Comparison of hardware performance of selected Phase II eSTREAM candidates

Kris Gaj, Gabriel Southern, and Ramakrishna Bachimanchi
ECE Department
George Mason University

Abstract

Five leading Phase 2 Profile 2 eSTREAM candidates have been implemented in hardware, targeting two main semiconductor technologies, Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs). An old GSM encryption algorithm A5/1 has been included in the study as well. All six ciphers have been analyzed from the point of view of the hardware efficiency, and two hardware architectures have been developed for each of them. The first of these architectures has been optimized for the minimum area, and the second for the maximum throughput to area ratio. Our study has revealed very large differences among all eSTREAM candidates in terms of the hardware efficiency, and have demonstrated a relative superiority of Trivium and Grain over other analyzed ciphers.

Keywords: eSTREAM, stream cipher, hardware, FPGA, ASIC

1. Introduction

Hardware implementation efficiency is one of the primary requirements for every cipher. This efficiency is particularly important in case of eSTREAM Profile 2 candidates which were designed with the special emphasis on their suitability for hardware implementations with limited number of gates, memory, and power supply [1].

In this paper, we compare hardware efficiency of four Profile 2 eSTREAM candidates qualified to Phase 2 as focus candidates (Grain, Mickey-128, Phelix, and Trivium), one additional Phase 2 cipher, Salsa20, and an old (and insecure) GSM standard A5/1 [2-4]. The basic features of these six stream ciphers are summarized in Table 1.

In Fall 2006, the first author was an instructor for a graduate course, ECE 545, Introduction to VHDL, focusing on designing digital systems with hardware description languages such as VHDL [5]. The third author was a teaching assistant for this course. As a primary project in this course, the students were given a task of implementing one of five selected eSTREAM candidates. Additionally, one student volunteered to implement an old standard, A5/1. Twenty students accepted the challenge and were asked to rank five eSTREAM ciphers in the order of their preference based exclusively on their first reading of the cipher specification. It is quite safe to assume that the students' preference reflected their perceived difficulty of implementing a particular cipher in VHDL, with the highest ranking (five) given to the cipher perceived as the easiest to implement. The results of this ranking are presented in Table 2.

Three ciphers, Trivium, Salsa20, and Mickey-128, were perceived by students as the easiest to implement. Grain was (surprisingly to the authors) ranked only as a fourth choice. Finally, Phelix was a far outsider, and was not a first choice of any of the twenty students.

The student preferences were taken into account in the final assignments, but each cipher was assigned to four students working on their implementations independently. At the end of the semester, the best out of four independent implementations of each cipher was selected. These implementations were revised by the authors of this paper in order to assure a full uniformity of the coding style and the detailed design choices. These revised codes were used in order to generate unified results presented in Section 4 of this paper.

This methodology led to six optimized implementations, of comparable quality, as the students taking the course had a similar background, were following the same design style (based on the use of

Table 1 Basic features of stream ciphers compared in this paper

Name	Authors	Key size [bits]	IV size [bits]	Internal state size [bits]	Basic components
Grain	M. Hall, T. Johansson, W. Meier	80	64	160	LFSR, NFSR, output function
Mickey-128	S. Babbage M. Dodd	128	0..128	320	LFSR, NFSR
Phelix	D. Whiting, B. Schneier, S. Lucks, F. Muller	≤ 256	128	288	Block function based on adders, rotators, and xors
Salsa20	D.J. Bernstein	256, 128	64	512	Hash function based on adders, rotators, and xors; used in the counter mode
Trivium	C. De Canniere, B. Preneel	80	80	288	LFSR, NFSR
A5/1	unknown	64	22	64	LFSR, clock control units

Table 2 Perceived difficulty of a hardware implementation of selected eSTREAM ciphers based on the survey of 20 GMU ECE students

Cipher	Perceived ease of implementation (5 – very easy; 1 – very difficult)
Trivium	3.36
Salsa20	3.32
Mickey-128	3.32
Grain	3.00
Phelix	2.00

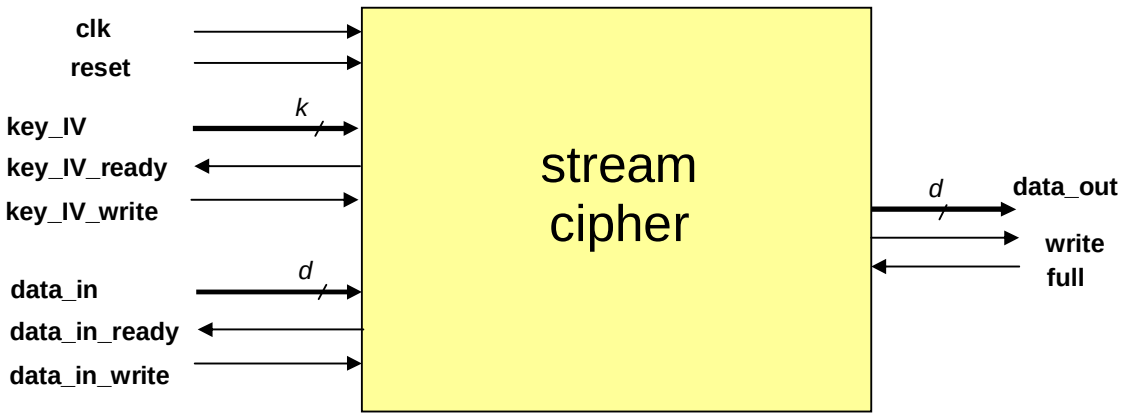

Fig. 1 Interface of a stream cipher used in our implementations

block diagrams and algorithmic state machine (ASM) charts, translated to VHDL), developed within the same amount of time (around 6 weeks). All designers used the same interface, shown in Fig. 1.

All six ciphers were described in portable VHDL, and then implemented using Field Programmable Gate Arrays (FPGAs) from the Spartan 3 Xilinx family, and synthesized using Synopsys tools targeting a semicustom ASIC technology based on the TSMC user libraries.

In Section 2, we describe the exact methodology and tools used by all designers. In Section 3, we present a few alternative hardware architectures and optimization options available for each cipher. In Section 4, we present and discuss major results. In Section 5, we compare these results to the results reported earlier in the literature. We summarize our findings and present conclusions in Section 6.

2. Methodology

All ciphers have been first designed using medium level block diagrams and algorithmic state machine (ASM) charts. These diagrams and charts have been then converted to synthesizable register-transfer level (RTL) VHDL code, without using any library components specific for a given technology or FPGA family. The code was debugged using either Aldec Active HDL or ModelSim Xilinx Edition VHDL simulators, depending on the student's preference. After the code was functionally correct, it was first synthesized using Synplicity Synplify Pro for Xilinx FPGAs, and then using Synopsys Design Compiler for ASICs. The choice of tools and their versions, affecting final results, is summarized in Table 3.

Table 3 Tools used for the implementation of the selected stream ciphers in the FPGA and ASIC technologies

Technology	FPGA	ASIC
VHDL simulation and debugging	Aldec Active HDL v. 7.1 ModelSim Xilinx Edition II	
Logic Synthesis	Synplicity Synplify Pro v. 8.5	Synopsys Design Analyzer X-2005.9
Implementation (mapping, placing and routing)	Xilinx ISE v. 8.1i	

In case of FPGAs, a low cost FPGA family, Spartan 3, fabricated in the 90 nm semiconductor technology was selected. For ASICs, the implementation is based on the standard-cell 90 nm library from TSMC, TCBN90G. Thus, both types of circuits use transistors of the same size.

The back end design was performed only for FPGAs. It consists of mapping, placing and routing. For both technologies, the timing of the circuit was characterized using static timing analysis, which returns the critical path in the circuit and the minimum clock period. Based on this data, the throughput of the circuit in Mbits per second, and the key setup latency in nanoseconds were computed.

All ciphers have been first optimized for minimum area. In most cases, the corresponding implementation was implied directly by the cipher specification. Then, an attempt was made to change the circuit structure in such a way to perform the same operation with the better ratio of the circuit throughput to the circuit area. Different parallelization methods were considered, wherever appropriate, in order to come up with an optimum design maximizing this ratio. The available design choices are described in more detail in the following chapter.

3. Choice of hardware architecture

Six selected ciphers represent three different types of stream ciphers, with different basic hardware architectures and optimization options in each case.

The first class of ciphers are ciphers based on linear and non-linear feedback shift registers (LFSRs and NFSRs) with a serial input to each register. Often, both types of circular structures are included

within the same cipher and interact with each other. This class of stream ciphers includes Grain, Trivium, and A5/1. In the basic architecture, implied by the cipher specification, each shift register is shifted by only one position per each clock cycle, and only one bit of the keystream is produced at a time. As a result, the maximum circuit throughput is equal to one bit divided by the minimum clock period, and if expressed in Mbits/s is numerically equal to the maximum clock frequency in MHz. The key and the IV are loaded one bit per clock cycle, so the key setup latency, expressed in clock cycles, is equal to the combined length of the key and the IV, incremented with the number of clock cycles required for the initialization run of LFSRs and NFSRs.

In order to increase the throughput, LFSRs and NFSRs can be shifted by *d* positions at a time, and the architecture produce *d* bits of the keystream per clock cycle. This *d*-parallel architecture increases the throughput by a factor close to *d*, but at the same time may have a significantly larger area, because the entire feedback logic must be repeated *d* times. Still, in majority of cases, the increase in the circuit throughput is a stronger function of *d* than the increase in the circuit area, and thus the maximum throughput to area ratio is achieved for the largest possible value of *d* supported by a given cipher.

This maximum value of the parallelization factor *d* can be determined by the analysis of the cipher structure, and in particular, the minimum distance between the serial entry of each shift register and the first tap position used in the feedback logic. Additionally, the allowed values of *d* may be limited to the proper divisors of the total length of each LFSR and NFSR. The parallelization factors *d*, selected using this approach, are equal to the following integer values: d=2, 4, 8, 16 for Grain; d=2, 4, 8, 16, 32, 64 for Trivium; and d=3, 4 for A5/1. The larger parallelization factors, although possible, are not likely to lead to the better throughput to area ratio.

The second type of a stream cipher represented in our group is a cipher that includes both LFSRs and NFSRs, but each of these registers has a parallel rather than serial input from the feedback loop. This parallel input combined with parallel output complicates the feedback loop, and makes its parallelization expensive in terms of both the design time and the circuit area. This type of ciphers is represented in our group by Mickey-128. The basic hardware architecture of Mickey-128, producing one bit of the keystream per clock cycle, is implied by the cipher specification. A parallelization, although likely possible, was not straightforward enough to be discovered by four graduate students who have attempted to implement and optimize this cipher.

Both Salsa-20 and Phelix have a structure similar to the structure of modern hash functions, and use similar internal operations: fixed-length rotations, additions mod 2^{32}, xor operations, etc. These operations simplify and speed up software implementations of both ciphers, especially on 32-bit platforms. It is worth noticing that both ciphers have been selected to Phase 2 as the *Focus Profile 1* candidates, i.e., as the leading candidates optimized for high speed implementations in *software*.

From the hardware point of view, a wide data path, consisting of 5 x 32 = 160 bits in Phelix, and 16 x 32 = 512 bits in Salsa20, leads to a relatively large circuit area, especially in the basic iterative architecture, known well from the hardware implementations of block ciphers and hash functions[7-9], and implied by the cipher specification.

The similarity to hash functions is not accidental; actually Salsa20 is described in the specification as a hash function used in the counter mode. The difference between the basic hardware architectures of Phelix and Salsa20 is that Phelix produces one 32-bit block of the keystream every clock cycle, while Salsa20, produces a large 512-block of the keystream every 10 clock cycles. This difference can be made insignificant for an end user by implementing an output buffer in Salsa20, refreshed every 10 clock cycles with a new output from the hash function, and read serially, 64-bits of the keystream at a time. In case this buffer is not emptied in time, the operation of the hash function is stalled.

The possible optimizations of the hardware implementations of Phelix and Salsa20 are aimed at reducing the circuit area without considerably affecting the circuit throughput. In Phelix, the area can be reduced by implementing a half-block function, instead of the block function, as a combinational logic, and executing the block function in two consecutive clock cycles. Since the critical path through the combinational logic is reduced by a factor close to two, and the number of clock cycles is multiplied by two, the overall effect on the circuit throughput may be limited. At the same time, the circuit area can be reduced considerably. This area could be further reduced by sharing a half-block function between encryption and key-mixing.

In Salsa20, the internal structure permits folding the internal combinational logic by a factor of 2, 4, or 8. The factor of two corresponds to executing the columnround and rowround using the same logic,

consisting of four instantiations of quaterround. Therefore, we refer to this architecture as a 4 x quaterround architecture. The factor of eight, corresponds to implementing only one instantiation of the quaterround in combinational logic, and using eight clock cycles to implement the entire doubleround. We refer to this architecture as a 1 x quaterround architecture.

Due to the time limitations, the optimized architectures of Salsa have not been fully implemented within the duration of the students' project, and as a result they are not explored in this version of the paper.

4. Results

The results of our FPGA implementations are summarized in Tables 4-9. In all cases the devices from the Xilinx Spartan 3 family are used. The devices from within a family are chosen in such a way that a selected FPGA is capable of holding the most area-consuming and the most pin-consuming architecture of the given cipher. In case of Trivium and Salsa20, the primary limitation comes from the number of pins required by the fastest considered architectures. All timing results are based on the minimum clock period after placing and routing obtained from the static timing analysis and verified using timing simulation.

Table 4 Performance of Grain for different values of the parallelization factor *d*
Xilinx Spartan 3, xc3s50pq208-5 [768 CLB slices]

Parallelization factor *d*	Maximum clock frequency	Minimum key setup time for *k=d*		Maximum throughput		Area		Throughput to area ratio	
	MHz	cycles	ns	Mbit/s	x basic	CLB slices	x basic	Mbit/s / CLB slices	x basic
1 (basic)	193	304	1575	193	1.0	122	1.0	1.58	1.0
2	168	152	905	336	1.7	147	1.2	2.29	1.4
4	170	76	447	680	3.5	173	1.4	3.93	2.5
8	161	38	236	1288	6.7	244	2.0	5.28	3.3
16	155	19	123	2480	12.8	356	2.9	6.97	4.4

Table 5 Performance of Trivium for different values of the parallelization factor *d*
Xilinx Spartan 3, xc3s400fg320-5 [3584 CLB slices]

Parallelization factor *d*	Maximum clock frequency	Minimum key setup time for *k=d*		Maximum throughput		Area		Throughput to area ratio	
	MHz	cycles	ns	Mbit/s	x basic	CLB slices	x basic	Mbit/s / CLB slices	x basic
1 (basic)	201	1312	6527	201	1.0	188	1.00	1.07	1.00
2	202	656	3248	404	2.0	189	1.01	2.14	2.00
4	203	328	1616	812	4.0	199	1.06	4.08	3.82
8	193	164	850	1544	7.7	199	1.06	7.76	7.26
16	191	82	429	3056	15.2	227	1.21	13.46	12.59
32	202	41	203	6464	32.2	264	1.40	24.48	22.90
64	190	21	108	12160	60.5	388	2.06	31.34	29.31

Table 6 Performance of A5/1 for different values of the parallelization factor *d*
Xilinx Spartan 3, xc3s50pq208-5 [768 CLB slices]

Parallelization factor *d*	Maximum clock frequency	Minimum key setup time for *k*=*d*		Maximum throughput		Area		Throughput to area ratio	
	MHz	cycles	ns	Mbit/s	x basic	CLB slices	x basic	Mbit/s / CLB slices	x basic
1 (basic)	174	186	1069	174	1.0	57	1.0	3.05	1.0
3	114	63	553	342	2.0	142	2.5	2.41	0.8
4	79	47	595	316	1.8	287	5.0	1.10	0.4

Table 7 Performance of Phelix for various architectures
Xilinx Spartan 3, xc3s200ft256-5 [1920 CLB slices]

Basic function implemented using combinational logic	Maximum clock frequency	Minimum key setup time for *k*=*d*=32		Maximum throughput		Area		Throughput to area ratio	
	MHz	cycles	ns	Mbit/s	x basic	CLB slices	x basic	Mbit/s / CLB slices	x basic
block	46	28	609	1472	1.00	1402	1.00	1.05	1.00
half-block	52	44	846	832	0.57	1197	0.85	0.70	0.66

For each cipher and the particular architecture we report maximum clock frequency in MHz, maximum encryption/decryption throughput in Mbit/s, area in the number of CLB slices, and the throughput to area ratio. Additionally, we report the minimum key setup time that includes the key and the IV loading time and any additional initialization operations required by the cipher specification.

In Tables 4-6, we compare the basic minimum-area architectures of Grain, Trivium, and A5/1, with the optimized *d*-parallel architectures discussed in Section 3. The parameter *d* is a parallelization factor that determines the number of bits of the keystream produced per clock cycle. The parameter *k*, which is the number of bits of the key and the IV loaded to the internal state per clock cycle, is selected to be equal to the value of *d*. This way, the increase in the circuit throughput is accompanied by the corresponding reduction in the key setup time.

For the maximum throughput, area, and the throughput to area ratio, we show the relative change compared to the basic architecture. One can see that the largest improvement in the maximum throughput and the maximum throughput to area ratio is possible in Trivium. In this cipher, for the parallelization factor *d*=64, the throughput increases by a factor of 60, and the throughput to area ratio by a factor of 29. These improvements are several times smaller in case of Grain, and in A5/1 they concern only throughput, and not the throughput to area ratio.

In Table 7, the results for the basic and the optimized architectures of Phelix are presented. For this cipher, the basic architecture is optimum from the point of view of the throughput and throughput to area ratio. The optimization is aimed at reducing the circuit area, and succeeds by producing the circuit smaller by 15% compared to the basic architecture.

In Tables 8 and 9, we characterize and compare the best architectures of all six ciphers, selected from the point of view of minimum area (Table 8), and the maximum throughput to area ratio (Table 9). The ciphers are listed in the order of their performance, according to the given optimization criterion. For the minimum area implementations, Grain is the best among the five considered

eSTREAM candidates. It outperforms Trivium by 54%, Mickey-128 by a factor of over two, Phelix by a factor of almost 10, and Salsa20 by a factor of over 12

Table 8 Comparison of architectures optimized for minimum area
Xilinx Spartan 3 family

Cipher	Maximum clock frequency	Minimum key setup time		Maximum throughput		Area		Throughput to area ratio	
	MHz	cycles	ns	Mbit/s	/ Grain	CLB slices	/ Grain	Mbit/s / CLB slices	/ Grain
A5/1 (d=1, k=1)	174	186	1069	174	0.90	**57**	**0.47**	3.05	1.93
Grain (d=1, k=1)	193	304	1575	193	1.00	**122**	**1.00**	1.58	1.00
Trivium (d=1, k=1)	201	1312	6527	201	1.04	**188**	**1.54**	1.07	0.68
Mickey-128 (d=1, k=1)	156	416	2667	156	0.81	**261**	**2.14**	0.60	0.38
Phelix (d=32, k=32) half-block	52	44	846	832	4.31	**1197**	**9.81**	0.70	0.44
Salsa20 (d=64, k=64) doubleround	23.5	5	213	1203	6.23	**1615**	**13.24**	0.75	0.47

Table 9 Comparison of architectures optimized for the maximum throughput to area ratio
Xilinx Spartan 3 family

Cipher	Maximum clock frequency	Minimum key setup time		Maximum throughput		Area		Throughput to area ratio	
	MHz	cycles	ns	Mbit/s	Trivium /Cipher	CLB slices	Cipher/ Trivium	Mbit/s / CLB slices	Trivium /Cipher
Trivium (d=64, k=64)	190	21	108	12160	1.0	388	1.00	**31.34**	**1.0**
Grain (d=16, k=16)	155	19	123	2480	4.9	356	0.92	**6.97**	**4.5**
A5/1 (d=1, k=1)	174	186	1069	174	69.9	57	0.15	**3.05**	**10.3**
Phelix (d=32, k=32) block	46	28	609	1472	8.3	1402	3.61	**1.05**	**29.8**
Salsa20 (d=64, k=64) doubleround	23.5	5	213	1203	10.1	1615	4.16	**0.74**	**42.1**
Mickey-128 (d=1, k=1)	156	416	2667	156	77.9	261	0.67	**0.60**	**52.4**

Table 10 Comparison of architectures optimized for minimum area
ASIC 90 nm TCBN90G TSMC library

Cipher	Maximum clock frequency	Minimum key setup time		Maximum throughput		Area		Throughput to area ratio	
	MHz	cycles	ns	Mbit/s	/ Grain	μm^2	/ Grain	Mbit/s / μm^2	/ Grain
A5/1 (d=1, k=1)	685	186	272	685	1.21	**1985**	**0.40**	0.345	3.00
Grain (d=1, k=1)	565	304	538	565	1.00	**4911**	**1.00**	0.115	1.00
Trivium (d=1, k=1)	840	1312	1562	840	1.49	**7428**	**1.51**	0.113	0.98
Mickey-128 (d=1, k=1)	457	416	910	457	0.81	**16232**	**3.31**	0.028	0.24
Phelix (d=32, k=32) half-block	316	44	139	5056	8.95	**53232**	**10.84**	0.095	0.83

Table 11 Comparison of architectures optimized for the maximum throughput to area ratio
ASIC 90 nm TCBN90G TSMC library

Cipher	Maximum clock frequency	Minimum key setup time		Maximum throughput		Area		Throughput to area ratio	
	MHz	cycles	ns	Mbit/s	Trivium/ Cipher	μm^2	Cipher/ Trivium	Mbit/s / μm^2	Trivium /Cipher
Trivium (d=64, k=64)	800	21	26	51200	1.0	13440	1.00	**3.810**	**1.0**
Grain (d=16, k=16)	495	19	38	7920	6.5	10548	0.78	**0.751**	**5.1**
A5/1 (d=4, k=4)	402	186	463	1606	31.9	3590	0.27	**0.447**	**8.5**
Phelix (d=32, k=32) half-block	316	44	139	5056	10.1	53232	3.96	**0.095**	**40.1**
Mickey-128 (d=1, k=1)	457	416	910	457	112.0	16232	1.21	**0.028**	**135.3**

Among the architectures optimized for the maximum throughput to area ratio, Trivium outperforms all other ciphers by a wide margin. Its throughput to area ratio is about 4.5 times higher than in Grain, 30 times higher than in Phelix, 42 times higher than in Salsa20, and 52 times higher than in Mickey-128. The advantage of Trivium is also very evident in terms of the throughput that reaches about 12 Gbit/s, and exceeds that of any other cipher by at least a factor of four.

The old standard A5/1 wins with all new eSTREAM candidates in terms of the minimum area, but it is worse than Trivium and Grain in terms of the throughput to area ratio. Additionally, it should be remembered that this cipher is long broken, and considered highly insecure.

In Tables 10 and 11, we present the similar comparison with the same codes implemented using the standard-cell ASIC approach. The TSMC 90 nm TCBN90G ASIC library is used for the synthesis and timing analysis. All results are post-synthesis only, and could change if the full back-end design (layout) was completed. The interconnect delays are estimated in the post-synthesis analysis using so called wireload model, which predicts these delays based on the number of gate inputs driven by each node, and statistical data concerning similar circuits implemented in the same technology [5].

The ranking of algorithms remains the same as in FPGA technology, with even larger differences between the best ciphers in each category and the remaining candidates.

In Table 12, we summarize the speed-up of the ASIC implementations vs. the corresponding FPGA implementations. In both cases the same underlying 90 nm semiconductor technology is used. The speed-up ranges between about 3 for the optimized architecture of Grain and over 6 for the optimized architecture of Phelix. This speed up is somewhat larger than the one earlier observed for equivalent implementations of block ciphers, such as AES and DES, where it varied between 1.5 and 3 [6]. The source of this speed up is the size overhead and extra delays introduced to the FPGA implementations by the reconfigurable cells and interconnects.

Table 12 Speed-up of a 90 nm TSMC standard-cell ASIC implementation over the Spartan 3 FPGA implementation

Cipher	Clock frequency in Spartan 3 FPGAs	Clock frequency in ASICs	ASIC vs. FPGA frequency ratio
	MHz	MHz	
Trivium (d=64, k=64)	190	800	4.2
Grain (d=16, k=16)	155	495	3.2
A5/1 (d=4, k=4)	79	402	5.1
Phelix (d=32, k=32) half-block	52	316	6.1

5. Comparison with previous work

In [10], eight eSTREAM candidates are compared in terms of their hardware efficiency based on the results of the ASIC implementation in 0.25 μm 5-metal CMOS technology. Among these eight candidates, three - Grain, Mickey, and Trivium - are the same as those in our study. The relative performance of these three algorithms reported in [10] is very similar to their relative performance described in this paper.

In [11], six eSTREAM candidates and AES, with several alternative architectures per each cipher, are compared using Xilinx Spartan 2 FPGAs, Altera Cyclone FPGAs, and ASIC 0.13 μm standard cell process. Among these ciphers, Grain, Trivium, and Phelix are the same as those in our group. The relative performance of these algorithms reported in [11] is very close to their relative performance described in our study.

6. Summary and conclusions

In this paper, we compare and contrast five leading Phase 2 Profile 2 eSTRAM candidates from the point of view of the hardware implementation efficiency. We also compare these ciphers vs. an old GSM encryption algorithm A5/1.

One of the most important findings of our study is that the relative differences between eSTREAM candidates in terms of all hardware performance measures are huge, much bigger than it was the case for block ciphers competing in the second round of the AES contest [7, 8].

Trivium and Grain outperform all other considered eSTREAM candidates in terms of the two most important optimization criteria, minimum area and maximum throughput to area ratio, by a factor of at least two. The only exception is a relatively smaller advantage of Trivium over Mickey-128 in terms of the area in the FPGA implementation.

In general, stream ciphers based on linear and non-linear shift registers once again show their advantage in terms of hardware efficiency over newer more complex designs intended to be efficient in both software and hardware.

Assuming no progress in the cryptanalysis of Trivium or Grain, one or both of these ciphers should be declared the winners of the eSTREAM competition.

Acknowledgments

The authors would like to thank all students in the Fall 2006 GMU ECE 545 Introduction to VHDL class for their effort on the development of hardware implementations of eSTREAM candidates, and in particular we would like to thank students who contributed their codes to this project: Son T. Nguyen (Grain), Lalitha Chikkam (Mickey-128), Chethan Ananth and Bhupathi Venkata N. Kakarlapudi (Phelix), Marcello Brito (Salsa20), and Sterling Brandon Stewart (A5/1).

References

[1] eSTREAM Phase 2 webpage, available at http://www.ecrypt.eu.org/stream/index.html

[2] Mark Briceno, “A pedagogical Implementation of A5/1,” available at http://jya.com/a51-pi.htm

[3] G. Kostopoulos, N. Sklavos, M.D. Galanis, and O. Koufopavlou, “VLSI Implementation of GSM Security: A5/1 and W7 Ciphers,” available at http://www.vlsi.ee.upatras.gr/~gkostop/Giorgos_WoWCAS.pdf

[4] M. D. Galanis, P. Kitsos, G. Kostopoulos, N. Sklavos, O. Koufopavlou, and C.E. Goutis, “Comparison of the Hardware Architectures and FPGA Implementations of Stream Ciphers,” available at http://www.vlsi.ee.upatras.gr/~mgalanis/pubs/icecs04_stream.pdf

[5] Webpage of ECE 545, Introduction to VHDL, Fall 2006, available at http://ece.gmu.edu/courses/ECE545/index.htm

[6] K. Gaj, “FPGA and cryptography: Is marriage in the cards?”, Proc. 2nd International Workshop on Cryptographic Architectures Embedded in Reconfigurable Devices - CryptArchi 2004, Abbey La Bussiere near Dijon, June 16 – 18, 2004.

[7] K. Gaj and P. Chodowiec, "Comparison of the hardware performance of the AES candidates using reconfigurable hardware," Third Advanced Encryption Standard (AES) Candidate Conference, New York, April 13-14, 2000.

[8] K. Gaj and P. Chodowiec, "Fast implementation and fair comparison of the final candidates for Advanced Encryption Standard using Field Programmable Gate Arrays," Proc. RSA Security Conf. - Cryptographer's Track, San Francisco, CA, April 8-12, 2001, pp. 84-99. Available at http://ece.gmu.edu/crypto/publications.htm

[9] R. Lien, T. Grembowski, K. Gaj, “A 1 Gbit/s Partially Unrolled Architecture of Hash Functions SHA-1 and SHA-512,” LNCS 2964, RSA Conference 2004, Cryptographers' Track, CT-RSA 2004, San Francisco, CA, Feb. 2004, pp. 324-338.

[10] F.K. Gürkaynak, P. Luethi, N. Bernold, R. Blattmann, V. Goode, M. Marghitola, H. Kaeslin, N. Felber and W. Fichtner, "Hardware Evaluation of eSTREAM Candidates: Achterbahn, Grain, MICKEY, MOSQUITO, SFINKS, Trivium, VEST, ZK-Crypt", available at http://www.ecrypt.eu.org/stream/hw.html

[11] T. Good, W. Chelton and M. Benaissa, "Review of stream cipher candidates from a low resource hardware perspective", available at http://www.ecrypt.eu.org/stream/hw.html