DATE : 2016/1/8

Department of Electrical and Electronic Information Engineering		ID	M123204
Name	Yusuke Ayuzawa		

Supervisor	Shuichi Ichikawa Naoki Fujieda

Abstract

Title	Acceleration of Modular Exponentiation by High-Radix Montgomery Multiplication

It is necessary to hide information when we deal with personal information, electronic money, and so on. The information is hidden by encryption such as RSA (Rivest, Shamir, Adleman), which is widely used for public key encryption. Since RSA relies on modular exponentiation, software processing with a general-purpose processor takes long time and consumes large electrical energy. Thus, there has been a number of studies on dedicated hardware for modular exponentiation.

If we perform modulo operation after calculating the exponentiation, the cost for the operations becomes large because of enlarged bit width. Binary method is widely used to solve this problem. The left-to-right binary method starts the calculation from the most significant bit, while the right-to-left binary method starts from the least significant bit. In the binary methods, the bit width stays constant by taking remainders during exponentiation.

Montgomery multiplication was proposed in 1985 by Montgomery. It calculates modular multiplication quickly using bit shift and bit mask. A high-radix Montgomery multiplication is often used with a binary method.

This paper examines two speedup techniques for the high-radix Montgomery multiplication on FPGA (Field Programmable Gate Array), which is a reconfigure logic device. The first method is the use of the right-to-left binary method. Left-to-right binary method has been used to reduce a logic scale in past studies. Though the right-to-left binary method requires more logic elements, it performs modular multiplication in parallel. Right-to-left binary method is adapted to Montgomery multiplication (Method 1). The second method is the use of multiple clock signals for the high-radix Montgomery multiplication. The module is separated by bit width and each part is driven by an individual clock. A register is added to reduce the number of comparison. It performed modulo and division by the proper connection between modules (Method 2). These two methods and their combination (Method 1+2) are evaluated.

The evaluation measure is the AT (Area-Time) product. The worst-case execution time is used for the time, while the number of slices is used for the area. The number of DSP (Digital Signal Processing) unit and BRAM (Block RAM) are included in the area using a conversion formula. The conversion formula is based on the area ratio that is observed in the floor planners of the CAD (Computer Aided Design) tool. The execution time became the half of the existing method in Method 1 at the cost of max. 87.9 % increase of LUT (Look Up Table) and registers. Method 2 was 25.1 times faster than the existing method without additional LUT and registers. Method 1+2 sped up the calculation by 50.8 times. Though Method 1+2 became larger than three precedent studies in the area, it was the smallest in the AT product.