Lab 2: Encryption and Decryption without Pre Round

EE4313 A

Raphael Kumar 0259929

Mikhail Istomin 0261785

Shing Cheng 0267834

10/20/2005

Abstract

The objective of this lab was to design and implement RC5 Encryption and Decryption without a pre or post round using fixed round keys with 64-bit input and output.

Procedure

The operation that was attempted for the first part of the exercise was as follows.

for i = 1 to 12 do

A = ((A xor B) <<< B) + S[2*i];

B = ((B xor A) <<< A) + S[2*i+1];

where i represents the current round of the 12 total rounds. The round keys were stored in a ROM and the appropriate SKEY was accessed utilizing the variable i which was incremented each round. The SKEYS were accessed using the following notation: skey(CONV_INTEGER(i_cnt)). The encryption portion of the program was created with three separate PROCESSes and a loop to go through all 12 rounds. After the program was completed, it was simulated with ModelSim to check its functional operation. Once its correct operation was verified, it was synthesized with Xilinx Project Navigator. This simulation created a timing model which could be used for timing analysis. The timing simulation was performed with ModelSim to determine if any glitches were present and to find delay.

For the second exercise, the decryption operation was to be done, without a post round.

for i = 12 to 1 do

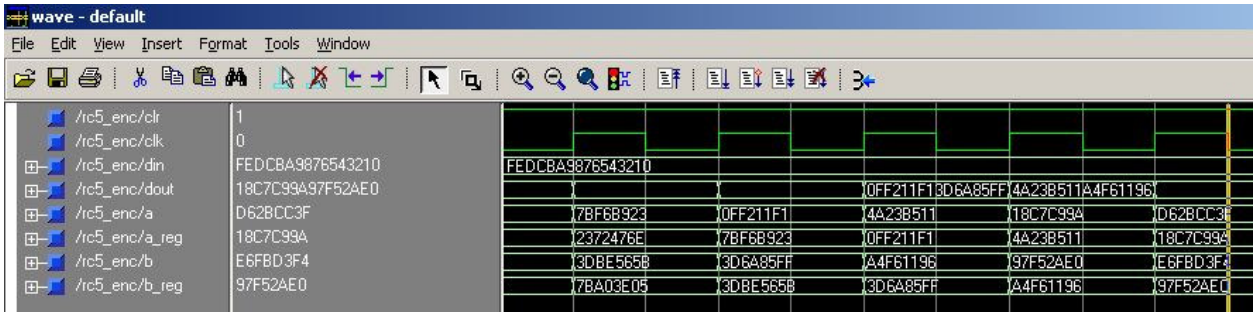
B = ((B – S[2*i+1]) >>> A) xor A;

A= ((A – S[2*i]) >>> B) xor B;

Just as above, a functional simulation was done with ModelSim, and a synthesis performed with Xilinx Project Navigator, with a timing simulation performed in ModelSim.

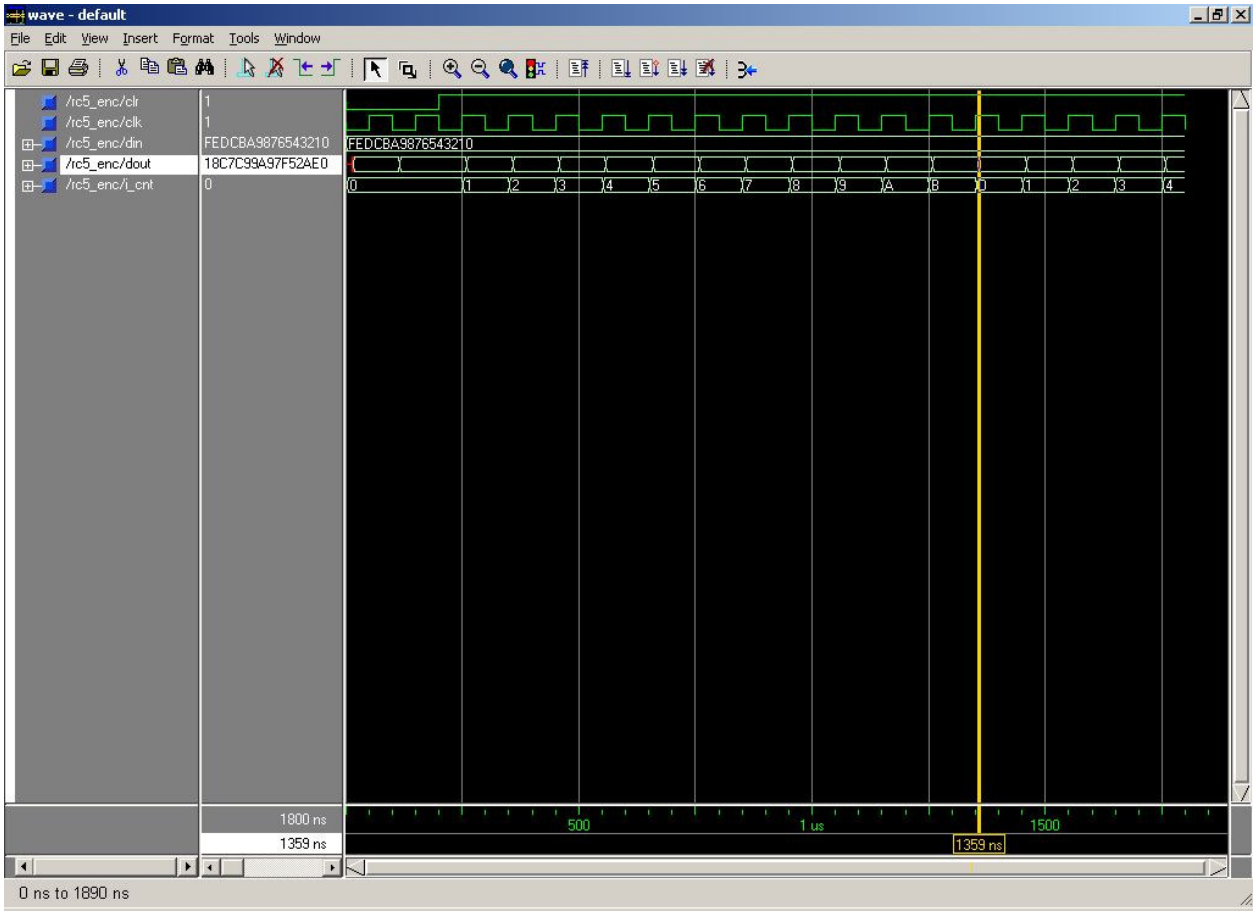
Data and Observations

The code used to perform both the encryption and decryption functions accompanies this document. Below is a snapshot of the functional simulation performed with the encryption code.



Full size versions of all snapshots accompany the report. The 64-bit input used in this example was FEDCBA9876543210 and the resulting encrypted output (without a pre round) was 18C799A97F52AE0. A sequential circuit is needed in this program because of the need for more than one round. Each round relies on the previous round, and therefore must utilize sequential circuits. Also, for the operations done to A and B, the second operation is dependent on the result from the first. CONV_INTEGER takes the parameter and converts it to an integer, which is necessary to look up the appropriate SKEY.

The timing simulation performed gave the following snapshot.



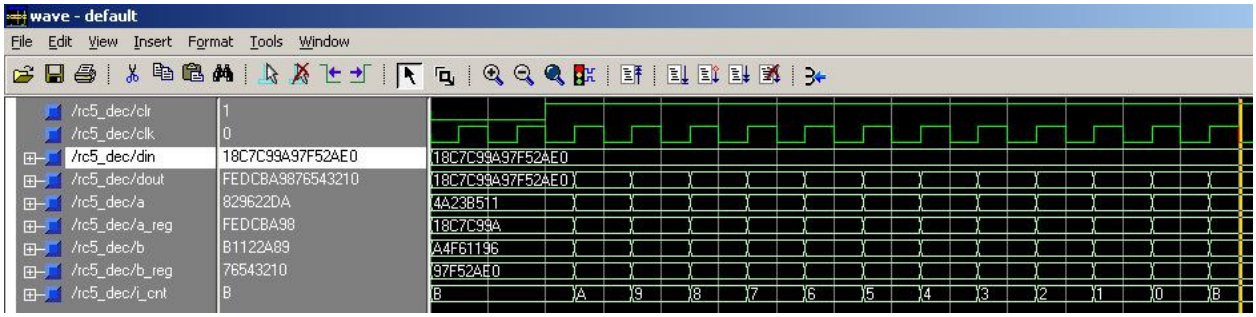
As can be seen in the above figure, the delay was 9 nanoseconds. The only noticeable glitch occurs in the very beginning of the program before the input is allowed to propagate. An examination of the timing and area reports gives the following.

Number of Slices:	367 out of 20480	1%
Number of Slice Flip Flops:	88 out of 40960	0%
Number of 4 input LUTs:	688 out of 40960	1%
Number of bonded IOBs:	129 out of 489	26%
Number of GCLKs:	1 out of 8	12%

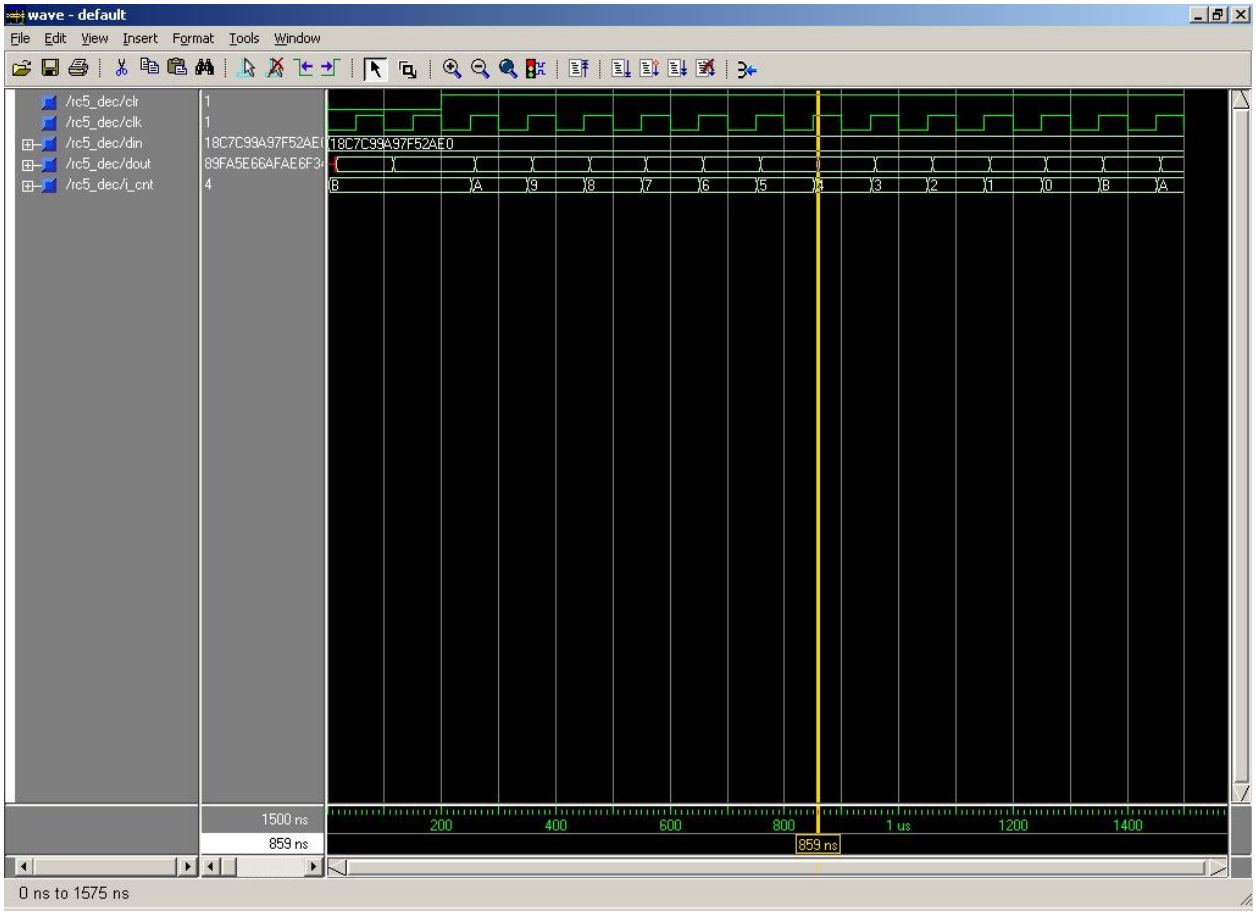
According to the timing report, the delay is 16.609 nanoseconds through 49 levels of logic, and the design was given a speed grade of -4. According to the report, 62.5% of the time was spent on logic, while 37.5% was spent on route. The total memory usage is 101088 kilobytes. It can

be inferred from the timing report that a very large amount of time was spent on logic, more than half. A simplification in logic could result in a smaller delay.

For the second exercise, a decryption operation was implemented without a post round. The functional simulation for this operation was performed utilizing the output of the previous operation.



The input for the 64-bit decryption was 187C99A97F52AE0 and the output was the original unencrypted plaintext, FEDCBA9876543210. A timing simulation was performed on the code and the following figure was produced.



According to the timing simulation performed in ModelSim, the delay is 9 nanoseconds, as can be seen in the above figure. The timing and area reports give the following information.

Number of Slices:	356 out of 20480	1%
Number of Slice Flip Flops:	77 out of 40960	0%
Number of 4 input LUTs:	671 out of 40960	1%
Number of bonded IOBs:	129 out of 489	26%
Number of GCLKs:	1 out of 8	12%

According to the timing report, the total delay of the design is 14.451 nanoseconds through 30 levels of logic, with a speed grade of -4. Of the total delay, 56.7% was spent on logic, and 43.3% spent on route. The total memory usage was 101088 kilobytes.

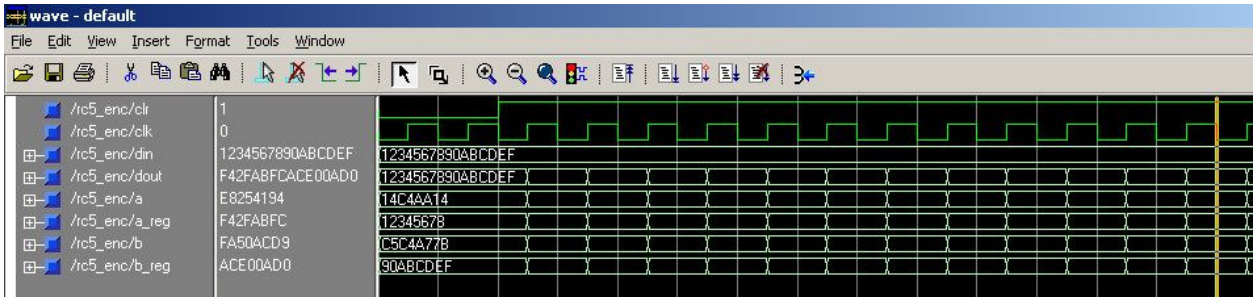
It can be inferred that the amount of time spent on logic created a significant portion of the delay. While there are less levels of logic than the encryption operation, simplification of the logic used could theoretically result in a more efficient design with less timing delay.

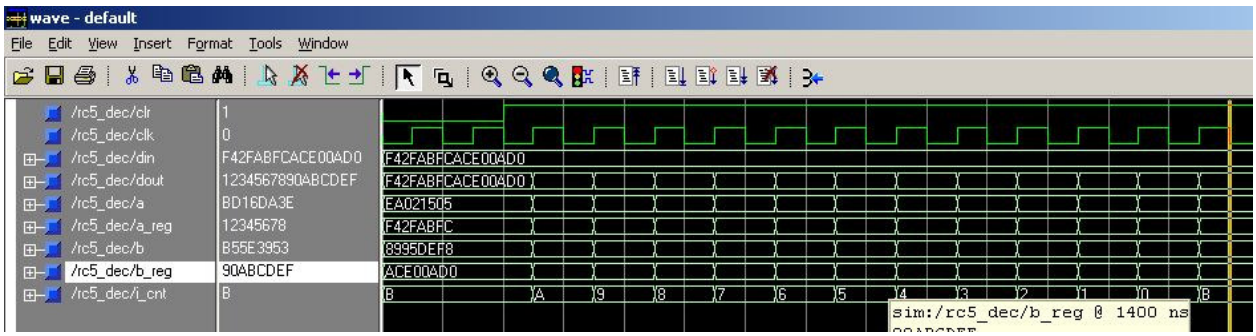
Two more test vectors were encrypted then decrypted to test whether the functions performed correctly. Here are the unencrypted and encrypted 64-bit vectors, including the first input test vector used.

Plaintext	Crypto text
FEDCBA1234567890	18C7C99A97F52AE
1234567890ABCDEF	F42FABFCACE00AD0
34D12E56F78A90BC	303D84D9D5003C1C

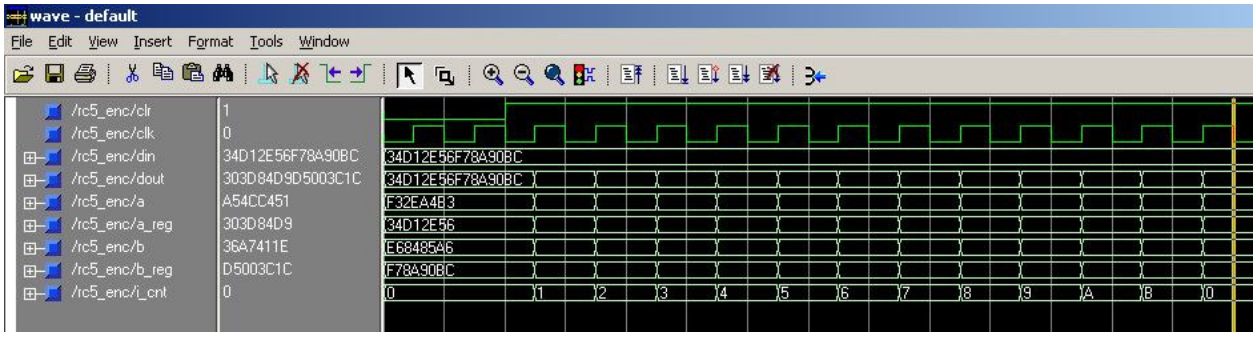
The decryption code was used to decrypt the resulting cipher text and the result was the original plaintext. The following figures show the encryption and decryption of the final two test vectors while the first vector can be seen above.

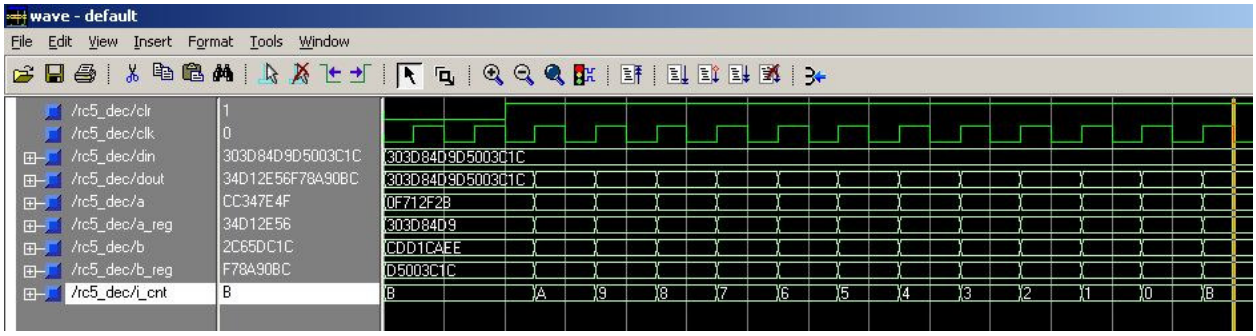
1234567890ABCDEF	F42FABFCACE00AD0





34D12E56F78A90BC 303D84D9D5003C1C





The experiment was a success with no major problems encountered. One problem involved the inability to produce a connected FPGA Synthesis Diagram with Project Navigator. Aside from this, the encryption and decryption performed as expected.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;   -- we will use CONV_INTEGER
ENTITY rc5_enc IS
PORT
(
clr : IN STD_LOGIC;  -- Asynchronous reset
clk : IN STD_LOGIC;  -- Clock signal
din : IN STD_LOGIC_VECTOR(63 DOWNTO 0);  -- 64-bit input
dout : OUT STD_LOGIC_VECTOR(63 DOWNTO 0) -- 64-bit output
);
END rc5_enc;

ARCHITECTURE rtl OF rc5_enc IS
SIGNAL i_cnt : STD_LOGIC_VECTOR(3 DOWNTO 0); -- round counter
SIGNAL ab_xor : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL a_rot : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL a : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL a_reg : STD_LOGIC_VECTOR(31 DOWNTO 0); -- register A
SIGNAL ba_xor: STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL b_rot : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL b : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL b_reg : STD_LOGIC_VECTOR(31 DOWNTO 0); -- register B
-- define a type for round keys
TYPE rom IS ARRAY (0 TO 23) OF STD_LOGIC_VECTOR(31 DOWNTO 0);
CONSTANT skey : rom:=rom'(X"46F8E8C5", X"460C6085", X"70F83B8A",
X"284B8303", X"513E1454", X"F621ED22",
X"3125065D", X"11A83A5D", X"D427686B",
X"713AD82D", X"4B792F99", X"2799A4DD",
X"A7901C49", X"DEDE871A", X"36C03196",
X"A7EFC249", X"61A78BB8", X"3B0A1D2B",
X"4DBFCA76", X"AE162167", X"30D76B0A",
X"43192304", X"F6CC1431", X"65046380" );

BEGIN
-- A=((A XOR B)<<<B) + S[2*i];
ab_xor <= a_reg XOR b_reg;
WITH b_reg(4 DOWNTO 0) SELECT
a_rot<=ab_xor(30 DOWNTO 0) & ab_xor(31) WHEN "00001",
ab_xor(29 DOWNTO 0) & ab_xor(31 DOWNTO 30) WHEN "00010",
ab_xor(28 DOWNTO 0) & ab_xor(31 DOWNTO 29) WHEN "00011",
ab_xor(27 DOWNTO 0) & ab_xor(31 DOWNTO 28) WHEN "00100",
ab_xor(26 DOWNTO 0) & ab_xor(31 DOWNTO 27) WHEN "00101",
ab_xor(25 DOWNTO 0) & ab_xor(31 DOWNTO 26) WHEN "00110",
ab_xor(24 DOWNTO 0) & ab_xor(31 DOWNTO 25) WHEN "00111",
ab_xor(23 DOWNTO 0) & ab_xor(31 DOWNTO 24) WHEN "01000",
ab_xor(22 DOWNTO 0) & ab_xor(31 DOWNTO 23) WHEN "01001",
ab_xor(21 DOWNTO 0) & ab_xor(31 DOWNTO 22) WHEN "01010",
ab_xor(20 DOWNTO 0) & ab_xor(31 DOWNTO 21) WHEN "01011",
ab_xor(19 DOWNTO 0) & ab_xor(31 DOWNTO 20) WHEN "01100",
ab_xor(18 DOWNTO 0) & ab_xor(31 DOWNTO 19) WHEN "01101",
ab_xor(17 DOWNTO 0) & ab_xor(31 DOWNTO 18) WHEN "01110",
ab_xor(16 DOWNTO 0) & ab_xor(31 DOWNTO 17) WHEN "01111",
ab_xor(15 DOWNTO 0) & ab_xor(31 DOWNTO 16) WHEN "10000",
ab_xor(14 DOWNTO 0) & ab_xor(31 DOWNTO 15) WHEN "10001",
ab_xor(13 DOWNTO 0) & ab_xor(31 DOWNTO 14) WHEN "10010",
ab_xor(12 DOWNTO 0) & ab_xor(31 DOWNTO 13) WHEN "10011",
ab_xor(11 DOWNTO 0) & ab_xor(31 DOWNTO 12) WHEN "10100",
ab_xor(10 DOWNTO 0) & ab_xor(31 DOWNTO 11) WHEN "10101",
ab_xor(9 DOWNTO 0) & ab_xor(31 DOWNTO 10) WHEN "10110",
ab_xor(8 DOWNTO 0) & ab_xor(31 DOWNTO 9) WHEN "10111",
```

```
ab_xor(7 DOWNTO 0) & ab_xor(31 DOWNTO 8) WHEN "11000",
ab_xor(6 DOWNTO 0) & ab_xor(31 DOWNTO 7) WHEN "11001",
ab_xor(5 DOWNTO 0) & ab_xor(31 DOWNTO 6) WHEN "11010",
ab_xor(4 DOWNTO 0) & ab_xor(31 DOWNTO 5) WHEN "11011",
ab_xor(3 DOWNTO 0) & ab_xor(31 DOWNTO 4) WHEN "11100",
ab_xor(2 DOWNTO 0) & ab_xor(31 DOWNTO 3) WHEN "11101",
ab_xor(1 DOWNTO 0) & ab_xor(31 DOWNTO 2) WHEN "11110",
ab_xor(0) & ab_xor(31 DOWNTO 1) WHEN "11111",
ab_xor WHEN OTHERS;
a<=a_rot + skey(CONV_INTEGER(i_cnt & '0'));

-- B=((B XOR A) <<<A) + S[2*i+1]
ba_xor <= b_reg XOR a;
WITH a(4 DOWNTO 0) SELECT
b_rot<=ba_xor(30 DOWNTO 0) & ba_xor(31) WHEN "00001",
ba_xor(29 DOWNTO 0) & ba_xor(31 DOWNTO 30) WHEN "00010",
ba_xor(28 DOWNTO 0) & ba_xor(31 DOWNTO 29) WHEN "00011",
ba_xor(27 DOWNTO 0) & ba_xor(31 DOWNTO 28) WHEN "00100",
ba_xor(26 DOWNTO 0) & ba_xor(31 DOWNTO 27) WHEN "00101",
ba_xor(25 DOWNTO 0) & ba_xor(31 DOWNTO 26) WHEN "00110",
ba_xor(24 DOWNTO 0) & ba_xor(31 DOWNTO 25) WHEN "00111",
ba_xor(23 DOWNTO 0) & ba_xor(31 DOWNTO 24) WHEN "01000",
ba_xor(22 DOWNTO 0) & ba_xor(31 DOWNTO 23) WHEN "01001",
ba_xor(21 DOWNTO 0) & ba_xor(31 DOWNTO 22) WHEN "01010",
ba_xor(20 DOWNTO 0) & ba_xor(31 DOWNTO 21) WHEN "01011",
ba_xor(19 DOWNTO 0) & ba_xor(31 DOWNTO 20) WHEN "01100",
ba_xor(18 DOWNTO 0) & ba_xor(31 DOWNTO 19) WHEN "01101",
ba_xor(17 DOWNTO 0) & ba_xor(31 DOWNTO 18) WHEN "01110",
ba_xor(16 DOWNTO 0) & ba_xor(31 DOWNTO 17) WHEN "01111",
ba_xor(15 DOWNTO 0) & ba_xor(31 DOWNTO 16) WHEN "10000",
ba_xor(14 DOWNTO 0) & ba_xor(31 DOWNTO 15) WHEN "10001",
ba_xor(13 DOWNTO 0) & ba_xor(31 DOWNTO 14) WHEN "10010",
ba_xor(12 DOWNTO 0) & ba_xor(31 DOWNTO 13) WHEN "10011",
ba_xor(11 DOWNTO 0) & ba_xor(31 DOWNTO 12) WHEN "10100",
ba_xor(10 DOWNTO 0) & ba_xor(31 DOWNTO 11) WHEN "10101",
ba_xor(9 DOWNTO 0) & ba_xor(31 DOWNTO 10) WHEN "10110",
ba_xor(8 DOWNTO 0) & ba_xor(31 DOWNTO 9) WHEN "10111",
ba_xor(7 DOWNTO 0) & ba_xor(31 DOWNTO 8) WHEN "11000",
ba_xor(6 DOWNTO 0) & ba_xor(31 DOWNTO 7) WHEN "11001",
ba_xor(5 DOWNTO 0) & ba_xor(31 DOWNTO 6) WHEN "11010",
ba_xor(4 DOWNTO 0) & ba_xor(31 DOWNTO 5) WHEN "11011",
ba_xor(3 DOWNTO 0) & ba_xor(31 DOWNTO 4) WHEN "11100",
ba_xor(2 DOWNTO 0) & ba_xor(31 DOWNTO 3) WHEN "11101",
ba_xor(1 DOWNTO 0) & ba_xor(31 DOWNTO 2) WHEN "11110",
ba_xor(0) & ba_xor(31 DOWNTO 1) WHEN "11111",
ba_xor WHEN OTHERS;

b<=b_rot + skey(CONV_INTEGER(i_cnt & '1'));  -- S[2*i+1]

-- A register
PROCESS(clr, clk)  BEGIN
IF(clr='0') THEN
a_reg<=din(63 DOWNTO 32);
ELSIF(clk'EVENT AND clk='1') THEN
a_reg<=a;
END IF;
END PROCESS;
-- B register
PROCESS(clr, clk)  BEGIN
IF(clr='0') THEN
b_reg<=din(31 DOWNTO 0);
ELSIF(clk'EVENT AND clk='1') THEN
b_reg<=b;
```

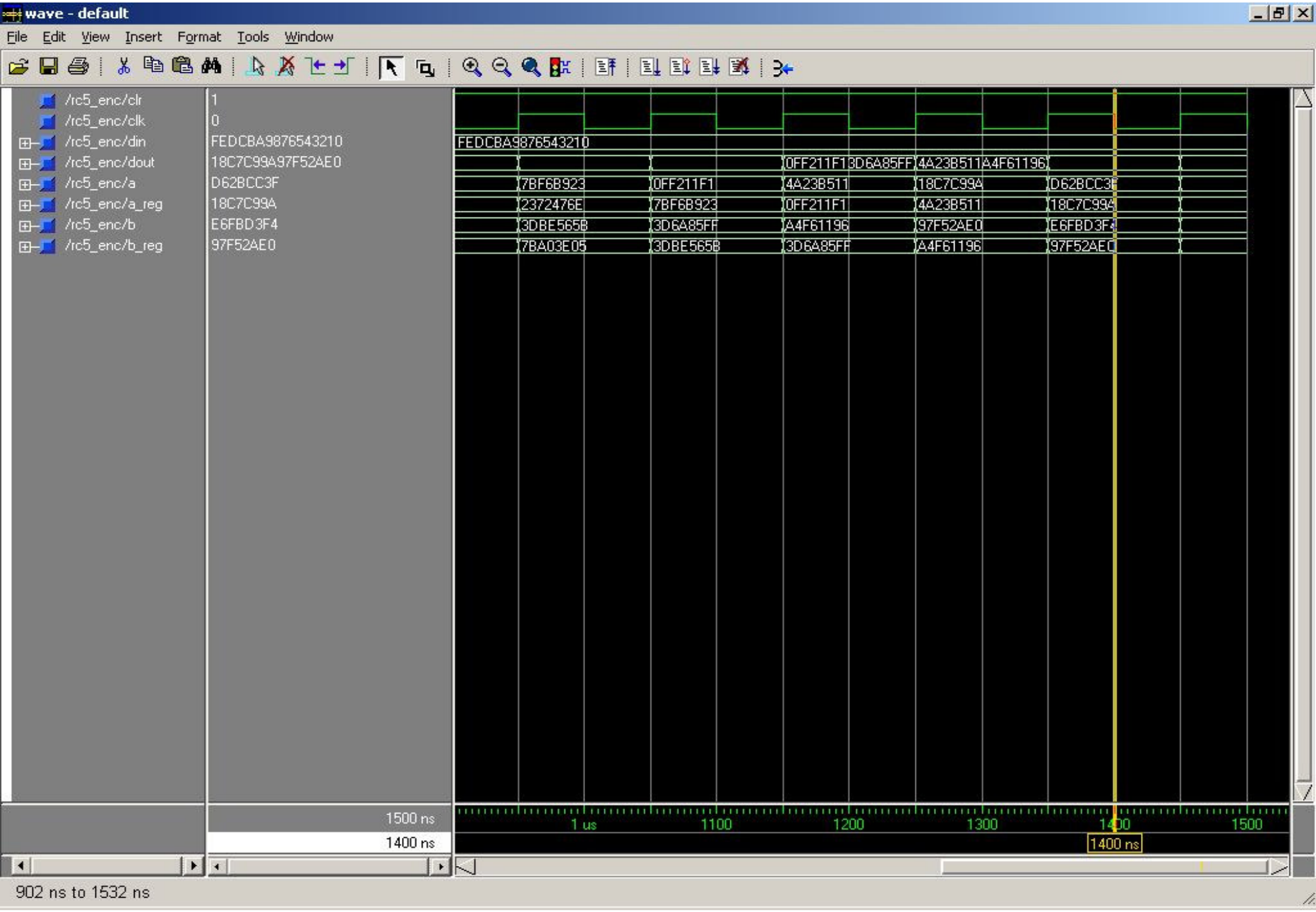
```
END IF;
END PROCESS;

-- round counter
PROCESS(clr, clk)  BEGIN
IF(clr='0') THEN
i_cnt<="0000";
ELSIF(clk'EVENT AND clk='1') THEN
IF(i_cnt="1011") THEN
i_cnt<="0000";
ELSE
i_cnt<=i_cnt+'1';
END IF;
END IF;
END PROCESS;
dout<=a_reg & b_reg;
END rtl;
```

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;   -- we will use CONV_INTEGER
ENTITY rc5_dec IS
PORT
(
clr : IN STD_LOGIC;  -- Asynchronous reset
clk : IN STD_LOGIC;  -- Clock signal
din : IN STD_LOGIC_VECTOR(63 DOWNTO 0);  -- 64-bit input
dout : OUT STD_LOGIC_VECTOR(63 DOWNTO 0) -- 64-bit output
);
END rc5_dec;

ARCHITECTURE rtl OF rc5_dec IS
SIGNAL i_cnt : STD_LOGIC_VECTOR(3 DOWNTO 0); -- round counter
SIGNAL bskey_minus : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL b_rot : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL a : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL a_reg : STD_LOGIC_VECTOR(31 DOWNTO 0); -- register A
SIGNAL ba_xor: STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL askey_minus : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL a_rot : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL b : STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL b_reg : STD_LOGIC_VECTOR(31 DOWNTO 0); -- register B

-- define a type for round keys
TYPE rom IS ARRAY (0 TO 23) OF STD_LOGIC_VECTOR(31 DOWNTO 0);
CONSTANT skey : rom:=rom'(X"46F8E8C5", X"460C6085", X"70F83B8A",
X"284B8303", X"513E1454", X"F621ED22",
X"3125065D", X"11A83A5D", X"D427686B",
X"713AD82D", X"4B792F99", X"2799A4DD",
X"A7901C49", X"DEDE871A", X"36C03196",
X"A7EFC249", X"61A78BB8", X"3B0A1D2B",
X"4DBFCA76", X"AE162167", X"30D76B0A",
X"43192304", X"F6CC1431", X"65046380" );


BEGIN


-- B=((B - S[2*i+1]) >>> A) XOR A

bskey_minus<=b_reg - skey(CONV_INTEGER(i_cnt & '1'));  -- S[2*i+1]
WITH a_reg(4 DOWNTO 0) SELECT
b_rot<= bskey_minus(0) & bskey_minus (31 DOWNTO 1) WHEN "00001",
bskey_minus(1 DOWNTO 0) & bskey_minus (31 DOWNTO 2) WHEN "00010",
bskey_minus(2 DOWNTO 0) & bskey_minus (31 DOWNTO 3) WHEN "00011",
bskey_minus(3 DOWNTO 0) & bskey_minus (31 DOWNTO 4) WHEN "00100",
bskey_minus(4 DOWNTO 0) & bskey_minus (31 DOWNTO 5) WHEN "00101",
bskey_minus(5 DOWNTO 0) & bskey_minus (31 DOWNTO 6) WHEN "00110",
bskey_minus(6 DOWNTO 0) & bskey_minus (31 DOWNTO 7) WHEN "00111",
bskey_minus(7 DOWNTO 0) & bskey_minus (31 DOWNTO 8) WHEN "01000",
bskey_minus(8 DOWNTO 0) & bskey_minus (31 DOWNTO 9) WHEN "01001",
bskey_minus(9 DOWNTO 0) & bskey_minus (31 DOWNTO 10) WHEN "01010",
bskey_minus(10 DOWNTO 0) & bskey_minus (31 DOWNTO 11) WHEN "01011",
bskey_minus(11 DOWNTO 0) & bskey_minus (31 DOWNTO 12) WHEN "01100",
bskey_minus(12 DOWNTO 0) & bskey_minus (31 DOWNTO 13) WHEN "01101",
bskey_minus(13 DOWNTO 0) & bskey_minus (31 DOWNTO 14) WHEN "01110",
bskey_minus(14 DOWNTO 0) & bskey_minus (31 DOWNTO 15) WHEN "01111",
bskey_minus(15 DOWNTO 0) & bskey_minus (31 DOWNTO 16) WHEN "10000",
bskey_minus(16 DOWNTO 0) & bskey_minus (31 DOWNTO 17) WHEN "10001",
```

```
bskey_minus(17 DOWNTO 0) & bskey_minus (31 DOWNTO 18) WHEN "10010",
bskey_minus(18 DOWNTO 0) & bskey_minus (31 DOWNTO 19) WHEN "10011",
bskey_minus(19 DOWNTO 0) & bskey_minus (31 DOWNTO 20) WHEN "10100",
bskey_minus(20 DOWNTO 0) & bskey_minus (31 DOWNTO 21) WHEN "10101",
bskey_minus(21 DOWNTO 0) & bskey_minus (31 DOWNTO 22) WHEN "10110",
bskey_minus(22 DOWNTO 0) & bskey_minus (31 DOWNTO 23) WHEN "10111",
bskey_minus(23 DOWNTO 0) & bskey_minus (31 DOWNTO 24) WHEN "11000",
bskey_minus(24 DOWNTO 0) & bskey_minus (31 DOWNTO 25) WHEN "11001",
bskey_minus(25 DOWNTO 0) & bskey_minus (31 DOWNTO 26) WHEN "11010",
bskey_minus(26 DOWNTO 0) & bskey_minus (31 DOWNTO 27) WHEN "11011",
bskey_minus(27 DOWNTO 0) & bskey_minus (31 DOWNTO 28) WHEN "11100",
bskey_minus(28 DOWNTO 0) & bskey_minus (31 DOWNTO 29) WHEN "11101",
bskey_minus(29 DOWNTO 0) & bskey_minus (31 DOWNTO 30) WHEN "11110",
bskey_minus(30 DOWNTO 0) & bskey_minus (31) WHEN "11111",
bskey_minus WHEN OTHERS;
b <= b_rot XOR a_reg;



-- A=((A - S[2*i]) >>> B) XOR B;
askey_minus<=a_reg - skey(CONV_INTEGER(i_cnt & '0'));
WITH b(4 DOWNTO 0) SELECT
a_rot <= askey_minus(0) & askey_minus (31 DOWNTO 1) WHEN "00001",
askey_minus(1 DOWNTO 0) & askey_minus (31 DOWNTO 2) WHEN "00010",
askey_minus(2 DOWNTO 0) & askey_minus (31 DOWNTO 3) WHEN "00011",
askey_minus(3 DOWNTO 0) & askey_minus (31 DOWNTO 4) WHEN "00100",
askey_minus(4 DOWNTO 0) & askey_minus (31 DOWNTO 5) WHEN "00101",
askey_minus(5 DOWNTO 0) & askey_minus (31 DOWNTO 6) WHEN "00110",
askey_minus(6 DOWNTO 0) & askey_minus (31 DOWNTO 7) WHEN "00111",
askey_minus(7 DOWNTO 0) & askey_minus (31 DOWNTO 8) WHEN "01000",
askey_minus(8 DOWNTO 0) & askey_minus (31 DOWNTO 9) WHEN "01001",
askey_minus(9 DOWNTO 0) & askey_minus (31 DOWNTO 10) WHEN "01010",
askey_minus(10 DOWNTO 0) & askey_minus (31 DOWNTO 11) WHEN "01011",
askey_minus(11 DOWNTO 0) & askey_minus (31 DOWNTO 12) WHEN "01100",
askey_minus(12 DOWNTO 0) & askey_minus (31 DOWNTO 13) WHEN "01101",
askey_minus(13 DOWNTO 0) & askey_minus (31 DOWNTO 14) WHEN "01110",
askey_minus(14 DOWNTO 0) & askey_minus (31 DOWNTO 15) WHEN "01111",
askey_minus(15 DOWNTO 0) & askey_minus (31 DOWNTO 16) WHEN "10000",
askey_minus(16 DOWNTO 0) & askey_minus (31 DOWNTO 17) WHEN "10001",
askey_minus(17 DOWNTO 0) & askey_minus (31 DOWNTO 18) WHEN "10010",
askey_minus(18 DOWNTO 0) & askey_minus (31 DOWNTO 19) WHEN "10011",
askey_minus(19 DOWNTO 0) & askey_minus (31 DOWNTO 20) WHEN "10100",
askey_minus(20 DOWNTO 0) & askey_minus (31 DOWNTO 21) WHEN "10101",
askey_minus(21 DOWNTO 0) & askey_minus (31 DOWNTO 22) WHEN "10110",
askey_minus(22 DOWNTO 0) & askey_minus (31 DOWNTO 23) WHEN "10111",
askey_minus(23 DOWNTO 0) & askey_minus (31 DOWNTO 24) WHEN "11000",
askey_minus(24 DOWNTO 0) & askey_minus (31 DOWNTO 25) WHEN "11001",
askey_minus(25 DOWNTO 0) & askey_minus (31 DOWNTO 26) WHEN "11010",
askey_minus(26 DOWNTO 0) & askey_minus (31 DOWNTO 27) WHEN "11011",
askey_minus(27 DOWNTO 0) & askey_minus (31 DOWNTO 28) WHEN "11100",
askey_minus(28 DOWNTO 0) & askey_minus (31 DOWNTO 29) WHEN "11101",
askey_minus(29 DOWNTO 0) & askey_minus (31 DOWNTO 30) WHEN "11110",
askey_minus(30 DOWNTO 0) & askey_minus (31) WHEN "11111",
askey_minus WHEN OTHERS;
a <= a_rot XOR b;



-- A register
PROCESS(clr, clk)  BEGIN
IF(clr='0') THEN
```

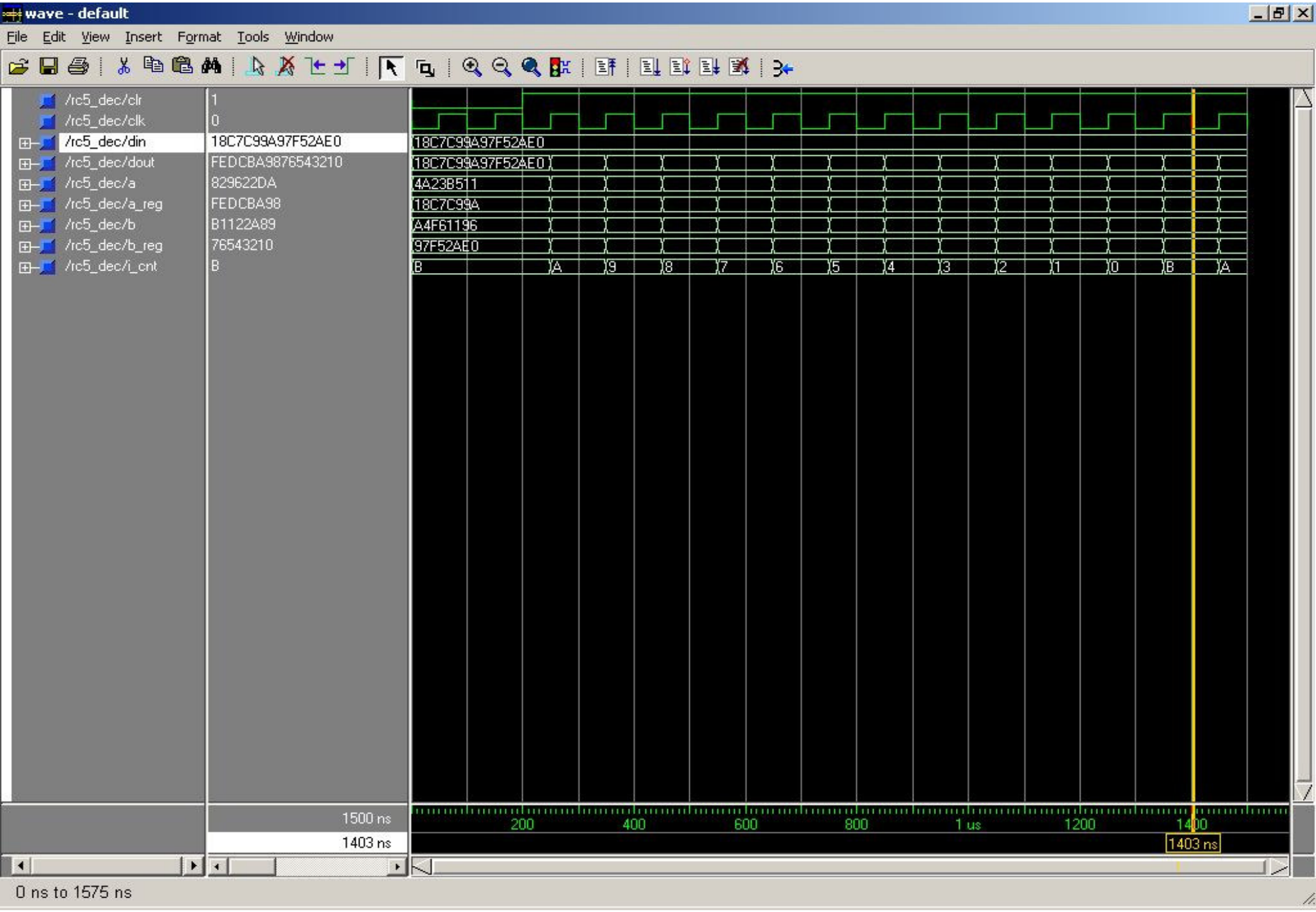
```
a_reg<=din(63 DOWNTO 32);
ELSIF(clk'EVENT AND clk='1') THEN
a_reg<=a;
END IF;
END PROCESS;
-- B register
PROCESS(clr, clk)  BEGIN
IF(clr='0') THEN
b_reg<=din(31 DOWNTO 0);
ELSIF(clk'EVENT AND clk='1') THEN
b_reg<=b;
END IF;
END PROCESS;

-- round counter
PROCESS(clr, clk)  BEGIN
IF(clr='0') THEN
i_cnt<="1011";
ELSIF(clk'EVENT AND clk='1') THEN
IF(i_cnt="0000") THEN
i_cnt<="1011";
ELSE
i_cnt<=i_cnt-'1';
END IF;
END IF;
END PROCESS;
dout<=a_reg & b_reg;
END rtl;
```

Encryption Function Simulation



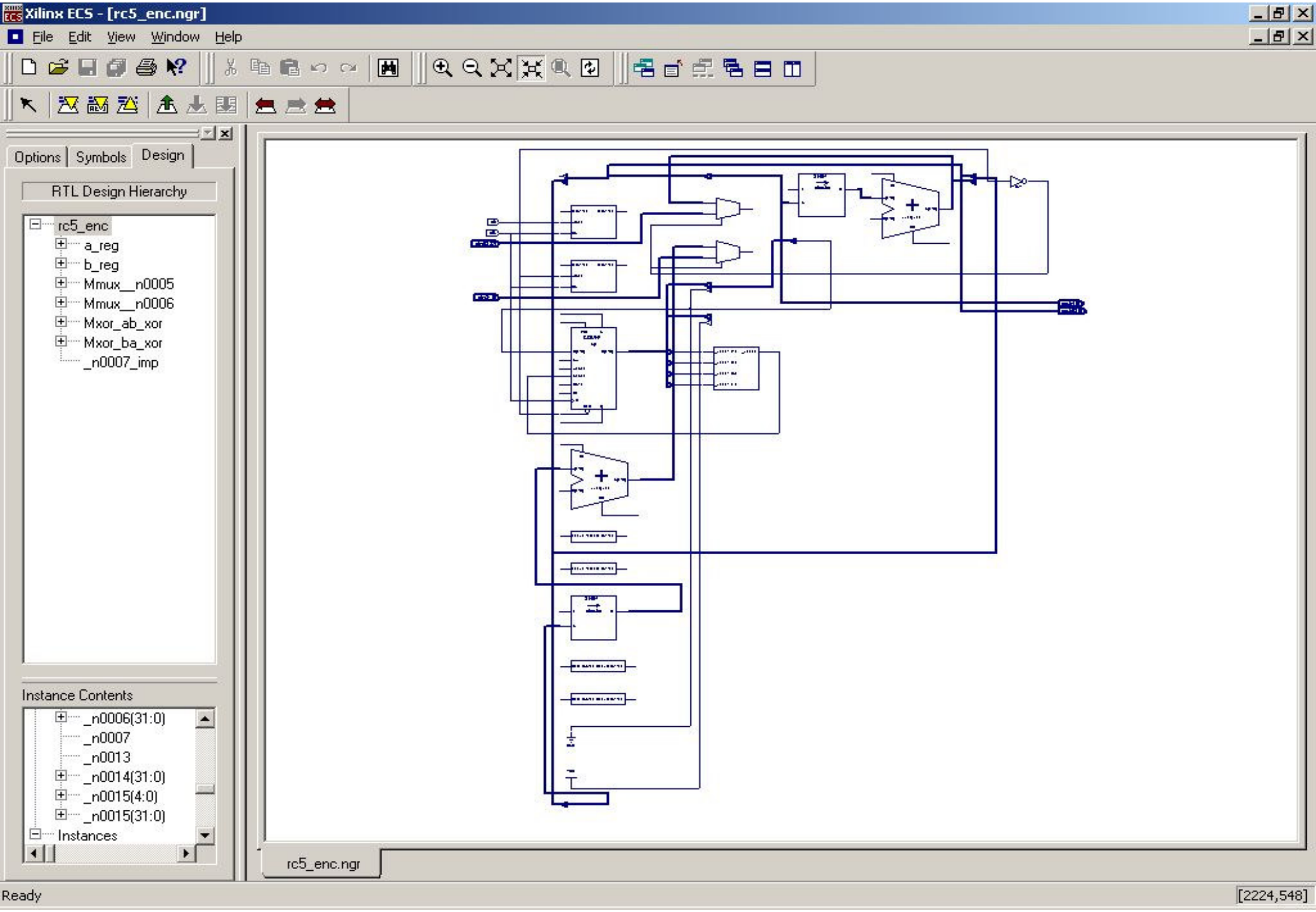
rc5_enc_funcsim.jpg

Decryption Functional Simulation



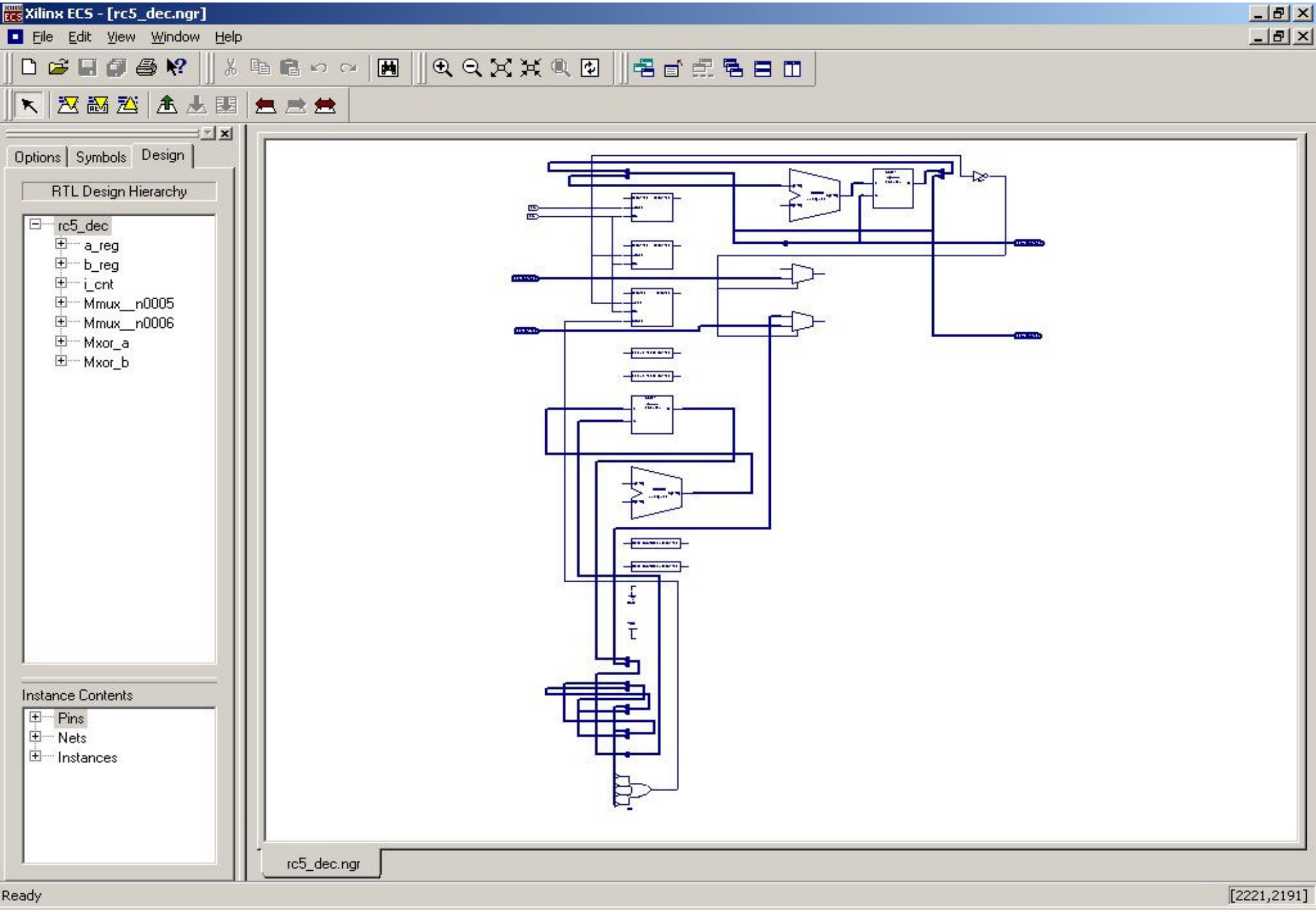
rc5_dec_funcsim.jpg

Encryption Schematic



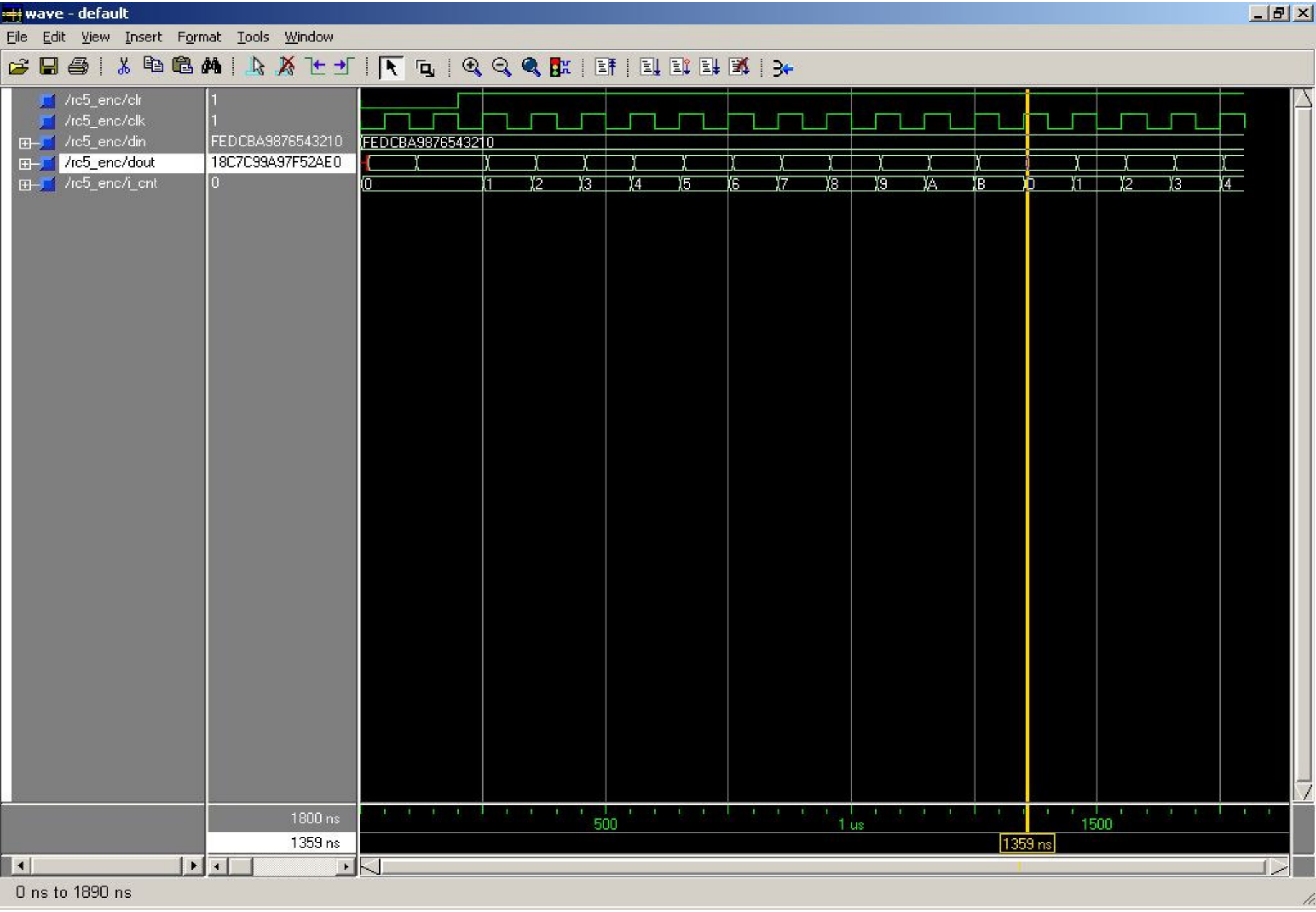
rc5_enc_synt.jpg

Decryption Schematic



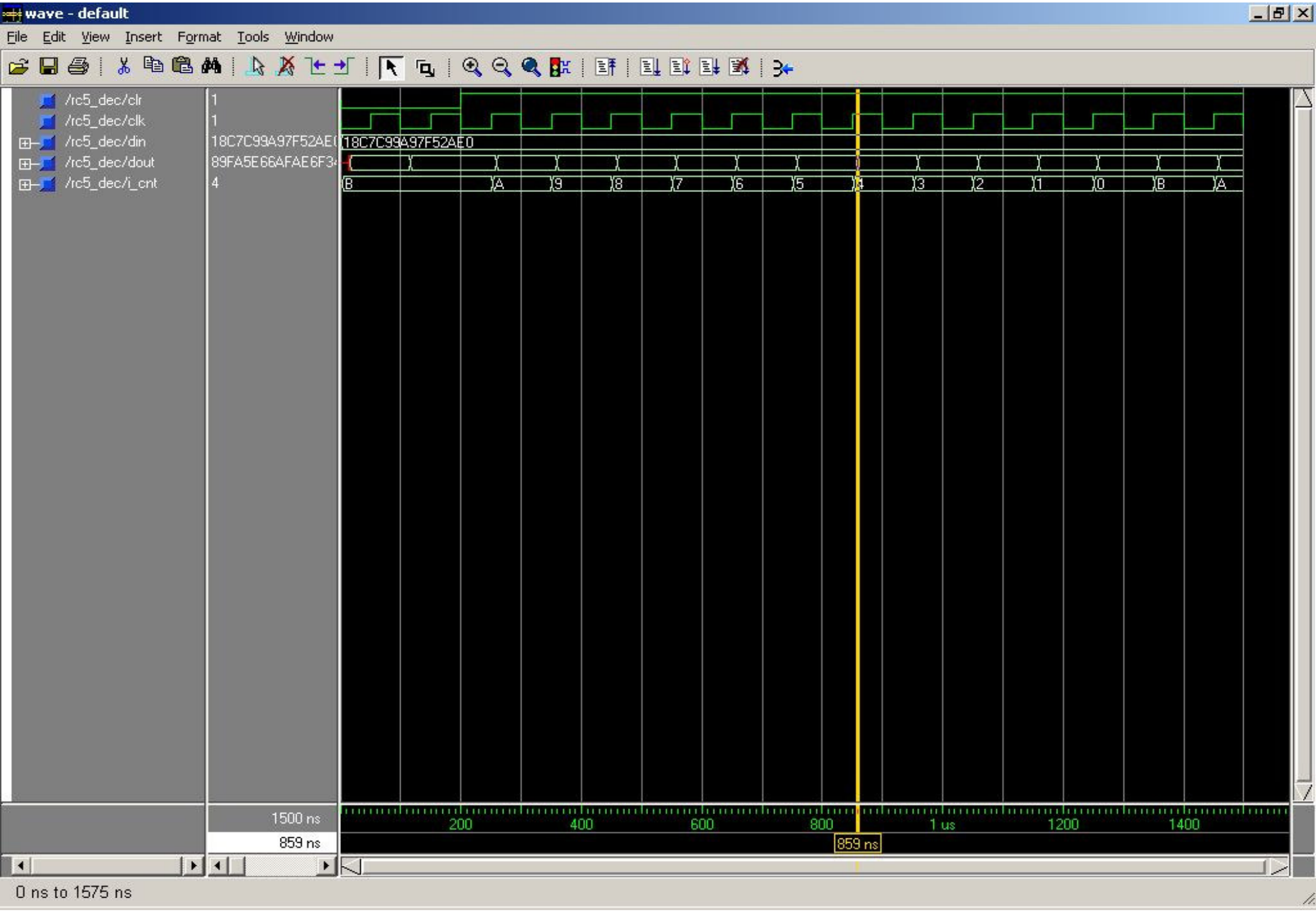
rc5_dec_synt.jpg

Encryption Timing Simulation



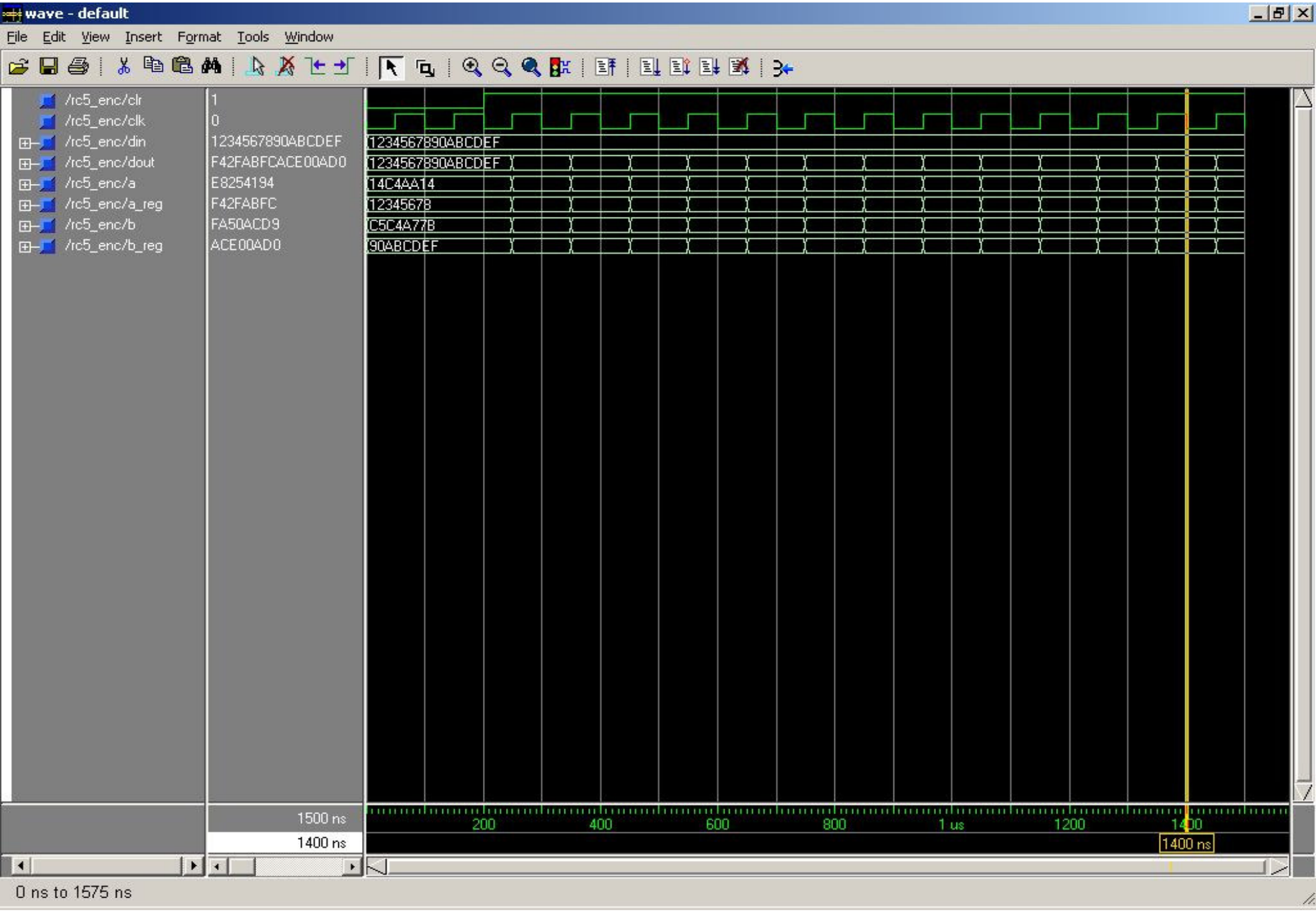
rc5_enc_timesim.jpg

Decryption Timing Simulation



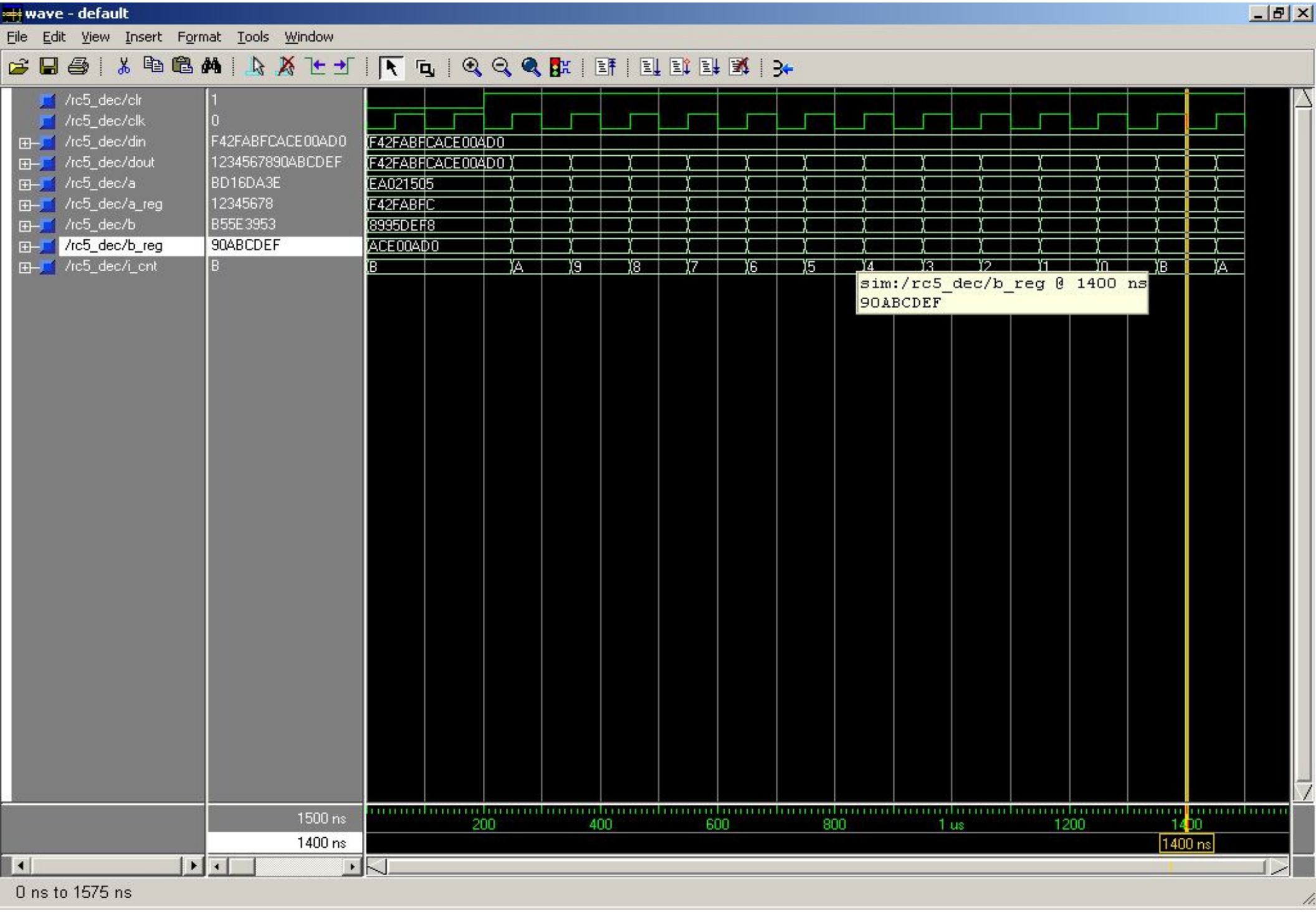
rc5_dec_timesim.jpg

Encryption 2nd Test Vector



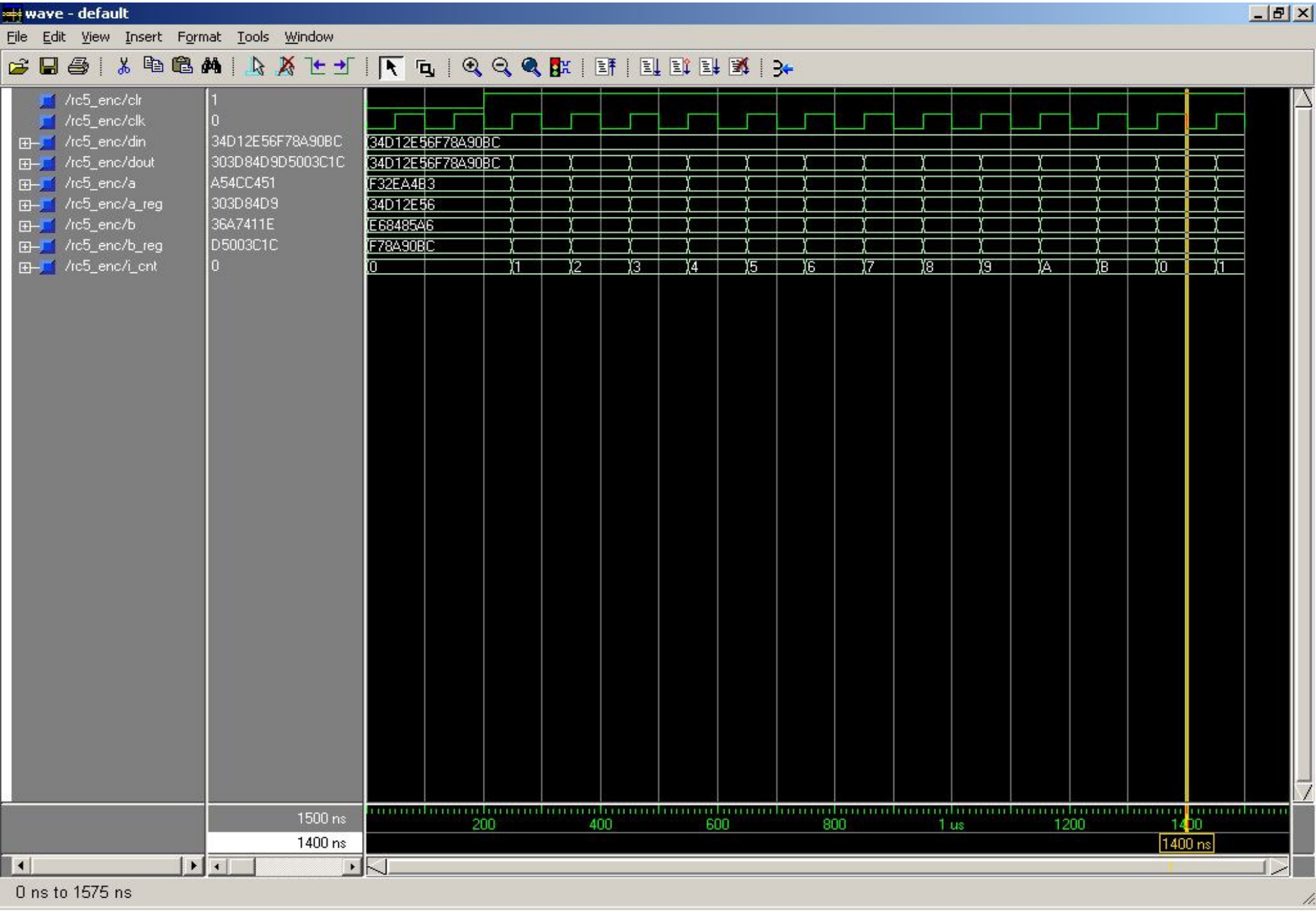
rc5_enc_test2.jpg

Decryption 2nd Test Vector



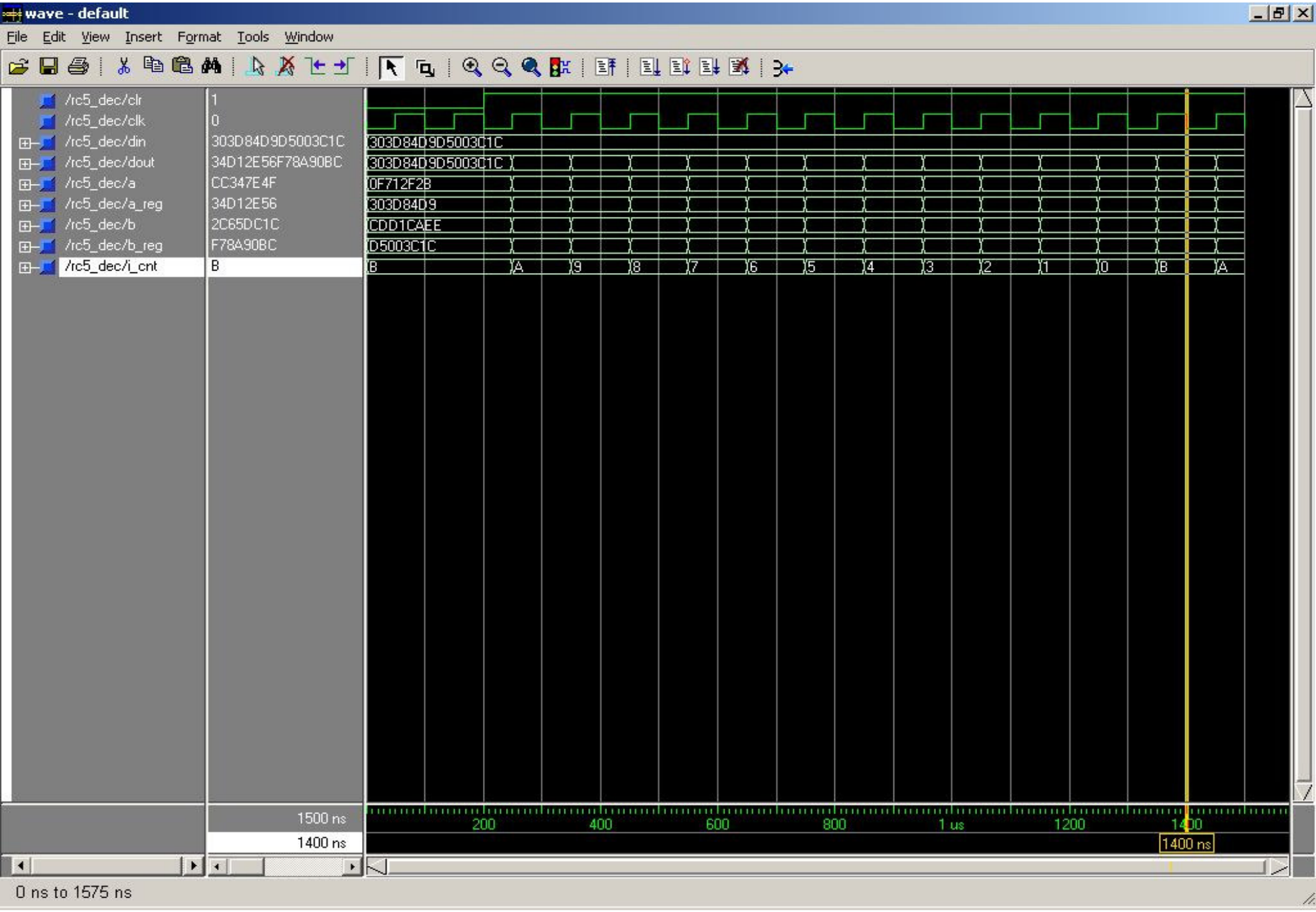
rc5_dec_test2.jpg

Encryption 3rd Test Vector



rc5_enc_test3.jpg

Decryption 3rd Test Vector



rc5_dec_test3.jpg

--> Parameter TMPDIR set to __projnav
CPU : 0.00 / 1.19 s | Elapsed : 0.00 / 1.00 s

--> Parameter xsthdpdir set to ./xst
CPU : 0.00 / 1.19 s | Elapsed : 0.00 / 1.00 s

--> Reading design: rc5_enc.prj

TABLE OF CONTENTS
  1) Synthesis Options Summary
  2) HDL Compilation
  3) HDL Analysis
  4) HDL Synthesis
     4.1) HDL Synthesis Report
  5) Low Level Synthesis
  6) Final Report
     6.1) Device utilization summary
     6.2) TIMING REPORT

Synthesis Options Summary

---- Source Parameters
Input File Name   : rc5_enc.prj
Input Format   : VHDL
Ignore Synthesis Constraint File   : NO

---- Target Parameters
Output File Name   : rc5_enc
Output Format   : NGC
Target Device   : xc3s2000-4fg676

---- Source Options
Entity Name   : rc5_enc
Automatic FSM Extraction   : YES
FSM Encoding Algorithm   : Auto
RAM Extraction   : Yes
RAM Style   : Auto
ROM Extraction   : Yes
ROM Style   : Auto
Mux Extraction   : YES
Mux Style   : Auto
Decoder Extraction   : YES
Priority Encoder Extraction   : YES
Shift Register Extraction   : YES
Logical Shifter Extraction   : YES
XOR Collapsing   : YES
Resource Sharing   : YES
Complex Clock Enable Extraction   : YES
Multiplier Style   : auto
Automatic Register Balancing   : No

---- Target Options
Add IO Buffers   : YES
Global Maximum Fanout   : 500
Add Generic Clock Buffer(BUFG)   : 16
Register Duplication   : YES
Equivalent register Removal   : YES
Slice Packing   : YES
Pack IO Registers into IOBs   : auto

```
---- General Options
Optimization Criterion             : Speed
Optimization Effort                : 1
Keep Hierarchy                     : NO
Global Optimization                : AllClockNets
RTL Output                         : Yes
Write Timing Constraints           : NO
Hierarchy Separator                : _
Bus Delimiter                      : <>
Case Specifier                     : lower
Top module area constraint         : 100
Top module allowed area overflow   : 5

---- Other Options
read_cores                         : YES
cross_clock_analysis               : NO

=========================================================================


=========================================================================
*                            HDL Compilation                            *
=========================================================================
Compiling vhdl file c:/raph/rc5_enc/../rc5_enc.vhd in Library work.
Architecture rtl of Entity rc5_enc is up to date.

=========================================================================
*                              HDL Analysis                             *
=========================================================================

Analyzing Entity <rc5_enc> (Architecture <rtl>).
WARNING:Xst:790 - c:/raph/rc5_enc/../rc5_enc.vhd line 74: Index value(s) does not
match array range, simulation mismatch.
WARNING:Xst:790 - c:/raph/rc5_enc/../rc5_enc.vhd line 112: Index value(s) does not
match array range, simulation mismatch.
WARNING:Xst:819 - c:/raph/rc5_enc/../rc5_enc.vhd line 115: The following signals are
missing in the process sensitivity list:
   din<63>, din<62>, din<61>, din<60>, din<59>, din<58>, din<57>, din<56>, din<55>,
din<54>, din<53>, din<52>, din<51>, din<50>, din<49>, din<48>, din<47>, din<46>,
din<45>, din<44>, din<43>, din<42>, din<41>, din<40>, din<39>, din<38>, din<37>,
din<36>, din<35>, din<34>, din<33>, din<32>.
WARNING:Xst:819 - c:/raph/rc5_enc/../rc5_enc.vhd line 123: The following signals are
missing in the process sensitivity list:
   din<31>, din<30>, din<29>, din<28>, din<27>, din<26>, din<25>, din<24>, din<23>,
din<22>, din<21>, din<20>, din<19>, din<18>, din<17>, din<16>, din<15>, din<14>,
din<13>, din<12>, din<11>, din<10>, din<9>, din<8>, din<7>, din<6>, din<5>, din<4>,
din<3>, din<2>, din<1>, din<0>.
Entity <rc5_enc> analyzed. Unit <rc5_enc> generated.


=========================================================================
*                             HDL Synthesis                             *
=========================================================================

Synthesizing Unit <rc5_enc>.
    Related source file is c:/raph/rc5_enc/../rc5_enc.vhd.
    Found 24x32-bit ROM for signal <$n0003> created at line 74.
    Found 24x32-bit ROM for signal <$n0004> created at line 112.
    Found 32-bit adder for signal <a>.
    Found 32-bit register for signal <a_reg>.
    Found 32-bit shifter rotate left for signal <a_rot>.
    Found 32-bit xor2 for signal <ab_xor>.
```

```
    Found 32-bit adder for signal <b>.
    Found 32-bit register for signal <b_reg>.
    Found 32-bit shifter rotate left for signal <b_rot>.
    Found 32-bit xor2 for signal <ba_xor>.
    Found 4-bit up counter for signal <i_cnt>.
    Found 64 1-bit 2-to-1 multiplexers.
    Summary:
        inferred   2 ROM(s).
        inferred   1 Counter(s).
        inferred   2 Adder/Subtracter(s).
        inferred   2 Combinational logic shifter(s).
Unit <rc5_enc> synthesized.


=========================================================================
HDL Synthesis Report

Macro Statistics
# ROMs                              : 2
  24x32-bit ROM                     : 2
# Registers                         : 2
  32-bit register                   : 2
# Counters                          : 1
  4-bit up counter                  : 1
# Multiplexers                      : 2
  2-to-1 multiplexer                : 2
# Logic shifters                    : 2
  32-bit shifter rotate left        : 2
# Adders/Subtractors                : 2
  32-bit adder                      : 2
# Xors                              : 2
  32-bit xor2                       : 2

=========================================================================

=========================================================================
*                          Low Level Synthesis                          *
=========================================================================
Library "C:/Xilinx/data/librtl.xst" Consulted

Optimizing unit <rc5_enc> ...

Mapping all equations...
Loading device for application Xst from file '3s2000.nph' in environment C:/Xilinx.
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block rc5_enc, actual ratio is 1.
FlipFlop b_reg_4_0 has been replicated 1 time(s)
FlipFlop b_reg_3_0 has been replicated 1 time(s)
FlipFlop b_reg_1_0 has been replicated 3 time(s)
FlipFlop i_cnt_2 has been replicated 2 time(s)
FlipFlop i_cnt_1 has been replicated 2 time(s)
FlipFlop i_cnt_3 has been replicated 2 time(s)
FlipFlop i_cnt_0 has been replicated 2 time(s)
FlipFlop b_reg_0_0 has been replicated 1 time(s)
FlipFlop b_reg_2_0 has been replicated 1 time(s)
FlipFlop b_reg_3_0 has been replicated 1 time(s)
FlipFlop i_cnt_0 has been replicated 1 time(s)
FlipFlop i_cnt_2 has been replicated 1 time(s)
FlipFlop i_cnt_1 has been replicated 1 time(s)
FlipFlop i_cnt_3 has been replicated 1 time(s)

=========================================================================
*                            Final Report                               *
```

```
================================================================================
Final Results
RTL Top Level Output File Name     : rc5_enc.ngr
Top Level Output File Name         : rc5_enc
Output Format                      : NGC
Optimization Criterion             : Speed
Keep Hierarchy                     : NO
Macro Generator                    : macro+

Design Statistics
# IOs                              : 130

Macro Statistics :
# ROMs                             : 2
#      24x32-bit ROM               : 2
# Counters                         : 1
#      4-bit up counter            : 1
# Logic shifters                   : 2
#      32-bit shifter rotate left  : 2
# Adders/Subtractors               : 2
#      32-bit adder                : 2

Cell Usage :
# BELS                             : 898
#      GND                         : 1
#      LUT1                        : 1
#      LUT2                        : 127
#      LUT2_D                      : 18
#      LUT2_L                      : 2
#      LUT3                        : 240
#      LUT3_D                      : 3
#      LUT3_L                      : 53
#      LUT4                        : 162
#      LUT4_D                      : 19
#      LUT4_L                      : 63
#      MUXCY                       : 66
#      MUXF5                       : 76
#      VCC                         : 1
#      XORCY                       : 66
# FlipFlops/Latches                : 88
#      FDCP                        : 72
#      FDCPE                       : 16
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 129
#      IBUF                        : 65
#      OBUF                        : 64
================================================================================

Device utilization summary:
---------------------------

Selected Device : 3s2000fg676-4

 Number of Slices:                      367  out of  20480     1%
 Number of Slice Flip Flops:             88  out of  40960     0%
 Number of 4 input LUTs:                688  out of  40960     1%
 Number of bonded IOBs:                 129  out of    489    26%
 Number of GCLKs:                         1  out of      8    12%


================================================================================
TIMING REPORT
```

```
NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 88    |
-----------------------------------+------------------------+-------+

Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 16.609ns (Maximum Frequency: 60.208MHz)
   Minimum input arrival time before clock: 4.176ns
   Maximum output required time after clock: 6.942ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
Delay:               16.609ns (Levels of Logic = 49)
  Source:            b_reg_1_0
  Destination:       b_reg_31_0
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: b_reg_1_0 to b_reg_31_0
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDCP:C->Q             18   0.494   1.066  b_reg_1_0 (b_reg_1_0)
     LUT2_D:I0->LO          1   0.382   0.100  mxor_ab_xor_result<1>1 (N13512)
     LUT4:I1->O             4   0.382   0.629  mshift_a_rot_sh21 (mshift_a_rot_sh2)
     LUT3:I1->O             1   0.382   0.000  mshift_a_rot_sh4011_G (N13113)
     MUXF5:I1->O            2   0.379   0.465  mshift_a_rot_sh4011
(mshift_a_rot_sh40)
     LUT3_D:I1->O           4   0.382   0.629  ker100761 (N10078)
     LUT4_D:I2->LO          1   0.382   0.000  mshift_b_rot_sh01_SW0 (N13410)
     MUXCY:S->O             1   0.295   0.000  madd_a_inst_cy_5 (madd_a_inst_cy_5)
     MUXCY:CI->O            1   0.046   0.000  madd_a_inst_cy_6 (madd_a_inst_cy_6)
     MUXCY:CI->O            1   0.046   0.000  madd_a_inst_cy_7 (madd_a_inst_cy_7)
     XORCY:CI->O           35   1.107   1.324  madd_a_inst_sum_7 (_n0015<3>)
     LUT2_D:I0->LO          1   0.382   0.100  mxor_ba_xor_result<3>1 (N13514)
     LUT4:I3->O             3   0.382   0.577  mshift_b_rot_sh39 (mshift_b_rot_sh3)
     LUT3:I1->O             1   0.382   0.000  mshift_b_rot_sh4111_G (N13238)
     MUXF5:I1->O            4   0.379   0.629  mshift_b_rot_sh4111
(mshift_b_rot_sh41)
     LUT4:I1->O             1   0.382   0.000  ker992611_F (N13086)
     MUXF5:I0->O            2   0.379   0.465  ker992611 (N9928)
     LUT4_L:I2->LO          1   0.382   0.000  madd_b_inst_lut2_110
(madd_b_inst_lut2_1)
     MUXCY:S->O             1   0.295   0.000  madd_b_inst_cy_6 (madd_b_inst_cy_6)
     MUXCY:CI->O            1   0.046   0.000  madd_b_inst_cy_7 (madd_b_inst_cy_7)
     MUXCY:CI->O            1   0.046   0.000  madd_b_inst_cy_8 (madd_b_inst_cy_8)
     MUXCY:CI->O            1   0.046   0.000  madd_b_inst_cy_9 (madd_b_inst_cy_9)
```

```
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_10 (madd_b_inst_cy_10)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_11 (madd_b_inst_cy_11)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_12 (madd_b_inst_cy_12)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_13 (madd_b_inst_cy_13)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_14 (madd_b_inst_cy_14)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_15 (madd_b_inst_cy_15)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_16 (madd_b_inst_cy_16)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_17 (madd_b_inst_cy_17)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_18 (madd_b_inst_cy_18)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_19 (madd_b_inst_cy_19)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_20 (madd_b_inst_cy_20)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_21 (madd_b_inst_cy_21)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_22 (madd_b_inst_cy_22)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_23 (madd_b_inst_cy_23)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_24 (madd_b_inst_cy_24)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_25 (madd_b_inst_cy_25)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_26 (madd_b_inst_cy_26)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_27 (madd_b_inst_cy_27)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_28 (madd_b_inst_cy_28)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_29 (madd_b_inst_cy_29)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_30 (madd_b_inst_cy_30)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_31 (madd_b_inst_cy_31)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_32 (madd_b_inst_cy_32)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_33 (madd_b_inst_cy_33)
    MUXCY:ci->o           1   0.046   0.000  madd_b_inst_cy_34 (madd_b_inst_cy_34)
    MUXCY:ci->o           0   0.046   0.000  madd_b_inst_cy_35 (madd_b_inst_cy_35)
    XORCY:ci->o           1   1.107   0.240  madd_b_inst_sum_35 (_n0014<31>)
    LUT3_L:I1->LO         1   0.382   0.000  mmux__n0006_i0_result1 (_n0006<31>)
    FDCP:d                    0.322          b_reg_31_0
    ----------------------------------------
    Total                    16.609ns (10.385ns logic, 6.224ns route)
                                      (62.5% logic, 37.5% route)

--------------------------------------------------------------------------------
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
Offset:              4.176ns (Levels of Logic = 3)
  Source:            clr
  Destination:       a_reg_14_0
  Destination Clock: clk rising

  Data Path: clr to a_reg_14_0
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:i->o            201   0.942   1.409  clr_ibuf (clr_ibuf)
     LUT3:i0->o             3   0.382   0.577  mmux__n0005_i17_result1 (_n0005<14>)
     LUT2:i1->o             1   0.382   0.240  a_reg_14__n00001 (a_reg_14__n0000)
     FDCP:clr                   0.244          a_reg_14_0
    ----------------------------------------
    Total                      4.176ns (1.950ns logic, 2.226ns route)
                                      (46.7% logic, 53.3% route)

--------------------------------------------------------------------------------
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
Offset:              6.942ns (Levels of Logic = 1)
  Source:            b_reg_2_0_1
  Destination:       dout<2>
  Source Clock:      clk rising

  Data Path: b_reg_2_0_1 to dout<2>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
```

```
     FDCP:C->Q             20   0.494   1.137  b_reg_2_0_1 (b_reg_2_0_1)
     OBUF:I->O                  5.311          dout_2_obuf (dout<2>)
    ---------------------------------------
    Total                       6.942ns (5.805ns logic, 1.137ns route)
                                        (83.6% logic, 16.4% route)

==========================================================================
CPU : 57.10 / 58.70 s | Elapsed : 57.00 / 58.00 s

-->

Total memory usage is 101088 kilobytes
```

--> Parameter TMPDIR set to __projnav
CPU : 0.00 / 1.02 s | Elapsed : 0.00 / 1.00 s

--> Parameter xsthdpdir set to ./xst
CPU : 0.00 / 1.02 s | Elapsed : 0.00 / 1.00 s

--> Reading design: rc5_dec.prj

TABLE OF CONTENTS
  1) Synthesis Options Summary
  2) HDL Compilation
  3) HDL Analysis
  4) HDL Synthesis
     4.1) HDL Synthesis Report
  5) Low Level Synthesis
  6) Final Report
     6.1) Device utilization summary
     6.2) TIMING REPORT

Synthesis Options Summary

---- Source Parameters
Input File Name   : rc5_dec.prj
Input Format   : VHDL
Ignore Synthesis Constraint File   : NO

---- Target Parameters
Output File Name   : rc5_dec
Output Format   : NGC
Target Device   : xc3s2000-4fg676

---- Source Options
Entity Name   : rc5_dec
Automatic FSM Extraction   : YES
FSM Encoding Algorithm   : Auto
RAM Extraction   : Yes
RAM Style   : Auto
ROM Extraction   : Yes
ROM Style   : Auto
Mux Extraction   : YES
Mux Style   : Auto
Decoder Extraction   : YES
Priority Encoder Extraction   : YES
Shift Register Extraction   : YES
Logical Shifter Extraction   : YES
XOR Collapsing   : YES
Resource Sharing   : YES
Complex Clock Enable Extraction   : YES
Multiplier Style   : auto
Automatic Register Balancing   : No

---- Target Options
Add IO Buffers   : YES
Global Maximum Fanout   : 500
Add Generic Clock Buffer(BUFG)   : 16
Register Duplication   : YES
Equivalent register Removal   : YES
Slice Packing   : YES
Pack IO Registers into IOBs   : auto

```
---- General Options
Optimization Criterion             : Speed
Optimization Effort                : 1
Keep Hierarchy                     : NO
Global Optimization                : AllClockNets
RTL Output                         : Yes
Write Timing Constraints           : NO
Hierarchy Separator                : _
Bus Delimiter                      : <>
Case Specifier                     : lower
Top module area constraint         : 100
Top module allowed area overflow   : 5

---- Other Options
read_cores                         : YES
cross_clock_analysis               : NO

=========================================================================


=========================================================================
*                            HDL Compilation                            *
=========================================================================
Compiling vhdl file C:/raph/rc5_dec/../decr/rc5_dec.vhd in Library work.
Entity <rc5_dec> (Architecture <rtl>) compiled.

=========================================================================
*                             HDL Analysis                              *
=========================================================================

Analyzing Entity <rc5_dec> (Architecture <rtl>).
WARNING:Xst:790 - C:/raph/rc5_dec/../decr/rc5_dec.vhd line 46: Index value(s) does
not match array range, simulation mismatch.
WARNING:Xst:790 - C:/raph/rc5_dec/../decr/rc5_dec.vhd line 85: Index value(s) does
not match array range, simulation mismatch.
WARNING:Xst:819 - C:/raph/rc5_dec/../decr/rc5_dec.vhd line 126: The following
signals are missing in the process sensitivity list:
   din<63>, din<62>, din<61>, din<60>, din<59>, din<58>, din<57>, din<56>, din<55>,
din<54>, din<53>, din<52>, din<51>, din<50>, din<49>, din<48>, din<47>, din<46>,
din<45>, din<44>, din<43>, din<42>, din<41>, din<40>, din<39>, din<38>, din<37>,
din<36>, din<35>, din<34>, din<33>, din<32>.
WARNING:Xst:819 - C:/raph/rc5_dec/../decr/rc5_dec.vhd line 134: The following
signals are missing in the process sensitivity list:
   din<31>, din<30>, din<29>, din<28>, din<27>, din<26>, din<25>, din<24>, din<23>,
din<22>, din<21>, din<20>, din<19>, din<18>, din<17>, din<16>, din<15>, din<14>,
din<13>, din<12>, din<11>, din<10>, din<9>, din<8>, din<7>, din<6>, din<5>, din<4>,
din<3>, din<2>, din<1>, din<0>.
Entity <rc5_dec> analyzed. Unit <rc5_dec> generated.


=========================================================================
*                             HDL Synthesis                             *
=========================================================================

Synthesizing Unit <rc5_dec>.
    Related source file is C:/raph/rc5_dec/../decr/rc5_dec.vhd.
    Found 24x32-bit ROM for signal <$n0003> created at line 46.
    Found 24x32-bit ROM for signal <$n0004> created at line 85.
    Found 32-bit xor2 for signal <a>.
    Found 32-bit register for signal <a_reg>.
    Found 32-bit shifter rotate right for signal <a_rot>.
    Found 32-bit subtractor for signal <askey_minus>.
```

```
    Found 32-bit xor2 for signal <b>.
    Found 32-bit register for signal <b_reg>.
    Found 32-bit shifter rotate right for signal <b_rot>.
    Found 32-bit subtractor for signal <bskey_minus>.
    Found 4-bit down counter for signal <i_cnt>.
    Found 64 1-bit 2-to-1 multiplexers.
    Summary:
        inferred   2 ROM(s).
        inferred   1 Counter(s).
        inferred   2 Adder/Subtracter(s).
        inferred   2 Combinational logic shifter(s).
Unit <rc5_dec> synthesized.


=========================================================================
HDL Synthesis Report

Macro Statistics
# ROMs                              : 2
  24x32-bit ROM                     : 2
# Registers                         : 2
  32-bit register                   : 2
# Counters                          : 1
  4-bit down counter                : 1
# Multiplexers                      : 2
  2-to-1 multiplexer                : 2
# Logic shifters                    : 2
  32-bit shifter rotate right       : 2
# Adders/Subtractors                : 2
  32-bit subtractor                 : 2
# Xors                              : 2
  32-bit xor2                       : 2

=========================================================================

=========================================================================
*                          Low Level Synthesis                          *
=========================================================================
Library "C:/Xilinx/data/librtl.xst" Consulted

Optimizing unit <rc5_dec> ...

Mapping all equations...
Loading device for application Xst from file '3s2000.nph' in environment C:/Xilinx.
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block rc5_dec, actual ratio is 1.
FlipFlop i_cnt_1 has been replicated 2 time(s)
FlipFlop i_cnt_3 has been replicated 2 time(s)
FlipFlop i_cnt_2 has been replicated 2 time(s)
FlipFlop i_cnt_0 has been replicated 2 time(s)
FlipFlop a_reg_1_0 has been replicated 1 time(s)

=========================================================================
*                            Final Report                               *
=========================================================================
Final Results
RTL Top Level Output File Name     : rc5_dec.ngr
Top Level Output File Name         : rc5_dec
Output Format                      : NGC
Optimization Criterion             : Speed
Keep Hierarchy                     : NO
Macro Generator                    : macro+
```

```
Design Statistics
# IOs                              : 130

Macro Statistics :
# ROMs                             : 2
#      24x32-bit ROM               : 2
# Registers                        : 12
#      1-bit register              : 12
# Logic shifters                   : 2
#      32-bit shifter rotate right : 2
# Adders/Subtractors               : 3
#      32-bit subtractor           : 2
#      4-bit subtractor            : 1

Cell Usage :
# BELS                             : 901
#      LUT1                        : 5
#      LUT2                        : 156
#      LUT2_L                      : 38
#      LUT3                        : 255
#      LUT3_D                      : 26
#      LUT3_L                      : 30
#      LUT4                        : 99
#      LUT4_D                      : 29
#      LUT4_L                      : 33
#      MUXCY                       : 64
#      MUXF5                       : 97
#      VCC                         : 1
#      XORCY                       : 68
# FlipFlops/Latches                : 77
#      FDC                         : 3
#      FDCP                        : 65
#      FDP                         : 9
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 129
#      IBUF                        : 65
#      OBUF                        : 64
=========================================================================

Device utilization summary:
---------------------------

Selected Device : 3s2000fg676-4

 Number of Slices:                      356  out of  20480     1%
 Number of Slice Flip Flops:             77  out of  40960     0%
 Number of 4 input LUTs:                671  out of  40960     1%
 Number of bonded IOBs:                 129  out of    489    26%
 Number of GCLKs:                         1  out of      8    12%


=========================================================================
TIMING REPORT

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
```

clk	BUFGP	77

Timing Summary:

Speed Grade: -4

Minimum period: 14.451ns (Maximum Frequency: 69.199MHz)
Minimum input arrival time before clock: 3.842ns
Maximum output required time after clock: 7.149ns
Maximum combinational path delay: No path found

Timing Detail:

All values displayed in nanoseconds (ns)

Timing constraint: Default period analysis for Clock 'clk'
Delay: 14.451ns (Levels of Logic = 30)
Source: i_cnt_0
Destination: a_reg_1_0
Source Clock: clk rising
Destination Clock: clk rising

Data Path: i_cnt_0 to a_reg_1_0

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
FDP:c->q	16	0.494	0.995	i_cnt_0 (i_cnt_0)
LUT4:i0->o	1	0.382	0.240	mrom__n0003_inst_mux_f5_321 (_n0003<0>)
LUT2_L:I1->LO	1	0.382	0.000	msub_bskey_minus_inst_lut2_01 (msub_bskey_minus_inst_lut2_0)
MUXCY:s->o	1	0.295	0.000	msub_bskey_minus_inst_cy_0 (msub_bskey_minus_inst_cy_0)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_1 (msub_bskey_minus_inst_cy_1)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_2 (msub_bskey_minus_inst_cy_2)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_3 (msub_bskey_minus_inst_cy_3)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_4 (msub_bskey_minus_inst_cy_4)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_5 (msub_bskey_minus_inst_cy_5)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_6 (msub_bskey_minus_inst_cy_6)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_7 (msub_bskey_minus_inst_cy_7)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_8 (msub_bskey_minus_inst_cy_8)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_9 (msub_bskey_minus_inst_cy_9)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_10 (msub_bskey_minus_inst_cy_10)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_11 (msub_bskey_minus_inst_cy_11)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_12 (msub_bskey_minus_inst_cy_12)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_13 (msub_bskey_minus_inst_cy_13)
MUXCY:ci->o	1	0.046	0.000	msub_bskey_minus_inst_cy_14

```
(msub_bskey_minus_inst_cy_14)
     XORCY:ci->o           2   1.107   0.465  msub_bskey_minus_inst_sum_15
(bskey_minus<15>)
     LUT3:i2->o            4   0.382   0.629  mshift_b_rot_sh141 (mshift_b_rot_sh14)
     LUT3:i2->o            1   0.382   0.000  mshift_b_rot_sh4011_g (n13796)
     MUXF5:i1->o           4   0.379   0.629  mshift_b_rot_sh4011
(mshift_b_rot_sh40)
     LUT3:i1->o            1   0.382   0.000  ker1124811_f (n14064)
     MUXF5:i0->o           3   0.379   0.577  ker1124811 (n11250)
     LUT4_D:I3->O         16   0.382   0.995  mxor_b_result<0>1_1
(mxor_b_result<0>1_1)
     LUT3:i0->o            4   0.382   0.629  mshift_a_rot_sh510 (mshift_a_rot_sh5)
     LUT3:i2->o            1   0.382   0.000  mshift_a_rot_sh3311_f (n13754)
     MUXF5:i0->o           4   0.379   0.629  mshift_a_rot_sh3311
(mshift_a_rot_sh33)
     LUT3:i1->o            1   0.382   0.000  mshift_a_rot_result<1>11_f (n13994)
     MUXF5:i0->o           2   0.379   0.465  mshift_a_rot_result<1>11 (_n0018<33>)
     LUT4_L:I3->LO         1   0.382   0.000  mmux__n0005_i30_result1 (_n0005<1>)
     FDCP:d                    0.322          a_reg_1_0
    ----------------------------------------
    Total                     14.451ns (8.198ns logic, 6.253ns route)
                                       (56.7% logic, 43.3% route)

--------------------------------------------------------------------------------
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
Offset:              3.842ns (Levels of Logic = 2)
  Source:            clr
  Destination:       i_cnt_1
  Destination Clock: clk rising

  Data Path: clr to i_cnt_1
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:i->o           194   0.942   1.409  clr_ibuf (clr_ibuf)
     LUT1:i0->o           12   0.382   0.865  i_cnt_3_aset_inv1 (i_cnt_0_3_n815)
     FDP:pre                   0.244          i_cnt_1
    ----------------------------------------
    Total                      3.842ns (1.568ns logic, 2.274ns route)
                                       (40.8% logic, 59.2% route)

--------------------------------------------------------------------------------
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
Offset:              7.149ns (Levels of Logic = 1)
  Source:            a_reg_2_0
  Destination:       dout<34>
  Source Clock:      clk rising

  Data Path: a_reg_2_0 to dout<34>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDCP:c->q             55   0.494   1.344  a_reg_2_0 (a_reg_2_0)
     OBUF:i->o                 5.311          dout_34_obuf (dout<34>)
    ----------------------------------------
    Total                      7.149ns (5.805ns logic, 1.344ns route)
                                       (81.2% logic, 18.8% route)

==============================================================================
CPU : 44.08 / 45.49 s | Elapsed : 44.00 / 45.00 s

-->
```

Total memory usage is 101088 kilobytes