DRM系统中OFDM模块的高效实现

何 苗，董在望，徐淑正

(清华大学电子工程系 北京 海淀区 100084)

【摘要】结合世界性数字广播(Digital Radio Mondiale，DRM)系统具有多种鲁棒性模式和带宽占用模式的特点，设计了适用于DRM系统的快速傅里叶变换算法。与其他快速傅里叶变换算法相比较，这种通用长度N的同址、顺序素因子算法点数选取灵活，运算精度高，可以实现同址、顺序运算，因此其存储量和数据传递次数少，运算量小，运算速度快，且程序结构规整，利于软、硬件实现。该算法也可用于其他采用正交频分复用调制技术的系统。

关 键 词 世界性数字广播； 快速傅里叶变换； 正交频分复用； 素因子算法
中图分类号 TN 911.72 文献标识码 A

High-Efficient Implementation of the OFDM Module for DRM Systems

HE Miao，DONG Zai-wang，XU Shu-zheng

(Department of Electronic Engineering, Tsinghua University Haidian Beijing 100084)

Abstract Combining the robustness modes and bandwidth occupation modes of Digital Radio Mondiale (DRM) systems, a method of computing Fast Fourier Transform (FFT) algorithm is presented for DRM systems. In comparison with other FFTs. This algorithm produces the in-place, in-order algorithm for variable transform sizes with high operational precision. Therefore it saves storage space, operation and run time. In addition, the program structure of this algorithm is very regular, thus it can be easily achieved by software and hardware. This algorithm is applicable to Orthogonal Frequency Division Multiplexing (OFDM) modules of other systems.

Key words digital radio mondiale; fast Fourier transform; orthogonal frequency division multiplexing; prime factor algorithm

1998年3月，世界上20多个国家的广播机构联合起来，设立了“世界性数字广播”(Digital Radio Mondiale，DRM)组织，提出了一种非专利的用于将现有AM广播数字化的技术标准[1-2]，共同开发AM波段的数字声音广播[3]。为适应30 MHz以下各广播频段的传输需要，该标准规定了多种数据率、带宽与调制方式。由于系统采用COFDM技术[4]，因此需要设计多种点数的快速傅里叶变换(Fast Fourier Transform，FFT)[5-6]。

FFT算法可以分为两大类：基于递归的算法和基于卷积的算法。第一类算法通过递归的方法对离散傅里叶变换(Discrete Fourier Transform，DFT)逐级分解，从而简化计算，主要有Cooley-Tukey算法、分裂基算法、Rader-Brenner算法。第二类算法将DFT转换成卷积进行计算，主要有Winograd(WFTA)算法[7]、素因子算法(Prime Factor Algorithm，PFA)[8]。DRM系统所规定的多种FFT点数具有素因子的特点，因此本文主要讨论素因子算法的实现。

1 素因子算法

素因子算法(PFA)[8]与Winograd算法[7]均基于卷积运算，不同的是Winograd算法是通过嵌套算法把所需的乘法部分合成在一起，放在算法中间，从而减少了乘法次数。但算法结构复杂，不能实现同址运算，对结果需要重新排序，存储量和数据传递次数较大，不利于软、硬件实现。而素因子算法由于采用素因子分解，运算中省去了级间旋转因子所需的乘法，运算量小，运算速度快，但由此会导致结果数据的乱序，需要重新排列。

对于素因子算法的不足，文献[9]提出了两种素因子算法：(1) 采用同址、顺序的方法，且仅需使用一对数组，减少了存储量和数据传递次数，提高了

收稿日期：2006－04－28
作者简介：何 苗(1979－)，男，硕士生，主要从事DRM接收机方面的研究.

运算速度，但需要对每一个特定的长度N进行单独设计。(2) 通过整序常数，并增加一对数组，实现对运算结果的重排。但额外增加了一组存储器，且降低了运算速度。

针对此情况，文献[10]提出了通用长度N的同址、顺序素因子算法，仅需一对数组就能够实现通用长度N的同址、顺序计算。与文献[9]提出的两种素因子算法相比，它省去了最后的顺序重排，且减少了存储量和数据传递次数，提高了运算速度。

文献[10]论证了从一维DFT映射到二维DFT的情况，本文讨论一维DFT映射到M维DFT这种更一般的情况。

$x(n)$的N点离散傅里叶变换为：

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk} \tag{1}$$

式中 $k=0,1,\cdots,N-1$；$W_N=\mathrm{e}^{-\mathrm{j}2\pi/N}$。素因子算法中的 N 为数集 $\{2,3,4,5,7,8,9,16\}$ 中 M 个两两互素因子的乘积，即：

$$N=N_1N_2\cdots N_M \tag{2}$$

式中 $M\leq 4$；$(N_i,N_j)=1$，$(i\neq j)$，$i,j=1,2,\cdots,M$。

为了得到顺序的输出，而避免通过整序常数对变换的结果重新排序，在一维DFT映射成 M 维DFT时，对 n 和 k 没有采用文献[9]中的下标映射关系，而是采用了相同的下标映射：

$$n=<\sum_{i=1}^{M}(\frac{N}{N_i})n_i>N \tag{3}$$

$$k=<\sum_{i=1}^{M}(\frac{N}{N_i})k_i>N \tag{4}$$

式中 $n_i,k_i=0,1,\cdots,N_i-1$；$i=1,2,\cdots,M$。将式(3)、式(4)代入式(1)中，得：

$$\hat{X}(k_1,k_2,\cdots,k_M)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\cdots\sum_{n_M=0}^{N_M-1}\hat{x}(n_1,n_2,\cdots,n_M)\times W_{N_1}^{(N/N_1)n_1k_1}W_{N_2}^{(N/N_2)n_2k_2}\cdots W_{N_M}^{(N/N_M)n_Mk_M} \tag{5}$$

式中 $\begin{cases}\hat{x}(n_1,n_2,\cdots,n_M)=x\left(<\sum_{i=1}^{M}(\frac{N}{N_i})n_i>N\right)\\ \hat{X}(k_1,k_2,\cdots,k_M)=X\left(<\sum_{i=1}^{M}(\frac{N}{N_i})k_i>N\right)\end{cases}$。采用如下的置换关系，令：

$$k_i'=\left\langle(N/N_i)k_i\right\rangle_{N_i}=\left\langle\left\langle N/N_i\right\rangle_{N_i}\left\langle k_i\right\rangle_{N_i}\right\rangle_{N_i}=\left\langle\left\langle N/N_i\right\rangle_{N_i}k_i\right\rangle_{N_i} \tag{6}$$

式中 $k_i=0,1,\cdots,N_i-1$；$i=1,2,\cdots,M$。将式(6)代入式(5)中，得：

$$\hat{X}(k_1',k_2',\cdots,k_M')=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\cdots\sum_{n_M=0}^{N_M-1}\hat{x}(n_1,n_2,\cdots,n_M)\times W_{N_1}^{n_1k_1'}W_{N_2}^{n_2k_2'}\cdots W_{N_M}^{n_Mk_M'} \tag{7}$$

式中 $\begin{cases}\hat{x}(n_1,n_2,\cdots,n_M)=x\left(<\sum_{i=1}^{M}(\frac{N}{N_i})n_i>N\right)\\ \hat{X}(k_1',k_2',\cdots,k_M')=X\left(<\sum_{i=1}^{M}(\frac{N}{N_i})k_i>N\right)\end{cases}$，$k_i'=0,1,\cdots,N_i-1$；$i=1,2,\cdots,M$。

由式(7)可得，每个小 N_i 点DFT的输出都是按照 k_i' 进行排序的。因此要得到顺序的输出，就要按照 k_i 对每个小 N_i 点DFT的输出重新排序，且由式(6)可得，对不同的 N，小 N_i 点DFT的输出排序不同。

按照以上的推导过程，对于输入数据，由输入映射式(3)，计算出 N_i 个输入数据的索引值，存入数组 $I[\ ]$。再由置换关系式(6)，对 N_i 个输出数据的地址重新排序，并将得到的置换序列存入数组 $NM[\ ]$。最后由输出映射式(4)，结合置换序列数组 $NM[\ ]$，得到 $II[NM[\ ]]$，从而计算出 N_i 个输出数据的索引值，存入数组 $II[\ ]$。这样，即可实现通用长度 N 的同址、顺序计算。

采用通用长度N的同址、顺序素因子算法，需要考虑以下三个方面：

(1) 对于 M 维DFT的计算，有一种简单而高效的算法[9]，把 M 维DFT当作二维DFT来处理。即先计算 $N_2N_3\cdots N_M$ 个长度为 N_1 的DFT，再计算 $N_1N_3\cdots N_M$ 个长度为 N_2 的DFT，……最后计算 $N_1N_2\cdots N_{M-1}$ 个长度为 N_M 的DFT。基于这种思想，M 次DFT计算中的每一次只需要由式(3)、式(4)得到一组二维的 n、k的下标映射关系，而非 M 维的下标映射关系，从而降低了运算复杂度。

(2) 不同的因子 N_i 和不同的因子数会导致不同的计算量，需要合理选取变换长度 N [9]。

(3) 对于小 N_i 点DFT的输出顺序重排，其中，2点DFT的输出不需要重排；由于3、4、5点DFT的输出顺序较为简单，对于固定的 N，可以先计算出输出顺序的值，然后写入程序中；如果 N 是变化的，根据结构化程序设计的思想，程序会根据式(6)得到小 N_i 点DFT的输出地址表，通过查表法进行顺序重排；而7、8、9、16点DFT的输出顺序较为复杂，均采用查表法进行顺序重排。

图1是通用长度 N 的同址、顺序素因子算法的流程图。

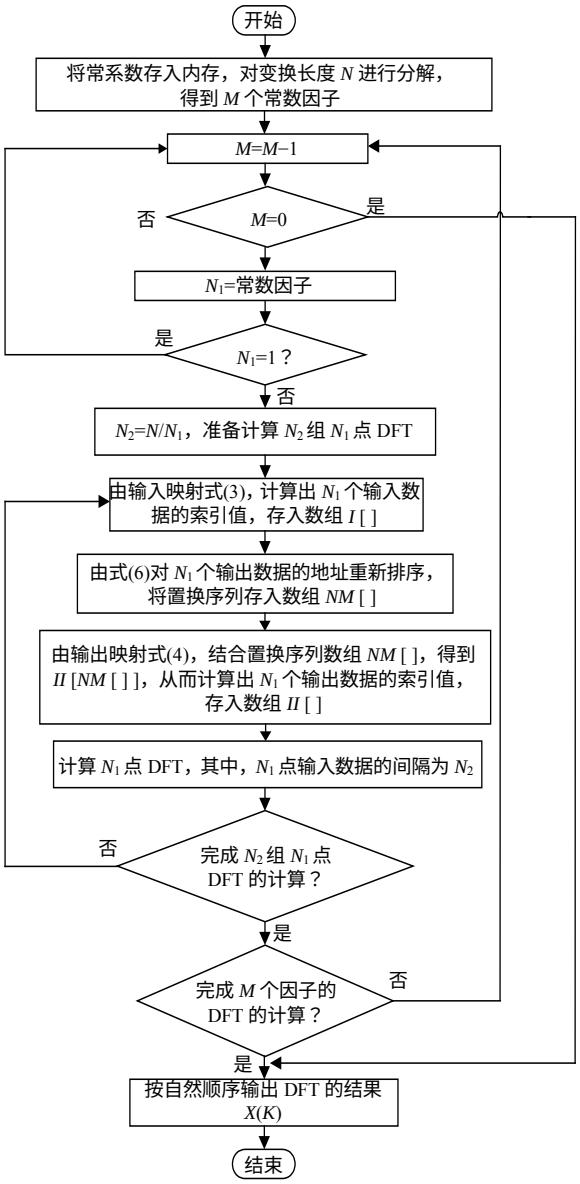

图1　通用长度N 的同址、顺序素因子算法流程图

图2是对不同FFT算法运算时间的比较曲线。其中，各种FFT算法的输入数据为复数；PFA2是本文采用的文献[10]中的算法；PFA1是文献[9]中通过整序常数重新排序的算法；WFTA是Winograd算法；Split-radix是分裂基算法。图中的数据是通过Automated QA公司开发的商业软件AQTime对各种算法所消耗的时钟周期数的测试结果。

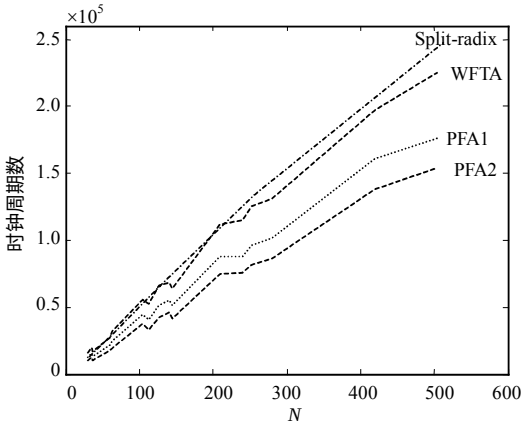

图2　各种FFT算法所消耗的时钟周期数

2　DRM系统OFDM模块的高效实现

在DRM系统中采用了OFDM调制方式，DRM信号有四种稳定性模式，对应于不同的信道条件。为了适应各种需要，系统还使用了六种带宽占用方式。不同模式和带宽对应着不同的数据率。

对于不同模式的DRM传输信号，每个码元包含的子载波数不同，且均不为2的整数次幂，对应的子载波数如表1所示[1]。由于改进的素因子算法在一定程度上结合了基于2的整数次幂的FFT算法和Winograd算法的优点，因此更适合DRM系统OFDM模块的实现，采用这种PFA有以下三个优点：

(1) 与基2、基4、分裂基算法相比，PFA的点数选取灵活，运算精度高。

(2) 与WFTA相比，加法次数大大减少，且程序结构简单，仅需一对数组就可以实现同址、顺序运算，因此存储量和数据传递次数明显减少，利于软、硬件实现。

(3) 通过实际验证，通用长度 N 的同址、顺序素因子算法比其他FFT算法的速度快。

因此，综合考虑各方面因素，最后选用了通用长度N的同址、顺序素因子算法实现OFDM模块。

表1　标准中不同模式的子载波数[1]和选用的FFT点数

稳定性模式	频带占用模式					
	0	1	2	3	4	5
A	101→112	113→120	202→210	226→240	410→420	458→504
B	91→112	103→112	182→210	206→210	366→420	410→420
C	—	—	—	138→144	—	280→280
D	—	—	—	88→90	—	178→180

如前所述，采用PFA，需要合理地选取变换长度 N [9]。以计算 $N = N_1 N_2$ 点DFT为例，若单独计算 N_1、N_2 点小 N，WFTA所需的乘法数分别为 M_1、M_2，加法数为 A_1、A_2，那么计算 N 点DFT所需的乘法数和加法数分别为：

$$\begin{cases} M_P = N_1 M_2 + N_2 M_1 \\ A_P = N_1 A_2 + N_2 A_1 \end{cases} \tag{8}$$

式中 $(N_1, N_2) = 1$。由式(8)可以看出不同的因子 N_i 和不同的因子数会导致不同的乘加数，即存在大 N 点FFT乘加数小于小 N 点乘加数的情况，再加上索引、寻址和数据传递所消耗的时间，大 N 点FFT的运算时间有可能少于小 N 点FFT的时间，因此需要合理地选取变换长度 N。如：$105 = 1*3*7*5$，$112 = 16*1*7*1$，但112点FFT的乘加数明显少于105点FFT的乘加数，且112点FFT的运算速度更快，35和36点、140和144点的FFT运算也都有这样的特点。计算FFT所消耗的时钟周期数可以参见图2。

根据DRM标准[1]和PFA运算量的上述特点，由此选定的FFT点数见表1。表中，“$\rightarrow$”左边表示标准中每一个码元的子载波数，“$\rightarrow$”右边表示选用的FFT点数。需要说明的是91→112、101→112、103→112和138→144这几种情况，尽管105点与标准中子载波数91、101、103点更接近，但是105点FFT的运算速度低于112点FFT的速度，因此选择112点作为变换长度 N。同理，没有选择140点，而选择了144点作为变换长度 N。

3 结 论

本文通过对各种FFT算法进行性能分析和比较，针对DRM系统中存在多种稳定性模式和带宽占用模式的特点，设计了适用于DRM系统的快速傅里叶变换(FFT)算法。与其他FFT算法相比较，这种通用长度 N 的同址、顺序素因子算法更适合DRM系统OFDM模块的实现。此外，该算法也可用于其他采用OFDM调制技术的系统。

参 考 文 献

[1] ETSI ES 201 980 v2.1.1. ETSI Standard. Digital radio mondiale (DRM)[S]. System Specification, 2004.

[2] ETSI ES 101 968 v1.2.1. ETSI Standard. Digital radio mondiale (DRM)[S]. Data Applications Directory, 2004.

[3] 李 栋. 数字声音广播[M]. 北京: 北京广播学院出版社, 2001.

[4] STOTT J. Digital Radio Mondiale: Key technical features[J]. Electronics & Communication Engineering Journal, 2002, 14: 4-14

[5] WEINSTEIN S B, EBERT P M. Data transmission by frequency-division multiplexing using the discrete Fourier transform[J]. IEEE Trans. Commun. Tech., 1971, COM-19: 628-634.

[6] 王文博, 郑 侃. 宽带无线通信OFDM技术[M]. 北京: 人民邮电出版社, 2003.

[7] WINOGRAD S. On computing the discrete Fourier transform[J]. Math. Comput., 1978, 32: 175-199.

[8] KOLBA D P, PARKS T W. A prime factor FFT algorithm using high speed convolution[J]. IEEE Trans. Acoust., Speech, Signal Processing, 1977, ASSP-25: 281-294.

[9] BURRUS C S, ESCHENBACHER P W. An in-place, in-order prime factor FFT algorithm[J]. IEEE Trans. Acoust., Speech, Signal Processing, 1981, ASSP-29: 806-817.

[10] ROTHWEILER J H. Implementation of the in-order prime factor transform for variable sizes[J]. IEEE Trans. Acoust., Speech, Signal Processing, 1982, ASSP-30: 105-107.

编 辑 漆 蓉