基于神经网络的重构指令预取机制及其可扩展架构

陈志坚,孟建熠,严晓浪,沙子岩
(浙江大学超大规模集成电路设计研究所,浙江杭州 310027)

摘 要: 针对动态可重构处理器的配置信息加载延时,提出了一种基于神经网络的可扩展的重构指令预取机制.增加感受器的历史指令信息,并结合感受器权重构建新型的感受器模型,通过权重与历史指令信息的协同训练学习重构指令调用规律.在处理器运行过程中,提前完成对后续重构指令的预测及配置信息的预取,隐藏指令重构成本.进一步提出了本方法的可扩展实现框架,神经网络的学习结果作为重构指令的关联信息,被移至内存并分布式存储.在重构指令预取时,完成对神经网络学习信息的加载.实验结果表明,该方法对重构指令的预测准确率达91%,综合性能平均提升40%.

关键词: 可重构处理器; 配置信息预取; 改进神经网络算法; 可扩展存储架构
中图分类号: TN302 **文献标识码:** A **文章编号:** 0372-2112 (2012) 07-1476-05
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.07.031

Algorithm/Architecture of NN-Based Configuration Prefetching

CHEN Zhi-jian, MENG Jian-yi, YAN Xiao-lang, SHA Zi-yan
(*Institute of VLSI Design Zhejiang University, Hangzhou, Zhejiang* 310027, *China*)

Abstract: Reconfigurable processor suffers a severely performance loss from reconfiguration overhead. A NN(neural network algorithm) based configuration prefetching algorithm was proposed in this paper to reduce the overhead. Not only the receptor weight but also the history RFUOP ID constructs an advanced receptor model. The neural network studies the RFUOP trace through collaborative training of receptor weight and history RFUOP ID. With the learning result, the neural network predicts next RFUOPs and completes configuration prefetching, overlapping the configuration loading with the computation on the host processor. Furthermore, an extensible architecture for neural network storage was proposed. As a component of RFUOP, reception information is located on off-chip memory and linked to corresponding configuration field. Once configuration was prefetched, neuron information was loaded and computed for next prediction. Experiments show that the NN-based prefetching algorithm can reach 91% prediction accuracy, while gaining performance improving by 40% on average.

Key words: reconfigurable processor; configuration prefetch; advanced neural network algorithm; extensible architecture

1 引言

面对嵌入式运算复杂度的指数级增长[1,2],传统嵌入式处理器技术已显得力不从心.可重构技术作为嵌入式处理器技术的重要发展趋势,可实现处理器运算性能的巨大提升[3].文献[4]显示相比传统处理器,可重构处理器的运算性能平均提升4.3~13.5倍.由于可重构处理器在运行过程中需要实时的载入指令配置信息,在配置信息加载期间,处理器由于无法执行重构指令而停顿并导致处理器的性能损失.指令动态重构产生的加载成本已成为阻碍可重构处理器发展的瓶颈.以DISC[4]处理器为例,其指令动态重构产生的运行延时占到处理器总执行时间的25%~71%.

目前解决重构成本的方法主要分三种:配置信息压缩、配置信息缓存和基于预测的配置信息预取.文献[5]对配置信息进行算法级压缩.文献[6]则从电路级对可重构部件进行改进以减少重构所需的配置信息大小.但这些方法对于降低重构成本的效果有限.文献[7]等提出配置信息缓存的方法,该方法通过缓存关键的配置信息,减少配置信息加载次数.这种方法虽然设计简单,但硬件成本较大.文献[8]进一步提出了提升配置缓存效率的方法.基于预测的配置信息预取[9~13]是近年来的主要研究方向.配置信息预取主要包括静态重构指令预测和动态重构指令预测两类.文献[12]的基于编译器的静态指令预测,通过编译器在代码中插入特殊指令实现配置信息预取.该方法无法解决分支之后的指令预测.

收稿日期:2010-03-30;修回日期:2012-03-01

文献[13]提出基于马尔可夫链的动态指令预测,但是由于马尔可夫预测的无前效性与程序的局部性原理不符,因此存在预测准确率低的缺点.

本文提出了一种基于神经网络的可扩展的重构指令预取机制,相比现有解决方法,不仅提高了配置信息的预取效率,同时降低了硬件实现成本.其内容包括:(1)引入感受器的历史指令信息,构建权重和历史指令信息协同工作的新型感受器模型.(2)运用改进的神经网络学习重构指令的调用规律.在处理器运行过程中,提前完成对后续重构指令的预测及配置信息的预取,将配置信息加载延时隐藏于处理器执行周期中.(3)提出本方法的可扩展实现框架,将神经网络训练结果移至内存并分布式存储.由于学习信息存储于内存,不受处理器核硬件资源的限制,使得本方法在降低处理器核硬件成本的同时获得良好的扩展性.

2　基于神经网络的重构指令预取机制

2.1　预测模型

神经网络算法模拟生物体神经系统的工作原理,具有自学习和自适应的特性,被广泛应用于模式识别、趋势预测等领域.本文利用神经网络算法的自学习特性,对程序中的重构指令执行轨迹进行学习并动态预测后续重构指令.在重构指令的执行过程中,重构指令之间的调用存在一定内在联系,表现为前序重构指令的调用历史一定程度决定了后续重构指令的调用.据此,本文的预测原理为:首先,为每条重构指令分配指令 ID,区分程序执行过程中不同功能的重构指令;其次,为每条重构指令的调用过程构建神经网络模型,并引入 h 条重构指令调用历史作为神经网络的输入模式;运用神经网络动态学习 h 条重构指令调用历史与后续重构指令之间的调用关系,并将学习结果保存在神经元感受器中;在处理器运行过程中,利用神经网络学习结果对当前的重构指令调用历史进行识别,提前预测后续重构指令并完成相应配置信息的预取.

本文在传统神经网络算法的基础上,增加感受器的历史指令信息,用于记忆重构指令调用历史.感受器权重与感受器的历史指令信息共同构成了新型的感受器模型,通过权重与历史指令信息的协同训练,实现对重构指令调用历史与后续重构指令之间的调用关系的学习.改进之后的神经网络预测模型如图 1 所示.该模型是为单条重构指令的调用过程构建的预测模型,由 n 个神经元构成,每个神经元代表一个预测元素;每个神经元上分布着 h 个感受器,用于分别感受 h 条历史重构指令的刺激.

由于每个感受器对应一对由历史指令信息和权重构成的学习信息,因此整个神经元表征了一个 h 维向量的感受器信息矩阵$[(\boldsymbol{f}_1,\boldsymbol{w}_1),\cdots,(\boldsymbol{f}_i,\boldsymbol{w}_i),\cdots,(\boldsymbol{f}_h,\boldsymbol{w}_h)]$.其中,$\boldsymbol{f}_i$ 是第 i 个感受器的历史指令信息,其内容为重构指令的 ID;$\boldsymbol{w}_i$ 是第 i 个感受器的权重,其内容为有符号整数,表示该感受器对神经元输出的作用强度和方向.当 h 条重构指令调用历史为$[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_h]$,整个神经元表示了一个相似度函数 $\mathrm{S}(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_i,\cdots,\boldsymbol{x}_h)$,如式(1)(2)所示.相似度函数的大小反映了该神经元成为预测指令的概率,相似值越高,成为预测指令的概率越大.

$$\mathrm{S}(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_i,\cdots,\boldsymbol{x}_h)=\sum_{1}^{h}\boldsymbol{b}_i\boldsymbol{w}_i \tag{1}$$

$$\boldsymbol{b}_i=\begin{cases}-1, & \text{if}(\boldsymbol{x}_i\neq\boldsymbol{f}_i)\\ 1, & \text{if}(\boldsymbol{x}_i=\boldsymbol{f}_i)\end{cases} \tag{2}$$

在重构指令的预测过程中,神经网络接受 h 条历史重构指令的刺激,分别计算各个神经元的相似值,最后选择相似度最大的神经元作为预测的重构指令,整个预测过程如图 1 所示.

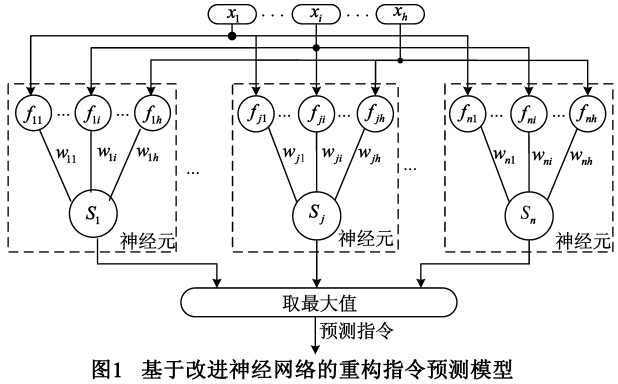


图1　基于改进神经网络的重构指令预测模型

2.2　训练算法

神经网络在动态运行过程中,不断接受由历史重构指令及其后续指令构成的激励的刺激,并依据训练算法调整感受器权重和历史指令信息,实现对重构指令调用关系的学习.神经网络的训练算法是影响神经网络预测效率和收敛速度的关键.本文提出的训练算法的核心思想是在预测错误的情况下,通过对相应神经元分别进行奖励和惩罚,调整神经元相似值的相对大小,体现重构指令执行的局部性特征并加快收敛速度.

定义　I_{cur}为当前重构指令,I_{next}为后续重构指令,I_{pred}为预测指令.神经网络的训练过程如算法 1 所示:当执行到后续重构指令时,判断指令预测是否正确.当 $I_{pred}=I_{next}$时,表示神经网络准确完成了预测,因此无需进行训练;当 $I_{pred}\neq I_{next}$时,表示预测不正确,此时需要对相关神经元的相似值大小进行调整.具体调整方法为:对 I_{next}的神经元进行奖励,增强其在当前输入模式下的相似值函数,同时,为加快收敛速度,对 I_{pred}的神经

元进行惩罚,减小其相似值函数.奖励过程如图 2 第 3 – 10 行所示,对于神经元 I_{next},当感受器的历史指令信息等于当前的历史重构指令时,表明该感受器与神经元相似值的作用方向一致,因此权重加一,否则权重减一.同时,为了在训练算法中体现重构指令执行的局部性原理,一旦更新完毕之后的权重小于 0 时,则对感受器权重取反,同时感受器的历史指令信息更新为当前的历史重构指令.惩罚过程如图 2 第 13 – 14 行所示,对于神经元 I_{pred},当感受器的历史指令信息等于当前的历史重构指令时,表明该感受器与神经元相似值的作用方向相反,因此感受器权重减一,否则权重保持不变.完整的神经网络训练算法的伪代码如算法 1 所示.

算法 1:改进神经网络预测算法的训练法则

```
1:  if( I_next != I_pred){                    //预测错误
2:    for(i = 0; i < h; i++){                 //奖励 I_next
3:      if(I_next.f_i == x_i)
4:        (I_next.w_i)++;
5:      else
6:        (I_next.w_i)--;
7:      if(I_next.w_i < 0){
8:        I_next.w_i = -I_next.w_i;
9:        I_next.f_i = x_i;
10:     }
11:   }
12:   for(j = 0; j < h; j++){                 //惩罚 I_pred
13:     if(I_pred.f_j == x_j)
14:       (I_next.w_j)--;
15:   }
16:}
```

3 可扩展的神经元信息存储框架

由于每个神经元包含 h 个感受器,每个感受器需要记忆权重和历史指令信息,因此预测集合为 n 的改进神经网络的存储结构如图 2 所示.硬件开销为 $Cost = n.(ID + h.(p + q))$bits,其中 ID 是每个神经元代表的的预测指令,p 是感受器历史指令信息,q 是感受器权重.

本文涉及的预测方法为每条重构指令的调用过程构建神经网络模型,因此其需要实现的神经网络个数随重构指令数的增加线性增长.重构指令越多,神经网络的个数越多,需要保存的神经元信息越庞大.目前常见的神经网络硬件实现方法是将神经元信息集中存储于处理器核内 SRAM 上,并与处理器内部的神经网络预测器直接运算获得预测结果[14].这种实现方式虽然设计简单直观,但是硬件开销大且无法随重构指令数的改变动态扩展,存在面向应用的扩展性差的缺点.

预测元素	感受器历史指令信息	感受器权重
I_1	f_{11} … f_{1h}	w_{11} … w_{1h}
…	…	…
I_n	f_{n1} … f_{nh}	w_{n1} … w_{nh}

图2 改进神经网络存储结构

由于动态可重构处理器在每次完成预测后从片外存储器载入目标指令的配置信息,基于该特征,本文进一步提出了神经元信息的可扩展存储框架.神经网络的训练结果作为重构指令的关联信息,从传统的核内 SRAM 移至片外存储器.在重构指令预取时,除了加载配置信息,同时加载目标指令的神经网络训练结果.由于该方法将神经元信息存储于片外,使得神经网络个数不再受处理器核内存储资源的限制,满足了可重构处理器面向应用重构不同数量指令的需求.

神经元信息的可扩展存储框架如图 3①所示.在每条指令配置信息之后预留 $n.(ID + h.(p + q))$bits 的内存空间,存放指令的神经网络训练结果.在可重构处理器运行过程中,除了预取目标指令的配置信息,同时预载入神经网络训练结果.通过神经网络训练结果与核内的神经网络预测器运算获得预测的后续指令.神经网络预测器的结构如图 3②所示,预测器仅需极少数硬件资源即可完成预算与指令预测功能.

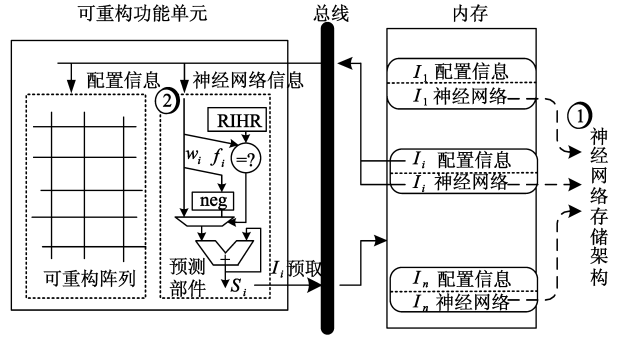


图3 可扩展的神经网络存储框架

4 实验及分析

本文以国产 32 位高性能嵌入式处理器 CK510[15]及其仿真平台为基础,以 Powerstone 基准测试程序为目标程序.首先,分析获得 Powerstone 基准测试程序的热点函数,并针对热点函数设计相应的重构指令.在此基础上,对基于神经网络的重构指令预取机制进行设计空间搜索,分析算法预测准确率与各个设计要素的内在联系.最后,将基于神经网络的预取机制与现有的马尔可夫预取模型[8]进行预测准确率和综合性能的对比.

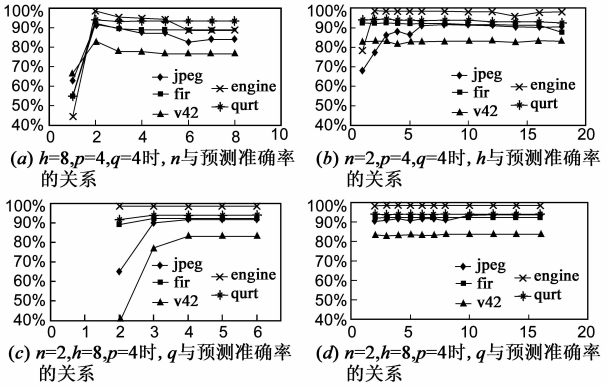
表 1 显示 Powerstone 各个基准测试程序的热点函数信息及重构指令分布情况.

基于神经网络的重构指令预取机制,其设计空间元素包括预测准确率 a,预测集合维数 n,历史重构指令条数 h,感受器历史指令信息的表示位数 p,感受器权重的表示位数 q.对神经网络预取机制进行设计空间

搜索,发现其性能与硬件开销的最佳平衡点对可重构嵌入式处理器设计意义重大.图 4(a ~ d)分别表示预测准确率 a 与设计空间元素 n、h、p、q 的关系.图 4(a),当预测集合维数 n 为 1 时,神经网络的预测准确率为 44% ~67%;当预测集合维数为 2 时预测准确率达 83% ~98%,准确率有明显提升.随着维数的继续增加,预测准确率基本保持不变,但较 $n=2$ 时略低.分析原因发现,$n=2$ 已基本满足不同重构指令运行的时间局部性特征,当维数继续增加时,神经网络对预测元素的训练开销增加,预测率反而略有下降.图 4(b),当历史重构指令条数由 1 增加到 6 时,预测准确率提升明显;当 $h>6$ 时,准确率达到饱和,基本保持不变.其原因在于,6 条历史重构指令的信息可基本识别不同重构指令的时间局部性和空间局部性特征,已能有效消除指令别名问题.图 4(c)显示对于重构指令数较少的应用(如 fir、engine、qurt),仅需较少的指令信息表示位数(2bit),但对于重构指令数较多的应用(如 jpeg.c 和 v42.c),所需的表示位数需要增加(4bit),因此感受器的历史指令信息表示位数与重构指令条数呈正比.图 4(d)显示权重表示位数对预测准确率影响较小.通过图 4 的研究发现,预测集合维度 n 是影响预测准确率的首要因素.历史重构指令条数 h,感受器的指令信息表示位数 p 也是影响预测准确率的重要因素.权重表示位数 q 对预测准确率影响则较小.

表 1　Powerstone 的热点函数及重构指令分布

基准程序	热点函数(个)	执行比重	重构指令(条)	重构指令执行次数	重构指令平均对应的汇编指令(条)
jpeg	5	80.3%	26	43 5581	9.3
fir	7	87.4%	16	22 8916	5.9
v42	6	68.7%	23	21 3792	7.1
engine	3	80.6%	7	1 7588	6.8
qurt	6	90.9%	13	1 6473	6.2



(a) h=8,p=4,q=4时,n与预测准确率的关系　(b) n=2,p=4,q=4时,h与预测准确率的关系

(c) n=2,h=8,p=4时,q与预测准确率的关系　(d) n=2,h=8,p=4时,q与预测准确率的关系

图4　重构指令预测算法的设计空间搜索

目前对于重构指令动态预测的研究主要是基于马尔可夫的预测模型[8].本文以 Powerstone 基准测试程序的总执行周期数作为处理器综合性能的反映,选取 $n=2$,$h=6$,$p=4$,$q=4$ 的神经网络预测模型与预测集合为 4,概率位数为 10bit 的马尔可夫模型(依据文献[17]结论,此时马尔可夫模型的预测准确率最好)进行预测准确率和综合性能的对比,结果分别如图 5、图 6 所示.

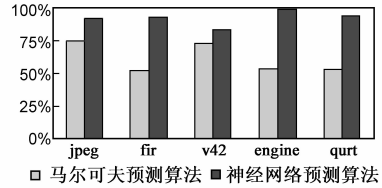


图5　神经网络模型与马尔可夫模型的预测准确率对比

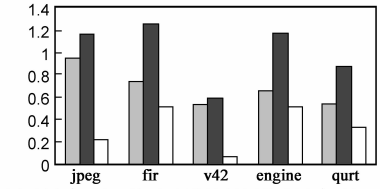


图6　神经网络预取机制与马尔可夫预取机制的综合性能对比

由图 5 可见,神经网络预测模型的预测准确率平均为 91%,相比马尔可夫预测模型的 65%,准确率提升 26%.其主要原因是马尔可夫模型仅通过重构指令的相对执行概率进行预测,无法识别指令执行过程中复杂控制流对于预测的影响.相比之下,神经网络预测模型通过改进的神经元学习重构指令之间的动态调用关系,实现对不同重构指令时间局部性和空间局部性的识别,达到较为理想的预测结果.

受益于重构指令预测准确率的巨大改进,本文提出的预取机制相比马尔可夫预取机制,整体性能有 40% 的提升.

5　结论与展望

本文重点讨论了阻碍可重构处理器发展的重构成本问题,深入研究和探索了基于神经网络的重构指令预取机制,同时提出了该方法的可扩展实现架构,极大的降低了可重构处理器的指令重构成本.本文所涉及方法为可重构处理器尤其是重构指令预取提供了积极的参考.

参考文献

[1] A Lodi, M Toma, et al. A VLIW processor with reconfigurable instruction set for embedded applications [J]. IEEE Journal of Solid-State Circuits, 2003, 38(11): 1876 – 1886.

[2] 孟建熠,丁永林,葛海通,等.一种 RTL 级数据通路 ODC 低功耗优化算法[J].电子学报,2010,38(7):1654 – 1659.
Meng Jian-yi, Ding Yong-lin, Ge Hai-tong, et al. A RTL level ODC algorithm for data path low power optimization [J]. Acta Electronica Sinica, 2010, 38(7): 1654 – 1659. (in Chinese)

[3] 宋宇鲲,高明伦,邓红辉,等.DReAC:一种新型动态可重

构协处理器[J].电子学报,2007,35(5):833 – 837.
Song Yun-kun, Gao Ming-lun, Deng Hong-hui, et al. DReAC: A novel dynamically reconfigurable Co-processor [J]. Acta Electronica Sinica, 2007, 35(5): 833 – 837. (in Chinese)

[4] M J Wirthlin, B L Hutchings. Dynamic instruction set computer [A]. Proceedings of the IEEE Symposium on FPGA's for Custom Computing Machines [C]. Napa Valley, CA, USA: IEEE Computer Society, 1995. 99 – 107.

[5] A Dandalis, V K Prasanna. Configuration compression for FPGA-based embedded systems [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2005, 13(12): 1394 – 1398.

[6] U Malik, O Diessel. On the placement and granularity of FPGA configuration [A]. Proceedings of the IEEE International Conference on Field-Programmable Technology [C]. Brisbane, Australia: IEEE Electron Devices Society, 2004. 161 – 168.

[7] Z Li, K Compton, S Hauck. Configuration cache management techniques for reconfigurable computing [A]. Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines [C]. Washington, DC, USA: IEEE Computer Society, 2000. 22 – 36.

[8] Q Yang, J-P Soininen, et al. A configuration locking technique to reduce the configuration overhead of run-time reconfigurable devices [A]. Proceedings of the International Symposium on System-on-Chip [C]. Tampere, Finland, 2007. 1 – 5.

[9] J Resano, D Mozos, et al. A hybrid prefetch scheduling heuristic to minimize at run-time the reconfiguration overhead of dynamically reconfigurable hardware [A]. Proceedings of the Conference on Design, Automation and Test in Europe [C]. Washington, DC, USA: IEEE Computer Society, 2006. 106 – 111.

[10] J RResano, et al. Efficiently scheduling runtime reconfigurations [J]. ACM Transactions on Design Automation of Electronic Systems, 2008, 13(4): 58 – 65.

[11] B B Wu, et al. Run-time configuration prefetching to reduce the overhead of dynamically reconfiguration [A]. IEEE International SoC Conference [C]. Las Vegas, NV, USA: IEEE Press, 2010. 305 – 308.

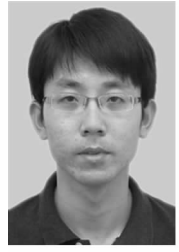
[12] X Tang, M Aalsma, R Jou. Compiler directed approach to hiding configuration latency in Chameleon processors [A]. Proceedings of the Roadmap to Reconfigurable Computing, 10th International Workshop on Field-Programmable Logic and Applications [C]. London, UK: Springer-Verlag, 2000. 29 – 38.

[13] Z Li, S Hauch. Configuration prefetching techniques for partial reconfigurable coprocessor with relocation and defragmentation [A]. ACM/SIGDA International Symposium on Field Programmable Gate Arrays [C]. Monterey, CA, United states: Association for Computing Machinery, 2002. 187 – 195.

[14] D Jimenez, C Lin. Neural methods for dynamic branch prediction [J]. ACM Transactions on Computer Systems, 2002, 20(4): 369 – 397.

[15] C-SKY High Performance Embedded Processor [DB/OL]. http://www.c-sky.com, 2008.

作者简介



陈志坚 男,1984 年 7 月出生于浙江洞头,2006 年获浙江大学自动化学士学位,现为浙江大学超大规模集成电路设计研究所硕博连读生,主要研究方向为高性能嵌入式处理器设计,可重构处理器设计.
E-mail: chenzj@vlsi.zju.edu.cn

孟建熠 男,讲师,1982 年 1 月出生于浙江上虞,分别于 2004 年和 2009 获浙江大学电子信息工程学士和电路与系统博士,主要研究方向为高性能低功耗嵌入式处理器设计.
E-mail: mengjy@vlsi.zju.edu.cn