Design Trade-offs in the Implementations of 14 Round 2 SHA-3 Candidates using Embedded Resources of Altera and Xilinx FPGAs

Marcin Rogawski, Rabia Shahid, Malik Umar Sharif, and Kris Gaj
George Mason University

Abstract

Cryptographic algorithms have been demonstrated in the past to take advantage of embedded resources available on modern FPGAs. For example, the fastest to date FPGA implementations of the Montgomery multiplication, a major building block of public key cryptographic algorithms, such as RSA, have been demonstrated using DSP units of high-performance Xilinx FPGAs. Advanced Encryption Standard, a major secret key cryptosystem used for bulk data encryption, has been sped up first by using Block Memories of Xilinx and Altera FPGAs, and then by using a combination of DSP units and Block RAMs in Xilinx Virtex 5 FPGAs.

In this paper, we focus on the use of embedded resources in efficient implementations of cryptographic hash functions. These algorithms have recently come to focus because of the contest for a new American federal hash function standard called Secure Hash Algorithm–3 (SHA-3), organized by NIST in the period 2007-2012. In this study, we take into account the set of all 14 Round 2 candidates to demonstrate advantages of using embedded resources for implementing modern hash functions, representing multiple security paradigms.

All algorithms are implemented using two major approaches: with and without using embedded resources of modern FPGAs. These implementations are then optimized, using the GMU benchmarking environment ATHENa , targeting four modern FPGA families: Spartan 3 and Virtex 5 from Xilinx, and Cyclone II and Stratix III from Altera.

Our results demonstrate significant savings in the amount of reconfigurable logic, resulting from the use of embedded block memories, especially high for functions based on large look-up tables, such as four AES-based candidates (ECHO, Fugue, Groestl and SHAvite-3), as well as BLAKE and Hamsi. The advantage of using DSP units and multipliers is much more limited, and typically associated with the significant performance drop. The main reason for that is that the majority of investigated hash functions use only addition, and cannot take any advantage of multipliers present in these units. Furthermore, the DSP-based addition is relatively less efficient in Stratix III compared to Virtex 5. This is due to the fact that in Stratix III, each DSP addition is preceded by multiplication whereas the DSP units in Virtex 5 give you the flexibility to bypass these multipliers.

Overall, embedded resources provide an interesting and important alternative to the use of basic reconfigurable logic resources in implementations of modern cryptographic hash functions. We believe that our study is the first one in the literature that looks comprehensively at utilizing embedded resources in a large class of hash functions, including SHA-2 and all 14 Round 2 SHA-3 candidates.