A High-Performance Area-Efficient AES Cipher on a Many-Core Platform

Bin Liu and Bevan M. Baas
Department of Electrical and Computer Engineering, University of California, Davis, CA 95616

Advanced Encryption Standard

- SubBytes: byte substitution from a look up table
- MixColumns: each column multiplies a fixed polynomial over GF(2^8)
- ShiftRows: cyclically shift by one, two and three bytes in the 2nd, 3rd and 4th row
- AddRoundKey: round key is added to byte blocks using a bitwise XOR operation

<table>
<thead>
<tr>
<th>KeySubWord</th>
<th>KeyRotWord</th>
<th>KeyXOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kc, K3, K4, K5</td>
<td>Kc, K3, K4, K5</td>
<td>Kc, K3, K4, K5</td>
</tr>
</tbody>
</table>

Targeted Fine-Grained Many-Core Platform

- AsAP2 Single Tile (164 total)
  - Area: 0.17 mm²
  - Transistors: 325,000
  - CMOS Tech.: 65nm low-leakage
  - Max. frequency: 1.2GHz @ 1.3 V
  - Data Memory: 128 x 16-bit
  - Instru. Memory: 128 x 32-bit

  - Throughput is 43% higher (9.5 cycles per block)
  - 16% fewer cores required (59 cores)

Preliminary Design of AES Cipher

- (N-1) times loop unrolling
- Throughput is 266 cycles per datablock
- 70 cores are used

Optimization I: Increased Throughput

- Cores running MixColumns workloads are 2x slower than others, so we parallelize each into two MixCol-8 programs
- Throughput is increased by 43% (152 cycles per block)
- Increased parallelization requires 10 more cores
- MixCol-8 cores are now bottlenecks

Optimization II: Reduced Number of Cores

- Before optimization
  - 22% average lMem usage
  - 43% average dMem usage
- Core merging should not introduce new bottlenecks or exceed memory limitations

- Step I: Combine the neighboring SubBytes and ShiftRows into one SubShift core
- Step II: Combine the neighboring KeyRot and KeyXOR into one KeySche core
- \( T_{EXE\_SUBSHIFT} = 148 \) cycles per data block
- 80% lMem and 100% dMem usage
- 24% lMem and 28% dMem usage

Optimized Design of AES Cipher

Comparison with Related Work

<table>
<thead>
<tr>
<th>Platform</th>
<th>Method</th>
<th>Tech. (nm)</th>
<th>Area (mm²)</th>
<th>Max Freq. (MHz)</th>
<th>Throughput (cycles/byte)</th>
<th>Scaled Throughput (Mbps)</th>
<th>Scaled Area (mm²)</th>
<th>Scaled Throughput/Area (Mbps/mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pentium 4 561</td>
<td>Intel</td>
<td>90</td>
<td>112</td>
<td>3600</td>
<td>16</td>
<td>2492</td>
<td>58.42</td>
<td>42.66</td>
</tr>
<tr>
<td>Athlon 64 3500</td>
<td>AMD</td>
<td>90</td>
<td>193</td>
<td>2200</td>
<td>10.6</td>
<td>2299</td>
<td>101</td>
<td>22.76</td>
</tr>
<tr>
<td>Core 2 Duo E6400</td>
<td>Intel</td>
<td>65</td>
<td>111</td>
<td>2130</td>
<td>19.9</td>
<td>1854</td>
<td>111</td>
<td>16.70</td>
</tr>
<tr>
<td>Core 2 Quad E6800 (one core)</td>
<td>Intel</td>
<td>65</td>
<td>143</td>
<td>2400</td>
<td>9.32</td>
<td>2060</td>
<td>143</td>
<td>14.41</td>
</tr>
<tr>
<td>Core 2 Quad Q6950 (one core)</td>
<td>Intel</td>
<td>65</td>
<td>143</td>
<td>2830</td>
<td>7.59</td>
<td>2065</td>
<td>112</td>
<td>18.44</td>
</tr>
<tr>
<td>Core i7 920 (one core)</td>
<td>Intel</td>
<td>65</td>
<td>133</td>
<td>2668</td>
<td>6.92</td>
<td>2135</td>
<td>133</td>
<td>16.05</td>
</tr>
<tr>
<td>This Work AsAP</td>
<td>65</td>
<td></td>
<td>159</td>
<td>252</td>
<td>45.63</td>
<td>252</td>
<td>45.63</td>
<td>155.70</td>
</tr>
</tbody>
</table>

- Compared to CPUs, our design achieves 3.6-10.7x higher throughput per chip area
- Compared to DSP, our design achieves 1.5x higher throughput
- Compared to GPU, our design achieves 3.4x higher throughput per chip area

Core-Scaling on Many-Core Platforms

- Fine-grain many-core processor arrays with large numbers of cores enable application libraries with varying numbers of cores
- Optimum run-time programs chosen with joint “core scaling” and supply voltage and clock frequency scaling
- New possibilities to tradeoff number of processors, performance and energy efficiency

Acknowledgments: The authors gratefully acknowledge support from NSF Grant 0940090, IBM Members and CAREER Award 0549007. SRC Grant 1598 and 1791, Csr Grant 1656. UC Micro, ST Microelectronics, Intel, and IntelX. The authors also acknowledge the support of the C252 Focus Center, one of the six research centers funded under the Force Center Research Program (FCRP), a Semiconductor Research Corporation entity.