Comparison of the Hardware Performance of the AES Candidates Using Reconfigurable Hardware

Master’s Thesis

Pawel Chodowiec
MS CpE Candidate, ECE
George Mason University

Advisor: Dr. Kris Gaj, ECE
George Mason University
Outline

- Introduction to AES contest
- Introduction to FPGAs and Hardware Implementations
- Modes of Operation
- Implementations in Basic Iterative Architecture
- Implementations in Pipelined Architecture
- Summary
Most Popular Secret-Key Ciphers

- DES (1977-1990): 56 bit key
- Triple DES (1999-2001)
- AES-contest (2001-2020)
- IDEA
- Blowfish
- RC5
- CAST
- Serpent
- Twofish
- RC6
- Mars
Deep Crack

Electronic Frontier Foundation, 1998

Total cost: $220,000

Average time of search: 4.5 days/key

1800 ASIC chips, 40 MHz clock
AES Contest - NIST Evaluation Criteria

- Security
- Software Efficiency
- Hardware Efficiency
- Flexibility
AES Contest Effort

June 1998

15 Candidates
from USA, Canada, Belgium, France, Germany, Norway, UK, Isreal, Korea, Japan, Australia, Costa Rica

Round 1
Security
Software efficiency

August 1999

5 final candidates
Mars, RC6, Rijndael, Serpent, Twofish

Round 2
Security
Hardware efficiency

October 2000

1 winner: Rijndael
Belgium
Which way to go?

**ASICs**
- High performance
- Low power
- Low cost (but only in high volumes)

**FPGAs**
- Off-the-shelf
- Low development costs
- Short time to the market
- Reconfigurability

Low power and low cost (but only in high volumes) are advantages of ASICs. FPGAs, on the other hand, offer off-the-shelf availability, low development costs, short time to the market, and reconfigurability.
Reconfigurability

External ROM and microprocessor enable changing an FPGA function in several milliseconds

Encryption vs. decryption vs. key scheduling

Various algorithms
Target FPGA devices

Xilinx Virtex - XCV 1000

• 0.22 μm CMOS process
• 12 288 CLB slices
• 32 4-kbit block RAMs
• 1 mln equivalent logic gates
• Up to 200 MHz clock

Configurable Logic Block slices (CLB slices)

Programmable Interconnects

Block RAMs
Methodology and Tools

Implementation

2. Synthesis and Implementation

Code in VHDL

Verification

1. Functional simulation
   Aldec, Active-HDL

Xilinx, Foundation Series v. 2.1

Netlist with timing

3. Timing simulation
   Aldec, Active-HDL

Bitstream

4. Experimental Testing
   USC-ISI, SLAAC-1V FPGA board
Top level block diagram

control

Control unit

input interface

encryption/decryption

output interface

output

key scheduling

memory of internal keys
Primary factor in choosing the encryption/decryption unit architecture

Symmetric-key cipher mode of operation:

1. Non-feedback cipher modes
   - ECB, counter mode

2. Feedback cipher modes
   - CBC, CFB, OFB
Non-feedback Counter Mode - CTR

\[ C_i = M_i \oplus \text{AES}(\text{IV}+i) \quad \text{for } i=0..N \]
Feedback cipher modes - CBC

\[ C_1 = \text{AES}(M_i \oplus IV) \]
\[ C_i = \text{AES}(M_i \oplus C_{i-1}) \quad \text{for } i=2..N \]
Feedback cipher modes
CBC, CFB, OFB
Typical Structure of a Block Cipher

Round Key[0] → Initial transformation

i := 1

Cipher Round

Round Key[i] → i < #rounds?

i := i + 1

#rounds times

Round Key[#rounds+1] → Final transformation
Basic iterative architecture

one round
Architectures suitable for feedback modes

register -> MUX -> combinational logic -> one round

round 1
round 2
.....
round \#rounds

MUX

round 1
round 2
.....
round \#rounds
Partial Loop Unrolling

K rounds

combinational logic

multiplexer

register

round 1

round 2

......

round K
Loop Unrolling: Speed vs. Area

Throughput

- basic architecture
- loop unrolling
- resource sharing

basic architecture
resource sharing
loop-unrolling

Area
MARS - General Structure

plaintext

\[ + \]

forward mixing

\[ \text{subkey} \]

keyed forward transformation

\[ \text{subkeys} \]

keyed backwards transformation

\[ \text{subkeys} \]

backwards mixing

\[ - \]

\[ \text{subkey} \]

ciphertext
MARS - Keyed Transformation

D3 from mix transf. \[\rightarrow\] D3 loop

D2 from mix transf. \[\rightarrow\] D2 loop

D1 from mix transf. \[\rightarrow\] D1 loop

D0 from mix transf. \[\rightarrow\] D0 loop

optional swap

optional rotation to the right

128-bit register

optional swap

Keyed transformation core

D3 loop \[\rightarrow\] D3 loop

D2 loop \[\rightarrow\] D2 loop

D1 loop \[\rightarrow\] D1 loop

D0 loop \[\rightarrow\] D0 loop
MARS - Keyed Transformation Core
MARS – E-function
RC6 Round

R[

feedback to R_0

feedback to R_1

feedback to R_2

feedback to R_3

S[2i]

S[2i+1]

F

\[ \begin{align*}
    R_0 & \rightarrow d \rightarrow e \\
    R_1 & \rightarrow d \rightarrow e \\
    R_2 & \rightarrow d \rightarrow e \\
    R_3 & \rightarrow d \rightarrow e \\
\end{align*} \]

A

B

RC6 Round
Rijndael – Different Circuits for Encryption and Decryption

plaintext

-> ByteSub

-> ShiftRow

-> MixColumn

-> ciphertext

subkey

-> InvByteSub

-> InvShiftRow

-> InvMixColumn

-> ciphertext

subkey

-> plaintext
Rijndael – Resource Sharing Between Encryption and Decryption

- Inversed element in Galois field
- Inversed affine transformation
- Affine transformation
- ShiftRow
- MixColumn
- InvShiftRow
- InvMixColumn
- Subkey
First basic architecture of Serpent - Serpent I1

128-bit register

32 x S-box 0

32 x S-box 1

8-to-1 128-bit multiplexer

linear transformation

output
Alternative basic architecture of Serpent: Serpent I8

128-bit register

round 0

32 x S-box 0
linear transformation

round 7

32 x S-box 7
linear transformation

one implementation round of Serpent = 8 regular cipher rounds
Twofish – Encryption/Decryption Round

F - function

<<<1

>>>1
Twofish – F-function

![Diagram of Twofish F-function](image-url)
My Results: Basic architecture - Speed

Throughput [Mbit/s]

- Serpent: 431 Mbit/s
- Rijndael: 414 Mbit/s
- Twofish: 177 Mbit/s
- RC6: 142 Mbit/s
- Mars: 61 Mbit/s
- 3DES: 59 Mbit/s
My Results: Basic architecture - Area

Area [CLB slices]

- Twofish: 1076
- RC6: 1137
- Rijndael: 2507
- Mars: 2744
- Serpent: 4507
- 3DES: 356
Comparison with results of other groups: Speed

Throughput [Mbit/s]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Worcester Polytechnic Institute</th>
<th>University of Southern California</th>
<th>My Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serpent I8</td>
<td>431</td>
<td>444</td>
<td>414</td>
</tr>
<tr>
<td>Rijndael</td>
<td>414</td>
<td>353</td>
<td>294</td>
</tr>
<tr>
<td>Twofish</td>
<td>177</td>
<td>173</td>
<td>104</td>
</tr>
<tr>
<td>Serpent I1</td>
<td>149</td>
<td>143</td>
<td>62</td>
</tr>
<tr>
<td>RC6</td>
<td>112</td>
<td>88</td>
<td>61</td>
</tr>
<tr>
<td>Mars</td>
<td>61</td>
<td>102</td>
<td></td>
</tr>
</tbody>
</table>
Comparison with results of other groups: Area

Area [CLB slices]

- **Our Results**
- University of Southern California
- Worcester Polytechnic Institute

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Worcester Polytechnic Institute</th>
<th>University of Southern California</th>
<th>Our Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serpent I1</td>
<td>2638</td>
<td>1250</td>
<td>5511</td>
</tr>
<tr>
<td>Rijndael</td>
<td>4312</td>
<td>2507</td>
<td>4621</td>
</tr>
<tr>
<td>Mars</td>
<td>4621</td>
<td>3528</td>
<td>4621</td>
</tr>
<tr>
<td>Serpent I8</td>
<td>4507</td>
<td>2744</td>
<td>7964</td>
</tr>
</tbody>
</table>

- **Rijndael**
- **Mars**
- **Serpent I8**

### Comparison
- **Serpent I8** has the highest Area [CLB slices] at 7964.
- **Serpent I1** has the lowest Area [CLB slices] at 5511.
My Results: Encryption in cipher feedback modes (CBC, CFB, OFB) - Virtex FPGA

Throughput [Mbit/s] vs. Area [CLB slices]

- Rijndael
- Serpent I8
- Mars
- Twofish
- RC6
- Serpent I1
NSA Results: Encryption in cipher feedback modes (CBC, CFB, OFB) - ASIC, 0.5 µm CMOS

Throughput [Mbit/s] vs. Area [mm²]

- Serpent I1
- RC6
- Twofish
- Rijndael
- Mars
Conclusions for feedback cipher modes (1) (CBC, CFB, OFB)

• **Speed** (throughput) should be the primary criteria of comparison

• **Basic iterative architecture** is the most appropriate for comparison and future implementations

• **Serpent** and **Rijndael** are over twice as fast as the next best candidate for all implementations
Conclusions for feedback cipher modes (2) (CBC, CFB, OFB)

• Results confirmed by
  - three independent university groups for FPGAs, and
  - NSA group for ASICs

• Results of comparison independent of implementation technology (FPGAs vs. ASICs)
Non-Feedback Cipher Modes
ECB, counter
Comparison for non-feedback cipher modes, e.g. Counter Mode - CTR

\[ C_i = M_i \oplus AES(IV+i) \quad \text{for } i=0..N \]
NSA approach: Traditional methodology

- One round, no pipelining
- \#rounds registers
  - Round 1 = one pipeline stage
  - Round 2 = one pipeline stage
  - Round \#rounds = one pipeline stage
My approach: New methodology

a) register
   one round, no pipelining

b) $k$ registers
   one round
   = $k$ pipeline stages

\[
\text{round 1} = k \text{ pipeline stages} \\
\text{round 2} = k \text{ pipeline stages} \\
\text{round } \text{#rounds} = k \text{ pipeline stages}
\]

c) $K \cdot k$ registers

\[
\text{round 1} = k \text{ pipeline stages} \\
\text{round 2} = k \text{ pipeline stages} \\
\text{round } K = k \text{ pipeline stages}
\]

d) $\text{#rounds} \cdot k$ registers

\[
\text{round 1} = k \text{ pipeline stages} \\
\text{round 2} = k \text{ pipeline stages} \\
\text{round #rounds} = k \text{ pipeline stages}
\]
My approach: Inner-Round Pipelining
Comparison of the traditional and new design methodologies (1)

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Throughput</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inner-round pipelining</td>
<td>K=2</td>
<td></td>
</tr>
<tr>
<td>inner-round pipelining</td>
<td>K=3</td>
<td></td>
</tr>
<tr>
<td>mixed inner and outer-round pipelining</td>
<td>K=2</td>
<td></td>
</tr>
<tr>
<td>mixed inner and outer-round pipelining</td>
<td>K=3</td>
<td></td>
</tr>
<tr>
<td>mixed inner and outer-round pipelining</td>
<td>K=4</td>
<td></td>
</tr>
</tbody>
</table>

$k_{opt}$ represents the optimal value for each methodology.
Comparison of the traditional and new design methodologies (2)

Latency

- basic architecture
- inner-round pipelining
- mixed inner and outer-round pipelining
- outer-round pipelining

Area

inner-round pipelining
mixed inner and outer-round pipelining
basic architecture
outer-round pipelining
NSA architecture: Full outer-round pipelining

#rounds registers

round 1
= one pipeline stage

round 2
= one pipeline stage

...  

round #rounds
= one pipeline stage

Total # of pipeline stages = #rounds
NSA Results: Full outer-round pipelining

CMOS ASIC 0.5 μm

Throughput [Gbit/s]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Throughput [Gbit/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serpent</td>
<td>8.0</td>
</tr>
<tr>
<td>Rijndael</td>
<td>5.7</td>
</tr>
<tr>
<td>Twofish</td>
<td>2.3</td>
</tr>
<tr>
<td>RC6</td>
<td>2.2</td>
</tr>
<tr>
<td>Mars</td>
<td>2.2</td>
</tr>
</tbody>
</table>
My approach:
Full mixed inner- and outer-round pipelining

$k$ registers

round 1
$= k$ pipeline stages

round 2
$= k$ pipeline stages

round #rounds
$= k$ pipeline stages

Total # of pipeline stages $= \#\text{rounds} \cdot k$
My Results: Full mixed pipelining

Virtex FPGA

Throughput [Gbit/s]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Throughput [Gbit/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serpent</td>
<td>16.8</td>
</tr>
<tr>
<td>Twofish</td>
<td>15.2</td>
</tr>
<tr>
<td>RC6</td>
<td>13.1</td>
</tr>
<tr>
<td>Rijndael</td>
<td>12.2</td>
</tr>
</tbody>
</table>
Speed-up compared to the basic architecture

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Our results</th>
<th>NSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rijndael</td>
<td>29.5</td>
<td>9.5</td>
</tr>
<tr>
<td>Serpent I8</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Serpent I1</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Twofish</td>
<td>86</td>
<td>21.5</td>
</tr>
<tr>
<td>RC6</td>
<td>91.5</td>
<td>21</td>
</tr>
<tr>
<td>Mars</td>
<td>38</td>
<td></td>
</tr>
</tbody>
</table>
My approach: Full Mixed Pipelining

Speed = \frac{\text{block size}}{\text{min}\_\text{clock}\_\text{period}}
My Results: Full mixed pipelining

Area [CLB slices]

- Serpent: 19,700 CLB slices
- Twofish: 21,000 CLB slices
- RC6: 46,900 CLB slices
- Rijndael: 12,600 CLB slices

Legend:
- dedicated memory blocks, RAMs
- 80 RAMs
Conclusions for non-feedback cipher modes (1)

ECB, counter

• All ciphers can achieve approximately the same speed. Area should be the primary criteria of comparison.

• Architecture with inner round pipelining combined with full outer round pipelining is the most appropriate for comparison and future implementations.

• Serpent, Twofish and Rijndael are the most cost-efficient and take approximately the same amount of area.
Conclusions for non-feedback cipher modes (2)

ECB, counter

• **No agreement** regarding the methodology and architecture used for comparison

• **NSA methodology favored** ciphers with
  • simple (fast) cipher round
    (Serpent and Rijndael)

• **My methodology**
  • fair
  • **practical** (superior throughput/area ratio)
Summary (1)

All five AES finalists implemented in basic architecture
  • The best throughput for four of them
  • The best throughput/area ratio

Four AES finalists implemented in mixed architecture
  • The highest throughput ever reported
Summary (2)

• **Basic iterative architecture** is the most appropriate for comparison and future implementations of ciphers in **feedback modes of operation**
  - Gives best throughput/area ratio
  - Throughput is the comparison criteria

• **Mixed architecture** is the most appropriate for comparison and future implementations of ciphers in **non-feedback modes of operation**
  - Area is the comparison criteria
  - Independent of number and complexity of rounds
Summary (3) - My Ranking of Ciphers

- **Serpent, Rijndael**
  - Rijndael offers better throughput/area ratio

- **RC6, Twofish**
  - Both ciphers slower in basic architecture

- **MARS**
  - Slow and large
Summary (4) - Survey filled by AES3 conference participants

# votes

0 10 20 30 40 50 60 70 80 90 100

Rijndael Serpent Twofish RC6 Mars
Possible Extensions of this Work

• Study implementations of key schedules
  • Forward and backward key schedules

• Study new modes of operation
  • New contest started by NIST

• Comparison of Stream and Block Ciphers
Questions....