文章编号:2095-6134(2015)02-0259-05

基于 FPGA 的高速浮点 FFT/IFFT 处理器设计与实现*

苏 斌[1,2†], 刘 畅[1], 潘志刚[1]

(1 中国科学院电子学研究所, 北京 100190; 2 中国科学院大学, 北京 100049)

(2014 年 3 月 6 日收稿; 2014 年 5 月 12 日收修改稿)

Su B, Liu C, Pan Z G. Design and implementation of high-speed floating points FFT processor based on FPGA[J]. Journal of University of Chinese Academy of Sciences, 2015,32(2):259-263.

摘 要 设计一种基于 FPGA 的改进的并行 FFT/IFFT 蝶形运算结构. 该结构采用按时间抽选的 FFT 基-2 蝶形算法对 IEEE 单精度浮点数构成的复数进行 8 路并行处理. 利用 Xilinx ISE 13.1 软件完成 FFT/IFFT 处理器的设计,并在 Virtex6 硬件平台上进行验证. 结果表明,利用这种 8 路并行结构设计的 FFT/IFFT 处理器可在合理利用硬件资源的同时提高运算速度及精度.

关键词 FPGA; 单精度浮点; FFT/IFFT; 基-2 蝶形算法; 并行结构

中图分类号:TN911 **文献标志码**:A **doi**:10.7523/j.issn.2095-6134.2015.02.016

Design and implementation of high-speed floating points FFT processor based on FPGA

SU Bin[1,2], LIU Chang[1], PAN Zhigang[1]

(1 *Institute of Electronics, Chinese Academy of Sciences, Beijing* 100190, *China*;
2 *University of Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract An improved parallel FFT/IFFT butterfly structure is presented. The structure, which is based on the decimation-in-time (DIT) Radix-2 algorithm, contains 8 paths in parallel. IEEE single precision floating complex FFT/IFFT processors are designed by using this parallel structure and achieved with Xilinx ISE 13.1 software. Verifications are carried out with Virtex6 FPGA of Xilinx. The results show that the FFT/IFFT processors can make use of the resources reasonably and enhance both the speed and precision of computation.

Key words FPGA; single precision floating; FFT/IFFT; Radix-2 algorithm; parallel structure

快速傅里叶变换(fast Fourier transformation, FFT)处理器是实时信号处理系统中最重要的运算单元之一,被广泛应用于雷达信号处理领域. 随着现代军工技术的发展,对雷达信号处理的实时性以及数据处理精度的要求越来越高. 目前,应用于雷达的信号处理器多采用数字信号处理器(digital signal processor, DSP)芯片. 但是 DSP 芯片内部结构固定,专用电路通过 RAM 内部指令集和数据进行工作,受指令周期与时钟的约束,同时 DSP 的运算资源固化,使得 DSP 实现效率和灵

* 国家自然科学基金(61100201)资助

† 通信作者,E-mail:subin_iecas@163.com

活性受到制约[1]. 现场可编程门阵列(field-programmable gate array, FPGA)直接针对硬件进行设计,具有实时性好、灵活性强、支持在线可编程、可充分进行设计开发和验证的优势[2]. 因此基于FPGA实现FFT/IFFT处理器,可有效地提高运算速度[2]. 同时,Xilinx公司的Virtex6芯片提供了浮点运算单元,可实现高精度的数据运算.

通常基于FPGA的FFT运算结构[3]分为递归结构、流水线结构以及全并行结构. 在数据的存取及运算方面,递归结构所占用的硬件资源最少,但只有一个运算单元,运算时间较长. 流水线结构是将本级运算结果直接用于下一级运算,运算速度有所提高,但需要消耗较大的存储空间. 全并行结构的运算单元数量与参与运算的点数成正比,运算速度最快,但需要耗费大量硬件资源.

本文设计了8路并行FFT/IFFT运算结构,利用该结构实现的FFT/IFFT处理器可稳定工作在200 MHz时钟下,计算误差小,合理利用FPGA芯片内部资源并提高了FFT/IFFT的运算速度.

1 时间抽选法基-2 FFT/IFFT

FFT算法主要分为2类:时间抽选法(decimation in time, DIT)和频率抽选法(decimation in frequency, DIF). 按时间抽选的FFT算法主要有基-2 FFT算法、基-4 FFT算法、分裂基FFT算法. 分裂基算法比较复杂,不易在高速电路中实现[4]. 相比于基-2 FFT算法,基-4算法无法进行2^n(n为奇数)点FFT运算,同时基-2 FFT算法的蝶形运算结构较为简单,降低了抽取地址生成的难度. 因此本文设计的FFT/IFFT运算结构选用基-2 FFT算法.

基-2 FFT迭代运算表达式为:

$$X_L(J) = X_{L-1}(J) + X_{L-1}(J+B)W_N^P, \quad (1)$$

$$X_L(J+B) = X_{L-1}(J) - X_{L-1}(J+B)W_N^P, \quad (2)$$

其中,L为蝶形运算级,$L = 1,2,\cdots,M$;J为蝶形组,$J = 0,1,2,\cdots,2^{L-1}-1$;$B$为两个数据间距,$B = 2^{L-1}$;$P$为旋转因子指数,$P = J \times 2^{M-L}$.

IFFT可以通过FFT实现. 从IDFT公式

$$x(n) = \mathrm{IDFT}[X(k)] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-nk} \quad (3)$$

与DFT公式

$$X(k) = \mathrm{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad (4)$$

的比较中可以看出,需将FFT旋转因子W_N^{nk}改为W_N^{-nk},之后乘以常数$1/N$便可实现IFFT[4].

将$X(k)$的实部与虚部互换得到$X_1(k)$,表达式为

$$X_1(k) = \mathrm{Im}(X(k)) + j\mathrm{Re}(X(k)). \quad (5)$$

对$X_1(k)$进行傅里叶变换可得$x_1(n)$,即

$$\begin{aligned} x_1(n) &= \mathrm{DFT}[X_1(k)] \\ &= \sum_{k=0}^{N-1}(\mathrm{Im}(X(k)) + j\mathrm{Re}(X(k)))W_N^{nk} \\ &= \sum_{k=0}^{N-1}(\mathrm{Im}(X(k)) + j\mathrm{Re}(X(k)))(\mathrm{Re}(W_N^{nk}) + j\mathrm{Im}(W_N^{nk})) \\ &= \sum_{k=0}^{N-1}(\mathrm{Im}(X(k))\mathrm{Re}(W_N^{nk}) - \mathrm{Re}(X(k))\mathrm{Im}(W_N^{nk})) + \\ &\quad j(\mathrm{Re}(X(k))\mathrm{Re}(W_N^{nk}) + \mathrm{Im}(X(k))\mathrm{Im}(W_N^{nk})). \end{aligned} \quad (6)$$

将$x_1(n)$的实部与虚部交换后除以N可得

$$\begin{aligned} &\frac{1}{N}\sum_{k=0}^{N-1} j(\mathrm{Im}(X(k))\mathrm{Re}(W_N^{nk}) - \mathrm{Re}(X(k))\mathrm{Im}(W_N^{nk})) + \\ &(\mathrm{Re}(X(k))\mathrm{Re}(W_N^{nk}) + \mathrm{Im}(X(k))\mathrm{Im}(W_N^{nk})) \\ &= \frac{1}{N}\sum_{k=0}^{N-1}(\mathrm{Re}(X(k)) + j\mathrm{Im}(X(k)))(\mathrm{Re}(W_N^{nk}) - j\mathrm{Im}(W_N^{nk})) \\ &= \frac{1}{N}\sum_{k=0}^{N-1}(\mathrm{Re}(X(k)) + j\mathrm{Im}(X(k)))(\mathrm{Re}(W_N^{-nk}) + j\mathrm{Im}(W_N^{-nk})) \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-nk} = x(n). \end{aligned} \quad (7)$$

根据式(6)、式(7),IFFT模块设计思想为:先将输入数据实部与虚部交换,经过FFT模块处理后将运算结果实部与虚部交换并乘以$1/N$.

2 FFT/IFFT处理器的硬件实现

2.1 FFT/IFFT处理器整体结构

FFT/IFFT处理器整体结构如图1所示,一路32位浮点数据通过I/O接口串行输入,每个复数据分为实部与虚部,数据的实部与虚部交替输入. 在地址产生单元的控制下根据FFT/IFFT选择控制信号将输入的数据存放于16个RAM中,其中8个存放实部,另外8个存放虚部. 在地址产生单元的控制下,蝶形运算单元分别到数据RAM与旋转因子RAM的相应位置取数,进行蝶形运算,并将运算结果存回RAM.

2.2 并行地址设计

为实现8路并行运算,每次从16个RAM中读取8个复数据,分为4组参与蝶形运算,参与蝶形运算的2个数需存放在不同RAM中. 设数据的

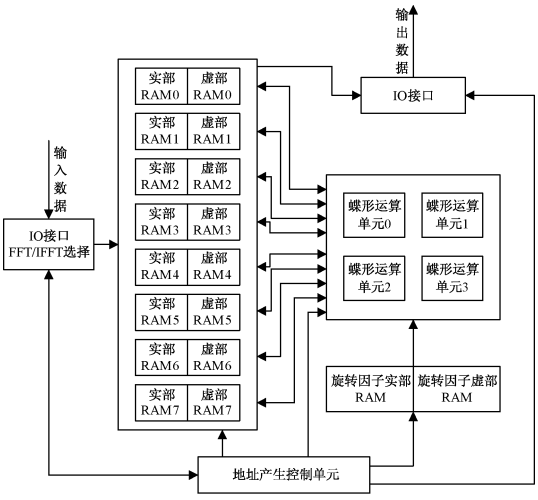

图 1　8 路并行 FFT/IFFT 系统框图

Fig. 1　Layout of 8-path parallel FFT/IFFT

序列号为 $a_n a_{n-1}\cdots a_2 a_1 a_0$, RAM 标号地址为 $b_2 b_1 b_0$,每块 RAM 中数据地址为 $x_{n-3}x_{n-4}\cdots x_1 x_0$. 数据存放的设计思想为:将数据按先后顺序均等分为4 部分,第 1 部分存放在 RAM0、RAM1 中,第 2 部分存放在 RAM2、RAM3 中,第 3 部分存放在 RAM4、RAM5 中,第 4 部分存放在 RAM6、RAM7 中,因此 RAM 标号地址高 2 位由数据序列号的高 2 位确定,即

$$b_2 = a_n, \tag{8}$$

$$b_1 = a_{n-1}. \tag{9}$$

由式(1)、式(2)可知参与运算的 2 个数的序号间隔为 2 的幂次,以二进制表示的 2 个数的序号仅有一位不同,可通过逐位异或进行分区. 因此,RAM 标号地址的最后一位由数据序列号的逐位异或得到,即

$$b_0 = a_n \wedge a_{n-1} \wedge a_{n-2} \wedge \cdots \wedge a_0. \tag{10}$$

每块 RAM 中数据地址由数据序列号的 a_{n-2} 到 a_1 位确定,即

$$x_{n-3}x_{n-4}\cdots x_2 x_1 x_0 = a_{n-2}a_{n-3}\cdots a_3 a_2 a_1. \tag{11}$$

本方案中先将输入数据的序号进行倒位序排列,以 32 点复数 FFT 为例,由式(8)－式(11)可以确定 RAM0—RAM7 中各位置存储的数据,其蝶形运算结构如图 2 所示

4 个蝶形运算单元同时从 16 个 RAM 中读取数据进行运算. 设参与运算的点数为 N,从第 1 级至倒数第 3 级蝶形运算,4 个单元在每一级的起始取数序号为 0,$N/4$,$N/2$,$3N/4$. 在倒数第 2 级,4 个单元的起始取数序号为 0,$N/8$,$N/2$,$5N/8$. 最后一级 4 个单元的起始取数序号为 0,$N/8$,$N/4$,$3N/8$. 由式(1)、式(2)可以确定 4 个蝶形运算单元每次运算的取数序号,根据序号到相应的 RAM 中读取数据.

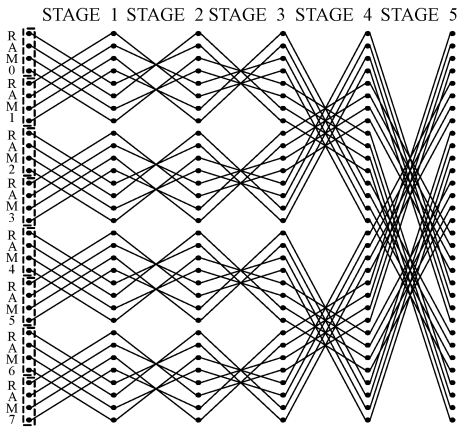

图 2　蝶形运算结构示意图

Fig. 2　Structure of butterfly computation

2.3　旋转因子存储器

N 点基－2 FFT 蝶形运算需要 $N/2$ 个旋转因子,每个旋转因子包括 2 个系数,即实部与虚部. 将旋转因子系数预置于 RAM,由于前面几级运算所需要的旋转因子都可以在最后一级参与运算的旋转因子中找到,因此仅需存储一份,按照旋转角度 $P = 0,1,2,\cdots,N/2-1$ 依次存储,数据存储形式为 32 位单精度浮点数,通过查找表方法得到蝶形运算过程中所需要的旋转因子[5]. 由于 4 个蝶形运算单元在同一时刻运算时所需的旋转因子不同,在存储空间足够大的情况下,可以为每个蝶形单元分配 1 个旋转因子存储 RAM. 为了节省片内存储空间,本文在 FFT/IFFT 模块中预存 2 份旋转因子,即每 2 个蝶形运算单元共用 1 个旋转因子存储 RAM.

2.4　浮点数蝶形运算模块设计

定点运算计算速度快,功耗低,但是运算误差大,浮点运算计算速度慢,功耗大,但运算精度高. 工程实现中,由于定点运算误差的存在,会造成重要信息的丢失,因此本文中的复数采用 IEEE 745 单精度浮点格式表示. Xilinx 公司提供了支持浮点数运算的 IP 核 Floating-Point[6],该 IP 核支持 IEEE 745 浮点数的加法、减法、乘法及除法运算.

在蝶形运算单元中包含浮点数乘法器、浮点数加法器和浮点数减法器 3 部分,结构框图如图 3 所示. 图中信号表示为:

1)输入数据的实部 $A_r = \mathrm{Re}(A), B_r = \mathrm{Re}(B)$.

2)输入数据的虚部 $A_i = \mathrm{Im}(A), B_i = \mathrm{Im}(B)$.

3)旋转因子的实部与虚部 $W_r = \mathrm{Re}(W)$, $W_i = \mathrm{Im}(W)$.

4)输出数据的实部 $X0_r = \mathrm{Re}(A + BW)$, $X1_r = \mathrm{Re}(A - BW)$.

5)输出数据的虚部 $X0_i = \mathrm{Im}(A + BW)$, $X1_i = \mathrm{Im}(A - BW)$.

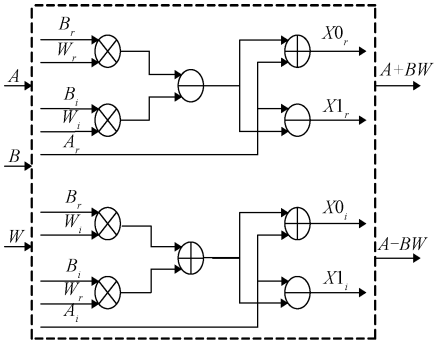

图 3 基 2 蝶形运算单元系统框图

Fig. 3 Schematic of Radix – 2 butterfly computation unit

3 FPGA 实现结果

使用 Xilinx 公司 Virtex6 XC6VLX240T 芯片对本文中的设计进行仿真，以 16 K 点 FFT/IFFT 运算为例，经过布线后的 FPGA 资源消耗如表 1 所示. 该模块最高工作频率可达 350 MHz.

完成程序仿真后，将程序加载到 FPGA 中，全局时钟频率为 200 MHz，输入 2 个含有高斯白噪声的复数正弦信号进行 FFT 运算，通过 Chipscope 软件采集 FPGA 处理后的实际数据，并将数据导入到 Matlab 中与 Matlab 仿真数据进行比较. 图 4 是 16 K 点 FFT 的 Matlab 仿真结果与 FPGA 实际输出结果. 图 5 为 Matlab 的仿真结果与 FPGA 运算结果的幅度误差值，误差的产生是由于 RAM 中的旋转因子为近似值，精度精确到小数点后 6 位. 由图 4 可知 FPGA 硬件运行结果与仿真基本一致，由图 5 可知 FPGA 芯片输出的实际值与 Matlab 仿真理想值的实、虚部误差可控制在小数点后 5 位，从而验证了此 FFT/IFFT 方案的正确性.

表 1 Virtex6 XC6VLX240T 实现 FFT/IFFT 资源消耗结果

Table 1 Resource consumption of FFT/IFFT based on Virtex6 XC6VLX240T

FPGA 资源	资源消耗	总数	利用率/%
Slice Registers	11 981	301 440	3
Slice LUTs	21 037	150 720	13
Bonded IOBs	72	600	12
Block RAM/FIFO	96	416	23

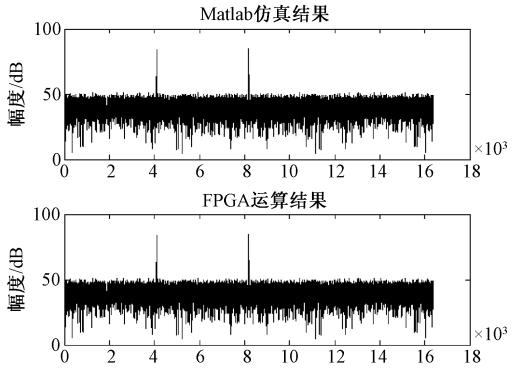

图 4 Matlab 理想值与 FPGA 运算结果对比

Fig. 4 Comparison between Matlab and FPGA results

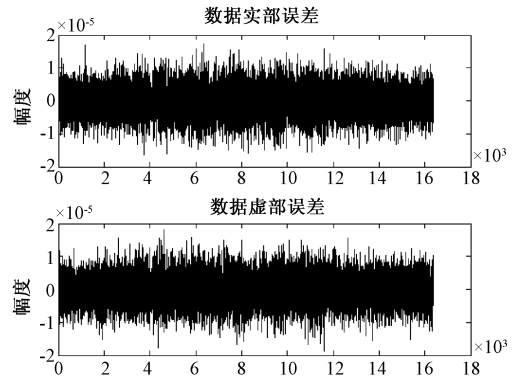

图 5 Matlab 理想值与 FPGA 运算结果幅度误差

Fig. 5 Amplitude errors between Matlab and FPGA results

表 2 是本文与其他 FFT 处理器运算性能的比较. 表 3 列出了与本文使用同类 FPGA 器件实现的 FFT 处理器硬件资源消耗情况. 从表 2 可以看出本文设计的 FFT/IFFT 处理器的运算精度与速度均高于文献[7-13]. 从表 3 看出，与文献相比，本文设计的 FFT/IFFT 处理器的硬件资源消耗量并未有过多增长.

表 2　FFT 运算性能对比

Table 2　Calculation performance comparison of FFT

文献	FFT 点数	精度	时钟频率/MHz	处理时间/μs	芯片类型
[7]	256	24 位定点	200	1.49	Xilinx XC6VLX240T
[8]	512	8 位定点	100	36.9	Altera EP2C8Q208C8N
[9]	1 024	16 位定点	100	26	XilinxVirtex－4 LX25
[10]	1 024	32 位定点	100	62.95	Altera EP2C35F484C8
[11]	1 024	32 位定点	100	62.96	Altera EP2C35F484C8
[12]	1 024	32 位浮点	200	750	Altera EP2C8Q208
[13]	4 096	28 位定点	100	82.7	Xilinx XC6VLX240T
本文	256	32 位浮点	200	1.387	Xilinx XC6VLX240T
本文	512	32 位浮点	200	2.987	Xilinx XC6VLX240T
本文	1 024	32 位浮点	200	6.507	Xilinx XC6VLX240T
本文	4 096	32 位浮点	200	30.827	Xilinx XC6VLX240T
本文	16 384	32 位浮点	200	143.467	Xilinx XC6VLX240T

表 3　硬件资源消耗对比

Table 3　Performance comparison on FPGA devices

文献	Slice Registers	Slice LUTs	Block RAM/FIFO	DSP48s	芯片类型
[7]256 点	5 264	6 043	8	24	Xilinx XC6VLX240T
[9]1 024 点	2 472	10 841	N/A	10	Xilinx VIRTEX4 LX25
[13]4 096 点	10 128	9 824	17	40	Xilinx XC6VLX240T
本文 4 096 点	9 768	14 036	32	48	Xilinx XC6VLX240T
本文 16 384 点	11 981	21 037	96	48	Xilinx XC6VLX240T

4　结束语

在 FFT/IFFT 处理器设计过程中,数据处理精度,硬件资源开销及运算速度是衡量处理器性能的关键指标. 本文介绍的 8 路并行运算结构在合理利用硬件资源的同时提高了运算速度,32 位浮点数处理模块保证了高精度的数据处理,通过与 Matlab 仿真值相比较,验证了实际输出值的正确性. 由于 Virtex6 XC6VLX240T 芯片实现的 FFT/IFFT 处理器的硬件资源利用率低,该处理器在数据处理速度上仍有提升空间,通过增加并行处理路数可进一步提高运算速度.

参考文献

[1] Patti A, Sezan M, Tekalp A. Super-resolution video reconstruction with arbitrary sampling lattices and nonzero aperture time[J]. IEEE Trans on Image Processing, 1997, 6(8):1 064-1 076.

[2] Sanchez M A, Garrido M, Lopez-Vallejo M, et al. Implementing FFT based digital channelized receivers on FPGA platforms [J]. Aerospace and Electronic Systems, 2008, 44(4):1 567-1 585.

[3] 李伟, 孙进平, 王俊, 等. 一种基于 FPGA 的超高速 32K 点 FFT 处理器[J]. 北京航空航天大学学报, 2007, 33(12): 1 440-1 443.

[4] 荣瑜, 朱恩. 一种高性能 FFT 蝶形运算单元的设计[J]. 东南大学学报:自然科学版, 2007,37(4):565-568.

[5] 杨靓, 黄巾, 刘红侠, 等. 一种高效的 FFT 处理器地址快速生成方法[J]. 信号处理, 2004(3): 251-257.

[6] Xilinx Company. LogiCORE IP Floating-Point Operator v5.0 [EB/OL]. America: Xilinx Company, (2011)[2014－02－20]. http://www.xilinx.com/products.

[7] Yang C, Chen H. A efficient design of a real-time FFT architecture based on FPGA[C]// Radar Conference, IET International. 2013:1-5.

[8] 栗旭光, 何国经, 刘江涛. 基于 FPGA IP 核的 FFT 实现与改进[J]. 电子科技,2013,26(12):96-100.

[9] Derafshi Z H, Frounchi J, Taghipour H. A high speed FPGA implementation of a 1024-point complex FFT processor[C]// Computer and Network Technology, 2010 Second International Conference on. 2010:312-315.

[10] Zhou S, Wang X C, Ji J J, et al. Design and implementation of a 1024-point high-speed FFT processor based on the FPGA [C]// Image and Signal Processing, 2013 6th International Congress on. 2013:1 112-1 116.

[11] 王文芳, 周盛, 王晓春, 等. 1024 点高速 FFT 处理器的 FPGA 设计与实现[J]. 国际生物医学工程杂志, 2011, 34(4):205-208.

[12] 刘健, 史彩娟, 赵丽莉. 基于 FPGA 的高速浮点 FFT 的实现研究[J]. 微型机与应用, 2012,31(14):79-84.

[13] Sun X, Qiu D L, Chen H, et al. An implementation of FFT processor[C]// Radar Conference 2013, IET International. 2013:1-4.