TCY3-06 VLSI Accelerator for Implementing Different Encryption/Decryption Algorithm for Security Processor

Name of students:  
Yau Lik Hang (04691297)  
Man Kwok Ho (04691792)  
Szeto Man Wa (04691560)  

Supervisor’s Name:  
Professor Tsui, Chi Ying
Introduction

The growing number of instances of breaches in information security in the last few years has shown an increasing demand for devices to perform security processing to protect private user data, like ID number or visa card number, transmitted over open channels. However, security processing presents a challenging workload for several electronic systems.

Therefore, hardware accelerators have been provided with the programmable processor in order to satisfy the system requirement.

Low-end systems:
- PDAs, wireless handsets, networked sensors, and smart cards.

High-end systems:
- routers, gateways, firewalls, storage servers, and web servers

Aim:

Our aim is to use the Advanced Encryption Standard (AES) algorithm to design a fast security processor to protect the data during transmission using VLSI technology. 128 bits key will be used to encrypt or decrypt the data.
Implementation

**System Block diagram**

**SubBytes** - This step is intended to provide adequate resistance from differential and linear cryptanalysis attacks.

**ShiftRows** - This step causes diffusion of the bits over multiple rounds.

**MixColumns** - The matrix obtained from the last step is multiplied with a standard matrix (in hexadecimal) to produce an output matrix.

**AddRoundKey** - Each byte of the state is combined with the round key.
## Measurement

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock speed (MHz)</td>
<td>100</td>
</tr>
<tr>
<td>Area (µm²)</td>
<td>348046</td>
</tr>
<tr>
<td>Total cell area</td>
<td>348049</td>
</tr>
<tr>
<td>Power</td>
<td>192</td>
</tr>
<tr>
<td>Total dynamic power (mW)</td>
<td></td>
</tr>
<tr>
<td>Cell leakage power (µW)</td>
<td>1.08</td>
</tr>
</tbody>
</table>

### Plaintext data (128 bits stream):

Input = 32 43 f6 a8 88 5a 30 8d 31 98 a2 e0 37 07 34

Key (128 bits stream):

Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab 17 15 88 09 cf 4f 3c

Ciphertext data (128 bits stream):

Output = 39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32