ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 37,No. 4,Jul. 2022,pp. 917－925
DOI:10. 16337/j. 1004-9037. 2022. 04. 019

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

低硬件成本256点FFT处理器的IP核设计

于　建，范浩阳

（河北民族师范学院物理与电子工程学院，承德 067000）

摘　要：设计了一种基于现场可编程门阵列（Field programmable gate array，FPGA）的低硬件成本256点快速傅里叶变换（Fast Fourier transform，FFT）处理器的IP核。采用按频率抽取的基-2^4算法和单路延迟负反馈（Single-path delay feedback，SDF）流水线架构用于减少旋转因子的复数乘法运算复杂度。为了降低硬件成本，提出了一种串接正则有符号数（Canonical signed digit，CSD）常数乘法器取代常用的布斯乘法器用来完成旋转因子W_{256}^{i}与对应序列的复数乘法运算，同时这种乘法器还能够移除存储旋转因子系数的只读存储器（Read only memory，ROM）。该处理器IP核基于QUARTUS PRIME平台进行综合，在Cyclone 10LP FPGA上实现。结果显示，该FFT处理器最高工作频率为100 MHz，对于24位符号数FFT运算，逻辑单元（Logic elements，LEs）使用量与记忆体位（Memory bits，MBs）使用量仅为3 978 LEs和6 456 MBs。

关键词：快速傅里叶变换；旋转因子；串接CSD常数乘法器；流水线架构；硬件成本

中图分类号：TN47　　**文献标志码：**A

Design of IP Core of Low Hardware-Cost 256-Point FFT Processor

YU Jian，FAN Haoyang

（Physics and Electronic Engineering College，Hebei Normal University for Nationalities，Chengde 067000，China）

Abstract：An IP core of low hardware-cost 256-point fast Fourier transform（FFT）processor is designed based on field programmable gate array（FPGA）. In order to reduce the complexity of twiddle factor calculation，the radix-2^4 algorithm based on decimation in frequency and the single-path delay feedback（SDF）pipelined architecture are adopted. For reducing hardware-cost，a cascade canonical signed digit（CSD）complex multiplier instead of conventional Booth multiplier is proposed for the operation of twiddle factor W_{256}^{i} multiplied by the corresponding sequences. Also，the proposed cascade CSD multiplier can remove read only memory（ROM）for storing coefficients of twiddle factors. The IP core is synthesized by using QUARTUS PRIME tool and is implemented on Cyclone 10LP FPGA. The result shows that the proposed FFT design can work under a maximum clock frequency of 100 MHz which occupies only 3 978 logic elements（LEs）and 6 456 memory bits（MBs）hardware-resource for 24-bit signed number FFT operation.

Key words: fast Fourier transform（FFT）；twiddle factor；cascade CSD constant multiplier；pipelined architecture；hardware-cost

基金项目：河北省自然科学基金（F2020101001）；河北省引进留学人员资助项目（C20210301）；河北省承德市科学技术研究与发展计划（202001B014）；河北民族师范学院科学技术研究项目（PT2019026）。
收稿日期：2021-04-11；**修订日期：**2022-07-13

引　言

快速傅里叶变换(Fast Fourier transform, FFT)是现代通信领域非常重要的技术,尤其在正交频分复用(Orthogonal frequency division multiplexing, OFDM)并行传输技术中的广泛应用[1],如IEEE 802.11a/g/n、全球微波互联接入(Worldwide interoperability for microwave access, WiMAX)、无线个人局域网(Wireless personal area networks, WPANs)和长期演进(Long term evolution ,LTE)等[2]。在不同的OFDM系统中,载波的数量正比于FFT变换长度,例如,WiMAX系统中FFT的采样点数支持256点FFT变换长度,因此设计一种有效的256点FFT处理器非常必要。

在OFDM系统中的物理层,FFT模块是最为复杂的运算模块,为了减少其硬件成本,已有很多学者对其进行了研究。李成诗等[3]在设计FFT处理器时,利用坐标旋转数字计算(Coordinate rotation digital computer, CORDIC)算法降低硬件成本消耗;王琳[4]提出通过优化基-4算法结构减少FFT处理器的硬件资源消耗;Fan等[5]利用布斯乘法器与CSD(Canonic signal digit)常数乘法器混合方案减少其所设计的FFT处理器的硬件成本;Wang等[6]为了减少硬件成本,提出了一种2乘法器和3加法器流水线型蝶形单元用于处理FFT处理器的实现;杨琳琳等[7]通过对蝶形单元结构的优化,减少乘法器的数目,从而达到减少FFT处理器硬件成本的目的;孙晓锋等[8]提出了基于Turbo结构的FFT处理器实现方案来降低硬件开销。

复数乘法器在FFT处理器的设计中占用主要的硬件资源,以往的这些研究在处理旋转因子 W_{256}^{i} 的复数乘法时,都只利用常用的布斯乘法器与只读存储器ROM来完成,无法有效地控制其所消耗的硬件资源。基于此,本文设计了一种基于现场可编程门阵列(Field programmable gate array, FPGA)的低硬件成本256点FFT处理器IP核,为了减少旋转因子的计算复杂度,采用了基-2^4算法和SDF流水线架构来完成设计,提出了一种串接CSD常数乘法器,通过简单的加法单元、移位单元以及选择单元来完成旋转因子 W_{256}^{i} 与输入序列的复数乘法运算,同时无须ROM对旋转因子系数进行存储,大幅降低了整个FFT处理器设计的硬件成本。

1　设计思考

若输入序列为$x(n)$,那么N点离散傅里叶变换(Discrete Fourier transform, DFT)$X(k)$定义为

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

式中旋转因子 $W_N^{nk}=\mathrm{e}^{-\mathrm{j}2\pi nk/N}$。

直接实现式(1)中的离散傅里叶变换需要消耗大量的硬件成本和计算时间。因此Tukey与Cooley提出了基-2快速傅里叶变换算法用于减少离散傅里叶变换的计算时间与硬件资源消耗。为了减少旋转因子计算复杂度,基-4算法应运而生,虽然能够降低旋转因子的复杂度,但是蝶形单元结构复杂,不易于硬件实现。后来基-2^k算法[9]出现,较于基-2算法,不但具有与其同样简单的蝶形单元结构,还能够像基-4算法一样简化旋转因子的复杂度。基于基-2^k算法的256点FFT处理器的计算包括7个阶段。不管k值如何,基-2^k算法在每个阶段都保留了与基-2算法相同的蝶形单元架构,但是k值的不同会导致不同阶段旋转因子的不同。表1所示为不同k值下,256点FFT在每个阶段旋转因子的详细分布,其中的"$-\mathrm{j}$"为普通运算,只需对输入序列实部和虚部的位置进行交换,再对虚部求反即可。由表1可知,由于具有更低的复数乘法运算量与更低复杂度的旋转因子,基-2^4算法在处理256点FFT计算方面更具优势,因此本文的设计采用基-2^4算法。

通常来说,FFT处理器架构一般可分为两种不同的类型:基于记忆体架构(Memory-based architec-

表1　256点基-2^k算法旋转因子序列分布

Table 1　Sequence distribution of 256-point FFT twiddle factor for radix-2^k algorithm

算法	旋转因子序列							计算量
	1	2	3	4	5	6	7	
基-2	W_{256}	W_{128}	W_{64}	W_{32}	W_{16}	W_8	$-j$	642
基-2^2	$-j$	W_{256}	$-j$	W_{64}	$-j$	W_{16}	$-j$	492
基-2^3	$-j$	W_8	W_{256}	$-j$	W_8	W_{32}	$-j$	504
基-2^4	$-j$	W_{16}	$-j$	W_{256}	$-j$	W_{16}	$-j$	480

ture, MBA)和流水线架构[10]。前者一般由主处理单元,即蝶形单元,存储单元与控制逻辑所组成,这种架构虽然能够减少所需的硬件资源和功耗,但是其数据吞吐量低、延迟高,无法满足数据的实时处理需求。而后者虽然相对来说消耗更多的硬件资源,但其具有更高的数据吞吐量。流水线架构的FFT处理器的设计又可以分为两种风格:前向反馈(Feedforward)和后向反馈(Feedback)。前向反馈又可分为单路延迟转换(Single-path delay commutator, SDC)和多路延迟转换(Multi-path delay commutator, MDC)[11];负向反馈又可分为单路延迟负反馈(Single-path delay feedback, SDF)和多路延迟负反馈(Multi-path delay feedback, MDF)[12]。与前向反馈风格相比,负向反馈风格具有更简单的控制逻辑,负向反馈风格中的MDF架构较SDF架构需要消耗更多的硬件资源。由于SDF架构具有硬件资源消耗低、易于实现等特点,因此本文的设计采用SDF流水线架构。

2　256点FFT处理器IP核设计

图1所示为基-2^4算法256点FFT的SDF流水线架构图。由图1可知,本文设计的FFT处理器架构由两种类型的蝶形单元(BFⅠ型和BFⅡ型)、不同容量的延迟缓冲单元(用于给蝶形单元的输入提供恰当的数据以及数据整理)以及⊗代表的复数乘法单元所组成。控制信号用于切换蝶形单元的类型并为旋转因子的复数乘法运算提供合适的控制逻辑。为了获得更少的硬件成本收益,复数乘法运算全部由CSD常数乘法器来完成。

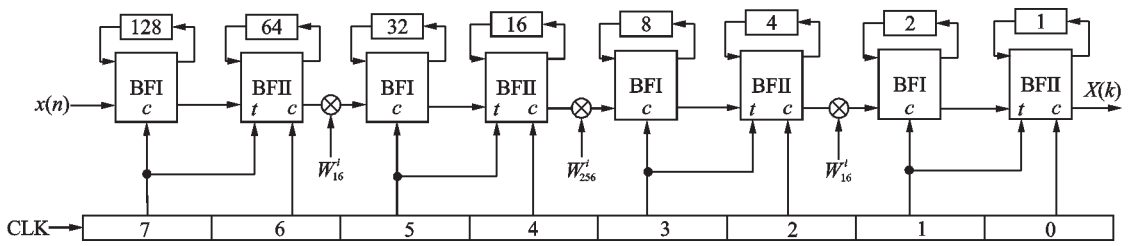


图1　基-2^4算法256点SDF FFT结构图

Fig.1　Block diagram of 256-point SDF FFT with radix-2^4 algorithm

2.1　蝶形单元

图2所示为本文所提出的256点FFT处理器IP核所需蝶形单元的具体结构,图2(a)为BFⅠ型,图2(b)为BFⅡ型。蝶形单元主要伴随缓冲单元的输入执行复数加法与复数减法运算[13],其中BFⅡ型蝶形单元除了进行基本的加减法运算,还要进行额外的"$-j$"运算,因此多出了额外的控制逻辑,如图2(b)所示。$x_r(n)$和$x_r(n+N/2)$代表输入序列的实部,$x_i(n)$和$x_i(n+N/2)$代表输入序列的虚部,$z_r(n)$和$z_r(n+N/2)$代表输出数据的实部,$z_i(n)$和$z_i(n+N/2)$代表输出数据的虚部。"t"和"c"是蝶形单元运算

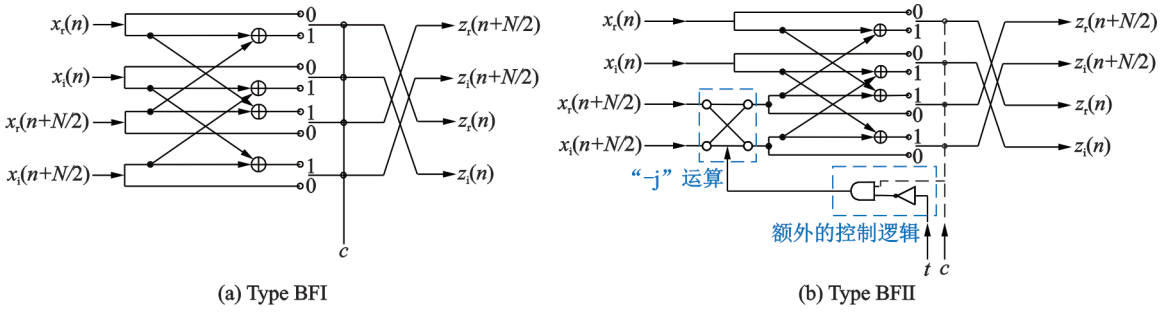


图2 蝶形单元架构

Fig.2 Block diagram of butterfly unit

时所需的控制信号。蝶形单元运算过程如下，输入数据序列的按照顺序储存到先进先出寄存器中，直到第$N/2$个数据（N为FFT点数），接下来的输入数据与先前存放在寄存器中的数据依次进行复数加减法运算。蝶形单元所进行的复数加法运算直接作为下一个阶段的输入数据，而减法运算结果存入下一阶段的先进先出延迟缓冲单元中。

2.2 CSD常数乘法器

一般来说，在特定阶段蝶形单元的输出数据都要与相应的旋转因子进行复数乘法运算来得到正确的输出。由图1可知，本文所设计的256点FFT处理器需要两种CSD常数乘法器用于旋转因子W_{16}^{i}与W_{256}^{i}的复数乘法运算。

2.2.1 适配于旋转因子W_{16}^{i}的CSD常数乘法器

在利用MATLAB对本文所设计的256点IP核进行建模时，表1中第2阶段与第6阶段的旋转因子W_{16}^{i}的i值分别由如下代码生成。

（1）第2阶段：tw=[zeros(1,16),ones(1,16),2*ones(1,16),3*ones(16)] i_stg2=[0.*tw,2.*tw,3.*tw]

（2）第6阶段：tw=repmat([0 0 0 0 0 2 4 6 0 1 2 3 0 3 6 9],16);i_stg6=tw(1,:)

由此可知，在设计CSD常数乘法单元时所需的旋转因子为：W_{16}^{0}，W_{16}^{1}，W_{16}^{2}，W_{16}^{3}，W_{16}^{4}，W_{16}^{6}，W_{16}^{9}。其中：$W_{16}^{0}=1$，$W_{16}^{4}=-\mathrm{j}$，$W_{16}^{6}=-\mathrm{j}\times W_{16}^{2}$，$W_{16}^{9}=-1W_{16}^{1}$，因此只需考虑3个旋转因子：$W_{16}^{1}$，$W_{16}^{2}$，$W_{16}^{3}$即可。又因为：$W_{16}^{1}=0.923\,9-\mathrm{j}0.382\,7$，$W_{16}^{2}=0.707\,1-\mathrm{j}0.707\,1$，$W_{16}^{3}=0.382\,7-\mathrm{j}0.923\,9$，所以只需对3个常数值（0.923 9，0.382 6，0.707 1）进行CSD化。表2所示为3个常数值的CSD表示，为了保证量化噪声比（Signal-to-quantization-noise ratio，SQNR），这里选择字长为12位，如果继续增加字长SQNR并无明显的变化。图3所示为不同k值的基-2^k算法在不同旋转因子字长的条件下所得到的SQNR值。由图3可知，当旋转因子字长为12位时，SQNR值达到了饱和。

表2 12位字长3常数值的CSD表示

Table 2 CSD representation of three constant values with 12-bit word-length

常数值	CSD表示											
0.923 9(Re{W_{16}^{1}})	1	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	1	0	0
0.707 1(Re{W_{16}^{2}})	1	0	$\bar{1}$	0	$\bar{1}$	0	1	0	1	0	0	0
0.382 7(Re{W_{16}^{3}})	0	1	0	$\bar{1}$	0	0	0	1	0	0	0	$\bar{1}$

为进一步节约硬件成本,找出能够在硬件实现上被重复利用的最优公共子表达式共享模块(Common sub-expression sharing block, CSE),表2中红色椭圆圈起的"101"或"$\bar{1}0\bar{1}$"即为最优的CSE,其中$\bar{1}$代表-1。

图4所示为适配于旋转因子W_{16}^{i}的CSD常数乘法器架构,其中"d"为输入的复数序列,输出端的2个4选1数据选择器完成输入的复数序列与旋转因子W_{16}^{i}进行复数乘法后的输出结果。综合后的结果显示,对比常用的布斯乘法器可节约72% LEs的使用量。

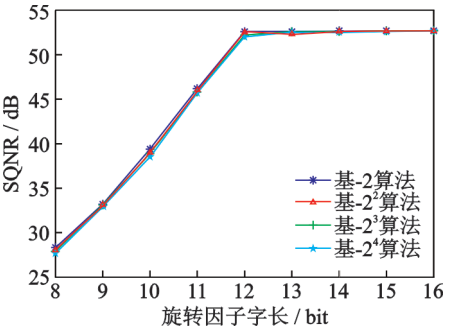


图3　不同旋转因子字长基-2^k算法SQNR比较

Fig.3　SNQR comparison of different twiddle factor word lengths for radix-2^k algorithm

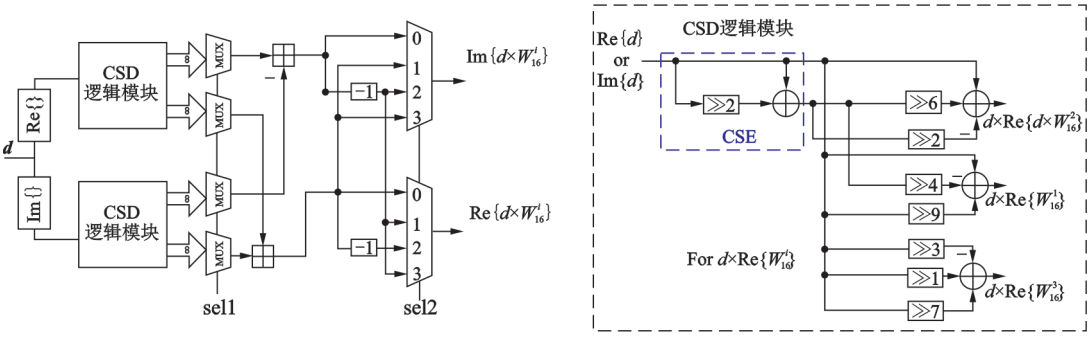


图4　适配于旋转因子W_{16}^{i}的CSD常数乘法器架构

Fig.4　Structure of CSD constant multiplier for twiddle factor W_{16}^{i}

2.2.2　适配旋转因子W_{256}^{i}的串接CSD常数乘法器

CSD常数乘法器的复杂度依赖于旋转因子常数值的个数,随着旋转因子常数值个数的增多,CSD常数乘法器的优势越来越不明显,当旋转因子W_N^i的N值超过64时,普通的CSD常数乘法器在设计上会相当复杂,与常用的布斯乘法器相比,在硬件成本控制上已无优势可言。为了处理旋转因子W_{256}^{i}的复数乘法运算,本文提出了一种串接CSD常数乘法器,通过对旋转因子W_{256}^{i}的i值进行合理的分解,达到减少旋转因子常数值的目的。根据旋转因子的对称性,将旋转因子W_{256}^{i}的i值平均划分到8个区域(A~H),设A区域旋转因子的表达式为$W_{256}^{p}=x_p+\mathrm{j}y_p$($p=0$~$32$),那么其他区域旋转因子的表达式可通过$A$区域的映射获得,详情如表3所示。

通过这样的方式,旋转因子W_{256}^{i}指数范围由$i=0$~255减少到$p=0$~31,再将参变量p分解为$p=4p_1+p_2$($p_1=1$~$8, p_2=0$~3),进一步缩小参变量指数的范围。最后,将分解后的常数值进行CSD化,并找出相应的可被重复利用的CSE完成设计。串接CSD常数乘法器需要进行2个阶段的复数乘法运算来完成一个完整的复数乘法运算,其具体的数学表达如式(2)所示。例如,如果要实现$d\times W_{256}^{13}$,令$p_1=3, p_2=1$即可。

$$d\times W_{256}^{p}=d\times W_{256}^{4p_1+p_2}=d\times W_{256}^{4p_1}\times W_{256}^{p_2}=d\times(\mathrm{Re}\{W_{256}^{4p_1}\}+\mathrm{j}\,\mathrm{Im}\{W_{256}^{4p_1}\})\times(\mathrm{Re}\{W_{256}^{p_2}\}+\mathrm{j}\,\mathrm{Im}\{W_{256}^{p_2}\}) \tag{2}$$

表4所示为适配于旋转因子W_{256}^{i}12组常数值的CSD表示,CSE_1为红色椭圆圈起的"101"(或$\bar{1}0\bar{1}$),CSE_2为蓝色椭圆圈起的"$10\bar{1}$"(或$\bar{1}01$),CSE_3为紫色椭圆圈起的"$1000\bar{1}$"(或$\bar{1}0001$)。

表3 8区域中旋转因子对应的映射

Table 3 Mapping of twiddle factor in eight symmetric regions

区域	p	实部	虚部	表达式
A	0～31	x_p	y_p	$x_p+\mathrm{j}y_p$
B	32～63	$-y_p$	$-x_p$	$-y_p-\mathrm{j}x_p$
C	64～95	y_p	$-x_p$	$y_p-\mathrm{j}x_p$
D	96～127	$-x_p$	y_p	$-x_p+\mathrm{j}y_p$
E	128～159	$-x_p$	$-y_p$	$-x_p-\mathrm{j}y_p$
F	160～191	y_p	x_p	$y_p+\mathrm{j}x_p$
G	192～223	$-y_p$	x_p	$-y_p+\mathrm{j}x_p$
H	224～255	x_p	$-y_p$	$x_p-\mathrm{j}y_p$

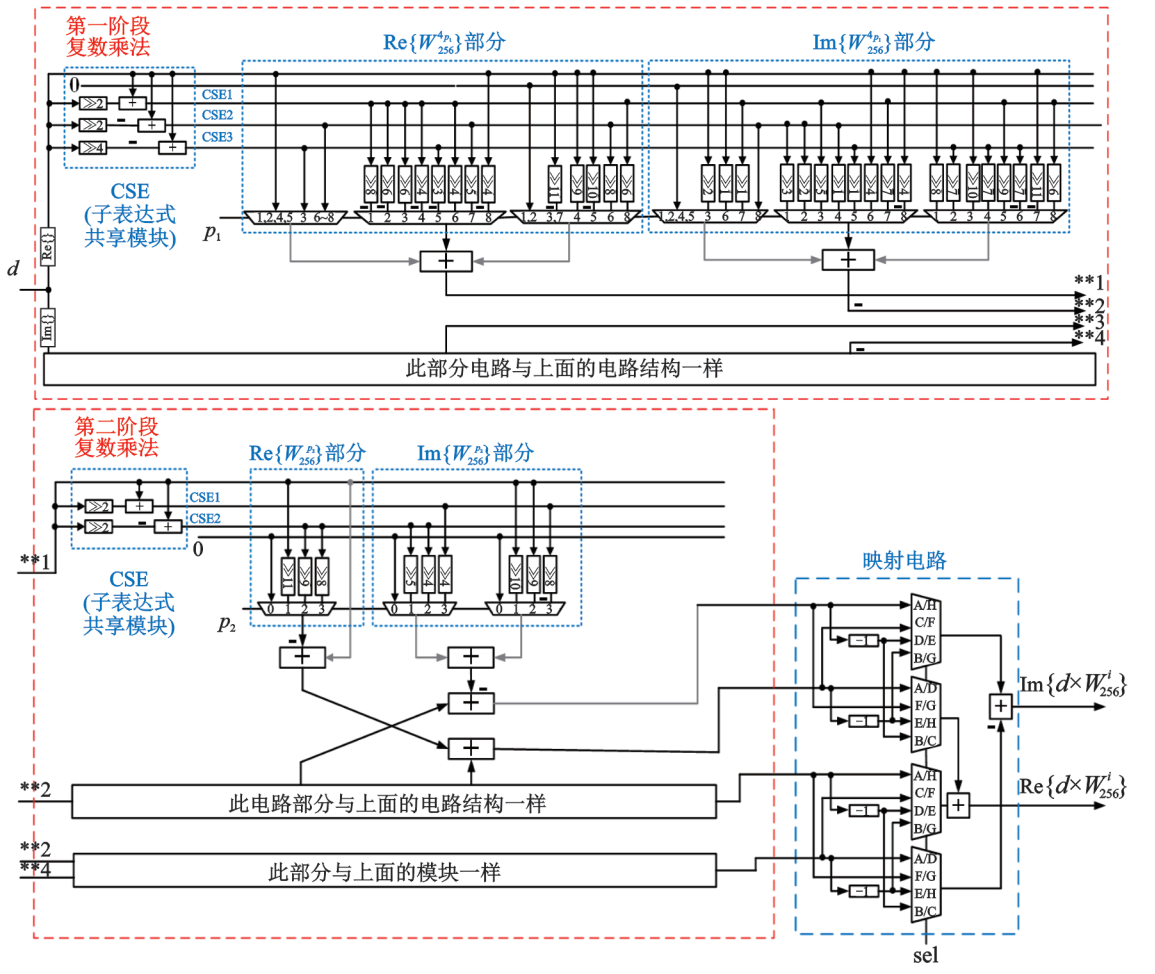
表4 12位字长12组常数值CSD表示

Table 4 CSD representation of twelve constant values with 12-bit word-length

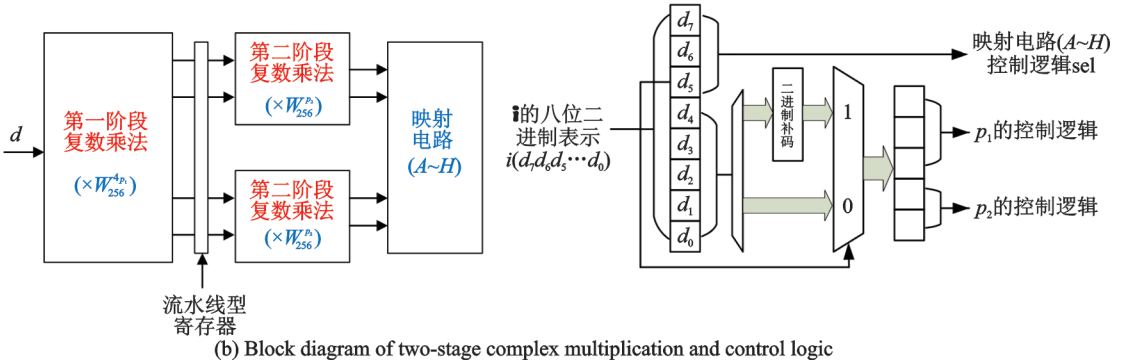
p_1	$\mathrm{Re}\{W_{256}^{4p_1}\}$													$-\mathrm{Im}\{W_{256}^{4p_1}\}$											
1	1	0	0	0	0	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	1	0	0	0
2	1	0	0	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	$\bar{1}$	0	0	0	$\bar{1}$
3	$\bar{1}$	0	0	0	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	0	$\bar{1}$	0	0	1	0	0	$\bar{1}$	0	$\bar{1}$	0	0	1	0
4	1	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	1	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	0	$\bar{1}$	0	0	0	$\bar{1}$
5	1	0	0	$\bar{1}$	0	0	0	$\bar{1}$	0	0	$\bar{1}$	0	0	$\bar{1}$	0	0	0	$\bar{1}$	0	0	0	$\bar{1}$	0	1
6	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	0	1	0	0	1	0	0	$\bar{1}$	0	0	0	$\bar{1}$
7	$\bar{1}$	0	$\bar{1}$	0	0	$\bar{1}$	0	$\bar{1}$	0	0	0	$\bar{1}$	0	$\bar{1}$	0	1	0	0	0	$\bar{1}$	0	1	0	$\bar{1}$
8	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	1	0	0	0	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	1	0	0	0
p_2	$\mathrm{Re}\{W_{256}^{p_2}\}$													$-\mathrm{Im}\{W_{256}^{p_2}\}$											
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	$\bar{1}$	0	0	0	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	1	0
2	1	0	0	0	0	0	0	0	0	0	$\bar{1}$	0	1	0	0	0	0	0	1	0	$\bar{1}$	0	0	1	0	0
3	1	0	0	0	0	0	0	0	0	$\bar{1}$	0	1	0	0	0	0	0	1	0	1	0	$\bar{1}$	0	$\bar{1}$	0

图5所示为适配于旋转因子 W_{256}^{i} 串接CSD常数乘法器的整体架构图。图5(a)所示为适配于旋转因子 W_{256}^{i} 的串接CSD常数乘法器的逻辑模块，其中最优CSE模块由加法单元右移模块完成，矩形方块即为右移模块，利用简单的硬件连接被放置在适当的位置，通过4选1选择单元与8选1选择单元用于得到最终的计算结果。

由于串接CSD乘法器需要两个阶段的乘法运算，会导致整体FFT处理器的运算速度降低，为了减少关键路径，在两阶段复数乘法接口处插入流水线型寄存器，如图5(b)左图所示。图5(b)右图所示为具体的控制逻辑，只需利用旋转因子 W_{256}^{i} 的指数 i，通过简单的二进制补码与选择操作便可获得对应于 p_1 和 p_2 的控制逻辑，综合后的结果显示其所消耗的硬件资源只占整体串接乘法器总硬件成本的0.96%。与常用的布斯乘法器相比，本文所提出的适配于旋转因子 W_{256}^{i} 的串接CSD常数乘法器能够节约34% LEs的使用量，同时能够移除存储旋转因子系数的ROM。



(a) Logic block of cascade CSD multiplier for twiddle factor W_{256}^{i}



(b) Block diagram of two-stage complex multiplication and control logic

图5　适配于旋转因子 W_{256}^{i} 串接CSD常数乘法器整体架构详解

Fig.5　Overall detailed cascade CSD multiplier for twiddle factor W_{256}^{i}

3　结果与比较

本文采用Verilog HDL对所设计的FFT处理器IP核进行建模,并利用Intel的QUARTUS　PRIME平台和Cyclone 10LP FPGA进行综合和性能评估。另外,与定点数运算相比,虽然浮点运算能够在一定程度上提高数据精度,但其所需的硬件电路复杂,硬件资源消耗大,而且运算速度慢,因此,本文的设计采用定

点字长方案,同时采取了逐级运算增加字长的方案,有效防止了数据溢出的问题。验证结果表明,对于256点24位符号数FFT运算,本文方案所设计的FFT处理器最大时钟频率可以达到100 MHz,延迟为255个时钟周期,只需消耗3 978 LEs和6 456 MBs的硬件资源。表5所示为本文方案与国内外其他设计方案在设计256点FFT处理器的比较结果。为了更加直观地进行比较,参考文献的设计方案同样是利用Verilog HDL进行建模,在基于Intel的QUARTUS PRIME平台和Cyclone 10LP FPGA进行综合和性能评估。由表5可知,文献[2]采用了基-16算法来减少旋转因子的复杂度,但其蝶形单元复杂不易于硬件实现,而且在处理256点FFT复数乘法运算时采用了常规的复数乘法器,硬件资源消耗高;文献[7]采用了基-4 MBA架构,虽然在一定程度上减少了实现时所需的硬件资源,但其延时大并在处理旋转因子复数乘法时采用了普通复数乘法器,相较于本文提出的全CSD常数乘法器方案,本文方案所占用的硬件资源更少、延时更小;文献[13]采用了多路径混合基MDC架构,在一定程度上提升了硬件使用效率与工作频率,但其硬件实现架构与控制逻辑复杂,而且同样利用普通复数乘法器完成256点FFT计算,硬件资源消耗高;文献[14]由于其所选择的算法在处理256点FFT计算时旋转因子更复杂、复数乘法的运算量更高(表1),因此在硬件成本控制方面并不理想。综上所述,本文方案在设计低硬件成本256点FFT处理器时更具优势。

表5 不同设计方案综合结果比较

Table 5 Performance comparison of the proposed scheme compared with previous implementations

来源	字长	算法与实现架构	时钟频率/MHz	延时/s	控制逻辑	硬件成本
文献[2]	10	R16SDF	101	255	简单	5 396 LEs 8 976 MBs
文献[7]	24	R4MBA	100	—	复杂	4 671 LEs 16 596 MBs
文献[13]	12	$R2^2$-4^2MDC	330	255	复杂	48 756 LEs 67 869MBs
文献[14]	12	R2MBA	100	—	复杂	4 989 LEs 18 986 MBs
本文	24	$R2^4$SDF	100	255	简单	3 978 LEs 6 456 MBs

图6所示为MODELSIM输出仿真结果的幅度值与MATLAB计算结果幅度值的比较(设输入序列的实部和虚部为1～256,即$x_r(n)$=[1：256],$x_i(n)$=[1：256]),其中"*"代表MATLAB计算结果,"△"代表MODELSIM输出仿真结果。由图6可知,两者的结果完全吻合,证明了设计的有效性。

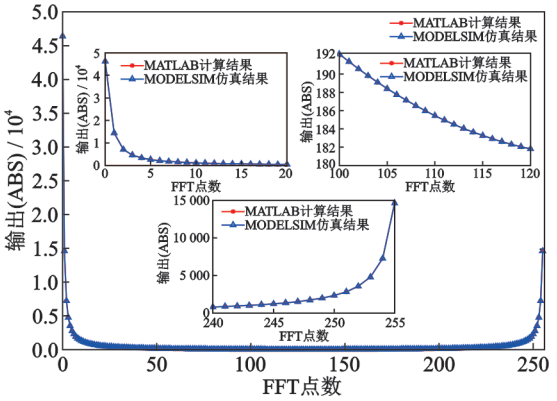


图6 MODELSIM仿真结果与MATLAB计算结果比较

Fig.6 Comparison between MODELSIM simulation result and MATLAB calculation result

4 结束语

本文设计了一种低硬件成本256点FFT处理器IP核,设计上采用了基-2^4算法与SDF流水线架构。为了最小化硬件资源的消耗,提出了一种适配于旋转因子W_{256}^{i}的串接CSD常数乘法器,利用旋转因子的对称性以及对旋转因子指数合理的分解,减少了旋转因子进行复数乘法运算所需的

常数值个数,令其所占用的硬件资源比常用的布斯乘法器减少了34%的LEs使用量,而且无须ROM对旋转因子系数进行存储。此外,为了保证处理器的工作频率,在串接CSD常数乘法器的设计中引入了流水线型寄存器,减少了关键路径,FFT处理器的最高工作频率达到了100 MHz,且延时只需255个时钟周期。基于QUARTUS PRIME平台的综合结果表明,对比已有的方案,本文方案在设计256点FFT处理器方面能够更加有效控制硬件成本。

参考文献:

[1] 李智勇,朱江. 面向5G的UFMC与OFDM技术频谱效率比较[J]. 信息通信,2017,174(6): 26-28.
LI Zhiyong, ZHU Jiang. Comparison of spectral efficiency between UFMC and OFDM for 5G[J]. Changjiang Information & Communications, 2017,174(6): 26-28.

[2] HUSSAIN M Z, PARVIN K N, ALI Z. Performance efficient FFT processor design[C]//Proceedings of International Conference on Global Trends in Signal Processing. Jalgaon: IEEE, 2017: 286-290.

[3] 李成诗,初建朋,李新兵,等. 基于CORDIC的一种高速实时定点FFT的FPGA实现[J]. 微电子学与计算机, 2004, 21(4): 88-96.
LI Chengshi, CHU Jianpeng, Li Xinbing, et al. A high speed real-time and fixed-point FPGA realization of a CORDIC based FFT processor[J]. Microelectronics & Computer, 2004, 21(4): 88-96.

[4] 王琳. 基于FPGA的低功耗可变精确度通用FFT处理器设计[D]. 济南:山东大学, 2012.
WANG Lin. Design of low power consumption and variable precision general FFT processor based on FPGA[D]. Jinan: Shandong University, 2012.

[5] FAN C P, LEE M S, SU G A. A low multiplier and multiplication costs 256-point FFT implementation with simplified radix-2^4 SDF architecture[C]//Proceedings of IEEE Asia Pacific Conference on Circuits and Systems. Singapore: IEEE, 2006: 1935-1938.

[6] WANG C, GAN W S, JONG C C. A low-cost 256-point FFT processor for portable speech and audio applications[C]//Proceedings of IEEE International Symposium on Integrated Circuits. Singapore: IEEE, 2007: 81-84.

[7] 杨琳琳,王新胜,王静. 低功耗浮点FFT处理器的设计[J]. 太赫兹科学与电子信息学报,2018, 16(2): 352-356.
YANG Linlin, WANG Xinsheng, WANG Jing. Design of a low power floating point FFT processor[J]. Journal of Terahertz Science and Electronic Information Technology, 2018,16(2): 352-356.

[8] 孙晓锋,刘晓杰,冀云成,等. 基于Turbo阵列的超高速流水线FFT设计与实现[J]. 中国集成电路, 2020,29(S4): 35-40,52.
SUN Xiaofeng, LIU Xiaojie, JI Yucheng, et al. Design and implementation of ultra high speed pipelined FFT based on Turbo array[J]. China Integrated Circuit, 2020,29(S4):35-40,52.

[9] CORTÉS A, VÉLEZ I, SEVILLANO J F. Radix r^k FFTs: Matricial representation and SDC/SDF pipeline implementation [J]. IEEE Transactions on Signal Processing, 2009, 57(7): 2824-2839.

[10] GANJIKUNTA G K, SAHOO S K. An area-efficient and low-power 64-point pipeline fast Fourier transform for OFDM applications[J]. Integration the VLSI Journal, 2016, 57: 125-131.

[11] 夏凯峰,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 Kaifeng, ZHOU Xiaoping, WU Bin. Generalized efficient address scheme for arbitrary point memory-based FFT processors[J]. Transaction of Beijing Institute of Technology, 2017, 37(9): 953-957.

[12] CHO T, LEE H. A high-speed low-complexity modified radix-2^5 FFT processor for high rate WPAN applications[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2013, 21(1): 187-191.

[13] JANG J K, KIM H K, SUNWOO M H, et al. Area-efficient scheduling scheme based FFT processor for various OFDM systems[C]//Proceedings of IEEE Asia Pacific Conference on Circuits and Systems. Chengdu, China: IEEE, 2018: 338-341.

[14] 刘剑丽,胡勤,黄山,等. 基于FPGA的快速傅里叶变换FFT处理器设计[J]. 通信与信息技术, 2020(1): 60-61, 68.
LIU Jianli, HU Qin, HUANG Shan, et al. Design of FFT processor based on FPGA[J]. Communication & Information Technology, 2020(1): 60-61,68.

作者简介:

于建(1979-),**通信作者**,男,副教授,研究方向:超大规模数字集成电路设计、虚拟仪器,E-mail: mjyujian1979@hotmail.com。

范浩阳(1987-),男,讲师,研究方向:超大规模数字集成电路设计,E-mail: 375004007@qq.com。

(编辑:王静)