APPLICATIONS OF CHAOS THEORY IN NUMERICAL DATASTREAMS CRYPTOGRAPHY. MATHEMATICAL MODEL AND IMPLEMENTATION METHOD

Authors: **Constantin Grozea**[*]
Dan Laurentiu Grecu[*]

[*] Main Researcher Engineer - Ministry of National Defence, Romania
[*] Main Researcher Engineer - Ministry of National Defence, Romania

Abstract:

In the military domain the signal voice coding constitutes a high concern in the field of sending news. The algorithms coding utilization based on the chaos theory open a new chapter in this field. The self-synchronize algorithms, from the cryptic schemes used, which use the chaos for coding, are useful due the fact that do not require any initial synchronize protocol. The synchronize achievement is even made using the flux of cryptic dates. In the present paperwork we suggest a self-synchronize algorithm of cryptic using the chaos theory and a scheme of practical achievement of this one with logical programming areas.

1. Theoretic notions

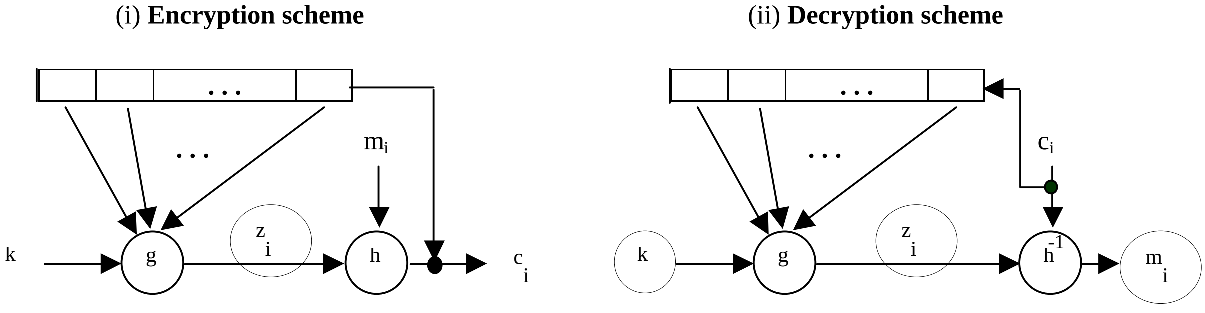
For the signal voice coding can use the cryptographic schemes that have a base with two principles: the initial synchronize method and the self-synchronization method. Usually, in the classic cryptography are preferred the synchronize schemes instead of those self-synchronizes, due the fact that they have a higher potential quality and their cryptographic properties can be proved easily. Although the self-synchronized encryption, when they are correctly realized (using unlinear functions), they have the advantage of combining the security of information with the facility of synchronize.

The coding using the self-synchronize schemes are those in which the pseudoaleator sequence (*key stream)* it is generates as a key function and a previous fix numbers of bits chosen by the crypto message. These coding can be named also asynchronous. These cryptographic functions can be described by the following equations:

$$\begin{aligned} s_i &= (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}) \\ z_i &= g(s_i, k) \qquad\qquad (1) \\ c_i &= h(z_i, m_i) \end{aligned}$$

Where $s_0 = (c_{-t}, c_{-t+1}, \ldots, c_{-1})$ it is the initial situation (unsecured), *k* is the key, *g* is the function that produces the pseudoaleator sequence z_i, and *h* is the out function.

The encryption and decryption processes are presented in figure 1.


(i) **Encryption scheme**

(ii) **Decryption scheme**

Figure 1. *The general model of a self-synchronization schemes*

The signal voice coding with the help of the self-synchronization snapshots codes have the following properties:

a). The self-synchronization: this is possible only if the bits from the crypto message are omitted or inserted, as the mapping decrypting depends by the fixed numbers of precedent characters of a crypto message. These codes are capable to restore the automatical right decrypted after the lost have synchronized, only with a fix number of bits from the plain message irretrievable.

b). The limited errors conduction: we suppose that the situation of a self-synchronize snapshot code depends on previous *t* bits from the crypto message. If one bit from the crypto message is modified (even omitted or inserted) during the transmission time, then the decrypted a till t bits can be incorrect, after that the right decrypted is resumed.

c). Active attacks: the second property implies the fact that the changing of the bits from the crypto message by an active opponent causes the incorrect decrypt of some others bits; it is why the probability of being detected by the decrypted is being improved (in compare with the snapshot synchronous codes). As a conclusion for the first property, it is more difficult now to detect a bits insert, an omission or a retort from the crypto message by an active opponent. That is why we need additional mechanisms in order to certify the source and the integrity guarantee of the dates.

d). Diffusivity of the statistics properties for a plain message: as a bit from the plain text influences the entire crypto message, the statistics properties of the plain message scatteres the whole crypto message.

In this paperwork we model and introduce a cryptographic voice scheme having as a base the unlinear functions (chaos). We start with a self-synchronized classical cryptor, described by the following relation:

$$y_n = c_o \oplus \left(c_1 \otimes y_{n-1}\right) \oplus \left(c_2 \otimes y_{n-2}\right) \oplus \ldots \oplus \left(c_k \otimes y_{n-k}\right) \oplus \ldots \qquad (2)$$

Where $\oplus$ and $\otimes$ represent the addition and multiplication of the modulus *m* in the ring Z_m.

The cryptor described with the relation (2) we added the unlinears functions result the block scheme presented in figure 2.

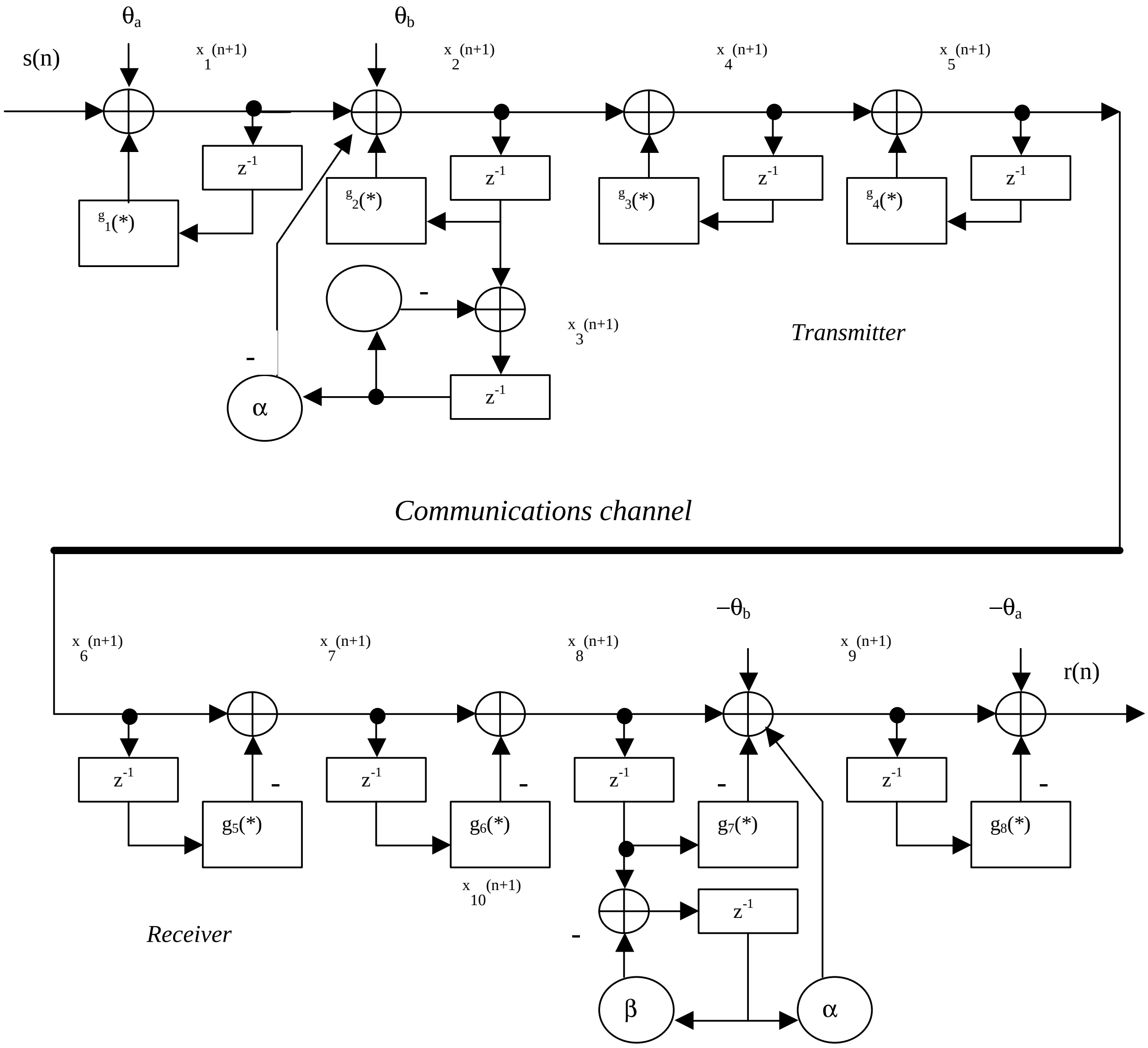


Figure 2. *The block scheme of a voice signal communication uses the chaos*

Between the variables presenting by block scheme there is the next mathematical relations:

a). For the transmitter:

$$\begin{aligned} x_1(n+1) &= g_1(x_1(n)) + s(n) + q_a \\ x_2(n+1) &= g_2(x_2(n)) - a x_3(n) + x_1(n+1) + q_b \\ x_3(n+1) &= x_2(n) - b x_3(n) \\ x_4(n+1) &= g_3(x_4(n)) + x_2(n+1) \\ x_5(n+1) &= g_4(x_5(n)) + x_4(n+1) \end{aligned} \tag{3}$$

b). For the receiver:

$$\begin{aligned} x_7(n+1) &= x_6(n+1) - g_5(x_6(n+1)) \\ x_8(n+1) &= x_7(n+1) - g_6(x_7(n)) \\ x_9(n+1) &= x_8(n+1) - g_7(x_8(n+1)) + a x_{10}(n) - q_b \\ x_{10}(n+1) &= x_8(n) - b x_{10}(n) \\ r(n) &= x_9(n+1) - g_8(x_9(n)) - q_a \end{aligned} \tag{4}$$

where:

- $x_k(n); k = 1, 2, 4, 5, 6, 7, 8, 9$ - internal statuses of the chaos system;
- $x_3(n), x_{10}(n)$ - internal statuses for the linear subsystem;
- $s(n)$ - digital voice input signal;
- $r(n)$ - digital voice output signal;
- $g_1(*), g_2(*), g_3(*), g_4(*), g_5(*), g_6(*), g_7(*), g_8(*)$ - unlinears functions;
- q_a, q_b - the threshold constants.

The following relation describes the unlinears functions:

$$g_k(x) = \begin{cases} kx + s\,; x < 0 \\ 0; x = 0 \\ kx - s\,; x > 0 \end{cases} ; k = \overline{1,8} \qquad (5)$$

The important parameter that decides the system characteristics represents the way of combine the unlinears functions. In this situation, from motives of simplicity of implementation, we consider the eight unlinears functions as being equals. In order to check the correctness of the mathematics way and to test the quality of the voice signal code, we made a computer programmer which implement the equations (3), (4) and (5). For testing the statistics quality of the snapshots generated by the transmitter, we used a testing programmer of data files that use ten statistics tests for appreciating the aleatory. Due the experiences made, we conclude that the scheme propose for voice coding has best statistics quality, and the self-synchronize property is verified.

2. The development micro system

The development micro system on which was put the cryptographic algorithm is made as having a central element, a FPGA circuit of AT40K20 series. It also has an EEPROM AT17C256 memory and a microcontroler type AT90S2313. The microcontroler is used to load the configuration programme in FPGA or in the configuration memory EEPROM. The microcontroler also assures a serial chain with a compatibil computer IBM-PC, by which follows to be charge the programme transfer is made in FPGA or EEPROM. The development microsystem.block scheme is presented in figure 3.

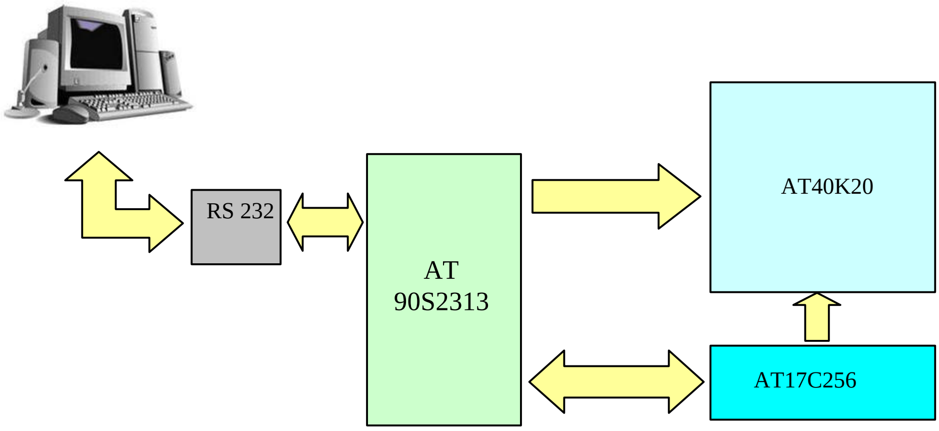


Figure 3 - *Block scheme for un development microsystem with FPGA*

The AT40K is a family of fully PCI-compliant SRAM-based FPGAs with distributed 10ns programmable synchronous/asynchronous dual port/single port SRAM, 8 global clocks, Cache Logic ability, automatic component generators, and range in size from 5,000 to 50,000 usable gates and support 3V and 5V designs. The Atmel's AT40K Cache Logic® FPGA is a symmetrical array of identical cells. The array is continuous from one edge to the other, except for bus repeaters spaced every four cells. The RAM can be configured as either a single-ported or dual-ported RAM, with either synchronous or asynchronous operation. At the lower right corner of each sector is a 32 x 4 FreeRAM™ block accessible by adjacent buses (figure 4).

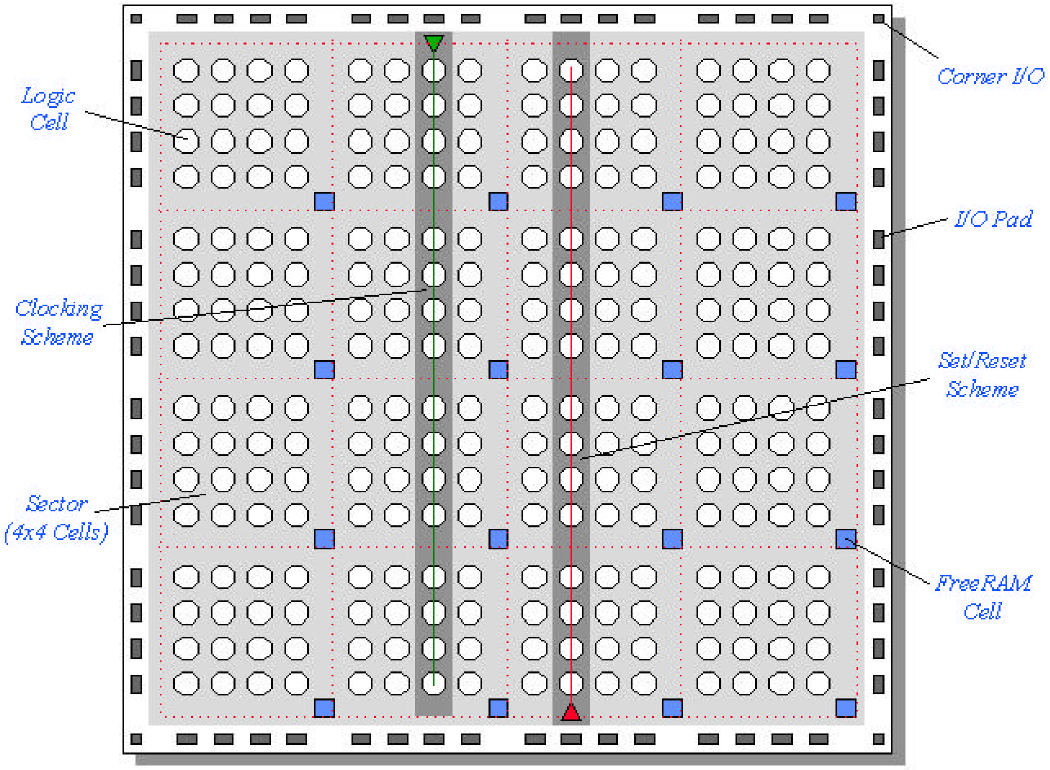


Figure 4 *- Block diagram of FPGA*

The power of the AT40K cell comes from the way it can be configured in a number of different "modes", making it very flexible for implementing a wide variety of digital designs (figure 5).

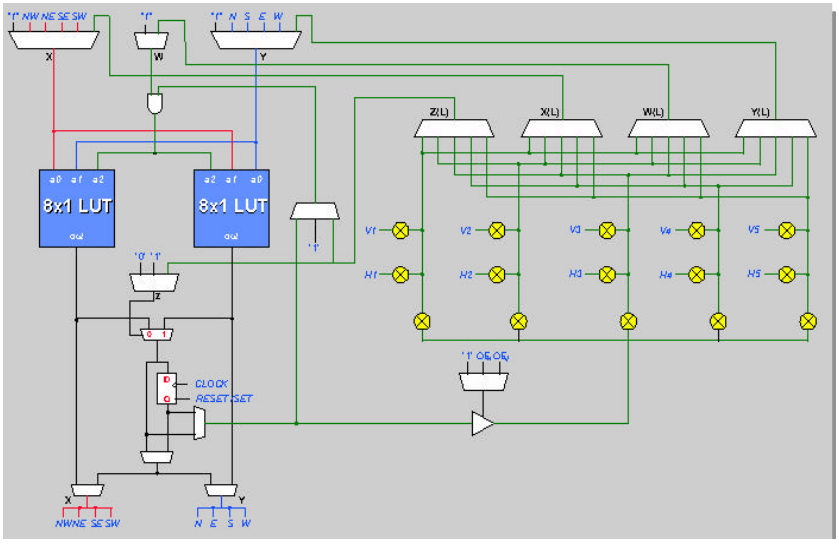


Figure 5 *- Block diagramm of a FPGA cell*

The AT40K logic cell can be configured in several "modes", as shown below figure 6.

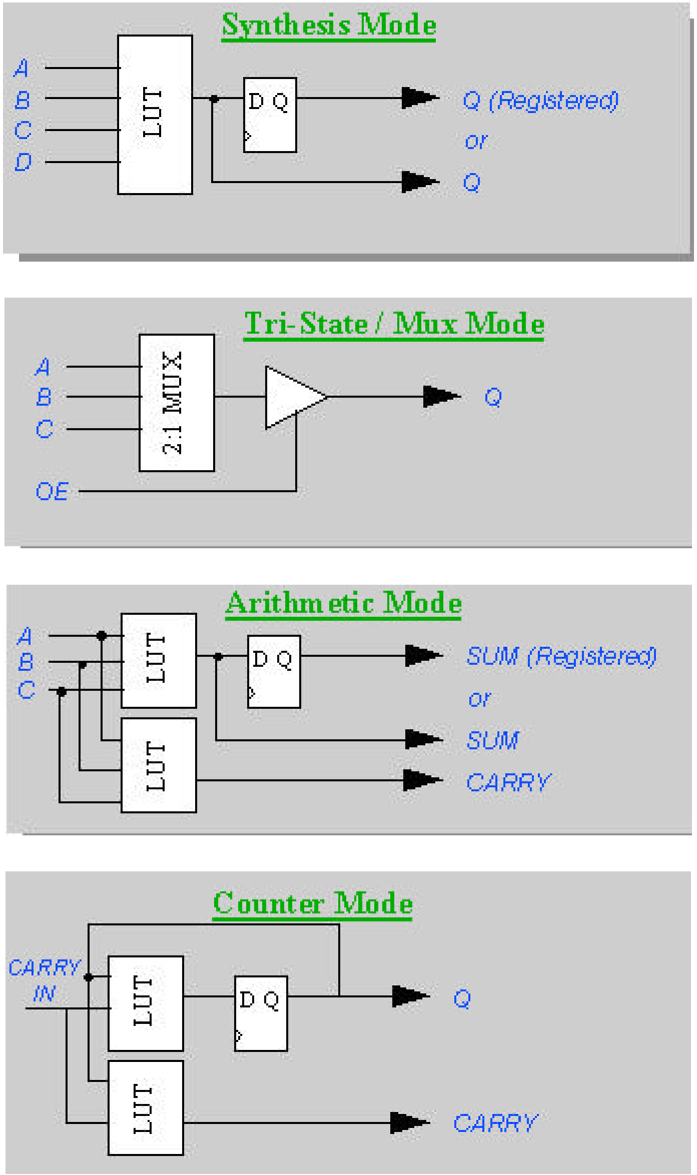


Figure 6 - *Possible modes of configuration for a cell*

The configuration of a FPGA it the loading process of the customer design. AT40 family it's based on SRAM memory witch could be configured any time. The time of configuration is less than 10 milliseconds. The configuration data can be transferred in the circuit in 6 modes. For this exists three input pin witch get the mode of configuration. For this pins it's possible to select one mode Master, four-mode slave and one asynchronous mode for directly accessing the memory of configurations.

The configuration will be done via four states. First status appears when the supply blocks is turned on. In this status the circuit erases the memory configuration. Next status appears when the user touches the reset button. In the other states, the configuration data it's loaded into configuration memory.

The design witch is implemented uses Mode 7 with the following singularities:

- The data source configuration: EEPROM or micro controller;
- The configuration pins used: RESET, CON, M0, M1, M2 and CCLK;
- The pins with double utilizations: D0, INIT, LDC and HDC;
- Optional pins with double utilisations: CSOUT, CHECK.

In Mode 7 the FPGA used in the system loads data from the EEPROM or micro controller. Maximum frequency will be 33 MHz, and the loading time is less than 4.7 milliseconds for AT 20K20.

The microcontroller was selected for its high performance and low power consumption. It contains 118 instructions that are executed in a single machine clock cycle. Also, it contains 2K of flash memory.

Characteristics:

- 128 bytes internal memory;
- 15 programmable I/O lines;
- 8 bits timer with separate prescaler;
- 16 bits timer with separate prescaler;
- Full-duplex UART;
- external interrupt pins;
- 20 pins device.

The configuration for AT40Kxx will be done with EEPROM memory. Serial bus used consists of two buses: data and clock. The data bus is bidirectional and the microcontroller has a build in mechanism for data control and transfer.

The information to be transmitted is grouped in frames. Each frame is preceded by a START condition end will be finished by a STOP condition. The frame consists in several bytes and each byte is composed by 8 bits and an acknowledge bit. The frame format allows the performing of data read and write.

The algorithm implementation was made by two functional modes of projection - one is encryption and the other is decryption - in order to verity the propuse solution (figure 7).

Each of these functionals modes contains the following blocks:

- A block of parallel to serial conversion for data which is following to be encrypted;
- A configure block of crypto key stream;
- The encryption / decryption block where the crypto function is impemented;
- A block of parallel to serial conversion for the serialize cryptostream;
- A block for tact division that is consisted in a time base for the encryption function.

This is the way of working:

- The datastream for being decryptioned is passed through a notebook of parallel to serial that it is read by a signal generated by the division block. This information is processed by the encryption block which after applies the encryption function offers the result to the block of parallel to serial conversion for the propose of the encrypyion information parallel to serial conversion;
- The scheme of decryption works in an analogous way mentioning that the encryption function is replaced with the decryption function. It takes the datasteam and after the application of the decryption function, offers the plain input datastream.

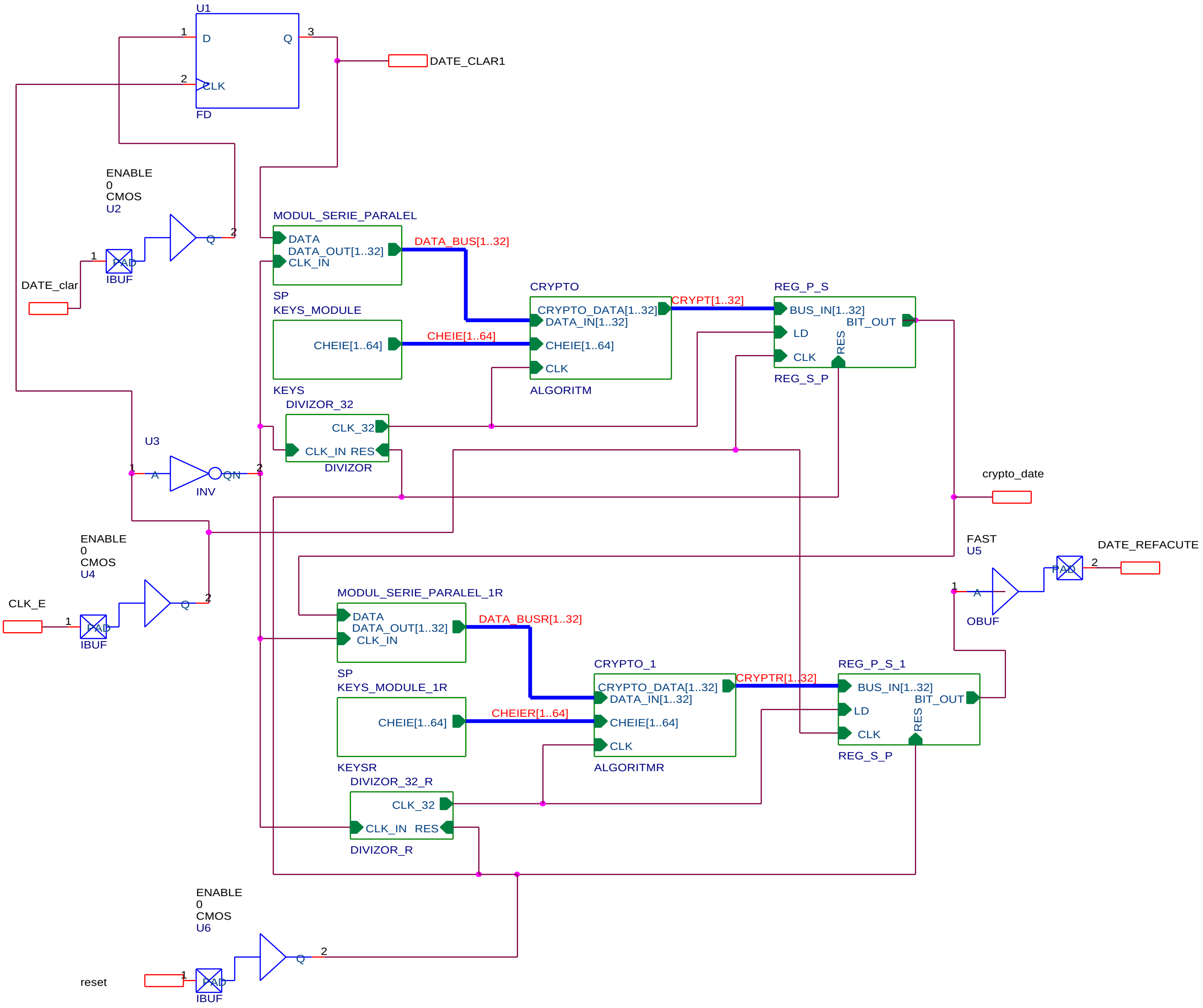


Figure 7. *The block scheme of the enctyption / decryption scheme implemented with FPGA*

3. Conclusions

From above mentioned, results that we can obtain a best coding using the chaos for encryption.

The simplicity of this way recommends this method to be used in the military field.

The above solution represents the advantage of simplicity, that means reduce costs for physics working, and also the insurance possibility of local management for encryption keys by using a microcontroller.

References

[1] Frank Dachselt, Kristina Kelber, Wolfgang Schwarz, Joos Vandewalle, “Chaotic versus classical stream ciphers - a comparative study”, IEEE Trans. Circ. & Syst., 1997, pp. 1 - 3

[2] A. Menezes, P. van Oorschot, S.Vanstone, ”Handbook of Applied Cryptography”, CRC Press, 1996.

[3] K. Kawaguchi, H. Kamata, „Secure communication systems based on Chaos synchronization using DSP”, DEC, Meiji University.