Digital Signal Processing for Embedded Communications and Biomedical Systems

Keshab K. Parhi
Distinguished McKnight University Professor
University of Minnesota, Minneapolis

http://www.ece.umn.edu/users/parhi

May 23, 2012
OUTLINE

• Communications Systems
  - Folding
  - Polar Decoders

• Biomedical Systems
  - Communication
    - Feature Computation and Classification
    - Monitoring

• IC Chip Security by PUF
Wireless Phone Timeline

Folding Transformation

• 3rd-order IIR filter

• See Parhi, VLSI Digital Signal Processing Systems, Wiley, 1999

• A possible folding set:

\[ A = \{A_0, A_1, A_2, A_3\}, \quad M = \{M_0, M_1, M_2, M_3\} \]
Folding Transformation (Cont’d)

- Folded 3\textsuperscript{rd}-order IIR filter

- Multiple algorithm operations are time-multiplexed to a single functional unit

- Area reduction!
Folding Transformation (Cont’d)

- 6th-order IIR filter (cascade of two 3rd-order IIR filter)

- Also can be folded into 1 multiplier and 1 adder

- A possible folding set with interleaved ordering:

\[ A = \{A_0, A_0', A_1, A_1', A_2, A_2', A_3, A_3' \} \]

\[ M = \{M_0, M_0', M_1, M_1', M_2, M_2', M_3, M_3' \} \]
Folding Transformation (Cont’d)

- Folded 6th-order IIR filter

- More Pipelining -> Low-Power, High-Speed

- Hierarchical Folding Algorithm:
  \[ D \rightarrow 2D \]
  \[ \text{switch } i \rightarrow \text{switch } 2i, 2i+1 \]
Advances in Coding Theory

- Turbo Codes
- LDPC Codes
- Polar Codes (Most Recent)
What are polar codes?

- Arıkan introduced polar coding in his breakthrough paper.
- Polar codes have provably capacity-achieving capability.
- They are applicable in a diverse set of scenarios.


Plot from UCSD Web link
Successive Cancellation (SC) decoding

Successive Cancellation (SC) is one of the most popular decoding algorithms. It is suitable for VLSI implementation for the FFT-like structure.

\[
L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = (-1)^{\hat{u}_{2i-1}} L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}),
\]

Type I PE:

\[
L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = 2 \text{artanh}\{\tanh[L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})/2] \cdot \tanh[L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})/2]\}.
\]
SC decoding algorithm

Successive cancellation (SC) is one of the most popular decoding algorithms.

It is suitable for VLSI implementation for the FFT-like structure.

Type I PE:
\[ L_{N}^{(2i)} (y_1^N, \hat{u}_1^{2i-1}) = \frac{1}{2} \begin{cases} \hat{u}_1^{2i-1} & \text{if } i \leq 1 \\ \hat{u}_1^{2i-2} \oplus \hat{u}_1^{2i-2} & \text{if } i > 1 \end{cases} \]

Type II PE:
\[ L_{N}^{(2i-1)} (y_1^N, \hat{u}_1^{2i-1}) = \frac{1}{2} \begin{cases} \hat{u}_1^{2i-1} & \text{if } i \leq 1 \\ \hat{u}_1^{2i-2} \oplus \hat{u}_1^{2i-2} & \text{if } i > 1 \end{cases} \]
SC decoding algorithm

Successive cancellation (SC) is one of the most popular decoding algorithms.

It is suitable for VLSI implementation for the FFT-like structure.

Type I PE:

\[
L_{N}^{(2i)}(y_{1}^{N}, \hat{u}_{1}^{2i-1}) = (-1)^{\hat{u}_{2i-1}} L_{N/2}^{(i)}(y_{1}^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L_{N/2}^{(i)}(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}),
\]

Type II PE:

\[
L_{N}^{(2i-1)}(y_{1}^{N}, \hat{u}_{1}^{2i-1}) = 2 \text{artanh}\{\tanh[L_{N/2}^{(i)}(y_{1}^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})/2] \cdot \tanh[L_{N/2}^{(i)}(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2})/2]\}.
\]
Successive cancellation (SC) is one of the most popular decoding algorithms. It is suitable for VLSI implementation for the FFT-like structure.

**Type I PE:**
\[
L_N^{(2i)}(y_N, \hat{u}_1^{2i-1}) = (-1)^i \hat{u}_1^{2i-1} L_N^{(i)}(y_1^{N/2}, \hat{u}_1^{2i-2} \oplus \hat{u}_1^{2i-2}) + L_N^{(i)}(y_1^{N/2+1}, \hat{u}_1^{2i-2}),
\]

**Type II PE:**
\[
L_N^{(2i-1)}(y_N, \hat{u}_1^{2i-1}) = 2 \text{artanh}\{\tanh[L_N^{(i)}(y_1^{N/2}, \hat{u}_1^{2i-2} \oplus \hat{u}_1^{2i-2})/2] \cdot \tanh[L_N^{(i)}(y_1^{N/2+1}, \hat{u}_1^{2i-2})/2]\}.
\]
SC decoding algorithm

Successive cancellation (SC) is one of the most popular decoding algorithms.

It is suitable for VLSI implementation for the FFT-like structure.

Type I PE:
\[ L_N^{(2i)}(y_1, \hat{u}_1^{2i-1}) = (-1) \hat{u}_2^{2i-1} \cdot L_N^{(i)}(y_1^{N/2}, \hat{u}_1^{2i-2} \oplus \hat{u}_1^{2i-2}) + L_N^{(i)}(y_1^{N/2+1}, \hat{u}_1^{2i-2}) \]

Type II PE:
\[ L_N^{(2i-1)}(y_1, \hat{u}_1^{2i-1}) = 2 \cdot \text{artanh} \{ \tanh[L_N^{(i)}(y_1^{N/2}, \hat{u}_1^{2i-2} \oplus \hat{u}_1^{2i-2})/2] \cdot \tanh[L_N^{(i)}(y_1^{N/2+1}, \hat{u}_1^{2i-2})/2] \} \]
Successive cancellation (SC) is one of the most popular decoding algorithms. It is suitable for VLSI implementation for the FFT-like structure.

Type I PE:
\[
L_{N}^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = (-1)^{\hat{u}_{2i-1}}L_{N/2}^{(i)}(y_{1}^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L_{N/2}^{(i)}(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2})
\]

Type II PE:
\[
L_{N}^{(2i-1)}(y_1^N, \hat{u}_1^{2i-1}) = 2\arctanh\{\tanh[L_{N/2}^{(i)}(y_{1}^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})/2]\} \cdot \tanh[L_{N/2}^{(i)}(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2})/2]\}.
\]
SC decoding algorithm

Successive cancellation (SC) is one of the most popular decoding algorithms.

It is suitable for VLSI implementation for the FFT-like structure.

However, the decoding latency is $2(N-1)$. $N$ over $2^{10}$ are always required.

Type I PE:

$\hat{L}_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = (-1)^{\hat{u}_{2i-1}} L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})$

Type II PE:

$\hat{L}_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-1}) = 2\text{artanh}\{\tanh[L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})/2]\cdot \tanh[L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})/2]\}$
Successive cancellation (SC) is one of the most popular decoding algorithms. It is suitable for VLSI implementation for the FFT-like structure. However, the decoding latency is 2(N-1). N over $2^{10}$ are always required.

How to reduce the latency?

**Type I PE:**
$L^{(2i)}_N(y_1^N, \hat{u}_{1}^{2i-1}) = (-1)^{\hat{u}_{2i-1}} L^{(i)}_{N/2}(y_{1}^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L^{(i)}_{N/2}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})$.

**Type II PE:**
$L^{(2i-1)}_N(y_1^N, \hat{u}_{1}^{2i-1}) = 2\text{artanh}\{\tanh[L^{(i)}_{N/2}(y_{1}^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})/2] \cdot \tanh[L^{(i)}_{N/2}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})/2]\}$. 

Stage 1  ---  Stage 2  ---  Stage 3

$\hat{u}_1 \rightarrow \hat{u}_8$

$\hat{u}_5 \rightarrow \hat{u}_{10}$

$\hat{u}_3 \rightarrow \hat{u}_{13}$

$\hat{u}_7 \rightarrow \hat{u}_{17}$

$\hat{u}_9 \rightarrow \hat{u}_{18}$

Type I PE: $\bigcirc$  \quad Type II PE: $\bullet$
Data flow graph (DFG) analysis

Marked each PE with red labels as indicated.

The derived DFG is single-rated.

Now we are able to derive the decoder architectures.

Get the DFG for it.
Latency-reduced architecture

- First, we would like to derive a multi-rate version of the previous DFG.
- Then, with the look-ahead manner, it can be further refined as follows.

C. Zhang, B. Yuan and K.K. Parhi, IEEE ICC 2012
Latency-reduced architecture

**Merged PEs** are used instead of Type I and Type II PEs.

The decoding latency is only **50%** of tree architecture.

Only half of the delay elements are employed by the **feedback part**.
Based on the previous analysis on the DFG, numerous decoder architectures can be obtained for different applications.
Body Area Network
Wireless Sensor Nodes in Healthcare

- http://ieeewban.wordpress.com/author/mikuslaw/
Wireless BAN

(a) Sensors/Electrodes, Wires, Central Control Unit, Receiving device, Local PC. Wireless distance, Wireless Chip: Bluetooth, 802.15.4 (Zigbee), WLAN.

(b) EEG, ECG, PH, WBAN, CCU, Receiving device, Remote PC. 1st Wireless distance, 2nd Wireless distance. Sensor node electronics with wireless capability. For long range transmission.
WBAN Applications

- Chronic disease monitoring
- Episodic patient monitoring
- Patients alarm monitoring
- Elderly people monitoring
Biomedical monitoring systems

ONLINE DETECTION

Electrodes

Recordings from Databases

Feature extraction

Feature selection

Classifier Training

OFFLINE TRAINING

Recordings from Databases

Feature extraction

Feature selection

Classifier Training

Selected feature set

Classifier Model

Feature extraction

Classification

Post-processing

Drug Delivery System/Create an alert

Spectral power
Wavelets
Auto-regressive coefficients
ICA

Linear SVM
Non-linear SVM
Adaboost

Moving average
Kalman
Closed-loop systems

http://ieeewban.wordpress.com/author/mikuslaw/
Compared with the SISO case, channel capacity increases \( \sim \min\{M,K\} \) times by using a \( M \times K \) antenna array.
Problem Statement - Transmitter

Transmitted Signal

Problem Statement – Access Point

- Timing? (signal arrival time)
- Channel information?
- Carrier frequency offset?
Solution

- Preamble

- Access Point
Contributions

• Achieve perfect timing synchronization when SNR ≤ 0dB (100% chance to find the correct timing)
  – Existing methods only have ≤ 40% chance to find the correct timing at the same SNR

• Zero BER is achievable when SNR ≥ 0dB
  – Existing methods have error floors, and may not achieve zero BER at any SNR (SNR >> 0dB)

Support Vector Machines

• Most widely used classification algorithm
  – Training based on quadratic optimization
• Non-linear SVMs (kernel based)
  • Map $x$ to some high dimensional space $\Phi: \mathbb{R}^d \rightarrow H$
  • The derived feature vectors are $\Phi(x_j)$
• Kernel function allows implicit calculation of dot products
• Learn a linear separator in high dimensional space
  $$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$$
• The final prediction is

$$f(x) = \sum_{i: a_i > 0} a_i y_i \Phi(x_i)^T \Phi(x) + b = \sum_{i: a_i > 0} a_i y_i K(x_i, x) + b$$
Illustration of SVM
SVM Classification

- Three popular kernels
- Linear

- Polynomial

\[ K(x_i, x) = x_i^T x \]

\[ f(x) = \text{sign}\left( \sum_{i: a_i > 0} a_i y_i x_i^T x + b \right) = \text{sign}\left( w^T x + b \right) \quad \text{where} \quad w = \sum_i a_i y_i x_i \]

\[ K(x_i, x) = [x_i^T x + 1]^p \]

\( p=2: \)

\[ K(x_i, x) = (z_i^T z)^2 \quad \text{where} \quad z = [x \ 1]^T \]

\[ = z_i^T z_i z_i^T z \]
SVM Classification

• Polynomial

\[
f(x) = \text{sign} \left( z^T W z + b \right)
\]
where

\[
W = \sum_{i:a_i > 0} a_i y_i z_i z_i^T
\]

• RBF Kernel

\[
K(x_i, x) = e^{-\frac{\|x-x_i\|^2}{\sigma^2}}
\]

\[
f(x) = \text{sign} \left( \sum_{i:a_i > 0} a_i y_i K(x_i, x) + b \right)
\]

• Further simplifications not possible for RBF
### Computational complexity

<table>
<thead>
<tr>
<th>Kernel</th>
<th>#words (memory)</th>
<th>#additions</th>
<th>#multiplications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>d</td>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>Polynomial (p=2)</td>
<td>$d^2$</td>
<td>$d^2$</td>
<td>$d(d+1)$</td>
</tr>
<tr>
<td>RBF</td>
<td>$N_{sv}(d+1)$</td>
<td>$2N_{sv}d$</td>
<td>$N_{sv}d$</td>
</tr>
</tbody>
</table>

* Additional $N_{sv}$ exponential operations for RBF

- Complexity depends on number of support vectors and # dimensions
Reducing the complexity

- Number of support vectors ($N_{sv}$)
  - Reduced SVM (RSVM) can be used
  - Number of SVs decrease while training
- Feature dimensionality ($d$)
  - Feature selection algorithms
  - SVM-RFE (Recursive Feature Elimination)
  - Adaboost, HPD, etc.,
- Optimizing the hardware
  - MAC and exponent operations
  - Memory requirements depends on word length
Configurable SVM processor
### TABLE IV
**Energy Consumption of the Proposed Architecture per Test Vector**

<table>
<thead>
<tr>
<th>Vdd (V)</th>
<th>$N_{sv}$</th>
<th>$d$</th>
<th>Energy (pJ) w/ opt</th>
<th>Energy (pJ) w/o opt</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>32</td>
<td></td>
<td>49.1</td>
<td>70.92</td>
<td>10 MHz</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td></td>
<td>81.8</td>
<td>118.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>128</td>
<td></td>
<td>147.3</td>
<td>212.76</td>
<td></td>
</tr>
<tr>
<td>2$^{nd}$ order Polynomial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>32</td>
<td></td>
<td>1570.6</td>
<td>22269.44</td>
<td>10 MHz</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td></td>
<td>5235.2</td>
<td>7564.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>128</td>
<td></td>
<td>18847.14</td>
<td>27223.28</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vdd (V)</th>
<th>$N_{sv}$</th>
<th>$d$</th>
<th>Energy (pJ)</th>
<th>kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>100</td>
<td>64</td>
<td>81.8</td>
<td>Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5235.2</td>
<td>Poly</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10633.61</td>
<td>RBF</td>
</tr>
</tbody>
</table>

### TABLE V
**Energy Consumption of the RBF Kernel per Test Vector**

<table>
<thead>
<tr>
<th>$N_{sv}$</th>
<th>$d$</th>
<th>Energy (pJ)$^1$</th>
<th>Energy (pJ)$^2$</th>
<th>Energy (pJ)$^3$</th>
<th>Energy (pJ)$^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>8</td>
<td>675.7</td>
<td>576.03</td>
<td>565.68</td>
<td>466.01</td>
</tr>
<tr>
<td>25</td>
<td>16</td>
<td>2002.37</td>
<td>1745.12</td>
<td>1635.61</td>
<td>1378.36</td>
</tr>
<tr>
<td>50</td>
<td>32</td>
<td>5257.1</td>
<td>4710.36</td>
<td>4156.83</td>
<td>3610.08</td>
</tr>
</tbody>
</table>

$^1$ with no optimizations  $^2$ with only non-uniform LUT  
$^3$ with only fixed-width MAC  $^4$ with both optimizations
FFT Architectures: Prior Designs

- 4-parallel delay-feedback

Contains 4 datapaths


Yuan Chen, et al. 2008
DIF Design

- 4-parallel feed-forward design (DIF)

- Requires 3N/2 delay elements
- Hardware utilization is 100%
DIF Design

- 8-parallel feed-forward design
DIT 4-parallel Architecture

- N-point FFT requires
  - N-4 delay elements
  - $4\log N$ complex adders
  - #multipliers depend on the algorithm

No delays at this stage
DIT 8-parallel Architecture

- Requires N-8 delay elements

No delays at two stages
DIT128-point FFT Architecture

• Hardware complexity
  – Complex adders: 28
  – Complex multipliers: 4+0.41
  – Delay elements: 124

M. Ayinala, M. Brown, K.K. Parhi, IEEE Trans. VLSI Systems, June 2012 (patent)
Seizure prediction

(a) Open-loop

(b) Closed-loop
**Seizure Prediction**

- **Objective**: Patient-specific prediction of seizures (5 min ahead) from EEG signal (6 electrodes)
- **Issues**: unbalanced data, feature selection

![Diagram of seizure prediction process]

- **System implementation details**:
  - **features**: power measured in 9 spectral bands for 4 differential channels. Total $4 \times 6 = 36$ features
  - **classifier**: Adaboost with decision stumps

EEG Data for Classification

- Parts of EEG data identified by medical experts: ictal, preictal (+), interictal(-)
- Preictal and interictal data used for classification
- Each data sample ~20 sec moving window

![EEG Signal - Preictal, Ictal, Interictal Region](image)

- At least 1-hour gap
- Preictal (Class +1)
- Interictal (Class -1)
Seizure Prediction

<table>
<thead>
<tr>
<th></th>
<th># Pat</th>
<th># Sz</th>
<th>Prediction Horizon</th>
<th>Sens (%)</th>
<th>FP/hr</th>
<th>Feature</th>
<th># Features</th>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>[6]</td>
<td>19</td>
<td>83</td>
<td>30</td>
<td>90.8</td>
<td>0.094</td>
<td>Correlation</td>
<td>20</td>
<td>SVM-RBF</td>
</tr>
<tr>
<td>[3]</td>
<td>9</td>
<td>18</td>
<td>15</td>
<td>100</td>
<td>0.17</td>
<td>AR coefficients</td>
<td>36</td>
<td>SVM-RBF</td>
</tr>
<tr>
<td>[1]</td>
<td>18</td>
<td>80</td>
<td>30</td>
<td>97.5</td>
<td>0.27</td>
<td>Spectral Power</td>
<td>36</td>
<td>SVM-RBF</td>
</tr>
<tr>
<td>Proposed</td>
<td>16</td>
<td>71</td>
<td>30</td>
<td>94.375</td>
<td>0.13</td>
<td>Spectral Power</td>
<td>36</td>
<td>Adaboost</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td></td>
<td></td>
<td>91.25</td>
<td>0.27</td>
<td>&quot;</td>
<td>36</td>
<td>SVM-Linear</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td></td>
<td></td>
<td>94.375</td>
<td>0.14</td>
<td>&quot;</td>
<td>4.8125</td>
<td>Adaboost</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td></td>
<td></td>
<td>67.1825</td>
<td>0.15</td>
<td>&quot;</td>
<td>5</td>
<td>SVM-Linear</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td></td>
<td></td>
<td>85.625</td>
<td>0.19</td>
<td>&quot;</td>
<td>10</td>
<td>SVM-Linear</td>
</tr>
</tbody>
</table>
Seizure Prediction

TABLE III
COMPARISON OF SEIZURE PREDICTION ALGORITHMS

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Sen%</th>
<th>FP/hr</th>
<th>FP %</th>
<th># Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>0.7917</td>
<td>36.3657</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>0.0833</td>
<td>4.1667</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>0.25</td>
<td>12.5</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>0.1667</td>
<td>8.33</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>75</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>75</td>
<td>0.0833</td>
<td>4.1667</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td>0.1667</td>
<td>8.333</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>100</td>
<td>0.1667</td>
<td>8.333</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>100</td>
<td>0.125</td>
<td>6.25</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>100</td>
<td>0.1667</td>
<td>8.33</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>100</td>
<td>0.1667</td>
<td>8.333</td>
<td>5</td>
</tr>
<tr>
<td>Mean</td>
<td>94.375</td>
<td>0.135</td>
<td>6.487</td>
<td>4.8125</td>
</tr>
</tbody>
</table>
Physical Unclonable Functions (PUFs)

“It is estimated that as much as 10% of all high-tech products sold globally are counterfeit which leads to a conservative estimate of 100 billion of revenue loss.”

[Guajardo et al, 2008]

- Security Challenges
  - Computing devices are becoming physically exposed
  - Adversaries may physically temper the devices and extract secret keys from non-volatile memory
  - Software-only protections are not enough
What is PUF?

- Extract secret keys from complex physical objects
- Due to manufacturing process variations, no two Integrated Circuits even with the same layouts are identical

Unclonable anti-counterfeiting marks for ICs!
Silicon MUX PUF

- All the multiplexers are identically designed.
- Each challenge creates two paths through the circuit.
- The response is generated by the racing result of the two paths.
- No special fabrication needed.
Characteristics of PUFs

• Security
  – Uniqueness: inter-chip variation
  – Unclonability: randomness
  – Unpredictability: hard to model

• Reliability
  – Intra-chip variation
  – Authentication robustness (add extra processing circuits, e.g., error correcting techniques)
Contributions

• Logically-reconfigurable PUFs (security)
• Systematic statistical analysis of (feed-forward) MUX PUFs
• Modified feed-forward path (reliability)
• Two-arbiter authentication scheme (reliability)

Logically-reconfigurable PUFs

- **Reconfigurable PUF circuit**
  - Alter the model of PUF circuit to update the challenge-response behavior, instead of re-mapping the challenge and response through pre- and/or post-processing
  - Several novel solutions, e.g.,
    - Reconfigurable feed-forward MUX PUF
    - MUX and DeMUX PUF
Reconfigurable feed-forward MUX PUF

• Ideas: using reconfigurable feed-forward path
  – Original MUX PUF can be modeled as a linear additive delay model
  – Feed-forward path: add nonlinearity to MUX PUF, improve the security

• Three types of feed-forward path: Cascade, Overlap, Separate
  – based on the beginning stage and the ending stage of the feed-forward path
  – experimental results have shown that the inter-chip and intra-chip characteristics of the 3 types are different
  – our statistical analysis has demonstrated that the mathematical models of the 3 types are different
Why reconfigurable PUFs?

Reconfigurability is desirable:
1. Application needs: updatable authentication keys
2. Improving the security, as the challenge-response behaviors can be updated (against modeling attacks).

Solutions for reconfigurability

- **Challenge-like**: Vulnerable to attacks & Poor performance.
- **Reconfigurable RO Silicon PUF**: The frequencies of ring oscillators are possible to be evaluated by attackers.
- **FPGA based**: Hard to implement: lower level design detail, symmetrical routing.
Conclusions

• Wireless communications systems for body area network will grow significantly
• Biomedical monitoring systems and drug delivery systems will grow
• Low-power DSP for biomedical monitoring will grow
• IC Chip Security by PUFs for biomedical systems
Acknowledgements

• Chuan Zhang, Bo Yuan (Polar)
• Te-Lung Kung (Wireless BAN)
• Manohar Ayinala (SVM, FFT)
• Yun-Sang Park, Lan Luo, Prof. T. Netoff (Epilepsy)
• Yingjie Lao (PUF)