Cryptography Engineering

“Blowfish Algorithm“

Bayan AlBajali

Introduction:

Symmetric algorithms, sometimes called conventional algorithms, are algorithms where the encryption key can be calculated from the decryption key and vice versa. In most symmetric algorithms, the encryption key and the decryption key are the same. These algorithms, also called secret-key algorithms, single-key algorithms, or one-key algorithms, require that the sender and receiver agree on a key before they can communicate securely. The security of a symmetric algorithm rests in the key; divulging the key means that anyone could encrypt and decrypt messages. As long as the communication needs to remain secret, the key must remain secret. Blowfish is a symmetric cryptographic algorithm.

Blowfish Algorithm:

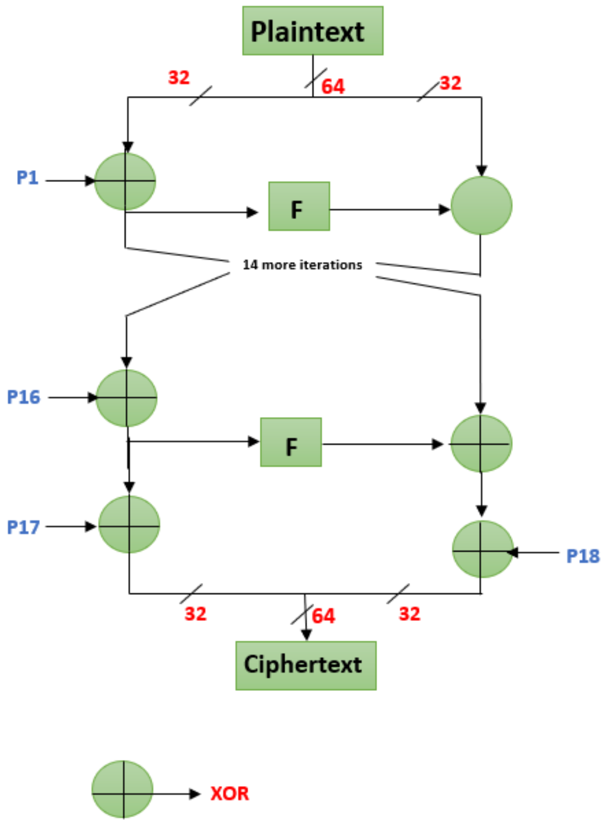
Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt messages. Blowfish is also a block cipher, meaning that it divides a message up into fixed length blocks during encryption and decryption. The block length for Blowfish is 64 bits; messages that aren't a multiple of eight bytes in size must be padded.

Advantages and Disadvantages of Blowfish Algorithm:

- Blowfish is a fast block cipher except when changing keys. Each new key requires a pre-processing equivalent to 4KB of text.
- It is faster and much better than DES Encryption.
- Blowfish uses a 64-bit block size which makes it vulnerable to birthday attacks.
- A reduced round variant of blowfish is known to be suceptible to known plain text attacks(2nd order differential attacks – 4 rounds).

Applications of Blowfish Algorithm:

- Bulk Encryption.
- Packet Encryption(ATM Packets)
- Password Hashing



The entire encryption process can be elaborated as:

Blowfish consists of sixteen rounds. For each round, first XOR the left half of the block with the subkey for that round. Then apply the f-function to the left half of the block, and XOR the right half of the block with the result. Finally, after all but the last round, swap the halves of the block. There is only one subkey for each round; the f-function consumes no subkeys, but uses S-boxes which are key dependent.
After the last round, XOR the *right* half with subkey 17, and the *left* half with subkey 18.

The f-function:

Blowfish uses four S-boxes. Each one has 256 entries, and each of the entries are 32 bits long. To calculate the f-function: use the first byte of the 32 bits of input to find an entry in the first S-box, the second byte to find an entry in the second S-box, and so on. The value of the f-function is ((S1(B1) + S2(B2)) XOR S3(B3)) + S4(B4) where addition is performed modulo 2^32.

Subkey generation:

Blowfish uses a large number of subkeys. These keys must be pre computed before any data encryption or decryption.

The P-array consists of 18 32-bit subkeys:
P1, P2...., P18.
2. There are four 32-bit S-boxes with 256 entries each:
S1,0 , S1,1 , ., S1,255;
S2,0, S2,1,..,, S2,255;
S3,0, S3,1,..., S3,255;
S4,0, S4,1,..,, S4,255;
The exact method used to calculate these subkeys will be described later

The Blowfish Encryption Algorithm:

is a Feistel network consisting of 16 rounds (Figure.1). The input is a 64-bit data element, X. Divide x into two 32-bit halves: XL, XR

: For i = 1to 16

xL = XL XOR Pi

xR = F(XL) XOR xR

Swap XL and xR

Swap XL and xR (Undo the last swap.)

xR = xR XOR P17

xL = xL XOR P18

Recombine xL and xR

The Blowfish Encryption pseudocode:

begin itemise
The input is a 64-bit data element, x
Divide x into two 32-bit halves: xL, xR
Then, for i = 1 to 16:

xL = xL XOR Pi

xR = F(xL) XOR xR

Swap xL and xR

After the sixteenth round, swap xL and xR again to undo the last swap.

Then, xR = xR XOR P17 and xL = xL XOR P18

Finally, recombine xL and xR to get the ciphertext

Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order. Implementations of Blowfish that require the fastest speeds should unroll the loop and ensure that all subkeys are stored in cache.

Implementation of Blowfish Encryption:

The code were written in Verilog and simulated with the Verilog simulator. The Verilog simulator ISIM is used here for the simulation. The key generation involving various steps and that can be used to generate the actual key for encryption. The decryption is done exactly same as the encryption except that the keys are used in the reverse order

blowfish_encryption:

C:/Users/user/Desktop/myVerilog/Blow-fish-master/hamdia blowfish/hamdiya_blowfish_encryption.v (/hamdiya_testbench_blowfish/hamdiya_enc) - Default

```
`timescale 1ns / 1ps
module hamdiya_blowfish_encryption(plainText,key,cipherText);
input [63:0]plainText;
input [448:0]key;
output [63:0]cipherText;
wire [31:0]ptr,ptl;
wire [31:0]k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13,k14;
wire [31:0]p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18;
wire [31:0]par1,par2,par3,par4,par5,par6,par7,par8,par9,par10,par11,par12,par13,par14,par15,par16,par17,par18;
wire [31:0]l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16;
wire [31:0]r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14,r15,r16;
wire [31:0]f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,f16,ctl,ctr;

assign ptr=plainText[31:0];
assign ptl=plainText[63:32];

assign k1=key[447:416];
assign k2=key[415:384];
assign k3=key[383:352];
assign k4=key[351:320];
assign k5=key[319:288];
assign k6=key[287:256];
assign k7=key[255:224];
assign k8=key[223:192];
assign k9=key[191:160];
assign k10=key[159:128];
assign k11=key[127:96];
assign k12=key[95:64];
assign k13=key[63:32];
assign k14=key[31:0];
assign p1=32'h243f6a88;
assign p2=32'h85a308d3;
assign p3=32'h13198a2e;
assign p4=32'h03707344;
assign p5=32'ha4093822;
assign p6=32'h299f31d0;
assign p7=32'h082efa98;
assign p8=32'hec4e6c89;
assign p9=32'h452821e6;
assign p10=32'h38d01377;
assign p11=32'hbe5466cf;
assign p12=32'h34e90c6c;
assign p13=32'hc0ac29b7;
assign p14=32'hc97c50dd;
assign p15=32'h3f84d5b5;
```

Transcript | hamdiya_blowfish_decryption.v | hamdiya_blowfish_encryption.v | hamdiya_feistel_function.v | sBox2.v | hamdiya_testbench_blowfish.v | sBox3.v | sBox4.v | sBox1.v | Library | Project

C:/Users/user/Desktop/myVerilog/Blow-fish-master/hamdia blowfish/hamdiya_blowfish_encryption.v (/hamdiya_testbench_blowfish/hamdiya_enc) - Default

```
assign k14=key[31:0];
assign p1=32'h243f6a88;
assign p2=32'h85a308d3;
assign p3=32'h13198a2e;
assign p4=32'h03707344;
assign p5=32'ha4093822;
assign p6=32'h299f31d0;
assign p7=32'h082efa98;
assign p8=32'hec4e6c89;
assign p9=32'h452821e6;
assign p10=32'h38d01377;
assign p11=32'hbe5466cf;
assign p12=32'h34e90c6c;
assign p13=32'hc0ac29b7;
assign p14=32'hc97c50dd;
assign p15=32'h3f84d5b5;
assign p16=32'hb5470917;
assign p17=32'h9216d5d9;
assign p18=32'h8979fb1b;

assign par1=p1^k1;
assign par2=p2^k2;
assign par3=p3^k3;
assign par4=p4^k4;
assign par5=p5^k5;
assign par6=p6^k6;
assign par7=p7^k7;
assign par8=p8^k8;
assign par9=p9^k9;
assign par10=p10^k10;
assign par11=p11^k11;
assign par12=p12^k12;
assign par13=p13^k13;
assign par14=p14^k14;
assign par15=p15^k1;
assign par16=p16^k2;
assign par17=p17^k3;
assign par18=p18^k4;

assign l1=par1^ptl;
hamdiya_feistel_function a1(l1,f1);
assign r1=ptr^f1;

assign l2=par2^r1;
hamdiya_feistel_function a2(l2,f2);
```

Transcript | hamdiya_blowfish_decryption.v | hamdiya_blowfish_encryption.v | hamdiya_feistel_function.v | sBox2.v | hamdiya_testbench_blowfish.v | sBox3.v | sBox4.v | sBox1.v | Library | Project

C:/Users/user/Desktop/myVerilog/Blow-fish-master/hamdia blowfish/hamdiya_blowfish_encryption.v (/hamdiya_testbench_blowfish/hamdiya_enc) - Default

```
assign l1=par1^ptl;
hamdiya_feistel_function a1(l1,f1);
assign r1=ptr^f1;

assign l2=par2^r1;
hamdiya_feistel_function a2(l2,f2);
assign r2=l1^f2;

assign l3=par3^r2;
hamdiya_feistel_function a3(l3,f3);
assign r3=l2^f3;

assign l4=par4^r3;
hamdiya_feistel_function a4(l4,f4);
assign r4=l3^f4;

assign l5=par5^r4;
hamdiya_feistel_function a5(l5,f5);
assign r5=l4^f5;

assign l6=par6^r5;
hamdiya_feistel_function a6(l6,f6);
assign r6=l5^f6;

assign l7=par7^r6;
hamdiya_feistel_function a7(l7,f7);
assign r7=l6^f7;

assign l8=par8^r7;
hamdiya_feistel_function a8(l8,f8);
assign r8=l7^f8;

assign l9=par9^r8;
hamdiya_feistel_function a9(l9,f9);
assign r9=l8^f9;

assign l10=par10^r9;
hamdiya_feistel_function a10(l10,f10);
assign r10=l9^f10;

assign l11=par11^r10;
hamdiya_feistel_function a11(l11,f11);
assign r11=l10^f11;
```

Transcript | hamdiya_blowfish_decryption.v | hamdiya_blowfish_encryption.v | hamdiya_feistel_function.v | sBox2.v | hamdiya_testbench_blowfish.v | sBox3.v | sBox4.v | sBox1.v | Library | Project

C:/Users/user/Desktop/myVerilog/Blow-fish-master/hamdia blowfish/hamdiya_blowfish_encryption.v (/hamdiya_testbench_blowfish/hamdiya_enc) - Default

```
assign l8=par8^r7;
hamdiya_feistel_function a8(l8,f8);
assign r8=l7^f8;

assign l9=par9^r8;
hamdiya_feistel_function a9(l9,f9);
assign r9=l8^f9;

assign l10=par10^r9;
hamdiya_feistel_function a10(l10,f10);
assign r10=l9^f10;

assign l11=par11^r10;
hamdiya_feistel_function a11(l11,f11);
assign r11=l10^f11;

assign l12=par12^r11;
hamdiya_feistel_function a12(l12,f12);
assign r12=l11^f12;

assign l13=par13^r12;
hamdiya_feistel_function a13(l13,f13);
assign r13=l12^f13;

assign l14=par14^r13;
hamdiya_feistel_function a14(l14,f14);
assign r14=l13^f14;

assign l15=par15^r14;
hamdiya_feistel_function a15(l15,f15);
assign r15=l14^f15;

assign l16=par16^r15;
hamdiya_feistel_function a16(l16,f16);
assign r16=l15^f16;

assign ctl=par18^l16;
assign ctr=par17^r16;

assign cipherText={ctl,ctr};

endmodule
```

Transcript | hamdiya_blowfish_decryption.v | hamdiya_blowfish_encryption.v | hamdiya_feistel_function.v | sBox2.v | hamdiya_testbench_blowfish.v | sBox3.v | sBox4.v | sBox1.v | Library | Project

blowfish_ Decryption:

C:/Users/user/Desktop/myVerilog/Blow-fish-master/hamdia blowfish/hamdiya_blowfish_decryption.v (/hamdiya_testbench_blowfish/hamdiya_dec) - Default

```
`timescale 1ns / 1ps
module hamdiya_blowfish_decryption(plainText,key,cipherText);
input [63:0]plainText;
input [448:0]key;
output [63:0]cipherText;
wire [31:0]ptr,ptl;
wire [31:0]k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13,k14;
wire [31:0]p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18;
wire [31:0]par1,par2,par3,par4,par5,par6,par7,par8,par9,par10,par11,par12,par13,par14,par15,par16,par17,par18;
wire [31:0]l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16;
wire [31:0]r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14,r15,r16;
wire [31:0]f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,f16,ctl,ctr;

assign ptr=plainText[31:0];
assign ptl=plainText[63:32];

assign k1=key[447:416];
assign k2=key[415:384];
assign k3=key[383:352];
assign k4=key[351:320];
assign k5=key[319:288];
assign k6=key[287:256];
assign k7=key[255:224];
assign k8=key[223:192];
assign k9=key[191:160];
assign k10=key[159:128];
assign k11=key[127:96];
assign k12=key[95:64];
assign k13=key[63:32];
assign k14=key[31:0];
assign p1=32'h243f6a88;
assign p2=32'h85a308d3;
assign p3=32'h13198a2e;
assign p4=32'h03707344;
assign p5=32'ha4093822;
assign p6=32'h299f31d0;
assign p7=32'h082efa98;
assign p8=32'hec4e6c89;
assign p9=32'h452821e6;
assign p10=32'h38d01377;
assign p11=32'hbe5466cf;
assign p12=32'h34e90c6c;
assign p13=32'hc0ac29b7;
assign p14=32'hc97c50dd;
assign p15=32'h3f84d5b5;
```

Transcript | hamdiya_blowfish_decryption.v | hamdiya_blowfish_encryption.v | hamdiya_feistel_function.v | sBox2.v | hamdiya_testbench_blowfish.v | sBox3.v | sBox4.v | sBox1.v | Library | Project

C:/Users/user/Desktop/myVerilog/Blow-fish-master/hamdia blowfish/hamdiya_blowfish_decryption.v (/hamdiya_testbench_blowfish/hamdiya_dec) - Default

```
assign p12=32'h34e90c6c;
assign p13=32'hc0ac29b7;
assign p14=32'hc97c50dd;
assign p15=32'h3f84d5b5;
assign p16=32'hb5470917;
assign p17=32'h9216d5d9;
assign p18=32'h8979fb1b;

assign par1=p1^k1;
assign par2=p2^k2;
assign par3=p3^k3;
assign par4=p4^k4;
assign par5=p5^k5;
assign par6=p6^k6;
assign par7=p7^k7;
assign par8=p8^k8;
assign par9=p9^k9;
assign par10=p10^k10;
assign par11=p11^k11;
assign par12=p12^k12;
assign par13=p13^k13;
assign par14=p14^k14;
assign par15=p15^k1;
assign par16=p16^k2;
assign par17=p17^k3;
assign par18=p18^k4;

assign l1=par18^ptl;
hamdiya_feistel_function a1(l1,f1);
assign r1=ptr^f1;

assign l2=par17^r1;
hamdiya_feistel_function a2(l2,f2);
assign r2=l1^f2;

assign l3=par16^r2;
hamdiya_feistel_function a3(l3,f3);
assign r3=l2^f3;

assign l4=par15^r3;
hamdiya_feistel_function a4(l4,f4);
assign r4=l3^f4;

assign l5=par14^r4;
hamdiya_feistel_function a5(l5,f5);
```

Transcript | hamdiya_blowfish_decryption.v | hamdiya_blowfish_encryption.v | hamdiya_feistel_function.v | sBox2.v | hamdiya_testbench_blowfish.v | sBox3.v | sBox4.v | sBox1.v | Library | Project

C:/Users/user/Desktop/myVerilog/Blow-fish-master/hamdia blowfish/hamdiya_blowfish_decryption.v (/hamdiya_testbench_blowfish/hamdiya_dec) - Default

```
assign l4=par15^r3;
hamdiya_feistel_function a4(l4,f4);
assign r4=l3^f4;

assign l5=par14^r4;
hamdiya_feistel_function a5(l5,f5);
assign r5=l4^f5;

assign l6=par13^r5;
hamdiya_feistel_function a6(l6,f6);
assign r6=l5^f6;

assign l7=par12^r6;
hamdiya_feistel_function a7(l7,f7);
assign r7=l6^f7;

assign l8=par11^r7;
hamdiya_feistel_function a8(l8,f8);
assign r8=l7^f8;

assign l9=par10^r8;
hamdiya_feistel_function a9(l9,f9);
assign r9=l8^f9;

assign l10=par9^r9;
hamdiya_feistel_function a10(l10,f10);
assign r10=l9^f10;

assign l11=par8^r10;
hamdiya_feistel_function a11(l11,f11);
assign r11=l10^f11;

assign l12=par7^r11;
hamdiya_feistel_function a12(l12,f12);
assign r12=l11^f12;

assign l13=par6^r12;
hamdiya_feistel_function a13(l13,f13);
assign r13=l12^f13;

assign l14=par5^r13;
hamdiya_feistel_function a14(l14,f14);
assign r14=l13^f14;

assign l15=par4^r14;
```

Transcript | hamdiya_blowfish_decryption.v | hamdiya_blowfish_encryption.v | hamdiya_feistel_function.v | sBox2.v | hamdiya_testbench_blowfish.v | sBox3.v | sBox4.v | sBox1.v | Library | Project

C:/Users/user/Desktop/myVerilog/Blow-fish-master/hamdia blowfish/hamdiya_blowfish_decryption.v (/hamdiya_testbench_blowfish/hamdiya_dec) - Default

assign l8=par11^r7;
hamdiya_feistel_function a8(l8,f8);
assign r8=l7^f8;

assign l9=par10^r8;
hamdiya_feistel_function a9(l9,f9);
assign r9=l8^f9;

assign l10=par9^r9;
hamdiya_feistel_function a10(l10,f10);
assign r10=l9^f10;

assign l11=par8^r10;
hamdiya_feistel_function a11(l11,f11);
assign r11=l10^f11;

assign l12=par7^r11;
hamdiya_feistel_function a12(l12,f12);
assign r12=l11^f12;

assign l13=par6^r12;
hamdiya_feistel_function a13(l13,f13);
assign r13=l12^f13;

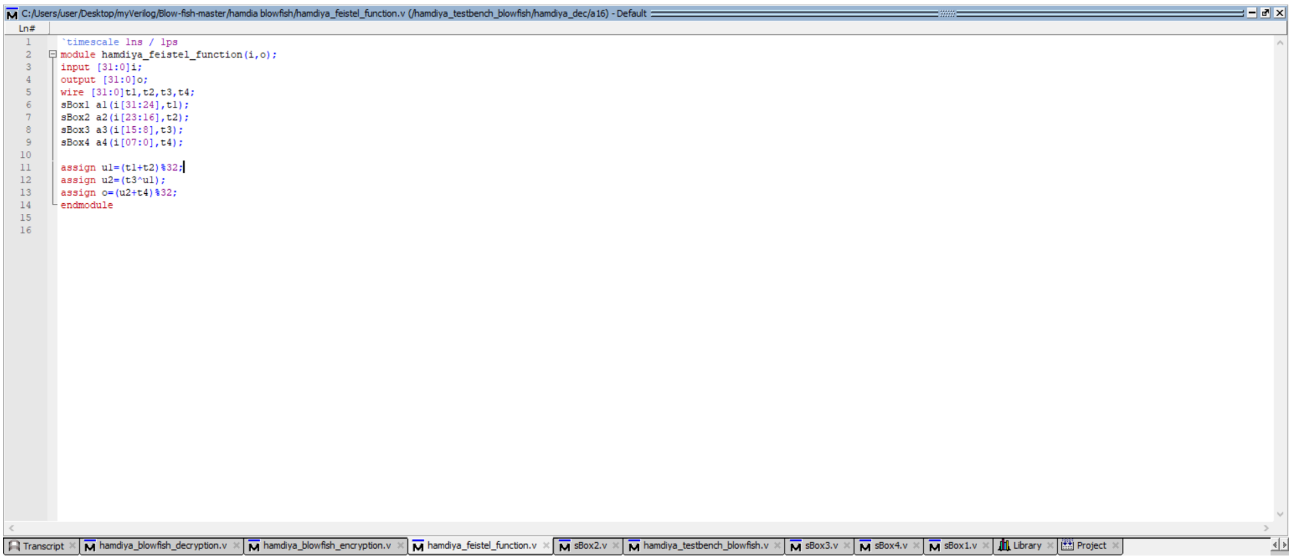
assign l14=par5^r13;
hamdiya_feistel_function a14(l14,f14);
assign r14=l13^f14;

assign l15=par4^r14;

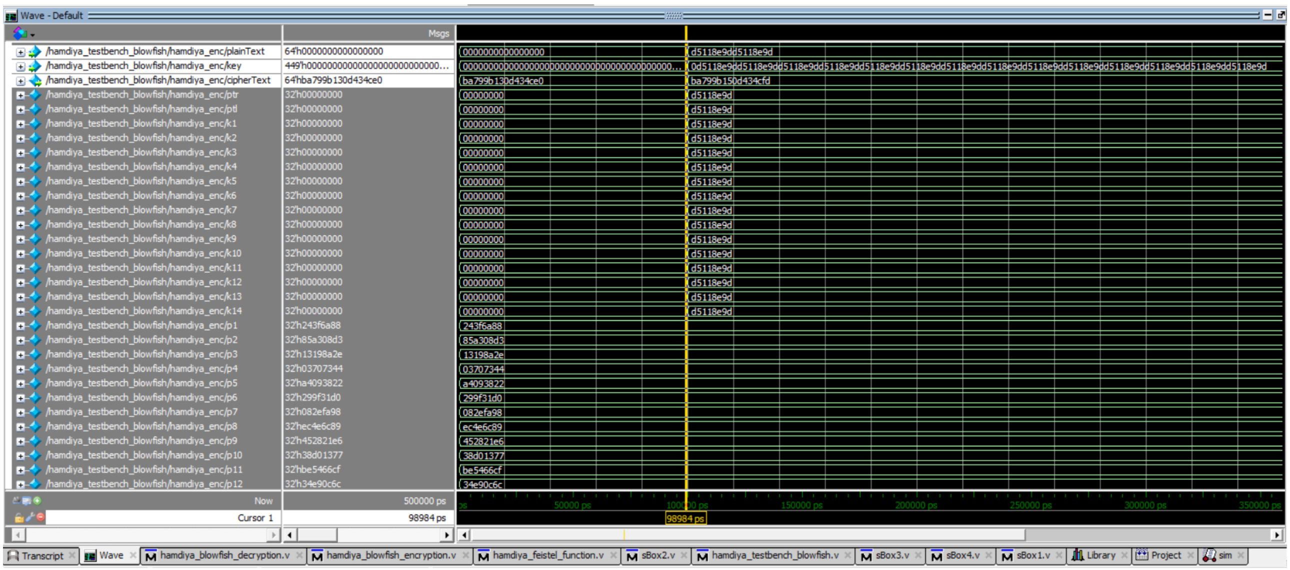
```
hamdiya_feistel_function a15(l15,f15);
assign r15=l14^f15;

assign l16=par3^r15;
hamdiya_feistel_function a16(l16,f16);
assign r16=l15^f16;

assign ctl=par1^l16;
assign ctr=par2^r16;

assign cipherText={ctl,ctr};

endmodule
```

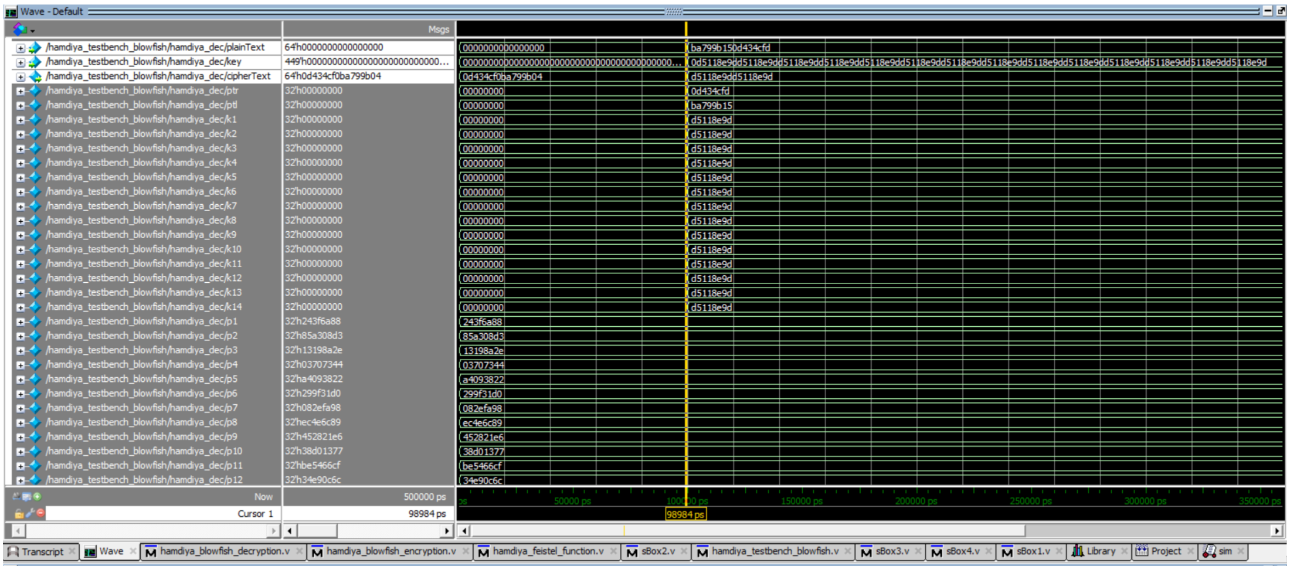
Transcript | hamdiya_blowfish_decryption.v | hamdiya_blowfish_encryption.v | hamdiya_feistel_function.v | sBox2.v | hamdiya_testbench_blowfish.v | sBox3.v | sBox4.v | sBox1.v | Library | Project

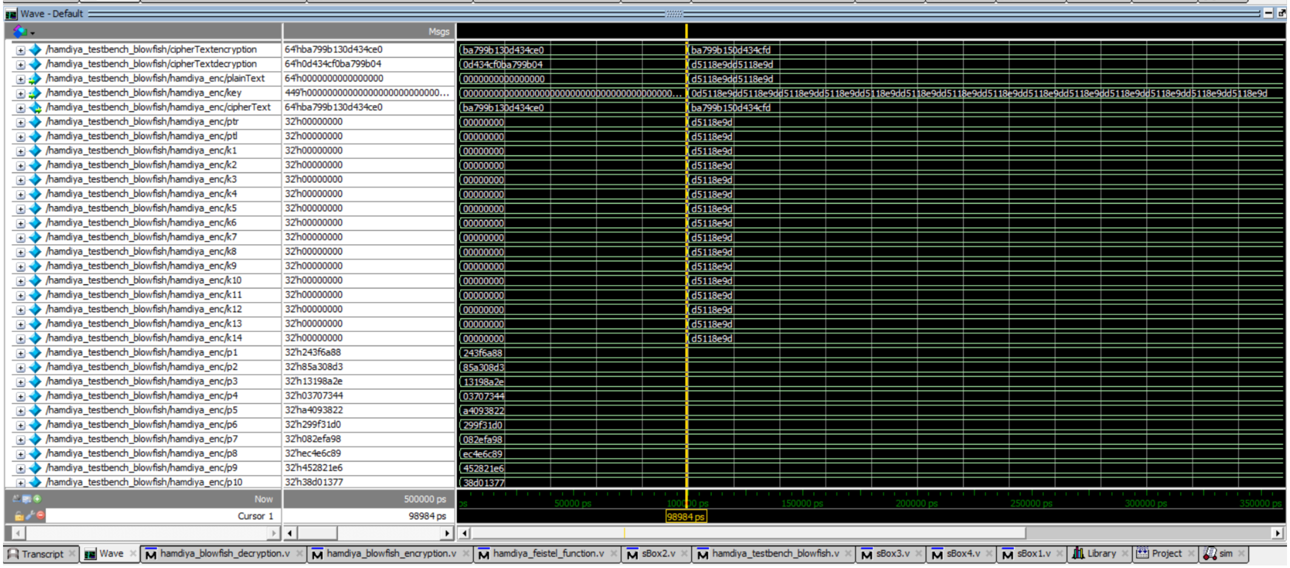
Feistel_Function:



Simulated result of encryption and decryption:




Wave - Default
Msgs
/hamdiya_testbench_blowfish/hamdiya_dec/plainText 64'h0000000000000000
/hamdiya_testbench_blowfish/hamdiya_dec/key
/hamdiya_testbench_blowfish/hamdiya_dec/cipherText 64'h0d434cf0ba799b04
/hamdiya_testbench_blowfish/hamdiya_dec/ptr 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/ptl 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k1 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k2 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k3 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k4 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k5 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k6 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k7 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k8 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k9 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k10 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k11 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k12 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k13 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/k14 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_dec/p1 32'h243f6a88
/hamdiya_testbench_blowfish/hamdiya_dec/p2 32'h85a308d3
/hamdiya_testbench_blowfish/hamdiya_dec/p3 32'h13198a2e
/hamdiya_testbench_blowfish/hamdiya_dec/p4 32'h03707344
/hamdiya_testbench_blowfish/hamdiya_dec/p5 32'ha4093822
/hamdiya_testbench_blowfish/hamdiya_dec/p6 32'h299f31d0
/hamdiya_testbench_blowfish/hamdiya_dec/p7 32'h082efa98
/hamdiya_testbench_blowfish/hamdiya_dec/p8 32'hec4e6c89
/hamdiya_testbench_blowfish/hamdiya_dec/p9 32'h452821e6
/hamdiya_testbench_blowfish/hamdiya_dec/p10 32'h38d01377
/hamdiya_testbench_blowfish/hamdiya_dec/p11 32'hbe5466cf
/hamdiya_testbench_blowfish/hamdiya_dec/p12 32'h34e90c6c
ba799b150d434cfd
d5118e9dd5118e9d
0d434cfd
ba799b15
d5118e9d
Now 500000 ps
Cursor 1 98984 ps
Transcript
Wave
hamdiya_blowfish_decryption.v
hamdiya_blowfish_encryption.v
hamdiya_feistel_function.v
sBox2.v
hamdiya_testbench_blowfish.v
sBox3.v
sBox4.v
sBox1.v
Library
Project
sim


Wave - Default
Msgs
/hamdiya_testbench_blowfish/cipherTextencryption 64'hba799b130d434ce0
/hamdiya_testbench_blowfish/cipherTextdecryption 64'h0d434cf0ba799b04
/hamdiya_testbench_blowfish/hamdiya_enc/plainText 64'h0000000000000000
/hamdiya_testbench_blowfish/hamdiya_enc/key
/hamdiya_testbench_blowfish/hamdiya_enc/cipherText 64'hba799b130d434ce0
/hamdiya_testbench_blowfish/hamdiya_enc/ptr 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/ptl 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k1 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k2 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k3 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k4 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k5 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k6 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k7 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k8 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k9 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k10 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k11 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k12 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k13 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/k14 32'h00000000
/hamdiya_testbench_blowfish/hamdiya_enc/p1 32'h243f6a88
/hamdiya_testbench_blowfish/hamdiya_enc/p2 32'h85a308d3
/hamdiya_testbench_blowfish/hamdiya_enc/p3 32'h13198a2e
/hamdiya_testbench_blowfish/hamdiya_enc/p4 32'h03707344
/hamdiya_testbench_blowfish/hamdiya_enc/p5 32'ha4093822
/hamdiya_testbench_blowfish/hamdiya_enc/p6 32'h299f31d0
/hamdiya_testbench_blowfish/hamdiya_enc/p7 32'h082efa98
/hamdiya_testbench_blowfish/hamdiya_enc/p8 32'hec4e6c89
/hamdiya_testbench_blowfish/hamdiya_enc/p9 32'h452821e6
/hamdiya_testbench_blowfish/hamdiya_enc/p10 32'h38d01377
ba799b150d434cfd
d5118e9dd5118e9d
d5118e9d
Now 500000 ps
Cursor 1 98984 ps
Transcript
Wave
hamdiya_blowfish_decryption.v
hamdiya_blowfish_encryption.v
hamdiya_feistel_function.v
sBox2.v
hamdiya_testbench_blowfish.v
sBox3.v
sBox4.v
sBox1.v
Library
Project
sim