快速傅里叶变换中计算倒序的新思路

刘大庆*　林浩然　陈树越
(常州大学信息科学与工程学院　常州　213164)

摘　要：为了提高快速傅里叶变换的运算效率，减少运算时间，该文研究了 FFT 中倒序序列的计算。研究发现不同长度的倒序序列不相互独立，它们之间有深刻的联系，长度为 N 的倒序序列可以由长度为 $N/2$ 的倒序序列生成。根据不同长度的倒序序列之间的相互关联性，给出了新的倒序序列的计算方法及相应的算法流程。通过计算仿真，验证了算法的正确性。该算法实现简单，而且运算效率高。与传统算法相比，新算法可将计算效率提高 3 个数量级。

关键词：数字信号处理；快速傅里叶变换；序列；倒序

中图分类号：TN911.72　　**文献标识码**：A　　**文章编号**：1009-5896(2018)03-0758-05

DOI: 10.11999/JEIT170595

New Approach for Calculating Inversed Order Sequence in FFT

LIU Daqing　LIN Haoran　CHEN Shuyue
(*School of Information Sciences and Engineering*, *Changzhou University*, *Changzhou* 213164, *China*)

Abstract: In order to improve the efficiency of Fast Fourier Transform (FFT) and reduce the computation time, an algorithm of inversed order sequence in FFT is studied. It is revealed that the inversed order sequences with different length N are not independent but have a deep connection, that is, the inversed order sequence with length N can be produced by the one with length $N/2$ according to a specific schedule. Based on the interconnectedness, a new approach for calculating the inversed order sequence with length N is proposed and the corresponding procedure flow is shown. The algorithm is simulated and the correctness of the algorithm is verified. The algorithm not only can be realized simply, but also has high efficiency. Compared with the traditional method, the new algorithm can improve the computing efficiency by three orders of magnitude.

Key words: Digital signal processing; Fast Fourier Transformation (FFT); Sequence; Inversed order

1　引言

数字傅里叶变换(Digital Fourier Transform, DFT)是信号分析与处理中的一种重要变换[1–10]。但是，直接计算 DFT 的计算量与序列的长度 N 的平方成正比，当 N 较大时，计算量非常大，这时可以利用快速傅里叶变换(Fast Fourier Transform, FFT)。如果不考虑倒序运算量，FFT 需要的计算量大致和 $N\log_2 N$ 成正比，在 $N=1024$ 时，运算效率大致可以提高 204.8 倍[1]。鉴于数值分析、数字信号处理、图像处理、光学和密码学等诸多领域都要用到傅里叶变换，FFT 具有独特的地位。

假设要计算长度为 $N=2^M$ 的序列的 FFT，无论是时域抽取法 FFT(DIT-FFT)还是频域抽取法 FFT(DIF-FFT)都要用到长度为 N 的倒序序列(倒序数组)。一般地，计算倒序序列时需要转码运算，例如将十进制转化为二进制和(或)将二进制转化为十进制。然而，用高级语言实现转码运算需要较多的判断和计算，程序实现较为复杂。因此，寻找一个简单、明晰的算法是有必要的。本文发现，不同长度的倒序序列事实上是互相联系的。利用该发现，我们开发了一种新的倒序算法，它不需要进行转码运算，不但实现简单，还可以极大地提高倒序序列的运算时间。

2　算法分析

为阐述倒序序列的运算，由文献[1,4,5]和计算长为 $N=2,4,8,16$ (或 $M=1,2,3,4$)的倒序序列列表如表 1 所示。其中，对于每个 N(或 M)，第 1 行为倒序的十进制形式，而第 2 行括号中的数字为倒序的二进制形式。

由表 1 可见，倒序序列元素的排列似乎杂论无章。然而，正如文献[1−10]指出的，这种排列事实上是有规律的。我们可将倒序序列的排列规律阐述如下：

考察该序列的第 k 个元素(其中 $k=0,1,\cdots,N-1$ 。若不加说明，序列的起始元素为第 0 元素)，设 k 可以 M 位二进制 $(k_{M-1}k_{M-2}\cdots k_1k_0)_2$ 表示，即

收稿日期：2017-06-21；改回日期：2017-11-28；网络出版：2017-12-27
*通信作者：刘大庆　liudq@cczu.edu.cn
基金项目：国家自然科学基金(51176016)
Foundation Item: The National Natural Science Foundation of China (51176016)

表 1　M=1,2,3,4 时的倒序序列

M	倒序															
1	0								1							
	(0)								(1)							
2	0				2				1				3			
	(00)				(10)				(01)				(11)			
3	0		4		2		6		1		5		3		7	
	(001)		(100)		(010)		(110)		(001)		(101)		(011)		(111)	
4	0	8	4	12	2	10	6	14	1	9	5	13	3	11	7	15
	(0000)	(1000)	(0100)	(1100)	(0010)	(1010)	(0110)	(1110)	(0001)	(1001)	(0101)	(1101)	(0011)	(1011)	(0111)	(1111)

$(k)_{10}=k_{M-1}\times 2^{M-1}+k_{M-2}\times 2^{M-2}+\cdots+k_1\times 2^1+k_0\times 2^0$，则第 k 个元素的二进制形式为 $(k_0k_1\cdots k_{M-2}k_{M-1})_2$，或该数为 $k_0\times 2^{M-1}+k_1\times 2^{M-2}+\cdots+k_{M-2}\times 2^1+k_{M-1}\times 2^0$。

为了明确与简化该规律，我们可定义 M 位倒序操作：

a 的 M 位倒序 $a^{I,M}$：设二进制数 $a\le 2^M-1$ 的形式为 $a=(a_0a_1\cdots a_{M-2}a_{M-1})_2$ (其中若 a 没有 M 位则前面补 0)，则 $a^{I,M}=(a_{M-1}a_{M-2}\cdots a_1a_0)_2$。

我们有如下推论：

(1) $\left((a)^{I,M}\right)^{I,M}$，即一个数的 M 位倒序的 M 位倒序是它自身。

(2)若 a 为偶数，$a=2n$, $n\le 2^{M-1}-1$ 的 $M-1$ 位二进制形式为 $n=(n_0n_1\cdots n_{M-3}n_{M-2})_2$ (不足补 0)，或 $n^{I,M-1}=(n_{M-2}n_{M-3}\cdots n_1n_0)_2$，则有 $a^{I,M}=\left(0n^{I,M-1}\right)_2=(0n_{M-2}n_{M-2}\cdots n_1n_0)_2$。

根据上述 M 位倒序的定义，设倒序序列的第 k 个元素为 a_k，则有 $(a_k)_2=((k)_2)^{I,M}$。

文献[1,4,5]给出了这一算法的详细计算和证明，表 1 给出了计算结果。传统上，基于这一思路，文献[11-18]给出了不同的倒序算法。

然而，不同长度的倒序序列是相互联系的，上述规律没有反映出这种联系。事实上，这种相互联系可由如下定理给出：

定理　长 $N=2^M$ 的倒序序列(以下称为目标序列)可由长 $N_1=2^{M-1}=N/2$ 的倒序序列(以下称为生成序列)按照如下规则生成：生成序列的每个元素 n 不变，同时后面插入一个新的元素 $n+N_1$。

例如 $N=16$ 的目标序列可由 $N_1=8$ 的生成序列生成。当生成序列元素为 3 时，则对应目标序列的两个元素，一个为 3，另一个为 $3+8=11$。即若生成序列的元素为 l，则对应目标序列 的两个元素，l 和 $l+N_1$。显然，该定理指出，不同长度的倒序序列不独立并具有深刻的联系。

以下证明长为 N 的目标序列确实可以按照如上方式由长为 $N_1=N/2$ 的生成序列生成。为简单起见，本证明若不加说明所有元素采用二进制形式。

由文献[1,4,5]，目标序列的第 $0,1,\cdots,N-1$ 个元素的二进制形式分别为：$(0)_2^{I,M}$, $(1)_2^{I,M}$, $(2)_2^{I,M}$, $(3)_2^{I,M}$, $\cdots$, $(N-2)_2^{I,M}$, $(N-1)_2^{I,M}$，或 $(\overbrace{00\cdots0}^{M})_2$, $(1\overbrace{00\cdots0}^{M-1})_2$, $(01\overbrace{00\cdots0}^{M-2})_2$, $(10\overbrace{00\cdots0}^{M-2})_2$, $(11\overbrace{00\cdots0}^{M-2})_2$, $\cdots$, $(\overbrace{11\cdots1}^{M-1}0)$, $(\overbrace{11\cdots1}^{M})$。显然，设目标序列的第 k 个元素为 a_k ($k\ge 0$)，则有推论：

(3) $a_{k+1}=\left(a_k^{I,M}+1^{I,M}\right)^{I,M}$, $a_{k+2}=\left(a^{I,M}+(10)_2^{I,M}\right)^{I,M}$, $\cdots$。以此类推。

(4)若 k 为偶数，则 $a_k^{I,M}=(k)_2$ 也为偶数；若 k 为奇数，则 $a_{k+1}^{I,M}=(k+1)_2$ 也为奇数。即若 k 为偶数，则 a_k 的最高位为 0；若 k 为奇数，则 a_k 的最高位为 1。

(5)由推论(3)和推论(4)，若 k 为偶数，$k=(k_{M-1}k_{M-2}\cdots k_2k_10)_2$，则 $a_k=(0k_1k_2\cdots k_{M-2}k_{M-1})_2$，并且有 $a_{k+1}=(1k_1k_2\cdots k_{M-2}k_{M-1})_2$。即若 k 为偶数，则 a_{k+1} 与 a_k 除最高位不同外，其余位上数字都相同。

(6)由推论(5)，若 k 为偶数，则 $a_{k+1}=a_k+(1\overbrace{00\cdots00}^{M-1})_2$，即第 k+1 个元素必然比第 k 个元素大 $(1\overbrace{00\cdots00}^{M-1})_2=2^{M-1}$。

现在按照顺序将 N 个元素 $a_k(k=0,1,\cdots,N-1)$ 每两个分为一组，共分为 $N_1=N/2$ 组。则第 n 组 $(n=0,1,2,N_1-1)$ 元素为 a_{2n} 和 a_{2n+1}。设 n 的 $M-1$ 位二进制形式为 $(n)_2=(n_{M-2}n_{M-3}\cdots n_1n_0)_2$ (位数不足前面补 0)，则由推论(4)，$2n$ 为偶数，它的 M 位二进制形式为 $(2n)_2=(n_{M-2}n_{M-3}\cdots n_1n_0 0)_2$，有 $a_{2n}=(0n_0n_1\cdots n_{M-3}n_{M-2})_2=\left(0n^{I,M-1}\right)_2$。由此，由推论(2)，按二进制的定义，显然 $a_{2n}=(n)_2^{I,M-1}$。又由倒序序列元素的排列规律[1,4,5]，$(n)_2^{I,M-1}$ 即为长为 $N_1=2^{M-1}$ 的倒序序列的第 n 个元素。同时，$2n$ 为偶数，则由

推论(6)，$a_{2n+1} = a_{2n} + 2^{M-1}$。从而，长 $N = 2^M$ 的目标序列可由长 $N_1 = 2^{M-1} = N/2$ 的生成序列按照如下规则生成：生成序列的每个元素 n 不变，同时后面插入一个新的元素 $n+N_1$。

而 $N_1 = 2^{M-1}$ 的倒序又可以由 $N_2 = 2^{M-2}$ 的倒序生成，以此类推。利用这一思路，我们可以构造如图 1(c)所示的程序框图。作为对比，图 1 中也列出了文献[1]和文献[18]计算倒序的程序框图(我们称之为算法 A 和算法 B)。

从图 1 可以看出，算法 A 和算法 B 程序框图比较复杂，都需要判断语句，而相较于传统算法 A 和算法 B，本文算法框图比较简单，同时没有判断语句，循环次数较少，可以较大地减少计算时间。

为了考察本算法的正确性，我们分别按照算法 A、算法 B 以及算法 C(即本文算法)计算了长度 N=1024, 4096, 16384, 65536 的倒序序列。这 3 种算法结果一致，证明了本文的程序能正确地生成倒序序列。

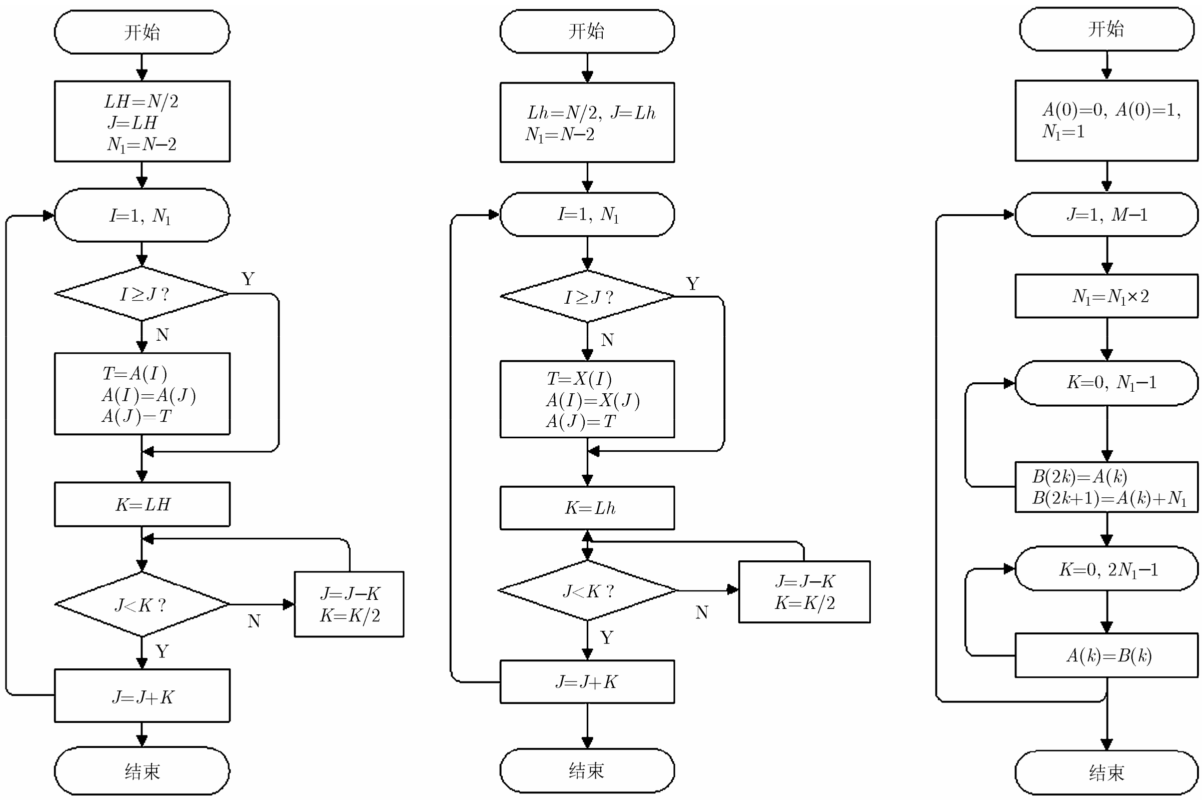
为了考察本文算法的效率，对于每个确定的 N 和算法(A, B, C)，我们在华硕 X550V 上用 VC6 分别计算了 10 次，发现每次的计算时间稍有差别，并不一致。这可能是由于 Windows 操作系统的环境干扰，但是差别不大，为此，对于确定的算法和 N，我们取 10 次计算的平均值作为算法的计算时间，得到表 2。

对于确定的 N，算法 B 比算法 A 耗时较少。算法 B 相较于算法 A 效率大致高了 20%左右，如表 2 第 5 列所示。然而，本文算法的计算时间大幅度减少，从表 2 的第 6，7 列可以看出，其计算时间大致为算法 A 和算法 B 的 1/1000 左右。因此，从计算时间的比较我们认为本文算法的计算效率相较于传统算法(A 和 B)大致提高了 3 个数量级。

3 结论

FFT 在数值分析、数字信号处理、图像处理、光学和密码学等诸多领域有着非常广泛的应用 。而倒序算法是 FFT 的基础，开发简单高效的倒序算法是必要的。通过分析，我们发现并且证明了，不同 N 的倒序排列是彼此联系的，即 $N = 2^M$ 的倒序可以由 $N_1 = 2^{M-1}$ 的倒序生成。基于这个背景，给出了倒序算法的新的计算思路和程序框图。我们通过 C 语言计算实例分析了新算法和传统算法的计算时间，发现本文算法不但实现简单，容易理解，而且运算效率极高，相对于传统的算法，效率提高了 3 个数量级。更为重要的是，本文算法指出不同长度的倒序序列不是相互独立的，它们有深刻的联系。

在做图像处理等运算时，我们有时需要不同 N



(a)算法 A 程序流程图 (b)算法 B 程序流程图 (c)本文程序流程图

图 1 文献[1]，文献[18]和本文的程序流程图

表 2 不同算法的平均耗时比较

N	算法 A 平均时间 t_A (ms)	算法 B 平均时间 t_B (ms)	本文算法平均时间 t_C (ms)	$t_A\,/\,t_B$	$t_A\,/\,t_C$	$t_B\,/\,t_C$
1024	267	157	0.099	1.7	2697	1586
4096	384	310	0.304	1.2	1263	987
16384	424	380	0.358	1.1	1184	1061
65536	613	516	0.501	1.2	1224	1030

时的 FFT，例如临时将 N 提高 2 倍，$N_0 = 2N$ ，这时，利用本文算法可以很容易由 N 的倒序序列生成 N_0 倒序序列，这是本文算法的另一个优越之处。

致谢　感谢西安电子科技大学的高西全教授阅读本文并提出宝贵意见。

参 考 文 献

[1] 高西全, 丁玉美. 数字信号处理[M]. 第 4 版, 西安: 西安电子科技大学出版社, 2016: 92–100.
GAO Xiquan and DING Yumei. Digital Signal Processing [M]. Fourth Edition, Xi’an: Xi’an Electronic and Science University Press, 2016: 92–100.

[2] 吴京. 信号分析与处理[M]. 修订版, 北京: 电子工业出版社, 2014: 129–142.
WU Jing. Signal Analysis and Processing [M]. Revised Edition, Beijing: Electronic Industry Press, 2014: 129–142.

[3] 柳群, BRIGHAM E O. 快速傅立叶变换[M]. 上海: 上海科学技术出版社, 1979.
LIU Qun and BRIGHAM E O. Fast Fourier Transform[M]. Shanghai: Shanghai Science and Technology Press, 1979.

[4] 张克俊, 唐勇波. DIT-FFT 的倒序算法[J]. 兵工自动化, 2005, 24(5): 46–48. doi: 10.3969/j.issn.1006-1576.2005.05.024.
ZHANG Kejun and TANG Yongbo. Reverse algorithm DIT-FFT[J]. *Ordnance Industry Automation*, 2005, 24(5): 46–48. doi: 10.3969/j.issn.1006-1576.2005.05.024.

[5] 朱冰莲, 方敏. 数字信号处理[M]. 第 2 版, 北京: 电子工业出版社, 2014: 99–114.
ZHU Binglian and FANG Min. Digital Signal Processing[M]. Second Editiong, Beijing: Publishing House of Electronics Industry, 2014: 99–114.

[6] 徐美清, 孙晨亮. 快速傅立叶变换 FFT 算法特点分析[J]. 科学与财富, 2016, 1(10): 52–54.
XU Meiqing and SUN Chenliang. Characteristics analysis of fast Fourier transform FFT algorithm[J]. *Journal of Science and Wealth*, 2016, 1(10): 52–54.

[7] 郑伟华. 快速傅立叶变换-算法及应用[D]. [博士论文], 湖南大学, 2015.
ZHENG Weihua. Fast Fourier transform algorithm and its application[D]. [Ph.D. dissertation], Hunan University, 2015.

[8] 郑宇凡. 浅谈 FFT(快速傅立叶变换)算法及其应用[J]. 科技展望, 2015, 25(29): 144–145. doi: 10.3969/j.issn.1672-8289.2015.29.132.
ZHENG Yufan. Discussion on FFT (fast Fourier transform) algorithm and its application[J]. *Science and Technology*, 2015, 25(29): 144–145. doi: 10.3969/j.issn.1672-8289.2015.29.132.

[9] 徐士良. 计算机常用算法[M]. 北京: 清华大学出版社, 1995: 287–291.
XU Shiliang. Computer Algorithm[M]. Beijing: Tsinghua University Press, 1995: 287–291.

[10] 程乾生. 数字信号处理[M]. 北京: 北京大学出版社, 2003: 7–23.
CHENG Qiansheng. Digital Signal Processing[M]. Beijing: Peking University Press, 2003: 7–23.

[11] 汪海兵, 徐淑正, 杨华中. 基于查找表的单基 FFT 原址倒序算法[J]. 清华大学学报, 2008, 48(1): 43–45. doi: 10.3321/j.issn:1000-0054.2008.01.012.
WANG Haibing, XU Shuzheng, and YANG Huazhong. The single-base FFT site reverse algorithm based on look-up table[J]. *Journal of Tsinghua University*, 2008, 48(1): 43–45. doi: 10.3321/j.issn:1000-0054.2008.01.012.

[12] 方志红, 张长耀, 俞根苗. 利用逆序循环实现 FFT 算法中倒序算法的优化[J]. 信号处理, 2004, 20(5): 533–535. doi: 10.3969/j.issn.1003-0530.2004.05.023.
FANG Zhihong, ZHANG Changyao, and YU Genmiao. To realize the optimization of reverse FFT algorithm in using reverse circulation[J]. *Journal of Signal Processing*, 2004, 20(5): 533–535. doi: 10.3969/j.issn.1003-0530.2004.05.023.

[13] 张学智, 蔡辉. 快速实现 FFT 的逆序方法[J]. 探测与控制学报, 2001, 23(2): 62–65. doi: 10.3969/j.issn.1008-1194.2001.02.023.
ZHANG Xuezhi and CAI Hui. Fast implementation of FFT reverse order method[J]. *Journal of Detection and Control*, 2001, 23(2): 62–65. doi: 10.3969/j.issn.1008-1194.2001.02.023.

[14] 刘仲, 陈海燕, 向宏卫. 使用融合乘加加速快速傅立叶变换计算的向量化方法[J]. 国防科技大学学报, 2015, 37(2): 72–78. doi: 10.11887/j.cn.201502015.
LIU Zhong, CHEN Haiyan, and XIANG Hongwei. By using fusion and accelerate the vectorization method of fast Fourier transform[J]. *Journal of National University of Defense Technology*, 2015, 37(2): 72–78. doi: 10.11887/j.cn.201502015.

[15] 薛会, 张丽, 刘以农. 非标准快速傅里叶变换算法综述[J]. CT

理论与应用研究, 2010, 19(3): 33–46. doi: 10.3969/j.issn.1003-2215.2014.11.015.

XUE Hui, ZHANG Li, and LIU Yinong. Overview of non standard fast Fourier transform algorithm[J]. *CT Theory and Applications*, 2010, 19(3): 33–46. doi: 10.3969/j.issn.1003-2215.2014.11.015.

[16] 张大炜. 一种新的级联 FFT 算法[J]. 舰船科学技术, 2016, 38(5): 60–63. doi: 10.3404/j.issn.1672-7619.2016.05.013.

ZHANG Dawei. A new cascade FFT algorithm[J]. *Ship Science and Technology*, 2016, 38(5): 60–63. doi: 10.3404/j.issn.1672-7619.2016.05.013.

[17] 李龙军, 王布宏, 夏春和. 基于迭代 FFT 算法的平面阵列交错稀疏布阵方法[J]. 电子与信息学报, 2016, 38(4): 970–977. doi: 10.11999/JEIT150749.

LI Longjun, WANG Buhong, and XIA Chunhe. Interleaved thinned linear arrays based on modified iterative FFT technigue[J]. *Journal of Electronics and Information Technology*, 2016, 38(4): 970–977. doi: 10.11999/JEIT150749.

[18] 陈慧, 周继惠, 郭春亮, 等. 基于 VB 的 FFT 算法的设计和实现[J]. 华东交通大学学报, 2003, 20(1): 94–96. doi: 10.3969/j.issn.1005-0523.2003.01.027.

CHEN Hui, ZHOU Jihui, GUO Chunliang, *et al.* Design and implementation of FFT algorithm based on VB[J]. *Journal of East China Jiaotong University*, 2003, 20(1): 94–96. doi: 10.3969/j.issn.1005-0523.2003.01.027.

刘大庆： 男，1971 年生，副教授，研究方向为电子科学与工程.

林浩然： 男，1995 年生，本科生，研究方向为电子科学与工程.

陈树越： 男，1963 年生，教授，研究方向为图像处理与分析.