ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.12, No.8, 1–6

# A signal degradation reduction method for memristor ratioed logic (MRL) gates

**Bosheng Liu**[1,2a)], **Ying Wang**[1,2b)], **Zhiqiang You**[3], **Yinhe Han**[1,2c)], **and Xiaowei Li**[1,2d)]

[1] *State Key Lab of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, China*

[2] *University of Chinese Academy of Sciences, Beijing, China*

[3] *College of Computer Science and Electronic Engineering, Hunan University, China*

a) *liubosheng@ict.ac.cn*

b) *wangying2009@ict.ac.cn*

c) *yinhes@ict.ac.cn*

d) *lxw@ict.ac.cn*

**Abstract:** This paper presents a design strategy of eliminating signal degradation for memristor ratioed logic (MRL) gates. Based on the strategy, a novel MRL-based one-bit full adder is proposed. The inverters in circuit can effectively eliminate the degradation and restore signal integrity. To evaluate the effectiveness of the proposed one-bit full adder, an eight-bit full adder is demonstrated as a study case. Compared to the previous MRL-based standard cell design, the proposed circuit can reduce 11.1% memristor cells, 22.2% CMOS transistors, 38.9% vias, 58% power. Compared to the previous MRL-based optimized design, the proposed design can reduce 11.1% memristor cells, 12.5% CMOS transistors, 98.1% power, 98.1% energy.

**Keywords:** full adder, memristor ratioed logic (MRL) gate

**Classification:** Integrated circuits

## References

[1] A. C. Torrezan, J. P. Strachan, G. Medeiros-Ribeiro and R. S. Williams: Nanotechnology **22** (2011) 485203. DOI:10.1088/0957-4484/22/48/485203

[2] A. Jog, A. K. Mishra, C. Xu, Y. Xie, V. Narayanan, R. Iyer and C. R. Das: Design Automat. Conf. (2012) 243. DOI:10.1145/2228360.2228406

[3] K. Schuegraf, M. C. Abraham, A. Brand, M. Naik and R. Thakur: IEEE J. Electron Devices Soc. **1** (2013) 66. DOI:10.1109/JEDS.2013.2271582

[4] E. Linn, R. Rosezin, S. Tappertzhofen, U. Bottger and R. Waser: Nanotechnology **23** (2012) 305205. DOI:10.1088/0957-4484/23/30/305205

[5] B. Liu, Z. You, X. Li, J. Kuang and Z. Qin: IEICE Electron. Express **10** (2013) 20130369. DOI:10.1587/elex.10.20130369

[6] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart and R. S. Williams: Nature **464** (2010) 873. DOI:10.1038/nature08940

[7] E. Lehtonen, J. Poikonen and M. Laiho: IEEE Int. Symp. on Circuits and



DOI: 10.1587/elex.12.20150062
Received January 19, 2015
Accepted March 24, 2015
Publicized April 8, 2015
Copyedited April 25, 2015

Systems (2012) 2441. DOI:10.1109/ISCAS.2012.6271792
[8] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny and U. C. Weiser: IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **22** (2013) 2054. DOI:10.1109/TVLSI.2013.2282132
[9] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser and E. G. Friedman: Proc. Int. Cellular Nanoscale Netw. Appl. (2012) 1. DOI:10.1109/CNNA.2012.6331426
[10] S. Kvatinsky, E. G. Friedman, A. Kolodny and U. C. Weiser: IEEE Trans. Circuits Syst. I, Reg. Papers **60** (2013) 211. DOI:10.1109/TCSI.2012.2215714

## 1 Introduction

Memristor has been considered as a promising candidate for memory designs in either server-scale or embedded computing systems due to its non-volatility, high-density, low leakage, low power and CMOS-compatible features [1, 2, 3].

A memristor, which stays either in high resistance state (HRS) or low resistance state (LRS), is suitable for logic operation [4, 5]. There are mainly two logic families on memristor-based logic designs. One is the IMPLY logic family. The memristors in these designs store logic values and/or perform logical operation in sequence [6, 7, 8]. In [8], an eight-bit full add operation can be fulfilled with 232 cycles in serial or 58 cycles in parallel. The other is the memristor ratioed logic (MRL) family. The MRL-based family is a typical hybrid memristor-CMOS logic design [9]. In a MRL design, both AND and OR logic gates consist of two antiserial memristors. Likewise, NAND and NOR gates are realized by adding an inverter at each output of the AND and OR logics.

The designs in IMPLY logic family or MRL family can take full advantage of the memristors. However, these designs with IMPLY logic family always suffer from complex sequential operations. Although MRL-based circuits are free from the additional overhead of sequential operation, they suffer from severe signal degradation issue, potentially leading to erroneous output. For instance, the degradation at the output comes up to 15% for an XOR logic [9].

In this paper, we present a novel and cost-effective MRL-based synthesize strategy to mitigate signal degradation in circuit designs. Based on the synthesis algorithm, a novel MRL-based one-bit full adder design is proposed. To evaluate the effectiveness of the proposed design, an eight-bit full adder composed by the proposed one-bit full adders is demonstrated for case study. Simulation results show that the proposed design can gain more profits than the previous designs.

## 2 Backgrounds and preliminaries

Fig. 1 depicts memristor ratioed logic (MRL) gates [9]. $V_A$, $V_B$ are inputs, while $V_{out}$ is an output. Two memristors, $m_1$ and $m_2$, are connected in opposite polarity directions. The thick black lines in $m_1$ and $m_2$ represent the polarity of the device. In an ideal case, the resistances of $m_1$ and $m_2$ satisfy $R_{LRS,m1} = R_{LRS,m2}$, $R_{HRS,m1} = R_{HRS,m2}$, and $R_{HRS} \gg R_{LRS}$. Fig. 1(a) shows an AND logic gate and its schematic. When $V_A$ equals to "1" and $V_B$ = "0", respectively, $m_1$ turns to HRS and


DOI: 10.1587/elex.12.20150062
Received January 19, 2015
Accepted March 24, 2015
Publicized April 8, 2015
Copyedited April 25, 2015

$m_2$ changes to LRS. Thus, $V_{out}$ equals to logic "0". Likewise, when $V_A$ = "0", $V_B$ = "1", $m_1$ changes to LRS and $m_2$ turns to HRS. Thus, $V_{out}$ = "0". Whereas, when the two inputs are in the same potential level, both "0s" or "1s", $V_{out}$ is "0" or "1" respectively. $V_{out}$ is the logic of ($V_A$ AND $V_B$).

In Fig. 1(b), two antiserial memristors are connected in opposite to Fig. 1(a). When $V_A$ = "1", $V_B$ = "0", $m_1$ turns to LRS, while $m_2$ changes to HRS. The output is logic "1". When $V_A$ = "0", $V_B$ = "1", $m_1$ changes to HRS and $m_2$ turns to LRS. The output is also "1". However, the $V_{out}$ is "0" when $V_A$ = "0" and $V_B$ = "0". Whereas the output is "1" when both inputs equals to "1". The output logic is ($V_A$ OR $V_B$). In Fig. 1(c), the NAND gate is realized by adding an inverter to the output of the two antiserial memristors that has AND logic. The NOR gate in Fig. 1(d) are realized in a similar way. The MRL-based NAND and NOR gates can be considered as the standard cells to replace the CMOS NAND and NOR gates in with less area consumptions.

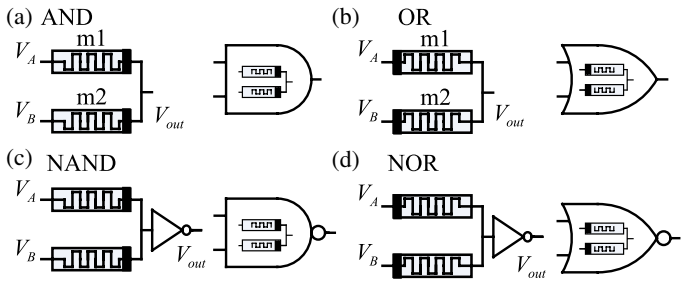


**Fig. 1.** MRL-based logic gates [9]. The AND and OR logic gates are realized with two memristive devices which are connected with different opposite polarities. The NAND and NOR gates are realized by connecting the AND and OR logic gates to an inverter respectively. (a) AND logic. (b) OR logic. (c) NAND gate. (d) NOR gate.

## 3 Designs and case study

### 3.1 Degradation-eliminating algorithm for MRL logic gates

Algorithm 1 depicts the synthesize process to generate a feasible design with less CMOS transistors for a given function $f(\cdot)$. The synthesis are realized in Design Compiler of Synopsis. $\Delta$ is a descent rate of area between 0 and 1. $num_{expected}$ is a supposed max amount of CMOS transistors used in circuit. $area_{AND,MRL}$ and $area_{OR,MRL}$ mean a preference of MRL-based AND and OR logic gates in design. A lower value of $area_{AND,MRL}$ and $area_{OR,MRL}$ in circuit can take more full advantage of the memristors. However, the signal degradation in circuit probably requires more relay buffers to amplify signals, which turns out more areas and power consumptions in turn. We use this line search algorithm to find an optional design with less CMOS transistors for a given logic function $f(\cdot)$. The step 3 in algorithm 1 connects every output of $f(\cdot)$ with an inverter, which can effectively eliminate the degradation and perform signal restoration.


DOI: 10.1587/elex.12.20150062
Received January 19, 2015
Accepted March 24, 2015
Publicized April 8, 2015
Copyedited April 25, 2015

### 3.2 Full adder design and case study

Fig. 2 depicts the proposed one-bit full adder obtained by the synthesis algorithm 1. The $num_{expected}$ we set is 16, which is the average number of the lowest cost of CMOS transistors in [9]. In this circuit, voltage inputs are $V_A$, $V_B$, and $V_{Cin}$. Where $V_{Cin}$ is a carry bit from previous stage. Outputs, $V_S$ and $V_{Cout}$, are sum bit and carry bit respectively. The outputs of the circuit are connected with inverters U13 and U14, which can effectively eliminate the degradation and perform signal restoration on its cascaded circuits. Moreover, the inverters U10, U13, U14, and the inverter in the MRL-based NOR gate U15 can eliminate the direct memristors connections among different MRL-based logic gates to reduce signal degradation.

The simulation is realized in LTspice. Each inverter in Fig. 2 consists of a P-channel Si1013 and an N-channel Si1555DL_N transistors. A TEAM model with Kvatinsky window in [10] is used to simulate a memristor with the parameters $R_{off} = 100000$, $R_{on} = 1000$, $i_{off} = 0.2\text{e-}6$, $i_{on} = -0.15\text{e-}6$, $K_{on} = -0.1$, $K_{off} = 0.1$, $Alpha_{on} = 3$, $Alpha_{off} = 3$, $x_{on} = 0$, $D = 3\text{e-}9$, $x_{off} = 3\text{e-}9$, $x_c = 107\text{e-}12$, $a_{on} = 0.1\text{e-}9$, $a_{off} = 2.9\text{e-}9$, $init\_state = 0.5$, $IV\_relation = 0$.

**Algorithm 1** Degradation-eliminating MRL-based logic search designs

**Input:**

$f(\cdot)$, CMOS standard cell library, $\Delta$, $num_{expected}$;

**Output:**

optional $f(\cdot)$

1: modify the standard cell library, set:
   $\text{area}_{\text{AND,MRL}} = \text{area}_{\text{OR,MRL}}$, $\text{area}_{\text{NAND,MRL}} = \text{area}_{\text{NOR,MRL}}$;
   where $\text{area}_{\text{AND,MRL}} = \text{area}_{\text{NAND,MRL}} = \text{area}_{\text{INV}}$
2: modify the leakage power of MRL gates
3: set $f'(\cdot) = NOT(f(\cdot))$
4: **repeat**
5: rebuild the standard cell library
6: load the standard cell library, $f'(\cdot)$
7: set attributes to only use the MRL and INV gates
8: synthesis $f'(\cdot) \rightarrow f''(\cdot)$
9: optional $f(\cdot) = NOT(f''(\cdot))$
10: count CMOS transistors $num_{optional\ f(\cdot)}$
11: set $\text{area}_{\text{AND,MRL}} = \text{area}_{\text{OR,MRL}} = (1 - \Delta) \times \text{area}_{\text{OR,MRL}}$
12: **until** $(num_{optional\ f(\cdot)} < num_{expected}) \parallel (\text{area}_{\text{AND,MRL}} < 0)$

Fig. 3 depicts the function simulation results of a one-bit full adder. Input voltages, $V_{Cin}$, $V_A$, and $V_B$, whose amplitudes equal to 3.0 V and periods are 6 ms. $V_S$ and $V_{Cout}$ are voltage outputs of sum bit and carry bit respectively. The supply voltage of every inverter is also 3.0 V. The output of $V_S$ is the logic of ($V_{Cin}$ XOR $V_A$ XOR $V_B$). Meanwhile, $V_{Cout}$ is the logic of $(V_A \cdot V_B + V_{Cin} \cdot (V_A \text{ XOR } V_B))$. The voltage reaches 3.0 V when the output logic is "1". To evaluate the effectiveness of the proposed one-bit full adder, we use an eight-bit full adder which is composed by the proposed one-bit full adder as the study case.


DOI: 10.1587/elex.12.20150062
Received January 19, 2015
Accepted March 24, 2015
Publicized April 8, 2015
Copyedited April 25, 2015

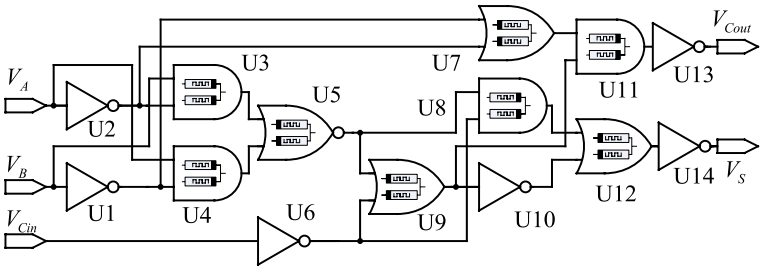


**Fig. 2.** One-bit full adder design. The output logics as $V_{out} = (V_A \cdot V_B + C_{in} \cdot (V_A \text{ XOR } V_B))$, $V_S = ((V_A \text{ XOR } V_B) \text{ XOR } V_{Cin})$, where $V_{Cin}$ is a bit carried in from the previous less significant stage.

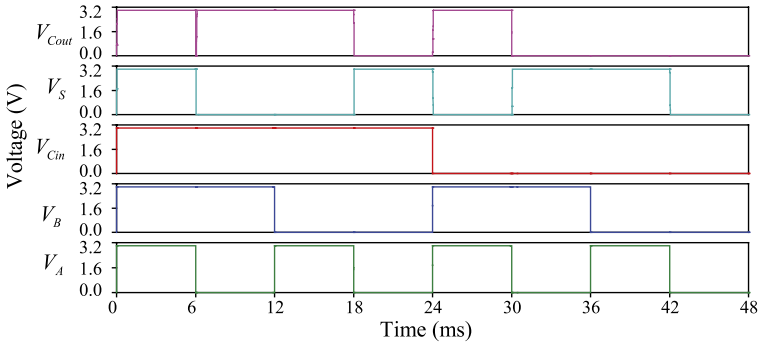


**Fig. 3.** Simulation results of a one-bit full adder.

Table I shows the comparison of different designs. The eight-bit full adder with the IMPLY methodology in [8] are realized in parallel. Both the previous works of standard cell design and the optimized design with parameter 3 in [9] are MRL-based logic family designs. The previous optimized design acquires the lowest number of CMOS transistors. Compared to the IMPLY methodology, the proposed design loss the advantages in the numbers of memristors, CMOS transistors, and vias. However, the performance of the IMPLY-based design are 72 cycles, but the proposed design only takes 1 cycle to generate formula. This also means the proposed design does not need a complex sequential control logic to perform the correct logic function. Compared to the previous standard cell design, the proposed design reduce the consumptions of memristor amount by 11.1%, the CMOS transistor by 22.2%, the via by 38.9%. Compared to the previous optimized design, the proposed architecture reduce the memristor count by 11.1%, the CMOS transistor by 12.5% at the expenses of 10% more vias.

**Table I.** Comparisons of eight-bit full adder designs

| | IMP methodology in [8] | standard cell design [9] | optimized design with parameter set 3 [9] | proposed |
|---|---|---|---|---|
| memristors | 58 | 144 | 144 | 128 |
| CMOS transistors | 0 | 144 | 128 | 112 |
| vias | 0 | 144 | 80 | 88 |
| cycles | 72 | 1 | 1 | 1 |


DOI: 10.1587/elex.12.20150062
Received January 19, 2015
Accepted March 24, 2015
Publicized April 8, 2015
Copyedited April 25, 2015

**Table II.** Power and energy comparisons of eight-bit full adder designs

| | Average power [normalized] | Total energy [normalized] |
|---|---|---|
| standard cell design [9] | 1 | 1 |
| parameter set 3 [9] | 22.5 | 167.9 |
| proposed | 0.42 | 3.16 |

Table II shows the power and energy comparisons of different designs. Our method reduces the power by 58%, compared to the standard cell design. With less inverters as the supply voltage sources than the standard cell design, the proposed design takes more time in memristor state exchange, which turns out more energy consumption. The proposed design can improve logic flexibility at the expense of 3.16× of energy. Compared to the previous optimized design with parameter 3, the proposed design can effectively eliminate the signal degradation with only 1.9% of its power or energy.

## 4 Conclusion

In this work, we proposed a strategy of reducing signal degradation for MRL-based logic designs. Based on that, we presented a one-bit full adder design. An eight-bit full adder is used in the case to study the effectiveness of the proposed architecture. Contrasted to the IMPLY logic family, the proposed design is a high performance combinational logic. Compared to the previous MRL-based standard cell design, the proposed design provides higher logic flexibility and only requires 11.1% fewer memristors, 22.2% fewer CMOS gates, 38.9% vias, 58% less power at the expense of 3.16× energy. Compared to the previous MRL-based optimized design, the proposed design save the amount of memristors by 11.1%, the CMOS transistors by 12.5%, both the power and energy by 98.1% at the expenses of 10% more vias.

## Acknowledgments

The work was supported in part by National Basic Research Program of China (973) under grant No. 2011CB302503, in part by National Natural Science Foundation of China (NSFC) under grant No. 6110001661221062).


DOI: 10.1587/elex.12.20150062
Received January 19, 2015
Accepted March 24, 2015
Publicized April 8, 2015
Copyedited April 25, 2015