

变维度 FFT 硬件加速器结构设计及 FPGA 实现

张多利,张玲佳,宋宇鲲

(合肥工业大学 电子科学与应用物理学院,安徽 合肥 230009)

摘 要: 本文设计了一种变维度 FFT 硬件加速器,其采用体-面-线的数据组织形式,提出了一种面划分^[1]兼多路并行的架构,从面和线 2 个层次展开计算,以面为基本存储单位,以线为基本计算单位,提高了 FFT 运算的并行度,减少了处理器间的数据交互,并通过乒乓预读取的设计和无冲突的地址调整,提高了整机的运算访存比. 本文设计的 FFT 加速器内含 32 个并行计算单元,支持 IEEE-754 标准下的 32 位单精度浮点数 32 点到 64 K 点一维 FFT 运算,32 点到 256 点的二维/三维 FFT 运算,且具有较强的可扩展性,可根据需要实现 $m \times n \times p$ 序列的 FFT 运算. 该设计已在 Xilinx Virtex6 FPGA 芯片上进行原型验证,最高工作频率 184.88 MHz.

关键词: FFT 硬件加速器; FFT 处理器; 地址调整模块; FPGA

中图分类号: TN47

文献标识码: A

文章编号: 1000-7180(2017)12-0034-06

Structure Design and FPGA Implementation of a Variable Dimension FFT Hardware Accelerator

ZHANG Duo-li, ZHANG Ling-jia, SONG Yu-kun

(School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009, China)

Abstract: In this paper, a variable dimension FFT hardware accelerator is designed. By using the data form of body-surface-line, the calculation is carried out from two levels of surface and line. It adopts surface partitioning^[1] and multiple parallel architecture to improve the parallelism of FFT. Furthermore by using Ping-Pong operation, pre-read and flexible address adjustment to conceal data transmitting time. In this paper, the FFT accelerator consists of 32 parallel computing units to carry out 2^n (n is from range of 5 to 16) single-precision floating-point 1-D FFT and 2^n (n is from range of 5 to 8) single-precision floating-point 2-D/3-D FFT. What is more, it is important that the structure has strong expansibility which can achieve FFT of the sequence of $m \times n \times p$. As a result the design's maximal frequency is up to 184.88 MHz. It has been successfully applied on the Xilinx Virtex6 FPGA chip.

Key words: FFT hardware accelerator; FFT processor; the module of address adjustment; FPGA

1 引言

FFT 在数字信号处理领域有着广泛应用,研究大点数 FFT 加速方法极有实用价值. 基于 FPGA 的 FFT 硬件实现是当前最有效的 FFT 加速方法,由于大点数/多维度 FFT 的计算量巨大,对处理器的运算能力和访存带宽要求非常高,所以 FFT 硬件加速器的设计需要综合考虑运算资源、本地存储规模和访存延迟等因素,提高运算访存比,达到资源与运算性能平衡.

FFT 加速器的设计通常围绕算法改进和结构完善两个方向展开,前人在这两个方面已有不少的研究成果.

文献[2]基于 Xilinx Virtex6 FPGA 芯片设计了一维、二维混合架构 FFT 处理器,实现了 32 点到 16 k 的任意 2^n 点 32 位单精度浮点数 FFT 运算. 该设计虽有二维架构的设计但只能运用二维的架构进行大点数一维 FFT 的计算,并不能实现真正的二维 FFT 运算.

文献[3]设计了基于 FPGA 的级联流水结构的

二维大点数 FFT 处理器. 该 FFT 处理器把序列长度为 N 的大点数 FFT 运算分解为 2 级小点数行列 FFT 运算. 列变换通过基-4FFT 算法, 完成固定 1 024 点 FFT 运算; 行变换通过基-2FFT 算法实现 2-1 024 k 点 FFT 运算. 此处理器的列变换只能完成固定 1 024 点 FFT 运算, 因此增加了任意序列 FFT 运算的行列分解复杂度.

文献[4]提出用于 802. 11ac 的 FFT/IFFT 处理器设计, 它是一种 8 通道 128 点的 MRMDC 处理器结构, 主要特点是数据输入输出的顺序调整结构. 但是此设计只进行了 128 点的设计, 因此通用性和扩展性不佳.

上述文献均针对固定维数的 FFT 运算展开讨论, 在实际应用中, FFT 运算器需要面对不同维度的 FFT 任务, 在处理高维度任务时, 需人工完成数据分解与重新组织. 为解决上述问题, 提高 FFT 运算器的应用范围, 本文设计了一种变维度 FFT 运算器.

2 FFT 算法原理

对于规模为 N 的一维序列 $x(N)$ 的 FFT 定义^[5] 为:

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

式中, $k = 0, 1, \dots, N-1$, $W_N = \exp(-\frac{i2\pi}{N})$. 由式 (1) 可推广到高维的情况, 如三维 FFT 序列. 对于规模 $N = N_z \times N_y \times N_x$ 的三维序列 $x(N_z, N_y, N_x)$ 的傅里叶变换定义为:

$$X(k_z, k_y, k_x) = \sum_{n_z=0}^{N_z-1} \sum_{n_y=0}^{N_y-1} G W_{N_y}^{n_y k_y} W_{N_z}^{n_z k_z} \quad (2)$$

其中:

$$G = \sum_{n_x=0}^{N_x-1} x(n_z, n_y, n_x) W_{N_x}^{n_x k_x};$$

$$k_z = 0, 1, \dots, N_z, W_{N_z} = \exp(-\frac{i2\pi}{N_z});$$

$$k_y = 0, 1, \dots, N_y, W_{N_y} = \exp(-\frac{i2\pi}{N_y});$$

$$k_x = 0, 1, \dots, N_x, W_{N_x} = \exp(-\frac{i2\pi}{N_x});$$

由式(2)可知, 三维 FFT 中各维之间数据相互无关, 因此, 对于 $N = N_z \times N_y \times N_x$ 的三维 FFT 处理过程为, 先对 x, y, z 任意一个方向先作一维 FFT, 再对其余两个方向中的任意一个方向作一维 FFT, 最后对剩下的方向作一维 FFT.

3 变维度 FFT 硬件加速器的结构设计

3.1 FFT 加速器架构

本文为了适应 FFT 运算器对不同维度、不同点数 FFT 的计算任务, 选择与之对应的体-面-线的数据组织形式. 考虑到 FFT 算法原理的特点和性能要求, 从面和线 2 个层次展开并行计算, 面为基本存储单位, 线为基本计算单位. 此设计方案不仅提高了数据的复用性, 减少数据交互, 而且降低了控制的复杂度.

以高维度三维 FFT 的计算为例: 设 $\forall F \in \{XY \text{ 面}, XZ \text{ 面}, YZ \text{ 面}\}$,

第一步:

取 $F_1 \in XY$, 令 $\forall L \in \{X, Y\}$, $L_1 \in X, L_2 \in Y$. 计算所有 L_1, L_2 的 FFT, 如图 1(a) 所示, 可获得 X 方向与 Y 方向的 FFT 计算结果;

第二步:

取 $F_2 \in XZ$, 令 $L_3 \in Z$. 计算所有 L_3 的 FFT, 如图 1(b) 所示, 可获得 Z 方向的 FFT 计算结果.

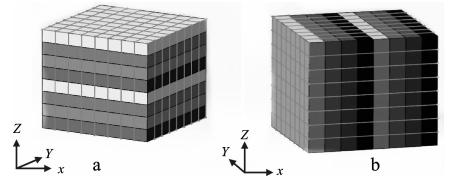


图 1 高维度 FFT 算法面划分并行算法

经过以上两个步骤, 即可完成整个三维 FFT 的运算.

本文设计的 FFT 加速器集成了 4 个独立的 FFT 处理器, 每个 FFT 处理器以面为基本存储单位, 依次独立完成同一个面在 2 个维度上的 FFT 运算, 整个加速器可以同时处理 4 个面的 FFT 运算. 同时, 每个处理器的私有存储单元支持多种寻址方式, 除了实现地址的无冲突操作以外, 还支持面的行列转置, 实现数据复用, 进一步提高计算访存比.

FFT 处理器模块是加速器的计算核心, 结构如图 2 所示. 包括控制器、数据传输单元、外存储控制单元、内存储控制单元、4 个 FFT 运算单元、存储单元的外存储为 1 组 4 个 $4k \times 64$ bit 双端口 RAM、内存储为 12 组 每组 4 个 32×64 bit 简单双端口 RAM.

3.2 FFT 运算单元

FFT 运算单元同时完成 4 条线的并行计算. 本

器的运算流程分别进行介绍.

当计算不高于 8 k 点时, 4 个 4 k 片上数据存储器进行乒乓操作, 通过控制器依次配置为操作数存储器组和结果存储器组, FFT 运算单元通过内存储控制单元和多路选择器同时访问操作数存储器组和结果存储器组, 每个时钟周期从操作数存储器组读取 4 个操作数进行 FFT 运算, 同时产生 4 个中间运算结果写入结果存储器组, FFT 运算每完成一级, 操作数存储器组和结果存储器组进行一次乒乓操作, 经过若干级乒乓操作完成整个 FFT 计算. 计算流程如图 5 所示, 运算时间流程如图 6 所示.

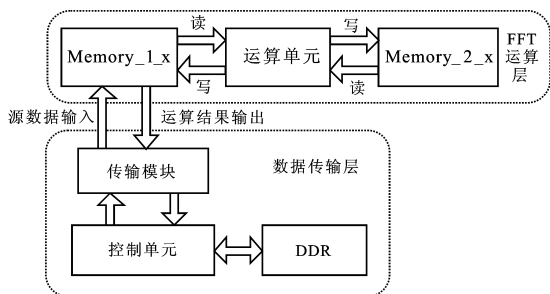


图 5 小点数一维 FFT 计算流程图

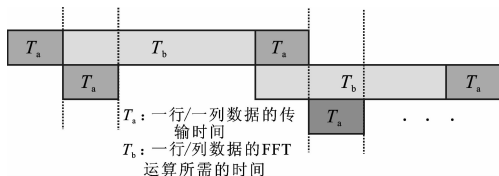


图 6 小点数一维 FFT 计算时间流程图

记一行/列的数据序列长度为 n , T_a 为一行/列的数据的搬运时间, T_b 为完成一行/列的 FFT 纯计

算时间, 则:

$$\begin{cases} T_a = \frac{n}{16} \\ T_b = \frac{n}{16} \times \log_2 n + 12 \end{cases} \quad (3)$$

结合公式(3)可算得一个 FFT 处理器完成所有的 FFT 运算的时间.

当计算大于 16 k 点一维 FFT 时, 16 k 外存储作为缓存, 每个 FFT 运算单元对应的 4×32 存储器进行内外乒乓操作, 通过控制器分别设置为操作数存储器组、结果存储器组及数据传输存储器组。数据传输单元通过外存储控制单元及多路选择器, 每个时钟周期从外存储器读取第一、二列/行的 8 个源数据, 每行/列取 4 个。再通过内存储控制单元写入数据传输存储器组, 完成第一、二列/行的数据传输。

第一列/行数据传输完成后,操作数存储器组、结果存储器组及数据传输存储器组进行一次外乒乓操作,将数据传输存储器组变为操作数存储器组,将操作数存储器组变为结果存储器组,将结果存储器组变为数据传输存储器组;数据传输单元读取第三、四列/行源数据,并写入数据传输存储器组,完成预读取操作.在数据传输的同时,每个 FFT 运算单元每个时钟周期从操作数存储器组读取 4 个操作数,产生 4 个结果数据写进结果存储器组.操作数存储器组与结果存储器组进行内乒乓操作完成第一、二列/行 FFT 运算.

控制器协调数据传输单元和 FFT 运算单元完成所有组 FFT 运算. 计算流程如图 7 所示, 运算时间流程如图 8 所示.

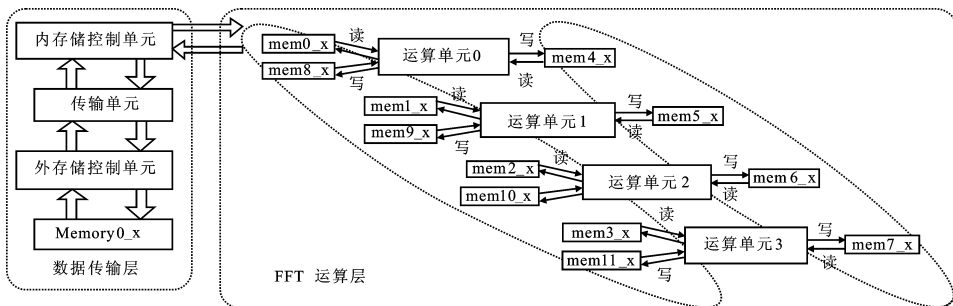


图 7 大点数一维/二维/三维 FFT 计算流程图

二/三维 FFT 的 FFT 处理器计算流程与大点数一维 FFT 计算流程相同.

设一行/列的数据序列长度为 n , T_a 为一个面内一行/列的数据的搬运时间, T_b 为完成一个面内一行/列的 FFT 纯计算时间, 则:

$$\begin{cases} T_a = \frac{n}{4} \\ T_b = \frac{n}{4} \times \log_2 n + 12 \end{cases} \quad (4)$$

结合公式(4)可算得一个 FFT 处理器完成一个

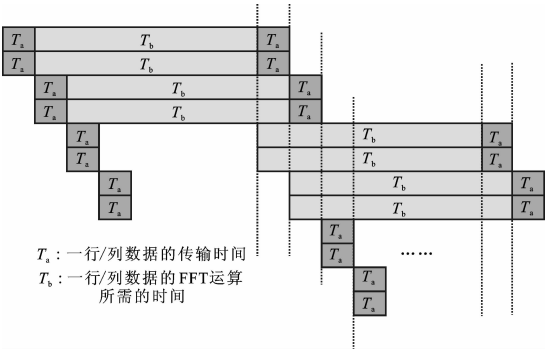


图 8 大点数一维/二维/三维 FFT 运算时间流程图

面的 FFT 运算所需的时间.

3.3 地址调整模块

地址调整模块是为了实现地址无冲突的操作、提高运算访存比而设计的模块. 针对不同的拓扑结构该模块的设计也会有所不同, 如文献[6-8]中采用的混合基的结构, 而本文采用的是基-2 的算法结构.

地址调整模块主要包括三个部分, 分别是源数据的写地址调整模块, 外存储的读地址调整模块, 内存存储的写地址调整模块.

(a)源数据的写地址调整模块

该地址调整模块指的是一个面的源数据写到外存储时对地址所做的调整. 外存储块由 4 个双端口 memory 组成, 一个周期可以对 8 个数据进行读写的操作. 外存储块在一个周期内写入 8 个源数据, 在与内存存储模块间数据交互时一个周期内也完成 8 个数据的传输. 该地址无冲突的模块设计伪代码如表 2 所示.

```
mode_w=sign_w(m,n,p);//sign_w is the function of loop.  
addr_w=group_switch(mode_w,addr_in);  
//group_switch is the function of select.
```

由此可知, 组间交叉开关和组内交叉开关根据地址标志位来判断存储位置. 该地址模块控制简单, 硬件实现十分方便.

(b)外存储的读地址调整模块

该地址调整模块用于内外存储块间数据传输. 为了提高运算访存比, 最有效的隐藏数据传输时间, 内存存储块采用预读取操作. FFT 运算单元一个周期从外存储块中无冲突的请求 8 个数据, 该无冲突的地址模块设计伪代码如下所示.

```
mode_r=sign_r(m,n,p);//sign_r is the function of loop.  
addr_r=group_switch_r(mode_r,addr_in);
```

//group_switch_r is the function of select.

如上所示, 每次读取 8 个数据, 结合 3.2 节中时间流程图可知, 每次并行取 2 条线的数据传输给内存存储单元. 组内交叉开关根据有效信号和地址标志位进行判断, 从而正确的进行数据传输.

(c)内存存储的写地址调整模块

内外存储块间数据传输时, 该模块用于内存存储块的写地址调整. 这不仅有利于提高运算访存比, 最有效的将数据传输时间隐藏, 更重要的是方便灵活的进行 FFT 运算过程中原位读写操作. 该调整模块的地址规律如表 1 所示, 基于 FFT 计算的倒序操作, 可得到存储在内存存储单元内的数据存储架构表, 如表 2 所示.

表 1 内存存储模块写地址

mem	存储地址(统一地址)							
m_0	0	4	...	$n/2+3$	$n/2+7$	...	$n-5$	$n-1$
m_1	1	5	...	$n/2+2$	$n/2+6$	...	$n-6$	$n-2$
m_2	2	6	...	$n/2+1$	$n/2+5$	...	$n-7$	$n-3$
m_3	3	7	...	$n/2$	$n/2+4$	...	$n-8$	$n-4$
地址	0	1	...	$n/8$	$n/8+1$	...	$n/4-2$	$n/4-1$

表 2 内存存储数据存储架构

mem	数据标号							
m_0	0	$n/8$	...	$3n/4+1$	$7n/8+1$	...	$n-1$	
m_1	$n/2$	$5n/8$	...	$n/4+1$	$3n/8+1$	...	$n-2$	
m_2	$n/4$	$3n/8$	...	$n/2+1$	$5n/8+1$	...	$n-3$	
m_3	$3n/4$	$7n/8$	...	1	$n/8+1$	...	$n-4$	
地址	0	1	...	$n/8$	$n/8+1$	...	$n/4-1$	

综上所述, 3 组地址调整模块遵循的是同一套地址规律. 在架构设计上只要根据地址标志位便可以简单的控制数据交换开关的工作, 控制简单, 有利于硬件实现.

4 FFT 加速器性能评估

4.1 资源消耗

大点数/多维度 FFT 大容量数据流的实时处理受到运算资源、存储带宽和访存延时等多种因素的制约, 提高访存比、平衡资源和运算性能是相关设计的难点与着眼点. 本文采用了可扩展设计, 平衡了资源和运算速度, 达到了较高的性价比. 整个设计在 Xilinx XC6VLX760-FF1760 FPGA 上硬件实现, 资源消耗如表 3 所示, 最高工作频率 184.88 MHz.

表 3 资源消耗		
Logic Utilization	Used/Available	Utilization/%
Slice Register	30 538/948 480	3
Slice LUTs	286 150/474 240	60
Block RAM	152/720	21
DSP48Es	640/864	74

4.2 实验性能评估

本文设计的变维度 FFT 处理器运算模式采用预读取操作隐藏蝶形单元的流水线延时和访存延时,在 Xilinx XC6VLX760-FF1760 FPGA 上硬件实现,实际测试得单路内的时间流如图 6 和图 8 所示.文献[9]中一维 FFT 加速器的设计、文献[10]中二维 FFT 加速器的设计、文献[11]中三维 FFT 加速器的设计与本文的计算方案相同、运算结构和加速方式相似,具有较强的参考性,故选作参照设计.

4.2.1 一维 FFT 运算的性能

根据文献[9]给出的信息,将本文设计的加速器实现的 128 点到 2 048 点序列一维 FFT 运算所花的周期与其对比,对比结果表明,本文设计的 FFT 处理器在进行一维 FFT 计算时有理想的处理速度.随着点数的增大,本文设计的 FFT 加速器的加速比系数越接近于 1,即接近理论值.所以当处理大点数 FFT 运算时,本文设计的 FFT 加速器具有明显优势.

4.2.2 二维 FFT 运算的性能

根据文献[10]中提及的对于二维 FFT 加速器的设计同本设计的分解方法相似,只是在存储方面,本文以面为基本存储单位减少了存储器之间的数据交互,而文献[10]的设计中 FFT 每级蝶形计算需要同步并且通过共享存储单元进行数据交互.两者的运算时间对比可知,在进行二维 FFT 运算时,文献[10]中提及的 3 种运算方式所用的周期数都高于本设计的周期数.本设计的加速比明显高于文献[10]设计的加速比,且随着点数的增大,加速比趋于稳定.综上可知,本文的设计有更好的运算性能,且优势明显.

4.2.3 三维 FFT 运算性能

进行三维 FFT 计算时,文献[11]以线为单位进行传输兼计算,而本文的设计以面为单位进行传输、以线为单位进行计算.两者在数据交互方式上不同,因此在运算访存比会有不同.本文的设计在运算周期上有一定的优势,但相比于一维/二维 FFT 计算时的优势是有所下降的.是两者加速比对比图可知,本文在 256 点以下的三维 FFT 运算时有明显的优势,但是随着点数的增加本文设计的加速比趋于稳定,而文

献[11]仍呈上升趋势.综上对比结果可知,本文的设计在 256 点以下 FFT 计算时有一定优势,而在大点数三维 FFT 计算时文献[11]的设计更有优势.

综合以上三组数据的对比,可知,本文的设计在一维/二维 FFT 的计算时具有一定优势.本文的设计在 256 点以下 FFT 计算时有一定优势,而在大点数三维 FFT 计算时文献[11]的设计更有优势.

5 结束语

变维度 FFT 从高吞吐、强实时性的应用角度出发,研究了一维、二维、三维 FFT 算法原理,并分析影响 FFT 硬件加速器性能的各种因素,设计出了一种变维度 FFT 硬件加速器.通过体-面-线的数据组织形式,以面为基本传输单位,以线为基本计算单位,设计了面划分兼多路并行的架构,提高了 FFT 运算的并行度.同时,通过 FFT 运算单元内 3 组存储器的乒乓及预读取操作,提高了运算访存比.最终在 Xilinx Virtex6 FPGA 芯片上,实现了 IEEE-754 标准下的 32 位单精度浮点数 32 点到 64 k 点一维 FFT 运算,32 点到 256 点的二维和三维 FFT 运算.此外,本设计方案具有较强的可扩展性,可根据需要实现 $m \times n \times p$ 序列的 FFT 运算.

参考文献:

[1] 方维,孙广中,吴超,等.一种三维快速傅里叶变换并行算法[J].计算机研究与发展,2011,48(3):440-446.

[2] Duoli Zhang, Lu Huang, Yukun Song, et al. Design and implemenation of 1D and 2D mixed architection fft processor in heterogeneous multi-core SoC based on FPGA [J]. International Jourual of Control and Automation, 2014, 7(6): 177-188.

[3] 王晓君,龙腾,周希元.二维级联流水结构大点数 FFT 运算器实现研究[J].无线电工程,2010,40(11): 19-22.

[4] 施隆照,郭冀闽.基于 802.11ac 的 FFT/IFFT 处理器设计[J].微电子学与计算机,2016,33(3):94-97.

[5] 郑君里.信号与系统[M].2 版.北京:高等教育出版社,2000: 135-146.

[6] Akira Nukada, SatoshiMatsuoka. Auto-tuning 3D FFT library for CUDA GPUs[C]// Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis. Portland, Oregon, 2009.

[7] Shousheng He, Mats Torkelson. A new approach to pipeline FFT processor[D]. Lund University, Lund, SWEDEN,1996.

acher. A CMOS chopper amplifier [J]. IEEE JSSC, 1987, SC-22(3): 335-342.

[4] Christian C Enz, Gabor C Temes. Circuit techniques for reducing the effects of op-amp imperfections: auto-zeroing, correlated double sampling, and chopper stabilization [J]. IEEE JSSC, 1981, SC-16(6): 1584-1614.

[5] Nick Van Helleputte, Mario Konijnenburg, Julia Petrine, A 345 μ W multi-sensor biomedical soc with bio-impedance, 3-channel ecg, motion artifact reduction, and integrated DSP [J]. IEEE JSSC, 2015, 50(1): 230-244.

[6] Qinwen Fan, Fabio Sebastiano, Johan H. Huijsing, et al. A 1.8 μ W 60 nV/Hz capacitively-coupled chopper instrumentation amplifier in 65 nm CMOS for wireless sensornodes[J]. IEEE JSSC, 2011, 46(7): 1534-1543.

[7] Yazicioglu R F , Torfs T, Merken P, et al. Texas instruments ADS1292R, low-power, 8Channel, 24Bit

AFE for biopotential measurements [J]. Microelectronics Journal, 2009, 40(9): 1313-1321.

作者简介:

陈 就 男, (1991-), 硕士研究生. 研究方向为低功耗、高精度模拟集成电路设计. E-mail: chenjiu@ime. ac. cn.

王小松 男, (1981-), 博士, 副研究员. 研究方向为高性能混合信号集成电路、生物电传感技术、物联网相关技术.

张海英 女, (1964-), 博士, 研究员. 研究方向为 3G/4G 射频技术、嵌入式多模、医疗电子系统集成技术、物联网相关技术.

刘 昱 男, (1975-), 博士, 研究员. 研究方向为高性能模拟/射频 CMOS 集成电路、硅基毫米波集成电路、超低功耗短距离无线通讯系统、物联网相关技术及医疗电子系统集成技术.

(上接第 39 页)

[8] Ainhoa Cortés, IgoneVélez, Juan F Sevillano. Radix $r^{[k]}$ FFTs: matricial representation and sdc/sdf pipeline implementation [J]. IEEE Transactions on Signal Processing, 2009, 57(7): 2824-2839.

[9] 张竺君. 基于 FPGA 的可变点 FFT 处理器的设计与实现[D]. 南京: 南京理工大学, 2009.

[10] 张全, 鲍华, 饶长辉, 等. GPU 平台二维快速傅里叶变换算法实现及应用[J]. 光电工程, 2016, 43(2): 69-75.

[11] Sharat B, Chandra Varma, Kolin Paul, et al. Accelerating 3D-FFT using hard embedded blocks in FPGAs [C]//2013 26th International Conference on VLSI De-

sign and the 12th International Conference on Embedded Systems. [s. l.], 2013.

作者简介:

张多利 男, (1976-), 博士, 研究员. 研究方向为多核处理器体系结构与设计方法、多媒体专用集成电路设计实现等.

张玲佳 女, (1992-), 硕士研究生. 研究方向为 SoC 设计.

宋宇鲲(通讯作者) 男, (1975-), 博士, 副研究员. 研究方向为面向数据密集与计算机密集应用的 SoC/MPSoC 体系结构与实现. E-mail: songyukun@hfut. edu. cn.