# Designing of QCA Schemes by Boundary Functions

V. Ostrovsky, I. Levin, O. Keren

Tel Aviv University, Bar Ilan University, Israel

# Outline

- Boundary functions
- Representation of Boolean functions by Boundary functions
- Boundary oriented decomposition
- Majority implementation by boundary functions
- Boundary Comparator
- QCA based Boundary Comparator
- Programmable Comparator Array (PCA)
- Synthesis of Boolean Functions by PCA
- Benchmark results
- Conclusions

# Definitions

Function $F\left(x_0, x_1, \ldots, x_{n-1}\right)$ is called threshold if it can be expressed as follows:

$$F\left(x_{n-1}, \ldots, x_0\right) = Sign\left(v_{n-1}x_{n-1} + \ldots + v_0 x_0 - a\right) =$$

$$= Sign\left(\sum_{j=0}^{n-1} v_j x_j - a\right),$$

where $SignA = \begin{cases} 1 & if \quad A \geq 0 \\ 0 & if \quad A < 0 \end{cases}$

$v_j$ - weight of input (variable) $x_j$;

$a$ - threshold.

# Definitions

A boundary function is a threshold function

having $\nu_i = 2^i$ weight of its $i^{th}$ arguments:

$$y = \Gamma\left(x_{n-1},\ldots,x_0; a\right) = \Gamma\left(a\right),$$

$x_{n-1},\ldots,x_0$ – arguments,

$a$ – threshold.

$$y = Sign\left(\sum_{i=1}^{n} x_i 2^i - a\right).$$

# Definitions

$$a = \sum_{i=0}^{n-1} \alpha_i 2^i; \quad a \in \left[0, 2^n - 1\right]$$

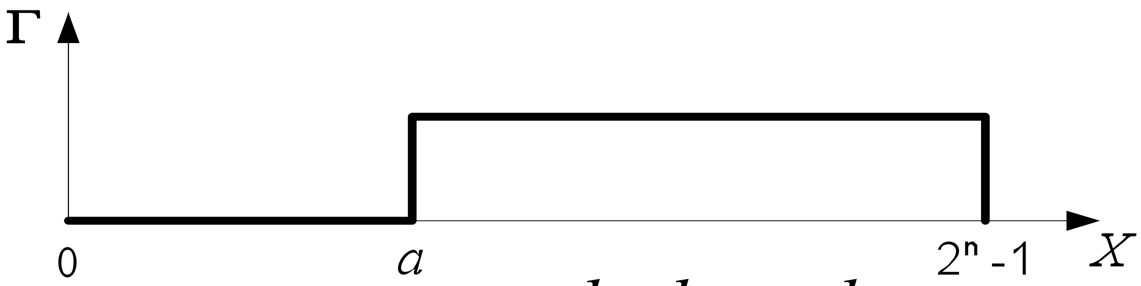
$$\vec{a} = \left(\alpha_{n-1}, \ldots, \alpha_0\right),\ \alpha_i \in \left\{0,1\right\}$$

One to one correspondence: $a \leftrightarrow \vec{a}$ and $x_i \leftrightarrow \alpha_i$

If $a \leftrightarrow \vec{a}$ and $b \leftrightarrow \vec{b}$, then from $\vec{a} > \vec{b}$ follows: $a > b$

But: from $a > b$ does not follow: $\vec{a} > \vec{b}$

# Graph of the boundary function




$a$ – *the bound*

| | | | | | | |
|---|---|---|---|---|---|---|
| | | 0 | 0 | 1 | 1 | $x_1$ |
| $x_3$ | $x_2$ | 0 | 1 | 1 | 0 | $x_0$ |
| 0 | 0 | 0 | 1 | 3 | 2 | |
| 0 | 1 | 4 | 5 | 7 | 6 | |
| 1 | 1 | 12 | 13 | 15 | 14 | |
| 1 | 0 | 8 | 9 | 11 | 10 | |

Example: ■ $\Gamma(9)$

# Boundary Representation of Boolean function

Any Boolean function $y = F\left(x_{n-1}, \ldots, x_0\right)$ can be defined by a set of its bound points $A = \left\{a_1, \ldots, a_k\right\}$

Notations: $y = F\left(x_{n-1}, \ldots, x_0; \left\{a_1, \ldots, a_k\right\}\right)$

$y = F\left(A\right)$ $y = F\left(x_{n-1}, \ldots, x_0; A\right)$ $y = F\left(a_1, \ldots, a_k\right)$

Example: $y_i = F_i\left(x_3, x_2, x_1, x_0\right)$ equals to 1 on the set of points: {3,4,5,6,11,12,13,14,15}.

Its bound points are $a_1 = 3; a_2 = 7; a_3 = 11$.

$y_i = F_i\left(x_3, x_2, x_1, x_0; \left\{3,7,11\right\}\right)$ or $y_i = F_i\left(3,7,11\right)$

# Boundary Representation of a Boolean function

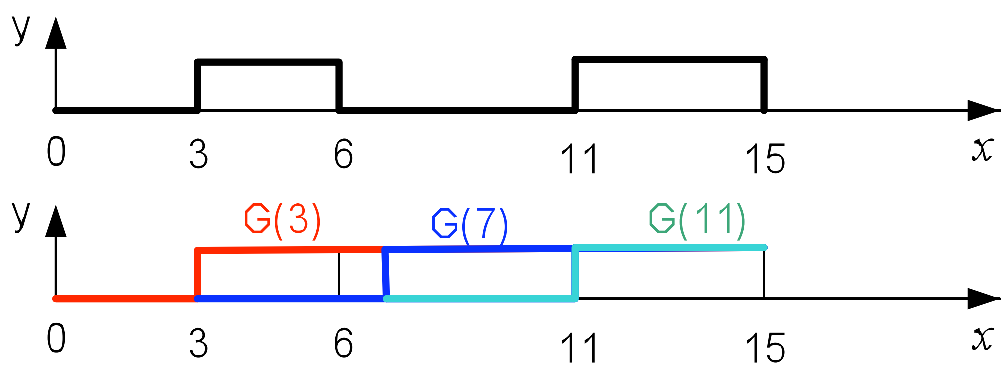
$$y = F(\{a_1,\ldots,a_k\}) = \Gamma(a_1)\overline{\Gamma(a_2)\ldots\overline{\Gamma(a_k)}} =$$

$$= \Gamma(a_1) \oplus \cdots \oplus \Gamma(a_k) =$$

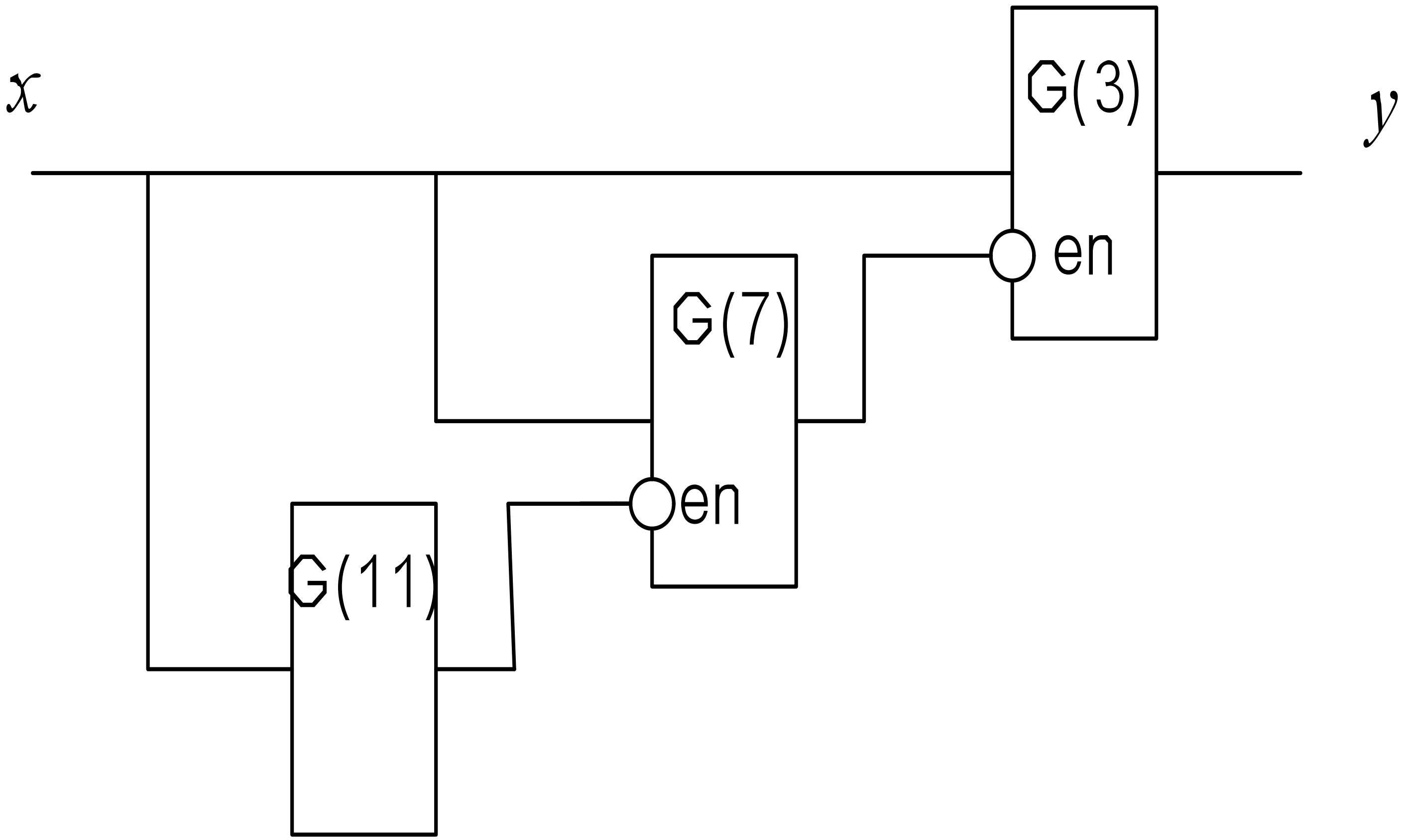
$$= \Gamma(a_1)\overline{\Gamma(a_2)} + \cdots + \Gamma(a_{k-1})\overline{\Gamma(a_k)}.$$

$$y_1 = F(\{3,7,11\})$$

$$y_1 = \Gamma(3)\overline{\Gamma(7)\overline{\Gamma(11)}} = \Gamma(3) \oplus \Gamma(7) \oplus \Gamma(11) = \Gamma(3)\overline{\Gamma(7)} + \Gamma(11).$$

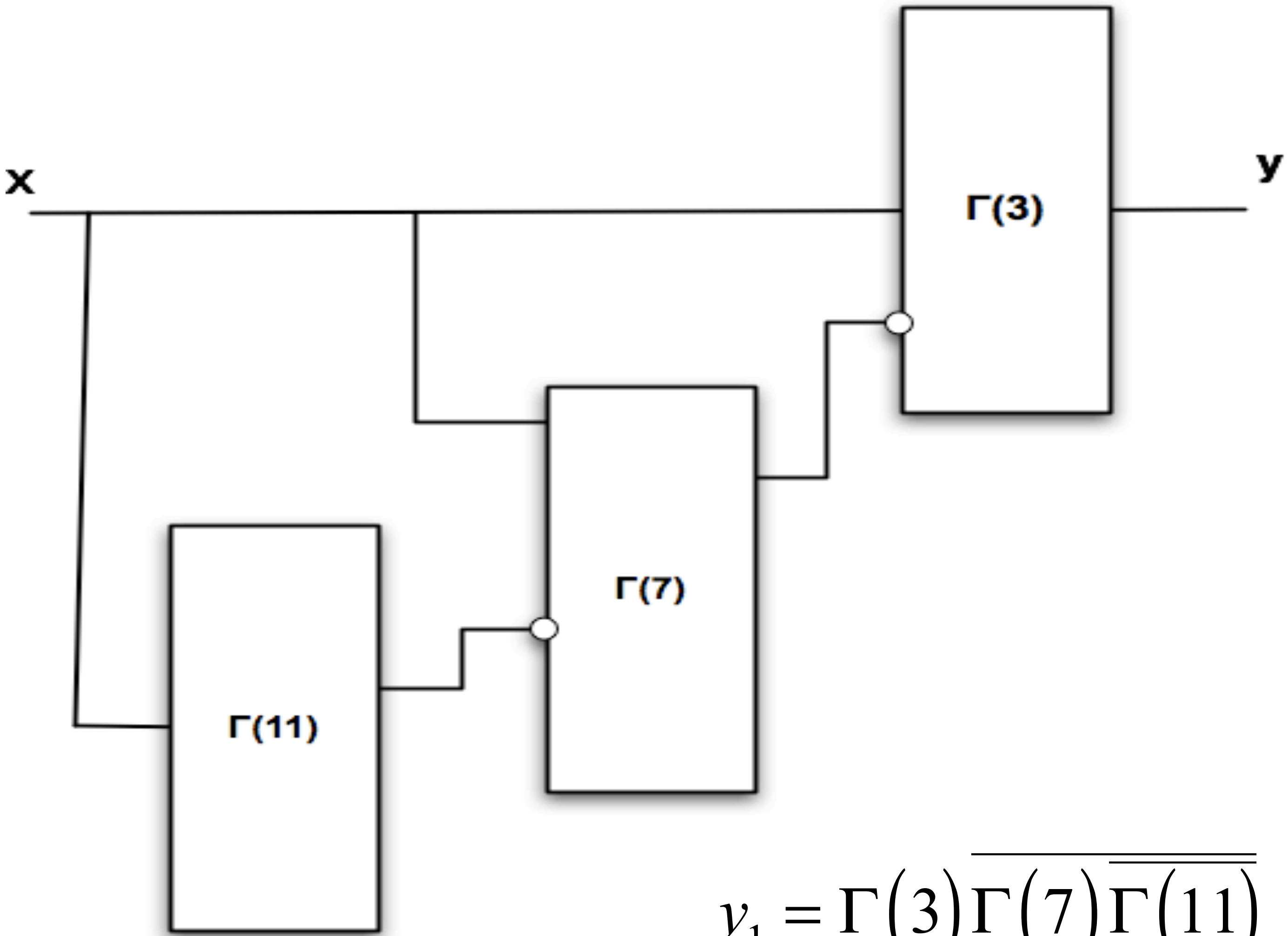


# Implementation of $y_1 = F(\{3,7,11\})$ by a network of boundary functions




$$y_1 = \Gamma(3)\overline{\Gamma(7)\overline{\Gamma(11)}}$$

# Implementation of $y_1 = F\left(\{3, 7, 11\}\right)$ by a network of boundary functions




$$y_1 = \Gamma\left(3\right)\overline{\Gamma\left(7\right)\overline{\Gamma\left(11\right)}}$$

# Interval functions

A logic function being equal to 1 only on interval $[a,b)$ is called ***interval function***

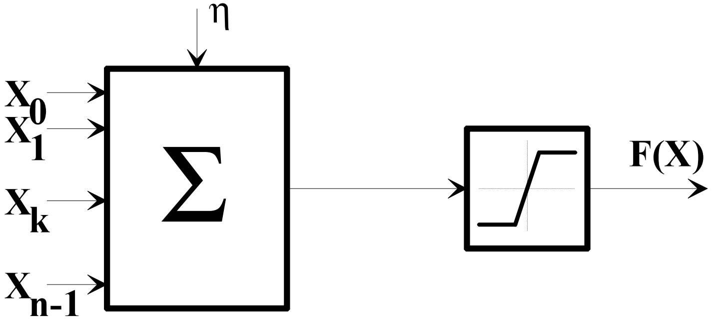
$$y = I\left(x_{n-1},\dots,x_0;[a,b)\right) \qquad y = I\left([a,b)\right)$$

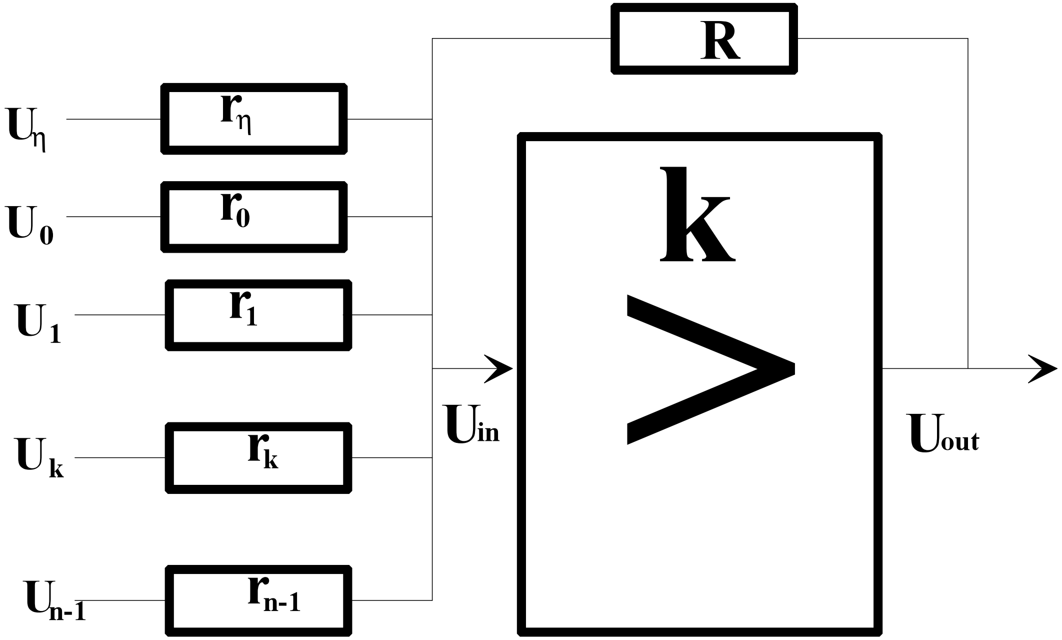
$$I\left([a,b)\right) = \Gamma(a) \oplus \Gamma(b) = \Gamma(a)\overline{\Gamma(b)}$$

Any Boolean function $F$ can be presented as

$$F = I_1 + \dots + I_s = \Gamma\left(I_1,\dots,I_s;0\dots01\right)$$

# Analog Implementation of a threshold element

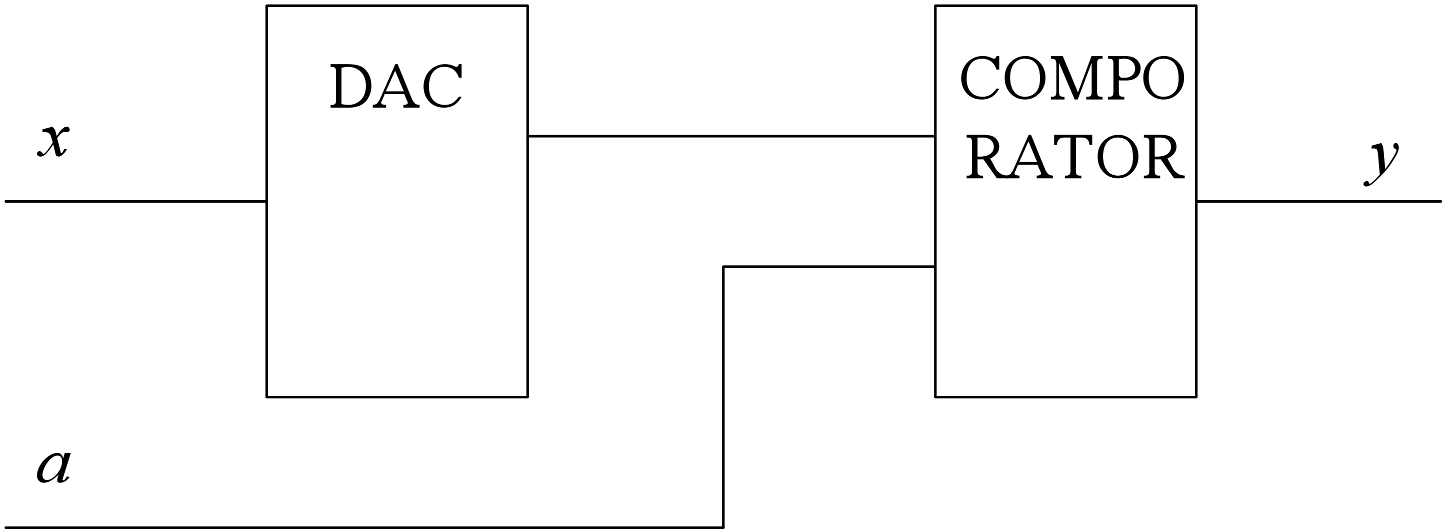



Example: ***Summing Amplifier***.

# Implementation

## 1. Analog implementation




Analog boundary gate

# Digital implementation

**Synthesis of boundary functions**

Put components of a binary vector of the threshold into correspondence with arguments of the function.

For example, *a* =001101001 :

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| $x_8$ | $x_7$ | $x_6$ | $x_5$ | $x_4$ | $x_3$ | $x_2$ | $x_1$ | $x_0$ |

Put “+” after 0 and “&” after 1

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| $x_8+$ | $x_7+$ | $x_6$ | $x_5$ | $x_4+$ | $x_3$ | $x_2+$ | $x_1+$ | $x_0$ |

If 0 follows 1, the expression that begins with '0’ appears in brackets:

$$\Gamma(001101001) = x_8 + x_7 + x_6 x_5 \left(x_4 + x_3 \left(x_2 + x_1 + x_0\right)\right)$$

# Example 1

$$y_1 = F\left(x_3, x_2, x_1, x_0; \left\{3, 7, 11\right\}\right)$$

$$\Gamma(3) = \Gamma(0011) = x_3 + x_2 + x_1x_0$$

$$\Gamma(7) = \Gamma(0111) = x_3 + x_2x_1x_0$$

$$\Gamma(11) = \Gamma(1011) = x_3(x_2 + x_1x_0)$$

$$y_1 = (x_3 + x_2 + x_1x_0)\overline{(x_3 + x_2x_1x_0)} + x_3(x_2 + x_1x_0) =$$

$$\overline{x}_3\left(x_2 + x_1x_0\right)\overline{\left(x_2x_1x_0\right)} + x_3(x_2 + x_1x_0)$$

$$y_1 = \overline{x}_1x_2 + \overline{x}_0x_2 + x_2x_3 + x_0x_1\overline{x}_2$$

# Example 2

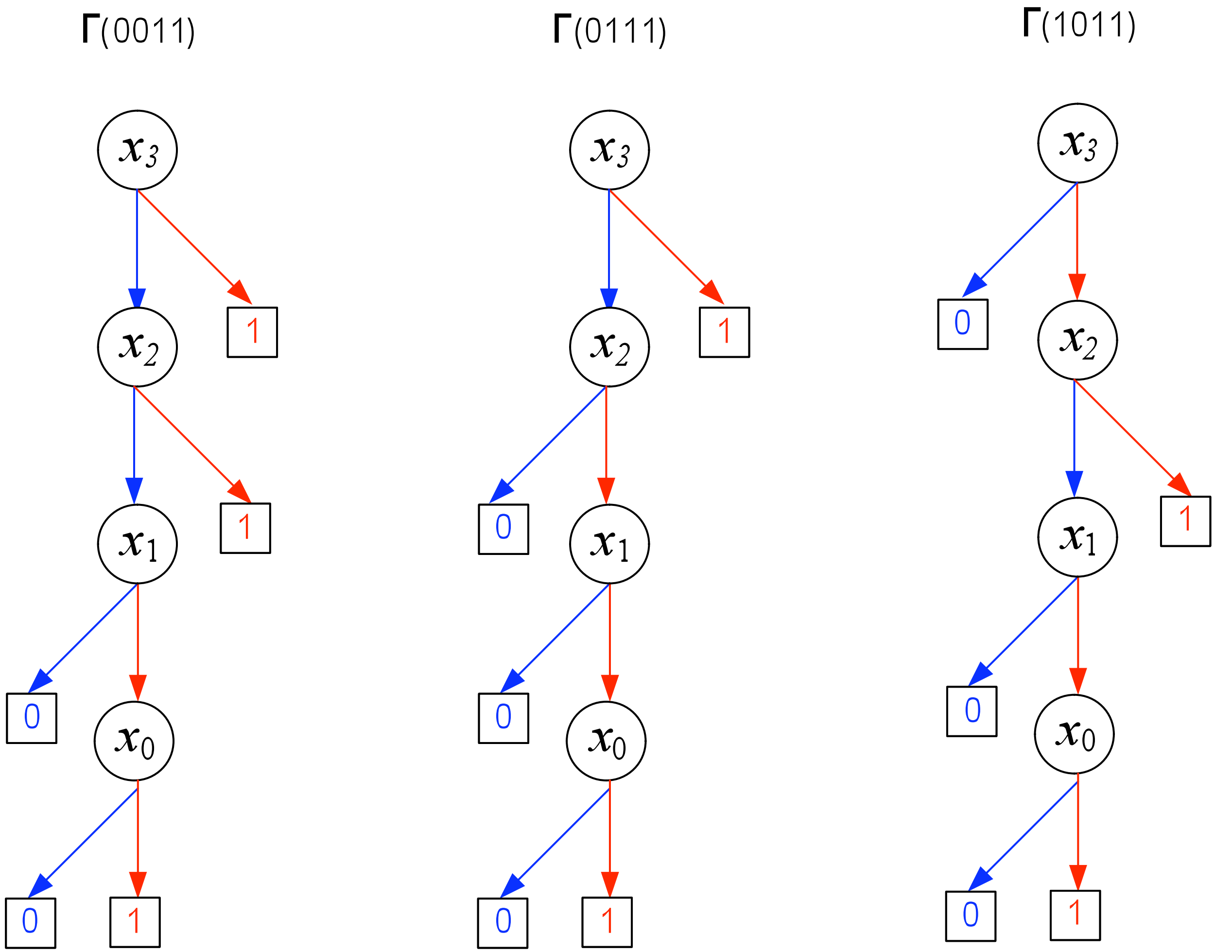
Function $y_2 = x_0 x_2 \overline{x}_3 + x_1 x_2 \overline{x}_3 + \overline{x}_1 x_3 + \overline{x}_2 x_3$

| | | 0 | 0 | 1 | 1 | $x_1$ |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 1 | 0 | $x_0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 1 | 1 | 1 | |
| 1 | 1 | 1 | 1 | 0 | 0 | |
| 1 | 0 | 1 | 1 | 1 | 1 | |
| $x_3$ | $x_2$ | | | | | |

$$y_2 = \Gamma(5)\overline{\Gamma(14)} = \left(x_3 + x_2\left(x_1 + x_0\right)\right)\overline{x_3 x_2 x_1}$$

# BDDs of Boundary Functions

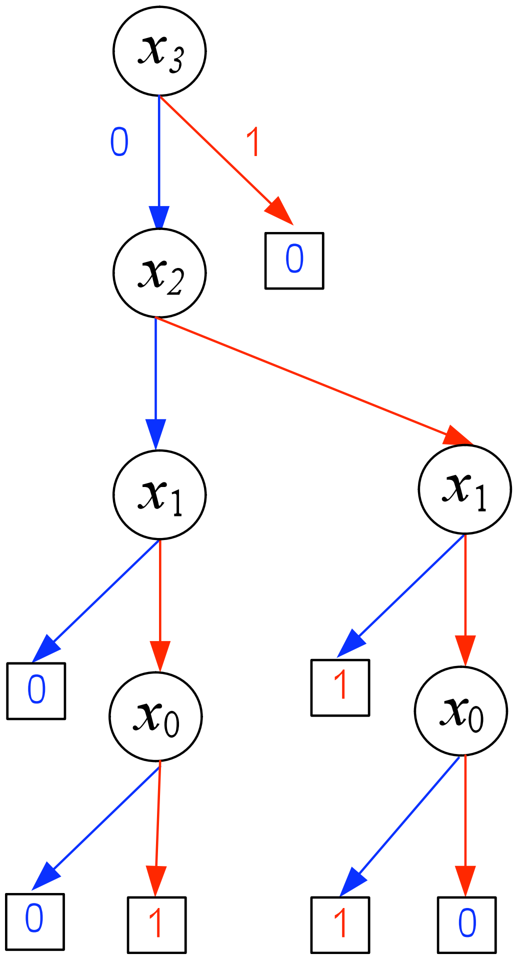
$$y_1 = F\left(\{3,7,11\}\right)$$



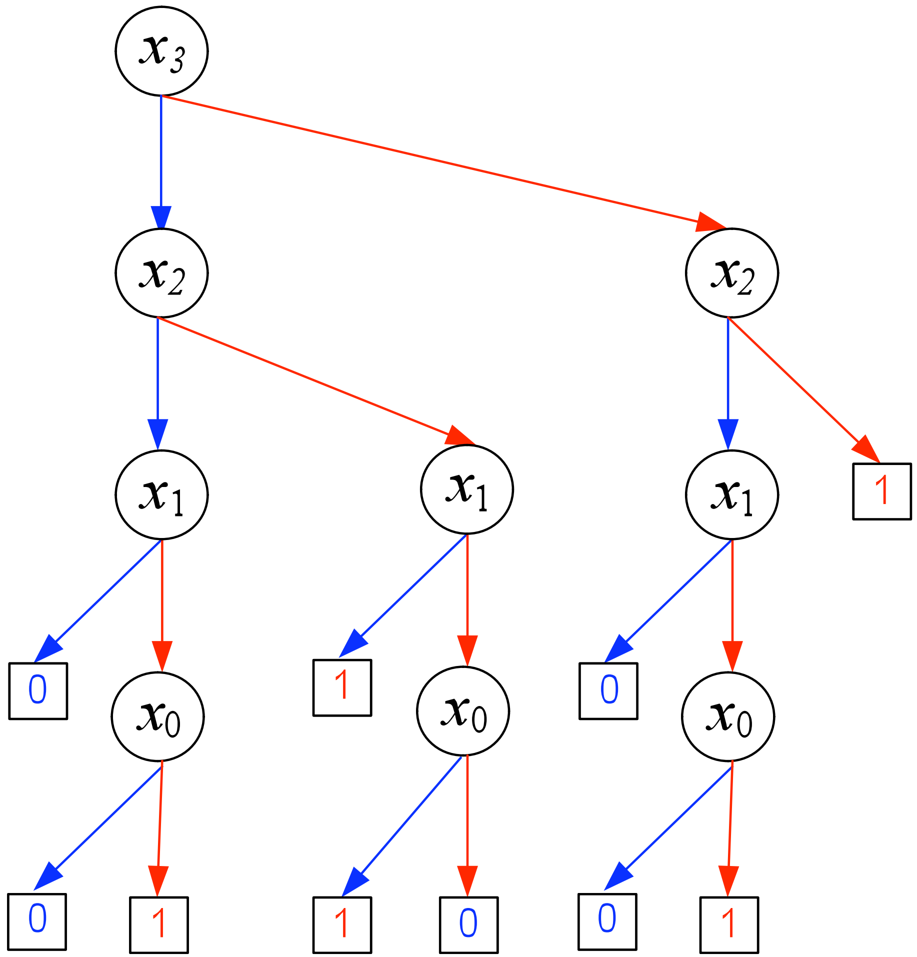

Separate BDDs

# Merging of separate BDDs

$\Gamma(0011)\,\overline{\Gamma(0111)}$




$\Gamma(0011)\,\overline{\Gamma(0111)}+\Gamma(1011)$



# Decomposition of Boundary Functions

For any boundary function

a simple disjoint decomposition exists :

$$\Gamma\left(X_a, X_b; a.b\right) = \Gamma\left(X_a, \Gamma\left(X_b; b\right); a.1\right)$$

For example : $\Gamma\left(01.111\right) = \Gamma\left(x_4, x_3, \Gamma\left(x_2, x_1, x_0; 111\right); 011\right) =$

$$= \Gamma\left(x_4, x_3, \left(x_2 x_1 x_0\right); 011\right) = x_4 + x_3\left(x_2 x_1 x_0\right)$$

Let $X_a = x_{n-1}$ and $a = \alpha_{n-1}$, then:

$$\Gamma\left(x_{n-1}, X_b; \alpha_{n-1}.b\right) = \Gamma\left(x_{n-1}, \Gamma\left(X_b; b\right); \alpha_{n-1}.1\right) =$$

$$= \alpha_{n-1} x_{n-1} \Gamma\left(b\right) + \bar{\alpha}_{n-1}\left(x_{n-1} + \Gamma\left(b\right)\right)$$

$$= maj\left(\Gamma\left(b\right), x_{n-1}, \bar{\alpha}_{n-1}\right)$$

# Partition on a set of arguments

Let $y = \Gamma\left(X; A\right)$, and $X = X_a \cup X_b; \left(X_a \cap X_b = \varnothing\right)$

Then : $y = \Gamma\left(X_a . X_b; a.b\right)$.

For example: $y_3 = \Gamma\left(x_4, x_3 . x_2, x_1, x_0; 01.101\right)$,

$X_a = \left\{x_4, x_3\right\}, X_b = \left\{x_2, x_1, x_0\right\}, a = 01, b = 101$

Theorem :

$\Gamma\left(X_a . X_b; a.b\right) = \Gamma\left(X_a; a+1\right) + \Gamma\left(X_a; a\right)\Gamma\left(X_b; b\right)$;

or : $\Gamma\left(a.b\right) = \Gamma\left(a+1\right) + \Gamma\left(a\right)\Gamma\left(b\right)$

In the example : $y_3 = \Gamma\left(x_4, x_3; 10\right) + \Gamma\left(x_4, x_3; 01\right)\Gamma\left(x_2, x_1, x_0; 101\right)$

$y_3 = x_4 + \left(x_4 + x_3\right)\left(x_2\left(x_1 + x_0\right)\right)$

In a case of a different partition:

$\Gamma\left(011.01\right) = \Gamma\left(100\right) + \Gamma\left(011\right)\Gamma\left(01\right) = x_4 + \left(x_4 + x_3 x_2\right)\left(x_1 + x_0\right)$

# Functional expansion of Boundary Functions

Two above - obtained formulae

$$\Gamma\left(x_{n-1}, X_b; \alpha_{n-1}.\, b\right) = maj\left(\Gamma\left(b\right), x_{n-1}, \bar{\alpha}_{n-1}\right) \text{ and}$$

$$\Gamma\left(a\,.\,b\right) = \Gamma\left(a+1\right) + \Gamma\left(a\right)\Gamma\left(b\right)$$

enable to express Boundary expansion formulae.

Let $\alpha_i$ be a certain bit of the threshold. Then :

$$\Gamma\left(a.\,\alpha_i.\,b\right) = \Gamma\left(a+1\right) + \Gamma\left(a\right)\Gamma\left(\alpha_i.\,b\right) = \Gamma\left(a+1\right) + \Gamma\left(a\right) maj\left(\Gamma\left(b\right), x_i, \bar{\alpha}_i\right)$$

For example :

$$\Gamma\left(01.\boldsymbol{1}.11\right) = \Gamma\left(x_4, x_3; 10\right) + \Gamma\left(x_4, x_3; 01\right) maj\left(\Gamma\left(x_1, x_0; 11\right), x_2, \boldsymbol{0}\right) =$$

$$= x_4 + \left(x_4 + x_3\right)\left(x_2 x_1 x_0\right)$$

$$\Gamma\left(0.\boldsymbol{1}.111\right) = \Gamma\left(x_4; 1\right) + \Gamma\left(x_4; 0\right) maj\left(\Gamma\left(x_2, x_1, x_0; 111\right), x_3, \boldsymbol{0}\right) =$$

$$= x_4 + \left(x_3 x_2 x_1 x_0\right)$$

# Important Cases of Expansion

The expansion can be simplified in the following important cases :

$$\alpha_i = \boldsymbol{1} \rightarrow \Gamma\left(a.\boldsymbol{1}.b\right) = \Gamma\left(a+1\right) + \Gamma\left(a\right)\left(\Gamma\left(b\right)x_i\right)$$

$$\alpha_i = \boldsymbol{0} \rightarrow \Gamma\left(a.\boldsymbol{0}.b\right) = \Gamma\left(a+1\right) + \Gamma\left(a\right)\left(\Gamma\left(b\right) + x_i\right)$$

$$\Gamma\left(\alpha_{n-1}.b\right) = maj\left(\Gamma\left(b\right), x_{n-1}, \bar{\alpha}_{n-1}\right)$$

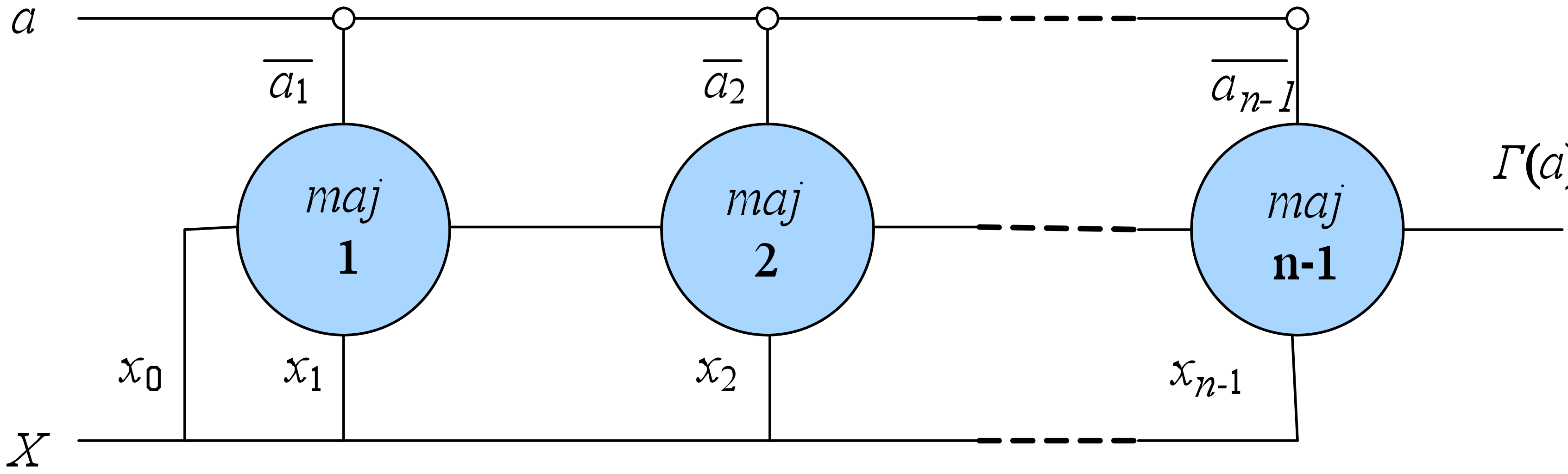
$$\Gamma\left(a.\alpha_0\right) = \Gamma\left(a+1\right) + \Gamma\left(a\right) maj\left(\Gamma\left(b\right), x_0, 0\right) = \Gamma\left(a+1\right) + \Gamma\left(a\right)x_0$$

$$\Gamma\left(X;a\right) = maj\left(\bar{\alpha}_{n-1}, x_{n-1}, maj\left(\bar{\alpha}_{n-2}, x_{n-2}, maj\left(\ldots maj\left(\bar{\alpha}_1, x_1, x_0\right)\right)\right)\right)$$

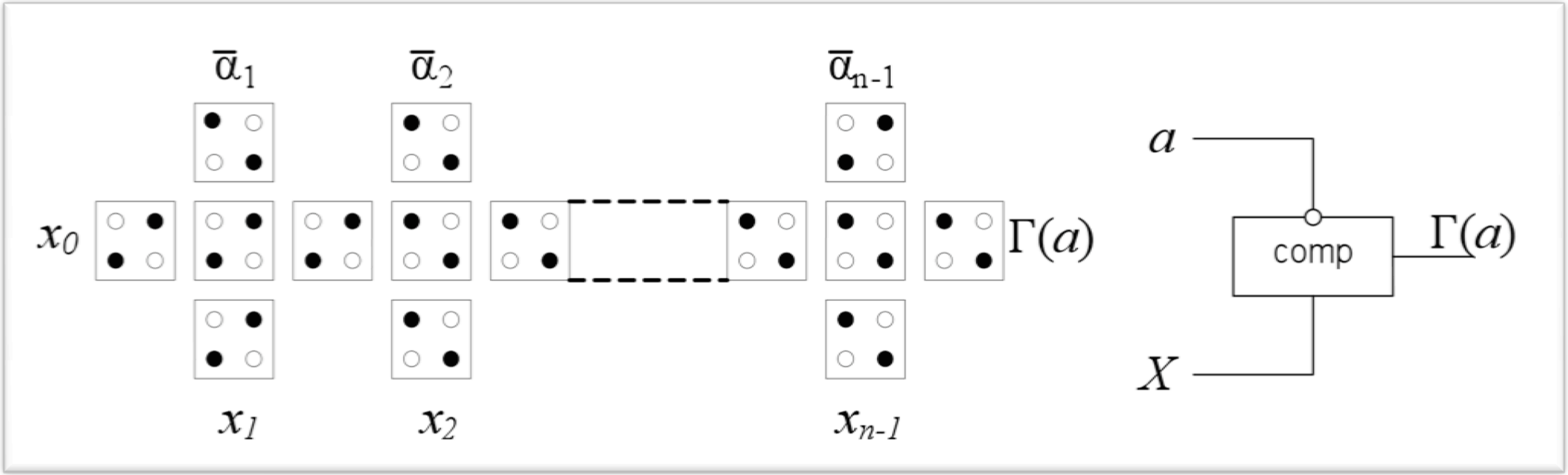
The last expression is used as the basic expansion for the proposed QCA implementation of Boolean functions.

# Majority comparator

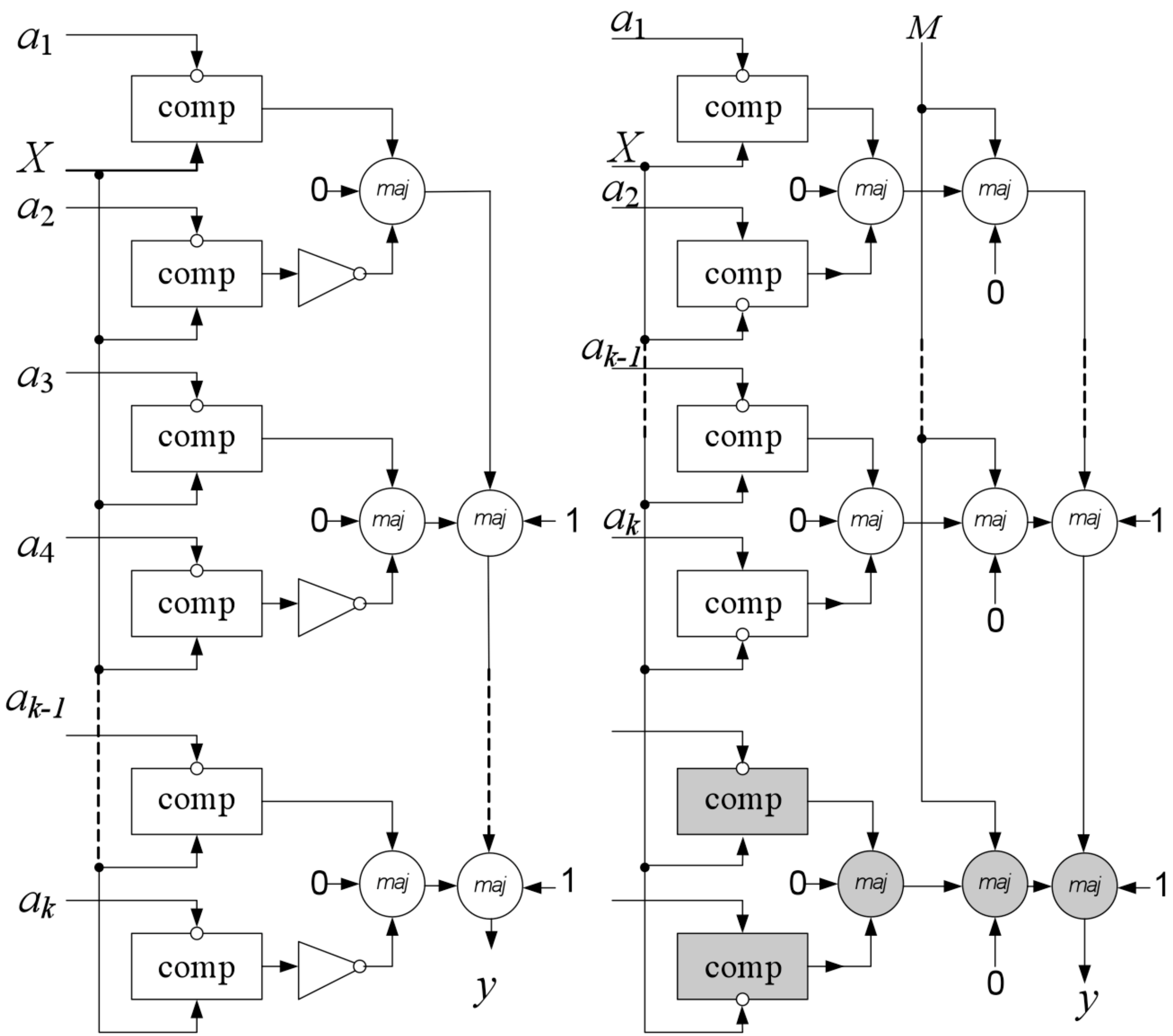
$$\Gamma(X;a) = maj\left(\bar{\alpha}_{n-1}, x_{n-1}, maj\left(\bar{\alpha}_{n-2}, x_{n-2}, maj\left(\ldots maj\left(\bar{\alpha}_1, x_1, x_0\right)\right)\right)\right)$$



# QCA implementation of the comparator



# Programmable Comparator Array



# Conclusions

- A novel universal quantum cellular automata gate - boundary comparator is presented
- A programmable comparator based array (PCA) based on this comparator is presented
- A method for implementation of the QCA schemes by the PCA is developed
- The PCA provides better solution in comparison with conventional methods for considerable number of benchmarks
- The main advantage of the proposed solution is its regularity, which provides the testability, a potential reconfigureability, reparability etc.