探索。本文以按时间抽取的基2FFT(DIT-FFT)算法为例，介绍FFT算法的基本原理，归纳FFT算法的运算规律，总结FFT算法的编程框图，列出MATLAB环境下软件实现的程序，从而建立起一种从算法理论到程序实现的全过程的完整概念。

1 DIT-FFT算法的基本原理

有限长序列x(n)的N点DFT定义为：$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn}$，式中$W_N=e^{-j\frac{2\pi}{N}}$，其整数次幂简称为旋转因子。直接进行DFT运算大约需要$2N^2$次三角函数计算，$4N^2$次实数乘法计算和$2N(2N-1)$次实数加法计算，且需许多索引和寻址操作[2]。文献[3]列出了直接DFT的MATLAB程序，这种直接DFT运算概念清楚、编程简单，但占用内存大、运算速度低，在实际工作中并不实用。基2FFT算法的基本思想是把原始的N点序列依次分解成一系列短序列，充分利用旋转因子的周期性和对称性，分别求出这些短序列对应的DFT，再进行适当的组合，得到原N点序列的DFT，最终达到减少运算次数，提高运算速度的目的。按时间抽取的基2FFT算法，先是将N点输入序列x(n)在时域按奇偶次序分解成2个$N/2$点序列$x_1(n)$和$x_2(n)$，再分别进行DFT运算，求出与之对应的$X_1(k)$和$X_2(k)$，然后利用图1所示的运算流程进行蝶形运算，得到原N点序列的DFT。只要N是2的整数次幂，这种分解就可一直进行下去，直到其DFT就是本身的1点时域序列。一个完整的8点DIT-FFT运算流程如图2所示[4]。图中的输入序列不再是顺序排列但有规律可循，数组A(存储地址)用于存放输入数据和每级运算的结果。

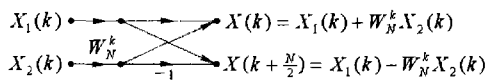

图1 DIT蝶形运算流图符号

2 DIT-FFT算法的运算规律及编程思想

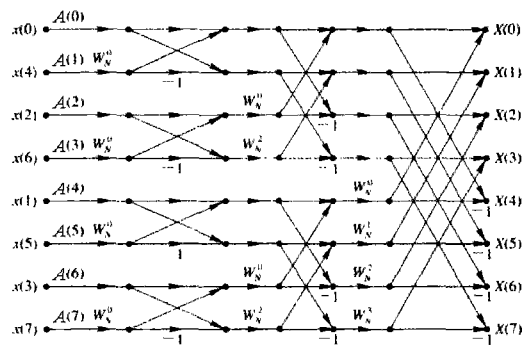

图2 8点DIT-FFT运算流程

为了编写DIT-FFT算法的运算程序，首先要分析其运算规律，总结编程思想并绘出程序框图。由图2可知，DIT-FFT算法的运算过程很有规律。

2.1 原位计算

对$N=2^M$点的FFT共进行M级运算，每级由$N/2$个蝶形运算组成。在同一级中，每个蝶的输入数据只对本蝶有用，且输出节点与输入节点在同一水平线上，这就意味着每算完一个蝶后，所得数据可立即存入原输入数据所占用的数组元素(存储单元)，这种原位(址)计算的方法可节省大量内存。

2.2 蝶形运算

实现FFT运算的核心是蝶形运算，找出蝶形运算的规律是编程的基础。蝶形运算是分级进行的；每级的蝶形运算可以按旋转因子的指数大小排序进行；如果指数

大小一样则可从上往下依次蝶算。对$N=2^M$点的FFT共有M级运算，用L表示从左到右的运算级数($L=1,2,\cdots,M$)。第L级共有$B=2^{L-1}$个不同指数的旋转因子，用R表示这些不同指数旋转因子从上到下的顺序($R=0,1,\cdots,B-1$)。第R个旋转因子的指数$P=2^{M-L}R$，旋转因子指数为P的第一个蝶的第一节点标号k从R开始，由于本级中旋转因子指数相同的蝶共有2^{M-L}个，且这些蝶的相邻间距为2^L，故旋转因子指数为P的最后一个蝶的第一节点标号k为：$(2^{M-L}-1)\cdot 2^L+R=N-2^L+R$，本级中各蝶的第二个节点与第一个节点都相距$B$点。应用原位计算，蝶形运算可表示成如下形式：$A_L(k)\Leftarrow A_{L-1}(k)+W_N^P A_{L-1}(k+B)$

$$A_L(k+B)\Leftarrow A_{L-1}(k)-W_N^P A_{L-1}(k+B)$$

总结上述运算规律，可采用如下运算方法进行DIT-FFT运算。首先读入数据，根据数据长度确定运算级数M，运算总点数$N=2^M$，不足补0处理。然后对读入数据进行数据倒序操作。数据倒序后从第1级开始逐级进行，共进行M级运算。在进行第L级运算时，先算出该级不同旋转因子的个数$B=2^{L-1}$(也是该级中各个蝶形运算两输入数据的间距)，再从$R=0$开始按序计算，直到$R=B-1$结束。每个R对应的旋转因子指数$P=2^{M-L}R$，旋转因子指数相同的蝶从上往下依次逐个运算，各个蝶的第一节点标号k都是从R开始，以2^L为步长，到$N-2^L+R$(可简取极值$N-2$)结束。考虑到蝶形运算有两个输出，且都要用到本级的两个输入数据，故第一个输出计算完毕后，输出数据不能立即存入输入地址，要等到第二个输出计算调用输入数据完毕后才能覆盖。这样数据倒序后的运算可用三重循环程序实现。整个运算流程如图3所示。

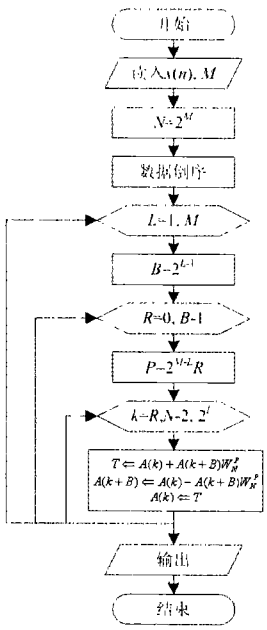

图3 DIT-FFT运算程序框图

2.3 序列倒序

为了保证运算输出的$X(k)$按顺序排列，要求序列$x(n)$倒序输入，即在运算前要先对输入的序列进行位序颠倒。如果总点数为$N=2^M$的$x(n)$的顺序数是用M位二进制数表示，则倒序数只需将顺序数的二进制位倒置即可[5]，按照这一规律用硬件电路和汇编语言很容易产生倒序数。但用MATLAB等高级语言实现倒序时，直接倒置二进制数位的方法不可取，还须找出产生倒序的十进制规律。

将十进制顺序数用I表示，与之对应的二进制数用IB表示。十进制倒序数用J表示，与之对应的二进制数用JB表示。JB是IB的位倒置结果，十进制顺序数I增加1，相当于IB最低位加1且逢2向高位进1，即相当于JB最高位加1且逢2向低位进1。JB的变化规律反映到J的变化分二种情况：如果JB的最高位是0($J<N/2$)，则直接由加1($J\Leftarrow J+N/2$)得到下一个倒序值；如果JB的最高位是1($J\geq N/2$)，则要先将最高位变0($J\Leftarrow J-N/2$)，再在次高位加1($J\Leftarrow J+N/4$)。但次高位加1时，同样要判断0、1值，如果是0($J<N/4$)，则直接加1($J\Leftarrow J+N/4$)，否则要先将次高位变0($J\Leftarrow J-N/4$)，再判断下一位。依此类推，直到完成最高位加1，逢2向右进位的运算。利用这一算法可按顺序数I的递增顺序，依次求得与之对应的倒序数J。为了节省内存，数据倒序可原址进行，当$I=J$时不需要交换，当$I\neq J$时需要交换数据。另外，为了避免再次调换前面已经调换过的一对数据，只对$I<J$的情况进行数据交换即可实现数据倒序操作。图3中数据倒序的程序框图如图4所示。

3 MATLAB程序实现

MATLAB提供的FFT函数是一个计算DFT的智能程序，能自动选择快速算法进行DFT运算，由于它是一个内建函数，用type命令看不到程序代码。为了体现编程思想，下面结合DIT-FFT程序框图，列出MATLAB环境下的实现程序。MATLAB的数组元素按序存储，可用下标寻访，但下标是从1开始的，所以在MATLAB程序中，寻访数组中的元素(数据)时，下标要在原序号上加1。旋转因子可按指数预先计算出来并存放到数组WN中，虽然占用了一些内存，但程序运行时可直接寻访调用，无需反复计算，可进一步提高运算速度。用MATLAB实现DIT-FFT算法的程序如下：

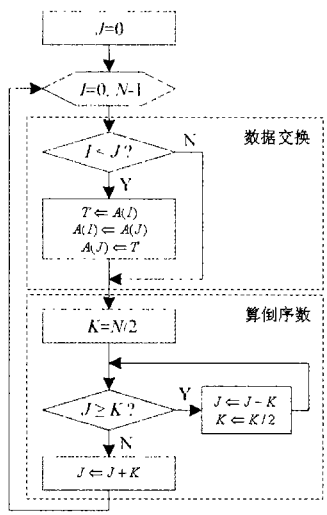

图4 数据倒序程序框图

```
%基2DIT-FFT运算的MATLAB程序
clc;close all;clear;format compact;
%输入数据并计算常量
xn=[0,1,2,3,4,5,6,7];%可取任意序列
M=nextpow2(length(xn)), N=2^M,
for m=0:N/2-1;%旋转因子指数范围
    WN(m+1)=exp(-j*2*pi/N)^m;%计算旋转因子
end
A=[xn,zeros(1,N-length(xn))]; %数据输入
disp('输入到各存储单元的数据:'),disp(A);
```

```
%数据倒序操作
J=0;%给倒序数赋初值
for I=0:N-1;%按序交换数据和算倒序数
    if I<J;%条件判断及数据交换
        T=A(I+1);A(I+1)=A(J+1);A(J+1)=T;
  end
  %算下一个倒序数
  K=N/2;
  while J>=K;
    J=J-K;K=K/2;
  end
  J=J+K;
end
disp('倒序后各存储单元的数据:'),disp(A);
%分级按序依次进行蝶形运算
for L=1:M;%分级计算
  disp('运算级次:'),disp(L);
  B=2^(L-1);
  for R=0:B-1;%各级按序蝶算
    P=2^(M-L)*R;
    for K=R:2^L:N-2;%每序依次计算
      T=A(K+1)+A(K+B+1)*WN(P+1);
      A(K+B+1)=A(K+1)-A(K+B+1)*WN(P+1);
      A(K+1)=T;
    end
  end
  disp('本级运算后各存储单元的数据:'),disp(A);
end
disp('输出各存储单元的数据:'),Xk=A,
disp('调用fft函数运算的结果:'),fftxn=fft(xn,N),
```

该程序严格按程序框图编写，思路清晰、容易理解，程序的运行过程在命令窗中一目了然。通过与fft函数运算的结果比对，程序编写正确，运算结果可靠。

4 结束语

FFT算法是在实践中应用广泛的成熟算法，MATLAB提供的FFT函数可以自动选择快速算法进行DFT计算，但由于它是内建函数而没有体现编程思想。文中介绍的FFT算法原理，归纳的运算规律，绘制的程序框图，列出的实现程序，既是学习基2DIT-FFT算法的重要内容，对其他算法从基本理论到软件实现的全过程学习也提供了借鉴和参考。

参考文献：

[1] Cooley J W, Tukey J W. An algorithm for the machine computation of complex Fourier series[J]. Mathematics of Computation, 1965,19(4):297-301.
[2] Proakis J G, Manolakis D G. 数字信号处理[M]. 方艳梅, 刘永清, 译. 北京:电子工业出版社, 2007:379-391.
[3] 陈怀琛, 吴大正, 高西全. MATLAB及在电子信息课程中的应用[M]. 第二版. 北京:电子工业出版社, 2004: 199-200.
[4] 刘顺兰, 吴杰. 数字信号处理[M]. 第二版. 西安:西安电子科技大学出版社，2008:119-123.
[5] 高西全, 丁玉美. 数字信号处理[M]. 第三版. 西安:西安电子科技大学出版社，2008:110-118.

作者简介：
张登奇(1968－)，男，湖南临湘人，硕士，
湖南理工学院信息与通信工程学院副教授。
主要研究方向：信号与信息处理
电话：13873099252
电子信箱：hnyyzdq@163.com
通信地址：湖南省岳阳市湖南理工学院信息与通信工程学院 张登奇（414006）