Middle-East Journal of Scientific Research 20 (1): 43-46, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.20.01.114104

Wireless Security System Based on FPGA

Surumbar Khuzhali

M. Tech-VLSI Design,
Bharath University, Chennai, India

Abstract: With the wireless communications technology in data security development, it increases the requirements for security and confidentiality, using the security management strategy and security technology faces new challenges. FPGA-based programmable devices provide secure communications for efficient processing. However, if the secure communication system will be fully integrated into the FPGA, the system overhead and the chip cost will increase substantially, with the upgrade problems. This issue presents a technique relying on encryption, using the wireless channel for confidential data transmission method. The system uses AES-128 as the encryption algorithm and uses the BCH algorithm for data protection. Hardware platform using the Xilinx Spartan-3AN FPGA family and 2.4GHz RF modules for the coprocessor.

Key words: Security management strategy · Ully integrated into the FPGA · Hardware platform

INTRODUCTION

The system implemented based on FPGA and 2.4GHz transceiver technology for wireless encrypted communications terminal. System characteristics are as follows:

- Data transmitted via the 2.4GHz band;
- AES-128 encryption algorithm for data processing;
- To prevent data errors, the system uses the BCH (31,16, 3) algorithm for error checking and correction.

Both of the communicating parties can initiate data transfer. If there is no data transfer request, the host will automatically enter sleep mode to reduce power consumption. In data security, in order to maximize the system's security protection, the data in system is not only encrypted but the corrupted data can be error correction. The data are self-correcting system feature that can be within a certain range of data repair (BCH (31, 16, 3)). If it is found that the value of the transmitted data parity error, error correction can be carried out locally, but beyond the correction range,

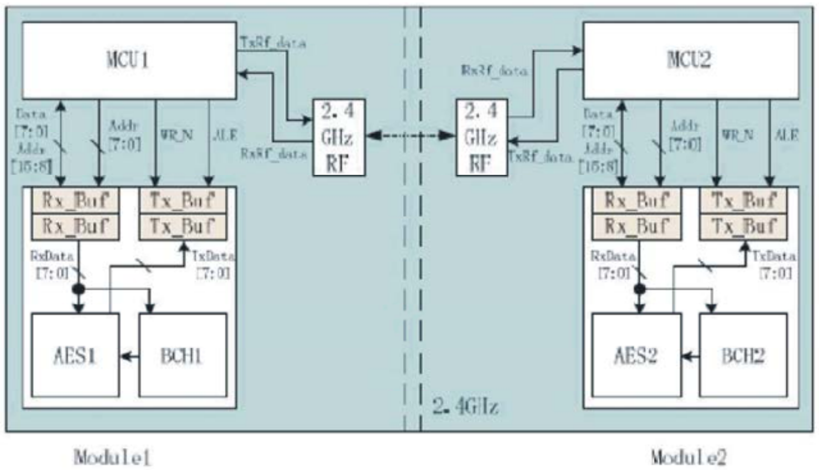


Fig. 1: Encryption Terminal Diagram

requires the sender to retransmit the packet. With the opening of wireless transmission channel, the system is for full data encryption. The function involved in the encryption algorithm is entirely implemented in hardware, so encryption and decryption to ensure maximum speed and reliability. AES key stored, under the control of the MCU, inside the FPGA [1].

FPGA Internal Memory Mapping: 2.4G-band communication module is used; the band has its own unique advantages. Compared with Bluetooth products, the manufacture of it is lower cost and providing higher data rates.

The Signal Functions as Follows:

CE: Digital input, chip enable activates RX or TX mode;

CSN: Digital input, SPI chip select; SCK : digital input, SPI clock;

MOSI: Digital input, SPI slave data input; MISO : digital output, SPI data output, with tri-state option;

IRQ: Digital output, maskable interrupt pin

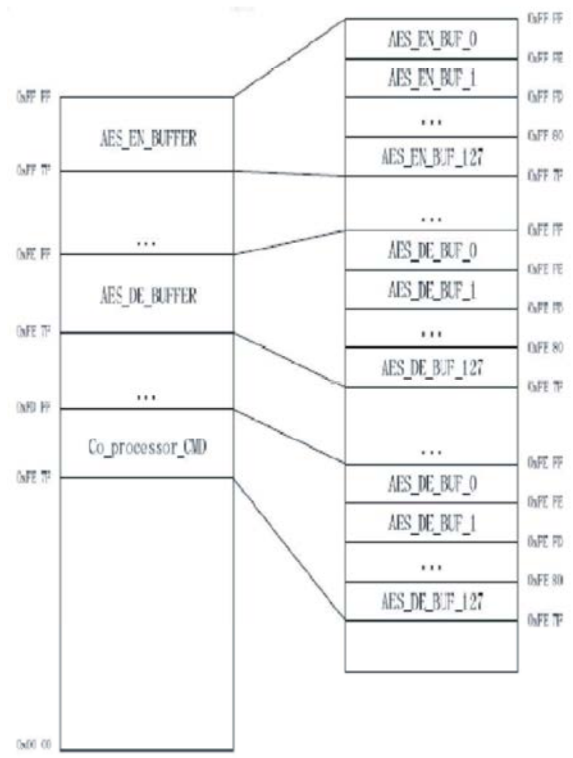


Fig. 3: FPGA Internal memory mapping

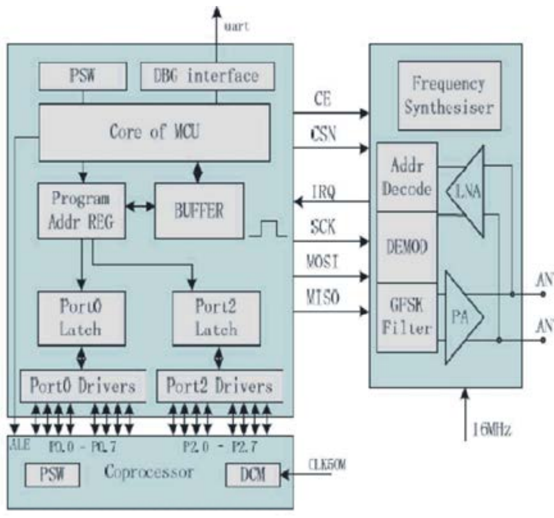


Fig. 4: Schematic Diagram of the Terminal

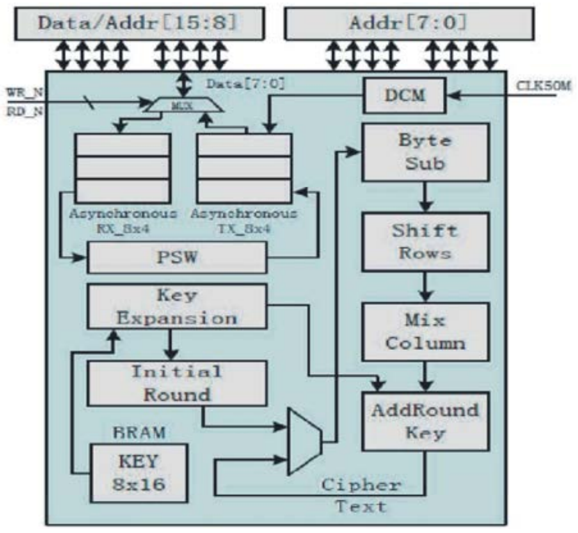


Fig. 5: The AES Comprocessor

Data Encryption Co-Processor: The Advanced Encryption Standard (AES; also referred to by its original name, Rijndael) specifies the latest standard in encryption for the protection of electronic information. The standard has been approved by the U.S. National Institute of Standards and Technology (NIST), which has made the specification publicly available in a Federal Information Processing Standards Publication (FIPS PUB 197). The motivation behind the new standard was the weakness of the existing Data Encryption Standard (DES) [2-4].

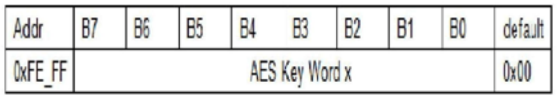
In addition to providing more security, AES is designed to lend itself to an easy implementation in hardware.

The AES Coprocessor: The 128-bit input data and initialization vector are each stored in Sixteen 8-bit registers (AES_EN_BUF) which are all write-only. The Encryption sequence is as follows:

- Write the 128-bit key in the Key Registers (AES_KEY).
- Write the initialization vector (or counter) in the Initialization Vector Registers.
- Set the bit DATRDY (Data Ready) in the AES Interrupt Enable register (AES_CMD), depending on whether an interrupt is required or not at the end of processing.
- Write the data to be encrypted/decrypted in the authorized Input Data Registers.
- Set the START bit in the AES Control register AES_CR to begin the encryption or the decryption process. The AES may be clocked through the Power Management Controller (PMC), so the programmer must first to configure the PMC to enable the AES clock. Here is the FPGA internal registers.

AES Key Word x Register:

TABLE I. AES_KEY, WRITE-ONLY

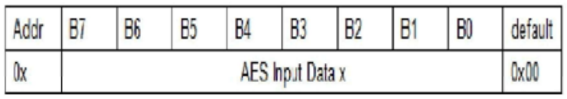


Addr	B7	B6	B5	B4 B3	B2	B1	B0	default
0xFE_FF	AES Key Word x							0x00

Aes_key, Write-only: The Sixteen 8-bit Key Word registers set the 128-bit cryptographic key used for encryption/decryption. KEYW1 corresponds to the first word of the key and respectively KEYW16 to the last one. These registers are write-only to prevent the key from being read by another application.

AES Input Data x Register:

TABLE II. AES_EN_BUF, WRITE-ONLY

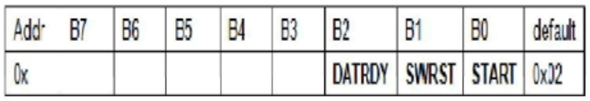


Addr	B7	B6	B5	B4	B3	B2	B1	B0	default
0x	AES Input Data x								0x00

Aes_en_buf, Write-only: The sixteen 8-bit Input Data registers set the 128-bit data block used for encryption/decryption. IDATA1 corresponds to the first word of the data to be encrypted/decrypted and IDATA16 to the last one. These registers are write-only to prevent the input data from being read by another application.

AES Control Register:

TABLE III. AES_DE_BUF, WRITE-ONLY



Add·	B7	B6	B5	B4	B3	B2	B1	B0	default
0x						DATRDY	SWRST	START	0x02

Aes_de_buf, Write-only:

Start: Start Processing
0 = No effect

1 = Starts manual encryption/decryption process.

SWRST: Software Reset 0 = No effect.

1 = Resets the AES. A software triggered hardware reset of the AES interface is performed.

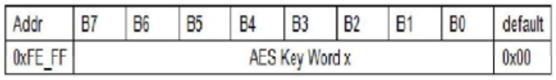
Datrdy: Data Ready Interrupt Enable

Data Error Correction Coprocessor: Bose-Chaudhuri-Hocquenghem (BCH) codes are a class of powerful random-error-correcting cyclic codes. The generator polynomial of the code is specified in terms of its roots from the Galois field GF(2") [2]. If α is a primitive element in the Galois field GF(2"), the generator polynomial g(x) is the lowest-degree polynomial over GF(2), which has α1, α2, α3…, α2t as its roots. Let (x) be the minimal polynomial of αi, then g(x) is the least common multiple of m1(x), m3(x), … m2t-1(x), that is: g(x) = m1(x) * m3(x) * … *m2t-1(x)

The degree of each minimal polynomial is m or less, the degree of g(x) is therefore at most m*t. In fact, the degree of the generator polynomial is 2*m for t = 2 and is 3*m for m > 4. During the first clock cycles, the two switches are connected to the "a" port and the k-bit message is input to the LFSR serially with most significant bit (MSB) first. Meanwhile, the message bits are also sent to the output to form the systematic part of the code word.

After k clock cycles, the switches are moved to the "b" port. At this point, the registers contain the coefficients of r(x).

TABLE I. AES_KEY, WRITE-ONLY



Addr	B7	B6	B5	B4	B3	B2	B1	B0	default
0xFE_FF	AES Key Word x								0x00

The BCH Decoding Involves Three Steps:

- Compute the syndrome from the received codeword.
- Find the error location polynomial from a set of equations derived from the syndrome.
- Use the error location polynomial to identify errant bits and correct them.

The system uses the BCH (31, 16,3) for data error correction. As the use of BCH (31, 16, 3) for data encoding, so the need to increase the 15bit data as parity-bit. If the user does not have the data error correction needs, you can bypass the BCH module.

CONCLUSION

Designed to ensure the quality of wireless transmission, it is fully taken into account data security and data encryption in using FPGA. If there is data validation error, the system can use the BCH error correction algorithm to reduce the packet loss rate. System is to achieve in a single PCB, FPGA signal without crossing to ensure maximum signal integrity. As can be seen from the data in Table IV, only a small number of LUT to achieve the coprocessor function and just modify the software can adapt to the new functional requirements.

REFERENCES

1. Rachh, R.R. and P.V. Ananda Mohan, 2008. Implementation of AES S-Boxes using combinational logic, Circuits and Systems, ISCAS IEEE.
2. Elbirt, Yip and Chetwynd, Paar, An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists, Dept. of Electr. & Comput. Eng., Worcester Polytech. Inst., MA, USA.
3. Gielata, A. and P. Russek, Wiatr, 0000. AES hardware implementation in FPGA for algorithm acceleration purpose, Dept. of Electron., AGH-UST, Cracow.
4. Serban, G., C. Anton, I. Tutanescu, L. Ionescu and A. Mazare, 0000. An implementation of BCH codes in a FPGA, Fac. of Electron., Commun. & Comput., Univ. of Pitesti, Pitesti, Romania.