

引用格式:李锦超,赵仲元,绳伟光. 基于相似性的粗粒度可重构指令压缩[J]. 微电子学与计算机,2020,37(8):21-26. [LI J C,ZHAO Z Y,SHENG W G. Similarity aware context reduction for coarse grained reconfigurable architectures[J]. Microelectronics & Computer,2020,37(8):21-26.]

基于相似性的粗粒度可重构指令压缩

李锦超,赵仲元,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:粗粒度可重构架构在能效比方面具有明显优势,然而其指令存储与传输过程的功耗代价过高.实验发现指令间具有明显的相似性,由此本文提出一种基于指令相似性的压缩技术,通过对指令的压缩、传输与解压,可以在不降低性能的前提下,优化架构的功耗和面积.针对同构和异构平台分别提出了指令分发模型和指令寄存器模型的解决方案,结合编译策略优化,最终与两种传统结构相比,面积效率比分别提升 36%和 181%,功耗效率比分别提升 33%和 118%.

关键词:粗粒度可重构架构;指令压缩;能效比;异构计算

中图分类号: TP303

文献标识码: A

文章编号: 1000—7180(2020)08—0021—06

Similarity aware context reduction for coarse grained reconfigurable architectures

LI Jin-chao,ZHAO Zhong-yuan,SHENG Wei-guang

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

Abstract: Coarse-Grained Reconfigurable Architectures (CGRAs) are promising computing platform with better efficiency. However, too much cost is wasted on the path of context storage and transmitting. With statistics of real case, there is obvious similarity between context. We propose one kind of context reduction method based on similarity of context. By making context compress, transmit, and decompress, this method can reduce power and area of CGRA obviously, with little overhead on performance. We propose context dispensing and register model for homogeneous heterogeneous platform. With the optimize of compiler, we get 33% and 118% promotion on power efficiency to tradition architecture, 36% and 181% on area efficiency.

Key words: CGRA; context reduction; power efficiency; heterogeneous computing

1 引言

粗粒度可重构架构(Coarse-Grained Reconfigurable Architecture, CGRA)被视为一种在后摩尔定律时代,平衡功耗与性能矛盾的优秀方案^[1],其在能效比方面相比其他架构,具有明显优势.近年来针对CGRA的主要研究热点是拓展其应用领域^[2]、进一

步提升性能和降低功耗^[3].

根据 Hycube^[4] 中的数据指出,CGRA 指令的存储与配置功耗占到总功耗的 40%.针对这一问题,国内外已有部分研究关注到如何优化 CGRA 指令的配置功耗.通过将指令缓存整合入处理单元内部,并利用指令共享和流水线技术可以减小 MIMD 结构的指令传输功耗^[5-6].指令内部各部分具有一定

的相关性,利用指令内部的相关性可以实现对指令的动态压缩^[7~9],并在运算前解压. DNestMap^[10]则是通过映射算法优化,来减少循环应用中指令切换的配置代价.

本文旨在提出一种利用指令间相似性的压缩方法,在保证系统性能的同时,减少控制通路的功耗代价.

2 CGRA 架构及指令特征

2.1 CGRA 架构组成

典型的 CGRA 架构如图 1 所示^[1]. 其具体结构有:共享数据存储器,全局指令存储器,指令存储器控制器和处理单元阵列. 在这个模型基础上,不同结构设计了多种处理单元阵列和共享数据存储器或全局指令存储器的互联方式. 不同的互联方式,极大的影响整个结构的性能与功耗.

同时,又根据处理单元阵列中处理单元是否相同,CGRA 可以分为同构平台和异构平台两类. 同构平台中,所有处理单元均具有运算和 Load/Store 能力,因而指令集设计具有通用性. 指令间切换由于存在普通运算指令向 Load/Store 指令切换的可能,指令翻转相对较多. 异构平台中,将运算与 Load/Store 分离,由 LSU(Load/Store Unit)执行 Load/Store 操作,由 ALU(Arithmetic and Logic Unit)执行其他算术操作. 因而,异构平台下可针对 ALU 和 LUS 指令特点,分别进行针对性的优化.

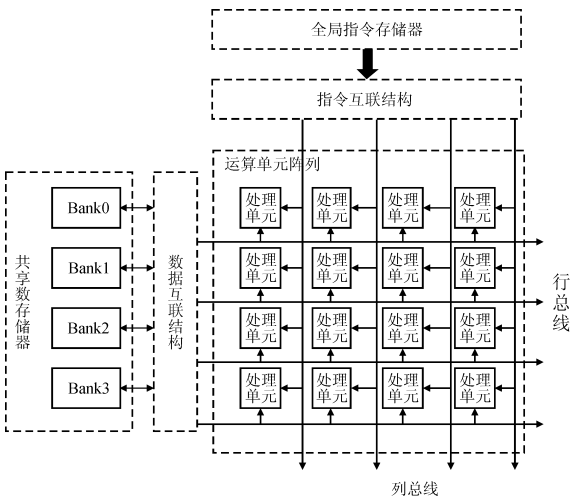


图 1 典型 CGRA 结构图

2.2 指令特征

CGRA 指令特征:

(1)在其运行过程中的指令平均翻转率仅为 11.79%,说明从一条指令到下一条实际的变化比特

数占比并不高,指令之间存在某种程度的相似性. 这种相似性可以体现为前后两条指令的输入来源的相关,立即数变化范围较小等.

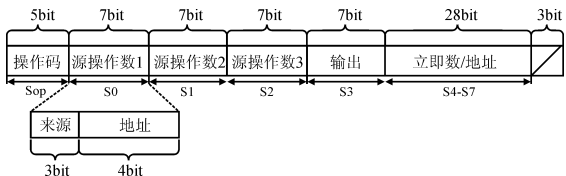
(2)指令内部具有冗余信息,对于一条具体的指令,指令可能没有完全利用全长度的位宽. 部分字段在某一具体指令中并未使用,但传输过程会产生代价.

(3)指令间的相似性可以通过改进编译器的映射策略得以提高. 基于以上特点,我们可以利用指令间的相似性进行压缩,减小控制通路的代价.

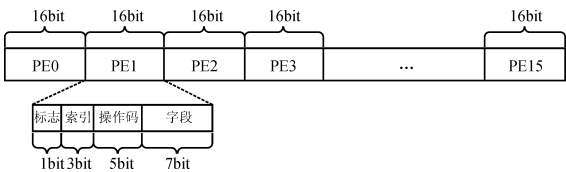
3 指令相似性压缩

3.1 指令集设计

本文未压缩基础指令集如图 2(a)所示. 考虑到通用性和可扩展性的要求,基础指令长度固定为 64 位,共由以下几部分组成:5 位的操作码字段,用来表示本条指令的具体操作类型;3 个源操作数字段,每个字段 7 位,用来表示 3 个源操作数(其中 3 位表示输入来源:立即数、共享数据存储器、全局寄存器、本地数据寄存器、本地或其他相联处理单元的输出生存器;4 位表示具体地址索引);输出也用 7 位表示(3 位表示输出去向:共享数据存储器、全局寄存器、本地数据寄存器或输出寄存器;4 位表示输出的具体地址索引);28 位的立即数/地址字段,用来表示操作数是立即数时的值或者 load 和 store 指令直接寻址时的地址.



(a) 基础指令集



(b) 压缩指令结构

图 2 基础指令集与压缩指令

3.2 指令分割及压缩

实践中我们发现,指令内部不同部分具有明显的相关性. 例如一条指令中源操作数的来源字段发生变化时,地址索引字段通常也会发生变化,故将其组合成一个 7 位的字段. 同样的,将其他部分也按照

7 位一个字段的进行分割,操作码与一条指令具有强耦合性,单独分割为一个字段 SOp. 如图 2 (a)所示,64 位的指令共分割为 S0 - S7 和 SOp 九个字段.

根据分割的字段,我们将处理单元的 SOp (5 位)以及相对上一条指令改变字段中的一个(7 位),打包入一个包中. 由于相对于上一条指令,本条指令可能有多个字段发生改变,故在包中加入 1 位的标志位表示本次传输是否为最后一个改变字段. 同时,因为共有 S0 - S7 八个字段,则需要一个 3 位的索引位作为本次传输的字段索引. 因此,一个包的长度为 16 位,共包括 1 位传输标志位,3 位索引位,5 位操作码和 7 位的一个改变字段. 一条压缩后的指令由 n 个包组成, n 为处理单元阵列中处理单元的个数如图 2(b)所示. 当发生一个处理单元有多个字段发生改变的情况时,需要多次传输,直到所有传输标志位置位时表示一次传输结束.

3.3 指令优化及映射

我们可以通过有效的指令优化和映射算法,显著的减少相邻指令改变的字段个数. 具体优化策略有以下几种.

(1)每条指令内部具有一定的冗余信息,例如对于 load 操作,源操作数 2 和 3 均为冗余信息,编译器可以通过将这些字段配置成与上一条或者下一条指令相同的信息,来减少改变的字段数.

(2)在处理单元映射过程中,可以在性能相同的几种配置方案中,选择改变字段数较少的结果. 如图 3 所示将图 3(b)所示算法,映射到图 3(a)所示 2×2 CGRA 阵列上,成功映射方案中的两种如图 3(c)和图 3(d)所示. 启动间隔表示多条指令循环执行时,一条指令在两次迭代间的间隔. 作为性能衡量的指标,两种方案的启动间隔均为 2,说明性能相同. 如图 4 所示,两种方案改变字段已用虚线框标出. 对于映射方案 1,处理单元 2 除操作码字段(SOp)外,还需要传输 2 个字段(S0 和 S3);但对于映射方案 2,除操作码字段外,每个处理单元最多只有一个字段发生改变,故方案 2 更优. 针对不同的应用,映射算法大多具有明显的优化空间.

(3)对于空操作(NOP)除操作码字段外,均为冗余信息,编译器可以自行配置其值,来减少改变的字段数. 在配置过程中,需充分考虑其上下两条指令的环境. 当 NOP 上下两条指令间有两个字段不同时,相比于将该条空操作配置为与前一条完全相同的方案,将其配置为与上下两条均有一个字段差异

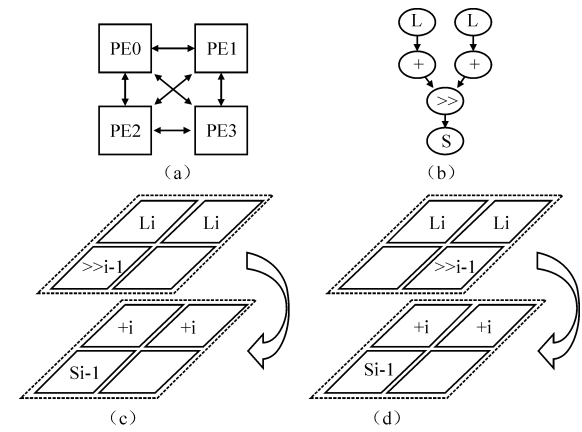


图 3 映射优化示例

	SOp	S0	S1	S2	S3	
Mapping1	L	fm	xxx	xxx	xxx	out
	+	PE	self	fm	0	xxx
	>>	PE	up	PE	diag	xxx
	S	PE	self	xxx	xxx	lm
Mapping2	L	fm	xxx	xxx	reg	xxx
	+	PE	self	fm	0	xxx
	NOP	xxx	xxx	xxx	xxx	xxx
	S	PE	right	xxx	xxx	lm

图 4 映射结果选择

的方案是更好的选择. 其他指令也尽可能配置为平衡上下两条指令的形式.

4 硬件结构设计

4.1 传统结构

传统 CGRA 的控制通路可分为有本地指令寄存器和无本地指令寄存器两种模型.

对于有本地指令寄存器的模型,处理单元阵列在开始运行前将一次任务需要的多条指令通过指令互联结构读入处理单元本地的指令寄存器,寄存器位宽为 64 位. 运行时,处理单元仅需从本地指令寄存器中提取指令. 一次传入可执行多次循环,均摊后传输的性能代价可忽略,故互联结构采用功耗损失较小的单总线模式. 由于在取指时没有性能损失,该模型具有最高的性能,将其作为我们的性能基准. 但是,本地指令寄存器给这个模型引入了很大的面积和功耗开销.

对于无本地指令寄存器模型,处理单元阵列在每次运行前,先从全局指令存储器中提取本次运行的指令并分发给各个处理单元,再启动运行. 由于避免了本地指令寄存器,该模型具有明显的功耗优势,但由于每次运行前的取指存在冲突,影响了整体性能. 考虑到阵列规模的可扩展性和性能与功耗的平

衡,互联结构采用能效比较高的列总线方案作为功耗基准.

4.2 面向同构的指令分发模型

图 5 为 4×4 规模指令分发模型的控制通路指令示意图.

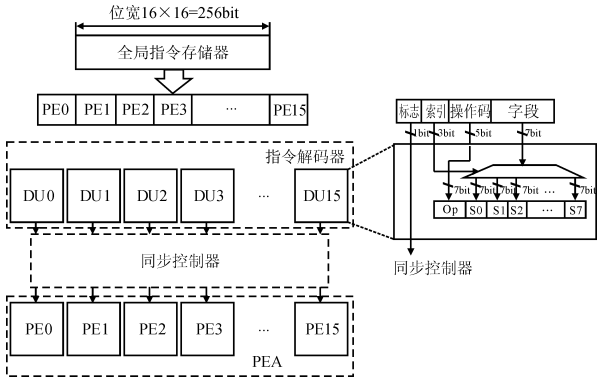


图 5 指令分发模型示意图

结构从上往下分别为:

(1)全局指令存储器. 位宽为 16×16 bit, 一次传输包含 16 个包, 对应 16 个处理单元.

(2)指令解码器. 接收同步器控制器发出的取指使能信号, 从全局指令存储器中提取指令, 将其按照指令相似性压缩规则解码, 修改对每个处理单元的 16 个指令寄存器, 循环读取配置信息直到读到的所有包中的传输标志位均有效, 则表示一次配置信息取指完成, 发送取指完成信号给同步控制器, 同时进入等待, 等待取指使能信号再次有效.

(3)同步控制器. 关键控制模块, 用于控制解码器取指和阵列运行. 接收解码器发出的取指完成信号, 处理单元阵列发出的运行完成信号. 当检测到两个信号均有效时, 向处理单元阵列发出执行使能信号, 处理单元在接到执行使能信号后首先提取暂存在解码器中对应寄存器的指令, 然后开始运行. 在执行使能信号发送后的一个时钟周期, 同步控制器向解码器发送取指使能信号, 开启下一条指令的提取解码过程.

(4)处理单元阵列. 由 16 个处理单元组成, 接收同步控制器的执行使能信号, 当其有效时从解码器中提取对应指令并运行, 运行完成时发送运行完成信号到同步控制器.

在指令分发模型中, 处理单元阵列的执行过程和指令解码器的取指过程形成了两级流水结构, 由于指令相似性压缩的优化, 减少了取指所用的时钟周期, 取指过程的性能损失基本可以被执行过程掩盖. 此时, 阵列性能将接近 4.1 节中提到的最好性能

基准. 同时又由于没有本地指令寄存器, 指令压缩后全局指令存储器容量减小, 该结构的在功耗和面积方面也具有明显优势.

4.3 面向异构的指令寄存器模型

相比同构平台, 异构平台有着特有的特点.

(1)LSU 与 ALU 指令具有完全不同的特点, 针对同构平台的优化, 在异构平台的 ALU 单元上同样具有良好效果. 而对于 LSU 指令, 异构平台可以进行采用进一步针对性的优化策略.

(2)为进一步提高 LSU 在循环迭代等应用需求下的执行能力, 异构平台下我们将 Load/Store 部分冗余字段加以修改利用, 以改善循环边界跳转情况下的运行性能.

(3)考虑到针对性的优化 LSU 指令后, 指令解码及分发方式将不再相同, 故在异构平台我们选择采用本地指令寄存器作为指令传输模式.

结合 LSU 以上特点我们将异构平台 LSU 指令集及压缩格式优化为如图 6 所示. 将原源操作数 2, 3 字段配置为偏移量字段, 将空闲的三位配置为循环控制字段. 其压缩格式做相应修改, 原操作码字段修改为 2 位的操作码和 3 位的循环控制字段.

2 位操作码. 由于在 LSU 中存储的指令只有 Load、Store 和空操作三种, 故可以将指令的操作码压缩为 2 位.

14 位偏移量字段. Load/Store 操作中存在基址偏移的寻址方式, 加入偏移字段可有效缓解同构平台中存在的不断修改常数字段来满足需求的情况. 偏移字段最低位为符号位, 支持正负偏移. 根据源操作数类型支持相对源操作数基址偏移和相对上一条地址偏移.

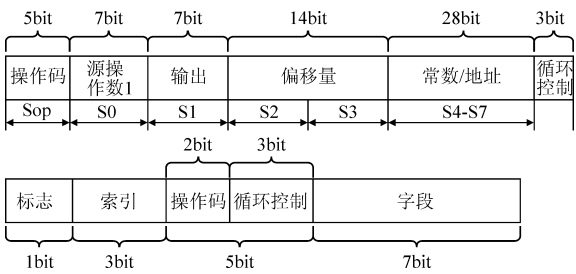


图 6 LSU 优化指令

3 位循环控制 (LC). 本文通过引入循环控制字段来减少指令切换. 循环控制字段由 2 位的偏移量标志位, 和 1 位立即数标志位组成. 2 位偏移量标志位中, 00 表示本条指令不叠加偏移量, 01 表示叠加本地偏移寄存器中的偏移量, 10 表示叠加指令中的偏移量, 11 表示叠加指令中的偏移量并将其写入本

地偏移寄存器.立即数标志位表示立即数是否要写入基址寄存器.

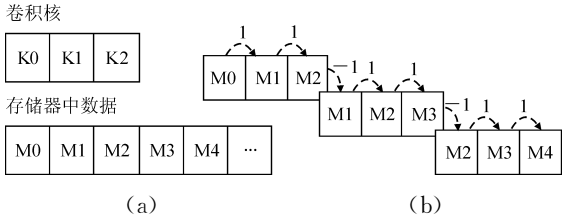


图 7 卷积应用示例

```
For(i= 0; i<N; i++)
  For(j= 0; j<3; j++)
    Result+= Mem[i+j]x kernel[j];
```

图 8 卷积代码

考虑如图 7(a)所示卷积应用示例,1×3 的卷积核对顺序存储在数据存储器中的数据进行卷积.图 7(b)所示为 LSU 读取存储器中数据的顺序,箭头所示为相对上一次地址的偏移量.循环代码如图 8 所示.

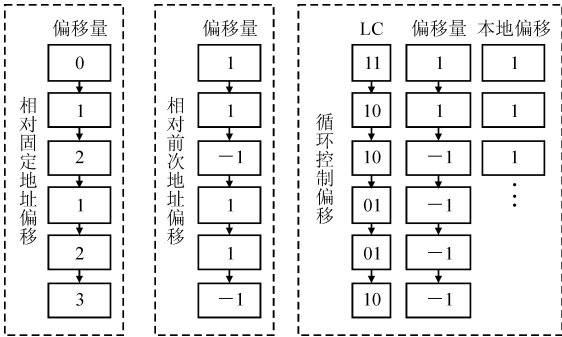


图 9 LSU 寻址方式

LSU 可采用图 9 所示三种方式读取数据.方式 1,相对立即数偏移,每次 Load 偏移字段均要改变;方式 2,相对前次指令地址偏移,在每次内层循环边界处偏移字段发生两次改变;方式 3,增加循环控制字段后的相对偏移,由于 LC 字段跟随操作码每次均要传输,仅需考虑偏移量字段改变,整个循环仅发生一次改变,需传输的字段明显减少.

5 实验结果

针对两种传统结构,和指令压缩同构异构两种模型,本文利用模拟器做出性能分析,基于 65 nm 工艺库利用 Design Compiler 及 PTPX 做出功耗和面积分析.以性能/面积作为面积效率衡量标准,以性能/功耗作为功耗效率衡量标准.

实验结果如图 10、图 11 所示.图中 A 表示传统有本地指令寄存器模型,B 表示传统无本地指令寄

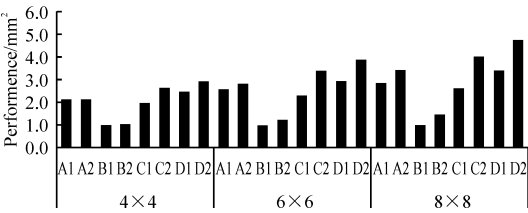


图 10 面积效率比

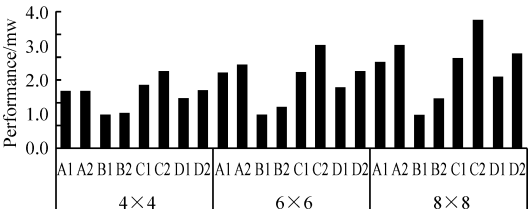


图 11 功耗效率比

存器模型,C 表示相似性压缩指令分发模型,D 表示相似性压缩指令寄存器模型. A1,A2 分别表示处理单元互联能力为距离为 1 和 2 的情况. B,C,D 情况相同. 4×4,6×6,8×8 分别代表阵列规模为 4,6,8 的情况,每种阵列规模下,数据都归一化到 B1.

实验结果表明,指令相似性压缩在不同阵列规模情况下都有明显的面积和功耗效率比提升. 分别来看,D 方案面积效率更具优势,原因在于采用本地指令寄存器将减少对全局指令存储器的位宽需求. C 方案功耗效率更具优势,原因在于 D 方案在实现时需要在处理单元本地增加状态机等控制逻辑模块,相比下 C 方案的集总控制模式更具功耗优势. 平均看来相比传统的两种 CGRA 结构,我们的方案(C 方案)具有 36%和 181%的面积效率比提升,33%和 118%的功耗效率比提升.

6 结束语

本文从实际应用角度出发,挖掘 CGRA 指令间具有的相似性,并提出了利用相似性的指令压缩及优化方法. 同时,针对不同平台的各自特点,设计了面向同构平台的指令分发模型和面向异构平台的指令寄存器模型,改善了 CGRA 指令配置代价过高的问题,最终取得了明显的面积和功耗效率提升. 未来将进一步在指令压缩的通用性上扩展.

参考文献:

[1] MEI B, VERNALDE S, VERKEST D, et al. ADRES: An architecture with tightly coupled VLIW processor and coarse-grained reconfigurable matrix[C] // International Conference on Field Programmable Logic

- and Applications. Springer, Berlin, Heidelberg, 2003: 61-70.
- [2] LIU D, YIN S, LIU L, et al. Polyhedral model based mapping optimization of loop nests for CGRAs[C]// Proceedings of the 50th Annual Design Automation Conference. ACM, 2013: 19.
- [3] YIN W Z, ZHAO Z Y, MAO Z G, et al. A fast and efficient compiler frame work for CGRAs[J]. Microelectronics & Computer, 2019, 36(8): 45-48.
- [4] KARUNARATNE, MANUPA, et al. "Hycube: A cgra with reconfigurable single-cycle multi-hop interconnect." 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC). IEEE, 2017.
- [5] KIM Y, KIEMB M, PARK C, et al. Resource sharing and pipelining in coarse-grained reconfigurable architecture for domain-specific optimization[C] // Proceedings of the conference on Design, Automation and Test in Europe-Volume 1. IEEE Computer Society, 2005: 12-17.
- [6] KIM Y, MAHAPATRA R N, PARK I, et al. Low Power Reconfiguration Technique for Coarse-Grained Reconfigurable Architecture [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 17(5): 0-603.
- [7] KIM Y, MAHAPATRA R N. Dynamic Context Compression for Low-Power Coarse-Grained Reconfigurable Architecture [J]. IEEE Transactions on Very Large Scale Integration Systems, 2009, 18(1): 15-28.
- [8] ASLAM, NAZISH, MILWARD, et al. Code Compression and Decompression for Coarse-Grain Reconfigurable Architectures [J]. IEEE Transactions on Very Large Scale Integration Systems, 16 (12): 1596-1608.
- [9] EGGER B, LEE H, KANG D, et al. A space- and energy-efficient code compression/decompression technique for coarse-grained reconfigurable architectures [C] // International Symposium on Code Generation & Optimization. IEEE Press, 2017.
- [10] KARUNARATNE M, TAN C, KULKARNI A, et al. Dnestmap: mapping deeply-nested loops on ultra-low power CGRAs [C] // 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC). IEEE, 2018: 1-6.

作者简介:

李锦超 男, (1994-), 硕士研究生. 研究方向为粗粒度可重构架构设计. E-mail: littledanni@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.

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

(上接第 20 页)

- [8] CHEN Y Z, JING N F, WANG Q. Implementation and evaluation of the rocket-IO based serial DDR memory [J]. Microelectronics & Computer, 2019, 36(6): 84-87.
- [9] JAVED S, GAGGATURA, DEEPANRAJ T. A power efficient active inductor-based receiver front end for 20 Gb/s high speed serial link [J]. AEUE - International Journal of Electronics and Communications, 2019, 12(7): 110-111.
- [10] XU C, LUO Z, LI J, et al. A high-speed transceiver with optical SerDes [C] // Applied Optics and Photonics

China, Beijing, China, 2015: 79-80.

作者简介:

许晋彰 男, (1994-), 硕士研究生. 研究方向为多处理器片间高速互连. E-mail: xujinzhang@sjtu.edu.cn.

景乃锋 男, (1982-), 博士, 副研究员. 研究方向为计算机体系结构.

蒋剑飞 男, (1979-), 博士, 助理研究员. 研究方向为高速数字集成电路设计、SoC 设计与系统集成.