2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

Mapping of Boolean Logic Functions onto 3D Memristor Crossbar

Naveen Murali G* P. Satya Vardhan* F. Lalchhandama* Kamalika Datta† Indranil Sengupta*
* Indian Institute of Technology, Kharagpur, India
Email: naveenmuralig@iitkgp.ac.in, satyavardhansv@iitkgp.ac.in, dampuiafanchun@iitkgp.ac.in, isg@iitkgp.ac.in
† National Institute of Technology, Meghalaya, India (Email: kdatta@nitm.ac.in)

***Abstract*—A novel 3D memristive crossbar architecture and Boolean logic computation on it is explored in the paper. The two approaches for 2D crossbar mapping- *Compact* and *Parallel mapping schemes*- are extended along with inclusion of possible *parallelism* between gates and the *pipelining* between the different layers is explored. The proposed 3D crossbar mapping schemes are compared in terms of t_{steps} and crossbar size required, and is also compared with the existing works on 2D crossbars for illustrating the possible advantages of 3D crossbars.**

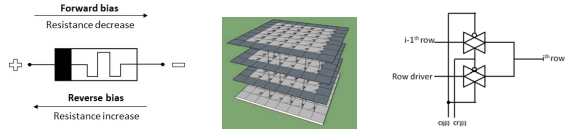
***Keywords*-Memristor, 3D crossbar, parallelism, pipelining**

I. Introduction

Memristors, first postulated by Leon Chua in 1971 [1], are two-terminal passive devices showing resistive switching characteristics depending on the total amount of charge flowing through it. The non-volatile nature and the ability to store the resistance state can be utilized to implement logic functions, as the two resistance states R_{on} and R_{off} can be considered as the two logic states 1 and 0 respectively. The symbolic representation of a memristor is shown in Fig. 1*a*.

Mapping of Boolean functions onto a 3D memristor crossbar have never been explored earlier. The extension of the crossbar structure from a single 2D layer to multiple stacked layers leading to a 3D structure improves the packing density. Also, the presence of multiple layers enables parallel gate operations and also pipelined operation between the layers.

In this paper, a novel 3D crossbar architecture is proposed and its effectiveness is demonstrated by mapping some basic arithmetic circuits like ripple carry adder (RCA) onto it. The rest of the paper is organized as follows: Section II discusses the proposed work, Section III presents the comparison results and the conclusion of the paper.



(a) (b) (c)

Fig. 1: (a) Memristor representation (b) Layout of a 3D memristor crossbar, (c) Multiplexer for inter-layer connections

II. Proposed work

A. Proposed 3D crossbar architecture

The proposed 3D crossbar structure (Fig. 1*a*) consists of multiple layers of 2D crossbars stacked on top of each other. The different layers of the 3D crossbar are interconnected by the driver circuit shown in Fig. 1*c*. The interconnection between the columns is not considered for now.

B. Parallel mapping of ripple-carry adder (RCA)

The parallel mapping approach discussed in [2] is extended here for 3D crossbar mapping of netlists, and is illustrated by mapping an N-bit RCA onto an N-layer 3D crossbar. The scheme has the advantage of having fixed sequence of micro-operations for any given netlists, and the scheduling of gates (RC or ASAP scheduling) in a given netlist can reduce the crossbar size and time-steps (t_{steps}) required for evaluation.

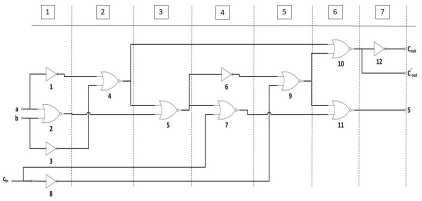


Fig. 2: ASAP schedule of full adder with G_{max} = 4

1) Non-pipelined structure: The non-pipelined implementation of a 3-bit RCA is illustrated in Fig. 3. Each layer consists of a full adder with 12 gates (numbered 1 to 12), with gates in each layer being evaluated using ASAP scheduling.

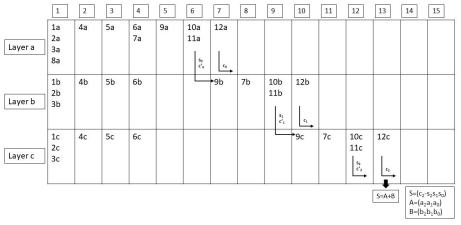


Fig. 3: 3D crossbar mapping of a 3-bit RCA

This requires only 13 levels for evaluation in contrast to the approach of [2] (15 levels). A comparison with respect to

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00106

number of levels, t_{steps}, and crossbar size is shown in Table I. Note that the crossbar size per-layer could be made 3×3 by following RC scheduling with G_{max} = 3.

2) Pipelined structure: The concept of bit-level pipelining can be realized on the same crossbar to improve throughput and t_{steps} for multiple sets of inputs. This is illustrated in Fig. 4 for $N = 3$ and $D = 2$ input sets (A, B) and (X, Y).

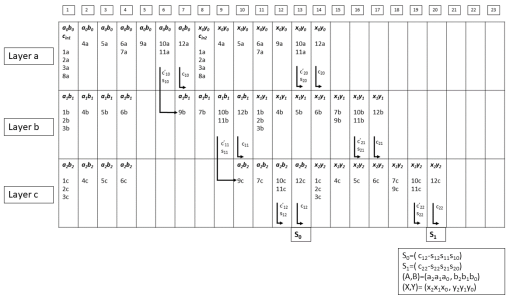


Fig. 4: Illustration of 3D pipelining

After the evaluation of (a_0, b_0) from layer-a at level-7, the next set of bits (x_0, y_0) are evaluated while the carry c_{10} from layer-a is propagated to layer-b evaluating the sum and carry bits (s_{11}, c_{11}) of bits (a_1, b_1), and so on. Output $S_0 = A + B$ is obtained at level-13, and $S_1 = X + Y$ at level-20,and so on. One output is obtained after every 7 levels or 7*5= 35 t_{steps}. The performance summary is shown in Table I for N-bit 3D RCA (pipelined & Non-pipelined) with D sets of inputs.

TABLE I: Comparison for parallel mapping implementation

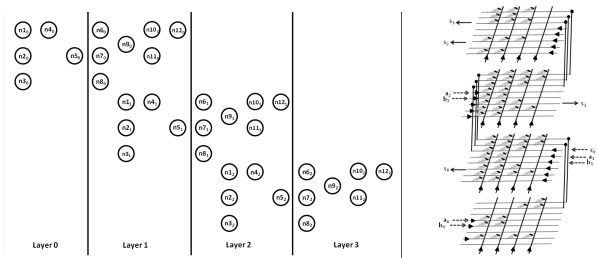
	As in [2]	3D Non-Pipelined	3D Pipelined
No. of levels	$4N + 3$	$3N + 4$	$3(N-1) + 7D$
Time-steps	$20N + 15$	$15N + 20$	$15(N-1) + 35D$
Throughput	$20N + 15$	$15N + 20$	35 t_{steps}

C. Compact Mapping of RCA onto 3D crossbar

Compact mapping approach (see [3]) has been extended as it results in fewer time steps.

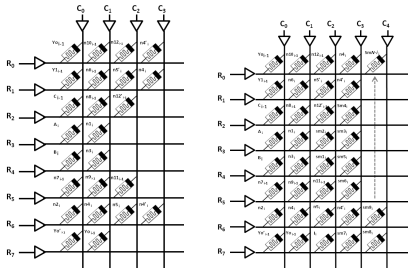
1) Non-pipelined structure: If the gates are mapped as shown in Fig. 6*a*, where Yo and Y1 are the outputs of gates n4, n5 of *i-1th* full adder, more parallel operations within and across the layers is possible. The black dots in Fig. 5*b* represent the circuit in Fig. 1*c*. The evaluation is carried out as follows: All the input bits A_i, B_i and C_0 are loaded(3 t_{steps}), gates from $n1_i$ to $n5'_i$ are evaluated simultaneously in all layers(8 t_{steps}), the remaining gates in all layers are evaluated sequentially and necessary bits are transferred across the layers when required((10+2)N t_{steps}). So the total number of t_{steps} required will be $12N + 11$, and the crossbar size will be 8×4.

2) Pipelined structure: From Fig. 6*b* if data are fed serially via the memristor $n11_{i-1}$, actual and inverted bits of the serial data (successive sum bits) are stored in *sm2, sm4, sm6,* etc. and *sm1, sm3, sm5,* etc. respectively. Similarly by pushing serial data via I_i, the reverse occurs. The layer-0 and layer-N of pipelined version are identical to that of N-bit RCA. The input bits are loaded appropriately, the evaluation of all the gates in all the layers is performed in parallel including storing successive sum bits and required bits are transferred across adjacent layers appropriately. For *approach-1* and *approach-2*, where storing the sum bits within the crossbar is considered and not considered respectively, latency can be estimated to be $N(20 + 2N) + 13$ and $18N + 13$ t_{steps} and throughput to be $(20 + 2N)$ and 18 t_{steps} respectively.



(a) (b)

Fig. 5: (a) Layer-wise netlist for 3-bit RCA, (b) 3D crossbar architecture for 3-bit RCA



(a) (b)

Fig. 6: (a) Compact mapping of layer-i of RCA, (b) Compact mapping of layer-i of pipelined RCA

III. Results and Conclusion

Table II shows that parallel mapping and compact mapping approaches provides area efficient and time efficient implementation respectively. Simulation of the proposed architectures along with row and column drivers, and also the mapping and evaluation of a general net-list onto the 3D crossbar can be considered as a future extension of the work.

TABLE II: Comparison of 8-bit RCA implementation

	Parallel Mapping		Compact Mapping	
Arithmetic block (with $N = D = 8$)	Time Steps	Crossbar Size	Time Steps	Crossbar Size
RCA of [2]	1400	4×3 / 10×3	-	-
Non-pipelined RCA	1120	4×3 / 3×3	856	8×4
Pipelined RCA	385	4×3 / 3×3	283	8×4
No.of layers	N		N+1	

References

[1] L. Chua. Memristor – the missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, September 1971.

[2] P. L.Thangkhiew, R. Gharpinde, V. C. Paturi, K. Datta, and I. Sengupta. Area efficient implementation of ripple carry adder using memristor crossbar arrays. In *11th Intl. Design and Test Symposium (IDT)*, pages 142–147, 2011.

[3] R. Gharpinde, P. L. Thangkhiew, K. Datta, and I. Sengupta. A scalable in-memory logic synthesis approach using memristor crossbar. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(2):355–366, February 2018.