FPGA implementation of variable precision Euclid’s GCD algorithm

ABSTRACT

Introduction: Euclid's algorithm is well-known for its efficiency and simple iterative to compute the greatest common divisor (GCD) of two non-negative integers. It contributes to almost all public key cryptographic algorithms over a finite field of arithmetic. This, in turn, has led to increased research in this domain, particularly with the aim of improving the performance throughput for many GCD-based applications. Methodology: In this paper, we implement a fast GCD coprocessor based on Euclid's method with variable precisions (32-bit to 1024-bit). The proposed implementation was benchmarked using seven field programmable gate arrays (FPGA) chip families (i.e., one Altera chip and six Xilinx chips) and reported on four cost complexity factors: the maximum frequency, the total delay values, the hardware utilization and the total FPGA thermal power dissipation. Results: The results demonstrated that the XC7VH290T-2-HCG1155 and XC7K70T-2-FBG676 devices recorded the best maximum frequencies of 243.934 MHz down to 39.94 MHz for 32-bits with 1024-bit precisions, respectively. Additionally, it was found that the implementation with different precisions has utilized minimal resources of the target device, i.e., a maximum of 2% and 4% of device registers and look-up tables (LUT’s). Conclusions: These results imply that the design area is scalable and can be easily increased or embedded with many other design applications. Finally, comparisons with previous designs/implementations illustrate that the proposed coprocessor implementation is faster than many reported state-of-the-art solutions. This paper is an extended version of our conference paper [1].

Keyword: Digital arithmetic; FPGA; Integrated circuit synthesis; Euclid's algorithm; GCD