



**IRWIN AND JOAN JACOBS  
CENTER FOR COMMUNICATION AND INFORMATION TECHNOLOGIES**

# **Latency Optimized Mapping of Logic Functions for Memristor Aided Logic (MAGIC)**

**Rotem Ben Hur, Nimrod Wald,  
Nishil Talati, and Shahar Kvatinsky**

**CCIT Report #908  
December 2016**

 Electronics  
Computers  
Communications

**THE ANDREW & ERNA VITERBI FACULTY OF ELECTRICAL ENGINEERING  
TECHNION—ISRAEL INSTITUTE OF TECHNOLOGY, HAIFA 3200003, ISRAEL**



# Technical Report – Latency Optimized Mapping of Logic Functions for Memristor Aided Logic (MAGIC)

*Rotem Ben Hur, Nimrod Wald, Nishil Talati, and Shahar Kvatinsky, Member, IEEE*

Andrew and Erna Viterbi Faculty of Electrical Engineering  
Technion – Israel Institute of Technology  
Haifa 3200003, ISRAEL

**Abstract**-This document describes in detail the optimization problem proposed in [1].

## I. LATENCY OPTIMIZATION PROBLEM

The latency optimizing problem of an in-memory Boolean function has two degrees of freedom: the locations of data and the execution time of different logic gates. The following variables are defined for each logic gate:

- $(\{R_{A_j}, C_{A_j}\}, \{R_{B_j}, C_{B_j}\}, \{R_{E_j}, C_{E_j}\})$  - Location (coordinates of memory cells) of the inputs and the output of NOR gate  $j$ .
- $(\{R_{A_k}, C_{A_k}\}, \{R_{E_k}, C_{E_k}\})$  - Location (coordinates of memory cells) of the inputs and the output of NOT gate  $k$ .
- $T_i$  - Clock cycle in which gate  $i$  (either NOT or NOR) is executed.

Therefore, each NOR gate and each NOT gate has 7 and 5 variables, respectively.

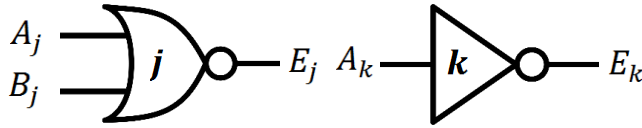


Figure 1 – Inputs and outputs of NOR and NOT gates

The execution of a given Boolean function is finished when the operations of all gates are completed, thus  $\max_j(T_j)$  is the latency of a specific mapping, where  $0 < j \leq \#gates$ . Therefore, the latency (in clock cycles) of the best mapping is the minimum latency out of all different mappings and is

$$Latency_{best\ mapping} = \min\{\max_j T_j\}, \quad (1)$$

$$0 < j \leq \#gates$$

The legal mappings are limited by location, connectivity and timing, and are restricted by the following constraints:

### 1) Location constraints:

- Every input and output (I/O) has to be mapped to a memory cell, thus the coordinates of each I/O are limited by the physical size of the memory.

$$\forall x_j \in \{A_j, B_j, E_j\}: (0 < C_{x_j} \leq Col_{num}) \cap (0 < R_{x_j} \leq Row_{num})$$

- Two different outputs cannot be placed in the same memory cell (whereas the same input may be used for two different gates, therefore their inputs share coordinates). In such a configuration, reusing of a cell (after resetting) is not supported. Note that this constraint is not mandatory and is used to simplify the problem at the cost of potentially using more cells.

$$\forall E_k, E_j: (C_{E_j} \neq C_{E_k}) \cup (R_{E_j} \neq R_{E_k})$$

- I/Os of each gate have to be located in the same row and different columns, or vice versa.  
 $\forall \text{gate } j: \left[ (C_{A_j} = C_{B_j} = C_{E_j}) \cap (R_{A_j} \neq R_{B_j} \neq R_{E_j}) \right] \cup \left[ (C_{A_j} \neq C_{B_j} \neq C_{E_j}) \cap (R_{A_j} = R_{B_j} = R_{E_j}) \right]$
- The execution of different NOR or NOT gates simultaneously is possible only when they are aligned in the rows or in the columns.

$\forall \text{gates } j, k:$

$$T_j \neq T_k \cup \left\{ \left\{ \left( (C_{A_j} = C_{A_k} \cap C_{B_j} = C_{B_k}) \cup (C_{A_j} = C_{B_k} \cap C_{B_j} = C_{A_k}) \right) \cap C_{E_j} = C_{E_k} \right\} \cap (R_{A_j} = R_{B_j} = R_{E_j} \cap R_{A_k} = R_{B_k} = R_{E_k}) \right\} \cup \left\{ \left\{ (R_{A_j} = R_{A_k} \cap R_{B_j} = R_{B_k}) \cup (R_{A_j} = R_{B_k} \cap R_{B_j} = R_{A_k}) \right\} \cap R_{E_j} = R_{E_k} \right\} \cap (C_{A_j} = C_{B_j} = C_{E_j} \cap C_{A_k} = C_{B_k} = C_{E_k}) \right\}$$

### 2) Connectivity constraints:

The following constraint is the only constraint which is determined by the connectivity of the gates, according to the netlist.

- Every output of gate  $h$  that is connected to an input of gate  $j$  has to be mapped to the same memory cell, and the execution of gate  $j$  has to be performed only after the execution of gate  $h$ . This configuration does not support movement of data within the memory array and is not mandatory.

$$\forall E_h, x_j \in \{A_j, B_j\} \text{ that are connected: } \left[ (C_{E_h} = C_{x_j}) \cap (R_{E_h} = R_{x_j}) \right] \cap (T_h < T_j)$$

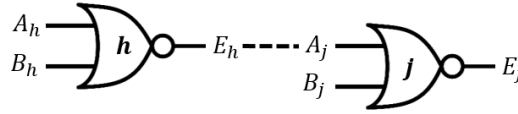


Figure 2 – output of gate  $h$  ( $E_h$ ), is connected to an input of gate  $j$  ( $A_j$ )

### 3) Timing constraints:

- The execution time of each gate is positive  
 $\forall \text{NOT gate } k, \text{ NOR gate } j: T_j, T_k > 0$
- The execution of a NOR gate and a NOT gate have to be done during different clock cycles since their number of inputs is different

$$\forall \text{NOT gate } k, \text{ NOR gate } j: T_j \neq T_k$$

## REFERENCES

- [1] R. Ben Hur, N. Wald, N. Talati, and S. Kvatinsky, “Synthesis and Mapping of Logic Functions for Memristor Aided Logic (MAGIC),” *Proceedings of the International Symposium of Circuits and Systems*, May 2017 [submitted].