An Efficient Method to Threshold Logic Functions Identification

Augusto Neutzling, Mayler Martins, Renato Ribas e André Reis
Threshold Logic – Motivations and challenges

- Main researches about Threshold Logic were in 1960, 1970
- Predominance of CMOS logic, based on ANDs and ORs

- Limits of CMOS devices scaling
- Threshold logic is a very efficient choice for development of nanotechnology based circuits. Like RTDs, QCAs, SET and Spintronics

- The great challenge is how to do efficiently a complete design flow with threshold logic gates.
- This requires that CAD tools are developed focused on this kind of logic
Threshold Logic Gates

Majority

X1  X2  X3

0   0   0  ->  1   1   1

0   0   0  ->  0   0   0

1   1   1  ->  1   1   1

1   1   1  ->  0   0   0

www.inf.ufrgs.br/logics
Threshold Logic Gates

Majority

\[ f(X_1, X_2, X_3) = \text{Maj}(X_1, X_2, X_3) \]

\[ \text{AND} \]

Threshold

\[ f(X_1, X_2, X_3) = \text{Threshold}(X_1, X_2, X_3) \]

\[ \text{OR} \]
Threshold Logic Gates

Weights

$f = X_1 + (X_2 \cdot X_3)$

Majority

$f = X_1 \cdot (X_2 + X_3)$
Threshold Logic - Basic Concepts

\[ X_1, X_2, \ldots, X_n \]

\[ W_1, W_2, \ldots, W_n \]

\[ f \]
\( X_1 + X_2X_3 \)

AND

\( X_1 \) \( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \)

\( X_2 \) \( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \)

\( X_3 \) \( \begin{array}{c} 1 \\ 3 \end{array} \)

\( f \)

OR

\( X_1 \) \( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \)

\( X_2 \) \( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \)

\( X_3 \) \( \begin{array}{c} 1 \\ 1 \end{array} \)

\( f \)

Maj

\( X_1 \) \( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \)

\( X_2 \) \( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \)

\( X_3 \) \( \begin{array}{c} 1 \\ 1 \end{array} \)

\( f \)

\( X_1X_2 + X_1X_3 \)

\( X_1 \) \( \begin{array}{c} 2 \\ 1 \\ 1 \end{array} \)

\( X_2 \) \( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \)

\( X_3 \) \( \begin{array}{c} 1 \\ 1 \end{array} \)

\( f \)

\( X_1 + X_2X_3 \)

\( X_1 \) \( \begin{array}{c} 2 \\ 1 \\ 1 \end{array} \)

\( X_2 \) \( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \)

\( X_3 \) \( \begin{array}{c} 1 \\ 1 \end{array} \)

\( f \)
Threshold Logic Gates

\[ f = x_1x_2 + (x_1+x_2)(x_3x_4+x_3x_5+x_4x_5) \]

\[ f = x_2(x_1+x_3+x_4) + x_4(x_3+x_1) \]

Full Adder
Threshold Logic – main advantages

- Economy in number of gates and interconnections, compared to traditional Boolean Function

- For instance, the method proposed in (Zhang, 2004)* allows up to 77% reduction in the gate count, with an average reduction being 55%
Threshold Logic – Motivations and challenges

- Main researches about Threshold Logic were in 1960, 1970
- Predominance of CMOS logic, based on ANDs and ORs

- Limits of CMOS devices scaling
- Threshold logic is a very efficient choice for development of nanotechnology based circuits. Like RTDs, QCAs, SET and Spintronics

- The great challenge is how to do efficiently a complete design flow with threshold logic gates.
- This requires that CAD tools are developed focused on this kind of logic.
Threshold Logic Identification

1. **Boolean Function**
2. **Is a TLF?**
   - **False** → **More than one TLG**
   - **True** → **Assign weights and Threshold Value**
3. **Threshold Logic Gate**
Threshold Logic Identification

- **Integer Linear Programming (ILP)**
  - Exact Algorithm
  - Large execution time
  - Is not scalable

- **Heuristic Approaches**
  - Does not find all solutions
  - Small execution time
  - Scalable

- **The state-of-art is:**
Experimental Results

Average Time per Function (miliSeconds)

- ILP
- Our Approach

Number of Variables

1.000,00
100.000,00
10.000,00
1.000,00
100,00
10,00
1,00
0,10
0,01

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

~30 sec
~0.003 sec
### Experimental Results

#### Number of NP classes identified TLF by each method

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TLF</td>
<td>TLF</td>
<td>%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>17</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>92</td>
<td>84</td>
<td>91.3</td>
</tr>
<tr>
<td>6</td>
<td>994</td>
<td>728</td>
<td>73.2</td>
</tr>
<tr>
<td>7</td>
<td>28262</td>
<td>9221</td>
<td>32.6</td>
</tr>
</tbody>
</table>


Thanks

Augusto Neutzling