New Circuit Minimization Techniques for Smaller and Faster AES SBoxes

Alexander Maximov and Patrik Ekdahl
Ericsson Research
Preliminaries

AES Round Function

- SubBytes is the only non-linear part
- 16 8x8 SBoxes needed for a full implementation
- Forward only or combined SBox
- In ASICs
  - Look-up table
  - Gate implementation

What to remember:
- New improved methods for circuit minimization.
- New SBox architecture which improves the critical path.
Preliminaries

Basic flow of AES SBox

Direct implementation of inversion over Rijndael field is very complex.
Previous work (low area)

Rijmen [Rij00] proposed (based on Itoh and Tsujii [IT88]) to use a composite field and do the inversion in $\text{GF}(2^4)$ instead.

— Satoh et al [SMT01] reduced inversion to $\text{GF}(2^2)$.
— Canright [Can05] investigated the importance of subfield representation.
Previous work (low depth)

Boyar, Peralta et al ([BP10a,BP10b,BP12,BFP18]) used a normal base $A=a_0Y + a_1Y^{16}$ and $A^{-1} = (AA^{16})^{-1}A^{16}$ (also based on Itoh and Tsujii [IT88]) to derive another implementation.

Several papers followed:
— Nogami et al [NNT+10], looking at mixed bases.
— Ueno et al [UHS+15], looking at redundant bases.
— Reyhani et al [RMTA18a,b], improving Boyar-Peralta (BP) search algorithm.
— Li et al [LSL+19], incorporating depth into BP algorithm.
Previous work (low depth)

Boyar, Peralta et al ([BP10a,BP10b,BP12,BFP18]) used a normal base $A = a_0 Y + a_1 Y^{16}$ and $A^{-1} = (AA^{16})^{-1}A^{16}$ (also based on Itoh and Tsujii [IT88]) to derive another implementation.

Several papers followed:
- Nogami et al [NNT+10], looking at mixed bases.
- Ueno et al [UHS+15], looking at redundant bases.
- Reyhani et al [RMTA18a,b], improving Boyar-Peralta (BP) search algorithm.
- Li et al [LSL+19], incorporating depth into BP algorithm.
Architectural starting point [BP12]

Base conversion and generation of linear parts of multiplications

Input U 8 — 22 bit Q — Mul-Sum — 4 bit X — Inverse GF(2^4) — 4 bit Y — 2 x Mul — 18 bit N — Output R

Base back-conversion and the affine transformation of the AES SBox.

Basic problem statement:
Given a binary matrix $M_{mxn}$ and the maximum allowed depth $maxD$,
find the circuit of depth $D \leq maxD$ with the minimum number of 2-input XOR gates
such that it computes $Y = M \cdot X$.

$$
\begin{align*}
 y_0 &= x_0 + x_2 + x_3 + x_4 \\
 y_1 &= x_1 + x_2 + x_4 \\
 y_2 &= x_0 + x_1 + x_3 + x_4
\end{align*}
$$

$$
M = \begin{pmatrix}
1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1
\end{pmatrix}
$$

Additional Input Requirement (AIR)
- Input signals may arrive with different delay $d_i$

Additional Output Requirement (AOR)
- Output signals may need to be ready earlier, $e_i \leq maxD$
Our contributions

— New techniques for minimizing the Top and Bottom matrices (area with delay constraints).
  — Introduced a probabilistic heuristic approach to the cancellation-free algorithm by Paar [Paa97].
  — New cancellation-allowed exhaustive search algorithm, based on BP-algorithm [BP10a].

— Floating Multiplexers for the combined SBox.

— New generalization of BP-algorithm, allowing other types of gates.
  — New metrics, with lots of speed up tricks for the distance function.
  — Stack algorithm with a search tree.

— New architecture that removes the Bottom matrix and reduces the overall depth.

— New circuit for the inverse operation.

— Additional Transformation Matrices.
Combined SBox with multiplexers
Combined SBox with multiplexers

Example:

\[ Y^F = X_0 + X_1 \]
\[ Y^I = X_0 + X_2 \]
\[ Y = \text{MUX}(\text{select}, X_0 + X_1, X_0 + X_2) \]

Replace with:

\[ Y = \text{MUX}(\text{select}, X_1, X_2) + X_0 \]

Generally:

\[ Y = A + \text{MUX}(\text{select}, B, C) \rightarrow \]
\[ Y = A + \Delta + \text{MUX}(\text{select}, B + \Delta, C + \Delta) \]
Boyar-Peralta algorithm [BP10a]

— Notion of a “point”.  
  — In original algorithm, this is a linear combination of input signals. Set of gates used $G=\{\text{XOR}\}$.

— Base set of known points $S$.

— Set of target points $T$, the rows $y_i$ of $M$.

— Metric using a distance function $\delta_i(S, y_i)$.

— Set of candidates $C$.

• Try all base pair $s_i, s_j$ in $S_t$ and form a candidate $c = g(s_i, s_j)$, in this case: $c = s_i + s_j$
• Calculate the new distance vector $\Delta$ based on $S_t \cup c$
• We save the candidate $c$ that gives the lowest distance $S_{t+1} = S_t \cup c$
• Repeat until the distance vector is all-zero.

\[ S_0 = (x_0, x_1, \ldots, x_4) = ([1,0,0,0,0], [0,1,0,0,0], \ldots, [0,0,0,0,1]) \]

\[
\begin{pmatrix}
1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 \\
\end{pmatrix} = 
\begin{pmatrix}
y_0 \\
y_1 \\
y_2 \\
\end{pmatrix}
\]

$\Delta = (\delta_0, \delta_1, \ldots, \delta_{n-1})$. 

\[
\Delta = (\delta_0, \delta_1, \ldots, \delta_{n-1}).
\]
BP for Linear Circuits with Floating Multiplexers

— Include MUX, NMUX in the set of gates.

— A **point** is now a tuple \( p = (F, I) \)
  — \( F \) and \( I \) are linear combinations of input signals
  — Translated into \( MUX(ZF, F \cdot X, I \cdot X) \)

— Input points \( X_k = (2^k, 2^k), k = 0, ... n - 1 \)

— Target points \( Y_k = (Y_k^F, Y_k^I), k = 0, ..., m - 1 \)

— Improved metrics and new algorithm (with lots of speed up) to calculate \( \delta_i(S, y_i|D_{max}) \).

— We keep track of AIR, and AOR at each stage.

— For the full Affine transformation, define the point as \( p = (f, F, i, I) \rightarrow MUX(ZF, F \cdot X + f, I \cdot X + i) \)
BP for any Nonlinear Circuit

— Allow all kinds of gates in G (XOR, AND, MUX, ... 2-input, 3-input...).

— A point is now the truth table of a Boolean function.
  — Combine points using truth tables and gate functionality.

— Target points are the truth table for every output signal of the nonlinear block.

— Applicable to circuits of maximum 4-5 input signals, and the number of output signals is not limited.

— Used to derive a smaller inversion circuit over GF(2^4).
Search Tree

- $S_r$, $S_{r+1}$, $S_{r+2}$, $S_{r+3}$
- 20-50 children
- ~400 total children

- Try to keep leaves from as many different branches as possible
Search Tree

— Try to keep leaves from as many different branches as possible
New architecture for lower depths

The Bottom matrix only depends on the multiplication of the 4-bit signal $Y$ with some linear combination of the input signal $U$

$$R = Y_0 \cdot M_0 \cdot U + \cdots + Y_3 \cdot M_3 \cdot U$$

where $M_i$ is an 8x8 matrix to be scalar multiplied by the $Y_i$ bit.

Calculate $M_i$ in parallel in Top matrix.

Assembling requires 56 gates (32NAND, 24XOR)
New circuit for the inversion in $\text{GF}(2^4)$

\[
\begin{align*}
Y_0 &= X_1 X_2 X_3 + X_0 X_2 + X_1 X_2 + X_2 + X_3 \\
Y_1 &= X_0 X_2 X_3 + X_0 X_2 + X_1 X_2 + X_1 X_3 + X_3 \\
Y_2 &= X_0 X_1 X_3 + X_0 X_2 + X_0 X_3 + X_0 + X_1 \\
Y_3 &= X_0 X_1 X_2 + X_0 X_2 + X_0 X_3 + X_1 X_3 + X_1
\end{align*}
\]

— In [BP12] they found a circuit of 17 gates and depth 4 (with base gates \{AND, XOR\}).
— By applying the BP-algorithm for general non-linear circuits, we managed to achieve 9 gates and depth 3.

\[
\begin{align*}
T_0 &= \text{NAND}(X_0, X_2) & T_3 &= \text{MUX}(X_1, X_2, 1) & Y_1 &= \text{MUX}(T_2, X_3, T_3) \\
T_1 &= \text{NOR}(X_1, X_3) & T_4 &= \text{MUX}(X_3, X_0, 1) & Y_2 &= \text{MUX}(X_0, T_2, X_1) \\
T_2 &= \text{XNOR}(T_0, T_1) & Y_0 &= \text{MUX}(X_2, T_2, X_3) & Y_3 &= \text{MUX}(T_2, X_1, T_4)
\end{align*}
\]

We also found a small conventional (no MUXes) circuit of 15 gates and depth 3.
Additional Transformation Matrices

Excluding the final constant from the affine transformation, we can write the SBox as:

\[ SBox(x) = x^{-1} \cdot A_{8x8} \]

In any field of characteristic 2, squaring, square root, and multiplication by a constant are linear functions. Thus, for any choice of \( \alpha = 1 \ldots 255 \), and \( \beta = 0 \ldots 7 \) we have:

\[ Z(x) = \left( \alpha \cdot x^{2^\beta} \right)^{-1} \]  

Top matrix

\[ SBox(x) = \frac{2^\beta \sqrt{\alpha \cdot Z(x)}}{A_{8x8}} \]  

Bottom matrix

— For Forward (Inverse) we have 2040 choices. Tried all!
— For Combined we have \( 2040^2 = 4,161,600 \) choices. Based on the heuristic algorithm, we selected candidates to run the full generic floating multiplexer algorithm.

A similar approach was independently proposed in [UHNA19] but they only considered multiplication.
## Forward SBox Results

### Previous Results

<table>
<thead>
<tr>
<th>SBox</th>
<th>Area Size/Gates</th>
<th>Critical Path/Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Std. gates</td>
<td>Tech. GE</td>
</tr>
<tr>
<td>Canright [Can05]</td>
<td>80×8+34×4+6×1</td>
<td>19×2+3×2+1×1</td>
</tr>
<tr>
<td>most famous design</td>
<td>120</td>
<td>226.4</td>
</tr>
<tr>
<td>Boyar et al [BP12]</td>
<td>94×8+34×4</td>
<td>13×2+3×2</td>
</tr>
<tr>
<td>our starting point</td>
<td>128</td>
<td>264.2</td>
</tr>
<tr>
<td>Boyar et al [Boy]</td>
<td>81×8+32×3</td>
<td>21×2+6×2</td>
</tr>
<tr>
<td>record smallest</td>
<td>113</td>
<td>220.7</td>
</tr>
<tr>
<td>Ueno et al [UHS+15]</td>
<td>91×8+48×2+13×2</td>
<td>10×8+5×2+1×1</td>
</tr>
<tr>
<td>record fastest, formulas from RMTA18a</td>
<td>151(+4)</td>
<td>270.71</td>
</tr>
<tr>
<td>Reyhani-Light [RMRA18a]</td>
<td>69×8+43×3+7×2</td>
<td>16×8+4×2+1×1</td>
</tr>
<tr>
<td>at CHES 2018</td>
<td>119(+4)</td>
<td>213.45</td>
</tr>
<tr>
<td>Reyhani-Fast [RMRA18a]</td>
<td>79×8+43×3+7×2</td>
<td>11×8+5×2+1×1</td>
</tr>
<tr>
<td>at CHES 2018</td>
<td>129(+4)</td>
<td>236.75</td>
</tr>
<tr>
<td>Ueno et al [UHNA19]</td>
<td>90×8+6×2+10×2+4×2</td>
<td>11×8+10×2+3×2</td>
</tr>
<tr>
<td>recent result</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Our Results

<table>
<thead>
<tr>
<th>SBox</th>
<th>Area Size/Gates</th>
<th>Critical Path/Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Std. gates</td>
<td>Tech. GE</td>
</tr>
<tr>
<td>Forward (fast)</td>
<td>77×8+1×2×3+2×2×3+6×1×2</td>
<td>7×8+1×2×1+1×1×2+2×1×2</td>
</tr>
<tr>
<td>fast with depth 12</td>
<td>130</td>
<td>243.04</td>
</tr>
<tr>
<td>Forward (tradeoff)</td>
<td>61×8+5×2×2+7×1</td>
<td>8×8+2×1×2+1×1×2</td>
</tr>
<tr>
<td>area/speed tradeoff</td>
<td>111</td>
<td>216.75</td>
</tr>
<tr>
<td>Forward (bonus)</td>
<td>58×8+6×2×2+7×1</td>
<td>18×8+2×1×2+1×1×2</td>
</tr>
<tr>
<td>new record smallest</td>
<td>102</td>
<td>195.10</td>
</tr>
</tbody>
</table>

### Area vs Clock period

![Area vs Clock period graph](image-url)
# Combined SBox Results

## Combined SBox

<table>
<thead>
<tr>
<th></th>
<th>Area Size/Gates</th>
<th>Critical Path/Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Std. gates</td>
<td>Tech. GE</td>
</tr>
<tr>
<td><strong>Previous Results</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canright [Can05]</td>
<td>94X+34ND+6NR+16MX (+2IV)</td>
<td>20X+3ND+2OR+5NR</td>
</tr>
<tr>
<td>most famous design</td>
<td>150 (+2)</td>
<td>297.64</td>
</tr>
<tr>
<td>Reyhani et al [RMTA1b]</td>
<td>81X+32ND+4OR+15NR+16MI (+8IV)</td>
<td>17X+2ND+3OR+6NR</td>
</tr>
<tr>
<td></td>
<td>149 (+8)</td>
<td>290.13</td>
</tr>
<tr>
<td>Ueno et al [UHNA19]</td>
<td>112X+73N+10DR+45AN+16I (+10IV)</td>
<td>11X+3AN+10OR+2MX (+11IV)</td>
</tr>
<tr>
<td><strong>Our Results</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined (fast)</td>
<td>77X+27N+41ND+6NR+13MX+12MI</td>
<td>6X+3XN+1ND+2NR+1MX+1MI</td>
</tr>
<tr>
<td>fast with depth 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined (tradeoff)</td>
<td>70X+21N+27ND+5NR+17MX+5MI</td>
<td>7X+4XN+1ND+2NR+1MX+1MI</td>
</tr>
<tr>
<td>area/speed tradeoff</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined (bonus)</td>
<td>70X+9N+27ND+6NR+16MX</td>
<td>15X+4XN+2ND+1NR+3MX</td>
</tr>
<tr>
<td>new record smallest</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Graph showing area vs. clock period for different SBox configurations](image)

- **Our - fast**
- **Our - tradeoff**
- **Our - bonus**
- **Reyhani**
- **Ueno'19**
- **Canright**
Alexander also applied the algorithms to the AES MixColumns circuits

<table>
<thead>
<tr>
<th>Previous results (XORs):</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>Jean et al, CHES 2017</td>
</tr>
<tr>
<td>97</td>
<td>Krantz et al, ToSC 2017</td>
</tr>
<tr>
<td>95</td>
<td>Banik et al, ePrint Archive Report 2019/856</td>
</tr>
<tr>
<td>94</td>
<td>Tan and Peyrin, ePrint Archive Report 2019/847</td>
</tr>
<tr>
<td>Alexander’s result:</td>
<td></td>
</tr>
</tbody>
</table>
Thank you.

www.ericsson.com