arXiv:2611.28605v1 [cs.ET] 21 Nov 2026

APUF-SNG：一种低成本、输出概率可调的仲裁器物理不可克隆函数随机数生成器

Shelby Williams、Mehran Moghadam、M. Hassan Najafi、Sercan Aygun、Arman Roohi 和 Gourav Datta

摘要—随机计算（SC）将数值编码为概率比特流，并利用简单逻辑门实现算术运算，为低成本、容错和高度并行计算提供了一种有前景的范式。然而，随机计算系统的精度和效率取决于随机数生成器（SNG）的质量。传统随机数生成器依赖伪随机或拟随机生成器，其确定性会带来周期性、相关性和偏差，从而限制了可实现的精度和稳健性。本文提出一种低成本、基于仲裁器物理不可克隆函数（APUF）的随机数生成器，在近亚稳态延迟条件下运行，以生成输出概率可控的真随机比特流。通过选择使路径延迟差趋近目标值的输入挑战，所得输出概率由器件固有噪声决定，无需传统随机数生成器和基于比较器的随机数生成器。所提出的设计继承了APUF结构硬件开销低、统计独立性强的特点。本文还建立了概率模型并提出挑战选择方法，用于表征和控制随机行为。一个8级 APUF-SNG在1024比特流下达到与最先进真随机随机数生成器相当的1%平均绝对误差，其占用面积为116μm^2 in 45nm，分别比基于真随机数生成器和基于Sobol序列的生成器设计小2.5× 和 7× 。

索引词—物理不可克隆函数、真随机性、随机计算、比特流生成、边缘人工智能。

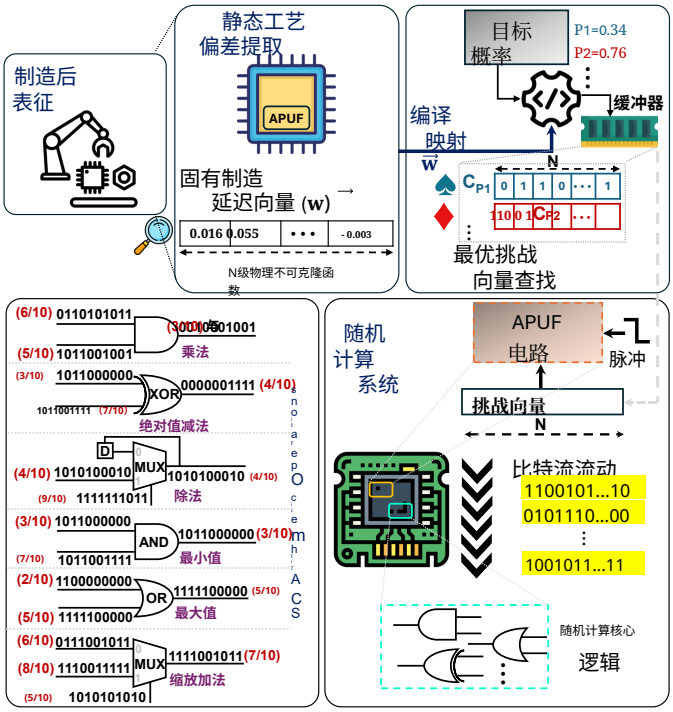

图1. 所提出的APUF-SNG设计概览。

I. 引言

随TOCHASTIC机计算（SC）是一种重新兴起的计算范式，它将数据编码为随机比特流，其中数值由观测到逻辑“1”的概率表示 [1], [2]。与依赖位置计数表示、需要复杂多比特算术单元的传统二进制计算不同，SC 使用极其简单的逻辑门进行计算：两个数值相乘可简化为按位与运算，缩放加法则通过多路复用器（多路复用器）实现，其他复杂函数只需使用相对较少的硬件 [3]。这些特性使随机计算特别适用于资源受限和容错应用，包括图像以及视频处理 [4], [5],机器学习推理 [6],信号处理 [7], 和神经网络加速 [8]。

随机计算系统中的一个关键瓶颈是随机数生成器（SNG），它负责将二进制编码的数据转换为随机比特流。SNG 通常由两个组件组成：随机数生成器（RNG）和二进制比较器。在每个时钟周期，比较器通过判断目标输入值是否大于 RNG 提供的当前随机数，生成一个输出比特。生成的比特流的整体质量、其中“1”和“0”的分布均匀性，以及各生成比特流之间的相关程度，直接决定了计算的精度、延迟和能效。性能较差的 SNG 会延长比特流收敛所需的长度，从而增加延迟和能耗。

大多数随机计算（SC）设计使用伪随机数生成器（PRNG）生成随机数，通常采用线性反馈移位寄存器（LFSR）实现 [1]。LFSR体积小巧，易于在硬件中实现，但其确定性和周期性会引入统计偏差和流间相关性，从而降低计算精度 [2]。最先进（SOTA）的研究用基于低差异序列的准随机数生成器（QRNG）取代了LFSR，例如——

本研究得到美国国家科学基金会（NSF）项目 2339701、2609436、2616837 以及美国国家航空航天局项目 80NSSC25C0335、Vernon & Ruby Langlinais 非捐赠研究基金、洛克希德·马丁公司捐赠教授基金以及来自 NVIDIA 和 Google 的捐赠。Mehran Moghadam 和 M. Hassan Najafi 任职于凯斯西储大学电气、计算机与系统工程系，位于克利夫兰，俄亥俄州，美国。电子邮箱：{moghadam, najafi}@case.edu。Sercan Aygun 任职于计算与信息学学院，路易斯安那大学拉法叶分校，路易斯安那州拉法叶，美国。电子邮箱：{c00591145，sercan.aygun}@louisiana.edu。Arman Roohi 任职于伊利诺伊大学芝加哥分校电气与计算机工程系，美国伊利诺伊州芝加哥。电子邮箱：aroohi@uic.edu。Gourav Datta 任职于电气、计算机与系统工程系，凯斯西储大学，美国俄亥俄州克利夫兰。电子邮箱：gourav.datta@case.edu。

差异（LD）序列，例如 Sobol [9],Halton [10],和 van der Corput [5]序列。低差异序列在样本空间中的数值分布更加均匀，使随机计算比特流能够更快地收敛到目标概率，缩短所需的位流长度，从而提高能效 [9]。基于PRNG和QRNG的SNG均为确定性的：其序列可根据种子预测，因此会引入偏差和周期相关性，对可达到的精度形成硬性上限。此外，这些生成器较高的硬件成本可能抵消采用随机计算所追求的硬件简洁性。

真随机数生成器（TRNG）为随机计算比特流生成提供了不同的基础。TRNG不依赖确定性算法，而是从热噪声、电子抖动和量子现象等本质上不可预测的物理过程中获取随机性。这种内在的非确定性消除了周期相关性，最大限度地减少了流间偏差，并提供了高熵；这些都是随机计算非常理想的特性。近期探索将TRNG用于随机计算的研究，大多包括基于电阻式随机存取存储器（RRAM）器件的设计 [11] 以及基于环形振荡器的真随机比特流生成 [12]。这些工作表明，基于TRNG的随机计算比特流生成器能够实现与基于线性反馈移位寄存器和Sobol的设计相当的准确性。然而，基于TRNG的随机数生成器设计仍未得到充分探索，同时提供真随机性、低硬件开销和可控输出概率的实现方案，仍是一个尚未解决且十分重要的研究问题。

物理不可克隆函数（PUF）是一类硬件安全原语，利用集成电路中固有且不可控的制造工艺变化来生成设备唯一的响应 [13]。在众多PUF架构中，仲裁器*PUF*（*APUF*）由于具有紧凑、规整的结构 [14]，尤其适合硅实现。在标准APUF中，两条电气等效的信号路径依次通过由激励控制的级联多路复用器级，并由输出端的锁存器确定哪条路径先到达。由于制造引起的延迟变化具有随机性，且实际上不可能被预测或复制，APUF的挑战-响应行为形成了设备特定指纹。传统APUF设计会避开两条路径延迟几乎相等的工作条件，因为这种近均衡条件会引入亚稳态和不可预测的输出行为。然而，已有研究表明，这些亚稳态挑战（其路径延迟差趋近于零）可以作为真随机性的来源：每次重新评估固定的近亚稳态挑战，都会产生一个独立且近似无偏的随机比特 [15]。这些设计将这种不稳定性视为一种安全特性：DTA-PUF从流水线内核的时序错误中获取响应 [16], ，而双模式电路则可在PUF和TRNG运算之间切换 [17], [18]。但这些设计都无法控制输出概率。相反，我们对近均衡区域进行校准，使存储的挑战能够产生任意目标输出概率，而这正是随机数生成器所需要的。

据我们所知，本研究提出了首个面向随机计算系统的基于 APUF 的真随机随机比特流生成器。图1概述了所提出的 **APUF-SNG** 设计。我们将 APUF 重新诠释为一种时间到概率转换原语：通过选择激励，使累积路径延迟差 Δ 达到特定目标值，即可控制仲裁器在任意一次评估中输出“1”的概率。具体而言，在热噪声、抖动和器件失配所产生的加性高斯噪声作用下，输出概率为$P(S=1) = \Phi(\frac{\Delta}{\sigma})$，其中 $\Phi(\cdot)$ 是标准正态累积分布函数（CDF），σ 表征噪声幅度。通过将每个输入值映射到能产生相应输出概率的激励，我们直接利用 PUF 的物理时序动态生成随机比特流，无需随机数生成器或比较器。所提架构继承了 APUF 的若干优良特性：1）由不同激励生成的任意两条比特流在统计上互不相关，满足精确执行随机计算乘法及许多其他随机计算操作的一项关键要求；2）随机性源自物理工艺变异和热噪声，而非确定性算法；3）核心生成器的硬件占用面积极小，仅由 PUF 延迟链和感测锁存器构成。

我们提出了一种解析延迟与噪声模型，以及一种挑战选择方法。通过评估对生成的比特流的统计质量进行了表征。所提出的生成器与最先进的随机比特流生成技术进行了比较，包括基于 LFSR 的伪随机数生成器、基于 Sobol 的量子随机数生成器，以及 TRUE-BSG 等近期的真随机方法 [12] 和 IMSNG [11]。本文的主要贡献总结如下：

➢**基于 APUF 的随机比特流生成：** 我们介绍了 **APUF-SNG**，一种基于 APUF 的真随机随机比特流生成器，可将器件级延迟变化直接转换为概率输出。我们建立了一种形式化概率模型，通过噪声驱动的决策过程将路径延迟差映射为输出概率，从而实现连续且细粒度的概率生成。

➢**通过激励选择控制概率：** 我们提出了一种挑战选择方法，用于识别对应于任意目标概率的近亚稳态工作点。该方法能够实现精确的概率调节，并消除传统随机计算系统中所采用的基于比较器的编码和外部随机数生成器的需求。

➢**硬件成本与精度评估：** 我们从精度、面积、功耗和能耗等方面，对所提出的生成器与基于最先进伪随机数生成器、量子随机数生成器和真随机数生成器的方法进行了全面评估，结果表明，该生成器在硬件成本显著降低的情况下，仍具备具有竞争力的效率。

II. 预备知识：仲裁器物理不可克隆函数

标准的仲裁器物理不可克隆函数会将两路信号跃迁输入由挑战控制的 多路复用器级联电路中，如图 2（a）所示。每个挑战位决定两路信号是沿同一路径传播，还是交叉至相反路径。在输出端，仲裁器（通常为锁存器）判定哪一路信号先到达。制造引起的工艺变化会使每条路径的延迟产生细微偏差，而累积

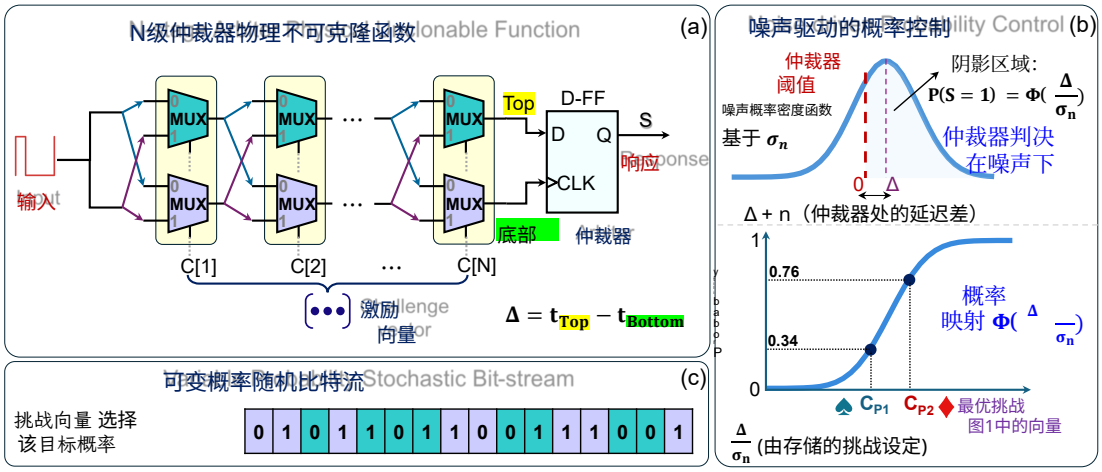

图 2。 （a）一个 N 级仲裁器物理不可克隆函数结构，（b）噪声影响下的仲裁器判决及其概率映射，以及（c）由此产生的随机计算比特流 .

延迟差的符号会产生依赖于激励的响应，从而形成独特的设备指纹。

在传统的（确定性）物理不可克隆函数运算中，通常会避开导致延迟差较小的激励，因为这会导致响应稳定性降低。相反，随机型仲裁器物理不可克隆函数有意针对这类近似平衡的延迟条件。在热噪声和抖动的影响下，对同一激励 进行重复评估会产生概率性输出，可将其建模为概率为 $p = P(S = 1)$ 的伯努利过程。这样，同一物理竞争机制便可改作随机数生成器使用。这一观点为所提架构提供了设计动机。我们不再将较小的延迟差视为不可靠性的来源，而是将其作为一种设计特性加以利用，以较低开销实现基于物理不可克隆函数的随机比特流生成。

形式化概率模型。设仲裁器输入端的累积路径延迟差为 $\Delta = T_{\text{Top}} - T_{\text{Bottom}}$。如果仲裁器判决受到零均值随机扰动 n 的影响，则有效判决变量变为$z = f_{latch}(\Delta + n)$。响应比特（见图 2(a)）为 $S = 1$，当 $z > 0$ 时；否则为 $S = 0$。因此， $P(S=1)=P(n>-\Delta)$。如果 n 服从方差为 σ_n^2 的高斯分布，则 $P(S=1) = \Phi(\frac{\Delta}{\sigma_n})$，其中 $\Phi(\cdot)$是标准正态累积分布函数。这个简单模型解释了关键的工作模式：❶ 如果 $|\Delta| \gg \sigma_n$，输出是可重复的（即确定性的）。❷ 如果 $|\Delta| \approx 0$，输出接近随机边界，且 $P(S = 1)$ 趋近于 0.5。❸ 如果 Δ 为适度的正值或负值，则该激励映射为一个概率介于 0 和 1 之间、可调的随机源。这个基于高斯噪声分布累积分布函数的概率模型，直接将依赖于激励的延迟失配与可观测的输出概率联系起来：产生延迟差 Δ 的激励将生成输出概率为 $P(S=1) = \Phi(\frac{\Delta}{\sigma_n})$ 的比特流。在实际中，激励搜索问题可简化为寻找一个激励 C，使其静态延迟差 Δ 符合目标值 $\Phi^{-1}(p_{\text{target}}) \cdot \sigma_n$，从而使针对 $|\Delta|$的优化成为所提架构的核心设计任务。

III. 所提架构

图 1概述了所提出的**APUF-SNG**设计流程，涵盖制造后表征、挑战选择，直至用于随机计算操作的随机比特流生成。图2 通过展示其底层物理机制对此作了补充。具体而言，图2（a）展示了一个作为真随机比特流发生器运行的 N级仲裁器物理不可克隆函数，其中仲裁器判决由累积路径延迟差 Δ决定。图2（b）展示了该判决如何变为概率性的：每次评估都会抽取一个新的噪声样本 n，其分布以 Δ为中心，因此判决结果为“1”的评估比例等于噪声密度曲线高于仲裁器阈值的面积， $P(S=1) = \Phi(\frac{\Delta}{\sigma_n})$。旁边的映射图表明，选择一个挑战即可将 $\frac{\Delta}{\sigma_n}$ 置于该曲线上选定的位置；因此，图 C_{P1}和 C_{P2}中的存储向量1 会产生目标 $p = 0.34$ 和 $p = 0.76$，而不只是 $p \approx 0.5$ 在 $\Delta \approx 0$时得到的无偏值。图2（c）展示了生成的随机计算比特流，其“1”的概率由所施加的挑战向量设定。

实际电路实现可描述如下。仲裁器物理不可克隆函数的每一级由一对可交叉延迟路径组成，并由一个挑战位控制。这些路径构成多路复用元件级联，根据施加的挑战向量有条件地交换或保持信号传播路径。信号在各级中传播时，工艺变化以及有意引入的失配或寄生失配会使两条路径之间产生累积延迟差。在输出端，时序比较器（仲裁器）判定哪个信号先到达，从而生成二进制响应。整体结构可表示为：（i）发射信号或时钟控制的转换源；（ii）由挑战控制的交叉级级联；（iii）各级相关的延迟失配；（iv）用于信号比较的时序仲裁器；以及（v）用于同步采样的可选输出寄存器。

当两条路径之间的延迟差较大时，仲裁器会产生确定性输出。然而，当延迟差趋近于零时，判决会对热噪声和局部器件变化等本征噪声源变得敏感。在这一近亚稳态区域，输出呈现概率行为，从而能够直接生成随机比特流。这一特性构成了所提架构的基础。

IV. 解析建模与仿真

A. 仲裁器PUF建模

对于一个 N 级仲裁器物理不可克隆函数（APUF），两条对称多路复用器链 [19], [20], [21], [22] 之间会发生信号竞争。施加到物理不可克隆函数（PUF）的挑战向量记为 $C \in \{0,1\}^N$。制造工艺变化由延迟向量$\vec{w} \in \mathbb{R}^{N+1}$表示：每一级包含一个失配项，另有一个末项表示最后一级与仲裁器之间的非对称性 [21],[23]。每个元素建模为 $w_i \sim \mathcal{N}(0, \sigma^2_{static})$。为确定输出仲裁器处的累积延迟差，我们定义奇偶校验向量 $\vec{\Psi} \in \{-1,1\}^{N+1}$，其中每个元素表示信号路径的交叉状态。二进制挑战位与奇偶校验向量之间的关系，对于 $i = 1$、.....、 N 和 $\Psi_{N+1} = 1$，可表示为 $\Psi_i = \prod_{j=i}^{N}(1 - 2C_j)$。顶部路径与底部路径之间的总静态延迟差 ΔD_{static}，是延迟向量与奇偶校验向量的线性组合。对于给定的芯片和挑战， $\Delta D_{static}(C) = \vec{w} \cdot \vec{\Psi}(C) = \sum_{i=1}^{N+1} w_i \Psi_i$。由于延迟分量 w_i 相互独立且服从正态分布，PUF 状态空间中的总静态延迟服从 $\Delta D_{static} \sim \mathcal{N}(0, (N+1)\sigma^2_{static})$ 分布（此处我们将 Ψ 视为奇偶校验向量，而 Φ 保留用于表示标准正态累积分布函数）。

B. 通过热噪声控制概率

要将物理不可克隆函数从确定性运算转换为真正的随机比特流生成，我们必须利用动态评估噪声。在运行时，热噪声和电源抖动会导致信号传播时间产生瞬时变化。我们将这种动态噪声建模为一个加性

算法 1基于APUF的随机比特流生成

输入： 目标概率 P_{target}、级数 N、位流长度 L、噪声参数 σ_{static}, $\sigma_{dynamic}$
输出： 最优激励 $\vec{C}_{opt}$，长度为 L 的随机比特流 $\mathcal{B}$
步骤
$\vec{w} \leftarrow \text{randn}(N+1,\ 1) \cdot \sigma_{static}$ ▷ 初始化芯片静态变化。仅用于仿真；根据硅上的挑战-响应对拟合
$\Delta D_{target} \leftarrow \text{norminv}(P_{target},\ 0,\ \sigma_{dynamic})$ ▷ 目标延迟有偏情况
步骤
$C_{opt} \leftarrow \varnothing$
$min_err \leftarrow \infty$, $\vec{C} \in \{0,1\}^N$: **对所有**
执行
$\Psi[N+1] \leftarrow 1$ ▷ Ψ：奇偶校验向量
对于 $j = N$ **递减至** 1，**执行**
$\Psi[j] \leftarrow (1 - 2 \cdot C[j]) \cdot \Psi[j+1]$
结束循环
$\Delta D_{static} \leftarrow \vec{w} \cdot \vec{\Psi}$ ▷ 静态延迟差
$current_err \leftarrow |\Delta D_{static} - \Delta D_{target}|$
如果 $current\ err < min\ err$ **则**
$min_err \leftarrow current\ err$, $best_delay \leftarrow \Delta D_{static}$, $\vec{C}_{opt} \leftarrow \vec{C}$
结束 if
结束 for
步骤
对于 $k = 1$ 到 L **执行**
$N_{th} \leftarrow \text{randn}() \cdot \sigma_{dynamic}$ ▷ 采样动态热噪声
$\Delta D_{total} \leftarrow best_delay + N_{th}$
如果 $\Delta D_{total} > 0$ **则**
$\mathcal{B}[k] \leftarrow 1$
else
$\mathcal{B}[k] \leftarrow 0$
结束如果
结束循环
返回 $\mathcal{B}$, $\vec{C}_{opt}$

从正态分布中抽取的随机变量 $N_{dynamic}$，其分布为 $N_{dynamic} \sim \mathcal{N}(0,\ \sigma^2_{dynamic})$。因此，单次评估中仲裁器处的瞬时延迟差为：$\Delta D_{total} = \Delta D_{static}(C) + N_{dynamic}$. 仲裁器解析竞争条件，生成二进制响应输出 $S \in \{0,1\}$。假设当顶部路径首先到达时仲裁器触发“1”（$\Delta D_{total} > 0$），则生成“1”的概率（记为 p）就是热噪声将总延迟推高至零以上的概率： $p = P(S=1) = P(\Delta D_{static}(C) + N_{dynamic} > 0)$。鉴于正态分布的对称性，我们可以利用标准正态分布的逆累积分布函数 Φ^{-1}，将任意目标概率 p_{target} 映射为特定所需的静态延迟，如下所示： $\Delta D_{target} = \sigma_{dynamic} \cdot \Phi^{-1}(p_{target})$。

C. 仿真方法

挑战空间是有限的。评估一个激励的开销为 $\mathcal{O}(N)$，因此，对于 $N \leq 16$，我们枚举全部 2^N 个激励（对于 $N = 16$ 为 65,536 个，约需 1.1×10^6 次乘加运算），并保留静态延迟最接近 ΔD_{target}的那个：$\vec{C}_{opt} = \text{argmin}_C |\Delta D_{static}(C) - \Delta D_{target}|$ （参见算法 1，C_{opt} 是为该概率级别存储的 N 比特向量）。枚举结果是精确的，因此决定精度的是覆盖范围，而非搜索。 2^N 延迟遵循 $\mathcal{N}(0, (N+1)\sigma^2_{static})$，在 $p \in [0.05, 0.95]$ 上，可实现概率之间的最大间隔对于 $N = 8$ 为 3%，对于 $N = 16$为 0.03%。后者远低于 1024 比特流的 1% 采样下限，这就是精度在 16 级时达到饱和的原因。

比特流生成： 选定的激励 $\vec{C}_{opt}$ 经过 L 次评估（其中 L 为比特流长度），每个周期都会注入独立的动态噪声 $N_{dynamic}$，从而生成最终比特流。为了评估生成的比特流的质量，我们进行 10,000 次蒙特卡洛仿真，将 $[0,1]$区间内的随机数转换为对应的比特流，并测量平均绝对误差（MAE）。我们使用随机交叉相关（SCC）指标 [24]评估生成的比特流对之间的相关性：

$$SCC = \begin{cases} \frac{ad-bc}{L \times min(a+b,a+c) - (a+b) \times (a+c)} & , if\ ad > bc \\ \frac{ad-bc}{(a+b) \times (a+c) - L \times max(a-d,0)} & , else \end{cases} \quad (1)$$

其中， a、 b、 c 和 d 分别表示从两个比特流对应比特位置取出的逻辑对 11、10、01 和 00。相关性由所有比特位置上重叠的逻辑 1 的数量决定。当重叠数最大时，比特流呈现最大相关性（$SCC = +1$）。当重叠数最小时，比特流呈现最小相关性（$SCC = -1$）。一个特别理想的情况是零相关性（$SCC = 0$），这是乘法等随机计算操作所需的，例如使用按位与运算进行乘法。

低相关性是**APUF-SNG**的结构性特征。每个比特在 $\mathbf{1}\{\Delta D_{static}(C) + n_k > 0\}$ 每次评估时都会使用新的噪声样本 $n_k \sim \mathcal{N}(0,\ \sigma^2_{dynamic})$，因此激励只决定均值；无论两个激励共享哪些路径，两条流的协方差都为零。图3通过随机配对证实了这一点，其高斯拟合以零为中心，离散程度为 $\frac{1}{\sqrt{L}}$。

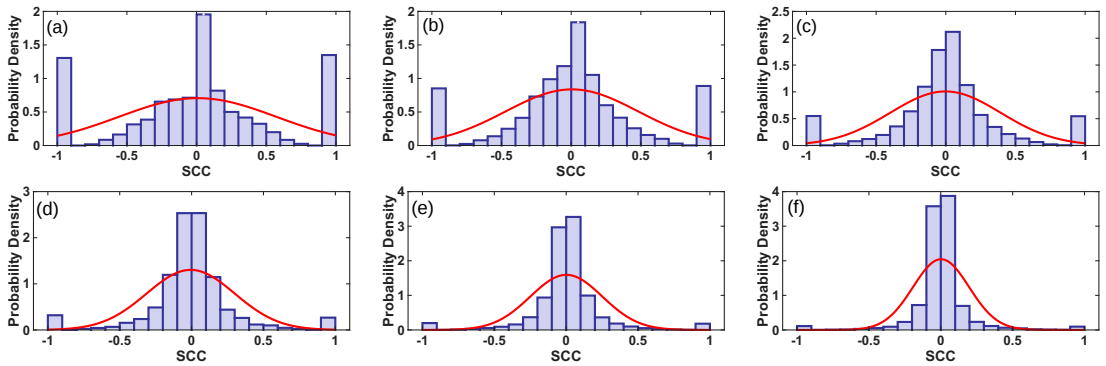

图3. 生成的随机计算比特流两两之间的 SCC 分布 使用 **APUF−SNG**16级生成的随机计算比特流，位流长度分别为 (a) 32 比特、(b) 64 比特、(c) 128 比特 ，(d) 256 比特、(e) 512 比特和 (f) 1024 比特s.

更具挑战性的情况表现相同：映射到相同目标概率的配对、共享所有路径的相同激励（在 L=1024 时标准差为 0.031，与 $\frac{1}{\sqrt{L}}$ 相符），以及从同一个 PUF 实例同时生成的八条流（正如随机计算数据通路所采用的方式），其 28 个成对随机计算相关系数仍以零为中心。

V. 评估结果与讨论

本评估基于加性延迟模型，该模型是标准的仲裁器物理不可克隆函数模型，并已在制造出的器件上得到验证 [21], [23], [20], ；此外还基于近亚稳态随机机制，该机制已由先前的物理不可克隆函数真随机数生成器设计在硅上进行测量 [17], [15], [18]。在硬件上，激励表依据各个激励测得的输出概率构建，而非依据模型，因此下文报告的精度并不取决于模型是否完全准确（第V-A节）。硅片测试揭示了电压、温度和老化漂移的幅度，第V-D节对此进行了界定。在所有实验中，$\sigma_{static} = 0.03$ 和 $\sigma_{dynamic} = 0.07$。对于 $N \leq 16$的设计，采用穷举枚举；对于 32 至 128 级的设计，则使用 10,000 个随机激励。动态噪声参数σ_{dynamic} 必须谨慎选择：取值过大会使 $\Phi(\frac{\Delta}{\sigma_{\text{dynamic}}})$ 曲线变平，降低概率分辨率，并损害比特流质量和算术精度；取值过小则会使激励搜索对 Δ中的量化误差过于敏感。所选的$\sigma_{\text{dynamic}} = 0.07$ 值在这两种相互制约的影响之间取得了平衡。

表 I 随机计算比特流生成方法的平均绝对误差（%）比较

方法	位流长度 (L)						
	16	32	64	128	256	512	1024
APUF−SNG–4 级	10.74	6.39	4.59	4.24	3.17	9.45	5.24
APUF−SNG–5 级	8.25	6.15	4.57	3.96	3.00	3.70	2.27
APUF−SNG–6 级	8.41	6.27	4.41	2.90	2.48	2.60	1.56
APUF−SNG–7 级	7.97	5.78	3.91	2.77	2.03	2.67	1.93
APUF−SNG–8 级	7.84	6.16	3.91	2.80	1.96	1.59	1.04
APUF−SNG–16级	7.90	5.48	3.95	2.76	1.95	1.42	1.01
APUF−SNG–32级	7.84	5.57	3.90	2.75	1.98	1.41	1.02
APUF−SNG–64级	7.86	5.63	3.94	2.76	1.96	1.39	1.01
APUF−SNG–128级	7.82	5.50	3.95	2.74	1.97	1.42	1.01
MATLAB（rand）	7.84	5.54	3.91	2.76	1.96	1.38	0.97
伪随机数生成器（$\log_2 L$-比特线性反馈移位寄存器）*	6.10	3.17	1.51	0.77	0.39	0.21	0.10
量子随机数生成器 (Sobol)	3.13	1.56	0.78	0.39	0.19	0.09	0.04
TRUE-BSG [12]†	7.94	5.62	3.97	2.77	1.97	1.37	1.06
IMSNG [11]‡	7.90	5.55	4.08	3.12	2.44	2.01	1.78

∗ 最大周期线性反馈移位寄存器。†、‡ 已选取最优案例。

A. 准确性评估

1）比特流生成：我们首先评估所提出的**APUF−SNG** 设计在生成随机计算比特流方面的表现，并将其与最先进的、基于伪随机数生成器和量子随机数生成器的比特流生成方法，以及采用 MATLAB rand 函数作为随机数生成器的软件生成器进行比较。对于每种方法，我们都将 $[0, 1]$ 区间内的 10,000 个随机实数转换为随机计算比特流。每个输入值转换为不同长度（$\forall L \in \{16, 32, 64, 128, 256, 512, 1024\}$）的随机计算比特流。为使伪随机数生成器方法达到最佳性能，我们实现了与比特流长度相适配的最大周期线性反馈移位寄存器（L）。对于量子随机数生成器方法，我们采用了MATLAB的Sobol 源生成器所生成的第一个 Sobol序列 [25]。表I报告了目标输入值与生成的比特流所表示数值之间的平均绝对误差。对于级数为 4 至 7 的**APUF−SNG** ，较小的状态空间限制了精度。**APUF−SNG**–4 级设计存在严重的量化惩罚，无法准确合成细粒度的目标概率。这导致误差缩放不一致，且在位流长度为 1024 时平均绝对误差高达 5.24%。这些浅级数设计的粗粒度延迟分辨率，使硬件无法有效抑制动态热噪声波动。级数增加到 8 后，平均绝对误差趋于稳定。**APUF−SNG**–16 级设计在 $L = 1024$时达到 1.01% 的平均绝对误差。32 至 128 级设计的误差均不超过 1.01%，因此 16 级即可达到精度上限。

与基于最先进真随机数生成器的随机计算比特流生成器相比，**APUF−SNG**–8级设计的精度与TRUE-BSG设计（1.06%）相当 [12] ，同时优于采用新兴非易失性ReRAM器件开发的真随机生成器IMSNG（1.78%） [11]。伪随机数生成器（线性反馈移位寄存器）和准随机数生成器（Sobol序列）方案的平均绝对误差最低，在1024比特时分别达到0.10%和0.04%。这在意料之中：利用准随机性不需要大量运行，因为其本质上具有确定性。Sobol序列可保证均匀的空间填充概率分布，从而避免纯粹由热噪声驱动的真随机数生成器中不可避免的随机比特聚簇（波动误差）。这一误差下限与噪声规律无关。当查找表根据实际噪声进行校准时， $L = 1024$ 在高斯分布、拉普拉斯分布、均匀分布和学生t分布t 噪声下均保持在$\sigma_{dynamic}$ 0.97%至0.99%之间；而在$\sigma_{dynamic}$ 的情况下则介于0.04至0.15之间。只有开环校准对噪声假设敏感：当实际噪声具有重尾特征，却假设噪声服从高斯分布时，平均绝对误差会升至

5.6%。在硅片上，查找表根据每个激励测得的概率生成，因此不会出现这种误差。尽管准随机数生成器能为随机计算提供最高的计算精度，但其硬件开销显著。所提出的8级 **APUF−SNG** 牺牲了这种分数形式的确定性精度，以实现轻量级硬件占用面积，同时兼具物理不可克隆性和内在的器件安全性。

2) 算术运算： 表 II 将准确性评估从原始比特流生成扩展到基本算术运算的计算精度。图 *1* 的随机计算算术运算面板展示了如何利用简单的逻辑元件在随机计算中实现基本算术运算。对于输入端需要相关比特流的算术运算（例如最小值、最大值、除法和绝对值减法），我们采用了 [26] 中提出的相关性操纵技术，以提高相关性水平。该方法仅在运算输入端增加组合逻辑，不会改变生成器本身。结果说明，浅层 APUF配置（4 至 7 级）固有的概率量化误差会如何在组合逻辑和时序随机计算逻辑中逐级传播。在 4 级设计中，对比特流相关性和精度敏感的运算（例如除法和绝对值减法）会出现严重退化。例如，在 128 比特下，除法运算产生不稳定的 24.48% 平均绝对误差，而绝对值减法的平均绝对误差最高达到 17.13%。与原始生成指标一致，平均绝对误差在 8 级时趋于稳定，并在 16 级时收敛。当位流长度为 1024 时，**APUF−SNG**–16 级设计实现了较低误差：乘法为 0.89%，缩放加法为 1.03%，绝对值减法为 1.47%，最小值为 1.01%，最大值为 1.03%。将 APUF状态空间扩展到 32、64 或 128 级并不能为任何运算带来收益（例如，乘法的误差仍保持在 0.89% 至 0.90%）。

APUF-SNG–8 级设计在所有算术运算中都能与 MATLAB (rand) 基准保持一致。值得注意的是，该设计在除法等时序逻辑运算中优于线性反馈移位寄存器。基于线性反馈移位寄存器的设计在基本乘法运算中具有更低的平均绝对误差（1024 比特时为 0.48%），但其固有的时间自相关会导致其在除法电路中完全失效，误差高达 8.39%。量子随机数生成器设计在所有随机性源中表现最佳（1024 比特时将乘法的平均绝对误差降至 0.07%），这并不意外，因为 LD 序列具有确定且均匀的空间分布。然而，部署量子随机数生成器需要复杂且开销高昂的硬件序列生成器，也无法获得真随机性的优势。

表 II 使用不同随机性源进行随机计算算术运算的平均绝对误差（MAE）（%）比较

方法	随机计算算术运算	位流长度（L）						
		16	32	64	128	256	512	1024
APUF-SNG（4 级）	乘法	8.74	5.51	3.86	3.59	2.58	7.15	4.15
	缩放加法	8.80	6.54	5.06	8.83	6.10	10.94	3.40
	绝对值减法	13.90	11.59	9.15	17.13	11.51	16.65	3.98
	除法	19.67	18.2	14.87	24.48	19.02	17.59	6.90
	最小值	8.78	8.61	6.80	11.37	9.73	7.05	3.03
	最大值	11.72	7.38	5.89	12.35	5.80	18.23	2.67
APUF-SNG（5 级）	乘法	7.12	5.55	3.97	3.25	2.45	3.10	1.76
	缩放加法	9.16	6.22	7.40	3.13	2.65	2.56	1.46
	绝对值减法	12.96			4.66	4.00	4.69	2.28
	除法	16.50 12.14 21.52			7.37	5.98	7.54	4.93
	最小值	7.67 5			3.26	2.55	2.70	1.69
	最大值	12.29 7.52 9.46			3.33	3.17	3.49	1.52
APUF-SNG（6 级）	乘法	7.16	5.54	3.89	2.55	2.07	2.13	1.30
	缩放加法	9.25	7.48	4.18	3.03	2.18	1.90	1.20
	绝对值减法 14.1			5.77	4.14	3.07	2.35	1.80
	除法	18.70	13.14	8.44	6.32	5.03	4.41	3.79
	最小值	8.14	5.72	3.86	2.87	2.11	1.65	1.19
	最大值	13.00	10.86	4.60	3.05	2.22	1.67	1.33
APUF-SNG（7 级）	乘法	7.05	4.95	3.51	2.53	1.76	2.32	1.77
	缩放加法	8.85	6.00	4.77	9.78	2.18	1.48	1.68
	绝对值减法 12.1		8.45	8.63	14.06	3.16	2.10	2.85
	除法	15.96	11.75	14.05	13.05	5.03	4.12	6.59
	最小值	7.55	5.50	6.34	5.24	2.00	1.42	2.45
	最大值	11.50	7.06	5.43	16.01	2.52	1.51	1.28
APUF-SNG（8 级）	乘法	6.99	5.19	3.51	2.50	1.72	1.37	0.91
	缩放加法	8.93	6.00	4.65	2.91	2.05	1.47	1.04
	绝对值减法	12.41	8.38	8.20	4.06	2.88	2.02	1.49
	除法	16.15	11.72	14.17	6.47	5.00	4.08	3.66
	最小值	7.59	5.45	6.46	2.78	1.98	1.41	1.05
	最大值	11.58	7.02	4.58	3.07	2.05	1.43	1.03
APUF-SNG（16 级）	乘法	6.90	4.85	3.47	2.48	1.75	1.26	0.89
	缩放加法	8.80	5.97	4.15	2.89	2.04	1.43	1.03
	绝对值减法	12.23	8.34	5.82	4.02	2.82	2.01	1.47
	除法	16.20	11.69	8.62	6.24	4.93	4.01	3.54
	最小值	7.57	5.44	3.92	2.75	1.98	1.39	1.01
	最大值	11.56	7.03	4.50	3.01	2.04	1.43	1.03
APUF-SNG（32级）	乘法	6.99	4.95	3.47	2.45	1.77	1.26	0.89
	缩放加法	8.84	5.98	4.15	2.88	2.01	1.43	1.04
	绝对值减法	12.28	8.42	5.76	3.97	2.81	2.03	1.46
	除法	16.00	11.70	8.38	6.32	4.91	4.03	3.50
	最小值	7.54	5.45	3.85	2.71	1.95	1.41	1.02
	最大值	11.56	7.02	4.49	3.00	2.01	1.43	1.02
APUF-SNG（64 级）	乘法	6.85	5.04	3.45	2.45	1.76	1.23	0.89
	缩放加法	8.86	6.02	4.10	2.88	2.02	1.45	1.04
	绝对值减法	12.40	8.42	5.79	4.01	2.84	2.05	1.45
	除法	16.14	11.61	8.54	6.21	4.89	4.03	3.54
	最小值	7.67	5.42	3.85	2.72	1.93	1.40	1.00
	最大值	11.53	7.08	4.48	2.99	2.03	1.45	1.02
APUF-SNG（128 级）	乘法	6.95	4.89	3.51	2.44	1.74	1.25	0.90
	缩放加法	8.85	6.01	4.13	2.88	2.03	1.44	1.04
	绝对值减法	12.36	8.24	5.70	3.97	2.81	2.01	1.48
	除法	16.16	11.41	8.33	6.09	4.85	4.04	3.56
	最小值	7.69	5.40	3.84	2.75	1.94	1.39	1.02
	最大值	11.59	7.06	4.47	2.96	2.03	1.41	1.03
MATLAB (rand)	乘法	6.97	4.90	3.50	2.43	1.72	1.23	0.86
	缩放加法	9.05	6.38	4.51	3.18	2.23	1.60	1.13
	绝对值减法	7.75	5.47	3.92	2.77	1.97	1.38	0.98
	除法	12.1	8.59	6.17	4.37	3.12	2.17	1.55
	最小值	7.89	5.54	3.92	2.78	1.94	1.37	0.99
	最大值	7.94	5.52	3.90	2.74	1.96	1.38	0.98
PRNG（$\log_2 L$ 比特线性反馈移位寄存器）	乘法	6.87	4.09	2.55	1.62	1.07	0.70	0.48
	缩放加法	8.54	5.32	3.18	1.97	1.32	0.82	0.26
	绝对值减法	5.35	2.88	1.43	0.68	0.42	0.20	0.10
	除法	13.6	10.54	9.50	8.74	8.82	8.31	8.39
	最小值	6.92	3.81	1.96	0.93	0.57	0.23	0.15
	最大值	4.96	2.64	1.38	0.68	0.37	0.14	0.11
QRNG（Sobol）	乘法	3.88	1.98	1.05	0.58	0.27	0.14	0.07
	缩放加法	3.79	2.59	0.94	0.47	0.33	0.12	0.06
	绝对值减法	2.08	1.04	0.52	0.26	0.13	0.06	0.03
	除法	4.86	3.26	2.86	2.71	2.72	2.76	2.75
	最小值	3.19	1.57	0.79	0.39	0.20	0.10	0.05
	最大值	3.09	1.56	0.77	0.39	0.20	0.10	0.05
TRUE-BSG [12]	乘法	7.05	5.02	3.52	2.47	1.74	1.24	0.88
	缩放加法	8.98	6.42	4.63	3.20	2.30	1.66	1.14
	绝对值减法	7.82	5.63	3.95	2.76	1.92	1.37	0.96
	除法	12.14	8.87	6.18	4.30	3.02	2.16	1.50
	最小值	7.85	5.74	4.04	2.82	1.97	1.41	0.98
	最大值	7.65	5.59	4.02	2.79	2.00	1.45	0.99
IMSNG [11]	乘法	7.00	5.03	3.60	2.63	2.03	1.67	1.45
	缩放加法	9.05	6.42	4.57	3.37	2.46	1.89	1.52
	绝对值减法	8.16	5.86	4.35	3.33	2.67	2.32	2.20
	除法	12.14	8.74	6.51	4.78	3.63	2.93	2.67
	最小值	7.81	5.55	4.00	2.98	2.28	1.83	1.67
	最大值	7.84	5.62	4.11	3.10	2.41	2.05	1.86

B. 硬件成本比较

接下来，我们比较所提出的生成器与最先进的基于线性反馈移位寄存器和 Sobol 的比特流生成方法的硬件成本。我们使用 Verilog 硬件描述语言描述了所有设计，并使用 Synopsys Design Compiler v2024.09-SP4 和 $45nm$ FreePDK 门库 [27] 进行了综合。表III 详细列出了不同方法的硬件成本。

APUF−SNG 呈现硬件线性扩展特性。随着状态空间从 4 级扩展到 8 级，硬件占用面积按比例从 54 增加到 $116\mu m^2$。这种线性

这种增长证实了级联多路复用器链在结构上的高效性。随着级数增加，关键路径延迟（CPL）也呈现出可预测的变化，反映了信号竞争条件下的物理传播延迟。16 级的**APUF−SNG** 配置实现了接近常见 LFSR–比较器基线（$233\mu m^2$）的硅面积（ $240\mu m^2$ ）。然而，传统线性反馈移位寄存器只能提供确定性伪随机性，且存在自相关问题。**APUF−SNG**设计在提供可靠真随机数生成器（TRNG）的同时，还具备固有的物理硬件

表 III 单一随机计算比特流生成的不同方法的硬件成本。（这些设计采用 $45nm$ 合成，精度为 8 位。）

方法	Area (μm^2)	CPL (ns)	功耗@最高频率 (mW)	能耗/周期 (pJ)	面积×延迟 ($\mu m^2 \times ns$)
APUF-SNG–4级	54	0.28	0.59	0.16	15.1
APUF-SNG–5级	70	0.31	0.63	0.19	21.7
APUF-SNG–6级	85	0.34	0.75	0.25	28.9
APUF-SNG–7 级	101	0.45	0.65	0.29	45.4
APUF-SNG–8 级	116	0.41	0.84	0.34	47.6
APUF-SNG–16 级	240	0.67	1.04	0.70	160.8
线性反馈移位寄存器 + 比较器	233	0.37	0.95	0.35	86.2
Sobol + 比较器 [9]	850	0.45	1.31	0.59	382
Sobol（基于有限状态机）[28]	281	0.36	0.72	0.26	101
TRUE-BSG [12]	295	–	–	0.05	–
IMSNG [11]	–	78.2	–	3420	–

不可克隆性，且不会增加面积。

与最先进的研究成果进行基准比较时，APUF-SNG 8级展现出具有竞争力的面积和速度优势。为确保与 TRUE-BSG [12]（其依赖高速环形振荡器）进行公平的架构比较，已将所需 8位二进制比较器的成本计入 TRUE-BSG 基准，使其总面积达到 295 μm^2，能耗为 0.05 pJ/周期。与这一调整后的基准相比，所提出的 APUF-SNG 8级面积小 2.5×（116 对比 295 μm^2）。尽管 TRUE-BSG 的单周期能耗更低，APUF-SNG 8级设计无需多位比较器，并兼具随机计算比特流生成和器件认证两种用途，无需专用密码核心。此外，APUF-SNG 8级优于 IMSNG 设计等替代域生成器 [11]。IMSNG 完全在 ReRAM阵列中运行，但由于繁重的混合信号读操作、参考电流生成和外围电路所带来的开销，其转换延迟较高（78.2 ns），能耗也显著偏高（3,420 pJ/周期）。相比之下，APUF-SNG 8级生成随机计算比特流的速度快两个数量级以上（关键路径延迟为 0.41ns），每周期能耗为 0.34 pJ 。此前的精度评估表明，准随机 Sobol序列的理论平均绝对误差最低，但表 III 展示了其高昂的硬件成本。带比较器的 Sobol生成器占用 850 μm^2， 7× APUF-SNG 8级。即使优化为基于有限状态机的拓扑结构（281 μm^2），Sobol生成器的物理面积仍大于 APUF-SNG 8级，且不具备任何物理不可克隆特性。

C. 总体设计效率

为进一步验证**APUF-SNG** 设计的有效性，我们评估了品质因数（*FoM*），该指标结合了精度和硬件（HW）成本，用于反映总体设计效率：$FoM = \frac{1}{\text{精度} \times \text{硬件成本}}$（平均绝对误差×面积×功耗×延迟）。图4 展示了*FoM* 在各种设计中的评估结果，所使用的数据提取自表I、II和III。结果一致表明，在所有随机计算比特流生成和算术运算中，**APUF-SNG**–8 设计均实现了与 TRUE-BSG [12] 相当的品质因数。尽管基于线性反馈移位寄存器和 Sobol 的设计具有更低的平均绝对误差，但其品质因数在随机计算算术运算中优于其他所有设计。在所有**APUF-SNG** 配置中，8 级设计始终

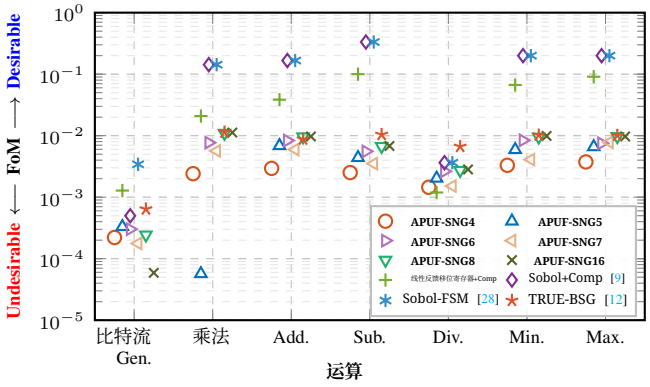

图 4. 不同设计方法的优值 比较。

处于最佳工作点，在精度和硬件成本之间实现了最佳平衡。更深的配置（16级及以上）无法带来显著的精度提升，却会增加面积和功耗，从而使其品质因数下降。

D. 校准、存储与稳定性

每颗芯片只需离线校准一次。根据数百个挑战-响应对拟合 $N+1$ 个 $\vec{w}$ 条目。任何物理不可克隆函数部署过程都会在注册期间收集这些数据，因此无需额外测量。随后，在 Intel Core i7-12700H 上使用单线程 Python 为 $N = 16$ 构建涵盖全部 255 个 8 位电平的表只需 43 毫秒；小型片上校准引擎所需的 1.1×10^6 次乘加运算则远不到 1 毫秒。每个电平存储一个 N 位激励： $N = 8$ 占 256 B，$N = 16$ 占 512 B。只使用固定常数集的应用只需存储这些常数。该表在不同工作条件下也能保持稳定。电压或温度变化会使竞争的两条路径几乎同比减速，因此这一影响在 Δ 中抵消。噪声 $\sigma_{dynamic}$ 按相同的标称延迟缩放，而 p 只取决于比值 $\frac{\Delta}{\sigma_{dynamic}}$。剩余影响则是二阶的。在模型中，噪声尺度变化 20% 会使平均绝对误差增加 2.9% ~~至 3.5%~~；固定偏移 $0.1\,\sigma_{dynamic}$（例如老化或仲裁器失配引起的偏移）会使其增加 2.8%。两者都不会导致性能骤降。接近 $p = 0.5$ 的电平受影响最小，因为此处噪声占主导。如果漂移超过这一程度，重新校准仍是同样的亚秒级离线步骤，并且可以由片上传感器触发。

VI. 结论

本研究将仲裁器PUF重新用作随机数生成器。在亚稳态点附近运行仲裁器PUF，可将路径延迟差转化为可控的输出概率，因此，存储的挑战取代了传统随机数生成器中的随机数生成器和比较器。延迟与噪声模型将每个目标概率映射到一个挑战，并通过枚举 2^N 挑战空间找到该挑战。仿真结果表明，8级**APUF-SNG** 在1024比特下的平均绝对误差为1.04%，与TRUE-BSG和软件真随机数生成器相当，且比特流之间互不相关（SCC $\approx$ 0）。在45 nm中综合后，8级设计占用 116 μm^2 ，每周期功耗为0.34pJ，比 TRUE-BSG小2.5× ，比带比较器的Sobol生成器小 7×。

参考文献

[1] W. Qian、X. Li、M. D. Riedel、K. Bazargan和D. J. Lilja，“基于随机逻辑的容错计算架构”，*IEEE Transactions on Computers*，第60卷，第1期，第93–105页，2011年。[2] A. Alaghi、W. Qian和J. P. Hayes，“随机计算的前景与挑战”，*IEEE Transactions on Computer-Aided Design of Integrated Circuitsand Systems*，第37卷，第8期，第1515–1531页，2017年。[3] M. Moghadam、S. Aygun和M. H. Najafi，“TranSC：使用随机逻辑的超越函数硬件感知设计”，*IEEE Journal on Emerging and Selected Topics in Circuits and Systems*，第1–15页，2026年。[4] P. Ghosh、R. Mukherjee、A. A. Saha、S. Naha、A. Das、A. Raha和M. K. Naskar，“演示摘要：一种用于边缘检测的低功耗实时随机计算硬件加速器”，载于*2025IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*，2025年，第1–2页。[5] M. S. Moghadam、S. Aygun、M. R. Alam和M. H. Najafi，“P2LSG：用于随机计算的二次幂低差异序列生成器”，载于第*29*届亚洲及南太平洋设计自动化会议 *(ASP-DAC)*，2024年，第38–45页。[6] Y. Liu、S. Liu、Y. Wang、F. Lombardi和J. Han，“面向机器学习应用的随机计算神经网络综述”，*IEEE Transactions on NeuralNetworks and LearningSystems*，第32卷，第7期，第2809–2824页，2021年。[7] T. Waho、A. Koyama和H. Hayashi，“Δ-Σ域信号处理：结合随机计算相关主题的综述”，载于*2023 IEEE*第*53*届多值逻辑国际研讨会 *(ISMVL)*，2023年，第88–93页。[8] Y. Zhang、S. Lin、R. Wang、Y. Wang、Y. Wang、W. Qian和R. Huang，“排序网络遇上并行比特流：一种基于随机计算的容错并行三值神经网络加速器”，载于*2020*年欧洲设计、自动化与测试会议及展览 *(DATE)*，2020年，第1287–1290页。[9] S. Liu和J. Han，“使用Sobol序列的能效随机计算”，载于欧洲设计、自动化与测试会议暨展览会 *(DATE)*，*2017*。电气电子工程师学会，2017年，第650–653页。[10] A. Alaghi和J. Hayes，“使用随机电路进行快速、精确的计算”，载于*DATE'14*，2014年3月，第1–4页。[11] J. P. C. de Lima、M. S. Moghadam、S. Aygun、J. Castrillon、M. H. Najafi和A. A. Khan，“使用ReRAM实现全存储器内随机计算”，载于*2025*年第*62*届*AC M/IEEE*设计自动化会议 *(DAC)*，2025年，第1–7页。[12] M. S. Moghadam、S. Williams、A. K. M. Masum、M. H. Najafi、S. Aygun和M. Bayoumi，“TRUE-BSG：用于快速、高效随机计算的真随机比特流生成器”，载于*2025 IEEE*电路与系统国际研讨会 *(ISCAS)*，2025年，第1–5页。[13] J. Aditya、R. J. Moore、J. Miskelly、J. Zhang、W. Liu、M. O'Neill和C. Gu，“近似计算中基于处理器的PUF硅前优化方法”，*IEEE Transactions on Dependable and Secure Computing*，第1–14页，2026年。[14] H. Yan、Y. Su、Q. Wu、Y. Ding、H. Li和Y. Zhang，“一种基于传输门阈值损耗的轻量级仲裁器PUF设计”，载于*2025 IEEE*第*16*届*ASIC*国际会议 *(ASICON)*，2025年，第1–4页。[15] B. Narayanapuram和J. Panda，“物以类聚：一种具有强PUF-TRNG功能的双模式候选电路”，载于*2023 IEEE*第*3*届技术、工程、管理与营销、创业及人才促进社会影响国际会议 *(TEMSMET)*，2023年，第1–6页。[16] I. Tsiokanos、J. Miskelly、C. Gu、M. O'Neill和G. Karakonstantis，“DTA-PUF：面向资源受限设备的动态时序感知物理不可克隆函数”，*ACM J. Emerg. Technol. Comput. Syst.*，第17卷，第3期，第32:1–32:24页，2021年。[17] X. Dong、L. Yao、P. Ding、J. Chen、M. Qiu、P. Kang、Z. Huang、Y. Lu、H. Liang和Y. Xu，“轻量级、高吞吐量、便携式多模式可重构集成PUF-TRNG”，*IEEE Transactions on Circuits and Systems I: Regular Papers*，第1–12页，2026年。[18] K. Pratihar、U. Chatterjee、M. Alam、R. S. Chakraborty和D. Mukhopadhyay，“物以类聚：一种实现强PUF-TRNG功能的双模式电路候选方案”，*IEEE Transactions on Computers*，第72卷，第6期，第1636–1651页，2023年。[19] Y. Gao、S. F. Al-Sarawi和D. Abbott，“物理不可克隆函数”，*Nature Electronics*，第3卷，第2期，第81–91页，2020年。 [在线]。网址：https://doi.org/10.1038/s41928-020-0372-5

[20] G. E. Suh 和 S. Devadas，“用于器件认证和密钥生成的物理不可克隆函数”，载于第*44*届年度设计自动化会议会议录，系列号 DAC '07。美国纽约州纽约市：计算机协会，2007年，第9–14页。 [在线]。获取地址：https://doi.org/ 10 1145 1278480 1278484 。/。

[21] B. Gassend、D. Clarke、M. van Dijk 和 S. Devadas，“硅物理随机函数”，载于第*9*届*ACM*计算机与通信安全会议会议录，系列号 CCS '02。美国纽约州纽约市：计算机协会，2002年，第148–160页。

[22] [ ] 10 1145 586110 586132 在线。获取地址：https://doi.org/。/。M. Majzoobi、F. Koushanfar 和 M. Potkonjak，“硬件安全测试技术”，载于*2008*年*IEEE*国际测试会议，2008年，第1–10页。[23] C. Herder、M.-D. Yu、F. Koushanfar 和 S. Devadas，“物理不可克隆函数及其应用：教程”，*IEEE*会议录，第102卷，第8期，第1126–1141页，2014年。[24] A. Alaghi 和 J. P. Hayes，“利用随机电路设计中的相关性”，载于*ICCD*，美国北卡罗来纳州阿什维尔，2013年，第39–46页。[25] S. Liu 和 J. Han，“利用并行Sobol序列实现能效随机电路”，*IEEE*超大规模集成电路系统学报，第26卷，第7期，第1326–1339页，2018年。[26] S. Asadi、A. H. Jalilvand、M. H. Najafi 和 M. Bayoumi，“Eco：面向低差异随机计算的增强型流内相关性操控”，*IEEE*超大规模集成电路系统学报，2025年。[27] NCSU，“NCSU FreePDK 45 纳米库”，https://eda.ncsu.edu/ downloads/，访问日期：2026年2月23日。[28] S. Asadi、M. H. Najafi 和 M. Imani，“一种用于低差异随机计算的低成本基于有限状态机的比特流生成器”，载于*2021*年欧洲设计、自动化与测试会议暨展览会（*DATE*），2021年，第908–913页。



Shelby Williams （电气电子工程师学会会员）于1996年和2020年分别获得美国路易斯安那大学拉法叶分校（路易斯安那州拉法叶）的电气工程与计算机科学理学学士和理学硕士学位，并在该校获得计算机工程博士学位。他目前的研究兴趣包括电子学、超大规模集成电路、微电子学、随机数生成器和脉冲神经元。



Mehran Moghadam （S'22）于2010年和2016年分别获得伊朗伊斯法罕大学计算机工程理学学士学位和计算机系统架构理学硕士学位。他在这两个专业的学习中均名列前茅。2022年，他开始在美国路易斯安那大学拉法叶分校计算与信息学院攻读计算机工程博士学位。2024年，他转入凯斯西储大学电气、计算机与系统工程系（俄亥俄州克利夫兰），美国，以继续攻读计算机工程博士学位。他入围了2024年ESWEEK和ICCAD的ACM SIGBED学生研究竞赛（SRC）决赛，并获选为DAC 2024和2026青年 Fellow。Mehran在ISLPED'26会议上获得最佳论文奖。此外，他于2026年获得IEEE电路与系统学会（CASS）博士生研究资助奖。他的研究兴趣包括新兴的非常规计算范式，例如节能随机计算模型、实时且高精度的超维计算系统、比特流处理、稳健的存内算术计算，以及面向边缘人工智能的低功耗传感器内/近传感器计算设计。



M. Hassan Najafi （S’15-M’18-SM’23）于2011年、2014年和2018年分别获得伊朗伊斯法罕大学计算机工程理学学士学位、伊朗德黑兰大学计算机体系结构理学硕士学位，以及美国明尼苏达大学双城分校电气工程博士学位。2018年至2024年，他在美国路易斯安那州拉法叶的路易斯安那大学拉法叶分校计算与信息学学院担任助理教授。目前，他是电气、

美国俄亥俄州克利夫兰凯斯西储大学计算机与系统工程系。他的研究兴趣包括随机计算、近似计算、一元处理、存内计算和超维计算。他已独立或合作发表 120 多篇同行评审论文，并获授 12 项美国专利，另有更多专利正在申请中。Najafi 博士于 2024 年获得 NSF CAREER奖，曾获 GLSVLSI’23 和 ICCD’17 最佳论文奖、GLSVLSI’24 和 DCAS’26 最佳海报奖、2018 年 EDAA杰出学位论文奖，以及明尼苏达大学博士论文奖学金。他曾担任 IEEE Journal on Emerging and Selected Topics in Circuits and Systems 的编辑，并担任多个 EDA会议的技术程序委员会委员。Najafi 博士是 IEEE高级会员，也是美国国家发明家院（NAI）高级会员。



Sercan Aygun （S’09-M’22-SM’24）于 2013 年获得土耳其埃斯基谢希尔奥斯曼加齐大学电气与电子工程理学学士学位，并同时修读计算机工程双专业。他于 2015 年获得伊斯坦布尔技术大学电子工程理学硕士学位，并于 2016 年获得阿纳多卢大学计算机工程第二个理学硕士学位。Aygun 博士于 2022 年获得伊斯坦布尔技术大学电子工程哲学博士学位。Aygun 博士获得了 ACM 最佳科学研究奖

SIGBED 学生研究竞赛（SRC）ESWEEK 2022、GLSVLSI’23 最佳论文奖，以及 GLSVLSI’24 和 DCAS’26 最佳海报奖。Aygun 博士的博士研究获得了土耳其电子制造商协会颁发的最佳科学应用博士奖，并在土耳其科学院评选的科学与工程博士论文奖中名列全国第一。目前，他是美国路易斯安那州拉法叶市路易斯安那大学拉法叶分校计算与信息学学院的助理教授。他的研究方向包括微型机器学习和新兴计算，涵盖随机计算与超维计算。



Arman Roohi （电气电子工程师学会高级会员）于 2019 年获得美国佛罗里达州奥兰多市中佛罗里达大学计算机工程博士学位。目前，他是美国伊利诺伊州芝加哥市伊利诺伊大学芝加哥分校电气与计算机工程系的副教授。在加入 UIC 之前，他曾于 2020 至 2024 年担任内布拉斯加大学林肯分校的助理教授，并于 2019 至 2020 年担任德克萨斯大学奥斯汀分校的博士后研究员。他的研究兴趣包括跨层

用于实现复杂机器学习任务和安全计算的协同设计，包括硬件安全与人工智能安全、可重构与自适应计算机架构，以及后 CMOS 计算。他已围绕这些主题发表了 50 多篇论文，其中包括最佳论文奖项、书籍章节和 STEM课程开发成果。他曾获得 DAC 2018博士论坛奖学金、2018 年 Frank Hubbard工程捐赠奖学金；2019 年获得 IEEE TRANSAC-TIONS ON EMERGING TOPICS IN COMPUTING 最佳论文奖项，并于 2017 年获得 IEEE TRANSACTIONS ON COMPUTERS 月度论文奖。



Gourav Datta 博士 是凯斯西储大学（CWRU）电气、计算机与系统工程系的助理教授。在此之前，他曾担任亚马逊通用人工智能的应用科学家，参与提升 Amazon Nova（一种下一代基础模型）的视频理解能力。他于 2023 年获得南加州大学哲学博士学位。他的研究兴趣包括面向边缘机器学习的高能效、低延迟算法—硬件协同设计。在攻读哲学博士期间，他获得了 USC Viterbi

工程学院的最高成就奖——2024年William Ballhaus最佳博士论文奖、2022年IEEE应用超导研究生奖学金、南加州大学电气与计算机工程系最佳研究助理（RA）奖，以及安纳伯格奖学金。他已在《自然·科学报告》、《神经科学前沿》、国际学习表征会议、欧洲计算机视觉会议、IEEE电路与系统汇刊I、DATE、冬季计算机视觉应用会议等顶级学术会议和期刊上发表了60多篇同行评审论文，并获提名角逐2022年超大规模集成电路与片上系统会议最佳论文奖。最后，他关于传感器内计算的研究曾多次受到Edge Impulse（微型机器学习领域领先的半导体知识产权公司）和南加州大学维特比工程学院的重点报道。