Fast implementations of secret-key block ciphers using mixed inner- and outer-round pipelining

Pawel Chodowiec, Po Khuon, and Kris Gaj
Electrical and Computer Engineering
George Mason University
4400 University Drive
Fairfax, VA 22030
(703) 993-1575
pchodowi@gmu.edu, pkhuon@gmu.edu, kgaj@gmu.edu

ABSTRACT

The new design methodology for secret-key block ciphers, based on introducing an optimum number of pipeline stages inside of a cipher round is presented and evaluated. This methodology is applied to five well-known modern ciphers, Triple DES, Rijndael, RC6, Serpent, and Twofish, with the goal to first obtain the architecture with the optimum throughput to area ratio, and then the architecture with the highest possible throughput. All ciphers are modeled in VHDL, and implemented using Xilinx Virtex FPGA devices. It is demonstrated that all investigated ciphers can operate with similar maximum clock frequencies, in the range from 95 to 131 MHz, limited only by the delay of a single CLB layer and delays of interconnects. Rijndael, RC6, Twofish, and Serpent achieve throughputs in the range from 12.1 Gbit/s to 16.8 Gbit/s; and Triple DES achieves the throughput of 7.5 Gbit/s. Because of the optimum speed to cost ratio, the proposed architecture seems to be very well suited for practical implementations of secret-key block ciphers using both FPGAs and custom ASICs. We also show that using this architecture for comparing hardware performance of secret-key block ciphers, such as AES candidates, operating in non-feedback cipher modes, leads to the more prudent and fairer analysis than comparisons based on other types of pipelined architectures.

General terms

Algorithms, Performance, Design, Security, Standardization.

Keywords

secret-key ciphers, fast architectures, pipelining, AES.

1. INTRODUCTION

Pipelining is a well-known technique used to speed up the operation of digital systems by processing multiple blocks of data at the same time. Traditionally, secret-key block ciphers, such as DES (Data Encryption Standard) [1], were designed to be implemented in hardware. As a result, they had a very simple and fast cipher round. This feature implied that only one type of pipelining, with rounds unrolled, and registers inserted between consecutive cipher rounds, was practical. We refer to this type of pipelining as outer-round pipelining.

The emergence of new block ciphers, optimized for software implementations, and based on elementary instructions of modern microprocessors, made this design methodology sub-optimum. A basic cipher round in modern ciphers such as Rijndael, RC6 and Twofish is quite complex, limiting the maximum clock frequency of their non-pipelined iterative hardware implementations. At the same time, the area necessary to repeat a single round of these ciphers within an integrated circuit may prohibit loop unrolling required for outer-round pipelining.

As a result, a new form of pipelining, with pipeline registers inserted inside of a cipher round became practical. We refer to this architecture as inner-round pipelining. The inner-round pipelining provides a substantial increase in the cipher speed at the cost of only small increase in the circuit area. Additionally, if the area available on the integrated circuit is large compared to the area used by the iterative architecture, the inner-round pipelining can be easily combined with the outer-round pipelining, leading to the fastest possible architecture of a given block cipher.

In this paper, we analyze the time-cost characteristics of the proposed architectures, and present the results of our hardware implementations of five secret-key block ciphers: RC6, Rijndael, Serpent, Twofish, and Triple-DES. The first four ciphers were the leading candidates for the new Advanced Encryption Standard (AES) [7]. From among these candidates, Rijndael was recently selected as a winner of the contest for the new American federal standard [7]. Triple DES is a current federal and banking standard and is planned to remain in use, along with AES, in the foreseeable future [1, 5].

All ciphers have been implemented first using the basic iterative architecture. This implementation was then extended with the pipeline registers inserted inside of the cipher round, which substantially increased the maximum clock frequency and the encryption throughput. Finally, all cipher rounds were unrolled, together with their internal registers, leading to the architecture with the maximum possible throughput.

2. PARAMETERS OF HARDWARE IMPLEMENTATIONS OF BLOCK CIPHERS

Hardware implementations of secret-key block ciphers can be characterized using several major parameters. *Encryption (decryption) throughput* is defined as the number of bits encrypted (decrypted) in a unit of time. Typically, the encryption and decryption throughputs are equal, and therefore only one parameter is reported. *Encryption (decryption) latency* is defined as the time necessary to encrypt (decrypt) a single block of plaintext (ciphertext). The encryption (decryption) latency and throughput are related by:

$$Throughput = block\_size \cdot number\_of\_blocks\_processed\_simultaneously \,/\, Latency \qquad (1)$$

In applications where the large amounts of data are encrypted or decrypted, throughput determines the total encryption/decryption time, and thus is the best measure of the cipher speed. In applications where a small number of plaintext (ciphertext) blocks is processed, the total encryption/decryption time depends on both throughput and latency.

Circuit area is another important parameter, which determines the cost of implementation and may impose a limit on the circuit architecture and speed. In FPGAs, it is common to express circuit area using the number of basic building blocks. In Xilinx Virtex FPGAs, these blocks are referred to as Configurable Logic Blocks (CLBs) or CLB slices (1/2 of CLB).

3. NEW METHODOLOGY FOR THE DESIGN OF SECRET-KEY BLOCK CIPHERS

3.1 Features of the new methodology

Traditional methodology for the design of high-performance implementations of secret-key block ciphers is shown in Fig. 1. This methodology is discussed among the others in [8, 9]. The basic iterative architecture, shown in Fig. 1a is implemented first, and its speed and area are determined. Based on these estimations, the number of rounds, *K*, that can be unrolled without exceeding the available area is found. This number must be a divisor of the total number of rounds, *#rounds*. The throughput and area of the circuit with partial outer-round pipelining increase proportionally to the value of *K*, as shown in Fig. 3; the encryption/decryption latency remains the same as in the basic iterative architecture,

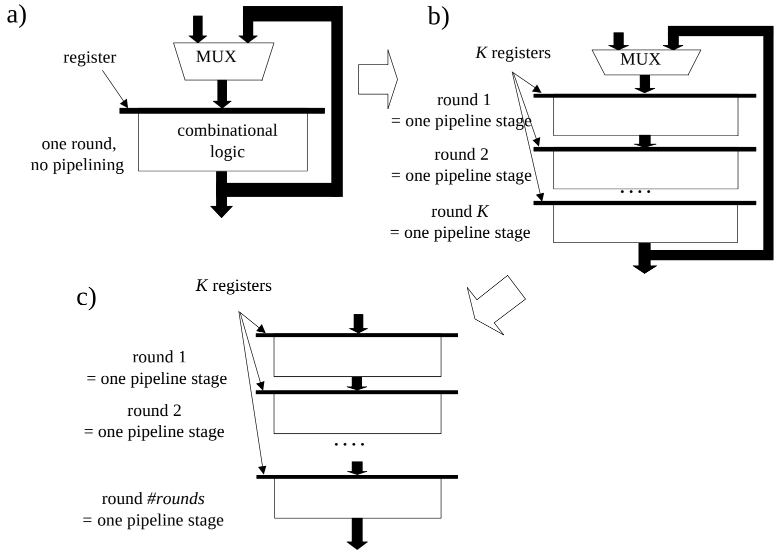

a)
register
MUX
combinational logic
one round, no pipelining
b)
K registers
MUX
round 1 = one pipeline stage
round 2 = one pipeline stage
....
round K = one pipeline stage
c)
K registers
round 1 = one pipeline stage
round 2 = one pipeline stage
....
round #rounds = one pipeline stage

Fig. 1 Traditional design methodology for secret-key block ciphers based on the outer-round pipelining.
a) basic iterative architecture b) partial (*K*-stage) outer-round pipelining; *K* < *#rounds* c) full outer-round pipelining.

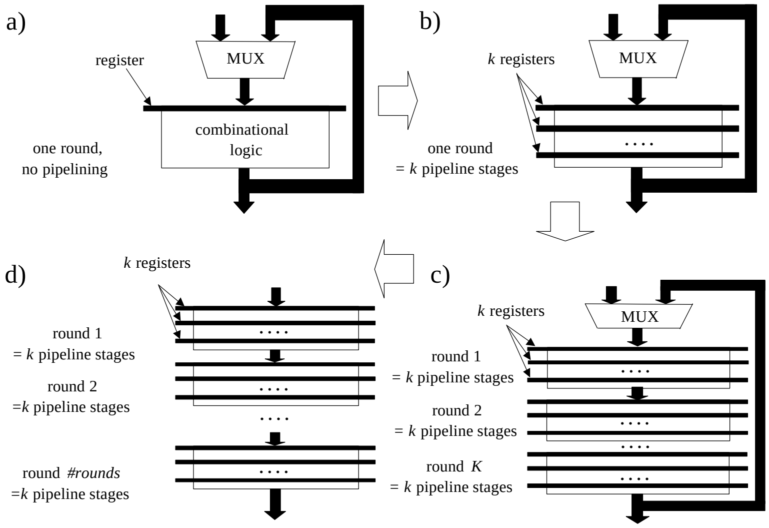


Fig. 2 New design methodology based on the mixed inner and outer-round pipelining.
a) basic iterative architecture b) inner-round pipelining, c) partial mixed inner and outer-round pipelining, $K < \#rounds$, d) full mixed inner and outer-round pipelining.

as shown in Fig. 4. If the available area is large enough to fit all cipher rounds, the feedback loop is no longer necessary, and full outer- round pipelining, shown in Fig. 1c, can be applied.

Our new methodology is shown in Fig. 2. The primary difference is that before loop unrolling, the optimum number of pipeline registers is inserted inside of a cipher round, as shown in Fig. 2b. The entire round, including internal pipeline registers is then repeated *K* times. The number of unrolled rounds *K* depends on the maximum available area or the maximum required throughput.

The primary advantage of the new methodology is shown in Fig. 3. Inserting registers inside of a cipher round significantly increases cipher throughput at the cost of only marginal increase in the circuit area. As a result, the throughput to area ratio increases until the number of internal pipeline stages reaches its optimum value k_{opt}. Inserting additional registers may still increase the circuit throughput, but the throughput to area ratio will deteriorate. The throughput to area ratio remains unchanged during the subsequent loop unrolling. The throughput of the circuit is given by

$$Throughput\ (K, k) = K \cdot block\_size \ / \ \#rounds \cdot \mathrm{T}_{\mathrm{CLKinner\_round}}\,(k) \qquad (2)$$

where k is the number of inner-round pipeline stages, K is the number of outer-round pipeline stages, and $\mathrm{T}_{\mathrm{CLKinner\_round}}\,(k)$ is the minimum clock period in the architecture with the k-stage inner-round pipelining.

For the given limit in the circuit area, mixed inner and outer-round pipelining offers significantly higher throughput compared to the pure outer-round pipelining (see Fig. 3). When the limit on the circuit area is large enough, all rounds of the cipher can be unrolled, leading to the highest possible value of the throughput given by

$$Throughput\ (\#rounds,\ k_{opt}) = block\_size \ / \ \mathrm{T}_{\mathrm{CLKmixed}}(k_{opt}) \qquad (3)$$

where k_{opt} is the number of inner-round pipeline stages optimum from the point of view of the throughput to area ratio, and $\mathrm{T}_{\mathrm{CLKmixed}}(k_{opt})$ is the minimum clock period in the architecture with the full mixed inner- and outer-round pipelining; $\mathrm{T}_{\mathrm{CLKmixed}}(k_{opt}) \approx \mathrm{T}_{\mathrm{CLKinner\_round}}(k_{opt})$.

The only side effect of our methodology is the increased latency of the circuit, i.e., the time necessary to encrypt a single block of data (see Fig. 4). This latency is given by

$$Latency(K, k) = \#rounds \cdot k \cdot \mathrm{T}_{\mathrm{CLKinner\_round}}\,(k) \qquad (4)$$

It does not depend on the number of rounds unrolled, *K*.

When the combinational portion of a cipher round is divided into *k* equal pipeline stages, the increase in latency is given by

$$DLatency(K, k) = \ \#rounds \cdot (k\text{-}1)(\mathrm{t_P} + \mathrm{t_{su}}) \qquad (5)$$

where $\mathrm{t_P}$ and $\mathrm{t_{su}}$ denote the propagation delay and the setup time of a register, respectively.
The increase in latency is typically small, in the range of single microseconds, and does not have any influence on the operation of the cryptographic system including a hardware implementation of the secret-key block cipher. This is particularly true for applications with a human operator present on at least one end of the secure communication channel.

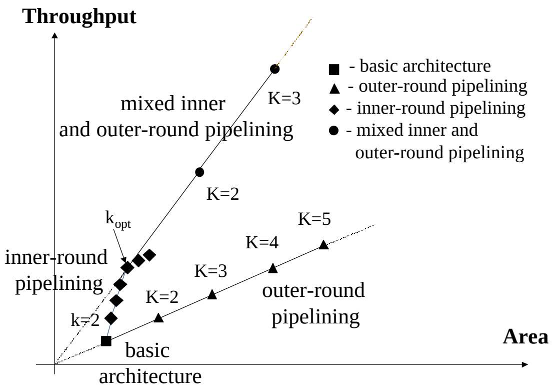


Fig. 3 Throughput vs. area dependence for the architectures developed according to the old and new design methodologies.

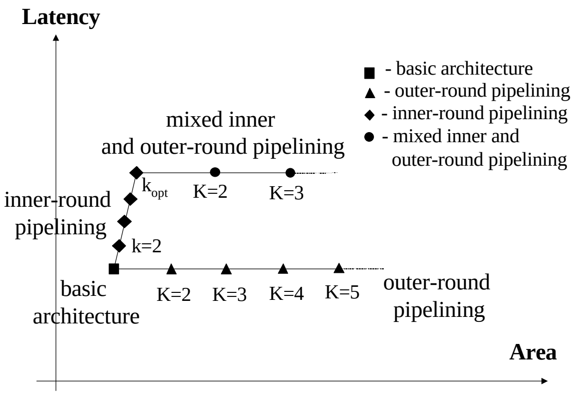


Fig. 4 Impact of the inner-round and outer-round pipelining on latency.

The input/output timing characteristics of three basic secret-key cipher architectures is shown in Fig. 5. In the basic iterative architecture, a new block of data must be fed into the system only once per *#rounds* clock cycles. In case of the inner round pipelining, there are periods of time when the input must be fed into the cryptographic core every clock cycle, even though an average input/output throughput is much lower (Fig. 5b). In the full mixed inner- and outer-round pipelining, input blocks are fed to the encryption unit every clock cycle (Fig. 5c).

3.2 Limits on the maximum clock frequency in the inner round pipelining

Throughput of the architecture with the mixed inner and outer-round pipelining is directly proportional to the maximum clock frequency for the inner round pipelining (see equation (2)). The following factors may limit the maximum clock frequency, $f_{CLKinner\_round}(k) = 1/T_{CLKinner\_round}(k)$ in this architecture:

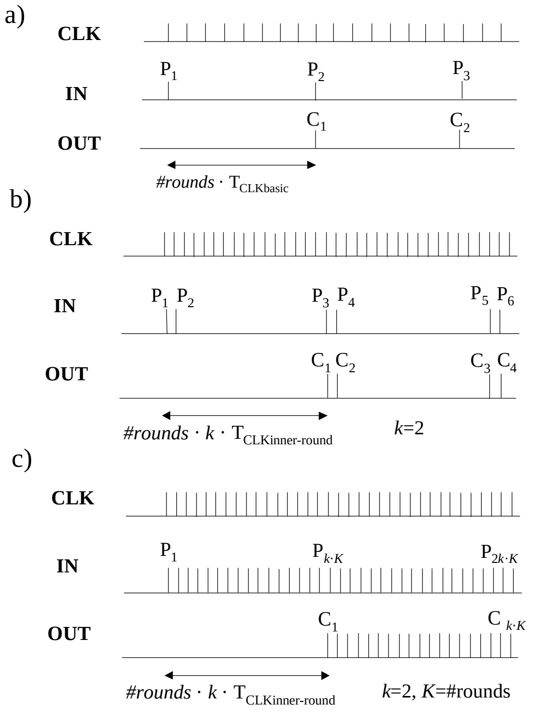


Fig. 5 Input/output timing characteristics of various architectures
a) basic iterative architecture b) mixed inner and outer-round pipelining; $K < \#rounds$ c) full mixed inner and outer-round pipelining

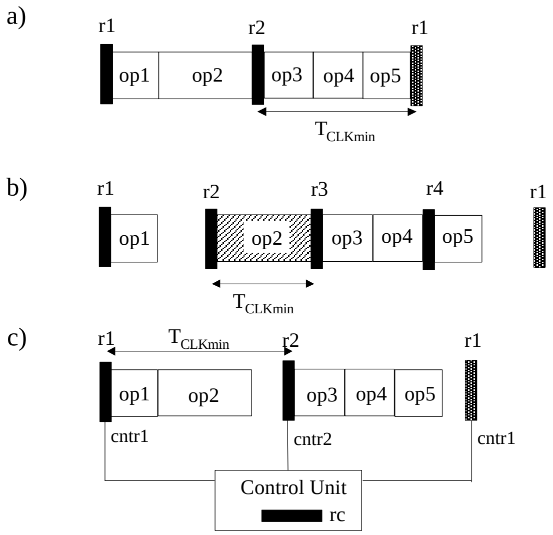


Fig. 6 Limits on the minimum clock period in the architecture with inner-round pipelining

a) ideal situation; evenly divided round , b) clock period limited by the largest indivisible operation, c) clock period limited by the control unit, i.e., the time necessary to generate and distribute control signals.

1. *delay of a single round divided by k*

For small values of *k*, it is usually possible to divide the combinational portion of a single round into *k* stages with equal (or at least approximately equal) delays. The delay of a single stage, equal to the delay a single round divided by *k*, determines the minimum clock period of the circuit, $T_{CLKinner\_round}$ (*k*), as shown in Fig. 6a.

2. *delay of the largest indivisible operation*

For some ciphers, when the number of internal pipeline stages *k* increases, it becomes more and more difficult to divide the combinational portion of a single round into stages with equal delays. At certain point, introducing additional internal registers to the circuit may require dividing an elementary operation of the cipher, such as an S-box or addition, into several stages. This division may be difficult to accomplish if the operation is performed using a standard library cell, special carry propagate circuitry, or if the operation is so simple that it cannot be easily divided into less complex atomic operations. This case is shown in Fig. 6b.

3. *delay of the control unit*

The control unit determines the data flow in the circuit. This unit is responsible for generating enable signals for all registers and memories in the circuit, and address inputs for all memories and major multiplexers. The time necessary to generate and distribute these signals, counted from the rising edge of the clock, may be greater than the time necessary to propagate data between two adjacent registers in the pipeline, as shown in Fig. 6c. This is especially true for control signals with large fanouts distributed globally to every stage of the pipeline.

4. *limit on the maximum latency*

Increasing the number of inner-round pipeline stages, *k*, increases the overall latency of the cipher, by a factor given in (5). If the specification of the cryptographic system imposes a limit on the maximum latency, L_{max}, this limit may determine the maximum possible number of inner-round pipeline stages, k_{max}, as shown in Fig. 7.

$$k_{max} \leq (L_{max} - L_{basic}) \,/\, \#rounds \cdot (t_P + t_{su}) \qquad (6)$$

5. *limit on the maximum input/output bandwidth*

We define the input/output bandwidth as a frequency of an external clock used to control the transmission of data between the integrated circuit and an external environment. The input/output bandwidth necessary to sustain the throughput of the circuit working in the mixed inner and outer-round pipelining is given by

$$\begin{aligned} Bandwidth &= Throughput(K, k)/bus\_width = \\ &= (K/\#rounds)\cdot(block\_size/bus\_width)\cdot f_{CLK\_inner\_round}(k), \end{aligned} \qquad (7)$$

where $f_{CLK\_inner\_round}(k)$ is a frequency of the clock for a *k*-round inner-round pipelining. The circuit is assumed to have two independent ports of the width *bus_width*, used for input and output respectively. In case of using the same bus for both input and output, the bandwidth must be at least twice as high to sustain the same throughput. The maximum bandwidth may limit

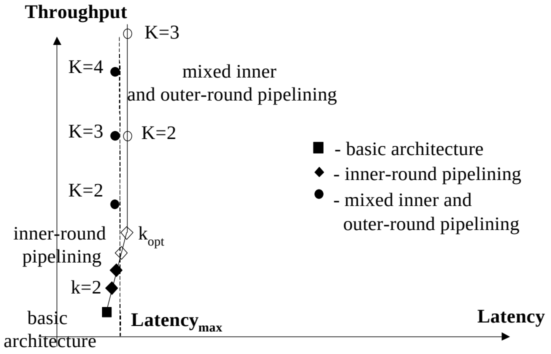


Fig. 7 Possible limits on the throughput due to the latency limit.

the maximum value of the product $K \cdot f_{CLK\_inner\_round}(k)$, and thus the maximum number of inner and outer-round pipeline stages.

4. RESULTS OF PIPELINED IMPLEMENTATIONS OF SELECTED SECRET-KEY BLOCK CIPHERS

4.1 Devices and tools used for implementations

We have implemented five selected ciphers using one of the largest currently available Xilinx Virtex FPGA devices, XCV-1000BG560-6. This device is fabricated using 0.22 µm CMOS process, and contains about one million of equivalent logic gates. All four ciphers were first described in VHDL, and their behavioral model verified using functional simulation based on standard test vectors. The revised VHDL code was then implemented using Xilinx Foundation Series v. 2.1. The only constraint specified during implementation was the minimum clock period. Timing characteristics of the circuit was extracted from the implementation reports generated by Xilinx tools, and confirmed using timing simulation. The results of our implementations of all five ciphers for three architectures shown in Figs. 2a,b,d are summarized in Figs. 9-12.

4.2 Number and location of pipeline registers

In both pipelined architectures the number and location of pipeline registers has been chosen in such a way that the critical path between any two adjacent registers includes only one level of CLBs.

No attempt was made to set the number of pipeline stages to the exact optimum in terms of the throughput to area ratio. This simplification has two major justifications. First, choosing an exact value of k_{opt} may lead to an irregular design, with pipeline registers inserted inside of the cipher elementary operations, and the VHDL code which is hard to develop, test, and maintain. Secondly, in the FPGA implementations, registers available in each Configurable Logic Block (CLB) often remain unused even though the combinational portion of a CLB is fully utilized. Typically, using these registers for additional pipeline stages does not increase the size of the circuit expressed in the number of CLBs. As a result, choosing the number of pipeline stages *k* slightly larger than the optimum often leads to the designs that have the same maximum throughput, and either the same or only marginally larger area.

Apart from introducing registers in the primary data paths, additional pipeline registers had to be introduced in all remaining data paths to properly synchronize the operation of the circuit, as shown in Fig. 8 for Twofish. Several adjacent registers introduced for the purpose of synchronization have been replaced by a FIFO (First-In First-Out) buffer to limit the number of required CLBs.

The number of pipeline stages per round of each cipher is shown in Fig. 8. Typically, one round of inner-round pipelining requires only one more pipeline stage than a single round of the mixed architecture. This pipeline stage is used to surround an input multiplexer (MUX) present in the feedback path of the inner-round pipelined architecture (see Fig. 2b).

For the purpose of a hardware implementation, Serpent is treated as a cipher with four extended rounds, we call implementation rounds [2]. Each implementation round is composed of 8 regular cipher rounds. When treated this way, Serpent can be implemented efficiently using both the basic iterative architecture, and the inner-round pipelining. In the

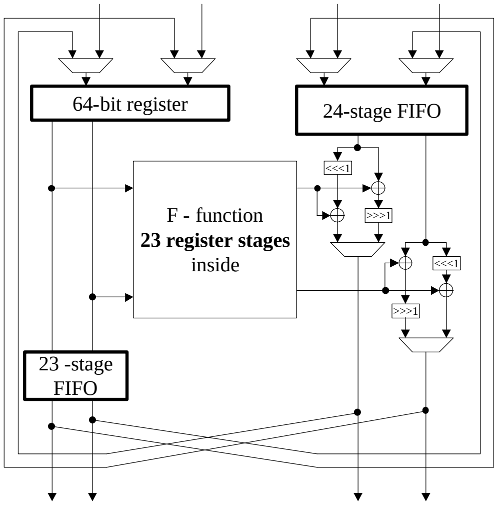


Fig. 8 Pipelining of the Twofish round.

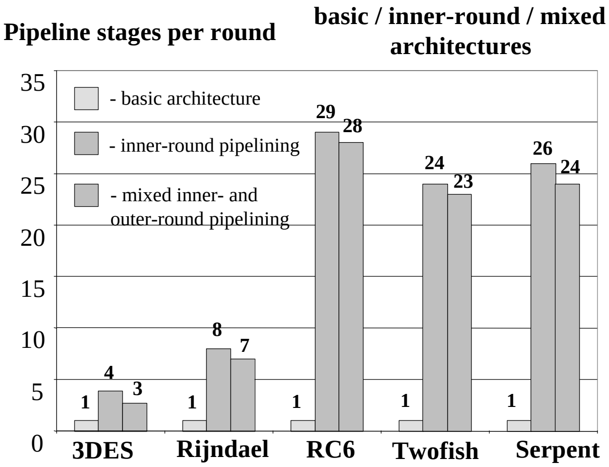


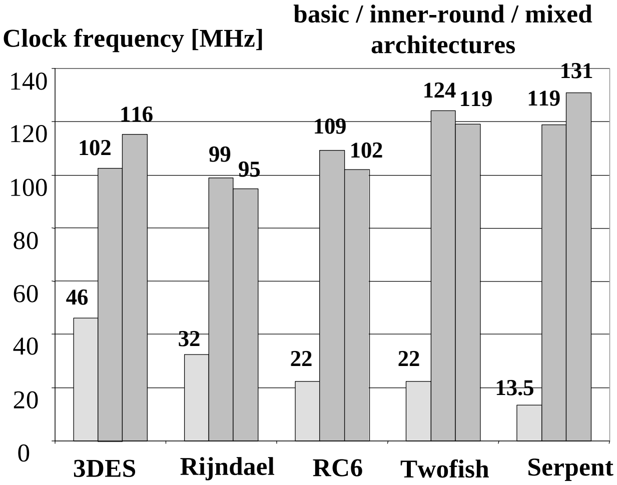
Fig. 9 Number of the pipeline stages per cipher round in each of the three implemented architectures: basic iterative architecture, inner-round pipelining, and full mixed inner- and outer-round pipelining.

inner-round pipelining, Serpent employs three pipeline stages per regular cipher round, and 24 pipeline stages per extended round.

4.3 Minimum clock period

In the basic iterative architecture, there exist large differences among the maximum clock frequencies of five investigated ciphers. These frequencies range from 46 MHz for Triple DES downto 13.5 MHz for Serpent, as shown in Fig. 9. This large spread of frequencies is caused by different delays through a single round of each cipher.

In the inner-round architecture (Fig. 2b), the minimum clock period does not depend any longer on the complexity of the cipher round, but is determined by the delay between two adjacent pipeline registers. In our implementations, this delay is equal to the delay of a single CLB level. This way, the critical paths have almost the same length for all ciphers and differ only in the delays of interconnects. The maximum clock frequencies of all five ciphers are within 12% from their mean value for inner-round pipelining, and within 16% from their mean value for mixed inner- and outer-round pipelining. Even small changes in the VHDL code, tool settings, tool versions, or target FPGA devices may reduce or even reverse these small



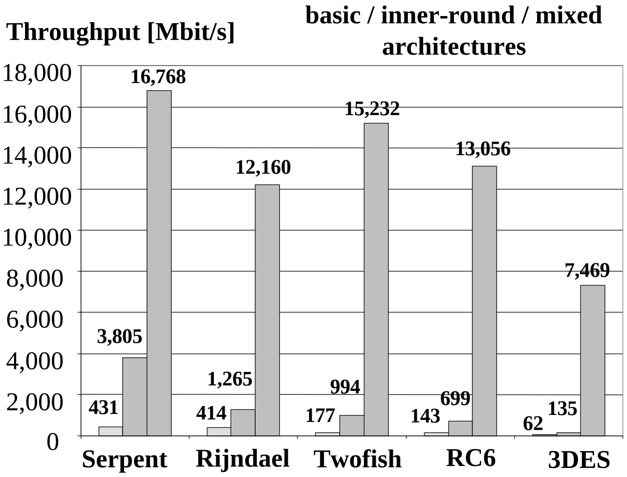


Fig. 10 Maximum clock frequency in each of the three implemented architectures: basic iterative architecture, inner-round pipelining, and full mixed inner- and outer-round pipelining.

Fig. 11 Maximum throughput in each of the three implemented architectures: basic iterative architecture, inner-round pipelining, and full mixed inner- and outer-round pipelining.

differences. For any practical purpose, including a comparison of performance among several ciphers, the maximum clock frequencies in the inner-round pipelining and the mixed inner- and outer-round pipelining are the same for all investigated ciphers.

For each cipher, the maximum clock frequency in the inner-round pipelining is within 15% from the corresponding clock frequency in the mixed pipelining. For three ciphers, inner-round pipelining can operate with the higher clock frequency than mixed pipelining. This effect can be explained by a much smaller area of the implementation with inner-round pipelining (see Fig. 13), which supports more efficient routing. Nevertheless, in our experiments, we also found out, that underutilization of the circuit area may have a negative effect on the clock frequency. For example, Triple DES, was shown to achieve 102 MHz in XCV-1000, where only 2% of the CLB slices are used by the circuit, and 161 MHz in the smaller device of the same family, XCV-150, where 19% of the CLB slices were utilized.

For two ciphers, Triple DES and Serpent, the order of frequencies was reversed, because the critical path in these ciphers includes access to the memory of internal keys. This access is slower in the inner-round pipelining, where all keys are stored in the CLB RAMs, compared to the mixed inner- and outer-round pipelining, where all internal keys are stored in the CLB registers.

4.4 Circuit throughput

The relative gain in the cipher throughput for the architecture with the inner-round pipelining, compared to the basic iterative architecture is illustrated in Fig. 11. This gain is the largest (greater than 5) for ciphers with the longest single round data path, i.e., Serpent and Twofish, and the smallest (only marginally greater than 2) for Triple DES, with the shortest single-round data path.

The formula for a cipher throughput after introducing the optimum number of inner-round pipeline stages, k_{opt} is

$$Throughput\ (k_{opt}) = block\_size\ /\ \#rounds \cdot \mathrm{T}_{\mathrm{CLKinner\_round}}\ (k_{opt}) \qquad (8)$$

Since the minimum clock period is almost identical for all algorithms, the throughput is determined by the ratio *block_size* / *#round*. Thus, this throughput is the largest for Serpent with the block size of 128 bits, and four *implementation* rounds only, and the smallest for Triple DES with the block size of 64 bits and 48 rounds.

For the architecture with the mixed inner- and outer-round pipelining the throughput is given by

$$Throughput\ (k_{opt}) = block\_size\ /\ \mathrm{T}_{\mathrm{CLKmixed}}\ (k_{opt}) \qquad (9)$$

The throughput does not depend any longer on the number of the cipher rounds. Since the maximum clock frequency tends to be similar for all ciphers, the throughputs are also approximately equal for all ciphers with the same block size. This is why all AES candidates have a throughput approximately equal, and twice as high as Triple DES.

4.5 Encryption latency

In Fig. 12, we report the increase in the encryption latency resulting from using the inner-round pipelining and full mixed inner- and outer-round pipelining. The latency increases by a factor of about 2.5-3 for Serpent and Rijndael, ciphers with a relatively simple cipher round, and by factors 4.4 and 6, respectively, for Twofish and RC6, ciphers with more complex cipher rounds. The absolute values of latency in architectures with full mixed inner- and outer-round pipelining are below 1μs for Serpent and Rijndael, below 2 μs for Triple DES, about 3 μs for Twofish, and below 6 μs for RC6. In majority of applications that require hardware-based high-speed encryption, the encryption/decryption throughput is a primary performance measure, and the aforementioned values of latency are fully acceptable.

4.6 Circuit area

The increase in the circuit area for both pipelined architectures is shown in Fig. 13, and is different for every cipher. In the inner-round pipelining, the increase is moderate, ranging from several tens of CLB slices for Triple DES, up to 2300 CLB slices for RC6. The implementations take from 3% to 50% of the total number of CLB slices, and easily fit within a single Virtex FPGA device XCV-1000.

For mixed pipelining, the circuit area increases proportionally to the number of cipher rounds. Triple DES has the smallest area requirements. Serpent and Twofish require almost twice as much area, and RC6 almost four times as large area as Triple DES. Comparing the area of Rijndael is made difficult by the use of dedicated memory blocks, Embedded Array Blocks (EABs), to implement large S-boxes. EABs are not used in implementations of any of the remaining AES candidates, and we are not aware of any formula for expressing the area of EABs in terms of the area used by CLB slices.

Triple DES is the only cipher that can be implemented using a single Virtex FPGA device XCV-1000. Serpent and Twofish can be implemented using two FPGA devices XCV-1000; Rijndael requires three, and RC6 requires four such devices.

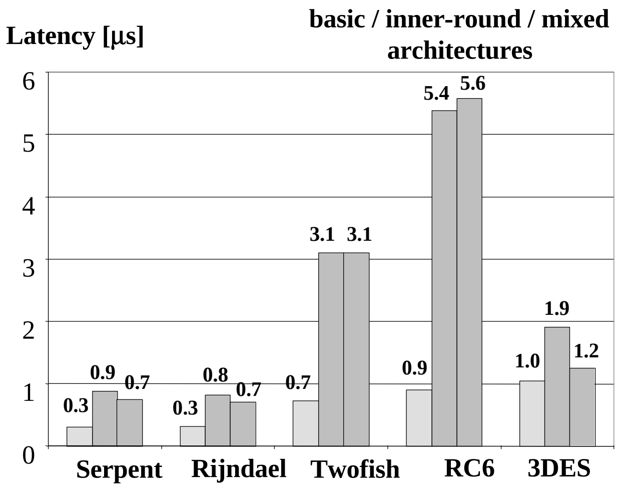


Fig. 12 Encryption latency in each of the three implemented architectures: basic iterative architecture, inner-round pipelining, and full mixed inner- and outer-round pipelining.

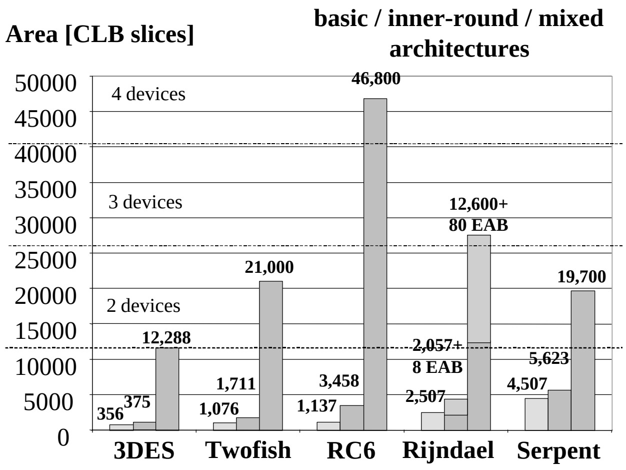


Fig. 13 Area in each of the three implemented architectures: basic iterative architecture, inner-round pipelining, and full mixed inner- and outer-round pipelining.

5. COMPARISON WITH RESULTS OF OTHER GROUPS

An architecture identical to our full mixed inner- and outer-round pipelining was used in [16] to implement DES. The throughput reported in this paper was 10.1 Gbit/s for Xilinx Virtex XCV300-6, compared to 7.5 Gbit/s obtained by our group for Triple DES implemented in XCV-1000-6. This difference can be attributed to the three times smaller area requirements of DES, supporting more efficient routing, and to the more focused and careful optimizations described in [16].

An architecture similar to our mixed inner- and outer-round pipelining was reported in [3]. The primary difference was that the number of the inner-round pipeline stages *k* was chosen to be small, ranging from 1 to 3, leading to designs sub-optimum from the point of view of both throughput and throughput to area ratio [15]. Speed-ups resulting from applying pipelining were from 3.4 to 5.7 times smaller than the speed-ups demonstrated in this paper and shown in Fig. 11 [15].

An advantage of our methodology over traditional methodology can be best demonstrated by comparing our results with results of the NSA team reported in [10]. The suite of ciphers implemented by the NSA team and our group was very similar. Four ciphers were identical in both suites; the implementation of Mars was reported only by the NSA team, and the implementation of Triple DES was reported only by our team.

In Fig. 14 we compare the speed-ups obtained by using full pipelining, reported by both groups. The speed-up is defined as a ratio of throughputs in the fully unrolled pipelined architecture vs. basic iterative architecture. These results clearly confirm the dependencies shown in Fig. 3, and described in section 3.1. The speed-ups obtained using our methodology are over three times larger than the speed-ups obtained using the traditional methodology for Rijndael, Twofish, and RC6. The comparable speed-up for Serpent is a result of different definitions of the basic iterative architecture used by both groups. Still, the absolute throughput of Serpent reported in this paper, 16.8 Gbit/s, is over twice as large as the throughput reported in [10], 8 Gbit/s.

The second difference between both methodologies is the effect of pipelining on the encryption/decryption latency. In the traditional methodology, pipelining does not increase, or even slightly reduces latency, as shown in Fig. 15. In our methodology the substantial increase of latency can be observed. This increase is particularly large for ciphers with the relatively complex cipher round. Nevertheless, the time necessary to encrypt a long stream of data is primarily a function of the encryption throughput, and is almost independent of the encryption latency. Since a primary function of the high-speed hardware cryptographic devices is to encrypt long streams of data, the increase in latency might not be critical in practical applications. This matter should be further investigated taking into account the properties of today's high-speed networks.

6. ADVANTAGES OF USING MIXED INNER AND OUTER-ROUND PIPELINING FOR COMPARING PERFORMANCE OF THE AES CANDIDATES

In papers presented at the Third AES Conference [11], several research groups presented their methodology for a fair comparison of the hardware performance of five AES finalists [2-4, 10, 12, 13]. For ciphers operating in the feedback cipher

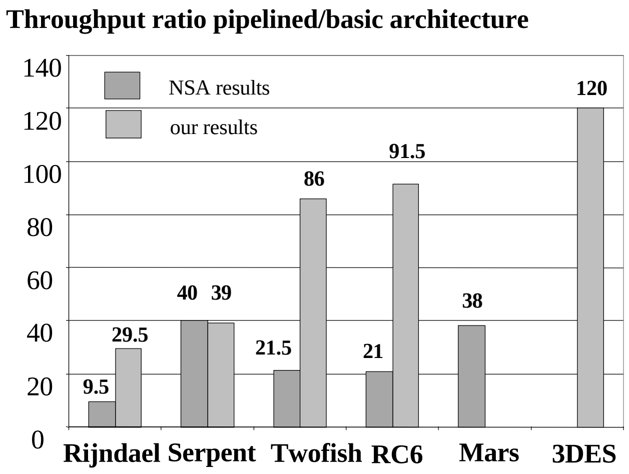


Fig. 14 Ratio of the encryption throughputs for the full pipelined architecture and the basic iterative architectures: NSA results for outer-round pipelining, and our results for the mixed inner- and outer-round pipelining.

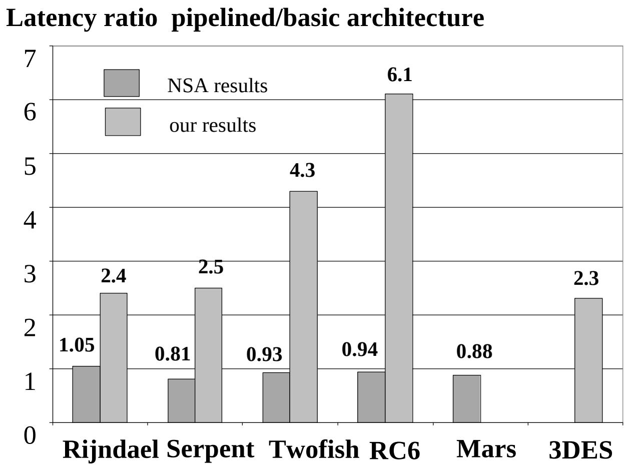


Fig. 15 Ratio of the encryption latencies for the full pipelined architecture and the basic iterative architectures: NSA results for outer-round pipelining, and our results for the mixed inner- and outer-round pipelining.

modes, such as CBC, CFB, OFB, a good agreement in both methodology and the results of comparison was achieved. For ciphers operating in the non-feedback cipher modes, such as ECB and counter mode, methodologies used by various groups were substantially different, leading to large differences in results and their interpretation [15].

In our opinion, a fair methodology for comparing hardware performance of secret-key ciphers should fulfill the following requirements.

a) It should be based on an architecture which is likely to be used in practical implementations, because of the superior throughput/area ratio.
b) It should not favor any group of ciphers or a specific internal structure of a cipher.

For feedback cipher modes, both conditions are very well fulfilled by the basic iterative architecture, and this architecture was commonly used for comparison [2, 3, 10, 13].

For non-feedback cipher modes, the decisions about the choice of the architecture varied and no consensus was achieved. The NSA team suggested the use of the full outer-round pipelining for comparison [10]. In our opinion, this choice does not fulfill either one of the formulated above requirements. As shown in Fig. 3, the outer-round pipelining offers significantly worse throughput to area ratio compared to the architecture with the mixed inner and outer-round pipelining. Therefore, the use of this architecture may lead to the sub-optimum designs, which are not likely to be used in practice. Secondly, the choice of the outer-round pipelining favors ciphers with a short and simple cipher round. Other ciphers, such as Mars and RC6, with the more complex internal round are adversely affected. This unequal treatment of AES candidates can be explained by investigating formulas that describe the encryption throughput in full outer-round pipelining (10) and full mixed inner- and outer-round pipelining (11). The encryption throughput in the full outer round pipelining is given by

$$Throughput_{\text{full\_outer\_round}} = block\_size \,/ \mathrm{T}_{\text{CLKbasic}} \qquad (10)$$

where $\mathrm{T}_{\text{CLKbasic}}$ is a delay of a single round, including register delay and setup time.

On the other hand, the throughput in the full mixed inner and outer-round pipelining is given by

$$Throughput_{\text{full\_mixed}} = block\_size \,/ \mathrm{T}_{\text{CLKmixed}}(k_{\text{opt}}) \qquad (11)$$

where $\mathrm{T}_{\text{CLKmixed}}(k_{\text{opt}})$ is the delay of a single pipeline stage for the optimum number of registers introduced inside of a single round. In FPGA implementations, this delay is determined by the delay of a single CLB slice and delays of interconnects between CLBs.

The conclusions based on these two different architectures and formulas (10) and (11) are completely different. When the full outer-round pipelining is used for comparison, there exist strong differences in cipher throughput. Ciphers with a simple and short basic round, such as Serpent and Rijndael achieve much greater throughput compared to the remaining candidates. When the full inner and outer-round pipelining is used for comparison, all candidates achieve almost exactly the same throughput. This is because the internal clock period $\mathrm{T}_{\text{CLKmixed}}(k_{\text{opt}})$ is almost identical for all ciphers implemented using this architecture, as shown in Fig. 10. Based on this observation, our conclusions are as follows

a) full mixed inner and outer-round pipelining should be the architecture of choice for comparing hardware performance of the AES candidates and other secret-key block ciphers in non-feedback cipher modes;

b) in the full mixed inner and outer-round pipelining, all ciphers can achieve the same throughput, so the primary criteria of comparison should be the area used to implement each cipher.

6. SUMMARY

The new methodology for the design of high-speed implementations of secret-key block ciphers has been proposed. It is based on introducing an optimum number of pipeline stages inside of a cipher round, before applying loop unrolling. This new methodology guarantees a substantial increase in the cipher throughput with a relatively insignificant penalty in the circuit area and latency. Five well-known modern block ciphers, with various characteristics, Triple DES, Rijndael, RC6, Serpent, and Twofish, have been implemented according to our methodology using Xilinx Virtex FPGA devices. It was shown that all these ciphers can achieve the maximum clock frequency in the range from 100 to 130 MHz. By applying our methodology to the current generation of the Xilinx Virtex FPGA devices, Rijndael, RC6, Twofish and Serpent, achieve the maximum throughput in the range from 12.1 to 16.8 Gbit/s, and Triple DES achieves the throughput of 7.5 Gbit/s. To our best knowledge, our implementations of four AES candidates are the fastest ever reported in any technology. The proposed methodology is very well suited for practical high-speed implementations of modern secret-key block ciphers in both FPGAs and custom ASICs. It also provides a fair way of comparing various secret-key block ciphers, including AES candidates, from the point of view of their hardware performance in non-feedback cipher modes.

7. ACKNOWLEDGMENTS

The authors would like to thank Tanvir Joy for his contribution to the design of Triple DES.

8. REFERENCES

[1] FIPS 46-2, "Data Encryption Standard," revised version to be issued as FIPS 46-3, National Institute of Standards and Technology, 1999.

[2] K. Gaj and P. Chodowiec, "Comparison of the hardware performance of the AES candidates using reconfigurable hardware," Proc. 3rd Advanced Encryption Standard (AES) Candidate Conference, New York, April 13-14, 2000.

[3] A. J. Elbirt, W. Yip, B. Chetwynd, C. Paar, "An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists," Proc. 3rd Advanced Encryption Standard (AES) Candidate Conference, New York, April 13-14, 2000.

[4] N. Weaver, J. Wawrzynek, "A comparison of the AES candidates amenability to FPGA Implementation," Proc. 3rd Advanced Encryption Standard (AES) Candidate Conference, New York, April 13-14, 2000.

[5] ANSI X9.52-1998, *Triple Data Encryption Algorithm Modes of Operation.* American National Standard Institute, 1998.

[6] FIPS 185, "Escrowed Encryption Standard (EES)," National Institute of Standards and Technology, 1994.

[7] "Advanced Encryption Standard Development Effort," http://www.nist.gov/aes.

[8] D. C. Wilcox, L. G. Pierson, P. J. Robertson, E. L. Witzke, and K. Gass, "A DES ASIC suitable for network encryption at 10 Gbps and beyond," Proc. 1st Int. Workshop on Cryptographic Hardware and Embedded Systems, CHES'99.

[9] A.J. Elbirt and C. Paar, "An FPGA Implementation and Performance Evaluation of the Serpent Block Cipher," Eighth ACM International Symposium on Field-Programmable Gate Arrays, Monterey, California, February 10-11, 2000. Preprint available at http://ece.wpi.edu/Research/crypt/publications/index.html.

[10] B. Weeks, M. Bean, T. Rozylowicz, C. Ficke, "Hardware performance simulations of Round 2 Advanced Encryption Standard algorithms," NSA's final report on hardware evaluations published May 15, 2000, available at http://csrc.nist.gov/encryption/aes/round2/r2anlsys.htm#NSA

[11] "Third AES Candidate Conference," http://csrc.nist.gov/encryption/aes/round2/conf3/aes3conf.htm.

[12] T. Ichikawa, T. Kasuya, M. Matsui, "Hardware Evaluation of the AES Finalists," Proc. 3rd Advanced Encryption Standard (AES) Candidate Conference, New York, April 13-14, 2000.

[13] A. Dandalis, V. K. Prasanna, J. D. Rolim, "A Comparative Study of Performance of AES Final Candidates Using FPGAs," *Proc. Cryptographic Hardware and Embedded Systems Workshop*, CHES 2000, Worcester, MA, Aug 17-18, 2000.

[14] C. Patterson, "A Dynamic FPGA Implementation of the Serpent Block Cipher," Proc. 2nd Int. Workshop on Cryptographic Hardware and Embedded Systems, CHES'00.

[15] K. Gaj and P. Chodowiec, "Hardware performance of the AES finalists - survey and analysis of results," available at http://ece.gmu.edu/crypto/AES_survey.pdf

[16] S. Trimberger, R. Pang, A. Singh, "A 12 Gbps DES Encryptor/Decryptor core in an FPGA," *Proc. Cryptographic Hardware and Embedded Systems Workshop*, CHES 2000, Worcester, MA, Aug 17-18, 2000.