Modelling DES Soft-Cores for information protection

BSc., MSc. Viktor Melnyk

Lviv National Polytechnic University

Lviv, Ukraine

Prof. Dr. Jürgen Bäsig

Georg-Simon-Ohm-Fachhochschule Nürnberg

Fachbereich
Elektrotechnik Feinwerktechnik Informationstechnik

Abstract

This paper discusses the possibilities of IP-technologies for information protection. An analysis of modern information protection Soft-Cores which support symmetric block ciphers is given. Development perspectives of the Soft-Cores are described. The DES Algorithm and the variations of the architecture of the DES Soft-Core models are considered, their analysis provided and they are compared with some existing DES Soft-Cores on the world market. Two architectures - iterative and pipelined - are identified. Twenty six variations of the DES Soft-Core are proposed and a comparative analysis of their architecture is given. The proposed DES Soft-Cores support all of the discussed functional possibilities. Thus a high performance is achieved, the equipment volume for their realization is lowered and they can be used effectively in a wide spectrum of the application fields.

ISSN 1616-0762 Sonderdruck Schriftenreihe der Georg-Simon-Ohm-Fachhochschule Nürnberg Nr. 9, Nürnberg 2002

Introduction

Modern computer systems allow the user to protect the data that are processed. Some ways of protection are user identification, distribution of the access to the information, provision of the integrity, confidentiality of the information, its protection from modification and destruction [1]. Some of these ways of protection are traditionally realized with the help of the cryptographic algorithms of the data transformation as a programmable module for the universal processors. However, such a realization with the advantage of flexibility of the program module usage and modification has a disadvantage, too: the problem of how to provide the module integrity, and the low datarate. The support of the program module integrity, which is responsible for the cryptographic transformation, is necessary to avoid the key-data distortion and, even more, the program module distortion. The low datarate of the program modules is caused by the discrepancy of the command set of the universal processors with the common operations used in cryptographic algorithms.

One of the possible ways to solve these problems is to use specialized VLSI circuits for data protection. The execution of the cryptographic algorithm with the specialized VLSI circuits allows to avoid the visibility of the algorithm and the key-data integrity. Free access to the content of a VLSI circuit is not possible.

Approach to the design of the Soft-Cores

Modern VLSI circuits of the data protection are characterized by the high level of integration, their high reliability and a wide spectrum of the functional possibilities in the different modes.
Soft-Cores are designed with a hardware description language like Verilog or VHDL. This leads to the approach that you can buy such Soft-Cores from other companies and include them in your own design. The customer is able to complete the core with additional functional blocks to obtain a data protection system built as a system-on-a-chip [4]. Such an approach is called "core-technology" [2, 3].
The use of a Soft-Core can be:

a model of a datapath and a control system with a simplified interface - such an approach allows the designer of a data protection system to use predesigned components or to develop his own specific interface logic and thus to match the requirements;
a complete description with a very specific interface - such an approach makes it possible to get a running data protection system very quickly, but this system can’t be adapted to match complex requirements.

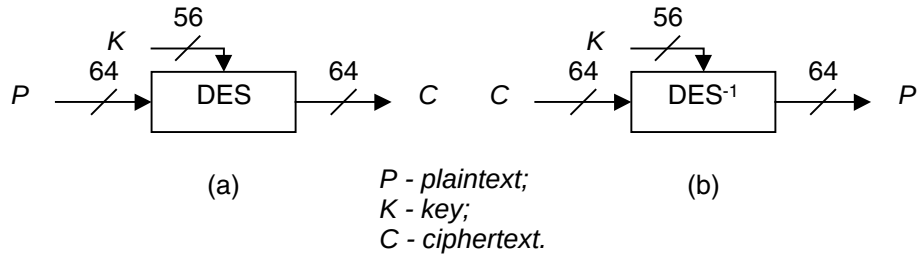
Flexibility in a selection of a data processing mode allows to use a VLSI circuit of a symmetric-key encryption in the data protection systems effectively. However, data protection systems use only a few modes. So, on the one hand, unused modes waste silicon, and such a realization makes it on the other hand impossible to achieve a high datarate. The alternative of the development of a symmetric-key VLSI circuit is a specialized datapath/control architecture for the specific data protection requirements.

DES cipher and DES cipher operating modes

The description of a symmetric-key encryption processes the input text by the fixed-size blocks of n bits (mostly n = 64). For the encipherment of text blocks that have a larger size, a solution is to divide them for some n-bit blocks, each of which can be processed separately.

The most often used algorithm of a symmetric-key encryption is the Data Encryption Standard (DES) [5, 7]. DES is a block cipher which processes plaintext blocks of n = 64 bits, producing 64-bit ciphertext blocks (see fig.1). The effective size of the secret key K is 56 bits; more precisely, the input key K is specified as a 64-bit key, 8 bits of which (bits 8, 16, ..., 64) may be used as parity bits.

Fig. 1: Information interface of the DES algorithm for encryption (a) and decryption (b)



Full details of DES are given in algorithm 1, fig.2 and fig.3. The encryption is proceeded in 16 stages or rounds. From the input key K, sixteen 48-bit subkeys K_i are generated, one for each round. Within each round, 8 fixed, 6-to-4 bit substitution mappings (S-boxes) S_i, collectively denoted S, are used. The 64-bit plaintext is divided into 32-bit halves L_0 and R_0. Each round is functionally equivalent, taking 32-bit inputs L_{i-1} and R_{i-1} from the previous round and producing 32-bit outputs L_i and R_i for $1 \le i \le 16$, as follows:

$$L_i = R_{i-1} \quad (1);$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i), \text{ where } f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i)) \quad (2).$$

Here E is a fixed expansion permutation mapping R_{i-1} from 32 to 48 bits (all bits are used once, some are used twice). P is another fixed permutation on 32 bits. An initial bit permutation (IP) precedes the first round; following the last round, the left and right halves are exchanged and, finally, the resulting string is bit-permuted by the inverse of IP.

A simplified view is that the right half of each round (after expanding the 32-bit input to 8 characters of 6 bits each) carries out a key-dependent substitution on each of the 8 characters, then uses a fixed bit transposition to redistribute the bits of the resulting characters to produce 32 output bits.

Algorithm 2 specifies how to compute the DES round keys K_i, each of which contains 48 bits of K. These operations make use of tables PC1 and PC2, which are called permuted choice 1 and permuted choice 2. To begin with, 8 bits (k_8, k_{16}, ..., k_{64}) of K are discarded (by PC1). The remaining 56 bits are permuted and assigned to two 28-bit variables C and D, and then for 16 iterations, both C and D are rotated either 1 or 2 bits, and 48 bits (K_i) are selected from the concatenated result.

Fig. 2: Data encipherment by DES algorithm

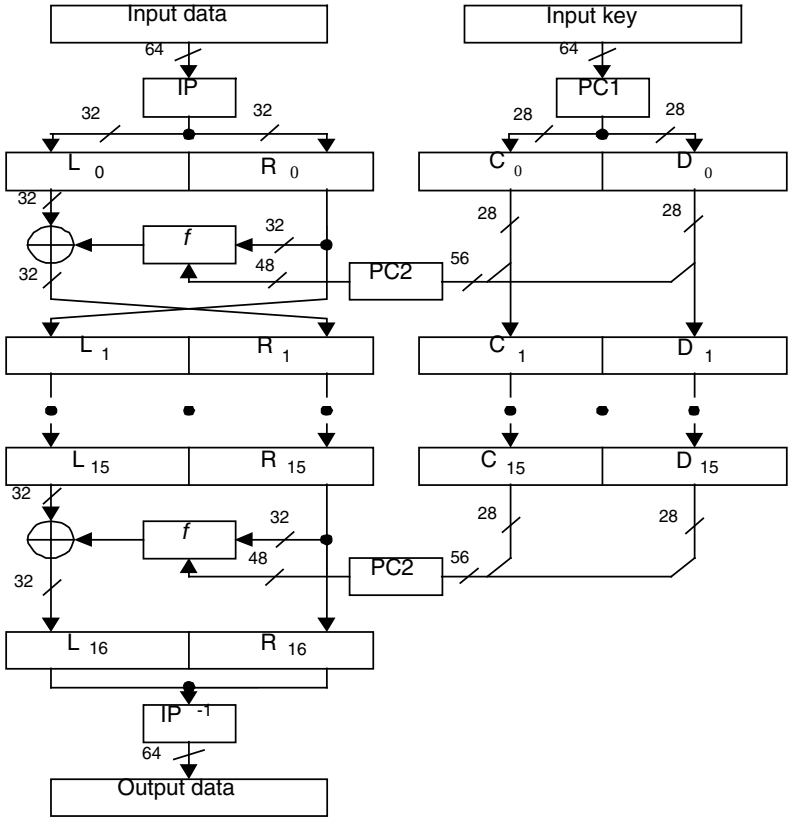
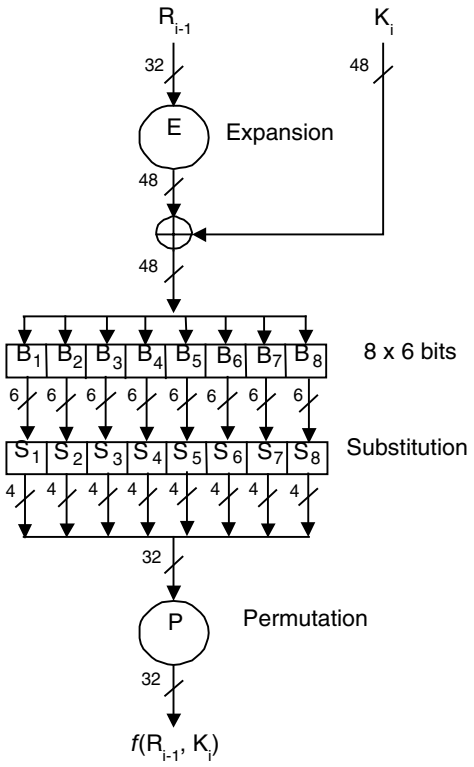


Fig. 3: DES inner function f



Algorithm 1: Data Encryption Standard (DES)

INPUT: plaintext m_1, ..., m_{64}; 64-bit key $K = k_1$, ..., k_{64} (includes 8 parity bits).
SUMMARY: 64-bit ciphertext block $C = c_1$, ..., c_{64}.

1. (key schedule) Compute sixteen 48-bit round keys K_i from K using algorithm 2;
2. $(L_0,R_0) \leftarrow IP\ (m_1, m_2, \ldots, m_{64})$;
3. (16 rounds) for i from 1 to 16, compute L_i and R_i using equations (1) and (2) above.
   Compute $f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$ as follows:
   (a) Expand $R_{i-1} \leftarrow r_1, r_2, \ldots, r_{32}$ from 32 to 48 bits (using E); $T \leftarrow E(R_{i-1})$;
   (b) $T' \leftarrow T \oplus K_i$. Represent T' as eight 6-bit character strings: $(B_1, ..., B_8) = T'$;
   (c) $T'' \leftarrow (S_1(B_1), S_2(B_2), \ldots, S_8(B_8))$;
   (d) $T''' \leftarrow P(T'')$;
4. $b_1, b_2, \ldots, b_{64} \leftarrow (R_{16}, L_{16})$ (Exchange final blocks L_{16}, R_{16});
5. $C \leftarrow IP^{-1}(b_1, b_2, \ldots, b_{64})$ (Transpose using IP^{-1}).

Algorithm 2: DES key schedule

INPUT: 64-bit key $K = k_1, k_2, ..., k_{64}$ (including 8 odd-parity bits).
SUMMARY: sixteen 48-bit keys, K_i, $1 \le i \le 16$.

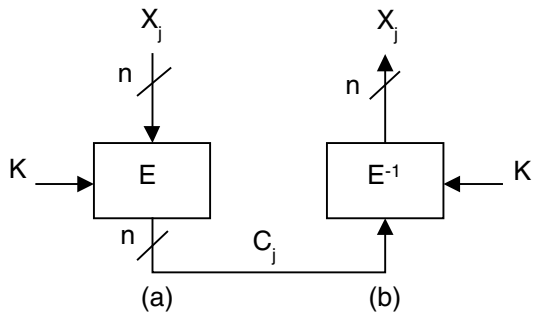
1. Define V_i as follows: $V_i = 1$ for i from {1, 2, 9, 16}; $V_i = 2$ otherwise
   (These are left-shift values for 28-bit circular rotations below);
2. $T \leftarrow PC1(K)$; represent T as 28-bit halves (C_0, D_0);
3. For i from 1 to 16, compute K_i as follows: $C_i = C_{i-1}$ **rol** V_i, $D_i = D_{i-1}$ **rol** V_i.
   $K_i = PC2(C_i, D_i)$ (**rol** denotes **left** circular shift).

DES decryption consists of the encryption algorithm with the same key but reversed key schedule, using in the order of (K_{16}, ..., K_1). Subkeys K_1, K_2, ..., K_{16} may be generated by algorithm 2 and used in reverse order, or generated in reverse order directly as follows. Note that after K_{16} is generated, the original values of the 28-bit registers C and D are restored (each has rotated 28 bits). Consequently, and due to the choice of shift-values, modifying algorithm 2 generates subkeys as follows: replace the left-shifts by right-shift rotates; change the shift value V_1 to 0.

Most widely used in block ciphers are the following four modes: ECB, CBC, CFB, and OFB [6, 7]. Each of them has its own advantages and disadvantages, but all of them use the DES cipher. In the following E_K denotes the encryption function. E_K^{-1} denotes the decryption function. Input plaintext $X = X_1, \ldots, X_t$ consists of n-bit blocks for ECB and CBC modes and r-bit blocks for CFB and OFB blocks, where r is a fixed value, $r \le n$.

Electronic Codebook (ECB) is the simplest mode of DES algorithm work. It is illustrated in fig.4 and in algorithm 3.

Fig. 4: ECB mode encipherment (a) and decipherment (b)



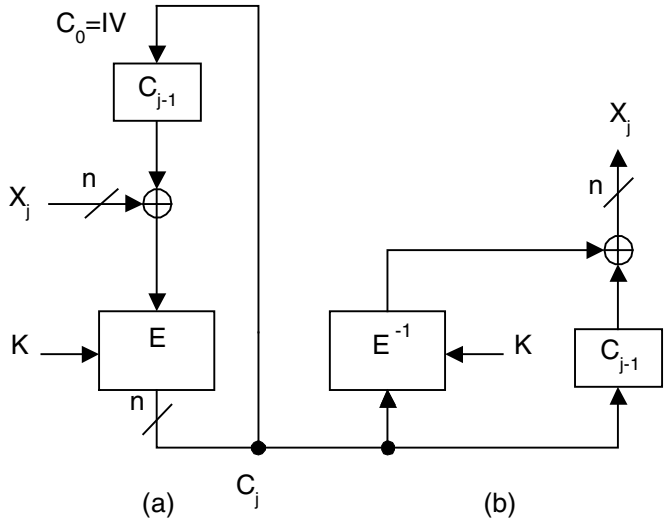
Algorithm 3: ECB mode of operation

INPUT: n-bit key K; n-bit plaintext blocks X_1, ..., X_t.
SUMMARY: produce ciphertext blocks C_1, ..., C_t; decrypt to recover plaintext.

1. Encryption: for $1 \le j \le t$, $C_j \leftarrow E_K(X_j)$;
2. Decryption: for $1 \le j \le t$, $X_j' \leftarrow E_K^{-1}(C_j)$.

The cipher-block chaining (CBC) mode of operation, specified in algorithm 4 and illustrated in fig.5, involves the use of an n-bit initialization vector, denoted IV.

Fig. 5: CBC mode encipherment (a) and decipherment (b)



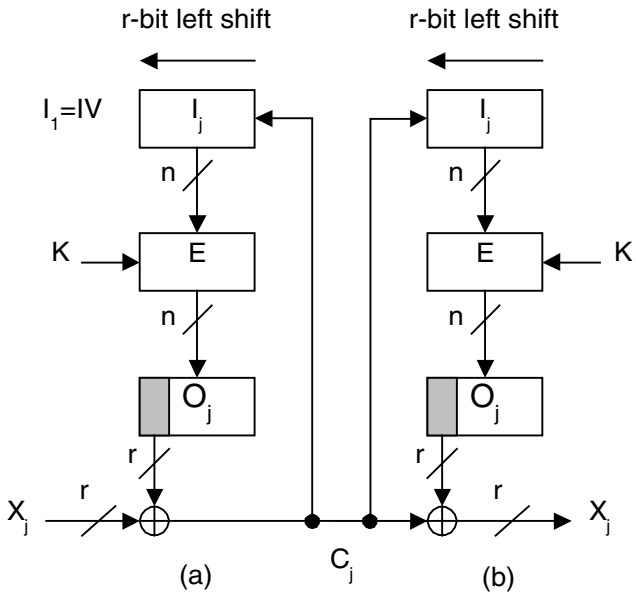
Algorithm 4: CBC mode of operation

INPUT: n-bit key K; n-bit key IV; n-bit plaintext blocks X_1, ..., X_t.
SUMMARY: produce ciphertext blocks C_1, ..., C_t; decrypt to recover plaintext.

1. Encryption: $C_0 \leftarrow IV$, for $1 \le j \le t$, $C_j \leftarrow E_K(C_{j-1} \oplus X_j)$;
2. Decryption: $C_0 \leftarrow IV$, for $1 \le j \le t$, $X_j' \leftarrow C_{j-1} \oplus E_K^{-1}(C_j)$.

While the CBC mode processes plaintext n-bits at a time (using an n-bit block cipher), some applications require that r-bit plaintext units are encrypted and transmitted without delay, for some fixed r<n (often r=1 or r=8). In this case, the cipher feedback (CFB) mode may be used, as specified in algorithm 5 and illustrated in fig.6.

Fig. 6: CFB mode encipherment (a) and decipherment (b)



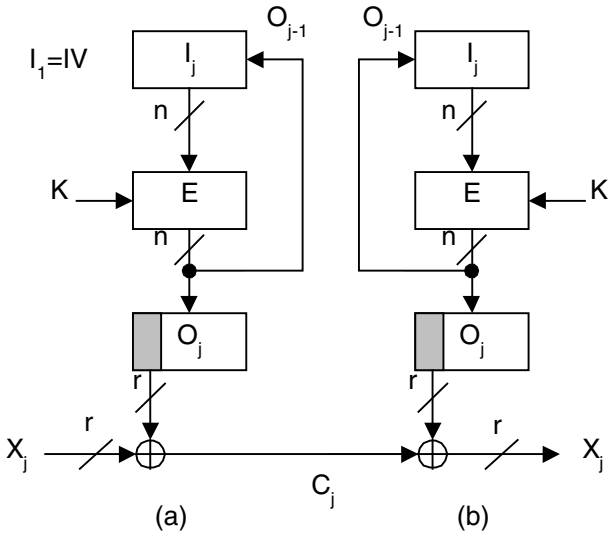
Algorithm 5: CFB mode of operation (CFB-r)

INPUT: n-bit key K; n-bit key IV; r-bit plaintext blocks X_1, ..., X_t.
SUMMARY: produce r-bit ciphertext blocks C_1, ..., C_t; decrypt to recover plaintext.

1. Encryption: $I_1 \leftarrow IV$ (I_j is the input value in a shift register) For $1 \le j \le U$:
   (a) $O_j \leftarrow E_k(I_j)$ (Compute the block cipher output);
   (b) $t_j \leftarrow$ r leftmost bits of O_j (Assume the leftmost is identified as bit 1);
   (c) $C_j \leftarrow X_j \oplus t_j$ (Transmit the r-bit ciphertext block C_j);
   (d) $I_{j+1} \leftarrow 2^{r*} I_j + C_j \bmod 2^n$ (Shift C_j into right end of shift register);
2. Decryption: $I_1 \leftarrow IV$. For $1 \le j \le U$, upon receiving C_j:
   $X'_j \leftarrow C_j \oplus t_j$, where t_j, O_j and C_j are computed as above.

The output feedback (OFB) mode of operation may be used for applications in which all error propagation must be avoided. It is similar to the CFB mode, and allows encryption of various block sizes. In opposite to the OFB mode the encryption function E supports the feedback of the output data. The two versions of the OFB mode share one n-bit block cipher. The ISO version (see fig.7 and algorithm 6) requires an n-bit feedback, and is more secure. The earlier FIPS version (see fig.7 and algorithm 7) allows r<n bits of feedback.

Fig. 7: OFB mode encipherment (a) and decipherment (b)



Algorithm 6: OFB mode with full feedback (per ISO 10116)

INPUT: n-bit key K; n-bit key IV; r-bit plaintext blocks X_1, ..., X_t.
SUMMARY: produce r-bit ciphertext blocks C_1, ..., C_t; decrypt to recover plaintext.

1. Encryption: $I_1 \leftarrow IV$. For $1 \le j \le n$ given plaintext block X_j:
   (a) $O_j \leftarrow E_k(I_j)$ (Compute the block cipher output);
   (b) $t_j \leftarrow$ r leftmost bits of O_j (Assume the leftmost is identified as bit 1);
   (c) $C_j \leftarrow X_j \oplus t_j$ (Transmit the r-bit ciphertext block C_j);
   (d) $I_{j+1} \leftarrow O_j$ (Update the block cipher input for the next block);
2. Decryption: $I_1 \leftarrow IV$. For $1 \le j \le n$, upon receiving C_j:
   $X'_j \leftarrow C_j \oplus t_j$, where t_j, O_j and C_j are computed as above.

Algorithm 7: OFB mode with r-bit feedback (per FIPS 81)

INPUT: n-bit key K; n-bit key IV; r-bit plaintext blocks X_1, ..., X_t.
SUMMARY: produce r-bit ciphertext blocks C_1, ..., C_t; decrypt to recover plaintext.

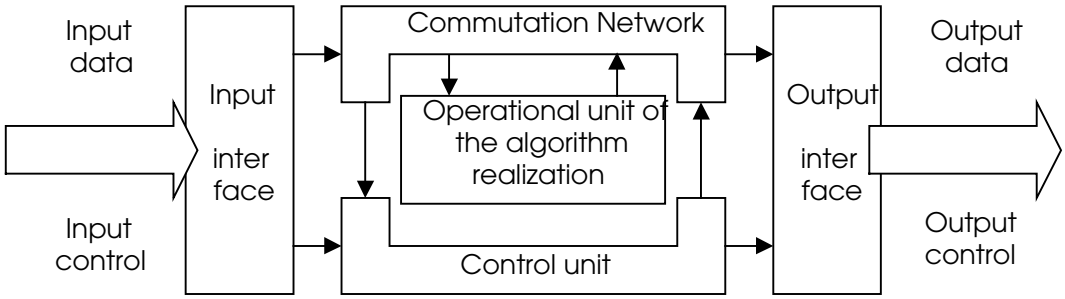
As per algorithm 6, but with $I_{j+1} \leftarrow O_j$ replaced by:
$I_{j+1} \leftarrow 2^r * I_j + t_j \bmod 2^n$ (Shift t_j into right end of shift register).

The ECB mode supports no dependencies between the text blocks which are processed. This decreases the reliability of the cipher, as an identical plaintext block causes the appearance of an identical ciphertext block, when the same key for their encipherment is used. For this reason, the ECB mode is not recommended for messages longer then one block, or if keys are reused for more then a single one-block message. Security may be improved by using other modes of work, where feedbacks are implemented. As a result, reliability is higher because each currently processed block depends on previously processed blocks.

Variations of the DES Soft-Core model

Let's take a closer look at the variations of the DES Soft-Cores for data protection. A general architecture of such core [8] is shown in fig.8.

Fig. 8: General architecture of the DES Soft-Core



The operational unit of the algorithm realization (OUAR), which represents a model of the core's datapath, is a hardware implementation of the DES algorithm and can be used with the described modes of the DES cipher. The Commutation Network configures the core in a certain mode. The input and the output interfaces control the acceptance of the input data, the storage of the intermediate results and the registering of the output data. The partitioning of the core with the described units shows the following behavior: the algorithm of the OUAR work remains fixed for all modes, other units (Commutation Network, Control Unit, input and output interfaces) show a dependency on the selected mode.

Let's consider the variations of the OUAR architecture. There are two basic architectures for the OUAR - iterative and pipelined. The iterative architecture is shown in fig.9, the pipelined - in fig.10.

Fig. 9: Iterative architecture of OUAR

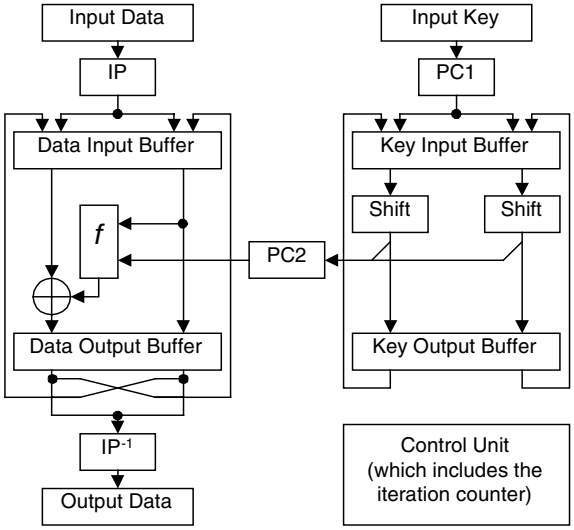
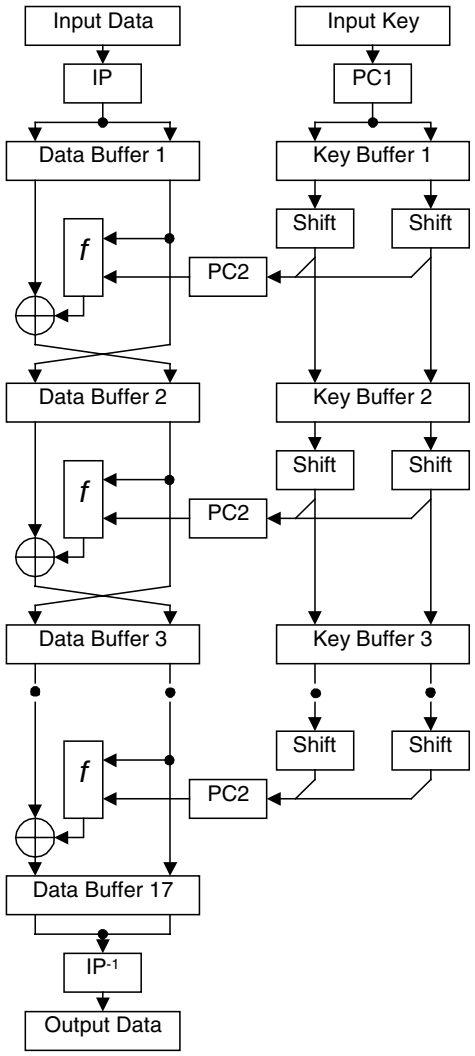


Fig. 10: Pipelined architecture of OUAR



The iterative solution of the OUAR unit generates synthesis results with less gatecount. The pipelined architecture is characterized by a high datarate.

Table 1 shows the possible variations for the DES Soft-Core functionality. The OUAR architecture and the functionality[1] are parameters. A Multimode Soft-Core performs data encipherment (encryption and decryption) in the four modes which are considered.

[1] mode, encipherment direction

Table 1: Possible variations for the DES Soft-Core implementation[2]

#	OUAR architecture	Functional orientation	ECB		CBC		CFB		OFB	
			E	D	E	D	E	D	E	D
1	Iterative	Multimode	✓	✓	✓	✓	✓	✓	✓	✓
2		Specialized	✓	✓						
3					✓	✓				
4							✓	✓		
5									✓	✓
6			✓							
7				✓						
8					✓					
9						✓				
10							✓			
11								✓		
12									✓	
13										✓
14	Pipelined	Multimode	✓	✓	✓	✓	✓	✓	✓	✓
15		Specialized	✓	✓						
16					✓	✓				
17							✓	✓		
18									✓	✓
19			✓							
20				✓						
21					✓					
22						✓				
23							✓			
24								✓		
25									✓	
26										✓

Possible variations of the DES algorithm and modes are shown in table 1. The expediences of an implementation for a DES-Soft-Core will be discussed.

Comparative analysis of DES Soft-Core models

Let's have a closer look now at the general advantages and disadvantages of the iterative and pipelined architectures of the OUAR-unit.

As has been said above, the iterative architecture of the OUAR-unit leads to a design with less gatecount and a low datarate. A pipelined architecture produces a high datarate and the gatecount is increased. One feature of the DES algorithm should be taken into account: data is processed within sixteen identical rounds. A fully iterative architecture of the OUAR-unit consists of one DES round, where the data block circulates sixteen times. A fully pipelined structure consists of sixteen rounds, separated by the pipeline registers. The pipelined OUAR-unit is able to process the data parallel in sixteen blocks. The performance of the pipelined OUAR-unit is sixteen times higher than the performance of the iterative one. If the combined iterative-pipelined architecture is realized[3], the datarate and the gatecount increase in comparison to a fully iterative architecture. It has been stated [9], that a good *trade off* for the number of

[2] E: encryption; D: decryption

[3] it could be represented as an iterative architecture, which contains some rounds of the pipeline

cycles to process one data block and the necessary gatecount are reached by the fully pipelined architecture of the OUAR-unit.

Multimode DES Soft-Core models

The performance of the core in the ECB mode (see fig.4) is the same as the maximum performance of its OUAR-unit. According to the requirements for a high datarate or for a low gatecount the use of an iterative or a pipelined architecture is possible. For the CBC mode it is not effective to use the pipelined architecture for the OUAR-unit (see fig.5), because the unit can process - due to the feedback - only one data block in time. In the case of decryption there are no feedbacks, that's why the pipelined OUAR-architecture is very suitable for a high datarate. The same situation can be seen in the CFB mode (see fig.6). In the case of the OFB mode (see fig.7) there are feedbacks which don't allow the processing of more than one data block in time. So, in this case the iterative OUAR-architecture is qualified, too. In summary the multimode DES Soft-Core creation with an iterative architecture fits the requirements for higher datarate and a low gatecount for most of the modes. The datarate decreases only for the ECB mode for encryption and decryption and the CBC and CFB modes for decryption using the pipelined architecture.

Specialized DES Soft-Core models

Generally a specialized DES Soft-Core is characterized by a small field of application in comparison with a multimode DES Soft-Core. The possibility to achieve a high datarate and a low gatecount compensates this disadvantage in most applications.

The ECB mode has no feedbacks, the usage of a pipelined/iterative OUAR-architecture makes sense and is possible. Two different cases occur in the CBC and CFB modes: iterative OUAR-architecture for encryption is effective, because a feedback doesn't allow to handle more than one data block in time; for decryption both architectures are effective, because there is no feedback. In this way both OUAR-architectures are possible. In the case of the OFB mode, there is a feedback, which doesn't allow to process more than one data block in time. In these cases only the iterative OUAR-architecture can be used.
The development of a dedicated Soft-Core with regard to the encipherment direction allows to achieve a low gatecount and to obtain a high datarate. For the Soft-Cores, dedicated to the ECB and OFB modes with regard to the encipherment direction, all the features which have been discussed in this paper remain. Considering the CBC and CFB modes regarding to the encipherment direction, the iterative OUAR-architecture should be favoured for encryption. For decryption both architectures are possible.

Let's have a closer look at a Soft-Core which supports some modes. The similarity of the algorithms of these modes should be checked. For a high datarate in the ECB, CBC and CFB modes for encryption the pipelined OUAR-architecture should be preferred. In other cases an iterative OUAR-architecture should be possible. The complexity of other core-units[4] isn't dramatically increased for these Soft-Cores. The described Soft-Core Architecture supports a wide spectrum of functional possibilities, but requires a higher gatecount for its realization.

[4] Commutation Network, Control Unit, Input and Output Interfaces

Available Soft-Cores for information protection

The available Soft-Cores can be characterized as shown in the following table.

Table 2: Some existing Soft-Cores for information protection

Producer	FPGA / ASIC Technology	Modes	Interface	Databus	Fre-quency (MHz)	Gatecount
Alatek	EPF6016-2	ECB	special	64 bit	39	623 LC
	EPF81500-2				36	634 LC
	EPF10K50V-1				43	663 LC
	XLC30-3				24	264 CLB
	XC4013XL-08				55	266 CLB
	V150-6				100	281 Slices
CAST	Virtex V150-6	ECB	special	64 bit	101	255 Slices
	EPF6016-2				37	540 LC
	EPF81500-2				35	540 LC
	EPF10K20-3				44	540 LC
	EPF10K30A-1				73	570 LC
	EPF10K50V-1			8 bit	27	574 LC
Inventra	CBA library	ECB	special	64 bit	100	4000 Gates
	CBA library	ECB, CBC			100	7000 Gates
Memec Design	XC4000E/XL	ECB	special	64 bit	43	316 CLB
	Spartan				25	316 CLB

The cores (table 2) have a very specialized functionality and most of them support the ECB Mode only. As has been said before the ECB mode has a low reliability, which is due to the lack of feedback. To develop a Soft-Core which supports the other modes - CBC, CFB and OFB - is an important task.

The described ideas have been realized in the project “Development of a Data Encryption Standard IP Generator”. An Internet-version of this generator is available online [5]. The Core Generator supports the download of 46 different synthesizeable VHDL-files. The use of these cores leads to a shorter time-to-market for new security products. The project was sponsored by AIF (Arbeitsgemeinschaft industrieller Forschungsvereiniungen “Otto von Guericke” e.V.).

Conclusion

In this paper the Soft-Cores for information protection have been analyzed. The analysis of the technical characteristics of existing Soft-Cores shows a high amount of specialization. Flexible designs, fast realization and high design quality require parameterizable models and libraries of Soft-Cores for information protection systems. Using them has the following advantages:

- the development process is dramatically simplified and rapid;
- a high level of integration is achieved and errors are virtually impossible;
- the customer can choose the cores which match his requirements.

The variations of the DES Soft-Cores architecture for information protection have been considered. The focus has been laid on the following points:

- principles of the data processing by the DES block cipher;
- modes of DES cipher work: ECB, CBC, CFB and OFB;
- variations of the multimode DES Soft-Core architecture that process the data in all the modes of the DES cipher work;

[5] http://CoreGenerator.nf.fh-nuernberg.de:8080/CoreGenerator/

- variations of the dedicated DES Soft-Core architecture that process the data in one certain mode of the DES cipher work for encryption and decryption;
- variations of the dedicated DES Soft-Core architecture that process the data in one certain mode of the DES cipher work for encryption only;
- variations of the dedicated DES Soft-Core architecture that process the data in one certain mode of the DES cipher work for decryption only.

These Soft-Cores are analyzed and compared with each other. Two basic variations of the architecture of the cores' datapaths are identified - the iterative and pipelined one. The iterative architecture provides a low datarate and a low gatecount. The pipelined version provides a high datarate and a high gatecount. An approach for a combined iterative-pipelined datapath is described. An estimation about the Soft-Cores' architectures that support some similar modes (for instance, CBF and OFB) is provided.

The characteristics of the existing DES Soft-Cores are considered. Their disadvantage in general is their limited functionality - an absence of the cores that work in CFB and OFB modes. In the paper twenty six variations of the DES Soft-Cores' architecture are shown. These cores have a wide functionality - all modes are supported - which allows using them in a wide spectrum of the application fields.

The described ideas have been realized in the project "Development of a Data Encryption Standard IP Generator". An Internet-version of this generator is available.online. The Core Generator supports the download of 46 different synthesizeable VHDL-files. The use of these cores leads to a shorter time-to-market for new security products.

Acknowledgement

The authors acknowledge AIF (Arbeitsgemeinschaft industrieller Forschungsvereiniungen "Otto von Guericke" e.V.) for supporting this project.

Literature

[1] H. Feistel, "Cryptography and computer privacy", Scientific American, Vol.228, N5, May 1993, pp. 15-23.
[2] P. Lapsey and J. Bier "DSP Cores Bring New Level of Integration"//Microprocessor report, August 1994.
[3] DSP Design Tools and Mothodologies, Berkeley Design Technology, Inc. (Fremont, California), 1995.
[4] M. Keating, P. Bricaud "Reuse Methodology Manual for System-On-a-Chip Design", Kluwer Academic Publishers, 1999, pp.224.
[5] FIPS 46, "Data Encryption Standard", Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C.
[6] FIPS 81, "Operational Modes of DES", Federal Information Processing Standard (FIPS), Publication 81, National Bureau of Standards, U.S. Department of Commerce, Washington D.C.
[7] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone "Handbook of Applied Cryptography", CRC Press, October 1996, 816 p.
[8] V. Melnyk, T. Korkishko "DES Cryptographic Processor". Report on the research project. Georg-Simon-Ohm Fachhochschule Nuernberg, 27 September 1999 - 28 November 1999, 178 p.
[9] T. Korkishko, A. Melnyk "Cryptographic Processor Architectures for DES Algorithm", Proceedings of AFRICON'99 conference, Cape Town, South Africa 1999, pp. 126-131.