Some Guidelines for Implementing Symmetric-Key Cryptosystems on Reconfigurable-Hardware

Arturo Díaz-Pérez, Nazar A. Saqib, and Francisco Rodríguez-Henriquez

Computer Science Section, Electrical Engineering Department
Centro de Investigación y de Estudios Avanzados del IPN
Av. Instituto Politécnico Nacional No. 2508, México D.F.
{nabbas@computacion.cs.cinvestav.mx, adiaz,francisco@cs.cinvestav.mx}

Abstract. This paper identifies the basic characteristics of cryptographic algorithms especially symmetric block ciphers for their implementation on hardware platforms. The basic primitives in symmetric ciphers are discusses and some implementation techniques are suggested for them. As an application, an FPGA implementation of DES is presented which achieves a throughput of 274 Mbits/s occupy just 165 CLB slices for a single round. The same guidelines well hold for other block ciphers like Advanced Encryption Standard (AES).

1 Introduction

Most of the cryptographic algorithms, especially symmetric block ciphers, are based in the principle of substitution and transposition to encrypt a plain-text message and to produce a cipher-message. Those transformations are based on well-understood mathematical problems using non-linear functions and linear modular algebra [1].

Implementation of criptographic algorithms mainly use bit-level operations and table look-ups. Bit-wise operators (XORs, AND/OR, etc.), substitutions, logical shifts and permutations are quite common operations. Such operations are well suited for their fast execution in hardware platforms. Furthermore, currently abundant memory resources in hardware platforms enhance encryption speed for the operations like substitution.

Since, in general-purpose processors instructions are executed in a sequential way, highly parallel architectures can be designed on hardware to achieve higher-performance compared to software implementations.

In this paper, we explore about general strategies for implementing symmetric-block ciphers in reconfigurable hardware. We search for the frequent operations involved in cryptographic algorithms to their hardware implementations. It is also explained how the bit level parallelism is exploited for the cryptographic algorithms using either iterative or pipeline approaches. We present a case of study for Data Encryption Standard (DES), which can be extended to the other similar cryptosystems. The reconfigurable type of hardware (Field Programmable Gate Arrays) platform is chosen for implementing one round of DES. Achieved throughput is 274 Mbits/s occupying just 166 CLB slices for XCV400E-8-BG560.

The rest of this paper is organized as follows: Section 2 describes the general characteristics in cryptographic algorithms and some wise techniques for their implementations. In Section 3 a short introduction to DES algorithm and its FPGA

implementation is presented. The implementation results and performance comparison with the existing state of the art DES implementations is presented in Sectionr 4. Section 5 includes concluding remarks and the future work.

2 General Characteristics in Cryptographic Algorithms

In this section some of the basic characteristics have been investigated especially for symmetric block ciphers. Some fundamental operations and the general structure will be discussed. The suitability of those operations on hardware platform is also a part of this section.

2.1 General Structure of a symmetric-key cipher

Symmetric ciphers are normally based on well-understood mathematical problems. A common method for the construction of a block cipher is to combine multiple rounds of repeated bit shuffling (Permutation), simple non-linear functions, implemented as lookup tables (S-Box) and linear mixing (in the sense of modular algebra) using XOR as it is shown in Figure 1.

Modern symmetric ciphers use those operations to achieve enough security strength against known attacks. The most important characteristics of a block cipher are: 1) its iterative nature where basic transformations are repeated several times (rounds), 2) increased block length ($\geq$128-bit), 2) complex permutation and substitution operations (adding more non-linear functions), and 4) Key schedule to obtain a sequence of keys from a single user-key.

DES is an example of a block cipher, which operates on a 64-bit block length and combine 16 rounds of repeated operations: permutations and substitutions. For current security applications, 64-bit block length is not sufficient against the brute force attacks. The new advanced encryption standard (AES) can support a data and key length of variable sizes of 128, 192 and 256 bits. AES repeats 11 rounds of the similar operations for the encryption of one block of data.

2.2 Software and hardware implementations

Cryptographic algorithms can be implemented in software, VLSI (Very Large Scale Integrated circuit), and reconfigurable logic platforms. The choice of the platform depends on algorithm performance, cost and flexibility. The performance requirements for the cryptosystems are directly concerned to the question what application needs to be secured. Low speed or moderately high speed network where very small amount of traffic is to be processed unpredictable, can be implemented in software. Secure mail, credit card transactions, e-commerce, are good examples of such applications. Software methods offer low cost and highly flexibly solutions where modification or upgrade is possible any time.

On the other hand, secure high-speed networks where a large amount of traffic is to be processed in unpredictable and in real time requiere high- peformance hardware implementations. Hardware implementations are required in telephone conversation,

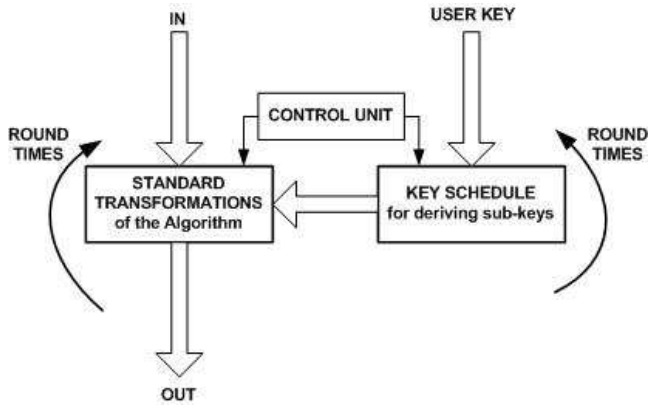

Fig. 1. General structure for a block cipher

video conferencing, streaming audio or encoded transmissions where data is coming at significant line speeds and must be treated in real time.

Hardware solutions can be specified on VLSI or reconfigurable logic devices. VLSI architectures for cryptographic algorithms offer fastest encryption rates if speed factor dominates all others including cost and size of the application. VLSI solutions are costly because of long development cycle for achieving final product. The cost per unit however decreases in bulk production. The main drawback of VLSI solutions is their lack of flexibility when any small change or modification in the circuit require new layout for the circuit.

Reconfigurable hardware solutions eliminate those VLSI drawbacks at the cost of small decrease in encryption rate for the cryptographic applications. Reconfigurable devices are highly flexible offering low cost solutions due to low development cycle for achieving final product. Run-time configuration in reconfigurable devices allows the use of more than one cryptographic algorithm.

2.3 Fundamental Operations of Cryptographic Algorithms

Symmetric or secret key cryptographic algorithms are based on well-understood mathematical and cryptographic principles. The most common primitives encountered in various cryptographic algorithms are permutation, substitution, rotation, bit-wise XOR, shift etc. This is one of the reasons for their fast encryption speed. On the other hand, asymmetric or public key cryptographic algorithms are based on the mathematical problems difficult to solve. The most common primitives in various such type of algorithms include modular addition/subtraction, modular multiplication, variable length rotations etc. Those primitives add to algorithmic strength but hard to implement: occupy more space and consume more time. Therefore those algorithms are not used to encrypt large data files and are applied to the other important cryptographic applications like key exchange, signature, verification etc.

A detail survey has been conducted by [2] to identify the basic operations involved in several cryptographic algorithms. The survey has been slightly updated as

Modular addition or	Blowfish, CAST, FEAL,GOST, IDEA, LOKI97,RC5, RC6, TEA, subtraction, SAFER K-64, Twofish, RC4, SEAL TWOPRIME, WAKE
Bitwise XOR	Blowfish, CAST, DEAL, DES, FEAL, GOST,IDEA, A5 RC4, RC5, Nanoteq, SEAL, TWOPRIME, WAKE, LOKI97 LOKI91, Madryga, MISTY, Rijndael, MMB, RC6, SAFER K-64, TEA, Twofish
Bitwise AND/OR	MISTY
Variable-length rotations	CAST, Madryga, RC5, RC6
Fixed-length rotations	DEAL, DES, CAST, FEAL, GOST, Serpent, RC6, Twofish
Modular multiplication	CAST, IDEA, RC6, MMB, Rijndael, Elliptic Curves
Substitution	Blowfish, DEAL, DES, LOKI91, LOKI97, Twofish, Rijndael
Permutation	DEAL, DES, ICE, LOKI91, LOKI97
Non-circular shifts	Serpent, TEA

Table 1. Primitives of cryptographic algorithms

shown in Table 1. From Table 1, it is clear that most of the cryptographic algorithms mainly include bit-wise operations like XOR, AND/OR etc. Long word length for the cryptographic algorithms is another characteristic, which is recommended by various international standards to attain sufficient security against brute force attacks. The recommended word length for the modern block ciphers is no less than 128-bits [3]. The new Advance Encryption Standard (Rijndael) can support a word length of 128, 192 and 256 bits. The high key/world length of cryptographic algorithms restrict parallel flow of the data on 8, 16, 32-bit machines resulting high time delays for the execution of the algorithms. To confuse the relationship between input and output, cryptographic algorithms perform a number of iterations on the same input data block for encryption. DES performs 16 iterations and AES support 10, 12, and 14 iterations depending on the word length.

2.4 Useful Properties for Implementing Symmetric Cryptosystems in FPGAs

Hardware implementations are intrinsically more physically secure: Key access and algorithm modification is considerably harder and some properties of summetric-key cryptographic algorithms well match for their implementation on reconfigurable devices like FPGAs.

1. Most of the block ciphers (DES and AES for example) include bit-level operations: XOR, AND, OR, etc. The abundance of bit-level operations in cryptographic algorithms makes their execution faster on FPGAs and additionally they occupy relatively less hardware resources. Permutation and logic shift are the two common operations among symmetric block ciphers. Both operations are free of cost as they do not occupy FPGA resources and can be implemented by rewiring the input bus connections, thus consuming no time for their executions.

2. Substitution is an important operation in symmetric block ciphers to add maximum non-linearity. In fact the strength of DES algorithm is based on its substitution boxes (SP-Boxes). The S-Box in AES is used for both encryption and key schedule algorithms. FPGAs offer two solutions for the implementation of substitution operation: CLBs and BlockRAMs. A CLB can be configured into memory mode providing the implementation of substation operation as a look-up table. Block RAMs are built-in fast memories included in modern models of FPGAs. The Virtex series devices contain more than 280 BRAMs of 4096 bits each. Block RAMs are faster than CLBs since routing overhead is required among a great number of CLBs.
3. Given the iterative nature of symmetric cryptosystems, various design strategies offer area-time tradeoff for the choice of implementing a number of rounds. In iterative looping only one round of the algorithm is implemented and n cycles are consumed for executing n iterations of the algorithm. The design achieves low throughput but occupies less FPGA resources. A pipeline design implements all the rounds by providing registers between any two consecutive rounds. At each clock cycle the results of a round is transferred to the next round. After n cycles the final output appears at the last round and then it appears at each successive clock cycle. The encryption rate in pipeline architecture increases at the ratio of $n : 1$ as compared to iterative looping at the increase of $1 : n$ ratio in FPGA resources. The encryption rate can be still improved in pipeline architecture by putting more registers inside the different steps of the same round which it is called sub-pipelining. FPGAs allow to implement only one round and then the same building block is copied n times.
4. Modern block ciphers operate on data blocks of 128 bits or more. Unlike software implementations on general-purpose microprocessors, FPGAs permit parallel execution of the whole data block. FPGAs can contain more than 1000 external pins to be programmable for inputs or outputs.
5. Symmetric block ciphers use a secret key and key schedule algorithm for the generation of sub-keys for each round. It is recommendable to change the secret key for different sessions. FPGAs also provide the benefit to store secret keys for different sessions.

3 FPGA implementation of DES

This section provides a short introduction to DES and explains the algorithm flow for encrypting one data block. An FPGA implementation is then presented.

3.1 Introduction to DES Algorithm

On August, 1974, IBM submitted a candidate (under the name LUCIFER) for cryptographic algorithm in response to the 2nd call from National Bureau of Standards (NBS), now the National Institute of Standards & Technology (NIST)[3], to protect data during transmission and storage. NBS launched an evaluation process with the help of National Security Agency (NSA) and finally adopted a modification of LUCIFER algorithm as the new Data Encryption Standard (DES) on July 15, 1977. The

Data Encryption Standard [4], known as Data Encryption Algorithm (DEA) by the ANSI [5] and the DEA-1 by the ISO [6] remained a worldwide standard for a long time and was replaced by the new Advanced Encryption Standard (AES) on October 2000. However, DES will remain in the public domain for number of years. It provides a basis for comparison for new algorithms and is also used in IPSec protocols, ATM cell encryption, the secure socket layer (SSL) protocol and in TripleDES. The detail description of DES algorithm can be seen at [7, 1, 8].

DES is a block cipher: It encrypts/decrypts data in 64-bit blocks using a 64-bit key (although effective key length is 56-bit). DES is a symmetric algorithm: The same algorithm and key are used for both encryption and decryption. DES is an iterative cipher: the basic building block (a substitution followed by a permutation) called a *round* is repeated 16 times. For each DES round, a sub-key is derived from the original key called *key schedule*. Key schedule for encryption and decryption is the same except for the minor difference in the order (reverse) of the sub-keys for decryption. A basic algorithm flow for encrypting/decrypting one block of data is shown in Fig. 2.

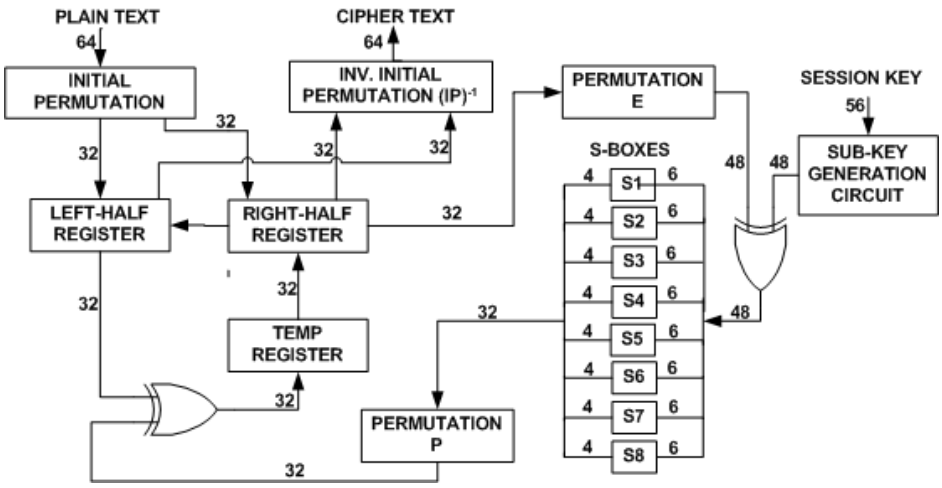

Fig. 2. DES Algorithm

Encryption begins with an *initial permutation* (IP), which scrambles the 64-bit plain-text in a fixed pattern. The result of the initial permutation is sent to two 32-bit registers, called the *right half register* and *left half register*. Those registers hold the two halves of the intermediate results through succeeding 16 iterations. The contents of the right half register are permuted (*permutation E*) and sent to an exclusive-OR unit along with the sub-key for each iteration. Note that some bits are selected twice, allowing the 32-bit register to expand to 48 bits. The 48-bit output of the exclusive-OR block is divided into eight groups (6-bits each) to address eight substitution memories (S-boxes). A *permutation P* is applied to 32-bit output from S-boxes and then feed into an exclusive-OR block along with the contents of the left half register. The output of this block is written into temporary register, concluding the first iteration.

At the next clock cycle, the contents of the temporary registers are written into the right half register and previous contents of the right half register are written into left half register. This process repeats through 16 iterations. After the 16 iterations, the right half and left half register contents are subjected to a final permutation IP^{-1}, which is the inverse of the initial permutation. The output of IP^{-1} is the 64-bit cipher-text.

3.2 FPGA Implementation of DES Algorithm

Figure 3 represents a block diagram for DES implementation on FPGA. As it has been earlier discussed that permutation is a simple operation on hardware devices. It can be implemented by changing bit positions for the outgoing bus (change of wires), hence it is free of cost. DES includes lot of permutation operations (initial, final, permutation E, permutation P). The building blocks for those operations in Figure 3 are therefore symbolic representations having no logic inside. Each S-box in DES occupy 64×4=256-bit memory, a total of 2048 bits for eight S-Boxes. The implementation for s-boxes consumes only 32 CLB slices configured in memory mode. Temporary registers and XOR blocks are the other operations which occupy FPGA resources.

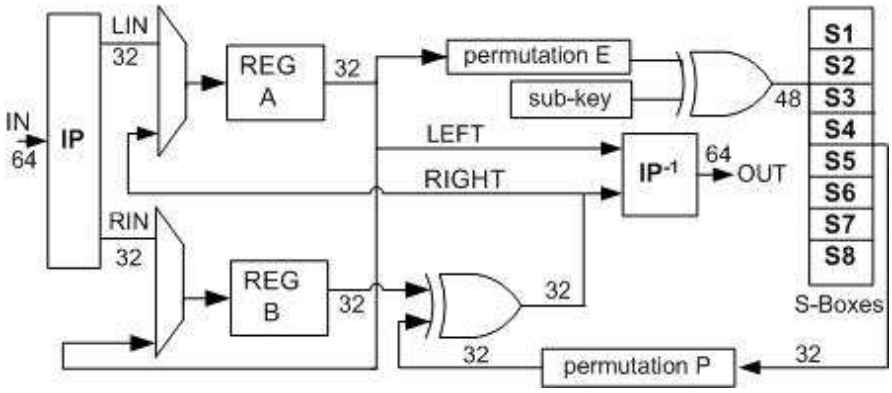

Fig. 3. DES implementation on FPGA

Three inputs: Chip Enable (CE), Clock (CLK), input data (IN) and the only output (OUT) are the four pins of DES chip. Chip enable (CE) activates the timing logic as well as the rest of the circuitry, in its low state (when it is '0'). The external clock CLK is the master clock for the whole circuit that is used to generate all the control signals to synchronize the data flow. When CE enables the circuit, The 64-bit at the input are permuted and divided into two halves RIN and LIN. At the first rising edge of the clock both halves are being transferred to the output of the two registers REGA and REGB. The right halves (REGA output) go through a number of operations: Permutation E; addition with sub-key; substitution (through S-Boxes); Permutation P and; addition with the original left half (REGB output). Before the

next clock comes, the old right half (RIGHT) is the input of the register REGB and the new left half (LEFT) is the input of the register REGA. The sixteen iterations are then executed. After 16th clock cycles the two halves RIGHT and LEFT are concatenated and the resulting block goes through the inverse permutation (IP^{-1}) resulting one encryption for a 64-bit input block. Notice that the usage of an eight DES S-Boxes parallel structure, results in a significant reduction of the critical path for encryption/decryption.

4 Performance Results and Comparison

FPGA implementation of DES targets VirtexE device XCV300fg456 by using Xilinx Foundation Tool F4.1i. It is coded in VHDL and occupies 165 (3%) CLB slices, 117 (1%) slice Flip Flops and 129 (41%) I/Os. The system runs at a frequency of 68.05 MHz (14.7ηS) and achieves a throughput of (68.05 64)/16=274 Mbits/s by taking 16 clock cycles to encrypt one data block (64-bits). Table 2 compares our results with state of the art implementations reported in the last years.

Author	Device used	CLB slices	Allowed Freq. (MHz)	Throughput (Mbits/s)
Wong et al. [9]	XC4020E	438	10	26.7
Biham(software) [10]	Alpha 8400	—	300	127
Kaps and Paar [11]	XC4028EX	741	25.18	402.7
Free-DES [12]	XCV400	5263	47.7	3052
McLoony, McCanny [13]	XCV1000	6446	59.5	3808
Sandia Laboratories [14]	ASIC	—	—	9280
Patterson (Jbits) [15]	XCV150	1584	168	10752
This work (FPGA)	XCV400	165	68.5	274

Table 2. Recent DES hardware/software implementations

A VLSI implementation of DES on static 0.6 micron CMOS technology at [14] is the fastest implementation of DES reported in the literature. Using pipeline approach, the encryption can be performed at the rate of $\geq$6.7 Gbs. A software implementation at [10] on Alpha 8400 achieves a throughput of 127 Mbits/s. Several FPGA implementations of DES have been reported in the literature achieving throughput ranges from 26 to 10752 Mbits/s using different design strategies. A DES implementation at [12] is a free DES cores which uses pipeline approach in ECB mode and achieves a data rate of 3052 Mbits/s. A java-based (Jbits) DES implementation at [15] achieves the fastest encryption rate of 10752 Mbits/s. DES implementation at [11] implement both 2-stage and 4-stage pipeline approaches obtaining throughput of 183.8 Mbits/s and 402.7 Mbits/s respectively. Almost all FPGA architectures for DES implement fully or partially pipeline approaches. Only the design at [9] implement all DES primitives and took 24 cycles to complete encryption for one data block. Our architecture can be compared with that design and shows an improvement of 173% in throughput occupying just 24 (5%) more CLB slices.

5 Conclusions

In this paper we presented some general strategies to implement cryptographic algorithms in reconfigurable hardware. It has been shown that most of the block ciphers share common operations for their standard transformations. Those are mainly bitwise or memory based operations. Logic shifts and permutations are also frequently found. Basic approaches for the implementation of one algorithm serve for others.

A one-round FPGA implementation of DES targets Xilinx VirtexE device XCV400e-8-bg560 occupying 165 CLB slices and achieves a throughput of 274 Mbits/s. The speed factor improves 10 times a similar one-round DES implementation at [9]. This one-round DES implementation will be further extended to DES pipeline architecture and also for 3DES implementations.

References

1. Menezes, A., Oorschot, P.V., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Boca Raton, FL (1997)
2. Charlwood, S., James-Roxby, P.: Evaluation of the XC6200-Series Architecure for Cryptographic Application. In: FPL 98, Lecture Notes in Computer Science 1482, Springer-Verlag Berlin Heidelberg 2003 (1998) 218–227
3. NIST: Announcing the advanced encryption standard(aes). Federal Information Standards Publication (2001)
4. X9.62, A. Federal Information Processing Standard (FIPS) 46, National Bureau Standards (1977)
5. (Revised):, A.X. National Standards for financial institution key management (wholesale), American Bankers Association (1986)
6. 8732:, I.D. Banking-key management (wholesale), Association for Payment Clearing Services (1987)
7. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, New York (1996)
8. Trappe, W., Washington, L.: Introduction to Cryptography with Coding Theory. Prentice Hall, Inc., Upper Saddle River, NJ 07458 (2002)
9. Wong, K., Wark, M., Dawson, E.: A single-chip fpga implementation of the data encryption standard (des) algorithm. In: IEEE Globecom Communication Conf., Sydney, Australia (1998) 827–832
10. Biham, E.: A fast new des implementation in software. In: 4th Int. Workshop on Fast Software Encryption, FSE97, Haifa, Israel, Springer-Verlag, 1997 (1997) 260–271
11. Kaps, J., Paar, C.: Fast des implementations for fpgas and its application to a universal key-search machine. In: Proc. 5th Annual Workshop on selected areas in cryptography-Sac' 98, Ontario, Canada, Springer-Verlag, 1998 (1998) 234–247
12. Core(2000), F.D.: (2000) URL: http://www.free-ip.com/DES/.
13. McLoone, M., McCanny, J.: High-performance fpga implementation of des using a novel method for implementing the key schedule. IEE Proc.: Circuits, Devices & Systems **150** (2003) 373–378
14. Wilcox, D., Pierson, L., Robertson, P., Witzke, E.L., Gass, K.: A des asic suitable for network encryption at 10 gbs and beyond. In: CHESS 99, LNCS 1717 (1999) 37–48
15. Patterson, C.: High performance des encryption in virtex fpgas using jbits. In: Field-programmable custom computing machines, FCCM' 00, Napa Valley, CA, USA, IEEE Comput. Soc., CA, USA, 2000 (2000) 113–121