VLSI Accelerator for Implementing Different Encryption/Decryption Algorithm for Security Processor

Group Members: CHIU Yeuk Shan, FUNG Ka Ho, NG Sze Ka

Project Code: TCY3b-06
Date: 18 April 2007

Project Supervisor: Dr. TSUI Chi-Ying
In this project, we designed a Very-Large-Scale Integration (VLSI) Accelerator that implements an Advanced Encryption Standard (AES) algorithm for security processor.

Objectives

- To provide a rapid processor for transferring secret data
- To minimize the area and delay time of the circuit

AES is divided into two parts. They are Encryption and Decryption.

- Encryption: To convert data into an unintelligible form, which is called ciphertext.
- Decryption: To convert the ciphertext back into its original form, which is called plaintext.
Encryption consists of four procedures which are AddRoundKey, SubBytes, ShiftRow and MixColumn.

A. AddRoundKey: each byte of the data in each round is combined with the round key that is derived from a key schedule.

B. SubBytes: a non-linear substitution step where each byte is replaced by another byte by checking look-up table (S-box).

C. ShiftRows: a transposition step where each row is shifted cyclically a certain number of steps.

D. MixColumns: a mixing operation which operates on the columns of the state, combining the four bytes in each column using a linear transformation.

Decryption consists of four procedures which are AddRoundKey, InvSubBytes, InvMixColumn and InvShiftRow.

A. AddRoundKey and InvShiftRow are the same as in Encryption.

B. InvSubBytes is inverse process of SubBytes in Encryption.

C. InvMixColumn is the inverse process of MixColumn in Encryption.
### Encryption

<table>
<thead>
<tr>
<th>128-bit Plaintext input (Hex)</th>
<th>128-bit Ciphertext output (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001122344556778899aabbccddeeff</td>
<td>69c4e0d86a7b0430d8cdb78070b4c55a</td>
</tr>
<tr>
<td>123456789abcdef123456789abcdefeff</td>
<td>6317d2b23649763aef21dfe1c156aa1</td>
</tr>
</tbody>
</table>

**Key:** 000102030405060708090a0b0c0e0d0e0f

<table>
<thead>
<tr>
<th>Area ($\lambda^2$)</th>
<th>Critical path delay (ps)</th>
<th>Clock period (ps)</th>
<th>Clock cycle</th>
<th>Extracted result time (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2800000</td>
<td>6491</td>
<td>12000</td>
<td>40</td>
<td>480000</td>
</tr>
</tbody>
</table>

### Decryption

<table>
<thead>
<tr>
<th>128-bit Ciphertext input (Hex)</th>
<th>128-bit Plaintext output (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>69c4e0d86a7b0430d8cdb78070b4c55a</td>
<td>0001122344556778899aabbccddeeff</td>
</tr>
<tr>
<td>6317d2b23649763aef21dfe1c156aa1</td>
<td>123456789abcdef123456789abcdefeff</td>
</tr>
</tbody>
</table>

**Key:** 2b7e151628aed2a6abf7158809cf4f3c

<table>
<thead>
<tr>
<th>Area ($\lambda^2$)</th>
<th>Critical path delay (ps)</th>
<th>Clock period (ps)</th>
<th>Clock cycle</th>
<th>Extracted result time (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3100000</td>
<td>6324</td>
<td>12000</td>
<td>77</td>
<td>924000</td>
</tr>
</tbody>
</table>