ON THE HARDWARE IMPLEMENTATIONS OF THE SHA-2 (256, 384, 512) HASH FUNCTIONS

N. Sklavos and O. Koufopavlou.

Electrical and Computer Engineering Department
University of Patras, Patras, GREECE
nsklavos@ee.upatras.gr

ABSTRACT

Couple to the communications wired and unwired networks growth, is the increasing demand for strong secure data transmission. New cryptographic standards are developed, and new encryption algorithms are designed, in order to satisfy the special needs for security. SHA-2 is the newest powerful standard in the hash functions families. In this paper, a VLSI architecture for the SHA-2 family is proposed. For every hash function SHA-2 (256, 384, and 512) of this standard, a hardware implementation is presented. All the implementations are examined and compared in the supported security level and in the performance by using hardware terms. This work can substitute efficiently the previous SHA-1 standard implementations, in every integrity security scheme, with higher offered security level, and better performance. In addition, the proposed implementations could be applied alternatively in the integrations of digital signature algorithms, keyed-hash message authentication codes and in random numbers generators architectures.

1 INTRODUCTION

The continued growth of communications in the last decades and the large range of the sensitive offered applications in the networks, have triggered the increasing demand for security. In some cases, the existing security standards have been proved insufficient. In order to be satisfied the special needs for cryptography, new security algorithms and cryptographic schemes are designed and developed.

The Secure Hash Algorithm-1 (SHA-1) [1], is the world's most popular hash function. Unfortunately, the security level of this design is limited to a level comparable to an 80-bit block cipher. The announced new AES Standard [2], which is specified in 128-, 192-, and 256-bit keys, drove the demand for a new SHA algorithm offering security comparable to the AES key strengths. On August
26, 2002, NIST announced the approval of FIPS 180-2 [3], Secure Hash Standard, which introduces the specifications of three new hash functions, SHA-2 (256, 384, 512).

Hash functions is a fundamental primitive category in modern cryptography, often called one-way hash functions. A hash function is a computationally efficient function, which maps binary strings of arbitrary length to binary strings of some fixed length, called hash-values [4].

In this paper, a VLSI architecture for the SHA-2 hash function family is proposed. Based on this architecture, every one of the SHA-2 (256, 384, 512) hash functions have been implemented separately, due to the different specifications. All the proposed implementations are examined and compared in the offered security level and in the performance by using hardware terms. From the comparisons results, with other related works on other hash functions implementations, it is proven that the proposed implementations perform better in all of the cases. This work can substitute the previous SHA-1 standard implementations, in every integrity security scheme, with higher supported security level and better hardware performance. In addition, the proposed implementations can be applied efficiently in the implementation of digital signature algorithms [5], keyed-hash message authentication codes [6] and in random numbers generators architectures [7].

This paper is organized as follows: in section two the SHA-2 standard is introduced. In the next section the proposed architecture for the SHA-2 hash functions family is presented in detail. In addition, the specifications and the design demands for each one of the proposed implementations SHA-2 (256), SHA-2 (384), SHA-2 (512), are also presented. The VLSI implementations synthesis results are given in the section 4 and comparisons with other related works are presented. Finally, conclusions are discussed in the last section.

0-7803-7761-3/03/$17.00 ©2003 IEEE

2 SECURE HASH FUNCTION 2

The SHA-2 standard [3] supersedes the existing FIPS 180-1 [1], adding three new algorithms SHA-2(256), SHA-2(384), and SHA-2(512), for computing a condensed representation (message digest) of electronic data (input message. The message digests range in length from 160 to 512 bits, depending on the algorithm.

These hash functions enable the determination of a message's integrity: any change to the message will, with a very high probability, results in a different message digest. Secure hash functions are basically used with other cryptographic algorithms, such as digital signature algorithms, keyed-hash message authentication codes, and in random numbers generators.

The three new hash functions specified in this standard are called secure because, for a given algorithm, it is computationally infeasible 1) to find a message that corresponds to a given message digest, or 2) to find two different messages that produce the same message digest. Any change to a message will, with a very high probability, results in a different message digest.

The hash functions of the SHA-2 family differ most significantly in the number of security bits that are provided for the hashed input data block. Security is directly related to the message digest length. In most of the cases, when a secure hash function is used in conjunction with another algorithm, there special specifications, which require the use of a secure hash function with a certain number of security bits. For example, if a message is being signed with a digital signature algorithm that provides 256-bit security, then that signature algorithm may require the use of a secure hash algorithm that provides 256-bit security, SHA-2(512). In addition, these three hash functions differ in terms of the block size and words of data that are used during hashing. Table 1 presents the basic properties of all four secure hash functions.

Hash Functions / Terms (bits)	SHA-1	SHA-2 (256)	SHA-2 (384)	SHA-2 (512)
Message Size	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block Size	512	512	1024	1024
Word Size	32	32	64	64
Tranf. Rounds	80	64	80	80
Mes. Digest	160	256	384	512
Security	80	128	192	256

Table 1: Secure Hash Functions Comparison

In the above table, with the rule "security" is defined a birthday attack [8] on a message digest of size n, which finally produces a collision with a work factor of approximately $2^{n/2}$.

3 PROPOSED SHA-2 FAMILY ARCHITECTURE

In order to implement the three hash functions of the SHA-2 family, a common architecture have been designed, and it is used for each one of them separately. This proposed architecture is illustrated in the Fig. 1.

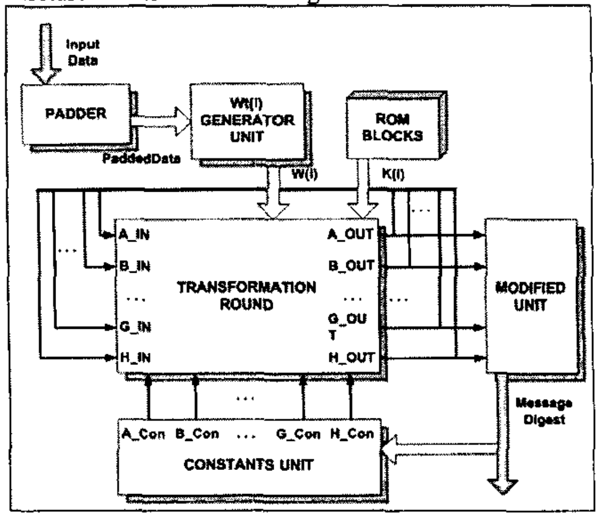

Figure 1: Proposed SHA-2 Family Architecture

The PADDER pads the input data and converts them to n-bit blocks (padded data). This operation is the same for these three hash functions, which sequentially process blocks of n-bit. The padded message is generated with the following process: a logic "1", followed by m "0"s, followed by a 64-bit integer are appended to the end of the input data message, to produce a padded message of length equal to a multiple of n. The 64-bit integer is equal to the length of the input data message. The padded data is a multiple of 512-bit for the SHA-2(256) and a multiple of 1024 for the SHA-2(384) and SHA-2(512). The padded data are divided into N equal data blocks $M^1,\ldots,M^N$ and are processed in order, using the following equations:

$$W(i)=M^i, \qquad 0<=i<=15, \quad (1)$$

$$W(i)=\sigma 1W(i-2)+W(i-7)+\sigma 0W(i-15)+W(i-16), 15<i, \quad (2)$$

where i indicates the number of the transformation round.

Every one of the hash functions uses a specified number of transformation rounds, which is determined in the Table I. The ROM Blocks are used for predefined K(i) constants support. These constants are different for each one of the implemented hash functions. For the SHA-2 (256), 64x32-bit ROM blocks are allocated, while each one of the SHA-2(384) and SHA-2(512) uses 80x64-bit ROM blocks. Finally, a last modification (MODIFIED UNIT) is performed to the data before the message digest is produced.

The basic TRANSFORMATION ROUND is working with 8 basic inputs (A_IN,...,H_IN) and 8 outputs (A_OUT,..,H_OUT), which are related with the following equations:

A_OUT=T1(E_IN,F_IN,G_IN,H_IN)+T2(A_IN,B_IN,C_IN) (3)
B_OUT=A_IN (4)
C_OUT=B_IN (5)
D_OUT=C_IN (6)
E_OUT=D_IN+T1(E_IN,F_IN,G_IN,H_IN) (7)
F_OUT=E_IN (8)
G_OUT<=F_IN (9)
H_OUT<=G_IN (10)

The working inputs are initialized to the constants values (A_Con,...,H_Con) before hash computation begins. In the above equations, two functions T1(x,y,z) and T2(x,y,z) are used for the desirable data transformation. The architectures of these functions are illustrated in the following Fig.2:

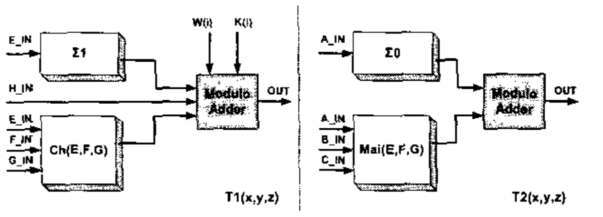

Figure 2: T1(x,y,z), T2(x,y,z) Functions Architectures

For the SHA-2(256) the 2^{32} modulo adder is used, while for the SHA-2(384) and SHA-2(512) the 2^{64} modulo adder is implemented. The Ch and Maj (Fig. 3) are common functions for the three hash functions while Σ1, Σ0 are different for each one of the hash functions.

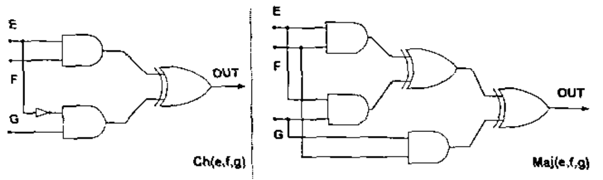

Figure 3: Ch, MAj Functions Architectures

The proposed architectures for the Σ1, Σ0 functions are shown in the next Fig. 4. In the same figure, the architectures of σ0, σ1 functions, for the generation of the Wt(i) data block, are also illustrated.

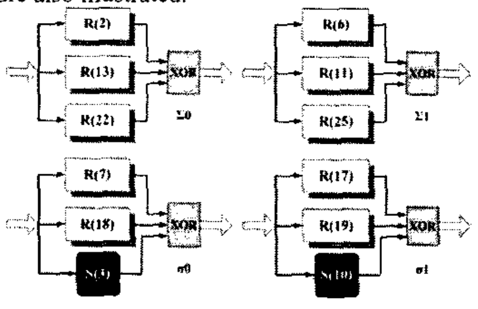

Figure 4: Σ0, Σ1, σ0, σ1 Functions Architectures

where R(n) indicates right rotation of the input by n-bit and S(n) indicates right shift by n-bits. The functions of Fig. 4 are similar but they are implemented by using different constants for each one of the SHA-2 family hash functions. The specified constants values for every hash function are given by the SHA-2 standard [3].

4 VLSI SYNTHESIS RESULTS

The proposed architectures have been captured by using VHDL and have been synthesized placed and routed using XILINX FPGA Virtex Device (v200pq240). The synthesis results for the proposed SHA-2 implementations are illustrated in the following Tables 2-4.

Allocated Area	Used / Available	Utilization
Fun. Generators	2120 / 4704	45 %
CLB Slices	1060 / 2352	45 %
Dffs or Latches	1651 / 4704	35 %
Frequency	83 MHz	

Table 2: SHA-2(256) Implementation Synthesis Results

Allocated Area	Used / Available	Utilization
Fun. Generators	3932 / 4704	83 %
CLB Slices	1966 / 2352	83 %
Dffs or Latches	3689 / 4704	78 %
Frequency	74 MHz	

Table 3: SHA-2 (384) Implementation Synthesis Results

Allocated Area	Used / Available	Utilization
Fun. Generators	4474 / 4704	95 %
CLB Slices	2237 / 2352	95 %
Dffs or Latches	3739 / 4704	79 %
Allocated Area	Used / Available	Utilization
Frequency	75 MHz	

Table 4: SHA-2 (512) Implementation Synthesis Results

In addition, the proposed implementations are compared in the terms of operating frequency, throughput and in the area-delay product. These comparisons results are illustrated in the following Fig.5:

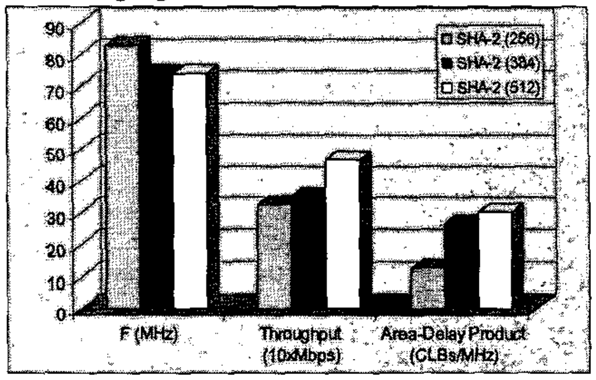

Figure 5: SHA-2 Functions Comparisons

The allocated area resources of the SHA-2(256) are at about 50% less than the allocated resources of the other two hash algorithms implementations. The operating frequency of SHA-2(384) and SHA-2(512) is measured at 75 and 74 MHz respectively, and they are less compared with the 84 MHz of the SHA-2(256). Furthermore, the throughput of the SHA-2(512) is better compared with the other two implementations. Finally the SHA-2(256) hash function is proven the best in the term of the area-delay product, compared with the other two hash functions integrations.

In addition, comparisons of the proposed SHA-2 hash functions implementations with other related works are illustrated in the Table 5. Since the SHA-2 hash family is a new standard only few related works [9] have been published until now in technical literature. Therefore, performance comparison is an enough difficult process. Although, in the Table 5 comparisons with other hash function families implementations [7], [9-11] are also given, in order to have a fair and detailed comparison of the proposed implementations.

Implementations	Covered Area	Frequency (MHz)	Bit-Rate (Mbps)
SHA-2 (256) [9]	1004 CLBs 10900 Gates	42.9 FPGA 59 ASIC	77 56
SHA-1 [9]	1004 CLBs 10900 Gates	42.9 FPGA 59 ASIC	119 86
SHA-1 [7]	2606 CLBs	37	233
SHA-1 [10]	-	90 SOFT.	40
SHA-1 [11]	-	N/A SOFT. 133 SOFT.	4.23 41.51
MD5 [9]	1004 CLBs 10900 Gates	42.9 FPGA 59 ASIC	146 107
MD5 [10]	-	90 SOFT.	114
Prop. SHA-2(256)	1060 CLBs	83	326
Prop. SHA-2(384)	1966 CLBs	74	350
Prop. SHA-2(512)	2237 CLBs	75	480

Table 5: Implementations Comparison Results

The SHA-2(256) performs better in terms of operating frequency and throughput compared with both the ASIC and the FPGA implementations of the related work [9]. Furthermore, all the three proposed implementations are compared with the previous SHA-1 hash function standard hardware implementations [7], [9], [11].

From the synthesis results of the Table 5, it is also proven that the proposed implementations performs better compared with the hardware designs works of [7], [9] and the software development of [10], [11]. Finally, the three proposed implementations have better synthesis results compared with the works of [9], [10], on the other well known hash family functions, MD5.

5 CONCLUSIONS

In this paper, three different implementations are proposed for the SHA-2 hash family. SHA-2 is the newest hash function standard. This hash functions family provides higher security level than the existing SHA-1. The proposed implementations are compared in terms of operating frequency, throughput and in the area-delay product. From the comparisons results, with other related works [9] and implementations of other hash functions families [7], [9-11] it is proven that the proposed implementations performs better in all of the cases. They can substitute efficiently the previous SHA-1 implementations, in communications protocols and networks integrity units, with higher supported security level and better achieved performance. This work can be applied efficiently in the implementation of digital signature algorithms, keyed-hash message authentication codes and in random numbers generators architectures.

6 REFERENCES

[1] SHA-1 Standard, National Institute of Standards and Technology (NIST), Secure Hash Standard, FIPS PUB 180-1, www.itl.nist.gov/fipspubs/fip180-1.htm

[2] "Advanced Encryption Standard", http://csrc.nist.gov.

[3] SHA-2 Standard, National Institute of Standards and Technology (NIST), Secure Hash Standard, FIPS PUB 180-2, www.itl.nist.gov/fipspubs/fip180-2.htm

[4] S. Bakhtiari, R.Safavi-Naini, J. Pieprzyk, "Cryptographic Hash Functions: A Survey ", Technical Report 95-09, Department of Computer Science, University of Wollongong, July 1995.

[5] National Institute of Standards and Technology (NIST), Digital Signature Standard, FIPS PUB 186-2, http://csrc.nist.gov/publications/fips/fips186-2.htm

[6] HMAC Standard, National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), http://csrc.nist.gov/publications/fips.htm

[7] N. Sklavos, P. Kitsos, K. Papadomanolakis and O. Koufopavlou, "Random Number Generator Architecture and VLSI Implementation", proc. of IEEE International Symposium on Circuits & Systems (ISCAS'02), USA, 2002.

[8] A.Menezes, P.van Oorchot, and S.Vanstone, *Handbook of Applied Cryptography*, CRC Press, Inc, October 1997.

[9] S. Dominikus, "A Hardware Implementation of MD4-Family Hash Algorithms", proc. of IEEE International Conference on Electronics Circuits and Systems (ICECS'02), Vol. III, pp.1143-1146, Croatia, 2002.

[10] H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160, a strengthened version of RIPEMD", Fast Software Encryption, LNCS 1039, Springer-Verlag 1996.

[11] Michael Roe, "Performance of Block Ciphers and Hash Functions-One Year Later", proc. of Second International Workshop for Fast Software Encryption '94, Leuven, Belgium, December 14-16, 1994.