Hardware implementation of random prime number generator

Chi Trung Ngo, Van Khanh Pham, Sang Tran, Jong-Phil Hong

School of Electrical Engineering, Chungbuk National University

trung@chungbuk.ac.kr, pvkhanh@chungbuk.ac.kr, sang@chungbuk.ac.kr,

jphong@chungbuk.ac.kr

랜덤 소수 생성기의 하드웨어 구현

오치충, 팜반카인, 트랑상, 홍종필

충북대학교 전기공학부

요 약

In this work, we propose a hardware architecture for prime number generation that combines the Sieve of Eratosthenes with the Miller-Rabin primality test in a unified design framework. The proposed architecture has been synthesized using a 28nm process, operating at a frequency of 40MHz. Our design has a power consumption of 3mW. In terms of area occupation, our implementation costs are 148k GEs.

Ⅰ. Introduction

Prime numbers play a crucial role in a wide range of fields, including mathematics, computer science, and cryptography. In particular, they form the foundation of many cryptographic algorithms. For instance, the RSA algorithm [1] relies on large prime numbers to generate secure keys, with its security depending on the computational difficulty of factoring large composite numbers. Furthermore, prime numbers underpin key exchange protocols such as Diffie-Hellman and are also used in digital signature schemes to ensure message authenticity and integrity. Motivated by the wide range of cryptographic applications and the inherent limitations of conventional approaches, this work presents a novel prime number generator that integrates the Sieve of Eratosthenes with the Miller-Rabin primality test [2] into a unified framework. By leveraging the early elimination capability of the sieve and the high reliability of probabilistic verification, the proposed method achieves a significant reduction in computational time compared to traditional standalone techniques.

Ⅱ. Method

Prime number generation is conventionally performed through deterministic sieve algorithms or probabilistic primality testing as shown in Fig. 1. Deterministic sieves, such as the Sieve of Eratosthenes and the Sieve of Atkin, provide efficient enumeration of primes within bounded ranges but exhibit prohibitive memory and computational complexity when extended to cryptographically significant magnitudes. Conversely, probabilistic methods, including randomized search with Miller-Rabin testing and incremental search strategies, scale more effectively to large integers but suffer from substantial computational overhead due to repeated primality evaluations. The inherent inefficiency of employing either class of algorithms in isolation renders them suboptimal for large-scale prime generation in cryptographic contexts. To mitigate these limitations, this work introduces a hybrid methodology that integrates sieve-based preselection with Miller-Rabin verification, thereby reducing candidate space while maintaining high probabilistic assurance of primality, ultimately improving both asymptotic efficiency and practical scalability.

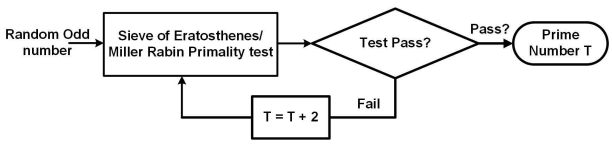

Fig. 1 Conventional prime number generation method.

Figure 2 illustrates the block diagram of the proposed prime number generation system, which operates on random number inputs and comprises three primary functional blocks: Initial Modification, Sieve of Eratosthenes (SoE), and the Miller-Rabin Primality Test. This work combie The Initial Modification stage enforces structural constraints by adjusting the most significant bit (MSB) and least significant bit (LSB) to guarantee the required numerical magnitude while ensuring that the candidate is odd. The SoE stage performs divisibility checks against a predefined set of small prime numbers, efficiently eliminating composite candidates at an early stage. Although this step significantly reduces the computational burden on subsequent primality testing, its implementation cost increases with the size of the prime set due to memory overhead associated with storing large divisors. Finally, the Miller-Rabin Primality Test is employed as the conclusive verification stage, offering probabilistic primality assurance

with an error probability lower than 2E-80, thereby ensuring both reliability and efficiency in large prime generation.

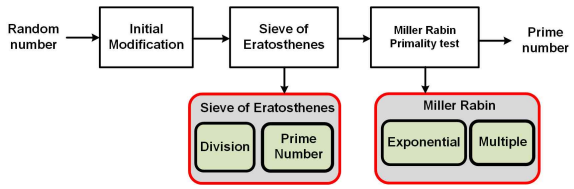

Fig. 2 Block diagram of proposed prime number generator system.

The flowchart of the proposed prime number generation system is presented in Fig. 3. The process begins with a random number obtained from the TRNG, which is modified by setting both the MSB and the LSB to one, thereby ensuring that the candidate is an odd number of the desired magnitude. The modified number is then subjected to the SoE, where divisibility is checked against the first 53 prime numbers. While increasing the number of small primes improves the filtering efficiency, it also introduces significant area overhead due to storage requirements. If the candidate passes the SoE test, it proceeds to the subsequent stage; otherwise, its value is incremented by two to preserve odd parity, and the SoE test is repeated. Candidates that successfully pass this preliminary screening are finally verified using the Miller-Rabin primality test, which provides probabilistic assurance of primality with a negligible error rate. The Miller-Rabin primality test is a probabilistic algorithm derived from the principles of the Fermat primality test. Let n be an odd integer expressed as $n = 1+d.2^e$, where d is odd. A random integer a, with $1 < a < n$, is selected as a test base. The number n is classified as a probable prime if either $a^d \equiv 1 (\mathrm{mod}\, n)$ or $a^{2^r.d} \equiv -1 (\mathrm{mod} n)$ holds for some $0 \leq r < e$. If neither condition is satisfied, n is declared composite. In the case where the conditions hold, n may either be prime or a composite number that behaves like a prime with respect to the chosen base a; in the latter situation, n is referred to as a strong pseudoprime to base a, and a is considered a non-witness. Conversely, when a demonstrates that n fails the test, it is termed a witness to the compositeness of n.

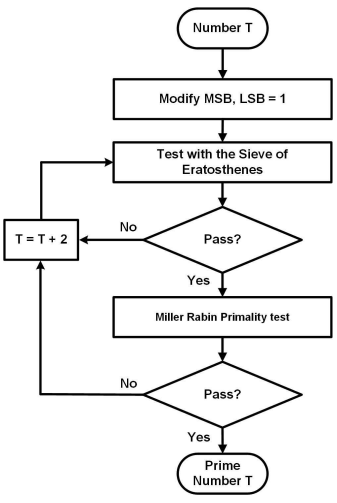

Fig. 3 System flowchart diagram.

If the number fails the test, two is added to it, and the process returns to the initial test to repeat all subsequent evaluations. If the number passes the test, it is identified as a prime number.

	This work
Technology	28
Input sequence (bits)	512
Frequency (MHz)	40
Area (GE)	148747
Power (mW)	3.03

Table I Performance summary of the proposed system..

Table I presents the hardware performance of the proposed prime number generator. The design has been validated and synthesized using a 28nm process technology, operating at a clock frequency of 40?MHz. The implementation requires 148,747GE and consumes 3.03mW of power. To evaluate the improvement, several random odd numbers were tested; for example, for the 64-bit hexadecimal number 0x43FE04DCB43295BB, the generation of a prime number using only the Miller-Rabin primality test required 5,612,500 clock cycles. In contrast, the proposed system achieves the same result in only 1,812,500 clock cycles, representing an improvement of approximately 300% in generation time.

Ⅲ. Conclusion

This paper proposed a hardware architecture of prime number generator by using the SoE and Miller Rabin primality test algorithm. The proposed architecture is synthesized in a 28nm process with an operating frequency of 40MHz. The synthesis of our design has yielded a power consumption of approximately 3mW. Additionally, our implementation cost at only 148,747 GEs in terms of area occupation.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 지역지능화혁신인재양성사업임(IITP-2025-RS-2020-II201462). 이 논문은 2025년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. RS-2020-NR049604)

참 고 문 헌

[1] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 2 (Feb. 1978), 120-126.

[2] M. E. O'Neill, "The Genuine Sieve of Eratosthenes," J. Funct. Program., vol. 19, no. 1, pp. 95-106, Jan. 2009.

[3] Rabin, M. (1980). Probabilistic algorithm for testing primality. Journal of Number Theory, 12 (1), 128-138.