Report

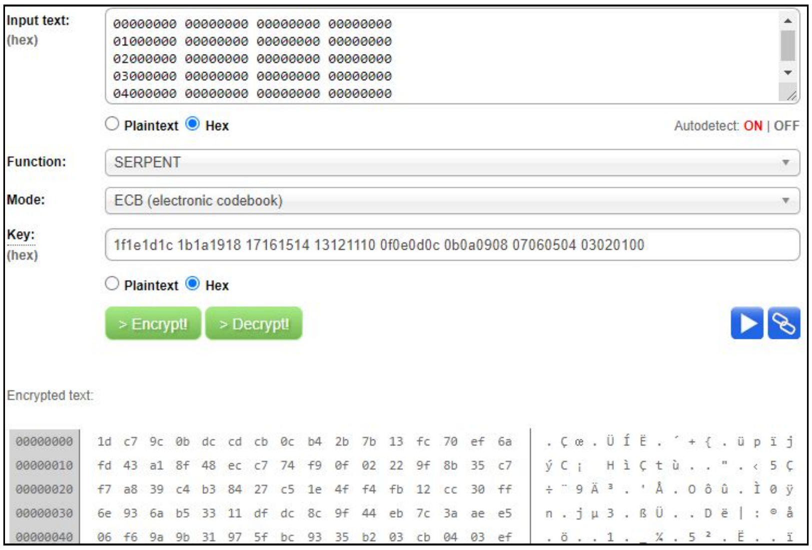
module structure

```
.
├── Report.pdf
├── tb_test.v                  // 測加密後解密
├── tb_practice.v              // 原本的testbench
└── Serpent
        ├── encrypt            // 加密用的
        │    ├── encryption.v // 32 rounds
        │    ├── round.v      // Key mixing → S_box → Transformation
        │    └── S_box.v      // S box
        ├── decrypt            // 解密用的
        │    ├── decryption.v // 32 rounds, 倒過來
        │    ├── round_inv.v  // Transformation_inv → S_box_inv → Key
mixing
        │    └── S_box_inv.v  // Inverse S box
        ├── keygen             // 生33個key用的
        │    └── key_gen.v    // 生33個key用的
        └── codegen.py         // 輔助用的
```

Test

```
iverilog tb_test.v
vvp a.out
iverilog tb_practice.v
vvp a.out
```

Result for tb_practice.v



Input text:
(hex)

```
00000000 00000000 00000000 00000000
01000000 00000000 00000000 00000000
02000000 00000000 00000000 00000000
03000000 00000000 00000000 00000000
04000000 00000000 00000000 00000000
```

○ Plaintext ● Hex

Autodetect: ON | OFF

Function: SERPENT

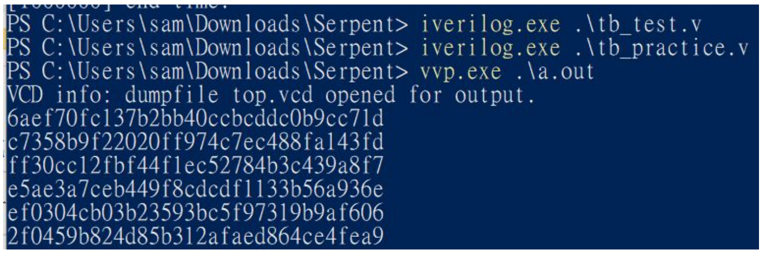
Mode: ECB (electronic codebook)

Key:
(hex)
1f1e1d1c 1b1a1918 17161514 13121110 0f0e0d0c 0b0a0908 07060504 03020100

○ Plaintext ● Hex

Encrypted text:

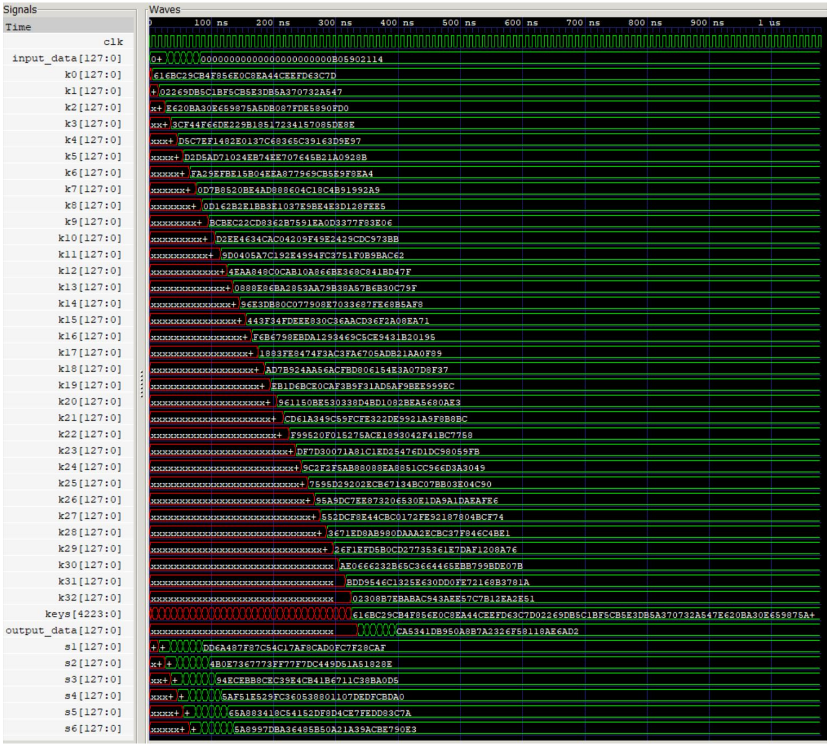
```
00000000  1d c7 9c 0b dc cd cb 0c b4 2b 7b 13 fc 70 ef 6a  . Ç œ . Ü Í Ë . ´ + { . ü p ï j
00000010  fd 43 a1 8f 48 ec c7 74 f9 0f 02 22 9f 8b 35 c7  ý C ¡   H ì Ç t ù . . " . ‹ 5 Ç
00000020  f7 a8 39 c4 b3 84 27 c5 1e 4f f4 fb 12 cc 30 ff  ÷ ¨ 9 Ä ³ . ' Å . O ô û . Ì 0 ÿ
00000030  6e 93 6a b5 33 11 df dc 8c 9f 44 eb 7c 3a ae e5  n . j µ 3 . ß Ü . . D ë | : ® å
00000040  06 f6 9a 9b 31 97 5f bc 93 35 b2 03 cb 04 03 ef  . ö . . 1 . _ ¼ . 5 ² . Ë . . ï
```



```
PS C:\Users\sam\Downloads\Serpent> iverilog.exe .\tb_test.v
PS C:\Users\sam\Downloads\Serpent> iverilog.exe .\tb_practice.v
PS C:\Users\sam\Downloads\Serpent> vvp.exe .\a.out
VCD info: dumpfile top.vcd opened for output.
6aef70fc137b2bb40ccbcddc0b9cc71d
c7358b9f22020ff974c7ec488fa143fd
ff30cc12fbf44f1ec52784b3c439a8f7
e5ae3a7ceb449f8cdcdf1133b56a936e
ef0304cb03b23593bc5f97319b9af606
2f0459b824d85b312afaed864ce4fea9
```

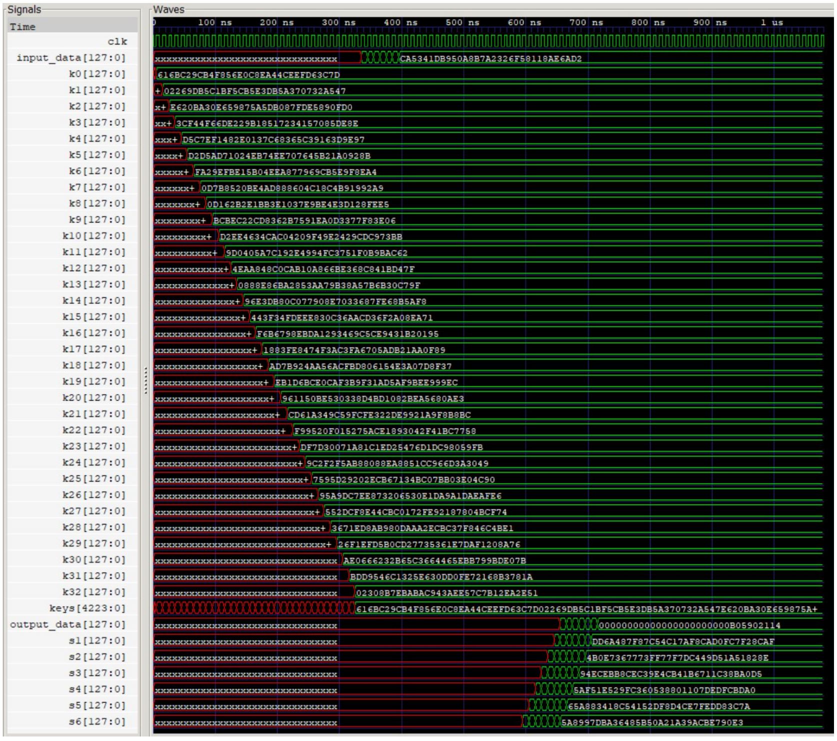
說明

Encryption:


Signals
Waves
Time
0
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1 us
clk
input_data[127:0] 0+ 000000000000000000000000B05902114
k0[127:0] 616BC29CB4F856E0C8EA44CEEFD63C7D
k1[127:0] 02269DB5C1BF5CB5E3DB5A370732A547
k2[127:0] E620BA30E659875A5DB087FDE5890FD0
k3[127:0] 3CF44F66DE229B18517234157085DE8E
k4[127:0] D5C7EF1482E0137C68365C39163D9E97
k5[127:0] D2D5AD71024EB74EE707645B21A0928B
k6[127:0] FA29EFBE15B04EEA877969CB5E9F8EA4
k7[127:0] 0D7B8520BE4AD888604C18C4B91992A9
k8[127:0] 0D162B2E1BB3E1037E9BE4E3D128FEE5
k9[127:0] BCBEC22CD8362B7591EA0D3377F83E06
k10[127:0] D2EE4634CAC04209F49E2429CDC973BB
k11[127:0] 9D0405A7C192E4994FC3751F0B9BAC62
k12[127:0] 4EAA848C0CAB10A866BE368C841BD47F
k13[127:0] 0888E86BA2853AA79B38A57B6B30C79F
k14[127:0] 96E3DB80C077908E7033687FE68B5AF8
k15[127:0] 443F34FDEEE830C36AACD36F2A08EA71
k16[127:0] F6B6798EBDA1293469C5CE9431B20195
k17[127:0] 1883FE8474F3AC3FA6705ADB21AA0F89
k18[127:0] AD7B924AA56ACFBD806154E3A07D8F37
k19[127:0] EB1D6BCE0CAF3B9F31AD5AF9BEE999EC
k20[127:0] 961150BE530338D4BD1082BEA5680AE3
k21[127:0] CD61A349C59FCFE322DE9921A9F8B8BC
k22[127:0] F99520F015275ACE1893042F41BC7758
k23[127:0] DF7D30071A81C1ED25476D1DC98059FB
k24[127:0] 9C2F2F5AB88088EA8851CC966D3A3049
k25[127:0] 7595D29202ECB67134BC07BB03E04C90
k26[127:0] 95A9DC7EE873206530E1DA9A1DAEAFE6
k27[127:0] 552DCF8E44CBC0172FE92187804BCF74
k28[127:0] 3671ED8AB980DAAA2ECBC37F846C4BE1
k29[127:0] 26F1EFD5B0CD27735361E7DAF1208A76
k30[127:0] AE0666232B65C3664465EBB799BDE07B
k31[127:0] BDD9546C1325E630DD0FE72168B3781A
k32[127:0] 02308B7EBABAC943AEE57C7B12EA2E51
keys[4223:0] 616BC29CB4F856E0C8EA44CEEFD63C7D02269DB5C1BF5CB5E3DB5A370732A547E620BA30E659875A+
output_data[127:0] CA5341DB950A8B7A2326F58118AE6AD2
s1[127:0] DD6A487F87C54C17AF8CAD0FC7F28CAF
s2[127:0] 4B0E7367773FF77F7DC449D51A51828E
s3[127:0] 94ECEBB8CEC39E4CB41B6711C38BA0D5
s4[127:0] 5AF51E529FC360538801107DEDFCBDA0
s5[127:0] 65A883418C54152DF8D4CE7FEDD83C7A
s6[127:0] 5A8997DBA36485B50A21A39ACBE790E3

每個cycle生4個omega，也就是一個key，總共33把key，要33個cycle；每個cycle生下一個state，pipeline做，第一把要等一個cycle，剩下的32個cycle，做32個round。故encryption要33個cycle，而input可以每個cycle都進來。

Decryption:

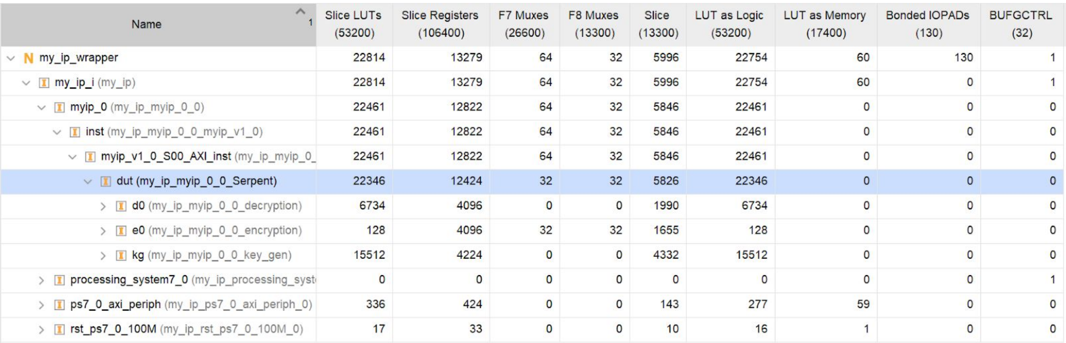

Signals
Waves
Time
0
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1 us
clk
input_data[127:0] CA5341DB950A8B7A2326F58118AE6AD2
k0[127:0] 616BC29CB4F856E0C8EA44CEEFD63C7D
k1[127:0] 02269DB5C1BF5CB5E3DB5A370732A547
k2[127:0] E620BA30E659875A5DB087FDE5890FD0
k3[127:0] 3CF44F66DE229B18517234157085DE8E
k4[127:0] D5C7EF1482E0137C68365C39163D9E97
k5[127:0] D2D5AD71024EB74EE707645B21A0928B
k6[127:0] FA29EFBE15B04EEA877969CB5E9F8EA4
k7[127:0] 0D7B8520BE4AD888604C18C4B91992A9
k8[127:0] 0D162B2E1BB3E1037E9BE4E3D128FEE5
k9[127:0] BCBEC22CD8362B7591EA0D3377F83E06
k10[127:0] D2EE4634CAC04209F49E2429CDC973BB
k11[127:0] 9D0405A7C192E4994FC3751F0B9BAC62
k12[127:0] 4EAA848C0CAB10A866BE368C841BD47F
k13[127:0] 0888E86BA2853AA79B38A57B6B30C79F
k14[127:0] 96E3DB80C077908E7033687FE68B5AF8
k15[127:0] 443F34FDEEE830C36AACD36F2A08EA71
k16[127:0] F6B6798EBDA1293469C5CE9431B20195
k17[127:0] 1883FE8474F3AC3FA6705ADB21AA0F89
k18[127:0] AD7B924AA56ACFBD806154E3A07D8F37
k19[127:0] EB1D6BCE0CAF3B9F31AD5AF9BEE999EC
k20[127:0] 961150BE530338D4BD1082BEA5680AE3
k21[127:0] CD61A349C59FCFE322DE9921A9F8B8BC
k22[127:0] F99520F015275ACE1893042F41BC7758
k23[127:0] DF7D30071A81C1ED25476D1DC98059FB
k24[127:0] 9C2F2F5AB88088EA8851CC966D3A3049
k25[127:0] 7595D29202ECB67134BC07BB03E04C90
k26[127:0] 95A9DC7EE873206530E1DA9A1DAEAFE6
k27[127:0] 552DCF8E44CBC0172FE92187804BCF74
k28[127:0] 3671ED8AB980DAAA2ECBC37F846C4BE1
k29[127:0] 26F1EFD5B0CD27735361E7DAF1208A76
k30[127:0] AE0666232B65C3664465EBB799BDE07B
k31[127:0] BDD9546C1325E630DD0FE72168B3781A
k32[127:0] 02308B7EBABAC943AEE57C7B12EA2E51
keys[4223:0] 616BC29CB4F856E0C8EA44CEEFD63C7D02269DB5C1BF5CB5E3DB5A370732A547E620BA30E659875A+
output_data[127:0] 000000000000000000000000B05902114
s1[127:0] DD6A487F87C54C17AF8CAD0FC7F28CAF
s2[127:0] 4B0E7367773FF77F7DC449D51A51828E
s3[127:0] 94ECEBB8CEC39E4CB41B6711C38BA0D5
s4[127:0] 5AF51E529FC360538801107DEDFCBDA0
s5[127:0] 65A883418C54152DF8D4CE7FEDD83C7A
s6[127:0] 5A8997DBA36485B50A21A39ACBE790E3

Decryption key要倒著xor，所以要先等key出來，故要33+32個cycle才能結束decryption。如果key先算好則可以只要32 cycle就結束。

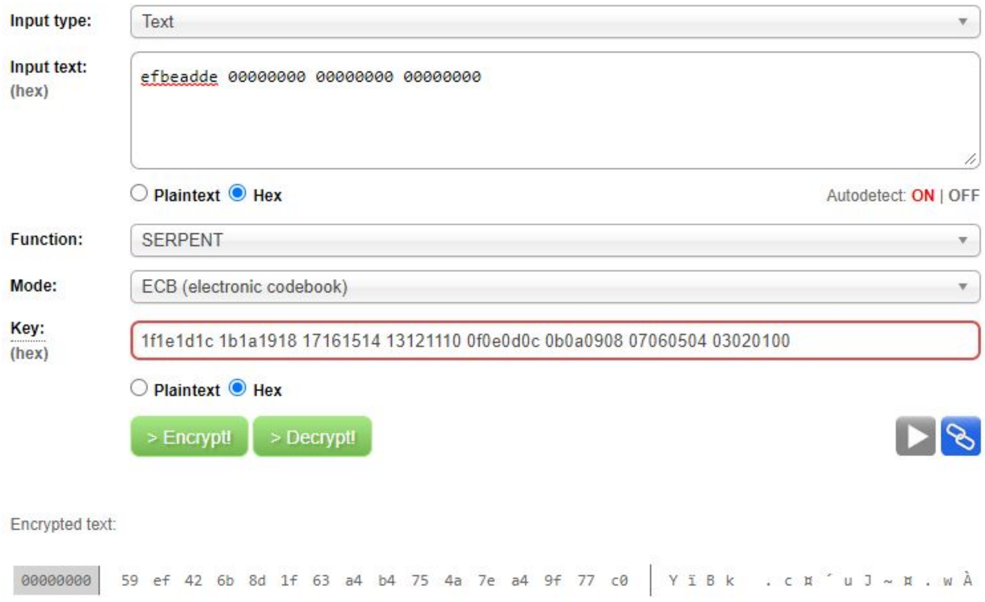
Test on board

```
always @(*)
begin
      case ( axi_araddr[ADDR_LSB+OPT_MEM_ADDR_BITS:ADDR_LSB] )
        4'h0   : reg_data_out <= slv_reg0;
        4'h1   : reg_data_out <= slv_reg1;
        4'h2   : reg_data_out <= slv_reg2;
        4'h3   : reg_data_out <= slv_reg3;
        4'h4   : reg_data_out <= slv_reg4;
        4'h5   : reg_data_out <= slv_reg5;
        4'h6   : reg_data_out <= slv_reg6;
        4'h7   : reg_data_out <= slv_reg7;
        4'h8   : reg_data_out <= slv_reg8;
        4'h9   : reg_data_out <= slv_reg9;
        4'hA   : reg_data_out <= out_128[127:96];
        4'hB   : reg_data_out <= out_128[95:64];
        4'hC   : reg_data_out <= out_128[63:32];
        4'hD   : reg_data_out <= out_128[31:0];
        4'hE   : reg_data_out <= slv_reg14;
        4'hF   : reg_data_out <= out_32[31:0];
        default : reg_data_out <= 0;
      endcase
end
Serpent  dut(    .i_clk( S_AXI_ACLK ),
                 .i_resetn(  slv_reg14[0] ),
                 .i_dir( slv_reg14[1] ),
                 .i_key( {64'h0001020304050607, slv_reg9, slv_reg8, slv_reg7, slv_reg6, slv_reg5, slv_reg4} ),
                 .i_new_block( slv_reg14[2] ),
                 .i_data( {slv_reg3, slv_reg2, slv_reg1, slv_reg0} ),
                 .o_ready( out_32[0] ),
                 .o_output( out_128 ) );
```

Resource



Name	Slice LUTs (53200)	Slice Registers (106400)	F7 Muxes (26600)	F8 Muxes (13300)	Slice (13300)	LUT as Logic (53200)	LUT as Memory (17400)	Bonded IOPADs (130)	BUFGCTRL (32)
my_ip_wrapper	22814	13279	64	32	5996	22754	60	130	1
my_ip_i (my_ip)	22814	13279	64	32	5996	22754	60	0	1
myip_0 (my_ip_myip_0_0)	22461	12822	64	32	5846	22461	0	0	0
inst (my_ip_myip_0_0_myip_v1_0)	22461	12822	64	32	5846	22461	0	0	0
myip_v1_0_S00_AXI_inst (my_ip_myip_0_	22461	12822	64	32	5846	22461	0	0	0
dut (my_ip_myip_0_0_Serpent)	22346	12424	32	32	5826	22346	0	0	0
d0 (my_ip_myip_0_0_decryption)	6734	4096	0	0	1990	6734	0	0	0
e0 (my_ip_myip_0_0_encryption)	128	4096	32	32	1655	128	0	0	0
kg (my_ip_myip_0_0_key_gen)	15512	4224	0	0	4332	15512	0	0	0
processing_system7_0 (my_ip_processing_syst	0	0	0	0	0	0	0	0	1
ps7_0_axi_periph (my_ip_ps7_0_axi_periph_0)	336	424	0	0	143	277	59	0	0
rst_ps7_0_100M (my_ip_rst_ps7_0_100M_0)	17	33	0	0	10	16	1	0	0



Input type: Text

Input text: (hex)

efbeadde 00000000 00000000 00000000

○ Plaintext ● Hex

Autodetect: ON | OFF

Function: SERPENT

Mode: ECB (electronic codebook)

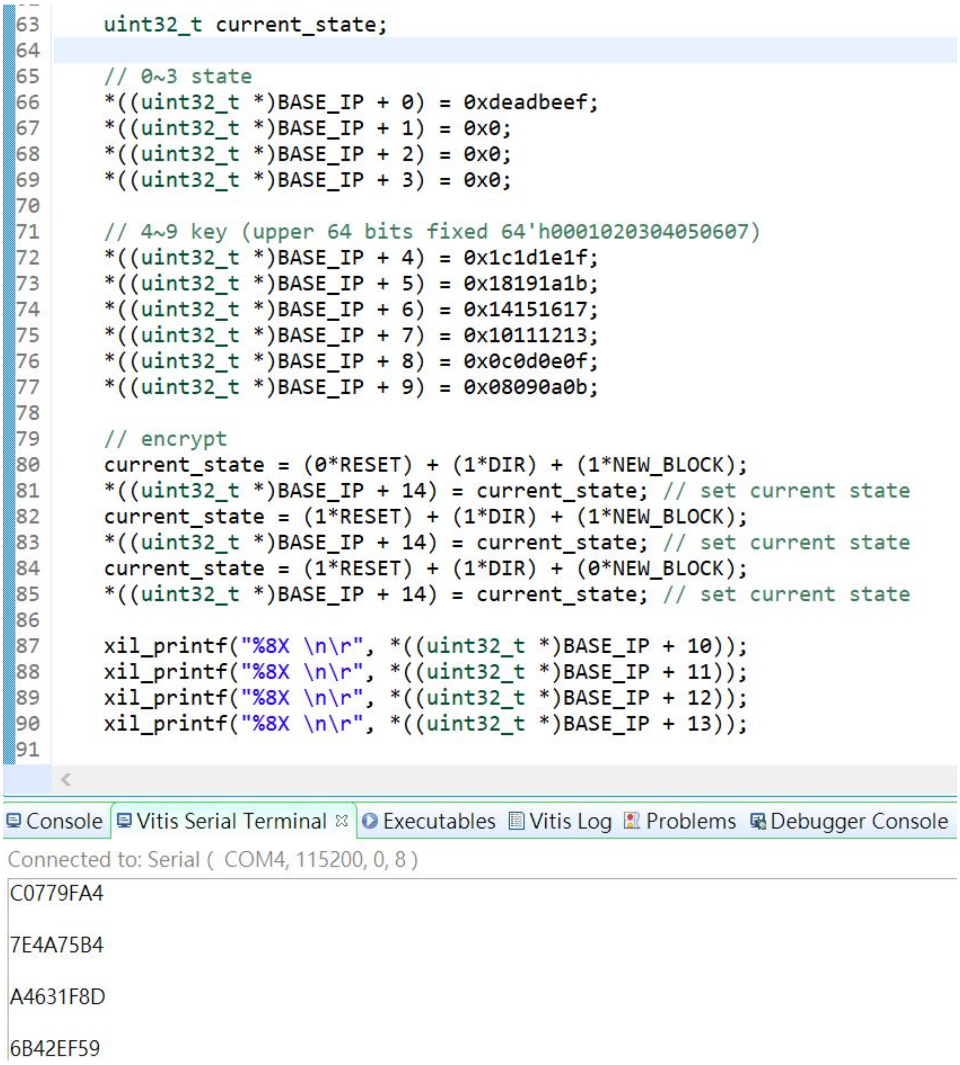
Key: (hex)

1f1e1d1c 1b1a1918 17161514 13121110 0f0e0d0c 0b0a0908 07060504 03020100

○ Plaintext ● Hex

Encrypted text:

```
00000000  59 ef 42 6b 8d 1f 63 a4 b4 75 4a 7e a4 9f 77 c0 | Y ï B k  . c ¤ ´ u J ~ ¤ . w À
```



```
    uint32_t current_state;

    // 0~3 state
    *((uint32_t *)BASE_IP + 0) = 0xdeadbeef;
    *((uint32_t *)BASE_IP + 1) = 0x0;
    *((uint32_t *)BASE_IP + 2) = 0x0;
    *((uint32_t *)BASE_IP + 3) = 0x0;

    // 4~9 key (upper 64 bits fixed 64'h0001020304050607)
    *((uint32_t *)BASE_IP + 4) = 0x1c1d1e1f;
    *((uint32_t *)BASE_IP + 5) = 0x18191a1b;
    *((uint32_t *)BASE_IP + 6) = 0x14151617;
    *((uint32_t *)BASE_IP + 7) = 0x10111213;
    *((uint32_t *)BASE_IP + 8) = 0x0c0d0e0f;
    *((uint32_t *)BASE_IP + 9) = 0x08090a0b;

    // encrypt
    current_state = (0*RESET) + (1*DIR) + (1*NEW_BLOCK);
    *((uint32_t *)BASE_IP + 14) = current_state; // set current state
    current_state = (1*RESET) + (1*DIR) + (1*NEW_BLOCK);
    *((uint32_t *)BASE_IP + 14) = current_state; // set current state
    current_state = (1*RESET) + (1*DIR) + (0*NEW_BLOCK);
    *((uint32_t *)BASE_IP + 14) = current_state; // set current state

    xil_printf("%8X \n\r", *((uint32_t *)BASE_IP + 10));
    xil_printf("%8X \n\r", *((uint32_t *)BASE_IP + 11));
    xil_printf("%8X \n\r", *((uint32_t *)BASE_IP + 12));
    xil_printf("%8X \n\r", *((uint32_t *)BASE_IP + 13));
```

```
Connected to: Serial ( COM4, 115200, 0, 8 )
C0779FA4

7E4A75B4

A4631F8D

6B42EF59
```