RISCV密码专用处理器能效概率模型与体系结构研究

李 伟*① 别梦妮① 陈 韬① 吴艾青① 南龙梅②
①(解放军信息工程大学 郑州 450000)
②(复旦大学专用集成电路与系统国家重点实验室 上海 200433)

摘 要：该文以高能效为目标，建立了密码专用处理器能效概率模型，并指导高能效密码专用处理器体系结构设计。该文将面向密码领域的专用指令处理器设计空间探索问题描述为“1”值在配置矩阵中的定位问题，通过引入概率矩阵进一步将定位问题转化为最优配置的概率问题，并基于机器学习思想提出了密码专用处理器最高能效概率模型。实验证明，该文提出的能效概率模型平均经过2300次迭代输出最终结果，且预测准确率达到92.7%。根据最高能效概率模型，对密码专用处理器设计空间进行探索，获取满足高能效需求的密码专用处理器运算单元集合，以扩展指令的方式将其集成到开源通用64位RISCV处理器核心Araine中，提出高能效密码专用处理器体系结构。将该处理器在CMOS 55 nm工艺下进行逻辑综合，结果表明，该文提出的RISCV密码专用处理器与扩展前相比面积增大了426874 μm^2，关键延迟增加了0.51 ns，完成密码算法总时间面积积增幅之和为0.46，执行常见密码算法能效比在1.61～35.16 Mbps/mW范围内。

关键词：密码处理器；机器学习；能效概率模型；高能效
中图分类号：TN918.4; TP316.4 **文献标识码**：A **文章编号**：1009-5896(2021)06-1541-09
DOI: 10.11999/JEIT210004

Research on Energy Efficiency Probability Model and Architecture of RISCV Cryptographic Processor

LI Wei① BIE Mengni① CHEN Tao① WU Aiqing① NAN Longmei②
①(*PLA Information Engineering University*, *Zhengzhou* 450000, *China*)
②(*State Key Laboratory of ASIC and System*, *Fudan University*, *Shanghai* 200433, *China*)

Abstract: This paper establishes an energy efficiency probability model for a dedicated cryptographic processor, and guides the design of the cryptographic processor. The design space exploration problem of a processor is designed as the positioning problem of "1" values in the configuration matrix. The probability matrix is introduced to transform the positioning problem into an optimal configuration probability problem. Based on the idea of machine learning, a probability model for the highest energy efficiency of a dedicated cryptographic processor is proposed. Experiments prove that the energy efficiency probability model in this paper outputs the final result after 2300 iterations on average, and the prediction accuracy rate reaches 92.7%. According to the highest energy efficiency probability model, a collection of computing units that meet high energy efficiency requirements can be obtained, and they are integrated into the open source general-purpose 64 bit RISCV processor core named Ariane. A dedicated processor for energy-efficient cryptography is built. The processor is synthesized under the CMOS 55 nm process, and the results show that compared with Ariane, the area of the proposed cryptographic processor increases by 426874 μm^2, the key delay increases by 0.51 ns, and the sum of the increasing total time area of the cryptographic algorithm is 0.46, the energy efficiency ratio of common cryptographic algorithms is within the range of 1.6～35.16 Mbps/mW.

Key words: Cryptographic processor; Machine learning; Energy efficiency probability model; High energy efficiency

1 引言

在万物互联的大数据时代，网络与信息安全问题日渐突出，小型设备承担的密码处理任务日益丰富，对其密码处理能力的要求也随之提高。研究一种高能效的密码芯片，在有限的面积和功耗下获取更高的密码处理性能具有重要应用价值和现实意义。

收稿日期：2021-01-04；改回日期：2021-04-07；网络出版：2021-04-16
*通信作者： 李伟 liwei12@fudan.edu.cn

在密码处理器实现方式中，通用微处理器具有很大的灵活性，但由于其运算粒度太细，运算速度远不如专用密码芯片。专用密码芯片采用ASIC方式直接针对1种或几种特定的算法进行硬件优化，密码处理性能高，但无法进行二次开发，其兼容性差、可扩展性差。专用指令处理器有领域专用指令集，用户可通过软件开发平台实现不同密码算法的可重构，处理性能也接近专用密码芯片的水平，是一种具有吸引力的折中方式。因此，本文以高能效为目标，对密码专用指令处理器体系结构设计关键技术展开研究。

首先，能效模型是高能效密码专用处理器设计的基础，需要构建密码处理器能效分析模型，指导处理器整体结构设计。

高能效密码处理器体系结构设计问题是一个设计空间探索问题。处理器体系结构设计空间探索通常是一个由设计人员根据经验反复试验以寻找合适的配置参数的过程。在这个过程中，设计人员需要对每一个可能影响处理器最终表现的参数进行选择，通过模拟其中部分选择方案来验证是否满足需求。这一过程往往需要大量的长时间的迭代，且很难确定最终选择方案是否为最佳方案。为了加快这一进程，且提高设计方案的准确度，有研究者提出将体系结构设计空间探索问题描述为回归问题的思想，采用线性回归、样条回归、神经网络等[1–6]分析模型进行参数预测和分析，取得了较好的成果。值得注意的是，随着机器学习技术的蓬勃发展，更多的研究者将其引入体系结构设计空间探索问题[7,8]，使得探索迭代速度和准确度大大提高。但是，这些研究聚焦于体系结构中的存储容量、指令发射并行度等显而易见的参数分析，而忽略了如单元粒度等不能使用具体数值表述的参数分析，而这部分参数往往更大程度地影响处理器的表现。

除此之外，精简高效的架构是高能效密码专用处理器设计的关键。在能效模型的指导下，还需寻求高能效的处理器架构，并设计精简的密码运算单元，共同构建高能效密码专用处理器。

在密码专用处理器架构中，有VLIW(超长指令字)、CISC(复杂指令集)、RISC(精简指令集)3种设计方案。其中VLIW指令集[9]架构把许多条指令连在一起，增加了运算的速度，实现了指令级并行。这种方式实际使用多倍的硬件资源来换取运算速度，其庞大的面积和功耗开销使得其能量效率大大降低。CISC指令集[10]架构构建的运算单元粒度大、延迟长、功耗大，同样不适合以高能效为目标的密码专用处理器。与之相比，RISC架构因指令数目少、指令格式固定、指令实现粒度小等特点使得其硬件实现开销小且能达到较高的工作频率，更适合以高能效为目标的密码专用处理器架构设计。RISC架构中，新推出的RISCV架构开源且配套开发环境完整，更适合以高能效为目标的密码专用处理器架构设计。

针对上述分析，本文以高能效为目标，深入研究密码专用指令处理器能效建模方法和体系结构设计技术。首先，将密码专用处理器运算单元设计空间探索问题描述为配置矩阵中的“1”值定位问题；然后，借鉴机器学习理念，引入“1”值停留在本位的概率参数，建立密码专用处理器运算单元高能效概率模型；最终，在该概率模型的指导下，提出一套高能效密码专用处理器运算单元并将该运算单元集合扩展到64位RISCV通用密码处理器中，完成高能效密码专用指令处理器的构建。

本文结构安排如下：第2节对密码专用指令集设计问题展开深入分析；第3节提出基于机器学习思想的密码专用处理器运算单元高能效概率模型；第4节在能效分析模型指导下完成高能效密码基本运算单元集合的设计并扩展到RISCV处理器中；第5节进行实验验证及分析，第6节总结本文工作。

2 高能效密码专用处理器设计问题分析

2.1 设计目标

在资源有限的嵌入式设备上，平衡资源和性能的关系显得更为重要，高能效逐渐成为研究者追求的目标。密码加速单元作为密码专用处理器计算能力的集中体现，其计算位宽和计算粒度影响着密码专用处理器的关键路径延迟和面积功耗开销，同时决定了完成一个密码任务的指令条数。在密码加速单元设计方案已知的情况下，可进一步确定密码处理器处理密码任务的能效。

处理器核心的能效被定义为芯片性能与功耗的比值，单位为bps/W。可见，描述处理器核心的能效需要采集芯片的性能和功耗两个参数。其中，性能参数在设计阶段可通过逻辑综合近似评估，而功耗需要在芯片后端设计完成后才能较为准确地采集。若直接采用性能与功耗的比值作为评价指标，则设计迭代需要更长的时间代价。在设计初期，研究者广泛采用更容易采集到的时间面积积参数来近似评估处理器的能效。

密码专用处理器是否满足高能效的目标需要在执行某一具体算法的过程中体现，算法集合不同所设计的加速单元集合必然不同。我们假设目标算法集合为$\{y_1, y_2, \cdots, y_m\}$，设一个密码处理器核心完成

算法y_l需要时间为T_l，密码处理器核心的总面积为A，那么，该密码处理器核心完成y_l算法的时间面积积W_l如式(1)。

$$W_l{=}T_l \cdot A \tag{1}$$

当综合考虑执行算法集合中所有算法的能效时，若简单将时间面积积相加，则总时间面积积必然受指令条数多的算法影响更大。因此，为平衡各密码算法体制差异对能效分析过程的影响，采用时间面积积增幅f_l之和作为最终评价标准，则总时间面积积增幅f表示如式(2)，其中W_{l0}表示不集成任何密码加速单元时，处理器实现密码算法y_l的时间面积积。

$$f=\sum_{l=1}^{m} f_l=\sum_{l=1}^{m}\frac{W_l}{W_{l0}} \tag{2}$$

综上所述，高能效密码专用处理器设计目标为找到使得总时间面积积增幅f值最小的一种体系结构。

2.2 问题描述

对目标算法集合展开理论分析，抽取出其细粒度共性逻辑，并以此分析结果为依据，确定基本密码加速单元集合的功能种类。对于同一种功能的密码运算，可用不同粒度、不同位宽的基本运算单元迭代实现。粒度和位宽大，其关键延迟和面积大，但实现算法所需时钟周期短；粒度和位宽小，则其关键延迟和面积小，但实现算法所需时钟周期长。我们难以通过定性分析判断采用何种粒度与位宽的密码基本运算单元能效更高。

综合考虑算法集合、运算单元功能、规模、关键路径延迟等参数的影响，一种配置处理器完成某一密码算法y_l的时间面积积W_l可表示为式(3)，总时间面积积增幅f表示为式(4)。

$$W_l{=}C_l \cdot \max_{0\le i\le n}\{k_i t_i\}\cdot\sum_{i=0}^{n} k_i s_i\ (k_0=1) \tag{3}$$

$$f=\sum_{j=1}^{m}\frac{C_l}{C_{l0}}\cdot\frac{\max_{0\le i\le n}\{k_i t_i\}\cdot\sum_{i=0}^{n} k_i s_i}{t_0\cdot s_0}\ (k_0=1) \tag{4}$$

式(3)与式(4)相关参数释义如表1所示。

由于运算单元x_i之间的区别主要来源于粒度、位宽和功能的差异，假设运算单元功能共a种，位宽取值范围为32, 64, 128, 256 bit 4种情况，粒度视单元种类不同而有不同选择，则$\{k_1,k_2,\cdots,k_n\}$的可能取值空间将不少于$2^{(4a)}$种。然而，位宽不同、功能相同的单元不会同时存在，$\{k_1,k_2,\cdots,k_n\}$的取值情况中存在一定的互斥关系，为体现这种互斥关系，可将$\{k_1,k_2,\cdots,k_n\}$改写为矩阵形式，矩阵中每行表示同一种功能单元，按粒度或位宽从左向右递增。改写后矩阵$\boldsymbol{K}$表示如式(5)所示。显然，矩阵$\boldsymbol{K}$每行有且仅有一位为1。

$$\boldsymbol{K}=\{k_{ij}|0<i\le a,0<j\le b\} \tag{5}$$

与$\boldsymbol{K}$矩阵相对应，将加速单元集合、面积集合、延迟集合也改写为矩阵形式，如式(6)—式(8)所示，则式(4)可进一步表述为式(9)。

$$\boldsymbol{X}=\{x_{ij}|0<i\le a,0<j\le b\} \tag{6}$$

$$\boldsymbol{S}=\{s_{ij}|0<i\le a,0<j\le b\} \tag{7}$$

$$\boldsymbol{T}=\{t_{ij}|0<i\le a,0<j\le b\} \tag{8}$$

$$f{=}\sum_{j=1}^{m}\frac{C_l}{C_{l0}}\cdot\frac{\max\{k_{ij}t_{ij},t_0\}\cdot\left(s_0+\sum k_{ij}s_{ij}\right)}{t_0\cdot s_0} \tag{9}$$

此时，我们将高能效密码专用处理器设计问题描述为矩阵$\boldsymbol{K}$中“1”值的定位问题。即当$\boldsymbol{K}$矩阵每行元素中的“1”位于什么位置时，使得总时间面积积增幅f取得最小值。

3 基于机器学习思想的密码专用处理器能效概率模型

3.1 高能效概率模型学习框架

观察式(9)，t_{ij}以及s_{ij}均可通过对密码加速单

表 1　参数列表

参数	约束	含义
x_i	$1\le i\le n$	可集成到处理器中的密码加速单元
y_i	$1\le l\le m$	算法集合中某目标密码算法
t_i	$1\le i\le n$	密码加速单元x_i对应的关键延迟
s_i	$1\le i\le n$	密码加速单元x_i对应的面积
k_i	$k_i\in[0,1]$	表示单元x_i是否为处理器扩展单元(1表示扩展)
t_0		表示未扩展密码运算单元时处理器关键延迟
s_0		表示为扩展密码运算单元时处理器面积
C_l		一种密码运算单元扩展配置下完成y_l算法所需时钟周期数
C_{l0}		未扩展密码运算单元时完成y_l算法所需时钟周期数

元进行逻辑综合获取其值，若C_l与矩阵$\boldsymbol{K}$之间的关系确定，则可利用最优化算法求解最优配置。然而，C_l与矩阵$\boldsymbol{K}$之间的关系无法使用某一具体函数形式表述，只有在矩阵$\boldsymbol{K}$已知的情况下，通过编译仿真获取其具体数值。考虑此时$\boldsymbol{K}$矩阵可能的取值情况有5^a种，无法通过穷举编译来确定C_l与矩阵$\boldsymbol{K}$的关系。因此，需要寻求其他方法建立f关于矩阵$\boldsymbol{K}$的模型。

在这种情况下，机器学习这种不依赖于具体逻辑而依赖于数据关系的建模方式充分展现了其优势，可有效辅助建立f关于矩阵$\boldsymbol{K}$的能效分析模型。

建立密码专用处理器能效分析模型的目的在于找到一个使得f取值最小的矩阵$\boldsymbol{K}$。在不能穷举的情况下，我们更倾向于找到一个最有可能使f取值最小的矩阵$\boldsymbol{K}$。即建立一个最高能效概率PW(矩阵$\boldsymbol{K}$使得f取值最小的概率)与矩阵$\boldsymbol{K}$之间的关系。

Shan等人[11]在构建功耗分析模型时，引入了一个通过概率矩阵获取最小代价函数的机器学习算法，其解决的数学问题与本文有一定的相似性。但是，Shan等人的目标是找到一个满足某一阈值范围的代价函数，引入的概率矩阵是为了反馈纠正代价函数的取值可能，使代价函数更逼近所设置的阈值。

借鉴上述概率矩阵转移学习算法思想，本文提出一种高能效概率模型学习框架，如图1所示。它由一个训练阶段和一个预测阶段组成。在训练阶段，该框架从设计空间采样$\boldsymbol{n}$种配置，训练一个最高能效概率模型。在预测阶段，根据最高能效概率模型对设计空间中所有配置进行预测，从中找到使得能效最高的概率最大的配置矩阵$\boldsymbol{K}$，即所求配置。

概率模型学习框架以最高能效概率为训练目标，可辅助建立起密码专用处理器能效概率模型。

3.2 概率矩阵转移算法

如图1所示，训练概率模型的关键即为概率矩阵转移算法，其直接决定了训练效率以及概率模型的准确度。为设计一个合理的概率矩阵转移算法，本文引入了一个概率矩阵$\boldsymbol{P}$表述最高能效概率PW与矩阵$\boldsymbol{K}$之间的关系。概率矩阵$\boldsymbol{P}$中元素p_{ij}表示“1”值停留在k_{ij}时总时间面积积增幅f最小的概率。由于$\boldsymbol{K}$矩阵中每行有且只有1个“1”元素，且每行“1”元素的位置相互独立，显然最高能效概率PW与p_{ij}具有如式(10)所示关系。其中，p_{ij}满足式(11)。根据概率矩阵$\boldsymbol{P}$，可以预测配置数据库中任一配置为最优配置的概率。

$$\mathrm{PW}=\frac{1}{a}\sum_{i=1}^{a}\sum_{j=1}^{b}k_{ij}p_{ij} \tag{10}$$

$$\sum_{j=1}^{b}p_{ij}=1,\ i\in[1,a] \tag{11}$$

初始情况下，概率矩阵$\boldsymbol{P}$中元素p_{ij}为该行元素个数的倒数，即“1”值出现在该行任意位置的概率相等，为$1/b$。在学习过程中，通过随机采样从未知配置集合中抽取一个配置，编译仿真计算的实际f值，使用该值与已标记的配置集合进行比较，判断新采样的配置在已知配置集合中是否为最优，并利用判断结果更新概率矩阵的值。经过多次迭代优化，当连续r次预测结果与实测结果一致时，停止训练，r是学习算法的一个控制参数，其值越大，训练准确度越高，训练量也越大。

综上，概率矩阵转移算法流程图如图2所示，其具体描述如下：

(1)初始化配置矩阵$\boldsymbol{K}$及概率矩阵$\boldsymbol{P}$。矩阵$\boldsymbol{K}$中第1列为“1”，其余元素为“0”；概率矩阵$\boldsymbol{P}$中元素p_{ij}为该行元素个数的倒数；仿真初始矩阵$\boldsymbol{K}$对应的总能效比f，记为$f_{\min}$；初始化控制参数$\mathrm{sum}=0$。

(2)随机选取矩阵$\boldsymbol{K}$的一种配置$\boldsymbol{K}_i$，利用概率矩阵$\boldsymbol{P}$计算已知配置集合中所有配置分别对应的最高能效概率，判断新选取矩阵$\boldsymbol{K}_i$是否为最优配置。

(3)仿真获取新选取矩阵$\boldsymbol{K}_i$对应的总能效比f_i，计算$\delta=f_i-f_{\min}$，若$\delta>0$意味着该配置不是最优配置，反之，该配置是当前已知最优配置。

(4)将步骤(2)预测结果与步骤(3)实测结果比对，若一致则$\mathrm{sum}=\mathrm{sum}+1$；否则，$\mathrm{sum}=0$。

(5)若$\delta>0$，即该配置不是最优配置，将该配置对应位置的p_{ij}减小一个参量α(α为概率矩阵调整参量，决定了算法的学习速率)，该行其余位置的p_{ij}增大α/b；反之，增大一个参量α，该行其余位置的p_{ij}减小α/b；更新概率矩阵$\boldsymbol{P}$。

(6)若$\mathrm{sum}=r$，则输出概率矩阵$\boldsymbol{P}$，算法结束；否则，返回操作步骤(2)。

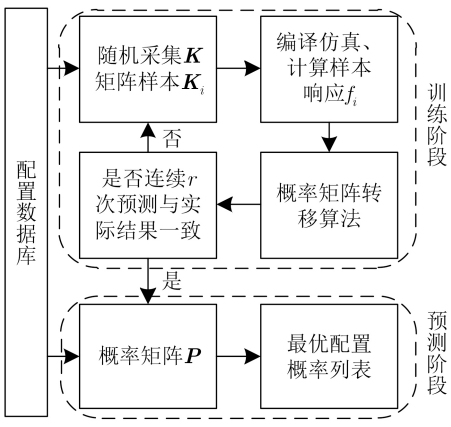

图 1 概率模型学习框架

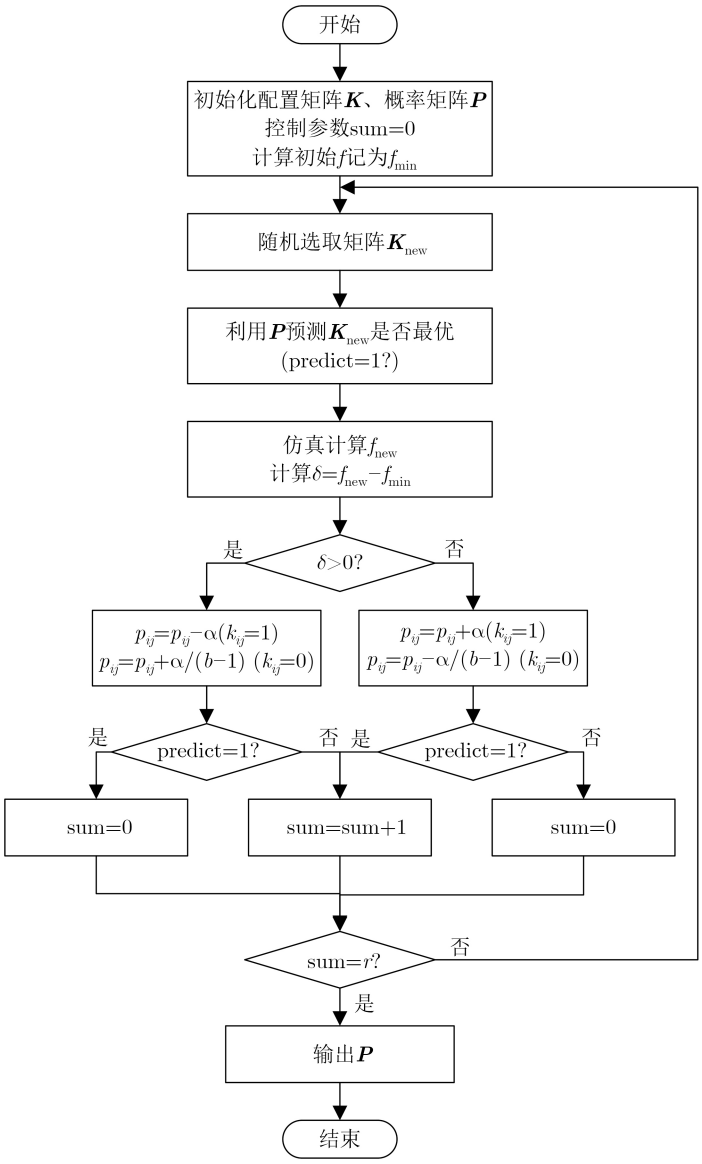

图 2 概率矩阵转移算法流程图

4　基于RISCV的高能效密码专用处理器体系结构研究

开源RISCV指令集架构因指令数目少、指令格式固定、指令实现粒度小等特点使得其硬件实现开销小且能达到较高的工作频率，在通用处理器领域大放异彩，是当前通用高能效处理器设计的首选框架。本文以通用RISCV处理器为平台，应用能效概率模型探索满足高能效需求的扩展密码加速单元集合，从而构建高能效RISCV密码专用处理器。

首先，我们借鉴了通用RISCV指令集架构，针对包括分组、序列、杂凑密码算法在内的10余种常用密码算法，抽取出细粒度共性逻辑，设计密码加速单元。基于这些加速单元采集能效模型所需相关数据，构建密码专用处理器配置数据库。然后，在密码专用处理器能效分析模型的指导下，探索密码处理器设计空间，提出用于密码运算的专用指令集。最后，根据密码专用指令集选出相应的密码运算单元，扩展构建高能效RISCV密码处理器。

4.1　配置参数采集

本文对包括分组、序列、杂凑密码算法在内的10余种常用密码算法的结构特征进行分析，对其运算类型进行分类，抽取出常见密码算法的细粒度共性逻辑，并以此为依据设计了相应的密码运算单元。

本文选取常用密码算法集合包括DES, AES, M4, IDEA, A5-1, SM3, MD5, SHA256, Grain, ZUC, SNOW, RC4, CHACHA20等。针对上述算法集合共设计密码加速单元12种，包括线性反馈移位寄存器、比特矩阵乘、三输入布尔函数、非线性反馈移位寄存器、比特置换、模加/减、模乘、S盒替代、有限域乘法、移位、筛选、插入等[12–15]。即

$\boldsymbol{K}$矩阵中参数a= 12。这12种密码运算单元分别依照32, 64, 128, 256 bit 4种运算位宽设计，即$\boldsymbol{K}$矩阵中参数b= 5。则共设计运算单元48种，设计配套指令57条(不同位宽指令视为同一条指令)，计算得$\boldsymbol{K}$矩阵设计空间大小为5^{12}。

在CMOS 55 nm工艺下分别对48种运算单元进行逻辑综合，构建$\boldsymbol{S}$矩阵与$\boldsymbol{T}$矩阵。

4.2 高能效密码专用处理器设计空间探索

根据4.1节分析，在GCC编译链中扩展自定义指令57条。选取标准C语言编写的算法程序作为标准输入。根据概率模型学习算法需求，编译统计不扩展密码运算单元情况下执行各密码算法所需指令条数，结果如图3所示。

根据图3中数据和式(9)，计算总时间面积积并初始化$f_{\min}$，同时初始化配置矩阵$\boldsymbol{K}$及概率矩阵$\boldsymbol{P}(p_{ij}=0.2)$，选取控制参数$r=20$, α=0.01，启动概率模型转移算法执行程序。经学习，获取运算单元集合参数如图4所示。

4.3 高能效密码专用处理器体系结构设计

由于密码运算不存在复杂的乘除、原子、浮点操作，且基本运算位宽多在32 bit或64 bit内，同时考虑对序列密码算法中线性反馈移位寄存器等大位宽操作的兼容，本文采用RV64I指令集作为基本指令集。该指令集仅支持两元一目的指令形式，只能实现最高128 bit输入64 bit输出。然而，在密码操作中，存在诸如线性反馈移位寄存器等需要128 bit数据输入128 bit数据输出的操作，又有诸如三输入布尔函数等需要3个64 bit数据输入的操作。级联实现指令开销更大，拖慢处理性能。考虑到RISCV指令格式中寄存器位置固定，遵循其设计原则，将2个源寄存器和1个目的寄存器均进行复用，即3个寄存器均可为源、目的寄存器以满足密码处理需要。而对于诸如连续模加等需要4个以上输入数据的指令，增加了一组专用寄存器作为指令中某一固定输入。扩展的指令格式如表2所示。其中，RL为两源两目的无立即数指令(例如128 bit置换)，SRI为两源两目的带立即数指令(例如线性反馈移位寄存器)，C为配置指令(例如配置有限域运算不可约多项式)。

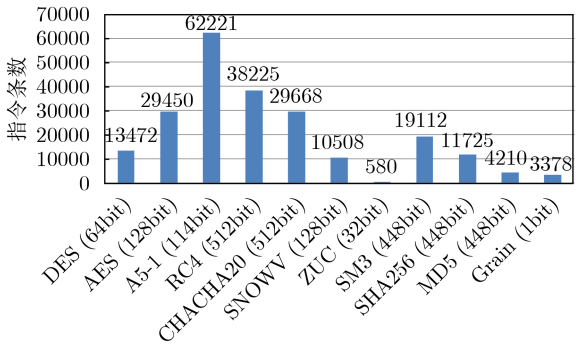

图 3 各密码算法所需原RISCV指令条数

依据高能效密码专用处理器设计空间探索结果和上述指令设计原则，设计密码加速单元。选取PULPino项目开源的64位RISCV处理器核心Araine为基本框架，扩展专用密码运算单元，并根据密码处理任务特征需求，对处理器基本框架进行了优化，扩展后的RISCV密码处理器结构示意图如图5所示。

5 实验验证及分析

5.1 能效概率模型准确率分析

为验证能效概率模型准确率，随机从设计空间中预先抽取200组数据，通过编译仿真获取其在给定密码算法集合下的总时间面积积增幅之和，作为测试集保存。

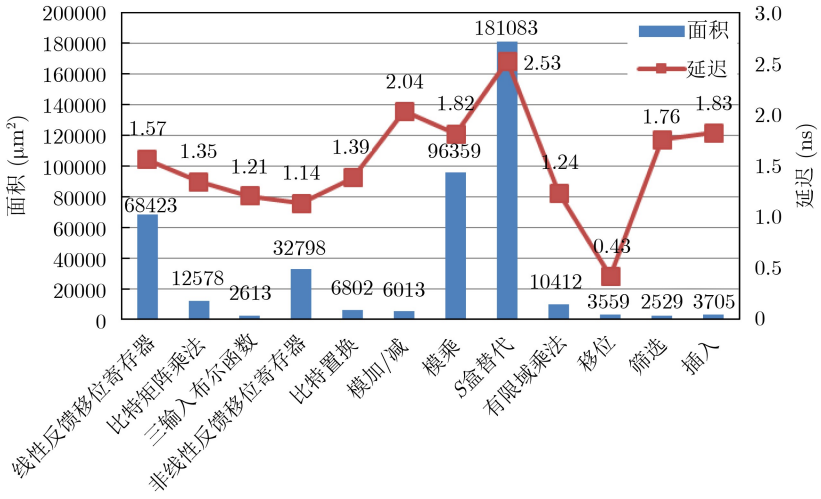

图 4 探索获取密码运算单元集合参数

表 2　指令模板

	[31:26]	[25]	[24:20]	[1915]	[14:12]	[11:7]	[6:0]
RL	Funt7		Rsd2	Rsd1	Funt3	Funt5	opcode
SRI	Imm[5:0]	Funt1	Rsd2	Rsd1	Funt3	Rds	opcode
C	Funt7		Rs2	Rs1	Imm[7:0]		opcode

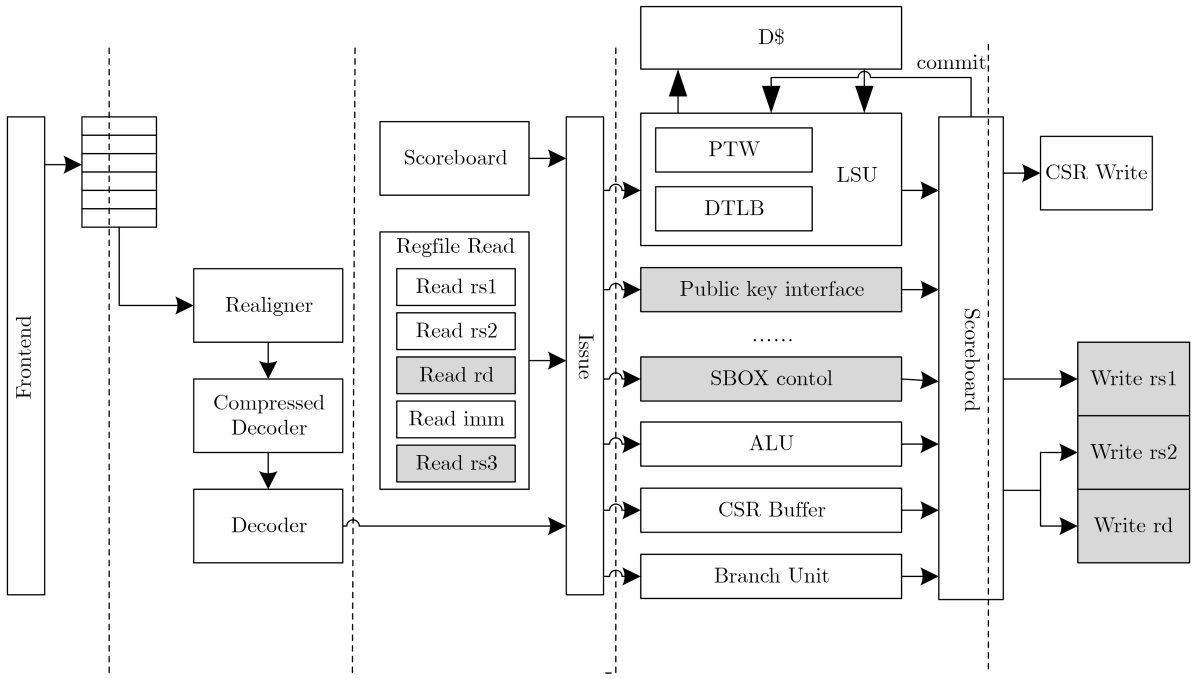

图 5 RISCV密码处理器核心结构示意图

依据4.2节设计空间搜索结果，利用概率矩阵预测测试集中配置的能效概率。由于预测结果是一个概率值，不能直接反映为总时间面积积增幅。一种解决方式是：分别根据总时间面积积增幅之和以及能效概率对测试集中配置进行排序，以排名位置为标准对模型准确率进行检验。

经测试，本文提出的能效概率模型平均经过2300次迭代后结束，预测准确度为92.7%。

5.2　密码专用处理器能效分析

为对本文提出的RISCV密码专用处理器进行验证，使用Verilog HDL对RISCV密码专用处理器进行描述，并在CMOS 55 nm工艺下进行了逻辑综合，结果如表3所示。与扩展前相比，本文提出的RISCV密码专用处理器面积增大了426874 μm^2，关键延迟增加了0.51 ns。

以典型AES, DES, SHA256, SNOWV, MD5等密码算法为例，其实现所需指令周期数结果如表4所示，需要注意的是，此处本文统计的是完成完整密码算法批处理所需指令，不包括密钥生成过程和初始化过程等。其时间面积积结果如图6所示。综上数据可知，本文提出的RISCV密码专用处理器完成密码算法总时间面积积增幅之和为0.46。

为进一步验证本文提出的RISCV密码专用处理器的能效优势，我们采集了RISCV密码专用处理器执行AES, DES, SHA256, SNOWV, SM3等密码算法的实测功耗与吞吐率(400 Mbps)，并与未扩展密码加速单元的Ariane处理器进行比较。结果如图7所示。

综合图6和图7数据，可以发现，本文提出的密码专用处理器不同程度地提高了目标集合中算法的

表 3　RISCV密码专用处理器性能

	Ariane	密码专用处理器
关键延迟(ns)	1.9	2.3
面积(μm^2)	1321154	1748028

表 4　RISCV密码专用处理器实现算法所需指令周期数

	Ariane	密码专用处理器
AES	29450	402
DES	13472	384
SHA256	11725	1192
SNOWV	10508	1026
MD5	4210	196

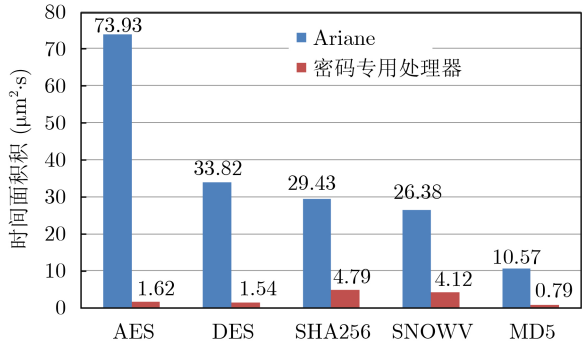

图 6 RISCV密码处理器核心实现算法的时间面积积

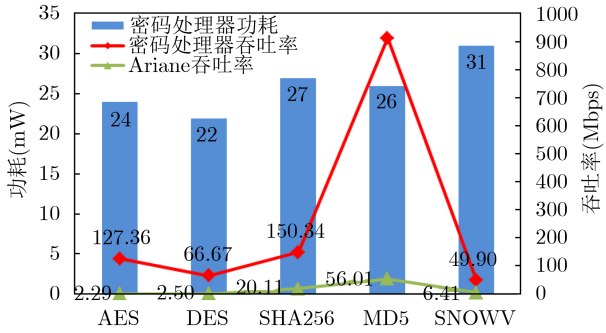

图 7 RISCV密码处理器能效分析

吞吐率，其中，对分组密码算法的加速效果最好，对序列密码算法的适应性略显不足，但仍然比Ariane处理器能效更优。由于Ariane处理器未披露其处理器功耗数据，我们依据时间面积参数，与Ariane处理器处理密码任务的能效进行对比。可以发现，密码专用处理器的处理密码任务的时间面积积相比Ariane处理器提高了6～55倍不等。

将本文提出的密码专用处理器与VLIW的密码专用处理器进行对比，为了对不同工艺下实现能效进行比对，本文参考了文献[16]中的工艺换算方法对能效进行了简单等价，其粗略评估结果如表5所示，表中数据均是标准化为55 nm工艺下的值。

由表5可知，与文献[17]相比，AES算法能效略低，为文献[17]的56%，但其余算法能效值提升了1.04～10.65倍，总体而言，对测试算法集合来说，能效比优于文献[17]。与文献[18]相比，本文结构在执行杂凑密码算法时具有明显优势，SHA256算法能效较之提高了21倍，MD5算法提高了54倍。虽然分组密码算法能效值较之低2～4倍，但文献[18]的数值是于无反馈模式下测得的，应用场景有限；而且，与文献[18]相比，本文提出的结构执行分组密码算法能效的降低幅度远低于杂凑密码算法能效的提升幅度，整体而言，本文提出的结构能效更优。

表 5 密码处理器能效对比分析(Mbps/mW)

	文献[17]	文献[18]	密码专用处理器
AES	9.39	20.74	5.30
DES	2.91	6.91	3.03
SHA256	3.55	0.26	5.56
SNOWV	–	–	1.61
MD5	3.30	0.64	35.16

6 结束语

面向领域专用的处理器设计方式中，专用指令处理器是灵活性与运算效率之间的折中选择。本文以高能效为目标构建了密码专用处理器能效概率模型，并在该模型指导下完成了密码专用处理器体系结构设计。本文首先将运算单元设计空间探索问题描述为矩阵定位问题，借鉴机器学习思想提出了概率矩阵学习框架，建立了面向密码领域的专用处理器运算单元能效分析模型。然后通过模型分析得到了针对一个密码算法集合的高能效密码运算单元集合。最终在通用处理器架构的基础上根据密码处理特征进行优化，并集成高能效密码运算单元集合，实现了一个高能效的RISCV密码专用指令处理器。

参 考 文 献

[1] AZIZI O, MAHESRI A, LEE B C, *et al*. Energy-performance tradeoffs in processor architecture and circuit design: A marginal cost analysis[J]. *ACM SIGARCH Computer Architecture News*, 2010, 38(3): 26–36. doi: 10.1145/1816038.1815967.

[2] DUBACH C, JONES T M, and O'BOYLE M F P. Exploring and predicting the architecture/optimising compiler Co-design Space[C]. The 2008 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, Atlanta, USA, 2008: 31–40.

[3] LEE B C and BROOKS D M. Accurate and efficient regression modeling for microarchitectural performance and power prediction[C]. The 12th International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, USA, 2006: 185–194.

[4] LEE B C and BROOKS D M. Illustrative design space studies with microarchitectural regression models[C]. The 13th IEEE International Symposium on High Performance Computer Architecture, Scottsdale, USA, 2007: 340–351.

[5] LEE B C and BROOKS D. Applied inference: Case studies in microarchitectural design[J]. *ACM Transactions on Architecture and Code Optimization*, 2010, 7(2): 1–37.

[6] PALERMO G, SILVANO C, and ZACCARIA V. ReSPIR: A response surface-based pareto iterative refinement for application-specific design space exploration[J]. *IEEE Transactions on Computer-Aided Design of Integrated*

Circuits and Systems, 2009, 28(12): 1816–1829. doi: 10.1109/TCAD.2009.2028681.

[7] CHEN Tianshi, GUO Qi, TANG Ke, *et al*. ArchRanker: A ranking approach to design space exploration[C]. The 41st ACM/IEEE International Symposium on Computer Architecture (ISCA), Minneapolis, USA, 2014: 85–96.

[8] CHEN Tianshi, CHEN Yunji, GUO Qi, *et al*. Effective and efficient microprocessor design space exploration using unlabeled design configurations[J]. *ACM Transactions on Intelligent Systems*, 2014, 5(1): 20.

[9] LI W, ZENG Xiaoyang, NAN Longmei, *et al*. A reconfigurable block cryptographic processor based on VLIW architecture[J]. *China Communications*, 2016, 13(1): 91–99. doi: 10.1109/CC.2016.7405707.

[10] 孟沫舒. CISC处理器复杂指令的实现方法研究[D]. [硕士论文], 中国科学院研究生院, 2012.

[11] SHAN Weiwei, ZHANG Shuai, XU Jiaming, *et al*. Machine learning assisted side-channel-attack countermeasure and its application on a 28-nm AES circuit[J]. *IEEE Journal of Solid-State Circuits*, 2020, 55(3): 794–804. doi: 10.1109/JSSC.2019.2953855.

[12] LI Wei, CHANG Zhongxiang, FENG Xiao, *et al*. Fast parallel extract-shift and parallel deposit-shift in general-purpose processors[C]. The 12th IEEE International Conference on Ubiquitous Intelligence and Computing, Beijing, China, 2015: 764–771.

[13] 马超, 李伟, 戴紫彬, 等. 新型可重构移位-置换单元研究与设计[J]. 电子学报, 2017, 45(5): 1025–1034. doi: 10.3969/j.issn.0372-2112.2017.05.001.

MA Chao, LI Wei, DAI Zibin, *et al*. A novel reconfigurable rotation-permutation unit research and implementation[J]. *Acta Electronica Sinica*, 2017, 45(5): 1025–1034. doi: 10.3969/j.issn.0372-2112.2017.05.001.

[14] 马超, 戴紫彬, 李伟, 等. RPRU: 一种面向处理器的比特抽取与移位统一架构[J]. 计算机研究与发展, 2018, 55(2): 426–437. doi: 10.7544/issn1000-1239.2018.20160775.

MA Chao, DAI Zibin, LI Wei, *et al*. RPRU: A unified architecture for rotation and bit-extraction operations in general-propose processor[J]. *Journal of Computer Research and Development*, 2018, 55(2): 426–437. doi: 10.7544/issn1000-1239.2018.20160775.

[15] 徐光明, 徐金甫, 常忠祥, 等. 序列密码非线性反馈移存器的可重构研究[J]. 计算机应用研究, 2015, 32(9): 2823–2826. doi: 10.3969/j.issn.1001-3695.2015.09.062.

XU Guangming, XU Jinfu, CHANG Zhongxiang, *et al*. Reconfigurability study on nonlinear feedback shift registers in stream cipher[J]. *Application Research of Computers*, 2015, 32(9): 2823–2826. doi: 10.3969/j.issn.1001-3695.2015.09.062.

[16] LIU Bin and BAAS B M. Parallel AES encryption engines for many-core processor arrays[J]. *IEEE Transactions on Computers*, 2013, 62(3): 536–547. doi: 10.1109/TC.2011.251.

[17] LI Wei, ZENG Xiaoyang, DAI Zibin, *et al*. A high energy-efficient reconfigurable VLIW symmetric cryptographic processor with loop buffer structure and chain processing mechanism[J]. *Chinese Journal of Electronics*, 2017, 26(6): 1161–1167. doi: 10.1049/cje.2017.06.010.

[18] SAYILAR G and CHIOU D. Cryptoraptor: High throughput reconfigurable cryptographic processor[C]. 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, USA, 2014: 155–161.

李　伟：男，1983年生，副教授，博士生导师，研究方向为密码处理器设计、ASIC专用芯片设计.

别梦妮：女，1997年生，硕士生，研究方向为智能化可重构芯片电路与架构.

陈　韬：男，1979年生，副教授，硕士生导师，研究方向为安全专用芯片设计.

吴艾青：男，1997年生，硕士生，研究方向为智能化可重构芯片电路与架构.

南龙梅：女，1981年生，博士生，研究方向为大规模集成电路设计、专用集成电路设计.

责任编辑：陈　倩