High Performance Cryptographic Engine PANAMA: Hardware Implementation

G. Selimis, P. Kitsos, and O. Koufopavlou

VLSI Design Labotaroty
Electrical & Computer Engineering Department,
University of Patras Patras, Greece
Email: gselimis@ee.upatras.gr

ABSTRACT

In this paper a hardware implementation of a dual operation cryptographic engine PANAMA is presented. The implementation of PANAMA algorithm can be used both as a hash function and a stream cipher. A basic characteristic of PANAMA is a high degree of parallelism which has as result high rates for the overall system throughput. An other profit of the PANAMA is that one only architecture supports two cryptographic operations – encryption/ decryption and data hashing. The proposed system operates in 96.5 MHz frequency with maximum data rate 24.7 Gbps. The proposed system outperforms previous any hash functions and stream ciphers implementations in terms of performance. Additional techniques can increase the achieved throughput about 90%.

1. INTRODUCTION

Today more and more sensitive data is stored digitally. Bank accounts, medical records and personal emails are some categories that data must keep secure. The science of cryptography tries to encounter the lack of security. Data confidentiality, authentication, non-reputation and data integrity are some of the main parts of cryptography. The evolution of cryptography drove in very complex cryptographic models which they could not be implemented before some years. The revolution of computers and especially CMOS technology permit the design and the implementation of systems with characteristics as limited area resources, low power consumption and high speed. Then due the CMOS technology known cryptographic standards were implemented and today they provide secure transactions.

The use of systems with increasing complexity, which usually are more secure, has as result low rate of throughput. Last years the authors of new cryptographic algorithms try to suit high complexity transformations in systems with the view of high throughput rates. The assistance of FPGA and ASIC technologies in this road is substantial. FPGAs especially, are used for efficient and flexible implementations. When a current algorithm is broken and a new standard is created (e.g. Advanced Encryption Standard-AES [1]), it is perceivable that field devices are upgraded with a new encryption algorithm [2]. In addition the hardware implementations are more efficiency in FPGAs than general purpose CPUs due to the fact that the algorithm specifications suits much better in FPGA structure.

Typical application with high speed requirements is encryption or decryption of video-rate in conditional access applications (e-g pay TV). The modern networks have been implemented to satisfy the demand for high bandwidth multimedia services. Then the switches which they are placed at the nodes of the network must provide high throughput. So if there is a need for secure networks, the systems in the network switches should not introduce delays. PANAMA [3] is a cryptographic module that can be used both as a cryptographic hash function and as stream cipher in applications with ultra high speed requirements

In this paper an efficient implementation of the PANAMA is presented. The introduced system works either as a hash function or stream cipher. The proposed implementation is suitable for application with ultra high speed data rates. Comparisons with other previous published hash functions and stream ciphers implementations prove that the proposed one performs better in terms of overall system throughput.

This paper is organized as follows: In section 2 the main features of Hash Functions and Stream Ciphers are presented. In section 3 the PANAMA specifications are given. The proposed Dual Operation Cryptographic Engine PANAMA is presented in detail in section 4. The FPGA synthesis results are given in section 5, and finally section 6 concludes the paper.

2. DEFINITIONS

2.1 Hash Functions

The operation of a Hash function H is to map an input of arbitrary length into a fixed number of output bits, the hash value. Hash functions are used in cryptography mainly for authentication and digital signature schemes. The requirements for a hash function are as follows:

- The input can be of any length.
- Fixed - length output.
- H(x) is relatively easy to compute for any given x.
- H(x) is one-way function which means that given a hash value h, it is computationally infeasible to find some input x such that H(x) = h.

- H(x) is collision-free which means it is computationally infeasible to find any two messages x and y such that H(x) = H(y).

2.2 Stream Ciphers

While block ciphers operate on large blocks of data, stream ciphers typically operate on smaller units of plaintext, usually bits. With a stream cipher, the transformation of these smaller plaintext units will vary, depending on when they are encountered during the encryption process. Stream cipher generates what is called a *keystream* (a sequence of bits used as a key). Encryption is accomplished by combining the keystream with the plaintext, usually with the bitwise XOR operation.

3. PANAMA ALGORITHM

The basic elements of PANAMA [3] algorithm are a finite state machine with a 544-bit state which called state α, an 8192-bit buffer and the state update transformation which denoted by ρ. When the data of buffer and state machine are updated then an iteration happens, Push or Pull. The three possible modes for the PANAMA module are Reset, Push and Pull. In Reset mode the state α and buffer are set to 0. In Push mode an 8-word input is applied and there is no output. In Pull mode there is no input and an 8-word output is delivered. The buffer behaves as a linear feedback shift register that ensures that input bits are injected into the state α over a wide interval of iterations. In the Push mode the input to the shift register is formed by the external input, in the Pull mode, by part of state α. Figure 1 shows the Push and Pull modes of PANAMA.

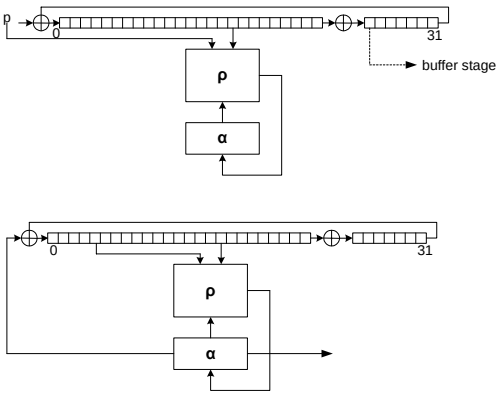

Figure 1. Push (above) and Pull (below) modes of PANAMA

The updating transformation ρ of the state has high diffusion and distributed nonlinearity. It combines four different transformations: one for nonlinearity (sigma stage-σ), one for bit dispersion (theta stage-θ), one for inter-bit diffusion (pi stage-π), and one of injection of buffer and input bits (gamma stage-γ). The stage updating transformation ρ is given by the formula:

$$\rho = \sigma \circ \theta \circ \pi \circ \gamma$$

Symbol "o" denotes the associative comparison of transformations where the right-most transformation executed first.

The PANAMA hash function transforms the information of arbitrary length to a hash result of 256 bits. It consists of two processes: message padding and iteration phase. During message padding the information is converted to a stream of data which its length is multiple of 256 while during iteration phase the cryptographic module follows the below sequence diagram (Table 1).

Table 1. The sequence diagram of the hash function

Time step t	Mode	Input	Output
0	reset	------	------
1,…,V	Push	p^t	------
V=1,…,V+32	Pull	------	------
V+33	Pull	------	H

The PANAMA stream encryption scheme is initialized by first loading the 256-bit key K, the 256-bit diversification parameter Q and performing 32 additional blank pull operations. During keystream generation an 8-word block z is delivered at the output for every iteration. The full scenario of encryption / decryption process is shown in Table 2.

Table 2. The sequence diagram of the stream encryption scheme

Time step t	Mode	Input	Output
0	reset	------	------
1,…,V	Push	p^t	------
V=1,…,V+32	Pull	------	------
V+33	Pull	------	H

In practice, the diversification parameter allows for frequent resynchronization without the need to change the key.

4. PROPOSED ARCHITECTURE

The proposed architecture is presented in Figure 2.

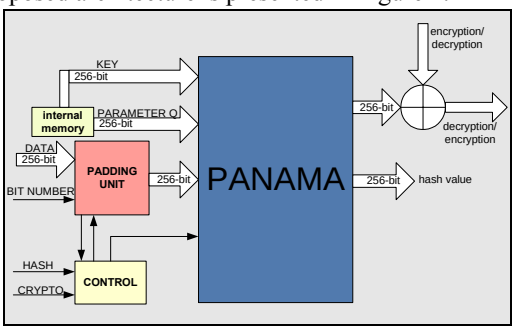

Figure 2. PANAMA - proposed architecture

The operation of PANAMA system has two options. It can operate as keystream generator for data encryption and also as hash function. In the following Figure 3 the PANAMA proposed VLSI implementation is presented in detail.

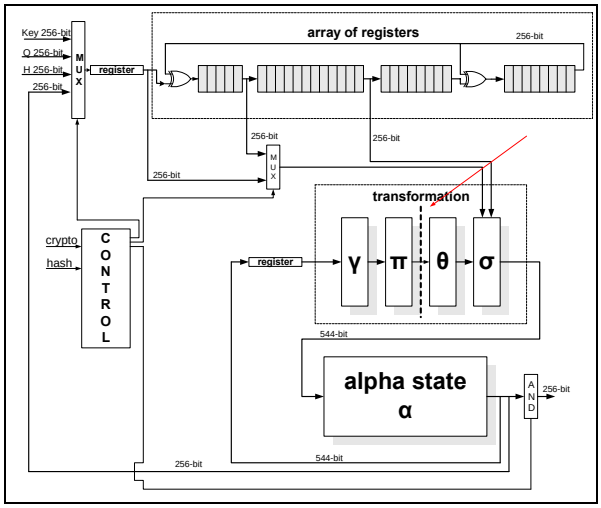

Figure 3. The PANAMA proposed architecture in details

The main components of proposed PANAMA crypto engine are: the alpha state α, the buffer, the transformation round ρ and the control unit. The alpha state has memory storage of 544 bits and it is implemented by (17 *32-bit) registers. The same policy has been applied for the construction of buffer. The buffer is an array of registers. As the Figure 3 shows shown that the buffer consists of 32 cells. Every cell has 8 words or 8*32-bit registers. The use of registers is preferred because the throughput value is increased drastically compared with RAM implementation.

The control unit is a Finite State Machine. It controls the output of the MUXes and of the AND gate. According the values of the signals Reset, CRYPTO and HASH the panama engine works either as a hash or data encryption. When CRYPTO=1 then the values of key and parameter Q are injected in the PANAMA block and after the appropriate processing a sequence of bits are generated. When HASH=1 then the PANAMA block orders as hash function and a sequence of 256 block of bits are injected in PANAMA block

The updating transformation is an array of four transformations each with its specific contribution. The aim of these transformations is to introduce high diffusion and distributed nonlinearity to system. The VLSI implementation of transformation round ρ is shown in Figure 4. γ is an invertible non linear transformation. It is composed of 18 basic components. Inside every component two 32- bit XOR operations are executed. The permutation π combines cyclic word shifts and a permutation of the word positions. The operation of π has been implemented with 18 shift registers which map every 32 bit input in the appropriate output while the internal bits of every stream are shifted cyclic. θ is an invertible linear transformation. It is composed of 18 main components which every one executes a two 32-bit XORs as shown in Figure 4. Finally transformation σ executes 18 32-bit XOR operations between the bits of θ's output and the bits of buffer or input words. It is composed of 18 32-bit XOR. Due the fact that one transformation is executed in a clock cycle the whole system has very high throughput rate.

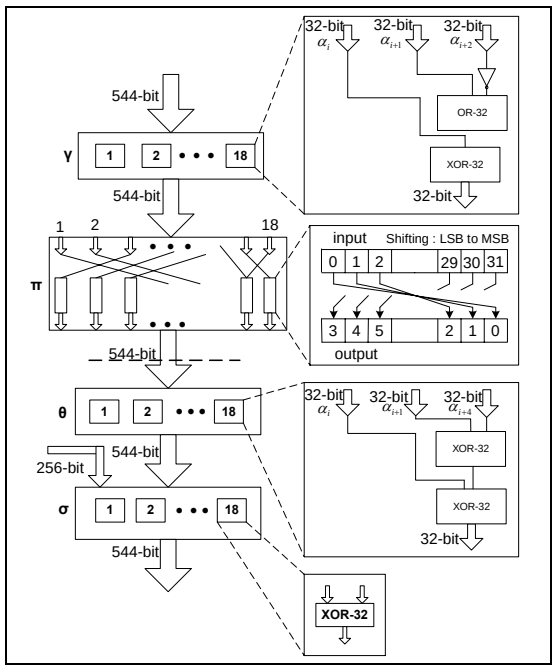

Figure 4. The VLSI implementation of transformation round ρ

In Figures 3 and 4 the arrow shows approximately the middle of the system's data path. In this point it could placed a negative edge triggered register. As a result clock period is reduced roughly in half. With this method [4] the throughput can be increased about 90%. Due to the fact that the system's frequency is multiplied by a factor about 2.The only penalty in this situation is the additional area resources due the use of one extra 544-bit register.

5. HARDWARE SYNTHESIS RESULTS

The proposed architecture has been captured by using VHDL. All the internal components of the design were synthesized placed and routed using XILINX FPGA device VIRTEX-E v405efg900. According to the Table 1 the hash operation demands 32 rounds to eject the first hash value. In this case throughput is computed by the function $\frac{0{,}0965 * 256}{32 + R} R$, where R is the number of 256-bit packets which are injected to the system. When the parameter R increases is easy to prove that the throughput increases. If only one packet (R=1) injected for data hashing the throughput is 748 Mbps and the system does not operates efficiently. Figure 5 shows the system's throughput in relation with the number of the sequent packets for hashing. Finally the maximum throughput can reach the value of 24.7 Gbps

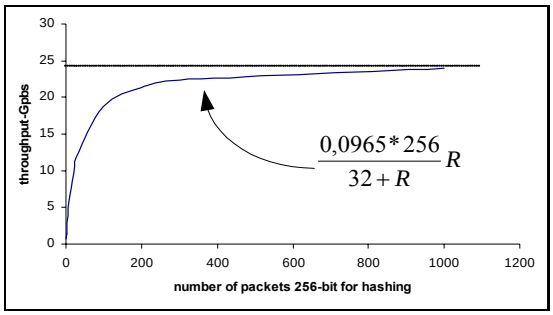

Figure 5 The relation throughput-number of sequent packets

For the encryption/decryption mode 34 rounds are demanded for the first key stream to be generated. After the initialization phase a sequence of 256-bit key stream is generated and 24.7 Gbps throughput value is achieved. Due to the proposed PANAMA implementation is the first, no previous performance metrics there are for this algorithm. So comparison with other previous hash functions and stream ciphers implementations are given in Table 3.

6. CONCLUSIONS

In this paper a first VLSI implementation of PANAMA algorithm is presented. The system operates both as stream cipher and a hash function. The comparison proves that the proposed implementation outperform any previous published hash function and stream cipher hardware implementations. The system was synthesized, placed and routed by using FPGA device. It reaches 24.7 Gbps throughput at 96.6 MHz. A useful technique in order to reduce the critical path is introduced with the usage of negative edge-triggered register.

Table 3. Experimental Results and comparisons

Implementation		FPGA	CLBs	Frequency-MHz	Throughput-Mbps
HASH FUNCTIONS	MD5 [4]	xilinx V1000FG680-6	880	21	165
	MD5 [4]	xilinx V1000FG680-6	4763	71.4	354
	SHA-1[5]	xilinx V300PQ240-6	2606	37	257
	SHA-1[6]	Altera EP20K1000EBC652-3	-	18	114
	MD5[[6]	Altera EP20K1000EBC652-3	-	18	192
	SHA-1[7]	xilinx V300E	1004	42.9	119
	MD5[7]	xilinx V300E	1004	42.9	146
	SHA-2[7]	xilinx V300E	1004	42.9	77
	RIPEMD[7]	xilinx V300E	1004	42.9	89
	SHA-2[8]	xilinx V200PQ240-6	1060	83	326
	SHA-2[8]	xilinx V200PQ240-6	1966	74	350
	SHA-2[8]	xilinx V200PQ240-6	2237	75	480
	Prop. PANAMA	xilinx -E V600EFG900	4524	95.6	reaches 24700*
STREAM CIPHERS	RC4[9]	xilinx XC4000E4013EPQ208-2	255	17.8	2.22
	RC4[10]	xilinx 2V250FG256	138	64	22
	MULTI-S01[12]	Hitachi's HG73C cell library	139.5K	140	9.1
	Prop. PANAMA	virtex-e V600EFG900	4524	95.6	24700

REFERENCES

[1] U.S Department of Commerce/National Institute of Standard and Technology.FIPS PUB 197, Specification for the Advanced Encryption Standard (AES), November 2001.

[2] T. Wollinger and C. Paar, "How secure are FPGAs in Cryptographic Applications?", in 13th International Conference on Field Programmable Logic and Applications-FPL 2003,Lisbon, Portugal, September 1-3, 2003.

[3] J. Daemen, and Craig Clapp, "Fast Hashing and Stream Encryption with PANAMA" Fast Software Encryption: 5th International Workshop, FSE'98, Paris, France, March 1998.

[4] P. Kitsos, M. D. Galanis, and O. Koufopavlou, "High-Speed Hardware Implementations of the KASUMI Block Cipher", accepted for presentation in IEEE International Symposium on Circuits & Systems (ISCAS'04), Canada, May 23-26, 2004.

[5] Janaka Deepakumara, Howard M. Heys and R. Vanketersam, "FPGA Implementation of MD5 hash algorithm", in proc. of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2001), Toronto, Ontario, May 2001.

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

[7] Yong Kyu Kang, Dae Won Kim, Taek Won Kwon, and Jun Rim Choi, "An Efficient Implementation of hash function processor for IPSEC", in proc. of third IEEE Asia-Pacific Conference on ASICs, Taipei, Taiwan, August 6-8, 2002.

[8] Snadra Dominicus, "A hardware implementation of MD4-family algorithms",in proc. of IEEE International Conference on Electronics Circuits and systems (ICECS 2002), Croatia, September 2002.

[9] N. Sklavos and O. Koufopavlou, "On the hardware implementation of the SHA-2 (256,384,512) hash functions", in proc. of IEEE International symposium on Circuits and systems (ISCAS 2003), may 25-28, Bangkok, Thailand, 2003.

[10] P. Hamalainen, M. Hannikainen, T. Hamalainen and J. Saarinen, "Hardware Implementation of the improved WEP and RC4 Encryption Algorithms for Wireless Terminals", the European Signal Processing Conference, September 5-8, 2000, Tampere, Finland, pp. 2289-2292.

[11] P. Kitsos, G. Kostopoulos, N. Sklavos, and O. Koufopavlou, "Hardware Implementation of the RC4 Stream Cipher", in 46th IEEE Midwest Symposium on Circuits & Systems '03, December 27-30, Cairo, Egypt, 2003.

[12] Hitachi, Ltd., "Self-Evaluation Report MULTI-S01", 2001.

* the accurate value of throughput is defined by the Figure 5