# EE116 FPGA-Based Digital System Design

FINAL PROJECT

AES Encryption

Hongxiang Gao

18797306

## Content

## 1. Introduction

The Advanced Encryption Standard (AES) is formal encryption method adopted by the National Institute of Standards and Technology of the US Government, and is accepted worldwide. This paper introduces AES and key management, and discusses some important topics related to a good data security strategy.

AES encryption uses a single key as a part of the encryption process. The key can be 128 bits (16 bytes), 192 bits (24 bytes), or 256 bits (32 bytes) in length. The term 128-bit encryption refers to the use of a 128-bit encryption key. With AES both the encryption and the decryption are performed using the same key. This is called a symmetric encryption algorithm. Encryption algorithms that use two different keys, a public and a private key, are called asymmetric encryption algorithms.

An encryption key is simply a binary string of data used in the encryption process. Because the same encryption key is used to encrypt and decrypt data, it is important to keep the encryption key a secret and to use keys that are hard to guess. Some keys are generated by software used for this specific task. Another method is to derive a key from a pass phrase. Good encryption systems never use a pass phrase alone as an encryption key.

## 2. Operation Strategy

A standard AES process has 11 rounds, 10 normal rounds and a special round. For each round, it can be divided into two parts: message processing and key processing. Every part consists of several blocks, which correspond to the sub-processes in the operation. For instance, a “key” part contains three blocks: a rotator for circular byte left shifting, an s-box for byte substitution and an XOR adder for adding round constant. A “message” part contains four blocks: a 16-element XOR adder for adding round key, a substitution block for byte substitution, a shifter for shifting row and a mixer for mixing the columns. Then use these elements to construct a simple round.

Follow the strategy above, one can construct the whole system.

# 3. System Construction

## 3.1 Key Part

### 3.1.1Rotator

#### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity rotate is
        Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
                      a2 : in STD_LOGIC_VECTOR (7 downto 0);
                      a3 : in STD_LOGIC_VECTOR (7 downto 0);
                      a4 : in STD_LOGIC_VECTOR (7 downto 0);
                      clk:in STD_LOGIC;
                      b1 : out STD_LOGIC_VECTOR (7 downto 0):=x"00";
                      b2 : out STD_LOGIC_VECTOR (7 downto 0):=x"00";
                      b3 : out STD_LOGIC_VECTOR (7 downto 0):=x"00";
                      b4 : out STD_LOGIC_VECTOR (7 downto 0):=x"00");
end rotate;

architecture Behavioral of rotate is

begin
process(clk)
begin
if clk='1' and clk'event then
        b1<=a2;
        b2<=a3;
        b3<=a4;
        b4<=a1;
end if;
end process;

end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_rotate is
--   Port ( );
end test_rotate;

architecture Behavioral of test_rotate is
component rotate is
     Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
          a2 : in STD_LOGIC_VECTOR (7 downto 0);
          a3 : in STD_LOGIC_VECTOR (7 downto 0);
          a4 : in STD_LOGIC_VECTOR (7 downto 0);
          clk: in STD_LOGIC;
          b1 : out STD_LOGIC_VECTOR (7 downto 0);
          b2 : out STD_LOGIC_VECTOR (7 downto 0);
          b3 : out STD_LOGIC_VECTOR (7 downto 0);
          b4 : out STD_LOGIC_VECTOR (7 downto 0));
end component;

signal a1:std_logic_vector(7 downto 0);
signal a2:std_logic_vector(7 downto 0);
signal a3:std_logic_vector(7 downto 0);
signal a4:std_logic_vector(7 downto 0);
signal b1:std_logic_vector(7 downto 0);
signal b2:std_logic_vector(7 downto 0);
signal b3:std_logic_vector(7 downto 0);
signal b4:std_logic_vector(7 downto 0);
signal clk:std_logic;
begin
ss:rotate port
map(a1=>a1,a2=>a2,a3=>a3,a4=>a4,b1=>b1,b2=>b2,b3=>b3,b4=>b4,clk=>clk);
process
begin
a1<=X"67";
a2<=X"20";
a3<=X"46";
a4<=X"75";
wait for 1000ns;
end process;
process
begin
```

```
clk<='0';
wait for 10ns;
clk<='1';
wait for 10ns;
end process;
end Behavioral;
```

## Simulation Result

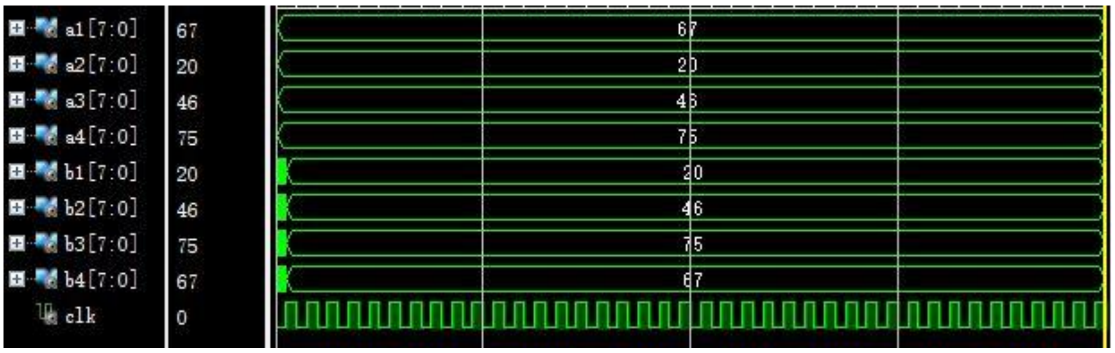


Figure1

Figure1 shows the simulation result of the rotator.

## 3.1.2 S box

### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity s_box is
	Port ( a : in STD_LOGIC_VECTOR (7 downto 0);
			b : out STD_LOGIC_VECTOR (7 downto 0);
			clk: in STD_LOGIC);
end s_box;

architecture Behavioral of s_box is
begin
process(clk)
begin
case a is
				when X"00"=>b<=X"63";
				when X"01"=>b<=X"7C";
				when X"02"=>b<=X"77";
				when X"03"=>b<=X"7B";
				when X"04"=>b<=X"F2";
				when X"05"=>b<=X"6B";
				when X"06"=>b<=X"6F";
				when X"07"=>b<=X"C5";
				when X"08"=>b<=X"30";
				when X"09"=>b<=X"01";
				when X"0A"=>b<=X"67";
				when X"0B"=>b<=X"2B";
				when X"0C"=>b<=X"FE";
				when X"0D"=>b<=X"D7";
				when X"0E"=>b<=X"AB";
				when X"0F"=>b<=X"76";
				when X"10"=>b<=X"CA";
				when X"11"=>b<=X"82";
				when X"12"=>b<=X"C9";
				when X"13"=>b<=X"7D";
				when X"14"=>b<=X"FA";
				when X"15"=>b<=X"59";
				when X"16"=>b<=X"47";
				when X"17"=>b<=X"F0";
				when X"18"=>b<=X"AD";
when X"19"=>b<=X"D4";
when X"1A"=>b<=X"A2";
when X"1B"=>b<=X"AF";
when X"1C"=>b<=X"9C";
when X"1D"=>b<=X"A4";
when X"1E"=>b<=X"72";
when X"1F"=>b<=X"C0";
when X"20"=>b<=X"B7";
when X"21"=>b<=X"FD";
when X"22"=>b<=X"93";
when X"23"=>b<=X"26";
when X"24"=>b<=X"36";
when X"25"=>b<=X"3F";
when X"26"=>b<=X"F7";
when X"27"=>b<=X"CC";
when X"28"=>b<=X"34";
when X"29"=>b<=X"A5";
when X"2A"=>b<=X"E5";
when X"2B"=>b<=X"F1";
when X"2C"=>b<=X"71";
when X"2D"=>b<=X"D8";
when X"2E"=>b<=X"31";
when X"2F"=>b<=X"15";
when X"30"=>b<=X"04";
when X"31"=>b<=X"C7";
when X"32"=>b<=X"23";
when X"33"=>b<=X"C3";
when X"34"=>b<=X"18";
when X"35"=>b<=X"96";
when X"36"=>b<=X"05";
when X"37"=>b<=X"9A";
when X"38"=>b<=X"07";
when X"39"=>b<=X"12";
when X"3A"=>b<=X"80";
when X"3B"=>b<=X"E2";
when X"3C"=>b<=X"EB";
when X"3D"=>b<=X"27";
when X"3E"=>b<=X"B2";
when X"3F"=>b<=X"75";
when X"40"=>b<=X"09";
when X"41"=>b<=X"83";
when X"42"=>b<=X"2C";
when X"43"=>b<=X"1A";
when X"44"=>b<=X"1B";
when X"45"=>b<=X"6E";
when X"46"=>b<=X"5A";
when X"47"=>b<=X"A0";
when X"48"=>b<=X"52";
when X"49"=>b<=X"3B";
when X"4A"=>b<=X"D6";
when X"4B"=>b<=X"B3";
when X"4C"=>b<=X"29";
when X"4D"=>b<=X"E3";
when X"4E"=>b<=X"2F";
when X"4F"=>b<=X"84";
when X"50"=>b<=X"53";
when X"51"=>b<=X"D1";
when X"52"=>b<=X"00";
when X"53"=>b<=X"ED";
when X"54"=>b<=X"20";
when X"55"=>b<=X"FC";
when X"56"=>b<=X"B1";
when X"57"=>b<=X"5B";
when X"58"=>b<=X"6A";
when X"59"=>b<=X"CB";
when X"5A"=>b<=X"BE";
when X"5B"=>b<=X"39";
when X"5C"=>b<=X"4A";
when X"5D"=>b<=X"4C";
when X"5E"=>b<=X"58";
when X"5F"=>b<=X"CF";
when X"60"=>b<=X"D0";
when X"61"=>b<=X"EF";
when X"62"=>b<=X"AA";
when X"63"=>b<=X"FB";
when X"64"=>b<=X"43";
when X"65"=>b<=X"4D";
when X"66"=>b<=X"33";
when X"67"=>b<=X"85";
when X"68"=>b<=X"45";
when X"69"=>b<=X"F9";
when X"6A"=>b<=X"02";
when X"6B"=>b<=X"7F";
when X"6C"=>b<=X"50";
when X"6D"=>b<=X"3C";
when X"6E"=>b<=X"9F";
when X"6F"=>b<=X"A8";
when X"70"=>b<=X"51";
when X"71"=>b<=X"A3";
when X"72"=>b<=X"40";
when X"73"=>b<=X"8F";
when X"74"=>b<=X"92";
when X"75"=>b<=X"9D";
when X"76"=>b<=X"38";
when X"77"=>b<=X"F5";
when X"78"=>b<=X"BC";
when X"79"=>b<=X"B6";
when X"7A"=>b<=X"DA";
when X"7B"=>b<=X"21";
when X"7C"=>b<=X"10";
when X"7D"=>b<=X"FF";
when X"7E"=>b<=X"F3";
when X"7F"=>b<=X"D2";
when X"80"=>b<=X"CD";
when X"81"=>b<=X"0C";
when X"82"=>b<=X"13";
when X"83"=>b<=X"EC";
when X"84"=>b<=X"5F";
when X"85"=>b<=X"97";
when X"86"=>b<=X"44";
when X"87"=>b<=X"17";
when X"88"=>b<=X"C4";
when X"89"=>b<=X"A7";
when X"8A"=>b<=X"7E";
when X"8B"=>b<=X"3D";
when X"8C"=>b<=X"64";
when X"8D"=>b<=X"5D";
when X"8E"=>b<=X"19";
when X"8F"=>b<=X"73";
when X"90"=>b<=X"60";
when X"91"=>b<=X"81";
when X"92"=>b<=X"4F";
when X"93"=>b<=X"DC";
when X"94"=>b<=X"22";
when X"95"=>b<=X"2A";
when X"96"=>b<=X"90";
when X"97"=>b<=X"88";
when X"98"=>b<=X"46";
when X"99"=>b<=X"EE";
when X"9A"=>b<=X"B8";
when X"9B"=>b<=X"14";
when X"9C"=>b<=X"DE";
when X"9D"=>b<=X"5E";
when X"9E"=>b<=X"0B";
when X"9F"=>b<=X"DB";
when X"A0"=>b<=X"E0";
when X"A1"=>b<=X"32";
when X"A2"=>b<=X"3A";
when X"A3"=>b<=X"0A";
when X"A4"=>b<=X"49";
when X"A5"=>b<=X"06";
when X"A6"=>b<=X"24";
when X"A7"=>b<=X"5C";
when X"A8"=>b<=X"C2";
when X"A9"=>b<=X"D3";
when X"AA"=>b<=X"AC";
when X"AB"=>b<=X"62";
when X"AC"=>b<=X"91";
when X"AD"=>b<=X"95";
when X"AE"=>b<=X"E4";
when X"AF"=>b<=X"79";
when X"B0"=>b<=X"E7";
when X"B1"=>b<=X"C8";
when X"B2"=>b<=X"37";
when X"B3"=>b<=X"6D";
when X"B4"=>b<=X"8D";
when X"B5"=>b<=X"D5";
when X"B6"=>b<=X"4E";
when X"B7"=>b<=X"A9";
when X"B8"=>b<=X"6C";
when X"B9"=>b<=X"56";
when X"BA"=>b<=X"F4";
when X"BB"=>b<=X"EA";
when X"BC"=>b<=X"65";
when X"BD"=>b<=X"7A";
when X"BE"=>b<=X"AE";
when X"BF"=>b<=X"08";
when X"C0"=>b<=X"BA";
when X"C1"=>b<=X"78";
when X"C2"=>b<=X"25";
when X"C3"=>b<=X"2E";
when X"C4"=>b<=X"1C";
when X"C5"=>b<=X"A6";
when X"C6"=>b<=X"B4";
when X"C7"=>b<=X"C6";
when X"C8"=>b<=X"E8";
when X"C9"=>b<=X"DD";
when X"CA"=>b<=X"74";
when X"CB"=>b<=X"1F";
when X"CC"=>b<=X"4B";
when X"CD"=>b<=X"BD";
when X"CE"=>b<=X"8B";
when X"CF"=>b<=X"8A";
when X"D0"=>b<=X"70";
when X"D1"=>b<=X"3E";
when X"D2"=>b<=X"B5";
when X"D3"=>b<=X"66";
when X"D4"=>b<=X"48";
when X"D5"=>b<=X"03";
when X"D6"=>b<=X"F6";
when X"D7"=>b<=X"0E";
when X"D8"=>b<=X"61";
when X"D9"=>b<=X"35";
when X"DA"=>b<=X"57";
when X"DB"=>b<=X"B9";
when X"DC"=>b<=X"86";
when X"DD"=>b<=X"C1";
when X"DE"=>b<=X"1D";
when X"DF"=>b<=X"9E";
when X"E0"=>b<=X"E1";
when X"E1"=>b<=X"F8";
when X"E2"=>b<=X"98";
when X"E3"=>b<=X"11";
when X"E4"=>b<=X"69";
when X"E5"=>b<=X"D9";
when X"E6"=>b<=X"8E";
when X"E7"=>b<=X"94";
when X"E8"=>b<=X"9B";
when X"E9"=>b<=X"1E";
when X"EA"=>b<=X"87";
when X"EB"=>b<=X"E9";
when X"EC"=>b<=X"CE";
when X"ED"=>b<=X"55";
when X"EE"=>b<=X"28";
when X"EF"=>b<=X"DF";
when X"F0"=>b<=X"8C";
when X"F1"=>b<=X"A1";
when X"F2"=>b<=X"89";
when X"F3"=>b<=X"0D";
when X"F4"=>b<=X"BF";
```

```
                    when X"F5"=>b<=X"E6";
                    when X"F6"=>b<=X"42";
                    when X"F7"=>b<=X"68";
                    when X"F8"=>b<=X"41";
                    when X"F9"=>b<=X"99";
                    when X"FA"=>b<=X"2D";
                    when X"FB"=>b<=X"0F";
                    when X"FC"=>b<=X"B0";
                    when X"FD"=>b<=X"54";
                    when X"FE"=>b<=X"BB";
                    when X"FF"=>b<=X"16";
                    when others=>b<=X"00";
                    end case;
end process;
end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity test_sbox is
--    Port ( );
end test_sbox;

architecture Behavioral of test_sbox is
component s_box is
        Port ( a : in STD_LOGIC_VECTOR (7 downto 0);
                        clk:in std_logic;
              b : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal a:std_logic_vector(7 downto 0);
signal b:std_logic_vector(7 downto 0);
signal clk:std_logic;
begin
uss:s_box port map(a=>a,b=>b,clk=>clk);
process
begin
a<=X"00";
wait for 5ns;
while(true) loop
        a<=a+X"01";
        wait for 5ns;
end loop;
end process;
process
begin
clk<='0';
wait for 5ns;
clk<='1';
wait for 5ns;
end process;
end Behavioral;
```

## Simulation Result

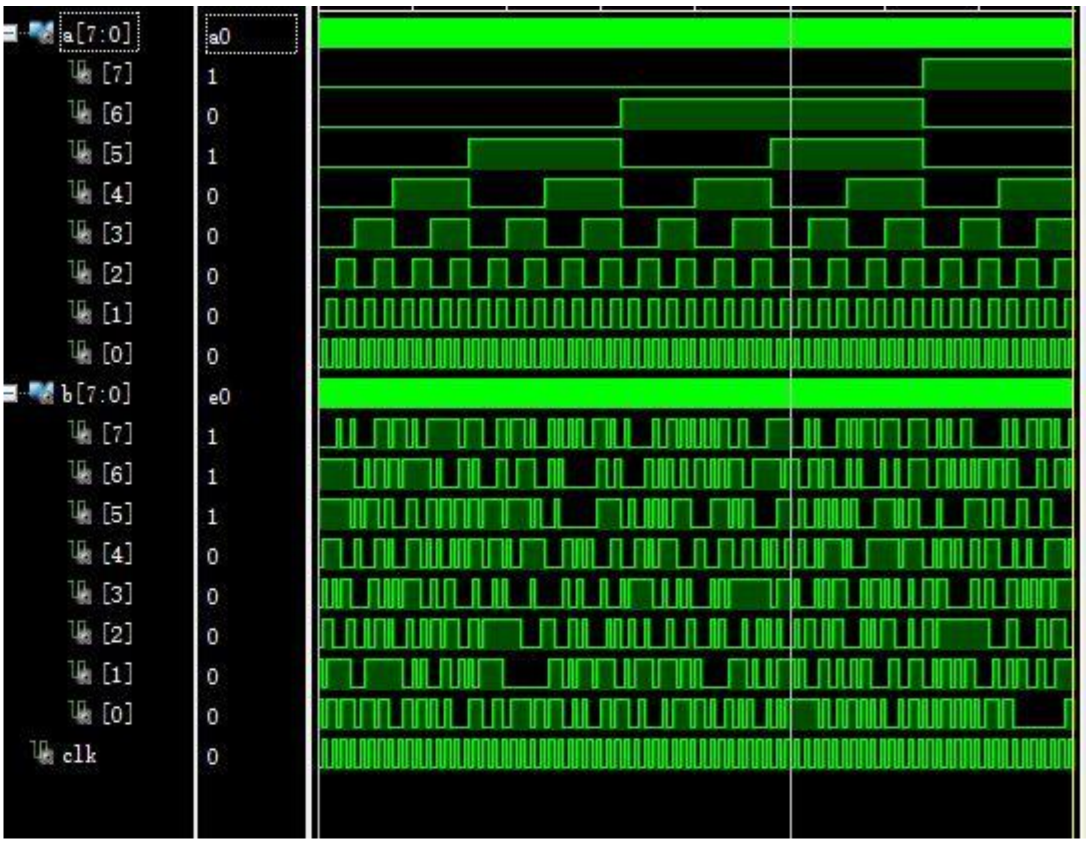


Figure2

Figure2 shows the simulation result of the s_box.

### 3.1.3 Key

#### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity key is
    Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
           a2 : in STD_LOGIC_VECTOR (7 downto 0);
           a3 : in STD_LOGIC_VECTOR (7 downto 0);
           a4 : in STD_LOGIC_VECTOR (7 downto 0);
           a5 : in STD_LOGIC_VECTOR (7 downto 0);
           a6 : in STD_LOGIC_VECTOR (7 downto 0);
           a7 : in STD_LOGIC_VECTOR (7 downto 0);
           a8 : in STD_LOGIC_VECTOR (7 downto 0);
           a9 : in STD_LOGIC_VECTOR (7 downto 0);
           a10 : in STD_LOGIC_VECTOR (7 downto 0);
           a11 : in STD_LOGIC_VECTOR (7 downto 0);
           a12 : in STD_LOGIC_VECTOR (7 downto 0);
           a13 : in STD_LOGIC_VECTOR (7 downto 0);
           a14 : in STD_LOGIC_VECTOR (7 downto 0);
           a15 : in STD_LOGIC_VECTOR (7 downto 0);
           a16 : in STD_LOGIC_VECTOR (7 downto 0);
           b1 : out STD_LOGIC_VECTOR (7 downto 0);
           b2 : out STD_LOGIC_VECTOR (7 downto 0);
           b3 : out STD_LOGIC_VECTOR (7 downto 0);
           b4 : out STD_LOGIC_VECTOR (7 downto 0);
           b5 : out STD_LOGIC_VECTOR (7 downto 0);
           b6 : out STD_LOGIC_VECTOR (7 downto 0);
           b7 : out STD_LOGIC_VECTOR (7 downto 0);
           b8 : out STD_LOGIC_VECTOR (7 downto 0);
           b9 : out STD_LOGIC_VECTOR (7 downto 0);
           b10 : out STD_LOGIC_VECTOR (7 downto 0);
           b11 : out STD_LOGIC_VECTOR (7 downto 0);
           b12 : out STD_LOGIC_VECTOR (7 downto 0);
           b13 : out STD_LOGIC_VECTOR (7 downto 0);
           b14 : out STD_LOGIC_VECTOR (7 downto 0);
           b15 : out STD_LOGIC_VECTOR (7 downto 0);
           b16 : out STD_LOGIC_VECTOR (7 downto 0);
           clk : in STD_LOGIC;
           round_num:in std_logic_vector(7 downto 0));
end key;
```

```
architecture Behavioral of key is
component rotate is
        Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
              a2 : in STD_LOGIC_VECTOR (7 downto 0);
              a3 : in STD_LOGIC_VECTOR (7 downto 0);
              a4 : in STD_LOGIC_VECTOR (7 downto 0);
              clk:in STD_LOGIC;
              b1 : out STD_LOGIC_VECTOR (7 downto 0);
              b2 : out STD_LOGIC_VECTOR (7 downto 0);
              b3 : out STD_LOGIC_VECTOR (7 downto 0);
              b4 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
component s_box is
        Port ( a : in STD_LOGIC_VECTOR (7 downto 0);
              b : out STD_LOGIC_VECTOR (7 downto 0);
              clk:std_logic);
end component;
component xor8 is
        Port ( a : in STD_LOGIC_VECTOR (7 downto 0);
              b : in STD_LOGIC_VECTOR (7 downto 0);
              c : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal tempa1:std_logic_vector(7 downto 0);
signal tempa2:std_logic_vector(7 downto 0);
signal tempa3:std_logic_vector(7 downto 0);
signal tempa4:std_logic_vector(7 downto 0);
signal tempb1:std_logic_vector(7 downto 0);
signal tempb2:std_logic_vector(7 downto 0);
signal tempb3:std_logic_vector(7 downto 0);
signal tempb4:std_logic_vector(7 downto 0);
signal tempc1:std_logic_vector(7 downto 0);
signal out1:std_logic_vector(7 downto 0);
signal out2:std_logic_vector(7 downto 0);
signal out3:std_logic_vector(7 downto 0);
signal out4:std_logic_vector(7 downto 0);
signal out5:std_logic_vector(7 downto 0);
signal out6:std_logic_vector(7 downto 0);
signal out7:std_logic_vector(7 downto 0);
signal out8:std_logic_vector(7 downto 0);
signal out9:std_logic_vector(7 downto 0);
signal out10:std_logic_vector(7 downto 0);
signal out11:std_logic_vector(7 downto 0);
signal out12:std_logic_vector(7 downto 0);
signal out13:std_logic_vector(7 downto 0);
signal out14:std_logic_vector(7 downto 0);
signal out15:std_logic_vector(7 downto 0);
signal out16:std_logic_vector(7 downto 0);
begin
rot:rotate port
map(a1=>a13,a2=>a14,a3=>a15,a4=>a16,clk=>clk,b1=>tempa1,b2=>tempa2,b3=>temp
a3,b4=>tempa4);
s_box1:s_box port map(clk=>clk,a=>tempa1,b=>tempb1);
s_box2:s_box port map(clk=>clk,a=>tempa2,b=>tempb2);
s_box3:s_box port map(clk=>clk,a=>tempa3,b=>tempb3);
s_box4:s_box port map(clk=>clk,a=>tempa4,b=>tempb4);
xor8s:xor8 port map(a=>round_num,b=>tempb1,c=>tempc1);
out1<=(a1 xor tempc1);
out2<=(a2 xor tempb2);
out3<=(a3 xor tempb3);
out4<=(a4 xor tempb4);
out5<=(out1 xor a5);
out6<=(out2 xor a6);
out7<=(out3 xor a7);
out8<=(out4 xor a8);
out9<=(out5 xor a9);
out10<=(out6 xor a10);
out11<=(out7 xor a11);
out12<=(out8 xor a12);
out13<=(out9 xor a13);
out14<=(out10 xor a14);
out15<=(out11 xor a15);
out16<=(out12 xor a16);
b1<=out1;b2<=out2;b3<=out3;b4<=out4;
b5<=out5;b6<=out6;b7<=out7;b8<=out8;
b9<=out9;b10<=out10;b11<=out11;b12<=out12;
b13<=out13;b14<=out14;b15<=out15;b16<=out16;
end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_key is
--    Port ( );
end test_key;

architecture Behavioral of test_key is
component key is
      Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
           a2 : in STD_LOGIC_VECTOR (7 downto 0);
           a3 : in STD_LOGIC_VECTOR (7 downto 0);
           a4 : in STD_LOGIC_VECTOR (7 downto 0);
           a5 : in STD_LOGIC_VECTOR (7 downto 0);
           a6 : in STD_LOGIC_VECTOR (7 downto 0);
           a7 : in STD_LOGIC_VECTOR (7 downto 0);
           a8 : in STD_LOGIC_VECTOR (7 downto 0);
           a9 : in STD_LOGIC_VECTOR (7 downto 0);
           a10 : in STD_LOGIC_VECTOR (7 downto 0);
           a11 : in STD_LOGIC_VECTOR (7 downto 0);
           a12 : in STD_LOGIC_VECTOR (7 downto 0);
           a13 : in STD_LOGIC_VECTOR (7 downto 0);
           a14 : in STD_LOGIC_VECTOR (7 downto 0);
           a15 : in STD_LOGIC_VECTOR (7 downto 0);
           a16 : in STD_LOGIC_VECTOR (7 downto 0);
           b1 : out STD_LOGIC_VECTOR (7 downto 0);
           b2 : out STD_LOGIC_VECTOR (7 downto 0);
           b3 : out STD_LOGIC_VECTOR (7 downto 0);
           b4 : out STD_LOGIC_VECTOR (7 downto 0);
           b5 : out STD_LOGIC_VECTOR (7 downto 0);
           b6 : out STD_LOGIC_VECTOR (7 downto 0);
           b7 : out STD_LOGIC_VECTOR (7 downto 0);
           b8 : out STD_LOGIC_VECTOR (7 downto 0);
           b9 : out STD_LOGIC_VECTOR (7 downto 0);
           b10 : out STD_LOGIC_VECTOR (7 downto 0);
           b11 : out STD_LOGIC_VECTOR (7 downto 0);
           b12 : out STD_LOGIC_VECTOR (7 downto 0);
           b13 : out STD_LOGIC_VECTOR (7 downto 0);
           b14 : out STD_LOGIC_VECTOR (7 downto 0);
           b15 : out STD_LOGIC_VECTOR (7 downto 0);
           b16 : out STD_LOGIC_VECTOR (7 downto 0);
           clk : in STD_LOGIC;
```

```
        round_num:in std_logic_vector(7 downto 0));
end component;
signal
a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,
b12,b13,b14,b15,b16,round_num:std_logic_vector(7 downto 0);
signal clk :std_logic;
begin
testkey:key port
map(a1=>a1,a2=>a2,a3=>a3,a4=>a4,a5=>a5,a6=>a6,a7=>a7,a8=>a8,a9=>a9,a10=>a10,
a11=>a11,a12=>a12,a13=>a13,a14=>a14,a15=>a15,a16=>a16,b1=>b1,b2=>b2,b3=>b3,
b4=>b4,b5=>b5,b6=>b6,b7=>b7,b8=>b8,b9=>b9,b10=>b10,b11=>b11,b12=>b12,b13=
>b13,b14=>b14,b15=>b15,b16=>b16,round_num=>round_num,clk=>clk);
process
begin
clk<='0';
wait for 50ns;
clk<='1';
wait for 50ns;
end process;
process
begin
round_num<=X"1b";
a1<=X"8e";
a2<=X"51";
a3<=X"ef";
a4<=X"21";
a5<=X"fa";
a6<=X"bb";
a7<=X"45";
a8<=X"22";
a9<=X"e4";
a10<=X"3d";
a11<=X"7a";
a12<=X"06";
a13<=X"56";
a14<=X"95";
a15<=X"4b";
a16<=X"6c";
wait for 100ns;
end process;
end Behavioral;
```

## Simulation Result

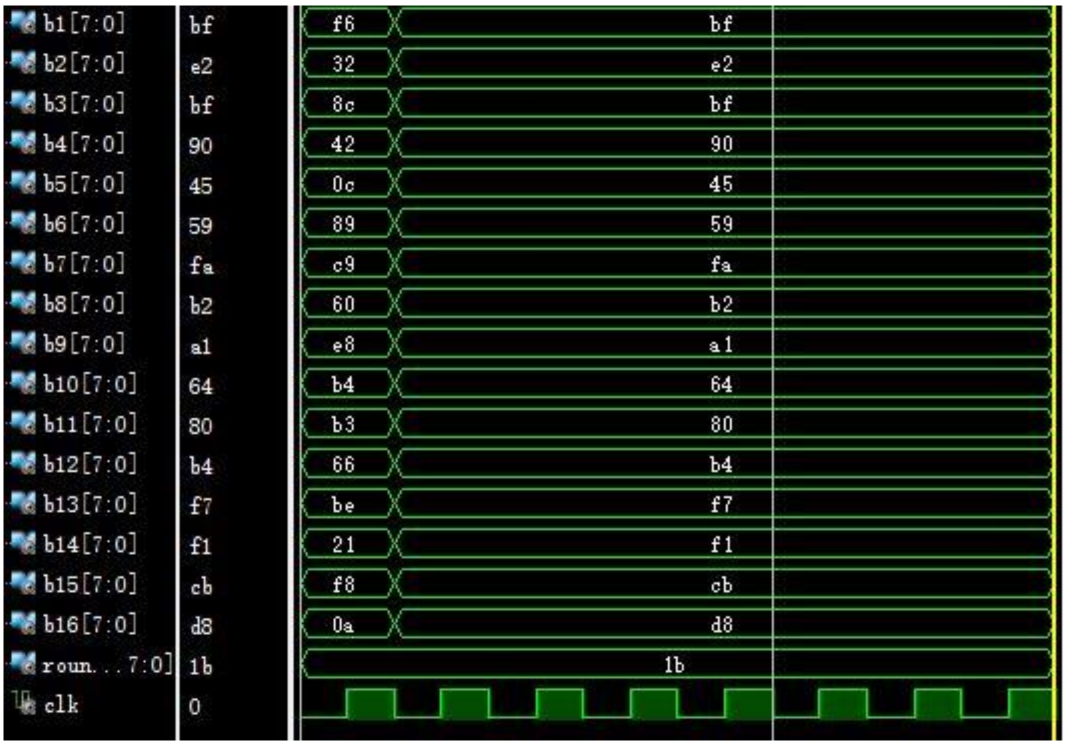


Figure3

Figure3 shows the simulation result of the key.

## 3.2 Message Part

### 3.2.1 Add round key

#### Code

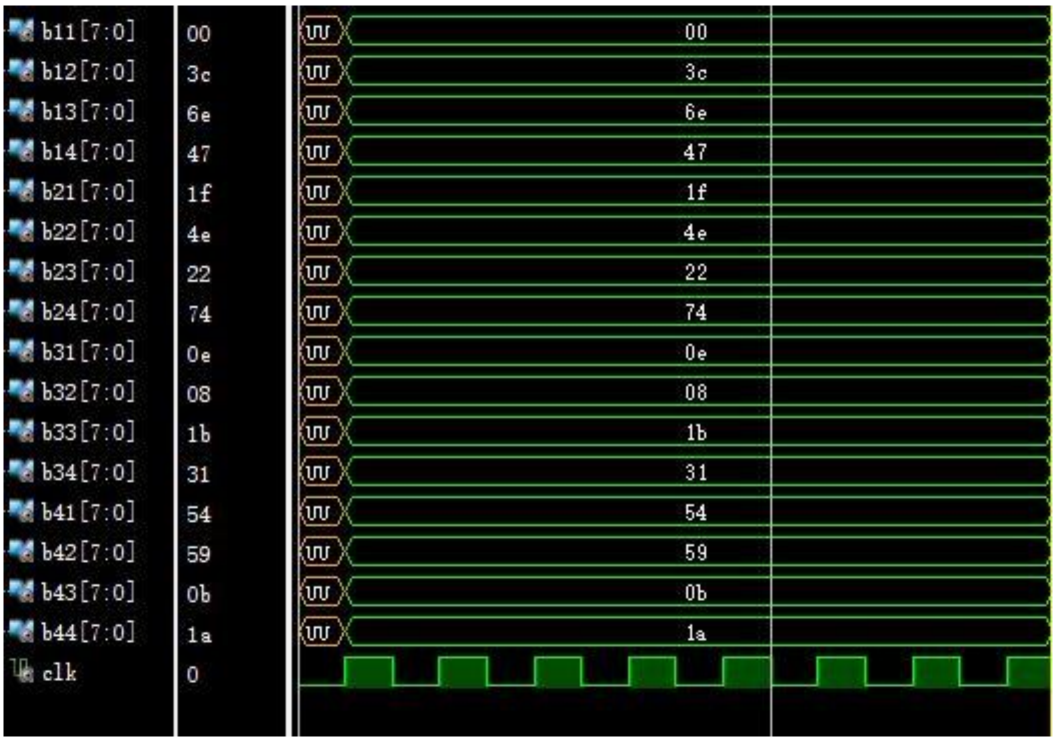
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity add_round_key is
	Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end add_round_key;

architecture Behavioral of add_round_key is

begin
process(clk)
begin
if clk='1' and clk'event then
b11<=a11 xor key11;b12<=a12 xor key12;b13<=a13 xor key13;b14<=a14 xor key14;
b21<=a21 xor key21;b22<=a22 xor key22;b23<=a23 xor key23;b24<=a24 xor key24;
b31<=a31 xor key31;b32<=a32 xor key32;b33<=a33 xor key33;b34<=a34 xor key34;
b41<=a41 xor key41;b42<=a42 xor key42;b43<=a43 xor key43;b44<=a44 xor key44;
end if;
end process;

end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_addroundkey is
--    Port ( );
end test_addroundkey;

architecture Behavioral of test_addroundkey is
component add_round_key is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal
a11,a12,a13,a14,a21,a22,a23,a24,a31,a32,a33,a34,a41,a42,a43,a44,key11,key12,key13,key14,
key21,key22,key23,key24,key31,key32,key33,key34,key41,key42,key43,key44,b11,b12,b13,b1
4,b21,b22,b23,b24,b31,b32,b33,b34,b41,b42,b43,b44:std_logic_vector(7 downto 0);
signal clk:std_logic;
begin
ark:add_round_key port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,key1
1=>key11,key12=>key12,key13=>key13,key14=>key14,key21=>key21,key22=>key22,key2
3=>key23,key24=>key24,key31=>key31,key32=>key32,key33=>key33,key34=>key34,key4
1=>key41,key42=>key42,key43=>key43,key44=>key44,b11=>b11,b12=>b12,b13=>b13,b
14=>b14,b21=>b21,b22=>b22,b23=>b23,b24=>b24,b31=>b31,b32=>b32,b33=>b33,b34
=>b34,b41=>b41,b42=>b42,b43=>b43,b44=>b44,clk=>clk);
process
begin
clk<='0';
wait for 50ns;
clk<='1';
wait for 50ns;
end process;
process
begin
a11<=x"54";a12<=x"4f";a13<=x"4e";a14<=x"20";
a21<=x"77";a22<=x"6e";a23<=x"69";a24<=x"54";
a31<=x"6f";a32<=x"65";a33<=x"6e";a34<=x"77";
a41<=x"20";a42<=x"20";a43<=x"65";a44<=x"6f";
```

```
key11<=x"54";key12<=x"73";key13<=x"20";key14<=x"67";
key21<=x"68";key22<=x"20";key23<=x"4b";key24<=x"20";
key31<=x"61";key32<=x"6d";key33<=x"75";key34<=x"46";
key41<=x"74";key42<=x"79";key43<=x"6e";key44<=x"75";
wait for 500ns;
end process;
end Behavioral;
```

## Simulation Result



```
b11[7:0]  00
b12[7:0]  3c
b13[7:0]  6e
b14[7:0]  47
b21[7:0]  1f
b22[7:0]  4e
b23[7:0]  22
b24[7:0]  74
b31[7:0]  0e
b32[7:0]  08
b33[7:0]  1b
b34[7:0]  31
b41[7:0]  54
b42[7:0]  59
b43[7:0]  0b
b44[7:0]  1a
clk       0
```

Figure4

Figure4 shows the simulation result of the add round key.

### 3.2.2 Sub

#### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity sub is
        Port ( clk : in STD_LOGIC;
              a11 : in STD_LOGIC_VECTOR (7 downto 0);
              a12 : in STD_LOGIC_VECTOR (7 downto 0);
              a13 : in STD_LOGIC_VECTOR (7 downto 0);
              a14 : in STD_LOGIC_VECTOR (7 downto 0);
              a21 : in STD_LOGIC_VECTOR (7 downto 0);
              a22 : in STD_LOGIC_VECTOR (7 downto 0);
              a23 : in STD_LOGIC_VECTOR (7 downto 0);
              a24 : in STD_LOGIC_VECTOR (7 downto 0);
              a31 : in STD_LOGIC_VECTOR (7 downto 0);
              a32 : in STD_LOGIC_VECTOR (7 downto 0);
              a33 : in STD_LOGIC_VECTOR (7 downto 0);
              a34 : in STD_LOGIC_VECTOR (7 downto 0);
              a41 : in STD_LOGIC_VECTOR (7 downto 0);
              a42 : in STD_LOGIC_VECTOR (7 downto 0);
              a43 : in STD_LOGIC_VECTOR (7 downto 0);
              a44 : in STD_LOGIC_VECTOR (7 downto 0);
              b11 : out STD_LOGIC_VECTOR (7 downto 0);
              b12 : out STD_LOGIC_VECTOR (7 downto 0);
              b13 : out STD_LOGIC_VECTOR (7 downto 0);
              b14 : out STD_LOGIC_VECTOR (7 downto 0);
              b21 : out STD_LOGIC_VECTOR (7 downto 0);
              b22 : out STD_LOGIC_VECTOR (7 downto 0);
              b23 : out STD_LOGIC_VECTOR (7 downto 0);
              b24 : out STD_LOGIC_VECTOR (7 downto 0);
              b31 : out STD_LOGIC_VECTOR (7 downto 0);
              b32 : out STD_LOGIC_VECTOR (7 downto 0);
              b33 : out STD_LOGIC_VECTOR (7 downto 0);
              b34 : out STD_LOGIC_VECTOR (7 downto 0);
              b41 : out STD_LOGIC_VECTOR (7 downto 0);
              b42 : out STD_LOGIC_VECTOR (7 downto 0);
              b43 : out STD_LOGIC_VECTOR (7 downto 0);
              b44 : out STD_LOGIC_VECTOR (7 downto 0));
end sub;

architecture Behavioral of sub is
```

```
component s_box is
        Port ( a : in STD_LOGIC_VECTOR (7 downto 0);
              b : out STD_LOGIC_VECTOR (7 downto 0);
              clk: in STD_LOGIC);
end component;
signal temp11:std_logic_vector(7 downto 0);
signal temp12:std_logic_vector(7 downto 0);
signal temp13:std_logic_vector(7 downto 0);
signal temp14:std_logic_vector(7 downto 0);
signal temp21:std_logic_vector(7 downto 0);
signal temp22:std_logic_vector(7 downto 0);
signal temp23:std_logic_vector(7 downto 0);
signal temp24:std_logic_vector(7 downto 0);
signal temp31:std_logic_vector(7 downto 0);
signal temp32:std_logic_vector(7 downto 0);
signal temp33:std_logic_vector(7 downto 0);
signal temp34:std_logic_vector(7 downto 0);
signal temp41:std_logic_vector(7 downto 0);
signal temp42:std_logic_vector(7 downto 0);
signal temp43:std_logic_vector(7 downto 0);
signal temp44:std_logic_vector(7 downto 0);
begin
sub11:s_box port map(a=>a11,b=>temp11,clk=>clk);
sub12:s_box port map(a=>a12,b=>temp12,clk=>clk);
sub13:s_box port map(a=>a13,b=>temp13,clk=>clk);
sub14:s_box port map(a=>a14,b=>temp14,clk=>clk);
sub21:s_box port map(a=>a21,b=>temp21,clk=>clk);
sub22:s_box port map(a=>a22,b=>temp22,clk=>clk);
sub23:s_box port map(a=>a23,b=>temp23,clk=>clk);
sub24:s_box port map(a=>a24,b=>temp24,clk=>clk);
sub31:s_box port map(a=>a31,b=>temp31,clk=>clk);
sub32:s_box port map(a=>a32,b=>temp32,clk=>clk);
sub33:s_box port map(a=>a33,b=>temp33,clk=>clk);
sub34:s_box port map(a=>a34,b=>temp34,clk=>clk);
sub41:s_box port map(a=>a41,b=>temp41,clk=>clk);
sub42:s_box port map(a=>a42,b=>temp42,clk=>clk);
sub43:s_box port map(a=>a43,b=>temp43,clk=>clk);
sub44:s_box port map(a=>a44,b=>temp44,clk=>clk);
process(clk)
begin
if clk='1' and clk'event then
b11<=temp11;b12<=temp12;b13<=temp13;b14<=temp14;
b21<=temp21;b22<=temp22;b23<=temp23;b24<=temp24;
b31<=temp31;b32<=temp32;b33<=temp33;b34<=temp34;
```

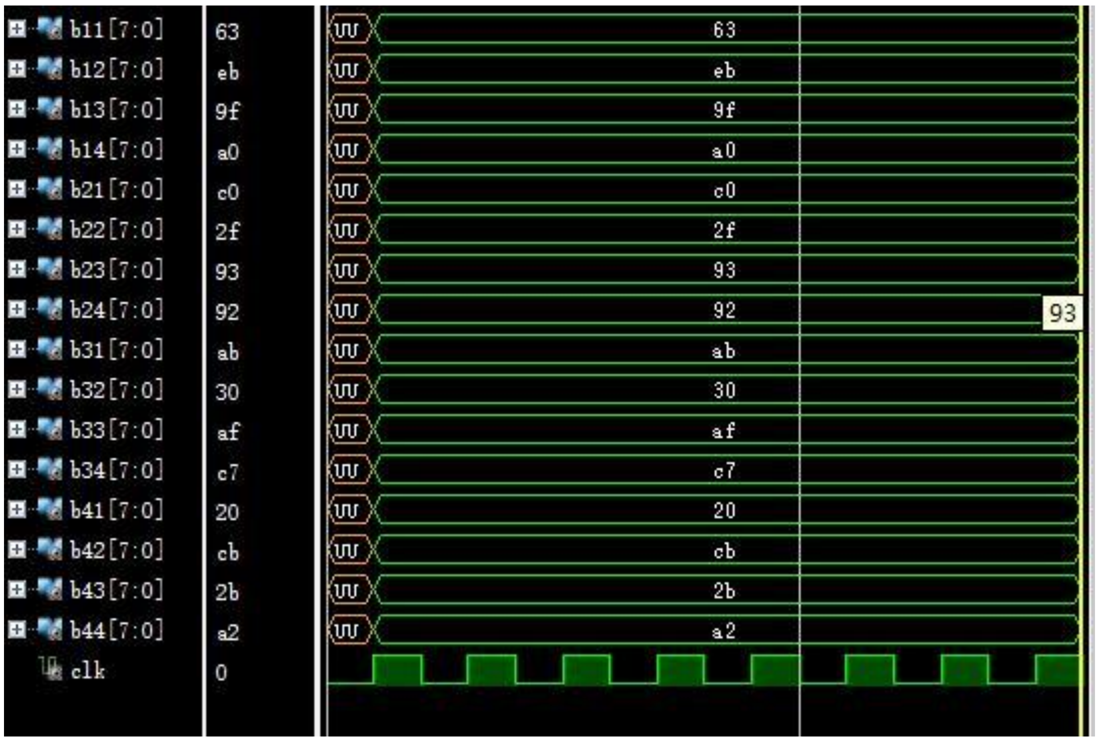
```
b41<=temp41;b42<=temp42;b43<=temp43;b44<=temp44;
end if;
end process;
end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_sub is
--    Port ( );
end test_sub;

architecture Behavioral of test_sub is
component sub is
        Port ( clk : in STD_LOGIC;
      a11 : in STD_LOGIC_VECTOR (7 downto 0);
      a12 : in STD_LOGIC_VECTOR (7 downto 0);
      a13 : in STD_LOGIC_VECTOR (7 downto 0);
      a14 : in STD_LOGIC_VECTOR (7 downto 0);
      a21 : in STD_LOGIC_VECTOR (7 downto 0);
      a22 : in STD_LOGIC_VECTOR (7 downto 0);
      a23 : in STD_LOGIC_VECTOR (7 downto 0);
      a24 : in STD_LOGIC_VECTOR (7 downto 0);
      a31 : in STD_LOGIC_VECTOR (7 downto 0);
      a32 : in STD_LOGIC_VECTOR (7 downto 0);
      a33 : in STD_LOGIC_VECTOR (7 downto 0);
      a34 : in STD_LOGIC_VECTOR (7 downto 0);
      a41 : in STD_LOGIC_VECTOR (7 downto 0);
      a42 : in STD_LOGIC_VECTOR (7 downto 0);
      a43 : in STD_LOGIC_VECTOR (7 downto 0);
      a44 : in STD_LOGIC_VECTOR (7 downto 0);
      b11 : out STD_LOGIC_VECTOR (7 downto 0);
      b12 : out STD_LOGIC_VECTOR (7 downto 0);
      b13 : out STD_LOGIC_VECTOR (7 downto 0);
      b14 : out STD_LOGIC_VECTOR (7 downto 0);
      b21 : out STD_LOGIC_VECTOR (7 downto 0);
      b22 : out STD_LOGIC_VECTOR (7 downto 0);
      b23 : out STD_LOGIC_VECTOR (7 downto 0);
      b24 : out STD_LOGIC_VECTOR (7 downto 0);
      b31 : out STD_LOGIC_VECTOR (7 downto 0);
      b32 : out STD_LOGIC_VECTOR (7 downto 0);
      b33 : out STD_LOGIC_VECTOR (7 downto 0);
      b34 : out STD_LOGIC_VECTOR (7 downto 0);
      b41 : out STD_LOGIC_VECTOR (7 downto 0);
      b42 : out STD_LOGIC_VECTOR (7 downto 0);
      b43 : out STD_LOGIC_VECTOR (7 downto 0);
      b44 : out STD_LOGIC_VECTOR (7 downto 0));
```

```
end component;
signal
a11,a12,a13,a14,a21,a22,a23,a24,a31,a32,a33,a34,a41,a42,a43,a44,b11,b12,b13,b14,b21,b22,
b23,b24,b31,b32,b33,b34,b41,b42,b43,b44:std_logic_vector(7 downto 0);
signal clk:std_logic;
begin
subs:sub port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,b11
=>b11,b12=>b12,b13=>b13,b14=>b14,b21=>b21,b22=>b22,b23=>b23,b24=>b24,b31=
>b31,b32=>b32,b33=>b33,b34=>b34,b41=>b41,b42=>b42,b43=>b43,b44=>b44,clk=>cl
k);
process
begin
clk<='0';
wait for 50ns;
clk<='1';
wait for 50ns;
end process;
process
begin
a11<=X"00";
a12<=X"3c";
a13<=X"6e";
a14<=X"47";
a21<=X"1f";
a22<=X"4e";
a23<=X"22";
a24<=X"74";
a31<=X"0e";
a32<=X"08";
a33<=X"1b";
a34<=X"31";
a41<=X"54";
a42<=X"59";
a43<=X"0b";
a44<=X"1a";
wait for 100ns;
end process;

end Behavioral;
```

## Simulation Result



```
b11[7:0]  63
b12[7:0]  eb
b13[7:0]  9f
b14[7:0]  a0
b21[7:0]  c0
b22[7:0]  2f
b23[7:0]  93
b24[7:0]  92
b31[7:0]  ab
b32[7:0]  30
b33[7:0]  af
b34[7:0]  c7
b41[7:0]  20
b42[7:0]  cb
b43[7:0]  2b
b44[7:0]  a2
clk       0
```

Figure5

Figure5 shows the simulation result of the sub.

## 3.2.3 Shift row

### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity shift_row is
	Port ( a11 : in STD_LOGIC_VECTOR (7 downto 0);
			a12 : in STD_LOGIC_VECTOR (7 downto 0);
			a13 : in STD_LOGIC_VECTOR (7 downto 0);
			a14 : in STD_LOGIC_VECTOR (7 downto 0);
			a21 : in STD_LOGIC_VECTOR (7 downto 0);
			a22 : in STD_LOGIC_VECTOR (7 downto 0);
			a23 : in STD_LOGIC_VECTOR (7 downto 0);
			a24 : in STD_LOGIC_VECTOR (7 downto 0);
			a31 : in STD_LOGIC_VECTOR (7 downto 0);
			a32 : in STD_LOGIC_VECTOR (7 downto 0);
			a33 : in STD_LOGIC_VECTOR (7 downto 0);
			a34 : in STD_LOGIC_VECTOR (7 downto 0);
			a41 : in STD_LOGIC_VECTOR (7 downto 0);
			a42 : in STD_LOGIC_VECTOR (7 downto 0);
			a43 : in STD_LOGIC_VECTOR (7 downto 0);
			a44 : in STD_LOGIC_VECTOR (7 downto 0);
			clk : in STD_LOGIC;
			b11 : out STD_LOGIC_VECTOR (7 downto 0);
			b12 : out STD_LOGIC_VECTOR (7 downto 0);
			b13 : out STD_LOGIC_VECTOR (7 downto 0);
			b14 : out STD_LOGIC_VECTOR (7 downto 0);
			b21 : out STD_LOGIC_VECTOR (7 downto 0);
			b22 : out STD_LOGIC_VECTOR (7 downto 0);
			b23 : out STD_LOGIC_VECTOR (7 downto 0);
			b24 : out STD_LOGIC_VECTOR (7 downto 0);
			b31 : out STD_LOGIC_VECTOR (7 downto 0);
			b32 : out STD_LOGIC_VECTOR (7 downto 0);
			b33 : out STD_LOGIC_VECTOR (7 downto 0);
			b34 : out STD_LOGIC_VECTOR (7 downto 0);
			b41 : out STD_LOGIC_VECTOR (7 downto 0);
			b42 : out STD_LOGIC_VECTOR (7 downto 0);
			b43 : out STD_LOGIC_VECTOR (7 downto 0);
			b44 : out STD_LOGIC_VECTOR (7 downto 0));
end shift_row;

architecture Behavioral of shift_row is
```

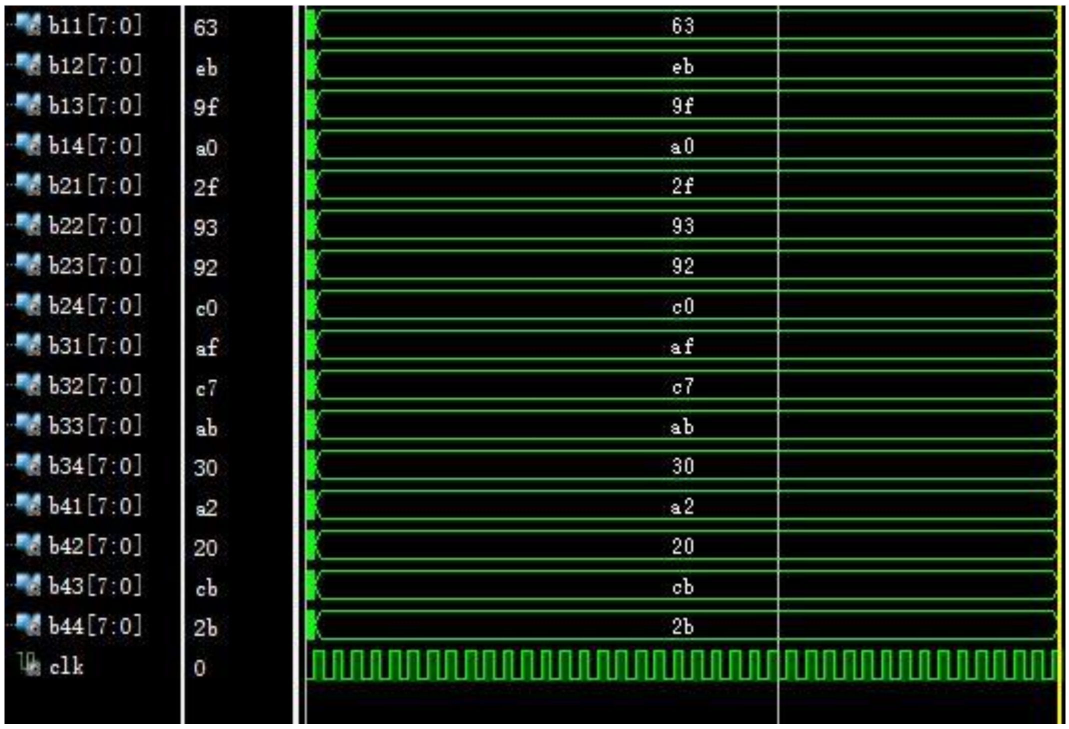
```
begin
process(clk)
begin
if clk='1' and clk'event then
b11<=a11;b12<=a12;b13<=a13;b14<=a14;
b21<=a22;b22<=a23;b23<=a24;b24<=a21;
b31<=a33;b32<=a34;b33<=a31;b34<=a32;
b41<=a44;b42<=a41;b43<=a42;b44<=a43;
end if;
end process;

end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_shiftrow is
--	Port ( );
end test_shiftrow;

architecture Behavioral of test_shiftrow is
component shift_row is
	Port ( a11 : in STD_LOGIC_VECTOR (7 downto 0);
	    a12 : in STD_LOGIC_VECTOR (7 downto 0);
	    a13 : in STD_LOGIC_VECTOR (7 downto 0);
	    a14 : in STD_LOGIC_VECTOR (7 downto 0);
	    a21 : in STD_LOGIC_VECTOR (7 downto 0);
	    a22 : in STD_LOGIC_VECTOR (7 downto 0);
	    a23 : in STD_LOGIC_VECTOR (7 downto 0);
	    a24 : in STD_LOGIC_VECTOR (7 downto 0);
	    a31 : in STD_LOGIC_VECTOR (7 downto 0);
	    a32 : in STD_LOGIC_VECTOR (7 downto 0);
	    a33 : in STD_LOGIC_VECTOR (7 downto 0);
	    a34 : in STD_LOGIC_VECTOR (7 downto 0);
	    a41 : in STD_LOGIC_VECTOR (7 downto 0);
	    a42 : in STD_LOGIC_VECTOR (7 downto 0);
	    a43 : in STD_LOGIC_VECTOR (7 downto 0);
	    a44 : in STD_LOGIC_VECTOR (7 downto 0);
	    clk : in STD_LOGIC;
	    b11 : out STD_LOGIC_VECTOR (7 downto 0);
	    b12 : out STD_LOGIC_VECTOR (7 downto 0);
	    b13 : out STD_LOGIC_VECTOR (7 downto 0);
	    b14 : out STD_LOGIC_VECTOR (7 downto 0);
	    b21 : out STD_LOGIC_VECTOR (7 downto 0);
	    b22 : out STD_LOGIC_VECTOR (7 downto 0);
	    b23 : out STD_LOGIC_VECTOR (7 downto 0);
	    b24 : out STD_LOGIC_VECTOR (7 downto 0);
	    b31 : out STD_LOGIC_VECTOR (7 downto 0);
	    b32 : out STD_LOGIC_VECTOR (7 downto 0);
	    b33 : out STD_LOGIC_VECTOR (7 downto 0);
	    b34 : out STD_LOGIC_VECTOR (7 downto 0);
	    b41 : out STD_LOGIC_VECTOR (7 downto 0);
	    b42 : out STD_LOGIC_VECTOR (7 downto 0);
	    b43 : out STD_LOGIC_VECTOR (7 downto 0);
	    b44 : out STD_LOGIC_VECTOR (7 downto 0));
```

```
end component;
signal
a11,a12,a13,a14,a21,a22,a23,a24,a31,a32,a33,a34,a41,a42,a43,a44,b11,b12,b13,b14,b21,b22,
b23,b24,b31,b32,b33,b34,b41,b42,b43,b44:std_logic_vector(7 downto 0);
signal clk:std_logic;
begin
shifta:shift_row port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,b11
=>b11,b12=>b12,b13=>b13,b14=>b14,b21=>b21,b22=>b22,b23=>b23,b24=>b24,b31=
>b31,b32=>b32,b33=>b33,b34=>b34,b41=>b41,b42=>b42,b43=>b43,b44=>b44,clk=>cl
k);
process
begin
clk<='0';
wait for 10ns;
clk<='1';
wait for 10ns;
end process;
process
begin
a11<=x"63";a12<=x"eb";a13<=x"9f";a14<=x"a0";
a21<=x"c0";a22<=x"2f";a23<=x"93";a24<=x"92";
a31<=x"ab";a32<=x"30";a33<=x"af";a34<=x"c7";
a41<=x"20";a42<=x"cb";a43<=x"2b";a44<=x"a2";
wait for 500ns;
end process;
end Behavioral;
```

## Simulation Result



```
b11[7:0]  63
b12[7:0]  eb
b13[7:0]  9f
b14[7:0]  a0
b21[7:0]  2f
b22[7:0]  93
b23[7:0]  92
b24[7:0]  c0
b31[7:0]  af
b32[7:0]  c7
b33[7:0]  ab
b34[7:0]  30
b41[7:0]  a2
b42[7:0]  20
b43[7:0]  cb
b44[7:0]  2b
clk       0
```

Figure6

Figure6 shows the simulation result of the shift row.

### 3.2.4 Mix Simple

#### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mix_sim is
    Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
           a2 : in STD_LOGIC_VECTOR (7 downto 0);
           a3 : in STD_LOGIC_VECTOR (7 downto 0);
           a4 : in STD_LOGIC_VECTOR (7 downto 0);
           b1 : out STD_LOGIC_VECTOR (7 downto 0);
           b2 : out STD_LOGIC_VECTOR (7 downto 0);
           b3 : out STD_LOGIC_VECTOR (7 downto 0);
           b4 : out STD_LOGIC_VECTOR (7 downto 0));
end mix_sim;

architecture Behavioral of mix_sim is
component mul2 is
    Port ( a : in STD_LOGIC_VECTOR (7 downto 0);
       b : out STD_LOGIC_VECTOR (7 downto 0));
end component;

component mul3 is
    Port ( a : in STD_LOGIC_VECTOR (7 downto 0);
       b : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal temp11:std_logic_vector(7 downto 0);
signal temp12:std_logic_vector(7 downto 0);
signal temp22:std_logic_vector(7 downto 0);
signal temp23:std_logic_vector(7 downto 0);
signal temp33:std_logic_vector(7 downto 0);
signal temp34:std_logic_vector(7 downto 0);
signal temp41:std_logic_vector(7 downto 0);
signal temp44:std_logic_vector(7 downto 0);

begin
mul2a:mul2 port map(a=>a1,b=>temp11);
mul3a:mul3 port map(a=>a2,b=>temp12);
mul2b:mul2 port map(a=>a2,b=>temp22);
mul3b:mul3 port map(a=>a3,b=>temp23);
mul2c:mul2 port map(a=>a3,b=>temp33);
```

```
mul3c:mul3 port map(a=>a4,b=>temp34);
mul2d:mul2 port map(a=>a4,b=>temp44);
mul3d:mul3 port map(a=>a1,b=>temp41);

b1<=(temp11 xor temp12 xor a3 xor a4);
b2<=(a1 xor temp22 xor temp23 xor a4);
b3<=(a1 xor a2 xor temp33 xor temp34);
b4<=(temp41 xor a2 xor a3 xor temp44);

end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_mix_sim is
--   Port ( );
end test_mix_sim;

architecture Behavioral of test_mix_sim is
component mix_sim is
    Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
        a2 : in STD_LOGIC_VECTOR (7 downto 0);
        a3 : in STD_LOGIC_VECTOR (7 downto 0);
        a4 : in STD_LOGIC_VECTOR (7 downto 0);
        b1 : out STD_LOGIC_VECTOR (7 downto 0);
        b2 : out STD_LOGIC_VECTOR (7 downto 0);
        b3 : out STD_LOGIC_VECTOR (7 downto 0);
        b4 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal a1:std_logic_vector(7 downto 0);
signal a2:std_logic_vector(7 downto 0);
signal a3:std_logic_vector(7 downto 0);
signal a4:std_logic_vector(7 downto 0);
signal b1:std_logic_vector(7 downto 0);
signal b2:std_logic_vector(7 downto 0);
signal b3:std_logic_vector(7 downto 0);
signal b4:std_logic_vector(7 downto 0);

begin
mix_sims:mix_sim port
map(a1=>a1,a2=>a2,a3=>a3,a4=>a4,b1=>b1,b2=>b2,b3=>b3,b4=>b4);
process
begin
a1<=x"63";
a2<=x"2f";
a3<=x"af";
a4<=x"a2";
wait for 500ns;
end process;

end Behavioral;
```

## Simulation Result

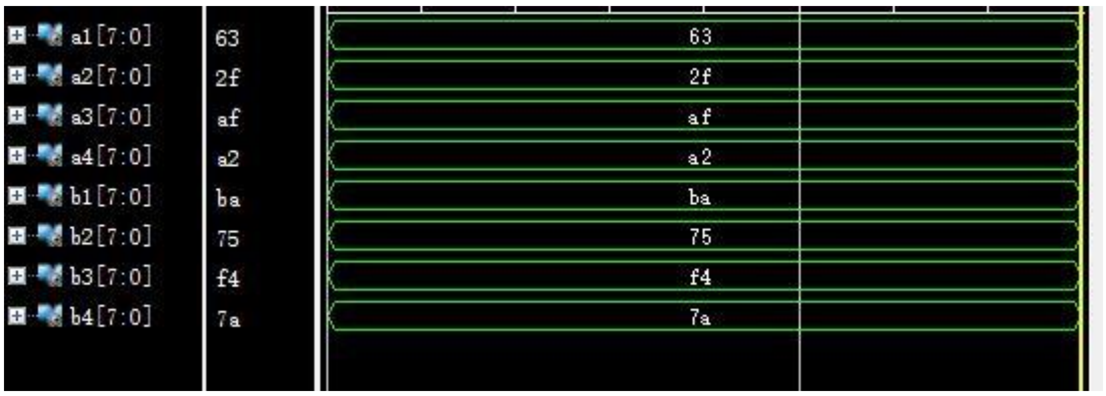


Figure7

Figure7 shows the simulation result of the mix simple.

### 3.2.5 Mix Column

#### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity mix_column is
    Port ( clk : in STD_LOGIC;
           a11 : in STD_LOGIC_VECTOR (7 downto 0);
           a12 : in STD_LOGIC_VECTOR (7 downto 0);
           a13 : in STD_LOGIC_VECTOR (7 downto 0);
           a14 : in STD_LOGIC_VECTOR (7 downto 0);
           a21 : in STD_LOGIC_VECTOR (7 downto 0);
           a22 : in STD_LOGIC_VECTOR (7 downto 0);
           a23 : in STD_LOGIC_VECTOR (7 downto 0);
           a24 : in STD_LOGIC_VECTOR (7 downto 0);
           a31 : in STD_LOGIC_VECTOR (7 downto 0);
           a32 : in STD_LOGIC_VECTOR (7 downto 0);
           a33 : in STD_LOGIC_VECTOR (7 downto 0);
           a34 : in STD_LOGIC_VECTOR (7 downto 0);
           a41 : in STD_LOGIC_VECTOR (7 downto 0);
           a42 : in STD_LOGIC_VECTOR (7 downto 0);
           a43 : in STD_LOGIC_VECTOR (7 downto 0);
           a44 : in STD_LOGIC_VECTOR (7 downto 0);
           b11 : out STD_LOGIC_VECTOR (7 downto 0);
           b12 : out STD_LOGIC_VECTOR (7 downto 0);
           b13 : out STD_LOGIC_VECTOR (7 downto 0);
           b14 : out STD_LOGIC_VECTOR (7 downto 0);
           b21 : out STD_LOGIC_VECTOR (7 downto 0);
           b22 : out STD_LOGIC_VECTOR (7 downto 0);
           b23 : out STD_LOGIC_VECTOR (7 downto 0);
           b24 : out STD_LOGIC_VECTOR (7 downto 0);
           b31 : out STD_LOGIC_VECTOR (7 downto 0);
           b32 : out STD_LOGIC_VECTOR (7 downto 0);
           b33 : out STD_LOGIC_VECTOR (7 downto 0);
           b34 : out STD_LOGIC_VECTOR (7 downto 0);
           b41 : out STD_LOGIC_VECTOR (7 downto 0);
           b42 : out STD_LOGIC_VECTOR (7 downto 0);
           b43 : out STD_LOGIC_VECTOR (7 downto 0);
           b44 : out STD_LOGIC_VECTOR (7 downto 0));
end mix_column;

architecture Behavioral of mix_column is
```

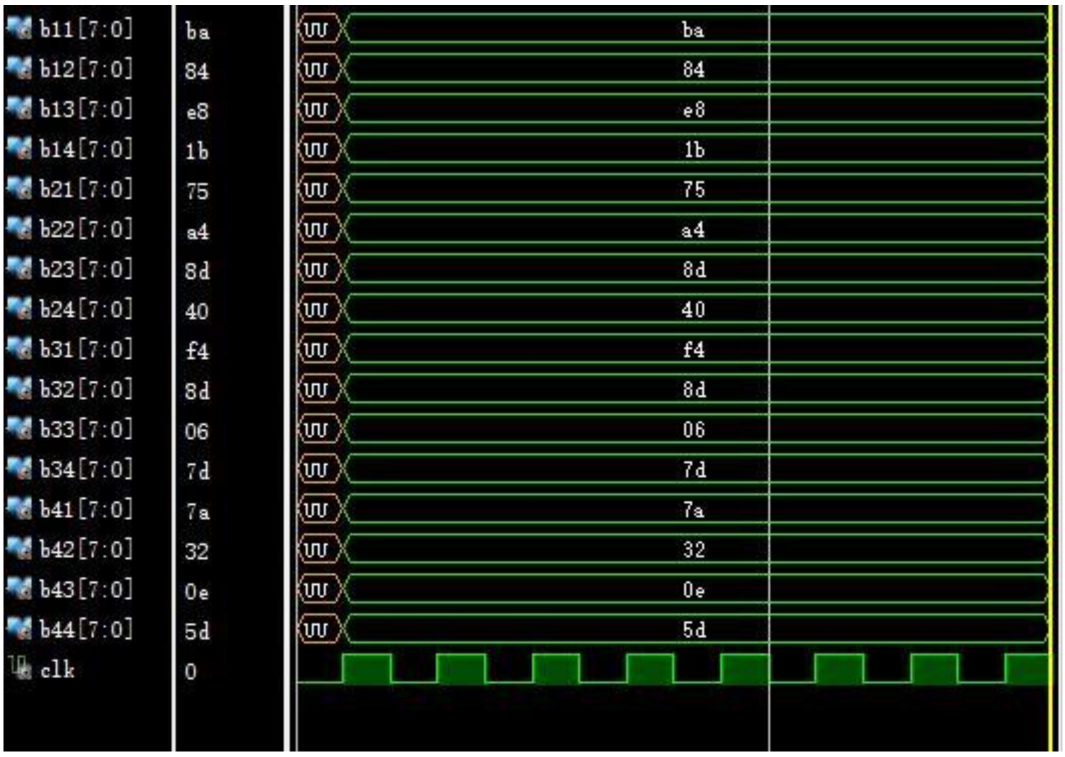
```
component mix_sim
        Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
              a2 : in STD_LOGIC_VECTOR (7 downto 0);
              a3 : in STD_LOGIC_VECTOR (7 downto 0);
              a4 : in STD_LOGIC_VECTOR (7 downto 0);
              b1 : out STD_LOGIC_VECTOR (7 downto 0);
              b2 : out STD_LOGIC_VECTOR (7 downto 0);
              b3 : out STD_LOGIC_VECTOR (7 downto 0);
              b4 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal temp11,temp12,temp13,temp14:std_logic_vector(7 downto 0);
signal temp21,temp22,temp23,temp24:std_logic_vector(7 downto 0);
signal temp31,temp32,temp33,temp34:std_logic_vector(7 downto 0);
signal temp41,temp42,temp43,temp44:std_logic_vector(7 downto 0);
begin
mix1:mix_sim port
map(a1=>a11,a2=>a21,a3=>a31,a4=>a41,b1=>temp11,b2=>temp21,b3=>temp31,b4=>t
emp41);
mix2:mix_sim port
map(a1=>a12,a2=>a22,a3=>a32,a4=>a42,b1=>temp12,b2=>temp22,b3=>temp32,b4=>t
emp42);
mix3:mix_sim port
map(a1=>a13,a2=>a23,a3=>a33,a4=>a43,b1=>temp13,b2=>temp23,b3=>temp33,b4=>t
emp43);
mix4:mix_sim port
map(a1=>a14,a2=>a24,a3=>a34,a4=>a44,b1=>temp14,b2=>temp24,b3=>temp34,b4=>t
emp44);
process(clk)
begin
if clk='1' and clk'event then
b11<=temp11;b12<=temp12;b13<=temp13;b14<=temp14;
b21<=temp21;b22<=temp22;b23<=temp23;b24<=temp24;
b31<=temp31;b32<=temp32;b33<=temp33;b34<=temp34;
b41<=temp41;b42<=temp42;b43<=temp43;b44<=temp44;
end if;
end process;
end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_mix is
--    Port ( );
end test_mix;

architecture Behavioral of test_mix is
component mix_column is
        Port ( clk : in STD_LOGIC;
              a11 : in STD_LOGIC_VECTOR (7 downto 0);
              a12 : in STD_LOGIC_VECTOR (7 downto 0);
              a13 : in STD_LOGIC_VECTOR (7 downto 0);
              a14 : in STD_LOGIC_VECTOR (7 downto 0);
              a21 : in STD_LOGIC_VECTOR (7 downto 0);
              a22 : in STD_LOGIC_VECTOR (7 downto 0);
              a23 : in STD_LOGIC_VECTOR (7 downto 0);
              a24 : in STD_LOGIC_VECTOR (7 downto 0);
              a31 : in STD_LOGIC_VECTOR (7 downto 0);
              a32 : in STD_LOGIC_VECTOR (7 downto 0);
              a33 : in STD_LOGIC_VECTOR (7 downto 0);
              a34 : in STD_LOGIC_VECTOR (7 downto 0);
              a41 : in STD_LOGIC_VECTOR (7 downto 0);
              a42 : in STD_LOGIC_VECTOR (7 downto 0);
              a43 : in STD_LOGIC_VECTOR (7 downto 0);
              a44 : in STD_LOGIC_VECTOR (7 downto 0);
              b11 : out STD_LOGIC_VECTOR (7 downto 0);
              b12 : out STD_LOGIC_VECTOR (7 downto 0);
              b13 : out STD_LOGIC_VECTOR (7 downto 0);
              b14 : out STD_LOGIC_VECTOR (7 downto 0);
              b21 : out STD_LOGIC_VECTOR (7 downto 0);
              b22 : out STD_LOGIC_VECTOR (7 downto 0);
              b23 : out STD_LOGIC_VECTOR (7 downto 0);
              b24 : out STD_LOGIC_VECTOR (7 downto 0);
              b31 : out STD_LOGIC_VECTOR (7 downto 0);
              b32 : out STD_LOGIC_VECTOR (7 downto 0);
              b33 : out STD_LOGIC_VECTOR (7 downto 0);
              b34 : out STD_LOGIC_VECTOR (7 downto 0);
              b41 : out STD_LOGIC_VECTOR (7 downto 0);
              b42 : out STD_LOGIC_VECTOR (7 downto 0);
              b43 : out STD_LOGIC_VECTOR (7 downto 0);
              b44 : out STD_LOGIC_VECTOR (7 downto 0));
```

```
end component;
signal
a11,a12,a13,a14,a21,a22,a23,a24,a31,a32,a33,a34,a41,a42,a43,a44,b11,b12,b13,b14,b21,b22,
b23,b24,b31,b32,b33,b34,b41,b42,b43,b44:std_logic_vector(7 downto 0);
signal clk:std_logic;
begin
mix:mix_column port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,b11
=>b11,b12=>b12,b13=>b13,b14=>b14,b21=>b21,b22=>b22,b23=>b23,b24=>b24,b31=
>b31,b32=>b32,b33=>b33,b34=>b34,b41=>b41,b42=>b42,b43=>b43,b44=>b44,clk=>cl
k);
process
begin
clk<='0';
wait for 50ns;
clk<='1';
wait for 50ns;
end process;
process
begin
a11<=X"63";
a12<=X"eb";
a13<=X"9f";
a14<=X"a0";
a21<=X"2f";
a22<=X"93";
a23<=X"92";
a24<=X"c0";
a31<=X"af";
a32<=X"c7";
a33<=X"ab";
a34<=X"30";
a41<=X"a2";
a42<=X"20";
a43<=X"cb";
a44<=X"2b";
wait for 100ns;
end process;

end Behavioral;
```

## Simulation Result



```
b11[7:0]  ba  ba
b12[7:0]  84  84
b13[7:0]  e8  e8
b14[7:0]  1b  1b
b21[7:0]  75  75
b22[7:0]  a4  a4
b23[7:0]  8d  8d
b24[7:0]  40  40
b31[7:0]  f4  f4
b32[7:0]  8d  8d
b33[7:0]  06  06
b34[7:0]  7d  7d
b41[7:0]  7a  7a
b42[7:0]  32  32
b43[7:0]  0e  0e
b44[7:0]  5d  5d
clk       0
```

Figure8

Figure8 shows the simulation result of the mix.

### 3.2.6 Normal Round

#### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity normal_round is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
```

```
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end normal_round;

architecture Behavioral of normal_round is
component add_round_key is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
```

```
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
component sub is
     Port ( clk : in STD_LOGIC;
    a11 : in STD_LOGIC_VECTOR (7 downto 0);
    a12 : in STD_LOGIC_VECTOR (7 downto 0);
    a13 : in STD_LOGIC_VECTOR (7 downto 0);
    a14 : in STD_LOGIC_VECTOR (7 downto 0);
    a21 : in STD_LOGIC_VECTOR (7 downto 0);
    a22 : in STD_LOGIC_VECTOR (7 downto 0);
    a23 : in STD_LOGIC_VECTOR (7 downto 0);
    a24 : in STD_LOGIC_VECTOR (7 downto 0);
    a31 : in STD_LOGIC_VECTOR (7 downto 0);
    a32 : in STD_LOGIC_VECTOR (7 downto 0);
    a33 : in STD_LOGIC_VECTOR (7 downto 0);
    a34 : in STD_LOGIC_VECTOR (7 downto 0);
    a41 : in STD_LOGIC_VECTOR (7 downto 0);
    a42 : in STD_LOGIC_VECTOR (7 downto 0);
    a43 : in STD_LOGIC_VECTOR (7 downto 0);
    a44 : in STD_LOGIC_VECTOR (7 downto 0);
    b11 : out STD_LOGIC_VECTOR (7 downto 0);
    b12 : out STD_LOGIC_VECTOR (7 downto 0);
    b13 : out STD_LOGIC_VECTOR (7 downto 0);
    b14 : out STD_LOGIC_VECTOR (7 downto 0);
    b21 : out STD_LOGIC_VECTOR (7 downto 0);
    b22 : out STD_LOGIC_VECTOR (7 downto 0);
    b23 : out STD_LOGIC_VECTOR (7 downto 0);
    b24 : out STD_LOGIC_VECTOR (7 downto 0);
    b31 : out STD_LOGIC_VECTOR (7 downto 0);
    b32 : out STD_LOGIC_VECTOR (7 downto 0);
    b33 : out STD_LOGIC_VECTOR (7 downto 0);
    b34 : out STD_LOGIC_VECTOR (7 downto 0);
    b41 : out STD_LOGIC_VECTOR (7 downto 0);
    b42 : out STD_LOGIC_VECTOR (7 downto 0);
    b43 : out STD_LOGIC_VECTOR (7 downto 0);
    b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
component shift_row is
     Port ( a11 : in STD_LOGIC_VECTOR (7 downto 0);
         a12 : in STD_LOGIC_VECTOR (7 downto 0);
         a13 : in STD_LOGIC_VECTOR (7 downto 0);
         a14 : in STD_LOGIC_VECTOR (7 downto 0);
         a21 : in STD_LOGIC_VECTOR (7 downto 0);
         a22 : in STD_LOGIC_VECTOR (7 downto 0);
         a23 : in STD_LOGIC_VECTOR (7 downto 0);
         a24 : in STD_LOGIC_VECTOR (7 downto 0);
         a31 : in STD_LOGIC_VECTOR (7 downto 0);
         a32 : in STD_LOGIC_VECTOR (7 downto 0);
         a33 : in STD_LOGIC_VECTOR (7 downto 0);
         a34 : in STD_LOGIC_VECTOR (7 downto 0);
         a41 : in STD_LOGIC_VECTOR (7 downto 0);
         a42 : in STD_LOGIC_VECTOR (7 downto 0);
         a43 : in STD_LOGIC_VECTOR (7 downto 0);
         a44 : in STD_LOGIC_VECTOR (7 downto 0);
         clk : in STD_LOGIC;
         b11 : out STD_LOGIC_VECTOR (7 downto 0);
         b12 : out STD_LOGIC_VECTOR (7 downto 0);
         b13 : out STD_LOGIC_VECTOR (7 downto 0);
         b14 : out STD_LOGIC_VECTOR (7 downto 0);
         b21 : out STD_LOGIC_VECTOR (7 downto 0);
         b22 : out STD_LOGIC_VECTOR (7 downto 0);
         b23 : out STD_LOGIC_VECTOR (7 downto 0);
         b24 : out STD_LOGIC_VECTOR (7 downto 0);
         b31 : out STD_LOGIC_VECTOR (7 downto 0);
         b32 : out STD_LOGIC_VECTOR (7 downto 0);
         b33 : out STD_LOGIC_VECTOR (7 downto 0);
         b34 : out STD_LOGIC_VECTOR (7 downto 0);
```

```
            b41 : out STD_LOGIC_VECTOR (7 downto 0);
            b42 : out STD_LOGIC_VECTOR (7 downto 0);
            b43 : out STD_LOGIC_VECTOR (7 downto 0);
            b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
component mix_column is
       Port ( clk : in STD_LOGIC;
            a11 : in STD_LOGIC_VECTOR (7 downto 0);
            a12 : in STD_LOGIC_VECTOR (7 downto 0);
            a13 : in STD_LOGIC_VECTOR (7 downto 0);
            a14 : in STD_LOGIC_VECTOR (7 downto 0);
            a21 : in STD_LOGIC_VECTOR (7 downto 0);
            a22 : in STD_LOGIC_VECTOR (7 downto 0);
            a23 : in STD_LOGIC_VECTOR (7 downto 0);
            a24 : in STD_LOGIC_VECTOR (7 downto 0);
            a31 : in STD_LOGIC_VECTOR (7 downto 0);
            a32 : in STD_LOGIC_VECTOR (7 downto 0);
            a33 : in STD_LOGIC_VECTOR (7 downto 0);
            a34 : in STD_LOGIC_VECTOR (7 downto 0);
            a41 : in STD_LOGIC_VECTOR (7 downto 0);
            a42 : in STD_LOGIC_VECTOR (7 downto 0);
            a43 : in STD_LOGIC_VECTOR (7 downto 0);
            a44 : in STD_LOGIC_VECTOR (7 downto 0);
            b11 : out STD_LOGIC_VECTOR (7 downto 0);
            b12 : out STD_LOGIC_VECTOR (7 downto 0);
            b13 : out STD_LOGIC_VECTOR (7 downto 0);
            b14 : out STD_LOGIC_VECTOR (7 downto 0);
            b21 : out STD_LOGIC_VECTOR (7 downto 0);
            b22 : out STD_LOGIC_VECTOR (7 downto 0);
            b23 : out STD_LOGIC_VECTOR (7 downto 0);
            b24 : out STD_LOGIC_VECTOR (7 downto 0);
            b31 : out STD_LOGIC_VECTOR (7 downto 0);
            b32 : out STD_LOGIC_VECTOR (7 downto 0);
            b33 : out STD_LOGIC_VECTOR (7 downto 0);
            b34 : out STD_LOGIC_VECTOR (7 downto 0);
            b41 : out STD_LOGIC_VECTOR (7 downto 0);
            b42 : out STD_LOGIC_VECTOR (7 downto 0);
            b43 : out STD_LOGIC_VECTOR (7 downto 0);
            b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;

signal
tempa11,tempa12,tempa13,tempa14,tempa21,tempa22,tempa23,tempa24,tempa31,tempa3
2,tempa33,tempa34,tempa41,tempa42,tempa43,tempa44 :std_logic_vector(7 downto 0);
```

```
signal
tempb11,tempb12,tempb13,tempb14,tempb21,tempb22,tempb23,tempb24,tempb31,temp
b32,tempb33,tempb34,tempb41,tempb42,tempb43,tempb44 :std_logic_vector(7 downto 0);
signal
tempc11,tempc12,tempc13,tempc14,tempc21,tempc22,tempc23,tempc24,tempc31,tempc32
,tempc33,tempc34,tempc41,tempc42,tempc43,tempc44 :std_logic_vector(7 downto 0);
begin
addkey:add_round_key port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,key1
1=>key11,key12=>key12,key13=>key13,key14=>key14,key21=>key21,key22=>key22,key2
3=>key23,key24=>key24,key31=>key31,key32=>key32,key33=>key33,key34=>key34,key4
1=>key41,key42=>key42,key43=>key43,key44=>key44,b11=>tempa11,b12=>tempa12,b1
3=>tempa13,b14=>tempa14,b21=>tempa21,b22=>tempa22,b23=>tempa23,b24=>tempa
24,b31=>tempa31,b32=>tempa32,b33=>tempa33,b34=>tempa34,b41=>tempa41,b42=>t
empa42,b43=>tempa43,b44=>tempa44,clk=>clk);
subs:sub port
map(a11=>tempa11,a12=>tempa12,a13=>tempa13,a14=>tempa14,a21=>tempa21,a22=
>tempa22,a23=>tempa23,a24=>tempa24,a31=>tempa31,a32=>tempa32,a33=>tempa33,
a34=>tempa34,a41=>tempa41,a42=>tempa42,a43=>tempa43,a44=>tempa44,b11=>tem
pb11,b12=>tempb12,b13=>tempb13,b14=>tempb14,b21=>tempb21,b22=>tempb22,b23
=>tempb23,b24=>tempb24,b31=>tempb31,b32=>tempb32,b33=>tempb33,b34=>tempb
34,b41=>tempb41,b42=>tempb42,b43=>tempb43,b44=>tempb44,clk=>clk);
sh_row:shift_row port
map(a11=>tempb11,a12=>tempb12,a13=>tempb13,a14=>tempb14,a21=>tempb21,a22=
>tempb22,a23=>tempb23,a24=>tempb24,a31=>tempb31,a32=>tempb32,a33=>tempb33
,a34=>tempb34,a41=>tempb41,a42=>tempb42,a43=>tempb43,a44=>tempb44,b11=>te
mpc11,b12=>tempc12,b13=>tempc13,b14=>tempc14,b21=>tempc21,b22=>tempc22,b23
=>tempc23,b24=>tempc24,b31=>tempc31,b32=>tempc32,b33=>tempc33,b34=>tempc3
4,b41=>tempc41,b42=>tempc42,b43=>tempc43,b44=>tempc44,clk=>clk);
mixs:mix_column port
map(a11=>tempc11,a12=>tempc12,a13=>tempc13,a14=>tempc14,a21=>tempc21,a22=>
tempc22,a23=>tempc23,a24=>tempc24,a31=>tempc31,a32=>tempc32,a33=>tempc33,a3
4=>tempc34,a41=>tempc41,a42=>tempc42,a43=>tempc43,a44=>tempc44,b11=>b11,b1
2=>b12,b13=>b13,b14=>b14,b21=>b21,b22=>b22,b23=>b23,b24=>b24,b31=>b31,b32
=>b32,b33=>b33,b34=>b34,b41=>b41,b42=>b42,b43=>b43,b44=>b44,clk=>clk);
end Behavioral;
```

## Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_normal_round is
--    Port ( );
end test_normal_round;

architecture Behavioral of test_normal_round is
component normal_round is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal
a11,a12,a13,a14,a21,a22,a23,a24,a31,a32,a33,a34,a41,a42,a43,a44,key11,key12,key13,key14,
key21,key22,key23,key24,key31,key32,key33,key34,key41,key42,key43,key44,b11,b12,b13,b1
4,b21,b22,b23,b24,b31,b32,b33,b34,b41,b42,b43,b44:std_logic_vector(7 downto 0);
signal clk:std_logic;
begin
nor_rnd:normal_round port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,key1
1=>key11,key12=>key12,key13=>key13,key14=>key14,key21=>key21,key22=>key22,key2
3=>key23,key24=>key24,key31=>key31,key32=>key32,key33=>key33,key34=>key34,key4
1=>key41,key42=>key42,key43=>key43,key44=>key44,b11=>b11,b12=>b12,b13=>b13,b
14=>b14,b21=>b21,b22=>b22,b23=>b23,b24=>b24,b31=>b31,b32=>b32,b33=>b33,b34
=>b34,b41=>b41,b42=>b42,b43=>b43,b44=>b44,clk=>clk);
process
begin
clk<='0';
wait for 50ns;
clk<='1';
wait for 50ns;
end process;
process
begin
a11<=x"54";a12<=x"4f";a13<=x"4e";a14<=x"20";
a21<=x"77";a22<=x"6e";a23<=x"69";a24<=x"54";
a31<=x"6f";a32<=x"65";a33<=x"6e";a34<=x"77";
a41<=x"20";a42<=x"20";a43<=x"65";a44<=x"6f";
```

```
key11<=x"54";key12<=x"73";key13<=x"20";key14<=x"67";
key21<=x"68";key22<=x"20";key23<=x"4b";key24<=x"20";
key31<=x"61";key32<=x"6d";key33<=x"75";key34<=x"46";
key41<=x"74";key42<=x"79";key43<=x"6e";key44<=x"75";
wait for 500ns;
end process;
end Behavioral;
```

## Simulation Result

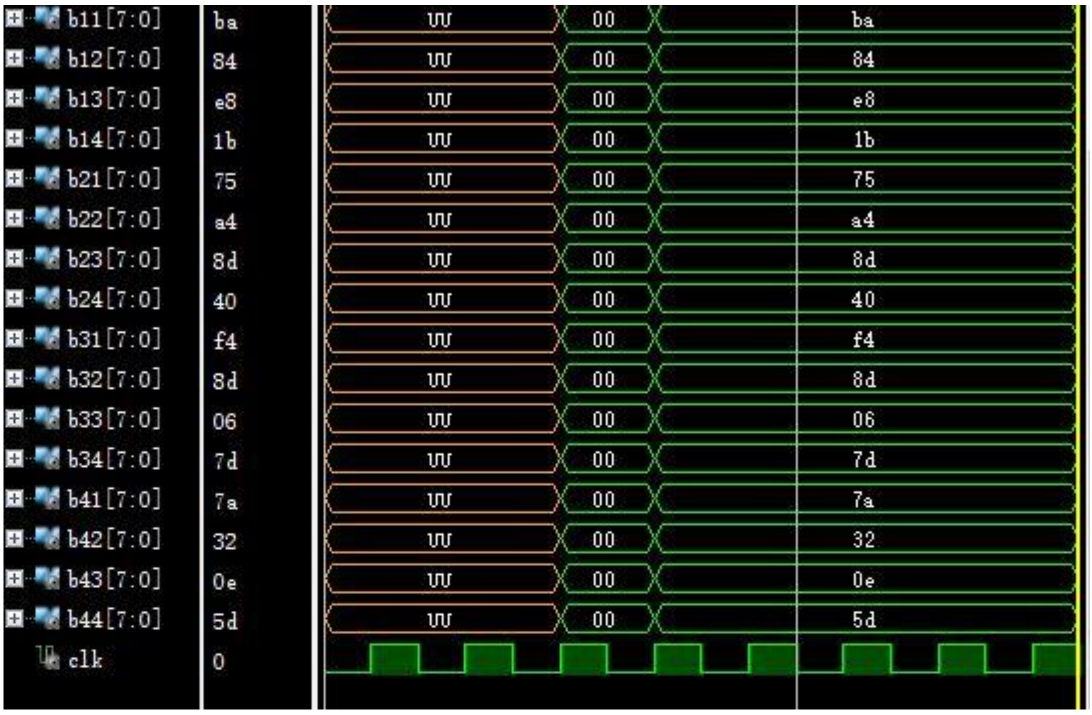


Figure9

Figure9 shows the simulation result of the normal round.

### 3.2.7 Final Round

#### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity rnd_final is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
```

```
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end rnd_final;

architecture Behavioral of rnd_final is
component add_round_key is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
```

```
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
component sub is
      Port ( clk : in STD_LOGIC;
    a11 : in STD_LOGIC_VECTOR (7 downto 0);
    a12 : in STD_LOGIC_VECTOR (7 downto 0);
    a13 : in STD_LOGIC_VECTOR (7 downto 0);
    a14 : in STD_LOGIC_VECTOR (7 downto 0);
    a21 : in STD_LOGIC_VECTOR (7 downto 0);
    a22 : in STD_LOGIC_VECTOR (7 downto 0);
    a23 : in STD_LOGIC_VECTOR (7 downto 0);
    a24 : in STD_LOGIC_VECTOR (7 downto 0);
    a31 : in STD_LOGIC_VECTOR (7 downto 0);
    a32 : in STD_LOGIC_VECTOR (7 downto 0);
    a33 : in STD_LOGIC_VECTOR (7 downto 0);
    a34 : in STD_LOGIC_VECTOR (7 downto 0);
    a41 : in STD_LOGIC_VECTOR (7 downto 0);
    a42 : in STD_LOGIC_VECTOR (7 downto 0);
    a43 : in STD_LOGIC_VECTOR (7 downto 0);
    a44 : in STD_LOGIC_VECTOR (7 downto 0);
    b11 : out STD_LOGIC_VECTOR (7 downto 0);
    b12 : out STD_LOGIC_VECTOR (7 downto 0);
    b13 : out STD_LOGIC_VECTOR (7 downto 0);
    b14 : out STD_LOGIC_VECTOR (7 downto 0);
    b21 : out STD_LOGIC_VECTOR (7 downto 0);
    b22 : out STD_LOGIC_VECTOR (7 downto 0);
    b23 : out STD_LOGIC_VECTOR (7 downto 0);
    b24 : out STD_LOGIC_VECTOR (7 downto 0);
    b31 : out STD_LOGIC_VECTOR (7 downto 0);
    b32 : out STD_LOGIC_VECTOR (7 downto 0);
    b33 : out STD_LOGIC_VECTOR (7 downto 0);
    b34 : out STD_LOGIC_VECTOR (7 downto 0);
    b41 : out STD_LOGIC_VECTOR (7 downto 0);
    b42 : out STD_LOGIC_VECTOR (7 downto 0);
    b43 : out STD_LOGIC_VECTOR (7 downto 0);
    b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
component shift_row is
     Port ( a11 : in STD_LOGIC_VECTOR (7 downto 0);
          a12 : in STD_LOGIC_VECTOR (7 downto 0);
          a13 : in STD_LOGIC_VECTOR (7 downto 0);
          a14 : in STD_LOGIC_VECTOR (7 downto 0);
          a21 : in STD_LOGIC_VECTOR (7 downto 0);
          a22 : in STD_LOGIC_VECTOR (7 downto 0);
          a23 : in STD_LOGIC_VECTOR (7 downto 0);
          a24 : in STD_LOGIC_VECTOR (7 downto 0);
          a31 : in STD_LOGIC_VECTOR (7 downto 0);
          a32 : in STD_LOGIC_VECTOR (7 downto 0);
          a33 : in STD_LOGIC_VECTOR (7 downto 0);
          a34 : in STD_LOGIC_VECTOR (7 downto 0);
          a41 : in STD_LOGIC_VECTOR (7 downto 0);
          a42 : in STD_LOGIC_VECTOR (7 downto 0);
          a43 : in STD_LOGIC_VECTOR (7 downto 0);
          a44 : in STD_LOGIC_VECTOR (7 downto 0);
          clk : in STD_LOGIC;
          b11 : out STD_LOGIC_VECTOR (7 downto 0);
          b12 : out STD_LOGIC_VECTOR (7 downto 0);
          b13 : out STD_LOGIC_VECTOR (7 downto 0);
          b14 : out STD_LOGIC_VECTOR (7 downto 0);
          b21 : out STD_LOGIC_VECTOR (7 downto 0);
          b22 : out STD_LOGIC_VECTOR (7 downto 0);
          b23 : out STD_LOGIC_VECTOR (7 downto 0);
          b24 : out STD_LOGIC_VECTOR (7 downto 0);
          b31 : out STD_LOGIC_VECTOR (7 downto 0);
          b32 : out STD_LOGIC_VECTOR (7 downto 0);
          b33 : out STD_LOGIC_VECTOR (7 downto 0);
          b34 : out STD_LOGIC_VECTOR (7 downto 0);
```

```
            b41 : out STD_LOGIC_VECTOR (7 downto 0);
            b42 : out STD_LOGIC_VECTOR (7 downto 0);
            b43 : out STD_LOGIC_VECTOR (7 downto 0);
            b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal
tempa11,tempa12,tempa13,tempa14,tempa21,tempa22,tempa23,tempa24,tempa31,tempa3
2,tempa33,tempa34,tempa41,tempa42,tempa43,tempa44 :std_logic_vector(7 downto 0);
signal
tempb11,tempb12,tempb13,tempb14,tempb21,tempb22,tempb23,tempb24,tempb31,temp
b32,tempb33,tempb34,tempb41,tempb42,tempb43,tempb44 :std_logic_vector(7 downto 0);
begin
addkey:add_round_key port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,key1
1=>key11,key12=>key12,key13=>key13,key14=>key14,key21=>key21,key22=>key22,key2
3=>key23,key24=>key24,key31=>key31,key32=>key32,key33=>key33,key34=>key34,key4
1=>key41,key42=>key42,key43=>key43,key44=>key44,b11=>tempa11,b12=>tempa12,b1
3=>tempa13,b14=>tempa14,b21=>tempa21,b22=>tempa22,b23=>tempa23,b24=>tempa
24,b31=>tempa31,b32=>tempa32,b33=>tempa33,b34=>tempa34,b41=>tempa41,b42=>t
empa42,b43=>tempa43,b44=>tempa44,clk=>clk);
subs:sub port
map(a11=>tempa11,a12=>tempa12,a13=>tempa13,a14=>tempa14,a21=>tempa21,a22=
>tempa22,a23=>tempa23,a24=>tempa24,a31=>tempa31,a32=>tempa32,a33=>tempa33,
a34=>tempa34,a41=>tempa41,a42=>tempa42,a43=>tempa43,a44=>tempa44,b11=>tem
pb11,b12=>tempb12,b13=>tempb13,b14=>tempb14,b21=>tempb21,b22=>tempb22,b23
=>tempb23,b24=>tempb24,b31=>tempb31,b32=>tempb32,b33=>tempb33,b34=>tempb
34,b41=>tempb41,b42=>tempb42,b43=>tempb43,b44=>tempb44,clk=>clk);
sh_row:shift_row port
map(a11=>tempb11,a12=>tempb12,a13=>tempb13,a14=>tempb14,a21=>tempb21,a22=
>tempb22,a23=>tempb23,a24=>tempb24,a31=>tempb31,a32=>tempb32,a33=>tempb33
,a34=>tempb34,a41=>tempb41,a42=>tempb42,a43=>tempb43,a44=>tempb44,b11=>b1
1,b12=>b12,b13=>b13,b14=>b14,b21=>b21,b22=>b22,b23=>b23,b24=>b24,b31=>b31,
b32=>b32,b33=>b33,b34=>b34,b41=>b41,b42=>b42,b43=>b43,b44=>b44,clk=>clk);

end Behavioral;
```

## 3.3 Big

### Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity big_round is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
```

```
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end big_round;

architecture Behavioral of big_round is
component normal_round is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
```

```
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;

component rnd_final is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
```

```
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;

component key is
        Port ( a1 : in STD_LOGIC_VECTOR (7 downto 0);
      a2 : in STD_LOGIC_VECTOR (7 downto 0);
      a3 : in STD_LOGIC_VECTOR (7 downto 0);
      a4 : in STD_LOGIC_VECTOR (7 downto 0);
      a5 : in STD_LOGIC_VECTOR (7 downto 0);
      a6 : in STD_LOGIC_VECTOR (7 downto 0);
      a7 : in STD_LOGIC_VECTOR (7 downto 0);
      a8 : in STD_LOGIC_VECTOR (7 downto 0);
      a9 : in STD_LOGIC_VECTOR (7 downto 0);
      a10 : in STD_LOGIC_VECTOR (7 downto 0);
      a11 : in STD_LOGIC_VECTOR (7 downto 0);
```

```
    a12 : in STD_LOGIC_VECTOR (7 downto 0);
    a13 : in STD_LOGIC_VECTOR (7 downto 0);
    a14 : in STD_LOGIC_VECTOR (7 downto 0);
    a15 : in STD_LOGIC_VECTOR (7 downto 0);
    a16 : in STD_LOGIC_VECTOR (7 downto 0);
    b1 : out STD_LOGIC_VECTOR (7 downto 0);
    b2 : out STD_LOGIC_VECTOR (7 downto 0);
    b3 : out STD_LOGIC_VECTOR (7 downto 0);
    b4 : out STD_LOGIC_VECTOR (7 downto 0);
    b5 : out STD_LOGIC_VECTOR (7 downto 0);
    b6 : out STD_LOGIC_VECTOR (7 downto 0);
    b7 : out STD_LOGIC_VECTOR (7 downto 0);
    b8 : out STD_LOGIC_VECTOR (7 downto 0);
    b9 : out STD_LOGIC_VECTOR (7 downto 0);
    b10 : out STD_LOGIC_VECTOR (7 downto 0);
    b11 : out STD_LOGIC_VECTOR (7 downto 0);
    b12 : out STD_LOGIC_VECTOR (7 downto 0);
    b13 : out STD_LOGIC_VECTOR (7 downto 0);
    b14 : out STD_LOGIC_VECTOR (7 downto 0);
    b15 : out STD_LOGIC_VECTOR (7 downto 0);
    b16 : out STD_LOGIC_VECTOR (7 downto 0);
    clk : in STD_LOGIC;
    round_num:in std_logic_vector(7 downto 0));
end component;

component add_round_key is
     Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
```

```
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;

signal
tempa11,tempa12,tempa13,tempa14,tempa21,tempa22,tempa23,tempa24,tempa31,tempa3
2,tempa33,tempa34,tempa41,tempa42,tempa43,tempa44:std_logic_vector(7 downto 0);
signal
tempb11,tempb12,tempb13,tempb14,tempb21,tempb22,tempb23,tempb24,tempb31,temp
b32,tempb33,tempb34,tempb41,tempb42,tempb43,tempb44:std_logic_vector(7 downto 0);
signal
tempc11,tempc12,tempc13,tempc14,tempc21,tempc22,tempc23,tempc24,tempc31,tempc32
,tempc33,tempc34,tempc41,tempc42,tempc43,tempc44:std_logic_vector(7 downto 0);
signal
tempd11,tempd12,tempd13,tempd14,tempd21,tempd22,tempd23,tempd24,tempd31,temp
```

```
d32,tempd33,tempd34,tempd41,tempd42,tempd43,tempd44:std_logic_vector(7 downto 0);
signal
tempe11,tempe12,tempe13,tempe14,tempe21,tempe22,tempe23,tempe24,tempe31,tempe
32,tempe33,tempe34,tempe41,tempe42,tempe43,tempe44:std_logic_vector(7 downto 0);
signal
tempf11,tempf12,tempf13,tempf14,tempf21,tempf22,tempf23,tempf24,tempf31,tempf32,te
mpf33,tempf34,tempf41,tempf42,tempf43,tempf44:std_logic_vector(7 downto 0);
signal
tempg11,tempg12,tempg13,tempg14,tempg21,tempg22,tempg23,tempg24,tempg31,temp
g32,tempg33,tempg34,tempg41,tempg42,tempg43,tempg44:std_logic_vector(7 downto 0);
signal
temph11,temph12,temph13,temph14,temph21,temph22,temph23,temph24,temph31,temph
32,temph33,temph34,temph41,temph42,temph43,temph44:std_logic_vector(7 downto 0);
signal
tempi11,tempi12,tempi13,tempi14,tempi21,tempi22,tempi23,tempi24,tempi31,tempi32,tem
pi33,tempi34,tempi41,tempi42,tempi43,tempi44:std_logic_vector(7 downto 0);
signal
tempj11,tempj12,tempj13,tempj14,tempj21,tempj22,tempj23,tempj24,tempj31,tempj32,tem
pj33,tempj34,tempj41,tempj42,tempj43,tempj44:std_logic_vector(7 downto 0);

signal
keytempa11,keytempa12,keytempa13,keytempa14,keytempa21,keytempa22,keytempa23,ke
ytempa24,keytempa31,keytempa32,keytempa33,keytempa34,keytempa41,keytempa42,keyt
empa43,keytempa44:std_logic_vector(7 downto 0);
signal
keytempb11,keytempb12,keytempb13,keytempb14,keytempb21,keytempb22,keytempb23,k
eytempb24,keytempb31,keytempb32,keytempb33,keytempb34,keytempb41,keytempb42,ke
ytempb43,keytempb44:std_logic_vector(7 downto 0);
signal
keytempc11,keytempc12,keytempc13,keytempc14,keytempc21,keytempc22,keytempc23,key
tempc24,keytempc31,keytempc32,keytempc33,keytempc34,keytempc41,keytempc42,keyte
mpc43,keytempc44:std_logic_vector(7 downto 0);
signal
keytempd11,keytempd12,keytempd13,keytempd14,keytempd21,keytempd22,keytempd23,k
eytempd24,keytempd31,keytempd32,keytempd33,keytempd34,keytempd41,keytempd42,ke
ytempd43,keytempd44:std_logic_vector(7 downto 0);
signal
keytempe11,keytempe12,keytempe13,keytempe14,keytempe21,keytempe22,keytempe23,ke
ytempe24,keytempe31,keytempe32,keytempe33,keytempe34,keytempe41,keytempe42,keyt
empe43,keytempe44:std_logic_vector(7 downto 0);
signal
keytempf11,keytempf12,keytempf13,keytempf14,keytempf21,keytempf22,keytempf23,keyte
mpf24,keytempf31,keytempf32,keytempf33,keytempf34,keytempf41,keytempf42,keytempf4
3,keytempf44:std_logic_vector(7 downto 0);
```

```
signal
keytempg11,keytempg12,keytempg13,keytempg14,keytempg21,keytempg22,keytempg23,k
eytempg24,keytempg31,keytempg32,keytempg33,keytempg34,keytempg41,keytempg42,ke
ytempg43,keytempg44:std_logic_vector(7 downto 0);
signal
keytemph11,keytemph12,keytemph13,keytemph14,keytemph21,keytemph22,keytemph23,k
eytemph24,keytemph31,keytemph32,keytemph33,keytemph34,keytemph41,keytemph42,ke
ytemph43,keytemph44:std_logic_vector(7 downto 0);
signal
keytempi11,keytempi12,keytempi13,keytempi14,keytempi21,keytempi22,keytempi23,keytem
pi24,keytempi31,keytempi32,keytempi33,keytempi34,keytempi41,keytempi42,keytempi43,ke
ytempi44:std_logic_vector(7 downto 0);
signal
keytempj11,keytempj12,keytempj13,keytempj14,keytempj21,keytempj22,keytempj23,keytem
pj24,keytempj31,keytempj32,keytempj33,keytempj34,keytempj41,keytempj42,keytempj43,ke
ytempj44:std_logic_vector(7 downto 0);

begin
rnd1:normal_round port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,key1
1=>key11,key12=>key12,key13=>key13,key14=>key14,key21=>key21,key22=>key22,key2
3=>key23,key24=>key24,key31=>key31,key32=>key32,key33=>key33,key34=>key34,key4
1=>key41,key42=>key42,key43=>key43,key44=>key44,b11=>tempa11,b12=>tempa12,b1
3=>tempa13,b14=>tempa14,b21=>tempa21,b22=>tempa22,b23=>tempa23,b24=>tempa
24,b31=>tempa31,b32=>tempa32,b33=>tempa33,b34=>tempa34,b41=>tempa41,b42=>t
empa42,b43=>tempa43,b44=>tempa44,clk=>clk);
rnd2:normal_round port
map(a11=>tempa11,a12=>tempa12,a13=>tempa13,a14=>tempa14,a21=>tempa21,a22=
>tempa22,a23=>tempa23,a24=>tempa24,a31=>tempa31,a32=>tempa32,a33=>tempa33,
a34=>tempa34,a41=>tempa41,a42=>tempa42,a43=>tempa43,a44=>tempa44,key11=>ke
ytempa11,key12=>keytempa12,key13=>keytempa13,key14=>keytempa14,key21=>keytem
pa21,key22=>keytempa22,key23=>keytempa23,key24=>keytempa24,key31=>keytempa31
,key32=>keytempa32,key33=>keytempa33,key34=>keytempa34,key41=>keytempa41,key4
2=>keytempa42,key43=>keytempa43,key44=>keytempa44,b11=>tempb11,b12=>tempb1
2,b13=>tempb13,b14=>tempb14,b21=>tempb21,b22=>tempb22,b23=>tempb23,b24=>t
empb24,b31=>tempb31,b32=>tempb32,b33=>tempb33,b34=>tempb34,b41=>tempb41,
b42=>tempb42,b43=>tempb43,b44=>tempb44,clk=>clk);
rnd3:normal_round port
map(a11=>tempb11,a12=>tempb12,a13=>tempb13,a14=>tempb14,a21=>tempb21,a22=
>tempb22,a23=>tempb23,a24=>tempb24,a31=>tempb31,a32=>tempb32,a33=>tempb33
,a34=>tempb34,a41=>tempb41,a42=>tempb42,a43=>tempb43,a44=>tempb44,key11=>k
eytempb11,key12=>keytempb12,key13=>keytempb13,key14=>keytempb14,key21=>keyte
mpb21,key22=>keytempb22,key23=>keytempb23,key24=>keytempb24,key31=>keytempb
```

```
31,key32=>keytempb32,key33=>keytempb33,key34=>keytempb34,key41=>keytempb41,k
ey42=>keytempb42,key43=>keytempb43,key44=>keytempb44,b11=>tempc11,b12=>tem
pc12,b13=>tempc13,b14=>tempc14,b21=>tempc21,b22=>tempc22,b23=>tempc23,b24=
>tempc24,b31=>tempc31,b32=>tempc32,b33=>tempc33,b34=>tempc34,b41=>tempc41,
b42=>tempc42,b43=>tempc43,b44=>tempc44,clk=>clk);
rnd4:normal_round port
map(a11=>tempc11,a12=>tempc12,a13=>tempc13,a14=>tempc14,a21=>tempc21,a22=>
tempc22,a23=>tempc23,a24=>tempc24,a31=>tempc31,a32=>tempc32,a33=>tempc33,a3
4=>tempc34,a41=>tempc41,a42=>tempc42,a43=>tempc43,a44=>tempc44,key11=>keyte
mpc11,key12=>keytempc12,key13=>keytempc13,key14=>keytempc14,key21=>keytempc2
1,key22=>keytempc22,key23=>keytempc23,key24=>keytempc24,key31=>keytempc31,key
32=>keytempc32,key33=>keytempc33,key34=>keytempc34,key41=>keytempc41,key42=>
keytempc42,key43=>keytempc43,key44=>keytempc44,b11=>tempd11,b12=>tempd12,b1
3=>tempd13,b14=>tempd14,b21=>tempd21,b22=>tempd22,b23=>tempd23,b24=>temp
d24,b31=>tempd31,b32=>tempd32,b33=>tempd33,b34=>tempd34,b41=>tempd41,b42=
>tempd42,b43=>tempd43,b44=>tempd44,clk=>clk);
rnd5:normal_round port
map(a11=>tempd11,a12=>tempd12,a13=>tempd13,a14=>tempd14,a21=>tempd21,a22=
>tempd22,a23=>tempd23,a24=>tempd24,a31=>tempd31,a32=>tempd32,a33=>tempd33
,a34=>tempd34,a41=>tempd41,a42=>tempd42,a43=>tempd43,a44=>tempd44,key11=>k
eytempd11,key12=>keytempd12,key13=>keytempd13,key14=>keytempd14,key21=>keyte
mpd21,key22=>keytempd22,key23=>keytempd23,key24=>keytempd24,key31=>keytempd
31,key32=>keytempd32,key33=>keytempd33,key34=>keytempd34,key41=>keytempd41,k
ey42=>keytempd42,key43=>keytempd43,key44=>keytempd44,b11=>tempe11,b12=>tem
pe12,b13=>tempe13,b14=>tempe14,b21=>tempe21,b22=>tempe22,b23=>tempe23,b24
=>tempe24,b31=>tempe31,b32=>tempe32,b33=>tempe33,b34=>tempe34,b41=>tempe
41,b42=>tempe42,b43=>tempe43,b44=>tempe44,clk=>clk);
rnd6:normal_round port
map(a11=>tempe11,a12=>tempe12,a13=>tempe13,a14=>tempe14,a21=>tempe21,a22=
>tempe22,a23=>tempe23,a24=>tempe24,a31=>tempe31,a32=>tempe32,a33=>tempe33,
a34=>tempe34,a41=>tempe41,a42=>tempe42,a43=>tempe43,a44=>tempe44,key11=>ke
ytempe11,key12=>keytempe12,key13=>keytempe13,key14=>keytempe14,key21=>keytem
pe21,key22=>keytempe22,key23=>keytempe23,key24=>keytempe24,key31=>keytempe31
,key32=>keytempe32,key33=>keytempe33,key34=>keytempe34,key41=>keytempe41,key4
2=>keytempe42,key43=>keytempe43,key44=>keytempe44,b11=>tempf11,b12=>tempf12,
b13=>tempf13,b14=>tempf14,b21=>tempf21,b22=>tempf22,b23=>tempf23,b24=>tempf
24,b31=>tempf31,b32=>tempf32,b33=>tempf33,b34=>tempf34,b41=>tempf41,b42=>te
mpf42,b43=>tempf43,b44=>tempf44,clk=>clk);
rnd7:normal_round port
map(a11=>tempf11,a12=>tempf12,a13=>tempf13,a14=>tempf14,a21=>tempf21,a22=>te
mpf22,a23=>tempf23,a24=>tempf24,a31=>tempf31,a32=>tempf32,a33=>tempf33,a34=>
tempf34,a41=>tempf41,a42=>tempf42,a43=>tempf43,a44=>tempf44,key11=>keytempf11
,key12=>keytempf12,key13=>keytempf13,key14=>keytempf14,key21=>keytempf21,key22
=>keytempf22,key23=>keytempf23,key24=>keytempf24,key31=>keytempf31,key32=>keyt
```

```
empf32,key33=>keytempf33,key34=>keytempf34,key41=>keytempf41,key42=>keytempf4
2,key43=>keytempf43,key44=>keytempf44,b11=>tempg11,b12=>tempg12,b13=>tempg1
3,b14=>tempg14,b21=>tempg21,b22=>tempg22,b23=>tempg23,b24=>tempg24,b31=>t
empg31,b32=>tempg32,b33=>tempg33,b34=>tempg34,b41=>tempg41,b42=>tempg42,
b43=>tempg43,b44=>tempg44,clk=>clk);
rnd8:normal_round port
map(a11=>tempg11,a12=>tempg12,a13=>tempg13,a14=>tempg14,a21=>tempg21,a22=
>tempg22,a23=>tempg23,a24=>tempg24,a31=>tempg31,a32=>tempg32,a33=>tempg33
,a34=>tempg34,a41=>tempg41,a42=>tempg42,a43=>tempg43,a44=>tempg44,key11=>k
eytempg11,key12=>keytempg12,key13=>keytempg13,key14=>keytempg14,key21=>keyte
mpg21,key22=>keytempg22,key23=>keytempg23,key24=>keytempg24,key31=>keytempg
31,key32=>keytempg32,key33=>keytempg33,key34=>keytempg34,key41=>keytempg41,k
ey42=>keytempg42,key43=>keytempg43,key44=>keytempg44,b11=>temph11,b12=>tem
ph12,b13=>temph13,b14=>temph14,b21=>temph21,b22=>temph22,b23=>temph23,b24
=>temph24,b31=>temph31,b32=>temph32,b33=>temph33,b34=>temph34,b41=>temph
41,b42=>temph42,b43=>temph43,b44=>temph44,clk=>clk);
rnd9:normal_round port
map(a11=>temph11,a12=>temph12,a13=>temph13,a14=>temph14,a21=>temph21,a22=
>temph22,a23=>temph23,a24=>temph24,a31=>temph31,a32=>temph32,a33=>temph33,
a34=>temph34,a41=>temph41,a42=>temph42,a43=>temph43,a44=>temph44,key11=>k
eytemph11,key12=>keytemph12,key13=>keytemph13,key14=>keytemph14,key21=>keyte
mph21,key22=>keytemph22,key23=>keytemph23,key24=>keytemph24,key31=>keytemph
31,key32=>keytemph32,key33=>keytemph33,key34=>keytemph34,key41=>keytemph41,k
ey42=>keytemph42,key43=>keytemph43,key44=>keytemph44,b11=>tempi11,b12=>temp
i12,b13=>tempi13,b14=>tempi14,b21=>tempi21,b22=>tempi22,b23=>tempi23,b24=>te
mpi24,b31=>tempi31,b32=>tempi32,b33=>tempi33,b34=>tempi34,b41=>tempi41,b42=>
tempi42,b43=>tempi43,b44=>tempi44,clk=>clk);
rnd10:rnd_final port
map(a11=>tempi11,a12=>tempi12,a13=>tempi13,a14=>tempi14,a21=>tempi21,a22=>te
mpi22,a23=>tempi23,a24=>tempi24,a31=>tempi31,a32=>tempi32,a33=>tempi33,a34=>t
empi34,a41=>tempi41,a42=>tempi42,a43=>tempi43,a44=>tempi44,key11=>keytempi11,k
ey12=>keytempi12,key13=>keytempi13,key14=>keytempi14,key21=>keytempi21,key22=>
keytempi22,key23=>keytempi23,key24=>keytempi24,key31=>keytempi31,key32=>keytem
pi32,key33=>keytempi33,key34=>keytempi34,key41=>keytempi41,key42=>keytempi42,ke
y43=>keytempi43,key44=>keytempi44,b11=>tempj11,b12=>tempj12,b13=>tempj13,b14=
>tempj14,b21=>tempj21,b22=>tempj22,b23=>tempj23,b24=>tempj24,b31=>tempj31,b3
2=>tempj32,b33=>tempj33,b34=>tempj34,b41=>tempj41,b42=>tempj42,b43=>tempj43,
b44=>tempj44,clk=>clk);

key1:key port
map(a1=>key11,a2=>key21,a3=>key31,a4=>key41,a5=>key12,a6=>key22,a7=>key32,a8=
>key42,a9=>key13,a10=>key23,a11=>key33,a12=>key43,a13=>key14,a14=>key24,a15=>
key34,a16=>key44,b1=>keytempa11,b2=>keytempa21,b3=>keytempa31,b4=>keytempa4
1,b5=>keytempa12,b6=>keytempa22,b7=>keytempa32,b8=>keytempa42,b9=>keytempa1
```

```
3,b10=>keytempa23,b11=>keytempa33,b12=>keytempa43,b13=>keytempa14,b14=>keyt
empa24,b15=>keytempa34,b16=>keytempa44,round_num=>x"01",clk=>clk);
key2:key port
map(a1=>keytempa11,a2=>keytempa21,a3=>keytempa31,a4=>keytempa41,a5=>keytem
pa12,a6=>keytempa22,a7=>keytempa32,a8=>keytempa42,a9=>keytempa13,a10=>keyte
mpa23,a11=>keytempa33,a12=>keytempa43,a13=>keytempa14,a14=>keytempa24,a15=>
keytempa34,a16=>keytempa44,b1=>keytempb11,b2=>keytempb21,b3=>keytempb31,b4=
>keytempb41,b5=>keytempb12,b6=>keytempb22,b7=>keytempb32,b8=>keytempb42,b9
=>keytempb13,b10=>keytempb23,b11=>keytempb33,b12=>keytempb43,b13=>keytemp
b14,b14=>keytempb24,b15=>keytempb34,b16=>keytempb44,round_num=>x"02",clk=>cl
k);
key3:key port
map(a1=>keytempb11,a2=>keytempb21,a3=>keytempb31,a4=>keytempb41,a5=>keytem
pb12,a6=>keytempb22,a7=>keytempb32,a8=>keytempb42,a9=>keytempb13,a10=>keyte
mpb23,a11=>keytempb33,a12=>keytempb43,a13=>keytempb14,a14=>keytempb24,a15=
>keytempb34,a16=>keytempb44,b1=>keytempc11,b2=>keytempc21,b3=>keytempc31,b4
=>keytempc41,b5=>keytempc12,b6=>keytempc22,b7=>keytempc32,b8=>keytempc42,b9
=>keytempc13,b10=>keytempc23,b11=>keytempc33,b12=>keytempc43,b13=>keytempc
14,b14=>keytempc24,b15=>keytempc34,b16=>keytempc44,round_num=>x"04",clk=>clk);
key4:key port
map(a1=>keytempc11,a2=>keytempc21,a3=>keytempc31,a4=>keytempc41,a5=>keytemp
c12,a6=>keytempc22,a7=>keytempc32,a8=>keytempc42,a9=>keytempc13,a10=>keytemp
c23,a11=>keytempc33,a12=>keytempc43,a13=>keytempc14,a14=>keytempc24,a15=>key
tempc34,a16=>keytempc44,b1=>keytempd11,b2=>keytempd21,b3=>keytempd31,b4=>k
eytempd41,b5=>keytempd12,b6=>keytempd22,b7=>keytempd32,b8=>keytempd42,b9=>
keytempd13,b10=>keytempd23,b11=>keytempd33,b12=>keytempd43,b13=>keytempd14
,b14=>keytempd24,b15=>keytempd34,b16=>keytempd44,round_num=>x"08",clk=>clk);
key5:key port
map(a1=>keytempd11,a2=>keytempd21,a3=>keytempd31,a4=>keytempd41,a5=>keytem
pd12,a6=>keytempd22,a7=>keytempd32,a8=>keytempd42,a9=>keytempd13,a10=>keyte
mpd23,a11=>keytempd33,a12=>keytempd43,a13=>keytempd14,a14=>keytempd24,a15=
>keytempd34,a16=>keytempd44,b1=>keytempe11,b2=>keytempe21,b3=>keytempe31,b4
=>keytempe41,b5=>keytempe12,b6=>keytempe22,b7=>keytempe32,b8=>keytempe42,b
9=>keytempe13,b10=>keytempe23,b11=>keytempe33,b12=>keytempe43,b13=>keytemp
e14,b14=>keytempe24,b15=>keytempe34,b16=>keytempe44,round_num=>x"10",clk=>clk
);
key6:key port
map(a1=>keytempe11,a2=>keytempe21,a3=>keytempe31,a4=>keytempe41,a5=>keytem
pe12,a6=>keytempe22,a7=>keytempe32,a8=>keytempe42,a9=>keytempe13,a10=>keyte
mpe23,a11=>keytempe33,a12=>keytempe43,a13=>keytempe14,a14=>keytempe24,a15=
>keytempe34,a16=>keytempe44,b1=>keytempf11,b2=>keytempf21,b3=>keytempf31,b4=
>keytempf41,b5=>keytempf12,b6=>keytempf22,b7=>keytempf32,b8=>keytempf42,b9=>
keytempf13,b10=>keytempf23,b11=>keytempf33,b12=>keytempf43,b13=>keytempf14,b1
4=>keytempf24,b15=>keytempf34,b16=>keytempf44,round_num=>x"20",clk=>clk);
```

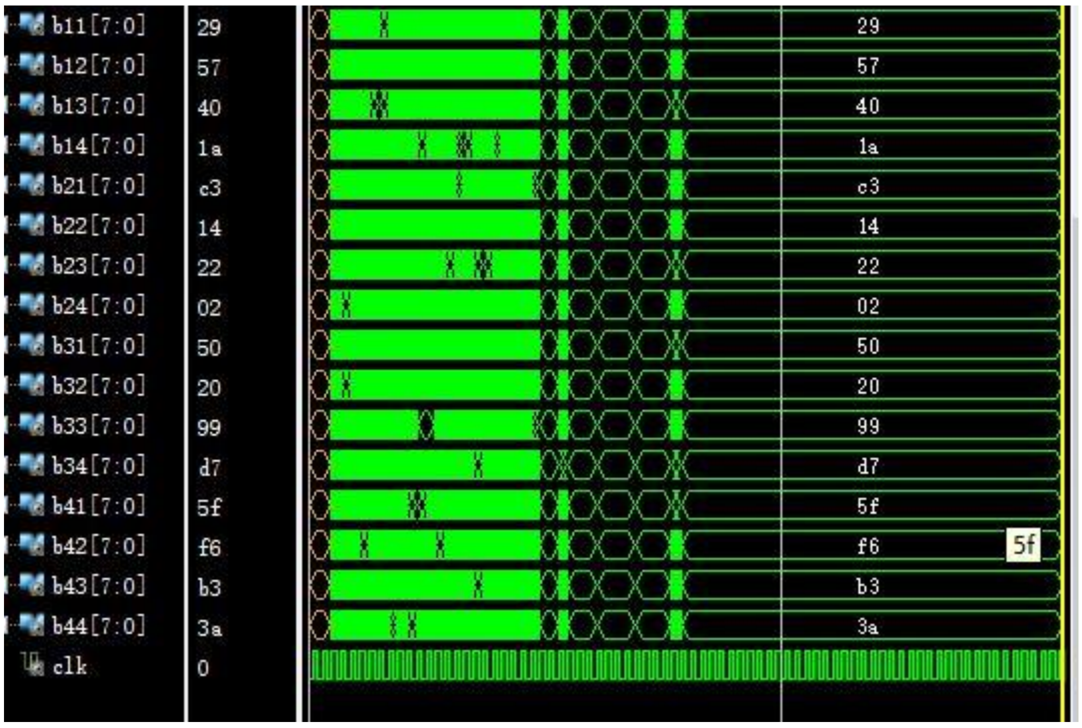
```
key7:key port
map(a1=>keytempf11,a2=>keytempf21,a3=>keytempf31,a4=>keytempf41,a5=>keytempf
12,a6=>keytempf22,a7=>keytempf32,a8=>keytempf42,a9=>keytempf13,a10=>keytempf2
3,a11=>keytempf33,a12=>keytempf43,a13=>keytempf14,a14=>keytempf24,a15=>keytem
pf34,a16=>keytempf44,b1=>keytempg11,b2=>keytempg21,b3=>keytempg31,b4=>keyte
mpg41,b5=>keytempg12,b6=>keytempg22,b7=>keytempg32,b8=>keytempg42,b9=>keyt
empg13,b10=>keytempg23,b11=>keytempg33,b12=>keytempg43,b13=>keytempg14,b14
=>keytempg24,b15=>keytempg34,b16=>keytempg44,round_num=>x"40",clk=>clk);
key8:key port
map(a1=>keytempg11,a2=>keytempg21,a3=>keytempg31,a4=>keytempg41,a5=>keytem
pg12,a6=>keytempg22,a7=>keytempg32,a8=>keytempg42,a9=>keytempg13,a10=>keyte
mpg23,a11=>keytempg33,a12=>keytempg43,a13=>keytempg14,a14=>keytempg24,a15=
>keytempg34,a16=>keytempg44,b1=>keytemph11,b2=>keytemph21,b3=>keytemph31,b
4=>keytemph41,b5=>keytemph12,b6=>keytemph22,b7=>keytemph32,b8=>keytemph42,
b9=>keytemph13,b10=>keytemph23,b11=>keytemph33,b12=>keytemph43,b13=>keytem
ph14,b14=>keytemph24,b15=>keytemph34,b16=>keytemph44,round_num=>x"80",clk=>c
lk);
key9:key port
map(a1=>keytemph11,a2=>keytemph21,a3=>keytemph31,a4=>keytemph41,a5=>keytem
ph12,a6=>keytemph22,a7=>keytemph32,a8=>keytemph42,a9=>keytemph13,a10=>keyte
mph23,a11=>keytemph33,a12=>keytemph43,a13=>keytemph14,a14=>keytemph24,a15=
>keytemph34,a16=>keytemph44,b1=>keytempi11,b2=>keytempi21,b3=>keytempi31,b4=
>keytempi41,b5=>keytempi12,b6=>keytempi22,b7=>keytempi32,b8=>keytempi42,b9=>k
eytempi13,b10=>keytempi23,b11=>keytempi33,b12=>keytempi43,b13=>keytempi14,b14
=>keytempi24,b15=>keytempi34,b16=>keytempi44,round_num=>x"1b",clk=>clk);
key10:key port
map(a1=>keytempi11,a2=>keytempi21,a3=>keytempi31,a4=>keytempi41,a5=>keytempi1
2,a6=>keytempi22,a7=>keytempi32,a8=>keytempi42,a9=>keytempi13,a10=>keytempi23,
a11=>keytempi33,a12=>keytempi43,a13=>keytempi14,a14=>keytempi24,a15=>keytempi
34,a16=>keytempi44,b1=>keytempj11,b2=>keytempj21,b3=>keytempj31,b4=>keytempj4
1,b5=>keytempj12,b6=>keytempj22,b7=>keytempj32,b8=>keytempj42,b9=>keytempj13,b
10=>keytempj23,b11=>keytempj33,b12=>keytempj43,b13=>keytempj14,b14=>keytempj2
4,b15=>keytempj34,b16=>keytempj44,round_num=>x"36",clk=>clk);
final_kick:add_round_key port
map(a11=>tempj11,a12=>tempj12,a13=>tempj13,a14=>tempj14,a21=>tempj21,a22=>te
mpj22,a23=>tempj23,a24=>tempj24,a31=>tempj31,a32=>tempj32,a33=>tempj33,a34=>t
empj34,a41=>tempj41,a42=>tempj42,a43=>tempj43,a44=>tempj44,key11=>keytempj11,k
ey12=>keytempj12,key13=>keytempj13,key14=>keytempj14,key21=>keytempj21,key22=>
keytempj22,key23=>keytempj23,key24=>keytempj24,key31=>keytempj31,key32=>keytem
pj32,key33=>keytempj33,key34=>keytempj34,key41=>keytempj41,key42=>keytempj42,ke
y43=>keytempj43,key44=>keytempj44,b11=>b11,b12=>b12,b13=>b13,b14=>b14,b21=>
b21,b22=>b22,b23=>b23,b24=>b24,b31=>b31,b32=>b32,b33=>b33,b34=>b34,b41=>b
41,b42=>b42,b43=>b43,b44=>b44,clk=>clk);
end Behavioral;
```

**Test Bench**

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_big_round is
--    Port ( );
end test_big_round;

architecture Behavioral of test_big_round is
component big_round is
        Port ( clk : in STD_LOGIC;
a11 : in STD_LOGIC_VECTOR (7 downto 0);
a12 : in STD_LOGIC_VECTOR (7 downto 0);
a13 : in STD_LOGIC_VECTOR (7 downto 0);
a14 : in STD_LOGIC_VECTOR (7 downto 0);
a21 : in STD_LOGIC_VECTOR (7 downto 0);
a22 : in STD_LOGIC_VECTOR (7 downto 0);
a23 : in STD_LOGIC_VECTOR (7 downto 0);
a24 : in STD_LOGIC_VECTOR (7 downto 0);
a31 : in STD_LOGIC_VECTOR (7 downto 0);
a32 : in STD_LOGIC_VECTOR (7 downto 0);
a33 : in STD_LOGIC_VECTOR (7 downto 0);
a34 : in STD_LOGIC_VECTOR (7 downto 0);
a41 : in STD_LOGIC_VECTOR (7 downto 0);
a42 : in STD_LOGIC_VECTOR (7 downto 0);
a43 : in STD_LOGIC_VECTOR (7 downto 0);
a44 : in STD_LOGIC_VECTOR (7 downto 0);
key11 : in STD_LOGIC_VECTOR (7 downto 0);
key12 : in STD_LOGIC_VECTOR (7 downto 0);
key13 : in STD_LOGIC_VECTOR (7 downto 0);
key14 : in STD_LOGIC_VECTOR (7 downto 0);
key21 : in STD_LOGIC_VECTOR (7 downto 0);
key22 : in STD_LOGIC_VECTOR (7 downto 0);
key23 : in STD_LOGIC_VECTOR (7 downto 0);
key24 : in STD_LOGIC_VECTOR (7 downto 0);
key31 : in STD_LOGIC_VECTOR (7 downto 0);
key32 : in STD_LOGIC_VECTOR (7 downto 0);
key33 : in STD_LOGIC_VECTOR (7 downto 0);
key34 : in STD_LOGIC_VECTOR (7 downto 0);
key41 : in STD_LOGIC_VECTOR (7 downto 0);
key42 : in STD_LOGIC_VECTOR (7 downto 0);
key43 : in STD_LOGIC_VECTOR (7 downto 0);
key44 : in STD_LOGIC_VECTOR (7 downto 0);
b11 : out STD_LOGIC_VECTOR (7 downto 0);
b12 : out STD_LOGIC_VECTOR (7 downto 0);
b13 : out STD_LOGIC_VECTOR (7 downto 0);
b14 : out STD_LOGIC_VECTOR (7 downto 0);
b21 : out STD_LOGIC_VECTOR (7 downto 0);
b22 : out STD_LOGIC_VECTOR (7 downto 0);
b23 : out STD_LOGIC_VECTOR (7 downto 0);
b24 : out STD_LOGIC_VECTOR (7 downto 0);
b31 : out STD_LOGIC_VECTOR (7 downto 0);
b32 : out STD_LOGIC_VECTOR (7 downto 0);
b33 : out STD_LOGIC_VECTOR (7 downto 0);
b34 : out STD_LOGIC_VECTOR (7 downto 0);
b41 : out STD_LOGIC_VECTOR (7 downto 0);
b42 : out STD_LOGIC_VECTOR (7 downto 0);
b43 : out STD_LOGIC_VECTOR (7 downto 0);
b44 : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal
a11,a12,a13,a14,a21,a22,a23,a24,a31,a32,a33,a34,a41,a42,a43,a44,key11,key12,key13,key14,
key21,key22,key23,key24,key31,key32,key33,key34,key41,key42,key43,key44,b11,b12,b13,b1
4,b21,b22,b23,b24,b31,b32,b33,b34,b41,b42,b43,b44:std_logic_vector(7 downto 0);
signal clk:std_logic;
begin
big_rnd:big_round port
map(a11=>a11,a12=>a12,a13=>a13,a14=>a14,a21=>a21,a22=>a22,a23=>a23,a24=>a24,
a31=>a31,a32=>a32,a33=>a33,a34=>a34,a41=>a41,a42=>a42,a43=>a43,a44=>a44,key1
1=>key11,key12=>key12,key13=>key13,key14=>key14,key21=>key21,key22=>key22,key2
3=>key23,key24=>key24,key31=>key31,key32=>key32,key33=>key33,key34=>key34,key4
1=>key41,key42=>key42,key43=>key43,key44=>key44,b11=>b11,b12=>b12,b13=>b13,b
14=>b14,b21=>b21,b22=>b22,b23=>b23,b24=>b24,b31=>b31,b32=>b32,b33=>b33,b34
=>b34,b41=>b41,b42=>b42,b43=>b43,b44=>b44,clk=>clk);
process
begin
clk<='0';
wait for 5ns;
clk<='1';
wait for 5ns;
end process;
process
begin
a11<=x"54";a12<=x"4f";a13<=x"4e";a14<=x"20";
a21<=x"77";a22<=x"6e";a23<=x"69";a24<=x"54";
a31<=x"6f";a32<=x"65";a33<=x"6e";a34<=x"77";
a41<=x"20";a42<=x"20";a43<=x"65";a44<=x"6f";
```

```
key11<=x"54";key12<=x"73";key13<=x"20";key14<=x"67";
key21<=x"68";key22<=x"20";key23<=x"4b";key24<=x"20";
key31<=x"61";key32<=x"6d";key33<=x"75";key34<=x"46";
key41<=x"74";key42<=x"79";key43<=x"6e";key44<=x"75";
wait for 500ns;
end process;
end Behavioral;
```

## Simulation Result



```
b11[7:0]   29
b12[7:0]   57
b13[7:0]   40
b14[7:0]   1a
b21[7:0]   c3
b22[7:0]   14
b23[7:0]   22
b24[7:0]   02
b31[7:0]   50
b32[7:0]   20
b33[7:0]   99
b34[7:0]   d7
b41[7:0]   5f
b42[7:0]   f6
b43[7:0]   b3
b44[7:0]   3a
clk        0
```

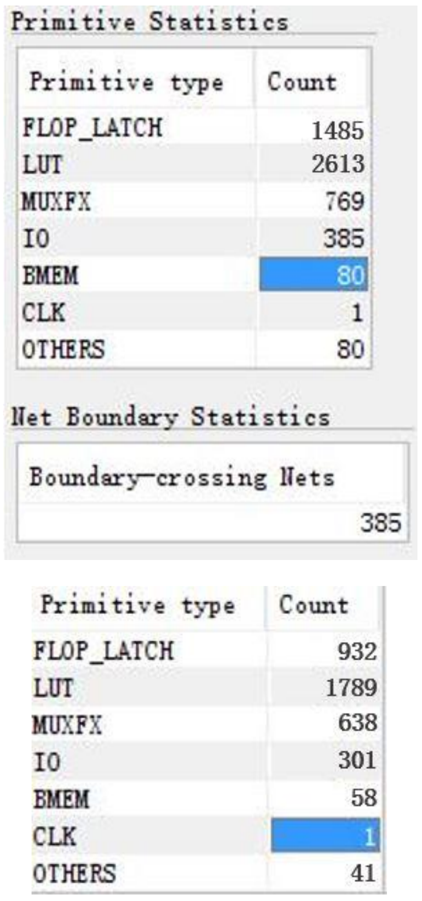
Figure10

Figure10 shows the simulation result of the big.

## 4. Optimization & Conclusion

At first time, I used the concept of FSM, the resource cost is high. Soon after, I apply the concept of assembly line. The cost of the resources is lowered, however, at the same time, the time consumption is increased.

## 5. Resource Comparison



Primitive Statistics

| Primitive type | Count |
|---|---|
| FLOP_LATCH | 1485 |
| LUT | 2613 |
| MUXFX | 769 |
| IO | 385 |
| BMEM | 80 |
| CLK | 1 |
| OTHERS | 80 |

Net Boundary Statistics

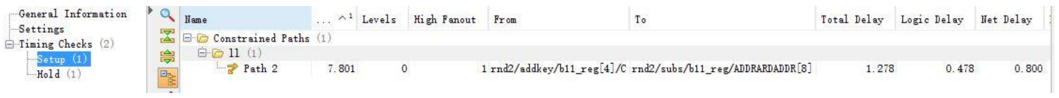
| Boundary-crossing Nets |
|---|
| 385 |

| Primitive type | Count |
|---|---|
| FLOP_LATCH | 932 |
| LUT | 1789 |
| MUXFX | 638 |
| IO | 301 |
| BMEM | 58 |
| CLK | 1 |
| OTHERS | 41 |

Figure11

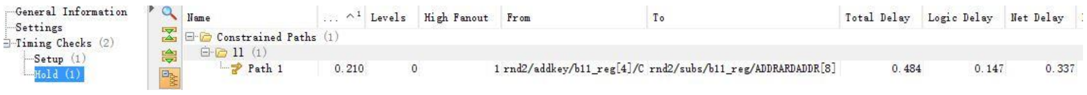
Figure11 shows the comparison of the two systems. The cost of the resource of the optimized one is much lower than the former one.

## 6. Time Consumption



General Information
Settings
Timing Checks (2)
Setup (1)
Hold (1)

| Name | ... ^1 | Levels | High Fanout | From | To | Total Delay | Logic Delay | Net Delay |
|---|---|---|---|---|---|---|---|---|
| Constrained Paths (1) | | | | | | | | |
| 11 (1) | | | | | | | | |
| Path 2 | 7.801 | 0 | 1 | rnd2/addkey/b11_reg[4]/C | rnd2/subs/b11_reg/ADDRARDADDR[8] | 1.278 | 0.478 | 0.800 |



General Information
Settings
Timing Checks (2)
Setup (1)
Hold (1)

| Name | ... ^1 | Levels | High Fanout | From | To | Total Delay | Logic Delay | Net Delay |
|---|---|---|---|---|---|---|---|---|
| Constrained Paths (1) | | | | | | | | |
| 11 (1) | | | | | | | | |
| Path 1 | 0.210 | 0 | 1 | rnd2/addkey/b11_reg[4]/C | rnd2/subs/b11_reg/ADDRARDADDR[8] | 0.484 | 0.147 | 0.337 |