物理设计探索用于埃米时代的线友好型领域专用处理器

Lorenzo Ruotolo, Lara Orlandic, Pengbo Yu, Moritz Brunion, Daniele Jahier Pagliari, Dwaipayan Biswas, Giovanni Ansaloni, David Atienza, Julien Ryckaert, Francky Catthoor, and Yukai Chen

摘要—本文介绍了针对机器学习（ML）领域特定处理器（DSIP）架构的物理设计探索，解决了先进 Ångstrom 时代技术中的互连效率挑战。该设计通过利用专门的记忆结构和 SIMD（单指令多数据）单元来减少线长并提高核心密度。五种配置使用 IMEC 的 A10 纳米片节点进行合成和评估。关键的物理设计指标在不同配置之间以及与最先进的 DSIP 基准 VWR2A 进行了比较。结果表明，我们的架构实现了超过 2× 的标准化线长降低和高于 3× 的密度，且所有配置中的指标变异性较低，使其成为下一代 DSIP 设计的一种有前景的解决方案。这些改进是在极少的人工布局干预下实现的，展示了该架构内在的物理效率及其适合低成本布线友好型实施的潜力。

***Index Terms*—领域特定处理器，机器学习，物理设计，纳米片，布线长度优化。**

I. 介绍

FIELDS 例如机器学习（ML）和数字信号处理（DSP）正在迅速发展，从而增加了其应用的复杂性 [1]. 这一趋势导致了定制加速器的日益采用，这些加速器通过在专门优化的硬件上执行内核来加快特定任务的速度，尤其是在边缘设备中。尽管与通用 CPU 相比，这些加速器非常高效，但它们缺乏灵活性，并且通常只加速工作负载的一部分。

特定领域的指令集处理器（DSIP）通过支持针对特定领域更广泛的指令集，提供了更加灵活的替代方案，能够在其目标领域内加速整个应用程序。通过缩小范围，它们在通用 CPU 的灵活性和应用专用加速器的效率之间实现了良好的折衷。近年来已经提出了几种 DSIP 架构。VWR2A [2] 和 R-Blocks [3] 是粗粒度可重构阵列（CGRA）设计，提供了灵活性但因类似脉动阵列的互连而遭受路由效率低下的问题。同样地，作为可扩展向量处理器的 AraXL [4] 也因其二维块基础布局未能高效扩展而遭受路由拥塞。

这些限制在先进技术节点中变得越来越重要。随着传统的缩放法则，如 Dennard 缩放和摩尔定律的失效，新的微架构解决方案需要被开发以持续进入埃米时代，在这个时代特征尺寸将低于一纳米。尽管晶体管仍在缩小，但互连线未能按比例缩小 [5]。虽然通过采用新材料来处理导线和电介质已经探索了工艺层面的缓解措施，解决布线瓶颈也要求架构级别的解决方案以实现 “对导线友好”。减少平均导线长度不仅减少了电容负载，从而降低峰值功率和总能耗，而且降低了电阻，有助于减轻 IR 降。此外，这种减少还改善了热行为，因为功率和温度紧密相关 [6]。

这些平均布线长度的减少必须在不牺牲可扩展性（以计算密度为衡量标准）的前提下实现，以便最大限度地提高每单位面积的性能。现有的用于 AI 或 DSP 负载的 DSIPs，如 VWR2A，尽管是在亚纳米节点中实现的，但仅部分采用了这些架构策略，因此未能充分利用先进工艺技术的潜力。解决这一限制需要在路由效率和密度的同时约束下重新思考架构设计。

为此，我们重新审视 DSIP 架构，重点在于互连效率和计算密度。受到这些挑战的启发，我们采用了来自 [7] 的基础设计，该设计集成了高效的缓存存储器（SPMs）和非常宽的寄存器（VWRs）[8] ，并结合了灵活的单指令多数据（软 SIMD）矢量功能单元（VFUs）[9], [10] 。本工作的主要贡献在于物理设计探索以及与类似

L. Ruotolo and D.J. Pagliari are with the Politecnico di Torino, Turin, Italy.

L. Orlandic, P.B. Yu, G. Ansaloni, D. Atienza are with École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland.

F. Catthoor is with the National Technical University of Athens, Athens, Greece. Email: catthoor@microlab.ntua.gr

L. Ruotolo, M. Brunion, D. Biswas, J. Rychaert, Y.K. Chen are with IMEC, Leuven, Belgium. Email: yukai.chen@imec.be

面向机器学习的 DSIP 进行比较评估。我们展示了关键设计指标的一致改进，实现了高达 50%的布线长度减少和核心密度 3× 的增长，在 IMEC 预测 A10（1 纳米）纳米片技术节点上验证了我们的方法。

II. 背景与相关工作

本工作中采用的 DSIP 架构集成了多个关键架构组件，每个都体现了创新特性。本节简要介绍这些组件，为理解后续的物理设计探索提供必要的背景。

1) 非常宽的寄存器: 最初在 [8] 中引入的非常宽寄存器（VWR）是一种内存组织设计，旨在提高传统寄存器文件架构的能量效率。位于功能单元附近，VWRs 充当前景存储器，存储内循环计算的数据。这减少了访问更高层级内存层次结构的成本较高的频率，从而最小化数据移动并增强局部性，这对于从高数据重用中获益的机器学习工作负载特别有利。

与多端口寄存器文件不同，每个单元都连接到多个位线和字线，每个 VWR 单元只连接到一条位线和一条字线。这种简化结构允许更快且更节能的并行访问。VWR 具有单个端口，带有两个不对称接口：一个宽接口用于内存传输，另一个窄接口用于数据路径操作。当与 SIMD 功能单元配对时，从 VWR 检索到的每个字包含多个子字，从而在保持并行性的同时减少访问开销。一旦缓冲的数据被消耗掉，新的数据将以更高的代价从较高层级的内存中获取。尽管 VWR 不具备传统寄存器文件那样的灵活性，无法执行针对各个子字的多次随机访问，但对于具有规则内存访问模式的工作负载（如 ML 推理或 DSP 内核）而言，这一限制并不重要。

多项研究表明了基于 VWR 的架构在能效方面的表现。在 [8] 中，VWR 相比集群式多端口寄存器文件，在 DSP 基准测试上实现了高达 10× 的能量节省。VWR2A 架构 [2] 展示了进一步的效率提升，与 ARM Cortex-M4 相比减少了 70.8%的系统级能耗，与 Ibex 核心相比减少了 74.4%，这主要是由于其优化的内部存储结构涉及了 VWRs，并且减少了昂贵的 SoC 互连如 AMBA AHB [11]。

2) 软 SIMD: 传统的 SIMD 架构是通过可以定义为硬件 SIMD（硬-SIMD）的方式来实现的，仅支持有限的一组 SIMD 位宽。这种方法限制了灵活性，并且限制了可以从细粒度或更小位宽并行性中受益的应用程序的潜在性能提升，例如量化 ML 模型。

相比之下，本工作采用了 Soft-SIMD（软件定义的 SIMD）VFUs 来提高数据并行工作负载的灵活性和效率。Soft-SIMD 最初在 [9] 中引入，它能够在运行时重新配置 SIMD 宽度，使得能够更精确地与应用级并行性需求对齐。本设计中使用的 Soft-SIMD VFUs 还结合了规范符号数（CSD）编码，促进了基于移位-加法的乘法器的使用。通过减少操作数中的非零数字数量，所需的移位-加法操作次数显著降低，从而降低了动态功耗。

如 [9] 所示，Soft-SIMD 虽然可能需要更多的乘法周期，但显著降低了总体能量延迟面积产品（EDAP），在均匀和异构数据量化基准场景中分别实现了高达 56.6% 和 72.9% 的改进。

III. 设计探索方法论

本研究中探讨的 DSIP 设计采用了基于方块的架构，遵循 ProVeT（处理器向量方块）模板，这是 AERO 框架的一个专门子集 [10], [12]。在此架构中，数据密集型工作负载的加速由称为**瓷砖**的专业单元处理，这些单元可以实例化为 2D 数组以在 SoC 上实现粗粒度并行性。单个方块的示意图如图 1 所示。本研究重点探讨了单个 DSIP 方块的物理设计探索。控制平面与 [7] 中的实现一致，被视为外部组件并且不包括在内。基于 ProVeT 的方块模板高度可配置，具有诸如 VWR 和 VFU 的数量和大小及其位宽等可调参数，从而促进了广泛且灵活的设计空间探索。

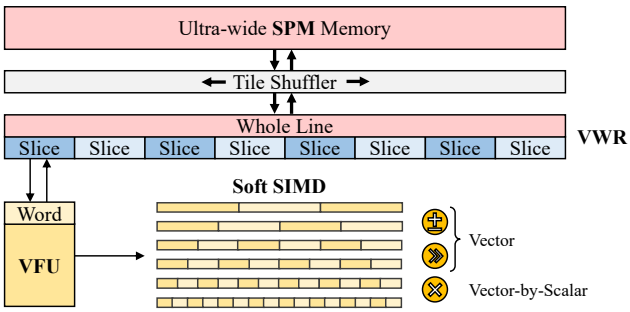

图 1. 简化后的内部瓷砖地板布局，突出显示了预期的布局。

在物理设计探索方面，我们的目标是一种高层次的布局方案，在这种方案中，每个瓦片内的 Soft-SIMD VFUs 通过直接互连与系统内存层次结构的最低级别（即 VWRs 和 SPM）直接接口连接，类似于其他计算靠近存储 (CnM) 架构方法 [13]. 这使典型的 CnM 设计效率提升成为可能，这种提升源于减少的数据移动和改进的局部性。

简化互连结构的使用也在物理实现中带来了实际优势。在布局阶段，穿越较少金属层的较短互连允许更紧密的标准单元打包和更高的可达到密度。在布线过程中，减少的走线复杂性降低了布线拥塞并缩短了设计封

闭时间。这些物理优势在传统设计中的多端口寄存器文件或基于交叉开关的互连中难以实现。

本节详细说明了瓷砖设计的三个主要组成部分：**存储层次结构**、**计算单元**以及它们之间的**互连**。

1) 存储层次结构: 所提出的架构中的内存层次结构被组织成三个层级：SPM、VWR 和本地 VFU 寄存器。

SPM 作为阵列中其他模块与系统主内存之间的主要数据接口，通过外部片上网络（NoC）进行访问。内部，SPM 由 SRAM 银行组成，通过共享控制信号并行控制。通过向所有银行广播相同的地址，可以同时访问同一行，形成一个 N 位宽、M 个字深的 L1 SPM。

在下一级别，第 II 节引入的 VWR 用作 SPM 和本地 VFU 寄存器之间的 L0 缓冲。每个 VWR 的宽度为 N 位，与 SPM 行的宽度匹配，并且深度为 1 位，使用标准锁存单元实现。VWR 在逻辑上被划分为与 VFU 数据路径宽度相匹配的字，并分组到切片中，每个 VFU 连接到一个切片。这允许仅直接访问分配切片中的字，从而减少了接口复杂性并利用了空间局部性。

如需访问，可以通过系统级直接内存访问（DMA）控制器或专用的片块混洗器使用较慢的数据重新排列间接地获取 VFU 切片之外的单词。该片块混洗器采用一个左移一位的移位器来实现 VWR 内的快速重配置。然而，其面积开销会随着支持的重新排列模式而增加，可能会减少直接连接 VWR-SPM-VFU 的空间。这种权衡必须基于工作负载局部性进行评估。

最终的存储层次由每个 VFU 中的本地寄存器组成。这些单字宽的寄存器存储中间操作数和结果。它们的内容根据活动的子字配置动态解释，符合软 SIMD 模型：每个字包含多个子字，从而实现细粒度并行。

2) 计算单元: 每个磁贴中的计算单元实现了第 II 节中描述的 Soft-SIMD 范式。每个磁贴包含可配置数量的 VFU，其数据路径宽度也可以参数化。虽然更宽的数据路径提供了更高的并行性，但也引入了更高的内部延迟（主要是通过 ALU 的进位链）和更大的面积开销。与布局的其余部分一致，VFU 的放置旨在最小化横向数据流长度。较少使用的组件，如数据打包单元 (DPU)，被放置在磁贴的最右侧以避免干扰关键路径。每个 VFU 的示意图见图 1，其功能细节见 [9]。

3) 互连: 每个 SPM 感测放大器输出直接连接到相应的 VWR 锁存接口，该接口与对齐的 VFU 共享相同的物理端口。这种布局最大限度地减少了水平数据路径长度，这是晶片中的主要布线方向，从而提高了局部性并降低了布线复杂度。与传统的寄存器文件相比，缺少复杂的多路复用结构允许组件之间进行短距离、位级、点对点连接。最节省布线的配置不使用任何晶片混频器，仅有一个 VWR，并且每个切片一个字，允许完全直接连接而无需中间布线或逻辑开销。

IV. 实验

为了展示所提出设计在不同架构配置下的线高效特性，我们通过改变关键组件的数量和大小（SPMs、VWRs 和 VFUs）探索了五个变体（A – E）。对于每个配置，都使用 Cadence 套件完成了标准的数字设计流程 [14] [15] 。为了确保一致性，布局区域被设置为由放置工具确定的最小布线区域。所有设计均采用 IMEC 的 A10 预测 PDK 实现，该 PDK 表示的是埃米时代、深度缩放纳米片全环绕栅极场效应晶体管（GAAFET）技术节点。

为了评估我们方法的优势，我们将它与基于 CGRA 范式的可重构 DSIP——VWR2A [2] 进行了比较。和我们的设计一样，它的目标是高计算密度和低功耗运行，针对机器学习和数字信号处理应用。它还集成了节能的内存组件，如 SPMs 和 VWRs，这些也是我们架构的重要组成部分。然而，使用传统的脉动阵列式处理元件（PE）互连和复杂的瓦片重排程序导致路由复杂度增加，从而使得面积和布线长度要求更高。为了进行有意义且公平的比较，我们使用相同的 A10 纳米片 PDK 合成了 VWR2A 的一个版本，确保两种设计都在相同的技术约束下进行了评估。在生成的五个配置中，配置 (E) 在聚合内存大小（24 KiB 对 32 KiB SPM 和 2304 KiB 对 3072 KiB VWRs）以及综合后的逻辑单元数量（304K 对 328K）上最接近 VWR2A，这使得可以直接进行架构级和物理层面的比较。

所有五种配置的关键架构参数，连同 VWR2A 基线一起，汇总于表 I 中。为了便于比较，我们还报告了每个单元的总内存大小（以字节为单位），计算方法是该单元所有实例的总容量。

A. 布局后结果比较

综合工具为五种配置生成的最终布局如图 2 所示。从视觉上看，我们的设计在所有配置中都明显比 VWR2A 更密集，从而导致更高的利用率和内部组件之间的连线更短。这一观察结果得到了表 II 中所列关键布线后指标数据的支持。

表 I
配置间磁瓦参数的汇总

单位	参数	A	B	C	D	E	VWR2A
	Columns	1	1	1	1	1	2
	Word Width [Bits]	96	192	96	192	192	32
	Tile Shuffler	✗	✗	✗	✓	✓	✓
特殊性能材料	Bank Size [Bits]	512×64					
	Banks Number	3	6	6	3	6	8
	Bitwidth	1536	3072	3072	1536	3072	4096
	Aggregate Size [KiB]	12	24	24	12	24	32
视野范围	Number	1	4	2	2	6	6
	Bitwidth	1536	3072	3072	1536	3072	4096
	Slices Per VWR	8	1	8	8	16	8
	Words Per Slice	2	16	4	1	1	32
	Words Per VWR	16	16	32	8	16	128
	Aggregate Size [B]	188	1536	750	375	2304	3072
虚拟功能单元	Number	8	1	8	8	16	8
	Datapath Bitwidth	96	192	96	192	192	32
	Aggregate Size [B]	96	24	96	192	384	32

结果表明，无论架构配置如何，所有设计的核心密度均超过核心区域的 40%，平均为 50.77%，标准差为 6.42%。此外，我们将线长与面积比作为标准化指标来独立评估线长效率的设计大小。在所有设计中，该比率平均值为 112.08，标准差为 28.28。这些趋势在图 3 中可视化，展示了设计的核心密度和归一化线长的变异性，其中横轴代表标准单元数量。

与 VWR2A 的基线相比，我们选择的配置 (E) 具有相似数量的标准单元和总逻辑面积。有趣的是，(E) 也显示没有失效端点 (FEP)，并且在寄存器到寄存器互连中的最差负松弛 (WNS) 为+0.004 ns，表明这个正定时裕量可能允许比 VWR2A 中使用的更高频率，从而有可能提高性能。此外，与 VWR2A 的 16.00%核心密度相比，(E) 实现了 53.89%。(E) 还具有几乎短 2× 的归一化互连线长，相比于 VWR2A。这两个指标在所有配置 (A-E) 中保持有利且在一个狭窄的变化范围内这一事实表明，我们的设计在扩展瓦片配置时维持了高计算密度和布线效率。

表 II
配置间的键指标

度量	A	B	C	D	E	VWR2A
Number of Standard Cells	81,121	139,447	121,482	187,564	304,173	327,714
Total Logical Area [μm^2]	3,372	6,648	6,092	5,517	10,632	15,881
reg2reg FEPs	17	199	0	3335	0	114
reg2reg WNS (Setup) [ns]	-0.004	-0.008	+0.002	-0.035	+0.004	-0.008
Wire length [μm]	275,894	917,486	468,085	651,732	1,548,251	4,716,330
Wire length-To-Area Ratio	81.82	138.01	76.84	118.13	145.62	296.98
Core Density	46.09%	48.30%	43.79%	61.77%	53.89%	16.00%

V. 结论

我们介绍了针对高效布线布局优化的 DSIP 架构的物理设计探索，这是未来埃时代技术的关键挑战之一。

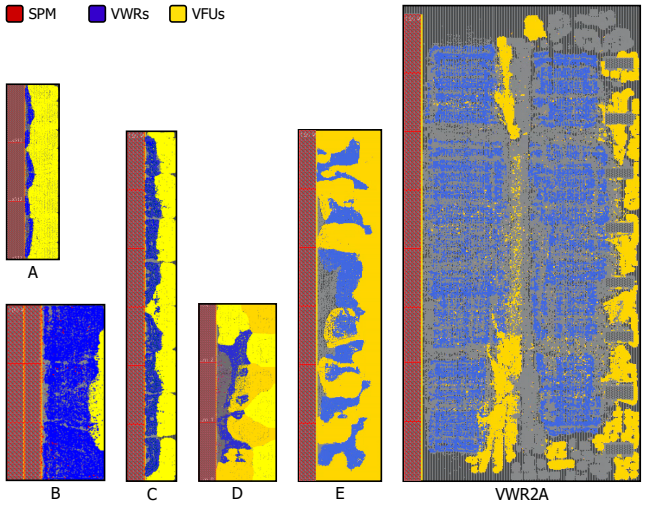

图 2. 配置 (A – E) 和 VWR2A 生成的物理布局的同尺度比较。SPM 银行（红色）位于左侧，而右侧的标准单元则被分组为 VWRs（蓝色）和 VFUs（黄色）。

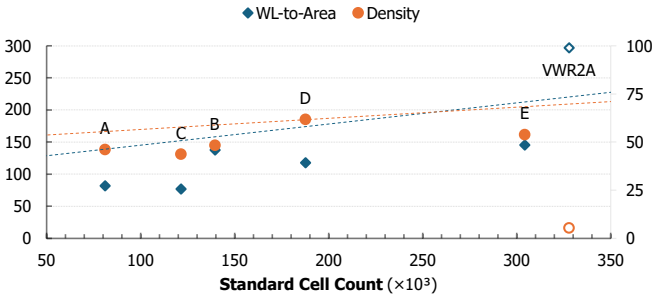

图 3. 显示 WL 到面积和核心密度在不同逻辑标准单元数量配置下相对于 VWR2A 架构的趋势图。

探索并比较了五种配置与类似的 CGRA 架构 VWR2A。后布局分析显示，我们的设计实现了超过 2× 倍更低的线长比面积比率和超过 3× 倍更高的核心密度。其他配置显示出一致的结果，证实了在广泛的设计空间中的鲁棒性，并证明这种架构布局是解决使用先进工艺节点带来能效挑战的一个有前景的解决方案。

这些改进是在除了 SPM 宏之外没有指导放置任何子模块的情况下实现的，极大地减少了设计工作量。未来的工作包括评估此类优化并将分析扩展到架构的功耗和热特性。

参考文献

[1] B. Peccerillo *et al.*, “A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives,” *Journal of Systems Architecture*, vol. 129, p. 102561, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762122001138

[2] B. W. Denkinger *et al.*, “Vwr2a: a very-wide-register reconfigurable-array architecture for low-power embedded devices,” in *Proceedings*

of the 59th ACM/IEEE Design Automation Conference, 2022, pp. 895–900.

[3] B. de Bruin *et al.*, "R-blocks: an energy-efficient, flexible, and programmable cgra," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 17, no. 2, May 2024. [Online]. Available: https://doi.org/10.1145/3656642

[4] N. K. Purayil *et al.*, "Araxl: A physically scalable, ultra-wide risc-v vector processor design for fast and efficient computation on long vectors," 2025. [Online]. Available: https://arxiv.org/abs/2501.10301

[5] Z. Tokei, "Scaling the back end of line – a toolbox filled with new processes, boosters and conductors," Sep 2019. [Online]. Available: https://tinyurl.com/37rc32wr

[6] M. R. Baklanov *et al.*, "Advanced interconnects: Materials, processing, and reliability," *ECS Journal of Solid State Science and Technology*, vol. 4, no. 1, p. Y1, dec 2014. [Online]. Available: https://dx.doi.org/10.1149/2.0271501jss

[7] J. Altayo *et al.*, "Addressing memory bandwidth scalability in vector processors for streaming applications," 2025. [Online]. Available: https://arxiv.org/abs/2505.12856

[8] P. Raghavan *et al.*, "Very wide register: An asymmetric register file organization for low power embedded processors," in *Design, Automation & Test in Europe Conference & Exhibition*, 2007, pp. 1–6.

[9] P. Yu *et al.*, "An energy efficient soft simd microarchitecture and its application on quantized cnns," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2024.

[10] F. Catthoor *et al.*, *Ultra-low energy domain-specific instruction-set processors*. Springer Science & Business Media, 2010.

[11] B. W. Denkinger *et al.*, "Acceleration of control intensive applications on coarse-grained reconfigurable arrays for embedded systems," *IEEE Transactions on Computers*, vol. 72, no. 9, pp. 2548–2560, 2023.

[12] S. Yang *et al.*, "Aero: Design space exploration framework for resource-constrained cnn mapping on tile-based accelerators," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 12, no. 2, pp. 508–521, 2022.

[13] S. Srinivasa *et al.*, "Trends and opportunities for sram based in-memory and near-memory computation," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*, 2021, pp. 547–552.

[14] "Cadence® Genus™ Synthesis Solution."

[15] "Cadence® Innovus™ Implementation System."