基于 FPGA的高速流水线 FFT算法实现

樊光辉,许 茹,王德清
(厦门大学通信工程系水声通信与海洋信息技术教育部重点实验室,福建省厦门市 361005)

摘 要:提出了在 FPGA(现场可编程门阵列)上实现 1 024点基 4-FFT(快速傅里叶变换)算法的设计方案。方案对 FFT算法的核心单元即蝶形运算单元的结构进行了分析和优化,用一个复乘器通过时序控制实现了和 3个复乘器同样的效率,而且对整个算法的流程采用了流水线式的工作控制方式,不仅节省了 FFT在 FPGA上实现时占用的硬件资源,并且极大地提高了算法的运算效率。最后给出了仿真实验结果,并同 MATLAB的 FFT运算结果进行了对比。结果显示,在 100 MHz时钟条件下,本方案完成 1 024点的基 4-FFT运算仅需 51.28 μs,完全满足高速 FFT运算的实时性要求。

关键词: FFT;基 4蝶形运算;流水线; FPGA
中图分类号: TP301.6

0 引 言

有限长序列的 DFT(离散傅里叶变换)特点是能够将频域的数据离散化成有限长的序列。但由于 DFT本身运算量相当大,限制了它的实际应用。FFT(快速傅里叶变换)算法是作为 DFT的快速算法提出[1],它将长序列的 DFT分解为短序列的 DFT,大大减少了运算量,使得 DFT算法在频谱分析、滤波器设计等领域得到了广泛的应用。

FPGA(现场可编程门阵列)是一种具有大规模可编程门阵列的器件,不仅具有专用集成电路(ASIC)快速的特点,更具有很好的系统实现的灵活性。FPGA可通过开发工具实现在线编程。与 CPLD(复杂可编程逻辑器件)相比,FPGA属寄存器丰富型结构,更加适合于完成时序逻辑控制。因此,FPGA为高速 FFT算法的实现提供了一个很好的平台。

1 基 4-FFT算法基本原理

在 FFT各类算法中,基 2-FFT算法是最简单的一种,但其运算量与基 4-FFT算法相比则大得多,分裂基算法综合了基 4和基 2算法的特点,虽然具有最少的复乘运算量,但其 L 蝶形运算控制的复杂性也限制了其在硬件上的实现[2],因此,本设计采用了基 4-FFT算法结构。

基 4-FFT算法的基本运算是 4点 DFT。一个 4点的 DFT运算的表达式为:

$$\begin{bmatrix} X'(0) \\ X'(2) \\ X'(1) \\ X'(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} X(0)W_N^0 \\ X(1)W_N^{K_0} \\ X(2)W_N^{2K_0} \\ X(3)W_N^{3K_0} \end{bmatrix} \quad (1)$$

式(1)对于输出变量进行了二进制倒序,便于在运算过程中进行同址运算[1],节省了运算过程中所需存储器单元的数量。

按 DIT(时间抽取)的 1 024点的基 4-FFT共需 5级蝶形运算,每级从 RAM中读取的数据经过蝶形运算后原址存入存储单元准备下一级运算。算法的第 1级为一组 N = 1 024点的基 4蝶形运算,共 256个蝶形,每个蝶形的距离为 256点;第 2级为 4组 N = 256点的基 4蝶形运算,每组 64个蝶形,每个蝶形的距离为 64点。后 3级类推。这种算法每一级的运算具有相对独立性,每级运算都采用同址运算,因此,本设计只使用了 2个 1 k ×16 bits的 RAM单元。运算过程中所需的旋转因子的值经过查询预设的正弦与余弦 ROM表得到。

2 1 024点 FFT算法模块的设计

本设计的总体框图如图 1所示。整个模块的输入包括 16位带符号实部和虚部数据输入、FFT启动信号,输出包括 16位带符号实部和虚部数据输出、输出有效数据区间标志。内部结构包括 2个 1 k ×16 bits的实部和虚部双口 RAM存储单元、蝶形运算单元、旋转因子生成模块(包括正弦因子查询表、余弦因子查询表和象限转换模块)、RAM和 ROM存储器地址控制单元、倒序模块以及时序总控制单元。

收稿日期: 2007-07-26; 修回日期: 2007-09-25。
基金项目:国家自然科学基金(60572106);厦门大学“985”二期信息创新平台项目。

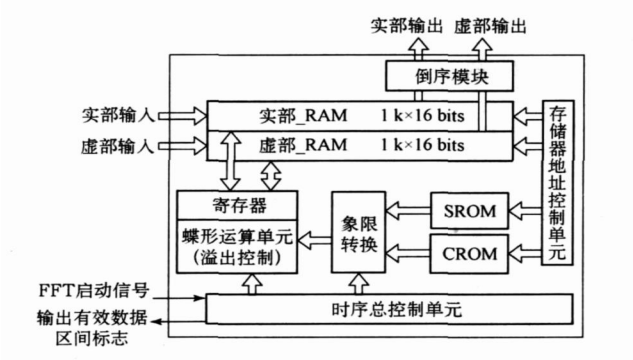

图 1　FFT算法的总体结构框图

下面对主要单元进行分析。

2.1　旋转因子产生模块

在整个 FFT运算过程中,需要存储一组旋转因子表用于蝶形运算,如第 1级运算需要的旋转因子有 $W_{1\,024}^{0}, W_{1\,024}^{1}, \cdots, W_{1\,024}^{1\,023}$,根据旋转因子的可约性,后几级运算所需的旋转因子都可以在这一组数据中查到,因此无需另外存储。为了更节省存储资源,本设计只在 ROM单元中存储了前 256个旋转因子数据,即第 1象限因子 $W_{1\,024}^{0}, W_{1\,024}^{1}, \cdots, W_{1\,024}^{255}$,。其余象限的因子通过象限转换后得到。这样便可以节省 3/4的 ROM存储单元的硬件资源。

2.2　蝶形运算单元

2.2.1　蝶形整体结构

蝶形运算单元包括输入输出寄存器、串/并转换、并/串转换和复数乘法器等。从基本的基 4蝶形运算表达式可以看出,每一级的输出数据在进入下一级运算之前都要首先与旋转因子 W_N^{nk} 进行相乘。本设计采用如图 2的蝶形运算器结构。

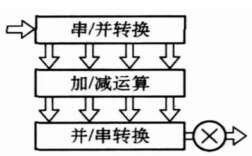

图 2　蝶形运算器结构

这种结构是经过优化的蝶形运算器结构,文献[3]给出了这一结构的具体分析,这样的结构与传统的需要 3个复乘单元的蝶形结果相比,因为采用了流水线控制,硬件上节省了 2个复乘单元,而输出同样只需 4个时钟周期,工作效率并未降低。在 FPGA设计中,一个乘法器的引入,尤其是高位数的乘法器的引入,将很大程度地影响系统整体的运行速率,并且将占用大量的资源。因此,这种改进方案更有利于 FFT算法的高效实现。

2.2.2　复乘器设计

对于复乘单元的设计,常见的复乘方式为:

$$x' + y'\mathrm{i} = (x + y\mathrm{i})(c + s\mathrm{i}) = xc - ys + \mathrm{i}(yc + xs) \tag{2}$$

式中:i为虚数单位。

这种乘法表达式需要 4个实数乘法运算和 2个加减运算,设计中对表达式进行如下变换:

$$x' + y'\mathrm{i} = (x + y\mathrm{i})(c + s\mathrm{i}) = x(c + s) - (y + x)s + \mathrm{i}[x(s + c) + (y - x)c] \tag{3}$$

式(3)这种复乘方式只需要 3个实数乘法运算和 5个加减就可以完成复乘运算,减少了乘法器数量。式中 $(c + s)$值可以在进行象限转换的同时通过计算得到,而无需另外存储。

2.2.3　数据溢出控制

为了防止数据计算过程中的溢出,上述蝶形单元中的加减法运算单元对于输入的 4个有符号复数数据采取了符号位扩展相加后再对计算结果进行 1/4倍压缩的方法进行计算。而对于乘法单元则采用了刻度(scaling)的方法[4],将复数数据(16位)与旋转因子(8位)相乘后,得到 24位数据结果刻度为 16位数据后,再存入 RAM单元中参与下一级运算。经过这样处理后,有效地防止了整个系统在运算过程中出现的数据溢出情况,保证了最终运算结果的可靠性。

2.3　地址产生与总时序控制

在 FFT运算过程中,地址的产生包括复数数据存储 RAM的读写地址(RAM_addr)产生和旋转因子表的读取地址产生。对于不同级运算情况下,RAM读写的控制必须按 DIT的倒序规则进行,这在程序中就需要若干个变量来控制。假设控制级数的变量是 L,每级的蝶形运算距离是 D,当前计算蝶形所在的组为第 S组,共 N组,当前计算蝶形所在组中的位置是第 A个蝶形,那么每个蝶形的 4个输入数据地址分别为:

$$\begin{cases} \mathrm{RAM\_addr1} = (N - S)(D \times 4) + A \\ \mathrm{RAM\_addr2} = \mathrm{RAM\_addr1} + d \\ \mathrm{RAM\_addr3} = \mathrm{RAM\_addr2} + d \\ \mathrm{RAM\_addr4} = \mathrm{RAM\_addr3} + d \end{cases} \tag{4}$$

ROM读取地址 ROM_addr可按如下式子计算得到:

$$\mathrm{ROM\_addr} = \begin{cases} iAN & iAN \leq 255 \\ 512 - iAN & 255 < iAN \leq 512 \\ iAN - 512 & 512 < iAN \leq 768 \\ 1\,024 - iAN & iAN > 768 \end{cases} \tag{5}$$

式中 $iAN = i \times A \times N$: $i = 2, 1, 3$,为输出 4点数据的倒序序号,当 i为 0时数据直接输出,无需对 ROM进行读取。

本设计中采用的 RAM模块为 QuartusⅡ软件中的双口 RAM模块,此模块存储与读取可以同时进行。系统单独完成一个蝶形运算总共需要 11个时钟周期,为了能够充分利用乘法器的运行效率,设计采用了流水线工作方式,平均完成一个蝶形运算只需 4个时钟周期。复数乘法器的工作时序占整个工作时序的 75%,具有较高的工作效率。

综上所述,可以得到如图 3所示流水线工作图。

时序	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	RAM_addr(a)				RAM_addr(b)				RAM_addr(a)				RAM_addr(b)			
			读A/B/C/D					读A/B/C/D				读A/B/C/D				读A/B
					ROM_addr				ROM_addr				ROM_addr			
								M1 M2 M3				M1 M2 M3				M1
								存A′/B′/C′/D′(a)				存A′/B′/C′/D′(b)				存A′

图 3 系统的流水线工作图

图 3中,RAM_addr为分别计算 4个数据地址,地址计算结果将交替存入寄存器组 a和 b中。这种控制方式类似于 Pingpong RAM的控制方式,适用于流水线工作时序中,可以较大地提高系统的工作效率。地址寄存器组 a(或 b)中的第 1个地址在用于保存完本次蝶形运算数据的第 1个计算结果数据之后的,将被立即写入下一个蝶形第 1个数据读取地址,可见这种流水线方式具有非常高的工作效率。

图 3中,ROM_addr为分别计算 3个旋转因子的地址,M1、M2、M3分别为每个蝶形单元的 3次复乘。蝶形运算单元对 4个输入数据 A/B/C/D进行计算,输出结果 4个数据为 A′/B′/C′/D′。可以看出,在这 16个时钟单元中,共有 4个蝶形运算同时处于流水线工作中,因此每个蝶形运算平均只需 4个时钟周期就可以完成。

需要指出的是,在所有蝶形运算结束后,即第 5级运算完成后,所存储在 RAM中的数据是四进制倒序的,为了能在输出端得到正确的 1 024点频域数据,在输出时必须进行四进制倒序输出[5],输出的数据可以直接用于后续的数据分析等工作。

2.4 FFT算法仿真结果

在 Quartus Ⅱ软件中利用 simulator tool工具在 100 MHz的时钟环境下对系统进行了仿真。输入时域数据为一个矩形窄脉冲信号,完成整个 FFT运算的耗时仅为 51.28 μs。仿真得到的矢量波形文件的部分结果如图 4所示。

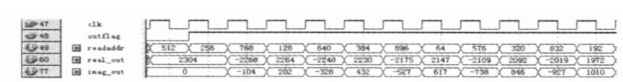

图 4 仿真输出矢量波形文件部分数据

将仿真输出结果转换成 tbl文件并利用 MATLAB软件读取后,得到如图 5所示的频谱数据图(实部数据部分)。

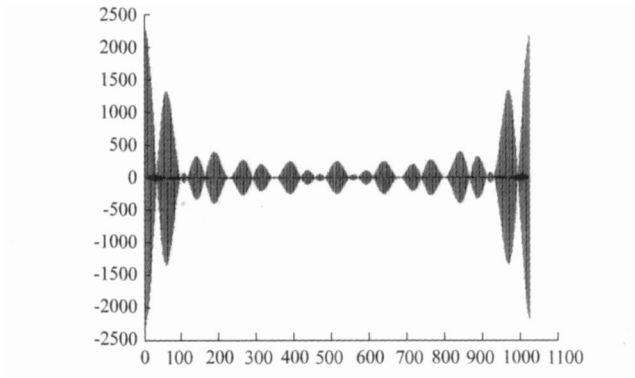

图 5 FPGA仿真输出的实部数据

图 6所示为 MATLAB自带 FFT( )函数对于输入相同 1 024点数据的 FFT计算结果(同样为实部数据部分)。

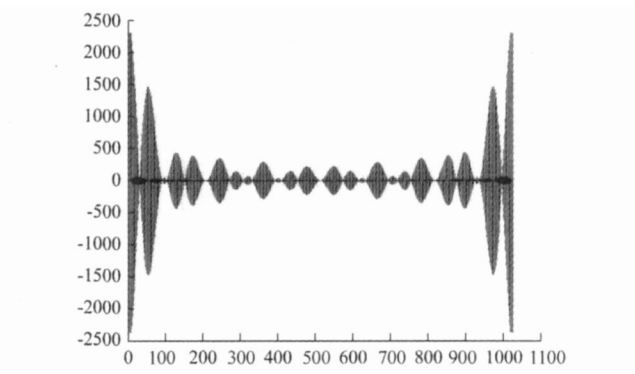

图 6 MATLAB仿真输出的实部数据

通过对比可以看到,本设计的仿真结果与 MATLAB计算的结果基本一致。只在较小值受到了有限字长效应的影响。就总体而言,本设计能够正确而高效地计算输入的 1 024点数据的频域数据值,数据能够有效地用于实际的频谱分析过程中。

3 结束语

1 024点基 4-FFT算法共需要 5级运算,每级需要计算 256个蝶形,由前所述,平均每个蝶形运算需要 4个时钟周期,所以理论上完成 1 024点 FFT的总时钟周期为 $N = 256 \times 4 \times 5 = 5120$;假设使用的时钟为 100MHz,那么将总共耗时 $T = 5120 \times (1/100) = 51.2\ \mu s$,这与仿真结果 51.28 μs基本一致。将所设计的 FFT程序模块在 Altera公司的自带 DSP单元的 stratix系列 FPGA上进行综合后,除了乘法器以及存储单元外,所占据资源仅为 1 619个逻辑单元。因此,本设计方案能够在 FPGA有限的资源下实现较高效率的 FFT算法。

(下转第 53页)

行口中断服务程序,处理接收的串行数据包。

4 结束语

该转换器成功应用于某煤矿风机监测系统的改造项目上。改造前,只有与风机检测单片机系统连接的计算机能观测该风机的运行状态[4],改造后,只要接入矿区局域网的计算机都可以观测到风机的运行状态。

虽然该转换器基本上实现了单片机系统与以太网的通信,但是转换器的功能比较简单,若能把 http协议嵌入到转换器,通过 Web的方式通信会更好。

参 考 文 献

[1]蒋峰. 基于 Internet的数据采集系统的设计与实现[J]. 广西轻工业, 2007, 23(1): 71-72.

[2]郭健,董猛. 实时操作系统 μC/OS-Ⅱ在 W78E516B上的移植[J]. 山西电子技术, 2006(5): 27-28.

[3] RTL8019AS手册[EB/OL]. http://www.efw520.com/down/dc0014_8019as_cn.pdf.

[4]刘小明. 矿井通风机监测系统[J]. 矿山机械, 2006(12).

陈传虎(1974-),男,硕士,讲师,主要从事信息与网络方面的教学与科研工作。

Design of Converter Between Serial Port and Ethernet Interface Based on RTL8019AS

CHEN Chuanhu

(Xuzhou Normal University, Xuzhou 221116, China)

Abstract: With the development of network technique, it is popular that a data acquisition and control system with single-chip microcomputer is connected to a network to share information resources. However, there is some troubles for the incompatibility between RS-232 serial port and RJ-45 network interface. A design of interface converter between RS-232 and RJ-45 is introduced in this paper. The converter consists mainly of several units such as control unit, network interface unit, power supply. All unit circuits are designed in the paper, as well as program flow and working process. The interface converter is applied successfully in fans monitoring system as present.

Keywords: RS-232 serial interface; RJ45 Ethernet interface; converter

(上接第 40页)

参 考 文 献

[1]程佩青. 数字信号处理教程[M]. 2版. 北京:清华大学出版社, 2003.

[2]韩泽耀,韩雁,郑为民. 一种高速实时定点 FFT处理器的设计[J]. 电路与系统学报, 2002, 7(1): 18-22.

[3] SANSALONI T, PEREZ-PASCUAL A, VALLS J. Area-efficient FPGA-based FFT processor[J]. Electronics Letters, 2003, 39(19): 1369-1370.

[4] MEYER-BAESE U. 数字信号处理的 FPGA实现[M]. 刘凌,胡永生,译. 北京:清华大学出版社, 2006.

[5]连冰,宫丰奎,张力,等. 基于 FPGA的快速傅立叶变换[J]. 国外电子元器件, 2003(12): 26-28.

樊光辉(1983-),男,硕士研究生,研究方向为水声无线通信网络技术。

Implementation of High Speed Pipelining FFT Algorithm Based on FPGA

FAN Guanghui, XU Ru, WANG Deqing

(Institute of Underwater Acoustic Communicaion Technology, Department of Communication Engineering, Xiamen University, Xiamen 361005, China)

Abstract: The paper proposes the implement scheme of the 1024 points radix-4 DIT FFT algorithm on FPGA. The butterfly unit, which is the core of the algorithm, is analysed and optimized. Due to timing controlling, the developed butterfly unit uses only one complex multiplier and has the same efficiency to the three multiplier structure. The processor is working in pipeline, which made the implement of the FFT algorithm on FPGA is both area and speed efficient. The simulation result is presented in the last section and compared with the FFT result using MATLAB. It shows that the total simulation time of the 1024 points FFT is 51.28 μs when operated at 100 MHz clock, which well meets the demands of high speed FFT alogrithm.

Keywords: FFT; Radix-4 butterfly operation; Pipeline; FPGA