Trivium hardware implementations for power reduction

J. M. Mora-Gutiérrez*,†, C.J. Jiménez-Fernández and M. Valencia-Barrero

Institute of Microelectronics of Seville (IMSE-CNM), CSIC–University of Seville, Seville, Spain

SUMMARY

This paper describes the use of parallelization techniques to reduce dynamic power consumption in hardware implementations of the Trivium stream cipher. Trivium is a synchronous stream cipher based on a combination of three non-linear feedback shift registers. In 2008, it was chosen as a finalist for the hardware profile of the eSTREAM project. So that their power consumption values can be compared and verified, the proposed low-power Trivium designs were implemented and characterized in 350-nm standard-cell technology with both transistors and gate-level models, in order to permit both electrical and logical simulations. The results show that the two designs decreased average power consumption by between 15% and 25% with virtually no performance loss and only a slight overhead (about 5%) in area.

KEY WORDS: Trivium; stream cipher; low power; lightweight cryptography; hardware implementation

1. INTRODUCTION

In the coming years, most communications systems in low-complexity devices with applications in portable health care or the Internet of things will use cryptographic techniques to ensure inviolability and confidentiality in data management. Hardware implementation in application-specific integrated circuit (ASIC) devices will require not only cryptographic algorithms but also algorithms for lightweight cryptography [1]. And in hardware implementations, the important measurements for evaluating lightweight properties are chip size and power consumption [2, 3].

Ciphers used in this type of cryptography include block ciphers and stream ciphers [4]. This article focuses on the latter. Stream ciphers are generally much faster than block ciphers, and they use less hardware resources, making them an ideal alternative when high throughput, low gate counts, and low power consumption are priority requirements.

The initiative known as eSTREAM [5] identified and published three new algorithms specially designed to ensure good performance in hardware (Grain, Mickey, and Trivium). These stream ciphers are already being used in embedded systems [1], wireless communications [6], and battery-powered and passively powered devices [7], where it is critical to have an algorithm that minimizes power consumption.

The objective of this work is to propose low-power ASIC implementations of Trivium based on standard-cell libraries in complementary metal–oxide–semiconductor (CMOS) process technology. Analysis of the Trivium algorithm suggests that parallelization is the most appropriate technique to

achieve a reduction in power consumption [8]. The parallelization technique was introduced by Schneider, Von Kaenen, and Piquet in 1995 [9].

In literature, few contributions about analyzing and reducing Trivium power consumption have been published. In field-programmable gate array implementations, some summaries of power results for eSTREAM candidates including Trivium were reported in [10, 11], but no techniques specifically aimed at reducing its power consumption were applied. In ASIC implementation, power results for a Trivium in a 130-nm CMOS technology were described in [12, 13] where 227 and 175 μW of average power were obtained at 10 MHz. Another set of power results for a Trivium in 130 and 350 nm was shown in [14], where 337 and 641 μW were obtained at 5 MHz. In [15], power consumption in a radix-16 Trivium optimized for passively powered devices was reduced by applying clock gating [16] and sleep mode logic as a means of reducing the effective clock frequency. Twenty-two clock cycles were needed to generate a 16-bit key stream, and source current values below 1 μA at 100 KHz and 1.5 V were obtained in 350-nm technology.

This paper focuses on Trivium hardware implementations for low-power applications, describing two different parallelization alternatives. The first alternative was the mixed-parallel low-power (MPLP) Trivium implementation, where parallelization was applied to flip-flops unaffected by non-linear feedback paths. The second was the full-parallel low-power (FPLP) implementation, where the parallelization technique was applied to all the flip-flops in the Trivium stream cipher, even though this meant redesigning non-linear feedback paths. This alternative was applied in an earlier study [17] in which good results were obtained for a Trivium even though only results from logical simulations were presented. That low-power implementation of Trivium was improved and updated to the FPLP version.

The applied technique reduces the internal flip-flop switching activity factor while maintaining the same external frequency, thus relating power reduction to the switching activities in the Trivium flip-flops.

In this work, we compared logical results with electrical simulations in a standard-cell CMOS technology that included transistor models for evaluating and comparing the accuracy of the power measurements. Our study also presents another low-power (MPLP) implementation alternative, which produced good results despite being less complex.

To compare the benefits of each of the proposed solutions, quantitative measurements of power consumption were made in the different designs. For this purpose, a detailed power consumption study was carried out at logic and electrical levels in a 350-nm technology that incorporated both transistors and gate-level models in order to permit both electrical and logical simulations.

The results show that this technique makes it possible to reduce dynamic power and average current by between 30% and 58%, with no performance loss and only a slight penalty in area (less than 5%).

The paper is organized as follows. Section 2 briefly describes the Trivium algorithm and its hardware implementation. Section 3 presents the architecture for an MPLP Trivium implementation with a non-full-parallel shift register, along with the corresponding power reduction results. Section 4 describes an alternative architecture for an FPLP Trivium implementation. Section 5 highlights the main differences between the MPLP and FPLP implementations, and, finally, some conclusions are presented in Section 6.

2. TRIVIUM HARDWARE IMPLEMENTATION

The hardware implementation of the Trivium [18] stream cipher is based on a 288-bit cyclic shift register (state register), with combinational logic to provide its non-linear feedback. It generates up to 2^{64} bits of pseudorandom key stream with an 80-bit secret key and an 80-bit initialization vector (IV).

As can be seen in Figure 1, the Trivium algorithm implementation, which generates one key stream bit in each clock cycle, comprises three shift registers of different lengths and combinational logic for the exclusive-or sum and the AND operations. The lengths of the shift registers are not the same; the first register has 93 bits, the second has 83, and the third has 111.

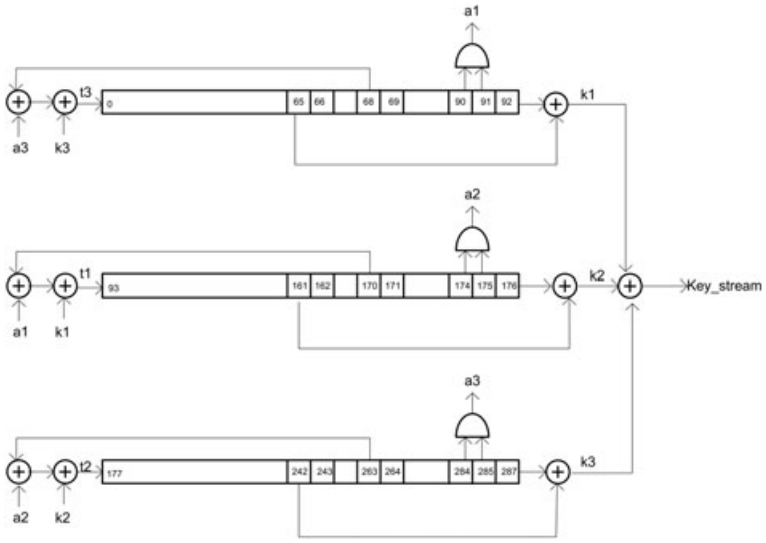

Figure 1. Trivium stream cipher schematic.

The state register is loaded with a secret key, an IV, and some ones and zeros to initialize the state of the Trivium. Once loaded, the state register must be shifted 1152 times (4×288) before a valid key stream can be obtained. The output *key stream* is an exclusive-or operation of signals from the three shift registers.

With this architecture and operation mode, most of the power is consumed by the flip-flops of the three shift registers. To reduce power consumption, we therefore focused on the shift register structure. Dynamic power depends on the switching activity factor (which represents the average fraction of clock cycles in which a signal transition occurs), clock frequency, supply voltage, and output capacitance. The more the logic transitions in the output, the greater the increase in switching power.

In our proposal, dynamic power in the Trivium shift registers is reduced by decreasing the switching activity. This is carried out by applying the shift register parallelization technique, while maintaining the same frequency and supply voltage.

To accurately estimate power consumption in the Trivium stream cipher, electrical and logical simulations were carried out in the different designs with the layout data. The post-layout netlist contained the clock buffer in the clock trees and the core cells, so the power consumption was the summation of the logic gates and the clock buffers. The input/output cells were not included.

Electrical-simulation-based analysis makes it possible to calculate power more accurately and in more detail than using logical simulation, although it also has the huge disadvantage of being extremely time-consuming.

A 350-nm CMOS process technology was chosen because it has logic and transistor-level models capable of performing logical and electrical simulations.

3. MIXED-PARALLEL LOW-POWER TRIVIUM

The parallelization technique cannot be applied directly to all the flip-flops in the Trivium state register because the outputs of some of them are involved in logical operations. In this version, the technique was applied only to the less significant bits of each shift register not used in the feedback. The bits in question are bits 0 to 63 in the first shift register, bits 93 to 160 in the second shift register, and bits 177 to 240 in the third shift register (196 out of 288 bits in the state register).

The application of the parallelization technique required a slight hardware modification in each shift register of the representation shown in Figure 1. As can be seen in Figure 2, the bits of the shift registers not involved in feedback or combinational operations were divided into two shift registers

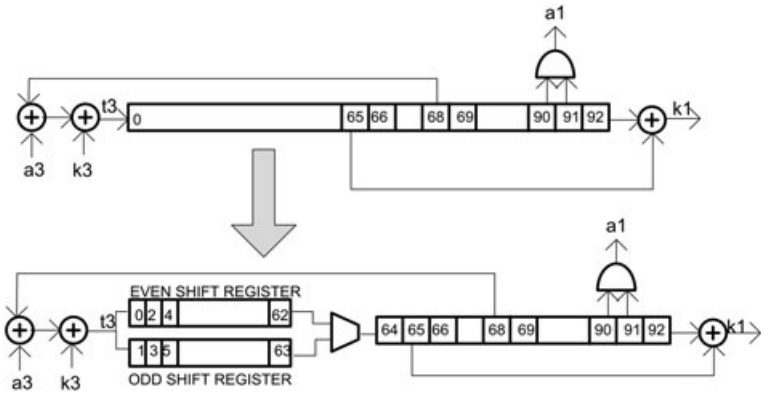

Figure 2. Shift register parallelization schematic.

denominated the odd and even shift registers, respectively. This modification made it necessary to introduce a flip-flop to generate a half-frequency clock. A multiplexer was also needed, to select the least significant bit in each shift register. The MPLP implementation is shown in Figure 3.

The state register was loaded in parallel with a secret key and an initialization vector, the even registers being loaded with the rising edge and the odd registers with the falling edge of an internal clock with half the frequency of the input clock.

The MPLP and standard Trivium implementations were described in VHDL, synthesized with *Design Vision* (*Synopsys*), and verified using the *ModelSim* simulation environment, with the same test vectors and using the same key and initialization vector as those presented in the Trivium reference files [5]. Simulations were also performed with different sets of keys and initialization vectors. The layout was implemented using the *Encounter Digital RTL-to-GDSII Implementation System* (*Cadence*).

As previously mentioned, dynamic power depends on the switching activity factor, which in turn represents the average fraction of clock cycles in which signal transition occurs. To estimate the effects of the parallelization technique on switching activity in the Trivium implementation, RTL simulations were performed to compare the number of transitions taking place in the state register flip-flops in each clock cycle. The average (*avg*) and maximum (*max*) numbers of flip-flops that changed in a clock cycle (0 to 1 and 1 to 0) are shown in Table I.

The average number of flip-flops switching their values in each clock cycle was 138 for the standard Trivium and 94 for the MPLP Trivium. The parallelization technique therefore reduces the number of flip-flop switching in each clock cycle by approximately 30%. This reduction was also seen in the

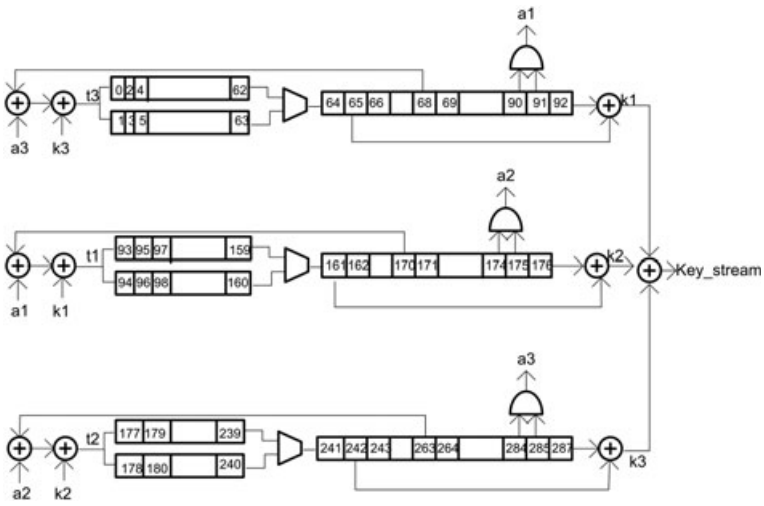

Figure 3. Mixed-parallel low-power Trivium schematic.

Table I. Switching registers per clock cycle in Trivium and MPLP Trivium.

	Switching register		Switching 0 to 1		Switching 1 to 0	
	Average	Maximum	Average	Maximum	Average	Maximum
Trivium	138	158	69	80	69	78
MPLP Trivium	94	117	47	58	47	59

MPLP, mixed parallel low power.

average number of transitions from levels 0 to 1 and 1 to 0. These results, combined with the fact that most power consumption occurs in the state register's flip-flops, suggest that a power reduction of about 30% is possible, although this needs to be corroborated by logical and electrical simulations.

The area report provided by the *Design Vision* synthesis tool for the MPLP and standard Trivium implementations is shown in Table II. The area of the MPLP version of Trivium implemented in a 350-nm technology is quite similar to the area of the standard version, because the numbers of flip-flops do not change (only one flip-flop is added, for the clock division) and the combinational area only has to accommodate three additional multiplexers. The conclusion is that the MPLP version has no area penalty in comparison with the standard version.

3.1. Mixed-parallel low power Trivium power consumption

To analyze power consumption more accurately and obtain a better idea of why it decreases, we measured current drawn from the power supply in post-layout electrical simulations.

Electrical simulations were carried out with a clock frequency of 25 MHz, but because of the complexity of the circuit and the computing time, it was not possible to simulate a high number of clock cycles. Figure 4 shows the waveform detail of the current supply for both implementations when simulating for 1 µs. Note that the power supply current peaks in the standard version of Trivium were very similar on both clock edges, whereas the MPLP Trivium showed a reduction in current peaks on the rising and falling edges of the clock, with peaks in the falling edges falling particularly sharply. Measurements from the electrical simulations show that the peak current for the MPLP Trivium was reduced by about 20% on the rising clock edge and by about 69% on the falling edge. Furthermore, average current consumption as measured by electrical simulation

Table II. Synopsys cell area report for Trivium and MPLP Trivium.

Synopsys report	350 nm		
	Trivium	MPLP Trivium	Reduction
Cell area	126 580 µm^2	129 165 µm^2	2%

MPLP, mixed parallel low power.

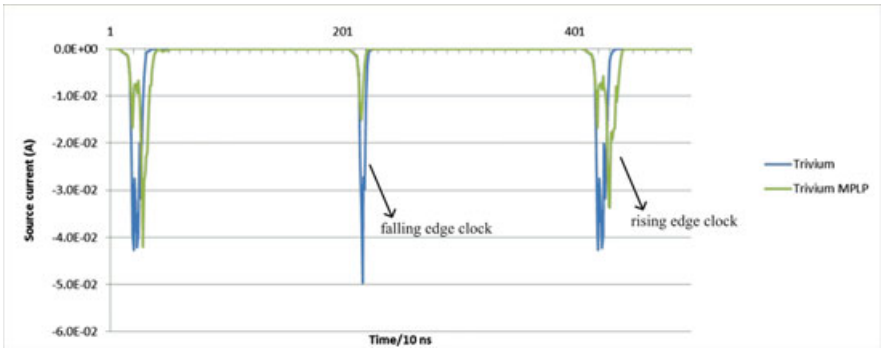

Figure 4. Power supply current post-layout electrical simulation. MPLP, mixed parallel low power.

decreased by about 25%, for the MPLP Trivium compared with the standard version. This is shown in Table III.

The reduction in power consumption shown in these results was slightly less than expected because of the reduced number of transitions in the flip-flops, shown in Table I. This is because clock tree power consumption is not considered in the table.

Dynamic power consumption was measured from logical simulations and compared with the electrical simulations. Logical simulations were carried out for more clock cycles because they are less time-consuming, although their results are less accurate than those of electrical simulations. Input patterns were the same in both simulations. The IV and key used were those presented in the Trivium reference files [5].

Power consumption was analyzed using *Encounter Digital Implementation System RTL to GDSII* tools with a switching activity file in a value change dump format. This file was generated with 1700 clock cycles (68-μs simulation) and a clock frequency of 25 MHz. As mentioned earlier, Trivium needs 1152 clock cycles to obtain a valid key stream. Capacitances and power models for wires and gates were taken from the technology library.

When the power consumption of the MPLP implementation was compared with that of the standard Trivium implementation, it was found that the dynamic power consumption of the MPLP version was about 25% lower than that of the standard version (Table IV). Again, the main reason for this reduction was the lower number of flip-flops changing in each clock cycle, as shown previously in Table I. This result is very similar to the measurements taken during the electrical simulation.

The parallelization of 196 of the 288 bits in the state register produced a power reduction of about 25% in the MPLP Trivium. If the parallelization technique could be applied to all 288 bits of the state register, an even greater reduction in consumption could be obtained. However, to do this, it was necessary to introduce some hardware modifications. A new, low-power, version incorporating such modifications is therefore presented in the following section.

4. FULL-PARALLEL LOW-POWER TRIVIUM

In the FPLP implementation, the parallelization technique was applied to all the flip-flops in the shift registers. This, however, required additional modifications in the structure of the Trivium, as discussed in [17].

Shift register parallelization of all the bits in the state register transforms each of the Trivium's shift registers into two half-length shift registers (odd and even). Figure 5 shows a schematic representation of the FPLP Trivium. The length of each shift register is indicated by the figures inside the odd and even registers.

Table III. Average power and current consumption measured by electrical simulation.

Electrical Simulation	Trivium	MPLP Trivium	Reduction (%)
Average power	4.09 mW	2.98 mW	25
Average current	1.22 mA	0.9 mA	25
Maximum peak current	51.2 mA	42.1 mA	17

MPLP, mixed parallel low power.

Table IV. Power consumption measured by logical simulation and the *Encounter* tool.

Power at 25 MHz at 3.3 V	350 nm		
	Trivium (mW)	MPLP Trivium (mW)	Reduction (%)
Dynamic	5.84	4.36	25
Switching	1.12	1.11	
Cell internal	4.72	3.24	31

MPLP, mixed parallel low power.

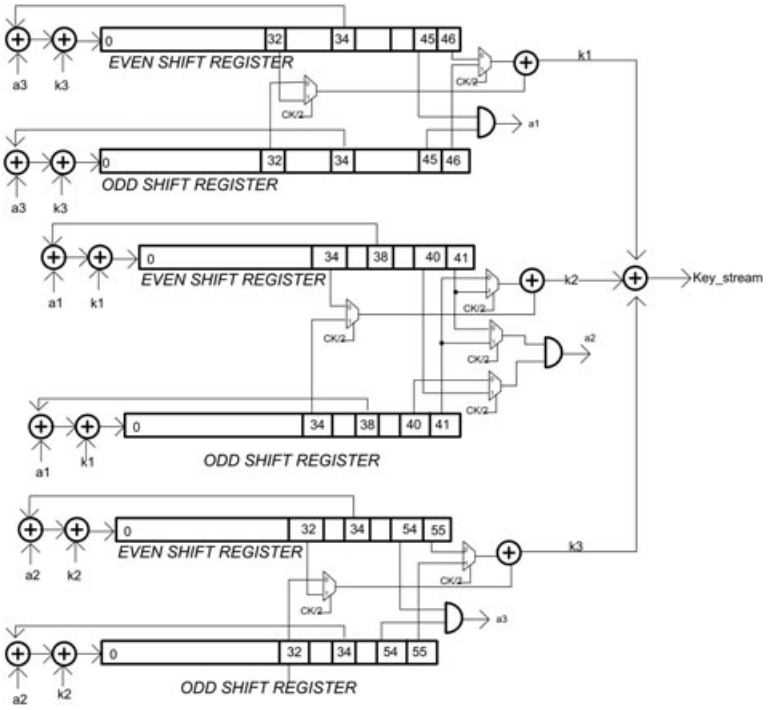

Figure 5. Full-parallel low-power Trivium schematic.

The generation of the input bits for each shift register and the generation of the key stream depend on bits stored in different positions in the shift registers. But the problem posed by this new structure is that the location of those bits depends on whether the clock cycle is odd or even. In one case, the bit to be retrieved is in the even register, and in another, it is in the odd register. So that the bits are correctly selected, glue logic must be introduced. This added logic basically involves the use of multiplexers, which, using the clock as the selection signal, will select the bit to be retrieved from the odd or even shift register (Figure 5).

The FPLP Trivium version was described and designed using VHDL. The resulting implementation was verified using the *ModelSim* simulation environment with a post-layout netlist. The FPLP implementation increases the number of the cells and nets because more multiplexers and combinational cells have to be added to implement the algorithm. Table V shows the Synopsys cell and net counts for a 350-nm technology. The FPLP version uses more cells (6.6%) and more nets (16%) than the standard Trivium and MPLP implementations.

As with the MPLP version, the amount of switching taking place in the shift register flip-flops was analyzed. In each clock cycle, the results for the average (*avg*) and maximum (*max*) number of flip-flops that change (from levels 0 to 1 and 1 to 0) were compared with those obtained for the standard version of the Trivium and are shown in Table VI.

As can be seen in the table, the average number of flip-flops changing their output in each clock cycle was 138 for the standard Trivium and 70 for the FPLP Trivium. This represents a reduction of 49%. This reduction also occurs in the average number of switches from levels 0 to 1 and 1 to 0.

Table V. Number of cells and nets reported by Synopsys in Trivium and FPLP Trivium.

Synopsys report	350 nm		
	Trivium	FPLP Trivium	Overhead (%)
Cell	617	748	6.6
Nets	792	921	16

FPLP, full parallel low power.

Table VI. Switching registers per clock cycle in Trivium and FPLP Trivium.

	Switching registers		Switching 0 to 1		Switching 1 to 0	
	Average	Maximum	Average	Maximum	Average	Maximum
Trivium	138	158	69	80	69	78
FPLP Trivium	70	86	35	43	35	43

FPLP, full parallel low power.

Regarding the area, the combinational area and the net area are inevitably larger in the FPLP Trivium because more multiplexers and combinational cells have to be added to implement the algorithm. Table VII shows the area estimation reports provided by the *Design Vision* synthesis tool for the FPLP and standard versions. The combinational area of the FPLP version is 19.1% larger than that of the standard version.

The non-combinational area, however, is quite similar in both designs because the numbers of flip-flops do not change (only one flip-flop is introduced, for the clock division). The FPLP version thus has a cell area penalty of about 4% while its net area increases by 8%.

4.1. Full-parallel low-power Trivium power consumption

As in the MPLP version, we closely analyzed the nature of power consumption in the FPLP version to identify exactly where power reduction occurs. The power consumption measurements were taken from electrical and logical simulations in Trivium layout circuits. The electrical simulations were carried out with a clock frequency of 25 MHz, simulating for 1 µs, as for the MPLP Trivium simulations.

Figure 6 shows the waveform detail of the power supply current flow for both implementations. Again, the power supply current peaks in the standard version of Trivium are very similar on both clock edges. In the FPLP Trivium, however, the current peaks rise on the rising edges of the clock and totally disappear on the falling edge.

Table VII. Synopsys area report for Trivium and FPLP Trivium.

Synopsys area report	350 nm		
	Trivium (μm^2)	FPLP Trivium (μm^2)	Overhead (%)
Combinational	26,990	32,159	19.1
Non-combinational	99,590	100,573	1
Cell	126,580	132,732	4
Net	17,559	19,017	8

FPLP, full parallel low power.

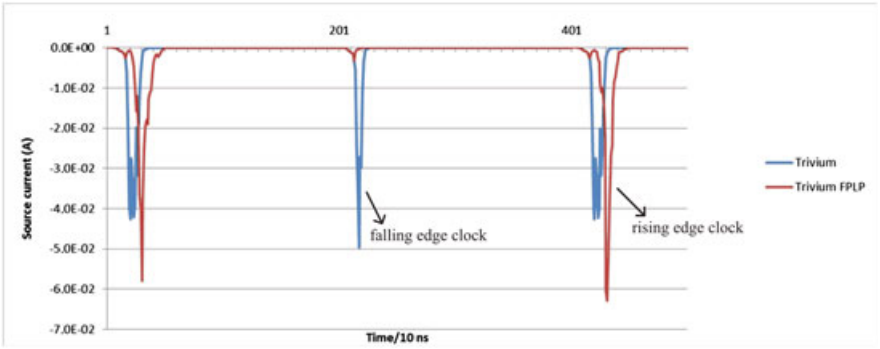

Figure 6. Power supply current post-layout electrical simulation. FPLP, full parallel low power.

Measurements from the electrical simulations showed that the peak current on the rising clock edge increased by about 25% in the FPLP Trivium implementation, because of the clock tree, but dropped sharply, by about 93%, on the falling clock edge. Furthermore, the average current consumption, as measured during the electrical simulation, decreased by 15% in the FPLP Trivium (Table VIII). The average power measurement produced by the electrical simulation indicates a power reduction of about 15% in the FPLP Trivium, as can be seen in the table.

As in Section 3.1, power consumption was again measured from logical simulations. The power consumptions of the two implementations (standard Trivium and FPLP Trivium) are shown in Table IX. When they are compared, the FPLP Trivium can be seen to have a cell dynamic power consumption 23% lower than that of the standard version. Switching power is very similar in both versions of Trivium.

This result differs slightly from the measurements produced by the electrical simulation, and the power reduction is lower than that obtained by estimating the number of flip-flop transitions by clock cycles because clock tree power consumption is now included.

Table VIII. Average power and current consumption measured by electrical simulation.

Electrical simulation	Trivium	FPLP Trivium	Reduction (%)
Average power	4.09 mW	3.39 mW	15
Average current	1.22 mA	1.03 mA	15
Maximum peak current	51.2 mA	63.5 mA	−25*

FPLP, full parallel low power.
*Means increasing.

Table IX. Power consumption measured by logical simulation and the *Encounter* tool.

Power at 25 MHz at 3.3 V	350 nm		
	Trivium (mW)	FPLP Trivium (mW)	Reduction (%)
Dynamic	5.84	4.46	23
Switching	1.12	1.13	
Cell internal	4.72	3.33	29

FPLP, full parallel low power.

Table X. Comparative summary of Trivium references.

Trivium	Dynamic power	Supply voltage (V)	Clock rate	Technology (nm)
Trivium [13]	175.1 μW	1.2	10 MHz	130
Trivium [12]	34.7 μW		1 MHz	130
Trivium [12]	227 μW		10 MHz	130
Trivium [12]	2.15 mW		100 MHz	130
Trivium [14]	337 μW	1.2	5 MHz	130
Trivium [14]	641 μW	3.3	5 MHz	350
Trivium radix-16 [15]	0.68 μA	1.5	100 kHz	350
Trivium [17]	1007 μW	1.8	25 MHz	180
FPLP [17]	712 μW	1.8	25 MHz	180
Trivium [17]	236 μW	1.2	25 MHz	130
FPLP [17]	178 μW	1.2	25 MHz	130
Trivium [17]	219 μW	1.2	25 MHz	90
FPLP [17]	179 μW	1.2	25 MHz	90
Trivium [this work]	5.8 mW	3.3	25 MHz	350
MPLP [this work]	4.3 mW	3.3	25 MHz	350
FPLP [this work]	4.4 mW	3.3	25 MHz	350

FPLP, full parallel low power; MPLP, mixed parallel low power.

5. COMPARISON BETWEEN MIXED-PARALLEL LOW-POWER AND FULL-PARALLEL LOW-POWER TRIVIUM

Compared with the standard version of Trivium, the MPLP version achieves a reduction in power consumption of about 25%, while the reduction achieved by the FPLP version is about 23–15%. Post-layout clock distribution has a negative effect on power consumption in the FPLP version. In this regard, clock paths and the clock buffer represent two important potential areas of improvement for the FPLP implementation.

Although the results obtained using electrical and logical simulations vary slightly in the FPLP Trivium, it can be concluded that both the MPLP and FPLP versions are able to reduce dynamic power.

Table X summarizes the power consumption of other Trivium implementations reported in literature, along with that achieved using our proposals. It is difficult to compare power consumption in the different Trivium implementations because the technologies used are different, despite their identical transistor sizes. If frequency and voltage in the 130-nm technology are scaled up, for example, the FPLP Trivium implementation [17] can be seen to have lower cell dynamic power consumption than the implementation described in [12, 13]. For the 350-nm technology, however, comparison is more difficult, not only because the technologies are different, but also because the low-power Trivium implemented in [15] is radix-16 and uses a reduced effective clock frequency, making scaling more indeterminate.

6. CONCLUSIONS

In this paper, two versions of low-power Trivium implementations using logic parallelization techniques (MPLP and FPLP) have been presented. The power consumption of both versions was estimated by electrical and logical simulation. Electrical simulations were possible because the 350-nm technology has transistor-level models of standard cells. Some recent technologies are not able to provide standard-cell libraries with transistor-level models.

The MPLP Trivium architecture offers a greater power reduction than the FPLP Trivium because of its post-layout clock distribution, in which the slightly less complex algorithm and logic used make the final area smaller. The technique proposed produced Trivium implementations with reductions in power consumption of between 25% and 15% and virtually no performance loss. With this technique, current peaks are much lower (more than 50%) on the falling edge and about 25–15% lower on the rising edge. The area penalty and cell number obtained with this technique is very low (less than 6%), while the reduction achieved in dynamic power consumption is noticeable.

ACKNOWLEDGEMENTS

This work was partially funded by Spanish government projects CITIES (TEC2010-16870), CESAR (TEC2013-45523-R), and LACRE (CSIC 201550E039).

REFERENCES

1. Manifavas C, Hatzivasilis G, Fysarakis K, Papaefstathiou Y. A survey of lightweight stream ciphers for embedded systems. *Security and Communication Networks* 2016; **9**(10):1226–1246. doi:10.1002/sec.1399.
2. Razmdideh R, Saneei M. Two novel low power and very high speed pulse triggered flip-flops. *International Journal of Circuit Theory And Applications* 2015; **43**:1925–1934. doi:10.1002/cta.2048.
3. Abed S, Mohd BJ, Al-bayati Z, Alouneh S. Low power Wallace multiplier design based on wide counters. *International Journal of Circuit Theory and Applications* 2012; **40**:1175–1185. doi:10.1002/cta.779.
4. Robshaw M, Billet O. New Stream Cipher Designs: The eSTREAM Finalists. In *Lecture Notes in Computer Science*, vol 4986. Springer-Verlag: Berlin Heidelberg, 2008.
5. eSTREAM: ECRYPT Stream Cipher Project, 2012. http://www.ecrypt.eu.org/stream
6. Gong J, Chen G, Li L, Li J. A secure authentication protocol for RFID based on Trivium. In *International Conference on Computer Science and Service System (CSSS)*, Nanjing, China, 2011; 107–109. doi: 10.1109/CSSS.2011.5974817
7. Kocheta M, Sujatha N, Sivakanya K, Srikanth R, Shetty S, Ananda Mohan PV. A review of some recent stream ciphers. In *International Conference on Circuits, Controls and Communications (CCUBE)*, Bengaluru, India, 2013; 1–6. doi: 10.1109/CCUBE.2013.6718558

8. Piquet C. *Low-power CMOS Circuits Technology, Logic Design and CAD Tools*. CRC/Taylor & Francis: Boca Raton, FL, 2006.
9. Schneider T, von Kaenel V, Piguet C. Low-voltage/low-power parallelized logic modules. In *Power and Timing Modeling for Performance of Integrated Circuits (PATMOS'95)*, 1995; 147–160.
10. Rogawski M. Hardware evaluation of eSTREAM Candidates: Grain, Lex, Mickey128, Salsa20 and Trivium. In *State of the Art of Stream Ciphers Workshop (SASC)*, Bochum, Germany, 2007.
11. Marmolejo-Tejada JM, Trujillo-Olaya V, Velasco-Medina J. Hardware implementation of Grain-128, Mickey-128, Decim-128 and Trivium. In *IEEE ANDESCON*, Bogotá, Columbia, 2010.
12. Good T, Chelton W, Benaissa M. Review of stream cipher candidates from a low resource hardware perspective. *SASC 2006 Stream Ciphers Revisited* 125, 2006.
13. Good T, Benaissa M. Hardware results for selected stream cipher candidates. *State of the Art of Stream Ciphers Workshop* (SASC 2007), eSTREAM. ECRYPT Stream Cipher Project, Report 2007/023, 2007.
14. Atani RE, Mirzakuchaki S, Atani SE, Meier W. On DPA-resistive implementation of FSR-based stream ciphers using SABL logic styles. *International Journal of Computers Communications & Control* 2008; **3**(4):324–335.
15. Feldhofer M. Comparison of low-power implementations of Trivium and Grain. *State of the Art of Stream Ciphers Workshop* (SASC 2007), eSTREAM. ECRYPT Stream Cipher Project, Report 2007/027, 2007.
16. Pouiklis G, Georgios CS. Clock gating methodologies and tools: a survey. *International Journal of Circuit Theory and Applications* 2015. doi:10.1002/cta.2107.
17. Mora-Gutiérrez JM, Jiménez-Fernández CJ, Valencia-Barrero M. Low power implementation of Trivium stream cipher. *PATMOS* 2012:113–120.
18. De Canniere C, Preneel B. Trivium: A stream cipher construction inspired by block cipher design principles. *State of the Art of Stream Ciphers Workshop* (SASC 2006), eSTREAM, ECRYPT Stream Cipher Project, Report 2006/021, 2006.