

嵌入式粗颗粒度可重构处理器的软硬件协同设计流程

于苏东,刘雷波,尹首一,魏少军

(1. 清华大学信息科学与技术国家实验室,清华大学微电子所,北京 100084)

摘 要: 面向多媒体应用的可重构处理器架构由主处理器和动态配置的可重构阵列 (Reconfigurable Cell Array, RCA) 组成. 协同设计流程以循环流水线和流水线配置技术为基础,采用启发式算法对应用中较大的关键循环进行了软硬件划分,使用表格调度算法实现了任务在 RCA 上的映射. 经过 FPGA 验证, H. 264 基准中的核心算法平均执行速度相比于 PipeRench, MorphoSys, 以及 TI DSP TMS320C64X 提高了 3.34 倍.

关键词: 可重构; 循环; 软硬件划分; 映射

中图分类号: TP311.52

文献标识码: A

文章编号: 0372-2112 (2009) 05-1136-05

Hardware-Software Co-Design Flow for Embedded Coarse-Grained Reconfigurable Processor

YU Su-dong¹, LIU Lei-bo¹, YIN Shou-yi¹, WEI Shao-jun¹

(1. Tsinghua National Laboratory for Information Science and Technology, Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

Abstract: The reconfigurable processor architecture for multimedia application consists of a host processor and a coarse-grained Reconfigurable Cell Array (RCA) as the coprocessor, which can be reconfigured dynamically. The proposed co-design flow is based on loop pipeline and pipelined reconfiguration technologies. Heuristic algorithm is used for hardware-software partition of big kernel loop and a table schedule algorithm for the mapping of task graph. They have been verified in FPGA with some kernels in H. 264 baseline. The average speedup is 3.34 times compared with PipeRench, MorphoSys, and TI DSP TMS320C64X.

Key words: reconfigurable; loop; hardware-software partition; mapping

1 引言

对于复杂的对媒体应用,专用集成电路 (Application Specific Integrated Circuits, ASIC) 有着很高的执行效率,但是缺乏灵活性,而通用处理器灵活性很高但性能不足. 为了解决该问题,人们提出了可重构处理器^[1,2]的概念. 一些研究成果如 MorphoSys^[3]、PipeRench^[4]和 DRESC^[5]已经证明粗颗粒度的可重构处理器能够有效地提高多媒体应用的执行速度. 可重构处理器的设计基于软硬件协同设计技术^[6]. 软硬件划分是协同设计的关键. 一种有效的划分策略是把占据大量任务执行时间的关键循环体^[7]映射到 RCA 上执行,这样能够加速执行速度,减少配置时间. 循环流水线技术^[8,9]会提高循环在 RCA 上的执行效率. 大多数过去的研究都只注意较小的循环^[3,5,9]. 在文献[6]和[10]中,较大的循环被直接划分到处理器和可重构阵列上而没有采用流水线的技术. 文献[9]中,较大的循环被划分为许多小的循环然后顺序的在可重构阵列上执行,这样会增加配置的时间.

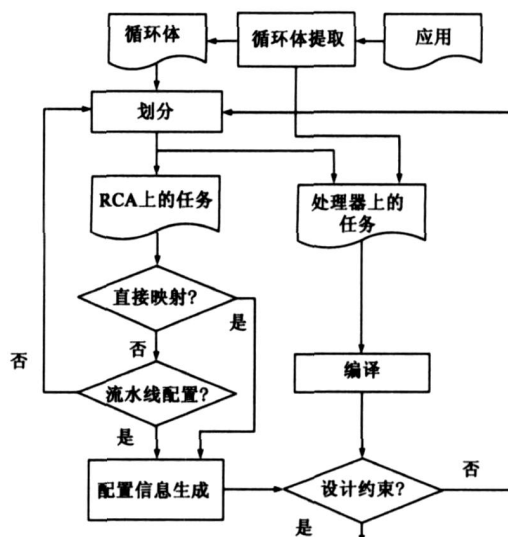


图1 设计流程图

PipeRench 提出了流水线配置技术,这种技术把较大循环的配置信息分成多级流水线,避免对循环的划分. 但该技术需要额外的配置时间. 以上的文献中都没

收稿日期:2008-06-20;修回日期:2009-02-06

基金项目:国家自然科学基金(No. 60506007, No. 60676012)

有考虑循环中数据依赖关系。

本文提出了嵌入式可重构处理器配置信息生成的软硬件设计流程(如图1)。该结构和设计流程面向数据规整且存在大量并行运算多媒体应用,如视频解码。设计流程中从应用中提取以任务图的形势表示的循环体,然后将其转化为 RCA 的配置信息。配置信息里包括每个可重构单元(Reconfigurable Cell, RC)的任务类型以及相互之间的连接关系。较大的循环将被划分为硬件部分和软件部分,硬件部分在 RCA 上执行,软件部分在通用处理器上执行。划分的标准是尽量减少在处理器上执行的任务节点数量,同时减少主处理器和协处理器之间的数据传输。较小的循环将采用表格调度算法被直接映射到 RCA 上,对于一些结构规整且关键路径较短的循环,在将采用流水线配置技术进行映射。如果以上方法都不能完成循环的映射过程,那么将对循环进行再次划分,减少在 RCA 上配置任务节点的数量。系

统在 Xilinx XC4VLX80 FPGA 上得到了验证,验证的应用包括 H. 264 基准中的整数离散余弦变换和运动估计。

2 硬件结构

图2是可重构处理器的硬件结构。主处理器为通用嵌入式处理器。协处理器 RCA 为网格结构,由多个 RC 组成。两者可以并行计算,并通过总线进行数据传输。DMA 负责数据的高速传输。

每个 RC 有一个算术逻辑单元(ALU)和三个多路选择器以及两个输出寄存器组成。临时数据寄存器存储计算中用到的中间数据。RC 的数据宽度可以根据应用的配置成 8 位,16 位或者 32 位。ALU 可以在一个周期内完成算术,逻辑,常数,移位以及乘法运算。同一行或者同一列的 RC 可以自由交换数据。RCA 和缓存有高速接口,数据和配置信息传输都可以在 2 个时钟周期内完成。RCA 每一行为一级,按照从上到下的顺序执行。

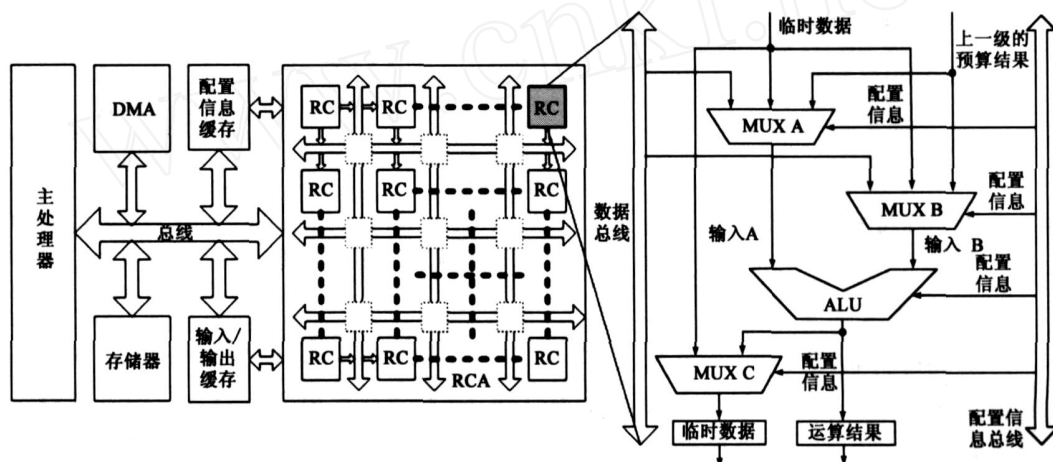


图2 可重构处理器的结构

3 软硬件设计流程

循环体由任务图 $G(V, E)$ 表示: V 表示任务节点的集合, E 表示节点之间边的集合, 每一个节点表示一个能够在 RC 上进行的运算, 每一条边表示输入输出或者节点之间的数据依赖关系。本文设定每个节点最多有 2 个输入 4 个输出。图3 是三个不同循环体的任务图表示。

3.1 循环流水线

如果循环的节点数目小于 RCA 中 RC 的数目, 且循环之间不存在数据依赖关系, 那么对于循环次数是 N 的循环来说, 执行时间(本文中所有执行时间均指时钟周期数)为式(1)。

$$T_{\text{cycle}} = \max(N \times M / B, C + N + S - 1) \quad (1)$$

N : 循环次数

M : 每一次循环的数据传输量(RCA 和外部主

处理器或者 DMA 的数据传输量)

B : 每个时钟周期的传输速率

C : 配置周期数

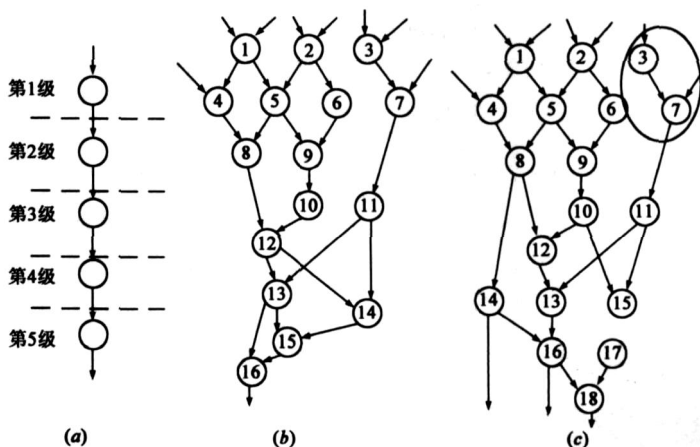


图3 循环体的任务图

S :一次循环在 RCA 上的执行周期数

如果循环中存在数据依赖关系,那么流水线必须中断以等待上一次循环的计算结果,假设需要插入 p 个等待周期,那么执行时间如式(2)。

$$T_{\text{cycle}} = \max(N \times M/B, C + N + S - 1 + N \times p) \quad (2)$$

当 N 很大时,存在数据依赖关系的循环的执行时间远远大于没有数据依赖关系的循环。

3.2 直接映射算法

本文采用表格调度算法来进行节点数量小于 RC 数量的任务映射。假设 RCA 中有 4×4 个 RC。

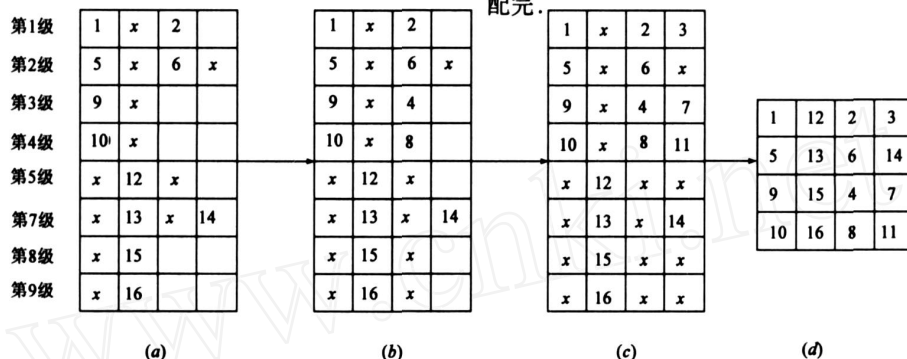


图4 直接映射算法

3.3 流水线配置和后台配置信息切换

当 RCA 中 RC 数量小于循环体节点数量时,可以采

第1步:计算出任务图的关键路径和每个节点可能被放置的级别以及关键路径的长度。

第2步:图4的表格有4列,和 RCA 的宽度相同,行数为关键路径长度。算法按照自由度从小到大进行节点映射,节点必须满足相互的数据依赖关系。“x”表示该位置已经被占用,不能分配节点。由于 RCA 一共有4层,那么如果第 n 层的位置被占用,那么所有该列 $n+4 \times k$ ($-4 < k < 4$) 的位置都被占用。

第3步:如果没有位置可以使用,那么在表格的最下方增加一行,然后回到第2步,直到所有的节点都分配完。

用流水线配置技术。对于 3×1 RCA 和图3(a)中的循环,流水线配置如图5所示。

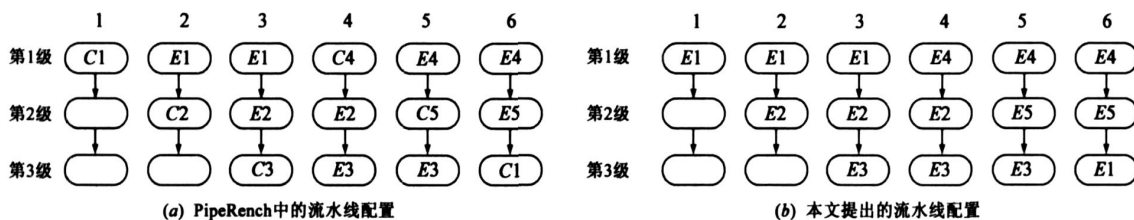


图5 流水线配置两种方式

图5(a)是 PipeRench 中的流水线配置方式, $C1$ 表示任务图第1级节点的配置。一般情况, v 级的任务图映射到 u 级的 RCA 上 ($u < v$), 当 $N \gg u$, 执行时间是式(3)。可以看出执行时间和 $v/(u-1)$ 成正比。

$$T_{\text{cycle}} = \max(N \times M/B, v + \left\lfloor \frac{N-1}{u-1} \right\rfloor v + (N-1) \% (u-1) + 1) \quad (3)$$

图5(b)采用了后台自动配置信息切换技术, 当 $N \gg u$, 执行时间如式(5)。

$$T_{\text{cycle}} = \max(N \times M/B, \frac{N}{u}v), N \gg u \quad (4)$$

当传输速度足够大时, 式(4)执行时间比式(3)减少 $1/u$ 。

3.4 循环体的划分算法

受到面积的限制, RC 数量有限, 因此多数的循环的节点数都大于 RC 的数量。这样的循环体必须进行软硬件划分。对循环体的软硬件划分有3条基本的规则。

1. 关键路径的节点在 RCA 上执行。
2. 尽量减少软硬件之间的数据传输。
3. 软件部分的节点数目应该最小化。

根据这些规则, 本文给出了一种启发式的构造算法。初始假设任务图所有的节点都是在 RCA 上执行, 划分算法描述如下:

第1步: 寻找任务图中的满足条件的边: 如果切断这条边, 那么任务图能分为两个部分。如果划分结果满足 RCA 大小约束, 那么划分完成。否则执行第2步。

第2步: 寻找任务图中的满足条件的非关键路径输入或者输出节点: 使得该节点划分到处理器执行后, 不

增加硬件部分的输入输出代价. 如果划分结果满足 RCA 大小约束, 那么划分完成. 否则执行第 3 步.

第 3 步: 寻找任务图中的满足条件的关键路径上输入或者输出节点: 该节点划分到处理器执行后, 不增加硬件部分的输入输出代价.

在图 3(c) 中, 节点 3 先被划分, 然后是节点 7. 在存在划分的情况下, N 次循环的执行时间如式 (5).

$$T_{\text{cycle}} = \max(N \times M_p / B, N \times A_p) + \max(N \times M_R / B, C + N + S_R - 1) \quad (5)$$

M_p : 在处理器上执行的节点的输入输出数量

M_R : 在 RCA 上执行的节点的输入输出数量

S_R : 划分之后一次循环在 RCA 上的执行时间

A_p : 划分到处理器上的节点数量

当 N 非常大时减少 M_p , M_R 和 A_p 能够大幅度的提高执行速度.

4 验证系统

验证系统称为可重构多媒体系统 (REconfigurable Multi-media System, REMUS). 通用嵌入式处理器 ARM926 作为主处理器, 协处理器 RCA 由 Xilinx XC4VLX80 FPGA 实现. 对于 32 位 4×4 RCA, 采用 SMIC 0.18 μm 1P6M lo-

gic 工艺, 面积 514,623 μm^2 , 相当于 60,000 等效 2 输入与非门.

整数余弦变换/反变换和运动估计是 H.264 中的核心算法. 整个 DCT 包括 8 次一维 DCT 和两次矩阵转置 (如图 6). 采用直接映射的方法, RCA 同时执行两行 DCT, 需要 4 个时钟周期. 矩阵转置需要 4 个时钟周期. DCT 的总执行时间为 16 个周期, 考虑数据传输的代价, 100 次 8×8 DCT 的执行时间是 800 个时钟周期. 8×8 DCT 快速算法和 4×4 DCT 类似. 100 次 8×8 DCT 的执行时间是 3200 个时钟周期.

$$SAD = \sum_{m=0}^N \sum_{n=0}^N |S_k(m, n) - S_{k-1}(m+i, k+j)| \quad (6)$$

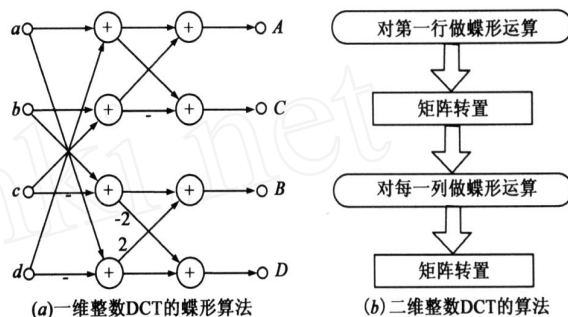


图6 整数DCT的任务图

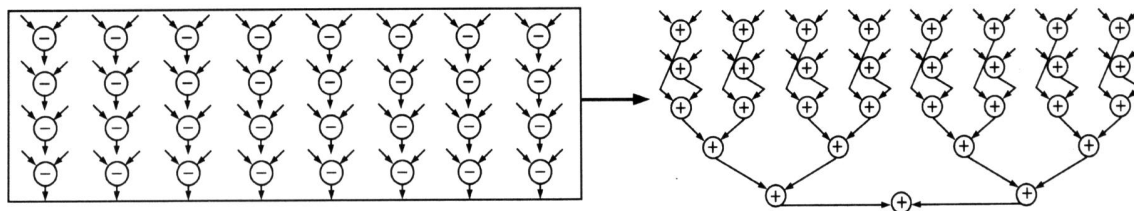


图7 8×8 ME的映射结果

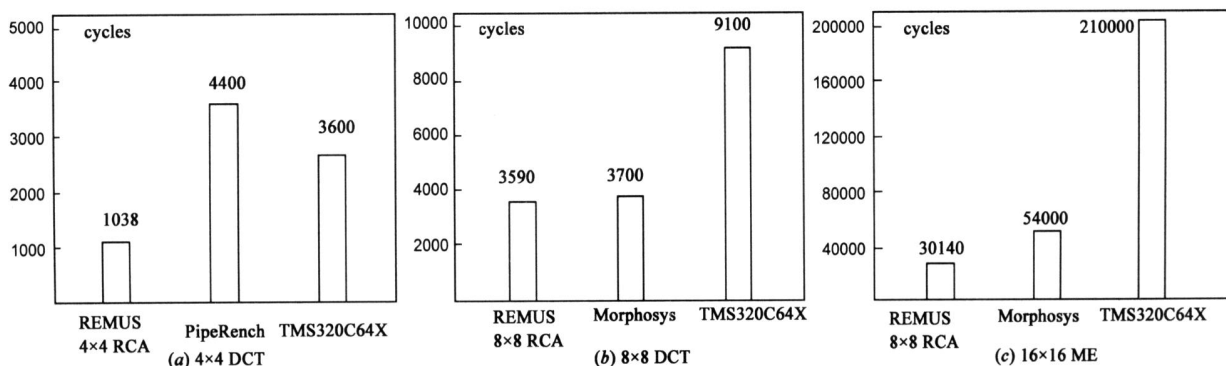


图8 性能比较

运动估计是视频编解码的核心部分, 公式如式 (6), 在 8×8 REMUS 上的映射如图 7. 8×8 ME 的执行时间是 10 个周期. 利用流水线配置技术, 100 次 16×16 ME 的执行时间是 14400 个时钟周期. 图 8 是 REMUS 和两种同类型的结构 (MorphoSys 和 PipeRench) 以及一种高性能 TI DSP (TMS320C64X^[11]) 的比较结果 (由于通信代

价, 实际值较理论值偏大). 从这些比较中可以看出, 采用了本文的硬件架构和软硬件设计流程, 相对于同类的结构以及高性能 DSP, REMUS 有 1.03 到 6.97 倍的性能提升.

5 结论

本文提出了一种新的针对多媒体应用 (例如 MP3,

MPEG4 和 H. 264 等) 可重构处理器架构和相应的软硬件协同设计流程, 该流程基于循环流水线技术自动的产生粗颗粒度可重构处理器的配置信息. 其中包括关键循环的划分算法, 映射算法, 以及流水线配置技术. 验证平台与其他可重构处理器以及 DSP 相比, 性能大约提高了 3.34 倍. 本文中的技术能够广泛的用于多媒体处理器的设计中.

参考文献:

- [1] R Hartenstein. A decade of reconfigurable computing: A visionary retrospective [A]. 2001, Design, Automation and Test in Europe Conference and Exposition (DATE 2001) [C]. Munich, Germany: IEEE Press Piscataway, NJ, USA, 2001. 642 - 649.
- [2] S Vassiliadis, S Wong, G N Gaydadjiev, K L M. Bertels, G K Kuzmanov, E Moscu Panainte. The molen polymorphic processor[J]. IEEE Transactions on Computers: 2004, 53 (11): 1363 - 1375.
- [3] H Singh, et al. MorphoSys: An integrated reconfigurable system for data-parallel and computation-intensive applications [J]. IEEE Trans. Computers, 2000, 49(5): 465 - 481.
- [4] S C Goldstein, et al. PipeRench: A reconfigurable architecture and compiler[J]. Computer, 2000, 33(4): 70 - 77.
- [5] Bingfeng Mei, et al. Exploiting loop-level parallelism on coarse-grained reconfigurable architectures using modulo scheduling [A]. 2003 Design, Automation and Test in Europe Conference and Exposition (DATE 2003) [C]. Munich, Germany: IEEE Computer Society, 2003. 10296.
- [6] Y Li, T Callahan, E Darnell, R Harr, U Kurkure, J Stockwoo. Hardware-software co-design of embedded reconfigurable architectures [A]. Proceedings, 37th Design Automation Conference (DAC 2000) [C]. Los Angeles, California, United States: ACM New York, NY, USA, 2000. 507 - 512.
- [7] IMPACT research group [Z]. University of Illinois, at Urbana-Champaign. <http://www.crhc.uiuc.edu/IMPACT/>, 1989 - 2008.
- [8] Kiran Bondalapati. Parallelizing DSP nested loops on reconfigurable architectures using data context switching [A]. 38th DAC [C]. Las Vegas, Nevada, United States: ACM New York, NY, USA, 2001. 273 - 276.
- [9] K Bondalapati, V Prasanna. Loop pipelining and optimization for run-time reconfiguration [A]. Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing [C]. Cancun, Mexico: Springer-Verlag London, UK, 2000. 906 - 915.
- [10] J Lee, K Choi, N Dutt. Compilation approach for coarse-grained reconfigurable architectures [J]. IEEE D&T, 2003, 20 (1 - 2): 26 - 33.
- [11] Texas Instruments Inc. TMS320C6000 Assembly Benchmarks at Texas Instruments: C64X DSP Benchmarks [Z]. www.ti.com, 2000.

作者简介:

于苏东 男, 1979 年 12 月出生于江苏盐城. 2001 年毕业于清华大学电子工程系微电子专业, 其后进入清华大学微电子学研究所集成电路设计专业, 现为硕博连读生, 主要研究领域为: 可重构处理器的硬件架构设计、可重构处理器的软硬件协同设计等. E-mail: ysd97@mails.tsinghua.edu.cn

刘雷波 男, 清华大学微电子学研究所副教授, 博士. 1999 年毕业于清华大学电子工程系无线电技术与信息系统专业, 获学士学位. 2004 年毕业于清华大学微电子学研究所集成电路设计专业, 获博士学位. 2004 年 9 月在清华大学微电子学研究所参加工作至今. 主要研究领域包括: 音频视频信号处理、超大规模集成电路设计、数字信号处理在安全领域的应用、图像压缩技术等. E-mail: liulb@mail.tsinghua.edu.cn