VHDL Implementation of Twofish Algorithm

Purnima Gehlot  
Mody Institute of Technology and Science  
Laxmangarh, Sikar  
Rajasthan, India  
purnima.gehlot20@gmail.com

Richa Sharma  
Mody Institute of Technology and Science  
Laxmangarh, Sikar  
Rajasthan, India  
sharma.r0707@gmail.com

S. R. Biradar  
Mody Institute of Technology and Science  
Laxmangarh, Sikar  
Rajasthan, India  
srbiradar@gmail.com

ABSTRACT
Every day hundreds and thousands of people interact electronically, whether it is through emails, e-commerce, etc. through internet. For sending sensitive messages over the internet, we need security. In this paper a security algorithms, Twofish (Symmetric key cryptographic algorithm) [1] has been explained. All the important modules of Twofish algorithm, which are Function F and g, MDS, PHT, are implemented on Xilinx – 6.1 xst software and there delay calculations has been done on FPGA families which are Spartan2, Spartan2E and VirtexE.

Keywords
Twofish, MDS, PHT, symmetric key, Function F and g.

1. INTRODUCTION
Twofish is a 128-bit block cipher and variable-length key up to 256 bits are acceptable in it. There is a 16-round Feistel network with a function F made up of four key-dependent 8-by-8-bit S-boxes [3], a fixed 4-by-4 maximum distance separable (MDS) matrix over GF(2^8), a pseudo-Hadamard transform (PHT), bitwise rotations, and a carefully designed key schedule. It is a symmetric key algorithm which is also called as private key algorithm because only one key used for both the encryption and decryption purpose.

Objective of the dissertation is to perform an efficient method of implementing a twofish algorithm with minimum delay and having high performance in terms of delay while maintaining the proper functionality of the system. The software used for the implementation of the algorithm is Xilinx 6.1 – xst and language used is VHDL (very high speed integrated circuit hardware description language). Simulation of encryption process of the twofish algorithm has been done using the Xilinx software. Inputs will be converted into binary form and given as input to the "Model-Sim Simulator" of Xilinx 6.1 xst. Structure of Twofish algorithm is shown in Figure 1.

The organisation of paper is like this, First of all the description of all the modules of algorithms with appropriate diagram has been given, then the results of delay has been shown.

2. DIFFERENT MODULES OF TWOFISH ALGORITHM
Mainly there are three main steps in this algorithm, input whitening, sixteen iterations of round function and output whitening. Input is combined mathematically with the keys (key is divided into bytes and XORed with the input bytes). This process was input whitening. After 16 rounds same process happens in output whitening. During the rounds, function F is the main module for it, which consists of two modules of function g, one MDS i.e. maximum distance separable matrix, a PHT i.e. pseudo hadamard transform and two adders of 32-bit for one round. To increase the complexity of the algorithm we can perform Endian function over the input bit stream. Different modules of twofish algorithms are:

2.1 ENDIAN FUNCTION
Endian Function is a transformation of the input data. It is used as an interface between the input data provided to the circuit and the rest of the cipher. It can be used with all the key-sizes [5]. Here 128-bit input is divided into 16 bytes from byte0 to byte15 and are rearranged to get the output of 128-bit.

Figure 1: Twofish Structure
PHT is a reversible transformation of a bit string that provides cryptographic diffusion. SAFER was the first cipher block which uses 8-bit PHTs extensively for diffusion [6]. Twofish uses a 32-bit PHT. The outputs from the two g functions are given as input to the PHT module, it performs some addition operations over it and produces output. It is a two input, two output function.

Given two inputs, a and b, the 32-bit PHT is:

\[ a' = a + b \mod 2^{32} \]
\[ b' = a + 2b \mod 2^{32} \]

### 2.3 MDS (MAXIMUM DISTANCE SEPARABLE)

A 4-by-4 MDS matrix is used in twofish algorithm. MDS matrices are useful building blocks for ciphers because they guarantee a certain degree of diffusion. There are only three coefficients in MDS matrix, 01, EF and 5B [4]. The result of a multiplication can be reduced to a series of XOR’s for each bit of the output.

\[ Z = [MDS] [Y] \]

### TABLE 1. DELAY AND FREQUENCY OF DIFFERENT MODULES

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Delay (ns)</th>
<th>Frequency (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endian</td>
<td>6.479</td>
<td>154.34</td>
</tr>
<tr>
<td>Function F</td>
<td>104.102</td>
<td>9.60</td>
</tr>
<tr>
<td>Function g</td>
<td>39.383</td>
<td>25.39</td>
</tr>
<tr>
<td>MDS</td>
<td>15.520</td>
<td>62.81</td>
</tr>
<tr>
<td>PHT</td>
<td>61.172</td>
<td>16.34</td>
</tr>
<tr>
<td>Encryption</td>
<td>105.794</td>
<td>9.40</td>
</tr>
</tbody>
</table>

The MDS matrix is given as:

\[
\begin{pmatrix}
    z_0 \\
    z_1 \\
    z_2 \\
    z_3
\end{pmatrix} =
\begin{pmatrix}
    01 & EF & 5B & 5B \\
    5B & EF & 01 & EF \\
    EF & 5B & 01 & EF \\
    EF & 01 & EF & 5B
\end{pmatrix}
\]

\[
y_0 \\
y_1 \\
y_2 \\
y_3
\]

### 2.4 FUNCTION g

It consists of two main elements, the key-dependent S-boxes and the MDS matrix. Input is splitted into 4-byte, processed and merged back into a 32-bit stream. Output from S-boxes are given to the MDS matrix, where it is multiplied with the fix matrix and we get the output.

### 2.5 FUNCTION F

Function F is the main operation of the algorithm. It consists of four key-dependent 8-by-8-bit S-boxes, a fixed 4-by-4 maximum distance separable matrix, a pseudo-Hadamard transform, bitwise rotations and a key schedule. The operations of this function can be explained easily by:

\[ T_0 = g(\text{R}_0) \]
\[ T_1 = g(\text{ROL}(\text{R}_1, 8)) \]
\[ F_0 = (T_0 + T_1 + K_{2t, 18}) \mod 32 \]
\[ F_1 = (T_0 + 2T_1 + K_{2t, 9}) \mod 32 \]

### 2.6 ENCRYPTION

Encryption is the process of converting the data from plain-text to cipher [2]. Here in this paper encryption of 128-bit input is considered for one round. The output of encryption will be given for output whitening and we will get the cipher-text as output.

### 3. RESULTS

The delay table have been generated as result is shown in Table 1. The interface and RTL for Encryption is shown in Figure 2.

![Figure 2: RTL for Encryption](image)

### 6. REFERENCES


