基于时间数字转换器（TDC）的谐振存内计算：采用层优化静态随机存取存储器映射的 INT8 卷积神经网络

Dhandeep Challagundla，*IEEE*学生会员，Ignatius Bezzam，*IEEE*会员，以及 Riadul Islam，*IEEE*高级会员

arXiv:2601.00434v1 [eess.SP] 1 Jan 2026

摘要——**近年来，存内计算架构已成为深度神经网络加速器的一种有前景的解决方案。乘累加被视为神经网络中的事实标准基本运算。借助存内计算固有的并行处理能力，可以更高效地执行需要大量乘加运算的神经网络。将权重存储在静态随机存取存储器中，进一步减少了大量数据搬移的需求，并提升了整体计算速度和能效。传统存内计算架构在模拟域中执行乘加运算，并使用模数转换器将模拟乘加值转换为数字输出。然而，这些模数转换器会显著增加面积和功耗，还会引入非线性。本文提出一种谐振时域存内计算（TDC-CiM）架构，通过使用时间数字转换器将模拟乘积累加结果数字化，以较低的功耗和面积成本免去模数转换器。专用的 8T SRAM单元可实现可靠的按位乘累加运算，而读出电路采用带有脉冲收缩延迟元件的 4 位 TDC，实现 1 GS/s 采样，功耗仅为 1.25 mW。此外，权重驻留数据映射策略与自动化 SRAM宏选择算法相结合，可在各种卷积神经网络工作负载中实现可扩展且高能效的部署。对六种卷积神经网络模型的评估表明，当静态随机存取存储器大小从 32 千字节扩展到 256 千字节时，该算法可将推理能耗最多降低 8×，同时在量化后将准确度损失维持在最低水平。通过采用台积电28纳米工艺的 8 千字节 SRAM存储器阵列，验证了所提架构的可行性。所提出的 TDC-CiM 架构实现了 320 每秒十亿次运算的吞吐量，能效达到每瓦每秒万亿次运算 38.46。**

索引词—**静态随机存取存储器 (SRAM)、存内计算 (CiM)、卷积神经网络 (CNN)、乘累加 (MAC)、时间数字转换器 (TDC)。**

I. 引言

基于冯·诺依曼架构的神经网络加速器目前正应用于边缘设备，以执行复杂任务。这些网络需要大量存储器

D. Challagundla 和 R Islam 任职于美国马里兰州巴尔的摩县马里兰大学巴尔的摩县分校计算机科学与电气工程系，邮编 21250，电子邮箱：riaduli@umbc.edu。I Bezzam 任职于美国加利福尼亚州米尔皮塔斯市 McCarthy Blvd 1525 号、邮编 95035 的 Rezonent Inc.，电子邮箱：i@rezonent.us。本研究部分由美国国家科学基金会（NSF）项目资助，项目编号：2138253；Rezonent Inc. 项目资助，项目编号：CORP0061；以及 UMBC 初创基金资助。版权所有 (c) 2025年 IEEE。允许将此材料用于个人用途。但如需将此材料用于任何其他用途，必须发送电子邮件至 pubs-permissions@ieee.org，向 IEEE 获取许可。

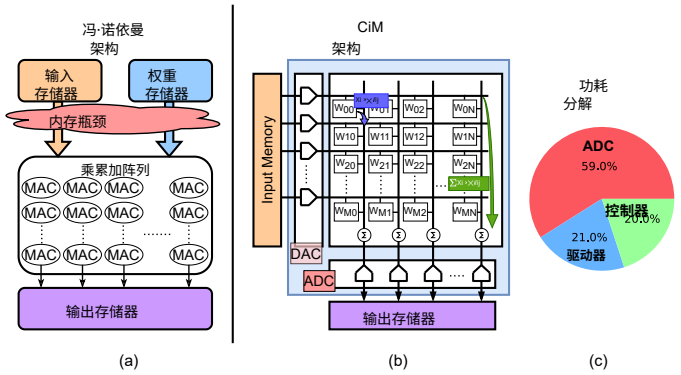

图 1. (a) 传统神经网络加速器使用处理单元进行乘累加计算，由于数据搬移量大，导致存储墙瓶颈。(b) 存内计算架构通过在存储单元中利用电荷累积实现模拟乘累加计算，并使用模数转换器，从而减少频繁数据搬移的需求；(c) 传统存内计算架构的功耗分解显示，模数转换器造成了总功耗的 59% [8]。

来访问输入和权重，从而造成存储墙瓶颈，降低处理器性能，如图 1(a) 所示。存内计算架构通过利用动态随机存取存储器 [1]–[3]和静态随机存取存储器 [4]–[7] 存储阵列内的并行计算，降低数据搬移带来的能耗开销。与传统神经网络一样，乘累加（MAC）运算是存内计算中的基本操作。存内计算架构并行执行输入与权重之间的多次乘法，并在电流域中将结果求和。位线电压对应这些乘累加运算的最终输出，如图 1(b) 所示。随后，使用模数转换器将模拟电压转换为数字输出位。然而，如图 1(c) 所示，这种模拟处理在存内计算架构的总功耗中占比过高（最高可达 59%） [8]。

与其他节能技术 [9]–[18] 不同，尽可能降低模数转换器带来的开销对于提升存内计算加速器的能效至关重要。一些存内计算架构通过降低稀疏输入下的模数转换器精度来实现这一目标，而另一些方案则采用结合非线性量化技术的低精度模数转换器 [19]。在混合信号设计中集成模数转换器的一个主要问题

架构（例如存内计算）需要确保模数转换器在各种工作条件下的性能一致，并能随工艺扩展。此外，由于工艺变化的影响，对模拟电路与数字组件进行验证需要采用成本高昂的先进测试方法。

为克服模数转换器的非线性和高功耗，越来越多的研究聚焦于时域模数转换器（TD模数转换器）：这类转换器将模拟电压转换为延迟，以便在数字域中进行处理 [20], [21]。在这项工作中，我们通过引入一种无模数转换器谐振时域存内计算（TDC-CiM）架构来缓解存内计算中的模数转换器问题。该架构采用时间数字转换器（TDC），在静态随机存取存储器存储单元中执行乘加运算，其功能与模数转换器相同，同时模拟数字电路的行为 [22]。具体而言，这项工作的主要贡献如下：

- 首次提出利用新型TDC执行乘加运算的无模数转换器谐振存内计算架构。
- 专用读端口8T SRAM单元，可实现无读干扰的逐位乘法。
- 一种自动化SRAM宏选择算法，可在权衡卷积核维度、所需并行度和权重驻留映射约束的基础上，将任意给定的量化卷积神经网络工作负载映射至能效最高的TDC-CiM配置。
- TDC功能验证与稳健性分析表明，其能效达到每转换步163 fJ。

II. 背景

作为一种新兴范式，基于静态随机存取存储器的存内计算架构在显著提升多种计算任务的处理速度和能效方面展现出巨大潜力，例如乘累加 [23]–[27]，内容可寻址存储器 [28],以及布尔逻辑 [29], [30]。此外，串联 [31]–[33] 和并联谐振 [34]–[38] 在能效计算方面也展现出巨大潜力。串联谐振可支持宽广的工作频率范围，而本研究针对静态随机存取存储器特性表征中采用的固定工作频率，采用并联谐振方案。

采用标准6T静态随机存取存储器单元的存内计算架构 [23]–[25] 通过沿位线传播信息来执行计算。这种计算方法需要激活多行，以便在单个访问周期内读取多个操作数。然而，这会导致位线上的电压变化幅度较大，从而引发严重的读扰问题，并可能导致数据损坏。 [27] 采用7T静态随机存取存储器单元执行乘加运算，利用读位线放电来消除读扰问题。然而，如果读位线电压大幅下降，这种方法可能会导致反向电流流动；当某一行未能放电时，反而可能使位线电压升高，从而在乘加计算中引入非理想因素并影响准确度。 [26] 采用9T1C架构执行乘加运算，但延迟较高，主要是因为按位乘法运算期间为电容器充电耗时较长。 [28] 采用11T静态随机存取存储器单元以执行布尔运算和内容寻址存储器运算，同时无需遵循按列计算的限制。然而，由于静态随机存取存储器单元中使用了额外的晶体管，其面积开销较大。本工作采用8T静态随机存取存储器位单元，可提供完整的读位线摆幅，避免 [27], 中报告的反向电流问题；与 [28]中的11T位单元相比，还能降低面积开销。

存内计算架构通常采用两种设计方法执行乘加运算：基于模拟的 [26] 和数字计算 [39]。第一类存内计算架构使用电流 [40]、电压 [41], 或脉冲宽度 [42] 来表示信息并执行乘加运算。这类架构的功耗较低，但会受到各种非理想因素的影响。相较之下，数字存内计算架构在执行多位运算时计算结果准确，但功耗较高。电荷域存内计算十分有效，因为累加在电容器上进行，电荷能够以较低损耗线性叠加，并且可以激活多行，同时保持准确度 [43]–[44]。采用在工艺、电压和温度（PVT）变化下稳定的金属-氧化物-金属电容器后，这些阵列每次运算支持的并行度高于电流域设计，且能耗更低 [45], [46]。在许多先前系统中，读出端的模数转换器一直是性能瓶颈。提高模数转换器的分辨率以实现更高精度的运算，虽能降低量化误差，却也会缩小1个最低有效位的电压步进。因此，相同的噪声会占据更大比例的一个码，模数转换器错误也会更加频繁。模数转换器的模拟前端还会占据主要功耗，且难以在不同工艺角下得到保障，从而增加能耗并引入部分和误差，降低整体准确度 [47]。为解决这些问题，本文提出一种无模数转换器存内计算架构，该架构执行乘加计算，并通过数字时间数字转换电路将模拟乘加值转换为数字输出。TDC-CiM采用基于电容器的计算实现低能耗计算，同时避免模数转换器的开销：它将乘加电压摆幅映射为时间，并通过由数字模块构成的紧凑型TDC进行数字化。TDC与乘加放电范围协同设计，可提供单调的4位读出结果；如有需要，可通过小型查找表将其线性化，同时保持INT8推理精度，并通过从转换环路中移除高分辨率模数转换器来降低转换器功耗。

文献中已报道了大量关于TDC电路的研究。 [48]中的TDC架构采用一个与锁相延迟环相结合的延迟元件，并通过计数器对脉冲数量进行计数，从而产生数字输出。另一种方法 [49]采用两条脉冲收缩延迟线和一个延迟稳定环，将脉冲输入转换为数字输出。 [21] 采用两步时间数字转换器和电压—时间转换器，为给定输入产生数字输出。与以往的TDC研究不同，本研究采用简单的脉冲收缩延迟元件和D触发器，根据输入电压捕获脉冲计数，并进一步将其编码为数字位。

III. 所提出的架构

在卷积神经网络（CNN）中，乘累加（MAC）运算是一项基本计算。本文提出了一种新颖的存内计算（CiM）架构，通过在存储器结构内部执行乘加运算，增强标准SRAM缓存。该设计如图2所示，将传统的SRAM阵列与二进制加权电容阵列相结合，通过电荷共享对8 位输入和8 位权重执行模拟乘积累加运算。由所提出的时间数字转换器构成的读出电路，将模拟MAC值转换为两组4位数字值，再将其相加，形成8位乘加输出。

A. 所提出的*TDC*存内计算架构

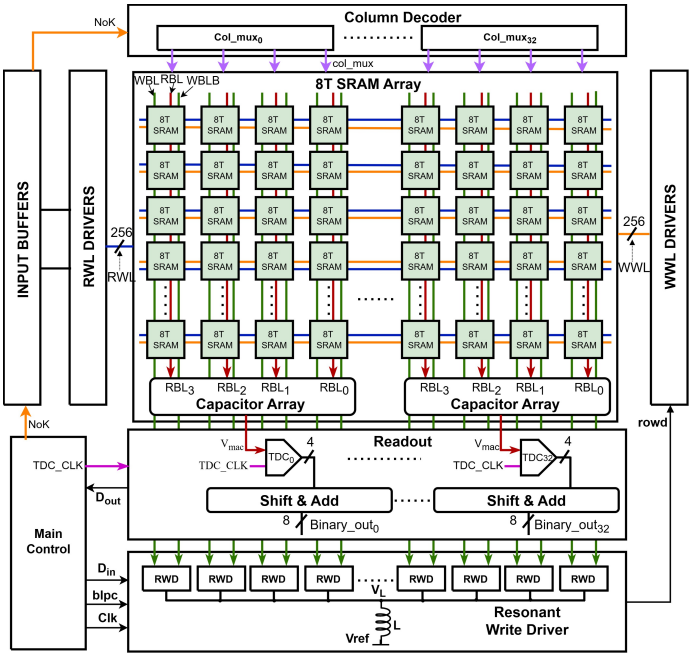

图2. 所提出的架构包括8T位单元阵列、用于传输输入的行字线驱动器、用于执行乘加运算的新型基于TDC的读出电路，以及用于回写乘加输出的谐振写入驱动器。

图2展示了所提出的TDC存内计算8T SRAM宏的架构，该架构旨在实现乘加运算。该结构包含一个8T SRAM 阵列、一个二进制加权电容器阵列，以及一个专门配置用于乘加运算的读出电路。乘加结果的回写通过能量回收谐振写入驱动器实现，从而支持低功耗写操作 [50]。借助用于写字线（WWL）和读字线（RWL）的行译码器，以及用于生成内部信号的外围控制电路，该架构支持与传统SRAM架构类似的常规读写操作。

8T SRAM 阵列由 256×256 个位单元组成，用于存储和读取数据，并采用垂直位线和水平字线。在传统的读写过程中，该架构设计为每次仅激活一个 RWL 驱动器或一个 WWL 驱动器。然而，九个 RWL 驱动器会同时启用，以对应输入特征图（IFM）的 3×3 内核，从而执行乘加运算。本工作采用权重驻留数据流方案，详见第三节E。在乘加运算过程中，列译码器会选择存储了权重的相应列。主控制单元生成一个指示卷积核数量（NoK）的信号，再对其进行译码以激活相应的列。8位滤波器权重

(W)以逻辑“1”或“0”的形式存储在位单元中，而8位 $IFMs$ 在2个周期内以一系列模拟脉冲的形式施加到 $RWLs$上。

B. 提出的多位乘法

在所提出的架构中，8位滤波器权重沿行方向存储在两组各含4个相邻8T SRAM位单元的单元组中。滤波器权重在TDC-CiM阵列中的映射如图4所示。8位 $IFMs$ 在随后两个时钟周期内以一系列4位输入脉冲的形式输入TDC-CiM阵列，以执行乘累加运算。

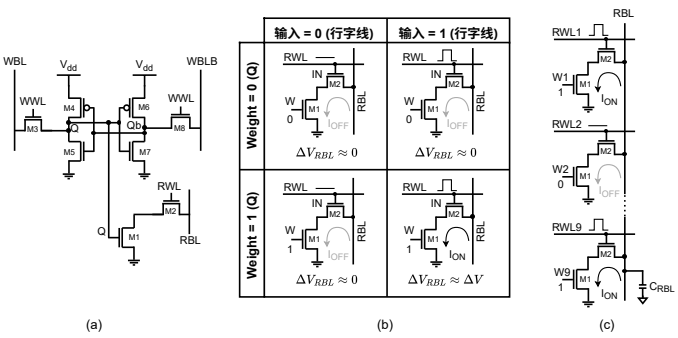

图3。（a）带有 $M1-M2$ 晶体管构成专用读取端口的8T SRAM位单元示意图；（b）使用8T位单元对输入和权重进行按位乘法，仅当输入和权重均为“1”时 ΔV 才会放电；（c）多行激活执行一系列按位乘法，共同完成按位乘积累加运算。

图3(a)展示了用于执行乘加运算的8T位单元晶体管级原理图。图3(b)说明了使用8T静态随机存取存储器位单元进行按位乘法的过程。在这里，输入 =“1”用单位脉冲表示，而输入 = “0”表示没有脉冲。当权重（Q值）设为“1”，且输入也为“1”时，会产生电流，使RBL线的电压降低 ΔV。因此，它在单个位单元中对存储的权重和输入执行1位乘法。由放电导致的电压降 ΔV 可以通过量化电容器（C_{RBL}）损失的电荷量（Q）来测得。电容器上剩余的电荷可写为：

$$Q_{\text{remaining}} = Q_{\text{initial}} - Q_{\text{discharged}} \tag{1}$$

由于 $Q = CV$，电容器（C_{RBL}）上的剩余电压（V_{rem}）为：

$$V_{\text{rem}} \cdot C_{RBL} = (V_{\text{dd}} \cdot C_{RBL}) - (I_{\text{ds}} \cdot T_{\text{dis}}) \tag{2}$$

其中，I_{ds} 是在时间段 T_{dis}内放出的电流。求解 V_{rem}：

$$V_{\text{rem}} = V_{\text{dd}} - \frac{I_{\text{ds}} \cdot T_{\text{dis}}}{C_{RBL}} \tag{3}$$

因此，电压降 ΔV 可以量化为：

$$\Delta V = V_{\text{dd}} - V_{\text{rem}} = \frac{I_{\text{ds}} \cdot T_{\text{dis}}}{C_{RBL}} \tag{4}$$

同时启用多行，使我们能够并行执行一系列此类单比特乘法，从而在 RBL 上产生集体放电，该放电表示各个乘法运算的总和，如图3(c)所示。对于已激活的位单元，此过程实质上在 RBL 上实现了按位乘加函数。即使按位乘法的结果为“0”，使用 8T SRAM 位单元也能克服 6T/7T 位单元中的反向充电电流限制。

由于 n 个活动行并行工作， RBL上的总电压降可表示为：

$$\Delta V_{\text{total}} = \left(\sum_{i=0}^{n} X_i \cdot W_i \right) \cdot \left(\frac{I_{\text{ds}} \cdot T_{\text{dis}}}{C_{RBL}} \right) \tag{5}$$

其中， X_i 和 W_i 分别表示输入特征和卷积核权重。按位乘加结果采样于两组电容器上：二进制加权按位电容和累加电容器 C_{acc}，如图4所示。二进制加权按位电容根据激活的行数放电。按位乘法完成后，$col\ mux$信号将切换为“ON”，使位电容与 C_{acc}之间实现电荷共享。 C_{acc} 输出节点的电压即为与输入和权重的模拟乘加结果相对应的 V_{mac} 电压。

图4展示了使用8T SRAM 单元阵列执行乘加运算的卷积神经网络层映射过程。RWL驱动器将8位 IFM 作为输入提供给静态随机存取存储器单元的行字线。输入特征图中的逻辑“1”会在行字线上产生一个单位脉冲，而逻辑“0”则不会产生脉冲。图5展示了该乘加运算的SPICE仿真结果。

8位固定值形式的卷积核权重水平存储在同一行内、紧邻8T SRAM 位单元的位置。每条读位线最初都连接到一个二进制加权电容器。在乘法阶段，初始预充电的读位线会根据激活行的数量逐步放电。图中所示的配置包含九条字线，对应一个 3×3 卷积核，这种卷积核通常用于卷积神经网络层。按位乘法过程完成后，我们将$col\ mux$信号置为“ON”，使二进制加权电容器连接到 C_{acc}。这会启动电荷共享机制，为.....充电

C_{acc}基于二进制加权电容器组合后的电荷产生最终累积电压V_{mac}，如图5所示。

累积电压 V_{mac} 在 C_{acc} 中以模拟形式表示乘加运算的结果。该模拟值 V_{mac}与 $TDC\ CLK$ 脉冲一起输入时间数字转换器。时间数字转换器将该 V_{mac} 电压转换为 4 位数字输出（$Binary\ out < 3:0 >$），表示部分乘加值；本例中该值为“11”。

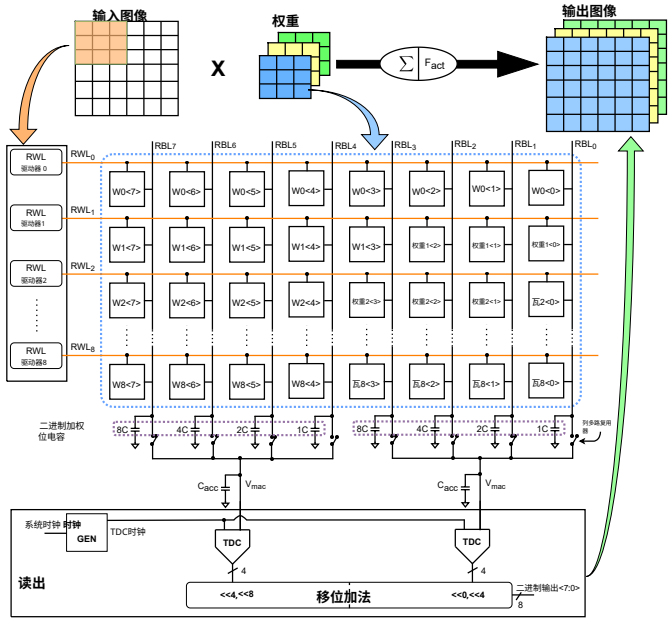

图4. 卷积运算的层映射：通过读字线驱动器施加输入数据，与静态随机存取存储器单元中的固定权重相乘，并通过二进制加权电容器沿每个位线执行按位乘累加运算，最终在 C_{acc}上通过电荷共享得到多位乘加输出；所得电压 V_{mac} 由时间数字转换器数字化。

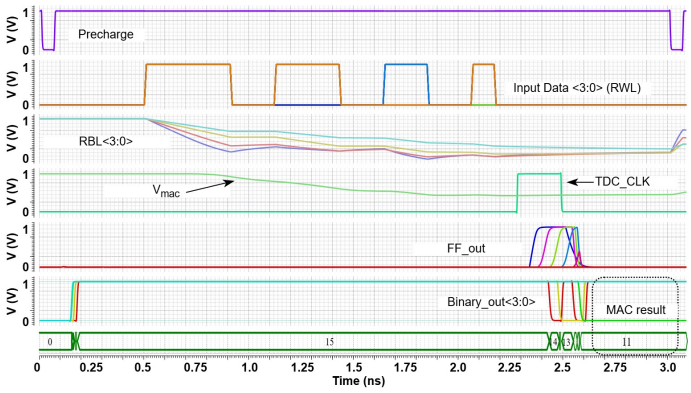

图5. 对 $RBLs$ 执行按位乘加运算并进行电荷共享的功能仿真，得到模拟乘加电压 V_{mac} ，再由时间数字转换器进行数字化。

在所提出的方案中，4 位输入序列与两组 4 位权重相乘，产生两个独立的 4 位部分乘加值。这些部分乘加结果在两个时钟周期内合并。在第一个时钟周期中，对输入特征图的低 4 位进行处理，相应的部分输出分别左移 0 位和 4 位。在第二个时钟周期中，以相同方式处理输入特征图剩余的高 4 位，部分输出分别左移 4 位和 8 位。随后将两个周期的移位部分和相加，得到最终的 8 位乘加结果。该最终和存储在输出缓冲区中，随后通过谐振写入驱动器 [51]写回 SRAM 阵列，以供下一层使用。

RBL 和 C_{acc} 的模拟放电速率

相对于活动行数量的关系如图 6 所示。随着激活的行数增加，RBL 和累加电容器的放电量也随之增加。

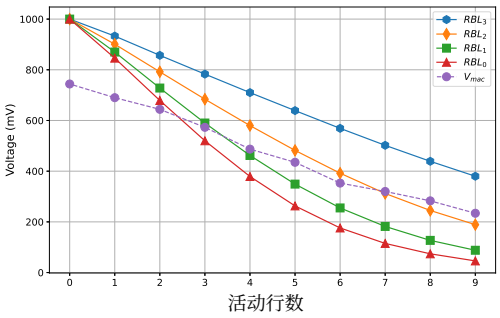

图 6. 读位线（RBL）和累加电容器 C_{acc} 的放电速率表明，随着激活的行数增加，电压呈线性下降。

C. 提出的 TDC 架构

图7展示了所提出的 4 位 TDC。它由一组脉冲收缩延迟元件及其对应的 D触发器（$DFFs$）组成。这些$DFFs$的输出（Q）作为温度计码转二进制编码器的输入，从而得到 4 位二进制输出。TDC电路的整体 SPICE 仿真如图8所示。TDC 接收乘加运算的模拟输出，该输出以电压（V_{mac}）的形式存储在 C_{acc}中，并将其转换为数字信号。TDC 还具有一个输入脉冲 $TDC\ CLK$，该脉冲由系统时钟通过基于缓冲器的延迟电路生成。

图7所示的晶体管 $M1-M5$ 构成脉冲收缩延迟元件，该元件由电压控制缓冲器组成。输入脉冲（$TDC\ CLK$）的上升沿传播速度会被限流晶体管 $M3$ 减慢，而下降沿则快速传播。按位乘法后通过电荷共享得到的V_{mac} 结果，作为电压控制输入提供给 TDC电路。每个延迟元件都会根据 V_{mac} 结果将输入脉冲收缩 ΔT。随着脉冲沿延迟线传播，每经过一个元件，脉冲宽度都会收缩 ΔT，直至脉冲完全消失。

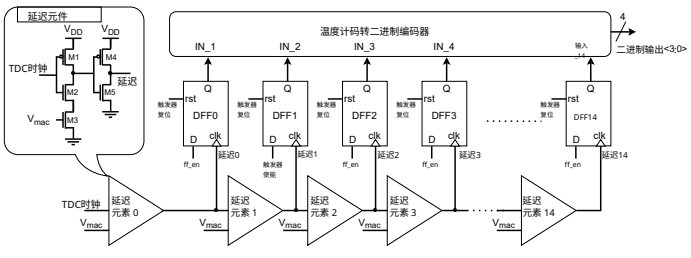

图7. 4 位 TDC 转换器架构由一组脉冲收缩电压控制延迟元件和 D触发器组成，用于生成与模拟乘加值 V_{mac} 相对应的脉冲计数，再通过基于多路复用器的温度计码到二进制编码器转换为数字输出。

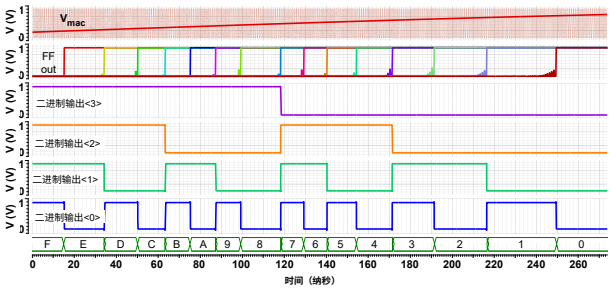

图8. TDC 的功能仿真表明，数字输出会随着输入电压斜坡信号 V_{mac} 线性下降，成功捕获了 4 位数值的完整范围。

每个脉冲收缩延迟元件都连接到正边沿触发型 DFF 的时钟引脚。DFF 会利用传播的脉冲锁存 “1”，直至脉冲消失，从而使后续的 DFF 保持“0”。图8所示的仿真通过施加电压斜坡信号，捕捉了将所有 $DFFs$ 置为“1”所需的电压。

Q 来自 $DFFs$，随后作为输入送至基于多路复用器的温度计码到二进制编码器。该转换器将一串“1”和“0”转换为一个4位二进制数。该二进制数是原始模拟电压的数字表示。此4位数字输出作为输入送至移位加法模块，用于最终的8位乘累加输出计算。

D. 谐振写入驱动器

如图9所示，写入路径采用了一种基于先前工作的能量回收型谐振驱动器，并配有电源升压功能 [51], [52]。串联电感位于写位线 $Wbl/Wblb$ 的放电路径中，另一端连接至参考电压 $Vref$。选择 $V_{\mathrm{ref}} = V_{\mathrm{DD}}/2$ 可最大限度地节省串联电感带来的能耗 [51]。在写操作期间，Wbl 或 $Wblb$ 从逻辑“1”切换到逻辑“0”时，原本会耗散的电荷将被转移并存储在 V_{ref} 节点。在随后的预充电阶段，这些存储的能量从 V_{ref} 返回位线，从而减少电源提供的能量，并使写入—预充电周期内的净电流为零。

图9中的谐振写入驱动器使用晶体管$M5-M8$，通过有条件地将 $Wbl/Wblb$连接到归纳器来实现谐振；该连接由系统时钟产生的 vsr信号控制。在图10所示的写入逻辑“1”情形中，vsr 信号使能 $Wblb$ 信号的放电路径，以写入数据“1”。vdn 信号则使能位线完全放电。在随后的预充电阶段，再次置位 vsr，使归纳器中存储的能量返回位线。因此，预充电使能（$BLPC$）无需将位线从0V开始充电，从而降低整体能耗。

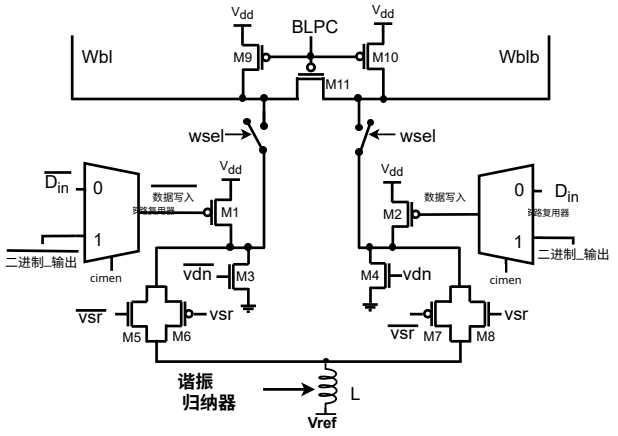

图 9. 与传统写入驱动器相比，谐振写入驱动器额外增加了 4 个晶体管（M5 至 M8）和一个共享电感，从而能够在每次写操作期间实现能量回收。

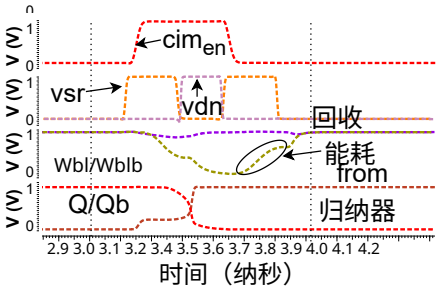

图 10. 谐振写入驱动器的功能验证，展示了在启用 cim_{en} 时 TDC输出（$binary\ out$）的写操作。

E. 权重驻留数据映射方案

图11展示了所提出的面向卷积神经网络工作负载的TDC-CiM宏单元所采用的完全权重驻留数据映射方案。在该方案中，每个卷积神经网络层的8位卷积核权重首先被加载并驻留在SRAM存储体中，只需一个初始化周期。这种配置将卷积核权重保存在各存储体本地，从而最大限度地减少不必要的数据搬移；与此同时，计算期间，输入特征图（IFM）通过行字线（RWL）驱动器动态输入。在数据映射方面，每个TDC-CiM行存储8位权重，而输入缓冲区使用两行，分别存储来自不同通道的输入特征图的4位最高有效位和4位最低有效位。权重驻留布局还为输入特征图在各存储体间的映射方式提供了更大的灵活性。当卷积核数量较多时，同一输入特征图可以并行广播至所有存储体，使每个存储体使用相同的输入数据处理不同的卷积核组。相反，当卷积核数量较少时，可以将输入特征图的多个副本分发到不同存储体，从而在同一周期内并行计算多个输入特征图分块。根据各层的需求，在卷积核并行与输入并行之间进行选择，有助于平衡吞吐量和利用率。每次矩阵向量乘法后，输出特征图将被写回

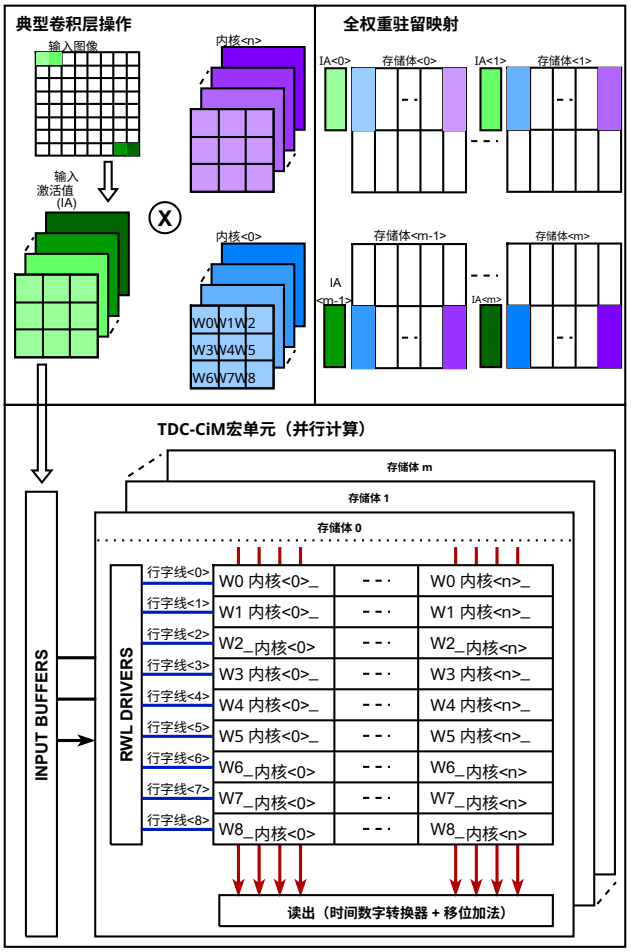

图 11. 所提出的 TDC-CiM宏单元采用完全行驻留权重映射方案，每个计算周期只需发送输入激活值，从而显著降低权重片外访问的延迟。

写入各存储体中，为下一层做好准备。该映射策略通过利用本地数据复用，显著减少片外权重传输并提高能效。

例如，在典型的 LeNet-5 层中，与大型现代卷积神经网络相比，卷积核数量相对较少。在这种情况下，所提出的权重驻留方案可将数量有限的卷积核映射到部分可用的 SRAM 存储体中，让其余存储体在同一计算周期内并行处理多个输入特征图分块。通过将输入数据的不同部分分配到各个独立存储体中并发处理，即使卷积核数量较少，该架构也能保持较高的利用率和吞吐量。对于卷积核数量较多的层，同一映射策略则可将单个输入特征图广播到所有存储体，以同时处理更多卷积核。这种灵活性确保 TDC-CiM 宏单元能够适应各种卷积神经网络层，同时减少冗余的存储器访问并最大限度地提高并行计算能力。所提出的权重驻留映射方案的控制电路包括一个可编程行译码器，可根据输入特征图位置选择性地激活各存储体中的多个行字线，以及一个列译码器

该列译码器能够索引存储在位单元中的静态卷积核权重。主控制器协调这些组件，实现输入特征图切片在各存储体间的同步广播和顺序乘加运算。此外，还需要周期级控制逻辑，以管理8位输入特征图的多周期处理；每次乘加运算使用两个4位部分累加。

算法 I. 将量化卷积神经网络工作负载映射到最优 TDC-CiM 架构 1：输入：预训练神经网络（NN）、SRAM宏（$SRAM_{list}$）；2：输出：rTD–CiM 架构；3：$QNN \leftarrow Quantize$（NN，8-bit）；▷将神经网络量化为 8 位整数精度 4：$LayerInfo$，$Weights \leftarrow Extract$（$QNN$）；▷提取逐层信息和权重 5：$SRAM_{size} \leftarrow IdentifySRAM$（$LayerInfo$，$Weights$）；▷ 根据层信息和卷积核数量确定一组 SRAM；6：对于所有 $SRAM$，执行 $SRAM_{size}$；▷遍历每个 SRAM 宏 7：$Metrics_{list}[SRAM] \leftarrow Evaluate$（$LayerInfo$，$Weights$，$SRAM$）；▷评估所选每个 SRAM 宏的功耗、延迟和能耗 8：结束循环 9：$BestSRAM \leftarrow SortEnergy$（$Metrics_{list}$）；确定能耗最低的 AIG 10：$L_{res} \leftarrow CalculateInductor$（$BestSRAM$）；▷ 计算所选 SRAM 的谐振电感器尺寸 11：输出：TDC-CiM 架构 $\leftarrow \{BestSRAM, L_{res}\}$；▷卷积神经网络模型所生成的 TDC-CiM 架构

算法 I 描述了如何选择最佳 SRAM 宏单元大小，以将量化卷积神经网络工作负载映射到所提出的 TDC-CiM 架构。该算法的输入是一个预训练神经网络和可用 SRAM 宏配置的列表，输出则是针对给定模型定制的 TDC-CiM 硬件配置。首先，在第 3 行将预训练卷积神经网络量化为 8 位整数精度，以实现高效的低比特存内计算。接着，第 4 行从量化后的模型中提取逐层结构信息及相应权重。利用这些信息，第 5 行根据层维度、参数数量和不同的并行度，筛选出一组适用的 SRAM 宏单元大小。

对于每个选定的 SRAM 宏，算法会在第 6–7 行评估其功耗、延迟和能耗表现，同时考虑特定工作负载和硬件约束。评估完成后，第 9 行的筛选步骤会选出总能耗最低的配置。最后，第 10 行计算所选 SRAM 大小所需的谐振电感器值。此方法将得到针对特定工作负载的 TDC-CiM 架构，在保持量化精度的同时最大限度地提高能效和性能。

IV. 实验结果

A. 实验设置

采用 28 nm台积电技术设计并仿真了一个 8 KB SRAM存储器实例。该 SRAM存储器阵列由 256×256 8 T 个位单元组成，使用 Cadence Virtuoso 实现，并使用 Cadence Spectre仿真器在 0.5 GHz 时钟频率下进行仿真。输入由九个行字线驱动器驱动，对应于 IFM 的 3×3 输入内核。执行乘加运算的电容器阵列由二进制加权按位电容和一个 C_{acc} 组成。按位最低有效位电容值设为 $1C = 4$ fF，而存储模拟乘加输出 V_{mac} 的累加电容器设为 32 fF。读出电路由 64 个 TDC模块组成，每个模块连接到一个电容器阵列。

系统级基准测试使用了六种神经网络模型，分别为 LeNet-5 [53]（MNIST [54]）、MobileNetV1 [55] 和 MobileNetV2 [56]（CIFAR-10 [57]）、SqueezeNet [58] 和 ResNet-18 [59]（ImageNet-1K [60]），以及 Tiny-YOLOv3 [61]（COCO [62]）。所有模型均采用训练后量化方法量化至 8 位精度，且未进行进一步微调。针对每个网络应用所提出的 SRAM 宏选择算法，以确定能最大限度降低能耗和推理延迟的最佳 SRAM 大小。能耗和延迟指标通过将每一层映射到相应的 SRAM 配置，并累加整个网络推理过程中逐层的估算值而得出。

B. TDC特性表征与比较

图12展示了用于分析TDC线性度的传输特性。施加于TDC的输入电压绘制在x轴上，基线偏移为 200 mV。所提出的TDC电路的完整电压范围为 200 mV至 800 mV。TDC的理想性能由蓝线表示，其特征是保持一致的单调递减函数。相比之下，所提出TDC电路的模拟实际传输特性以红色显示，偏离了线性函数。该特性与图6所示 C_{acc} 上观察到的线性放电相一致。实际传输曲线的偏移可以通过进行微分非线性（DNL）校正

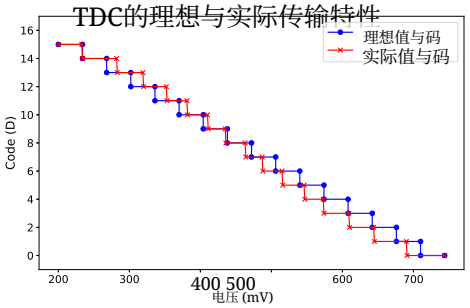

图 12。TDC 的仿真传输特性显示，其偏离线性理想特性曲线，但跟踪模拟乘累加输出电压 V_{mac}。

并进行积分非线性（INL）分析，分别如图13和图14所示。

图13展示了所提出的时间数字转换器的微分非线性（DNL）特性。DNL用于衡量连续数字码之间的实际步长相对于理想步长的偏差，以最低有效位（LSB）表示。DNL值为正表示两个连续数字码之间的实际步长大于1个LSB；DNL值为负则表示步长小于1个LSB。如果DNL值达到-1 LSB，则意味着出现缺码，这是时间数字转换器中一种严重的非线性误差。不过，图中所有DNL值均在 ± 0.5 LSB以内，属于时间数字转换器特性表征中可接受的范围。该图显示，数字码“0”的最大DNL值为 +0.4 LSB，数字码“7”的最小DNL值为-0.3 LSB。

图14展示了所提出的时间数字转换器（TDC）的积分非线性（INL）图。INL用于衡量转换器的线性度，表示在转换器的整个范围内，输入到输出的映射相对于理想函数的最大偏差。INL图呈现出初始正偏差，表明TDC早期的输出码大于理想线性系统的预期值。随着输入值增大，INL曲线逐渐下降，最终低于零，说明TDC的输出码逐渐低于线性模型的预测值。该图显示，数字码“12”对应的INL最大值为 +0.5个最低有效位，数字码“2”对应的INL最小值为-1个最低有效位。

表 I 比较了多篇设计文献中的各种模数转换器架构。本研究所用时间数字转换器的分辨率为 4 位，采样率为 1 GS/s。该时间数字转换器的电源电压为 1V。其信号噪声失真比为 19.45 dB，无杂散动态范围为 22.4 dB。该时间数字转换器以 V_{mac} 作为输入，无需使用传统时间数字转换器架构通常所需的电压-时间转换器。这种方法降低了时间数字转换器整体框架的功耗。与 [20] 相比，所提出的时间数字转换器功耗降低了 71%；与 [21]，相比，功耗降低了 45.6%，后两者均为时域模数转换器（$TD\ \ ADC$）。此外，与 [64] 相比，所提出的时间数字转换器功耗降低了 98%；与 [63]，相比，功耗降低了 96%

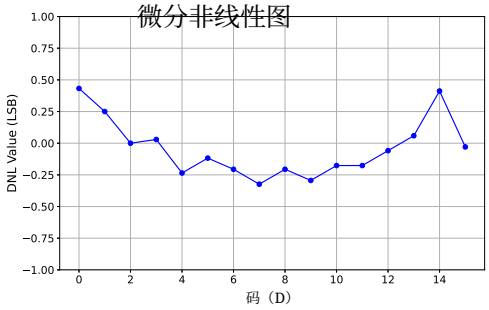

图13。DNL特性图显示了-0.3/+0.4个最低有效位的可接受偏差。

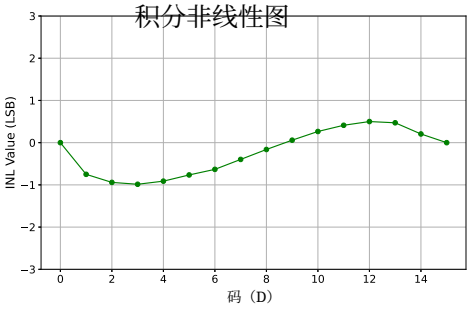

图 14. TDC 的 INL 特性图显示，数字码“2”的最小偏差为 -1 个最低有效位，而“12”的最大偏差为 +0.5 个最低有效位。

它们是 $SAR\ \ ADCs$，通常用于传统存内计算架构。该时间数字转换器的 Walden 品质因数（FoM）为 162.8 fJ/转换步，比 [63] 高 12%，比 [64] 高 14%。这一与位分辨率直接相关的品质因数，可通过增加时间数字转换器的输出位数来降低。

C. 工艺、电压和温度变化分析

为评估所提出的 TDC 设计的稳健性，我们进行了大量蒙特卡洛仿真，以考虑工艺、电压和温度变化 [65]。我们针对供电电压变化对所提出的 TDC 进行了蒙特卡洛分析，在每个电压角下使用 2000 个样本，供电电压相对于标称值变化 $\pm 10\%$。结果如图 15 所示，展示了供电电压对功耗和延迟特性的影响。在 $V_{\mathrm{DD}} = 0.9\mathrm{V}$ 时，平均功耗为 $334.8\ \mu\mathrm{W}$，标准差为 $1.72\ \mu\mathrm{W}$；平均延迟为 247 ps，标准差为 2.68 ps。将供电电压提高至 $V_{\mathrm{DD}} = 1\mathrm{V}$ 后，平均功耗为 $432\ \mu\mathrm{W}$（$\sigma = 2.29\ \mu\mathrm{W}$），平均延迟为 285 ps（$\sigma = 2.09$ ps）。在 $V_{\mathrm{DD}} = 1.1\mathrm{V}$ 时，平均功耗进一步增至 $512.37\ \mu\mathrm{W}$，标准差为 $2.93\ \mu\mathrm{W}$；平均延迟则降至 309.56 ps，标准差为 1.79 ps。延迟变化的均值对应于单个延迟级的脉冲宽度，脉冲宽度越大，整体 TDC 延迟越低。这些结果证实，较高的供电电压会增加功耗但降低延迟，体现了 TDC 设计中电压缩放所预期的权衡。

我们还针对所提出的时间数字转换器进行了温度变化下的蒙特卡洛分析，每个温度角使用 2000 个样本。温度在三个具有代表性的工作点之间变化：$0\,^{\circ}\mathrm{C}$、$27\,^{\circ}\mathrm{C}$ 和 $100\,^{\circ}\mathrm{C}$，以评估热条件对功耗和延迟性能的影响。如图 16 所示，在 $0\,^{\circ}\mathrm{C}$ 时，平均功耗为 $430.6\ \mu\mathrm{W}$，标准差为 $3.4\ \mu\mathrm{W}$；平均延迟为 287.5 ps，标准差为 2.05 ps。在标称温度 $27\,^{\circ}\mathrm{C}$ 下，平均功耗为 $432\ \mu\mathrm{W}$，标准差为 $\sigma = 2.29\ \mu\mathrm{W}$，平均延迟为 285 ps，标准差为 $\sigma = 2.09$ ps。在 $100\,^{\circ}\mathrm{C}$ 的高温工作条件下，平均功耗增加至 $485.44\ \mu\mathrm{W}$，标准差为 $2.13\ \mu\mathrm{W}$，而平均延迟

表I 与先前时间域和逐次逼近型模数转换器架构的比较显示，该时间数字转换器架构的功耗分别比 [20]，低45.6%，比 [63]低96%。

参考	APCCAS'22 [20]	欧洲固态电路会议'23 [63]	射频集成电路'18 [64]	JSSC'19 [21]	本工作
架构	时间域 模数转换器	逐次逼近 模数转换器	逐次逼近 模数转换器	时间域 模数转换器	TDC
工艺（nm）	28	28	28	65	**28**
分辨率（位）	9	6	8	8	**4**
采样率（GS/s）	0.5	1.4	8.8	1	**1**
电源（V）	0.9	0.9	1.5	1	**1**
信号噪声失真比（dB）	54.69	67	38.4	45	**19.45**
无杂散动态范围（dB）	55.16	NR	48.9	60.3	**22.4**
功耗（mW）	4.27	32	83.4	2.3	**1.25**
品质因数（fJ/转换步长）	19.29	143.2	139.5	18.7	**162.8**

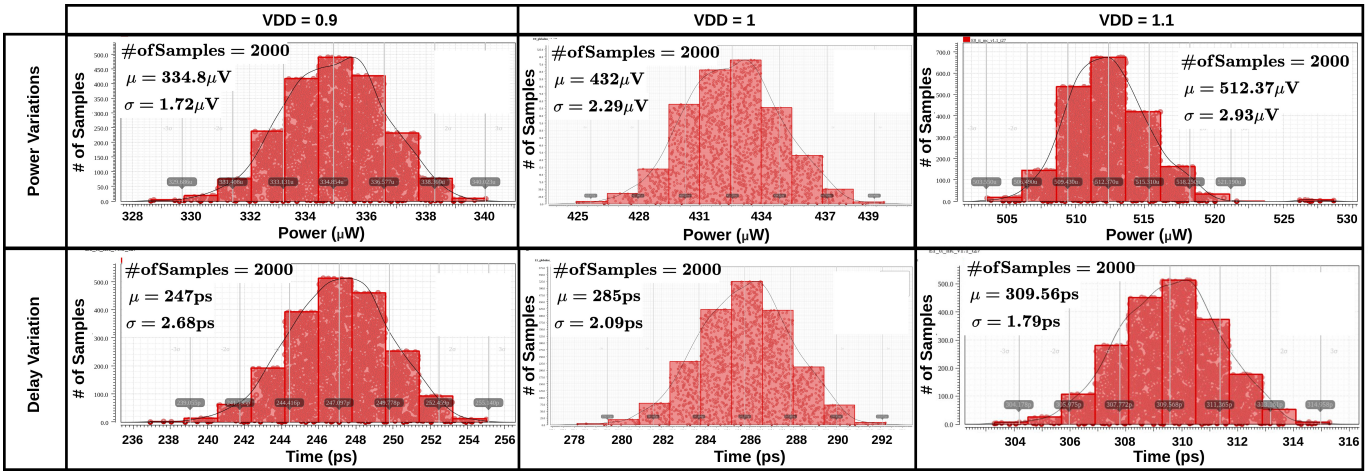

图 15. 所提出 TDC 在电源电压变化下的蒙特卡洛分析。顶行分别显示 V_{DD} = 0.9V、1V 和 1.1 V 下的功耗变化。底行显示各电压角对应的延迟变化分布。每种情况下均使用 3σ 工艺变化和 ±10% 电源电压变化进行了总计 2000 次样本仿真，结果表明，随着电源电压升高，平均功耗增加，而平均延迟降低，这与延迟级中电压缩放的预期行为一致。

则降至 279.82 ps，标准差为 2.2 ps。与电压变化结果类似，延迟变化的平均值代表单个延迟级的脉冲宽度；脉冲宽度越大，整体时间数字转换器延迟越低。这些结果表明，温度升高会略微增加功耗并降低延迟，这是因为高温会增强载流子迁移率。

D. 使用神经网络模型进行基准测试

我们在六种不同的神经网络模型上评估了所提出的 TDC-CiM 架构，模型规模从具有 60K 个参数的小型网络到具有超过 11M 个参数的大型模型不等。具体而言，我们使用了在 MNIST [54] 数据集上训练的 LeNet-5 [53]模型、在 CIFAR-10 [57]数据集上训练的 MobileNetV1 [55]和 MobileNetV 2 [56] 、在 ImageNet-1K [60]数据集上训练的 SqueezeNet [58] 和 ResNet，以及在 COCO [62]数据集上训练的 Tiny-YOLOv3 [61] 。

表II报告了全精度准确率和量化后的 INT8 准确度，表明量化带来的准确度损失极小。预训练模型未经进一步微调便进行了量化。根据模型大小，采用不同大小的静态随机存取存储器可实现更高能效的实现。图17展示了这些神经网络基准测试的功耗、延迟和能耗对比。我们针对九种不同的 SRAM 宏单元大小和不同数量的存储体，对功耗、延迟和能耗进行了详细分析。

图 17(a) 比较了在不同 SRAM 宏单元大小下，各神经网络基准测试在所提出的 TDC-CiM 架构上运行时的总体功耗。对于给定的工作负载，无论 SRAM 大小如何，总功耗都基本保持不变。这种现象可以从并行度与周期数之间的平衡来解释：例如，16 KB 的 SRAM 宏单元存储容量是 8 KB 宏单元的两倍，因此每个时钟周期可并行执行的乘加运算数量也增加一倍。然而，对于固定模型大小的相同总乘加运算量，较小的 8 KB 配置需要两倍的周期才能完成。因此，总动态开关活动几乎保持不变，而静态功耗开销则有效地分摊到整个计算过程中，从而使不同 SRAM

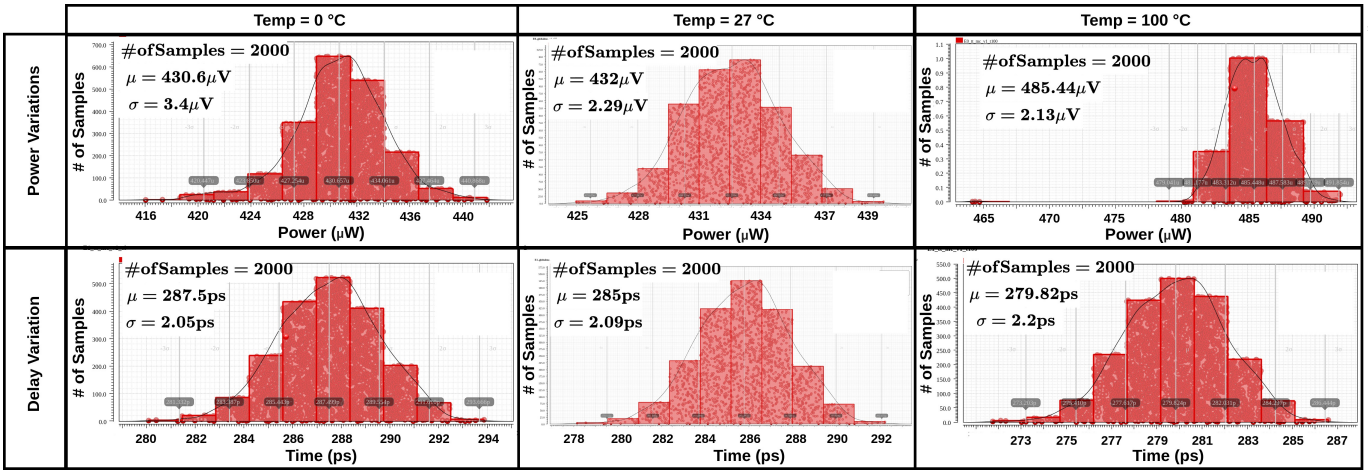

图16. 所提出TDC在温度变化下的蒙特卡洛分析。顶行显示 0 °C、 27 °C和 100 °C下的功耗变化，底行显示相应的延迟变化。每个工艺角均使用2000个样本，并采用 3σ 工艺变化，结果表明，较高温度会使功耗略有增加、延迟有所降低，同时在整个工作范围内保持稳定的性能。

宏尺寸。

图17(b)展示了所有神经网络基准测试在不同SRAM宏单元大小下的推理延迟。平均而言，将宏单元大小从8 KB增大到24 KB时，延迟降低了70%，凸显了更大的本地存储在实现每周期更多乘加运算并减少串行计算周期数量方面的优势。进一步将大小从16 KB增大到64 KB，可使平均延迟进一步降低约60%，体现出并行数据访问能力的提升以及对重复权重获取需求的减少。同样地，将宏单元从128 KB扩展到256 KB可使平均延迟提升约45%，这表明由于可用并行度相对于工作负载大小趋于饱和，在更大的宏单元大小下收益会逐渐递减。这一明确趋势证实，TDC-CiM架构中的更大SRAM宏能够有效利用周期内并行性和数据复用，最大限度地减少内存瓶颈，并加速各种网络规模下的推理性能。

图 17(c) 展示了所有基准模型在不同 SRAM 宏单元大小下每次推理的平均能耗。平均而言，将宏单元从 8 KB 扩大到 32 KB，可使总能耗降低 67%，这主要是因为减少了获取权重与激活值所需的连续内存访问周期数。将宏单元大小从 32 KB 进一步增至 96 KB，可使平均能耗额外降低 73.6%，凸显了更高数据复用率和更少片外事务的优势。将 SRAM 扩大到 256 KB（从 96 KB 起）可使能耗进一步降低 62%，表明即使 SRAM 内存大小较大，宏单元仍能带来显著的效率提升。这些结果证实，所提出的 TDC-CiM 架构能够利用大型 SRAM 阵列最大限度地执行存内运算，并减少计算单元与存储单元之间冗余的数据搬移，从而实现显著节能。

表II总结了基准测试所用的神经网络模型，包括数据集、参数数量、INT8量化前后的准确度，以及为实现最佳推理性能而选用的SRAM宏大小。

平均而言，对五种经过评估的卷积神经网络模型进行INT8量化后，准确度平均下降0.75%。为每种模型选择最佳SRAM大小，需要在存储容量与所需的并行计算水平之间取得平衡。LeNet-5 [53] 等较小的模型，由于参数总量较少，采用相对较小的24 KB SRAM块即可实现最高效率。相比之下，ResNet-18 [59] 和Tiny-YOLOv3 [61] 等较大的模型需要显著更多的片上存储，以充分利用高并行度的乘加运算并尽量减少重复的权重获取，因此采用较大的256 KB SRAM配置是合理的。尽管与其他ImageNet类别模型相比，SqueezeNet [58] 的参数数量相对较少，但其架构设计因包含压缩与扩展模块，导致每个参数对应的运算数量较多。因此，为维持较高的并行度并减少总周期数，它需要容量更大的128 KB SRAM，这一需求超出了其原始参数数量所暗示的水平。表II还表明，为每个网络选择合适的SRAM大小，既能将量化导致的准确率下降控制在较低水平，又能通过存内计算并行性显著改善能耗和延迟。

E. 与先前工作的乘累加比较

表 III 将所提出的 TDC-CiM 架构与现有能够执行乘加运算的存内计算架构进行了比较。所提出的架构使用 8T 位单元，以 0.5 GHz 时钟频率对 8 位输入和 8 位权重执行乘加运算。表 III 报告了我们通过对 $512 \times 256$8T TDC-CIM 宏单元进行晶体管级布局后 SPICE 仿真得到的数据，该宏单元包括字线和位线驱动器、局部控制电路以及 4 位 TDC。所有仿真均采用 28 纳米 CMOS 工艺，在 TT 工艺角、 25 °C 和 $VDD = 1.0$ V 条件下进行。根据乘加放电范围，TDC 输入范围为 200 至 800 mV。INT8 推理由两个 4 位乘累加周期实现，并采用移位加法累加。吞吐量根据仿真的周期数和时钟频率计算

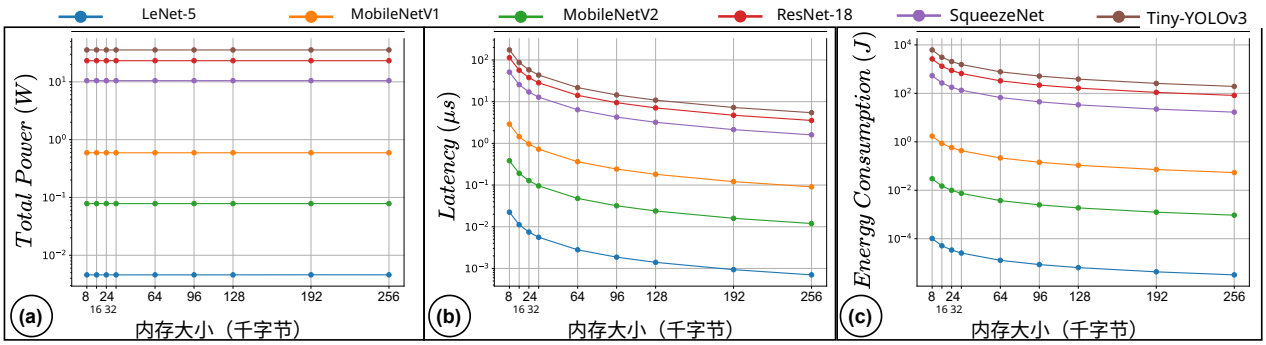

图 17. 对 SRAM 宏（容量范围为 8 千字节至 256 千字节）上的卷积神经网络模型进行评估。(a) 不同存储器容量下的总功耗基本保持不变，较深的模型功耗更高。(b) 随着 SRAM 容量增大，推理延迟降低；由于存储器访问次数减少，较深模型的延迟降幅更大。(c) 能耗随存储器容量增大而降低，这表明减少片外数据传输可提升能效。

表 II 神经网络模型的基准分析表明，无论采用全精度还是 INT8 数据类型，准确率下降都很小；同时还揭示了实现最佳能效所需的最优 SRAM 大小。

模型	数据集	参数	全精度准确度	INT8 准确度（软件）	INT8 准确度（TDC-CiM）	SRAM 大小
LeNet-5 [53]	MNIST [54]	61.47K	98.76%	98.7%	98.7%	24 千字节
MobileNetV1 [55]	CIFAR-10 [57]	3.19M	92.45%	89.73%	89.6%	64 千字节
MobileNetV2 [56]	CIFAR-10 [57]	2.35M	96.43%	96.48%	96.1%	64 千字节
SqueezeNet [58]	ImageNet-1K [60]	1.24M	58.18%	56.50%	55.80%	128 千字节
ResNet-18 [59]	ImageNet-1K [60]	11.68M	67.61%	67.14%	67.1%	256 千字节
Tiny-YOLOv3 [61]	COCO [62]	8.85M	33%（平均精度均值）	31%（平均精度均值）	31%（平均精度均值）	256 千字节

表 III 提出的 TDC存内计算架构与先前存内计算研究的比较表明，其能效比 [66] 高 59.7%，吞吐量比 [47]高 6.25× 。

	本文 JSSC'23 [66]	JSSC'24	[47] TCAS-I'24	[67] 国际固态电路会议'22	[68]
工艺（nm）	**28**22 28 28 28				
单元类型	**8T**	9T	8T	8T1C	6T
阵列大小	**16KB**	128千字节	16KB	64KB	1MB
精度（输入/权重）	**8/8**	8/8	8/8	8/8	8/8
供电电压（V）	**1**	1.1	0.9	0.8	0.9
频率（GHz）	**0.5**	0.24	0.2	0.25	-
吞吐量（每秒十亿次运算）	**320**	1000	51.2	256	1241
能效（每瓦每秒万亿次运算）	**38.46**	15.5	22.2	71.17	37.01
计算密度（TOPS/mm^2）	**2.1**	4	1.6	1.29	-

对于 16 千字节 SRAM 宏，采用配置的并行度。能效（每瓦每秒万亿次运算）为吞吐量除以宏功耗，其中功耗包括存内计算阵列、字线和位线驱动器、TDC 读出电路及局部控制电路的动态功耗和漏电功耗。

TDC-CiM的吞吐量达到 320 $GOPS$ ，能效为38.46 $TOPS/W$。在 [66], 中，采用具有差分输入的9T双极性单元执行异或运算，但连续电流通路会导致漏电功耗较高。相比之下，本研究采用的8T单元在空闲时会断开电容器，从而有效降低漏电功耗。因此，与 [66]相比，所提出的架构工作频率提高一倍，能效提升59.7%。在 [47], 中，基于数模转换器/模数转换器的存内计算操作会导致较高的能耗，在 [67]中，存内计算操作会导致较高的能耗，而用于存内计算的數據转换器占总面积的50%以上。相比之下，所提出的TDC存内计算架构无需模数转换器，显著降低了能耗开销和面积占用。因此，与 [47], 相比，TDC-CiM的吞吐量提升了 6.25× ，面积效率更高 1.31× ，能效提升了 42.2%；此外，与 [67]相比，其频率更高 2×、吞吐量高出20%，面积效率也更高 1.62×。

五、结论

本文提出了一种面向低功耗存内计算的 TDC-CiM 架构。该架构利用 8T SRAM 位单元执行按位乘法，并采用二进制加权电容阵列在存储器内部直接完成乘加运算。定制的时间数字转换器（TDC）将模拟乘积累加结果数字化，无需使用传统存内计算设计中常见的高能耗、高面积开销模数转换器。所提出的 TDC 以 1.25 mW 的功耗实现了 1 GS/s 的采样频率，信号噪声失真比为 19.45 dB，Walden 品质因数为 162.8 fJ/转换步。

为支持卷积神经网络工作负载的可扩展部署，提出了一种采用

全权重驻留映射方案的自动化静态随机存取存储器宏选择算法。对六种神经网络进行基准测试，静态随机存取存储器大小从 8 千字节扩展至 256 千字节，结果表明，增大静态随机存取存储器大小可显著降低能耗：从 8 千字节增至 32 千字节时，能耗最多降低 67%；从 32 千字节增至 96 千字节时，额外降低 73.6%；从 96 千字节增至 256 千字节时，进一步降低 62%。这些改进归因于内存访问周期减少、存内数据重用增加以及乘加并行性提升。

所提出的TDC存内计算架构实现了320 GOPS的吞吐量和38.46 TOPS/W的能效，证实了其在能耗受限的边缘人工智能应用中的潜力，适用于需要并行、低延迟乘加运算的场景。

参考文献

[1] G. Singh、A. Wagle、S. Khatri和S. Vrudhula，“Cidan-xe：利用人工神经元在DRAM中进行存内计算”，《电子学前沿》，第3卷，文章编号834146，2022年。[2] F. Zhang、S. Angizi、J. Sun、W. Zhang和D. Fan，“Aligner-d：利用DRAM存内计算加速DNA短读段比对”，*IEEE*新兴与精选电路和系统主题期刊，第13卷，第1期，第332–343页，2023年。[3] G. Singh和S. Vrudhula，“面向生成式人工智能的可扩展、高能效存内处理架构”，*IEEE*新兴与精选电路和系统主题期刊，第15卷，第2期，第285–298页，2025年。[4] I. Choi、E. J. Choi、D. Yi、Y. Jung、H. Seong、H. Jeon、S.-J. Kweon、I.-J. Chang、S. Ha和M. Je，“一种采用电流复用差分环控振荡器的基于SRAM的混合计算存内宏”，*IEEE*新兴与精选电路和系统主题期刊，第12卷，第2期，第536–546页，2022年。[5] X. Li、Y. Pan、Q. Jin、L. Chen、Y. Lou、B. Wu、J. Long、Y. Zhou、C. Peng、X. Wu和Z. Lin，“采用自循环10T SRAM单元的全阵列布尔逻辑存内计算宏，适用于AES系统”，*IEEE*《超大规模集成（*VLSI*）系统汇刊》，第33卷，第8期，第2214–2224页，2025年。[6] Z. Lin、Z. Zhu、H. Zhan、C. Peng、X. Wu、Y. Yao、J. Niu和J. Chen，“基于10T SRAM及水平和垂直解耦读端口的双向存内计算”，*IEEE*固态电路期刊，第56卷，第9期，第2832–2844页，2021年。[7] J. Wang、X. Wang、C. Eckert、A. Subramaniyan、R. Das、D. Blaauw和D. Sylvester，“14.2 一种支持位串行整数/浮点运算的计算型SRAM，用于可编程存内向量加速”，载于*IEEE*国际固态电路会议（*ISSCC*），2019年，第224–226页。[8] S. Kim、S. Kim、S. Um、S. Kim、K. Kim和H.-J. Yoo，“Neuro-CIM：具有运算门控/停止和数模网络的无ADC神经形态存内计算处理器”，*IEEE*固态电路期刊，第58卷，第10期，第2931–2945页，2023年。[9] S. K. Gonugondla、M. Kang和N. R. Shanbhag，“通过片上训练实现的抗变化存内机器学习分类器”，*IEEE*固态电路期刊，第53卷，第11期，第3163–3173页，2018年。[10] R. Islam、H. Fahmy、P.-Y. Lin和M. R. Guthaus，“差分电流模式时钟分配”，载于*2015*年*IEEE*第*58*届中西部电路与系统国际研讨会（*MWSCAS*），2015年，第1–4页。[11] B. Jyothi、B. R. Reddy和M. Jhamb，“用于超低功耗应用的全环绕碳纳米管场效应晶体管：采用失配级联传输晶体管绝热逻辑”，《集成》，第100卷，文章编号102260，2025年。[12] R. Islam，“负电容时钟分配”，*IEEE*《计算新兴主题汇刊》，第9卷，第1期，第547–553页，2021年。[13] M. Nurul Achmadiah、A. Ahamad、C.-C. Sun和W.-K. Kuo，“面向物联网系统边缘设备的高能效快速目标检测”，*IEEE*《物联网期刊》，第12卷，第11期，第16 681–16 694页，2025年。[14] M. Guthaus和R. Islam，“电流模式时钟分配”，2017年10月10日，美国专利9,787,293。[15] Y. Ji、Y. Li和I. Lee，“适用于高温应用的低功耗亚阈值电压基准”，*IEEE*《电路与系统汇刊*I*：常规论文》，第72卷，第6期，第2529–2542页，2025年。[16] R. Islam，“一种低功耗、高可靠性的抗单粒子瞬态诱发双节点翻转加固型锁存器和触发器”，《加拿大电气与计算机工程期刊》，第42卷，第2期，第93–101页，2019年。

[17] S. Wan、H. Wu、J. Cheng 和 X. Cao，《一种具有低温度系数、低功耗和高 PSRR 的亚阈值 CMOS 电压基准》，《模拟集成电路与信号处理》，第 124 卷，21，2025 1，第 p. .[18] R. Islam 和 M. R. Guthaus，《电流模式时钟分配》，载于 IEEE 国际电路与系统研讨会（ISCAS），2014 年，第 1203–1206 页。[19] U. Saxena、I. Chakraborty 和 K. Roy，《迈向无需 ADC 的存内计算加速器，实现高能效深度学习》，载于 2022 年欧洲设计、自动化与测试会议及展览会（DATE），2022 年，第 624–627 页。[20] Y. Zhao、F. Ye 和 J. Ren，《采用非二进制逐次逼近 TDC 的 500-MS/s、9 位时域 ADC》，载于 2022 年 IEEE 亚太电路与系统会议（APCCAS），2022 年，第 209–212 页。[21] K. Ohhata，《一种采用高线性度动态 VTC 的 2.3-mW、1-GHz、8 位全时域两步 ADC》，IEEE固态电路期刊，第 54 卷，第 7 期，第 2038–2048 页，2019 年。[22] D. Challagundla、I. Bezzam 和 R. Islam，《面向乘加运算的谐振时域存内计算（rTD-CiM）无 ADC 架构》，载于 2024 年大湖区 VLSI 研讨会论文集，丛书 GLSVLSI ’24。美国纽约：美国计算机协会，2024 年，第 268–271 页。 [在线]。可访问：https://doi.org/ 10 1145 3649476 3658773 。/。J. Lou、F. [23] Freye、C. Lanius 和 T. Gemmeke，《面向边缘 AI 设备、能效达 2.06 pops/w 的可扩展时域存内计算 BNN 引擎》，载于 2023 年大湖区 VLSI 研讨会论文集，丛书 GLSVLSI ’23。美国纽约：美国计算机协会，2023 年，第 665–670 页。 [在线]。可访问：https://doi.org/ 10 1145 3583781 3590220 。/。S. Angizi、A. Roohi、M. [24] Taheri 和 D. Fan，《使用剩余数系统加速基于乘加的应用的存内处理：一项比较研究》，载于 2021 年大湖区 VLSI 研讨会论文集，丛书 GLSVLSI ’21。美国纽约：美国计算机协会，2021 年，第 265–270 页。 [在线]。可访问：https://doi.org/ 10 1145 3453688 3461529 。/。 [25] H. Kim、T. Yoo、T. T.-H. Kim 和 B. Kim，《Colonnade：用于神经网络处理的可重构 SRAM 数字位串行存内计算宏》，IEEE固态电路期刊，第 56 卷，第 7 期，第 2221–2233 页，2021 年。[26] K. Xiao、X. Cui、X. Qiao、J. Song、H. Luo、X. Wang 和 Y. Wang，《一种采用分层容量衰减器和输入稀疏性优化 ADC、用于 4 位乘加运算的 28 nm、32 kb SRAM 存内计算宏》，IEEE电路与系统汇刊第二部分：简报，第 70 卷，第 6 期，第 1816–1820 页，2023 年。[27] D. Kushwaha、A. Sharma、N. Gupta、R. Raj、A. Joshi、J. Mishra、R. Kohli、S. Miryala、R. Joshi、S. Dasgupta 和 A. Bulusu，《一种采用电荷共享、支持 4 位乘累加运算的 65 nm 存内计算 7T SRAM 宏》，载于 2022 年 IEEE 国际电路与系统研讨会（ISCAS），2022 年，第 1556–1560 页。[28] F. Wei、X. Cui、S. Zhang 和 X. Zhang，《一种用于双向阵列内逻辑/内容寻址存储器运算的 11T静态随机存取存储器单元》，IEEE电路与系统汇刊第二部分：简报，第 1–1 页，2023 年。[29] S. Zhang、X. Cui、F. Wei 和 X. Cui，《一种基于 SRAM 阵列实现任意布尔函数的面积高效存内实现方法》，IEEE Transactions on Computers，第 72 卷，第 12 期，第 3416–3430 页，2023 年。[30] D. Challagundla、I. Bezzam、B. Saha 和 R. Islam，《用于布尔逻辑的谐振存内计算（RCiM）10T SRAM 宏》，载于 IEEE第41届国际计算机设计会议（ICCD），2023 年，第 110–117 页。[31] D. Challagundla、I. Bezzam 和 R. Islam，《14 纳米 FinFET 中串联谐振时钟的设计自动化》，《电路、系统与信号处理》，第 42 卷，第 12 期，第 7549–7579 页，2023 年。[32] R. Islam、D. Challagundla 和 I. Bezzam，《降低宽带串联谐振时钟偏斜的系统与方法》，2024 年 10 月 10 日，美国专利申请 18/627,479。[33] D. Challagundla，《利用 FinFET 宽带时钟网络中的谐振能量回收降低功耗与偏斜》，博士学位论文，2022 年。 [在线]。可访问：http: //proxy-bc.researchport.umd.edu/login?url= https://www.proquest.com/ dissertations-theses/power-skew-reduction-using-resonance-energy/ docview/2720940389/se-2[34] R. Islam，《采用软错误稳健能量回收触发器的低功耗谐振时钟》，Journal of Electronic Testing，第 34 卷，第 4 期，第 471–485 页，2018 年。

[35] J. Rosenfeld 和 E. G. Friedman，《全局谐振H树时钟分配网络的设计方法》，IEEE《超大规模集成（VLSI）系统汇刊》，第 15 卷，148，2007 2，135 no. pp. – .[36] R. Islam，《高速、能效优异的软错误容错触发器》，博士学位论文，Citeseer，2011 年。[37] Y. Mizukawa、J. Ranta、U. Ahmed、D. Bla˘zevi´c 和 P. Rasilo，《谐振电路对自由振动下磁致伸缩能量采集器最优设计的影响》，IEEE磁学汇刊，第 60 卷，第 8 期，第 1–14 页，2024 年。[38] P.-Y. Lin、H. A. Fahmy、R. Islam 和 M. R. Guthaus，《利用补偿电容最小化 LC 谐振时钟资源》，载于 IEEE 国际电路与系统研讨会（ISCAS），2015 年，第 1406–1409 页。[39] H. You、W. Li、D. Shang、Y. Zhou 和 S. Qiao，《一种用于神经网络处理的 1–8 位可重构数字 SRAM 存内计算宏》，IEEE《电路与系统汇刊I：常规论文》，第 1–13 页，2024 年。[40] C. Eckert、X. Wang、J. Wang、A. Subramaniyan、R. Iyer、D. Sylvester、D. Blaaauw 和 R. Das，《神经缓存：深度神经网络的缓存内位串行加速》，载于 2018 年 ACM/IEEE第45届年度国际计算机体系结构研讨会（ISCA），2018 年，第 383–396 页。[41] A. Biswas 和 A. P. Chandrakasan，《Conv-SRAM：一种用于低功耗卷积神经网络、支持存内点积计算的高能效 SRAM》，IEEE固态电路期刊，第 54 卷，第 1 期，第 217–230 页，2019 年。[42] J. Song、Y. Wang、M. Guo、X. Ji、K. Cheng、Y. Hu、X. Tang、R. Wang 和 R. Huang，《TD-SRAM：用于二值神经网络的基于时域的存内计算宏》，IEEE《电路与系统汇刊I：常规论文》，第 68 卷，第 8 期，第 3377–3387 页，2021 年。[43] S. Cheon、K. Lee 和 J. Park，《一种面向三值神经网络、能效达 2941 TOPS/W 的电荷域10T SRAM存内计算》，IEEE《电路与系统汇刊I：常规论文》，第 70 卷，第 5 期，第 2085–2097 页，2023 年。[44] J. Song、X. Tang、X. Qiao、Y. Wang、R. Wang 和 R. Huang，《一种 28 nm、16 Kb、位宽可扩展的电荷域转置 6T SRAM 存内计算宏》，IEEE《电路与系统汇刊I：常规论文》，第 70 卷，第 5 期，第 1835–1845 页，2023 年。[45] Y. Chen、G. Yin、M. Zhou、W. Tang、Z. Yang、M. Lee、X. Du、J. Yue、J. Liu、H. Yang、Y. Liu 和 X. Li，《SAMBA：采用非线性补偿全并行模拟加法器树的单ADC多位累加存内计算》，IEEE《电路与系统汇刊I：常规论文》，第 70 卷，第 7 期，第 2762–2773 页，2023 年。[46] Z. Chen、Z. Wen、W. Wan、A. Reddy Pakala、Y. Zou、W.-C. Wei、Z. Li、Y. Chen 和 K. Yang，《PICO-RAM：一种具有原位多位电荷计算和兼容6T薄单元的版图的 PVT不敏感模拟存内计算SRAM宏》，IEEE固态电路期刊，第 60 卷，第 1 期，第 308–320 页，2025 年。[47] K. Lee、J. Kim 和 J. Park，《一种采用基于位线电荷共享的SRAM内DAC/ADC操作的 28 nm、50.1-TOPS/W P-8T SRAM 存内计算宏设计》，IEEE固态电路期刊，第 59 卷，第 6 期，第 1926–1937 页，2024 年。[48] P. Chen、S.-I. Liu 和 J. Wu，《一种低功耗、高准确度 CMOS时间数字转换器》，载于 1997年IEEE国际电路与系统研讨会（ISCAS），第 1 卷，1997 年，第 281–284 页，卷 1。[49] E. Raisanen-Ruotsalainen、T. Rahkonen 和 J. Kostamovaara，《一种低功耗 CMOS时间数字转换器》，IEEE固态电路期刊，第 30 卷，第 9 期，第 984–990 页，1995 年。[50] R. Islam、B. Saha 和 I. Bezzam，《谐振能量回收SRAM架构》，IEEE电路与系统汇刊第二部分：简报，第 68 卷，第 4 期，第 1383–1387 页，2021 年。[51] D. Challagundla、I. Bezzam 和 R. Islam，《ArxRCIM：面向特定应用的谐振SRAM存内计算的体系结构探索》，IEEE《超大规模集成（VLSI）系统汇刊》，第 33 卷，第 1 期，第 179–192 页，2025 年。[52] R. Islam、B. Saha 和 I. Bezzam，《谐振能量回收SRAM架构》，IEEE电路与系统汇刊第二部分：简报，第 68 卷，第 4 期，第 1383–1387 页，2021 年。[53] Y. LeCun、L. Bottou、Y. Bengio 和 P. Haffner，《基于梯度的学习在文档识别中的应用》，载于 IEEE会议录，第 86 卷，第 11 期。IEEE，1998 年，第 2278–2324 页。[54] Y. LeCun、C. Cortes 和 C. J. Burges，《手写数字的 MNIST 数据库》，http://yann.lecun.com/exdb/mnist，1998 年。[55] A. G. Howard、M. Zhu、B. Chen、D. Kalenichenko、W. Wang、T. Weyand、M. Andreetto 和 H. Adam，《MobileNets：面向移动视觉应用的高效卷积神经网络》，载于 arXiv 预印本 arXiv:1704.04861

[56] M. Sandler、A. Howard、M. Zhu、A. Zhmoginov 和 L.-C. Chen，“Mobilenetv2：倒置残差与线性瓶颈”，载于*IEEE*计算机视觉与模式识别会议论文集，2018年，第4510–4520页。[57] A. Krizhevsky 和 G. Hinton，“从微型图像中学习多层特征”，多伦多大学，技术报告，2009年。[58] F. N. Iandola、S. Han、M. W. Moskewicz、K. Ashraf、W. J. Dally 和 K. Keutzer，“SqueezeNet：以少50倍的参数实现AlexNet级准确度，且模型大小仅为¡1*07360*，2016*1602*MB”，载于*arXiv*预印本 *arXiv:* 。。[59] K. He、X. Zhang、S. Ren 和 J. Sun，“用于图像识别的深度残差学习”，载于*IEEE*计算机视觉与模式识别会议论文集，2016年，第770–778页。[60] J. Deng、W. Dong、R. Socher、L.-J. Li、K. Li 和 L. Fei-Fei，“ImageNet：大型分层图像数据库”，载于*IEEE*计算机视觉与模式识别会议论文集，2009年，第248–255页。[61] J. Redmon 和 A. Farhadi，“YOLOv3：渐进式改进”，https://pjreddie.com/darknet/yolo/，2018年，arXiv预印本 arXiv:1804.02767。[62] T.-Y. Lin、M. Maire、S. Belongie、J. Hays、P. Perona、D. Ramanan、P. Doll´ar 和 C. L. Zitnick，“Microsoft COCO：情境中的常见物体”，载于欧洲计算机视觉会议。施普林格，2014年，第740–755页。[63] L. M. Santana、E. Martens、J. Lagos、P. Wambacq 和 J. Craninckx，“一种在28纳米CMOS中采用数字交叉耦合的70 MHz带宽时分交织噪声整形逐次逼近寄存器辅助ΔΣ模数转换器”，载于*ESSCIRC 2023-IEEE*第*49*届欧洲固态电路会议（*ESSCIRC*），2023年，第389–392页。[64] X. S. Wang、C.-H. Chan、J. Du、C.-H. Wong、Y. Li、Y. Du、Y.-C. Kuan、B. Hu 和 M.-C. F. Chang，“一种在28纳米CMOS中采用互补双环路辅助缓冲器、SFDR达50 dB的8.8 GS/s 8位时分交织逐次逼近模数转换器”，载于*2018*年*IEEE*射频集成电路研讨会（*RFIC*），2018年，第88–91页。[65] R. Islam、H. A. Fahmy、P. Y. Lin 和 M. R. Guthaus，“DCMCS：高鲁棒性低功耗差分电流模式时钟与综合”，*IEEE*《超大规模集成（*VLSI*）系统汇刊，第26卷，第10期，第2108–2117页，2018年。[66] H. Wang、R. Liu、R. Dorrance、D. Dasalukunte、D. Lake 和 B. Carlton，“一种面向边缘推理、基于C-2C梯形网络的8位乘累加单元电荷域SRAM存内计算宏，采用22纳米FinFET工艺”，*IEEE*固态电路期刊，第58卷，第4期，第1037–1050页，2023年。[67] X. Qiao、Q. Guo、X. Tang、J. Song、R. Wei、M. Li、R. Wang 和 Y. Wang，“一种用于有符号操作数计算和批归一化实现的16.38 TOPS、4.55 POPS/W SRAM存内计算宏”，*IEEE*《电路与系统汇刊*I*：常规论文》，第71卷，第4期，第1706–1718页，2024年。[68] P.-C. Wu、J.-W. Su、Y.-L. Chung、L.-Y. Hong、J.-S. Ren、F.-C. Chang、Y. Wu、H.-Y. Chen、C.-H. Lin、H.-M. Hsiao、S.-H. Li、S.-S. Sheu、S.-C. Chang、W.-C. Lo、C.-C. Lo、R.-S. Liu、C.-C. Hsieh、K.-T. Tang、C.-I. Wu 和 M.-F. Chang，“一种28纳米、1 Mb时域存内计算6T SRAM宏，延迟为6.6 ns，吞吐率为1241 GOPS、能效为37.01 TOPS/W，用于边缘AI设备中的8位乘加运算”，载于*2022*年*IEEE*国际固态电路会议（*ISSCC*），第65卷，2022年，第1–3页。



Dhandeep Challagundla （IEEE学生会员）获得了美国马里兰州马里兰大学巴尔的摩县分校（UMBC）的硕士学位，目前正在该校计算机科学与电气工程系攻读博士学位。他的研究兴趣包括能效计算、存内计算、静态随机存取存储器设计、低功耗电路设计、混合信号集成电路设计和电子设计自动化工具。



Ignatius Bezzam教授 于2015年获得圣克拉拉大学电气工程博士学位，并于1983年获得印度理工学院马德拉斯分校的工学学士学位。Bezzam博士在模拟混合信号集成电路设计领域拥有多项重要专利，并在国际顶级会议上发表论文，包括国际固态电路会议、欧洲固态电路会议和电路与系统汇刊。Bezzam博士与全球团队合作，成功完成了30次首硅流片，并在硅谷、欧洲和亚洲拥有33年的新一代芯片设计经验。



Riadul Islam 现任马里兰大学巴尔的摩县分校计算机科学与电气工程系助理教授。在加州大学圣克鲁兹分校攻读博士学位期间，Riadul 设计了首个电流脉冲触发器/寄存器，由此实现了首个面向高性能微处理器的一对多电流模式时钟分配网络。2017年至2019年，他在美国密歇根大学迪尔伯恩分校担任助理教授。他是 IEEE 高级会员，也是计算机协会、IEEE电路与系统学会、IEEE-CAS 超大规模集成电路系统与应用技术委员会（VSA-TC）以及 IEEE固态电路学会的成员。他持有两项美国专利，并发表了多篇 IEEE/ACM/MDPI/Springer Nature 期刊论文和会议论文。他目前的研究兴趣包括面向各种应用的数字、模拟和混合信号 CMOS 集成电路/片上系统；模拟、数字和混合信号集成电路的验证与测试技术；硬件安全；控制器局域网；用于微处理器和现场可编程门阵列设计与分析的计算机辅助设计工具；汽车电子；以及生物芯片。他是施普林格《电路、系统与信号处理》期刊的副编辑。他曾担任以下会议的技术程序委员会（TPC）成员：IEEE/ACM国际计算机辅助设计会议（ICCAD 2022）、ACM五大湖超大规模集成电路研讨会（GLSVLSI 2020、GLSVLSI 2021、GLSVLSI 2022）、2020年第57届IEEE/ACM设计自动化会议（DAC）晚期突破研究会议、2021年IEEE计算机学会超大规模集成电路年度研讨会（ISVLSI），以及2021年IEEE国际消费电子会议（ICCE）。Riadul 获得了2021年美国国家科学基金会ERI奖、2021年马里兰州产业合作伙伴计划（MIPS）奖和2021年马里兰州创新倡议（MII）奖。