S-DES Verilog Implementation

Απόστολος Κονταρίνης

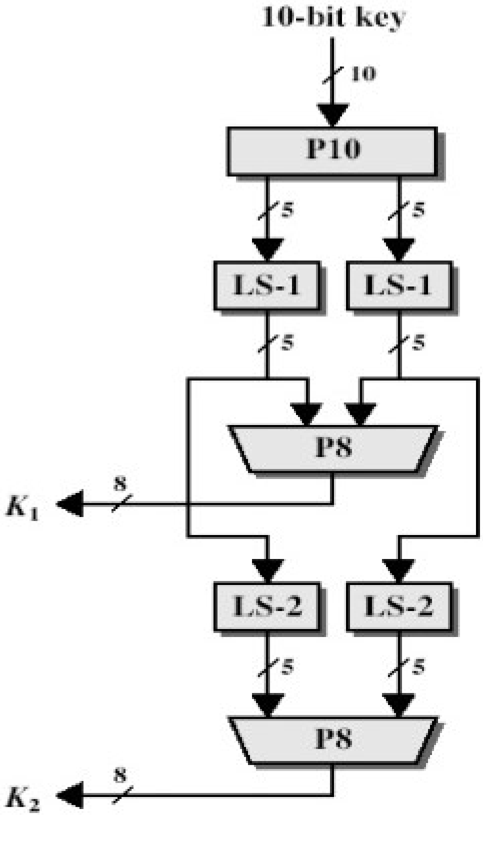
December 2021

Contents

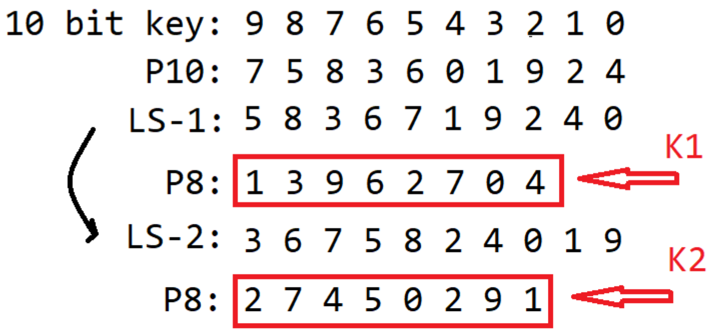
1 Algorithm Analysis

1.1 Key Generation

The Simple DES algorithm has an input key of 10 bits. From this key 2 subkeys are created using the following steps:



Following the steps we derive to:

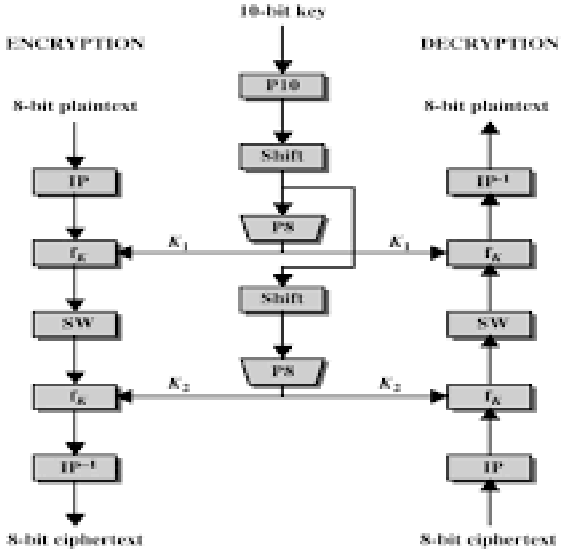


```
10 bit key: 9 8 7 6 5 4 3 2 1 0
       P10: 7 5 8 3 6 0 1 9 2 4
      LS-1: 5 8 3 6 7 1 9 2 4 0
        P8: 1 3 9 6 2 7 0 4
      LS-2: 3 6 7 5 8 2 4 0 1 9
        P8: 2 7 4 5 0 2 9 1
```

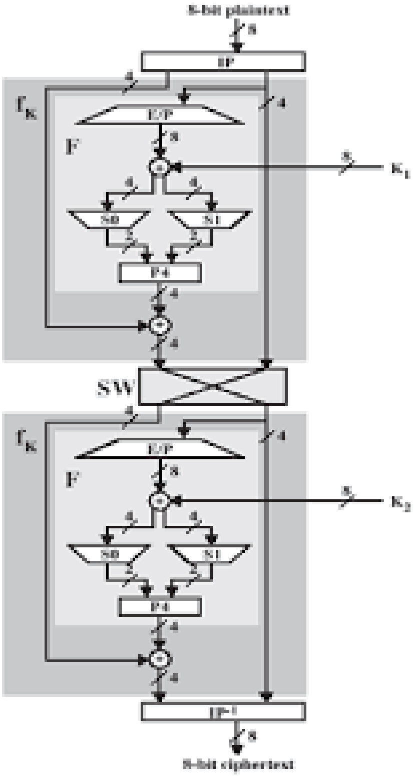
In code:
Key 1: $K1 = key[1], key[3], key[9], key[6], key[2], key[7], key[0], key[4];$
Key 2: $K2 = key[2], key[7], key[4], key[5], key[0], key[2], key[9], key[1];$

1.2 Algorithm

The Encryption and Decryption of S-DES are the same operation with the opposite sequence of keys.

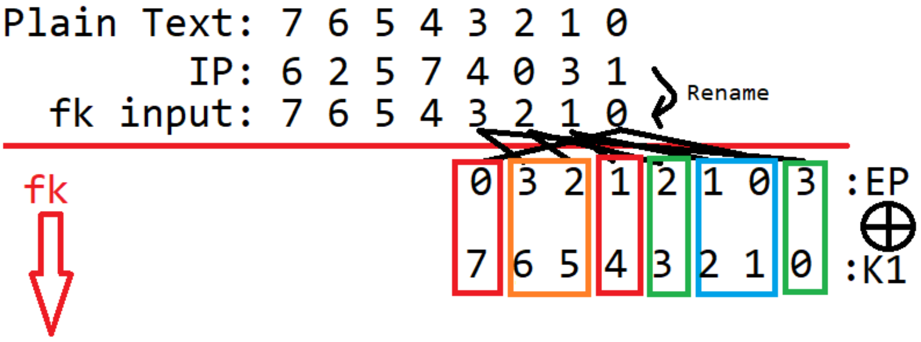


1.2.1 Function fk



The insides of the fk function can be seen here:

By following the steps of the fk function we arrive at:
(Rename means that the bits are renamed after their new positions)

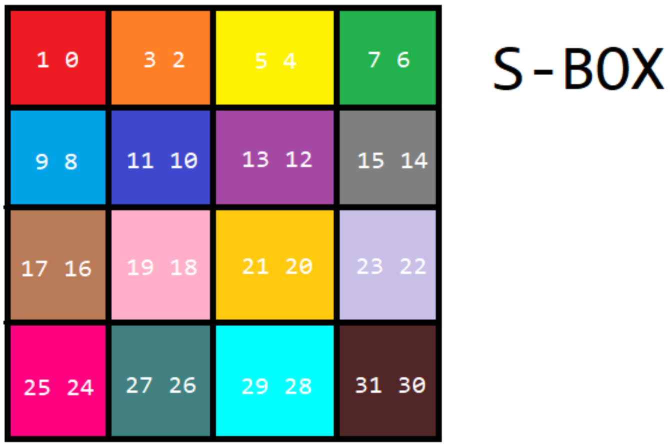


```
Plain Text: 7 6 5 4 3 2 1 0
        IP: 6 2 5 7 4 0 3 1
  fk input: 7 6 5 4 3 2 1 0
```

```
0 3 2 1 2 1 0 3 :EP
                 ⊕
7 6 5 4 3 2 1 0 :K1
```

```
row0[1] = fk input[0] xor K1[7]
row0[0] = fk input[1] xor K1[4]
col0[1] = fk input[3] xor K1[6]
col0[0] = fk input[2] xor K1[5]
row1[1] = fk input[2] xor K1[3]
row1[0] = fk input[3] xor K1[0]
col1[1] = fk input[1] xor K1[2]
col1[0] = fk input[0] xor K1[1]
```

The S-boxes are 32-bit inputs with the following form:



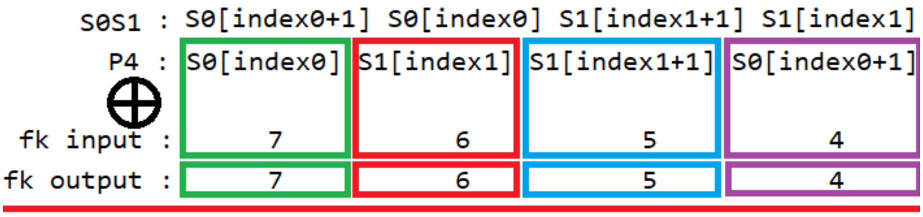
S-BOX

1 0	3 2	5 4	7 6
9 8	11 10	13 12	15 14
17 16	19 18	21 20	23 22
25 24	27 26	29 28	31 30

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

The index of each block is:
$index = 8 * row[1:0] + 2 * col[1:0] + 0/1$
Something that can be translated into a 5-bit index. 3 left shifts for the row bits and 1 left shift for the column bits.
S-Box0:
$index0[4] = row0[1] = fk\_input[0]\ xor\ K1[7]$
$index0[3] = row0[0] = fk\_input[1]\ xor\ K1[4]$
$index0[2] = col0[1] = fk\_input[3]\ xor\ K1[6]$
$index0[1] = col0[0] = fk\_input[2]\ xor\ K1[5]$
$index0[0] = 0$
S-Box1:
$index1[4] = row1[1] = fk\_input[2]\ xor\ K1[3]$
$index1[3] = row1[0] = fk\_input[3]\ xor\ K1[0]$
$index1[2] = col1[1] = fk\_input[1]\ xor\ K1[2]$
$index1[1] = col1[0] = fk\_input[0]\ xor\ K1[1]$
$index1[0] = 0$

Finally the last 2 steps of the fk function.

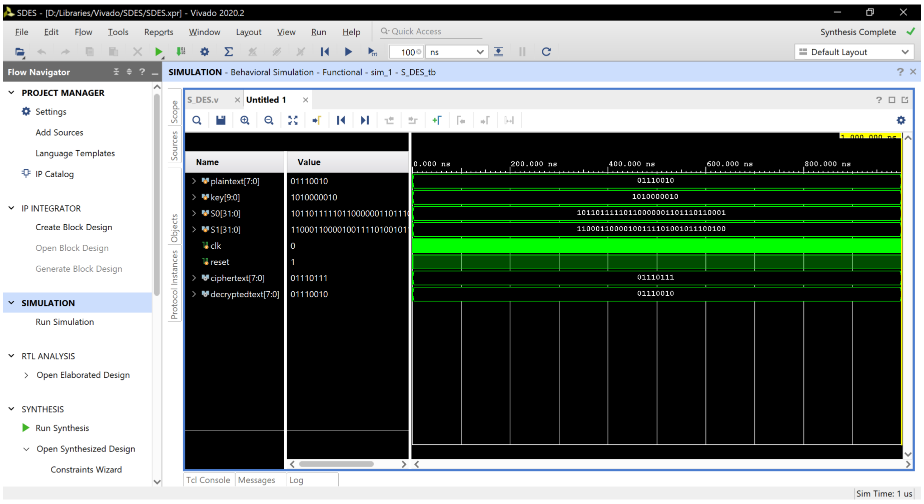


```
      S0S1 : S0[index0+1] S0[index0] S1[index1+1] S1[index1]
        P4 : S0[index0]   S1[index1] S1[index1+1] S0[index0+1]
        ⊕
  fk input :      7            6           5            4
 fk output :      7            6           5            4
```

2 Design

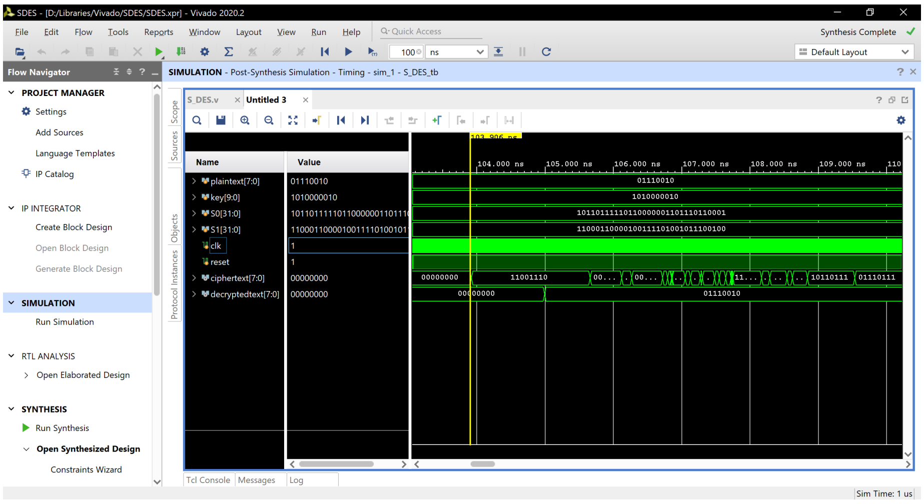
The Xilinx Vivado Software was used for the Design, Synthesis, Implementation and Pre/Post Synthesis Simulation. The board used is the ZedBoard Zynq Evaluation and Development Kit (xc7z020clg484-1).

2.1 Pre Synthesis Simulation



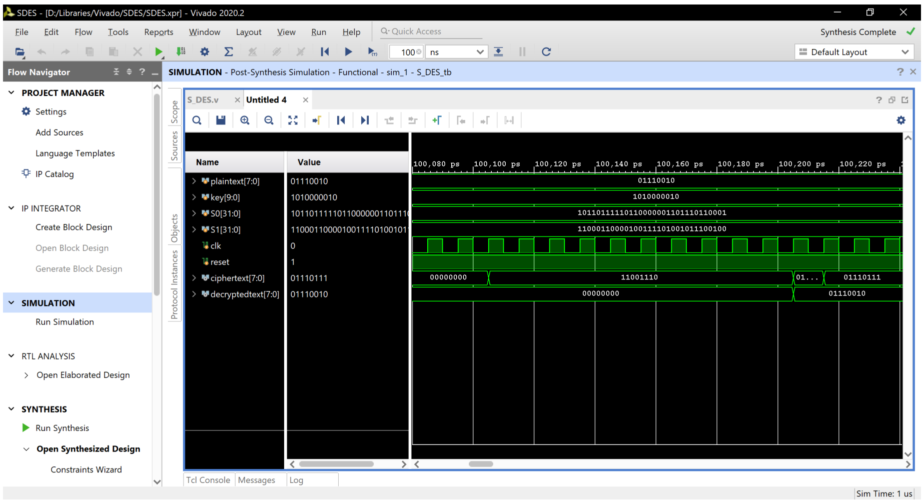
Name	Value
plaintext[7:0]	01110010
key[9:0]	1010000010
S0[31:0]	101101111101100000011011101
S1[31:0]	110001100001001111010010101
clk	0
reset	1
ciphertext[7:0]	01110111
decryptedtext[7:0]	01110010

2.2 Post Synthesis Timing Simulation



Name	Value
plaintext[7:0]	01110010
key[9:0]	1010000010
S0[31:0]	101101111101100000011011101
S1[31:0]	110001100001001111010010101
clk	1
reset	1
ciphertext[7:0]	00000000
decryptedtext[7:0]	00000000

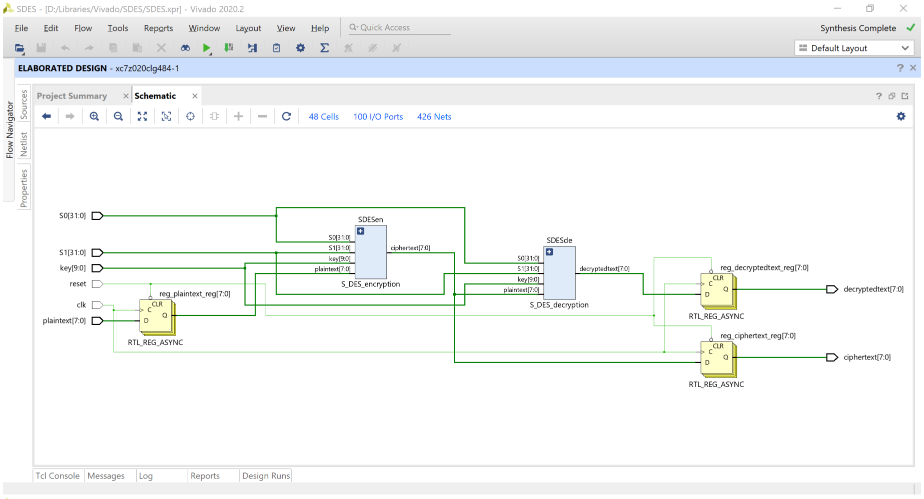
2.3 Post Synthesis Functional Simulation

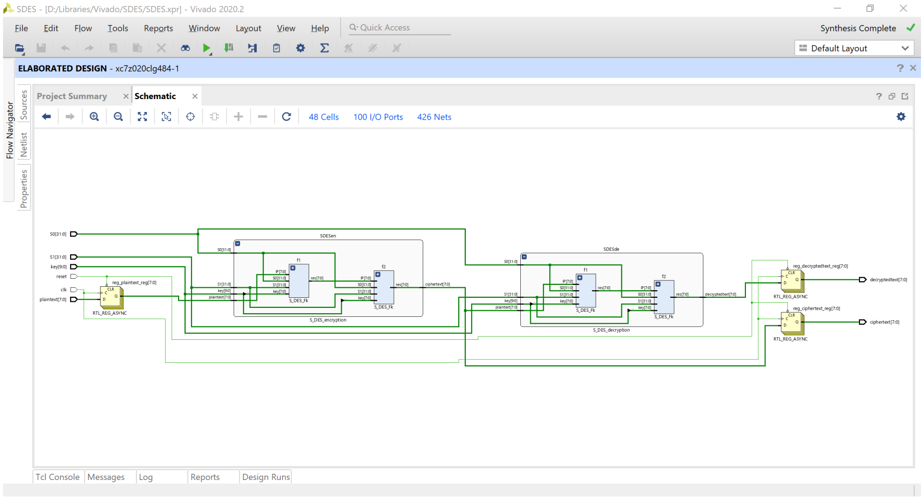


SIMULATION - Post-Synthesis Simulation - Functional - sim_1 - S_DES_tb

Name	Value
plaintext[7:0]	01110010
key[9:0]	1010000010
S0[31:0]	1011011111011000000110111(
S1[31:0]	1100011000010011110100101
clk	0
reset	1
ciphertext[7:0]	01110111
decryptedtext[7:0]	01110010

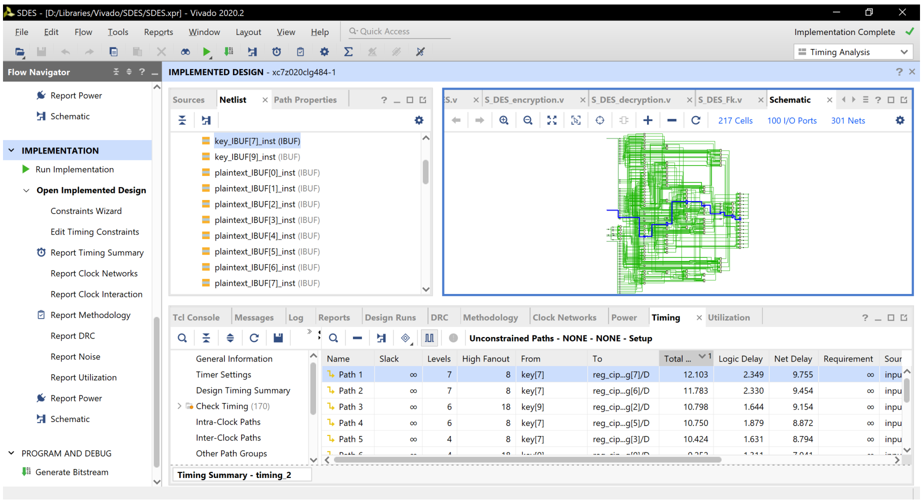
2.4 Hierarchical





2.5 Timings

The maximum delay of the circuit is
$max\_dalay = 12.103 + 2.761 = 14.864$
The maximum frequency of the design is
$max\_freq = 1/14.864$


SDES - [D:/Libraries/Vivado/SDES/SDES.xpr] - Vivado 2020.2

IMPLEMENTED DESIGN - xc7z020clg484-1

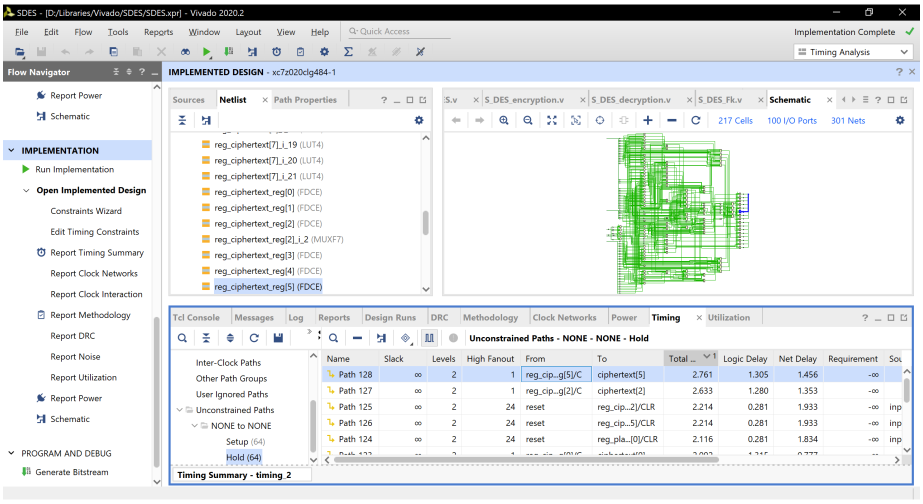
Netlist:
- key_IBUF[7]_inst (IBUF)
- key_IBUF[9]_inst (IBUF)
- plaintext_IBUF[0]_inst (IBUF)
- plaintext_IBUF[1]_inst (IBUF)
- plaintext_IBUF[2]_inst (IBUF)
- plaintext_IBUF[3]_inst (IBUF)
- plaintext_IBUF[4]_inst (IBUF)
- plaintext_IBUF[5]_inst (IBUF)
- plaintext_IBUF[6]_inst (IBUF)
- plaintext_IBUF[7]_inst (IBUF)

Schematic: 217 Cells, 100 I/O Ports, 301 Nets

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Sour
Path 1	∞	7	8	key[7]	reg_cip...g[7]/D	12.103	2.349	9.755	∞	inpu
Path 2	∞	7	8	key[7]	reg_cip...g[6]/D	11.783	2.330	9.454	∞	inpu
Path 3	∞	6	18	key[9]	reg_cip...g[2]/D	10.798	1.644	9.154	∞	inpu
Path 4	∞	6	8	key[7]	reg_cip...g[5]/D	10.750	1.879	8.872	∞	inpu
Path 5	∞	4	8	key[7]	reg_cip...g[3]/D	10.424	1.631	8.794	∞	inpu

Timing Summary - timing_2


SDES - [D:/Libraries/Vivado/SDES/SDES.xpr] - Vivado 2020.2

IMPLEMENTED DESIGN - xc7z020clg484-1

Netlist:
- reg_ciphertext[7]_i_19 (LUT4)
- reg_ciphertext[7]_i_20 (LUT4)
- reg_ciphertext[7]_i_21 (LUT4)
- reg_ciphertext_reg[0] (FDCE)
- reg_ciphertext_reg[1] (FDCE)
- reg_ciphertext_reg[2] (FDCE)
- reg_ciphertext_reg[2]_i_2 (MUXF7)
- reg_ciphertext_reg[3] (FDCE)
- reg_ciphertext_reg[4] (FDCE)
- reg_ciphertext_reg[5] (FDCE)

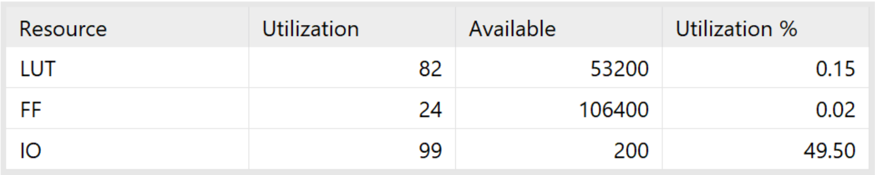
Schematic: 217 Cells, 100 I/O Ports, 301 Nets

Unconstrained Paths - NONE - NONE - Hold

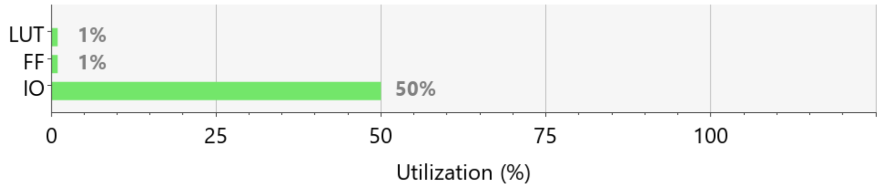
Name	Slack	Levels	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Sou
Path 128	∞	2	1	reg_cip...g[5]/C	ciphertext[5]	2.761	1.305	1.456	-∞	
Path 127	∞	2	1	reg_cip...g[2]/C	ciphertext[2]	2.633	1.280	1.353	-∞	
Path 125	∞	2	24	reset	reg_cip...2]/CLR	2.214	0.281	1.933	-∞	inp
Path 126	∞	2	24	reset	reg_cip...5]/CLR	2.214	0.281	1.933	-∞	inp
Path 124	∞	2	24	reset	reg_pla...[0]/CLR	2.116	0.281	1.834	-∞	inp

Timing Summary - timing_2

2.6 Utilization



Resource	Utilization	Available	Utilization %
LUT	82	53200	0.15
FF	24	106400	0.02
IO	99	200	49.50



3 Third Party Verification

A simple online S-DES calculator to verify the above outputs can be found here.