INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

Design Of Area Optimized Aes 128 Algorithm Using Mixcolumn Transformation

Anumol Mathai
P.G. Student, Department Of ECE., SBC Engineering College, India

Dr. M. Sathyanarayana
Professor, Department Of ECE., SBC Engineering College, India

Abstract:
In cryptography, the Advanced Encryption Standard (AES), also known as Rijndael, is a block chiper adopted as an encryption standard. Rijndael is an encryption algorithm that has been designed with the state of art in the cryptographic research and is still believed very secure by most of the people. The algorithm accepts blocks of size 128, 192, or 256 bits. Independently, the key length can be 128, 192, or 256 bits as well. All encryptions are done in a certain number of rounds, which varies between 10, 12, and 14, and it depends on the size of the block length and the key length chosen. An encryption module is used to generate all the intermediate encryption data, and a separate key-scheduling module is used to generate all the sub-round keys from the initial key. The project is intended to design and implement AES algorithm and to maximize the encryption throughput while minimizing the area consumption at the same time maximizing the throughput will minimize the critical paths and solve the memory access conflicts. The VHDL code can be simulated to verify its functionality. Then gate level design equivalent will be synthesized targeting FPGA. Xilinx software is used for Design Entry, Simulation and Synthesis.

Key words: *Iterative algorithm, Data encryption standard, Cipher text, AES, FPGA, encryption, decryption, Rijndael, block cipher*

1.Introduction

The need for privacy has become a high priority for both governments and civilians desiring protection from signal and data interception. Widespread use of personal communications devices has only increased demand for a level of security on previously insecure communications. Both DES (Data Encryption Standard) and AES are defined as symmetric key block ciphers, with the main difference being the bit length of the key (56 bit for DES). These symmetric-key encryption schemes use the same key for both the sender and receiver, and as a result eliminate the need for the verification server needed in public keying. Symmetric keying lends itself to work independently of an open network and in turn a higher level of system interoperability. Ever since DES was phased out in 2001 and its successor, the Advanced Encryption Standard (also known as Rijndael) took its place, various AES implementations have been proposed both in software and hardware. This paper presents a low cost and low power hardware architecture for the Advanced Encryption Standard (AES). In 1997, the National Institute of Standards and Technology promoted worldwide research into a replacement for DES, or the widely accepted Data Encryption Standard. To minimize cost, focusing on efficiency reduced overall hardware complexity. By incorporating most of the algorithm complexity into the controller, components are reused and efficiency increased. A Verilog hardware implementation is also presented, utilizing a field programmable gate array (FPGA) as a prototyping platform. A focus on low power and cost allows for scaling of the architecture towards vulnerable portable communications devices in consumer and military applications such as cellular phones, PDAs, digital radios, pagers, and similar lower speed communication embedded systems.The importance of cryptography applied to electronic data transactions has acquired an inevitable relevance during the last few decades. Large volumes of information in various fields, such as financial and legal files, medical reports, bank services via Internet, telephone conversations, and e-commerce transactions are generated and interchanged among millions of users everyday. All these examples of applications and several others deserve a significant treatment from the security point of view, not only in the transport of such information but also in its storage. In this sense, cryptography techniques ,particularly at hardware levels, are especially applicable. Hence this implementation will find application in wireless security like military communication and mobile telephony where there are a greater emphasis on the complexity of computation and on the speed of communication. In cryptography, the AES, also called as Rijndael , is a block cipher adopted as an encryption standard by the US government, which specifies an encryption algorithm capable of protecting sensitive information [1, 2].The Rijndael algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data into an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plaintext. The AES algorithm is supports keys length of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits [3, 4],thus the name AES-128,AES-192 and AES-256 respectively The hardware implementation of the AES algorithm can provide high performance, low cost for specific applications and reliability ,compared to its software counterparts.

2.Description Of Aes Algorithm

The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data to an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plain-text.

2.1.Aes Encryption

The AES algorithm operates on a 128-bit block of data and executed Nr - 1 loop times. A loop is called a round and the number of iterations of a loop, Nr, can be 10**,** 12, or 14 depending on the key length. The key length is 128, 192 or 256 bits in length respectively.

The first and last rounds differ from other rounds in that there is an additional AddRoundKey transformation at the beginning of the first round and no MixCoulmns transformation is performed in the last round. Here we use the key length of 128 bits (AES-128) .

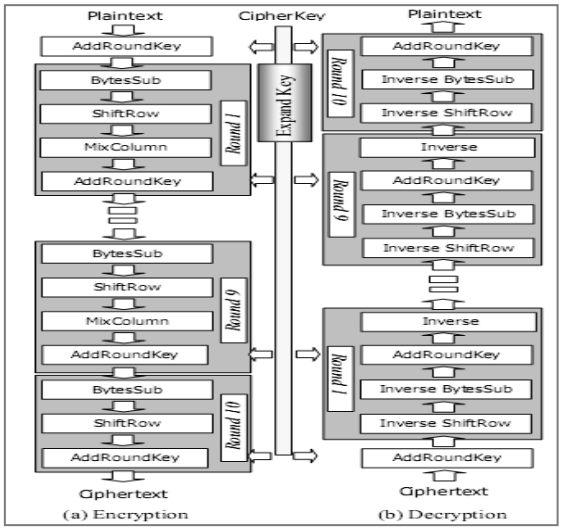

Figure : 1

2.1.1.Subbyte Transformation

The SubBytes transformation is a non-linear byte substitution, operating on each of the state bytes independently. The SubBytes transformation is done using a once pre-calculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. More details of the method of calculating the S-box is given in table II. This is a more efficient method than directly implementing the multiplicative inverse operation followed by affine transformation. This approach avoids complexity of hardware implementation and has the significant advantage of performing the S-box computation in a single clock cycle, thus reducing the latency.

2.1.2.Shiftrow Transformation

In ShiftRows transformation, the rows of the state are cyclically left shifted over different offsets. Row 0 is not shifted; row 1 is shifted one byte to the left; row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left.

2.1.3.Mixcolumns Transformation

In MixColumns transformation, the columns of the state are
Considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial c(x), given by:
c(x)={03}x3 + {01}x2 + {01}x + {02}

2.1.4.Addroundkey Transformation

In the AddRoundKey transformation, a Round Key is added to the State - resulted from the operation of the MixColumns transformation - by a simple bitwise XOR operation.The RoundKey of each round is derived from the main key using the KeyExpansion algorithm [2]. The encryption/decryption algorithm needs eleven 128-bit RoundKey, which are denoted RoundKey[0] RoundKey[10] (the first RoundKey [0] is the main key)

2.2.Aes Decryption

Decryption is a reverse of encryption which inverse round transformations to computes out the original plaintext of an encrypted cipher-text in reverse order. The round transformation of decryption uses the functions AddRoundKey, InvMixColumns, InvShiftRows, and InvSubBytes successively.

2.2.1.Addroundkey

AddRoundKey is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order. The description of the other transformations will be given as follows.

2.2.2.InvShiftRowsTransformation

InvShiftRows exactly functions the same as ShiftRows, only in the opposite direction. The first row is not shifted, while the second, third and fourth rows are shifted right by one, two and three bytes respectively

2.2.3.Invsubbytes Transformation

The InvSubBytes transformation is done using a once-pre-calculated substitution table called InvS-box. That InvS-box table contains 256 numbers (from 0 to 255) and their corresponding values. InvS-box is presented in Table II.

2.2.4.Invmixcolumns Transformation

In the InvMixColumns transformation, the polynomials of
degree less than 4 over GF(28), which coefficients are the
elements in the columns of the state, are multiplied modulo ($x4+ 1$) by a fixed polynomial $d(x) = \{0B\}x3 + \{0D\}x2 + \{09\}x + \{0E\}$, where {0*B*}, {0*D*}; {09}, {0*E*} denote hexadecimal values. In the next section, a description of the proposed design based on FPGA implementation of AES encryption/decryption function is detailed.

3.Proposed Work

In the existing system we have used number of multiplication and addition operations .In addition more registers are used to store the intermediate result. So this will consume more power and use more area .The overall system performance decreased. Instead we will use a particular mixcolumn and inverse mixcolum transformations to reduce the power and area.

3.1.Mixcolumn

The mixing of input bits along with XOR operations are performed. Here the number of registers used for storing intermediate results are reduced and area too.

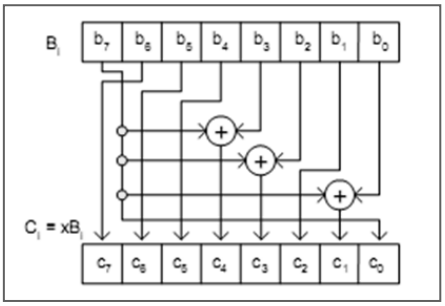

Figure 2: AX2 Fixed Coefficient Multiplier

3.2.Inverse Mixcolumn

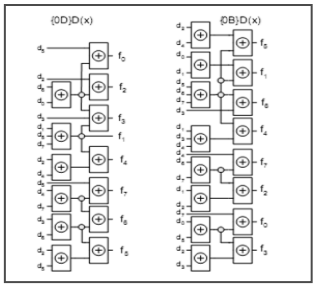

Figure 3: F(X)={0D}D(X) And F(X)={0B}B(X)

4.Result

The design has been coded by Verilog HDL. All the results are synthesized and simulated basing on the Quatus 9.0, the Model Sim – Altera 6.4a and EP20K400CB652C7 device. The results of simulating the encryption/decryption algorithm from the ModelSim simulator are shown in Fig.4 and Fig.5.

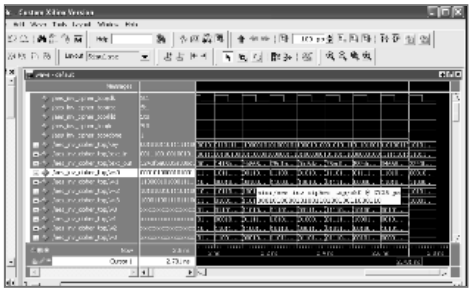

Figure 4: Simulation Result For Encryption

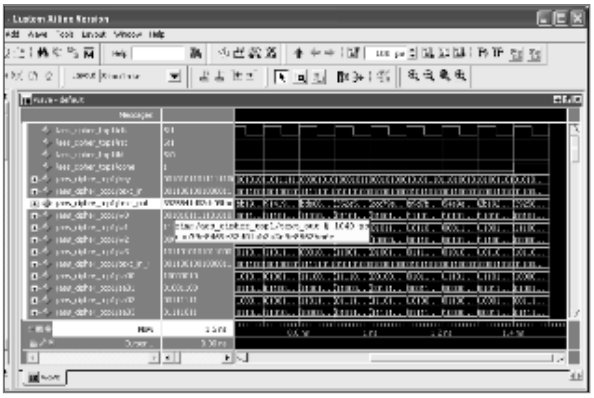

Figure 5: Simulation Result For Decryption

5.Conclusion

The Advanced Encryption Standard algorithm is a symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys with lengths of 128, 192, and 256 bits. An efficient FPGA implementation of 128 bit block and 128 bit key AES algorithm has been presented in this paper. The design is implemented on Altera using APEX20KC FPGA which is based on high performance architecture. The proposed design is implemented based on the iterative approach for cryptographic algorithms. Our architecture is found to be better in terms of latency, throughput as well as area. The design is tested with the sample vectors provided by FIPS 197 [2].

	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

x

Table 1: Algorithm Specification

	Key Length *(Nk words)*	**Block Size *(Nb words)***	**Number of Rounds *(Nr)***
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Table 2: S-Box

6.References

1. Raneesha K, Rema Vellody and R Nandakumar "Hardware efficiency comparison of AES implementations", 2012 International Conference on Communication Systems and Network Technologies.
2. NIST, "ADVANCED ENCRYPTION STANDARD (AES, Rijndael)", FIPS-197, November 2001.
3. . Hoang Trang and Nguyen Van Loi "An efficient FPGA implementation of the Advanced Encryption Standard algorithm", SBCCI 2002, pp. 197-202
4. Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, and Daniel A. Spielman " Efficient Erasure Correcting Codes", IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 47, NO. 2, FEBRUARY 2001.
5. Roohi Banu and Tanya Vladimirova, "On-Board Encryption in Earth Observation Small Satellites"
6. 1-4244-01 74-7/06/$20.00 ©2006 IEEE.
7. M. S. Gussenhoven and E. G. Mullen "Space Radiation Effects Program: An
8. Overview", IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 40, NO. 2, APRIL 1993.
9. . Farhan Aadil, Shahzada Khayyam Nisar, Wajahat Abbas, Asim Shahzad "Reusable IP core for Forward Error Correcting Codes", International Journal of Basic & Applied Sciences IJBAS-IJENS Vol: 10 No: 01.