分类号：TN402　　单位代码：10363
密　级：公开　　学　　号：2220320146

基于抖动和亚稳态采样的异或门链 TRNG 设计方法研究

RESEARCH ON XOR GATE CHAIN TRNG DESIGN METHOD BASED ON JITTER AND METASTABLE SAMPLING

学 生 姓 名 ：叶新伟
校 内 导 师 ：倪天明（教授）
专业学位类别：电子信息（085400）
研究方向(领域)：硬件安全

论文答辩日期：2025 年 5 月 10 日

二、安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：叶亲伟

日期：2025 年 5 月 6 日

三、安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 □，在 ____ 年解密后适用本版权书。

本学位论文属于

不保密 ☑。

学位论文作者签名：[illegible]　　指导教师签名：[illegible]

日期：2025 年 5 月 16 日　　日期：2025 年 5 月 16 日

基于抖动和亚稳态采样的异或门链 TRNG 设计方法研究

摘 要

真随机数发生器（True Random Number Generator, TRNG）是密码学、统计学和信息技术等领域的关键，其输出的随机序列在加密算法、安全通信和仿真模拟中具有不可替代的作用。传统基于环形振荡器（Ring Oscillator, RO）的 TRNG 设计虽然被广泛使用，但其依赖抖动的累积来生成随机性，导致时间消耗长、效率低，难以满足高性能应用需求。本文提出了一种基于异或门链采样的新型 TRNG 设计方法，克服了传统方案的局限性，并在现场可编程门阵列（Field Programmable Gate Array, FPGA）平台上实现了高效的硬件实现。

本文的创新点主要包括以下三个方面：

1. 提出了一种新的抖动捕获机制：在传统基于 RO 的 TRNG 基础上，通过逻辑门的短脉冲抑制效应精细化捕获时钟抖动，实现了对抖动的高效利用，同时结合亚稳态熵源，显著提高了随机性生成的质量与效率。

2. 优化了硬件结构与资源效率：所提出的 TRNG 仅占用 25 个查找表（Look Up Table, LUT）和 6 个触发器（DFF），在 Xilinx Artix-7 和 Virtex-7 FPGA 平台上分别实现了 360MBps 和 330MBps 的高吞吐量，满足了高性能应用的需求。

3. 验证了高随机性与环境适应性：生成的随机序列通过了 NIST SP800-22 和 NIST SP800-90B 两项权威测试标准的全面评估，测试结果优秀，通过率高。此外，在不同电压（0.8V 至 1.1V）和温度（20°C 至 80°C）条件下的测试结果表明，所提出的 TRNG 具有良好的环境适应性和鲁棒性，同时，在 UltraScale+ VCU118 FPGA（16nm 工艺）上进行了严格测试，实验结果表明其熵质量和吞吐量仍然很高，进一步验证了方法的鲁棒性与跨平台适用性。

实验结果表明，本文提出的 TRNG 设计在随机性、稳定性和硬件效率等方面表现优秀。其 NIST 测试通过率超过 96%。与其他基于 FPGA 的先进 TRNG 相比，所提出的设计在硬件资源消耗和吞吐量方面具有显著优势，展现了更高的综合性能，所研究成果为真随机数发生器的硬件实现提供了新的设计思路。

关键词：真随机数发生器，环形振荡器，抖动，短脉冲，现场可编程门阵列

Research on XOR gate chain TRNG design method based on jitter and metastable sampling

Abstract

True Random Number Generator (TRNG) is the key in the fields of cryptography, statistics and information technology, and its output random sequence plays an irreplaceable role in encryption algorithms, secure communication and simulation. Although the traditional TRNG design based on Ring Oscillator (RO) is widely used, it relies on the accumulation of jitter to generate randomness, resulting in long time consumption and low efficiency, which is difficult to meet the needs of high performance applications. In this paper, a novel TRNG design method based on XOR Gate chain sampling is proposed, which overcomes the limitations of traditional schemes and achieves efficient hardware implementation on the Field Programmable Gate Array (FPGA) platform.

The innovation of this paper mainly includes the following three aspects:

1.A new jitter capture mechanism is proposed: based on the traditional RO-based TRNG, the clock jitter is captured by the short pulse suppression effect of the logic gate, which realizes the efficient use of jitter. At the same time, the quality and efficiency of randomness generation are significantly improved by combining with the stable entropy source.

2.Optimized hardware structure and resource efficiency: the proposed TRNG only takes up 25 lookup tables (LUT) and 6 triggers (DFF), and achieves high throughput of 360MBps and 330MBps respectively on Xilinx Artix-7 and Virtex-7 FPGA platforms, meeting the needs of high-performance applications.

3.High randomness and environmental adaptability have been verified: The generated random sequences have passed the comprehensive evaluation of two

authoritative test standards, NIST SP800-22 and NIST SP800-90B. The test results are excellent and the pass rate is high. Furthermore, the test results under different voltages (0.8V to 1.1V) and temperatures (20° C to 80° C) show that the proposed TRNG has good environmental adaptability and robustness. Meanwhile, strict tests have been conducted on the UltraScale+ VCU118 FPGA (16nm process). The experimental results show that its entropy quality and throughput are still very high, further verifying the robustness and cross-platform applicability of the method.

The experimental results show that the TRNG design proposed in this paper performs well in terms of randomness, stability and hardware efficiency. Its NIST test pass rate exceeds 96%. Compared with other advanced TRNNS based on FPGA, the proposed design has significant advantages in terms of hardware resource consumption and throughput, demonstrating higher comprehensive performance. The research results provide a new design idea for the hardware implementation of true random number generators.

KEYWORDS: True random number generator, Ring oscillator, Jitter, Short pulse, Field programmable gate array.

目 录

图目录

表目录

第一章 绪论

1.1 研究背景与意义

随着信息技术的高速发展，数据存储与交换变得越来越频繁，大量数据成为网络犯罪分子的攻击目标。苹果公司的一项研究报告表明数据泄露已呈泛滥之势，严重威胁着全球敏感及个人消费者数据[1]。

从数据泄露的规模来看，2013 至 2022 年期间，其总数增长超两倍，仅过去两年就有 26 亿条个人记录被暴露，而 2023 年更是持续恶化。仅美国在 2023 年前 9 个月的数据泄露事件相比以往任何一年就多出近 20%。其中，云数据成为重灾区，超 80%的泄露事件涉及云存储数据，且从 2021 到 2022 年针对云基础设施的攻击近乎翻倍。

勒索软件团伙及协调活动是数据泄露加剧的重要因素。截至 2023 年 9 月，勒索软件攻击数量相比 2022 年前三季度增加近 70%，甚至超过了 2022 年全年总和，这导致美英等国账户泄露数量在 2023 年上半年较 2022 年同期增加一倍多。

面对如此严峻的数据安全形势，科技行业积极应对。密码学是信息安全领域用来对抗信息窃取的重要手段，而随机数在密码学中占有非常重要的地位，我们日常使用的密码学算法基本都要用到一些对于窃密者来说必须是难以破解的机密数据，而这些数据基本都是随机数 [51]。对于一次一密的加密系统而言，其安全性从根上来说完全依赖于密钥，包括对称加密算法（DES、AES 等）的密钥和非对称加密算法（RSA、DSA 等）的密钥对，而这些密钥则必须为随机数[3]。现代密码学系统安全性的一个重要来源就是用来产生机密数据所使用的随机数的随机性，密码学家 Bruce Schneier 曾说过：“尽管随机数序列是一个被较少讨论的密码学问题，但是它比其它任何密码学问题都重要[4]。”

例如，近年来，随着用户和互联网企业安全意识的提高和 HTTPS（超文本传输安全协议）成本的下降，HTTPS 已经越来越普及。很多互联网巨头也在力推 HTTPS，比如谷歌的 Chrome 浏览器在访问 HTTP（超文本传输协议）网站时会在地址栏显示不安全的提醒，微信要求所有的小程序必须使用 HTTPS 传输协议，苹果也要求所有在 AppStore 上架的应用必须采用 HTTPS，国内外的大部分

主流网站也都已迁移至 HTTPS。HTTPS 自 1994 年被首次使用以来，已经几乎取代了过去的 HTTP（超文本传输协议）协议，这背后最大的原因就是不安全。

由图 1-1 可知，HTTP 在传输数据的过程中，所有的数据都是明文传输[5]。由于传输过程中没有保密措施，自然没有安全性可言，特别是一些敏感数据，比如用户密码和信用卡信息等，一旦被第三方获取，后果不堪设想。因此，HTTP 被扩展为了现在流行的 HTTPS 协议。

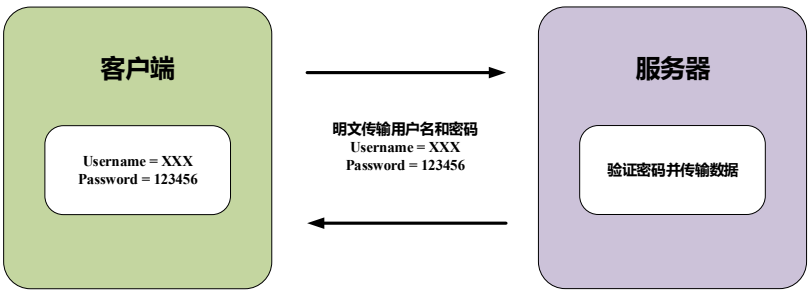


图 1-1 HTTP 数据传输过程

HTTPS 的具体通信过程如图 1-2 所示，客户端向服务器发送 HTTPS 连接请求，服务器选择双方支持的 SSL/TLS 协议版本并发送证书，客户端验证证书有效性后生成随机对称密钥，用服务器证书公钥加密后发送给服务器，服务器用私钥解密得到对称密钥，之后双方用此对称密钥加密传输数据，完成数据传输后，客户端和服务器按需关闭连接，期间 SSL/TLS 协议保障了数据传输的安全可靠，防止数据遭窃取、篡改与冒充。而在整个传输过程中，数据的保密完全依赖秘钥及随机数，一但秘钥被人获取，那么整个数据传输系统的安全就不复存在[6]。因此，需要提供不可预测的随机数，来保证整个系统的安全性。

随机数不仅在信息安全领域有着重要的应用，而且在扩频通信、博彩、赌博机、音乐和图像合成、模拟和测试等多个领域都有着广泛的应用[7]。在一些科学研究（如蒙特卡洛模拟)中也经常使用到随机数，而且在一些自然现象的研究中，随机性更是必不可少的条件。

随机数发生器作为现代电子及通信系统中必不可少的一部分。按其产生方式可以分为真随机数发生器（True Random Number Generator, TRNG）和为随机数发生器（Pseudo Random Number Generator, PRNG）。伪随机数发生器通常用于快速生成所需的随机数，其本质上是一个数学模型或公式，具有以其状态数为界的

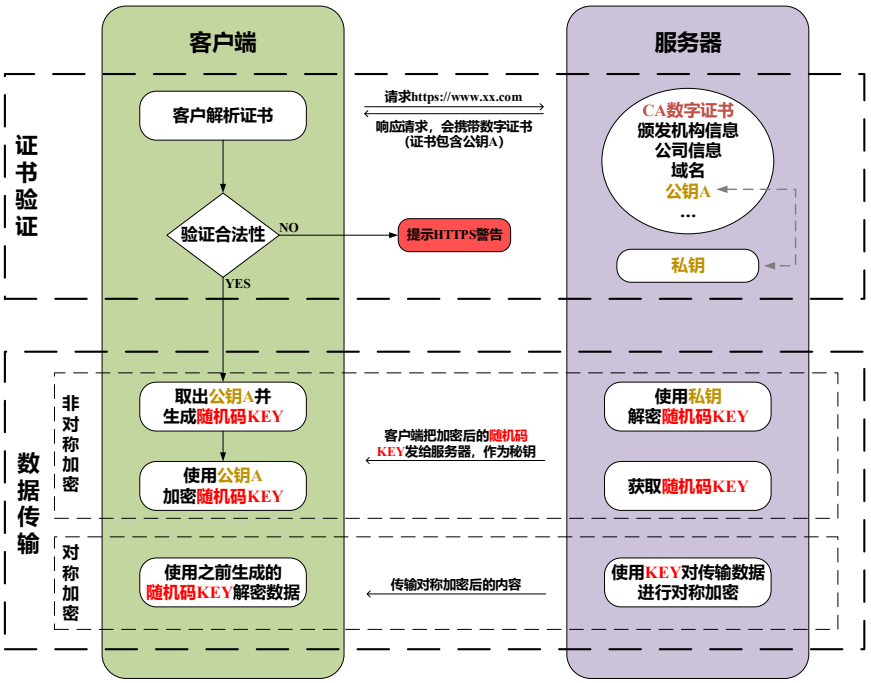


图 1-2 HTTPS 加密、解密、验证及数据传输过程

周期性，输出完全由其初始状态决定，该状态通常被称为“种子”。PRNG 易于实现且统计特性良好，但其安全性却较差，一旦种子被获取，产生的随机数就很容易被预测[8]。而 TRNG 利用物理过程中随机量作为熵源，可以产生不可预测性的随机数，和 PRNG 相比，TRNG 是独立且不可预测的[9]。在密钥生成、向量初始化和统计模拟等 场合中，TRNG 生成的随机序列不仅可以提供卓越的统计特性，还能保证足够的安全性。因此，TRNG 具有重要的研究意义。

1.2 国内外研究现状

当下，随机数发生器研究范畴广泛，内容多元。从理论到应用，从技术创新到性能提升，为多领域提供关键支撑。

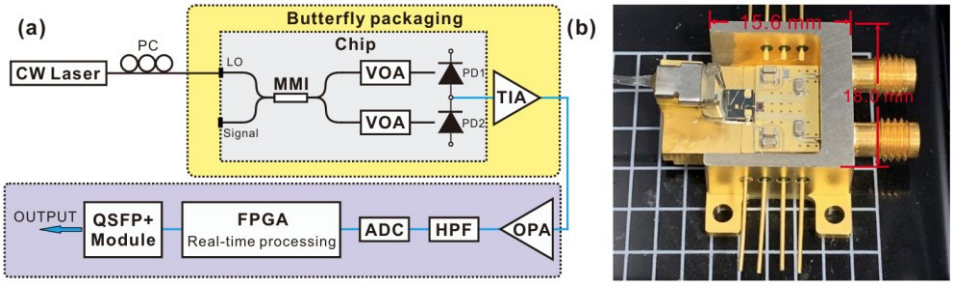
最初，人们从自然现象如放射性物质衰变、热噪声等获取随机数，这些物理过程具有内在随机性，是早期真随机数的主要来源。但这种方式受限于环境条件复杂、难以精确控制和稳定产生等问题，例如放射性物质存在安全隐患且其衰变过程不易于小型化、集成化设备的使用。

随着电子技术发展，利用电子元件的噪声特性（如电阻的热噪声、半导体器件的散粒噪声等）产生随机数成为新途径。像基于雪崩二极管的电路，通过雪崩

击穿产生的随机电流来提取随机数[10]。不过这类方法存在稳定性、随机性质量有待提高、对电路参数敏感等缺点，易受温度、电源波动等因素干扰，所以使用起来并不方便。

如今，真随机数发生器设计更为精巧复杂，结合多种物理机制和数字处理技术。例如将混沌系统与数字电路结合，利用混沌系统对初始条件极度敏感的特性生成随机数，再经数字逻辑处理[11]、后处理算法优化，提升随机性质量和生成速率，可集成到芯片中，广泛应用于加密、通信、模拟等众多领域，满足对高质量真随机数不断增长的需求，推动着真随机数技术向更高效、更可靠方向持续发展。

例如，中国科学技术大学教授潘建伟、张军等联合浙江大学教授储涛研究组，通过研制硅基光子集成芯片和优化实时后处理，实现了速率达 18.8Gbps（千兆比特每秒）的实时量子随机数发生器（Quantum Random Number Generator, QRNG）[12]，其结构图见图 1-3。



（a）实现原理图　　（b）随机数发生器实物

图 1-3 QRNG 模块原理图

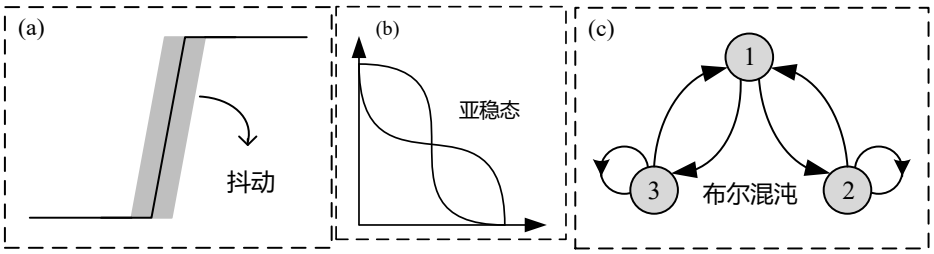
当下，已经有多家企业已将 TRNG 商业化。例如，华为将 TRNG 技术应用在通信设备和安全芯片里面，提升了设备的安全性，通过这种方法来保证信息传输和存储过程中，数据的安全性性和完整性。图 1-4 为他们所开发的服务器密码机和云服务器密码机。

图 1-4 华为服务器密码机和云服务器密码机

在集成电路行业蓬勃发展的背景下，FPGA 因其独特的设计灵活性、出色的兼容性以及高效的开发速度等显著优势，于真随机数发生器的应用领域脱颖而出，已逐步发展成为当下最为流行且广泛应用的实现平台之一，为 TRNG 技术的进一步发展与突破提供了坚实的硬件基础和广阔的创新空间。

当前 TRNG 的主要研究方向有三个部分，分别是熵源、采样电路和后处理电路。其中，熵源作为随机性的根基，像量子噪声、热噪声等物理现象能产生天然随机信号，为 TRNG 提供原始随机信息，其特性研究至关重要；采样电路负责将熵源的微弱随机物理信号转化为数字信号，要求具备高灵敏度与精度，平衡速度与精度，确保信息准确采集与转换。后处理部分则运用算法对采样后的原始数字信号优化提纯，消除偏差与相关性，使其符合随机性统计标准，满足加密、通信等领域对随机数的严格要求。这三部分的详细说明如下：

第一部分是在熵源上进行创新，目前，在 TRNG 中常用来产生真随机数的熵源主要有抖动[13]~[22]、亚稳态[14][18][21]~[27]、热噪声[33]、量子效应[12][38]~[40]、混沌[41]~[47]及散粒噪声等。具体如图 1-5 所示，抖动是信号的实际位置与其理想位置之间的偏差，通常由 RO[15][48]极其变体[49]或 PLL[50][51]等产生。在长时间的积累下可以获得较好的随机性，但会降低吞吐量，且硬件资源消耗较大；在数字电路中如果数据信号转换违反了寄存器的建立时间和保持时间，则寄存器的输出可能进入亚稳态[52]。在 FPGA 中，通常使用触发器的时序违规[53]或交叉耦合的反相器[28]等作为产生亚稳态的来源。利用亚稳态电路作为熵源的 TRNG 虽然结构简单，但是由于工艺偏差，布局布线等原因，生成的随机数偏差较高，需要冗杂的后处理电路；基于混沌的 TRNG 一般通过混沌系统（如布尔混沌）的内在随机性和不可预测性，而混沌系统对于初始条件的高度敏感可以让基于此种方式的 TRNG 所产生的随机数难以预测。



（a）抖动　　（b）亚稳态　　（c）混沌

图 1-5 真随机数发生器常用熵源

为了提升 TRNG 的随机效果，很多研究者在熵源上做出了许多改进，如[14][22]等利用混合熵源，来提升随机性。此外，还有一些利用环形电路中信号追赶机制的真随机数发生器[54][55]，来减轻电路的复杂度。

第二部分是利用不同的采样电路对信号进行处理。D 触发器是最常见的采样单元[56]，具有良好的稳定性和可靠性,其内部的逻辑结构和工作原理经过了长期的实践验证和优化,在真随机数发生器中,能够稳定地对随机源信号进行采样和转换,减少了因器件本身的不稳定或故障而导致随机数产生错误或偏差的可能性。通常可以用周期信号采样非周期信号，或者非周期信号采样周期信号。而相干采样[57][58]是由两个具有相似周期的振荡信号的结构组成，它利用两个相近信号频率之间的微小差异来区分抖动积累，从而显著提高捕获抖动的概率，比直接采样获得更高的输出比特率，但同时也会使累积抖动的时间大大增加，从而降低 TRNG 的吞吐率。在以 PLL 和 DCM 等时钟信号中的抖动作为熵源时，拍频检测[59][60]也是常用的方法。利用抽头延迟链[13]或快速进位链可以同时提高采样精度及速度，如图 1-6 所示，但所需的硬件开销也会随之上升。在熵源随机性较低，且噪声累积速度慢的情况下，也会使用计数器进行采样[54]，通过增加长时间的计数，牺牲吞吐量来获得良好的随机性。

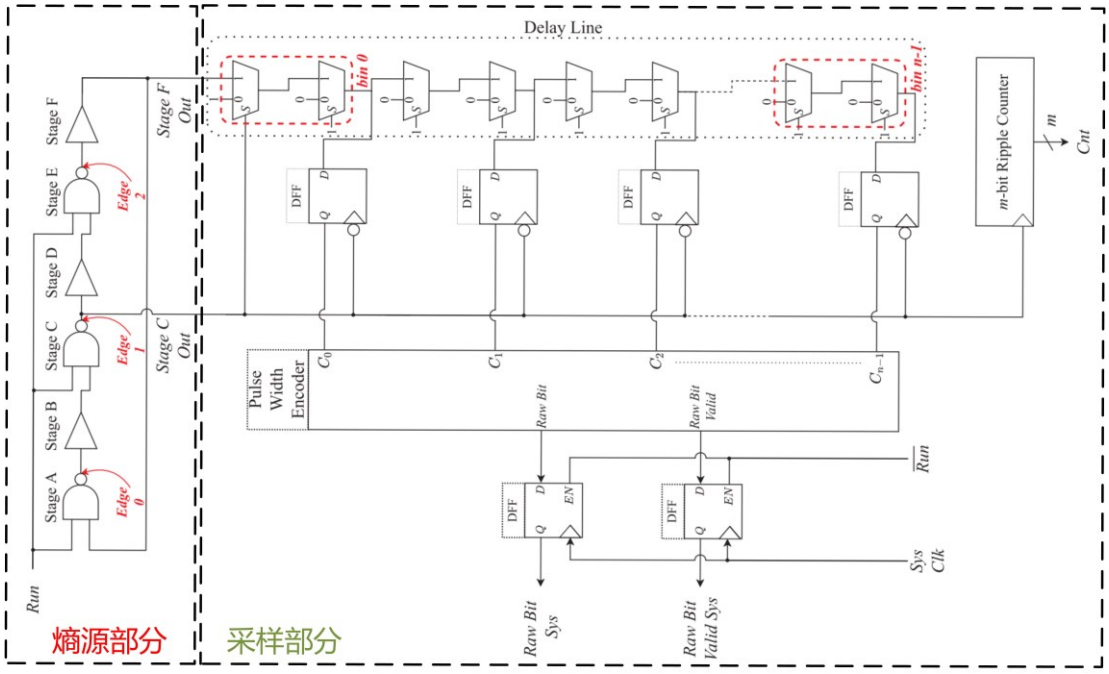


图 1-6 利用延迟链作为采样方式

第三部分是提出和改进电路的后处理模块，在数字电路中，常见的后处理模

块有异或处理，冯诺依曼后处理和线性反馈移位寄存器纠偏（LFSR）等等。以异或处理为例，它是以成倍的硬件资源为代价，在不损失吞吐量的情况下，提高输出的随机性和偏置，适用于大多数的 TRNG。如图 1-7 所示，p和q为两个 TRNG 输出随机数为 1（或 0）的概率，$0 < p$，$q < 1$，将两个输出进行异或后，其输出为 1（或 0）的概率和熵分别为：

$$p_O = p(1-q) + q(1-p) = \frac{1}{2} - 2(p-\frac{1}{2})(q-\frac{1}{2}) \qquad (1\text{-}1)$$

易证p_O比p和q更接近 1/2，对应的香农熵就更接近 1，具有更好的随机性。故常常利用多个相同的 TRNG 模块进行异或，得到处理后的最终输出，设每个模块的输出都是独立同分布的，输出随机数为 1（或 0）的概率为p_0，则n个模块的异或输出为 1（或 0）的概率为：

$$p_{o_n}(p_0, n) = p_0 \sum_{i=0}^{n} (1-2p_0)^i = \frac{1-(1-2p_0)^{n+1}}{2} \qquad (1\text{-}2)$$

由上式可知，模块的数量越多，最终输出的偏置越低，随机性越好。

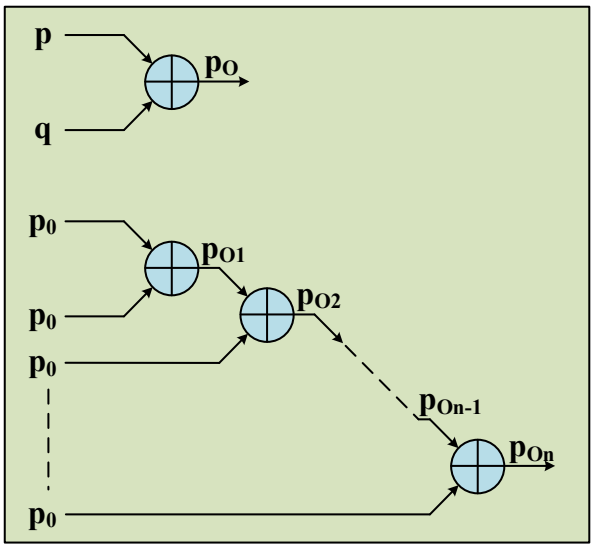


图 1-7 异或后处理

图 1-8 是利用线性反馈移位寄存器纠偏，它的的特点是下一状态由当前状态与反馈多项式决定。处理后的真随机数的熵值和 LFSR 的压缩率直接相关，后处理后真随机数的比特相关性也和处理前随机序列比特间相关性有直接相关，所以基于 LFSR 后处理的方法只能用于输出比特间相关性低的场合。

图 1-9 描述的是一个冯诺依曼后处理电路模块。具体过程是将随机数中每两个比特作为一个块(a_1, a_2)若$a1 = a2$，则将块整体丢弃，若$a1 \neq a2$，则输出$a = a1$。若原始比特偏差为k，则$P(a=1) = (0.5+k)(0.5-k)$，该方法通过牺牲吞

吐量来提高随机数的质量，但电路结构较为简单，且易于实现。

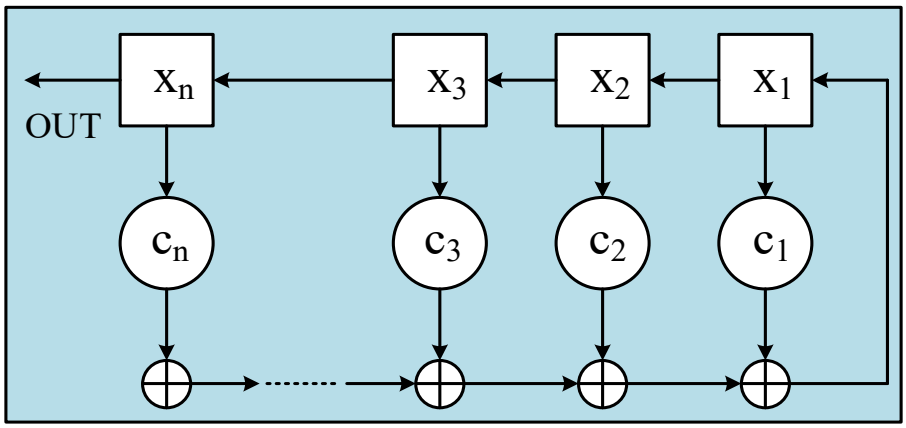


图 1-8 线性反馈移位寄存器纠偏

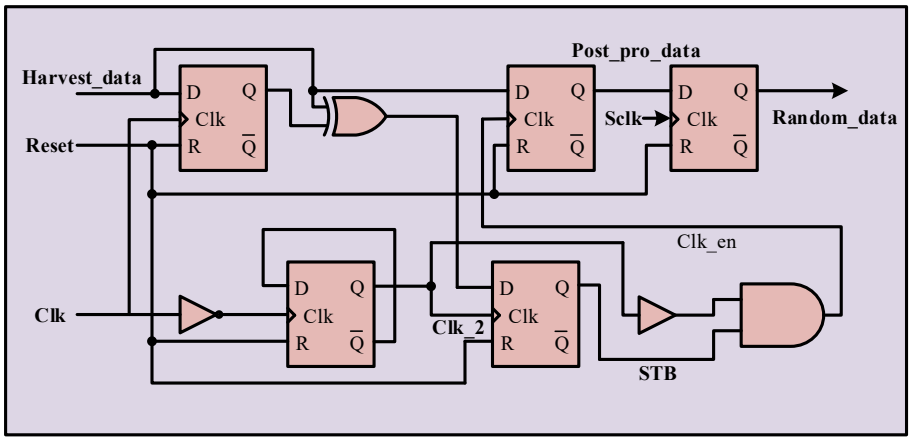


图 1-9 冯诺依曼后处理

1.3 本文的研究内容及组织结构

1.3.1 本文的研究内容

本文主要是进行真随机数发生器的熵源分析、量化方式改进、提高吞吐量及降低硬件开销的方法进行研究。传统基于 RO 和 PLL 等真随机数发生器需要大量的抖动累积时间或叠加更多电路模块提升最终输出的随机性。通过布局布线等方式来调节振荡电路的内部延迟，可以提升采样时的随机性，但会提高电路的复杂度，导致其难以移植，甚至会降低真随机数发生器的输出速率。为了更高效地利用时钟抖动，常常利用进位链或抽头延迟链来高精度采样信号的抖动区域，这样可以减小抖动的累积时间，但是这会带来大量的硬件开销，增加电路设计的成

本。而以亚稳态为熵源的真随机数发生器，虽然结构简单，但由于工艺偏差等因素，导致亚稳态难以产生，需要严格的设计来满足对称布局，使得电路几乎无法移植，而且，产生亚稳态的锁存器结构每输出一个随机数，都需要进行重启，同时，产生的亚稳态需要较长的时间来输出稳定值，故吞吐率较低。因此，提高对熵源采集效率的同时，降低电路的硬件资源，提高真随机数发生器的输出速率是需要深入探究并加以解决的关键要点之一。

高品质的真随机数发生器常常应用于各种场景，不仅需要输出高度随机的随机数，还应具备强大的抗干扰能力。面对外界复杂的环境变化（温湿度等），都可以稳定运行。因此，提高真随机数发生器运行的稳定性和抗干扰可靠性也是需要解决的问题之一。

针对上述问题，本文在以下的几个方面进行了研究：

深入分析时钟抖动的来源和累积过程，为 TRNG 的设计提供理论依据和数学模型。；优化电路设计，即在提高电路采样精度的同时，降低资源消耗，实现更轻量化的电路设计，同时，结合多种熵源，以此来更高效地利用噪声源；对于产生亚稳态的苛刻条件，常用方法是严格的对称布局，或者利用大量的触发器列阵，前者对工艺要求极其严格，后者则占用大量硬件资源，本文在无需布局的情况下，提高亚稳态发生的概率，并将亚稳态与时钟抖动这两种熵源相结合，进一步提高电路的随机性;；可靠性与稳定性是大多数电路所需要的，设计的真随机数发生器需要在各种环境下（工艺、电压及温度等）稳定运行。此外，设计电路的随机性质量、吞吐率、硬件开销及可移植性也会被验证。

1.3.2 本文组织结构

本文的结构如下：第一章为绪论，介绍真随机数发生器的研究背景和意义，总结国内外研究现状，并明确本文的研究内容和整体框架；第二章为随机数基础，介绍随机数的基本概念、特性及其分类，分别讨论 PRNG 和 TRNG 的工作原理，并介绍了随机数的统计方法。第三章为 TRNG 设计，重点研究基于环形振荡器（RO），分析其随机性来源和熵源特性，提出一种基于新型采样链的量化抖动同时结合亚稳态的设计思路，并详细说明电路结构。第四章为所设计的 TRNG 的实现与测试，通过在 FPGA 平台上实现所设计的 TRNG，并对它生成的随机序列

进行随机性分析，包括重启测试、偏移度测试、自相关测试以及温压测试，同时与现有结构进行对比分析。第五章总结全文的研究成果，并对未来研究方向进行展望。通过上述内容，本文系统地研究了 TRNG 的设计、实现与性能分析，为相关领域的研究提供了理论支持和实践参考。

第二章 随机数相关概念及发生器原理

2.1 随机数的基本概念

2.1.1 随机数的定义

随机数是在一定范围内按照随机分布的序列。从数学角度看，在样本空间里面，每个数被选中的可能性都符合一定的概率分布。例如，在均匀分布的随机数中，区间[a,b]内的任意一个数被选中的概率都是相等的。在计算机科学中，随机数主要由特定算法生成。而随机数发生器根据其生成方式，主要可以分为两类：

伪随机数发生器：将一个初始值（种子）输入给一个复杂的算法，经过一系列复杂的计算后，可以得到看似随机的数。由于生成数字的算法是确定的，所以只要种子相同，生成的序列就必然相同，因此它是“伪随机”的。

真随机数发生器：以自然环境中的不确定性（如热噪声、核衰变等现象）为熵源，将其随机性提取出来后，可以生成真正不可预测的随机数。

随机数在生活中有很多应用，比如：在密码学领域，随机数常常被当作密钥，密钥的随机性越高，则加密信息就越安全；而在统计学中，可以通过生成大量的随机数模拟一些复杂的行为，如蒙特卡洛模拟等；在游戏开发过程中，随机数可以用于生成随机事件（如暴击率等等），增加游戏的趣味性和不确定性。

2.1.2 随机数的特性

不可预测性：这是随机数最为显著的特性之一。真正的随机数生成过程就像是大自然的神秘“抽签”，完全脱离人类的掌控与预知。以彩票为例，彩票号码通过专门的随机数生成设备产生，购买者即便穷尽各种数据分析方法，也无法从过往的开奖号码中找到固定规律，从而准确预测下一期的中奖号码。每一次开奖都是独立的随机事件，前一次的结果对下一次毫无影响，这使得彩票结果充满了不确定性。在计算机模拟复杂物理现象时，随机数的不可预测性同样至关重要。例如模拟分子的布朗运动，分子在各个时刻的运动方向和速度变化都由随机数决定，由于其不可预测性，才能真实地反映出分子运动的无序状态。

独立性：随机数的生成彼此之间不存在关联，每个随机数的诞生都是一个独立的“个体事件”。以掷骰子来说，每次掷骰子的结果都不会受到之前掷骰结果的影响。无论前一次掷出的是“6”还是“1”，下一次掷骰子时，每个点数（1-6）出现的概率依然是六分之一。在统计学的抽样调查中，为了保证样本的随机性和代表性，需要利用随机数来抽取样本。从总体中抽取的每个样本都是依据独立生成的随机数确定的，这样就避免了因样本间的相互影响而导致的偏差，确保了调查结果的可靠性。

无偏差性：理论上，足够优秀的随机数是没有偏差的，这就表示在大量的样本数据中，随机数在取值范围内的每个数值出现的概率基本相同，不会偏向特定值。例如，商店做活动时经常放置一个大转盘，而转盘会被等分为多个涂有不同颜色的区域。顾客转动指针或者转盘，当转动完全停止时，指针指向每块区域的概率应该是相同的；此外，利用计算机生成随机数时，如果程序生成 0 到 9 之间的随机数，理论上经过很多次的重复生成后，输出每个数的次数几乎相差无几。

均匀分布性：若随机数在给定的区间内生成，那么它们在该区间内的分布是均匀的。这意味着在该区间的任何一个子区间内，随机数出现的概率与子区间的长度成正比。例如，在区间[1,100]内生成随机数，当生成的随机数数量足够大时，落在[1,50]和[51,100]这两个长度相等的子区间内的随机数数量大致相同。在计算机图形学中，为了模拟自然场景中的随机分布现象，如草地上草叶的分布、星空背景中星星的分布等，就需要利用具有均匀分布性的随机数来确定每个元素的位置，从而营造出逼真的自然效果。

不可重现性：所谓真随机数必然不能重现；简单地说，真随机数发生器生成一次随机数的过程不可能再完全重复地经历一遍，因为随机的定义就意味着它是不可再现的。与之相反的是伪随机，则是可以通过固定的算法和种子值导出。假如两次用同一颗种子与同一个算法，那么导出的就是完全一样，完全可以重现，并不存在不可重现这一特性。密码、安全认证等场景绝对不允许有可重现的情况发生。比如说在一次一密通信系统中，我们每次传输使用的都是不相同的密钥，如果不这样的话，很容易使得通讯被复制破解。如果密文都可重现了，那么明文也很容易就可以找出来了。

2.2 随机数及其发生器的分类

2.2.1 伪随机数发成器

伪随机数发生器（Pseudorandom Number Generator, PRNG)，是一种通过确定性算法生成类似于随机数序列的算法或设备，它依赖于一个初始值，即种子，在给定种子的情况下，按照特定的算法生成相应的伪随机数序列，如图 2.1 左边所示。

伪随机数发生器生成的数字序列并非真正意义上的随机数，具有一定的可预测性和可重复性，只要种子和算法确定，生成的随机数序列就是完全确定的。但在很多应用场景中，其生成的序列能够满足一定的随机性需求，表现出类似随机数的统计特征，例如在一定范围内的分布具有均匀性等。常见的伪随机数发生器算法有线性同余发生器（Linear Congruential Generator, LCG）、梅森旋转算法（Mersenne Twister）等，产生随机数的具体过程如下：

（1） 线性同余法

线性同余法是一种简单常用的伪随机数生成算法。这种伪随机算法通过一个递推公式来生成随机数序列，公式如下：

$$X_{n+1} = (aX_n + c)\ mod\ m \qquad (2\text{-}1)$$

其中，X_n是序列中的第n个随机数，a是乘子（乘数），c是增量（加数），m是模数，初始值X_0（通常称为种子）是给定的。利用线性同余法生成随机数简单高效，只需要一次乘法、加法和取模运算就能生成下一个随机数。然而，输出地结果常具有周期较短地缺点，若如果参数选择不当，生成的随机数序列极有可能会出现明显的规律。例如，m较小，或者a、c的值不合适，可能会导致生成的随机数分布很不均匀。

（2） 梅森旋转算法

梅森旋转算法是一种更复杂、高质量的伪随机数生成器。它基于线性反馈移位寄存器的原理，但进行了改进。它的周期非常长，达到了 219937-1，这是一个梅森素数。

其生成过程首先需要初始化一个长度为$n = 624$的数组（状态向量），这个数

组的元素是 32 位整数。通过一个种子来初始化这个数组，种子可以是用户提供的任意整数。然后，通过一系列复杂的位运算和变换来更新这个状态向量。具体来说，它会根据当前状态向量中的元素计算出下一个状态向量，这个过程涉及到矩阵乘法和位旋转等操作。从更新后的状态向量中提取伪随机数。一般是通过对状态向量中的元素进行组合和变换，得到一个 32 位或 64 位的伪随机数。

梅森旋转算法生成的伪随机数具有很高的质量，其周期长，分布均匀。它能够满足大多数需要高质量随机数的应用场景，如模拟实验、密码学中的密钥生成等。缺点是相对复杂，实现起来比线性同余法等简单算法要困难一些，并且需要更多的计算资源和存储空间来维护状态向量。

（3） 基于密码学的伪随机数发生器

这种生成器通常基于密码学算法，如哈希函数或分组密码。例如，使用哈希函数（如 SHA256）来生成伪随机数。可以从一个初始种子（如系统时间、用户输入等）开始，将种子输入到哈希函数中。假设种子是一个字节序列S，通过计算$H(S)$得到一个固定长度的哈希值，这个哈希值可以作为伪随机数。如果需要生成更多的随机数，可以将上一次生成的哈希值作为新的种子再次输入到哈希函数中，即$S_{n+1} = H(S_n)$，以此来生成一系列的伪随机数。该方法安全性高，适合用于需要高度安全的随机数生成场景。计算效率可能相对较低，因为密码学算法通常比较复杂，需要更多的计算资源。而且其生成的随机数可能受到种子质量的影响，如果种子被攻击者知晓或者可预测，可能会影响随机数的安全性。

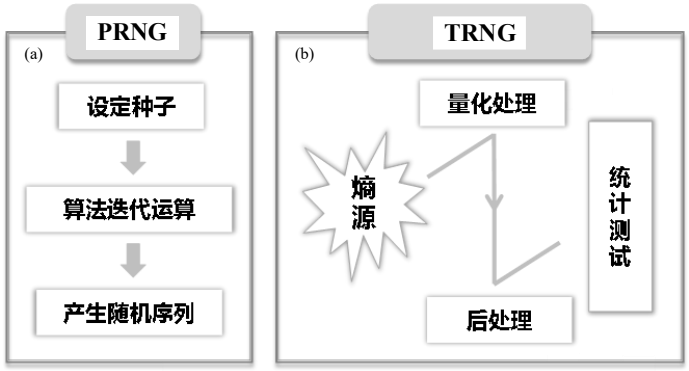
2.2.2 真随机数发成器

在一个典型的 TRNG 结构中，首先要从目标随机源中提取出随机信号，然后通过采样随机信号来生成一个数字序列，最终通过后处理模块来提升序列的质量，如图 2-1 右边所示。目前，熵源主要包括热噪声、核衰变、宇宙辐射以及其他随机物理现象[5]。而热噪声是很多熵源中最广泛使用的，它是由导体中电子的随机运动生成的。

真随机数发生器在数字电路领域有着广泛且关键的应用，其设计方法丰富多样，各具特色。以下几种是数字电路中常见的真随机数发生器设计方式：

一种常见的设计方法是利用低频信号对高频信号进行采样，从而捕获高频信

号中的随机抖动或噪声，生成随机数，如图 2-2 所示。高频信号源通常是一个环



（a）伪随机数发生器实现过程（b）真随机数发生器实现过程

图 2-1 真随机数发生器和伪随机数发生器生成过程

形振荡器、PLL 或其他类型的振荡器，其输出信号具有较高的频率和一定的抖动（Jitter）。低频采样信号通常是一个稳定的低频时钟信号，用于对高频信号进行采样。由于高频信号的抖动，采样结果会具有一定的随机性。可以使用 D 触发器作为采样电路。将高频信号连接到 D 触发器的数据输入端，低频采样信号连接到时钟输入端。D 触发器的输出即为采样结果，是一个随机比特。当低频信号对高频信号进行采样时，由于高频信号的抖动，采样时刻的高频信号状态（高电平或低电平）是随机的。通过多次采样，可以捕获到一系列随机比特。通过这种方法得到的随机数质量较低，通常需要对采样结果进行后处理，消除偏差并提高随机数的均匀性。同时，产生随机数的速度慢，无法满足高速信号传输系统的要求。

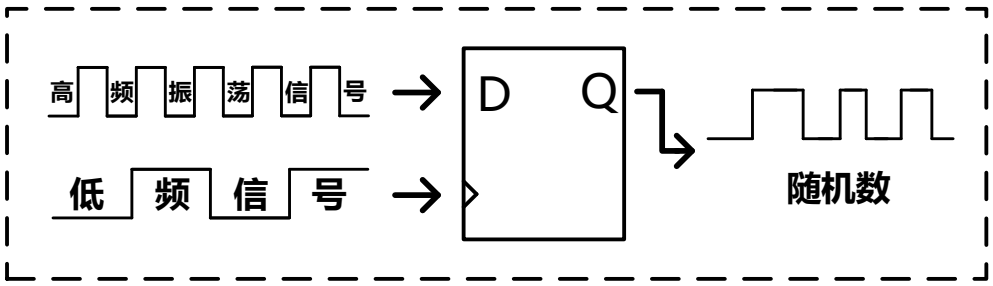
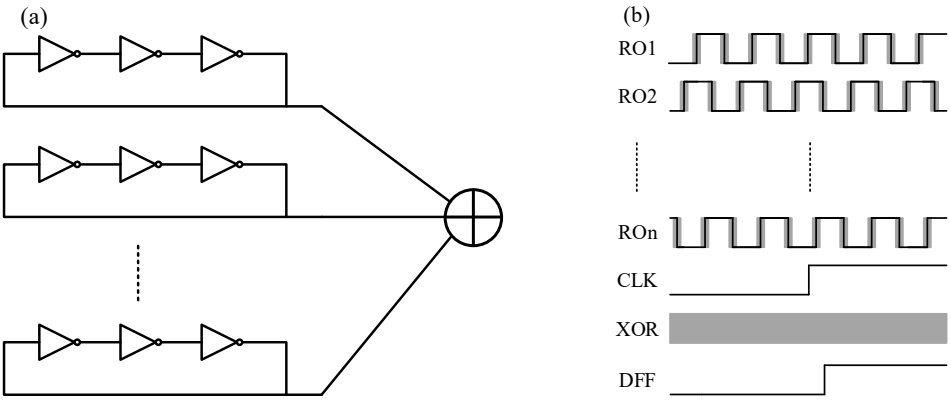


图 2-2 基于振荡器的真随机数发生器

为了解决前面的问题，伍斯特理工学院 Sunar 教授团队让多个振荡器并行工作并通过异或操作生成随机信号，能够显著提高随机数的吞吐量[47]。这种并行化设计使得该 TRNG 能够以较高的速率生成随机数。通过将多个振荡器的输出异或，能够在时域中生成更多的抖动区域，如图 2-3（b）所示，其中，阴影部分

为信号的抖动区域，通过异或处理，可以将所有信号的抖动区域叠加，从而提高抖动填充率，经过足够时间的累积，甚至可以被抖动区域完全填充。填充率的提升意味着更多的采样点具有随机性，增强了随机数生成的质量。相对于前者采用单一的振荡信号作为熵源，避免了低频采样高频信号的设计依赖于单一高频信号的抖动，填充率较低，且需要复杂的校准和优化才能达到较高的随机性。



（a）基于多组环形振荡器的 TRNG 结构图　（b）通过异或叠加抖动区域

图 2-3 通过多组 RO 并行异或改善 TRNG 效果

基于电路中的亚稳态现象生成随机数是一种常用的生成方法。如图2-4所示，通过 RS 锁存器在特定情况下进入一种不稳定的状态（亚稳态），然后通过 XOR 操作来提取随机数。

具体工作原理如下：当 CLK 信号为 0 时，两个与非门的输出都为 1，当 CLK 从 0 变为 1 时，RS 锁存器的内部会形成振荡信号，由于此时的电路为双稳态结构，导致电路无法立即稳定下来，进入一种不确定的中间状态。这种状态最终会稳定为一个值，但输出结果到底为 0 还是 1 是不可预测的，因为它受设备内部噪声、物理扰动以及制造工艺偏差的影响。正是这种不可预测性，为生成真随机数提供了基础。

这种设计不仅简单高效，还非常节省硬件资源，同时功耗也很低，所以在基于 FPGA 的 TRNG 设计中很常见。

然而，基于亚稳态现象的随机数发生器也存在一些问题。由于亚稳态的产生和维持受制于工艺偏差、温度及电压变化等，这可能会导致输出结果的随机性不一致，在不同设备或环境下可能会出现一定的偏差或相关性。其次，因为亚稳现象产生需要一些特定条件而使得其每次产生随机数需花费一段时间，即需要重启

才能继续工作，导致速度比较慢。因此，解决这些问题的方法是在设计时加一定的后处理电路或者其他校正过程（哈希函数等）加以弥补。该种方法对于硬件资源有限制的时候有利，但是也提醒设计者，在应用此方案之前还是应适当对该方法做一些调整才是最佳的做法。

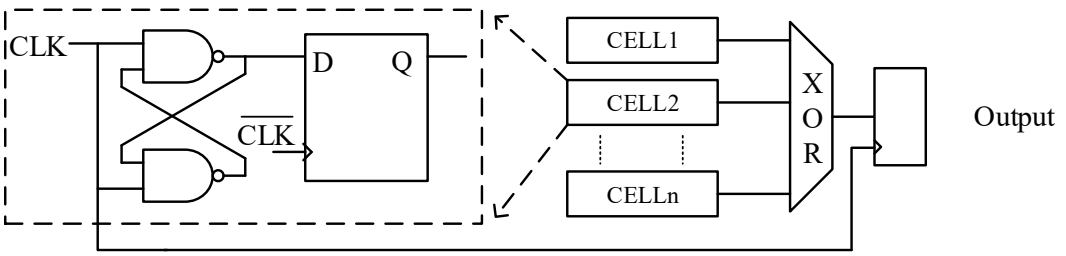


图 2-4 基于亚稳态的真随机数发生器结构

为了解决亚稳态电路偏差高，随机性差，Michal 提出了一种过渡效应环形振荡器（Transition Effect Ring Oscillator，TERO）。该设计来源于双稳态电路在亚稳态事件时产生的振荡轨迹。其核心思想是通过在双稳态电路中加入延迟单元，使电路强制振荡形成振荡呀问题来提取随机性，振荡次数的高度方差由 FPGA 逻辑单元的内部噪声决定。

一个典型的 TERO 类型的 TRNG 通常由与非门，反相器和计数器构成，如图 2-5 所示，通过再两个交叉耦合的与非门之间加入偶数个反相器，可以形成一个振荡环，通过控制 CLK 信号从 0 转换为 1 时，通过 NAND 门将两个边缘（一个上升沿，一个下降沿）注入到偶数级环形振荡器的相反节点。由于偶数级环形振荡器的特性，两个边缘会在环形中传播并最终崩溃从而形成稳态，这种从亚稳态变成稳态的过程被称为振荡亚稳态。崩溃时间取决于边缘的延迟差异和随机抖动。崩溃时间服从逆高斯分布，通过计数器计数崩溃前的循环次数并取其最低有效位（LSB）作为随机数输出。TERO 的优势在于其结构简单，且对 FPGA 内部噪声的敏感性显著高于传统的环形振荡器（RO），同时对全局扰动的敏感性较低，具有更高的随机性。然而，TERO 对工艺变化较为敏感，尤其是在偶数级环形振荡器中，工艺变化可能导致崩溃时间的均值和方差发生显著变化，影响随机性。同时，由于其严重受工艺变化影响，为了该影响，TERO 通常需要额外的调谐机制（如自动调谐环路或动态配置），增加了设计和实现的复杂性。且该结构每输出一个随机数，仍需要对电路进行重启，崩溃时间的分布（逆高斯分布）使得随

机数的生成速度难以精确控制，严重影响了随机数发生器的性能。

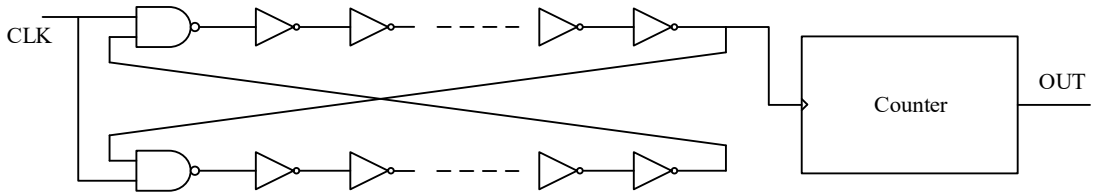


图 2-5 基于边沿追赶的真随机数发生器结构

2.3 随机数的统计度量

2.3.1 香农熵

香农熵主要用于度量随机变量所蕴含的不确定性程度。对于离散型随机变量X，其取值有n种可能$x_1, x_2, \ldots, x_n$，对应概率分别为$p_1, p_2, \ldots, p_n$，香农熵$H(X)$的计算公式为：

$$H(X) = -\sum_{i=1}^{n} p_i log_2 p_i \quad (2\text{-}2)$$

以抛一枚均匀硬币为例，出现正面和反面的概率为$p_1 = p_2 = \frac{1}{2}$，将其代入公式可得：

$$H(X) = -(\frac{1}{2} log_2 \frac{1}{2} + \frac{1}{2} log_2 \frac{1}{2}) \quad (2\text{-}3)$$

比特，这直观地表明每次抛硬币这一随机事件结果的不确定性为 1 比特。$H(X)$越大，则表明混乱程度越高，即随机性越高。香农熵具有非负性，只有当随机变量的取值完全确定（即某一取值概率为 1）时，香农熵才为 0。对于真随机数发生器而言，其输出同抛硬币一样，只有两种结果（0 或 1），故其香农熵最大值为 1，此时真随机数发生器的输出表现最好；为 0 时，则完全没有随机性。

2.3.2 最小熵

最小熵是在最坏情况下，来衡量随机变量的不确定性。是评估随机数的重要参数，对于离散型随机变量X，最小熵$H\infty(X)$的计算公式是：

$$H_\infty(X) = -log_2(max p_i) \quad (2\text{-}4)$$

假设有一个独立同分布的随机变量，它的输出结果有一定的偏差，输出 1 的概率为0.9，输出1的概率为0.1，那么它的最小熵为$H_\infty(X) = -log_2(0.9) \approx 0.15$

比特。在密码学领域，最小熵是衡量密钥的重要标准，如果设置的密钥最小熵很低，则攻击者可以通过一定次数的尝试猜出密钥，给安全系统造成极大的风险，所以，最小熵可以帮助人们更加有效地设计保护措施，降低信息泄露地风险

2.4 本章小结

本章对随机数基本原理、随机数发生器以及随机数发生器的统计度量方法进行了梳理，为随机数发生器特别是真随机数发生器设计提供理论基础。按照发生原理，随机数可以分成伪随机数（随机性取决于一个确定的算法）和真随机数（随机性取决于物理上的不确定性），并且具有不可预测性、不相关性、无偏差性等特征，广泛应用于密码领域、统计领域和模拟实验等；以此随机数发生器则可以分为伪随机数发生器和真随机数发生器，其中 PRNG 适合应用在需要重复使用的情形，TRNG 一般用于安全性较高的应用，香农熵和最小熵则用于度量随机数的质量，并作为评价随机性的重要参数。这一部分内容为随机数发生器的进一步研究提供了理论基础，对于后续真随机数发生器设计研究的开展也具有积极意义。

第三章 基于新型采样链精度量化抖动的 TRNG

3.1 基于 RO 的随机数发生器

3.1.1RO 的随机性来源

时钟抖动是电路设计中的一种常见现象，指的是时钟信号在时间轴上的不确定性和偏差。具体来说，时钟抖动描述了时钟信号的边沿（上升沿或下降沿）相对于理想时间位置的微小偏移。这种偏移可能是随机的（如由噪声引起的随机抖动）或确定的（如由电源干扰引起的周期性抖动）。下文将会以环形振荡器为例来对抖动进行熵源分析。

如图 3-1 所示，环形振荡器由奇数个反相器首尾连接而成，电路结构非常简单。信号在环路中传播过程中能够产生周期性振荡信号。其工作原理基于电路中的延迟反馈：信号经过每个反相器中会产生时间延迟，奇数个反相器的串联使得信号在环路中不断反转形成振荡从而输出一个周期性波形。输出波形通常近似于方波，但可能存在较大的边沿抖动，如图 3-2 所示，当信号在环路中不断传播时，信号中的抖动会随之累积。本文将对抖动累积过程以及采样时的输出结果可能进行分析。

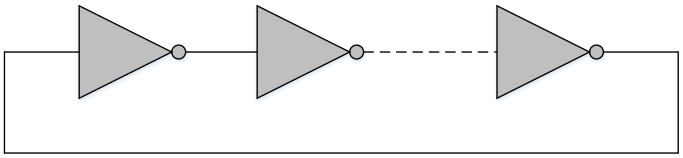


图 3-1 奇数个反相器首尾相接形成环形振荡器

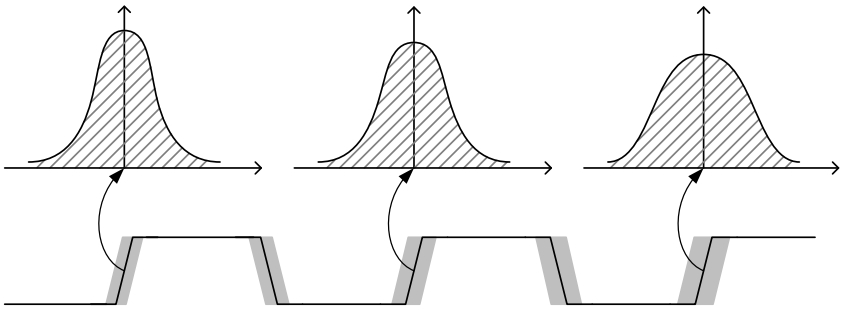


图 3-2 抖动在电路中不断累积

基于环形振荡器的 TRNG，其随机性来自于两个采样间隔之间的 RO 输出的信号中积累的时序抖动。生成的随机比特流主要取决于抖动累积的方式。在[66]

中，建立了 RO 的相位噪声与其阶数之间关系的表达式：

$$L(f)_{min} = \frac{8}{3\eta} N \frac{KT}{p} \left(\frac{V_{DD}}{V} + \frac{V_{DD}}{IR}\right)\left(\frac{f_0}{f}\right)^2 \tag{3-1}$$

其中，N是 RO 的阶数；K是玻尔兹曼常数；T是绝对温度；η、VDD、V、R和I是常数；f_0是 RO 的频率；f是偏移频率；p是 RO 的功耗。当频率不固定时，增加N会降低频率f_0，但是一个周期内可以产生更多噪声。频域相位噪声与时域平均抖动之间的关系如下。

$$\sigma_{RMS} = \frac{\Delta\phi}{2\pi f_0}, \Delta\phi = \sqrt{2\int_{f_1}^{f_2} L(f)\, df} \tag{3-2}$$

对于基于 RO 的 TRNG 中抖动积累的现象，特别是随机抖动分量和确定性抖动分量对总体累积抖动的影响，还需要进行详细地分析。鲁汶大学 Viktor Fischer 教授团队对[61]报道的抖动产生和积累模型进行了改进[62]。具体分析过程见 3.1.2 节。

3.1.2RO 电路的熵源分析

环振荡器是由奇数个反相器组成的自由运行振荡器（见图 3-1)。每个反相器在两个连续的半周期内，会传播时钟信号的上升沿和下降沿，将其一般化后如图 3-3 所示，只包含一个与非门和任意数量的延迟单元（门）。

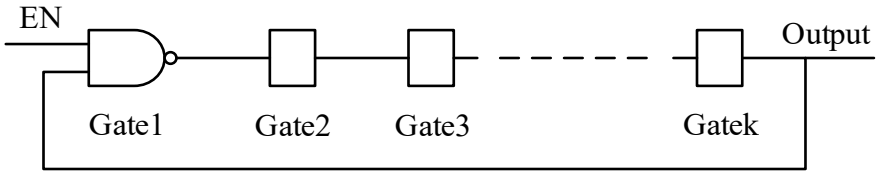


图 3-3 与非门和延迟单元构成的振荡器

RO 可使能或停止时钟信号的产生，以减少消耗，也可以使随机数发生器与其他的事件进行同步。图 3-3 中环形振荡器产生的时钟信号的半周期H_j被设为：

$$H_j = \sum_{i=1}^{k} d_{i,j} \tag{3-3}$$

其中，$k \in \{3,4,5, \ldots, n\}$，是延迟单元（包括与非门）的数量，$d_{i,j}$是半周期$j$中的第$i$个延迟单元的延迟，它被定义为：

$$d_{i,j} = D_i + \Delta d_{i,j} = D_i + \Delta d_{Li,j} + \Delta d_{Gi,j} \tag{3-4}$$

其中Di是第i个门的固定延迟（包括第i个门和$(i+1)$个门之间的连接延迟），

$\Delta d_{i,j}$是由单个局部源（$\Delta d_{Li,j}$）和通用全局源（$\Delta d_{Gi,j}$）引起的延迟变化。式(3-2)中第i个门在半周期j内的局部抖动可以表示为：

$$\Delta d_{Li,j} = \Delta d_{LGi,j} + \Delta d_{LDi,j} \tag{3-5}$$

其中，$\Delta d_{LGi,j}$是来自局部源的独立高斯抖动，而$\Delta d_{LDi,j}$是局部确定性抖动。式(3-2)中的全局抖动来自所有的门的共同抖动源，可以表示为：

$$\Delta d_{Gi,j} = K_i(\Delta D + \Delta d_{GGj} + \Delta d_{GDj}) \tag{3-6}$$

其中，ΔD表示由于温度或电源偏差引起的缓慢延迟变化，Δd_{GGj}来自电源的高斯噪声，Δd_{GDj}表示由电源快速变化引起的全局确定性抖动。K_i是全局抖动和门i抖动的比例系数。

为了简化模型，丢弃了不应该或没有促进随机数生成的局部和全局抖动源。式（3-3）中的局部确定性抖动$\Delta d_{LDi,j}$不被考虑在内，因为它是相对较小且无法控制。式（3-4）的缓慢变化的ΔD也被忽略，因为其并不直接参与随机数的生成。此外，在等式（3-4）中的全局高斯抖动Δd_{GGj}也不被使用，因为它与全局确定性抖动难以区分。因此，在作者所提出的简化模型里，使用的器件中第i个门的延迟变化可以表示为：

$$\Delta d_{i,j} = \Delta d_{LGi,j} + K_i \Delta d_{GDj} \tag{3-7}$$

基于环振荡器的 TRNG 使用 1 到 N 个（几乎相同的）环形振荡器，来产生 N 个带有抖动的快速时钟信号。然后，它们的异或输出被一个慢的参考时钟在 D 触发器中进行采样，以得到所谓的数字噪声信号[64]，所得到的信号的熵可以在后处理模块中得到增强。在环形振荡器中，抖动最重要的一个方面是它在时间上的积累。对环形振荡器抖动积累的分析是 TRNG 中提取仿真和评估的主要目标。通过公式 3-1 和公式 3-2 可以将环形振荡器的第j个半周期表示为：

$$H_j = \sum_{i=1}^{k} D_i + \Delta H_{LGaccj} + \Delta H_{GDaccj} \tag{3-8}$$

其中：

$$\Delta H_{LGaccj} = \sum_{i=1}^{k} \Delta d_{LGi,j} \tag{3-9}$$

它是局部高斯抖动的累积，服从正态分布$N(0, \sigma_{acc})$，假设单个门的高斯抖

动是独立的，并服从正态分布$N(0,\sigma)$，则$\sigma_{acc}=\sigma\sqrt{k}$。

此外：

$$\Delta H_{GDaccj}=\sum_{i=1}^{k}\Delta d_{GDi,j}=\sum_{i=1}^{k}K_i\,\Delta d_{GDj} \tag{3-10}$$

是累积的全局确定性抖动。在l个时钟周期内，RO 中累积的总抖动可以表示为：

$$\Delta T_{tot}=\sum_{j=1}^{2l}\Delta H_{LGaccj}+\sum_{j=1}^{2l}\Delta H_{GDaccj} \tag{3-11}$$

其中：

$$\sum_{j=1}^{2l}\Delta H_{GDaccj}=\sum_{j=1}^{2l}\sum_{i=1}^{k}K_i\,\Delta d_{GDj} \tag{3-12}$$

是k个门在l个周期（T）内累积的全局确定性抖动。如果振荡器中所有门的高斯抖动具有相同的统计参数（这在真实设备中通常是正确的），此外，若每个半周期的高斯抖动是独立的，那么累积的抖动将有标准差：

$$\sigma_{tot}=\sqrt{\sum_{j=1}^{2l}{\sigma_{acc}}^2}=\sqrt{2kl}\sigma \tag{3-13}$$

如公式（3-10）和（3-11）所示，在l个时钟周期内，RO 中累积的总抖动是一个服从正态分布的随机变量，其均值为$(\sum_{i=1}^{k}K_i)(\sum_{j=1}^{2l}\Delta d_{GDj})$，标准差为$\sqrt{2kl}\sigma$。然后，假设$\Delta d_{GDj}$为常数，则确定性抖动随$l$线性累积，而高斯抖动累积为$\sqrt{l}$。

中国科学院数据保障与通信安全研究中心的马原团队在[64]中介绍了具体的累积过程，以基于振荡器的 TRNG 的典型示例（见图 2-2）。一个稳定的慢时钟信号采样一个不稳定的快速振荡器信号来产生随机位。随着采样间隔的增加，快速振荡器信号的抖动也在累积。产生随机比特位的基础是在单个慢信号周期持续时间内，快速信号周期（半周期）数量的不可预测性。

基于振荡器的 TRNG 中的重要变量如图 3-4 所示，其中X_k是两次反转之间的时间间隔。假设X_k是独立同分布的，其原因将在第 3.4 节中进行讨论。而半周期的均值和方差分别记为μ和σ^2，即$\mu=E(X_k)$，$\sigma^2=Var(X_k)$。采样时间具有相同间隔 s，分别表示为s_0、s_1、…，s_i，即$s_i=is$。等待时间W_i表示s_i到下面最近的采样边沿的时间距离。在$(s_{i-1},s_i]$之间的边沿数是由R_i决定，第 i 个采样比特$B_i=(B_{i-1}+R_i)mod2$。

值得注意的是，在B_{i-1}中加入R_i可以视作一种后处理，但该操作对信息熵没

有影响，无法提供随机性，因此为了方便，取其余的$B_i = R_i mod 2$。

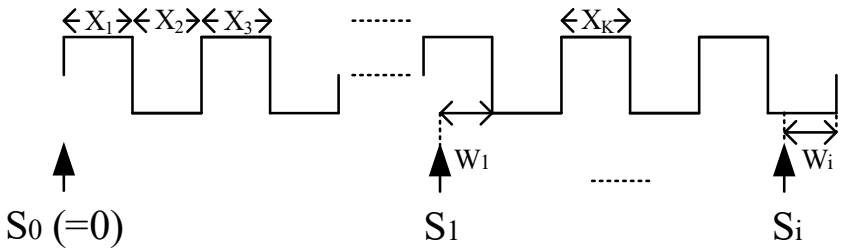


图 3-4 振荡信号的采样过程

根据上述内容，总结文献[65]，可以得到重要结果。令$R_i = \min\{k|T_k > s\}$，其中，$T_k = X_1 + X_2 + \cdots + X_k$，这意味着在$k$增加的过程中，$R_i$对应的是第一个$k$，使得$T_k$大于$s$。此时：

1

$$Prob(R_i = k+1) = Prob(T_k \le s) - Prob(T_{k+1} \le s) \tag{3-14}$$

由中心极限定理可以推导出 Tk 的分布：

$$Prob\left(\frac{T_k - k\mu}{\sigma\sqrt{k}} \le x\right) \to \Phi(x), k \to \infty \tag{3-15}$$

其中，$\Phi(x) = \int_{-\infty}^{x} e^{-t^2/2} dt/\sqrt{2\pi}$为标准正态分布$N(0,1)$的累积分布函数。然后可以得到：

$$\begin{aligned} Prob(R_i = k+1) &= Prob(T_k \le s) - Prob(T_{k+1} \le s) \\ &\approx \Phi\left((v-k)\cdot\frac{\mu}{\sigma\sqrt{k}}\right) - \Phi\left((v-k-1) - \frac{\mu}{\sigma\sqrt{k+1}}\right) \end{aligned} \tag{3-16}$$

其中，$v = s/\mu$表示频率比。则采样的比特B_i的概率分布为:

$$Prob(B_i = b_i) = Prob(R_i mod 2 = b_i) = \sum_{j=1}^{\infty} Prob(R_i = 2j - b_i), b_i \in \{0,1\} \tag{3-17}$$

在基于振荡器的 TRNG，特别是基于 RO 的 TRNG 中，抖动在信号中所占比例非常小，即$\sigma/\mu \ll 1$。k的可能值被限制在均值v附近的一个小区间内。此外，由于快振荡器信号一般比慢时钟的频率高几十倍，所以v的值较大。假设$\sqrt{k} \approx \sqrt{k+1} \approx \sqrt{v}$。令$q = \sigma\sqrt{v}/\mu$作为质量因子，用于评价 TRNG 的质量，可以得到：

$$\begin{aligned} Prob(R_i = k+1) &\approx \Phi\left((v-k)\cdot\frac{\mu}{\sigma\sqrt{k}}\right) - \Phi\left((v-k-1)\cdot\frac{\mu}{\sigma\sqrt{k+1}}\right) \\ &\approx \Phi\left(\frac{v-k}{q}\right) - \Phi\left(\frac{v-k-1}{q}\right) \end{aligned} \tag{3-18}$$

对于$B_i = 1$的情况，可以得出：

$$Prob(B_i = 1) = \sum_{j=1}^{\infty} Prob(R_i = 2j - 1) \approx \sum_{j=1}^{\infty} \Phi\left(\frac{v-2j}{q}\right) - \Phi\left(\frac{v-2j-1}{q}\right) \quad (3\text{-}19)$$

这可以描述为图 3-5 中正态分布曲线下面 0 和 1 的交错列的面积之和。

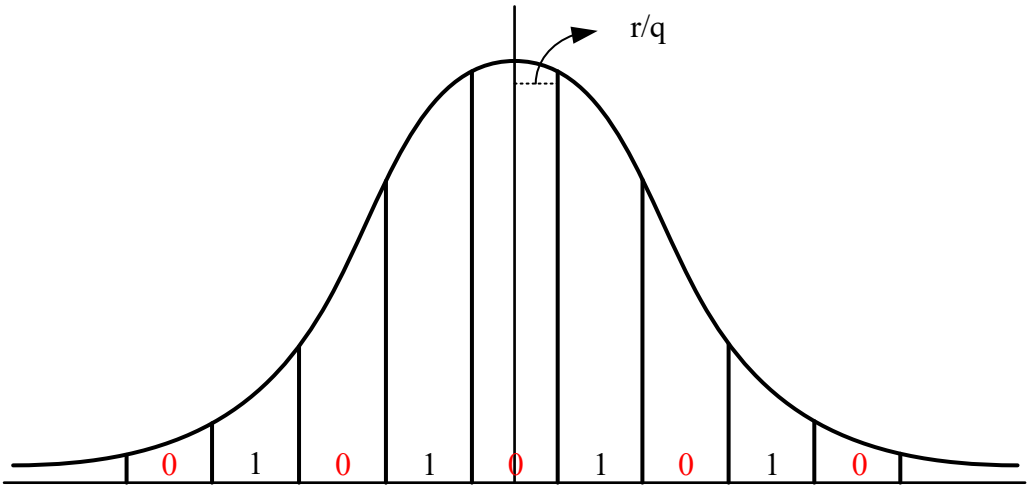


图 3-5 采样比特的正太分布

为便于表示，W_i被设置为 0。正态分布曲线与 x 轴之间的面积（等于 1）以$1/q$间隔进行划分，每列的面积对应的R_i的概率等于每个k的概率。q越大，列被划分得越细，这意味着的面积“0”和“1”更接近。另一个影响因素是，除了q之外，r的值也会影响采样位的偏差。变量r是v的分数部分，即$r = vmod1$。其位置由r/q确定，如图 3-5 所示。显然，当$W_i = 0$和$r = 0$时，概率“0”和“1”的面积相等，而与q无关。当$W_i = 0$时，最无偏的情况是$r = 0.5$，其中概率“0”和“1”之间的距离具有相同q的时候，与其他的情况相比，偏差是最小的。因此，即使在最坏（最无偏）的情况下，一个稳定的 TRNG 设计也应该有足够的熵。

3.2 基于异或门链的真随机数发生器结构

本结构在基于 RO 的 TRNG 架构上提出了一种新的采样方式，如图 3-6 所示。它由两个同阶的 RO、异或门链和一个 TFF 构成。不使用 DFF 直接采样 RO 输出，将两个 RO 的相同节点进行异或，产生的信号边沿差值形成一个脉冲，如图 3-7 所示。再将所有异或门输出进行异或，可以得到一个高频脉冲 P（信号每经过一个反相器就产生一次脉冲），产生的脉冲信号用于触发 TFF。由于固有延迟差的影响，产生的脉冲信号会周期性变化。

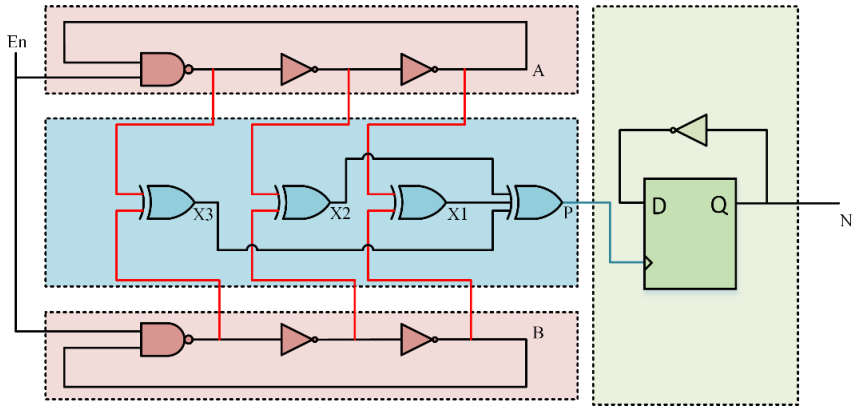


图 3-6 提出 TRNG 的基本组成部分

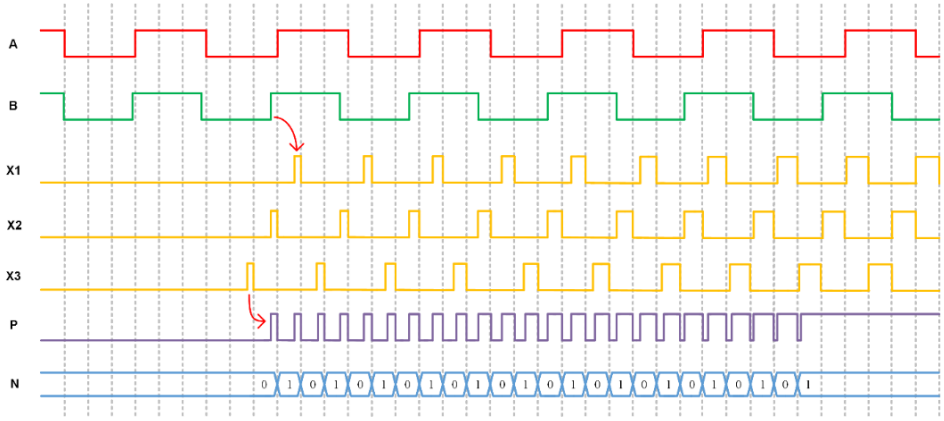


图 3-7 异或门链产生高频脉冲

在实际逻辑器件中，逻辑门无法对输入信号做瞬时响应，当输入信号维持时间较短时，由于输入脉冲太窄，器件响应特性慢，不能响应如此快的脉冲，产生了“短脉冲抑制效应”，逻辑门将不能完全响应，无法产生对应的输出波形[21]。故 RO 的边沿差只有在一定范围内才能产生有效脉冲，设能使 TFF 计数的最小边沿差值为T_{pb}，如图 3-8 所示，当两个信号的边沿差小于T_{pb}时，无法产生有效脉冲信号。

因此，在频域可以将其划分为无效区域（阴影部分），无法进行计数；以及有效区域（空白部分），可以进行计数，如图 3-9 所示。随机性来源于有效区域的过渡时间。

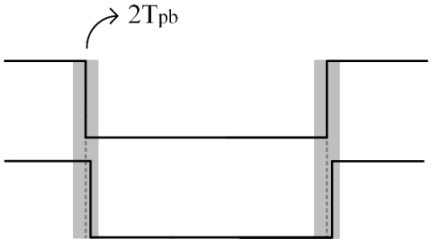


图 3-8 短脉冲抑制区域

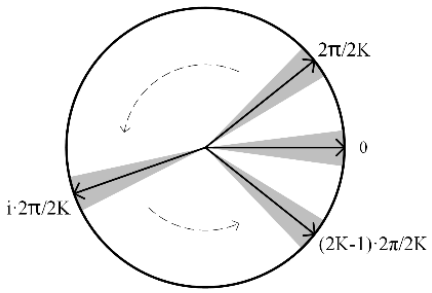


图 3-9 两个 RO 的相位差周期性变化

电路的具体运行过程如下：

（1）当 EN 信号为高电平时，两个 RO 同时开始振荡，开始时其信号边沿差为零，所有异或门的输出为 0，TFF 不进行计数；

（2）随着时间增加，边沿差所形成的脉冲宽度会呈周期性地逐渐变大，到达图 6 中的空白区域，可以产生有效脉冲，TFF 进行计数；

（3）当两个 RO 边沿差为一个逻辑门的延迟左右时，到达图 3-9 中的阴影区域，此时脉冲信号不再振荡，TFF 停止计数。

3.3 熵源的随机性分析

由于工艺偏差和不对称的布局布线等原因导致路径延迟不同，两个同阶 RO 的振荡周期会存在细微差异，其振荡波形宽度可以用半周期差与抖动之和来表示。一个K阶 RO 的第j个半周期：

$$T_j = t + \Delta T_j \tag{3-20}$$

其中，t表示半周期的均值，ΔT表示半周期的抖动，服从高斯分布$N(0,\sigma^2)$。

$$t = \sum_{i=1}^{k} D_i \tag{3-21}$$

其中，D_i是第i个反相器（与非门）的延迟和线延迟之和。对于两个同阶 RO（RO1 和 RO2），其边沿差一开始几乎为 0，设 RO2 周期更长，经历了l个半周期所经历的时间分别为：

$$T_{1j} = \sum_{j=1}^{l}\left(t_1 + \Delta T_{1j}\right) \tag{3-22}$$

$$T_{2j} = \sum_{j=1}^{l}\left(t_2 + \Delta T_{2j}\right) \tag{3-23}$$

其中，T_{1j}和T_{2j}分别是 RO1 和 RO2 的第j个半周期。由于T_{1j}和T_{2j}时间相近，

设该过程中产生的抖动ΔT_{1j}和ΔT_{2j}是分布相同的随机变量，即$\Delta T_{1j}\sim N(0,\sigma^2)$且$\Delta T_{2j}\sim N(0,\sigma^2)$。因此，两个 RO 相同节点异或产生的边沿差为：

$$\begin{aligned}\Delta\phi &= T_{2j}-T_{1j}\\ &= \sum_{j=1}^{l}(t_2-t_1)+\sum_{j=1}^{l}\left(\Delta T_{2j}-\Delta T_{1j}\right)\end{aligned} \tag{3-24}$$

由于t_2-t_1是固定值，因此，$\Delta\Phi$服从高斯分布，即$\Delta\Phi\sim N(l(t_2-t_1),2l\sigma^2)$。

据前文所述，由于逻辑门存在短脉冲抑制效应，TFF 计数的最小边沿差值为T_{pb}，则边沿差在无效区域应满足：

$$\sum_{i=1}^{m}D_i+T_{pb}<\Delta\phi<\sum_{i=1}^{m+1}D_i-T_{pb} \tag{3-25}$$

边沿差在有效区域应满足：

$$\sum_{i=1}^{m}D_i-T_{pb}<\Delta\phi<\sum_{i=1}^{m}D_i+T_{pb} \tag{3-26}$$

由于振荡过程存在抖动，所以，边沿差在有效区域内，所产生的有效脉冲数存在一定的偏差，可以带来随机性。设开始计数时，输出为 0，一直到进入无效区域，所需要的半周期数为X。由于不同门之间延迟的差异，第一个异或门到达无效区域后，后面的异或门仍可能提供有效脉冲，但是对结果影响很小，不进行计算。由于$\Delta\Phi$服从高斯分布，故边沿差$\Delta\Phi$在有效区域内，所经历的时间应服从逆高斯分布[7]，则在此期间 RO 所经历的半周期数X满足$X\sim IG(\mu,\lambda)$，其中：

$$\mu=\frac{D_i-2T_{pb}}{t_2-t_1},\lambda=\left(\frac{D_i-2T_{pb}}{\sigma}\right)^2 \tag{3-27}$$

本结构所使用的 RO 为 3 阶，故在半周期内可以产生 3 个脉冲，所以 TFF 的输出在一个半周期内会反转三次，假设刚进入有效区域时，TFF 的输出为 0，所以，X为奇数时输出结果为 1，X为偶数时输出结果为 0。即：

$$\begin{aligned}&P(Y_{out}=1)\\ &=\sum_{j=0}^{\infty}\int_{2j}^{2j+1}\left(\frac{\lambda}{2\pi x^3}\right)^{\frac{1}{2}}\exp\left(-\frac{\lambda}{2\mu^2x}(x-\mu)^2\right)dx\end{aligned} \tag{3-28}$$

$$\begin{aligned}&P(Y_{out}=0)\\ &=\sum_{j=0}^{\infty}\int_{2j+1}^{2j+2}\left(\frac{\lambda}{2\pi x^3}\right)^{\frac{1}{2}}\exp\left(-\frac{\lambda}{2\mu^2x}(x-\mu)^2\right)dx\end{aligned} \tag{3-29}$$

所生成的随机数香农熵为：

$$H=-\left(p\log_2(p)+(1-p)\log_2(1-p)\right) \tag{3-30}$$

其中，p是 0 或 1 的概率，对于一个拥有良好性能的 TRNG，其p的值应该接近 0.5，H值应该接近 1。

值得注意的是，本设计所利用的另一种熵源是亚稳态，这是在采样过程中（TFF 计数）发生的。在 1.2 节中，本文对亚稳态进行了简要描述，即亚稳态是由触发器的时序违规造成的。Altera 公司对亚稳态现象进行了具体的描述和评估方法[52]。具体过程如图 3-10 所示。

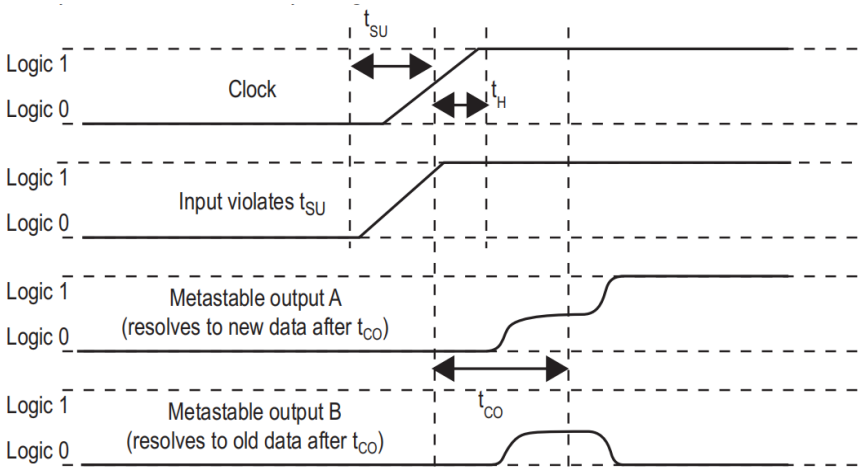


图 3-10 亚稳态的输出

其中，t_{SU}（建立时间）表示在时钟上升沿到来之前数据需要保持稳定的最小时间，t_H（保持时间）表示时钟上升沿到来后数据稳定的最小时间，t_{CO}表示从时钟上升沿到达至可以读取寄存器输出的时间（可以读取稳定输出）。

图中显示了亚稳信号：当时钟信号转换时，输入信号从低状态转换到高状态，违反了寄存器的t_{SU}（建立）最小时间要求。数据输出信号有两种可能结果，在图中的示例中，可能的输出信号从低状态开始，进入亚稳态，徘徊在高、低状态之间。信号输出 A 描述最终输出逻辑 1 状态，输出 B 则代表最终回到原逻辑 0 状态。在这两种情况下，输出转换到稳态（“0”或“1”）时间超过了寄存器的稳定时间t_{CO}。

韩晓泉团队进行类似的描述[67]，当触发器违反了建立和保持时间时，触发器可能会进入三种不同的状态：它能够识别输入信号，因此给出一个新的值；无法识别输入信号，保持旧值；不能识别输入信号，触发器进入亚稳态。当触发器进入亚稳态时，无法准确地知道输出何时会稳定到一个新值。这个时间可能是一

毫秒、一秒、一分钟甚至一年。所以亚稳可以定义触发器的输出不能在某一时刻达到稳定状态。

如图 3-11 所示，t_{CO}的时间长短受到了t_{SU}和t_H的影响，当触发器信号时间满足t_{SU}和t_H时，t_{CO}为触发器的典型输出（通常会在数据手册表中给出）；当触发时间不满足t_{SU}或t_H时，即建立时间或保持时间过短，则t_{CO}就会变长。随着过渡时间的变短，当t_{CO}大于一个触发器的最大值时，触发器进入亚稳态，此时的建立时间和保持时间的范围（小于最小时间要求）称为亚稳态窗（W）。当输入信号边缘靠近时钟边缘时，t_{CO}变得越来越长其大小与t_{SU}和t_H参数之间呈指数关系。这意味着信号在到达亚稳态窗之前，不会发生亚稳态，反之会发生亚稳态。

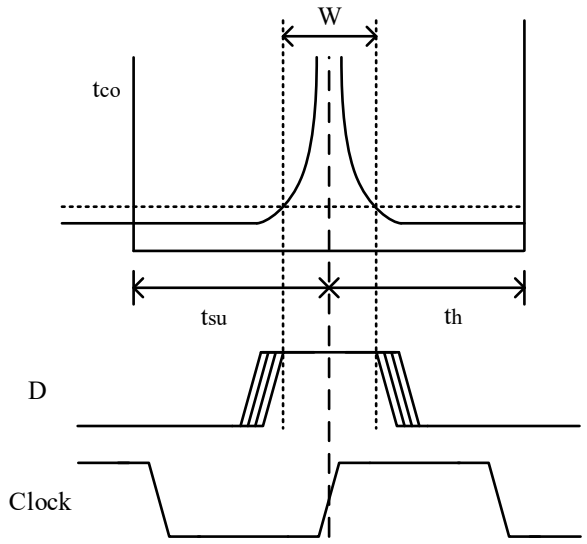


图 3-11 亚稳态窗

尽管在数字系统中避免亚稳态的发生是设计者极力避免的情况，但是合理地利用亚稳态可以使 TRNG 设计更加优秀。相比传统利用亚稳态的电路通过增加大量电路消耗来提升亚稳态的发生概率，本设计中触发器的采样时钟（见图 3-6 和 3-7）输入的信号会缓慢的周期性变化，这个过程可以提升信号到达亚稳态窗的概率，从而丰富本设计的熵源，增加随机性。

3.4 电路整体结构

所提出的真随机数发生器的整体架构如图 3-10 所示，由三个 CELL（如图 3 所示）组成，经过采样和异或处理后生成随机比特。由于两个 RO 之间的边沿差受固有延迟差的影响会周期性变化，因此无需对电路进行重启。通过异或门链对

RO 的延迟差进行量化，随后利用 D 触发器进行采样。由于采样过程中存在的短脉冲抑制效应，使得电路能够精确捕捉到延迟差的细微变化，从而提升了随机性生成的精度和效率。

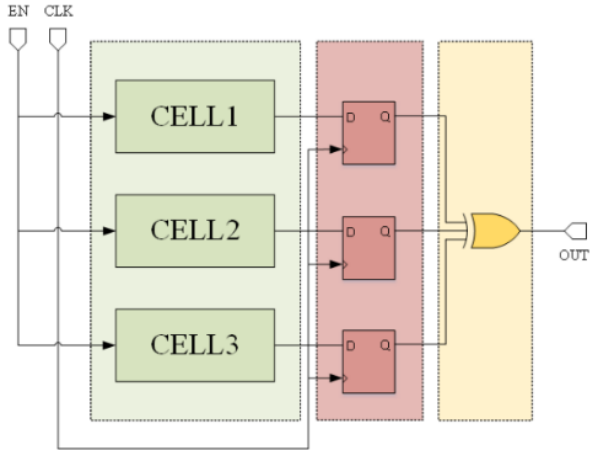


图 3-12 提出 TRNG 的整体结构

3.5 本章小结

本章围绕熵源展开分析和建模，并据此设计出高质量真随机数发生器电路。为此，本章围绕由 RO 提供的抖动时钟作为物理噪声随机源的 TRNG 进行讨论，通过 RO 熵源模型建立抖动在 RO 中的积累机制，分析抖动对随机数生成的影响，探讨相位噪声与 RO 阶数的建模，研究局部高斯抖动和全局确定性抖动的对随机数产生过程的影响；围绕时钟抖动累积设计具有较好随机性的 TRNG，利用异或门链和 T 型触发器（TFF）提取电路信号变化增强随机性，利用短脉冲抑制效应和由亚稳态产生的不确定性相结合来产生高质量随机数；对 RO 半周期差和抖动进行建模，并以此来证明 TRNG 产生的随机性。

第四章 真随机数发生器的 FPGA 实现与随机性分析

4.1 实验平台

4.1.1 硬件开发平台

FPGA 是一种能够灵活、动态调整的半导体器件，多用于数字电路设计、信号处理、嵌入式、通讯、人工智能等数字领域。FPGA 的基本结构是可编程逻辑单元（称为 CLB）、互连线（称为 Interconnect）、I/O 块（称为 IOB）、存储块（称为 Block RAM）、专用的硬件模块（如 DSP 和时钟管理单元）。FPGA 的优势为高度可编程性、并行计算能力、低延迟和高灵活度。能够用于通信、图像处理、人工智能加速等领域。

Virtex-7XC7VX785T 是 Xilinx 公司基于 28 纳米的 FPGA 平台器件，包含 78.5 万个逻辑单元、广泛的存储容量和 DSP 资源，支持复杂数字设计和信号处理；具有丰富的 I/O 资源、灵活的时钟及 FMC 扩展，适于高带宽、复杂的系统。

Virtex-7 系列面向高性能计算和高速通讯，如数据中心和 5G；Artix 系列面向低成本、低功耗、中等规模的应用，如工业控制和消费电子。在 FPGA 上实现电路设计验证能够保障电路设计的可移植、可兼容、可适应，提升电路性能和资源利用率，并测试电路的可靠和稳定。

本设计应用配套的 Xilinx Vivado2018.3 开发软件，可以开发 Verilog 和 VHDL，它提供的模块化开发架构大大缩短了开发周期。而且 Vivado2018.3 拥有一个集成的综合、布线工具、时序/功率分析以及交互式分析软件。可以为高性能的设计提供很大的帮助。

FPGA 的设计主要包括需求分析及设计规划，然后依据硬件描述语言实现逻辑设计、功能仿真，进行综合及实现，生成逻辑网表并经过布局布线，时序分析与优化，进而输出比特流文件并下载到 FPGA 中，最后对 FPGA 硬件验证。

FPGA 高度灵活的优势允许它快速应对各种应用需求，并且通过动态重构来实现实现多种功能的目的，在高效处理大量数据及高性能计算方面具有并行计算能力和超低延时的优势，成为现目前高性能设计的重要核心部件。

综上所述，FPGA 的架构和性能十分独特，在数字电路设计、信号处理和通信以及人工智能设计等方面都有广阔的应用前景，配合着优秀的开发工具与验证手段的使用，使用者可以最大限度地利用 FPGA。

FPGA 作为一种强大的硬件平台，为数字电路设计和高性能计算提供了极大的灵活性和效率。本文实验所采用的 FPGA 有两款 Xilinx Artix-7 和 Virtex-7，如图 4-1 和 4-2 所示。这两款 FPGA 芯片在硬件资源、性能、功耗、和成本上有所差异。

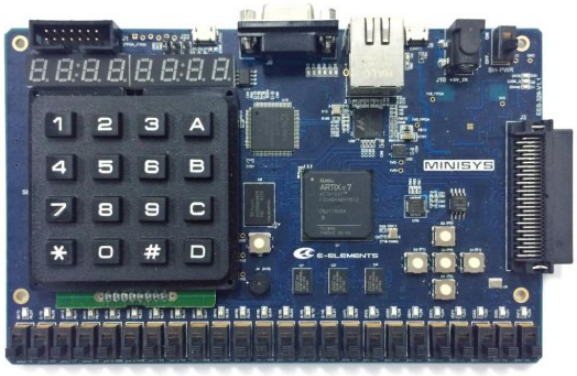


图 4-1 Xilinx Artix-7 FPGA 开发板

Xilinx Artix-7 XC7A100T 是一款功能强大的 FPGA，拥有 15850 个逻辑 Slice，每个 Slice 包含 4 个 6 输入查找表（LUT）和 8 个触发器，其中近 12.5%的查找表可配置为 64-bit 分布式 RAM 或 32 位 SRL（或两个 16 位 SRL16），使得综合工具能够充分利用这些逻辑和存储资源。此外，片内集成了 135 个 36Kbit 的 Block RAM，每个均可作为两个独立的 18Kbit Block RAM 使用，通过 Vivado 的 IP 集成器，可以灵活配置为单端口、双端口等多种类型的 RAM。XC7A100T 还配备了 240 个 DSP48E1 数字信号处理单元，每个 DSP48E1 包含一个预加器、一个 25x18 乘法器、一个加法器以及一个累加器，支持高性能信号处理任务。时钟管理方面，该器件拥有 6 个时钟管理模块（CMT），每个模块包含 1 个混合式时钟管理器（MMCM）和 1 个锁相环（PLL），能够生成宽频率范围的时钟信号并有效滤除抖动。内部时钟频率最高可达 450MBps，例如在 Minisys 实验板上通常采用 100MBps 主频进行设计。这些丰富的资源使 XC7A100T 成为高性能嵌入式系统、信号处理和通信应用的理想选择。

图 4-2 Xilinx Virtex-7 FPGA 开发板

XilinxVirtex-7XC7VX785T 是一款基于 28nm 工艺的逻辑资源丰富的可编程逻辑器件，适用于高密度、高性能的应用。具有 785000 个逻辑单元，包括丰富的存储器资源和具有高性能数字信号处理功能的 DSP Slice；复杂数字设计；快速信号处理操作；实现高速串行接口，包括 GTX/GTH 支持高速收发器，实现 PCIe®和以太网等高速串行协议，支持高速数据通信；时钟管理灵活，支持 FMC 扩展。为高性能计算、5G 通信，航空航天等高性能系统，提供高带宽和复杂的设计解决方案。

Virtex-7 FPGA 面向高密度，应用在数据中心和 5G 等领域，Artix 系列低成本、低功耗，应用在工业控制和消费电子产品上。在不同款 FPGA 上验证设计电路，可以验证设计是否易移植、易兼容，最大化设计资源的利用，并找到设计中可能的瓶颈。

为了验证不同工艺平台中所提出 TRNG 的表型，本文在 Xilinx UltraScale+ VCU118 FPGA（16nm 工艺）上进行了严格测试作为对比，如图 4-3 所示。

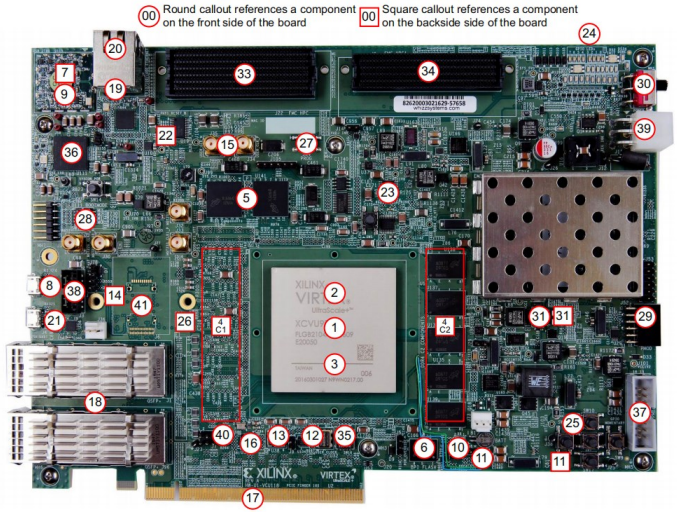


图 4-3 Xilinx UltraScale+ VCU118 FPGA 开发板

4.1.2 软件开发平台

本实验采用的 Vivado2018.3 设计套件对 FPGA 电路设计进行了优化，可支持硬件描述语言 Verilog 和 VHDL（本文采用 Verilog），Vivado2018.3 提供综合、布局布线和优化，提供了分析时序、功耗的工具，可用于高性能设计的实现。

在仿真与调试方面，Vivado 2018.3 提供了强大的功能支持。其内置的 Vivado Simulator 能够进行功能仿真和时序仿真，开发者还可以通过 Integrated Logic Analyzer（ILA）和 Virtual Input/Output（VIO）工具对硬件进行实时调试，快速定位和解决问题。此外，Vivado 2018.3 支持 IP Integrator，通过图形化界面集成各种 IP 核，极大提高了系统级设计的效率。对于复杂设计，该工具还支持 Partial Reconfiguration，允许在不影响其他部分的情况下动态修改部分逻辑。

Vivado 2018.3 的用户界面简洁直观，分为多个功能区域。具体功能如图 4-4 所示。

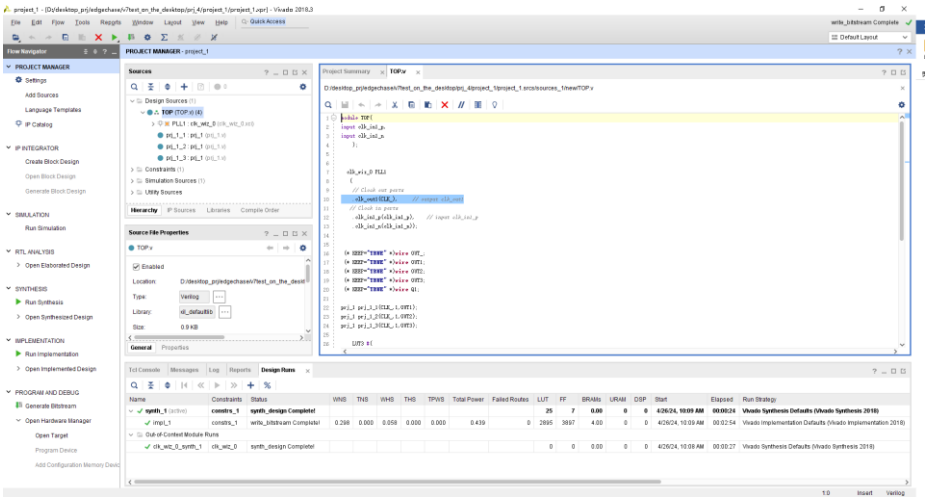


图 4-4 Vivado 2018.3 用户界面

右上角为主要代码编辑区，用于编写硬件描述语言；左侧为设计流程导航面板，包含综合、实现、生成比特流等步骤；中间区域为设计概览和分析面板，开发者可以在此查看资源利用率、添加约束条件、进行时序优化等操作。通过这些功能，Vivado 2018.3 为复杂的 FPGA 和 SoC 设计提供了全流程支持，是当前 FPGA 设计领域的主流工具之一。

如图 4-5 所示，为全面评估提出的 TRNG 电路稳健性，本文采用德州仪器提供的数字电压控制模块及其配套软件 Fusion Digital Power Designer 对电压温度

图 4-5 Fusion Digital Power Designer 调节温度

进行动态调整和监控。红色框区域为电压调节监控模块，支持精准控制和实时监控电压以确保设置范围内电压变化稳定。而绿色框区域是温度监测模块。TRNG 电路在不同环境条件下的性能表现，可以通过对电压和温度的双重变化实验进行系统分析，作为影响电路稳定的关键因素，温度和电压的变化很可能造成随意数输出的起伏。如果提出的 TRNG 电路在各种极端条件下仍能保持稳定的输出性能，则能充分证明其适应复杂多变场景的高路棒和可靠性。

4.2 实验实现

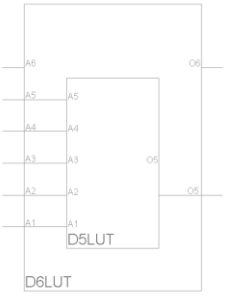
4.2.1 硬件语言实现

利用硬件原语实现电路设计特别适用于对实时性要求较高的应用场景，它一种是直接利用底层硬件资源进行功能设计的高效方式。硬件原语是 Verilog 语言中用来表示电路中基本逻辑门、触发器、多路选择器等基本模块。这些硬件原语通常由 Verilg 编译器和目标设备的库文件中定义其行为和功能。设计人员通过使用硬件原语，直接操作硬件逻辑单元，通过这种方法可以让系统性能更加可靠且优化资源利用率。

（1） 利用查找表搭建逻辑门

查找表（LUT）是 FPGA 中的核心组件，本质上是一种小型的随机存取存储器（Random Access Memory，简称 RAM），通过预先存储逻辑功能的真值表来

实现复杂的数字逻辑。开发者在 FPGA 中使用硬件描述语言（HDL）描述逻辑功能后，开发软件会计算出所有可能的输入组合及其输出结果，并将其写入 LUT 中。例如，一个 4 输入的与门（AND gate）需要一个 16x1 的 RAM 来存储所有可能的输入组合（如 A、B、C、D）及其输出结果，LUT 通过输入信号作为地址索引，快速查找并输出逻辑值。当一个 6 输入查找表（LUT6）作为地址传递给 LUT 时，它包含 64 个存储单元，能够存储 6 个输入变量的所有可能组合及其输出结果。本文提出的电路结构在 Artix-7、Virtex-7 和 UltraScale+ VCU118 开发板上实现，基于 LUT6 原语例化，通过具体的代码实现，并例化多个 LUT 原语，灵活构建复杂电路结构。如图 4-6 所示，图（a）为查找表的结构图，图（b）为利用 6 输入查找表搭建的异或门例化代码。这种设计高效利用了 FPGA 的逻辑资源，显著提升了电路设计的灵活性和可扩展性。由于 LUT 基于 SRAM 工艺，FPGA 掉电后会丢失配置信息，因此需要外部配置芯片在上电时加载数据。这种结构使 LUT 能够快速实现从简单逻辑门到复杂数字系统的功能，成为 FPGA 设计中不可或缺的组件。



（a）查找表结构图

```
LUT6 #(
   .INIT(64'h6996966996696996)
) xor1 (
   .O(x01),
   .I0(w01),
   .I1(w02),
   .I2(w03),
   .I3(w11),
   .I4(w12),
   .I5(w13)
);
```

（b）查找表例化代码

图 4-6 查找表的调用及例化代码

（2） 触发器调用

在 Vivado 中，FDCE 是一种带异步清零（CLR）和时钟使能（CE）的上升沿触发的 D 触发器原语，它可以直接映射到 FPGA 底层的硬件资源中。它可以在 Language Templats 中调用，如图 4-7 所示。以下是 FDCE 的端口定义和功能描述：C：时钟输入端口，上升沿有效，触发器在时钟上升沿时采样 D 端的数据；CE：时钟使能端口，高电平有效，当 CE=1 时，触发器正常工作，当 CE=0 时，触发器保持当前状态；CLR：异步清零端口，高电平有效，当 CLR=1 时，触发器的输出 Q 被强制清零为 0，不受时钟信号影响；D：数据输入端口，存储的数

据在时钟上升沿时被采样；Q：数据输出端口，输出触发器存储的值。

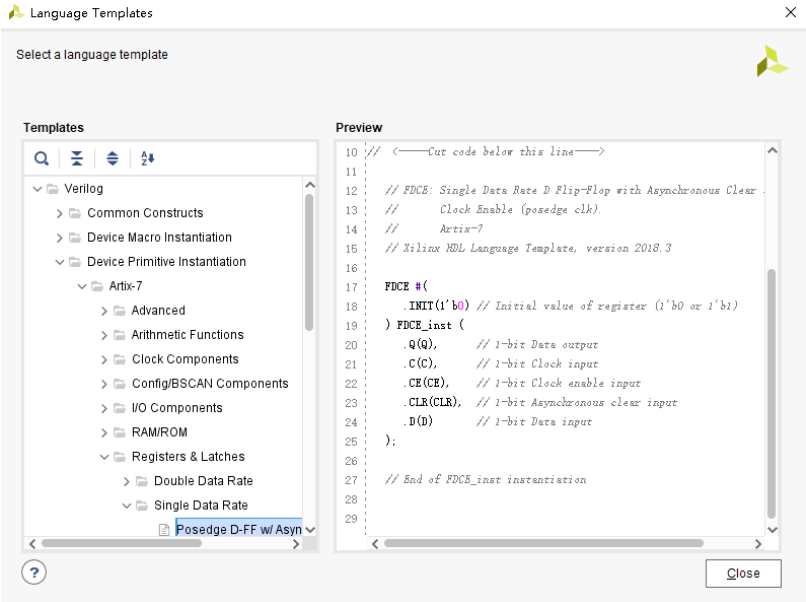


图 4-7 FDCE 的调用及例化代码

以下是本文中 FDCE 的 Verilog 实现示例：

如图 4-8 所示，在该示例中，C 端连接采样时钟 clk，该时钟决定了 TRNG 输出的吞吐量；D 端连接异或门链采样电路的输出；Q 端输出采样结果。CE 端固定为高电平 1，使触发器始终处于工作状态；由于本结构利用的是两个 RO 的信号边沿不断变化，不需要进行重启，所以 CLR 端固定为高电平 1。

通过这种方式，FDCE 能够高效地实现采样和控制功能，同时充分利用 FPGA 的硬件资源。

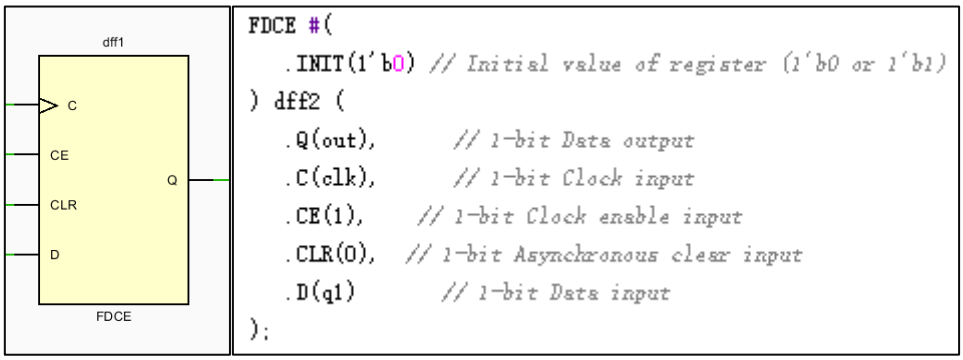


图 4-8 触发器的调用

（3） IP 核的调用与配置

IP 核（Intellectual Property Core）作为预设计且已验证功能模块，在 FPGA 设计中可以明显简化设计过程，提高开发效率。vivado 提供了丰富 IP 核库。设计者可以通过 IPCatalog 浏览并选择合适的 IP 核，双击进入配置界面后，根据数

据宽度、时钟频次等具体需求进行参数设置。配置完成后，点击通用按钮生成 IP 核文件，vivado 自动生成相应的 HDL 代码和约束文件等，最终将生成的 IP 核心模块进行例化，并在顶层设计中将端口与设计信号连接起来，完成融合。

在本次 TRNG 设计中，采样时钟频率为 360MBps，该高频率的采样时钟信号通过 MMCM（Mixed-Mode Clock Manager）信号产生，如图 4-9 所示。MMCM 是 FPGA 内部的高灵活度时钟生成模块，其作用是为系统提供多频时钟输出并根据用户设计中的不同要求实现不同类型的时钟分配。MMCM 的一个重要特点是宽范围频率的时钟产生，它与 FPGA 芯片内部的可编程分频器、乘法器、相位调整器等有关，可以根据需要灵活地提供输出时钟频率与相位的设定。此外，MMCM 支持在线调整功能，即可以在运行状态下根据输入的电压状态或其他的变量条件对输出时钟频率进行调整。在实际设计中，通过在 Vivado 中调入 MMCM 的 IP 核，调整其输入时钟频率与输出频率的范围以及相位关系，即可得到满足要求的 360MBps 的采样时钟信号以实现 TRNG 的设计。

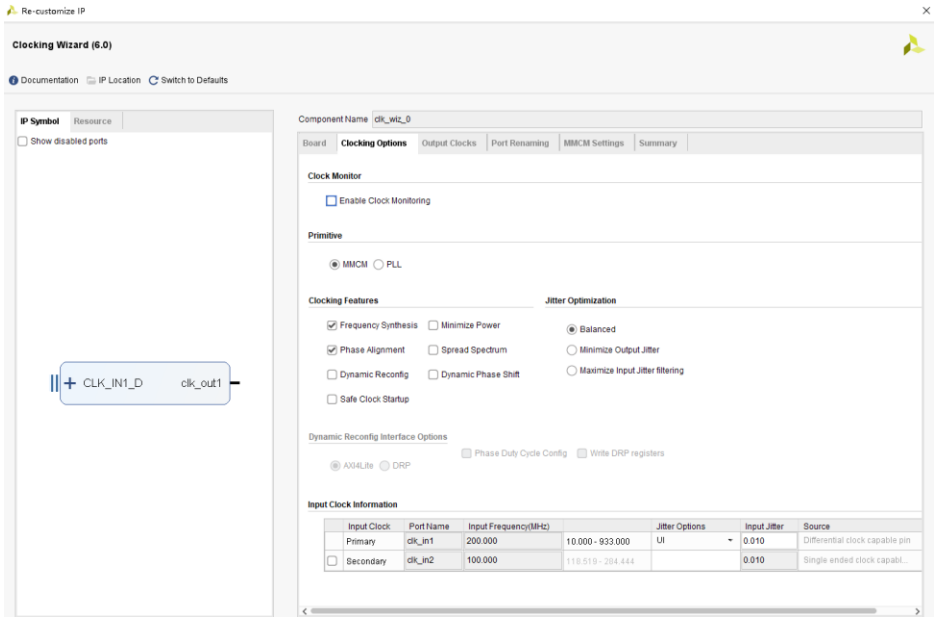


图 4-9 IP 核的调用

（4） Setup debug 抓取波形

针对基于 FPGA 的 TRNG 设计，通过 Setup Debug 抓波形来获取输出的随机数，这是最为有效快捷的调试方法，具体的过程如下：在设计时通过在 HDL 文件中设置（*mark_debug=”true”*）属性标记需要查看的信号，或者在 Vivado 中通过 Setup Debug 直接在综合后的网表中选择“Mark Debug”标记调试的重要信

号，如图 4-10，Vivado 会自动根据标记生成调试核，常使用 ILA 来实现该功能，ILA 是一种硬件调试工具，可实时抓取波形并在开发环境进行显示；配置调试核时，通过设置关键参数如采样深度（设定波形捕获的长度），触发条件（用于捕获特定的事件的波形，包括某信号值、上升沿等、组合逻辑状态等）以及时钟域（设置 ILA 抓取信号波形的采样时钟必须与所要调试的信号的时钟相同，以防止产生采样错误）；配置完成之后生成比特流文件并将其下载至 FPGA 当中，在硬件管理器（Hardware Manager）当中可以设置 ILA 抓取条件，条件一旦满足 ILA 会捕获所要的信号波形并通过硬件管理器传输展示给用户。开发者可依据这些波形检查信号行为是否如预期一样发生，找出逻辑错误或时间延迟等问题，并进一步改进设计，这些设计可以迅速调试 FPGA design 并提高设计中的缺陷探测速度。

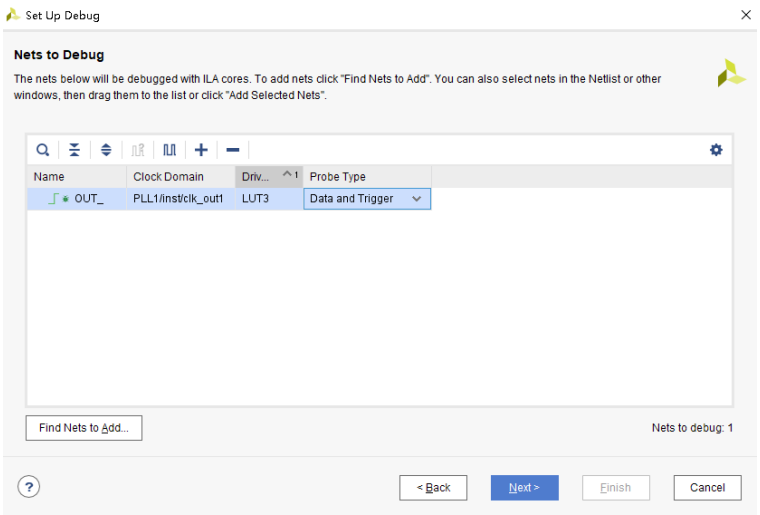


图 4-10 利用 Set up debug 抓取波形

4.2.2RTL 电路结构

RTL（Register Transfer Level）电路结构是数字电路设计中的一种核心抽象层次，用于描述数据在寄存器之间传输和处理的过程。它通过硬件描述语言（如 Verilog 或 VHDL）定义寄存器、组合逻辑以及控制信号的行为，既能清晰地表达硬件功能，又便于综合为实际的逻辑电路。RTL 设计兼具可读性、可综合性和模块化特性，广泛应用于处理器、通信系统和 FPGA/ASIC 设计中，是硬件实现的关键桥梁。本设计的 RTL 电路结构见图 4-11。

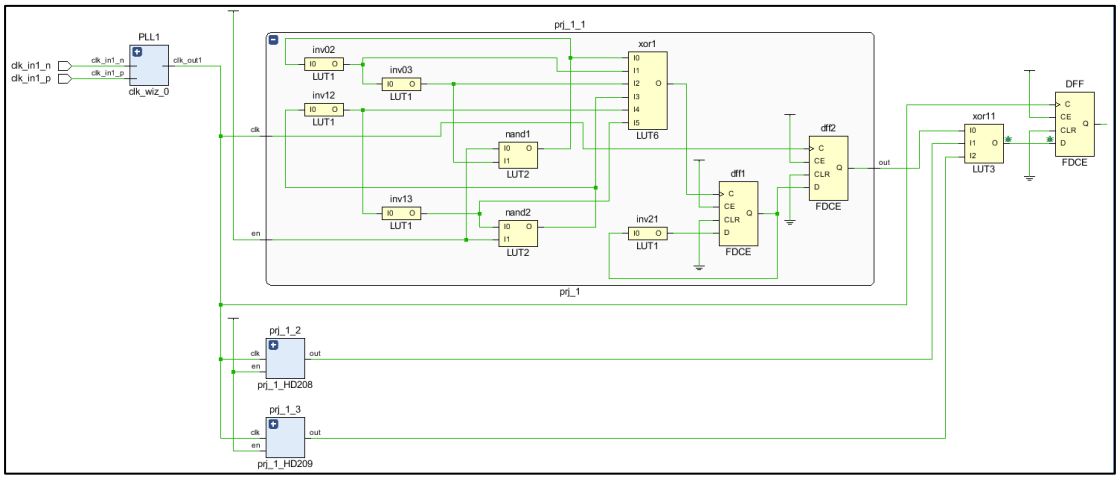


图 4-11 RTL 电路结构

其中，PLL1 为产生采样时钟的 MMCM 模块，它的输入有两个，分别为 clk_in_n 和 clk_in_p 两个差分时钟信号，时钟信号的频率为 200MBps，通过该模块，可以得到 360MBps 的最终输出频率；prj_1 模块为产生随机性的核心模块，以异或链为核心的 TRNG 不断产生高频脉冲信号，对其计数的过程，以及亚稳态现象的发生都在该模块中；通过将三个 prj_1 模块的再 xor11 上进行异或处理，然后对其输出进行采样，得到最终的 TRNG 输出，可以将三个 pri_1 熵源模块进行叠加看，提高整体设计的随机性。

4.2.3EDA 工具自动布局布线

自动布线比手工布线具有更好的移植性。一方面，自动布线可以快速迁移到其它芯片或 FPGA 工艺和架构平台，无论是更换芯片或者换用其它平台的 FPGA，设计可快速迁移到其它平台，而手工布线的结果针对的是一些特定的平台，要重新移植就需要根据这个新的平台调整布线，费时、费力。此外，自动布线的约束和优化策略对各种平台都通用和可重用，只要开发者重新设定一些参数，就可以快速将设计迁移到其他平台，而手工布线的结果几乎都是为某个平台定制的，没有通用性。另一方面，自动布线生成的物理文件（GDSII 或者比特流）遵循业界的通用格式，可直接用于其它厂家的硬件实现，而手工布线的结果需要对其进行许多改动再迁移到新的平台。因此，自动布线省去了许多移植环节，加之移植的结果可用于不同的平台硬件，实现较高程度的可移植性。本设计布线效果图（自动布线）如图 4-12。

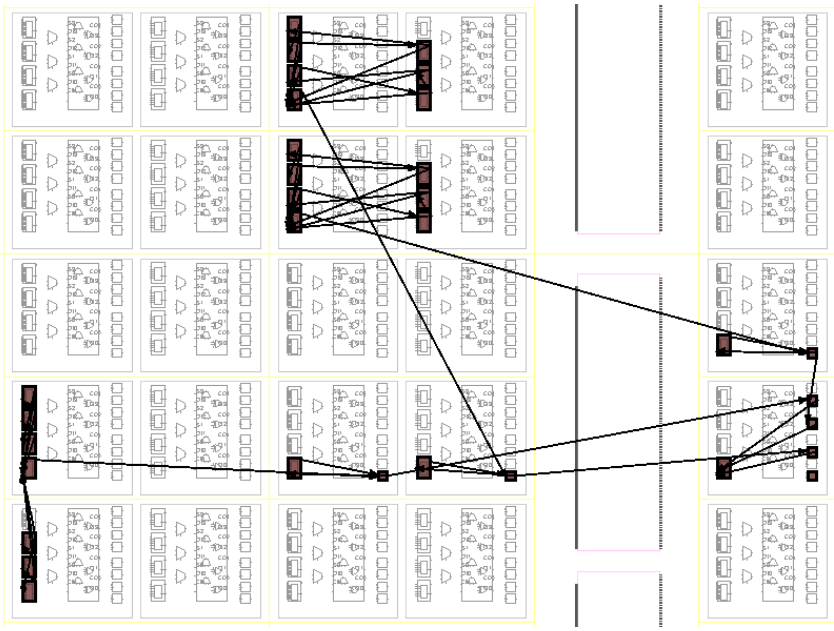


图 4-12 EDA 工具自动布局布线图

可以看到，一个 CLB 中，有两个 Slice，其中的小方块即为所使用的 LUT 和触发器，可以看出总体的布局布线并不对称。

4.2.4 随机序列的采集

为了方便地捕获本实验的 TRNG 产生的随机数并快速处理数据，在数据抓取过程中通过编写 TCL 脚本设置 ILA 对 TRNG 输出的随机数信号不断进行抓取，提取过程见图 4-13。

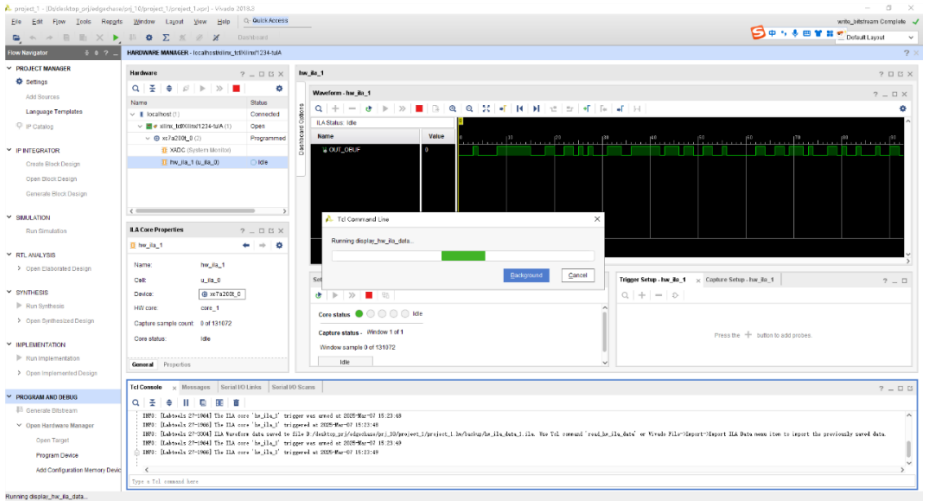


图 4-13 逻辑分析仪抓取数据

ILA 抓取的随机数存为一个.csv 的文件，可方便管理，避免长时间抓取时数据丢失。同时编写的 Python 语言脚本软件可自动读取该.csv 文件，提取抓取到

的随机数位序列，可进行高速处理；然后对抓取数据进行转换，并将转换后的数据合并、规范成一个含有一百万个二进制随机数的.txt 的随机数文件。该二进制文件可直接运用到随机性测试（NIST 测试套件），以及对生成数据所对应的统计分析，从而检验 TRNG 性能的（如熵值、自相关性、统计的均匀性等）。

4.3 实验结果分析

4.3.1NIST SP800-22 测试

NIST SP800-22 是美国国家标准与技术研究院（NIST）发布的一项权威随机性测试套件[68]，旨在评估随机数发生器（RNG）和伪随机数发生器（PRNG）输出序列的统计特性，以确保其满足密码学应用的安全性要求。该测试套件包含 15 项统计测试，具体如下：

1. 频率测试（Frequency Test）：检验序列中 0 和 1 的比例是否接近 0.5。
2. 块内频率测试（Block Frequency Test）：评估固定大小子的序列中 0 和 1 的分布是否均匀。
3. 游程测试（Runs Test）：检验序列中连续 0 或 1 的游程长度是否符合随机分布。
4. 最长游程测试（Longest Run of Ones in a Block Test）：分析块内最长连续 1 的分布是否随机。
5. 矩阵秩测试（Rank Test）：评估二进制矩阵的秩分布是否符合随机预期。
6. 离散傅里叶变换测试（Discrete Fourier Transform Test）：检测序列中是否存在周期性模式。
7. 非重叠模板匹配测试（Non-overlapping Template Matching Test）：检验特定模式在序列中出现的频率是否随机。
8. 重叠模板匹配测试（Overlapping Template Matching Test）：类似于非重叠模板匹配测试，但允许模式重叠。
9. 通用统计测试（Maurer’s Universal Statistical Test）：评估序列的压缩和复杂性。
10. 线性复杂度测试（Linear Complexity Test）：分析序列种的线性反馈移位寄存器（LFSR）复杂度。

11. 序列测试（Serial Test）：检验所有可能 m 位模式的频率是否均匀。
12. 近似熵测试（Approximate Entropy Test）：评估序列相邻模式的频率是否随机。
13. 累积和测试（Cumulative Sums Test）：检验序列的累积和是否偏离随机分布。
14. 随机游走测试（Random Excursions Test）：评估序列中随机游走的统计特性。
15. 随机游走变体测试（Random Excursions Variant Test）：对随机游走测试的进一步扩展。

以上各测试项可以测试序列的均匀性、独立性、不可预测性。将得到的随机数序列转换成二进制序列并进行依次测试，通过求出 p 值得到是否通过测试的结果。如果 p 值大于显著性水平（0.01），则说明随机序列的随机特性良好。NISTSP800-22 在密码学领域占有非常重要的位置，也是在加密算法、密钥生成等安全应用中对随机数发生器的性能验证的一个科学且规范的方法。给研究工作者与开发者提供了一个评价的标准与范例，是进行随机数发生器设计重要参照。

为了验证所提出 TRNG 的可移植性，本文在 Xilinx-Artix7 和 Virtex7 两种 FPGA 开发板上实现了该 TRNG，并在每个开发板中收集了 100M 测试数据，测试结果如表 4-1 所示。结果表明，所提出的 TRNG 结构具有良好的随机性，各测试项在不同的 FPGA 上 P 值均明显高于 0.01，15 项测试的最小通过率为 96%。因此，所提出的 TRNG 能够生成真正的随机数序列。

表 4-1 NIST SP 800-22 测试结果

开发套件	Artix-7			Virtex-7		
随机性测试项目	P 值	通过率	总结果	P 值	通过率	结果
近似熵测试	0.460847	96%	通过	0.493381	96%	通过
块内频率测试	0.505201	100%	通过	0.527789	100%	通过
累加和测试	0.451448	98%	通过	0.453762	97%	通过
离散傅里叶变换测试	0.530463	99%	通过	0.488243	100%	通过
频率测试	0.453313	96%	通过	0.445735	97%	通过
线性复杂度测试	0.556501	99%	通过	0.498923	98%	通过
块内最长游程测试	0.483516	98%	通过	0.501157	100%	通过
非重叠模块匹配测试	0.496432	99%	通过	0.498010	99%	通过
重叠模块匹配测试	0.453149	100%	通过	0.483221	100%	通过
随机游动测试	0.283367	98%	通过	0.315676	98%	通过
随机游动状态频数测试	0.290957	99%	通过	0.315612	99%	通过
二元矩阵秩测试	0.476661	100%	通过	0.536313	100%	通过
游程测试	0.494245	99%	通过	0.455160	98%	通过
序列测试	0.506148	98%	通过	0.526344	100%	通过
通用统计测试	0.491044	100%	通过	0.495330	100%	通过

4.3.2NIST SP800-90B 测试

本文设计的真随机数发生器分别在不同的 FPGA 平台（Xilinx Virtex-7、Artix-7、UltraScale+ VCU118）上实现和验证，经过 NIS TSP800-90B 随机数检测流程[69]，得到所设计的真随机数发生器输出的最小熵分别为为 0.904603（平台：Virtex-7）、0.912537（平台：Artix-7）。同时，说明设计的真随机数发生器在具有很高的随机性的同时适用于不同的硬件平台。

随机数最小熵是度量随机性的一个重要参数，熵值越大，意味着随机性更加强。本文设计的 TRNG 满足熵值要求，完全适合可以作为安全应用的随机源。本文提供的全部测试结果如表 4-2 所示，包括了具体的统计测试结果和统计分析。

表 4-2 NIST SP 800-90B 测试结果

开发套件	Artix-7		Virtex-7	
测试项目	P(max)	h-min	P(max)	h-min
最频值	0.501416	0.99592	0.501333	0.996158
碰撞	0.53418	**0.904603**	0.53125	**0.912537**
马尔科夫	3.70677e-39	0.997383	3.99066e-39	0.996551
压缩	0.5	1	0.5	1
元组	0.528634	0.91966	0.522944	0.935271
最长重复字串长度	0.503199	0.990798	0.52032	0.94253
多个 MCW	0.501086	0.996871	0.501401	0.995964
滞后	0.500788	0.997728	0.500902	0.9974
多个 MMC	0.500781	0.997748	0.500881	0.997461
LZ78Y 算法	0.500951	0.997258	0.501544	0.995553

4.3.3 重启测试

为了进一步确认 TRNG 输出随机数的真正随机性，本文设置了重启实验，即 TRNG 重新启动之后，得到其输出结果，通过这些结果比较可以判断出所设计 TRNG 输出的随机数序列具有不可预知性和不可重复性。每次 TRNG 重启，抽取的随机数各不相同则说明每次重新启动 TRNG 时都会得到一个新的随机数序列，即 TRNG 输出的序列具有真正随机性。

实验过程中重启 TRNG6 次，记录重启 TRNG 后分别得到的输出序列。结果如图 4-14 所示，可见 6 次重启后的采样数据之间的差异。由图可见，重启 TRNG 后每次得到的随机序列都大不相同，说明使用了所提出 TRNG 的在不同的初始

化下产生随机序列也不同，因而得到的随机序列的输出不可预测并且不重复。

上述实验结果可进一步说明所设计的 TRNG 的稳定与有效性，可以认为所设计的 TRNG 能够满足真随机数生成器的设计要求。重启测试也给验证 TRNG 实际运行中的性能提供了一个参考标准。

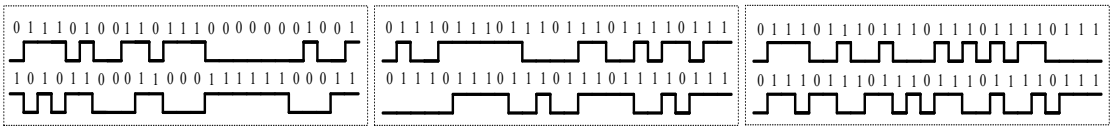


图 4-14 6 次重启测试结果

4.3.4 电压和温度和功耗测试

本文在 Xilinx Virtex-7 中对所提出结构的性能进行了 NIST SP800-22 测试套件在不同环境下（温度为 20°C，40°C，60°C，80°C 下，电压为 0.8V，0.9V，1.0V 和 1.1V）的测试。测试结果如图 4-15 和图 4-16 所示。图中 1~15 为 NIST SP800-22 的 15 个测试项，如果一个测试项的结果大于红色阈值线，那么就代表这个测试项测试通过了。

本文所设计的结构在绝大多数条件下通过了 NISTSP800-22 的 15 项测试，除了 60°C 时有 1 项测试略低于标准的要求。验证了本文设计的结构在温度在 20°C～80°C 以及电压在 0.8V～1.1V 的大范围内仍能产生质量满足要求的真随机数，验证了在大范围温压差环境下的鲁棒性。

以上实验结果也证实了本文提出结构在实际应用中将带来的可能性，尤其是对于工业级别以及高安全性级别，在面对温度和电压漂移时依然稳定，因此是一种可行的随机数生成器。

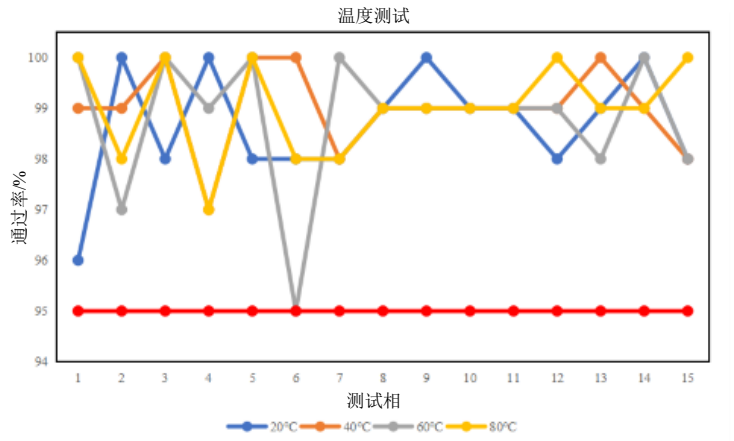


图 4-15 温度测试结果

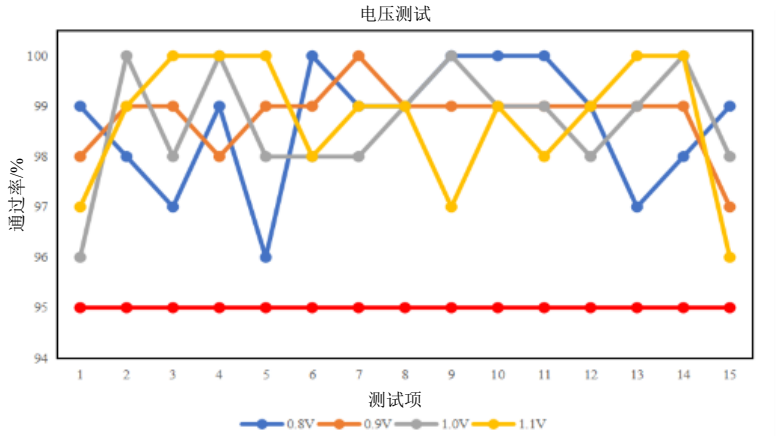


图 4-16 电压测试结果

此外，为了进一步验证本文所提出结构在保证性能的前提下，延长设备续航、减少发热、降低成本并提升可靠性。本文还进行了功耗测试，结果见表 4-3。

表 4-3 功耗测试结果

On-chip	Power(W)
Clocks	0.001
Signals	0.001
Logic	0.001
BRA	0.001
I/O	0.002

4.3.5 自相关测试

序列中在不同时刻随机发生的比特之间互相关性用自相关系数来评价，为进一步检验所构建结构的随机数发生器质量，从所涉及的 TRNG 电路中提取出一万个连续的比特，再借助 Python 平台计算出自相关函数，并将滞后值设定为 1~100，分析自相关因子（Autocorrelation Function，ACF）来判断变量之间的互相关性。

通过图 4-17 可以得知，ACF 的值大部分在-0.002~0.002 范围内变化，从而可以得知所生成随机序列的自相关性较低，因为 ACF 的值远小于阈值（一般取 0.05），证明序列的相关性较小。这样的小自相关特征表示生成的随机序列独立性较强，不容易被相关分析攻击，进而验证出本结构的随机性质量良好。

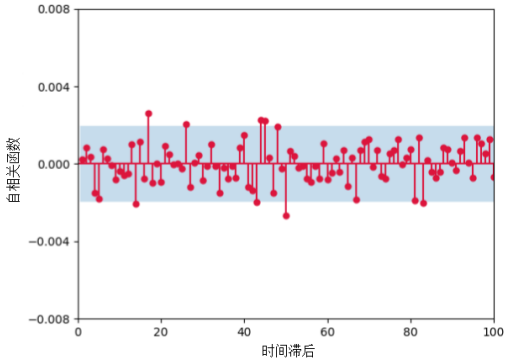


图 4-17 自相关测试结果

4.3.6 偏差测试

偏差测试主要用于检测生成序列是否具有明显的偏向性（如偏向“0”或“1”）。对于理想的随机序列，比特“0”和“1”的出现概率应接近相等，在统计学上即可表示为$P(0) \approx P(1) \approx 0.5$。

本次偏差检验时的具体方案是在 Xilinx Artix-7 FPGA 平台上产生连续 100 万个比特以黑白像素图像的形式表现出来，直观地对比特分布的均匀度进行判断，其中，比特序列中的“0”和“1”分别用黑、白表示像素。而黑白像素图像如图 4-18 所示，图中可以看出，黑、白像素分布均匀，不存在局部区域偏向“0”或“1”的情况，这表明产生的比特序列中“0”、“1”出现概率接近相等，序列几乎不存在偏移，具有良好的随机性。

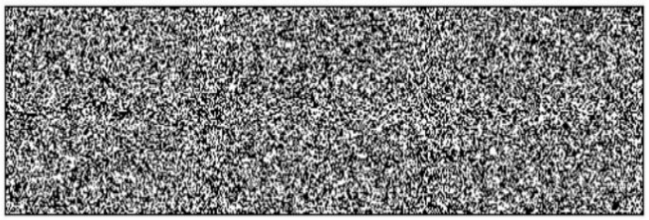


图 4-18 偏差测试结果

4.4 与现有结构对比分析

为了全面评估所提出结构的性能，本文将其与近期文献中发表的 TRNG 结构进行了详细对比，结果如表 4-3 所示。文献[14]利用抖动和亚稳态作为熵源，虽然实现了随机数生成，但其吞吐量仅为 12.5MBps，且需要手动布局布线，增加了设计复杂度。文献[28]基于 RS 锁存器生成亚稳态作为熵源，但亚稳态的生成条

件较为苛刻，限制了其实际适用性。文献[49]的结构在硬件资源消耗方面优于本文，但其吞吐量仅为 100MBps，并且需要多相时钟进行采样，增加了系统设计的复杂性。文献[22]虽然在硬件资源消耗方面表现出色，但在 Artix-7 开发板上的吞吐量不到本结构的一半，限制了其性能。文献[13]和[53]分别采用抽头延迟链和触发器阵列对时钟抖动进行精细化采样，但无论是在硬件资源消耗还是吞吐量方面，均无法与本文结构相媲美。文献[70]通过可编程延迟链提高了熵源的随机性，但其硬件开销是本结构的 20 多倍，且吞吐量仅为 6MBps，难以满足高速应用需求。文献[15]同样利用两个 RO 的边沿差生成随机数，但其所需的环长度远高于本结构，增加了硬件实现的复杂度。综合对比结果表明，与现有结构相比，本文所提出的 TRNG 结构在显著减少硬件资源消耗的同时，实现了更高的吞吐量，充分证明了其在性能与效率上的显著优势。

表 4-4 结果对比

对比方法	实验平台	硬件开销	吞吐量
Tcas-I(2022)[14]	Spartan-6	36LUTs	12.5
IEICE-TIS(2012)[28]	Virtex-4	128LUTS,256DFFs	3.85
Asian-HOST(2022)[49]	Virtex-6/Virtex-7/Aritx-7	4MUXs,4DFFs,1LUT	100
Tcas-II(2024)[22]	Aritx-7/Kintex-7	12LUTs,10DFFs	150/200
Tcas-I(2022)[13]	Aritx-7	32LUTs,55DFFs,17Carry4	12.5
DAC(2023)[53]	Aritx-7	24LUTs,33DFFs	275.8
Tcas-II(2020)[70]	Spartan-3A	528LUTs,177DFFs	6
本文	Aritx-7/Virtex-7	25LUTs,6DFFs	360/330

4.5 本章小结

本章基于 Xilinx Artix-7、Virtex-7 FPGA 和 UltraScale+ VCU118 平台，实现了真随机数生成器的设计与性能验证。通过硬件描述语言和RTL结构完成设计，利用 EDA 工具进行自动布局布线，并通过 ILA 抓取随机序列进行测试。NIST SP800-22 和 SP800-90B 测试结果显示，本文设计的 TRNG 在两种平台上均表现出高随机性，最小熵值分别为 0.904603 和 0.912537，测试通过率超过 96%。重启测试验证了输出序列的不可预测性，温度（20°C 至 80°C）和电压（0.8V 至 1.1V）测试则证明了 TRNG 的鲁棒性。自相关测试和偏差测试进一步表明，生成序列具有低相关性和均匀分布特性。与现有结构对比，所提 TRNG 在减少硬件资源消耗的同时实现了更高吞吐量，展现了显著的性能与效率优势。

第五章 总结与展望

5.1 总结

本文围绕基于环形振荡器和亚稳态电路的真随机数生成器设计与实现展开研究，提出了结构优化方案，并在 FPGA 平台上完成了硬件实现与性能验证。研究工作主要分为以下几个部分：

1. 理论分析与设计优化：通过对环形振荡器和亚稳态电路的理论研究，提出了结构优化方案，有效提高了随机性生成的稳定性与效率。
2. FPGA 实现：在 Xilinx Artix-7、Virtex-7 FPGA 和 UltraScale+ VCU118 平台上完成了 TRNG 的硬件描述语言设计，并通过 RTL 电路结构与自动布局布线实现了物理设计。
3. 性能验证：采用 NIST SP800-22 和 SP800-90B 测试套件对生成的随机序列进行了全面评估，结果表明 TRNG 在两种平台上均满足高随机性要求，最小熵值分别达到 0.904603 和 0.912537，测试通过率超过 96%。重启测试、温度与电压变化测试进一步验证了 TRNG 的不可预测性和鲁棒性。对比实验表明，所提出的 TRNG 在资源消耗与吞吐量方面，展现了较好的性能优势。

本文的研究成果为真随机数生成器的硬件实现提供了新的设计思路，并在随机性、稳定性和资源效率等方面取得了显著进展。

5.2 展望

尽管本文在 TRNG 设计与实现方面取得了一定成果，但仍存在一些值得深入研究的方向：首先，针对资源受限的场景，可探索更低功耗的 TRNG 设计方案，以提升 TRNG 的；其次，为应对潜在的物理攻击（如侧信道攻击），需研究更具安全性的 TRNG 设计，增强其在密码学应用中的抗攻击能力；最后，可以探索 TRNG 与其他硬件模块（如密钥生成器、加密引擎）的深度融合，为安全通信与数据处理提供一体化解决方案。这些方向将为未来 TRNG 技术的发展提供更多可能性。

参考文献

[1].Stuart E. Madnick.The Continued Threat to Personal Data: Key Factors Behind the 2023 Increase [R]. America, 2023-12-07 https://www.apple.com/newsroom/2023/12/report-2-point-6-billion-records-compromised-by-data-breaches-in-past-two-years/.

[2].谭阳. 关于随机数生成算法的研究[D]. 湖南师范大学，2008.

[3].胡月. 基于生物特征的真随机数发生器的研究与设计[D]. 重庆大学，2008.

[4].Johnson, Anju P ,Mukhopadyay,et al.An Improved DCM-Based Tunable True Random Number Generator for Xilinx FPGA[J]. IEEE Transactions on Circuits and Systems, II. Express briefs, 2017.

[5].孟震. HTTPS 监控系统的设计与实现[D]. 青岛科技大学，2018.

[6].佘冉君. SSL 安全研究及实现[D]. 西安电子科技大学，2015.

[7].Kruh, Louis. Digital Security in a Networked World[J]. Cryptologia, 2001(Jan).

[8].Kelsey J, Schneier B, Wagner D A, et al. Cryptanalytic attacks on pseudorandom number generators[J]. Fast Software Encryption, 1998, 1372: 168-188.

[9].Chen T , Ma Y , Lin J ,et al.A Lightweight Full Entropy TRNG with On-chip Entropy Assurance[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021, (99): 1-1.

[10].杨建会. 基于 InGaAs/InP 雪崩二极管的单光子探测及其噪声分析[D]. 北京邮电大学，2015.

[11].韩倩.基于振荡环的真随机数发生器研究与设计[D]. 合肥工业大学，2023.

[12].Bai B , Huang J , Qiao G R ,et al.18.8 Gbps real-time quantum random number generator with a photonic integrated chip[J]. AIP Publishing LLC AIP Publishing, 2021(26).

[13].Grujić M and Verbauwhede I. TROT: A three-edge ring oscillator based true random number generator with time-to-digital conversion[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2022, 69(6): 2435-2448.

[14].Sala R D, Bellizia D and Scotti G. High-throughput FPGA-compatible TRNG architecture exploiting multistimuli metastable cells [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2022, 69(12): 4886-4897.

[15].Cao Y, Zhao X, Zheng W, Y Zheng et al. A new energy-efficient and high throughput two-phase multi-bit per cycle ring oscillator-based true random number generator[J]. IEEE Transactions on Circuits and Systems I: Regular

Papers, 2022, 69(1): 272-283.

[16].Peetermans, A and Verbauwhede I. An energy and area efficient, all digital entropy source compatible with modern standards based on jitter pipelining[J]. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022, 2022(4): 88-109.

[17].Johnson A P, Chakraborty R S and Mukhopadyay D. An improved DCM-based tunable true random number generator for Xilinx FPGA[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2017, 64(4): 452-456.

[18].Şarkışla M A and Ergün S. Ring oscillator based random number generator using wake-up and shut-down uncertainties[C] // 2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), Hong Kong, China, 2018: 104-108.

[19].Robson S , Leung B , Gong G .Truly Random Number Generator Based on a Ring Oscillator Utilizing Last Passage Time[J]. Circuits and Systems II: Express Briefs, IEEE Transactions on, 2014, 61(12):937-941.

[20].Prada-Delgado M A , Martinez-Gomez C , Baturone I .Auto-Calibrated Ring Oscillator TRNG Based on Jitter Accumulation[C] // 2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2020.

[21].Fang L H, Guo Z J, Shuang G L, et al. Autonomous boolean network regulation based on logic gates' response characteristics[J]. Acta Phys. Sin., 2021, 70(5): 050502.

[22].Yang S,Liang H, Hu R, et al. Lightweight hybrid entropy source true random number generator based on jitter and metastability[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2024, 71(7): 3513-3517.

[23].Cui J, Yi M, Cao D et al. Design of true random number generator based on multi-stage feedback ring oscillator[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2022, 69(3): 1752-1756.

[24].Cherkaoui A , Fischer V , Aubert A ,et al.A Self-timed Ring based True Random Number Generator with Monitoring and Entropy Assessment[C] // IEEE International Symposium on Asynchronous Circuits & Systems.IEEE, 2013.

[25].姚亮,黄正峰,梁华国,等. 利用细粒度采样的低开销双输出异或门真随机数发生器研究[J]. 电子与信息学报，2023,45(09):3295-3301.

[26].B. Valtchanov, V. Fischer and A. Aubert, Enhanced TRNG based on the coherent sampling[C] // 2009 3rd International Conference on Signals, Circuits and Systems (SCS), Medenine, Tunisia, 2009, pp. 1-6.

[27].Tokunaga C, Blaauw D and Mudge T. True random number generator with a

metastability-based quality control[J]. IEEE Journal of Solid-State Circuits, 2008, 43(1): 78-85.

[28].Hata, Hisashi & Ichikawa, Shuichi. (2012). FPGA Implementation of Metastability-Based True Random Number Generator[J]. IEICE Transactions. 95-D. 426-436.

[29].C. Li, Q. Wang, J. Jiang and N. Guan, A metastability-based true random number generator on FPGA[C] // 2017 IEEE 12th International Conference on ASIC (ASICON), Guiyang, China, 2017, pp. 738-741.

[30].P. Z. Wieczorek and K. Gołofit, Dual-Metastability Time-Competitive True Random Number Generator[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 61, no. 1, pp. 134-145, Jan. 2014.

[31].R. Della Sala, D. Bellizia and G. Scotti, A Novel Ultra-Compact FPGA-Compatible TRNG Architecture Exploiting Latched Ring Oscillators[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 3, pp. 1672-1676, March 2022.

[32].倪天明,俞俊勇,彭青松等. 基于亚稳态叠加单元的高吞吐量真随机数发生器设计[J]. 电子与信息学报，2024,46(05):2289-2297.

[33].Mohammadhadi D, Venkatasubramaniyan A B, Kapoor G et al. Unified analog PUF and TRNG based on current-steering DAC and VCO[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2020, 28(11): 2280-2289.

[34].C. S. Petrie and J. A. Connelly, A noise-based IC random number generator for applications in cryptography[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 47, no. 5, pp. 615-621, May 2000.

[35].M. Bucci, L. Germani, R. Luzzi, A. Trifiletti and M. Varanonuovo, A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC[J]. IEEE Transactions on Computers, vol. 52, no. 4, pp. 403-409, April 2003.

[36].R. Brederlow, R. Prakash, C. Paulus and R. Thewes, A low-power true random number generator using random telegraph noise of single oxide-traps[C] // 2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers, San Francisco, CA, USA, 2006, pp. 1666-1675.

[37].M. Matsumoto, S. Yasuda, R. Ohba, K. Ikegami, T. Tanamoto and S. Fujita, 1200μm2 Physical Random-Number Generators Based on SiN MOSFET for Secure Smart-Card Application[C] // 2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers, San Francisco, CA, USA, 2008,

pp. 414-624.

[38].Liu, Y., Zhu, M. Y., Luo, B., Zhang, J. W., & Guo, H. (2013). Implementation of 1.6 Tbs−1 truly random number generation based on a super-luminescent emitting diode[J] Laser Physics Letters, 10(4), 045001.

[39].Stefanov A, Gisin N, Guinnard O, Zbinden H. Optical quantum random number generator[J]. Journal of Modern Optics, 2000, 47(4): 595-598.

[40].金振阳,万相奎,庞恺,等.脉冲激光相位涨落的量子随机数发生器[J]. 光通信研究，2022,(06):27-34+63.

[41].沈海斌,赵梦恋,李晓明,等.基于混沌的高速随机数发生器[J]. 电路与系统学报，2003，(05):92-96.

[42].H. Rohail and R. Ramzan, A High Throughput True Random Number Generator using Metastability and Chaos[C] // 2022 20th IEEE Interregional NEWCAS Conference (NEWCAS), Quebec City, QC, Canada, 2022, pp. 40-44.

[43].A.Zacharias, C. G. Gisha and B. A. Jose, Chaotic Ring Oscillator Based True Random Number Generator Implementations in FPGA[C] // 2020 24th International Symposium on VLSI Design and Test (VDAT), Bhubaneswar, India, 2020, pp. 1-6.

[44].Hosokawa Y, Nishio Y. Simple chaotic circuit using CMOS ring oscillators[J]. International Journal of Bifurcation and Chaos, 2004, 14(07): 2513-2524.

[45].Y. Luo, W. Wang, S. Best, Y. Wang and X. Xu, A High-Performance and Secure TRNG Based on Chaotic Cellular Automata Topology[C] // in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 12, pp. 4970-4983, Dec. 2020.

[46].解旭辉,胡汉平,郑俊,等.一种面向硬件实现的模数混合混沌真随机数发生器[J]. 密码学报(中英文)，2024,11(06):1399-1414.

[47].刘正文,易茂祥,杨云,等.一种高吞吐率自治布尔网络真随机数发生器[J]. 电子测量与仪器学报，2023,37(09):102-109.

[48].B. Sunar, W. J. Martin and D. R. Stinson, A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks[J]. IEEE Transactions on Computers, vol. 56, no. 1, pp. 109-119, Jan. 2007.

[49].Yao L, Liang H, Zhang H, et al. A lightweight M_TRNG design based on MUX cell entropy using multiphase sampling[C] // Chip Hong Chang, 2022 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), Singapore, Singapore: IEEE, 2022: 1-4.

[50].O.Petura, U. Mureddu, N. Bochard and V. Fischer, Optimization of the PLL

based TRNG design using the genetic algorithm[C] // 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 2017, pp. 1-4.

[51].B.Colombier, N. Bochard, F. Bernard and L. Bossuet, Backtracking Search for Optimal Parameters of a PLL-based True Random Number Generator[C] // 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2020, pp. 1-6.

[52].Altera Corporation. Understanding Metastability in FPGAs[R/OL]. America (2009-07)[2024-12-30].https://www.intel.com/content/www/us/en/content-details/650346/understanding-metastability-in-fpgas.html

[53].Z. Lu, H. Qidiao, Q. Chen, Z. Liu and J. Zhang, An FPGA-Compatible TRNG with Ultra-High Throughput and Energy Efficiency[C] // 2023 60th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2023, pp. 1-6.

[54].K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw and D. Sylvester, 16.3 A 23Mb/s 23pJ/b fully synthesized true-random-number generator in 28nm and 65nm CMOS[C] // 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 2014, pp. 280-281,.

[55].K. Yang, D. Blaauw and D. Sylvester, An All-Digital Edge Racing True Random Number Generator Robust Against PVT Variations[J]. in IEEE Journal of Solid-State Circuits, vol. 51, no. 4, pp. 1022-1031, April 2016, doi: 10.1109/JSSC.2016.2519383.

[56].N. Bochard, F. Bernard and V. Fischer, Observing the Randomness in RO-Based TRNG[C] // 2009 International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico, 2009, pp. 237-242.

[57].Chen et al., A Design of High-Efficiency Coherent Sampling Based TRNG With On-Chip Entropy Assurance[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 12, pp. 5060-5073, Dec. 2023.

[58].Peetermans, V. Rozic and I. Verbauwhede, A Highly-Portable True Random Number Generator Based on Coherent Sampling[C] // 2019 29th International Conference on Field Programmable Logic and Applications (FPL), Barcelona, Spain, 2019, pp. 218-224.

[59].A.P. Johnson, R. S. Chakraborty and D. Mukhopadyay, An Improved DCM-Based Tunable True Random Number Generator for Xilinx FPGA[J] IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 4, pp.

452-456, April 2017.

[60].M. Drutarovsky, M. Varchola and M. Petrvalsky, Remotely testable setup of soft CPU with cryptographic TRNG coprocessor extension embedded into Altera FPGA[C] // 2013 23rd International Conference Radioelektronika (RADIOELEKTRONIKA), Pardubice, Czech Republic, 2013, pp. 136-140.

[61].Valtchanov, Boyan et al. Modeling and observing the jitter in ring oscillators implemented in FPGAs[C] // 2008 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems(2008): 1-6.

[62].Fischer, Viktor et al. Enhancing security of ring oscillator-based trng implemented in FPGA[C] // 2008 International Conference on Field Programmable Logic and Applications (2008): 245-250.

[63].Schindler, Werner and Wolfgang Killmann. Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications[J]. Workshop on Cryptographic Hardware and Embedded Systems(2002).

[64].Ma, Yuan et al. Entropy Evaluation for Oscillator-Based True Random Number Generators[J]. Workshop on Cryptographic Hardware and Embedded Systems(2014).

[65].Killmann, W., Schindler, W. (2008). A Design for a Physical RNG with Robust Entropy Estimators[C] // Oswald, E., Rohatgi, P. (eds) Cryptographic Hardware and Embedded Systems – CHES 2008. CHES 2008. Lecture Notes in Computer Science, vol 5154. Springer, Berlin, Heidelberg.

[66].Hajimiri A, Limotyrakis S and Lee T H. Jitter and phase noise in ring oscillators[J]. IEEE Journal of Solid-State Circuits, 1999, 34(6): 790-804.

[67].J. Wu, Y. Ma, J. Zhang, Y. Kong, H. Song and X. Han, Research on metastability based on FPGA[C] // 2009 9th International Conference on Electronic Measurement & Instruments, Beijing, China, 2009, pp. 4-741-4-745.

[68].Rukhin A, Soto J, Nechvatal J, et al. SP 800-22. A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications[M]. USA: National Institute of Standards & Technology, 2010: 1-79.

[69].Turan M S, Barker E, Kelsey J, et al. 800-90B. Recommendation for the Entropy Sources Used for Random Bit Generation, document, Gaithersburg[M]. USA: National Institute of Standards & Technology, 2018: 1-84.

[70].N. Nalla Anandakumar, S. K. Sanadhya and M. S. Hashmi, FPGA-Based True Random Number Generation Using Programmable Delays in Oscillator-Rings[J].

IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 3, pp. 570-574, March 2020.

攻读学位期间参与的学术活动及成果情况

[1]**叶新伟**,倪天明,吴昊,等.一种精度量化抖动的真随机数发生器设计[J/OL].微电子学,1-9[2025-04-30].https://doi.org/10.13911/j.cnki.1004-3365.240213.

致 谢

三年的硕士时光，像是被按下了快进键的电影，每一帧都在脑海中快速闪过，却又无比清晰。回首这段攻读硕士学位的旅程，我在工科科研的海洋中奋力前行，经历了无数次的挫折与迷茫，也收获了成长与蜕变。从最初面对复杂理论的困惑，到在实验中反复摸索，每一步都倾注了无数的心血，但也正是这些磨砺让我逐渐成长为一名成熟的科研人。

在此，我要向我的导师倪天明教授致以最诚挚的感谢。从论文选题的确定，导师便凭借其敏锐的学术洞察力，为我指明了研究方向。在论文撰写过程中，大到整体框架的搭建，小到标点符号的规范，导师都给予了细致入微的指导。每一次的修改意见，都饱含着导师对学术的严谨态度和对我的殷切期望。他不辞辛劳地审阅我的论文，耐心解答我在研究中遇到的各种难题，让我在科研的道路上少走了许多弯路。导师不仅是我学术上的引路人，更是我人生道路上的榜样，他的言传身教将永远激励着我不断前行。

我也要感谢我的两位小导师，卞景昌老师和聂牧老师。在实验复现工作中，他们凭借丰富的经验和专业知识，为我提供了宝贵的建议和技术支持，帮助我克服了一个又一个实验难关。在论文的审阅批改过程中，他们从不同的角度提出建设性意见，使我的论文得以不断完善。他们的帮助让我在科研的过程中更加自信，也让我深刻体会到了团队协作的力量。

实验室的同门师兄弟和师弟师妹们，是我这三年科研生活中最温暖的陪伴。我们一起讨论问题、分享实验心得，在彼此遇到困难时相互鼓励、互相帮助。还记得那些一起熬夜做实验的日子，虽然辛苦，但我们相互陪伴，共同坚持。在这个团结友爱的大家庭里，我感受到了浓浓的情谊，也收获了珍贵的友谊。我们共同成长，共同进步，这段经历将成为我人生中最宝贵的财富。

此外，我还要感谢我的家人。父母一直是我最坚实的后盾，他们默默付出，给予我无尽的关爱和支持，让我能够安心追求学业。你们的理解和鼓励是我前进的动力，让我在遇到困难时能够勇往直前。

最后，感谢本次论文评审老师在百忙之中审阅我的论文，你们的宝贵意见和建议对我至关重要 ，将帮助我在学术道路上继续成长。