

大点数 FFT 在同构多核系统中的映射实现

曲冬明, 王晓蕾, 张多利, 宋宇鲲, 杜高明
(合肥工业大学 电子科学与应用物理学院, 安徽 合肥 230009)

摘 要: 大点数 FFT 被广泛地应用于图像处理、雷达信号处理、卫星通信、生物医学等高吞吐、高实时性的应用中, 同时, 在过去几年中多核系统的发展一直是主流方向. 高效、低资源消耗、且便于多核系统实现的大点数 FFT 映射方案极有研究价值. 本文介绍一种在同构多核系统中实现大点数 FFT 的映射方案. 该方案在片上存储消耗和资源运算节点(PE)的负载之间取得了很好的折中. 本文采用了 2 维 FFT 原理和基 2 时间抽取(DIT)FFT 方法将大点数 FFT 分成若干较小规模, 同时送到多个资源运算节点(PE)中并行计算. 为了减少处理器执行时间和编程任务量, 本文巧妙的将每级蝶型运算的源数据和结果数据分别访问相同的存储区. 同时, 本文设计了一种流水线结构来实现多个大点数 FFT 计算, 提高了运算的并行性.

关键词: 多核系统, 浮点处理器, 大点数 FFT, 映射方案, 并行计算

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2018)12-0090-05

The Implementation of Large FFT on Homogeneous Multi-core Systems

QU Dong-ming, WANG Xiao-lei, ZHANG Duo-li, SONG Yu-kun, DU Gao-ming

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

Abstract: Large FFT is widely applied to high throughput and high real-time applications such as image processing, radar signal processing, satellite communications and biomedicine. At the same time, the development of multi-core system has been the mainstream in the past few years. So it is valuable to study the implementation of large FFT on the multi-core systems which is high efficient and low resource consumption Extremely. This paper presents one implementation of large FFT on homogeneous multi-core system and achieves a good balance between the on-chip memory consumption and the payload in the process elements (PE). This paper adopts the 2-dimension FFT principle and radix-2 decimation-in-time (DIT) FFT method, which divides the large FFT into several shorter ones and computes each part with the process elements (PE) on the homogeneous multi-core system. In order to reduce the time of processor execution and reduce programming tasks, this paper make the source data and the results of data of each butterfly access to the same storage area skillfully. In addition, a pipeline structure is introduced for multiple FFTs computing at the same time aiming at improving the parallelism.

Key words: Multi-core System, Floating processors, Large FFT, implementation method, Mapping scheme

1 引言

同构多核系统因具有通用处理器的灵活性和 ASIC 的高性能, 成为了集成电路产业中解决计算密集型和数据密集型任务的主流架构. 而 FFT 运算是上述任务中一种典型的处理过程, 在二维信号处

理过程中有非常重要的意义^[1-2]. 因此, 研究大点数 FFT 在多核系统中的映射具有重要价值. 然而由于其片上存储资源的限制, 大点数 FFT 通常消耗计算时间并导致数据在不同资源运算节点之间频繁交互.

为解决上述问题, 本文通过如下工作实现了一

种高效映射方案. (1)根据 FFT 的点数给出了合适的网络规模以及 PE 个数;(2)根据 PE 的本地存储空间和运算功能,使每级蝶型运算的源数据和结果数据分别访问相同的存储区;(3)本文设计了一种流水线结构来实现多个大点数 FFT 计算,提高了运算的并行性^[3]. (4)分析硬件资源和计算周期,并将结果数据与 MATLAB 比较,进行误差分析. 由于基 2 FFT 涵盖范围更广,本文采用二维 FFT 原理和基 2 FFT 方法^[4-8].

2 2-D FFT

基于 PE 本地存储空间的限制,二维 FFT 具有重要的意义,其 DFT 公式如下:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^k, (n, k = 0, 1, 2, \cdots, N-1)$$

令 $N = N_1 \times N_2$, 则有如下公式:

$$\begin{cases} n = m_1 \times N_1 + m_2 \\ k = k_1 \times N_2 + k_2 \end{cases} \quad (1)$$

式中, $m_1 = 0, 1, 2, \cdots, N_2 - 1$,
 $m_2 = 0, 1, 2, \cdots, N_1 - 1$,
 $k_1 = 0, 1, 2, \cdots, N_1 - 1$,
 $k_2 = 0, 1, 2, \cdots, N_2 - 1$.

则存在公式:

$$X(k_1 N_2 + k_2) =$$
$$\sum_{m_2=0}^{N_1-1} \left[\sum_{m_1=0}^{N_2-1} X(m_1 N_1 + m_2) W_{N_2}^{m_1 k_2} \right] W_{N_1}^{m_2 k_1} W_N^{m_2 k_2} \quad (2)$$

在(2)式中,中间括号内的部分恰好是 N_2 -FFT. 由此得出:

$$X_1 = (m_2, k_2) = \sum_{m_1=0}^{N_2-1} x(m_1 N_1 + m_2) W_{N_2}^{m_1 k_2} \quad (3)$$

假设 $G(m_2, k_2) = X_1(m_2, k_2) \cdot W_N^{m_2 k_2}$
则有式:

$$X(k_1 N_2 + k_2) = \sum_{m_2}^{N_1-1} G(m_2, k_2) W_{N_1}^{m_2 k_1} \quad (4)$$

式(4)恰好为 N_1 -FFT. 由上述推理,我们将 N -FFT 分成了二维 FFT. 首先对原始数据进行 N_2 -FFT 计算,然后将结果数据乘上中间旋转因子 $W_N^{m_2 k_2}$,之后得到 N_1 -FFT 的最终结果,如图 1 所示.

由此可知,一维 N 点 FFT 的计算时间是 $\frac{N}{2} \log_2 N$, 然而二维 FFT 因需要额外 N 倍的复数乘法运算,计算时间为 $\frac{N}{2} \log_2 N + N$. 尽管二维 FFT 存在这样的缺点,在计算时可将多个小点数的 FFT 送

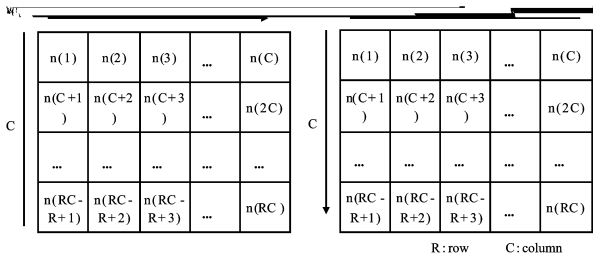


图 1 2-D FFT 变换矩阵

到不同的 PE 中并行计算,从而隐藏了二维 FFT 额外的计算时间.

3 多核系统平台

本文研究围绕面向高密度计算的多核系统设计展开,多个通用浮点处理器(General Float-point Processor, GFP)作为主要运算资源挂载到片上网络上,在主控制器的协调下共同完成计算任务. 综合考虑计算性能要求和资源消耗,本文设计了如图 2 所示 5x4 的 2D mesh 网络. 系统的片上网络按照功能分为配置网络(CFG)、状态网络(STA)和数据网络(PCC)三类,分别用于实现控制命令下行,状态信息上行和数据传输功能. 系统集成了 4GB 的 DDR 存储器 and 以太网口(ET). GFP 作为运算节点通过网络接口挂载到三层网络上,共同协调完成任务.

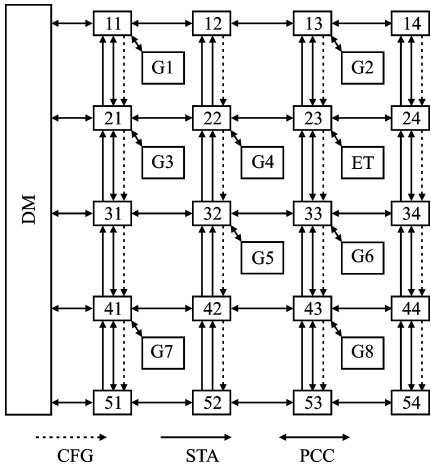


图 2 多核系统结构图

3.1 GFP 的结构及功能介绍

GFP 运算簇提供符合 IEEE-754 标准的 32 位浮点运算能力,主要完成非规则的浮点复数/实数运算,包括:乘法、加减法、除法、开方、取正余弦等运算. 采用“微控制器(MCU)+特殊寄存器(Reg_file)+浮点运算单元(CAU)+存储管理单元(MMU)”

的架构,能够同时完成二路 32 位宽的浮点运算.

GFP 可工作在存储、流、脉动三种运算模式,本文的映射方案中使用到存储和流两种模式,存储模式运算过程被分为接收数据、计算和发送数据 3 个部分,三者串行执行. GFP 可以根据算法特点选择合适的地址跳变方式进行数据读写,最大程度地提高运算灵活性. 流模式中,数据以流水的方式进入 DFP 并执行相应的运算类型,存储管理单元(MMU)重构成 FIFO 用来缓存数据. 数据量不受存储容量的限制. 多核系统中可将多个 GFP 锁定为流模式及相应的运算类型,从而形成一条无阻塞数据通路,以达到高效率运算.

由于本文中的映射方案采用倒序输入,顺序输出的方式,因此需要进行源数据生成. 其方法有两种,第一种是通过软件编程,微控制器 MCU 根据编程指令调整数据. 其伪码如图 3(左)所示,然而这种方法所需要的额外周期数是普通蝶形器的 13 倍,无法适应于大规模 FFT 计算. 于是我们采用了第二种方案,在 MMU 存储管理单元中设计了地址跳转单元,如图 3(右)所示,使其在较短周期内利用较少的硬件消耗完成位反转操作.

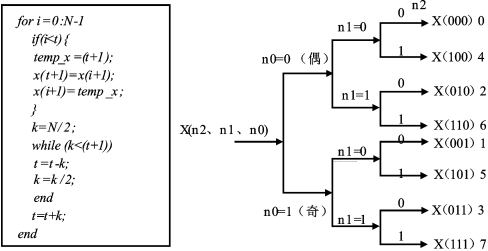


图 3 源数据生成的两种方式

4 算法映射

4.1 单一 GFP 算法映射

本文采用基 2FFT 蝶型算法,倒序输入、顺序输出. GFP 中共有 8 组存储区,每组存储区分为实部和虚部两部分,深度均为 1 K,实部虚部统一编址. 1 K 点 FFT 在单个 GFP 中的映射方案如图 4 所示,该蝶型运算共需要 10 级. 而每级所需的旋转因子不同,若执行每级蝶型运算时都向 DDR 请求旋转因子虽然节省了本地的存储空间,但反复建立路由电路传输小量数据会造成网络阻塞,运算效率降低. 因此本文一次性将所有级数需要的旋转因子全部存入到本地存储区.

当 $L(\text{Level})=1$ 时,共分为 5 步执行完毕. Step1:将 RAM0 中的数据进行位反转存储到

RAM1 中,同时将旋转因子存储到 RAM7 中; Step2:将源数据奇偶分离,分别存储到 RAM2, RAM3 中; Step3:由于第 0 级蝶形运算的旋转因子为 W^0 ,可直接将 RAM2 和 RAM3 中的源数据做复数加减,结果数据存储到 RAM0 中. 至此第 0 级蝶形运算结束,但中间结果数据并未保存至 RAM1 中,无法重复执行 Step2. 因此继续执行 Step4:将源数据 a 分离出来存储到 RAM2 中; Step5:将 RAM0 中的源数据 b 与 RAM7 中的旋转因子进行复数乘法,结果数据存到 RAM3 中. 此时可以发现源数据所处的存储区与 Step3 中的源数据存储区相同,且将要执行的运算与 Step3 也相同.

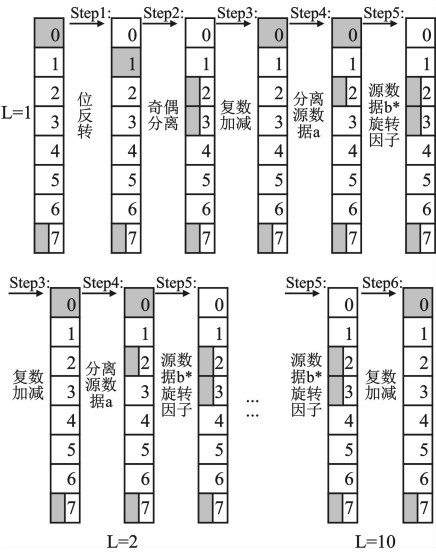


图 4 1K 点 FFT 映射方案

因此当 $L=2$ 时,只需重复执行 Step3, Step4, Step5 即可. 由于每次执行完 Step5 后的中间结果数据都存在 RAM2 和 RAM3 中,所以 $L3, L4, \dots, L9$ 同 $L2$ 一样,循环执行 3、4、5 步即可. 当 $L=10$ 时,执行 Step6,将最终结果数据存储到 RAM0 中. 该映射方案合理分配了存储单元,减少了软件编程的工作量和处理器执行周期.

4.2 多核系统算法映射

8 K 点 FFT 在该多核系统中映射方案如图 5 所示. 系统上电后每个 GFP 所需要的程序指令和源数据通过网口被下发到 DDR 中,由于 GFP 本地存储资源有限,每个 GFP 分别运算 2 K 点 FFT,4 个 GFP 并行运算. 整个映射分为三个阶段:第一阶段:通过 GFP0 和 GFP1 进行源数据生成,将前 4 K 数据送到 GFP0 中,进行奇偶分离,将分离后的偶数送到 GFP1 中,同理接受 GFP1 发来的 2 K 数据如图

中路径 b. 将 GFP0 中的 4 K 数据再进行奇偶分离, 2 K 奇数送到 GFP2 中如路径 c. 同时, 通过 DDR 向 GFP2~5 发送 1 K 的旋转因子. 至此, 第一阶段完成.

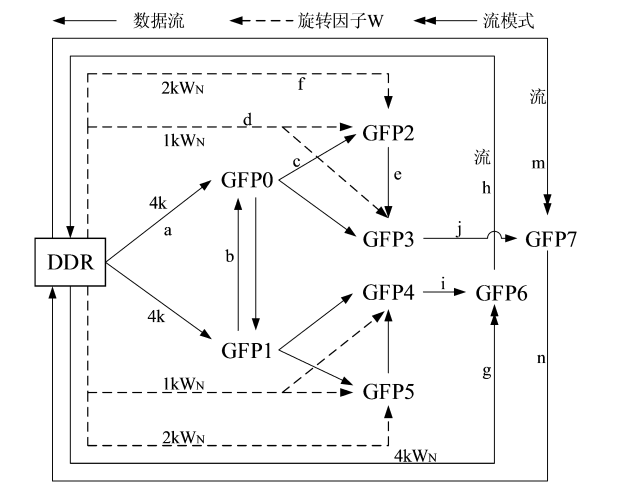


图 5 8K 点 FFT 映射方案

第二阶段根据前面提到的映射方案, 每个 GFP 分别执行 11 级的蝶型运算. 生成 2 K 的结果数据, 同时 GFP2 和 GFP5 接受 DDR 传来的 2 K 的旋转因子, 并与结果数据相乘, 分别将新的结果送到 GFP3 和 GFP4 中进行复数加减如路径 e. 至此, 第二阶段完成. 第三阶段需要将 GFP3 和 GFP4 中的 4 K 数据整合到一起, 而 GFP 中的存储空间为 8 K 无法存放结果数据. 因此采用流模式, 先锁定路径 h, i, g. 将 DDR 发来的 4 K 旋转因子与 GFP4 中的 4 K 数据相乘, 结果直接流到 DDR 中. 再锁定路径 j, m, n, 将刚才的 4 K 结果数据与 GFP3 中的 4 K 数据做复数加减. 生成 8 K 的最终结果数据流进 DDR 中.

4.3 大规模点数 FFT 映射

由上述方案中可看出, 当 GFP0 和 GFP1 执行完第一阶段后始终处于空闲状态, 因此针对大点数 FFT 分成多个 FFT 运算时, 如图 6 所示, 本文提出一种流水线就够, 使其能够隐藏并行的计算时间. 在图 5 的方案中, 阶段一和阶段三所占时间都较短, 而阶段二在计算 11 级蝶形运算时, 消耗的周期是其他阶段的 9 倍多. 这种不平衡性限制了流水线的效率, 为解决上述问题, 在运算资源允许的情况下可将阶段二的 FFT 计算分成多个小段进行. 图 6(b) 为分成两段后的优化效果.

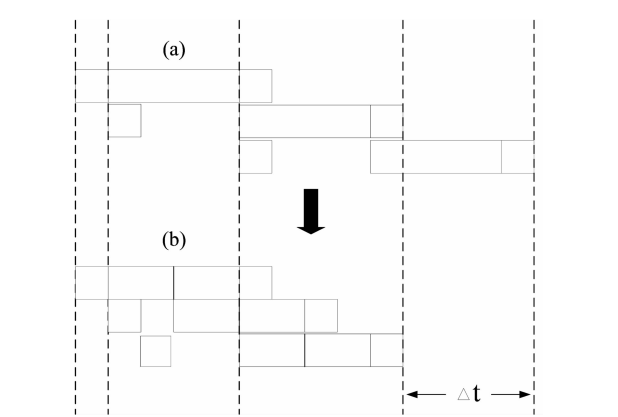


图 6 流水线结构实现多个 FFT 映射

5 实验性能分析

本文在 Xilinx xc7v2000t 的 FPGA 芯片上进行了原型验证. 表 1 给出了单个 GFP 和多核系统的资源消耗和百分比.

表 1 多核系统资源消耗

	Register	LUT	BRAM	DSP
1 个 GFP	16 408	15 374	17	14
百分比(%)	0.66	1.25	1.31	0.64
多核系统	162 439	177 132	158	112
百分比(%)	6.53	14.5	12.16	5.12

表 2 给出了 4 组实验的 FFT 运算周期数和最大相对误差. 由于多核系统片上存储空间有限, 本文中的 16 K 采用二维 FFT 原理实现, 而其他三组实验均采用一维 FFT 原理实现. 实验中源数据通过 MATLAB 中 randn(1, 16K) 函数生成, 同时将结果数据与 MATLAB 中的 FFT 结果进行对比, 记录最大的相对误差. 从表中可以看出, 尽管 FFT 运算中前后级的数据相关性较强, 多级蝶形运算无法并行执行. 但该映射方案仍然展现出了一定的并行度.

表 2 FFT 执行周期和最大相对误差

点数	16K	8K	4K	2K
单核周期	565 564	283 762	123 882	51 540
多核系统	192 416	81 264	46 260	18 944
虚部误差	5.11E-3	1.39E-3	9.26E-3	6.91E-4
实部误差	4.97E-3	3.01E-3	1.28E-3	7.51E-4

6 结束语

大点数 FFT 实现是许多领域的瓶颈, 而二维 FFT 原理减少了片上存储空间的需求, 对多核系统实现大点数 FFT 具有重要意义. 本文提出了一种在

同构多核系统上映射较大点数 FFT 的有效方法. 如 32 K,64 K 甚至更大点数的 FFT 均可按照本文的方法分成若干个较短的 FFT,再结合本文所述映射方法实现. 另外,IFFT 也具有相似的数学表达式,同样可以被映射到相同的结构中而不需要更多的硬件消耗.

参考文献:

[1] Liang C, Huang X. Mapping Parallel FFT Algorithm onto SmartCell Coarse-Grained Reconfigurable Architecture[C]// IEEE International Conference on Application-Specific Systems. Architectures and Processors. IEEE,2009, 231-234.

[2] Ingemarsson C, Källström P, Qureshi F, et al. Efficient FPGA Mapping of Pipeline SDF FFT Cores[J]. IEEE Transactions on Very Large Scale Integration Systems, 2017(99):1-12.

[3] Friedrich J, Le H, Starke W, et al. The POWER8 TM, processor: Designed for big data, analytics, and cloud environments[C]// IEEE International Conference on Ic Design & Technology. IEEE, 2014;1-4.

[4] 李大鹏,张盛兵,罗旻. 32 位 RISC 微处理器“龙腾 R2”浮点流水线的设计和实现[J]. 微电子学与计算机, 2006,23(01):188-191.

[5] Andor Krizsan, Dr. Daniela Volke, Stefanie Weinert, et al. Generator of dynamically reconfigurable processor[J]. Artificial Life and Robotics, 2015, 20(2):

103-108.

[6] Ilija M. Milićic, Djordje Ladjinovic. Computer simulation of two frequency response excitation 1D dynamic model using FFT and IFFT algorithm[C]// International Conference - Contemporary Achievements in Civil Engineering. 2016.

[7] Qureshi I A, Qureshi F. Generalized twiddle factor index mapping of radix-2 fast Fourier transform algorithm[C]// IEEE International Conference on Software Engineering and Service Science. IEEE, 2014.

[8] Schmittfull M, Vlah Z. FFT-PT: Reducing the 2-loop large-scale structure power spectrum to one-dimensional, radial integrals[J]. Physical Review D, 2016, 94(10).

作者简介:

曲冬明 男,(1992-),硕士研究生. 研究方向为集成电路设计与测试. E-mail:765488048@qq. com.

王晓蕾 女,(1978-),硕士,副研究员. 研究方向为集成电路设计理论研究

宋宇鲲 男,(1975-),博士,副研究员. 研究方向为面向数据高密度和计算密集应用的 SoC/MPSO 体系结构与实现.

张多利 男,(1976-),博士,硕士生导师. 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.

杜高明 男,(1977-),博士,硕士生导师. 研究方向为片上网络体系结构性能分析.

(上接第 89 页)

[9] 郭强,张晓萌,朱若函. 基于粒子群优化的模糊神经网络垂直切换算法[J]. 微电子学与计算机, 2015,32(8) : 12-18.

[10] 3GPP TR 25. 942, Radio frequency (RF) system scenarios (Release 10) [S].

[11] 张吉军. 模糊层次分析法[J]. 模糊系统与数学, 2000, 14(2) : 80-88.

[12] 陶洋, 刘小虎, 陶玲. 基于 QoS 的异构网络垂直切换算法[J]. 计算机工程, 2016, 42(12) : 91-96.

[13] 孔令斌, 王军选, 孙长印. 多网络共存下基于贝叶斯

的垂直切换算法[J]. 电讯技术, 2016, 56(2) : 122-127.

作者简介:

杨明极 男,(1971-),博士,教授. 研究方向为移动通信系统.

王 娅(通讯作者) 女,(1993-),硕士研究生. 研究方向为无线通信技术. E-mail:wyaya7766@163. com;

马 池 女,(1992-),硕士研究生. 研究方向为无线通信技术.