Alternative FPGA Implementations of some SHA-3 Finalists* (Abstract)

Julien Francq[1] and Jean-Baptiste Rigaud[2]

[1]EADS Defence & Security, Cyber Security Customer Solutions Center,
[2]École Nationale Supérieure des Mines de Saint-Étienne
[1]Julien.Francq@cassidian.com, [2]rigaud@emse.fr

Cryptographic hash functions are involved in widely-used protocols such as signature schemes, Message Authentication Codes (MAC) or encryption schemes. Recently, some of them have been successfully attacked, and serious attacks have been published against the NIST approved cryptographic hash functions SHA-1, and the SHA-2 family. Consequently, NIST has decided to develop one or more additional hash functions through a public competition in order to specify its future hash standard SHA-3. The goal is to publish the augmented and revised Hash Function Standard by 2012. The SHA-3 competition is similar to the development process of the Advanced Encryption Standard (AES) that took place in 1997, and chose the new symmetric primitive in 2001.

In round three of the SHA-3 Contest, 5 finalists remain for consideration. These hash algorithms are available for public comment and scrutiny, and such research is vital to the selection process. In particular, NIST has stated that computational efficiency of the algorithms in hardware, over a wide range of platforms, is addressed.

It appears that most of the hardware architectures proposed in the litterature for SHA-3 candidates are basic. In this paper, we will focus on optimized implementations of three finalists: `BLAKE`, `Keccak`, and `JH`. We hope that this talk will provide a significant contribution in the hardware benchmarking of the SHA-3 candidates.

Our first results concern `BLAKE`. At first sight, it seems challenging to implement only one circuit which can compute all the digest sizes requested by the NIST. Interoperability problems can then occur. We will show that it is possible to merge all the circuits needed for each digest size in only one at a reasonable cost. Moreover, thanks to the use of the so-called "unfolding method" and optimized adders, we will show that our `BLAKE` implementation can achieve a high throughput on Virtex-5 FPGAs. Let's recall that high-throughput hash function implementations are beneficial, for example in network server applications.

Our other results concern `Keccak`. We have successfully applied the unfolding method on this algorithm: we will show that the throughput of `Keccak` can be significantly improved.

Our last results concern `JH`. We will show that its throughput can be raised thanks to an extensive use of the Block RAMs of Virtex-5 FPGAs.

* This work is partially supported by the French Agence Nationale de la Recherche through the SAPHIR2 project under Contract ANR-08-VERS-014.

At the end of this talk, our results will be put back in the wider context of the SHA-3 competition.

In the case that `BLAKE`, `Keccak` or `JH` would become the SHA-3 standard, it will be embedded in future FPGA devices dedicated to cryptographic applications: this is why this talk submission can be of interest for CryptArchi 2011 Workshop.