Project proposal for Cpr E 583

Blake Vermeer, Piriya Kris Hall, and Rohit Zambre

***Abstract*—The abstract goes here.**

***Index Terms*—IEEEtran, journal, LaTeX, paper, template.**

I. Introduction

THE discovery and verification of prime numbers is crucial for security systems.

II. Core Problems

After researching the topic of prime number generation and verification, we have identified three main core problems that need to be solved:

A. Generation of Cunningham Chains

B. Generation of Prime Numbers

C. Verification of Generated Prime Numbers

Since the prime number generators work on the principal of generating numbers that has a reasonably high probability of being prime, the numbers must be verified to insure that they are actually prime. This is a very resource intensive problem since it requires testing if the potential prime number number is evenly divisible by a set of numbers. Since it is prohibitively expensive to test if large prime numbers are definitively prime, probabilistic methods have been developed that will determine if a number is prime with a high probability. One of these probability methods of determining if a number is prime is called the Rabin-Miller primality test.

1) Rabin-Miller Primality Test: The Rabin-Miller primality test is a probabilistic method of determining if a number is prime. The algorithm is based around the idea of testing if a number in question is divisible by a subset of small primes. If it is not divisible by any of the small primes in the set then it is a prime number with a probability that is determined by the number of small primes in the set. The larger the set of small prime numbers is allows for the primality of the number in question to be determined with greater confidence. Since the probability of correctly identifying a prime number is directly related to the number of small primes tested again the test number, the accuracy of the Rabin-Miller primality test can be easily customized. [1]

Now to dive into the theory of the Rabin-Miller primality test. The Rabin-Miller theory is based on the contrapositive of Fermat's little theorem which states that for a prime number n:

$$a^{n-1} \equiv 1 \pmod{n}$$

Therefore, the Rabin-Miller primality test can show that a number is not positive of for a number n if we can find a number a such that:

$$a^d \not\equiv 1 \pmod{n}$$

and

$$a^{2^r d} \not\equiv -1 \pmod{n}$$

for all $0 \leq r \leq s-1$, then n is not prime. [2]

The Rabin-Miller Primality test is a good test to implement on hardware since the test can be parallelized by testing many small primes again the number in question concurrently and the math operations involved would be fairly expense in terms of cycles if implemented in software because of its sequential nature. Since we are attempting to implement this design in hardware let's first examine how to do the modulus operation in hardware.

One way to implement a modulus operation in hardware is to use the Montgomery algorithm to perform a modulo-multiply.

III. Conclusion

The conclusion goes here.

References

[1] Cheung, R., Brown, A., Luk, W., Cheung, P.: A scalable hardware architecture for prime number validation. In: IEEE Int. Conf. on Field-Programmable Technology, pp. 177-184 (2004)

[2] A. Daly and W. Mamane. Efficient architectures for implementing montgomery modular multiplication and RSA modular exponentiation on reconfigurable logic. In *Tenth ACM International Symposium on Field-Programmable Gate Arrays* pages 40-49, February 2002.

[3] Miller-Rabin Primality Test. On: *Wikipedia*, 27 Oct. 2014. <http://en.wikipedia.org/wiki/Miller%E2%80%93Rabin_primality_test>

[4]

[5]

[6]

[7]

[8]

[9]

[10]