

引用格式:尹文志,赵仲元,毛志刚,等.一种快速高效的粗粒度可重构架构编译框架[J].微电子学与计算机,2019,36(8):45-48. [YIN W Z, ZHAO Z Y, MAO Z G, et al. A fast and efficient compiler framework for CGRAs [J]. Microelectronics & Computer, 2019, 36(8): 45-48.]

一种快速高效的粗粒度可重构架构编译框架

尹文志,赵仲元,毛志刚,王 琴,绳伟光

(上海交通大学 电子信息与电气工程学院,上海 200240)

摘 要: 利用硬件和软件协同的设计技术来进一步提高粗粒度可重构加速器在处理循环时的编译时间与面积效率(单位面积的性能).在硬件方面将处理单元内部的寄存器堆结构优化,用旁路互联的方式替代.软件方面基于这种结构提出了一种新颖,高效的循环映射算法.该算法相对于同期的研究算法,极大的缩小了搜索最优解决方案的空间.利用前向贪婪和反向回溯迭代运行,可以获得快速而又稳定的编译时间,同时保证了接近最优解的性能.在上述硬件与软件协同的解决方案下,架构的面积与计算效率得到了提升.实验数据显示,将本文的编译框架与最新技术比较,编译速度可提升 1955 倍,面积效率提升到 1.36 倍.

关键词: 粗粒度可重构架构;面积高效;映射算法;回溯

中图分类号: TP312

文献标识码: A

文章编号: 1000—7180(2019)08—0045—04

A fast and efficient compiler framework for CGRAs

YIN Wen-zhi, ZHAO Zhong-yuan, MAO Zhi-gang, WANG Qin, SHENG Wei-guang

(School of Electronic Information and Electrical Engineering, Shanghai JiaoTong University, Shanghai 200240, China)

Abstract: This paper provides a hardware and software co-design technique to optimize the compilation time and area efficiency of loop acceleration on Coarse-Grained Reconfigurable Architectures (CGRAs). From the hardware prospective, the architecture of each processing element (PE) is optimized by replacing register file within PE with bypass logic. On the software side, we develop a novel and efficient loop mapping algorithm, which greatly shrinks the search space for the optimal solution. The iteration of forward greedy placement and backward recovery obtains a fast and stable compile speed and guarantees performance close to the optimal solution. This hardware and software co-design method improves the area and computation efficiency. Experiment result shows that our framework improves 1955x in compile speed and obtains a 1.25x area efficiency.

Key words: CGRA; area efficient; mapping; backtrack

1 引言

近些年来,粗粒度可重构架构(CGRA)由于其兼具灵活性和高效性的特点而获得了学术界和工业界的广泛关注,被用来加速图像和视频处理应用^[1],深度神经网络^[2],分布式计算系统^[3]等,是一种适合数据和计算密集型应用的计算平台.

典型的 CGRA 由若干处理单元(PE)按一定的

拓扑结构连接组成.编译器将应用的循环内核映射到 PE 阵列上,生成配置信息.优秀的映射算法能带来较高的循环执行效率和应用加速比.目前,有很多关于循环映射方法的研究工作^[4-8]. REGIMap^[6]利用 PE 内部的本地寄存器堆,将映射问题转变成寻找最大子团的问题,获得了不错的性能.但是本地寄存器堆的引入导致面积效率不高,最大子团搜索的时间复杂度也较大.

收稿日期:2018-11-09;修回日期:2018-12-12

基金项目:国家自然科学基金项目(61176037,61201059)

本文采用优化后的 CGRA 结构,并针对该结构提出了一种新颖的循环映射算法,能在极短的编译时间内获得较优性能,从而提高 CGRA 的面积效率.

2 相关技术

2.1 CGRA 架构

图 1(a)展示了 CGRA 的系统框图,该架构由指

令存储器、数据存储器、指令控制器和处理单元阵列组成.图 1(b)为 PE 的内部结构,每个 PE 包含输入端口,算术逻辑单元,输出寄存器、本地寄存器堆和指令寄存器. PE 间以二维圆环的方式连接,每个机器周期 PE 从指令寄存器中读取配置信息然后执行.图 1(c)为本文优化后的 PE 结构,去掉了本地寄存器堆,节省了面积,在一定的面积预算下能容纳更多的 PE,获得更强的计算能力.

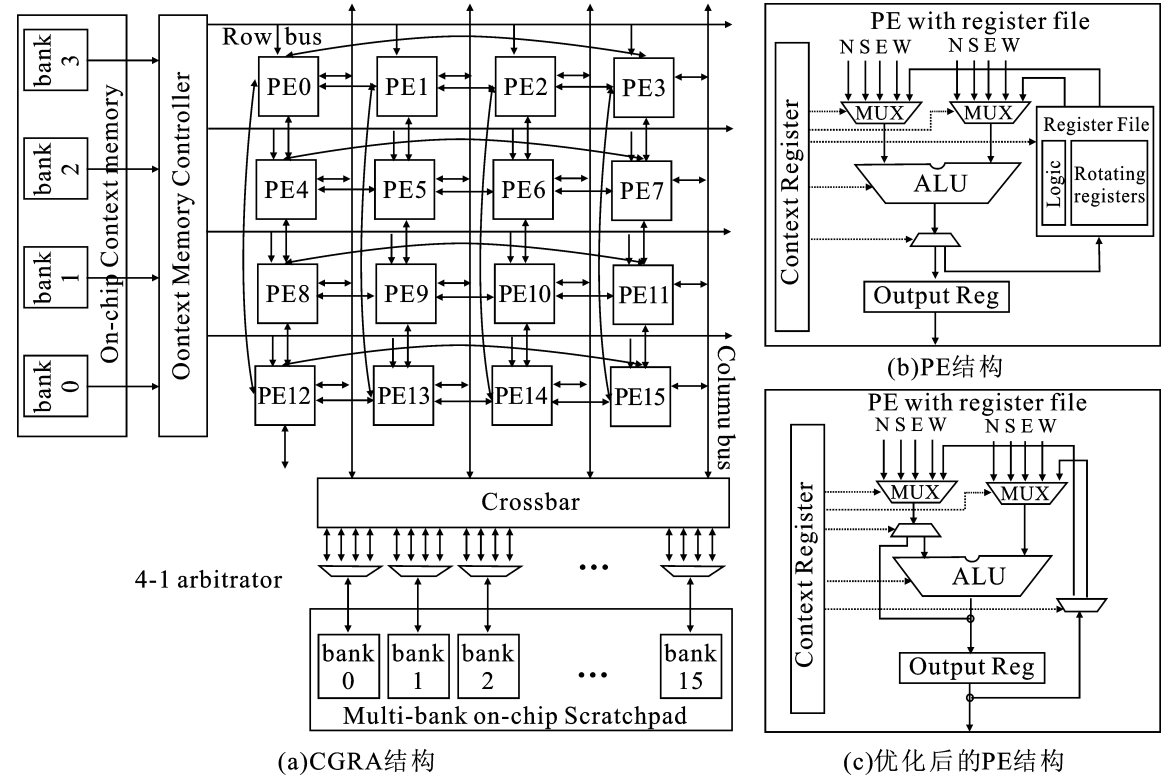


图1 CGRA 结构图

2.2 编译技术

编译器提取应用中的循环内核,将循环体转变成数据流图,然后将算子映射到 PE 阵列上.每一次循环迭代以流水线的形式组织,以一个固定的时间间隔依次启动,这个间隔称为循环启动间隔(II). II 越小,循环执行结束的时间就越短.编译器优化的目标是在合理的编译时间内最小化 II 的值.

3 循环映射算法

粗粒度可重构架构的映射是一个 NP 完全问题^[5].在面大尺寸的时间扩展 CGRA 图和复杂的循环内核时,由于搜索空间的指数增长,基于最大公共子图^[9]的算法会变得非常得慢.一般性的贪婪搜索算法虽然快,但不能保证优秀的性能.本文提出的映射算法在调度阶段优化了节点数量,在放置阶段

结合前向贪婪和反向回溯的技术,编译速度快且性能优异.

3.1 调度

本算法采用基于迭代的模调度算法^[10],首先计算理想的最小循环启动间隔(MII),以 MII 为 II 初值进行模调度.算子调度如图 2. 输入数据流图(DFG)中的每个算子都被调度在一个对 II 取模后的时间片上,每个节点都应在其输入节点之后的时间片上发射.对于间隔多个周期发射的节点对,需要插入额外的路由节点以保持正确的数据依赖关系,这一改变称为平衡^[5].之后,采取资源共享技术^[4,8]将两个具有相同源节点的路由节点合并成一个节点,减少路由节点以节省资源.图 2(a)是一幅 5 个节点的原始 DFG,图 2(b)是平衡后的 DFG,在周期 0 的节点 a 和周期 2 的节点 b、c 之间分别插入了一

个路由由节点 a_r . 图 2(d)是资源共享后的 DFG,两个路由由节点合并成了一个,节点数量减少,占用更少的资源.

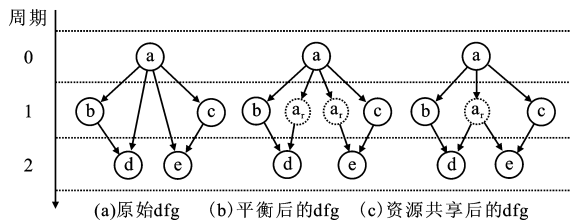


图 2 算子调度

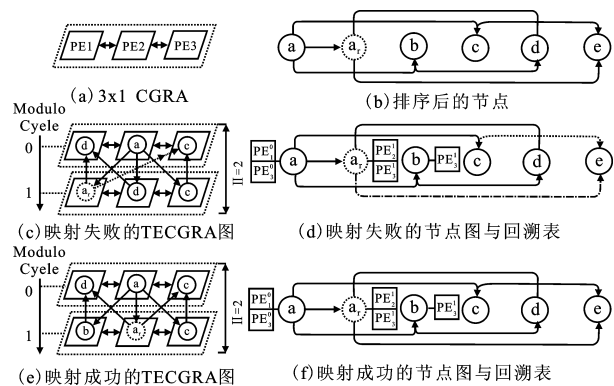


图 3 放置与互连过程

3.2 放置与互连

调度之后,II 被提取出来,数据流图中每个算子的发射时间确定.将 CGRA 资源图在时间片上复制 (II-1) 次,得到时间扩展的 CGRA 图 (TECGRA). 图 3(a) 是一副大小为 3×1 的 CGRA 资源图,图 3(c) 是其在 $II=2$ 下扩展后的 TECGRA 图. 令 $D_s = (V_D, E_D)$ 表示调度后的数据流图, V_D 为算子节点, E_D 为算子节点之间的边,反映数据依赖关系. $R_{II} = (V_R, E_R)$ 表示指定 II 下的 TECGRA 图, V_R 为 PE 节点,代表计算资源, E_R 为各 PE 节点之间的边,反映 PE 之间的互连结构. $P = (V_D, V_R)$ 是一个放置对,表示 D_s 的算子 V_D 被放置到 R_{II} 的资源节点 V_R 上. 为所有算子节点都找到一个放置对 P ,且 D_s 中每条边 E_D 都在 R_{II} 上有对应的边 E_R ,这就是循环内核的映射过程.

本文提出的映射策略由前向贪婪和反向回溯两部分组成.前向贪婪放置在编译时间上有优势,能在较短的时间内解决放置问题.反向回溯在前向放置失败时寻找一条新的映射路径,有利于在较低的 II 下映射成功,提高编译性能.

3.2.1 前向贪婪放置

在图 D_s 中,指向节点 V_D 的边的数量称为 V_D 的

入度,由 V_D 指出的边的数量是 V_D 的出度.入度和出度反映了节点与其相邻节点之间的约束.本算法选择具有最大约束的节点作为根节点 $V_{D_{root}}$,采用深度优先搜索 (DFS) 算法对 D_s 中的节点进行排序,然后按顺序依次向前映射.该拓扑顺序保证了除根节点外,每个节点在被放置时,至少有一个相邻节点已经放置.图 3(b) 展示了对图 2(c) 所示数据流图算子节点进行 DFS 排序后的结果.

算子节点在放置时可能有多个有效的候选槽 V_R ,根据公式 (1) 计算所有候选槽的优先级.式中 N_R 表示该候选槽 V_R 通过连接所有未被使用的 PE 节点而访问到的 PE 槽的数量总和, N_D 表示与 V_D 有连接的还未放置节点的数量.如果 N_R 小于 N_D ,则放置 V_D 到 V_R 上将无法满足后续节点的资源需求,该槽优先级为零.如果 N_R 等于或大于 N_D ,则资源数量越接近后续节点数量,该候选槽的优先级越高.该选择方案计算简单,不需要花费大量的时间,且能最大化资源的利用率,给后续的节点留下尽可能多的资源,有效地提高放置的成功率.

$$\text{Priority} = \begin{cases} 0, & N_R < N_D \\ \frac{N_D}{N_R}, & N_R \geq N_D \end{cases} \quad (1)$$

3.2.2 反向回溯

前向贪婪放置并不总是能生成有效的映射结果,尤其是当 II 较小或数据流图较复杂时.为了提高映射的成功率并获得一个更好的性能,本文提出了一种反向回溯的算法,该算法利用一种称为回溯表的数据结构.在前向放置中,每次从多个候选槽中进行选择时,回溯表会保存放置该数据流图节点时其他未被选择的槽.反向回溯由两步组成:反向恢复和更新.

当前向放置在节点 V_{D1} 失败时,回溯到最近的 V_{D1} 已放置的相连接的节点 V_{D2} ,如果回溯表中 V_{D2} 没有其他的槽,则回溯到距离 V_{D1} 第二近的节点.对于回溯节点和 V_{D1} 之间的所有节点,清除回溯表中对应的记录.这一步就是反向恢复.记 PE_i^t 为 TECGRA 图中 t 时刻的第 i 个 PE 槽.图 3(c) 为前向放置失败时的 TECGRA 图,如图中虚线所示,放置最后一个节点 e 时,只有 PE_3^2 空闲,而 e 的前驱节点 a_r 所在的 PE_{11}^1 槽与 PE_3^2 无连接, a_r 与 e 的数据依赖无法满足,映射失败.图 3(d) 展示了当前映射下的回溯表,此时回溯点为节点 c (图中虚线),但 c 没有其他候选槽.继续回溯到节点 a_r (图中点划线),其候选槽有两个,于是清空 a_r 和 e 之间所有节点在回

溯表中的记录.

反向恢复后,查找回溯表,改变回溯节点的位置到表中优先级最高的槽 V_{R2} ,并将 V_{R2} 从表中删除.之后继续放置回溯点之后的节点,重新建立回溯表.这一步称为更新,保留了 V_{R2} 之前节点的映射结果,为 V_{R2} 之后的节点提供了一条新的映射路径.交替重复前向与反向过程直到所有节点映射完成.若已经回溯到根节点,而根节点的回溯表为空,则增加II,重新调度和映射,直至映射成功.如图3(e)所示,将回溯点 a_r 映射到回溯表中第一个候选槽 PE_2^r 后,继续前向映射,最后所有节点均映射成功.此时的回溯表为图3(f)所示,找到了最优性能的映射方案.

4 实验结果

为了评估本文的编译框架的性能和效率,软件上,从不同的 Benchmark 中选取了 8 个不同应用领域的循环内核,分别采用本文算法与 REGIMap^[6]进行映射;硬件上,使用 RTL 代码实现了图 1 中两种不同的 CGRA 架构,其中 PE 阵列规模为 4x4,本地寄存器堆大小为 4,并使用 DesignCompiler 工具在 32 nm 工艺下进行综合.

表 1 循环内核映射性能与编译时间

循环内核	节点数量	本文算法			REGIMap		
		II	IPC	编译时间(s)	II	IPC	编译时间(s)
aes3	120	8	15	0.362 4	8	15	114.56
bezier1	41	3	13.67	0.005 6	3	13.67	13.18
conven1	41	3	13.67	0.028 9	3	13.67	9.430 7
fft	92	8	11.5	0.024 7	—	—	—
gemm1	22	2	11	0.003 2	2	11	0.838 6
ifft5	70	5	14	0.011 7	6	11.67	0.603 4
md5	47	10	4.7	0.298 4	12	3.92	1 273.57
unstructured2	23	3	7.67	0.012 1	2	11.5	1.398 8
平均值			11.39	0.103 2		11.49	201.94

表 1 列出了每个循环内核的算子节点数量,以及采用本文算法和 REGIMap 分别对循环内核进行编译后的性能和编译时间.对于 fft,REGIMap 映射失败,本文算法映射成功率更高,适用于更广泛的应用.将循环内核算子数量除以 II 来估计每个内核编译后的每周周期指令数(IPC).不纳入 FFT,本文算法的平均 IPC 为 11.39,REGIMap 为 11.49,说明即使在缺少本地寄存器优势的情形下,本文算法的回溯机制仍能带来与 REGIMap 一致的编译性能.对

于 aes3、md5 这类节点数量多或 DFG 图复杂的循环内核,本文算法均能在稳定且少量的时间内完成.而 REGIMap 由于采用的是最大子团搜索算法,在面对此类应用时搜索空间指数增长,编译速度慢.本文算法的平均编译时间为 0.1032 s,而 REGIMap 为 201.94 s,本文算法获得了 1 955 倍的速度提升.

表 2 不同框架的面积与效率

框架	面积(mm ²)	平均 IPC	面积效率(IPC/mm ²)
本文框架	0.154 3	11.39	73.82
REGIMap	0.211 5	11.49	54.33

表 2 为本文框架与 REGIMap 的面积与面积效率.优化 PE 结构,去掉寄存器堆后,本文框架的 CGRA 在面积上节省了 27%,以实现每 IPC 所占用的面积来衡量面积效率,本文框架在面积效率上提升了 36%.

5 结束语

本文采用软硬件协同的技术,针对粗粒度可重构架构提出了一种快速高效的编译框架,简化了处理单元的结构,节省了面积,并基于该简化结构提出了一种新颖的循环映射算法.算法结合了前向贪婪和反向回溯两种思想,保证了编译时间的稳定和性能的优异.实验证明,本文框架与当前先进的 REGIMap 框架相比,编译速度提升了 1955 倍,面积效率达到了 1.36 倍.

参考文献:

[1] EGGER B, LEE H, KANG D, et al. A space-and energy-efficient code compression/decompression technique for coarse-grained reconfigurable architectures [C]//2017 IEEE/ACM International Symposium on Code Generation and Optimization (CGO). Austin: IEEE, 2017: 197-209.

[2] CHEN Y H, EMER J, SZE V. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks [C]//2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA). Seoul: IEEE, 2016: 367-379.

[3] GAO M, KOZYRAKIS C. HRL: Efficient and flexible reconfigurable logic for near-data processing[C]//2016 IEEE International Symposium on High Performance Computer Architecture (HPCA). Barcelona: IEEE, 2016: 126-137.

- al. A Novel Four-Dimensional Hyperchaotic Four-Wing System With a Saddle-Focus Equilibrium[J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2017, 64(3):339-343.
- [4] XUE H W, DU J, LI S L, et al. Region of interest encryption for color images based on a hyperchaotic system with three positive Lyapunov exponents[J]. Science Direct, 2018, 106(1):506-516.
- [5] WANG Z, SUN C, JIN F, et al. A widely amplitude-adjustable chaotic oscillator based on a physical model of HP memristor[J]. IEICE Electronics Express, 2018, 15(8):1-6.
- [6] ZENG T, CHANG S Q, FAN H Y, et al. Design and processing of a novel chaos-based stepped frequency synthesized wideband radar signal[J]. Sensors, 2018, 18(4):1-23.
- [7] 梅蓉, 陈谋. 一类超大范围超混沌系统的动力学分析和电路实现[J]. 工程科学与技术, 2012, 44(5): 171-175.
- [8] ZHONG G, MAN K F, CHEN G R. A systematic approach to generating n-scroll attractors [J]. International Journal of Bifurcation & Chaos, 2002, 12(12):2907-2915.
- [9] TLELO-CUAUTLE E, PANO-AZUCENA A D, RANGEL-MAGDALENO J J, et al. Generating a 50-scroll chaotic attractor at 66 MHz by using FPGAs [J]. Nonlinear Dynamics, 2016, 85(4):1-15.
- [10] LAI Q, ZHAO X W, RAJAGOPAL K, et al. Dynamic analyses, FPGA implementation and engineering applications of multi-butterfly chaotic attractors generated from generalised Sprott C system[J]. Pramana, 2018, 90(1):6-17.

作者简介:

周 围 男, (1971-), 博士, 教授. 研究方向为电路与系统、通信系统及信号处理等.

吴周青(通讯作者) 男, (1991-), 硕士研究生. 研究方向为混沌理论及其应用研究. E-mail: cejin@foxmail.com

(上接第 48 页)

- [4] CHEN L, MITRA T. Graph minor approach for application mapping on CGRAs[C]//2012 International Conference on Field-Programmable Technology. Seoul: IEEE, 2012: 285-292.
- [5] HAMZEH M, SHRIVASTAVA A, VRUDHULA S. EPIMap: Using epimorphism to map applications on CGRAs[C]// DAC Design Automation Conference 2012. San Francisco: IEEE, 2012: 1280-1287.
- [6] HAMZEH M, SHRIVASTAVA A, VRUDHULA S. REGIMap: register-aware application mapping on coarse-grained reconfigurable architectures (CGRAs) [C]//Proceedings of the 50th Annual Design Automation Conference. Austin: ACM, 2013: 18.
- [7] DAVE S, BALASUBRAMANIAN M, SHRIVASTAVA A. RAMP: resource-aware mapping for CGRAs[C]//2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC). San Francisco: IEEE, 2018: 1-6.
- [8] ZHAO Z, SHENG W, JING N, et al. Resource-saving compile flow for coarse-grained reconfigurable architectures [C]//2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig). Cancun: IEEE, 2015: 1-8.
- [9] LEVI G. A note on the derivation of maximal common subgraphs of two directed or undirected graphs[J]. Calcolo, 1973, 9(4): 341-352.
- [10] RAU B R. Iterative module scheduling: An algorithm for software pipelining loops[C]//The 27th Annual IEEE/ACM International Symposium on Microarchitecture. San Jose: IEEE, 1994: 63-74.

作者简介:

尹文志 男, (1994-), 硕士研究生. 研究方向为粗粒度可重构处理器编译器后端技术. E-mail: mrwensim@sjtu.edu.cn

赵仲元 男, (1989-), 博士研究生. 研究方向为可重构计算的编译架构.

毛志刚 男, (1962-), 教授. 研究方向为可重构硬件结构设计技术研究.

王 琴 女, (1975-), 副教授. 研究方向为大规模集成电路设计、先进集成电路设计方法.

绳伟光 男, (1977-), 助理研究员. 研究方向为高可靠集成电路设计.