Maejo Int. J. Sci. Technol. **2010**, *4*(01), 125-135

Maejo International Journal of Science and Technology

ISSN 1905-7873
Available online at www.mijst.mju.ac.th

Full Paper

Field programmable gate array (FPGA) implementation of novel complex PN-code-generator- based data scrambler and descrambler

Ghulam M. Bhat [1], Muhammad Mustafa [2], Shabir A. Parah [2,*], Javaid Ahmad [2]

[1] University Science Instrumentation Centre (USIC), University of Kashmir, Srinagar-190006, India
[2] Department of Electronics and Instrumentation Technology, University of Kashmir Srinagar-190006, India

* Corresponding author, e-mail: Shabireltr@gmail.com

Received: 4 August 2009 / Accepted: 17 March 2010 / Published: 7 April 2010

Abstract: A novel technique for the generation of complex and lengthy code sequences using low-length linear feedback shift registers (LFSRs) for data scrambling and descrambling is proposed. The scheme has been implemented using VHSIC hardware description language (VHDL) approach which allows the reconfigurability of the proposed system such that the length of the generated sequences can be changed as per the security requirements. In the present design consideration the power consumption and chip area requirements are small and the operating speed is high compared to conventional discrete I.C. design, which is a pre-requisite for any system designer. The design has been synthesised on device EP2S15F484C3 of Straitx II FPGA family, using Quarts Altera version 8.1. The simulation results have been found satisfactory and are in conformity with the theoretical observations.

Keywords: complex code generator, VHDL, FPGA, scrambler, descrambler

Introduction

Since the birth of the cellular industry, security has been a major concern for both service providers and subscribers. Service providers are primarily concerned with preventing fraudulent operations such as cloning or subscription fraud while subscribers are mainly concerned with privacy issues. With the advent of second-generation digital technology platforms like TDMA/CDMA-IS-41, operators were able to enhance their network security by using improved encryption algorithms and other means. The noise-like signature of a CDMA signal over the air interface makes eavesdropping very difficult. This is due to the CDMA "long code," a pseudo-random noise (PN) sequence of long length, which is used to scramble voice and data transmissions. The PN sequence is represented as a sequence of 1's and 0's with certain properties and the sequences are generally categorised into two classes: (1) periodic sequences and (2) aperiodic sequences. The class of sequences used in spread-spectrum communication is usually periodic. There are many types of periodic sequences, some popular ones being (i) maximal-length linear shift register sequences (m-sequences), (ii) quadratic residue sequences (q-r sequences), (iii) Hall sequences, and (iv) twin primes. Among all these sequences, the most commonly used ones are m-sequences.

The PN key generation is of paramount importance for any secure communication system. Maximal sequences are easily generated by linear feedback shift registers (LFSRs). A LFSR [1-2] consists of a shift register and a feedback network (or a parity generator) consisting of only modulo-2 adders (XOR gates). The output of the feedback network is applied to the input of the shift register. The feedback network provides output logic 0 when an even number of input is at logic 1 and generates logic 1 when an odd number of input is at logic 1 state. Maximal-sequence codes generated by using LFSRs are not adequately secure when smaller lengths of LFSRs are employed.

Many circuits using LFSR for generation of complex codes have been reported in the literature [3-5]. Feedback-with-carry shift registers (FCSRs) and schemes using various combinations of LFSRs have been proposed for complex key generation but seem to be expensive for commercial application [6]. The security of the encrypted data is a direct function of the number of stages of shift register used to generate the key. This means, to increase data security, the number of stages of the shift register is to be increased, which leads to an increase in the complexity of the system in terms of power, space and cost [7].

In this paper a relatively low-length shift register for the generation of highly secure m-sequence generator is presented, wherein the feedback tappings keep on changing in a pseudo-random manner, which makes the generated codes quite complex. The complex code generated is used to scramble incoming plain text. At the receiving end, the same code is generated and successfully used to decrypt the transmitted data. The simplicity of the circuit along with the complexity of the generated codes makes the circuit attractive for secure message communication applications. The proposed technique of complex code generation is modelled in VHSIC hardware description language (VHDL), synthesised and simulated for a field programmable gate array (FPGA) target device. This approach allows the reconfigurability [8-10] of the proposed system such that the key complexity can be further enhanced as per the security requirements. Further, this type of implementation offers many advantages over conventional IC design vis-à-vis dynamic power consumption, space occupied and stray capacitances [11].

Description of Proposed Complex-Code-Based Data Scrambler and Descrambler

The proposed scheme is shown in Figure 1. The heart of the system is a complex code generator, as shown in Figure 2, which generates a long and highly complex key stream. The various available PN codes at the output of the complex code generator (Q1 to Qn) are applied as input to the AND gates of the scrambler that act as tap gains. The tap gain outputs are applied to the logical function that produces a complex key to scramble the plain text. The complex key generated is XORed with the plain text to produce cipher text, also known as cryptogram. At the receiving end the same key is generated and used to decrypt the incoming cipher text. The whole system has been modelled using VHDL and hence one can exploit the reconfigurability feature of Description Languages to enhance the system security and optimise the power consumption and speed of operation. The complex code generator used to scramble the incoming plain text uses an 8-bit LFSR which can generate 16 different sets of 255-bit code sequences, depending upon 16 valid sets of feedback tappings, viz. {8,4,3,2}, {8,6,5,4}, {8,6,5,3}, {8,5,3,2}, {8,6,5,2}, {8,6,3,2}, {8,5,3,1}, {8,7,4,3}, {8,6,5,1}, {8,7,3,2}, {8,7,6,1}, {8,7,2,1}, {8,7,6,5,2,1}, {8,7,6,3,2,1}, {8,6,4,3,2,1} or {8,7,6,5,3,2}. Any one of these sets of feedback tappings can be used at a time so that a particular combination of the output of the LFSR is connected back to its input through a modulo-2 adder. Thus, any one of the above sets of feedback connections can be selected at a time to generate the corresponding code-sequence, in part or in full, depending on the time for which the selected feedback remains connected. If these feedback connections are changed synchronously in a random manner, the output sequence (Q's in Figure 2) also changes correspondingly. For simplicity of demonstration of the scheme, only seven sets of feedback connections, viz. {8,4,3,2}, {8,6,5,4}, {8,6,5,3}, {8,5,3,2}, {8,6,5,2}, {8,6,3,2} and {8,6,5,1}, are chosen here. These sets of feed-back connections are obtained by XORing various output streams of the LFSR. One set of these connections can be selected at a time with the help of an 8-to-1 line multiplexer (MUX) controlled by a 3-bit word generated by another PN sequence as shown in Figure 2. Since '000' state is to be avoided as the control word of the MUX (because the PN sequence generates the required control word), only seven input signals of the MUX (one at a time) will be chosen depending upon the control word. Thus, feedback tappings are changed randomly, selecting one at a time out of the set of seven different sets. Hence, depending upon the value of N (the divide-by factor in the circuit), together with the initial state of the code generator and the initial state of the MUX, a complex code sequence is generated. The complex codes generated (Q1 to Q8) are used as input to AND gates to generate the tap gains. These tap gains are given to the logic function generator which manipulates the data and produces the complex key, the key being modulo-2-added with plain text and thus resulting in a cryptogram. The cryptogram is applied to the shifting logic so as to increase the complexity of the key and hence the security of data. At the receiving end, the same key is generated and used for the successful decryption of received data.

Simulation Results and Verification

The proposed scheme has been modelled in VHDL, synthesised and simulated for target device EP2S15F484C3 of Straitx II FPGA family using Quarts Altera version 8.1. It is important to mention that Stratix II devices can be used for implementing memory functions and complex logic functions such as digital signal processing, wide data-path manipulation, data transformation, and microcontrollers. The high-pin-count Stratix II devices contain a two-dimensional row- and column-based architecture to implement custom logic. The devices provide adaptive look-up tables (ALUTs), memory bits, and adaptive logic modules (ALMs). They also include input and output low-voltage differential signalling (LVDS) channels and provide dedicated circuitry to support differential I/O standards at up to 1 Gbps when using dynamic phase alignment (DPA) and 840 Mbps when not using

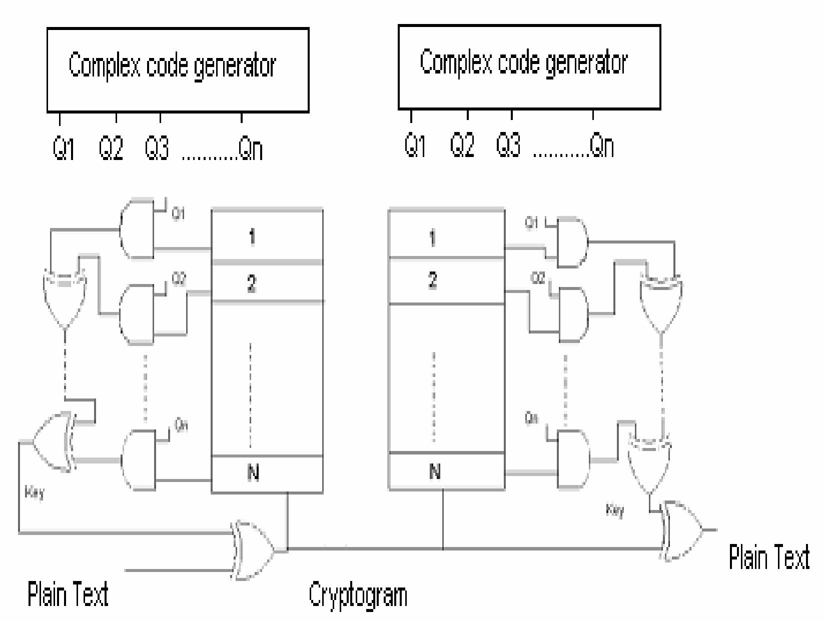


Figure 1. Complex-code-generator-based data scrambler and descrambler

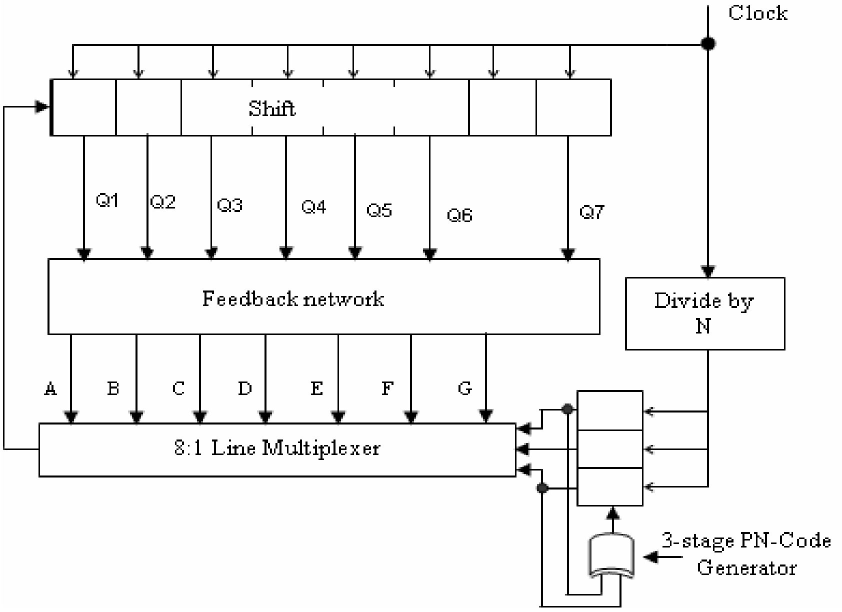


Figure 2. Complex code generator

DPA. Stratix II devices provide enhanced phase-locked loops (PLLs), fast PLLs, and regional clock networks to increase performance, and also provide advanced clock interfacing and clock-frequency synthesis. The devices also contain 16 dedicated clock pins for controlling signals with large fan-outs. In addition, all Stratix II devices include enhanced and fast (PLL) circuitry.

The wave forms obtained at various check nodes (output terminals at which output codes are available) of the complex code generator corresponding to all the 255 × 7 sequence combinations have been investigated and found in conformity with the theoretical observations. As shown in Figure 3, 'Clk' is an input signal which drives the PN sequence generator. PN (7) to PN (0) are the resultant output PN sequences. For seed value (initial value with which LFSR is loaded) of '11111111' and feedback tapping (8, 4, 3, 2), theoretically, the LFSR output sequence (with PN(7) as MSB and PN(0) as LSB) passes through the following states with subsequent clock cycles: '01111111', '00111111', '10011111', '01001111', '00100111', '000100111', '000010011', '10000100', and so on. (The pattern generated is a function of feedback tapping. Hence, the next pattern depends on the previous one and the feedback tapping.) The wave form obtained for LFSR seed value '11111111' and selected tap position (8, 4, 3, 2) for a few clock cycles is presented in Figure 3, which confirms that the results obtained are in agreement with the theoretical values (shown later.)

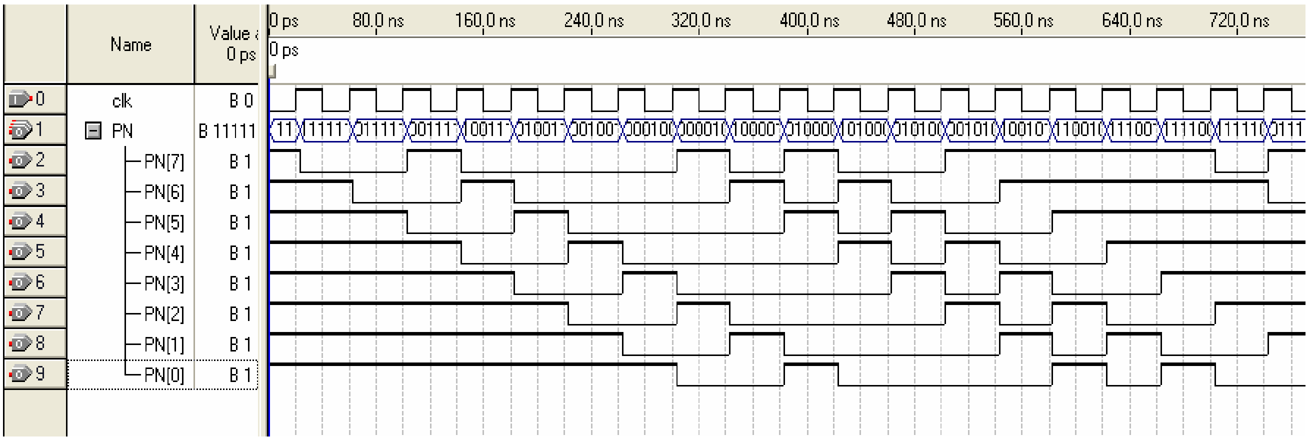


Figure 3. Wave forms obtained at various check points of complex code generator

As shown in Figure 3, prior to application of clock pulse, the value of output codes from PN(7) to PN(0) is '11111111'. With the arrival of positive-going edge of first clock pulse the output changes to '01111111'. The arrival of subsequent clock pulses then keeps on changing the output to '00111111', '10011111', '01001111', '00100111', '000100111', '000010011', '10000100' and so on in that sequence, which is in complete agreement with the theoretical results. The complex codes obtained from the complex code generator as shown in Figure 3 are used to scramble incoming data stream at the transmitter, as shown in Figure 4. At the receiving end, the same codes are generated for successful decryption of data.

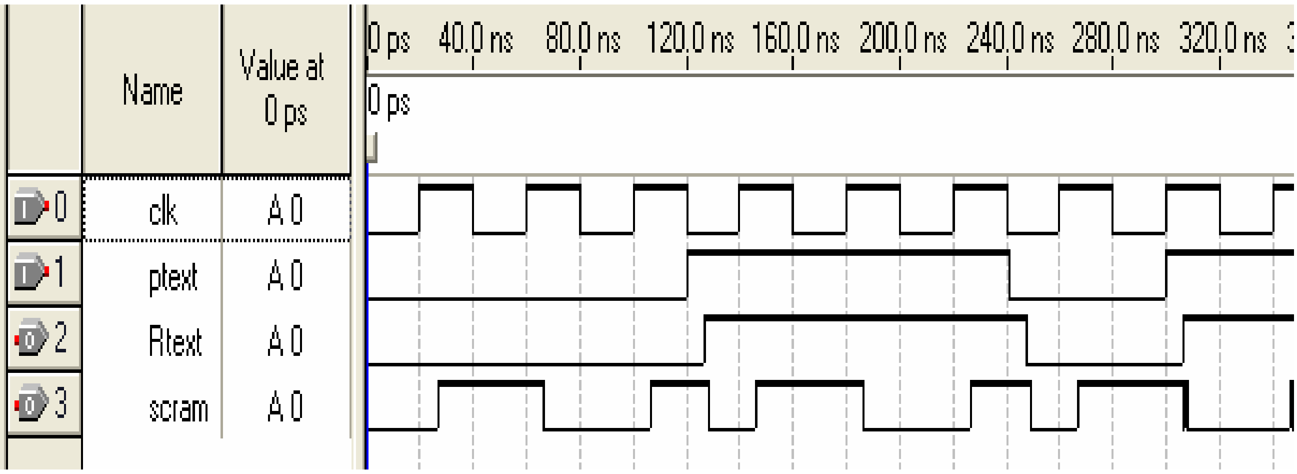


Figure 4. Wave forms obtained at various check points of data scrambler and descrambler

The simulation results are shown in Figure 4. 'Clk' represents the input signal clock used to drive the system, 'Ptext' is the input plain text at the transmitting end, and 'Rtext' represents the received data after decryption at the receiver. 'Scram' represents the scrambled signal corresponding to the input data. As shown in the figure, transmitted data and the received version of it are in complete agreement. However, the wave forms of the transmitted data and its scrambled version do not resemble at all.

The RTL viewers of the implemented designs, viz. scrambler and descrambler, are shown in Figures 5 and 6 respectively. Figure 7 shows RTL viewer of the implemented complex code generator. Finally, a comparison of the codes generated by the proposed scheme with those generated conventionally [3] is presented in Table 1. The comparison is depicted graphically in Figure 8. It is clear that the codes generated using the same length of LFSR are much more complex (more lengthy) than those generated using the conventional PN code generator.

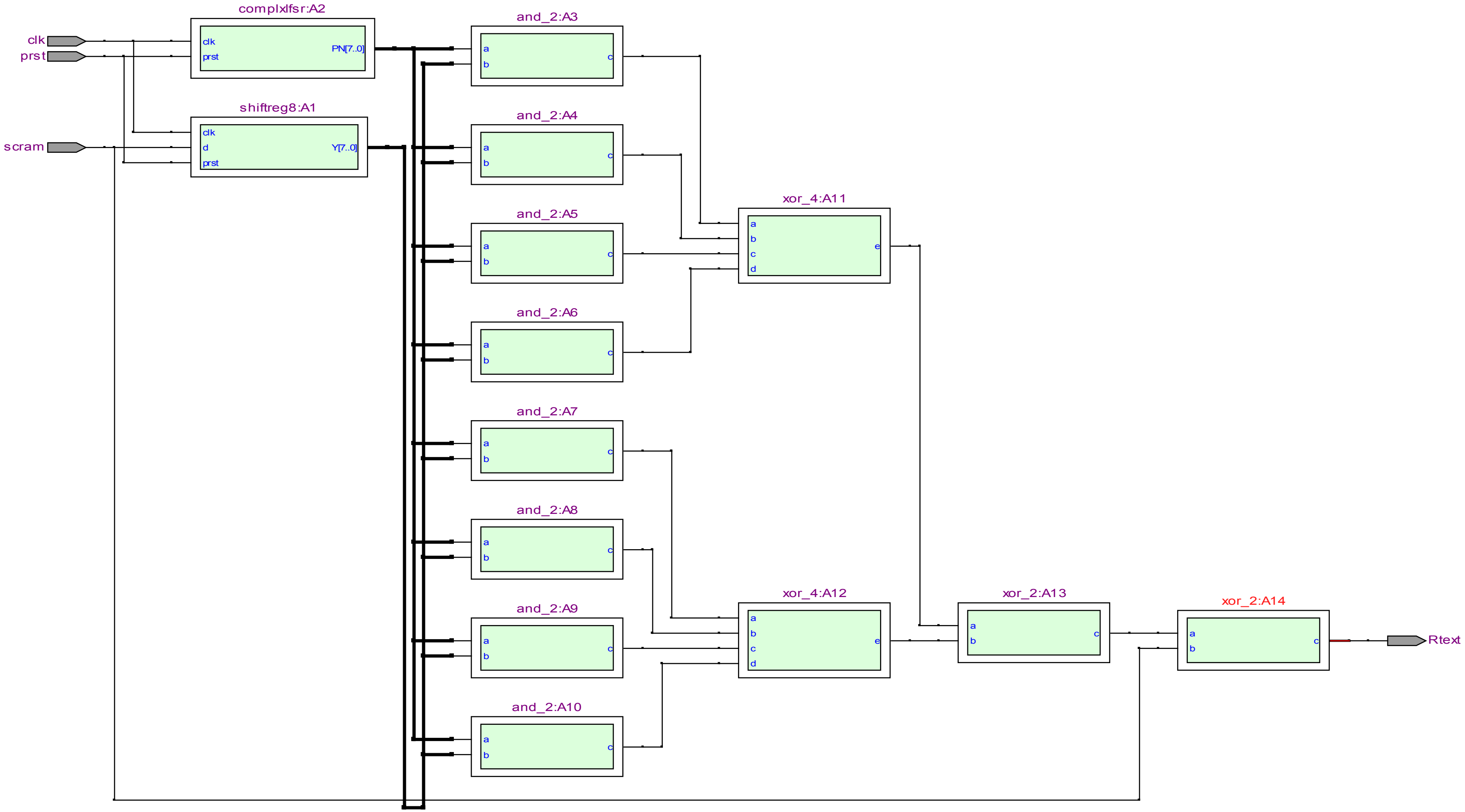


Figure 5. RTL viewer of complex-code-based data scrambler

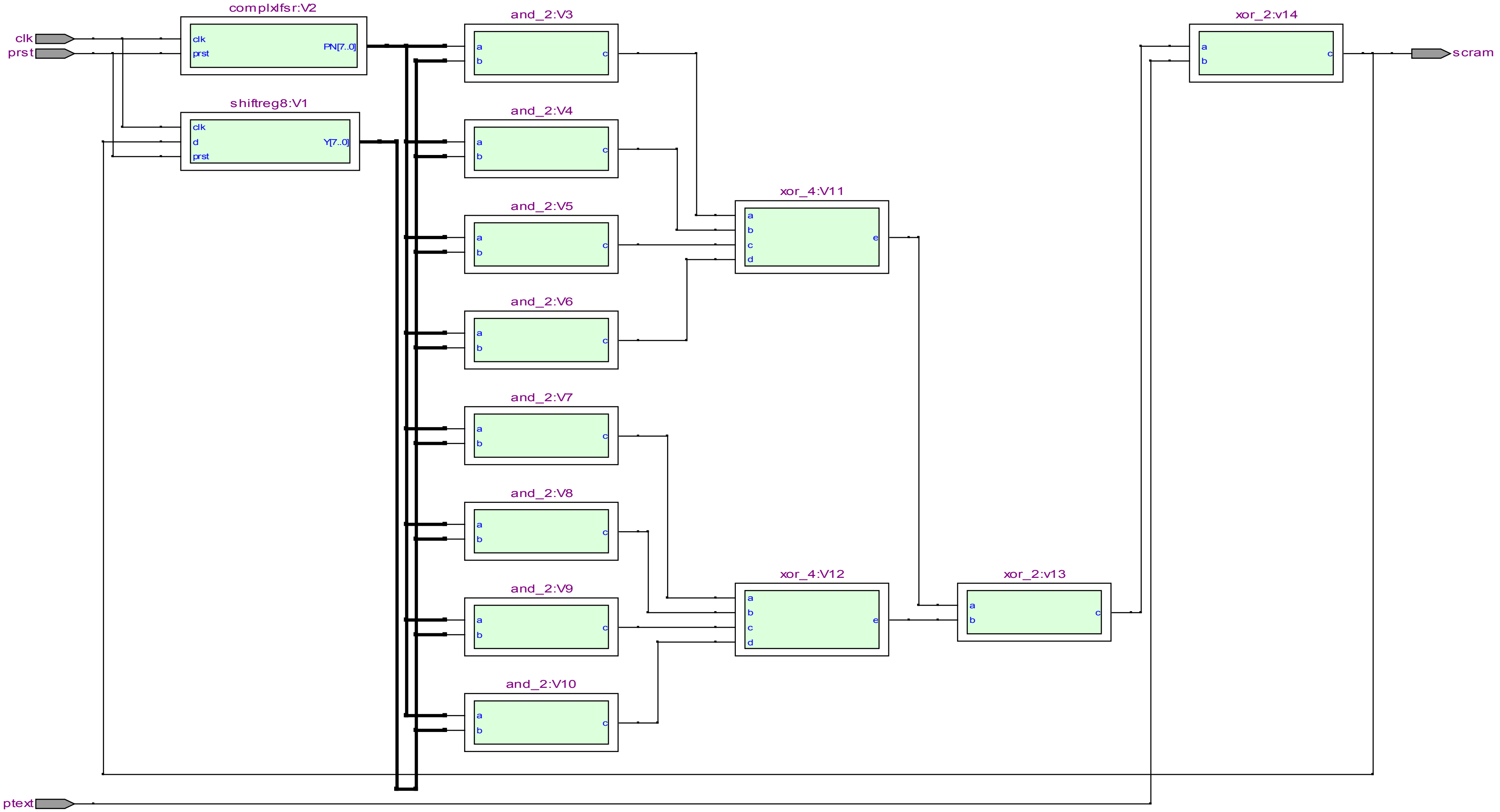


Figure 6. RTL viewer of complex-code-based data descrambler

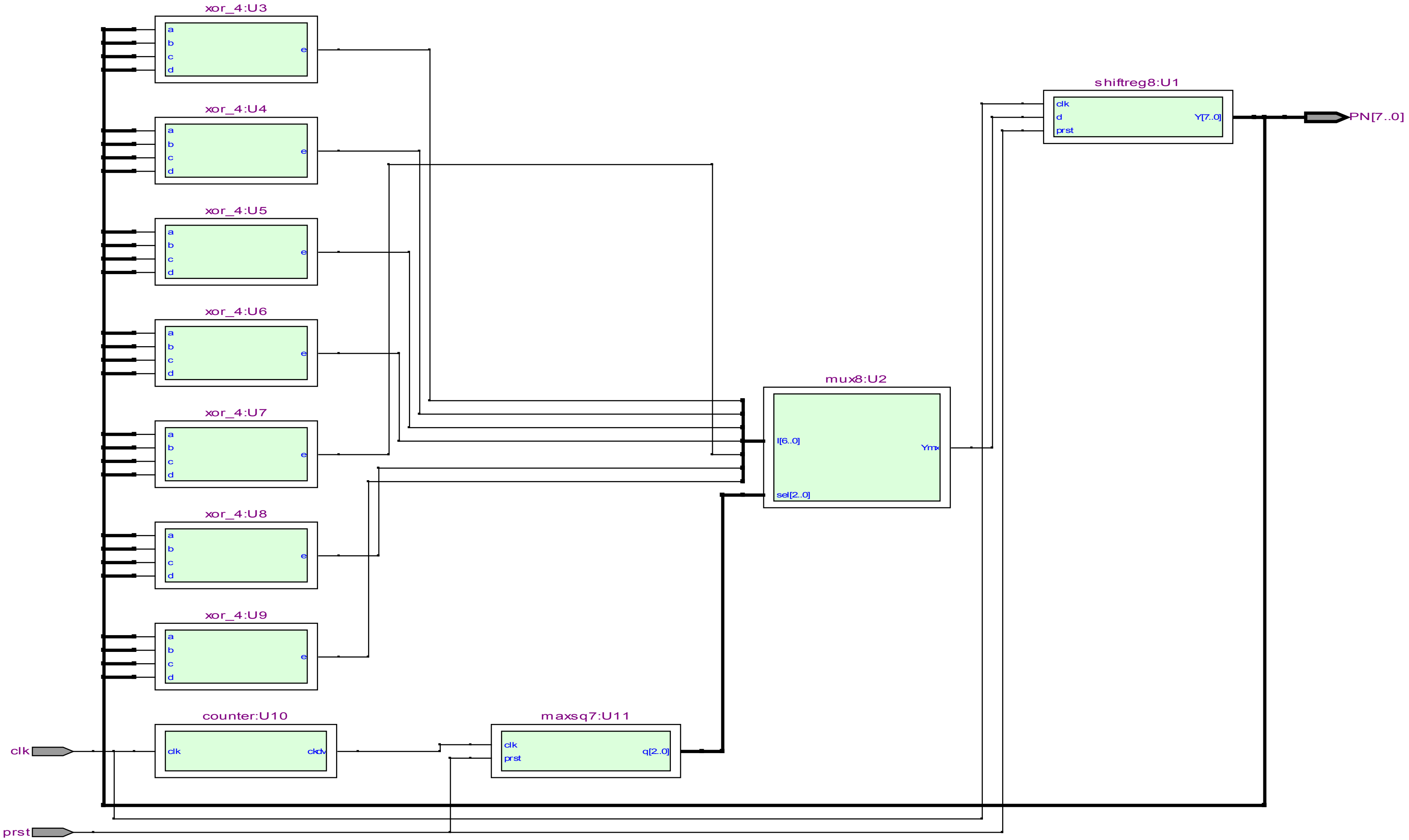
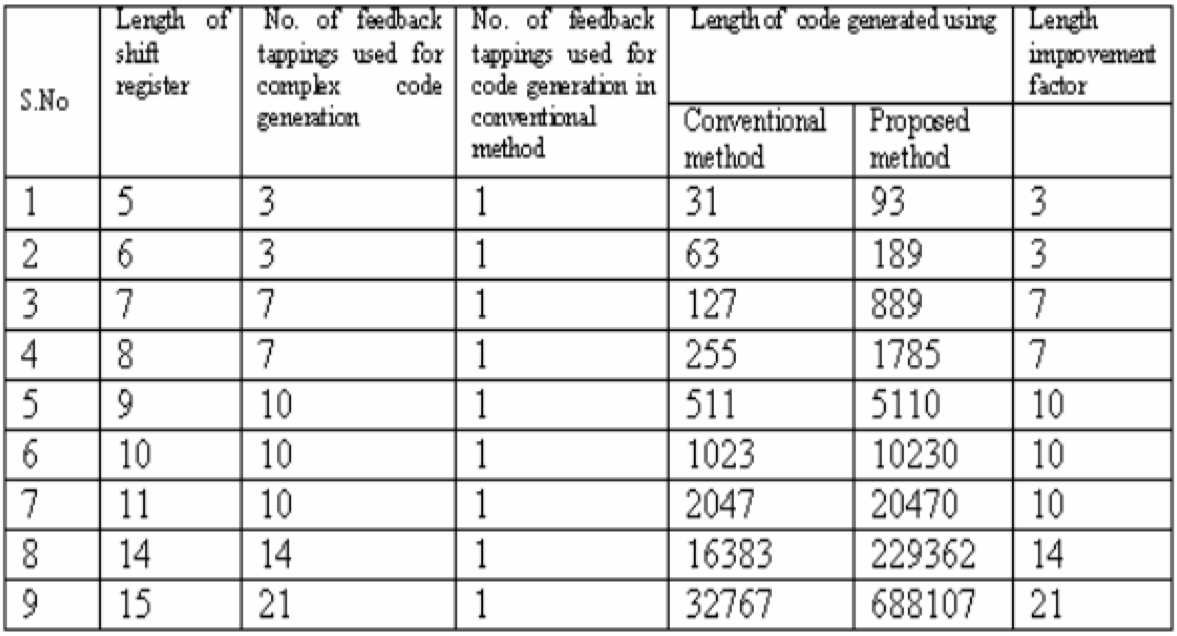


Figure 7. RTL viewer of complex code generator

Table 1. Resource utilisation summary



S.No	Length of shift register	No. of feedback tappings used for complex code generation	No. of feedback tappings used for code generation in conventional method	Length of code generated using		Length improvement factor
				Conventional method	Proposed method	
1	5	3	1	31	93	3
2	6	3	1	63	189	3
3	7	7	1	127	889	7
4	8	7	1	255	1785	7
5	9	10	1	511	5110	10
6	10	10	1	1023	10230	10
7	11	10	1	2047	20470	10
8	14	14	1	16383	229362	14
9	15	21	1	32767	688107	21

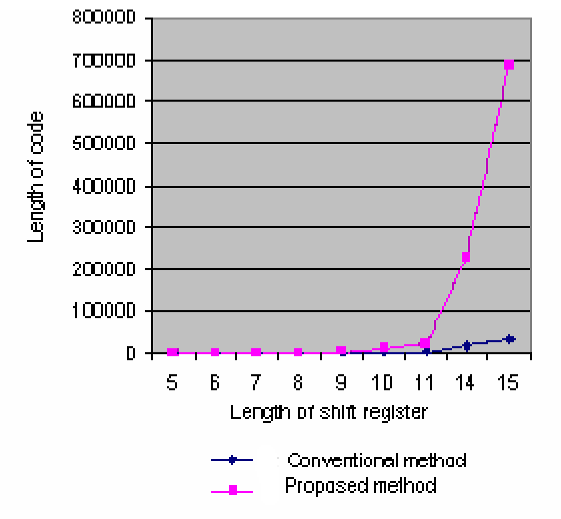


Figure 8. Code length comparison

Conclusions

A new modified scheme for complex PN-code-based data scrambler and descrambler has been presented. The proposed scheme uses a complex code generator, capable of generating lengthy codes using a LFSR with relatively less number of stages, for data encryption and decryption. The proposed scheme has been modelled in VHDL synthesised and simulated for target device EP2S15F484C3 of Straitx II FPGA family using Quarts Altera version 8.1 However, for defense and aerospace applications where design certainly carries special technical challenges, the choice for the target device can be Xilinx Virtex-5Q family of FPGAs. This provides 65-nm, high-density and high- performance FPGA technology suitable for secure communication systems. The proposed scheme is capable of providing a range of applications in Spread Spectrum Modulation, Code Division Multiple Access and Global Positioning Systems. The scheme can be synthesised and implemented on any of the existing CPLD and FPGA systems as per the degree of optimisation required.

References

1. B. Schneier, "Applied Cryptography", 2nd Edn., John Wiley & Sons Inc., New York, **1996**.
2. P. Pedron, "Linear feedback shift registers", *US Patent 5105376* (**1992**).
3. W. Ahmad and G. M. Bhat, "Scrambler for data", *Electron. World*, **1997**, *3*, 227-228.
4. M. Izharudin, O. Farooq and A. Waseem, "FPGA-based Implementation of Stream Ciphering Circuit", Proceedings of International Conference on ROVSIP, **2005**, Penang, Malaysia, pp. 16-19
5. R.C. Dixon, "Spread-Spectrum Systems with Commercial Applications", John Wiley & Sons Inc., New York, **1994**.
6. A. Wasim, "Development of low-cost secure communication techniques", AICTE (R&D) Project (**2002**), Department of Electronics Engineering, Aligarh Muslim University, Aligarh, India.
7. K. Jarvinen, M. Tommiska and J. Skytta,"Comparative survey of high-performance cryptographic algorithm implementations on FPGAs", *IEE Proc. Inform. Secur.*, **2005**, *152*, 3-12.
8. F. X. Standaert, G. Piret, G. Rouvroy and J. J. Quisquater, "FPGA implementations of the ICEBERG block cipher", *Proc. ITCC*, **2005**, *1*, 556-561.
9. K. Gaj and P. Chodowiec, "Fast implementation and fair comparison of the final candidates for advanced encryption standard using field programmable gate arrays", in "Topics in Cryptology - CT-RSA 2001" (Ed. D. Naccache), Springer, San Francisco, **2001**, pp. 84-99.
10. R. Mita, G. Palumbo and M. Poli, "Pseudo random sequence generators with improved inviobility performance", *IEE Proc. Circuits Devices Syst.*, **2006**, *153*, 375-382.
11. M. Feldhofer, J. Wolkerstorfer and V. Rijmen, "AES implementation on a grain of sand", *IEE Proc. Inf. Secur.*, **2005**, *152*, 13-20.