

**АНАЛІЗ ТА РЕАЛІЗАЦІЯ СПРАВЖНІХ ГЕНЕРАТОРІВ
ВИПАДКОВИХ ЧИСЕЛ НА ОСНОВІ АРХІТЕКТУРИ FPGA**

Білоцерківець О.Г

e-mail: oleksii.bilotserkivets@nure.ua

Науковий керівник – к.т.н., проф. Воргуль О.В.

Харківський національний університет радіоелектроніки, каф. МІРЕС
м. Харків, Україна

True random number generators (TRNGs) are essential for cryptographic systems, secure communication protocols, and stochastic modeling. Unlike pseudo-random generators, TRNGs obtain randomness from physical processes, ensuring higher unpredictability. This paper presents the analysis and implementation of a TRNG based on field-programmable gate array (FPGA) architecture. The proposed design uses ring oscillators as entropy sources, exploiting phase jitter caused by intrinsic noise in silicon structures. To increase entropy quality, outputs from multiple oscillators are combined using an XOR-based post-processing stage. Special synthesis constraints are applied to preserve combinational loops during hardware implementation. The generated bitstreams are evaluated using statistical tests from the NIST SP 800-22 suite. The results confirm that FPGA-based TRNGs can produce high-quality random numbers suitable for cryptographic applications.

Генерація випадкових чисел є фундаментальною складовою криптографічної безпеки, захищених протоколів зв'язку та стохастичного моделювання. У цифрових системах розрізняють два основні типи генераторів випадкових чисел: генератори псевдовипадкових чисел (PRNG) та генератори справжніх випадкових чисел (TRNG).

Генератори PRNG використовують детерміновані алгоритми, що розширюють коротке початкове значення у довгу послідовність бітів. Проте такі послідовності можуть бути передбачуваними у випадку компрометації внутрішнього стану алгоритму. На відміну від них, генератори TRNG отримують випадковість безпосередньо з фізичних процесів, що забезпечує значно вищий рівень непередбачуваності та робить їх особливо важливими для систем з високими вимогами до безпеки.

Програмовані вентильні матриці (FPGA) є ефективною платформою для реалізації TRNG завдяки їх реконфігурованій архітектурі та наявності численних джерел фізичного шуму в кремнієвій структурі.

Надійність генератора TRNG на основі FPGA значною мірою визначається якістю джерела ентропії. Найбільш поширеними джерелами випадковості в інтегральних схемах є: тепловий шум, дробовий шум, мережливий шум.

Тепловий шум, також відомий як шум Джонсона–Найквіста, виникає внаслідок хаотичного теплового руху електронів у провідниках, що при-

зводить до випадкових флуктуацій напруги. У ресурсах FPGA ці мікроколивання можуть впливати на пороги перемикання логічних елементів, таких як таблиці пошуку (LUT) та тригери.

Іншим важливим явищем є метастабільність. Вона виникає у випадку порушення вимог до часу встановлення або утримання сигналу тригером. У такій ситуації вихід тригера може тимчасово перебувати у нестабільному проміжному стані між логічними значеннями «0» та «1», перш ніж встановиться у стабільний стан. Час переходу з метастабільного стану значною мірою залежить від випадкових шумових факторів, що робить це явище цінним джерелом ентропії.

Однією з найпоширеніших архітектур TRNG для FPGA є генератори на основі кільцевих осциляторів. Кільцевий осцилятор складається з непарної кількості інверторів, з'єднаних у замкнений контур зворотного зв'язку. Через відсутність стабільної робочої точки схема починає коливатися з частотою, яка визначається затримками поширення сигналу в логічних елементах і міжз'єднаннях. Основним джерелом випадковості у таких системах є фазовий джиттер. Джиттер представляє собою короточасні відхилення моментів переходу цифрового сигналу від їх ідеального положення в часі.

У FPGA ці відхилення накопичуються під час проходження сигналу через петлю зворотного зв'язку осцилятора. На величину джиттера впливають різні фактори, зокрема: флуктуації напруги живлення, перехресні перешкоди між сусідніми сигналами, температурні градієнти на кристалі.

У результаті період коливальності осцилятора T не є сталим і змінюється у часі:

$$T = T_0 + \Delta T(t)$$

де T_0 – номінальний період, а $\Delta T(t)$ – випадкова складова, обумовлена шумом. Якщо сигнал такого осцилятора дискретизувати стабільним незалежним тактовим сигналом системи, фазова невизначеність призводить до випадкових змін значення біта на виході.

Використання одного кільцевого осцилятора може призводити до небажаних ефектів, таких як частотне блокування або інжекційне блокування. У цьому випадку частота осцилятора синхронізується із тактовим сигналом дискретизації або його субгармонікою, що значно знижує рівень ентропії.

Для уникнення цього ефекту застосовується архітектура з декількома осциляторами. У такій системі використовуються кілька кільцевих осциляторів з різною кількістю інверторних ступенів. Оскільки довжини кілець є взаємно простими, фазові співвідношення між сигналами осциляторів стають складними та неперіодичними.

Виходи кількох незалежних джерел ентропії об'єднуються за допомогою XOR-дерева. Операція XOR виконує роль простого концентратора ентропії: якщо хоча б один із вхідних сигналів є справді випадковим і ста-

тистично незалежним, вихідний сигнал також буде випадковим. Така апаратна постобробка дозволяє зменшити статистичне зміщення, яке може виникати в окремих осциляторах.

Реалізація TRNG потребує спеціального керування процесом синтезу. Стандартні інструменти автоматизованого проєктування, такі як Vivado або Quartus, намагаються оптимізувати логіку, усуваючи надлишкові елементи та комбінаторні цикли. Однак у випадку TRNG саме ці цикли є ключовим елементом роботи генератора.

Для збереження структури осцилятора застосовуються спеціальні атрибути синтезу. Наприклад, атрибут “dont_touch” забороняє оптимізацію або об’єднання логічних елементів у ланцюгу осцилятора.

Крім того, можуть використовуватися обмеження розміщення, які задають конкретне розташування логічних блоків на кристалі FPGA. Це дозволяє забезпечити симетричну структуру осциляторів та ізолювати їх від високошвидкісних або шумових компонентів системи.

Завершальним етапом розробки TRNG є перевірка статистичних властивостей згенерованих бітових послідовностей. Оскільки людина не може надійно оцінити випадковість послідовності, застосовуються стандартизовані набори тестів.

Одним із найпоширеніших є набір тестів NIST SP 800-22. Він включає декілька статистичних перевірок, серед яких: Frequency Test – перевіряє рівномірність розподілу нулів та одиниць; Runs Test – аналізує довжини послідовностей однакових бітів; Approximate Entropy Test – оцінює частоту появи перекривних бітових шаблонів.

Успішне проходження цих тестів підтверджує, що генератор здатний формувати високоякісні випадкові послідовності, придатні для криптографічних застосувань, зокрема для генерації ключів.

Генератори справжніх випадкових чисел на основі FPGA та кільцевих осциляторів забезпечують ефективний компроміс між складністю апаратної реалізації та якістю отримуваної ентропії. Глибоке розуміння фізичних джерел шуму та правильне керування процесом синтезу дозволяє інженерам створювати надійні автономні генератори випадкових чисел без необхідності використання зовнішніх апаратних компонентів.

Список використаних джерел:

1. Sunar B., Martin W., Stinson D. A provably secure true random number generator with built-in tolerance to active attacks // IEEE Transactions on Computers. 2007. Vol. 56, № 1. P. 109–119.
2. National Institute of Standards and Technology. A statistical test suite for random and pseudorandom number generators for cryptographic applications : NIST Special Publication 800-22. Gaithersburg : NIST, 2010. 131 p.
3. Altera Corporation. True random number generator design in FPGA devices : Application Note. San Jose : Altera Corporation, 2014. 18 p.