

DMT 中一种格型结构算法的 DSP 实现分析

赵慧民, 张光昭, 余顺争

(中山大学电子与通信工程系, 广东广州 510275)

摘要: DMT(离散多音频)调制/解调是 ADSL(不对称数字用户线)应用的标准传输技术. 尽管 DMT 能达到比其他调制技术更高的速率, 但其计算的复杂性也非常高, 尤其是调制/解调核心程序 FFT/IFFT 的运算. 因此, 本文探索了一种特别适合 DSP 处理的时间递归(time recursive)格型 FFT/IFFT 算法结构. 该结构与其直接运算($N = 256$)相比仅需原来 11% 的乘法运算量和 9% 的加法运算量, 从而使 ADSL 实时性、应用性大大提高.

关键词: DMT; FFT/IFFT; 格型结构算法

中图分类号: TN911.72 **文献标识码:** A **文章编号:** 0372-2112(2001)10-1342-04

The DSP Implementation and Analysis of the Lattice Architecture Algorithm In DMT

ZHAO Huimin, ZHANG Guangzhao, YU Shunzheng

(Dept. of Electronics and Communication Engineering, Zhongshan University, Guangzhou, Guangdong 510275, China)

Abstract: The discrete multitone (DMT) modulation/demodulation is the standard transmission technique in the application of asymmetric digital subscriber lines (ADSL). Although the DMT can achieve higher data rate compared with other modulation/demodulation schemes, its computational complexity is too high, especially to compute the modulation/demodulation kernel FFT/IFFT. Hence, a lattice algorithm based on the time recursive approach is presented, which is very suitable to run in programmable DSP. The algorithm requires only 11% number of multipliers and 9% number of adders compared with the direct implementation ($N = 256$), so it improves greatly ADSL reality and applicability.

Key words: DMT (discrete multitone); FFT/IFFT; lattice architecture algorithm

1 引言

Internet 接入的发展特别需要一种高速数据传输技术. 为了克服传统电话线上传输的“瓶颈”问题, 人们提出了几种新的调制/解调方法^[1]. 其中, 由于 DMT 结合了先进的 DSP(数字信号处理)技术, 如动态比特分配、多维音频编码等等, 所以它能实现最高的传输速率. 现在, ADSL 标准组织已正式把 DMT 作为物理层的传输技术标准. 但是, DMT 的一个主要缺点是运算量偏大, 特别是 N 较大时, FFT/IFFT 的计算量大. 为此, 本文提出了一种时间递归传输编码的格型 FFT/IFFT 算法结构, 以提高运算效率, 改善了 ADSL 实现的实时性能. 文中首先把 FFT/IFFT 分解成一个 DCT(离散余弦变换)和一个 DST(离散正弦变换)传输函数的和式, 通过利用傅立叶变换的对称性以及 DCT 和 DST 的快速算法, 得出替代复数域 FFT/IFFT 的一种新方案. 该方案仅需在实数域操作; 然后, 本文利用了时间递归格型结构实现了这种算法. 这个结构计算了时域输入序列传输的数据且仅需要 $O(N)$ 量级的硬件复杂性, 并且当 $N = 256$ 时, 它只需要直接蝶形 FFT/IFFT 运算中的 11% 的

乘法和 9% 的加法运算, 而且它们实现结果保持一致.

2 IFFT 的模型

2.1 IFFT 的推导

FFT/IFFT 在 DMT 系统中的结构如图 1 所示. 这里, 按 ADSL 标准取 $N = 256$. 在传输一边, 为确保 IFFT 仅产生实数输出, 其输入的约束条件是^[2]:

$$x(n) = x^*(2N - n), n = 0, 1, 2, \dots, N - 1 \quad (1)$$

这里, $x(n) = x_r(n) + jx_i(n)$ 被编码成满足 $x(0) = x(N) = 0$ 的复数值. 长度为 $2N$ 数据序列的 IFFT 为:

$$X(K) = \sum_{n=0}^{2N-1} x(n) \cdot W_{2N}^{nk}, k = 0, 1, 2, \dots, 2N - 1 \quad (2)$$

这里, $W_{2N}^{nk} = \exp(-j \frac{2\pi nk}{2N})$. 把 n 分为前后两部分并利用条件 $x(0) = x(N) = 0$, 式(2)成为:

$$X(k) = \sum_{n=1}^{N-1} x(n) \cdot W_{2N}^{nk} + \sum_{n=N+1}^{2N-1} x(n) \cdot W_{2N}^{nk} \quad (3)$$

接着, 利用输入数据的共轭对称性简化式(2)成为:

$$\begin{aligned}
 X(k) &= 2j \sum_{n=1}^{N-1} x_r(n) \cdot \cos \frac{2\pi nk}{2N} \\
 &\quad + \sum_{n=1}^{N-1} x_i(n) \sin \frac{2\pi nk}{2N} \\
 &= 2j [MDCT(k) + MDST(k)], \\
 k &= 0, 1, \dots, 2N-1 \quad (4)
 \end{aligned}$$

从式(4)也可以看到, FFT 的运算能分解成为两个部分的实数域计算. 第一部分是具有 $x_r(n)$ 输入的类似 DCT 操作, 第二部分是具有 $x_i(n)$ 输入的类似 DST 操作. 这里, 把第一部分命名为修正 DCT(MDCT), 第二部分叫修正 DST(MDST). 注意, MDCT(k) 和 MDST(k) 仅涉及实数值计算, 并且对 $k = 1, \dots, N-1$ 有:

$$\begin{aligned}
 MDCT(k) &= MDCT(2N-k) \\
 MDST(k) &= -MDST(2N-k)
 \end{aligned} \quad (5)$$

因此, 主要是对于 $k = 1, 2, \dots, N-1$, 计算 MDCT(k) 和 MDST(k). 然后, 再对 $k = N, N+1, \dots, 2N-1$ 扩展其运算范围. 这种简化运算大约可节省原有计算的 50% 硬件开销.

2.2 IFFT 的格型结构

IFFT 的格型结构模型可通过时间递归的方法来实现^[3], 在此主要说明 MDCT 和 MDST 双级有效的实现问题. 首先, 考虑 MDCT 的格型结构. 根据文献[3]推导, 并定义 MDCT 连续输入数据流从 $x_r(t)$ 开始至 $x_r(t+N-1)$ 结束, 则有:

$$X_c(k, t) = \sum_{n=t}^{t+N-1} x_r(n) \cdot \cos \frac{2\pi(n-t)k}{2N} \quad (6)$$

这里, $k = 0, 1, \dots, 2N-1$. 新到来数据项 $x_r(t+N)$ 之后, MDCT 修正为:

$$X_c(k, t+1) = \sum_{n=t+1}^{t+N} x_r(n) \cos \frac{2\pi(n-t-1)k}{2N} \quad (7)$$

重新把式(7)写为

$$X_c(k, t+1) = \overline{X}_c(k, t+1) \cos \frac{\pi \cdot k}{N} + \overline{X}_s(k, t+1) \sin \frac{\pi \cdot k}{N} \quad (8)$$

这里, 分别有

$$\overline{X}_c(k, t+1) \triangleq \sum_{n=t+1}^{t+N} x_r(n) \cos \frac{2\pi(n-t)k}{2N} \quad (9)$$

$$\overline{X}_s(k, t+1) \triangleq \sum_{n=t+1}^{t+N} x_r(n) \sin \frac{2\pi(n-t)k}{2N} \quad (10)$$

另一方面, 通过删除上一个数据项 $x_r(t)$ 并修正现在新的数据项 $x_r(t+N)$ 可推导出关系式:

$$\overline{X}_c(k, t+1) = X_c(k, t) - x_r(t) + (-1)^k x_r(t+N) \quad (11)$$

对于块处理, 其初始状态为零, 可把上式进一步简化为:

$$\overline{X}_c(k, t+1) = X_c(k, t) + (-1)^k x_r(t+N) \quad (12)$$

类似地, 可推导出 $\overline{X}_s(k, t+1)$ 与 $X_s(k, t)$ 的关系为:

$$\overline{X}_s(k, t+1) = X_s(k, t) \quad (13)$$

其中 $X_s(k, t)$ 定义为:

$$X_s(k, t) = \sum_{n=t}^{t+N-1} x_i(n) \sin \frac{2\pi(n-t)k}{2N}, \quad k = 0, 1, \dots, 2N-1 \quad (14)$$

按照式(7)~(10)中的推导, 可以发现:

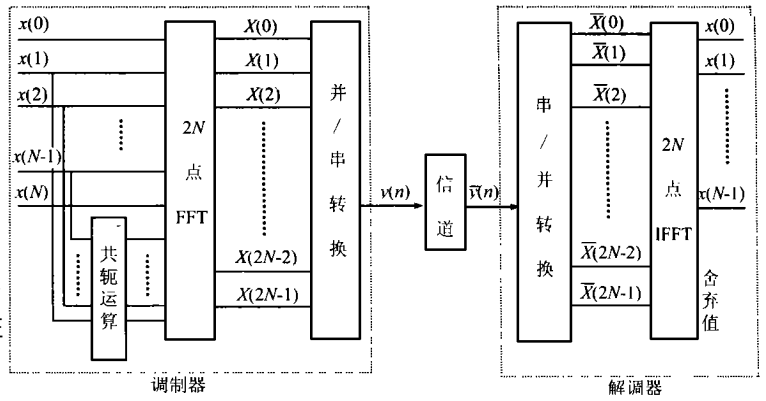


图 1 FFT/IFFT 在 DMT 系统中的方框图

$$X_s(k, t+1) = \overline{X}_s(k, t+1) \cos \frac{\pi \cdot k}{N} - \overline{X}_c(k, t+1) \sin \frac{\pi \cdot k}{N} \quad (15)$$

这里, 分别有

$$\overline{X}_c(k, t+1) \triangleq \sum_{n=t+1}^{t+N} x_i(n) \cos \frac{2\pi(n-t)k}{2N} \quad (16)$$

$$\overline{X}_s(k, t+1) \triangleq \sum_{n=t+1}^{t+N} x_i(n) \sin \frac{2\pi(n-t)k}{2N} \quad (17)$$

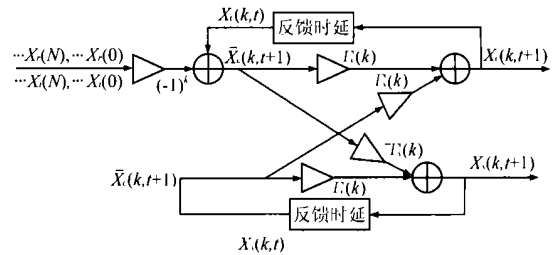


图 2 IFFT 的双级 MDCT 和 MDST 格型 (M_k, k = 1, 2, ..., N-1) 运算结构

注意, 式(16)和(17)除其输入是 $x_i(n)$ 以外其余都类似于式(9)和(10). 因此, (12)、(13)的修正式除把输入变为 $x_i(n)$ 后也用于式(16)、(17)的修正式计算. 通过以上数学变换, 导出同时产生 MDCT 和 MDST 输出的格型模型如图 2 所示. 其中, $\Gamma_c(k) \triangleq \cos \frac{\pi \cdot k}{N}$, $\Gamma_s(k) \triangleq \sin \frac{\pi \cdot k}{N}$. 对于 $k = 0$ 和 $k = N$ 两种特殊情况, IFFT 的输出分别成为: $X(0) = \sum_{n=0}^{N-1} x_r(n)$ 和 $X(N)$

$= \sum_{n=0}^{N-1} (-1)^n x_r(n)$. 这两种特殊情况通过累加器可以实现, 在模型中用 SC1 标识. 全部的 IFFT 实现结构如图 3 所示. 这里, $\downarrow N$ 代表了在时间上 $x_r(N)$ 和 $x_i(N)$ 完成模块阵列运算后对 MDCT/MDST 结果的亚抽样. 阵列模型由 $N-1$ 个格型结构和一个特殊情况结构 SC1 组成. 扩展电路由 N 次乘法和 $N-1$ 次加法组成. 当 $x_r(n)/x_i(n)$ 作为输入数据时, 乘法器分别分离 MDCT/MDST 值输出. 然后, 根据等式(5), 扩展 MDCT/MDST 的 $2N$ 个值输出. 寄存器用于储存运算的中间结果. 最后的 MDCT/MDST $2N$ 个值作为 IFFT 的数据, 左移 1 比特进行乘 2 运算操作.

从图 3 可见, 这个结构是模块化、标准化, 不需全局数据通信, 完全可由单片 DSP 实现. 若选用 TI 公司 TMS320C6201

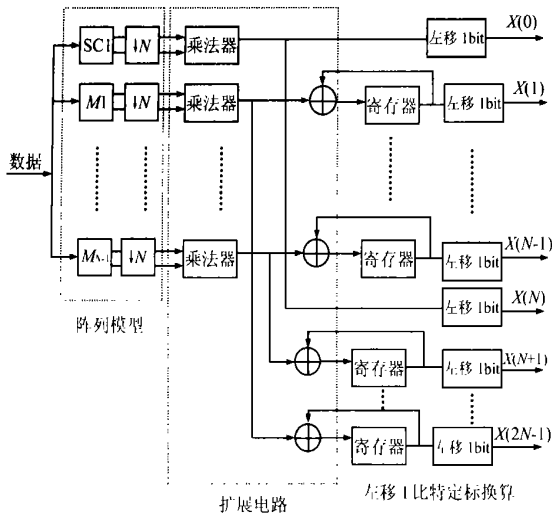


图 3 基于格型模块的全部 IFFT 运算结构

实现的话, 可为此开辟独立缓存区, 当 ADSL 系统输入比特流到来时, 直接调用其结果作为 N 维调制输出, 发送至均衡信道传输即可, 实时性能可以大大提高。

3 FFT 的模型及其格型结构

在图 1 的接收边, 进行 FFT 以解调接收的信号。类似 IFFT 的分析过程, 分解 FFT 为(推导略):

$$\begin{aligned} \bar{x}(n) &= \frac{1}{2N} \left[\sum_{k=0}^{2N-1} \bar{X}(k) \cos \frac{2\pi \cdot nk}{2N} + j \sum_{k=0}^{2N-1} \bar{X}(k) \sin \frac{2\pi \cdot nk}{2N} \right] \\ &= MDCT(n) + jMDSF(n), \quad n = 0, 1, \dots, 2N-1 \end{aligned} \quad (18)$$

它和 IFFT 的区别是其实数值需要通过 $\frac{1}{2N}$ 换算。另外, 在 DMT 系统中, 后续的 N 点 FFT 输出是前续 N 点 FFT 的共轭对称, 所以可忽略 $\bar{x}(n), n = N, N+1, \dots, 2N-1$ 的计算, 也节约了 50% 的操作时间。

在文献[3]中已说明了关于 FFT 的实数域输入数据的格型运算结构。但在 DMT 系统中, 其输入是 $2N$ 点数据而我们只需要 N 点的 FFT 输出。按照式(6)~(10)的结果, 时间递归 MDCT 公式成为:

$$\tilde{x}_r(n, t+1) = \tilde{x}_r(n, t+1) \cos \frac{\pi \cdot n}{N} + \tilde{x}_i(n, t+1) \sin \frac{\pi \cdot n}{N} \quad (19)$$

类似地, 也可以推导出 MDST 的计算公式, 这里从略。全部时间递归 FFT 的格型结构除其输入为 $\bar{X}(k)$ 及乘法因子不同以外, 其它与图 2 相似, 均由 4 次乘法和 3 次累加组成。对于 $n=0$ 的特殊情况, FFT 可被简化为 $\tilde{x}(0) = \sum_{k=0}^{2N-1} \bar{X}(k)$, 可通过一次累加操作来实现(用 SC2 标识)。全部的 FFT 结构如图 4, 其操作过程如下: (1) 数据 $\bar{X}(k), k = 0, 1, \dots, 2N-1$, 送入模块阵列计算 MDCT 和 MDST 的输出; (2) 通过右移 $\log_2(2N)$ 比特运算完成 $1/2N$ 的换算操作; (3) 按照式(18), 把实部和虚部合成得到 FFT 的复数域运算结果。

4 复杂性分析

在直接 FFT/IFFT 的运算中, $2N$ 点需要 $\log_2(2N)$ 级蝶形结构, 每一级由 N 次乘法和 $2N$ 次累加组成。因其输入是复数域值, 一次复数乘法需要 4 次实数相乘和 2 次实数相加, 再加上 1 次复加需 2 次实数相加, 所以总共需要 $4N \log_2(2N)$ 次实乘和 $6N \log_2(2N)$ 次实加。

表 1 $2N$ 点 FFT/IFFT 的乘法比较

N	FFT			IFFT		
	蝶形 O_1	格型 O_2	CR	蝶形 O_1	格型 O_2	CR
256	9216	1020	0.111	9216	1020	0.111
512	20480	2044	0.100	20480	2044	0.100
1024	45056	4092	0.091	45056	4092	0.091
2048	98304	8188	0.083	98304	8188	0.083
4096	212992	16380	0.077	212992	16380	0.077
8192	458752	32764	0.071	458752	32764	0.071

表 2 $2N$ 点 FFT/IFFT 的加法比较

N	FFT			IFFT		
	蝶形 O_1	格型 O_2	CR	蝶形 O_1	格型 O_2	CR
256	13824	1277	0.092	13824	766	0.055
512	30720	2557	0.083	30720	1534	0.050
1024	67584	5117	0.076	67584	3070	0.045
2048	147456	10237	0.069	147456	6142	0.042
4096	319488	20477	0.064	319488	12286	0.038
8192	688128	40957	0.060	688128	24574	0.036

而在本文中, 图 3 的 IFFT 仅需实数运算。SC1 进行 2 次相加, 每一格型完成 4 次相乘和 2 次相加。全部的运算再加上额外的 $2(N-1)$ 次加法和 N 次乘法, 需要 $4(N-1)$ 实乘和 $5(N-1) + 2$ 次实加。而在图 4 中, 共需 $4(N-1)$ 实乘和 $3(N-1) + 1$ 次实加运算。 $2N$ 点 FFT/IFFT 运算的复杂性比较结果上表 1~2 所示。其中, 复杂性比较系数 CR 定义为: $CR \triangleq \frac{O_2}{O_1}$ 。这里, O_1 和 O_2 分别是蝶形和格型结构的乘法(或表 2 中的加法)运算次数。

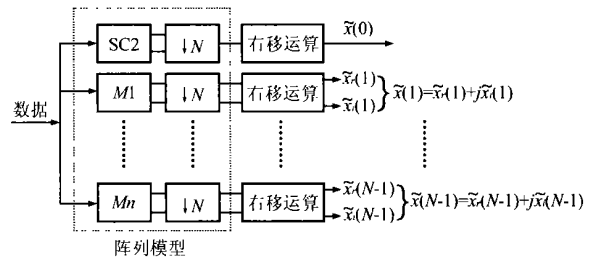


图 4 基于格型模块的全部 FFT 运算结构

由表可见, 在 $N = 256$ 时, 乘法 $CR = 11\%$, 加法 $CR = 9\%$ 。而且, N 越大, CR 越小。这说明, 用格型结构进行 FFT/IFFT 运算比其直接运算节约了大量的 DSP 硬件资源, 尤其对于 N 较大的计算。在选用 TMS320C6201 进行处理时, 由于它的指令周期为 5ns, 对常规 1024 点的 FFT/IFFT 需要 70μs, 据此用本文提出的格型结构 FFT/IFFT 算法与其常规算法相比, 其执行时间效率如图 5 所示。

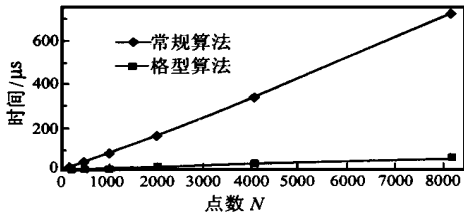


图 5 格型 FFT/IFFT 与其常规算法执行时间比较

由图可见, 格型结构算法大大提高了运算的执行时间, 这使得 DSP 在 ADSL 的应用中有效克服了由于 DMT 的处理而导致的较大时延问题^[4], 为用户的 Internet 高速宽带传输提供了保障。

5 结论

本文提出一种在 DMT 系统中易于 DSP 处理的 FFT/IFFT 实现算法, 该算法重新利用并分解了 FFT/IFFT 运算, 避免了复数域的操作。同时, 本文使用时间递归的格型结构实现了此算法, 并保持了与常规算法结果的一致性, 但为 DSP 节约了大量的硬件资源, 从而大大提高了 ADSL 系统的实时性能。

参考文献:

- [1] Zdzislaw papir. Competing for throughput in the local loop [J]. IEEE Communications Magazine, May 1999: 61- 66.
- [2] J S Chow, J C Tu, J M Cioffi. A discrete multion transceiver system for HDSL applications [J]. IEEE J. select. Areas Commun, Aug, 1991, 9: 895- 908.

- [3] K J Ray Liu, C T Chiu, et al. Optimal unified architectures for the real time computation of time recursive discrete sinusoidal transforms [J]. IEEE Trans. Circuits and Systems for Video Technology, April 1994, 4 (2): 168- 180.
- [4] I K Czajkowski. High speed copper access: a tutorial overview [J]. Electronics & Communication Engineering Journal, June 1999: 125 - 148.

作者简介:



赵慧民 男, 1966 年出生, 1991 年和 1997 年在西北工业大学分别获得学士学位与硕士学位。现在广州中山大学电子与通信工程系攻读博士学位, 任中山大学 TF DSP 实验室主任。主要研究方向为 ATM over ADSL 的 DSP 应用技术, 已发表学术论文十篇。



张光昭 男, 63 岁, 1959 年毕业于中山大学, 广州中山大学电子与通信工程系教授, 博士生导师, 中国电子学会会士, 广东省电子学会副理事长。现主要研究方向为多媒体通信、计算机网络和无线接入网。