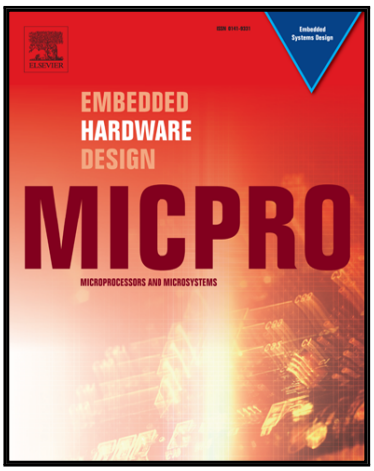

A Fast Digit Based Montgomery Multiplier Designed for FPGAs with DSP Resources

Erdem Özcan, Serdar S. Erdem

PII: S0141-9331(17)30458-1
DOI: 10.1016/j.micpro.2018.06.015
Reference: MICPRO 2715

To appear in: *Microprocessors and Microsystems*

Received date: 11 October 2017
Revised date: 13 April 2018
Accepted date: 22 June 2018

Please cite this article as: Erdem Özcan, Serdar S. Erdem, A Fast Digit Based Montgomery Multiplier Designed for FPGAs with DSP Resources, *Microprocessors and Microsystems* (2018), doi: 10.1016/j.micpro.2018.06.015

A Fast Digit Based Montgomery Multiplier Designed for FPGAs with DSP Resources

Erdem Özcan[a,b], Serdar S. Erdem[a,*]

[a] *Department of Electronics Engineering, Gebze Technical University, P.K: 141, 41400 Gebze, Kocaeli, Turkey.*
[b] *Informatics and Information Security Research Center (BILGEM), 41470, Gebze, Kocaeli, Turkey.*

Abstract

A fast Montgomery multiplier design utilizing the DSP resources in modern FPGAs is presented. In the proposed design, the operand size is the multiples of 528 bits and the digit size is 48 bits. The design has 48×48 bit digit multipliers built from the DSP slices performing 24×16 bit multiplications and a carry select accumulator built from the DSP slices performing 48 bit additions. The proposed Montgomery multiplier works iteratively. In each iteration, a digit of an operand is multiplied by the digits of the other, the result is accumulated, and reduced by Montgomery method. An iteration takes not one but eight cycles to keep the digit multiplier count low and save some hardware resources. The proposed design is implemented for Virtex-7 FPGAs. The performance results are comparable with the best results in the literature. Substantial savings in FPGA logic resources are obtained.

Keywords: Montgomery modular multiplication, carry-select addition, FPGA, DSP, RSA cryptosystem

1. Introduction

The Public key cryptosystems Diffie Hellman [1], RSA [2], and elliptic curve cryptography [3, 4] are widely used in today's communication systems. Modular multiplication with large integers is the main computation in these cryptosystems. Straightforward modular multiplication is a costly operation since it requires division. The Montgomery algorithm [5] is an efficient modular multiplication technique trading the costly division for a multiplication with a precomputed value depending on the divisor. Modular multiplication with Montgomery algorithm is briefly called Montgomery multiplication. Montgomery multiplication with a divisor θ needs the precomputation

$$\psi = -\theta^{-1} \bmod 2^{\delta}$$

where δ is the digit size of the multiplicands. In Montgomery multiplication, one multiplicand is multiplied with the digits of the other multiplicand, one digit at a time. After each multiplication, the partial product is accumulated and reduced by using the precomputation ψ.

Then, at least n/δ clock cycles are needed to complete a Montgomery multiplication where n is the multiplicand bit length. Obviously, the multiplier latency decreases as the digit size δ gets larger. However, the multiplier area increases too. This area increase is very significant in cryptographic applications where computations with several hundred or even thousand bit numbers are common.

Bit serial Montgomery multipliers (digit size $\delta = 1$) are abundant in the literature [6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. These bit serial architectures process the multiplier bits, $\delta = 1$ bit at a time. In cryptographic applications, the divisor θ is typically an odd integer. Thus, the precomputation $\psi = -\theta^{-1} \bmod 2^{\delta} = 1$ always when $\delta = 1$. Though the bit serial multipliers have low area complexity, their latencies are high, they need at least $n/\delta = n$ clocks cycles to perform a single modular multiplication.

Digit based Montgomery multipliers (digit size $\delta > 1$, precomutation $\psi = -\theta^{-1} \bmod 2^{\delta}$) are also proposed in the literature to obtain speed improvement at the expense of area [13, 16, 17, 18, 19]. A multiplier with $\delta = 2$ is presented in [13] and another one with $\delta = 3$ is presented in [16]. Also, the works in [17, 18] propose multiplier designs which can be implemented for arbitrary digit sizes δ. However, the work in [17] puts a serious restriction on the divisor θ such that the precomputation $\psi = -\theta^{-1} \bmod 2^{\delta} = \pm 1 \bmod 2^{\delta}$. Also, the Montgomery multiplier in [19] uses a method based on canonic signed digit recoding and converts one of the operands into a sparse integer representation to speed up the multiplication.

The FPGAs are important implementation platforms. They can be used to implement any hardware design by their programmable logic blocks and routing. Moreover, modern FPGAs have a significant number of DSP blocks, which have advanced arithmetic capabilities. These DSP blocks commonly have 18×18 or 25×18 or 36×36 bit multipliers. Thus, one or more DSP blocks can be configured to implement a $\delta \times \delta$ bit multiplier. Such a multiplier is usually used to multiply the δ bit digits of the large operands

*Corresponding author
Email addresses: erdemozcan@gtu.edu.tr (Erdem Özcan), serdem@gtu.edu.tr (Serdar S. Erdem)