Abstract

UNTIL RECENTLY restricted to the prototyping of digital circuits, FPGAs (Field-Programmable Gate Arrays) are today capable, for some applications, of competing in performance with ASICs (Application-Specific Integrated Circuits). The latest families of FPGAs contain, for example, logic cells conceived specifically to optimize the implementation of arithmetic operators and synchronous memories to simplify table-based computation. Moreover, FPGAs can be more cost-efficient than ASICs for small series and can exploit better fabrication technologies. The conception of high-performance systems requires nevertheless a considerable effort in order to efficiently exploit the resources of the FPGA.

We illustrate these issues through the study of a series of coprocessors dedicated to the implementation of the RC6 and IDEA encryption algorithms. The main difficulty of the RC6 algorithm resides in the computation of the function $f(X) = (X \cdot (2X + 1)) \bmod 2^w$, where X is a w-bit unsigned number. We propose an operator that exploits the architectural features of Xilinx's Virtex family of FPGAs to efficiently evaluate $f(X)$. For 32-bit operands, our circuit is 55 % more compact and 18 % faster that the circuit proposed by NSA. CryptoBooster, our first prototype proposed for IDEA, achieves a throughput of 102 Mbits/s on a XCV1000-4 FPGA of the Virtex family. However, exploiting the multimedia instructions of Intel's and HP's latest processor, Itanium, allows the latter to outperform our circuit. This observation motivated us to study in more detail the arithmetic details exploited by the IDEA encryption algorithm. The throughput of CryptoBooster II, the system we developed following this analysis, exceeds 3Gbits/s on the same FPGA. Since all of IDEA's arithmetic operations are executed in LSDF (Least Significant Digit First) mode, a serial processing unit can be developed by interconnecting and synchronizing the necessary operators. The system thus obtained, called CryptoBooster III, implements a digit-level pipeline.

If a serial system requires operations in MSDF (Most Significant Digit First) mode, such as division or square-root, the insertion of registers to memorize intermediate results between circuits operating in different transmission modes becomes necessary. These constraints reduce the throughput of the system and complicate its control. A redundant representation of numbers allows the propagation of the carry in a sum to be suppressed, and consequently the serial evaluation of any function in MSDF mode. This approach to computation, called on-line arithmetic, has been shown to be very efficient for applications such as embedded control, genome sequence analysis, or artificial neural networks.

M. Ercegovac, the father of on-line arithmetic, has defined a set of methods for the conception of on-line operators for arithmetic and algebraic functions. We propose a new approach for the on-line computation of some elementary functions ($\sin(x)$, $\cos(x)$, e^x, ...), based on table look-ups and on the evaluation of polynomials. Finally, we describe the VHDL library of on-line arithmetic operators realized using this approach.