Cryptographic Techniques for High Security Enabled Near Field Communication

Ashwin S, Priyatam Ch

Abstract— This paper presents the design and implementation of Near-Field Communication (NFC) architecture and implementation with high security support enabled by AES algorithm. NFC standards include exchange of data within a close proximity range and data transmission protocols. Due to the recent developments and applications of Near Field Communication technology, security has to be provided with reduction in complexity. For achieving this, an AES algorithm of 128-bit based Near Field Communication Cryptographic Tag architecture controlled by an 8 bit microcontroller has been designed in our work. Recent applications of RFID tags are controlled and monitored by finite state machines embedded in its hardware. But with the support of additional functionality like security, their design complexity drastically increases. Cryptographic techniques monitored by microcontroller leads to the reduction of this design complexity, because the hardware requirement for AES algorithm is very less compared to other Asymmetric cryptography techniques. This Cryptographic technique has been successfully synthesized, simulated and implemented using Xilinx ISE DS 13.2.2.using VHDL source code.

Index Terms—8BitMicrocontroller, Finite state machines, AESAlgorithm, Cryptography, RFID, NFC.

1 INTRODUCTION

THE main characteristic of NFC is that it is a wireless communication interface with a working distance limited to about 10 cm. The interface operates in different modes. The modes are distinguished whether a device creates its own RF field or whether a device retrieves the power from the RF field generated by another device. Radio-frequency identification (RFID) technology is the enabler for a variety of new applications. Many of these new applications will require RFID tags to support additional functionality, which increases their design complexity. Especially security functionality will play an important role. In order to cope with this increased complexity of the tags, new design concepts such as programmable approaches are necessary. Cryptography is the science of information and communication security that enables the confidentiality through an insecure channel communication. It prevents unauthorized alteration of use. It uses a cryptographic system to transform a plaintext into a cipher text, using most of the time a key. There exists certain cipher that doesn't need a key at all. The AES is the winner of the contest, held in 1997 by the US Government, after the data encryption standard was found too weak because of its small key size and the technological advancements in processor power. It takes an input block of a certain size, usually 128, and produces corresponding output block of the same size. The transformation requires a secret key the second input. Cryptography development has been a major concern in the fields of mathematics, computer science and engineering. One of the main classes in cryptography today is the symmetric key cryptography, where a same key will be used for the encryption and decryption processes. Our work relates to the field of secure tag design [1], which are:

1) First low-resource RFID tag that supports symmetric cryptography,
2) First low-area implementation of AES,
3) Flexible tag design based on a microcontroller for protocol handling and steering of the cryptographic module.

The remainder of this work is organized as follows. Section II provides a brief description of the deployed low-resource 8-bit microcontroller. In Section III the analyzed architecture of NFC tags Digital part is introduced in short. The AES algorithm and its processing architecture are presented in Section IV. The implementation of AES algorithm supported by 8 bit microcontroller and the analyzed area requirement output is shown in Section V, followed by conclusions in Section 6.

2. DESCRIPTION OF THE LOW RESOURCE 8 BIT MICROCONTROLLERS

2.1 Structure

The 8-bit microcontroller [2] performs the entire controlling tasks with focus on low chip area and low power consumption. The microcontroller bases on a Harvard architecture with separate program and data memory. The program memory ROM is implemented as lookup table in hardware and is divided into several pages. Each page can hold up to 256 16-bit instruction words. A maximum of 256 pages is supported. Data memory is realized as register with up to 64 8-bit registers. The register consists of three special-purpose (SP) registers, input-output (I/O) registers, and general-purpose (GP) registers. SP registers are used for indicating status information of the arithmetic-logic unit (ALU), the paging mechanism of the program ROM, and a dedicated accumulator (ACC) register. I/O registers allow interfacing of external modules. GP registers are used for computing and temporarily storing data. The

instruction set of the microcontroller involves 36 instructions, which are divided into arithmetic operations, logical operations and control operation. Most of the operations are executed within a single clock cycle.

2.2 Hardware and Software Functionality

The control complexity increases with integrating security and data management [1] features to a tag. Data has to be shifted from one component to another component. Commands that are breakup into number of blocks need to be rebuilt. Tag architecture with microcontroller can manage such a surged control complexity than a conventional state machine .Using a microcontroller; have to fulfill the requirements in terms of chip area and power consumption. Only a very simple microcontroller can be deployed for small chip size and clocked with the lowest possible clock frequency for low power consumption. Basic tag functionality is directly handled by framing logical hardware circuit. Since controlling complexity of basic tag functionality is low, implementation in hardware is achievable [1].Advanced tag functionality, leads to high control complexity.

3 OVERVIEW OF THE NFC TAG ARCHITECTURE DIGITAL PART

This presents the design and implementation Cryptographic protected tags for new RFID application operates at 13.56 MHZ and works passively. We target a low-area design that requires as little resources as possible such that the tag production does not exceed the practical limits of a possible commercial launch. The input data is directly feed to the NFC tag through framing logic. The area can be reduced by reusing a common 8-b microcontroller. The tag architecture consists of: a low resource 8 bit microcontroller [2], a framing logic (FL), a crypto unit (CU), and a memory unit. The microcontroller acts as the central element that controls all operations on the tag. The microcontrollers program is stored in an internal ROM and communicates via an AMBA bus with the crypto unit and framing logic.

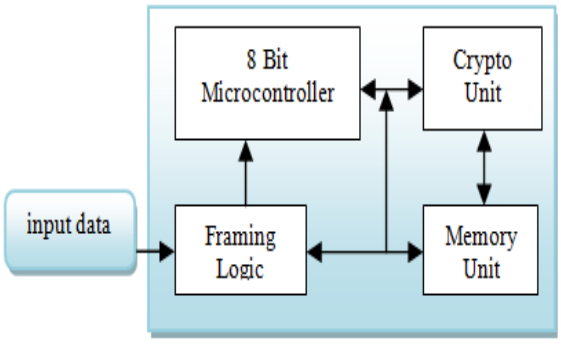

Fig 1 NFC Tag Architecture Digital Part

The FL is connected to the analog front-end and provides a byte interface for the microcontroller. But in our case the digital part of the NFC tag alone is taken for consideration .so, input data is given directly to the FL through rs232 interface. Cryptographic operations are processed within the CU that is accessed by the microcontroller via micro-code patterns. RAM stores temporary results. EEPROM for permanently storing data in files i.e. certified secret key, and memory for storing constants (ROM) are located in the memory unit.

3.1 Framing Logic

The FL[1] is an interface that converts serial data into parallel data and also handles the basic tag functionality .Contains the following components, Receive-and-transmit (RxTx) unit, Control unit, and AMBA interface.

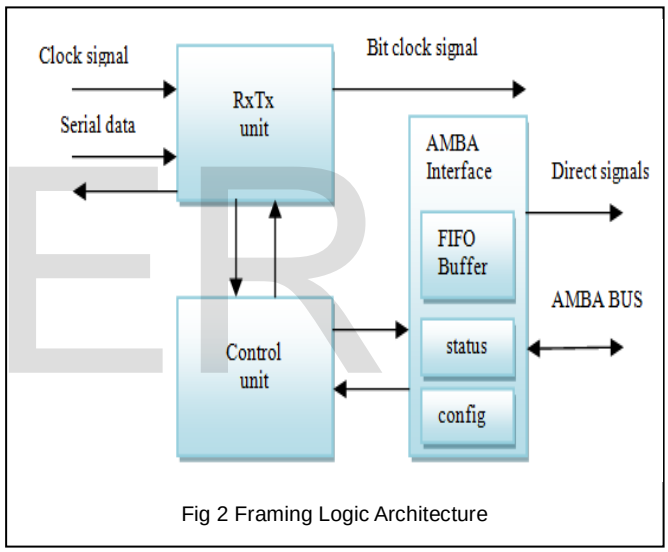

Fig 2 Framing Logic Architecture

The RxTx unit is the interface between the serial data signals of the RS232 and the parallel data signals of the control unit. Additionally, the RxTx unit is provided with an external clock signal .This provides a bit-clock signal to all components of the tag's digital part. But in this analysis our attention is only towards the Digital part of the system .so, input signal (data) is given directly to the RxTx Unit. Incoming serial data from the RS232 interface is first sampled by the RxTx unit, decoded into bits, transformed to byte data. The AMBA interface places this data into a FIFO buffer stores up to 6 Bit that is accessed by the microcontroller over the AMBA bus. The outgoing data from the microcontroller is first placed in the FIFO buffer of the FL and then transmitted to the RxTx unit by the control unit. The AMBA interface connects the FL with the AMBA bus. The AMBA interface also contains a status register that provides information about the internal state of the FL i.e. about the presence of data in the FIFO bus and a configuration register.

All the components can be accessed by the microcontroller via the AMBA bus.

3.2 8 Bit Microcontroller

Both memories are freely scalable i.e. their size can be adjusted during the design phase based on the requirements of the desired application. The program memory ROM is implemented as lookup table in hardware and is divided into several pages. A maximum of 256 pages is supported. Data memory is realized as register with up to 64 8-bit registers. There are three special purpose registers, I/O registers, and general purpose registers. SP registers give the status information of the ALU unit, ROMs paging mechanism, and an accumulator register. Interfacing with external modules is allowed by I/O registers. Thus the microcontroller controls the entire module during both data transmission and reception.

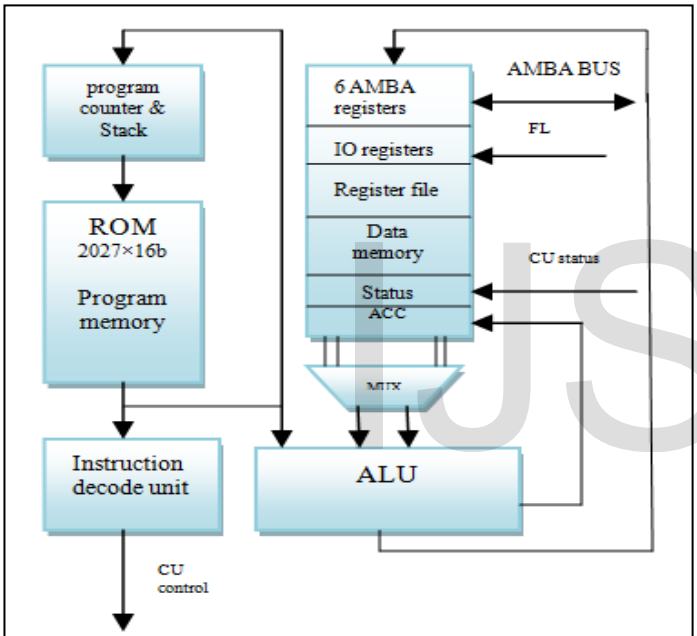

Fig 3 An8 Bit Microcontroller Architecture

GP registers are available for storing temporary data. Microcontroller's instruction set consists of 36 instructions, which are divided into logical operations, arithmetic operations, and control operations. Most of the operations are executed in a single clock cycle.

3.3 Cryptographic Unit

The cryptographic unit basically consists of three parts: a controller, RAM, and a data path [3]. The controller communicates with other modules on the tag to exchange data and it sequences the ten rounds of an AES encryption [6]. Therefore, it addresses the RAM accordingly and generates control signals for the data path. The RAM stores the 128-bit State and a 128-bit round key. These 256 bits are organized as 32 bytes to suit the intended 8-bit architecture.

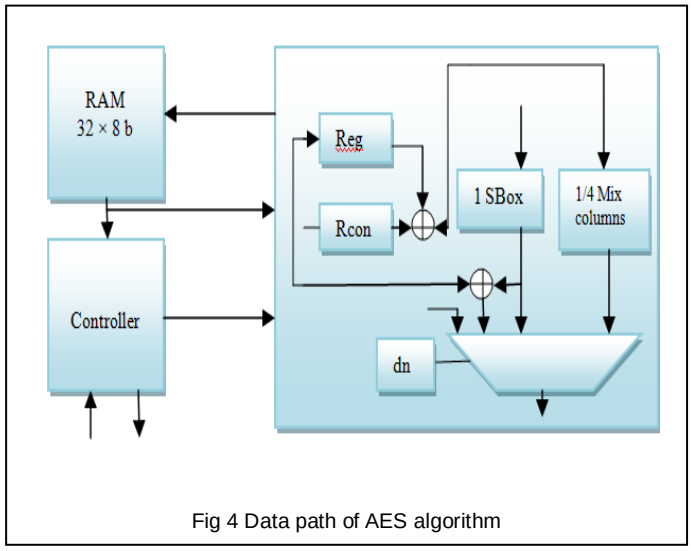

Fig 4 Data path of AES algorithm

The memory unit is composed of three memory types that are RAM, ROM, and EEPROM which are 16-bit linear dual-port memory space. A dual-port RAM showed to be advantageous since it allows reading of two words within one clock cycle. Also writing into one port and reading from the other is possible.

4 PROCESSING ARCHITECTURE OF AES ALGORITHM

AES is composed of 4 main stages for both encryption and decryption process in order to produce the final cipher text. The 4 main stages in the encryption process are Sub bytes transformation, shift rows, Mix columns and AddRoundKey. Similarly the decryption process also consists of four stages which just perform the inverse operations as that of the encryption process as follows inverse Sub bytes transformation, inverse Shift Rows, inverse mix columns and AddRoundKey step is common to both encryption and decryption.

Our work of NFC is based on AES128 algorithm [4]. The AES is a symmetric algorithm where the same key is used for both encryption and decryption.AES helps to transform the 128 bits block in to a new block which has the same size. At first the given input data is changed into a matrix form which is composed of 8 bit elements. After this transformation AES [5] will need to go through 4 main stages to produce the final cipher text. There is the SubByte step which involve substitution of bytes with the corresponding values from a fixed 8 bit look up table. Next stage is Shift Rows where rows of bytes are shifted an incremental one position to the left depend on which row it is i.e. row 2 will shift 1 position; row 3 will shift 2 position. First row will remain unchanged. Mix columns step every column is multiplied by a fixed polynomial. And at last, AddRoundKey step is performed where for each round of AES step, a subkey will be obtained from key schedule of AES algorithm and it will be continued for ten rounds.

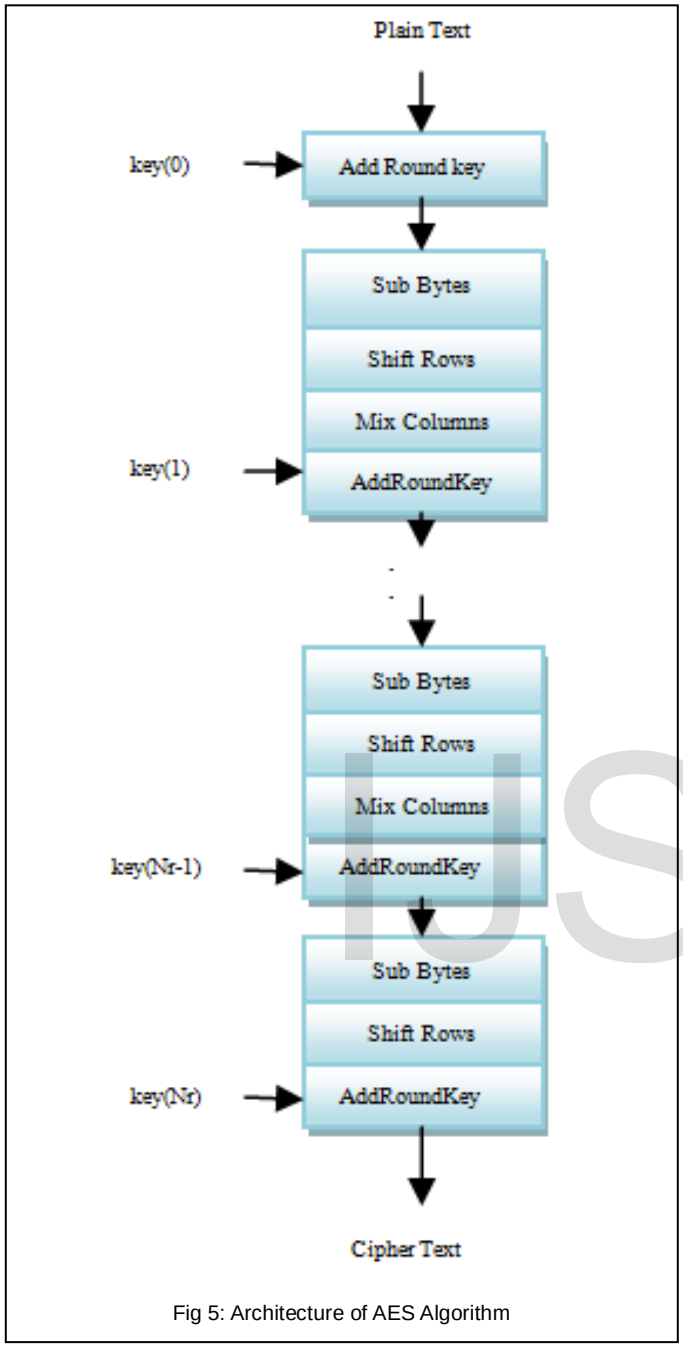

Fig 5: Architecture of AES Algorithm

As shown in figure 5 Nr represents the number of rounds and finally the encrypted data is obtained. Decryption is performed in the same manner as encryption but just the inverse operations are performed as similar to the steps discussed above.

5 IMPLEMENTATION RESULTS

We have implemented our flexible tag platform in VHDL and designed it towards low resource usage and low area requirement in Xilinx ISE design suite 13.2.2. AES is a flexible algorithm for hardware implementations because it covers the entire range of applications. AES hardware implementations can be applied in embedded systems for its low resource requirements. In this process the complete data encryption and decryption is done using AES algorithm which is controlled by an 8 bit microcontroller .But in the previous systems hardware components control is done by the state machines fixed in it which required about nearly 49950 gate elements. But by using our architecture design area requirement is nearly reduced into half as shown in figures 6 and 7.

round Project Status (11/29/2013 - 10:27:26)			
Project File:	nfc.xise	Parser Errors:	No Errors
Module Name:	micro_controller	Implementation State:	Placed and Routed
Target Device:	xc6vlx760l-1Lff1760	•Errors:	No Errors
Product Version:	ISE 13.2	•Warnings:	51 Warnings (51 new)
Design Goal:	Balanced	•Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	All Constraints Met
Environment:	System Settings	•Final Timing Score:	0 (Timing Report)

Device Utilization Summary				[-]
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	101	948,480	1%	
Number used as Flip Flops	101			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	116	474,240	1%	
Number used as logic	104	474,240	1%	
Number using O6 output only	79			
Number using O5 output only	1			
Number using O5 and O6	24			
Number used as ROM	0			
Number used as Memory	9	132,480	1%	
Number used as Dual Port RAM	0			
Number used as Single Port RAM	0			
Number used as Shift Register	9			

Fig 6 The Chip-Area Of The Microcontroller Unit

The above table shows the chip-area of the microcontroller unit result in terms No of flip flops, Registers and LUTs are clearly. This result shows that area requirement based on this microcontroller architecture is much less on comparison with conventional state machine approach.

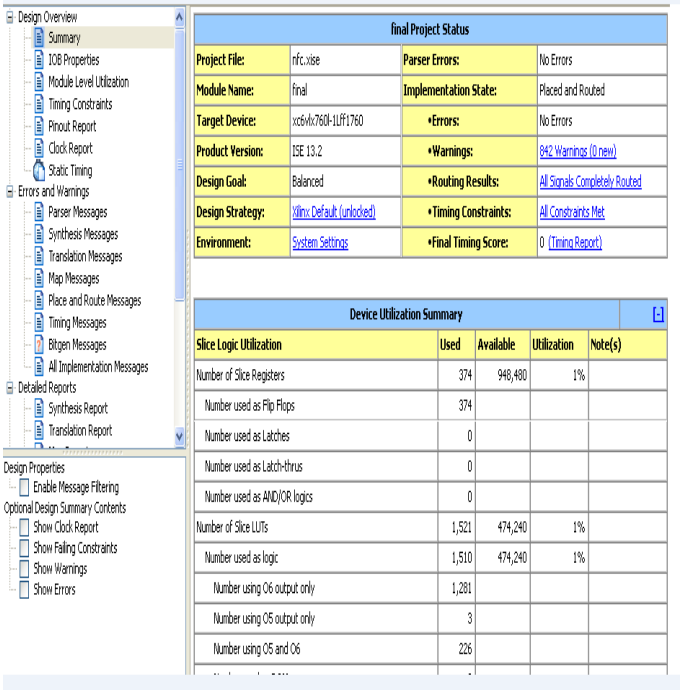

Fig 7 The chip-area results AES algorithm in NFC tag architecture

The above table shows the chip-area results in terms No of flip flops. The total no of Flip flops required for implementing this AES algorithm in NFC tag architecture is also given in figure 7 as per the obtained simulation output.Please note that math equations might need to be reformatted from the original submission for page layout reasons. This includes the possibility that some in-line equations will be made display equations to create better flow in a paragraph. If display equations do not fit in the two-column format, they will also be reformatted. Authors are strongly encouraged to ensure that equations fit in the given column width.

6 CONCLUSION AND FUTURE WORK

In this paper presents a flexible NFC-tag architecture that provides enhanced security features using symmetric as well as asymmetric cryptography. As a main contribution, the work described an entire "real-world" RFID system, including all hardware components needed for a practical chip fabrication. The design shows that significant resources can be saved by applying a microcontroller-based architecture instead of using a finite-state machine-based controlling. The reason lies in the fact that the controller can be simply reused by many hardware components, such as the CU or the RFID FL that would require more area when implemented as individual hardware modules. For example, AES encryption and decryption has been realized with an area overhead of only 374 no of slice registers. In the future, We plan to further analyze the design regarding enhanced implementation attacks, and to reduce the area requirement. This can be completely realized as a micro program, which reduces further chip-area requirements while increasing flexibility and assembly-based implementation convenience. The proposed system is planned to be designed using RSA algorithm which is an asymmetric cryptography in VHDL, simulated using Xilinx software and implemented using FPGA Spartan 3e. Being an asymmetric cryptography it can resists against unexpected security attacks. But it is a very big challenge to integrate asymmetric algorithm in NFC tag design because of its huge area requirement. So, we are currently working towards the implementation of RSA algorithm in NFC tag architecture along with area and power minimization.

REFERENCES

[1] Thomas Plos, Michael Hutter, Martin Feldhofer, MaksimiljanStiglic, And Francesco Cavaliere," security-enabled near-field communication tag with flexible architecture supporting asymmetric cryptography" IEEE transactions on very large scale integration (vlsi) systems, vol. 21, no. 11, November 2013.

[2] Hannes Gro and Thomas Plos," On Using Instruction-Set Extensions for Minimizing the Hardware-Implementation Costs of Symmetric-Key Algorithms on a Low-Resource Microcontroller" Institute for Applied Information Processing and Communications (IAIK),Graz University of Technology.

[3] M.Feldhofer, S.Dominikus, And J.Wolkerstorfer, "Strong Authentication For RFID Systems Using The AES Algorithm," in proc. ches, vol. 3156.aug. 2004, pp

[4] p. Hamalainen, T. Alho, M. Hannikainen, And T. D. Hamalainen,"Design and Implementation of low-area and low-power AES Encryption hardware core," in proc. 9th euromicro conf. digit. syst. design, sep. 2006, pp. 577–583.

[5] Ricci, M. Grisanti, I. De Munari, and P. Ciampolini, "Design of a 2 µW RFID baseband processor featuring an AES cryptographyprimitive," in Proc. Int. Conf. Electron., Circuits Syst., Sep. 2008, pp. 376–379.

[6] Shyamal Pampattiwar, "Literature Survey on NFC, Applications and Controller" International Journal of Scientific & Engineering Research, Volume 3, Issue 2, February-2012