IDEA Encryption

Hardware encryption today, for tomorrow's telecommunications needs!

Group P

Stanton Champion
Harris Schneiderman
Randy Letness
Sam Poutasse
Overview

- What is IDEA?
- Logic and System Timing
- Chip Components
- Magic Layout
- System Performance Analysis
What is IDEA?

- IDEA was invented by Xuejia Lai and James Massey in 1990 (first called PES)
- IDEA is a block cipher encryption algorithm
  - Takes a key and a block of plaintext bits, and then outputs "gibberish"
  - Can be reversed by using same algorithm and key
  - Normal block and key lengths are 64 and 128 bits
  - Our chip uses a 16 bit block and 32 bit key
  - Easily scaled up or down in size
Why IDEA?

• Used in real world because of large key
  – With $10$ trillion dollars, it would take $10^{11}$ years to brute force 128 bits
  – With the computational ability of short-eared, it would take 1.8 days to brute force 32 bits

• Fairly fast

• Thought to be most secure algorithm available today

• Widely used in things like PGP, SSH, and SSL
Algorithm Flow

Key #1
Message Block → Round 1: IDEA → Encrypted Block #1

Key #2
Round 2: IDEA → Encrypted Block #2

Key #n
Round n: IDEA → Encrypted Block #n

Output Transformation

Encrypted Message
Algorithm Operations

- 14 step algorithm
- 4-bit Addition modulo $2^4$
- 4-bit XOR
- 4-bit Multiplication modulo $2^4+1$
  - multiply and divide
  - zero sub-block is considered to represent $2^4$
  - thus, the multiplicative inverse of 0 is 0
- left-rotate 1 bit of 4-bit operand
IDEA Simulator

4-bit IDEA Simulator

Input

Output

Encrypt

Decrypt

Key

00110101111010110001110001001

Rounds

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Copy To Input

Input Text

Output Text

Convert To Binary
Logic and System Timing

- 3 – 4 Step Cycle
- 2 input ALU ops (add, mod multiply, xor)
- 1 input ALU ops (move, left rotate)
- Standard V\(_{\text{BS}}\)\(_{\text{A}}\) signals
- Special Q\(_{\text{A}}\) signals
- Asynchronous Reset
Round Counter

- Keeps track of current round number
- Enable Signal (count_en)
- Asynchronous reset
- Auto-decrement after each round
Round Counter
State Machine Controller

- Connection with round counter
- Input and output handling
- 3 Critical Branches
- Control of register file
- Control of ALU
- Control of MUX
State Machine
Arithmetic Logic Unit

- Input and output registers (A, B, C)
- Mov and rot operations
- Add/XOR unit
- Mult unit
- Output MUX
Arithmetic Logic Unit Block Diagram
Register File

- 4 bit input/output, 18 rows
- Data and control passes through length and width of register
- 5->18 decoder selects rows
- IO pin determines input or output (makes decoder 6->36)
One Bit Latch
### Four Bit Latch

<table>
<thead>
<tr>
<th>reg_cont</th>
<th>latch3</th>
<th>latch3</th>
<th>latch3</th>
<th>latch3</th>
</tr>
</thead>
<tbody>
<tr>
<td>reg_cont_0</td>
<td>latch3_0</td>
<td>latch3_1</td>
<td>latch3_2</td>
<td>latch3_3</td>
</tr>
</tbody>
</table>
Register File
(18 4-bit Registers)

<table>
<thead>
<tr>
<th>register1_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>register1_1</td>
</tr>
<tr>
<td>register1_2</td>
</tr>
<tr>
<td>register1_3</td>
</tr>
<tr>
<td>register1_4</td>
</tr>
<tr>
<td>register1_5</td>
</tr>
<tr>
<td>register1_6</td>
</tr>
<tr>
<td>register1_7</td>
</tr>
<tr>
<td>register1_8</td>
</tr>
<tr>
<td>register1_9</td>
</tr>
<tr>
<td>register1_10</td>
</tr>
<tr>
<td>register1_11</td>
</tr>
<tr>
<td>register1_12</td>
</tr>
<tr>
<td>register1_13</td>
</tr>
<tr>
<td>register1_14</td>
</tr>
<tr>
<td>register1_15</td>
</tr>
<tr>
<td>register1_16</td>
</tr>
<tr>
<td>register1_17</td>
</tr>
</tbody>
</table>
System Performance Analysis

• Longest Path: ALU input through Multiplier to ALU output determined using Crystal

• Spice Analysis of longest path for rise time and fall time
• Fall Time $\sim$ 30ns
- Rise Time ~ 40ns
- Maximum Clock Frequency ~ 1/40ns ~ 25MHz
Full Chip Layout
(without routing)
Full Chip Layout
(without routing)