RTL Implementation of AES Encryption Algorithm

Prepared by:
Shehab Bahaa

128-Bit AES Project Report

Overview

This project is an RTL-based implementation of AES with a 128-bits key. It uses ten rounds for 128-bit keys as shown in figure (1). Each round comprises a series of operations as follows:

- Byte substitution
- Rows shifting
- Matrix multiplication
- Adding with a key specific for each round

An important required specification was to achieve a latency of 21 clock cycles, i.e., the core provides a valid ciphertext after 21 clock cycles from asserting valid state and key. To achieve this specification, the latency of each block is as follows:

- The latency of the pre-round transformation is one clock cycle.
- The latency of each round is two clock cycles. Moreover, inside each round block:
  - SubBytes block is sequential with one clock cycle latency.
  - ShiftRows block is combinational.
  - MixColumns is combinational
  - AddRoundKey is sequential with one clock cycle latency.

In parallel with these blocks, The keyExpansion block is sequential and produces the round key with a rate of one round key per two cycles.

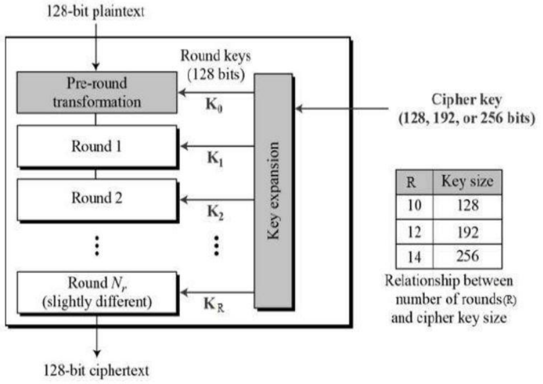


Figure (1)

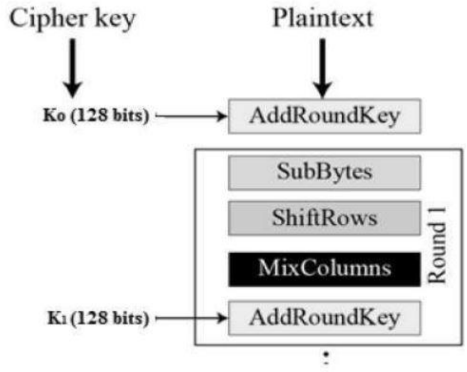
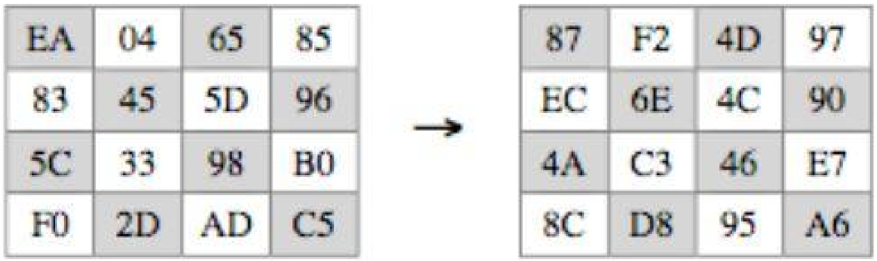


Figure (2)

The following section shows how each block and the entire design was tested, and the results of each test.

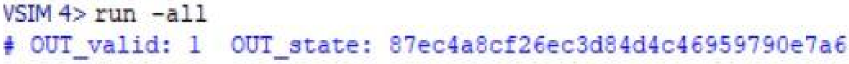
SubBytes:



EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

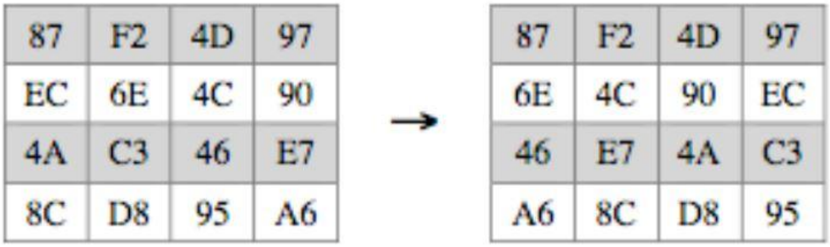
→

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6



```
VSIM 4> run -all
# OUT_valid: 1  OUT_state: 87ec4a8cf26ec3d84d4c46959790e7a6
```

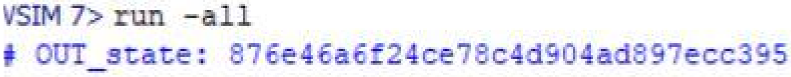
ShiftRows:



87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

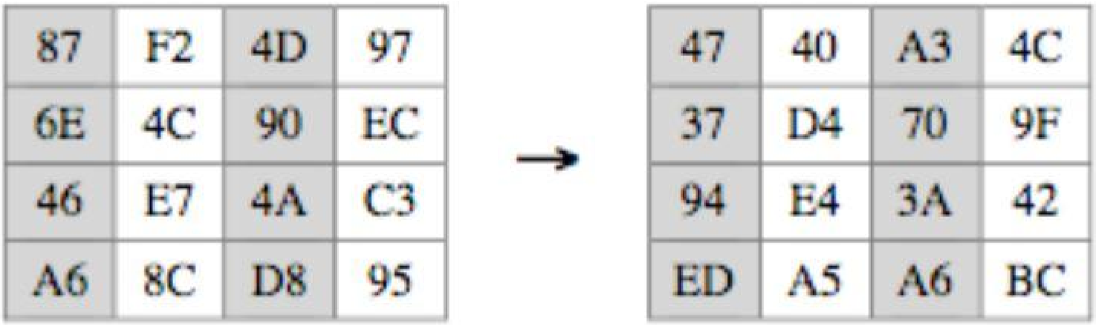
→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95



```
VSIM 7> run -all
# OUT_state: 876e46a6f24ce78c4d904ad897ecc395
```

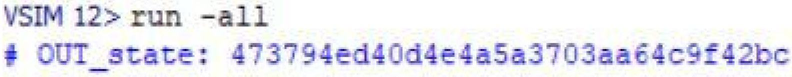
MixColumns:



87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

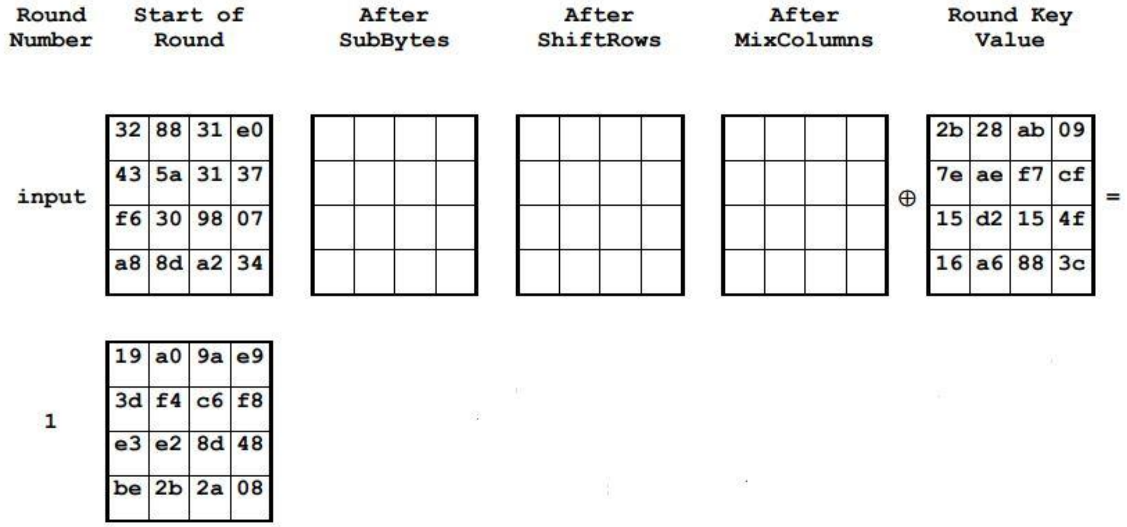
→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC



```
VSIM 12> run -all
# OUT_state: 473794ed40d4e4a5a3703aa64c9f42bc
```

AddRoundKey:



Start of Round (input):

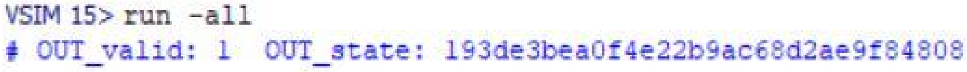
32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

Round Key Value:

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

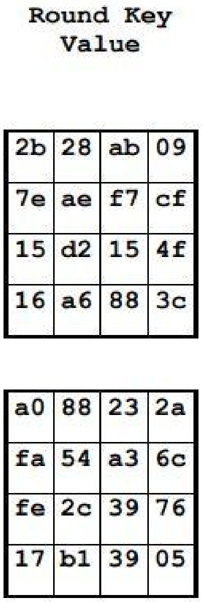
Start of Round (1):

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08



```
VSIM 15> run -all
# OUT_valid: 1  OUT_state: 193de3bea0f4e22b9ac68d2ae9f84808
```

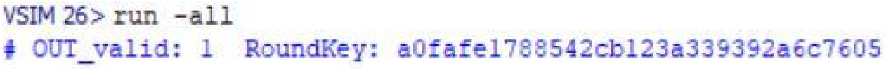
KeyExpansion for a single round:



Round Key Value

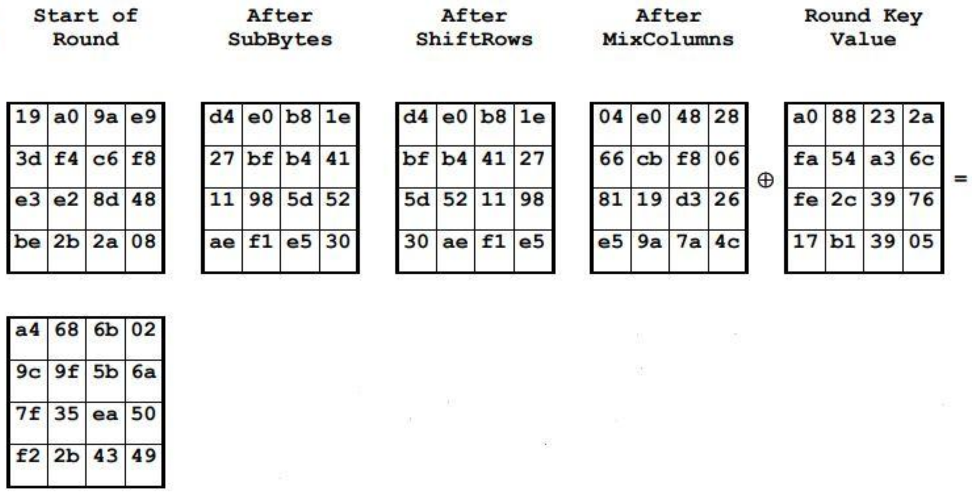
2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05



```
VSIM 26> run -all
# OUT_valid: 1  RoundKey: a0fafe1788542cb123a339392a6c7605
```

Round:



Start of Round

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

After SubBytes

d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

After ShiftRows

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

After MixColumns

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

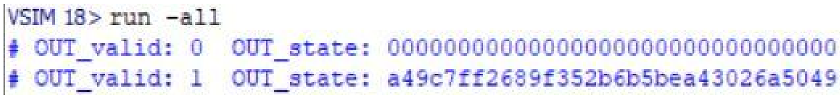
⊕

Round Key Value

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05

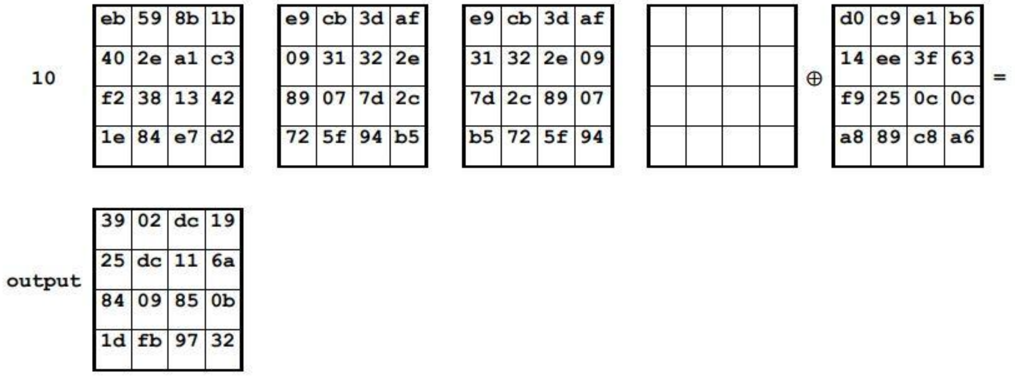
=

a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49



```
VSIM 18> run -all
# OUT_valid: 0  OUT_state: 00000000000000000000000000000000
# OUT_valid: 1  OUT_state: a49c7ff2689f352b6b5bea43026a5049
```

LastRound:



10

eb	59	8b	1b
40	2e	a1	c3
f2	38	13	42
1e	84	e7	d2

e9	cb	3d	af
09	31	32	2e
89	07	7d	2c
72	5f	94	b5

e9	cb	3d	af
31	32	2e	09
7d	2c	89	07
b5	72	5f	94

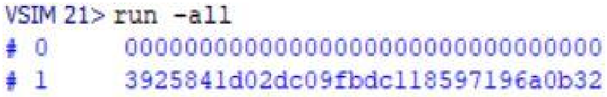
⊕

d0	c9	e1	b6
14	ee	3f	63
f9	25	0c	0c
a8	89	c8	a6

=

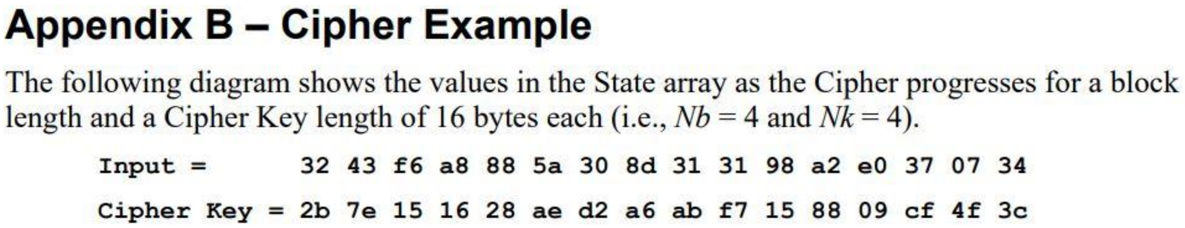
output

39	02	dc	19
25	dc	11	6a
84	09	85	0b
1d	fb	97	32



```
VSIM 21> run -all
# 0     00000000000000000000000000000000
# 1     3925841d02dc09fbdc118597196a0b32
```

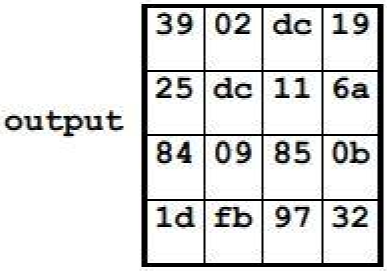
AES_128Core:



Appendix B – Cipher Example

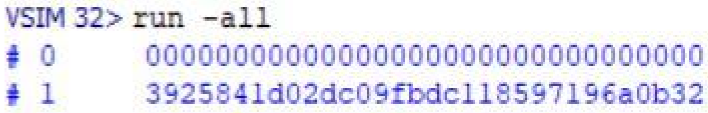
The following diagram shows the values in the State array as the Cipher progresses for a block length and a Cipher Key length of 16 bytes each (i.e., *Nb* = 4 and *Nk* = 4).

```
Input =       32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c
```



output

39	02	dc	19
25	dc	11	6a
84	09	85	0b
1d	fb	97	32



```
VSIM 32> run -all
# 0      00000000000000000000000000000000
# 1      3925841d02dc09fbdc118597196a0b32
```

References

[1] FIPS 197, Advanced Encryption Standard (AES)
[2] TutorialsPoint, Advanced Encryption Standard
[3] Washington University's Lecture in AES