International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 6, December 2024, pp. 6838~6845
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i6.pp6838-6845

Design of decryption process for advanced encryption standard algorithm in system-on-chip

Joseph Anthony Prathap[1], Mrinal Raj[2], Ritu Patnaik[1]
[1]Department of Electronics and Communication Engineering, School of Engineering, Presidency University, Bengaluru, India
[2]Department of Computer Science Engineering, School of Computer Science, Presidency University, Bengaluru, India

Article Info

Article history:

Received Mar 23, 2024
Revised Jul 21, 2024
Accepted Aug 6, 2024

Keywords:

Advanced encryption standard algorithm
Area and power analysis
Cadence electronic design automation
Field-programmable gate array
Hardware description language
Integrated circuit layout

ABSTRACT

This paper concentrates on the development of system-on-chip for the decryption algorithm in the advanced encryption standard (AES). This method includes the transformation of cipher text into plain text and consists of 4 sub-tasks based on the resolution. In this work, the 128-bit resolution is utilized to perform 10 rounds of transformation with the round key added at every round generated by the key expansion algorithm. Though there are many cryptography algorithms, the AES is simple, secure, faster in operation, and easy to develop compared to the others. The system-on-chip (SOC) design for the decryption of the AES depends on the synthesizable hardware description language (HDL) code development for all 10 rounds of processes with the key expansion algorithm. The lookup tables (LUTs) are used for the inverse S-box in the HDL code. The proposed SOC is designed using the Cadence electronic design automation (EDA) tools by making use of the synthesized HDL code for the proposed methods and analyzed for the very large-scale integration (VLSI) parameters.

Corresponding Author:

Joseph Anthony Prathap
Department of Electronics and Communication Engineering, School of Engineering, Presidency University
Itgalpur, Rajanakunte, Yelahanka, Bengaluru, India
Email: japtuhi1116@gmail.com

1. INTRODUCTION

Numerous real-time applications, including internet services, identity verification, banking authentication, and sensing technology, heavily rely on data security. The technique of disguising original material in a different format and retrieving it using the notion of a key is known as cryptography. Cryptography algorithms come in a variety of forms, including Rivest-Shamir-Adleman (RSA), elliptic curve cryptography (ECC), advanced encryption standard (AES), and data encryption standard (DES). The AES is the most widely utilized of these algorithms due to its high level of security, ease of construction, and speed of operation.

To increase data security, the cryptography algorithm uses diffusion and confusion. The picture encryption paradigm, which may be reused from previously generated data, can lead to both greater throughput and power savings [1]. The AES algorithm is used in several applications for its simplicity in design. The AES algorithm when implemented with an image encryption scheme is fast and safe from attacks [2]. The efficiency of the AES in image encryption has improved when bit permutation is used to replace the Mix column in the algorithm [3]. The field-programmable gate array (FPGA) is used for the realization of 128-bit AES algorithm-based image applications by incorporating the ZigBee module that can present low lookup tables (LUTs) [4]. The unnecessary switching in the AES-based ECG signals is eradicated using the Dynamic pipelined asynchronous model [5]. The issues of security, computing limitations, and increased power consumption when implementing AES can be overcome by a MicroBlaze processor [6].

The AES algorithm has been designed in several compiler versions, but the real-time implementation of the AES algorithm still requires enhancement in terms of area and power. The utilization of advanced FPGA devices has proven to be meritorious in the realization of Cryptography algorithms. The hardware description language (HDL) code for the AES algorithms is to minimize the latency and present less complexity in real-time hardware [7]. The AES algorithm when realized in FPGA is suitable for applications where area and security are high priorities [8], [9]. The performance of the AES algorithm is enhanced by utilizing a duty cycle-based accessing technique when implemented with the look-up table concept in the FPGA device [10].

The computation time can be minimized and efficiency can be maximized for the FPGA-based AES algorithm when accelerated in electronic codebook (ECB) mode [11]. The avalanche effect of the AES algorithm is enhanced with a slight enhancement in the computation time of the AES algorithm when the shift rows are randomized [12]. The computation speed for the AES algorithm is easy even though there is a renewal in the key [13]. The FPGA implementation of AES algorithm cryptography removes the threat of quantum computing by making use of parallelism in computation [14]. The area of the FPGA utilization is minimized by 50% when the look-up table for the 128-bit resolution of the AES algorithm [15]. In the FPGA realization of the AES algorithms, the composite field arithmetic is utilized to optimize the inverse S-Box [16]. The error identification in the AES algorithm using the neural network can retrieve the key with ease when implemented using the FPGA [17].

The FPGA/ASIC realization of the AES has proved to be efficient in parameters such as area, speed, and power [18]. The parameters of the cost, speed, and security of the cryptography algorithms are compared to find the best suit for real-time [19]. Power, performance, and throughput are all superior to the FPGA version of the AES algorithm [20]. To minimize device utilization in FPGA, the hardware implementation of the AES algorithm with 128-bit resolution makes use of the modified S-box structure and Vedic multiplier in the mix column stage [21]. For the traditional compiler architecture, the FAC-V achieves great speed and minimal FPGA cost when implementing the AES algorithm [22]. The current cipher architecture, when downloaded into an FPGA device, reduces latency, boosts throughput, is flexible, and uses little power [23]. The FPGA implementation of the cryptography algorithms has given way for the integrated circuit (IC) layout design to be fabricated. The layout for the 128-bit AES algorithm can be derived using electronic design automation (EDA) tools [24]. To summarize, there is a need for dedicated and customizable AES design which could be used as real IC device for any cryptographic application. In this paper, the IC layout for the developed very high-speed integrated circuit (VHSIC) hardware description language (VHDL) code of the decryption AES algorithm is designed by making use of the EDA tools. This paper aims at the feasibility of the system-on-chip (SOC) design for decryption of the AES algorithm using Cadence EDA tools.

2. THE PROPOSED IC CHIP FOR DECRYPTION OF AES ALGORITHM

The proposed method demonstrates the SOC design for the decryption algorithm of AES. The proposed SOC-AES decryption algorithm involves in development of synthesizable HDL code for the AES decryption techniques that could be used as application specific IC design for cryptographic applications. Procedurally, the AES decryption method includes 4 sub-processes in every round that involve i) inverse substitute byte, ii) inverse shift rows, iii) inverse mix columns, and iv) add round-key, but in the last round, the mix column step is excluded as shown in Figure 1.

- Inverse substitute bytes: As shown in Figure 2, this procedure, known as substitution, aids in the transformation by processing the state using a nonlinear byte substitution table (S-box) and acting on each of the state bytes separately. Using an 8-bit substitution box, the state array is replaced with a substitute byte $S(a(i,j))$ for each byte $a(i,j)$ in the inverse substitute bytes step. This step is exactly the reverse of the substitute bytes in the encryption method.
- Inverse shift rows: The state's rows are the subject of the shift rows step, which cyclically changes each row's bytes by a predetermined offset. The first row remains unaltered for AES. Every byte in the second row is moved to the right. Likewise, the offsets of two and three are used to shift the third and fourth rows, respectively. Bytes from each column of the input state make up each column of the output state of the shift rows step in this manner. This step is crucial to prevent the columns from being decrypted separately, which would cause AES to split into four separate block ciphers.
- Inverse mix column: The state is matrix multiplied with Galois fields (GF) in the inverse mix columns stage. The column value of the predefined matrix, or Galois matrix, is multiplied by each row value in the state. To obtain a plain-text column, the column multiplication results are XORed together.
- Add round key: The state and subkey are joined in the add round key phase. Rijndael's key schedule is used to derive a subkey from the main key for each round; each subkey has the same size as the state. By applying bitwise XOR to each byte in the state and the matching byte in the subkey, the subkey is appended.

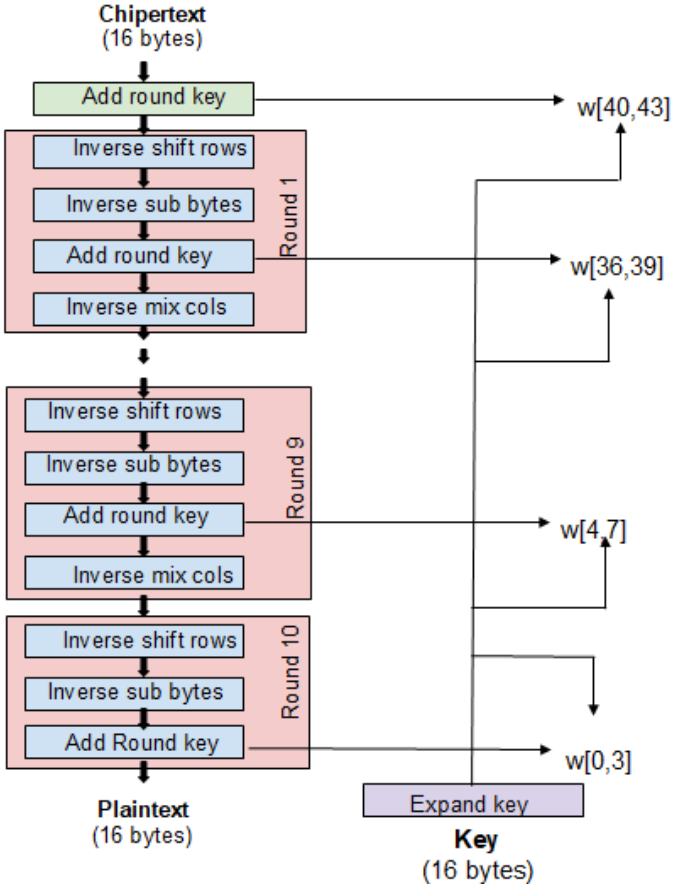


Figure 1. Block diagram of decryption algorithm in AES method to generate the plain text

	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

S_0'	S_4'	S_8'	S_{12}'
S_1'	S_5'	S_9'	S_{13}'
S_2'	S_6'	S_{10}'	S_{14}'
S_3'	S_7'	S_{11}'	S_{15}'

S_0	S_4	S_8	S_{12}
S_1	S_5	S_9	S_{13}
S_2	S_6	S_{10}	S_{14}
S_3	S_7	S_{11}	S_{15}

Figure 2. Inverse substitute bytes derived using the (16×16) S-Box

3. RESULTS AND DISCUSSION

The operation of the proposed methods is validated using the values assigned for the plaintext and key with its corresponding ciphertext. Considering an example and understanding the AES algorithm and its implications. The cipher text derived from the encrypted algorithm is as given and represented in hexadecimal. The cipher text, key, and plain text to be obtained using the decryption of the AES algorithm are given below.

$Ciphertext$: $9E756943661D7C5561F3F9781F5E32DE$
Key: $0A1B2C3D4E5F6789ABCDEF0123456789$
$Plaintext$: $0123456789ABCDEF0123456789ABCDEF$

The expansion of the 16-byte key into 10-round keys is displayed in Table 1. Word by word, as previously said, this procedure is carried out, with one column of the word round-key matrix corresponding to each four-byte word. The four round-key words produced for every round are displayed in the left-hand column.

Table 1. Key expansion steps for the decryption process using AES

Rounds	Keywords	Auxiliary Function
Start	$W_0 = 0A\ 1B\ 2C\ 3D$	$rotword(W_3) = 45\ 67\ 89\ 23 = X_1$
	$W_1 = 4E\ 5F\ 67\ 89$	$subword(X_1) = 6E\ 85\ A7\ 26 = Y_1$
	$W_2 = AB\ CD\ EF\ 01$	$rcon(1) = 01\ 00\ 00\ 00$
	$W_3 = 23\ 45\ 67\ 89$	$Y_1 \oplus rcon(1) = 6F\ 85\ A7\ 26 = Z_1$
I	$W_4 = W_0 \oplus Z_1 = 65\ 9E\ 8B\ 1B$	$rotword(W_7) = 49\ 64\ 1A\ A3 = X_2$
	$W_5 = W_4 \oplus W_1 = 2B\ C1\ EC\ 92$	$subword(X_4) = 3B\ 43\ A2\ 0A = Y_2$
	$W_6 = W_5 \oplus W_2 = 80\ 0C\ 03\ 93$	$rcon(2) = 02\ 00\ 00\ 00$
	$W_7 = W_6 \oplus W_3 = A3\ 49\ 64\ 1A$	$Y_2 \oplus rcon(2) = 39\ 43\ A2\ 0A = Z_1$
II	$W_8 = W_4 \oplus Z_2 = 5C\ DD\ 29\ 11$	$rotword(W_{11}) = 59\ A2\ 0A\ 54 = X_3$
	$W_9 = W_8 \oplus W_5 = 77\ 1C\ C5\ 83$	$subword(X_2) = CB\ 3A\ 67\ 20 = Y_3$
	$W_{10} = W_9 \oplus W_6 = F7\ 10\ C6\ 10$	$rcon(3) = 04\ 00\ 00\ 00$
	$W_{11} = W_{10} \oplus W_7 = 54\ 59\ A2\ 0A$	$Y_3 \oplus rcon(3) = CF\ 3A\ 67\ 20 = Z_3$
III	$W_{12} = W_8 \oplus Z_3 = 93\ E7\ 4E\ 31$	$rotword(W_{15}) = B2\ EF\ A8\ 47 = X_4$
	$W_{13} = W_{12} \oplus W_9 = E4\ FB\ 8B\ B2$	$subword(X_3) = 37\ DF\ C2\ A0 = Y_4$
	$W_{14} = W_{13} \oplus W_{10} = 13\ EB\ 4D\ A2$	$rcon(4) = 08\ 00\ 00\ 00$
	$W_{15} = W_{14} \oplus W_{11} = 47\ B2\ EF\ A8$	$Y_4 \oplus rcon(4) = 3F\ DF\ C2\ A0 = Z_4$
IV	$W_{16} = W_{12} \oplus Z_4 = AC\ 38\ 8C\ 91$	$rotword(W_{19}) = 9A\ A5\ 29\ 1C = X_5$
	$W_{17} = W_{16} \oplus W_{13} = 48\ C3\ 07\ 23$	$subword(X_4) = B8\ 06\ A5\ 9C = Y_5$
	$W_{18} = W_{17} \oplus W_{14} = 5B\ 28\ 4A\ 81$	$rcon(5) = 10\ 00\ 00\ 00$
	$W_{19} = W_{18} \oplus W_{15} = 1C\ 9A\ A5\ 29$	$Y_5 \oplus rcon(5) = A8\ 06\ A5\ 9C = Z_5$
V	$W_{20} = W_{16} \oplus Z_5 = 04\ 3E\ 29\ 0D$	$rotword(W_{23}) = 4F\ C1\ 86\ 0B = X_6$
	$W_{21} = W_{20} \oplus W_{17} = 4C\ FD\ 2E\ 2E$	$subword(X_5) = 84\ 78\ 44\ 2B = Y_6$
	$W_{22} = W_{21} \oplus W_{18} = 17\ D5\ 64\ AF$	$rcon(6) = 20\ 00\ 00\ 00$
	$W_{23} = W_{22} \oplus W_{19} = 0B\ 4F\ C1\ 86$	$Y_6 \oplus rcon(6) = A4\ 78\ 44\ 2B = Z_6$
VI	$W_{24} = W_{20} \oplus Z_6 = A0\ 46\ 6D\ 26$	$rotword(W_{27}) = 21\ E6\ 21\ F0 = X_7$
	$W_{25} = W_{24} \oplus W_{21} = EC\ BB\ 43\ 08$	$subword(X_6) = FD\ 8E\ FD\ 8C = Y_7$
	$W_{26} = W_{25} \oplus W_{22} = FB\ 6E\ 27\ A7$	$rcon(7) = 40\ 00\ 00\ 00$
	$W_{27} = W_{26} \oplus W_{23} = F0\ 21\ E6\ 21$	$Y_7 \oplus rcon(7) = BD\ 8E\ FD\ 8C = Z_7$
VII	$W_{28} = W_{24} \oplus Z_7 = 1D\ C8\ 90\ AA$	$rotword(W_{31}) = 3C\ 12\ 24\ FA = X_8$
	$W_{29} = W_{28} \oplus W_{25} = F1\ 73\ D3\ A2$	$subword(X_7) = EB\ C9\ 36\ 2D = Y_8$
	$W_{30} = W_{29} \oplus W_{26} = 0A\ 1D\ F4\ 05$	$rcon(8) = 80\ 00\ 00\ 00$
	$W_{31} = W_{30} \oplus W_{27} = FA\ 3C\ 12\ 24$	$Y_8 \oplus rcon(8) = 6B\ C9\ 36\ 2D = Z_8$
VIII	$W_{32} = W_{28} \oplus Z_8 = 76\ 01\ A6\ 87$	$rotword(W_{35}) = 53\ 93\ 04\ 77 = X_9$
	$W_{33} = W_{32} \oplus W_{29} = 87\ 72\ 75\ 25$	$subword(X_8) = ED\ DC\ F2\ F5 = Y_9$
	$W_{34} = W_{33} \oplus W_{30} = 8D\ 6F\ 81\ 20$	$rcon(9) = 1B\ 00\ 00\ 00$
	$W_{35} = W_{34} \oplus W_{31} = 77\ 53\ 93\ 04$	$Y_9 \oplus rcon(9) = F6\ DC\ F2\ F5 = Z_9$
IX	$W_{36} = W_{32} \oplus Z_9 = 80\ DD\ 54\ 72$	$rotword(W_{39}) = 93\ 33\ 73\ FD = X_{10}$
	$W_{37} = W_{36} \oplus W_{33} = 07\ AF\ 21\ 57$	$subword(X_9) = DC\ C3\ 8F\ 54 = Y_{10}$
	$W_{38} = W_{37} \oplus W_{34} = 8A\ C0\ A0\ 77$	$rcon(10) = 36\ 00\ 00\ 00$
	$W_{39} = W_{38} \oplus W_{35} = FD\ 93\ 33\ 73$	$Y_{10} \oplus rcon(10) = EA\ C3\ 8F\ 54 = Z_{10}$
X	$W_{40} = W_{36} \oplus Z_{10} = 6A\ 1E\ DB\ 26$	
	$W_{41} = W_{40} \oplus W_{37} = 6D\ B1\ FA\ 71$	
	$W_{42} = W_{41} \oplus W_{38} = E7\ 71\ 5A\ 06$	
	$W_{43} = W_{42} \oplus W_{39} = 1A\ E2\ 69\ 75$	

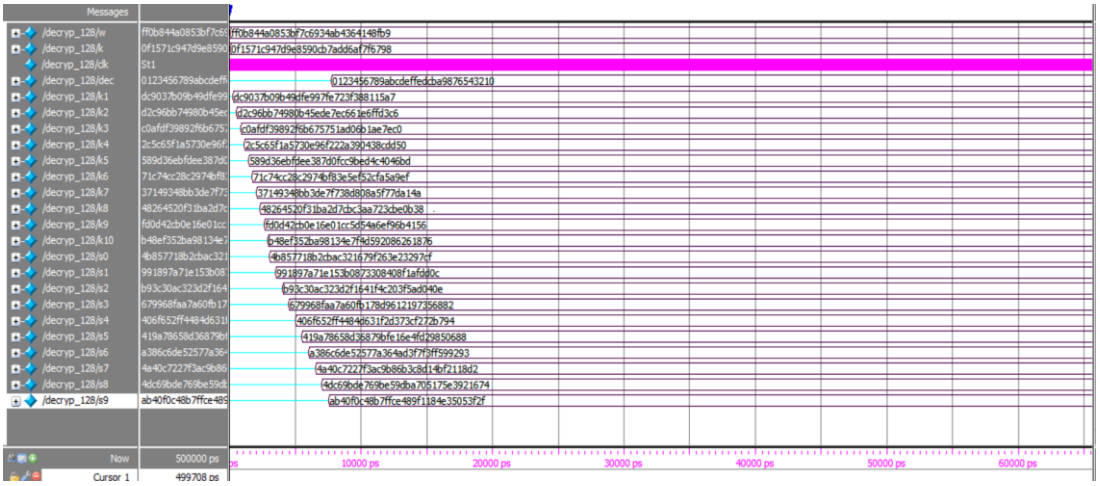
The AES is a symmetrical encryption algorithm and thus same key is used for both the encryption and decryption process. The difference is in the orientation of the keys utilized for the encryption and decryption is complementary. The state's progress through the AES decryption procedure is displayed in

Table 2. The state's value at the beginning of a round is displayed in the first column. The state for the first row is simply the cipher-text matrix arrangement. Following the inverse sub-bytes, inverse shift-rows, and inverse mix-columns transformations, the value of the State for that round is displayed in the second, third, and fifth columns, respectively. The values after the round key manipulation are seen in the fourth column. In the first round of decryption, only the cipher text is XORed with $W[40,43]$ as depicted in the block diagram, and then resulted values are represented in the 4th column of Table 2. The remaining rounds of the decryption include the inverse shift row, inverse sub-bytes, adding round keys, and inverse mix-columns. This is continued for the next 10 rounds to obtain the original plain text as given in the matric format.

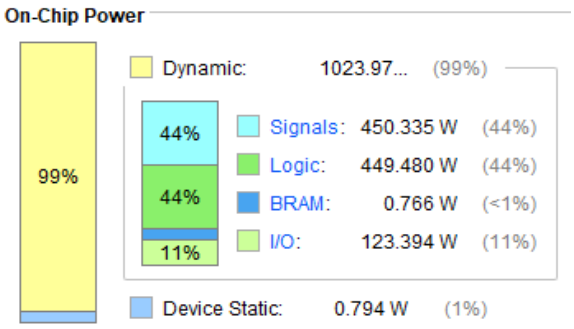
The simulation result for the evaluated cipher text for the given 128 bits of plain text is shown in Figure 3(a). The input for the developed code is 128 bits of cipher text and 128 bits of plain text. The behavioral model for the HDL code is used for the manipulation of the proposed decryption of the AES algorithm. The power report for the proposed method is evaluated using the Xilinx Tool as depicted in Figure 3(b) with the corresponding device utilization in Table 3. The IC layout for the decryption process is shown in Figure 4. The power manipulated in the cadence tool is given in Figure 5. Table 4 represents the comparison of the proposed method in terms of the device utilization in Xilinx tool. The proposed method is better than the existing method [25] that it consumes only 1.19% of FFs compared to the 13.03%. Also, the LUTs usage is 50% less than the existing method and block random-access memory (BRAM) is 2.22% compared to the 24.84%.

Table 2. Decryption steps depicting the state progress at each round of AES algorithm

Iteration	Start of round				After inverse shift rows				After inverse sub-byte				After round key				After the inverse mix column			
0	9E	66	61	1F	NOT APPLICABLE				NOT APPLICABLE				F4	0B	86	05	NOT APPLICABLE			
	75	1D	F3	5E									6B	AC	82	BC				
	69	7C	F9	32									B2	86	A3	5B				
	43	55	78	DE									65	24	7E	AB				
I	F4	0B	86	05	F4	0B	86	05	BA	9E	DC	36	3A	99	56	CB	50	5A	7A	C5
	6B	AC	82	BC	BC	6B	AC	82	78	05	AA	11	A5	AA	6A	82	52	FA	8C	7B
	B2	86	A3	5B	A3	5B	B2	86	71	57	3E	DC	25	76	9E	EF	93	9E	1F	33
	65	24	7E	AB	24	7E	AB	65	A6	8A	0E	BC	D4	DD	79	CF	FF	A6	32	E4
II	50	5A	7A	C5	50	5A	7A	C5	6C	46	BD	07	1A	C1	30	70	79	68	A7	AA
	52	FA	8C	7B	7B	52	FA	8C	03	48	14	F0	02	3A	7B	A3	53	87	52	B4
	93	9E	1F	33	1F	33	93	9E	CB	66	22	DF	6D	13	A3	4C	60	62	F8	41
	FF	A6	32	E4	A6	32	E4	FF	C5	A1	AE	7D	42	84	8E	79	7D	E1	6B	B9
III	79	68	A7	AA	79	68	A7	AA	AF	F7	89	62	B2	06	83	98	23	6E	3E	80
	53	87	52	B4	B4	53	87	52	C6	50	EA	48	0E	23	F7	74	28	83	8A	BD
	60	62	F8	41	F8	41	60	62	E1	F8	90	AB	71	2B	64	B9	F5	0A	5E	65
	7D	E1	6B	B9	E1	6B	B9	7D	E0	05	DB	13	4A	A7	DE	37	79	4E	24	3A
IV	23	6E	3E	80	23	6E	3E	80	32	45	D1	3A	92	A9	2A	CA	AF	BB	35	5E
	28	83	8A	BD	BD	28	83	8A	CD	EE	41	CF	8D	55	2F	EE	8D	69	9F	EE
	F5	0A	5E	65	5E	65	F5	0A	9D	BC	77	A3	F0	FF	50	45	3F	AC	EF	14
	79	4E	24	3A	4E	24	3A	79	B6	A6	A2	AF	90	AE	05	8E	64	D3	15	4B
V	AF	BB	35	5E	AF	BB	35	5E	1B	FE	D9	9D	1F	B2	CE	96	F1	42	4E	06
	8D	69	9F	EE	EE	8D	69	9F	99	B4	E4	6E	A7	49	31	21	AC	90	ED	25
	3F	AC	EF	14	EF	14	3F	AC	61	9B	25	AA	48	B5	41	6B	0E	DB	60	4C
	64	D3	15	4B	D3	15	4B	64	A9	2F	CC	8C	A4	01	63	0A	07	46	1E	B9
VI	F1	42	4E	06	F1	42	4E	06	2B	F6	B6	A5	87	BE	ED	B9	22	FF	C8	D6
	AC	90	ED	25	25	AC	90	ED	C2	AA	96	53	FA	69	BE	C9	C9	FC	1D	CC
	0E	DB	60	4C	60	4C	0E	DB	90	5D	D7	9F	1C	5A	9D	3A	69	95	8B	E3
	07	46	1E	B9	46	1E	B9	07	98	E9	DB	38	09	CA	5A	11	EA	D1	CA	A2
VII	22	FF	C8	D6	22	FF	C8	D6	94	7D	B1	4A	07	99	A2	0D	BC	AA	D3	43
	C9	FC	1D	CC	CC	C9	FC	1D	27	12	55	DE	C0	E9	BE	6C	F2	34	ED	7C
	69	95	8B	E3	8B	E3	69	95	CE	4D	E4	AD	80	C6	A9	42	75	DD	C2	6C
	EA	D1	CA	A2	D1	CA	A2	EA	51	10	1A	BB	60	A2	B8	13	1C	57	F1	63
VIII	BC	AA	D3	43	BC	AA	D3	43	78	62	A9	64	24	15	5E	30	38	F6	A3	38
	F2	34	ED	7C	7C	F2	34	ED	01	04	28	53	DC	18	38	0A	6F	66	9A	FE
	75	DD	C2	6C	C2	6C	75	DD	A8	B8	3F	C9	81	7D	F9	6B	E2	35	01	46
	1C	57	F1	63	57	F1	63	IC	DA	2B	00	C4	CB	A8	10	CE	07	7D	B7	1F
IX	38	F6	A3	38	38	F6	A3	38	76	D6	71	76	13	D	F1	D5	2B	C6	AC	AC
	6F	66	9A	FE	FE	6F	66	9A	0C	06	D3	37	92	C7	DF	7E	07	BF	28	28
	E2	35	01	46	01	46	E2	35	09	98	3B	D9	82	74	38	BD	F9	AC	AC	AC
	07	7D	B7	1F	7D	B7	1F	07	13	20	CB	38	08	B2	58	22	BE	33	33	33
X	2B	C6	AC	AC	2B	C6	AC	AC	0B	C7	AA	AA	01	89	01	89				
	07	BF	28	28	07	BF	28	28	38	F4	EE	EE	23	AB	23	AB				
	F9	AC	AC	AC	F9	AC	AC	AC	69	AA	AA	AA	45	CD	45	CD				
	BE	33	33	33	BE	33	33	33	5A	66	66	66	67	EF	67	EF				



(a)



(b)

Figure 3. The proposed AES decryption algorithm using the Xilinx tool: (a) simulation results and (b) power report

Table 3. Resource utilization for the proposed AES decryption algorithm using Xilinx tool

Resource	Estimation	Available	Utilization
LUT	9647	63400	15.22
FF	1504	126800	1.19
BRAM	3	135	2.22
IO	385	210	183.33
BUFG	1	32	3.13

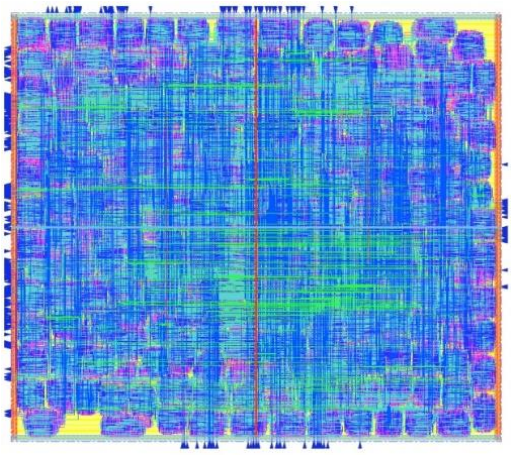


Figure 4. IC layout for the proposed AES decryption algorithm using the Cadence EDA tool

```
Total Power
-----------------------------------------------------------------------------------------
Total Internal Power:      150.63420027            58.5713%
Total Switching Power:     101.34523308            39.4062%
Total Leakage Power:         5.20152204             2.0225%
Total Power:               257.18095540
-----------------------------------------------------------------------------------------


Group                           Internal   Switching     Leakage       Total  Percentage
                                Power      Power         Power         Power  (%)
-----------------------------------------------------------------------------------------
Sequential                         26.09       3.185      0.3897       29.67       11.54
Macro                                  0           0           0           0           0
IO                                     0           0           0           0           0
Combinational                      124.5       98.16       4.812       227.5       88.46
Clock (Combinational)                  0           0           0           0           0
Clock (Sequential)                     0           0           0           0           0
-----------------------------------------------------------------------------------------
Total                              150.6       101.3       5.202       257.2         100
-----------------------------------------------------------------------------------------


Rail                  Voltage   Internal   Switching     Leakage       Total  Percentage
                                Power      Power         Power         Power  (%)
-----------------------------------------------------------------------------------------
VDD                       0.9      150.6       101.3       5.202       257.2         100
```

Figure 5. Power calculation for the proposed AES decryption algorithm using Cadence tool

Table 4. Comparison for the proposed AES decryption method with the existing methods

Method	[25]	Proposed method
LUT	71947 (31.23%)	9647 (15.2%)
FF	60040 (13.03%)	1504 (1.19%)
BRAM	77.5 (24.84%)	3 (2.22%)

4. CONCLUSION

The decryption process for the 128 bits of the AES algorithm is successfully implemented using the EDA tools. The proposed method has low area occupancy compared to the existing method. The application specific SOC is designed using the Cadence EDA Tools and the power analysis generated is low. Future work can be incorporated with the machine learning algorithm to identify the errors in the modified AES algorithms.

ACKNOWLEDGEMENTS

The authors would like to thank the Presidency University, Bengaluru for providing the VLSI Cadence lab in utilizing the EDA tools and for the contribution to this research article.

REFERENCES

[1] S. Hong, J. Im, S. M. M. Islam, J. You, and Y. Park, "Enabling energy efficient image encryption using approximate memoization," *Journal of Semiconductor Technology and Science*, vol. 17, no. 3, pp. 465–472, 2017, doi: 10.5573/JSTS.2017.17.3.465.

[2] H. V Gamido, A. M. Sison, and R. P. Medina, "Implementation of modified AES as image encryption scheme," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 6, no. 3, Sep. 2018, doi: 10.11591/ijeei.v6i3.490.

[3] H. V. Gamido, A. M. Sison, and R. P. Medina, "Modified AES for text and image encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 942–948, 2018, doi: 10.11591/ijeecs.v11.i3.pp942-948.

[4] T. L. Prasanna, N. Siddaiah, B. M. Krishna, and M. R. Valluri, "Implementation of the advanced encryption standard algorithm on an FPGA for image processing through the universal asynchronous receiver-transmitter protocol," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 6114–6122, Dec. 2022, doi: 10.11591/ijece.v12i6.pp6114-6122.

[5] E. S. Selvapriya and L. Suganthi, "Design and implementation of low power advanced encryption standard cryptocore utilizing dynamic pipelined asynchronous model," *Integration*, vol. 93, Nov. 2023, doi: 10.1016/j.vlsi.2023.102057.

[6] O. Azzouzi, M. Anane, M. Koudil, M. Issad, and Y. Himeur, "Novel area-efficient and flexible architectures for optimal Ate pairing on FPGA," *The Journal of Supercomputing*, vol. 80, no. 2, pp. 2633–2659, Jan. 2024, doi: 10.1007/s11227-023-05578-5.

[7] M. A. Rabbi Emon *et al.*, "Advanced encryption standard for embedded applications: an FPGA-based implementation using VHDL," in *2021 3rd IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, Dec. 2021, pp. 120–124, doi: 10.1109/MENACOMM50742.2021.9678241.

[8] U. Lee, H. K. Kim, J. Lee, and M. H. Sunwoo, "Area-efficient intellectual property (IP) design of advanced encryption standard," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 10, pp. 3797–3801, Oct. 2023, doi: 10.1109/TCSII.2023.3293999.

[9] B. Balaji Naik, Y. M. Sandesh, N. Kumar, and K. S. Vasundhara Patel, "Implementation of the AES algorithm on FPGA," in *International Conference on Emerging Research in Computing, Information, Communication and Applications*, 2024, pp. 89–99.

[10] D.-S. Kundi, A. Aziz, and N. Ikram, "A high performance ST-Box based unified AES encryption/decryption architecture on FPGA," *Microprocessors and Microsystems*, vol. 41, pp. 37–46, Mar. 2016, doi: 10.1016/j.micpro.2015.11.015.

[11] Y. Asfia, S. G. Khawaja, M. Asad, and A. Mirza, "Framework for live migration of FPGA based ECB-mode AES-128 accelerator," in *2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, May 2022, pp. 1–6, doi: 10.1109/ICoDT255437.2022.9787468.

[12] O. C. Abikoye, A. D. Haruna, A. Abubakar, N. O. Akande, and E. O. Asani, "Modified advanced encryption standard algorithm

for information security," *Symmetry*, vol. 11, no. 12, Dec. 2019, doi: 10.3390/sym11121484.
[13] W.-K. Lee, H. J. Seo, S. C. Seo, and S. O. Hwang, "Efficient implementation of AES-CTR and AES-ECB on GPUs with applications for high-speed FrodoKEM and exhaustive key search," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 6, pp. 2962–2966, Jun. 2022, doi: 10.1109/TCSII.2022.3164089.
[14] M. Bahadori, K. Jarvinen, and V. Niemi, "FPGA implementations of 256-bit SNOW stream ciphers for postquantum mobile security," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 11, pp. 1943–1954, Nov. 2021, doi: 10.1109/TVLSI.2021.3108430.
[15] Y. Zhou, G.-M. Tang, J.-H. Yang, P.-S. Yu, and C. Peng, "Logic design and simulation of a 128-b AES encryption accelerator based on rapid single-flux-quantum circuits," *IEEE Transactions on Applied Superconductivity*, vol. 31, no. 6, pp. 1–11, Sep. 2021, doi: 10.1109/TASC.2021.3075604.
[16] Y.-T. Teng, W.-L. Chin, D.-K. Chang, P.-Y. Chen, and P.-W. Chen, "VLSI architecture of S-Box with high area efficiency based on composite field arithmetic," *IEEE Access*, vol. 10, pp. 2721–2728, 2022, doi: 10.1109/ACCESS.2021.3139040.
[17] H. Wang and E. Dubrova, "Tandem deep learning side-channel attack on FPGA implementation of AES," *SN Computer Science*, vol. 2, no. 5, Sep. 2021, doi: 10.1007/s42979-021-00755-w.
[18] G. Manoj, J. Roopa Jayasingh, P. S. Divya, and Saravanan, "Comparative analysis of low power implementation for AES algorithm in ARTIX 7 FPGA & ASIC," *Przeglad Elektrotechniczny*, vol. 2023, no. 6, pp. 23–26, 2023, doi: 10.15199/48.2023.06.05.
[19] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight cryptography algorithms for resource-constrained IoT devices: a review, comparison and research opportunities," *IEEE Access*, vol. 9, pp. 28177–28193, 2021, doi: 10.1109/ACCESS.2021.3052867.
[20] K. Kalaiselvi and H. Mangalam, "Power efficient and high performance VLSI architecture for AES algorithm," *Journal of Electrical Systems and Information Technology*, vol. 2, no. 2, pp. 178–183, Sep. 2015, doi: 10.1016/j.jesit.2015.04.002.
[21] C. Arul Murugan, P. Karthigaikumar, and S. S. Priya, "FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-Box techniques for compact applications," *Automatika*, vol. 61, no. 4, pp. 682–693, Oct. 2020, doi: 10.1080/00051144.2020.1816388.
[22] T. Gomes, P. Sousa, M. Silva, M. Ekpanyapong, and S. Pinto, "FAC-V: An FPGA-based AES Coprocessor for RISC-V," *Journal of Low Power Electronics and Applications*, vol. 12, no. 4, Sep. 2022, doi: 10.3390/jlpea12040050.
[23] J. Damodharan, E. R. Susai Michael, and N. Shaikh-Husin, "High throughput PRESENT cipher hardware architecture for the medical IoT applications," *Cryptography*, vol. 7, no. 1, Feb. 2023, doi: 10.3390/cryptography7010006.
[24] R. Kumar *et al.*, "A time-/frequency-domain side-channel attack resistant AES-128 and RSA-4K crypto-processor in 14-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 4, pp. 1141–1151, Apr. 2021, doi: 10.1109/JSSC.2021.3052146.
[25] P. Nannipieri *et al.*, "VLSI design of advanced-features AES cryptoprocessor in the framework of the European processor initiative," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 2, pp. 177–186, Feb. 2022, doi: 10.1109/TVLSI.2021.3129107.

BIOGRAPHIES OF AUTHORS

Joseph Anthony Prathap was born in 1981 in Puducherry. He has obtained B.E in electronics and communication and M.Tech. VLSI design degrees in 2003 and 2007 respectively, and the Ph.D. in FPGA based power converters in 2017 from Annamalai University, Tamil Nadu, India. He has put in 19 years of service in teaching and research. He is currently an associate professor in the Department of Electronics and Communication Engineering at Presidency University, Bengaluru, Karnataka, India. He is a senior member of IEEE and has initiated for the formation of the IEEE Students Chapters in his career. His research interest includes VLSI design, development of digital switch patterns, FPGA control techniques for power converters, photovoltaic power electronics converters. He can be contacted at email: japtuhi1116@gmail.com.

Mrinal Raj is currently pursuing a B.Tech. degree in her third year at Presidency University, Bengaluru, Karnataka, India, with a keen interest in research and a strong enthusiasm for the field. As an enthusiast in the realm of research, she is an active student member of the Institute of Electrical and Electronics Engineers (IEEE), where she engages with peers and professionals in the field, staying updated on the latest advancements and opportunities. Her passion for exploration and discovery fuels her commitment to contribute meaningfully to the scientific community. She can be contacted at: mrinalraj12@gmail.com.

Ritu Patnaik is currently pursuing third year of Bachelor of Technology in electronics and communication engineering, at Presidency University, Bengaluru, Karnataka, India. She is an active student member of the Institute of Electrical and Electronics Engineers (IEEE), where she engages with peers and professionals in the field, staying updated on the latest advancements and opportunities. Her research interests include VLSI design, EDA tools usage in real application, applications of artificial intelligence, and cross compiling. She can be contacted at email: ritu.pnaik@gmail.com.