2016 IEEE 46th International Symposium on Multiple-Valued Logic

Notes on Majority Boolean Algebra

Anupam Chattopadhyay*, Luca Amarú†, Mathias Soeken‡, Pierre-Emmanuel Gaillardon∓, and Giovanni De Micheli‡

*School of Computer Engineering, Nanyang Technological University, Singapore
†Synopsys Inc., USA
∓Electrical and Computer Engineering (ECE), University of Utah, USA
‡Integrated Systems Laboratory, EPFL, Switzerland
*anupam@ntu.edu.sg

***Abstract*—A Majority-Inverter Graph (MIG) is a homogeneous logic network, where each node represents the majority function. Recently, a logic optimization package based on the MIG data-structure, with 3-input majority node (M_3) has been proposed [2], [30]. It is demonstrated to have efficient area-delay-power results compared to state-of-the-art logic optimization packages. In this paper, the Boolean algebraic transformations based on majority logic, i.e., majority Boolean algebra is studied. In the first part of this paper, we summarize a range of identities for majority Boolean algebra with their corresponding proofs. In the second part, we venture towards heterogeneous logic network and provide reversible logic mapping of majority nodes.**

I. Introduction

In Boolean logic, the majority function is defined to be true if at least half of the inputs are true. Formally speaking, for n odd, $M_n(x_1, x_2, \ldots, x_n) = 1$, if and only if $[x_1 + x_2 + \cdots + x_n] \geq \lceil \frac{n}{2} \rceil$. A Boolean algebra is defined over a set of binary values $\mathbb{B} = \{0, 1\}$ and the basic operations AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). Several Boolean algebra operations can be taken together to form a *complete* set of laws, from which other laws can logically derived. This allows *axiomatization* of Boolean algebra. In [2], it is proved that Boolean algebra, when defined over the set of $\{\mathbb{B}, M, \neg, 0, 1\}$[1] is complete and sound under the laws (Ω) defined as following.

Associativity $\Omega.A$
$M(x, u, M(y, u, z))$
$= M(z, u, M(y, u, x))$
$= M(y, u, M(z, u, x))$
Commutativity $\Omega.C$
$M(x, y, z) = M(y, x, z) = M(z, y, x)$
Distributivity $\Omega.D$
$M(x, y, M(u, v, z)) = M(M(x, y, u), M(x, y, v), z)$
Majority $\Omega.M$
$M(x, y, y) = y;\ M(x, \bar{x}, z) = z$
Inverter Propagation $\Omega.I$
$\overline{M}(x, y, z) = M(\bar{x}, \bar{y}, \bar{z})$

Henceforth, we refer to the aforementioned axiomatized Boolean algebra as *Majority Boolean algebra*.

[1] M_3 is referred as M, unless mentioned otherwise

A. Applications

Novel post-CMOS devices and systems, such as the Quantum-dot Cellular Automata (QCA) system, have reported realization of a majority Boolean logic gate [12]. Majority logic circuits with Quantum Flux Parametron (QFP) technology, which can achieve significantly higher clock speed compared to CMOS technologies, have been experimentally demonstrated at [14]. Li *et al.* have demonstrated 3-input majority logic gate using DNA strand displacement [16]. Interestingly, the study with devices capable of majority logic manipulation is not recent. As early as in 1960, Goto *et al.* demonstrated high-speed logical circuits with Esaki diodes (a.k.a. Tunnel Diodes). There, the majority logic circuit is realized first, and Boolean logic operations ($\wedge, \vee$) are implemented as special cases of Majority logic.

Independent of the majority-demonstrating non-CMOS devices, it was shown in [2] that, majority Boolean logic transformations can be helpful in optimization of CMOS-based logic circuits, with better or comparable results against state-of-the-art logic optimization packages, e.g., AND-Inverter-Graphs (AIGs, [7]) and Binary Decision Diagrams (BDDs, [24]).

For the circuit complexity theorists, majority functions provide an interesting problem in the efficient circuit construction. Valiant showed the existence of polynomial-size, logarithmic-depth monotone formula for Majority function [26]. A variation of this construction also achieving logarithmic depth, using $T(2, 3)$ gates, is proposed at [10]. There $T(k, m)$ is threshold Boolean function, that evaluates true iff at least k of its m inputs are true. The proposed construction is utilized for linear and scalable secret sharing schemes.

Majority functions are also studied in relation to social choices and voting system. Corresponding theoretical study was done in [15], [18], where it was conjectured [15] and then proved [18] that for independent binary variables, where each input has low influence, majority function is most stable to noise, e.g., input bit-flips.

B. Previous Studies

Despite the wide range of applications for majority function, due to lack of interest in logic circuit community, majority Boolean algebra did not receive intensive study. In past, this was studied in the context of devices demonstrating majority

0195-623X/16 $31.00 © 2016 IEEE
DOI 10.1109/ISMVL.2016.21

operations, as it is being studied right now. To the best of our knowledge, majority decision logic was introduced in [17] and an axiom set is proposed in [8]. In these works, the notation of $\{x\#y\#z\}$ is used to indicate $M(x, y, z)$. Akers proposed a different notation style $(x(\mathrm{Z})y)$ and introduced a range of Majority identities matching with Boolean identities in [3]. For efficient manipulation of majority logic networks, transformation rules have been proposed in [17], [2]. These works restricted the study in homogeneous M_3 logic networks. Efficacy of M_5 has been studied in [19], [1] in the context of QCA. Expression of complex Boolean gates via M_5 and M_7 is explored in [25].

In this paper, we adopt the axiomatic system proposed at [2]. We revisit the identities of Boolean algebra from the previous works and provide proofs of identities, for the sake of completeness. We also study the properties of majority Boolean algebra for heterogeneous logic networks. Finally, the reversible logic synthesis from majority networks is briefly reviewed.

II. Homogeneous Majority Logic Network

In this section first, corresponding to the standard Boolean algebra, we define a set of monotone and non-monotone laws for Majority Boolean algebra $\{\mathbb{B}, M, \neg, 0, 1\}$. We provide a list of identities to operate on majority Boolean logic networks. For trivial results and for proofs reported earlier [2], proofs are skipped. Finally, we establish correspondence between classical Boolean logic operators and Majority Boolean operations in the last two subsections.

A. Monotone and Non-monotone Laws

Among the monotone laws, *associativity*, *commutativity*, *distributivity*, and *majority* are already stated in the axiom Ω. From $\Omega.M$, the following laws follow immediately:

Identity Pair
$M(x, 0, 1) = x$
Annihilator Pair
$M(x, 0, 0) = 0;\ M(x, 1, 1) = 1$
Idempotence
$M(x, x, x) = x$
Absorption
$M(x, x, y) = x$

Among the non-monotone laws, *inverter propagation* is already stated in the axiom Ω. We further define,

Involution
$\neg\neg M(x, y, z) = M(x, y, z)$

B. Majority Boolean Algebra: Identities

In this section, we list a comprehensive set of identities reported in the literature as well as present several new ones. For variable substitution (x replaced by y) within an expression z, the notation $z_{x/y}$ is used [2]. According to the effect achieved by the identity, those are grouped in *expansion*, *contraction*, or *reshaping* types. Note that, for easy reference, we listed also several axiomatic identities here. Several identities from the expansion can be applied in reverse direction to achieve expansion effect, e.g., M2 and M10 do the same transformation in opposite direction.

Expansion

M1: $x = M(v, \bar{v}, x)$
M2: $M(x, y, M(u, v, z)) = M(M(x, y, u), M(x, y, v), z)$
M3: $M(x, y, z) = M(x, y, M(x, y, z))$

Contraction

M4: $M(x, x, y) = x$
M5: $M(x, \bar{x}, y) = y$
M6: $M(x, 1, M(x, y, 0)) = x$
M7: $M(x, 1, M(x, y, 1)) = M(x, y, 1)$
M8: $M(x, y, \overline{M}(x, y, z)) = M(x, y, \bar{z})$
M9: $M(x, y, M(\bar{x}, \bar{y}, z)) = M(x, y, z)$
M10: $M(M(x, y, u), M(x, y, v), z) = M(x, y, M(u, v, z))$
M11: $M(w, M(w, x, y), M(\bar{w}, x, z))) = M(w, x, y)$
M12: $M(x, M(u, v, z), M(\bar{u}, \bar{v}, \bar{z})) = x$

Reshaping

M13: $M(x, y, z) = M(x, y, z_{x/\bar{y}}) = M(x, y, z_{y/\bar{x}})$
M14: $M(x, u, M(y, \bar{u}, z)) = M(x, u, M(x, y, z))$
M15: $M(M(\bar{w}, \bar{x}, z), y, M(w, x, z))$
$= M(M(w, \bar{x}, y), z, M(\bar{w}, x, y))$
M16: $M(x, M(\bar{x}, z, u), M(\bar{x}, \bar{z}, u))$
$= M(M(\bar{x}, z, u), M(x, u, \bar{z}), u)$

In [8], a case-by-case analysis of the equality of the variables is done to provide several proofs. We note that, such an analysis is difficult for large expressions. Instead, the variable substitution technique proposed in [2] (here M13) suffices for deriving the proof from the axioms. Few exemplary proofs are demonstrated. Identities M1 and M3, used for the proofs can be easily derived from Ω.

Lemma 1. *Identity M3 can be derived from* Ω.

Proof: $M(x, y, z)$
$= M(x, y, M(x, \bar{x}, z))$ (M1)
$= M(x, y, M(x, y, z))$ (M13)

■

Lemma 2. *Identity M15 can be derived from* Ω.

Proof: $M(M(\bar{w}, \bar{x}, z), y, M(w, x, z))$
$= M(M(w, y, M(\bar{w}, \bar{x}, z)), M(x, y, M(\bar{w}, \bar{x}, z)), z)$ ($\Omega.D$)
$= M(M(w, y, M(\bar{x}, y, z)), M(x, y, M(\bar{w}, y, z)), z)$ (M13)
$= M(M(y, z, M(w, \bar{x}, y)), M(y, z, M(\bar{w}, x, y)), z)$ ($\Omega.A$)
$= M(y, z, M(z, M(w, \bar{x}, y), M(\bar{w}, x, y)))$ ($\Omega.D$)
$= M(M(w, \bar{x}, y), z, M(y, z, M(\bar{w}, x, y)))$ ($\Omega.A$)
$= M(M(w, \bar{x}, y), z, M(y, \neg M(w, \bar{x}, y), M(\bar{w}, x, y)))$ (M13)
$= M(M(w, \bar{x}, y), z, M(y, M(\bar{w}, x, \bar{y}), M(\bar{w}, x, y)))$ ($\Omega.I$)
$= M(M(w, \bar{x}, y), z, M(\bar{w}, x, M(y, y, \bar{y})))$ ($\Omega.D$)
$= M(M(w, \bar{x}, y), z, M(\bar{w}, x, y))$ ($\Omega.M$)

■

C. Correspondence with Classical Boolean Algebra

Besides the ability to manipulate in majority Boolean algebra, it is also important to establish transformations between classical Boolean algebra $\{\mathbb{B}, \vee, \wedge, \neg, 0, 1\}$. The following equations allow such bi-directional transformations.

$$x \vee y = M(x, y, 1) \tag{1}$$

$$x \wedge y = M(x, y, 0) \tag{2}$$

$$M(x, y, z) = xy \oplus yz \oplus zx = xy + yz + zx \tag{3}$$

Though, the reduction of a Disjunctive/Conjunctive Normal Form (CNF/DNF) specification or an AND-Inverter Graph (AIG) to majority logic network can be accomplished with the aforementioned equations, it introduces redundancy in the circuit, which needs to be further optimized. This can be simply demonstrated with the following scenario.

$M(x, y, z) = xy + yz + zx$
$= M(x, y, 0) + M(y, z, 0) + M(z, x, 0)$
$= M(x, y, 0) + M(M(y, z, 0), M(z, x, 0), 1)$
$= M(M(x, y, 0), M(M(y, z, 0), M(z, x, 0), 1), 1)$

Therefore, it is necessary to develop powerful theorems connecting classical and majority Boolean algebra. One such theorem is proposed in [17]. The theorem is presented in Equations 4 and 5.

$$\begin{aligned} &x \cdot \sum_{i=1}^{n-1} f_i + \bar{x} \cdot \prod_{i=n}^{2n-2} f_i \\ &= m_n[x, (xf_1 + \bar{x}f_n), \cdots, (xf_{n-1} + \bar{x}f_{2n-2})], \end{aligned} \tag{4}$$

where f_i is a Boolean function without x as one of the literals. m_n can be expressed as following.

$$m_n(v_1, v_2, \cdots, v_n) = M(v_1, m_{n-1}(v_1, \cdots, v_{n-1}), v_n) \tag{5}$$

and m_3 is M. In [17], a proof of above theorem based on induction is provided. The utility of this theorem can be shown by fixing n to a certain value. For $n = 3$, equation 4 takes the following form.

$$x(f_1 + f_2) + \bar{x}(f_3 \cdot f_4) = M(x, (xf_1 + \bar{x}f_3), (xf_2 + \bar{x}f_4)) \tag{6}$$

Here, the internal Boolean expressions, such as $(xf_1 + \bar{x}f_3)$, can *not* be further decomposed with the same technique.

$(xf_1 + \bar{x}f_3)$
$= (x(f_1 + 0) + \bar{x}(f_3 \cdot 1))$
$= M(x, (xf_1 + \bar{x}f_3), \bar{x})$
$= (xf_1 + \bar{x}f_3)$

Rather, one needs to use the basic equations (equations 1, 2) to derive majority Boolean logic formulation. Nevertheless, the proposed theorem is useful if the original Boolean logic expression can be cleverly arranged. This is shown in [17] by deriving a majority logic expression for a full adder circuit. [2] Another decomposition, which does not require such an arrangement, is provided in [3] without proof. This is studied in the following subsection.

D. Majority-based Decomposition

The majority-based decomposition rule is presented in the following theorem.

Theorem 3. *The decomposition of* $F(x, y, \ldots, u)$ *to* $M(x, M(\bar{x}, y, F(y, y, \ldots, u)), M(\bar{x}, \bar{y}, F(\bar{y}, y, \ldots, u)))$ *follows classical and Majority Boolean algebra axioms.*

Proof:
$M(x, M(\bar{x}, y, F(y, y, \ldots, u)), M(\bar{x}, \bar{y}, F(\bar{y}, y, \ldots, u)))$
$= M(x, M(\bar{x}, y, F(x, y, \ldots, u)), M(\bar{x}, \bar{y}, F(x, y, \ldots, u)))$
$= M(x, M(\bar{x}, y, F), M(\bar{x}, \bar{y}, F))$ (for clarity)
$= xM(\bar{x}, y, F) + xM(\bar{x}, \bar{y}, F) + M(\bar{x}, y, F)M(\bar{x}, \bar{y}, F)$
$= x(\bar{x}y + yF + \bar{x}F) + x(\bar{x} \cdot \bar{y} + \bar{y}F + \bar{x}F)$
$\quad +(\bar{x}y + yF + \bar{x}F)(\bar{x} \cdot \bar{y} + \bar{y}F + \bar{x}F)$
$= xyF + x\bar{y}F + \bar{x}yF + \bar{x} \cdot \bar{y}F + \bar{x}F$
$= xyF + x\bar{y}F + \bar{x}F$
$= xF + \bar{x}F$
$= F = F(x, y, \ldots, u)$

■

In [2], *Majority Inverter Graph (MIG)* is defined as a homogeneous logic network, where the nodes represent a 3-input majority function (M_3) and the edges can be complemented. The majority-based decomposition procedure can be repeatedly applied to derive MIG for the given Boolean function, as shown in the following figure 1.

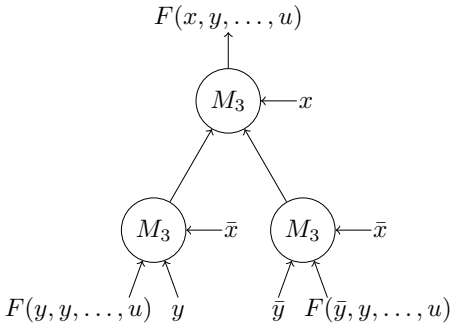

Fig. 1. Majority Inverter Graph

We illustrate the efficacy of the majority logic decomposition by developing the majority expression for the full adder sum, $S = a \oplus b \oplus c_i$. Naïve application of equations 1 and 2 would result in a majority logic network of depth 4, where the primary inputs are considered to be at depth 0. By Theorem 3,
$S = M(a, M(\bar{a}, b, (b \oplus b \oplus c_i)), M(\bar{a}, \bar{b}, (\bar{b} \oplus b \oplus c_i)))$
$= M(a, M(\bar{a}, b, c_i), M(\bar{a}, \bar{b}, \bar{c}_i))$
$= M(a, M(\bar{a}, b, c_i), \neg M(a, b, c_i))$

Note that this is a majority network with logical depth of 2. In this case, $M(a, b, c_i)$ is shared with the circuit for

[2]There is a typo in equation 15, last expression of [17]. The corresponding Fig. 3 is correctly drawn.

generating carry bit. For a ripple-carry adder formation, the following result can be obtained easily.

Lemma 4. *An n-bit adder can be represented with $3n$ nodes in MIG.*

Further, from the decomposition procedure, the following lemma can be stated.

Lemma 5. *The MIG representation of an n-variable Boolean function can have at most $3(2^{n-1}-1)$ M_3 nodes.*

Proof: From the decomposition procedure outlined in Theorem 3, it requires 3 nodes per variable decomposition. From these 3 nodes, 2 nodes are further decomposed. For these nodes, 2 edges lead to primary inputs and one edge leads to a reduced function. The decomposition continues until a majority function with only one variable is obtained, for which a primary input can be deduced. This gives rise to the following series for n-variable Boolean function.
$1 + 2(1 + 1 + 2(1 + 1 + 2(1 + \cdots + 2(1))))$
$= 1 + 2^2 + 2^3 + \cdots + 2^{n-1} + 2^{n-1}$
$= 3(2^{n-1} - 1)$ ■

It can be immediately observed that this bound is tight for 2-variable Boolean functions. For a MIG representation, $(a \oplus b)$ requires the most count of nodes, which is $3(2^{1-1}-1) = 3$. For large arithmetic circuits, it has been shown that $M_n,\ n > 3$ can be beneficial for compactness of the representation as well as the final implementation if a rich cell library is available [19], [1]. An improved bound for the MIG node count is presented recently in [23], where minimum MIG representations are pre-computed for functions up to 4 variables. In the following section, heterogeneity in majority Boolean algebra is explored.

III. Heterogeneous Majority Logic Network

In this section, we first look into the axioms from Ω and report similar identities for the 5-input majority gate, M_5. For a general case of M_n, where $n > 3$, a full axiomatic system is recently developed and proposed in [30]. Our goal is slightly different in the sense that we want to explore a mix of diverse majority nodes. Therefore, identities combining M_5 and M_3 are expressed. Using those identities, results presented in [19] are derived. Finally, the rationale for heterogeneous majority Boolean network is investigated from the perspective of logical depth. In that context, capabilities of a configurable majority gate are studied.

A. Basic Identities of M_5

Similar to the axioms defined in Ω (section I), we define a set of identities for M_5 as following. We refer to those as Φ.

Associativity $\Phi.A$
$M_5(x, y, z, u, M_5(a, b, y, z, u))$
$= M_5(a, y, z, u, M_5(x, b, y, z, u))$
$= M_5(b, y, z, u, M_5(a, x, y, z, u))$

Commutativity $\Phi.C$
$M_5(u, v, x, y, z) = M_5(x, y, z, u, v) = M_5(u, x, v, y, z)$

Distributivity $\Phi.D$
$M_5(v, x, y, z, M_5(a, b, c, d, e)) = M_5(M_5(v, x, y, z, a),$
$M_5(v, x, y, z, b), M_5(v, x, y, z, c), d, e)$

Majority $\Phi.M$
$M_5(x, x, x, u, v) = x;\ M_5(x, \bar{x}, y, u, v) = M_3(y, u, v)$

Inverter Propagation $\Phi.I$
$\neg M_5(x, y, z, u, v) = M_5(\bar{x}, \bar{y}, \bar{z}, \bar{u}, \bar{v})$

We omit the proofs since the proofs for the general case of $n > 3$ are already presented in [30].

B. Identities on M_5 and M_3

Any majority function can be decomposed into an expression containing smaller majority functions. This is useful for deriving the identities as well as for computing the complexity of majority-based representation for a given Boolean function. For M_5, the expression is as following.

$$M_5(x, y, z, u, v) = (x \oplus y)M_3(z, u, v) + xy(z + u + v) + \bar{x} \cdot \bar{y}(zuv) \quad (7)$$

By studying equation 7, following identities, apart from the basic identities mentioned in the previous subsection, can be derived. Some of these are listed in [1] (Table 1).

M17: $M_5(1, 1, z, u, v) = M_3(z, 1, M_3(v, u, 1))$
M18: $M_5(0, 0, z, u, v) = M_3(z, 0, M_3(v, u, 0))$
M19: $M_3(z, u, M_3(z, v, x)) = M_5(z, z, u, v, x)$
M20: $M_5(x, y, x, y, z) = M_3(x, y, z)$

Lemma 6. *Identity M19 follows Φ and classical Boolean algebra axioms.*

Proof: $M_3(z, u, M_3(z, v, x))$
$= zu + (z + u)(zv + vx + zx)$
$= zu + zv + zvx + zx + uzv + uvx + uzx$
$= z(u + v + x) + uvx$
$= (z \oplus z)M_3(u, v, x) + zz(u + v + x) + \bar{z}\bar{z}(uvx)$
$= M_5(z, z, u, v, x)$ (following equation 7) ■

By identity M19, it is possible to simplify the sum S of a full-adder. To start with, we have the M_3 expression as following.
$S = M_3(a, M_3(\bar{a}, b, c_i), \neg M_3(a, b, c_i))$
$= M_3(a, M_3(\bar{a}, b, c_i), \bar{c}_{i+1})$
$= M_3(a, M_3(\bar{c}_{i+1}, b, c_i), \bar{c}_{i+1})$
$= M_5(\bar{c}_{i+1}, \bar{c}_{i+1}, a, b, c_i)$ (M19)

This simplification is used in [19], [1] to derive a 2-element heterogeneous majority logic network for a full adder. Thus, it is possible to have an n-bit adder represented with $2n$ nodes in MIG, which is 50% smaller compared to the M_3-only representation (Lemma 4). An interesting point to note is that for a majority-based full-adder structure, the sum logic is deeper (depth 2) compared to the carry logic (depth 1).

In contrast to equation 7, the following equation 8 provides an elaboration of M_5 only in terms of M_3.

$$\begin{aligned} M_5(x_1, x_2, x_3, x_4, x_5) = \\ M_3(M_3(x_1, x_2, x_3), M_3(x_1, x_4, x_5), \\ M_3(x_1, M_3(x_2, x_4, x_5), M_3(x_2, x_3, x_4))) \end{aligned} \quad (8)$$

Lemma 7. *Equation 8 follows* Φ.

Proof: If both the first two constituents of the top-level M_3 operator in the RHS expression is true then, at least 3 members of the set $\{x_1, x_2, x_3, x_4, x_5\}$ are true. This satisfies the equivalence. If only one of these first two constituents are true, we can have two cases. Case I: x_1 is true. In this case, the LHS expression is true if both members of either of the sets $\{x_2, x_3\}$ and $\{x_4, x_5\}$ are true. That is evaluated in the final constituent expression of the top-level M_3 operator. Case II: x_1 is false. In this case, the LHS expression is true if exactly 3 members from the set $\{x_2, x_3, x_4, x_5\}$ are true. If 3 elements from that 4-member set is chosen in any combination, it will evaluate to be true. Thus, the final constituent expression of the top-level M_3 operator holds true and validates the equivalence. ■

$$M_5(x_1,x_2, x_3, x_4, x_5) = M_3(M_3(x_1, x_2, x_3), M_3(x_1, x_4, x_5), M_3(x_1, M_3(x_3, x_4, x_5), M_3(x_2, x_3, x_5))) \quad (9)$$

From the proof, it naturally follows that the equation 8 can be also expressed with a different combination of elements, such as the one given in the equation 9. It can also be noted that the identities M17, M18, M19 and M20 can be derived from the equation 8. A general approach for such decompositions is recently developed in [11].

C. Boolean function Complexity Analysis for Majority Logic Network

Several complex Boolean gates have been derived from M_5 and M_7 in [1] and [25] respectively, albeit in the context of QCA. In [25], it is proposed to use several inputs of a majority gate as configuration inputs and to set those as 0/1 to derive different complex multi-input logic gates out of the rest inputs. This can be considered as a generic field-programmable implementation with diverse majority gates and then, configuring some input bits to determine the functionality. From the perspective of circuit complexity theory [21], [27], this technique raises several interesting questions as following.

- For a given n-input Boolean function, what are the *lower* and *upper bounds* of the *size* of a homogeneous/heterogeneous MIG?
- For a given n-input Boolean function, what are the *lower* and *upper bounds of the depth* of a homogeneous/heterogeneous MIG?

We use the standard definitions of depth and size, i.e., the depth of a node is the length of the longest path from any primary input to the node. The depth of an MIG is the largest depth of a node. The size of an MIG is its number of nodes [2]. While a complete study of these questions is beyond the scope of the current paper, we briefly look into these.

It can be observed that with a depth-1 M_3 network, it is not possible to express all 2-variable Boolean functions. Among the 3-variable Boolean functions only $ab \oplus bc \oplus ca$ can be expressed. However, by setting one variable to 0 and 1, depth-1 M_3 network can express 2-input AND and 2-input OR gates respectively. Formally, we define a notation with $M_n(d, s)$, where d and s indicates depth and size respectively. Then, we have the following lemma, where B_2 represent the set of all 2-input Boolean functions. The most complex B_2 function for M_3 is $a \oplus b$.

Lemma 8. $M_3(2, 3) \equiv B_2$.

Lemma 8 can be extended to show that $M_3(4, 9) \equiv B_3$, where the 2-variable constituent functions are created by $M_3(2, 3)$. The 2-variable functions are generated from the 3-variable function using Shannon expansion. The size of a majority network can be reduced if M_5 is used. Assuming unrestricted fanout, one may implement the most complex B_2 circuit for majority gates, using $M_5(\neg M_3(a, b, 0), \neg M_3(a, b, 0), a, b, 0)$. Hence, the following lemma can be easily established. Note that, for simpler functions one may use the M_3, which is a constituent of M_5, i.e., $M_5 \supseteq M_3$.

Lemma 9. $M_5(2, 2) \equiv B_2$.

Corollary 10. $M_5(4, 7) \equiv B_3$

Proof: For the Boolean function $a \oplus b$, $M_5(2, 2)$ suffices. However, with Shannon expansion, B_3 requires the implementation of $\bar{a}f_1 + af_2$, for which no smaller M_5 expansion than utilizing three M_3 gates are known. ■

IV. Reversible Logic Mapping of Majority Nodes

Due to the efficiency of the MIG structure and potential realization of majority nodes in several emerging technologies, reversible logic mapping of majority nodes is interesting to study. One such mapping for M_3 is reported in [29].

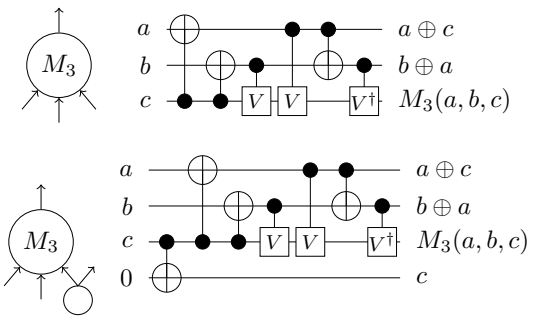

Fig. 2. Mapping of Majority Nodes to Reversible Circuits (I)

In [29], a compact implementation of majority logic is presented using only 6 two-qubit gates. This is shown in the top of Fig. 2. A naïve extension of that mapping to account for sharing of the inputs requires 3 ancilla lines and 9 gates. The case for one shared input is shown in the bottom of Fig. 2. However these implementations are not optimal. By applying an exact synthesis method, we could prove that a realization with 3 Toffoli gates is the smallest in terms of gate count. This is shown in the Fig. 3. Also shown is the realization which preserves all the inputs, requiring 9 gates and 1 ancilla line.

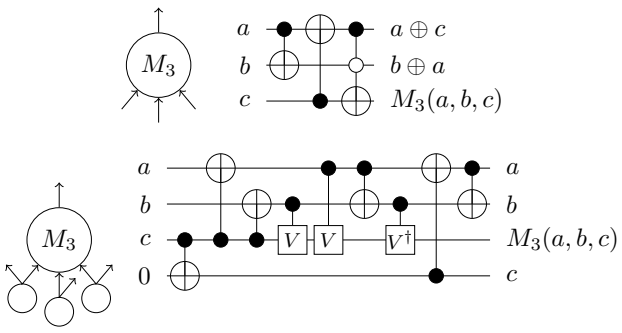

Fig. 3. Mapping of Majority Nodes to Reversible Circuits (II)

Remark: It can be noted that the reversible circuit for an M_3 node is comparable in terms of gate count, QC and ancilla of an equivalent reversible circuit for a BDD node.

V. Summary and Outlook

Due to the emergence of new computing devices as well as the increasing complexity of logic circuits, majority-based circuit design is receiving renewed research attention. In this perspective, this paper made several contributions. Firstly, by summarizing classic and contemporary works, a comprehensive set of techniques for majority Boolean algebra operations is presented. The inter-relation between majority Boolean algebra and classical (AND-OR) Boolean algebra is investigated. Furthermore, heterogeneous majority Boolean networks are discussed along with their efficient mapping for reversible circuits.

Multiple, independent lines of research can be pursued from here, such as, extending current MIG synthesis flows to utilize the identities presented here; study of canonicity for Majority Inverter Graph, if a strict variable order and fixed decomposition procedure is adopted; and studying the effects of heterogeneous majority logic, particularly for arithmetic circuits.

References

[1] R. Akeela and M. Wagh, "A five input majority gate in quantum-dot cellular automata," in *NSTI Nanotech*, vol. 2, pp. 978–981, 2011.

[2] L. Amarú, P.-E. Gaillardon and G. De Micheli, "Majority-Inverter Graph: A Novel Data-Structure and Algorithms for Efficient Logic Optimization," in *Proceedings of the 51st Annual Design Automation Conference (DAC '14)*.

[3] S. B. Akers, Jr., "On the Algebraic Manipulation of Majority Logic," in *IRE Transactions on Electronic Computers*, vol. EC-10, no. 4, pp. 779, 1961, doi=10.1109/TEC.1961.5219289.

[4] J. Baugh et al., "Quantum information processing using nuclear and electron magnetic resonance: review and prospects," in *ArXiV, available online at http://arxiv.org/abs/0710.1447*, October, 2007

[5] A. Barenco et al., "Elementary gates for Quantum Computation," in *Physical Review A*, vol. 52, no. 5, pp. 3457–3467, 1995, doi:10.1103/PhysRevA.52.3457.

[6] C. H. Bennett, "Logical reversibility of computation," in *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525–532, 1973.

[7] Berkeley Logic Synthesis and Verification Group, "ABC: A System for Sequential Synthesis and Verification," available online at http://www.eecs.berkeley.edu/∼alanmi/abc/.

[8] M. Cohn and R. Lindaman, "Axiomatic Majority-Decision Logic," in *IRE Transactions on Electronic Computers*, vol. EC-10, no. 1, pp. 17–21, March 1961, doi: 10.1109/TEC.1961.5219147.

[9] R. Cuykendall and D. R. Andersen, "Reversible optical computing circuits," in *Optics Letters*, vol. 12, no. 7, pp. 542–544, 1987.

[10] I. Damgård, J. Kölker, P. Bro Miltersen, "Secret Sharing and Secure Computing from Monotone Formulae," in *IACR eprint Archive*, Avaiable online at http://eprint.iacr.org/2012/536.pdf.

[11] R. Devadoss, K. Paul and M. Balakrishnan, "MajSynth : An n-input Majority Algebra based Logic Synthesis Tool for Quantum-dot Cellular Automata," in *24th International Workshop on Logic Synthesis*, 2015.

[12] A. Imre, G. Csaba, L. Ji, A. Orlov, G. H. Bernstein, W. Porod, "Majority Logic Gate for Magnetic Quantum-Dot Cellular Automata," in *Science*, vol. 311, no. 5758, pp. 205–208, 2006, DOI: 10.1126/science.1120506.

[13] E. Goto, K. Murata, K. Nakazawa, K. Nakagawa, T. Moto-Oka, Y. Matsuoka, Y. Ishibashi, H. Ishida, T. Soma and E. Wada, "Esaki Diode High-Speed Logical Circuits," in *IRE Transactions on Electronic Computers*, vol. EC-9, no. 1, pp. 25–29, March 1960, doi: 10.1109/TEC.1960.5221600.

[14] W. Hioe, M. Hosoya, S. Kominami, H. Yamada, R. Mita and K. Takagi, "Design and operation of a Quantum Flux Parametron bit-slice ALU," in *IEEE Transactions on Applied Superconductivity*, vol. 5, no. 2, pp. 2992–2995, June 1995, doi: 10.1109/77.403221.

[15] S. Khot, G. Kindler, E. Mossel and R. O'Donnell, "Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?," in *Proceedings of 45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 146–154, 17-19 Oct. 2004, doi: 10.1109/FOCS.2004.49.

[16] W. Li, Y. Yang, H. Yan and Y. Liu, "Three-Input Majority Logic Gate and Multiple Input Logic Circuit Based on DNA Strand Displacement," in *Nano Letters*, vol. 13, no. 6, pp. 2980–2988, May 2013, doi: 10.1021/nl4016107.

[17] R. Lindaman, "A Theorem for Deriving Majority-Logic Networks Within an Augmented Boolean Algebra," in *IRE Transactions on Electronic Computers*, vol. EC-9, no. 3, pp. 338–342, Sept. 1960, doi: 10.1109/TEC.1960.5219856.

[18] E. Mossel, R. O'Donnell and K. Oleszkiewicz, "Noise stability of functions with low influences invariance and optimality," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS '05)*, pp. 21–30, DOI=10.1109/SFCS.2005.53.

[19] K. Navi, S. Sayedsalehi, R. Farazkish and M. R. Azghadi, "Five-Input Majority Gate, a New Device for Quantum-Dot Cellular Automata," in *Journal of Computational and Theoretical Nanoscience*, vol. 7,no. 8,pp. 1–8,2010, DOI=10.1166/jctn.2010.1517.

[20] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press, 2000.

[21] C. E. Shannon, "The synthesis of two-terminal switching circuits," in *Bell System Technical Journal*, vol. 28,no.1, January, 1949, pp. 59–98.

[22] M. Saeedi and I. L. Markov,"Synthesis and Optimization of Reversible Circuits - A Survey," in *CoRR abs/1110.2574*, http://arxiv.org/abs/1110.2574, 2011.

[23] M. Soeken, L. Amarú, P.-E. Gaillardon and G. De Micheli,"Optimizing Majority-Inverter Graphs With Functional Hashing," in Proceedings of IEEE/ACM DATE, 2016.

[24] F. Somenzi, "CUDD: CU Decision Diagram Package," available online at http://vlsi.colorado.edu/∼fabio/CUDD/.

[25] W. J. Townsend and J. A. Abraham, "Complex Gate Implementations for Quantum Dot Cellular Automata," in *Proceedings of 4th IEEE Conference on Nanotechnology*, pp. 625–627, 16-19 Aug. 2004, doi: 10.1109/NANO.2004.1392440.

[26] L. G. Valiant, "Short monotone formulae for the majority function," in *Journal of Algorithms*, vol. 5, no. 3, pp. 363–366, September 1984, doi: 10.1016/0196-6774(84)90016-6.

[27] I. Wegener, "The Complexity of Boolean functions," in *Wiley-Teubner Series in Computer Science*, ISBN: 3-519-02107-2, 1987.

[28] R. Wille and R. Drechsler, "BDD-based Synthesis of Reversible Logic for Large Functions," in *Proceedings of DAC*, pp. 270–275, 2009.

[29] G. Yang, W. N. N. Hung, X. Song and M. Perkowski, "Majority-based reversible logic gates," in *Elsevier Theoretical Computer Science*, vol. 334, no. 1–3, pp. 259–274, April 2005, doi:10.1016/j.tcs.2004.12.026.

[30] L. Amarù, P.-E. Gaillardon, A. Chattopadhyay and G. De Micheli, "A Sound and Complete Axiomatization of Majority-n Logic," in *IEEE Transactions on Computers*, 2016, doi: 10.1109/TC.2015.2506566.