

高效可配置浮点 FFT 处理器设计

桑红石, 高 伟

(华中科技大学 图像识别与人工智能研究所多谱信息处理技术国家级重点实验室, 湖北 武汉 430074)

摘 要: 为了克服高精度浮点 FFT 处理器具有较大资源开销的设计瓶颈, 采用基于单口存储器的 FIFO 构建共享蝶形结构的 $R2/2^2$ SDF 流水可配置结构. 采用适合浮点设计的基 $2/2^2$ 算法实现流水结构, 不仅有利于可配置电路的实现, 还能够有效减少复数乘法次数, 提高复数乘法器的计算效率. 采用双倍数据位宽的单口存储器实现 FIFO 存储器, 有效避免了双口存储器面积和功耗较大的问题. 改进的蝶形共享结构实现两级蝶形的合并, 解决了单路径延迟反馈流水线结构蝶形单元利用率低的问题. 与传统流水线结构 FFT 处理器设计相比, 有效降低了浮点设计中的资源开销, 提高了计算单元的利用效率.

关键词: 快速傅里叶变换; 可配置浮点设计; 共享蝶形; 单路径流水结构

中图分类号: TN911.7

文献标识码: A

文章编号: 1000-7180(2012)04-0036-05

An Efficient Architecture Design of Reconfigurable Float-point FFT Processor

SANG Hong-shi, GAO Wei

(National Key Laboratory of Science and Technology on Multi-spectral Information Processing,
Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science
and Technology, Wuhan 430074, China)

Abstract: Large resource cost is the design bottleneck of high-precision float-point FFT processor, a novel $R2/2^2$ SDF reconfigurable architecture using shared-butterfly which employs single-port-based FIFO. Radix $2/2^2$ algorithm and pipeline architecture, which is suitable for float-point design, can reduce the multiplicative complexity and improve the multiplication efficiency. The FIFO memory using double-width single-port ram can avoid the larger area and power coat of dual-port ram. Two butterfly units can be merged by the proposed shared-butterfly architecture, which solves the low utilization factor problem of traditional single-path-delay-feedback architecture. The float-point design cost is efficiently reduced and the calculator utilization factor is improved, compared with the traditional pipeline method.

Key words: fast fourier transform; reconfigurable float-point design; shared-butterfly; single-path-delay-feedback pipeline architecture

1 引言

快速傅里叶变换(FFT)是离散傅里叶变换(DFT)的快速实现形式,是数字信号处理中的常用运算.随着技术的不断发展,对于FFT计算精度和面积的要求越来越高.目前,国内外的FFT处理器

在数据格式方面大多集中于定点格式,部分设计使用块浮点格式以提高计算精度^[1].定点格式计算简单,但是需要考虑数据溢出检测和处理机制,数据动态范围小^[2].块浮点格式通过共享指数来减少数据存储,但需要专门设计块浮点整理单元在一组处理完后调整共享指数,带来连续流水的困难,并且也存

在数据溢出的问题. 浮点格式具有较大资源和功耗, 但计算精度高, 处理的数据动态范围大, 能够有效地降低大量计算带来的累积误差, 非常适合于如雷达成像系统中回波信号的处理. 并且由于不存在数据溢出问题, 便于使用流水线结构加快计算速度, 但目前对于浮点格式实现的研究文献较少, 较多见于 FPGA 厂商提供的浮点 FFT 的 IP 核^[3].

在 VLSI 实现结构方面, 主要分为基于存储器迭代方式的结构^[4-5]和单路径(SDF)/多路径(MDC)流水线结构^[6]. 基于存储器迭代和多路径流水方式的设计面积开销较大, 并且多路径流水对于外部带宽需求大. 而单路径流水结构具有较高的计算速度和较低的面积开销, 因此为了满足高速高精度系统的应用需求, 本文基于单路径流水结构, 提出基于一种基于蝶形共享结构的低开销 $R2/2^2$ SDF 流水结构, 降低浮点 FFT 设计的面积开销, 提高 SDF 流水结构中蝶形计算单元的使用效率, 减少复数乘法次数和复数乘法器数目.

2 浮点 FFT 算法

2.1 混合基算法原理

对于一组序列 $x(n)$ ($n=0, 1, 2, \dots, N-1$), 其离散傅里叶变换公式为

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

其中 $k = 0, 1, 2, \dots, N-1$ (1)

Cooley-Tukey 及以此为基础的混合基和分裂基算法, 由于具有较好的规则性, 通常被用于 FFT 处理器的 VLSI 实现. 其中分离基算法的 L 型结构在控制上较为复杂, 所以在可配置 FFT 结构中常常优先考虑混合基算法. 混合基算法将较大点数的序列重新划分为多个小点数计算的多维计算^[7], 小点数的计算进一步划分, 最后可以使用基本的蝶形结构加减法运算来实现, 从而降低计算量.

对于一组 $N=r_1 r_2$ 长度的数据序列, 时域变量 n ($0 \leq n < N$) 可以表示为 $n=n_1 r_2 + n_0$, 其中 $n_1=0, 1, 2, \dots, r_1-1$ 且 $n_0=0, 1, 2, \dots, r_2-1$. 若令 $N=r_2 r_1$, 频域变量 k ($0 \leq k < N$) 表示为 $k=k_1 r_1 + k_0$, 其中 $k_1=0, 1, 2, \dots, r_2-1$ 且 $k_0=0, 1, 2, \dots, r_1-1$, 那么得到 FFT 组合基计算公式:

$$\begin{aligned} X(k) &= X(r_1 k_1 + k_0) = X(k_1, k_0) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \\ &= \sum_{n_0=0}^{r_2-1} \sum_{n_1=0}^{r_1-1} x(r_2 n_1 + n_0) W_N^{(r_2 n_1 + n_0)(r_1 k_1 + k_0)} \end{aligned}$$

$$\begin{aligned} &= \sum_{n_0=0}^{r_2-1} \sum_{n_1=0}^{r_1-1} x(n_1, n_0) W_N^{r_2 n_1 r_1 k_1} W_N^{r_2 n_1 k_0} W_N^{r_1 k_1} W_N^{n_0 k_0} \\ &= \sum_{n_0=0}^{r_2-1} \left[\sum_{n_1=0}^{r_1-1} x(n_1, n_0) W_{r_1}^{n_1 k_0} \right] W_N^{n_0 k_0} W_{r_2}^{n_0 k_1} \\ &= \sum_{n_0=0}^{r_2-1} [X_1(k_0, n_0) W_N^{n_0 k_0}] W_{r_2}^{n_0 k_1} = X_2(k_0, k_1) \end{aligned} \quad (2)$$

首先将二维序列分为 r_2 个 r_1 长度的一维序列, 对 r_2 个这样的序列分别做 DFT 计算得到 $X_1(k_0, n_0)$, 然后将上述计算结果分别乘以旋转因子 $W_N^{n_0 k_0}$, 最后对 r_1 个 r_2 长度的序列分别做 DFT 计算, 得到最后结果, 这样就将大点数的 FFT 转换为小点数的 FFT 计算.

2.2 基 $2/2^2$ 与基 $2/2^2/2^3$ 算法比较

算法的基数高低不但能够影响计算的并行性和速度, 还会影响到复数乘法的计算量和复数乘法器数目, 基 $2/2^m$ 的组合基算法能够在灵活配置 2 的幂次方点数计算的同时, 尽量提高计算基数以提升性能. 但是另一方面高基数算法具有较大的硬件开销, 在 SDF 流水线结构中, 更高基的算法在结构上可以使用单纯基 2 结构的多级级联实现^[8], 从而能够在减少复数乘法计算量的同时, 保持与基 2 算法相同的蝶形资源开销.

在定点和块浮点格式设计中, 常常采用最高基 8 的算法, 而对于 R8SDF 结构则采用 $R2/2^2/2^3$ SDF 结构, 可以设计一种可配置的 $R2/2^2/2^3$ 蝶形结构^[8]. 但是对于浮点设计而言, 基 2^3 结构中的第二级乘数因子 W_8^1 和 W_8^3 仅采用若干次移位加法会影响计算精度, 并且这些浮点加法面积甚至超过浮点乘法面积. 所以为了保证计算精度, 若采用 $R2/2^2/2^3$ 算法则在一个基 2^3 蝶形结构中需要采用两个复数乘法器.

对浮点格式的 $R2/2^2$ 和 $R2/2^2/2^3$ 算法 SDF 实现结构中的复数乘法次数和复数乘法器数目的比较如表 1 所示. 与定点格式的统计不同, 这里对于旋转因子 W_8^1 和 W_8^3 的乘法需要计算在复数乘法次数中, 而对于 $-j$ 和 1 则通过设计控制电路避免复数乘法运算.

从复数乘法次数的比较可以看出, 基 $2/2^2$ 算法与基 $2/2^2/2^3$ 算法基本相当, 甚至少于基 $2/2^2/2^3$ 算法的复数乘法次数; 从两种 SDF 结构的复数乘法器硬件开销看, 基 $2/2^2$ 算法的开销也较小, 并且 $R2/2^2$ SDF 与 $R2/2^2/2^3$ SDF 结构具有相同的数据吞吐

率能力,这表明对于浮点格式的计算而言,基 $2/2^2$ 算法与基 $2/2^2/2^3$ 算法相比更适合于 FFT 的结构设计,具有更低的面积开销和更高的复数乘法器的有效利用率.

表 1 浮点 $R2/2^2$ SDF 与 $R2/2^2/2^3$ SDF 处理时复数乘法比较

点数 (N)	R2/2 ² SDF 结构		R2/2 ² /2 ³ SDF 结构	
	浮点复乘 次数	浮点 复乘器	浮点复乘 次数	浮点 复乘器
64	76	2	80	3
128	214	3	252	4
256	492	3	508	4
512	1 238	4	1 208	5
1 024	2 732	4	2 926	6
2 048	6 486	5	6 364	6
4 096	13 996	5	14 264	7
8 192	32 086	6	32 622	8

3 改进的 $R2/2^2$ SDF 可配置结构

图 1 是基于改进共享蝶形结构的 $R2/2^2$ SDF 可配置整体结构图,与传统单路径流水整体结构不同的是将相邻两级的蝶形合并. 主要包括 FIFO 地址产生和配置逻辑、共享蝶形和存储器读写控制逻辑、ROM 地址产生和可配置旋转因子数据变换电路和其它一些可配置选通逻辑. 不同的点数模式具有不同的数据入口,每两种相邻点数共用一种数据入口,同时需要改变每级 FIFO 存储器的使用模式.

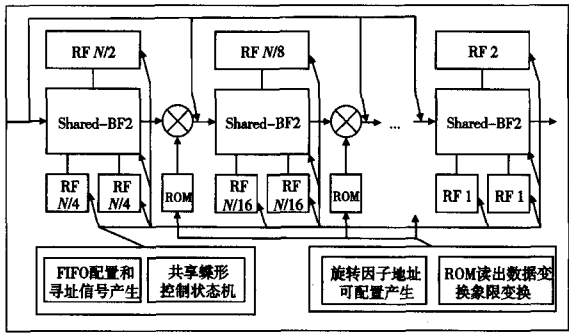


图 1 改进的 $R2/2^2$ SDF 可配置整体结构

3.1 单口存储器的 FIFO 设计

单口存储器相比于双口存储器^[5]具有面积小和功耗低的优势^[9],对于大点数和高精度的 FFT 计算而言,存储器的数据调度更是频繁,因而有必要基于单口存储器设计具有同时读写能力的 FIFO 存储器,这对于降低浮点 FFT 处理器的面积和功耗非常

重要.

文献[10]中提出了使用单口存储器构建 FIFO 的方法,过多的增加位宽虽然可以减少对于存储器的访问次数,但是却额外增加了不必要的数据串并和并串开销,并且数据位宽过大也会带来功耗和时序上的问题. 为此本文为结合浮点设计数据信号位宽较大的特点,使用 64 bit+64 bit 双倍数据位宽的单口存储器来构建 FIFO,如图 2 所示为采用双倍数据宽度,具有同时读写能力的单口存储器. 一个 64 bit 数据需要等待下一个时钟的 64 bit 数据,写入 128 bit 的单口存储器中,这样在一个写时钟内写入两个数据,而每两个系统时钟进行一次写操作,在写操作的下一个时钟进行双倍数据位宽的读操作,只需不断翻转读写使能信号就可以满足同时读写存储器的数据需求.

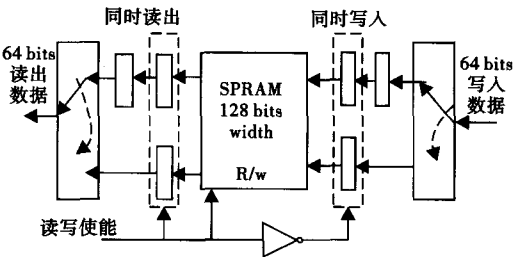


图 2 双倍数据位宽构成的同时读写存储器

3.2 共享蝶形结构设计

如图 3(a)所示, SDF 流水结构存在蝶形单元未被充分利用的缺点,蝶形单元在流水状态下仅有一半时间进行有效计算. 为此可以做出如图 3(b)所示的时序安排的改进, RAM $k+1$ 和 RAM $k+3$ 对应的蝶形计算和 FIFO 读写做部分交换,那么可以将相邻两级的蝶算和 FIFO 存储器读写控制合并,就可以减少一半的蝶形单元的计算开销,并充分利用蝶形计算单元.

图 4 所示为针对自然序输入和倒位序输出的 SDF 流水结构提出的共享蝶形结构.

对于一组长度为 N 的序列,传统 SDF 流水结构的第 k 级和第 $k+1$ 级做合并,则图 4 中 $D=N/2^k$,其具体的数据调度步骤如下:

- (1) 通过 D 个时钟周期将 D 个数据写入 MEM1;
- (2) 输入 D 个数据,与 MEM1 中数据蝶算和常数- j 乘法后,结果 oup_tp 依次写入存储器 MEM2a 和 MEM2b,并将 odn_tp 写回 MEM1;
- (3) 经过 $D/2$ 个时钟将 MEM2a、MEM2b 同位置数据依次读出做蝶算,其中 oup_fn 输出, odn_fn

图 5 蝶形共享结构的面积性能

5 结束语

针对浮点 FFT 处理器实现中资源开销较大的问题,提出基于蝶形共享策略的 $R2/2^2$ SDF 流水结构,在保证高速流水处理的前提下,提高浮点计算单元的利用率并降低面积开销。

通过比较分析发现基 4 算法是实现浮点 FFT 流水结构的较好算法,在复数乘法次数和复数乘法器数目方面优于使用较多的基 8 算法。在使用单口存储器构建 FIFO 读写时序的前提下,提出一种蝶形共享的可配置流水结构,不但有效提高了蝶算单元的使用效率,而且减少了面积开销。

本文提出的方法有效解决了浮点设计中资源开销瓶颈,可以应用于其他高精度可配置的 FFT/IFFT 处理器设计中。

参考文献:

- [1] Yuan chen, Lin Yuwei, Lee Chenyi. A block scaling FFT-IFFT processor FOR WiMax applications[C]//IEEE Asian Solid-State Circuits Conference. Hangzhou: IEEE, 2006: 203-206.
- [2] 云霄. 可配置 FFT/IFFT 处理器的设计及其 FPGA 构造[D]. 西安: 西安电子科技大学, 2009.
- [3] LogiCORE IP fast fourier transform v8.0[EB/OL]. (2010-09-21). [2011-06-20]. http://china.xilinx.com/support/document-ation/ip_documentation/ds808_xfft.pdf.
- [4] 吴斌, 姜鑫, 周玉梅. 基于 802.11a 的 FFT/IFFT 处理器设计[J]. 微电子学与计算机, 2011, 28(4): 61-64.
- [5] Xiaojin Li, Zongsheng Lai, Jianmin Cui. A low power and small area FFT processor for OFDM demodulator[J]. IEEE Transactions on Consumer Electronics, 2007, 53(2): 274-277.
- [6] 何星, 张铁军, 侯朝焕. 流水线结构 FFT/IFFT 处理器的设计与实现[J]. 微电子学与计算机, 2007, 24(4): 141-147.
- [7] 姜鑫, 唐广, 吴斌. 可配置高精度 FFT/IFFT 的设计[J]. 微电子学与计算机, 2010, 27(7): 44-48.
- [8] Hao Xiao, An Pan, Yun Chen, et al. Low-cost reconfigurable VLSI architecture for fast fourier Transform[J]. IEEE Transactions on Consumer Electronics, 2008, 54(4): 1617-1622.
- [9] Yang Yaoxian, Li Jinfu, Liu Hsiangning. Design of cost-efficient memory-based FFT processors using single-port memories[C]//IEEE International Conference on SOC. Hsinchu: IEEE, 2007: 29-32.
- [10] Ambarish Mukund Sule. Design of pipeline fast fourier transform processors using 3 dimensional integrated circuit technology[D]. Raleigh: North Carolina State University, 2007.

作者简介:

桑红石 女, (1970—), 副教授. 研究方向为图像处理和集成电路设计等。

高伟 男, (1987—), 硕士研究生. 研究方向为数字信号处理的 VLSI 设计。

(上接第 35 页)

- [9] Gupta P, Balkman J. PALAC: PAccket lookup and classification simulator[EB/OL]. [2011-06-10]. http://klamath.stanford.edu/tools/PALAC/SRC/DOCUMENT/palac_manual5.pdf.
- [10] Taylor D, Turner J. ClassBench: A packet Classification Benchmark[J]. IEEE/ACM Trans on Networking, 2007, 15(3): 499-511.

作者简介:

李金库 男, (1976—), 博士, 副教授. 研究方向为网络体系架构\计算机系统及网络安全。

马建峰 男, (1963—), 博士, 教授, 博士生导师. 研究方向为无线/移动网络体系及安全。

张德运 男, (1941—), 教授, 博士生导师. 研究方向为网络系统结构、网络安全。