FPGA implementation of ECDSA for Blockchain

Shoi Tachibana†, Shunsuke Araki†, Seiji Kajihara†, Shigeyuki Azuchi‡, Yukishige Nakajo‡, Hideki Shoda‡

†Kyushu Institute of Technology, Iizuka, Japan
‡Chaintope Inc., Iizuka, Japan

ABSTRACT

In this paper, we address Field Programmable Gate Array (FPGA) implementation of the Elliptic Curve Digital Signature Algorithm(ECDSA), which is suitable for cryptocurrencies in blockchain. Although the ECDSA requires high computational efforts, utilization of a specific logic circuit allows us quick and easy computation. In this paper, we give implementation results of the ECDSA on an FPGA, and mention the superiority of FPGA implementation by comparing its simulation result with that of software based computation using a CPU.

INTRODUCTION

Blockchain technology that is applicable to cryptocurrencies or financial trading, has been spreading rapidly. In almost all cryptocurrencies such as Bitcoin record, transactions can be executed in a blockchain. The transactions must be signed with digital signatures, and then verified by all nodes. After verification, valid transactions are stored by embedding them in the blockchain. Because the nodes must verify all digital signatures for valid/invalid transactions, speeding up the process of the digital signature is one of important issues for the blockchain.

The Elliptic Curve Digital Signature Algorithm (ECDSA) is widely used as a digital signature method to guarantee the validity of the transactions in the blockchain. There are some methods for speeding up the process of the digital signature with ECDSA, e.g., high-speed calculation of scalar multiplication over an elliptic curve and hardware implementation of calculation over the elliptic curve, and so on. Among others, the hardware implementation can be performed more efficiently not only in calculation time but also in terms of power consumption as compared with a CPU.

In this paper, we address implementation of ECDSA calculation on an FPGA and give evaluation results on the calculation time of signature verification process, in which it is shown FPGA implementation is superior to software based calculation with a CPU.

BLOCKCHAIN AND HARDWARE IMPLEMENTATION

In ordinary blockchains, correctness of a transaction is guaranteed by verifying the digital signature [1][2]. This verification process is carried out as soon as the participating nodes which are called “Full Nodes” in Bitcoin receive the transaction. So it is desirable that these nodes have a suitable processing ability which can verify many signatures. In other words, it is hard that low-resource devices such as a wireless LAN access point become the full node. There are some solutions to compute them efficiently such as employing a digital signature algorithm with fast signature generation/ verification and developing fast basic arithmetic units, namely a modular exponentiation and an elliptic curve calculation. An alternative approach is hardware implementation of signature calculation. A development of Application Specific Integrated Circuits (ASICs) for mining is a well-known approach with respect to high-speed operation by hardware, but the development cost is very high. Since even the most popular Bitcoin has at most 10,000 full nodes, the solution by ASICs will be unprofitable. Additionally, changing a digital signature algorithm and an elliptic curve means replacement by a new ASIC or a new device.

ECDSA

All transactions in the blockchain must be signed by any digital signature algorithm. The ECDSA which is one of widely used algorithms is a Digital Signature Algorithm (DSA) over elliptic curves. In the rest of this paper, we focus on verifications of the ECDSA, because one node must process many transactions.

Let n be the order of the base point G of the elliptic curve. A signature (r, s) for a message m by the ECDSA must satisfy that $r, s \in [1, n\text{-}1]$. In the verification of the ECDSA, we derive a point over the elliptic curve by the following equation:

$$(x_1, y_1) = u_1 G + u_2 G,$$

where $u_1 = z/s \bmod n$, $u_2 = r/s \bmod n$ and z is the $|n|$ high-order bits of the hash value $H(m)$. Last, we confirm $r \equiv x_1 \pmod n$ for the valid signature. In our research we implemented this verification of the ECDSA over Secp256k1, which is the parameter of the elliptic curve, on a hardware.

CIRCUITIZED FUNCTIONS FOR ELLIPTIC CURVE COMPUTATIONS

In order to realize a high-speed processing of the ECDSA by hardware implementation, we must design circuit modules of not only elliptic curve computations but also fundamental arithmetic operations for 256-bit integers. Specifically, the functions of the modules are addition and doubling, modulo arithmetic, and so on. These modules are defined individually, but they are almost the same as functions in common programing languages. For a modular multiplicative inverse calculation, it is realized by using extended Euclidean algorithm.

In addition to those described above, we must design a circuit for elliptic curve computations. In essence, a scalar multiplication over the elliptic curve is equal to additions of the point over the elliptic curve for a coefficient number. Of course, so long processing time would be required. In general, a double-and-add method would be used as the scalar multiplication over the elliptic curve in order to shorten this calculation time.

BLOCKCHAIN AND FPGA IMPLEMENTATION

If any full node receives a boost from an accelerator of elliptic curve computation, namely a hardware device, we would be able to regard the devise with low resources as a full node. As shown in the previous section, however, due to low

volume efficiency and changing elliptic curves and digital signature algorithms, ASICs are not suitable for this purpose. Therefore, we would employ an FPGA for fast processing of the digital signature algorithm.

The FPGA realizes a logic circuit described as a program in Hardware Description Language (HDL), and a user can freely design the circuit easily and reconfigure it by rewriting the program. So there is an advantage that the development period can be reduced and the development cost is low compared with ASICs. In the circuit, several operations can be performed in parallel, the process can be divided into some sub-process and the sub-process can be performed at the same time. The parallelism of calculation results in higher performance than software-based calculation with a CPU which cannot parallelize the process. Therefore, employing the FPGA as the accelerator can be a reasonable solution for speed-up.

FPGA IMPLEMENTATION FOR ECDSA

In this section, we discuss about key points of FPGA design for implementing the ECDSA. The points are below:

1. Circuit area

Different from software, hardware implementation needs to be considered circuit size because the FPGA has limited logic elements on the chip. There are some logic functions which are used repeatedly in one process. It is important for reduction of the circuit size to share one circuit block with several operations by changing its inputs.

2. Circuit performance to minimize operation timing

An FPGA operates with system clocks, and logic circuits are executed with the clocks at the same time. For fast calculation, it is necessary to operate as much logics as possible simultaneously. Circuit performance can go up by preparing enough computational resource, while circuit size would be large. Actually calculation time of less circuit is slower, because the repeated use of the same module causes some calculations to wait for their turns.

3. State machines

A state machine is implemented as a synchronous sequential circuit that determines the next state from primary inputs and the current state. It is important for fast calculation to optimize state transition and state assignment.

RESULT AND DISCUSSION

For evaluation using an FPGA device, we supposed Intel Cyclone IV whose operating frequency is 50MHz and prepared all values used for verifications for measuring the operating time, and executed logic simulation in our experiment. Besides, for comparison of the calculation time, we implemented the algorithm in a software running on a PC with Intel Core i7-7700 CPU whose operating frequency is 3.60GHz. A processing would be performed at 20 nsec per clock. In our experiment, we implemented the verification scheme of the ECDSA and evaluated the calculation time by logic simulation for the FPGA. Since the logic simulation is based on only the logic structure of the circuit and inputs values, it does not depend on the execution environment. In this implementation of the ECDSA, no special method such as a Montgomery modular multiplication has been used, and adjustment is made so that the operation per clock can be performed without waste.

Figure 1 shows output values of the calculated coordinates by the logic simulation and expectation value and calculation time. Table 1 gives the result of calculation time. The calculation was completed in 145.52 ms, and the outputs were matched with expected ones. In addition, the calculation time using the CPU was measured only for the real time when the program was running with the program written in ruby. The calculation time of the CPU is 142.99 ms, and the calculation time of the FPGA is longer than the calculation time of the CPU in our research. However, in the case of Bitcoin, since the number of transactions processed per second is about 6 to 7, practical use is also possible with this performance. In addition, the basic algorithm of the ECDSA is directly implemented on the FPGA in the current experiment. Because acceleration techniques for the ECDSA have been devised, improvement of the performance could be expected. Moreover, the clock frequency of the FPGA is 50 MHz, while the clock frequency of the CPU is 3.60GHz. Since power consumption necessary for computation is proportional to the clock frequency, the computation using the FPGA could be done with much less power than the CPU. Because low power consumption is desired especially for edge computing or IoT devices, this is a major advantage of FPGA implementation.

Figure 1: the computing time of logic simulation

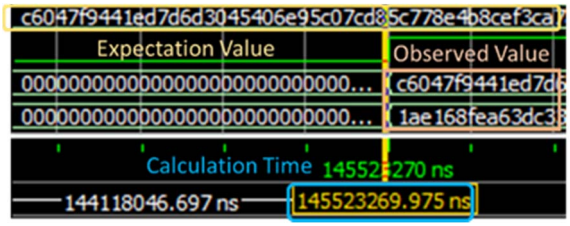

Table1: comparison of calculation time

device	frequency	calculation time
FPGA: Intel Cyclone IV	50 MHz	145.52 ms
CPU: Intel Core i7-7700	3.6 GHz	142.99 ms

CONCLUSION

In order to make it possible to realize a part of operations on the full node of the blockchain with a low resource device, we proposed the ECDSA computation on an FPGA, and confirmed that the computation time by the FPGA is comparable to the time by a CPU while the FPGA computation would have an advantage of low power consumption. Our future works are to estimate electric requirements with real devices and to employ high efficient computational methods as fast elliptic curve computations.

REFFERENCES

[1] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto Currencies*, NTT Publishing Co., Ltd., 2017 (in Japanese).

[2] S. Yamasaki, S. Azuchi, and S. Tanaka, *Blockchain Programming Introduction to Cryptocurrency*, Kodansha Ltd., 2017 (in Japanese).

[3] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, https://bitcoin.org/bitcoin.pdf (last accessed 31 Jan. 2019)