Design and Implementation of SOC in NIOS-II Soft Core Processor for Secured Wireless Communication

R.Ramachandran
Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India.

J. Thomas Joseph Prakash
H.H. The Rajah's College
Pudukkottai – 622001, Tamilnadu, India

ABSTRACT

In the modern world, the information that could benefit the individual / a group also can be used against such individual or group. Encryption is the technique of converting a plaintext (original data packet) into cipher text (encrypted message) which can be decoded back into the original message. There are several types of data encryptions schemes available which form the basis of network security. Our proposed work deals with the security based wireless communication system, with the NIOS-II soft core processor. Security in wireless communication is most essential, especially where hacking and tampering are threats of the data packet. Hence introduction of suitable security bits (Key) with the actual data packet is most important aspect in wireless communication to avoid such threats. It is really a significant and interesting area for the researcher. In this work, the data encryption standard technique is used for the safety transmission and reception, and implemented it successfully with the NIOS-II soft core processor.

Keywords

Wireless communication, NIOS-II soft core processor, FPGA, SOC, Data Encryption.

1. INTRODUCTION

In recent years, the wireless communication plays a vital role in industries, home and government departments [1]. It is the emerging field in electronic communication. One example of its growth is cellular systems. Cellular systems have experienced high growth over the last decade. Wireless networks currently replace wired networks in many homes, companies and other campuses. Many new applications, including smart homes, wireless sensor networks, automated factories and high ways, and remote telemedicine, are emerging from research ideas into concrete systems [2]. Although the wireless communication is having numerous applications, hacking and tampering the data packet are also quite common. There are plenty of techniques of hacking the data; especially in wireless communication is underway. Such hacking methods [3] are; 1. Diverse hacker attack methods, 2. Social Engineering, 3. Social spying, 4. Sniffing and more. In order to avoid hackings and tampering the data packets a suitable encryption technique has to be implemented. There are plenty of methods can be used to encrypt data packets, all of which can easily be implemented through software, but not so easily decrypted when either the original or its encrypted data stream are unavailable. (When both source and encrypted data are available, code-breaking becomes much simpler, though it is not necessarily easy). The best encryption methods have little effect on system performance (Slowing the system), and may contain other benefits such as data compression [4]. We have used the data encryption technique to implement the secured data transmission and reception with NIOS-II soft core processor [5]. It is having many features such as RISC (Reduced Instruction Set Computer) architecture, 32 bit processor, and its architecture consist of separate instruction and data bus (Harvard architecture). The SOPC (System on programmable chip) builder is a tool used in conjunction with the Quartus II CAD tool Software [6]. It allows the designer to easily create a system based on the NIOS II processor, by selecting the desired functional units and specifying their parameters. The NIOS II processor is a configurable soft core processor. Configurable means that can be added or removed any features or resources of the processor on the basis of performance or cost wise. Soft core means the processor core is not fixed in silicon chip, but can be targeted to any Altera FPGA family.

This paper is organized as 1. Overview, 2. Implementation, 3. Result and analysis, and 4. Conclusion.

2. OVERVIEW

Our approach to achieve the aim of the proposed work is based on Altera NIOS II embedded soft core processor that provides a highly configurable device and having excellent versatility. We have used Altera Cyclone Field Programmable Gate Array (FPGA) and its peripherals to make the design to be constructive. A well known Data Encryption Standard (DES) Algorithm[7] is chosen for Encryption for secured data transmission. The evaluation results obtained from the experiment are shown for analysis and verification of our design. The figure (1) shows block diagram of the proposed work. Referring the block diagram, the functional modules are 1. Various Wireless sensor outputs, usually analog signals (The Electrical output), 2. A/D converter, 3. Cyclone II FPGA, 4. GSM modem.

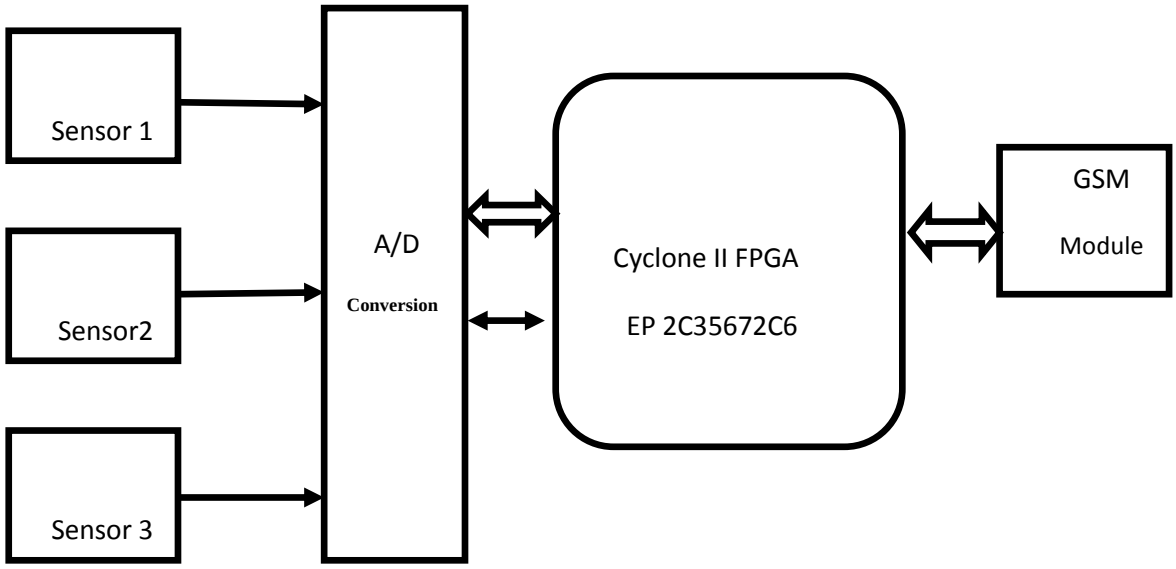

Figure (1) – Block diagram of the proposed work

2.1 Analog to Digital Converter

Analog to Digital converter [8] is configured with processor is 8- bit converter, compatible with ADC 0809. The 8- channel (Inputs IN0 to IN7) multiplexer can directly access any of 8- single ended analog signals to be transmitted with the set of address lines A,B,C. The clock chosen for conversion 600Khz.. The ADC 809 also having many desirable features, such as no zero or full scale adjust required, and 0V to 5V input range with single 5V power supply.

2.2 NIOS Embedded processor

The NIOS II Embedded processor is a general purpose configurable Soft core processor [9,10]. In practice, most FPGA designs implement some extra logic in addition to the processor system. The processor could be virtually realized on any of an Altera FPGA by using a System On a Programmable Chip (SOPC) builder. NIOS II embedded processor is having, Custom instructions; means Nios II instructions, user-defined instructions accept values from up to two 32-bit source registers and optionally write back a result to a 32-bit destination register. Nios II families are 1. NIOS II/f (fast), 2. NIOS II/s (standard), and 3. NIOS II/e (economy). Any one of these can be used for design, depends on the requirement.

2.3 DE2 development board

DE2 development board consists of Cyclone II EP2C35F672C6 with EPCS16 16-Mbit serial configuration device, and many standard peripherals. It provides a high performance, low cost design with modern Altera devices and tools. With Quartus II CAD tool, the designer can develop hardware design using HDLs in the on board FPGA. A set of memory such as 8 MB SDRAM, 512 KB SRAM are also available. Hence, DE2 development board is much suitable platform for running NIOS II soft core processor. The figure (2) shows Altera's DE2 development board.

2.4 RS-232 Serial Interface

An RS-232 interface [12] has many characteristics, 2. It allows bidirectional full-duplex communication, 3. It can communicate at a maximum speed, at the rate of 10KBytes/s. Three are important pins for communication, they are RxD (Receive Data- Pin No. 2), TxD (Transmit Data- Pin No.3) and GND (Ground pin No. 5).

2.4.1 Serial communication

Data is sent on bit by bit basis, that is one bit at a time; a single wire is used for each direction. Since computers usually need at least several bits of data, the data is "serialized" before being sent. Data is commonly sent by chunks of 8 bits. The LSB (data bit 0) is sent first, the MSB (bit 7) last.

2.4.2 *Asynchronous communication*

This interface uses an "asynchronous" protocol. In asynchronous data transfer, handshake signals are used for testing the readiness of the receiver. In our case the RS232 is

function in the mode. In this mode of transmission, the speed and format has to decided before start the transmission. The TxD line sends logic "1" as long as the line is idle. The start bit (logic "0") to be send before each byte of transmission. After the "start", data comes in the agreed speed and format, so the receiver can interpret it. The stop bit is usually logic "0".

The common baud rates of RS 232 Serial interface are 1200,9600,38400, and 115200. The speed can be easily calculated as, for example if the baud rate is 115200. T = 1/115200 = 8.7µs. If 8-bits data to be transmitted, that lasts 8 x 8.7µs = 69µs. But in the Asynchronous format of transmission, each byte requires an extra start and stop bit, so actually need of 10 x 8.7µs = 87µs. That translates to a maximum speed of 11.5KBytes per second.

2.5 SIM 300 GSM modem

The figure (3) shows SIM 300 GSM modem. A GSM modem exposes an interface that allows applications such as NowSMS to send and receive messages over the modem interface.

Figure (3) - SIM 300 GSM modem

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. Every command line starts with "AT" or "at". That's why modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems, such as ATD (Dial), ATA (Answer), ATH (Hook control) and ATO (Return to online data state), are also supported by GSM/GPRS modems and mobile phones.

2.6 Data Encryption Standard Algorithm

Data Encryption Standard Algorithm is chosen to achieve a good secured data transmission [14,15]. The implementation of this algorithm has been done with C++ language. A DES key consists of 64 binary digits of which 56 bits are randomly generated and used directly by the algorithm. In this technique, data can be recovered from cipher (Encrypted output) only by using exactly the same key used to encipher it. Unauthorized recipients of the cipher who know the algorithm but do not have the correct key cannot derive the original data algorithmically. It is also possible to retrieve the original data by deciphering the text with original key and algorithm. A standard algorithm based on a secure key thus provides a basis for exchanging encrypted computer data by issuing the key used to encipher it those authorized to have the data. The security level has been proven to be competent for efficiently defending the linear and differential cryptanalysis. The characteristics of this algorithm make it appealing security scheme for our proposed work.

3. IMPLEMENTATION

For transferring secured data transmission through wireless, we have designed NIOS II processor with necessary peripheral interface. The entire implementation process is thus assigned to two stages; 1. Hardware implementation, and 2. Software Implementation.

3.1 *Hardware Implementation*

In Our design, we have effectively utilized the resources available in the Altera NIOS II embedded processor technology. In the laboratory we have taken the analog data which is then successfully converted to equivalent binary bits. These are the output bits of the ADC, which is 64- bit known as plain text. With these binary bits (Plain text), C++ program was executed in the NIOS II processor on DE2 development board to get the cipher text for secured data transmission, and then the cipher text was transmitted serially with RS 232 serial interface to the SIM 300, for transmitting the secured data (Cipher text) over the space. At another end the cipher text was received successfully with another SIM 300, and then the cipher text was deciphered. Since this process of communication is purely asynchronous, the excellent configurable and programmable properties of the NIOS II processor enable us to simply implement the encryption and decryption process on a single DE2 development board significantly reducing the cost of the platform. The core of the DE2 development board consists of Cyclone II EP2C35F672C6 with EPCS16 16- Mbit serial configuration device. The basic peripherals chosen are UART controller, A/D controller, On chip memory, I/O controller, and a timer in SOPC builder for the lowest Logic Elements (Les) consumption. The figure (4) Shows how the NIOS II soft core processor with the necessary peripherals are implemented in FPGA. The completed SOPC design is then downloaded via JTAG UART to DE2 Board.

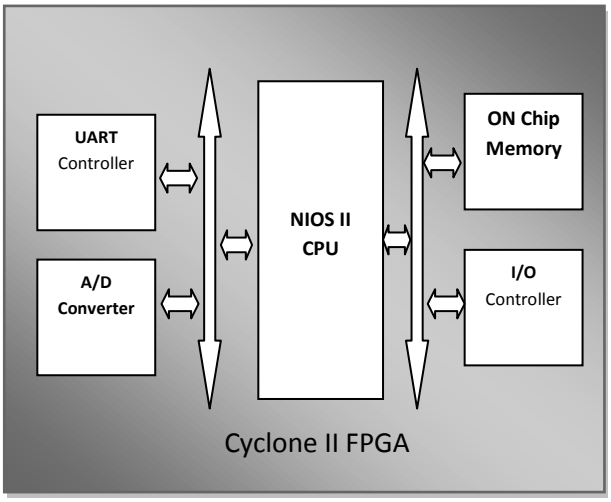

Figure (4) - NIOS II CPU implementation on FPGA

3.2 Software Implementation

Having built up the hardware, the Data Encryption Standard (DES) algorithm needs to be programmed in software. The software development environment of NIOS II processor that called NIOS II Integrated Development Environment (IDE) is based on the C/ C++ compiler. The program was written in C++, and the specifications of DES carefully coded with suitable parameters.

The algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 64- bit key (the left most bit of a block is bit number one). Deciphering must be accomplished by using the same key as for enciphering, but with the schedule of addressing the key bits altered so that the deciphering process is the reverse of the enciphering process. A block to be enciphered is subjected to initial permutation **IP**, then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation $\mathbf{IP^{-1}}$. The key-dependent computation can be simply defined in terms of a function ***f,*** called the cipher function, and a function **KS,** called the key schedule. The figure (5) shows the illustration details of the DES.

DES algorithm

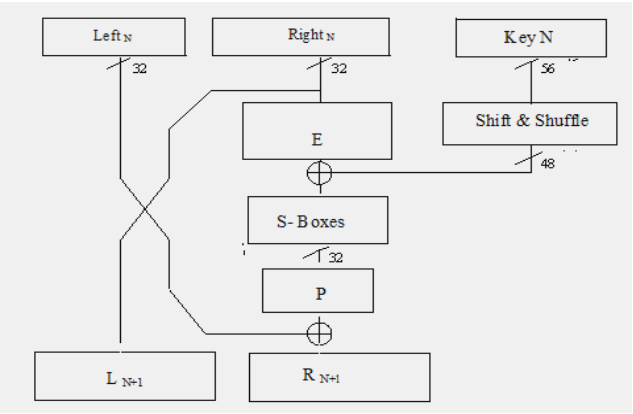

Figure (5) – DES illustration

Input:

T: 64 bits of clear text
k1, k2, ..., k16: 16 round keys
IP: Initial permutation
FP: Final permutation
f(): Round function

Output:

C: 64 bits of cipher text

Algorithm:

T' = IP(T), applying initial permutation

(L0, R0) = T', dividing T' into two 32-bit parts

(L1, R1) = (R0, L0 ^ f(R0, k1))

(L2, R2) = (R1, L1 ^ f(R1, k2))

.....

C' = (R16, L16), swapping the two parts

C = FP(C'), applying final permutation

where ^ is the XOR operation.

1. The round function f(R,k) is defined as:

Input:

R: 32-bit input data
k: 48-bit round key
E: Expansion permutation
P: Round permutation
s(): S boxes function

Output

R' = f(R,k): 32-bit output data

Algorithm

X = E(R), applying expansion permutation and returning 48-bit data
X' = X ^ k, XOR with the round key
X'' = s(X'), applying S boxes function and returning 32-bit data
R' = P(X''), applying the round permutation

2. The S boxes function s(X) is defined as:

Input:

X: 48-bit input data
S1, S2, ..., S8: 8 S boxes - 4 x 16 tables

Output:
X' = s(X): 32-bit output data

Algorithm:

(X1, X2, ..., X8) = X, dividing X into 8 6-bit parts
X' = (S1(X1), S2(X2), ..., S8(X8))
where Si(Xi) is the value at row r and column c of S box i with
r = 2*b1 + b6
c = 8*b2 + 4*b3 + 2*b3 + b4
b1, b2, b3, b4, b5, b6 are the 6 bits of the Xi

4. RESULT AND ANALYSIS

In this work, the designed System on chip (SOC) on NIOS- II processor successfully implemented. The resultant hardware configuration implemented, as shown in the figure (6) as follows.

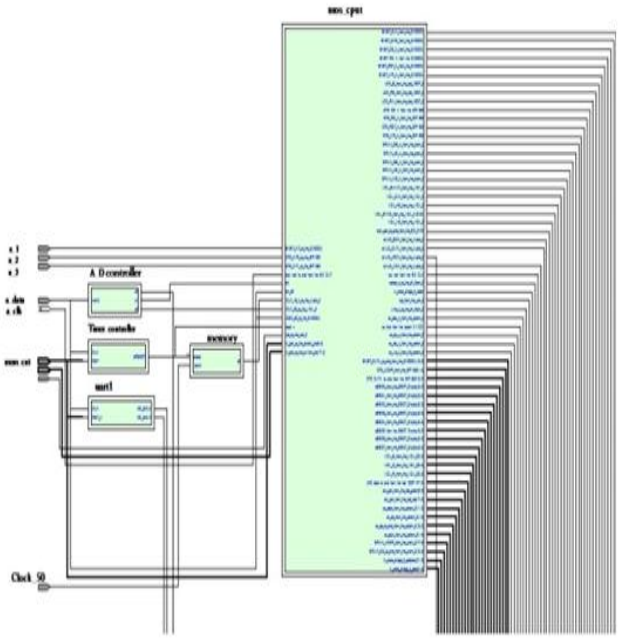

Figure (6)- RTL View of the NIOS II processor implementation

Test conducted on the prototype work by giving the analog data (Thermistor output) and transmitted successfully with encryption through GSM modem SIM 300. Also we have received appropriate message as output. The NIOS II/e (economy core) is designed to minimize the cost of the design. In this work, the total logic elements consumed only 20%, total logic functions are used only 17%. Other parameters are used in the work as shown in the table (1).

Family	**Cyclone II**
Device	EP2C35F672C6
Total Logic elements	6,574 / 33,216 (20%)
Total combinational functions	5,557 / 33,216 (17%)
Dedicated Logic registers	3,524 /33,216 (11%)
Total registers	3641
Total pins	429 /475 (90%)
Total virtual pins	0
Total memory bits	387,584/483,840 (80%)
Embedded multiplier 9-bits	4/70 (6%)
Total PLLs	2/4 (50%)

Table (1) - consumption details of NIOS II core processor

5. CONCLUSION

Based on our proposed work hardware implementation platform, we suggest a novel methodology for transmitting wireless data. This method takes into account the level of security, execution speed, memory usage and power consumption altogether, all of which are desired properties for wireless data transmission. With the comprehensive consideration of all these factors for certain application in wireless communication could be found with high efficiency and low cost. This paper has proposed a comprehensive approach of secured data transmission in wireless communication, and the feasibility of the design has been studied. Certain questions are to be answered in terms of speed in the future study.

REFERENCES

[1] David Tse, Pramod Viswanath, Fundamentals of Wireless Communication 2005

[2] Andrea Goldsmith, Wireless communication

[3] Cyrus Peikari, Seth Fogie, Wireless Maximum Security, Chapter 6, Sams publishing, 2003.

[4] H.Chen, A.Perrig." Security and privacy in Sensor Networks" Computer, Vol. 36, Issue 10, Oct. 2003, pp 103-105.

[5] Nios II processor Reference handbook, http://www.altera.com/

[6] http://users.ece.gatech.edu/~hamblen/DE1/DE1_CDROM/DE1_tutorials/tut_sopc_introduction_verilogDE2.pdf.

[7] Federal Information Processing Standards Publication 46-3, 1999 October 25.

[8] Data sheet of National Semiconductor, www.national.com

[9] Altera NIOS II Software Developer's Handbook, http://www.altera.com/

[10] Altera Quartus II Handbook, http://www.altera.com/

[11] DE2 Development and Education Board, http://www.altera.com/

[12] RS 232 Serial interface, http://www.fpga4fun.com/SerialInterface1.html

[13] http://en.wikipedia.org/wiki/Subscriber_Identity_Module .

[14] Alfred Menezes, Paul van Oorschot , Scott Vanstone, Handbook of Applied Cryptography.

[15] Jonathan Katz , Yehuda Lindell**,** Introduction to Modern Cryptography: Principles and Protocols (Chapman & Hall/CRC Cryptography and Network Security Series).