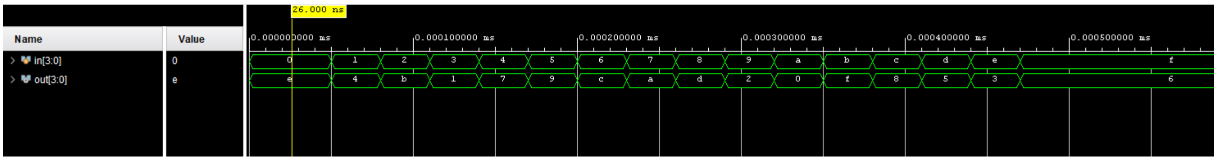
Assignment 7

Muhammed Erkmen

In this assignment, i applied BORON cryptology algorithm in verilog.

SBOX_LAYER TESTBENCH

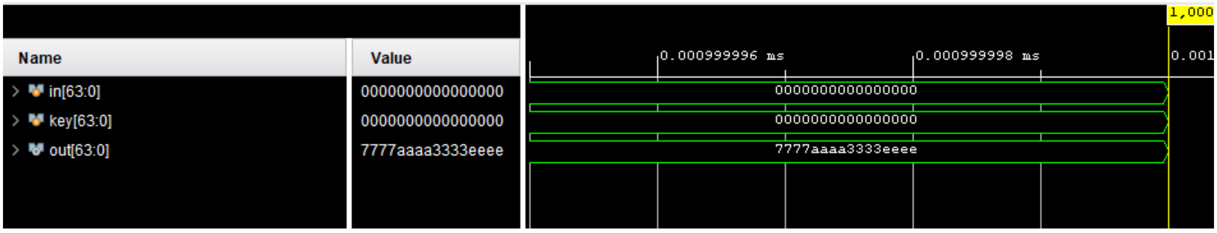


Name | Value
in[3:0] | 0
out[3:0] | e

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E	4	B	1	7	9	C	A	D	2	0	F	8	5	3	6

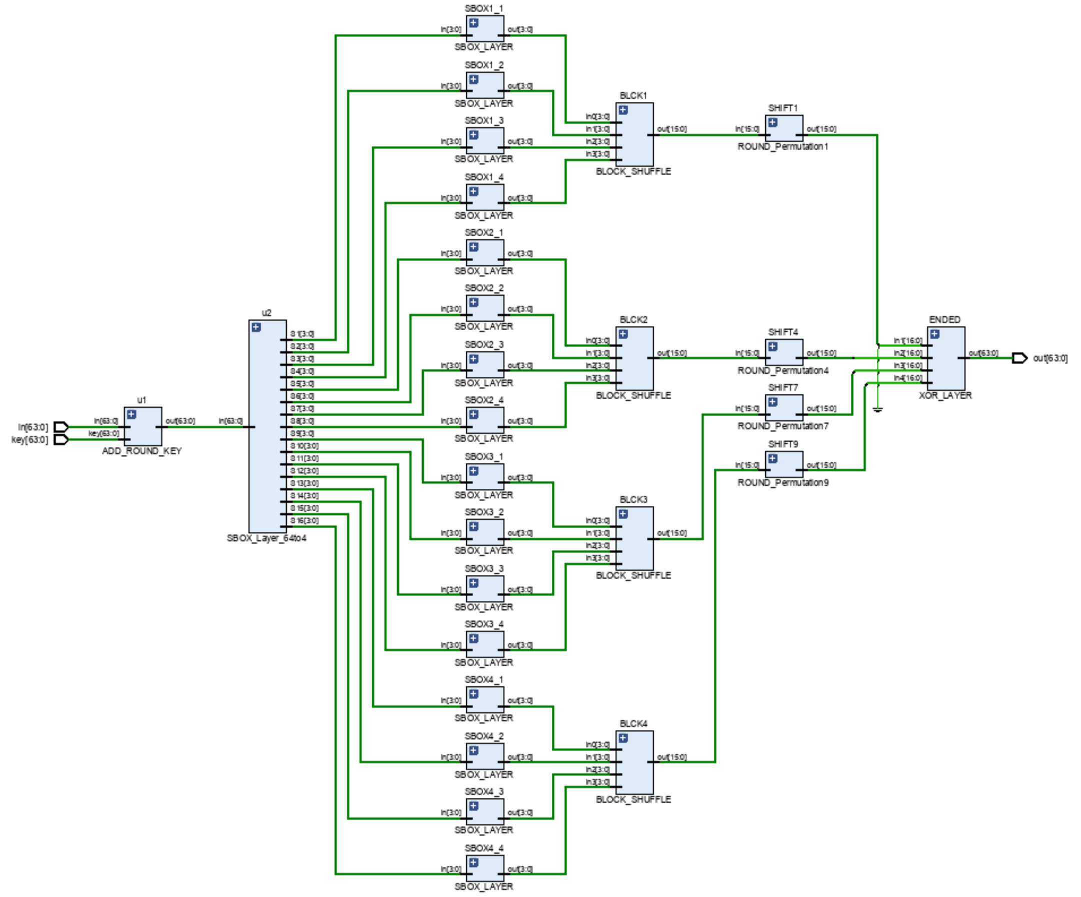
Question1

a) This is simulation result of the single round of BORON for input 64’d0 and key 80’d0. Output expected as 7777aaaa3333eeee and simulation results verified my code.



Name | Value
in[63:0] | 0000000000000000
key[63:0] | 0000000000000000
out[63:0] | 7777aaaa3333eeee

Schematic which is expected in assignment7 question1-a:



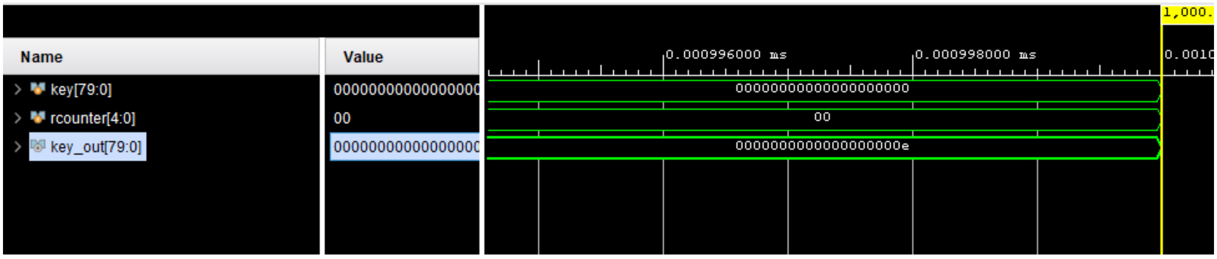
1-b)

In this part of the question 1, i designed key scheduler of the BORON algorithm. Key algorithm is simple as 13 bit shifting left, then applying S-BOX to the bits KEY[3:0]. And last step of key creating is KEY[63:59] XOR RoundCount.

Here is simulation results.

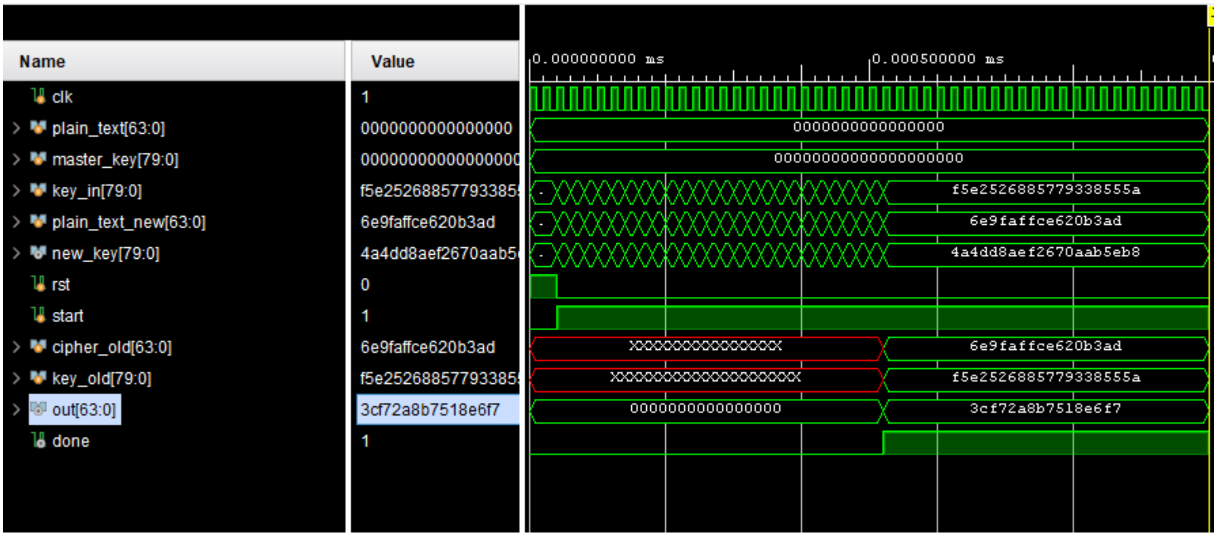
input => 00000000000000000000

output =>0000000000000000000e



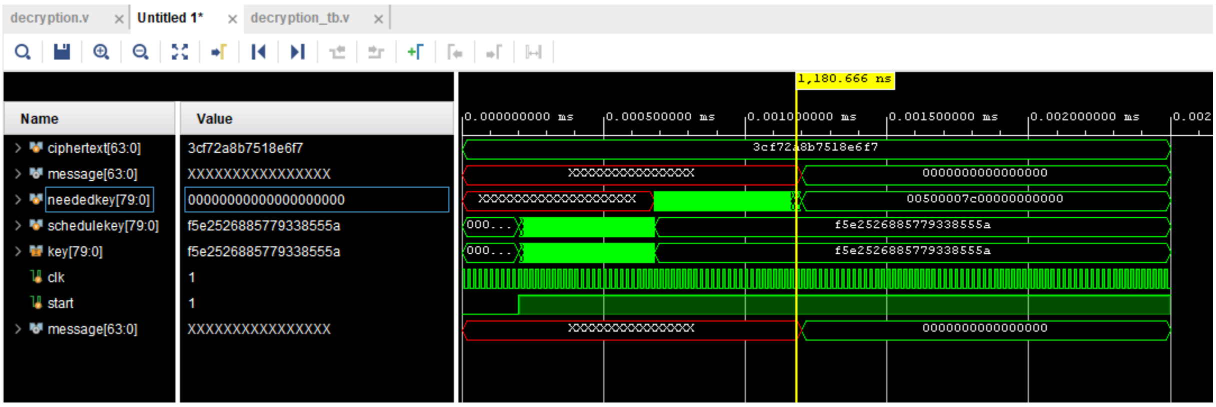
Name	Value
key[79:0]	0000000000000000
rcounter[4:0]	00
key_out[79:0]	0000000000000000

c)Encryption includes 25 cycle of single BORON round. In every step, algorithm uses the new generated key. And on the last cycle, algorithm has an XOR between new plaintext and 25th key. I applied this and got the expected 3cf72a8b7518e6f7.



Name	Value
clk	1
plain_text[63:0]	0000000000000000
master_key[79:0]	0000000000000000
key_in[79:0]	f5e25268857793385
plain_text_new[63:0]	6e9faffce620b3ad
new_key[79:0]	4a4dd8aef2670aab5
rst	0
start	1
cipher_old[63:0]	6e9faffce620b3ad
key_old[79:0]	f5e25268857793385
out[63:0]	3cf72a8b7518e6f7
done	1

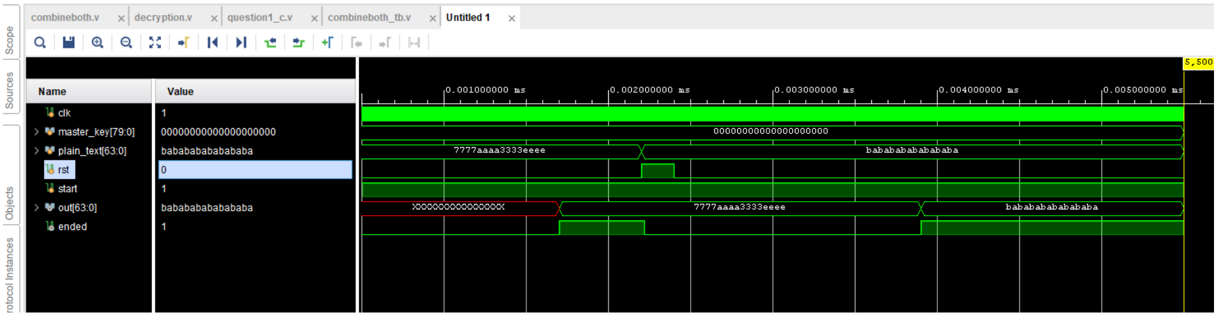
2)Decryption algorithm is exact inverse of the encryption algorithm. I calculated inverse of every step and created decryption algorithm. So the result of this application, if input is the 25th key (f5e2526885779338556a), we get 0 as we should in the output.



Name	Value
ciphertext[63:0]	3cf72a8b7518e6f7
message[63:0]	XXXXXXXXXXXXXXXX
neededkey[79:0]	00000000000000000000
schedulekey[79:0]	f5e2526885779338555a
key[79:0]	f5e2526885779338555a
clk	1
start	1
message[63:0]	XXXXXXXXXXXXXXXX

3) COMBINE BOTH

In this step i combined decryption and encryption together which completes the whole task in 75 clock cycle. This combined module has a start input, when start input is 1, encryption starts first. When encryption is completed, it sends the encrypted data and a start signal to the decryption module. After that, decryption makes its job as expected and we get the data we give 75 clock cycle before.



Name	Value
clk	1
master_key[79:0]	00000000000000000000
plain_text[63:0]	babababababababa
rst	0
start	1
out[63:0]	babababababababa
ended	1