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

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

CCIT Report #908 December 2016

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:

- $\left(\left\{R_{A_j}, C_{A_j}\right\}, \left\{R_{B_j}, C_{B_j}\right\}, \left\{R_{E_j}, C_{E_j}\right\}\right)$ - Location (coordinates of memory cells) of the inputs and the output of NOR gate *j*.
- $\left(\left\{R_{A_k}, C_{A_k}\right\}, \left\{R_{E_k}, C_{E_k}\right\}\right)$ - 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.

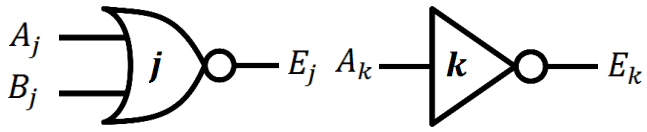

Figure 1 – Inputs and outputs of NOR and NOT gates

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

$$\boldsymbol{Latency_{best\ mapping} = \min_{0 < j \leq \#gates}\left\{\max_j T_j\right\}}, \quad (1)$$

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.

$$\forall \mathbf{x_j} \epsilon \{\mathbf{A_j}, \mathbf{B_j}, \mathbf{E_j}\}: \left(\mathbf{0} < \boldsymbol{C_{x_j}} \leq \boldsymbol{Col_{num}}\right) \cap \left(\mathbf{0} < \boldsymbol{R_{x_j}} \leq \boldsymbol{Row_{num}}\right)$$

- 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.

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

- I/Os of each gate have to be located in the same row and different columns, or vice versa.

$$\forall \boldsymbol{gate\ j}: \left[\left(\boldsymbol{C_{A_j} = C_{B_j} = C_{E_j}}\right) \cap \left(\boldsymbol{R_{A_j} \neq R_{B_j} \neq R_{E_j}}\right)\right] \cup \left[\left(\boldsymbol{C_{A_j} \neq C_{B_j} \neq C_{E_j}}\right) \cap \left(\boldsymbol{R_{A_j} = R_{B_j} = R_{E_j}}\right)\right]$$

- The execution of different NOR or NOT gates simultaneously is possible only when they are aligned in the rows or in the columns.

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

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

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.

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

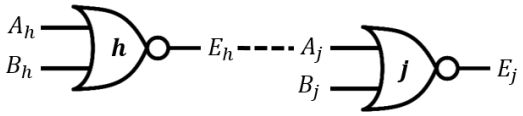

Figure 2 – output of gate *h* (E_h), is connected to an input of gate *j* (A_j)

3) Timing constraints:

- The execution time of each gate is positive

$$\forall \boldsymbol{NOT\ gate\ k, NOR\ gate\ j}: \boldsymbol{T_j, T_k > 0}$$

- 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

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

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].