DOI:10.19652/j.cnki.femt.2406303

基于 Cooley-Tukey-WFTA 算法的 DFT-S-OFDM 系统的优化研究

秦 昊[1,2] 尤文斌[2]
(1.山西银河电子设备厂 太原 030006;2.中北大学电气与控制工程学院 太原 030051)

摘 要:针对高速宽带无线通信系统的需求,现有的离散傅里叶变换离散傅里叶变换(discrete Fourier transform,DFT)扩展正交频分复用实现方法在资源利用和计算速度方面存在局限性。为了解决这些问题,提出了一种基于 Cooley-Tukey 算法和 Winograd-Fourier 变换(WFTA)算法相结合的优化方案。通过算数优化减少了乘法器的使用,实现了 1 024 点快速傅里叶变换(fast Fourier transform,FFT)的基-2 时间抽取(decimation-in-time,DIT)蝶形算法结构优化。同时引入了 WFTA 算法,利用移位化简操作,完成了 3 点 DFT 的资源分配。设计并优化了现场可编程门阵列(field-programmable gate array, FPGA)资源,成功实现了 3 072 点 FFT 在 FPGA 上的运行。FPGA 仿真和平台测试的结果表明,相较于 Xilinx LTE FFT IP 核处理算法,该优化算法在乘法器资源消耗方面减少了 24.96%,处理速度提升了 14.07%。此外,该算法大幅度降低了 FFT 的计算复杂度,显著提升了离散傅里叶变换扩频正交频分复用(DFT-S-OFDM)系统中 DFT 的整体性能。综上所述,所提出优化算法为 DFT-S-OFDM 技术在实际通信系统中的高效实现提供了新的解决方案。

关键词:Cooley-Tukey 算法;WFTA 算法;资源优化;FPGA
中图分类号: TN911.72 **文献标识码**:A **国家标准学科分类代码**: 510.4030

Research on optimization of DFT-S-OFDM system based on the Cooley-Tukey-WFTA algorithm

Qin Hao[1,2] You Wenbin[2]
(1. Shanxi Yinhe Electronic Equipment Factory, Taiyuan 030006, China;
2. College of Electrical and Control Engineering, North University of China, Taiyuan 030051, China)

Abstract: Aiming at the requirements of high-speed broadband wireless communication systems, the existing discrete Fourier transform(DFT) extended orthogonal frequency division multiplexing implementation methods have limitations in resource utilization and calculation speed. To solve these issues, this paper proposes an optimization approach that combines the Cooley-Tukey algorithm with the Winograd Fourier transform algorithm (WFTA). Arithmetic optimizations were applied to reduce the use of multipliers, enabling the optimization of the radix-2 DIT butterfly structure for a 1 024-point fast Fourier transorm(FFT). Additionally, the WFTA algorithm was introduced, using shift-based simplifications to optimize resource allocation for 3-point DFTs. FPGA resources were designed and optimized, resulting in the successful implementation of a 3 072-point FFT on an FPGA. The results of FPGA simulation and platform test show that, compared to the Xilinx LTE FFT IP core, the proposed optimization reduced multiplier resource consumption by 24.96% and increased processing speed by 14.07%. Moreover, this algorithm significantly reduced the computational complexity of the FFT, greatly enhancing the overall performance of DFT in DFT-S-OFDM system. In summary, the optimization proposed in this paper provides an efficient solution for implementing DFT-S-OFDM technology in practical communication systems.

Keywords: Cooley-Tukey algorithm; WFTA algorithm; resource optimization; FPGA

收稿日期:2024-09-29

0 引 言

在现代通信系统中，正交频分复用(orthogonal frequency division multiplexing，OFDM)技术因其高效的数据传输速率和强抗多径干扰能力，已成为无线通信标准的核心组成部分。然而，OFDM信号的高峰均功率比(peak to average power ratio，PAPR)在一定程度上限制了功率放大器的效率，进而影响了信号传输质量[1]。为了解决这一问题，离散傅里叶变换扩频正交频分复用(discrete fourier transform spread orthogonal frequency division multiplexing，DFT-S-OFDM)技术通过在逆离散傅里叶变换之前对信号进行预处理，有效降低了PAPR并改善了传输性能[2]。DFT-S-OFDM作为一种先进的调制技术，利用离散傅里叶变换(discrete Fourier transform，DFT)对调制符号进行优化处理，增强了子载波之间的关联性。通过在适当位置插入零值并对经过DFT处理的符号进行离散傅里叶逆变换(IDFT)处理，符号被精确地映射到各自的子载波上[3]。

快速傅里叶变换(fast Fourier transform，FFT)是DFT及其逆变换的高效计算方法，广泛应用于科学、数学和工程等领域[4]。在高带宽应用场景下，随着DFT-S-OFDM系统中使用的FFT点数增大，硬件资源消耗显著增加，并导致处理延迟增大，这在实时通信系统中成为一个重要问题[5]。Cooley-Tukey FFT算法是应用最为广泛的FFT算法[6]，但对于非2的幂次长度的输入效率较低。为减少DFT运算中的乘法次数，降低计算复杂度，并提高DFT计算的实时性，提出了Winograd-Fourier变换算法(Winograd Fourier transform algorithm，WFTA)[7]，该算法通过分解旋转因子，显著减少了乘法运算量，更适合硬件实现，但其仅支持部分固定点数的计算，不适用于大规模数据处理。许鹏飞等[8]借助XILINX核完成1 024点FFT运算后，进一步开展3点DFT运算，从而实现3 072点FFT。Liu等[9]则运用Good-Thomas算法进行排序，接着依次进行3点DFT运算与1 024点FFT运算，直至完成3 072点运算。洪钦智等[10]设计了支持多数据块混合处理的块浮点处理架构，可实现64～4 096点FFT处理。随着通信系统带宽的增加，DFT-S-OFDM系统中使用的FFT点数也在不断增加，实现方法通常需要大量的硬件资源，尤其是在处理高点数的FFT时。这些方法在现场可编程门阵列(field-programmable gate array，FPGA)实现中会占用大量的乘法器和加法器，从而增加了硬件成本和功耗。目前DFT-S-OFDM系统的优化方案大多侧重于算法层面的改进，但在硬件资源利用率和计算延迟方面的优化仍然不足，难以满足实时通信系统的高效处理需求。这些问题的存在促使了对更高效、更灵活的DFT计算方法的进一步研究。

针对DFT-S-OFDM技术的需求，本文结合了Cooley-Tukey FFT算法和WFTA算法，提出了一种新的Cooley-Tukey-WFTA混合算法。该算法对1 024点FFT中的基-2蝶形结构进行了资源优化，并利用WFTA算法优化了3点DFT的资源分配，成功优化了3 072点FFT的计算。通过算数运算优化了FPGA程序的乘法器结构，并成功将该算法在FPGA硬件平台上应用。

1 相关算法

1.1 Cooley-Tukey 混合基 FFT 算法

Cooley-Tukey混合基FFT算法能够灵活适应不同长度的变换运算[11]，显著提高了FFT在工程应用中的通用性。不同因式分解组合对算法的计算复杂度具有直接影响。当点数N为复合数(即可分解为多个因子的乘积)时，该算法可以应用于其变换运算[12]。对于点N，其DFT定义为：

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi j}{N}kn} = \sum_{n=0}^{N-1} x_n W_N^{kn} \tag{1}$$

式中：$k=0,1,2,\cdots,N-1$，为N个DFT的系数；n为$0,1,2,\cdots,N-1$；$W_N^{kn}=e^{-2\pi j/N}$是$e^{-2\pi j}$的N次根；系数W_N^{kn}是DFT的旋转因子。

根据式(1)可知，其加法运算复杂度为$O(N(N-1))$，乘法运算复杂度为$O(N^2)$[13]。由于计算量较大，因此需提高其效率，采用混合基FFT算法实现N点的FFT计算。将数N分解为N_1和N_2形式，此时的$N=N_1N_2$，令式(1)中$k=N_2k_1+k_2$，$k_1\in[0,N_1-1]$，$k_2\in[0,N_2-1]$以及$n=N_1n_2+n_1$，$n_1\in[0,N_1-1]$，$n_2\in[0,N_2-1]$代入式(1)，得到：

$$X_{N_2k_1+k_2} = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x_{N_1n_2+n_1} e^{-\frac{2\pi j}{N_1N_2}(N_1n_2+n_1)(N_2k_1+k_2)} =$$
$$\sum_{n_1=0}^{N_1-1}\left(e^{-\frac{2\pi j}{N}n_1k_2}\right)\left(\sum_{n_2=0}^{N_2-1} x_{N_1n_2+n_1} e^{-\frac{2\pi j}{N_2}n_2k_2}\right)e^{-\frac{2\pi j}{N_1}n_1k_1} \tag{2}$$

式(2)可分为两个DFT变换，分别为N_1个内部长度为N_2的变换和N_2个外部长度为N_1的变换；其中第2个括号实际为N_1个长度为N_2的DFT变换，每组为$x_{N_1n_2+n_1}$，$n_1\in[0,N_1-1]$，同时乘以旋转因子，再做N_2个长度为N_1的DFT变换[14]。式(1)中以求和形式表示DFT，也可用向量矩阵相乘的形式表达：

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_k \\ \vdots \\ X_{(N-1)} \end{bmatrix} = \begin{bmatrix} (W_N^{kn}) \\ (k,n=0,1,\cdots,N-1) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(n) \\ \vdots \\ x(N-1) \end{bmatrix} \tag{3}$$

其中，DFT矩阵如下：

$$\begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^1 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_N^0 & W_N^2 & W_N^{2k} & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_N^0 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \tag{4}$$

式(1)中 $W_N^{kn}=\mathrm{e}^{-2\pi jkn/N}$,结合欧拉公式 $\mathrm{e}^{\pm j\theta}=\cos\theta\pm\sin\theta$ 可知旋转因子有周期性,即 $W_N^{k+N}=W_N^k$;对称性,$W_N^{k+N/2}=-W_N^k$;若 m 是 N 的约数,$W_N^{mkn}=W_{N/m}^{kn}$;$W_N^0=1$ 和 $W_N^{N/2}=-1$。

为降低算法的复杂度,可以利用旋转因子性质,可将复杂度降低到 $O(N\log N)$,其中,Cooley-Tukey FFT 算法是最为成熟且应用最广泛的算法。其核心思想是分治法,将 N(N 为 2 的幂次)点分解为两个序列:一个为偶数点序列,另一个为奇数点序列。通过分别计算这两个子序列的 FFT,最终得到原始点序列的 FFT。每一级运算都能显著降低复杂度,直到最后一级分解为两个点为止,过程如下:

$$C(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn}=A(k)+W_N^kB(k) \tag{5}$$

其中,$k=0,1,\cdots,N/2-1$,$A(k)$和 $B(k)$分别表示 N 点 DFT 的偶数序列和奇数序列。利用周期性,$C(k)$的周期为 N,即 $C(k)=C(k+n)$;而 $A(k)$和 $B(k)$的周期都为 $N/2$;结合 $W_N^{N/2}=-1$ 和 $W_N^{k+N/2}=-W_N^k$,可将式(4)、(5)简化为最优解,得出:

$$D(k+N/2)=A(k)-W_N^kB(k) \tag{6}$$

式(6)可以用蝶形图(图 1)来表示[15]。

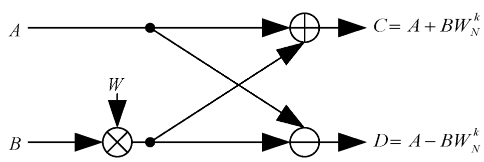


图 1 Cooley-Tukey 基-2 DFT 蝶形图

Fig. 1 Cooley-Tukey base-2 DFT dish chart

1.2 小数组 WFTA 算法

小数组 WFTA 算法是整个 WFTA-FFT 算法的核心基础[16]。小数组 WFTA 适用于信号长度 $N=2,3,4,5,6,7,8,9,16$ 的 DFT,Winograd 提出的这一独特算法能够有效简化这些特定长度的 DFT 运算。由此,可将式(4)可以写成:

$$\boldsymbol{W}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & \cdots & W \end{bmatrix} \tag{7}$$

小数组 WFTA-FFT 算法的主要目的是将矩阵式(7)分解为简化的标准形式 $\boldsymbol{W}=\boldsymbol{CBA}$。

其中,矩阵 $\boldsymbol{C}$ 和 $\boldsymbol{A}$ 的元素仅包含 0 和 ±1,矩阵 $\boldsymbol{B}$ 是一个对角线为非零元素的对角矩阵。这样一来,只有在数据与矩阵 $\boldsymbol{B}$ 进行运算时才涉及乘法运算,极大地减少了乘法运算量。给出的式(7)的分解过程参考文献[17]。

2 Cooley-Tukey-WFTA 混合优化算法

2.1 1 024 点 FFT 的基-2 DIT 蝶形优化算法

本文采用的 1 024 点 FFT 基于 Cooley-Tukey 基-2 DIT 结构[18]。输入端表示 FFT 的输入数据序号,输出端表示 OFDM 频域子载波序号。相邻两级之间的数字代表旋转因子的指数参数。整个 1 024 点 FFT 结构包含 10 级蝶形运算[19]。为分析旋转因子与中间级输出字长对 FFT 性能的影响,每一级的输出信号和旋转因子均可独立调节。

从图 1 可以看出,基-2 DIT 蝶形算法涉及复数加法、复数减法和复数乘法,如图 2 所示。

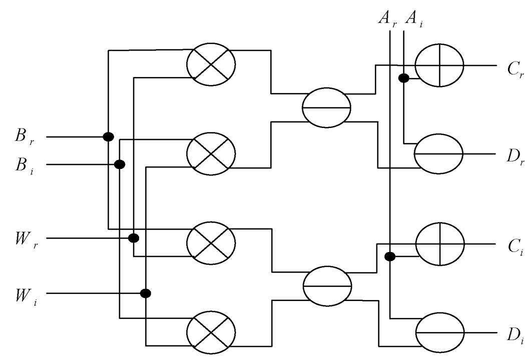


图 2 传统基-2 DIT FFT 蝶形算法结构

Fig. 2 Traditional base-2 DIT FFT butterfly algorithm structure

传统的基-2 DIT 蝶形算法由 4 个乘法器、3 个加法器和 3 个减法器模块组成[20]。从数学上来看,输出 C 和 D 可以表示为:

$$\begin{cases}(C_r+\mathrm{i}C_i)=(A_r+\mathrm{i}A_i)+(B_r+\mathrm{i}B_i)(W_r+\mathrm{i}W_i)\\(D_r+\mathrm{i}D_i)=(A_r+\mathrm{i}A_i)-(B_r+\mathrm{i}B_i)(W_r+\mathrm{i}W_i)\end{cases} \tag{8}$$

将实部和虚部扩展,上述方程可得到:

$$\begin{cases}(C_r+\mathrm{i}C_i)=A_r+B_rW_r-B_iW_i+\mathrm{i}A_i+\\ \quad \mathrm{i}B_rW_i+\mathrm{i}B_iW_r\\(D_r+\mathrm{i}D_i)=A_r-B_rW_r+B_iW_i+\mathrm{i}A_i-\\ \quad \mathrm{i}B_rW_i-\mathrm{i}B_iW_r\end{cases} \tag{9}$$

为便于 FPGA 实现,将实部和虚部独立分开,式(9)改写为:

$$\begin{cases}C_r=A_r+B_rW_r-B_iW_i\\C_i=A_i+B_rW_i+B_iW_r\\D_r=A_r-B_rW_r+B_iW_i\\D_i=A_i-B_rW_i-B_iW_r\end{cases} \tag{10}$$

式(8)中的复数运算可以分解为简单的乘法、加法和减法运算,如式(10)。通过算术优化,上述4个方程中的运算可以进一步简化,式(10)中包含具有公共乘法因子的重复项。通过向方程中增加和减去 ***BW***,式(10)可以被重新改写为:

$$\begin{cases} C_r = A_r + B_r W_r - B_i W_i + B_r W_i - B_r W_i \\ C_i = A_i + B_r W_i + B_i W_r + B_i W_i - B_i W_i \\ D_r = A_r - B_r W_r + B_i W_i + B_r W_i - B_r W_i \\ D_i = A_i - B_r W_i - B_i W_r + B_r W_i - B_r W_i \end{cases} \tag{11}$$

通过重新排列重复项,式(11)可以改写为:

$$\begin{cases} C_r = A_r + B_r(W_r + W_i) - W_i(B_r + B_i) \\ C_i = A_i + B_i(W_r - W_i) + W_i(B_r + B_i) \\ D_r = A_r - B_r(W_r + W_i) + W_i(B_r + B_i) \\ D_i = A_i - B_i(W_r - W_i) - W_i(B_r + B_i) \end{cases} \tag{12}$$

式中:$(B_r + B_i)$项出现在所有方程中,可以将其视为4个计算中的共享加法器;$(W_r + W_i)$和$(W_r - W_i)$项不涉及任何算术计算,因为旋转因子的值是常数,并且具有对称性和周期性,因此可以存储在ROM中进行读取。式(12)涉及的算术计算如图3所示。优化后的基-2 DIT蝶形算法使用了3个乘法器和7个加法器/减法器模块。

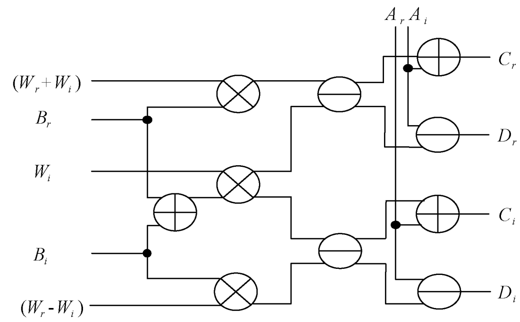


图3　优化的基-2 DIT FFT蝶形算法结构

Fig. 3　Optimized structure of base-2 DIT FFT butterfly algorithm

表1为在传统和优化后的基-2 DIT蝶形算法中,1 024点FFT所需的乘法器数量。

表1　传统和优化基-2 DIT FFT算法的计算复杂度

Table 1　Computational complexity of traditional and optimized base-2 DIT FFT algorithms

DFT长度	级数	全部蝶形数量	传统模式所需乘法器	优化后所需乘法器
1 024	10	5 120	20 480	15 360

通过使用经过算术优化的基-2 DIT蝶形算法来实现1 024点FFT,可以显著节省乘法器的使用。这种优化后的基-2 DIT蝶形算法应用于1 024点DIT FFT,如图4所示。优化后的基-2 DIT蝶形算法结构标记为“R-2”,每个阶段的蝶形计算需要从ROM中读取旋转因子。

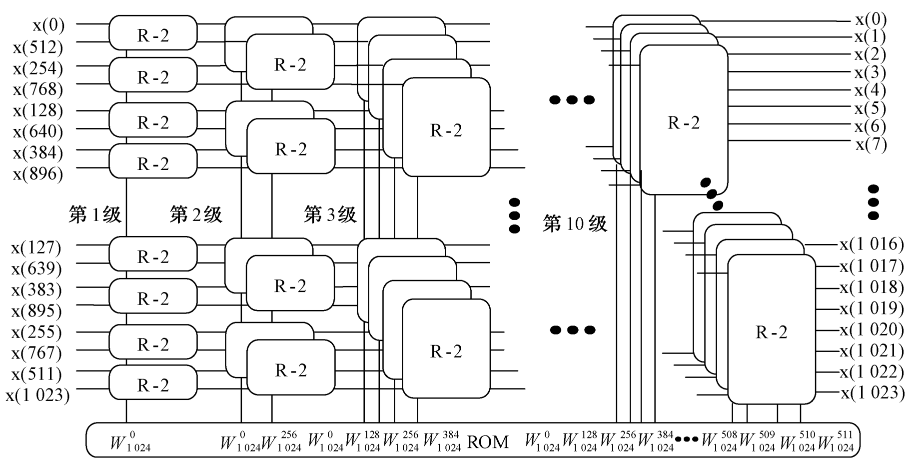


图4　1 024点DIT-FFT架构

Fig. 4　1 024 point DIT-FFT architecture

基-2 DIT蝶形算法通过算术优化有效降低了单个蝶形的计算复杂度,因此可使用优化后的基-2 DIT蝶形算法来实现1 024点DIT的FFT算法。

2.2　利用WFTA算法优化3点DFT

3点DFT模块也是实现3 072点FFT的关键模块。如果根据3点DFT变换的定义直接实现,则至少需要两个复数乘法器[21]。3点的WFTA算法如下:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3^1 & W_3^2 \\ 1 & W_3^2 & W_3^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 3/2 & \\ & & -\mathrm{j}\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix} \tag{13}$$

为减少乘法器的使用，根据式(13)可知，3点DFT结构实现[22]如图5所示，其中$x(0)$、$x(1)$和$x(2)$表示3点DFT的输入信号，$Y(0)$、$Y(1)$和$Y(2)$表示3点DFT的运算结果[23]。

$$\begin{cases} Y(0) = x(0) + x(1) + x(2) \\ Y(1) = x(0) + x(1)e^{-j2\pi/3} + x(2)e^{-j4\pi/3} = \\ \quad x(0) + x(1)(-1/2 - j\sqrt{3}/2) + \\ \quad x(2)(-1/2 + j\sqrt{3}/2) = \\ \quad x(0) - 1/2(x(1) + x(2)) - \\ \quad j\sqrt{3}/2(x(1) - x(2)) \\ Y(2) = x(0) + x(1)e^{-j4\pi/3} + x(2)e^{-j2\pi/3} = \\ \quad x(0) + x(1)(-1/2 + j\sqrt{3}/2) + \\ \quad x(2)(-1/2 - j\sqrt{3}/2) = x(0) - \\ \quad 1/2(x(1) + x(2)) + j\sqrt{3}/2(x(1) - \\ \quad x(2)) \end{cases} \tag{14}$$

由图5可知，3点WFTA使用了6个复数加减法器。通过化简移位操作，该算法实际上仅需1个复数乘法器(1个复数乘法器包括4个乘法器和3个加法器)，从而节约了资源。此外，该结构采用了流水线操作方式，进一步提高了实现效率[23]。

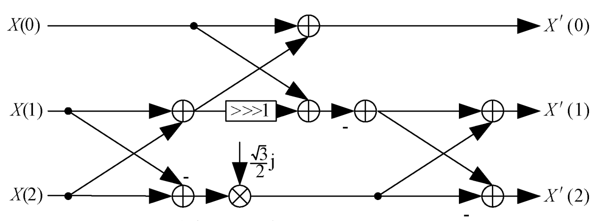


图5 3点WFTA流程

Fig. 5 3 point WFTA flow chart

2.3 Cooley-Tukey-WFTA算法资源对比

3 072点FFT算法由3个1 024点FFT和1个3点DFT模块实现。优化后的基-2 DIT蝶形算法结构在乘法器资源消耗方面比传统基-2 DIT蝶形算法结构减少了24.96%。具体的乘法器资源如表2所示。

表2 乘法器资源使用情况

Table 2 Multiplier resource usage

	1 024点FFT	3点DFT	3 072点FFT
传统模式	20 480	8	61 448
算法优化后	15 360	4	46 088

3 优化算法的FPGA实现

本文提出的3 072点FFT算法采用了优化后的基-2 DIT蝶形算法结构和优化后的3点WFTA算法。本文比较了传统基-2 DIT算法和优化后的基-2 DIT算法在1 024点FFT中的乘法利用率，并对优化后的3 072点FFT处理速率与Xilinx LTE FFT IP核处理速率进行了比较。

3.1 总体架构

针对DFT-S-OFDM架构，本文设计输出数据速率为40.96 MHz，子载波间隔为10 kHz，因此需要进行4 096点FFT运算，并且为了降低PAPR，且由于DFT和IDFT处理不能完全抵消[24]，最终所以选择3 072点FFT进行预处理。

在FPGA上实现的3 072点FFT运算时，根据Cooley-Tukey混合基算法原理[25]，将3 072点FFT分解为3个1 024点FFT和1个3点DFT。整体结构如图6所示。

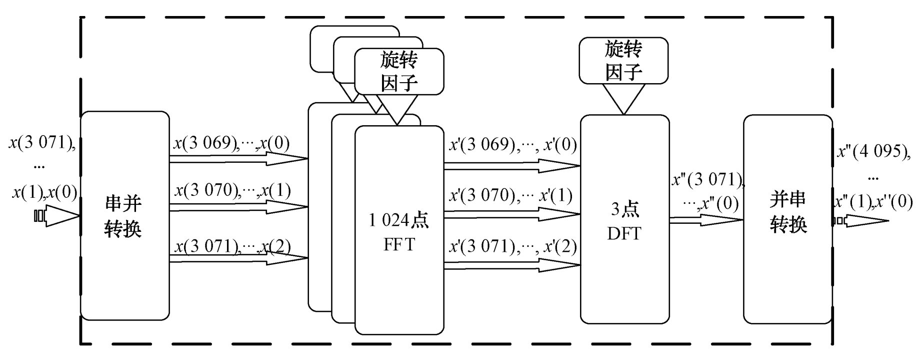


图6 整体结构

Fig. 6 Overall structure drawing

将输入数据进行串并转换，数据按列输入，形成3行1 024列，数据格式如图7所示。

对每一行数据进行1 024点FFT运算，并与对应的旋转因子相乘进行系数调整，得到3路并行数据。1 024点FFT运算方案通过将多个基-2蝶形模块串联进行流水线处理，实现数据的连续处理。每个基-2模块配有相应的RAM，用于存储输入数据和中间级数据，如图8所示。

结合小数组WFTA算法，对3行1 024点FFT的输出数据进行3点DFT运算，并与对应的旋转因子相乘，依次完成1 024次3点DFT运算，得到3路并行数据。随后，将这3路并行数据通过RAM跳地址方式进行交织，并在中间插入1 024个0，以实现扩频至4 096点输出。

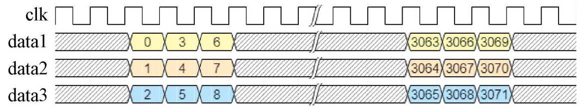


图 7　数据输入格式

Fig. 7　Data entry format

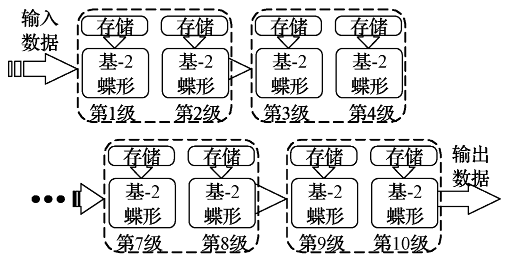


图 8　基 2 流水线处理

Fig. 8　Base 2 pipeline processing

由于 FPGA 中无法直接计算浮点数，为了防止数据溢出并避免数据长度的增加，同时为后续处理提供合适长度的数据，需要对输出结果进行处理。输入数据为 32 bit，为了在 FPGA 中进行运算，将旋转因子扩大 32 768 倍存入 ROM 中。计算完成后，输出结果右移 15 bit，以保证精度并防止数据溢出。

3.2　算法的 FPGA 适应性改进

传统基-2 DIT 算法与优化后的基-2 DIT 蝶形算法在 RTL 实现上的对比如图 9 所示。

与传统的基-2 DIT 算法相比，优化后的基-2 DIT 蝶形算法以增加一个加法器为代价，减少了一个乘法器。优化后的基-2 DIT 蝶形算法减少了一个乘法器，在 1 024 点 FFT 的实现中，乘法计算复杂度由 $O(20\ 480)$ 降低到 $O(15\ 360)$，这一优化显著降低了计算复杂度。

3 点 WFTA 算法通过式(13)实现 DFT，其中 1/2 的计算通过右移操作完成，仅需一个复数乘法器，计算复杂度由 $O(2)$ 降低到 $O(1)$。

整个算法和未优化前的算法相比，系统乘法复杂度从 $O(40\ 960)$ 降低到 $O(15\ 360)$。

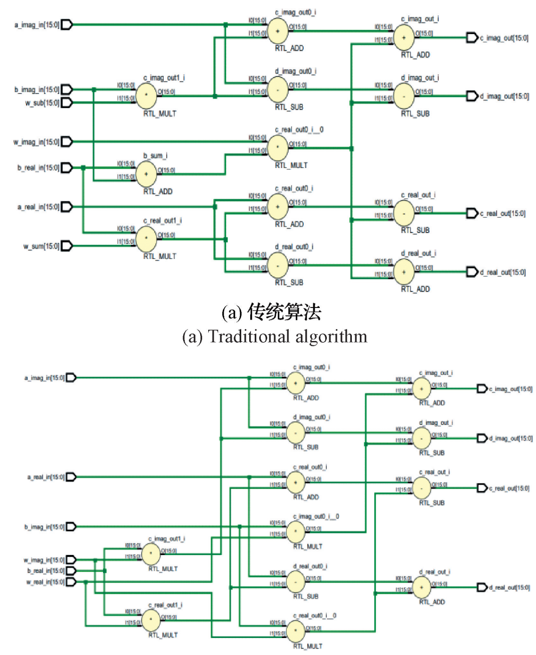
3.3　FPGA 上 3 072 点 FFT 算法的性能测试与验证

1)处理速率对比

将 Xilinx LTE FFT IP 核(LTE 核)与本文设计的 3 072 点 FFT 算法进行对比，仿真结果如图 10 所示。在处理相同数据时，本文方法的处理时间约为 29.57 μs，而 LTE 核的处理时间约为 34.41 μs。因此，本文方法的处理速度比 LTE 核快 14.07%。

2)FPGA 平台测试

FPGA 平台如图 11 所示，利用 Cooley-Tukey-WFTA 算法优化在 FPGA 上实现 3 072 点 FFT。将所得结果与数据仿真软件的结果进行对比，如图 12 所示，时域输入数据为周期锯齿波函数，利用 Cooley-Tukey-WFTA 算法优化所得结果与传统方法所得结果散点图可以看出，二者数据结果基本一致；在幅频曲线中，仿真数据和 FPGA 实现的数据在横坐标频率点均为 1、385、769、1 153、2 516、2 945、3 329、3 713 Hz 时，纵坐标幅值的区模后分别为 10 752、4 014.09、2 172.23、1 663.01、1 536、1 661.16、



(a) 传统算法

(a) Traditional algorithm

(b) 优化算法

(b) Optimized algorithm

图 9　基-2 DIT 算法 RTL 对比

Fig. 9　RTL comparison diagram of base-2 DIT algorithm

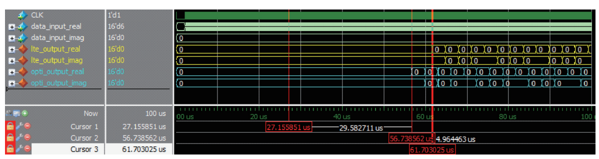


图 10　处理时间对比

Fig. 10　Processing time comparison chart

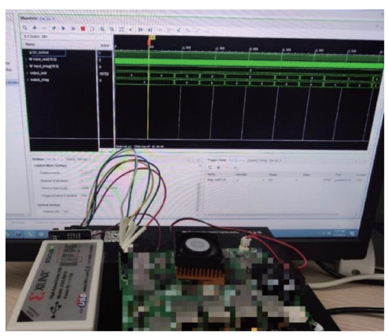


图 11　FPGA 平台测试

Fig. 11　FPGA platform testing

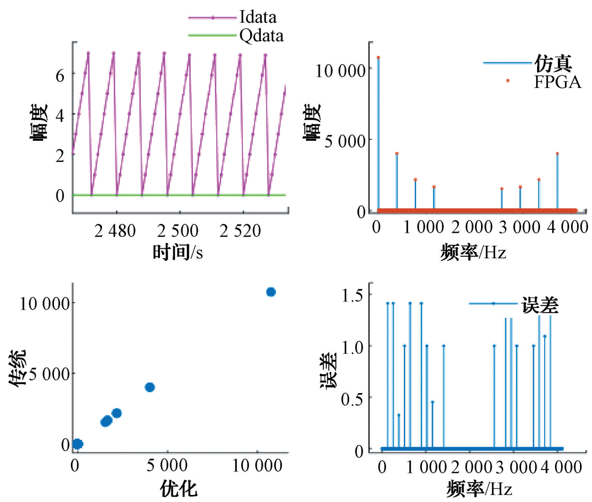


图 12　FPGA 结果与仿真数据对比

Fig. 12　Comparison of FPGA results with simulation data

2 172.23、4 014.85 dB;通过 FFT 进行数据处理后,FPGA 输出的数据与仿真结果在部分频率点上存在微小误差,误差在 2 dB 以内。通过对比,二者整体频率和幅值曲线基本一致。

4　结　论

本文提出了一种适用于 DFT-S-OFDM 通信系统的 3 072 点优化 FFT 算法,并成功在 FPGA 平台上实现。为解决现有 DFT-S-OFDM 实现方法在资源利用和计算速度上的不足,本文结合 Cooley-Tukey FFT 算法和 WFTA 算法,设计了 Cooley-Tukey-WFTA 算法,并成功应用于 3 072 点 FFT 优化。此方法通过优化 1 024 点 FFT 的基-2 DIT 蝶形算法结构,并引入 WFTA 算法优化 3 点 DFT 的乘法器资源,实现了高效的 FFT 计算。仿真和实验结果表明,该算法在乘法器资源消耗上比传统方法减少了 24.96%,且处理速度提升了 14.07%,大大降低了 FFT 的计算复杂度,证明了其在实际应用中的有效性。与传统算法相比,本文提出的 3 072 点 FFT 优化算法在资源消耗和性能方面展现出显著优势,为 DFT-S-OFDM 技术在实际通信系统中的高效实现提供了新的解决方案。

参 考 文 献

[1] SALEH M A, HAMOOD M, AHMED M. Reducing peak to average power ratio (PAPR) of OFDM systems using complex BIFORE precoding transform [C]. IOP Conference Series: Materials Science and Engineering, 2021, 1058(1):012067.

[2] 李轩,张雪冰. 基于 DFT 扩频技术的 FBMC/OQAM 系统峰均比研究[J]. 电子测量技术,2022,45(21):175-180.
LI X, ZHANG X B. Research on peak-to-average power ratio of FBMC/OQAM system based on DFT spread spectrum technology [J]. Electronic Measurement Technology, 2022, 45(21):175-180.

[3] SAHIN A, YANG R, BALA E, et al. Flexible DFT-S-OFDM: Solutions and challenges [J]. IEEE Communications Magazine, 2016, 54(11):106-112.

[4] 龚彤艳,张广婷,贾鹏鹏,等. 一种偶数基 Cooley-Tukey FFT 高性能实现方法[J]. 计算机科学,2020,47(1):31-39.
GONG T Y, ZHANG G T, JIA H P, et al. A high-performance implementation method of even-base Cooley-Tukey FFT [J]. Computer Science, 2020, 47(1):31-39.

[5] 陈山枝,孙韶辉,康绍莉,等. 6G 星地融合移动通信关键技术[J]. 中国科学:信息科学,2024,54(5):1177-1214.
CHEN SH ZH, SUN SH H, KANG SH L, et al. Key technologies of 6G satellite-terrestrial integrated mobile communication [J]. Scientia Sinica (Informationis), 2024, 54(5):1177-1214.

[6] 郭俊,刘鹏,杨昕遥,等. 大点数 FFT 在"申威 26010"上的并行优化[J]. 浙江大学学报(工学版),2024,58(1):78-86.
GUO J, LIU P, YANG X Y, et al. Parallel optimization of large-point FFT on "Shenwei 26010" [J]. Journal of Zhejiang University (Engineering Science), 2024, 58(1):78-86.

[7] 高立宁,马潇,刘腾飞,等. 基于超大点数 FFT 优化算法的研究与实现[J]. 电子与信息学报,2014,36(4):998-1002.
GAO L N, MA X, LIU T F, et al. Research and implementation of optimization algorithm based on ultra-large-point FFT [J]. Journal of Electronics & Information Technology, 2014, 36(4):998-1002.

[8] 许鹏飞,樊宁波,胡向晖. 一种 3 072 点 FFT 运算的实现方法:CN102831099B[P]. 2015-04-22.
XU P F, FAN N B, HU X H. An implementation method of 3 072-point FFT operation: CN102831099B [P]. 2015-04-22.

[9] LIU L, CHENG C, CUI Y, et al. Data processing method and processor based on 3 072-point fast Fourier transformation, and storage medium: US2018165250A1[P]. 2018-06-14.

[10] 洪钦智,王志君,郭一凡,等. 一种支持多数据块混合处理的 FFT 优化方法[J]. 西安电子科技大学学报,2022,49(6):42-50.
HONG Q ZH, WANG ZH J, GUO Y F, et al. An FFT optimization method supporting multi-data block hybrid processing [J]. Journal of Xidian University,

2022，49(6):42-50.
[11] 杨筱玥.功率分析仪电能参数计算研究与实现[D].成都:电子科技大学,2022.
YANG X Y. Research and implementation of power analyzer electrical energy parameter calculation [D]. Chengdu: University of Electronic Science and Technology of China, 2022.
[12] ZHOU Y X, SHAO L. Design of mixed-radix FFT algorithm based on FPGA[C]. 2022 7th International Conference on Communication, Image and Signal Processing (CCISP), 2022:418-422.
[13] 李浩,王厚军,肖磊,等.高速并行数字调制信号的产生与实现[J].仪器仪表学报,2023,44(2):110-118.
LI H, WANG H J, XIAO L, et al. Generation and implementation of high-speed parallel digital modulation signals [J]. Chinese Journal of Scientific Instrument, 2023, 44(2): 110-118.
[14] 侯晓晨,孟骁,陈昊.基于FPGA的混合基FFT算法设计与实现[J].太赫兹科学与电子信息学报,2021,19(2): 303-307.
HOU X CH, MENG X, CHEN H. Design and implementation of hybrid radix FFT algorithm based on FPGA [J]. Journal of Terahertz Science and Electronic Information Technology, 2021, 19(2): 303-307.
[15] 张建宏,武锦辉,刘吉,等.基于FPGA的多普勒雷达测速系统[J].国外电子测量技术,2019,38(12): 72-75.
ZHANG J H, WU J H, LIU J, et al. Doppler radar speed measurement system based on FPGA [J]. Foreign Electronic Measurement Technology, 2019, 38(12):72-75.
[16] 武铮,金旭,安虹.申威26010众核处理器上Winograd卷积算法的研究与优化[J].计算机研究与发展,2024, 61(4): 955-972.
WU ZH, JIN X, AN H. Research and optimization of Winograd convolution algorithm on Shenwei 26010 many-core processor [J]. Journal of Computer Research and Development, 2024, 61(4): 955-972.
[17] SAM PAUL S, BIBIN GLITTAS A X, SELLATHURAI M. Reconfigurable 2, 3 and 5-point DFT processing element for SDF FFT architecture using fast cyclic convolution algorithm[J]. Electronics Letters, 2020, 56(12):592-594.
[18] GARRIDO M. A survey on pipelined FFT hardware architectures [J]. Journal of Signal Processing Systems, 2022, 94(11):1345-1364.
[19] 吴琦,唐泽华,汪微龙,等.IM-DDO OFDM-PON中全并行1 024定点FFT优化的研究与实现[J].光通信技术,2016, 40(10):4.
WU Q, TANG Z H, WANG W L, et al. Research and implementation of all-parallel 1 024 fixed-point FFT optimization in IM-DDO OFDM-PON [J]. Optical Communication Technology, 2016, 40(10):4.
[20] 王海淼.基于FPGA的实时FFT分析方法研究[D].哈尔滨:哈尔滨工业大学,2021.
WANG H M. Research on real-time FFT analysis method based on FPGA [D]. Harbin: Harbin Institute of Technology, 2021.
[21] 夏凯锋,周小平,吴斌.任意点存储器结构FFT处理器地址策略[J].北京理工大学学报,2017,37(9): 953-957.
XIA K F, ZHOU X P, WU B. Address strategy of FFT processor with arbitrary point memory structure [J]. Transactions of Beijing Institute of Technology, 2017, 37(9):953-957.
[22] 马翠梅,陈禾,杜青.一种小面积的基-3蝶形单元设计[J].北京理工大学学报,2013,33(10): 1067-1071.
MA C M, CHEN H, DU Q. Design of a small-area radix-3 butterfly unit [J]. Transactions of Beijing Institute of Technology, 2013, 33(10):1067-1071.
[23] 孙重磊.基于FPGA的24点离散傅里叶变换结构设计[J].电子科技,2012,25(9):132-135.
SUN ZH L. Design of 24-point discrete Fourier transform structure based on FPGA [J]. Electronic Science and Technology, 2012, 25(9):132-135.
[24] 陈晨,费丹,郑鹏,等.5G TM信号无线信道测量平台研究与实现[J].电子测量与仪器学报,2023,37(11): 91-99.
CHEN CH, FEI D, ZHENG P, et al. Research and implementation of 5G TM signal wireless channel measurement platform [J]. Journal of Electronic Measurement and Instrumentation, 2023, 37(11):91-99.
[25] YANG C, WU J, XIANG S, et al. A high-throughput and flexible architecture based on a reconfigurable mixed-radix FFT with twiddle factor compression and conflict-free access [J]. IEEE Transactions on Very Large Scale Integration Systems, 2023, 31(10): 1472-1485.

作者简介

秦昊,硕士研究生,助理工程师,主要研究方向为动态测试与智能控制技术。
E-mail:455245405@qq.com
尤文斌,博士,教授,主要研究方向为动态测试与智能控制技术。