

# 基于 RRAM 的运用 MIG 逻辑设计的加法器电路

刘文楷<sup>1</sup>, 范冬宇<sup>1</sup>, 戴 澜<sup>1</sup>, 张 锋<sup>2</sup>

(1 北方工业大学 电子信息与工程学院, 北京 100144; 2 中国科学院 微电子研究所 微电子重点研究室, 北京 100029)

**摘 要:** 本文采用 SMIC 0.18  $\mu\text{m}$  CMOS 工艺, 设计了一种基于阻变随机存储器(RRAM)的加法器电路. 区别于传统的蕴含逻辑(IMP)型加法器, 本设计选取新型的多数表决器逻辑(Majority-Inverter Graph, MIG)实现加法运算. 文中设计并验证了基于 MIG 逻辑的一位全加器的全部运算, 证明了新逻辑在存储器计算方面的可行性. 与传统 RRAM 一位全加器相比较, 本设计的一位全加器因为采用新的逻辑方法使得 RRAM 多个单元可以并行进行数据操作, 从而减少了运算步骤, 及原有冗长运算引起的误差, 有效提高了运算速度, 为存储器内部计算提供了新的思路.

**关键词:** 阻变随机存储器; 多数表决器逻辑; 存储器计算; 加法器

**中图分类号:** TN431.2

**文献标识码:** A

**文章编号:** 1000-7180(2017)12-0050-05

## Design of Adders for RRAM-based In-memory Computing Using Majority-Inverter Graphs

LIU Wen-kai<sup>1</sup>, FAN Dong-yu<sup>1</sup>, DAI Lan<sup>1</sup>, ZHANG Feng<sup>2</sup>

(1 School of Electronic Information and Engineering, North China University of Technology,  
Beijing 100144, China; 2 Key Laboratory of Microelectronics Devices & Integrated Technology,  
Institute of Microelectronics Chinese Academy of Sciences, Beijing 100029, China)

**Abstract:** Implemented with 0.18  $\mu\text{m}$  CMOS process, an adder circuit based on RRAM was presented. Different from the traditional RRAM adder based on implication operation, in the paper, we propose an emerging approach for the synthesis of RRAM-based adders using novel Majority-Inverter Graphs (MIGs). Experimental results show a full adder using MIG can achieve the whole function. Compared with the full adder based on implication, it reduces the number of computing steps obviously by parallel operation among several RRAM cells which lowers the error and improves the computing speed at the same time. This design also provides a new way of in-memory computing.

**Key words:** RRAM; MIG; In-memory computing; full adders

### 1 引言

存储器设备是集成电路产业中最为重要的技术, 在传统的冯·诺依曼计算机体系结构中, 计算机的处理设备(CPU)与存储设备是分开的, 处理器与存储器通过一个数据通道连接, 即总线. 而这种结构也有着它的弊端, 虽然随着技术的发展, 处理器的处理速度不断提高, 存储器的存储量不断增大, 但是计算机的处理速度并没有得到很大的提升, 其原因在

于总线的容量决定了在任意时刻计算机能处理数据量的极限, 这也被称作“冯·诺依曼瓶颈”<sup>[1-2]</sup>. 而要解决这一问题的方法之一就是需要把处理器与存储器结合在一起.

阻变随机存储器(RRAM)是一种新型的具有发展潜力的存储器件, 由它构成的存储器不仅具有优秀的存储功能, 同时具备用它进行完备的布尔逻辑计算的能力, 即用高低阻值来代表不同的逻辑值 0 或 1(在本文中高阻值一律用“0”表示, 低阻值用

“1”表示),这样便实现了存储与计算在同一个器件上的统一.用 RRAM 作为计算机的核心,将存储和计算结合在同一个器件上,打破了经典冯·诺依曼计算机体系结构的弊端,为未来新型计算机提供了新的发展思路.

## 2 基于 RRAM 运用 MIG 逻辑设计加法器的背景和基本原理

### 2.1 电路设计的背景

阻变存储器(RRAM)不仅有很好的存储特性,而且还可以做逻辑计算.惠普实验室在 2010 年使用 RRAM 器件进行逻辑蕴含(IMP)操作实现了逻辑运算<sup>[3]</sup>,不同于传统的电压或者电量表示状态逻辑值,RRAM 的逻辑计算使用阻值的变换来表示的,这样做的优点是当外加电压结束后计算结果依旧可以保持在 RRAM 中,即没有其他操作电压施加在 RRAM 上时阻值不变,这个特性又称为非易失性.所以用 RRAM 进行逻辑计算实现了将存储和计算的有机统一,大大拓宽了 RRAM 的研究范围.但是基于蕴含逻辑(IMP)的 RRAM 计算因为需要的 RRAM 单元较多,而且操作步骤复杂,其结果是会放大每一步的误差,所以本文介绍了一种基于新型的布尔逻辑优化逻辑表示结构——MIG,基于 MIG 的 RRAM 逻辑运算方法可以有效的简化运算步骤和所需的 RRAM 单元.

### 2.2 多数表决器逻辑(MIG)定义

Majority-Inverter Graph (MIG)是一种新型的数据逻辑表示结构<sup>[4]</sup>,它是由多数门逻辑(the majority gate, MAJ)和取反逻辑组成的.其公式如(1)所示:

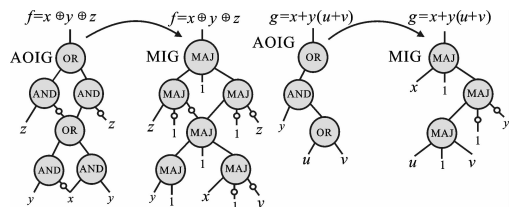
$$M(x, y, z) = x \cdot y + x \cdot z + y \cdot z \quad (1)$$

与传统的数据结构,如二元决策图(Binary Decision Diagrams, BDDs)和“与非”图表法(And-Inverter Graphs, AIGs)相比,在逻辑优化方面 MIG 具有更好的实验结果,特别是在传播延迟优化方面<sup>[4]</sup>.更重要的是, MIG 与基于 RRAM 的电路逻辑综合有很高的匹配性,这完全是因为在 RRAM 中可以内置一个用电阻来表示的多数门逻辑(the majority gate).

### 2.3 多数门逻辑(MAJ)

如式(1)所示, MAJ 逻辑的结果取决于三个输入  $x, y, z$ ,而通过控制三个输入中的其中一个输入的“0”或“1”可以实现另外两个输入的“与”操作或“或”操作,例如当  $z$  取“0”时,  $M(x, y, z)$  的结果实际

上是  $x, y$  进行“与”操作的结果逻辑值;当  $z$  取“1”时,  $M(x, y, z)$  的结果实际



(a)公式(2)的表示图 (b)公式(3)的表示图

图1 AOIG表示与 MIG表示的区别

上是  $x, y$  进行“或”操作的结果逻辑值.所以 MAJ 逻辑可以代替传统的“与”、“或”逻辑.再加上取反的操作,组成 MIG 逻辑可以完全取代传统的“与”,“或”,“非”图式法(AND/OR/Inverter Graphs, AOIGs).例如图1所示,分别对比了 MIGs 和 AOIGs 表示公式(2)和(3).

$$f = x \oplus y \oplus z \quad (2)$$

$$g = x + y(u + v) \quad (3)$$

### 2.4 多数门逻辑(MAJ)在 RRAM 中的实现

阻变存储器(RRAM)是一个双端器件,其内部的电阻可以在高低阻态之间进行转变,若将高阻态定义为逻辑“0”,低阻态定义为逻辑“1”,则 RRAM 可以进行逻辑值的转换.如果指明上电极逻辑为  $P$ ,下电极逻辑为  $Q$ ,施加在 RRAM 器件两端的操作电压表示为  $V_{PQ}$ .假设电压  $V_{SET}, V_{CLEAR}, V_{COND}$  (不会改变 RRAM 阻态的电压)分别对应逻辑值为  $(P=1, Q=0), (P=0, Q=1)$  和  $(P=Q)$ ,则 RRAM 下一时刻的阻值( $R'$ )与逻辑值  $P, Q$  和此刻阻值  $R$  的转换关系真值表如图2所示<sup>[5]</sup>.图2所示 MAJ 操作的真值表可以表示为式:

$$\begin{aligned} R' &= \text{MAJ}(P, Q, R) = (P \cdot \bar{Q}) \cdot \bar{R} + (P + \bar{Q}) \cdot R \\ &= P \cdot R + \bar{Q} \cdot R + P \cdot \bar{Q} \cdot \bar{R} \\ &= P \cdot R + \bar{Q} \cdot R + P \cdot \bar{Q} \cdot R + P \cdot \bar{Q} \cdot \bar{R} \\ &= P \cdot R + \bar{Q} \cdot R + P \cdot \bar{Q} \\ &= M(P, \bar{Q}, R) \end{aligned} \quad (4)$$

由此可见,如果将  $Q$  取反,只要知道 RRAM 此时此刻的阻态,通过控制施加在器件两端的电压,就可以用一个 RRAM 单元很容易地实现一个三个输入变量的 MAJ 逻辑门操作.

### 2.5 基于 RRAM 的 MIG 逻辑操作

既然一个 RRAM 单元就可以当作 MAJ 门来应用,那么要完成一个三输入的 MIG 逻辑操作,即

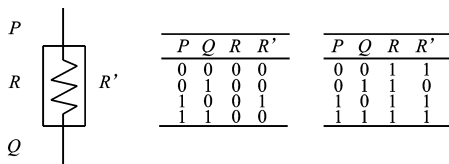


图2 RRAM 单元固有的 MAJ 操作

$M(x, y, z)$  的具体方法如下。

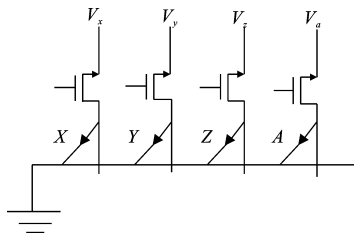


图3 基于 MAJ 操作完成一个 MIG 逻辑电路图

图3所示阻变存储器 X, Y, Z 分别是输入单元, 存储三个输入变量  $x, y, z$ , 而阻变存储器 A 作为工作单元, 存储 MIG 逻辑计算的中间值, 阻变存储器 Z 同时还作为输出单元, 在 MIG 逻辑计算的最后存储计算所得的结果。具体步骤如下:

01:  $X = x, Y = y, Z = z, A = 0$

02:  $P_A = 1, Q_A = y, R_A = 0 \rightarrow R'_A = \bar{y}$

03:  $P_Z = x, Q_Z = \bar{y}, R_Z = z \rightarrow R'_Z = M(x, y, z)$

第一步, 为写入, 将  $x, y, z$  写入输入 RRAM 单元中, 工作 RRAM 单元 Z 通过  $V_{\text{CLEAR}}$  置为高阻态 (逻辑“0”) 为存储计算中间值做准备; 第二步, 通过 MAJ 门逻辑将  $y$  的反存入 A 单元中。具体操作如下, 在阻变存储器 A 单元上施加逻辑值为 ( $P=1, Q=\sim y$ ) 对应的电压 (电压对应逻辑参见本章第二小节), 原本逻辑值为“0” (高阻态) 的 A 阻变存储器, 下一时刻会变为与输入  $y$  逻辑值相反的逻辑值。最后在 Z 单元上施加逻辑值为 ( $P=x, Q=\sim y$ ) 所对应的电压, 根据 MAJ 门逻辑, 原本逻辑值为  $z$  的输入单元, 下一时刻会变为  $M(x, y, z)$  计算结果的逻辑值。这样仅用三步就完成了 一个 MIG 逻辑的计算, 对比用传统的蕴含逻辑 (IMP) 设计的计算 MIG 逻辑, 基于 MAJ 门逻辑的设计在操作步骤和所参加运算的 RRAM 单元两方面都要更方便快捷。基于 IMP 操作的 MIG 电路的步骤需要 10 个步骤, 且参与操作的 RRAM 单元将达到 6 个<sup>[5]</sup>。

### 3 基于 MIG 逻辑的加法器高效设计

#### 3.1 一位全加器的设计思路

加法器的设计目的是减少运算所占用的时间和

空间的复杂度<sup>[6]</sup>, 同时要提高计算的可靠性, 所以尽量少的步骤是使步骤之间误差最小化的最优方法。首先以一个一位全加器为例,  $x, y$  和  $c$  为输入, 其中  $x, y$  表示两个加数,  $c$  表示来自低位的进位, 布尔函数  $C$  和  $S$  的逻辑值是全加器的输出, 其中  $C$  表示向高位的进位,  $S$  表示加法器求和结果的逻辑值。可以得到  $C$  和  $S$  的逻辑表达式为:

$$S = x \oplus y \oplus c \quad (5)$$

$$C = xy + xc + yc \quad (6)$$

由式(6)可以看出向高位的进位  $C$  正好是一个三个输入变量  $x, y, c$  组成的 MIG 逻辑运算。所以相对于传统的 IMP 逻辑, 用 MIG 逻辑在计算向高位的进位  $C$  时步骤更少。

求和  $S$  需要通过 MIG 逻辑的特有公式进行化简后再进行操作。原  $S$  求和公式的深度如图 1(a) 所示深度为 4, 经过化简只需要深度是 2 就可以完成, 化简后的  $S$  用 MIG 表示如下<sup>[4]</sup>:

$$S = M(M(c, y, x), x, M(\bar{c}, \bar{y}, \bar{x})) \quad (7)$$

在这公式的基础上对基于 MIG 逻辑的一位全加器进行设计。

#### 3.2 基于 MIG 逻辑的一位全加器电路设计

如图 4 所示, 该一位全加器电路采用一维纳米线交叉杆结构 (Crossbar), 将七个 RRAM 存储单元连接在一起。该结构中, 多个 RRAM 通过一条公共线进行连接。因为每个单元都是独立的, 所以如果彼此不相互调用逻辑, 便可以同时进行操作, 具体操作如下所述。

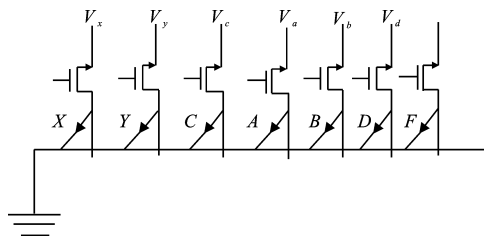


图4 基于 MIG 逻辑的一位全加器电路

在该一位加法器电路中, 输入 RRAM 集合为  $X, Y, C$ , 输出 RRAM 集合为  $C$  和  $A$ , 工作 RRAM 集合为  $A, B, D, F$ , 其中  $A$  不仅是工作 RRAM 单元, 同时还是输出单元存储求和结果  $S$ , 而  $C$  作为输入单元的同时在运算结束后也作为输出单元用来存储向高位的进位。

在运算顺序上, 因为  $C$  单元中存储的来自低位的进位值会被向高位的进位值所取代, 而计算函数  $A$  时不需要用到来自低位的进位值, 所以在计算顺

序上可以先计算 C. 下面是一位全加器的完整步骤:

- 01: $X = x, Y = y, C = c, A = 0, B = 0, D = 0,$   
 $F = 0;$
- 02: $P_A = 1, Q_A = c, R_A = 0 \rightarrow R'_A = \bar{c};$   
 $P_B = 1, Q_B = x, R_B = 0 \rightarrow R'_B = \bar{x};$   
 $P_D = 1, Q_D = y, R_D = 0 \rightarrow R'_D = \bar{y};$   
 $P_F = 1, Q_F = x, R_F = 0 \rightarrow R'_F = \bar{x};$
- 03: $P_B = c, Q_B = \bar{y}, R'_B = \bar{x} \rightarrow R''_B = M(c, y, \bar{x});$   
 $P_A = \bar{y}, Q_A = x, R'_A = \bar{c} \rightarrow R'_A = M(\bar{y}, \bar{x}, \bar{c});$   
 $P_C = x, Q_C = \bar{y}, R_C = c \rightarrow R'_C = M(x, y, c)$   
 $= xy + xc + cy = C_{out};$
- 04: $P_A = M(c, y, \bar{x}), Q_A = \bar{x}, R'_A = M(\bar{y}, \bar{x}, \bar{c})$   
 $\rightarrow M(M(c, y, \bar{x}), x, M(\bar{y}, \bar{x}, \bar{c})) = x \oplus y \oplus c = S$

步骤 1: 在 RRAM 单元中写入数据,  $X$  写入加数,  $Y$  写入被加数,  $C$  写入来自低位的进位值, 其他工作单元  $A、B、D、F$  先置为高阻值状态即逻辑 0;

步骤 2: 这个步骤主要是将输入的状态取反保存在工作 RRAM 单元中, 为后续的操作做准备, 因为要多次使用加数  $x$  的反, 所以存在两个工作单元中, 这里因为取反操作彼此互不干扰所以可以进行并行操作, 对所有的输入同时取反;

步骤 3: 在这一步中计算出  $C_{OUT}$  向高位的进位同时为计算求和  $S$  做准备, 因为这三个工作 RRAM 单元的操作也彼此独立, 所以可以同时进行;

步骤 4: 计算求和  $S$  存入 RRAM 单元  $A$  中.

通过该加法器的过程可知, 基于 MIG 逻辑的加法器设计与传统的基于蕴含逻辑(IMP)的加法器有着很大的不同. 传统的基于蕴含逻辑的加法器, 其本质是通过电阻的分压原理, 这就要求对负载电阻的取值有严格的要求, 而 MIG 逻辑的电路则是通过逻辑值的不同来控制施加在 RRAM 单元上的电压, 避免了对负载电阻的使用, 因为没有负载电阻所以 RRAM 单元的操作彼此独立, 可以在取反等操作中并行执行, 这样极大的缩减了计算的步骤. 同时, 中间进位结果保存在 RRAM 单元  $C$  中, 每次的进位结果运算后都会被存储, 因此  $n$  位全加器整合到了一个器件上, 而不是像 CMOS 加法器电路那样一位加法器与存储设备相连接. 这样的计算电路实现了存储与计算的结合, 为未来新型的计算机的发展提供了一种新的思路.

3.3 RRAM 加法器设计方法比较与评价

在参与运算的 RRAM 数量最小的前提下, 针对一位全加器, 基于 IMP 逻辑<sup>[3]</sup>、“与或”逻辑<sup>[7]</sup>、CRS 结构<sup>[8]</sup>和本文提到的基于 MIG 逻辑的电路设计, 图 5 对操作步骤进行了比较.

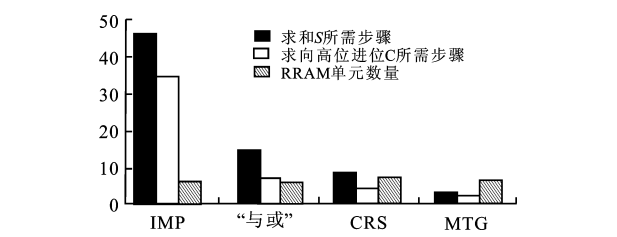


图 5 不同方法基于 RRAM 的一位全加器操作步骤比较

从图 5 中可以看到, 基于 MIG 逻辑设计与传统的 IMP 设计相比, 所需要用到的 RRAM 单元是相同的, 但是 MIG 逻辑电路在求和  $S$  所需的步骤上相比于 IMP 逻辑电路减少了 87.5%, 在求向高位的进位  $C$  所需的步骤上减少了 91.7%, 当加数和被加数为  $n$  位的时候, 求和  $S$  和向高位的进位  $C$  的计算序列长度分别相差  $42n, 34n$ . 与其他几种方法对比 MIG 逻辑的一位全加器也具有明显的步骤优势. 因此, 基本操作的减少, 显著提高了 RRAM 逻辑计算的灵活性. 从另一个方面考虑, 基于 RRAM 的逻辑计算的实现不仅仅局限于蕴含操作(IMP), 也可以根据实际情况来挑选不同的逻辑进行设计.

4 电路仿真结果

4.1 RRAM 单元模型仿真结果

本文采用 SMIC 0.18  $\mu\text{m}$  实现了一种基于 MIG 逻辑的一位全加器. 全加器中的 RRAM 元件电学特性运用 Verilog-A 描述, 且 RRAM 单元的相关参数如表 1 所示. 因为 Verilog-A 语言描述原因  $V_{\text{CNOD}}$  电压分为正负两个电压, 分别对应低阻和高阻的  $V_{\text{CNOD}}$ , RRAM 电学特性仿真结果如图 6 所示.  $V_{\text{SET}}$  电压的作用是将 RRAM 置为低阻值, 即逻辑值表示为“1”;  $V_{\text{CLEAR}}$  电压的作用是将 RRAM 无条件置为高阻态, 即逻辑值表示为“0”.

表 1 全加器中 RRAM 单元相关参数

| 参数          | 符号                 | 数值             |
|-------------|--------------------|----------------|
| SET 电平电压    | $V_{\text{SET}}$   | 1.8 V          |
| CLEAR 电平电压  | $V_{\text{CLEAR}}$ | -2 V           |
| CNOD 电平电压   | $V_{\text{CNOD}}$  | 1/-1 V         |
| RRAM 读电压    | $V_{\text{READ}}$  | 0.2 V          |
| 逻辑为 0 的高阻阻值 | $R_{\text{off}}$   | 87 k $\Omega$  |
| 逻辑为 1 的低阻阻值 | $R_{\text{on}}$    | 1 200 $\Omega$ |

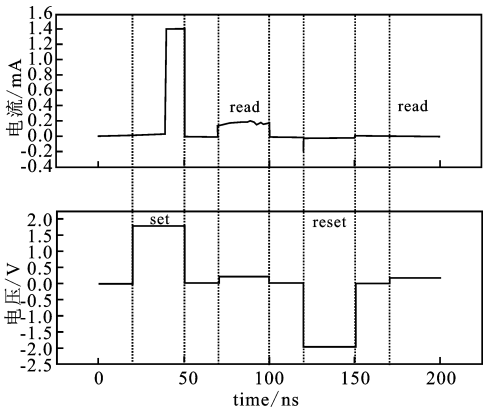


图 6 RRAM 单元电学特性仿真

4.2 基于 MIG 逻辑的一位全加器仿真结果

图 7 所示的是本文所述,基于 MIG 逻辑的一位全加器功能的仿真图. 图中列出了所有的八种一位全加器的可能情况,其中高阻态代表逻辑“0”,低阻态代表逻辑“1”. 从图中可以看出基于 MIG 逻辑的一位全加器可以完成所有的一位全加操作,并且步骤远少于传统的基于“蕴含”逻辑的一位全加器,有效提高了运算速度,减少了因为多次步骤操作所带来的误差. 图中画出了各个 RRAM 单元的阻值变化曲线,及 A、B、C、D、F 五个在一位全加器中有操作的 RRAM 单元上所施加的激励电压曲线,需要注意的是高阻值代表逻辑“0”而低阻值代表逻辑“1”.

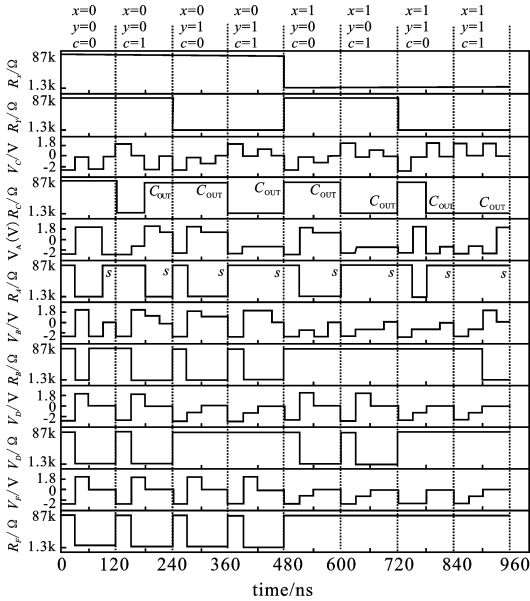


图 7 基于 MIG 逻辑的一位全加器的仿真结果

5 结束语

针对未来存储与计算功能相结合的新型设备,本文提出了基于 MIG 逻辑的 RRAM 一位全加器.

相比传统的蕴含逻辑 RRAM 的加法器,操作中多个 RRAM 单元的并行操作,极大减少了运算步骤. 利用 Verilog-A 和 SMIC0.18  $\mu\text{m}$  工艺,本文最终实现了 RRAM 的一位全加器,经过仿真,验证了 MIG 在 RRAM 上的简易操作方法,为未来的基于 RRAM 的存储与运算相结合的电路设计提供了一种新的方法. 但是,基于 MIG 逻辑的 RRAM 运算电路还存在外围控制电路复杂,读取次数偏多的问题,这还需要工程师进一步的研究和改进.

参考文献:

[1] Chua L O, Kang S M. Memristive devices and systems [J]. the IEEE, 1976, 64(2):209-223.  
[2] Esch J. Prolog to circuit elements with memory: memristors, memcapacitors, and meminductors [J]. the IEEE, 2009, 97(10):1715-1716.  
[3] Borghetti J, Snider G S, Kuekes P J, et al. 'Memristive' switches enable 'stateful' logic operations via material implication. [J]. Nature, 2010, 464(7290):873.  
[4] Amaru L, Gaillardon P E, De Micheli G. Majority-inverter graph: a novel data-structure and algorithms for efficient logic optimization[C]// Design Automation Conference. San Francisco, ACM, 2014:1-6.  
[5] Shirinzadeh S, Soeken M, Gaillardon P E, et al. Fast logic synthesis for RRAM-based in-memory computing using Majority-Inverter Graphs[C]// Design, Automation & Test in Europe Conference & Exhibition. Nice, France, IEEE, 2016:948-953.  
[6] 宋焕章,王保恒,张春元. 计算机原理与设计(下册)—存储与外设[M]. 长沙:国防科技大学出版社,1999.  
[7] 张娜. 基于忆阻器的加法器和乘法器高效设计与模拟[D]. 长沙:国防科学技术大学,2011.  
[8] Yang Y, Mathew J, Pontarelli S, et al. Complementary resistive switch-based arithmetic logic implementations using material implication[J]. IEEE Transactions on Nanotechnology, 2016, 15(1):94-108.

作者简介:

刘文楷 男,(1968-),副教授. 研究方向为光子器件及集成技术.  
范冬宇(通讯作者) 男,(1991-),硕士研究生. 研究方向为模拟集成电路设计. E-mail: siriusfan@icloud.com.  
戴 澜 男,(1975-),副教授. 研究方向为大规模集成电路设计.  
张 锋 男,(1977-),副研究员. 研究方向为高速低功耗存储器件设计.