International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 2; Issue: 4; April-2014

COMPARATIVE ANALYSIS OF FPGA DESIGN OF AES CORE ARCHITECTURE FOR EXTERNAL MEMORY DEVICE

Mr. Pradeep,
4th Sem M.Tech. Student,
Dept of ECE, BNMIT, Bangalore
pradeep4658@gmail.com

Dr. P. A. Vijaya
Professor, Dept. of ECE,
BNMIT, Bangalore
pavmkv@gmail.com

***Abstract*: The need for security has been increasing day by day. Various algorithms for encryption and decryption have been proposed, but all are having some problem with them. Some having less security, some consumes more area, some are having less speed and throughput. So our objective is to propose high effective AES core hardware architecture for implementing it to encrypt/decrypt the data in portable hard disk drive system that apply to effectively in the terms of speed, scale size and power consumption to comply with minimum speed of 5 Gbps (USB3.0). The 128 bits data path of two different AES architectures design has been proposed. Basic Iterative AES reuses the same hardware for all the ten iterations. One Stage Sub Pipelined AES, with one stage of outer pipelining in the data blocks that both of them are purely 128 bits data path architecture that different from the previous public paper.**

Keywords : AES,FPGA Design

1. INTRODUCTION

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data. It has been adopted by the U.S. government and is now used worldwide. It supersedes DES. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

The advanced encryption standard(AES), standardized by NIST, National Institute of Standards and Technology, is a cryptographic algorithm replacement to DES (Data Encryption Standard) algorithm as the federal standard to protect sensitive information. AES has already received widespread use because of its high security, high performance in both hardware and software. Many implementations are done in software but it seems to be too slow for fast applications such as routers and some wireless communication systems. AES is a 128 bit symmetric data block cipher with 128, 192 or 256 bits key. The data block was described by arrays of bytes in 4 x 4 matrix (Called "State") and it has four basic steps operation ; Sub Bytes, (or S-Box), ShiftRow, MixColumn, and AddRoundKey. These four steps are also known as layers. The four layer steps describe one round of the AES. Number of rounds is made vary according to the key size. The AES with 128-bit key size operates iteratively on those four basic steps for 10 rounds. However, the first and the final rounds are arranged in a slightly different manner compared to others. All four layers have their corresponding inverse operations. The deciphering is, therefore, the reverse order of the ciphering process. Operation steps are similar and at the comparable complexity. Moreover, both processes can share same set of designed hardware .The uses of AES in FDE application do exist in both software and hardware form, however they are not so widely published. Software-based AES is also vulnerable to attacks. In contrast, in the pure hardware implementation is more robust. AES IP cores are also available commercially in the form of both ASIC and FPOAs, to obtain the higher data rate AES (Obits/second), a technique of parallelization and pipelining can be combined. The implementations are physically secure since tempering by an outside attacker is naturally difficult. It's also a cost-effective solution for many application specific systems. Although our design is individual, it has drawn many useful ideas and modified some technique reported in to improve system throughput.

2. LITERATURE SURVEY

In2004. The Advanced Encryption Standard (AES) S-Box was the only non-linear structure of the AES algorithm; it dominated the hardware complexity of the AES cryptographic module. For Wireless Sensor Network applications, S-Box must offer low power, small area and high security. In this paper, a full-custom S-Box implementation with low-power, compact and well DPAresistant property was proposed. The energy-efficient Pass Transmission Gate (PTG) and three-input exclusive-OR (XOR) gate based on three-transistor XOR element were used to realize the functionality of S-Box, it resulted in the reduction of the area tradeoff and power consumption. The PTG-based latches, controlled by asynchronous latch controllers, were inserted in the data path to block the propagation of the dynamic hazards. A countermeasure against Differential Power Analysis, based on random delay insertion, was presented to attain high-security. The resulting implementation on a 0.25μm CMOS process is very suitable for the source-limited wireless sensor node chips. In1989. In this contribution, we derive a novel parallel formulation of the standard Itoh-Tsujii algorithm for multiplicative inverse computation over GF(2m). The main building blocks used by our algorithm are: field multiplication, field squaring and field square root operators. It achieves its best performance when using a special class of irreducible trinomials, namely, $P(X) = Xm + Xk + 1$, with m and k odd numbers and when implemented in hardware platforms. Under these conditions, our experimental results show that our parallel version of the Itoh-Tsujii algorithm yields a speedup of about 30% when compared with the standard version of it. Implemented in a Virtex 3200E FPGA device, our design is able to compute multiplicative inversion over GF (2193) after 20 clock cycles in about 0.94μS. In1993. Performance evaluation of the Advanced Encryption Standard candidates has led to intensive study of both hardware and software implementations. However, although plentiful papers present various

implementation results, it seems that efficiency could still be greatly improved by applying good design rules adapted to devices and algorithms. This paper addresses various approaches for efficient FPGA implementations of the Advanced Encryption Standard algorithm. As different applications of the AES algorithm may require different speed/area tradeoffs, we propose a rigorous study of the possible implementation schemes, but also discuss design methodology and algorithmic optimization in order to improve previously reported results. We propose heuristics to evaluate hardware efficiency at different steps of the design process. We also define an optimal pipeline that takes the place and route constraints into account. Resulting circuits significantly improve previously reported results: throughput is up to 18.5 Gbits/sec and area requirements can be limited to 542 slices and 10 RAM blocks with a ratio throughput/area improved by at least 25% of the best-known designs in the Xilinx Virtex- E technology. In2001. This paper presents the implementation of byte substitution (S-Box) layer of an AES system. The implementation is based on composite field technique. To support the designed S-Box, AES core as well as Key Scheduling unit were also designed; however with loose optimization. The S-Box is also designed with a number of pipeline stage options. These enable the system throughput customization. FPGA validation has confirmed to us the promise of the whole AES system design.

3. AES Encryption Operation

The AES round constitutes a fixed set of transformations applied to the State array. A separate Key Expansion unit isused to generate keys for each round of AES algorithm.

In each round, a data block is transformed by a sequence of operations:

- Addroundkey: the key schedule of the current round is added to data block by a simple using a XOR operation.
- SubBytes: replaces each byte of the 16 bytes of data block using the S-box lookup table value of that byte. The contents of an S-box is the multiplicative inverse in Galois Field (GF) (28), followed by an affine transformation.
- Shiftrows: obtains a new data block by cyclically shifting the block rows. The bytes of row i are shifted$i$ times, where $0 \leq i \leq 4$.
- Mixcolumns: transforms each column of the state array by multiplying it with a constant GF polynomial. It operates on the state column by column, treating each column as a four term polynomial. The columns are considered as polynomials over $GF(28)$ and multiplied x4+ 1 with a fixed polynomial a(x) given by: $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$

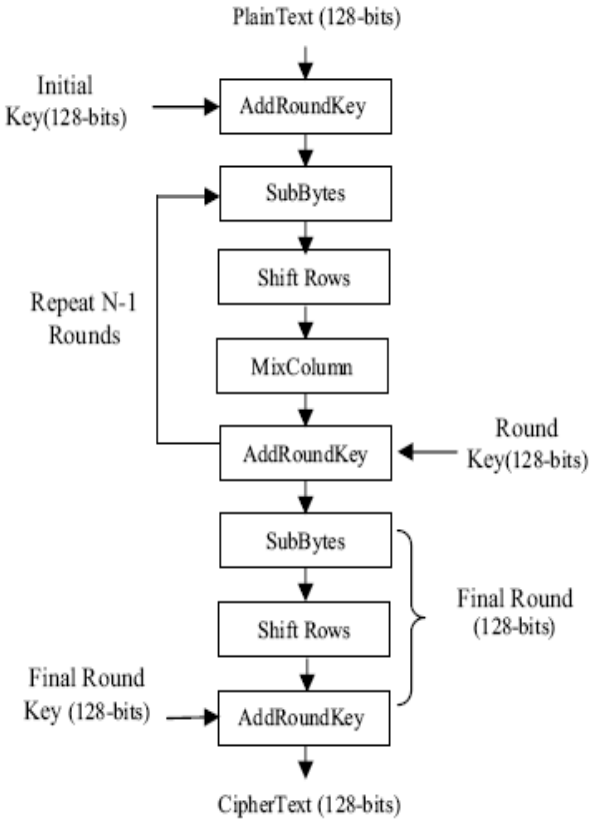
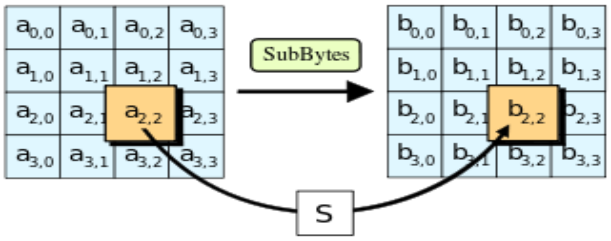


Fig 1: AES Encryption Round operation

3.1 *Round operation*



In the SubBytes step, each byte $a_{i,j}$ in the state matrix is replaced with a SubByte $S(a_{i,j})$ using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over $GF(28)$, known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), i.e.,$S(a_{i,j}) \neq a_{i,j}$, and also any opposite fixed points, i.e., $S(a_{i,j}) \oplus a_{i,j} \neq 0xFF$.

3.2 ShiftRows

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For blocks

of sizes 128 bits and 192 bits, the shifting pattern is the same. Row n is shifted left circular by n-1 bytes. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. The importance of this step is to avoid the columns being linearly independent, in which case, AES degenerates into four independent block ciphers.

3.3 MixColumns

In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher.

During this operation, each column is multiplied by the known matrix that for the 128-bit key is:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

The multiplication operation is defined as: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial un-shifted value. After shifting, a conditional XOR with 0x1B should be performed if the shifted value is larger than 0xFF.In more general sense, each column is treated as a polynomial over $GF(2^8)$ and is then multiplied modulo x4+1 with a fixed polynomial$c(x) = 0x03\ .\ x^3 + x^2 + x + 02$. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from$GF(2)[x]$. The MixColumns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.

3.4 *AddRoundKey*

In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

3.5 *ShiftRow Transformation*

The ShiftRow process operates on individual row with individual offset byte of state. In the state arrangement, data are fed into a square matrix in row order. To operate the ShiftRow transformation, register#l to store the whole data before byte swapping. This can result in the unsmooth data flow. However, the implementation is not very difficult. Due to this, the ShiftRow transform throughput is 128 bits per clock cycle for support the high throughput of hard disk has been designed. Register #2 is to be a pipeline arrangement (see Fig. 3 below) such that the data are arranged in order and ready for the following operation, MixColumntransformation.

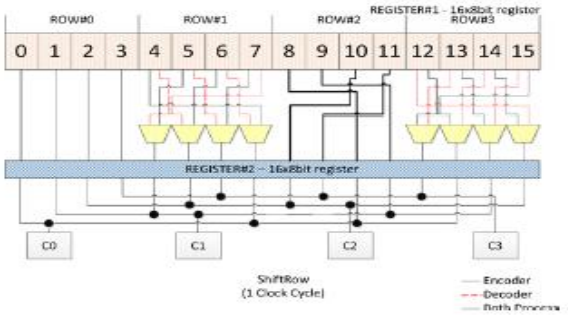


Fig 3Shift Rowand Inverse Shift RowSwitches with Mix Column-TRANSFORM Ready

3.6 *MixColumn Transform*

The mix column transformation operates on each column individually. Each byte of a column is mapped into a new value that is a function of all four bytes in that column. The transformation can be expressed by the following matrix multiplication on State.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} = \begin{bmatrix} c'_{0,0} & c'_{0,1} & c'_{0,2} & c'_{0,3} \\ c'_{1,0} & c'_{1,1} & c'_{1,2} & c'_{1,3} \\ c'_{2,0} & c'_{2,1} & c'_{2,2} & c'_{2,3} \\ c'_{3,0} & c'_{3,1} & c'_{3,2} & c'_{3,3} \end{bmatrix}$$

Each element in the product matrix is the sum of products of elements of one row and one column. In this case, the individual additions and multiplications are performed in GF(2^8). The MixColumn transform of an AES can be expressed asC'(x)= C(x)a(X)$_{\mathrm{mod}(x4+])}$and each column is considered as apolynomial with coefficients Ci,c define in GF(2^8). Themultiplication is modulo X^4 + 1 and a(x) is given by a(x)= ao+a_1x+a_2x^2+a_3x^3 where a0 = {02}, al = a2 = {OI} and a3 = {03}respectively. The inverse MixColumn matrix can be written in similar way b(x) = bo +b1x +b2x^2+b3$x^3$is defined as the inverse transform polynomial where bo = {OE} ,bl = {09}, b2= {OD} and b3 = {OB} respectively. Ci,c is computed in oneclock cycle (or one column per one clock cycle) with four parallel fixed-coefficient multipliers, followed by a summing operation as shown in Fig4.

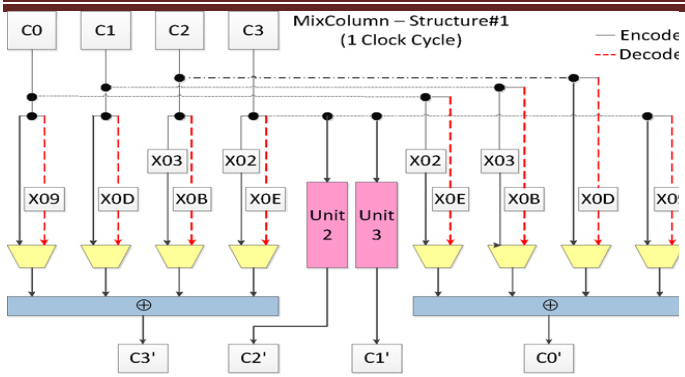


Fig 4 MixColumn and Inverse MixColumn Transform

3.7 KeyScheduling Transformation

The Key Scheduling expands the initial 128-bit cipher keysto generate the round keys. In this paper, the round keys applied to the data transformation for encryption ordecryption are calculated on-the-fly.The implementation of the key generation for encryption isillustrated in Fig 3.3. Every word (32 bits) of the next state is theXOR of the current word in the same position with its leftneighboring word. For example,the word in C1 is calculated as $w'[C_1] = w[C_1] \oplus w[C_0]$. For words in the position C1 its neighboring word is in position C3. A transformation RotWordis applied to the word in position C3 prior the XOR, followed by an XOR with the round constant Rcon. So, we need two128 bit register to store round keys. The Rot Word register isa circular word shift register. And The RCon is a feedback word shift register. For the one stage sub pipe1ined AESstructure, we have two different KeyScheduling modules which share the load of ten iterations. The KeyScheduling#lgenerates the first five keys and the KeyScheduling#2generates the last five keys. The entire KeyScheduling module generates three set of keys for iteration. Two set of keys from the KeyScheduling#1,#2 and the third, by accumulating the four 32 bits of outputs of the input block.

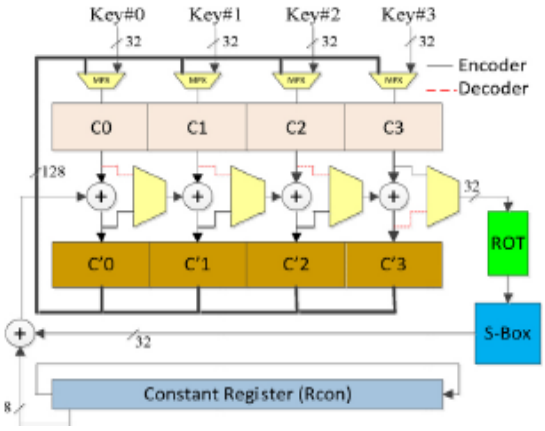


Fig 5Key Scheduling Structure

4. EXISTING SYSTEM

AES has already received widespread use because of its high security, high performance in both hardware and software. Many hardware architectures were proposed, but most of them were simple implementations according to the Rijndael specification. More recently, design strategies and implementation approaches were proposed for the implementation of block ciphers in reconfigurable hardware while other papers focused on some interesting algorithmic optimizations, especially for the highly expensive substitution box of Rijndael.

Many implementations are done in software but it seems to be too slow for fast applications such as routers and some wireless communication systems. The various AES hardware implementation architectures and optimizations have been suggested for different applications. Those to achieve high speed are usually very expensive in hardware complexity such as hard disk, ATM switch, etc. The large area of such architectures may not be suitable for some practical low-end embedded applications and do not require high speed or throughput, but is area critical. Therefore, reducing hardware resources to gain a compact and efficient implementation circuit is an increasing demand. Also in the existing model both encryption and decryption are done separately and they use the parallel architecture which results in less speed and throughput.

5. PROPOSED SYSTEM

This paper presents the design decisions and area optimizations to obtain a high throughput, over 30 Gbits/s AES processor. With loop unrolling and outer-round pipelining techniques, throughputs of 30 Gbits/s to 70 Gbits/s are achievable. The proposed AES core architecture can be chosen upon speed or throughput requirement for supporting portable hard disk data encryption. A one stage pipelined with LUT S-Box can give the throughput of more than 5 Gbps as high but it consumed more resource than composite field S-Box about 30%.It should be more efficiency than previous paper in the operation speed (the area wasn't significant different) and throughput.

The two Stage Sub-pipeline structure is an improvement of the basic iterative architecture with respect to speed. It has just two stage of pipeline within the data block. The data block is replicated twice. In two stage the same work load is shared by two data blocks. The hardware used in the data block is just more than twice the hardware used in Basic iterative design. Also deciphering is the reverse order of the ciphering process. Operation steps are similar and at the comparable complexity. Moreover, both processes can share same set of designed hardware. It's also a cost-effective solution for many application specific systems. Although our design is individual, it has drawn many useful ideas and modified some technique reported in to improve system throughput.

ADVANTAGES

1. Good speed and throughput due to pipelined architecture.
2. Both encryption and decryption in same architecture.
3. Consumes less area,
4. Complexity reduced,
5. Less expensive.

5.1 *TWO STAGE PIPELINED:*

In two stage the same work load is shared by three data blocks. The hardware used in the data block is just more than twice the hardware used in Basic iterative design. Also deciphering is the reverse order of the ciphering process. Operation steps are

AES Structure	CLB Slice	IOs	Freq. max (MHz)	TP(Gbps)
Basic Iteration AES	2230	391	239.16	3.78
One stage Pipelined	3100	391	481.25	6.16
Two stage pipelined	3784	391	698.14	9.25
Two stage pipelined with reduced shift row operation	4167	391	601.27	8.39

similar and at the comparable complexity. Moreover, both processes can share same set of designed hardware. It's also a cost-effective solution for many application specific systems. Although our design is individual, it has drawn many useful ideas and modified some technique reported in to improve system throughput.

5.2 *REDUCED ROUND OPERATION:*

Shift rows and mix column operations are reduced In the two stage sub pipelined architecture so the computation time is reduced

TWO STAGES PIPELINED:

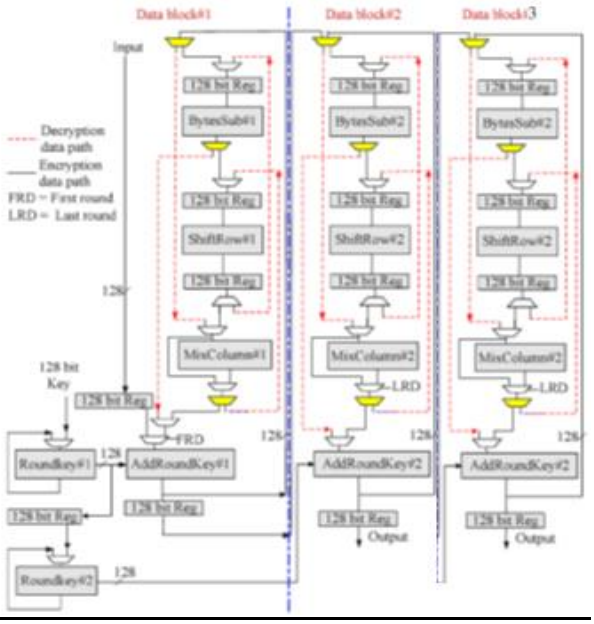


Fig.7 A 128 bits data path of the basic iteration (Data block#l) and two stage sub pipelined AES (Data block #2 and #3)

6. RESULT & DISCUSSION

Basic Iteration AES:

The basic iteration architecture gives Result with number of CLB Slice 2230 along with IOs 391 and Freq 239.16 MHz with throughput 3.78Gbps.

One Stage pipelined:

One stage pipelined architecture is the improvement over basic iterative architecture in terms of computation time with number of CLB Slices 3100 along with IOs 391 and Freq 481.25MHz with throughput of 6.16Gbps.

Two Stages pipelined:

Two stage pipelined architecture is the improvement over one stage pipelined architecture in terms of computation time with CLB Slices 3784 along with IOs 391 and Freq 698.14MHz with throughput 9.25Gbps.

Table I: Result of a Designed AES core Architecture

7. CONCLUSION

The proposed AES core architecture can be chosen upon speed or throughput requirement for supporting portable hard disk data encryption. A one stage pipelined with LUT S-Box can give the throughput of more than 5 Gbps as high but it consumed more resource than composite field S-Box about 30%.It should be more efficiency than previous paper in the operation speed (the area wasn't significant different) and throughput. A synthesizable VHDL code is developed for the implementation of both encryption and decryption process.The design is verified via the FPGA implementation with Xilinx family.

8. REFERENCES

[1] J. Daemen and V. Rijmen, AES Proposal: Rijndael (Version 2). NIST AES Website; http://csrc.nist.gov/publications/ and http://csrc.nist.gov/CryptoToolkit/aes/rijndaellRijndael Pdf
[2] NIST Federal Information Processing Standards (FIPS PUB197) Advanced Encryption Standard (2001, Nov.). [Online].Available:http://www.nist.gov/aes
[3] IP Cores, Ultra-Compact, Advanced Encryption StandardCore,availableat; http://www.ipcores.com/AES1.pdf
.[4] S. Chantarawong,P.Noo-intara, and S. Choomchuay, "An Architecture for S-Box Computation in the AES," Proc ofInformation and Computer Engineering Workshop2004 (ICEP2004), January 2004,pp.157-162.
[5] Alireza Hodjat, and Ingrid Verbauwhede, "Minimum area cost for a 30 to 70 Gbits/s AES processor," in Proc. of the 2004 IEEE Computer society Annual Symposium on VLSI, pp. 83-88, Feb.2004.
[6] Sumio Morioka, Akashi Satoh," An Optimized S-Box Circuit Architecture for Low Power AES Design "Tokyo Research Laboratory, IBM Japan Ltd.,2003.

AUTHORS PROFILE:

Mr. Pradeep is doing his M.Tech from the Dept. of E&C Engg, BNMIT, Bangalore, Karnataka, India. This paper is based on the project work done by him under the guidance of Dr. P. A. Vijaya.

Dr. P.A Vijaya did her B.E. from MCE, Hassan and M.E. and Ph.D. from IISc, Bangalore. She worked in MCE, Hassan, Karnataka for about 27 years. Presently, she is aProfessor in the Dept. of E&C Eng., BNMIT, Bangalore, Karnataka, India. Two students have obtained Ph.D. degree under her guidance and four more are doing Ph.D. Her research interests are in the areas of Pattern Recognition, Image Processing, Embedded Systems, Real time Systems, Network Security and Operating Systems.