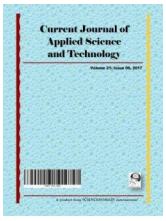


Current Journal of Applied Science and Technology

27(3): 1-16, 2018; Article no.CJAST.41481
ISSN: 2457-1024
(Past name: British Journal of Applied Science & Technology, Past ISSN: 2231-0843, NLM ID: 101664541)

Derivation of All Particular Solutions of a 'Big' Boolean Equation with Applications in Digital Design

Ali Muhammad Rushdi[1*] and Sultan Sameer Zagzoog[1]

[1]Department of Electrical and Computer Engineering, King Abdulaziz University, P.O.Box 80204, Jeddah 21589, Saudi Arabia.

Authors' contributions

This work was carried out in collaboration between the two authors. Author AMR designed the study, performed the analysis, solved the examples and wrote the manuscript. Author SSZ managed the literature search and drew the figures. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/CJAST/2018/41481
Editor(s):
(1) Wei Wu, Professor, Department of Applied Mathematics, Dalian University of Technology, China.
Reviewers:
(1) Samuel Asante Gyamerah, Kwame Nkrumah University of Science and Technology, Ghana.
(2) Prashant Kumar, Zeal College of Engineering and Research, India.
Complete Peer review History: http://www.sciencedomain.org/review-history/24676

Original Research Article

Received 26th February 2018
Accepted 9th May 2018
Published 18th May 2018

ABSTRACT

This paper considers the problem of solving a system of Boolean equations over a finite (atomic) Boolean algebra other than the two-valued one. The paper outlines classical and novel direct methods for deriving the general parametric solution of such a system and for listing all its particular solutions. A detailed example over B_{256} is used to illustrate these two methods as well as a third method that starts by deriving the subsumptive solution first. The example demonstrates how the consistency condition forces a collapse of the underlying Boolean algebra to a subalgebra, and also how to list a huge number of particular solutions in a very compact space. Subsequently, the paper proposes some potential applications for the techniques of Boolean-equation solving. These techniques are very promising as useful extensions of classical techniques based on two-valued Boolean algebra.

**Corresponding author: E-mail: arushdi@yahoo.com, arushdi@kau.edu.sa, arushdi@ieee.org, alirushdi@gmail.com;*

Keywords: Big Boolean algebra; parametric solution; listing of particular solutions; digital design; direct and inverse arithmetic; integer factorization; Diophantine equations.

1. INTRODUCTION

A prominent "misnomer" in mathematical and engineering circles is the term 'Boolean algebra'. This term is widely used to refer to switching algebra, which is just one particular case of a 'Boolean algebra' that has 0 generators, 1 atom and two elements belonging to $\mathrm{B}_2 = \{0, 1\}$. The term 'Boolean algebra' refers to an algebra of a finite or infinite cardinality [1-3]. The term 'finite Boolean algebra' covers, in fact, a countably infinite number of atomic algebras described by natural numbers n $(n \geq 0)$, such that an algebra has n generators, $\mathrm{N} = 2^n$ atoms, and $2^{\mathrm{N}} = 2^{2^n}$ elements. The inadvertent use of the general term Boolean algebra to refer to its particular case of a switching algebra B_2 leads to many problems and misconceptions, since the switching algebra is much simpler than a generalized finite (atomic) Boolean algebra. Therefore, many authors [4-14] started to label a finite Boolean algebra other than B_2 (i.e., one with n generators $(n > 0)$) as a "big" Boolean algebra. Table 1 (which originally appeared in [14]) lists some finite Boolean algebras specifying the numbers of generators, atoms and elements for each algebra. It is clear from the table that there is a finite Boolean algebra for every nonnegative integer, and that B_2 is just the smallest and simplest member among an infinite multitude of finite (atomic) Boolean algebras.

Any system of 'big' Boolean equations can be reduced to a single Boolean equation $\{g(\boldsymbol{Z}) = 1\}$ or $\{\bar{g}(\boldsymbol{Z}) = 0\}$ [4,5]. The main types of solutions of such a Boolean equation are the general solutions, (which could be subsumptive or parametric) and the particular solutions. The reader is referred to a plethora of modern texts and recent papers [2-18] to understand the meaning of these types and get acquainted with their interrelationships and derivation methods. An exclusive enumeration of particular solutions is obtained from any of the two types of general solutions *via* an expansion tree. The number of children nodes for any parent node is equal to the number $2^{\mathrm{N}} = 2^{2^n}$ of elements of the underlying Boolean algebra, possibly divided by a number of the form $2^k, k = 0,1, \ldots, \mathrm{N}$ [7]. In particular, in the conventional method for producing a general parametric solution, the number of parameters used is minimized [4,5,14] producing compact algebraic solutions with parameters belonging to the underlying Boolean algebra. Contrarily to this convention, Rushdi and Ahmad [10,12,16] proposed a novel method for producing a general parametric solution that does not attempt to minimize the number of parameters used, but instead used independent parameters belonging to the two-valued Boolean algebra B_2 for each asserted atom that appears in the discriminants of the function $g(\boldsymbol{Z})$. The parametric solution obtained sacrifices minimality of parameters and algebraic expressions for ease, compactness and efficiency in listing all particular solutions. These solutions are given by permutative additive formulas expressing a weighted sum of asserted atoms of $g(\boldsymbol{Z})$, with the weight of every atom (called its contribution) having a number of alternative possible values equal to the number of appearances of the atom in the discriminants of $g(\boldsymbol{Z})$. These alternatives are based on a set of orthonormal tags, and hence could be listed in a rectangle divided into disjoint cells. This rectangle resembles a Karnaugh map, and is, in fact, a Karnaugh map, possibly with some adjacent cells combined. The representation suggested allows the possibility of listing a huge number of particular solutions within a very small space. The reason of this possibility is that an arbitrarily-selected contribution of a particular atom can be combined with any of the possible contributions of each of the other atoms. The combination via the additive (ORing) operation is simple and straightforward.

The organization of the remainder of this paper is as follows. Section 2 reviews the conventional and novel direct methods for deriving a general parametric solution and listing all particular solutions for a system of Boolean equations. Section 3 presents a detailed illustrative example in which a general parametric solution is obtained *via* three methods, which are (a) a direct compact solution, (b) a compact solution obtained indirectly *via* a subsumptive solution, and (c) a direct permutative additive solution leading to compact listing of all particular solutions. Section 4 proposes a set of possible applications in digital design which entail direct and inverse arithmetic operations. Section 5 concludes the paper.

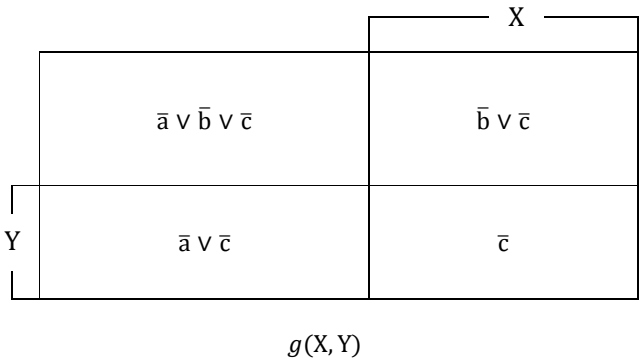


Fig. 1. The natural map (VEKM) for the function g(X,Y) equated to 1 in equation (29)

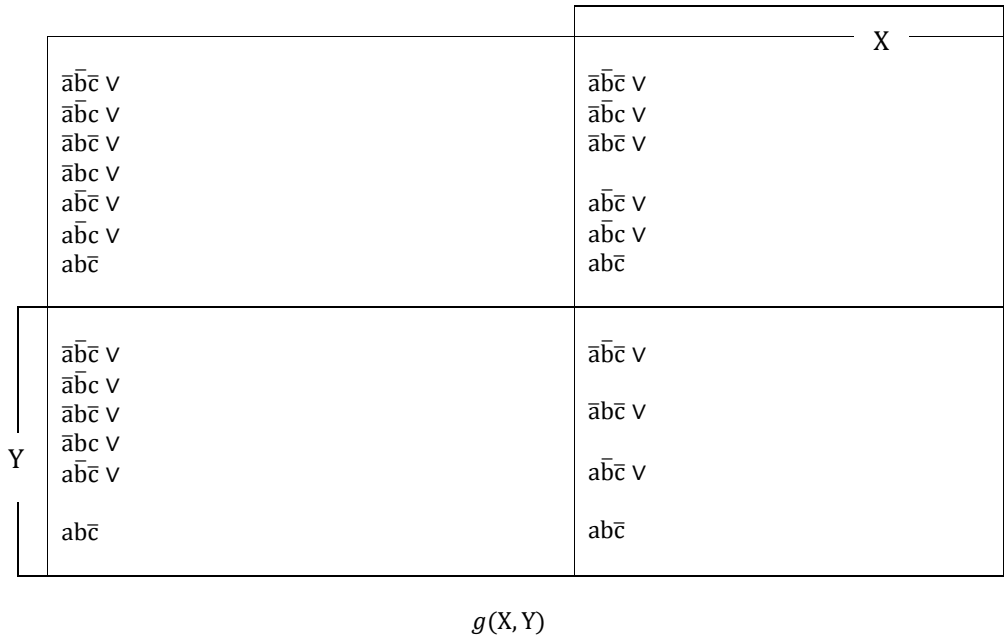


Fig. 2. The natural map of Fig. 1 redrawn with entries written as minterm expansions, or equivalently as disjunctions of atoms of $\mathbf{FB(a, b, c)}$

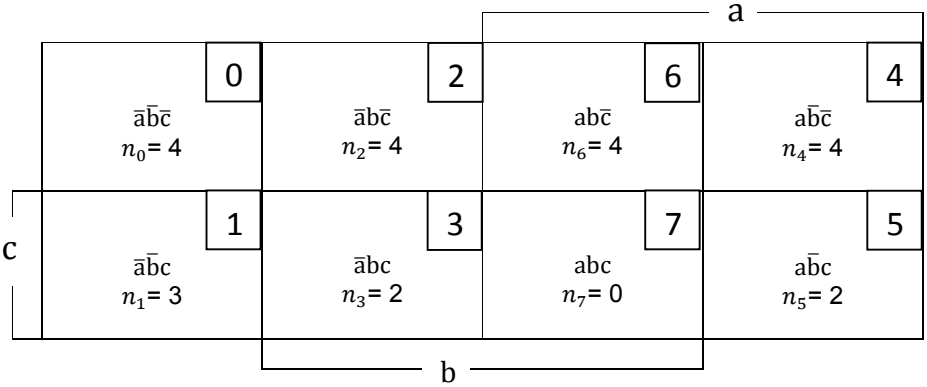
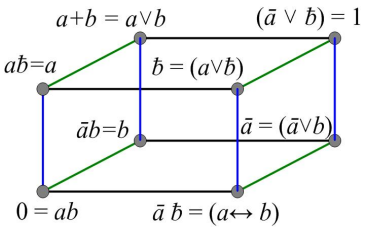
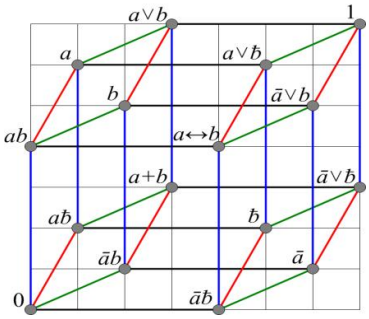


Fig. 3. Karnaugh-map listing of the number of appearances $n_i (0 \leq n_i \leq 4)$ of each of the eight atoms i of $\mathbf{FB(a, b, c)}$ in the four cells of the map in Fig. 1



The lattice of B_{16}, collapsed under the condition ab = 0 so as to represent B_8



A hypercube lattice indicating the partial ordering among the 16 elements of B_{16}

Fig. 4. Visualization of the concept of algebra collapse (the hypercube $\mathbf{B_{16}}$ collapses to a cube $\mathbf{B_8}$ when one of its four atoms (here ab) is nullified

	X	
	$\bar{a}\bar{b}c \vee \bar{a}bc \vee a\bar{b}c \vee \bar{c}$	$\bar{a}\bar{b}c \vee a\bar{b}c \vee \bar{c}$
Y	$\bar{a}\bar{b}c \vee \bar{a}bc \vee \bar{c}$	$\bar{c}$

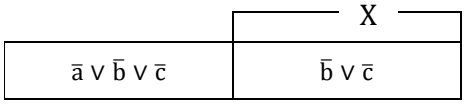
$g(\mathrm{X}, \mathrm{Y})$

Fig. 5. The natural map in Fig. 2 with the four atoms $\bar{a}\bar{b}\bar{c}, \bar{a}b\bar{c}, a\bar{b}\bar{c}$ and $ab\bar{c}$ combined as $\bar{c}$. Such a combination is equivalent to using the same set of orthonormal tags $\{\bar{u}\bar{v}, \bar{u}v, u\bar{v}, uv\}$ individually in the same way with instances of each of these atoms or collectively with their total disjunction $\bar{c}$

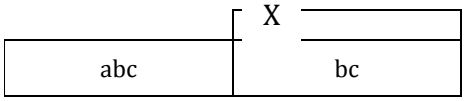
	X	
	$\bar{a}\bar{b}c\ (\bar{u}\bar{v} \vee uv) \vee$ $\bar{a}bc\ (u \vee \bar{v}) \vee$ $a\bar{b}c\ (\bar{u} \vee v) \vee$ $\bar{c}\ (\bar{u}\bar{v}) \vee$ $d(\mathrm{abc})$	$\bar{a}\bar{b}c\ (u\bar{v}) \vee$ $a\bar{b}c\ (u\bar{v}) \vee$ $\bar{c}\ (u\bar{v}) \vee$ $d(\mathrm{abc})$
Y	$\bar{a}\bar{b}c\ (\bar{u}v) \vee$ $\bar{a}bc\ (\bar{u}v) \vee$ $\bar{c}\ (\bar{u}v) \vee$ $d(\mathrm{abc})$	$\bar{c}\ (uv) \vee$ $d(\mathrm{abc})$

$\mathrm{G}_1(\mathrm{X}, \mathrm{Y};\ \mathrm{a}, \mathrm{b}, \mathrm{c};\ u, v)$

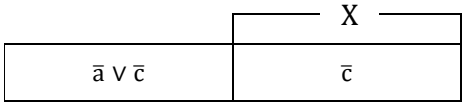
Fig. 6. The auxiliary function $\mathbf{G_1}$ with instances of each atom tagged by members of an orthonormal set. Common parameters are used for different atoms



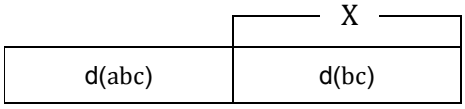
$g(\mathrm{X},0)$



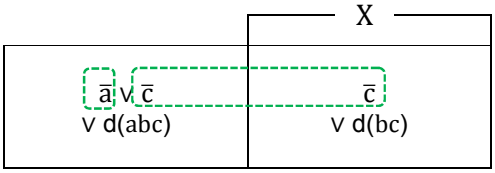
$\bar{g}(\mathrm{X},0)$



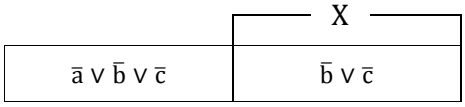
$g(\mathrm{X},1)$



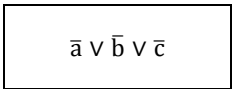
$s_{\mathrm{Y}} = \bar{g}(\mathrm{X},0)g(\mathrm{X},1) \vee \mathrm{d}\big(\bar{g}(\mathrm{X},0)\big) = 0$



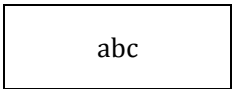
$t_{\mathrm{Y}} = g(\mathrm{X},1) \vee \mathrm{d}\big(\bar{g}(\mathrm{X},0)\big) = \bar{\mathrm{c}} \vee \bar{\mathrm{a}}\bar{\mathrm{X}}$



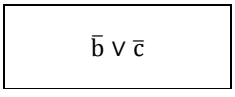
$g_1(X) = g(\mathrm{X},0) \vee g(\mathrm{X},1) = 0$



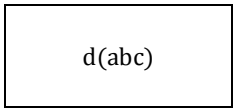
$g_1(0)$



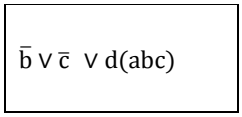
$\overline{g_1}(0)$



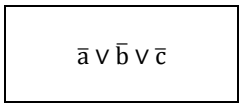
$g_1(1)$



$$s_{\mathrm{X}} = \overline{g_1}(0)\; g_1(1) \vee \mathrm{d}\left(\overline{g_1}(0)\right) = 0$$

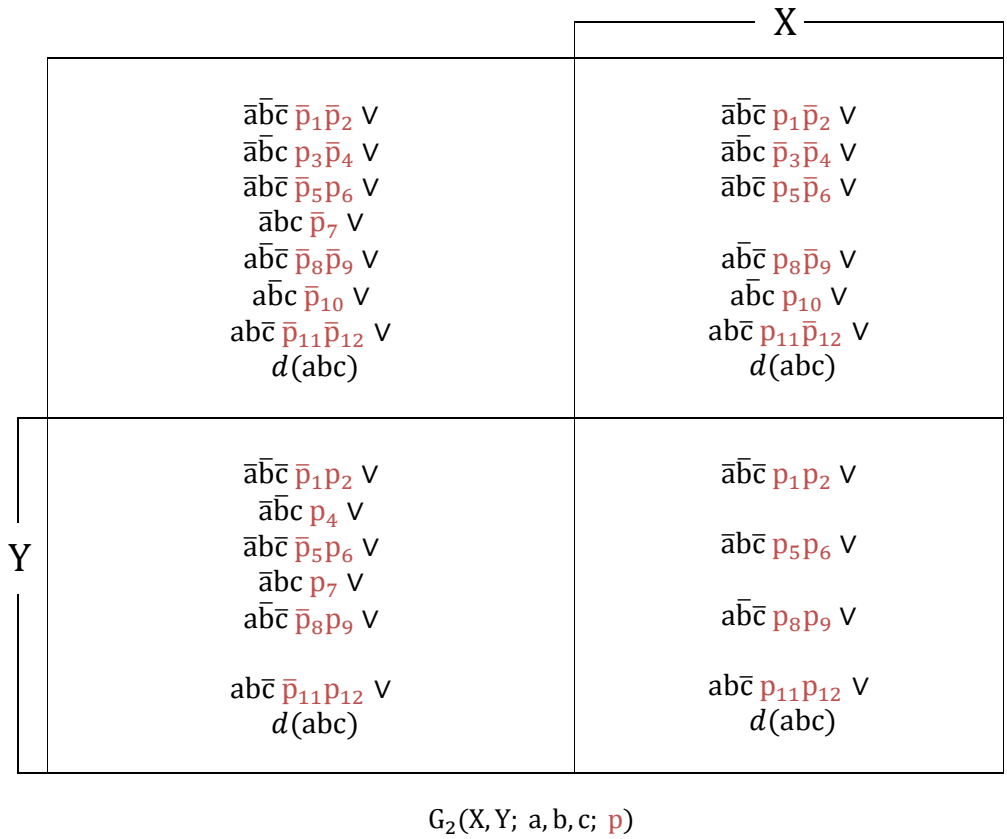


$$t_{\mathrm{X}} = \; g_1(1) \vee \mathrm{d}\left(\overline{g_1}(0)\right) = \bar{\mathrm{b}} \vee \bar{\mathrm{c}}$$



$$g_0 = \; g_1(0) \vee g_1(1) = 1$$

Fig. 7. Various VEKMs used in the derification of the most compact subsumptive solution of (29)



$$\mathrm{G}_2(\mathrm{X}, \mathrm{Y};\; \mathrm{a}, \mathrm{b}, \mathrm{c};\; \mathrm{p})$$

Fig. 8. The auxiliary function G_2 with instances of each atom tagged by members of an orthonormal set. Independent parameters are used for different atoms, and hence combining the four $\bar{\mathrm{c}}$ atoms does not work any more

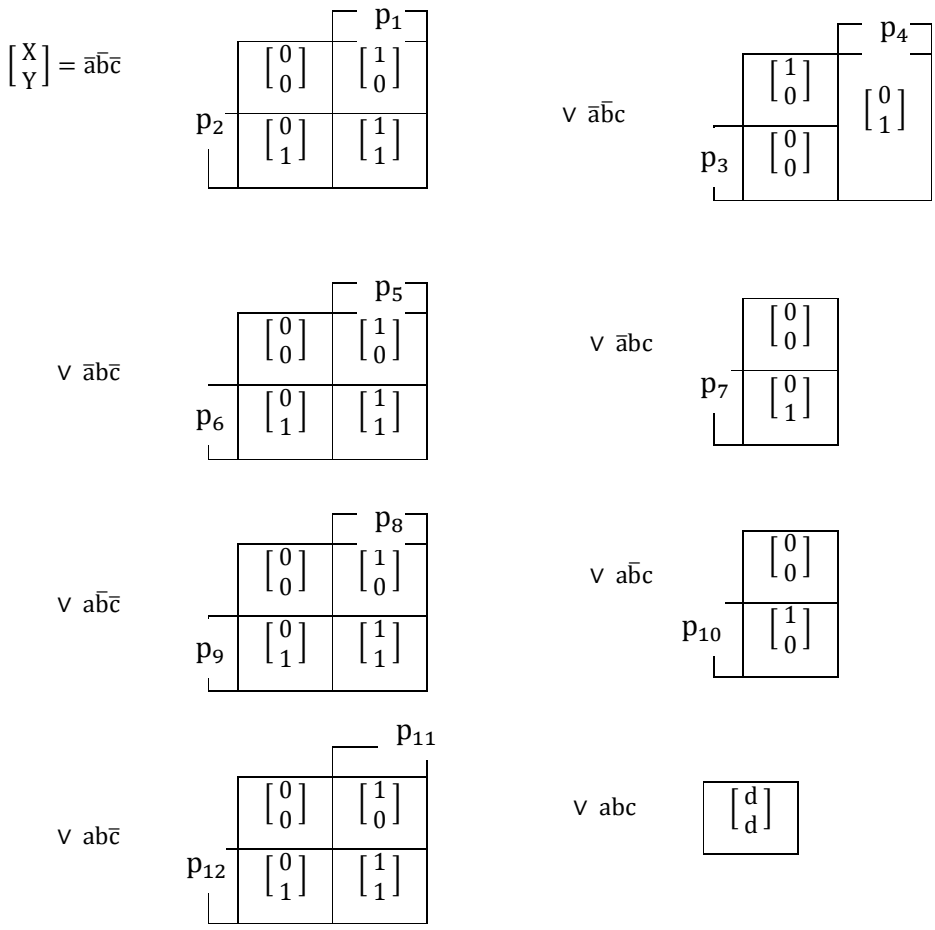


Fig. 9. A permutative additive formula listing all the 3072 particular solutions of Equation (29)

2. DIRECT METHODS FOR CONSTRUCTING PARAMETRIC SOLUTIONS

2.1 Reduction of a System of Boolean Equations into a Single Equation

Consider a system of l Boolean equations of the form

$$\boldsymbol{s}(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \boldsymbol{t}(\mathbf{X},\mathbf{Y},\mathbf{Z}) \quad (1)$$

Here, the vectors $\mathbf{X},\mathbf{Y},\mathbf{Z},\boldsymbol{s},\text{and }\boldsymbol{t}$ belong to $\mathrm{B}_2^k,\mathrm{B}_2^m,\mathrm{B}_2^n,\mathrm{B}_2^l,\text{and }\mathrm{B}_2^l$, respectively. The input arguments in (1) are partitioned into inputs $\mathbf{X}$ (typically controllable), outputs $\mathbf{Z}$ (typically observable), and intermediate variables $\mathbf{Y}$ (frequently neither controllable nor observable, and hence need to be dispensed with in same way). The system (1) can be expanded into scalar equations of the form

$$s_i(\mathbf{X},\mathbf{Y},\mathbf{Z}) = t_i(\mathbf{X},\mathbf{Y},\mathbf{Z}), \qquad 1 \le i \le l, \quad (2)$$

where occasionally we might have $t_i = 0$ or $t_i = 1$. The system (1) of equations is exactly equivalent to a single Boolean equation of the form

$$h(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}) = 1\,, \quad (3)$$

Or

$$r(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}) = 0\,, \quad (4)$$

Where

$$h(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}) \equiv \wedge_{i=1}^{l}(s_i \odot t_i), \quad (5)$$

and

$$r(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}) = \bar{h}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}) \equiv \vee_{i=1}^{l}(s_i \oplus t_i). \quad (6)$$

The symbols ∧, ∨, ⊕, and ⊙ in Equations (5) and (6) depict the AND operator, the OR operator, the XOR (Exclusive-OR) operator and the XNOR (coincidence or equivalence) operator, respectively.

2.2 Suppression of Intermediate (undesirable) Variables

This subsection offers a means to dispense with the undesirable variables $\mathbf{Y}$. This means is called "suppression" rather than "elimination" since the latter term is reserved for another technical meaning [4]. The *resultant of suppression* of the intermediate variables $\mathbf{Y}$ from the Boolean equation (4) (called the parent equation) is the derived Boolean equation [17].

$$f(\boldsymbol{X},\boldsymbol{Z}) = 0\,, \tag{7}$$

Where

$$f(\boldsymbol{X},\boldsymbol{Z}) \equiv \vee_{\mathbf{A}\in\{0,1\}^m}\ r(\boldsymbol{X},\boldsymbol{A},\boldsymbol{Z})\,. \tag{8}$$

The solutions of the derived equation (7) are exactly those of the parent equation (4) that do not involve the suppressed variables $\mathbf{Y}$ [17]. Dually, if equation (3) is used as a parent equation, then the resultant of suppression of the variables $\boldsymbol{Y}$ is now the derived Boolean equation [12].

$$g(\boldsymbol{X},\boldsymbol{Z}) = 1\,, \tag{9}$$

Where

$$g(\boldsymbol{X},\boldsymbol{Z}) \equiv \wedge_{\mathbf{A}\in\{0,1\}^m}\ h(\boldsymbol{X},\boldsymbol{A},\boldsymbol{Z})\,. \tag{10}$$

and the solutions of the derived equation (9) are exactly those of the parent equation (3) that do not involve the suppressed variables $\mathbf{Y}$ [12].

2.3 Algebraic Construction of Parametric Solutions

We seek solutions of the Boolean equation (9) where $g(\boldsymbol{X},\boldsymbol{Z})\colon B_2^{k+n}\to B_2$, is a two-valued Boolean function of k two-valued variables $\boldsymbol{X} = [\,X_1\,X_2\ldots\,X_k\,]^T$ and n two-valued variables $\boldsymbol{Z} = [\,Z_1\,Z_2\ldots\,Z_n\,]^T$. However, we do not need a listing of binary solutions for $\boldsymbol{X}$ and $\boldsymbol{Z}$, but instead we want to express $\boldsymbol{Z}$ in terms of $\boldsymbol{X}$. We view $g(\boldsymbol{X},\boldsymbol{Z})$ as $g(\boldsymbol{X};\boldsymbol{Z})$ or simply $g(\boldsymbol{Z})$ and rewrite (9) as

$$g(\boldsymbol{Z}) = 1, \tag{11}$$

where $g(\boldsymbol{Z})\colon B_{2^K}^n\to B_{2^K}$, and B_{2^K} is the free Boolean algebra $FB(X_1,X_2\ldots\ldots X_k)$ with k generators (namely, X_1, X_2,…, X_k), $K = 2^k$ atoms and $2^K = 2^{2^k}$ elements. Now we express $g(\boldsymbol{Z})$ by its Minterm Canonical Form (MCF), or Minterm Expansion [4, 12]

$$g(\boldsymbol{Z}) \equiv \vee_{\mathbf{A}\in\{0,1\}^n}\ g(\boldsymbol{A})\,\boldsymbol{Z}^{\boldsymbol{A}}\,. \tag{12}$$

For

$\boldsymbol{Z} = [\,Z_1\,Z_2\ldots\,Z_n\,]^T \in B_{2^K}^n$,
$\boldsymbol{A} = [\,a_1\,a_2\ldots\,a_n\,]^{\boldsymbol{T}} \in \{0,1\}^n$, the symbol $\boldsymbol{Z}^{\boldsymbol{A}}$ is defined as

$$\boldsymbol{Z}^{\boldsymbol{A}} = {Z_1}^{a_1}\ {Z_2}^{a_2}\ldots\ {Z_n}^{a_n}, \tag{13}$$

Where

$$Z_i^{a_i} = Z_i\ \odot a_i = \begin{cases}\overline{Z_\iota}\,, & when\ \ a_i = 0\\ Z_i\,, & when\ \ a_i = 1\end{cases} \tag{14}$$

For $\mathbf{A}\in\{0,1\}^n$, the symbol $\boldsymbol{Z}^{\boldsymbol{A}}$ spans the minterms of $\boldsymbol{Z}$, which are the 2^n elementary or primitive products

$$\overline{Z_1}\,\overline{Z_2}\ldots\overline{Z_{n-1}}\,\overline{Z_n},\quad \overline{Z_1}\,\overline{Z_2}\ldots\overline{Z_{n-1}}\,Z_n,\quad \ldots.,\quad Z_1\,Z_2\ldots Z_{n-1}\,Z_n. \tag{15}$$

The constant values $g(\boldsymbol{A})$ in equation (7) are elements of B_{2^K} called the discriminants of $g(\boldsymbol{Z})$. These discriminants are the entries of the natural map (VEKM) of $g(\boldsymbol{Z})$ which has an input domain $\{0,1\}^n \subseteq B_{2^K}^n$. The Boolean algebra $B_{2^K} = FB(X_1,X_2\ldots\ldots X_k)$ has generators X_i ($1\le i\le k$) which look like variables. We now observe that the minterms of $\boldsymbol{X}$, which are the $2^k = k$ elementary or primitive products

$$\overline{\mathrm{X}_1}\,\overline{\mathrm{X}_2}\ldots\overline{\mathrm{X}_{\mathrm{k}-1}}\,\overline{\mathrm{X}_\mathrm{k}},\quad \overline{\mathrm{X}_1}\,\overline{\mathrm{X}_2}\ldots\overline{\mathrm{X}_{\mathrm{k}-1}}\,\mathrm{X}_\mathrm{k},\quad \ldots,\quad \mathrm{X}_1\,\mathrm{X}_2\ldots\mathrm{X}_{\mathrm{k}-1}\,\mathrm{X}_\mathrm{k}, \tag{16}$$

are exactly the atoms of the underlying Boolean algebra, to be called T_i $(0\le i\le (K-1))$. The function $g(\boldsymbol{A})$ is given by

$$g(\boldsymbol{A}) = \vee_{i=0}^{K-1}\ (e_i(\boldsymbol{A})\wedge T_i\,), \tag{17}$$

where the symbol $e_i(\boldsymbol{A})$ denotes an indicator of the event that atom T_i appears in the expression of $g(\boldsymbol{A})$, *i.e.,* namely, Note that if a specific atom

T_i dose not appear in $g(\boldsymbol{A})$, then $e_i(\boldsymbol{A}) = g(\boldsymbol{A})/\,T_i$ is necessarily 0 since $g(\boldsymbol{A})$ is a disjunction of atoms that are all orthogonal to T_i. Atom T_i appears in the cell $\boldsymbol{A}$ of the natural map for $g(\boldsymbol{Z})$.

$$e_i(\boldsymbol{A}) = \begin{Bmatrix} 1, & if\ T_i \rightarrow g(\boldsymbol{A}) \\ 0, & otherwise \end{Bmatrix} = g(\boldsymbol{A})/T_i \quad (18)$$

where the symbol $(r\ /\ s) = (r)_{s=1}$ denotes the Boolean quotient of r by s [4,9]. Now, we define n_i $(0 \leq n_i \leq 2^n)$ as the total number of actual appearances of T_i in the expression (17) for $g(\boldsymbol{A})$, *i.e.*,

$$n_i = \sum_{\mathbf{A} \in \{0,1\}^n} e_i(\boldsymbol{A})\,. \quad (19)$$

The total number $N_{unconditional}$ of unconditional particular solutions of (9) over B_{2^K} (as it is) is given by

$$N_{unconditional} = \prod_{i=0}^{K-1} n_i. \quad (20)$$

This number is zero if some $n_i = 0$. To avoid such a situation, we require the *consistency condition* that all atoms T_i such that $n_i = 0$ must be forbidden or nullified. This means that the underlying Boolean algebra loses these atoms and hence collapses to one of its strict subalgebras. The number of solutions over this reduced Boolean algebra is

$$N_{conditional} = \prod_{\substack{i=0 \\ n_i \neq 0}}^{K-1} n_i. \quad (21)$$

Now we introduce a set of parameters $\boldsymbol{p}_i$ $(0 \leq i \leq (K-1),\ n_i \neq 0)$ to construct an orthonormal set of tags to attach to instances of appearances of the asserted atom T_i in the discriminants $g(\boldsymbol{A})$. The number of parameters for atom T_i (the length of vector $\boldsymbol{p}_i$) is given by

$$l(\boldsymbol{p}_i) = \lceil \log_2 n_i \rceil, \quad 0 \leq i \leq (K-1), \quad n_i \neq 0. \quad (22)$$

The parameters $\boldsymbol{p}_i$ can be used to generate a set of $n_i \leq 2^{l(\boldsymbol{p}_i)}$ orthonormal tags $\{t_1, t_2 \ldots t_{n_i}\}$, such that

$$t_1 \vee t_2 \vee \ldots \vee t_{n_i} = 1, \quad (23)$$

$$t_{j_1} \wedge t_{j_2} = 0 \qquad \forall\, j_1, j_2 \in \{1, 2, \ldots, n_i\}. \quad (24)$$

When $n_i = 2^{l(\boldsymbol{p}_i)}$ the set of orthonormal tags can be visualized as the products of cells in a Karnaugh map whose map variables are the underlying parameters. If $2^{l(\boldsymbol{p}_i)-1} < n_i < 2^{l(\boldsymbol{p}_i)}$, some cells of such a map are merged, and the map reduces to a map-like structure [10,12,16].

When each appearance of an atom T_i is tagged by a particular member of its orthonormal set of tags, an auxiliary function $G(\boldsymbol{Z}, \boldsymbol{p_i})$ $(0 \leq i \leq K-1,\ ni \neq 0$ results. The parametric solution is now given by [5,7,12,13].

$$Z_u = \vee_{\{\mathbf{A} \in \{0,1\}^n | A_u = 1\}}\ G(\boldsymbol{A}, \boldsymbol{p_i}).\ 1 \leq u \leq, (0 \leq i \leq K-1,\ ni \neq 0. \quad (25)$$

The total number E of parameters used in (25) to construct the tags for all atoms is given by [10,12,16].

$$E = \sum_{i=1}^{k} l(\boldsymbol{p}_i) = \sum_{i=1}^{k} \lceil \log_2(n_i) \rceil. \quad (26)$$

The conventional method is to select the parameter vectors from a shared pool of parameters so as to minimize the number of parameters used. This number is now rewritten as E' given by [10,12,14].

$$E' = \max_i\ l(\boldsymbol{p}_i) = \max_i \lceil \log_2 n_i \rceil = \left\lceil \log_2\ (\max_i n_i) \right\rceil. \quad (27)$$

However, parameters used must then belong to the underlying Boolean algebra (possibly collapsed due to the consistency condition). We now propose to use independent parameters $\boldsymbol{p}_i$ for each atom T_i $(0 \leq i \leq K-1,\ n_i \neq 0)$. The expressions (25) will not be as compact as they are in the conventional case, but the independent parameters $\boldsymbol{p}_i$ now belong to the two-valued Boolean algebra B_2 [5,7], a fact that facilitates the generation of all particular solutions as has been documented by Rushdi & Ahmad [10,12,14] and as will be seen in the next section.

3. ILLUSTRATIVE EXAMPLE

The problem studied in this section is taken from an old text on Boolean algebra [19] that supplied a general parametric "solution" *via* a non-constructive theorem-proof technique. However, this alleged solution fails to satisfy the equation it is intended to solve.

Consider the Boolean function

$$f(\mathrm{X}, \mathrm{Y}) = \mathrm{c}(\mathrm{a} \vee \mathrm{x})(\mathrm{b} \vee \mathrm{y}) \quad (28)$$

where $= \mathrm{B}_{256}^2 \rightarrow \mathrm{B}_{256}$, and $\mathrm{B}_{256} = \mathrm{FB}(\mathrm{a}, \mathrm{b}, \mathrm{c})$. A solution of the equation $\{f = 0\}$ expresses the dependent variables X and Y in terms of the

independent " variables" a, b and c which are treated herein as generators of the underlying Boolean algebra. We complement the function f to obtain $g = \bar{f}$ so as to solve the equivalent equation

$$g(\mathrm{X},\mathrm{Y}) = \bar{\mathrm{c}} \vee \bar{\mathrm{a}}\bar{\mathrm{X}} \vee \bar{\mathrm{b}}\bar{\mathrm{Y}} = 1 \qquad (29)$$

Where $g = \mathrm{B}_{256}^2 \to \mathrm{B}_{256}$. In the next three subsections, we offer three solutions of (29). These are a conventional parametric solution obtained directly, a conventional parametric solution obtained via a subsumptive one, and a permutative additive parametric solution.

3.1 A Conventional Parametric Solution Obtained Directly

The Boole-Shannon expansion of g *w. r. t.* its two arguments X and Y is

$$g(\mathrm{X},\mathrm{Y}) = (\bar{\mathrm{a}} \vee \bar{\mathrm{b}} \vee \bar{\mathrm{c}})\bar{\mathrm{X}}\,\bar{\mathrm{Y}} \vee (\bar{\mathrm{a}} \vee \bar{\mathrm{c}})\bar{\mathrm{X}}\,\mathrm{Y} \vee (\bar{\mathrm{b}} \vee \bar{\mathrm{c}})\mathrm{X}\,\bar{\mathrm{Y}} \vee (\bar{\mathrm{c}})\mathrm{X}\,\mathrm{Y} \qquad (30)$$

and hence its natural map (variable-entered Karnaugh map (VEKM)) is as shown in Fig. 1. Each of the entries of this map is a function of the "entered variables" or generators a, b and c and is a disjunction of some of the eight atoms of $\mathrm{FB}(\mathrm{a},\mathrm{b},\mathrm{c})$ as shown in Fig. 2. The numbers of appearances of these atoms in the cells of the map of Fig. 2 are listed in Fig. 3, which immediately shows that:

1. The atom abc does not appear at all in any of the cells of the map in Fig. 2. This means that this atom must be nullified, *i.e.,* the consistency condition of equation (29) is

$$\mathrm{abc} = 0 \qquad (31)$$

When the Boolean algebra $\mathrm{B}_{256} = \mathrm{FB}(\mathrm{a},\mathrm{b},\mathrm{c})$ loses its abc atom, it *collapses* into a subalgebra of 7 atoms only, *i.e.,* it collapses to B_{128}. The sizes of B_{256} and B_{128} are too large to be amenable to visualization. However, to give the reader a glimpse of the meaning of algebra collapse, we present in Fig. 4 a hypercube lattice representing B_{16}, and then represent its collapse to B_8 when it loses the ab atom.

2. The number of particular solutions of (29) is the product of the numbers of appearances of asserted atoms in Fig. 2, namely

$$\mathrm{N}_{particular} = 2^2 * 3^1 * 4^4 = 3072 \qquad (32)$$

The minimum number of parameters k needed for a parametric solution of (29) is

$$k = \lceil \log_2 4 \rceil = 2 \qquad (33)$$

In fact, this number is expected to be less than or equal to the number of variables involved [4], which is $n = 2$ herein.

To facilitate obtaining a parametric solution with a minimum number of parameters, we note that each of four atoms $\bar{\mathrm{a}}\bar{\mathrm{b}}\bar{\mathrm{c}}$, $\bar{\mathrm{a}}\mathrm{b}\bar{\mathrm{c}}$, $\mathrm{a}\bar{\mathrm{b}}\mathrm{c}$, and $\mathrm{ab}\bar{\mathrm{c}}$ is omnipresent in the map of Fig. 2, and might be combined into $\bar{\mathrm{c}}$, as shown in Fig. 5. Note that such a combination is permissible since it is equivalent to using the same set of orthonormal tags individually in the same way with instances of each of these atoms or collectively with their total disjunction $\bar{\mathrm{c}}$. In Fig. 6, we construct an auxiliary function $\mathrm{G}_1(\mathrm{X},\mathrm{Y};\ \mathrm{a},\mathrm{b},\mathrm{c};\ u,v)$, where we attach tags from the orthonormal set $\{\bar{u}\bar{v}, \bar{u}v, u\bar{v}, uv\}$ to the term $\bar{\mathrm{c}}$ (that has 4 appearances), attach tags from the orthonormal set $\{(\bar{u}\bar{v} \vee uv), \bar{u}v, u\bar{v}\}$ to the atom $\bar{\mathrm{a}}\bar{\mathrm{b}}\mathrm{c}$ (that has 3 appearances), and attach tags from the orthonormal set $\{(\bar{u} \vee v), u\bar{v}\}$ to atom $\mathrm{a}\bar{\mathrm{b}}\mathrm{c}$ (that has 2 appearances). We have chosen the orthonormal sets above with an eye on getting the most compact solution. In fact, we do not care about how cumbersome the entries in the $\bar{\mathrm{X}}\,\bar{\mathrm{Y}}$-cell are, since they do not affect the final solution. However, we have only a single tag per each of the other three cells, namely tag $\bar{u}v$ in the $\bar{\mathrm{X}}\mathrm{Y}$-cell, tag $u\bar{v}$ in the $\mathrm{X}\bar{\mathrm{Y}}$-cell, and tag uv in the XY-cell. We might add the nullified atom abc don't-care in each of the cells of Fig. 5. Our final solution is

$$\mathrm{X} = (\bar{\mathrm{a}}\bar{\mathrm{b}}\mathrm{c} \vee \mathrm{a}\bar{\mathrm{b}}\mathrm{c} \vee \bar{\mathrm{c}})u\bar{v} \vee \bar{\mathrm{c}}uv \vee \mathrm{d}(\mathrm{abc}) \qquad (33a)$$

$$\mathrm{Y} = (\bar{\mathrm{a}}\bar{\mathrm{b}}\mathrm{c} \vee \bar{\mathrm{a}}\mathrm{bc} \vee \bar{\mathrm{c}})\bar{u}v \vee \bar{\mathrm{c}}uv \vee \mathrm{d}(\mathrm{abc}) \qquad (33b)$$

These formulas might be simplified by ignoring the don't-care parts and involving the reflection law to obtain

$$\mathrm{X} = u(\bar{\mathrm{b}}\mathrm{c}\bar{v} \vee \bar{\mathrm{c}}(\bar{v} \vee v)) = u(\bar{\mathrm{b}}\bar{v} \vee \bar{\mathrm{c}}) \qquad (34a)$$

$$\mathrm{Y} = v(\bar{\mathrm{b}}\mathrm{c}\bar{u} \vee \bar{\mathrm{c}}(\bar{u} \vee u)) = v(\bar{\mathrm{a}}\bar{u} \vee \bar{\mathrm{c}}) \qquad (34b)$$

Substitution of the solution (34) in (28) yields

$$f(\mathrm{X},\mathrm{Y}) = \mathrm{c}(\mathrm{a} \vee u\bar{\mathrm{b}}\bar{v} \vee u\bar{\mathrm{c}})(\mathrm{b} \vee v\bar{\mathrm{a}}\bar{u} \vee v\bar{\mathrm{c}}) = \mathrm{abc} = 0 \qquad (35)$$

while its substitution in (29) yields

$$g(\mathrm{X},\mathrm{Y}) = \bar{\mathrm{c}} \vee \bar{\mathrm{a}}(\bar{u} \vee (\mathrm{b} \vee v)\mathrm{c}) \vee \bar{\mathrm{b}}(\bar{v} \vee (\mathrm{a} \vee u)\mathrm{c})$$
$$= \bar{\mathrm{c}} \vee \bar{\mathrm{a}}\bar{u} \vee \bar{\mathrm{a}}\mathrm{b} \vee \bar{\mathrm{a}}v \vee \bar{\mathrm{b}}\bar{v} \vee \mathrm{a}\bar{\mathrm{b}} \vee \bar{\mathrm{b}}u(\bar{u} \vee (\mathrm{b} \vee v)\mathrm{c})$$
$$= \bar{\mathrm{a}} \vee \bar{\mathrm{b}} \vee \bar{\mathrm{c}} = 1 \quad (36)$$

where the consensus $\bar{\mathrm{a}}\bar{\mathrm{b}}$ of $\bar{\mathrm{a}}\bar{u}$ and $\bar{\mathrm{b}}u$ is added to the second line in (36), and then combined with $(\bar{\mathrm{a}}\mathrm{b} \vee \mathrm{a}\bar{\mathrm{b}})$ to produce $(\bar{\mathrm{a}} \vee \bar{\mathrm{b}})$ which then absorbs $(\bar{\mathrm{a}}\bar{u} \vee \bar{\mathrm{a}}v \vee \bar{\mathrm{b}}\bar{v} \vee \bar{\mathrm{b}}u)$. The results of the aforementioned substitution verify the solution and partially explains why the consistency condition is needed.

3.2 A Conventional Parametric Solution Obtained *Via* a Subsumptive Solution

In this subsection, we utilize Variable-Entered Karnaugh Maps (VEKMs) to derive a general subsumptive solution for (29) in the most compact form. Fig. 7 presents the VEKMs used to obtain such a solution according to the method in [18,20]. The final result obtained is

$$0 \leq \mathrm{Y} \leq \bar{\mathrm{c}} \vee \bar{\mathrm{a}}\,\bar{\mathrm{X}} \quad (37a)$$

$$0 \leq \mathrm{X} \leq \bar{\mathrm{b}} \vee \bar{\mathrm{c}} \quad (37b)$$

Subject to the consistency solution

$$\bar{\mathrm{a}} \vee \bar{\mathrm{b}} \vee \bar{\mathrm{c}} = 1 \quad (37c)$$

Note that this consistency condition is equivalent to the one in (31), being the result of complementing both sides in it.

We can convert the subsumptive solution (37) into a parametric one, namely

$$\mathrm{X} = \mathrm{u}(\bar{\mathrm{b}} \vee \bar{\mathrm{c}}) \quad (38a)$$

$$\mathrm{Y} = \mathrm{v}(\bar{\mathrm{c}} \vee \bar{a}\bar{\mathrm{X}}) = \mathrm{v}(\bar{\mathrm{c}} \vee \bar{\mathrm{a}}\bar{u} \vee \bar{\mathrm{a}}\mathrm{bc})$$
$$= \mathrm{v}(\bar{\mathrm{c}} \vee \bar{\mathrm{a}}\bar{u} \vee \bar{\mathrm{a}}\mathrm{b}) \quad (38b)$$

Substituting this solution into (28), one obtains

$$f(\mathrm{X},\mathrm{Y}) = \mathrm{c}\big(\mathrm{a} \vee u\bar{\mathrm{b}} \vee u\bar{\mathrm{c}}\big)(\mathrm{b} \vee v\bar{\mathrm{c}} \vee v\bar{\mathrm{a}}\bar{u} \vee v\mathrm{ab}{=}\mathrm{abc}{=}0 \quad (39)$$

This solution is not symmetric like the one in (34). Hence, it can be used to generate a third solution, *via.,*

$$\mathrm{Y} = \mathrm{v}(\bar{\mathrm{a}} \vee \bar{\mathrm{c}}) \quad (40a)$$
$$\mathrm{X} = \mathrm{u}(\bar{\mathrm{c}} \vee \bar{b}\bar{\mathrm{Y}}) = \mathrm{u}(\bar{\mathrm{c}} \vee \bar{\mathrm{b}}\bar{\mathrm{v}} \vee \mathrm{a}\bar{\mathrm{b}})$$
$$= \mathrm{v}(\bar{\mathrm{c}} \vee \bar{\mathrm{a}}\bar{u} \vee \bar{\mathrm{a}}\mathrm{b}) \quad (40b)$$

Though the solutions (38) and (40) are not symmetric like the one in (34), they enjoy the advantage that in each of them one variable is dependent on a single parameter rather than the two parameters.

3.3 A Permutative Additive Parametric Solution

Despite the elegance, compactness, and symmetry of the solutions in (34), (38) or (40), they are not readily useful for producing a list of all particular solutions, since each of the two parameters u and v should be assigned an independent value that equals a specific element in B_{128}. The expansion tree used for this purpose should explore all $128 \times 128 = 16384$ combinations of (u, v) values, and will finally settle on 3072 solutions. Our alternative method to avoid the use of such an expansion tree is to use independent parameters for each individual atom, as shown in Fig. 8. The number of parameters used increases dramatically from 2 to 12, and though a detailed algebraic solution will be cumbersome when compared with the earlier solutions in (34), (38) or (40), it is nevertheless a permutative additive formula that lists all 3072 particular solutions of equation (29). This formula is given concisely in Fig. 9. To obtain the value for the vector $[\mathrm{X}\,\mathrm{Y}]^{\mathrm{T}}$, one chooses any of the possible values associated with each atom. Two examples of the particular solutions (subject to the condition $\mathrm{abc} = 0$) are

$$\begin{bmatrix}\mathrm{X}\\ \mathrm{Y}\end{bmatrix} = \bar{\mathrm{a}}\bar{\mathrm{b}}\bar{\mathrm{c}}\begin{bmatrix}0\\0\end{bmatrix} \vee \bar{\mathrm{a}}\bar{\mathrm{b}}\mathrm{c}\begin{bmatrix}0\\0\end{bmatrix} \vee \bar{\mathrm{a}}\mathrm{b}\bar{\mathrm{c}}\begin{bmatrix}0\\0\end{bmatrix} \vee \bar{\mathrm{a}}\mathrm{bc}\begin{bmatrix}0\\0\end{bmatrix} \vee \mathrm{a}\bar{\mathrm{b}}\bar{\mathrm{c}}\begin{bmatrix}0\\0\end{bmatrix} \vee \mathrm{a}\bar{\mathrm{b}}\mathrm{c}\begin{bmatrix}0\\0\end{bmatrix} \vee \mathrm{ab}\bar{\mathrm{c}}\begin{bmatrix}0\\0\end{bmatrix} \vee \mathrm{abc}\begin{bmatrix}0\\0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix} \quad (41)$$

$$\begin{bmatrix}\mathrm{X}\\ \mathrm{Y}\end{bmatrix} = \bar{\mathrm{a}}\bar{\mathrm{b}}\bar{\mathrm{c}}\begin{bmatrix}0\\1\end{bmatrix} \vee \bar{\mathrm{a}}\bar{\mathrm{b}}\mathrm{c}\begin{bmatrix}0\\1\end{bmatrix} \vee \bar{\mathrm{a}}\mathrm{b}\bar{\mathrm{c}}\begin{bmatrix}0\\1\end{bmatrix} \vee \bar{\mathrm{a}}\mathrm{bc}\begin{bmatrix}0\\1\end{bmatrix} \vee \mathrm{a}\bar{\mathrm{b}}\bar{\mathrm{c}}\begin{bmatrix}1\\0\end{bmatrix} \vee \mathrm{a}\bar{\mathrm{b}}\mathrm{c}\begin{bmatrix}1\\0\end{bmatrix} \vee \mathrm{ab}\bar{\mathrm{c}}\begin{bmatrix}1\\0\end{bmatrix} \vee \mathrm{abc}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}\mathrm{a}\\ \bar{\mathrm{a}}\end{bmatrix} \quad (42)$$

Note that any of the particular solutions if substituted in (28) reduces it to $\{\mathrm{abc} = 0\}$, and if substituted in (29) reduces it to $\{\bar{\mathrm{a}} \vee \bar{\mathrm{b}} \vee \bar{\mathrm{c}} = 1\}$.

4. PROPOSED APPLICATIONS IN DIGITAL DESIGN

The conventional realm of combinational digital design is the domain of propositional logic or two-valued Boolean algebra. This domain can be (and has been) extended through the use of sequential circuits, evolvable hardware, first-order predicate logic, big Boolean algebras, *etc.* We will now discuss a few possibilities for the use of big Boolean algebras in digital design.

We arbitrarily restrict our discussion here to the design of basic arithmetic circuits (Addition, Subtraction, Multiplication, and Division). However, we note that the subtraction $(\mathbf{X_1} - \mathbf{X_2})$ can be implemented as the addition $(\mathbf{X_1} + (-\mathbf{X_2}))$ through the use of any number system that covers both positive and negative numbers (such as the sign-magnitude system, the once-complements system, or (preferably) the twos-complements system) [21-25]. Therefore, we will not elaborate any more on subtraction since it is simply a form of addition. Similarly we do not consider division since it can be implemented *via* repeated multiplication. Since we are interested in both direct and inverse problems, we need to consider each of the following four cases:

4.1 Direct Addition

In direct addition, we consider a problem of the form

$$\mathbf{Z} = \mathbf{X_1} + \mathbf{X_2} \tag{43}$$

in which the two k-bit numbers $\mathbf{X_1}$ and $\mathbf{X_2}$ are added to produce the n-bit number $\mathbf{Z}$ (where $n = (k+1)$). Typically, (43) is implemented *via* full-adder modules or stages $(0 \le i \le (k-1))$, which obtain (with $\mathrm{Y}_{-1} = 0, \mathrm{Z}_{k+1} = \mathrm{Y}_k$)

$$\mathrm{Z}_i = Sum(\mathrm{X}_{1i}, \mathrm{X}_{2i}, \mathrm{Y}_{i-1}) = \mathrm{X}_{1i} \oplus \mathrm{X}_{2i} \oplus \mathrm{Y}_{i-1} \tag{44a}$$

$$\mathrm{Y}_i = Carry(\mathrm{X}_{1i}, \mathrm{X}_{2i}, \mathrm{Y}_{i-1}) = \mathrm{X}_{1i}\mathrm{X}_{2i} \vee \mathrm{X}_{1i}\mathrm{Y}_{i-1} \vee \mathrm{X}_{2i}\mathrm{Y}_{i-1} \tag{44b}$$

Here, an intermediate variable Y_i is called the carry-out for stage i and also the carry-in for stage $(i+1)$. Full design of the required adder circuit requires combining all the $2k$ conditions in (44a) and (44b) into a single equation of the form (3) and then suppressing the undesirable intermediate variables $\mathbf{Y}$ to obtain an equation of the form (9).

4.2 Inverse Addition

By inverse addition, we mean solving the Diophantine Equation [26-31].

$$\mathbf{X} = \mathbf{Z_1} + \mathbf{Z_2}, \quad \mathbf{Z_1} \ge \mathbf{0},\ \mathbf{Z_2} \ge \mathbf{0} \tag{45}$$

for the two n-bit integers $\mathbf{Z_1}$ and $\mathbf{Z_2}$ given the k-bit integer $\mathbf{X}$ (where $n = k$). Each of $\mathbf{Z_1}$ and $\mathbf{Z_2}$ is a nonnegative integer less than or equal to $\mathbf{X}$ (and hence definitely less than 2^{k} **)**. The integers $\mathbf{Z_1}$ and $\mathbf{Z_2}$ are usually called the additive components of $\mathbf{X}$. There are exactly $\mathbf{X}+1$ solutions to (45), which can be obtained by arbitrarily assigning one of $\mathbf{X}+1$ values to $\mathbf{Z_1}$ $(0 \le \mathbf{Z_1} \le \mathbf{X})$, and then performing the subtraction $(\mathbf{X} - \mathbf{Z_1})$ to obtain $\mathbf{Z_2}$. This procedure might be extended to obtain hardware solvers for more general Diophantine equations.

4.3 Direct Multiplication

In direct multiplication, it is desired to multiply two k-bit integers $\mathbf{X_1}$ and $\mathbf{X_2}$ to produce an n-bit integer $\mathbf{Z}$ (where n = 2k), namely

$$\mathbf{Z} = \mathbf{X_1} * \mathbf{X_2} \tag{46}$$

Typically, this operation is achieved *via* repeated addition. However, it might be implemented directly *via* Boolean-equation solving.

4.4 Inverse Multiplication

By inverse multiplication, we mean factorizing a k-bit positive integer $\mathbf{X}$ into two positive integers $\mathbf{Z_1}$ and $\mathbf{Z_2}$ of sizes n_1 bits and n_2 bits, respectively. Hence, these two integers must satisfy

$$\mathbf{X} = \mathbf{Z_1} * \mathbf{Z_2}, \mathbf{Z_1} > 0,\ \mathbf{Z_2} > 0 \tag{47}$$

The aforementioned problem of Inverse Factorization is of fundamental importance in many serious applications, the most prominent among which is cryptography [32-38].

First, we consider $\mathbf{X}$ to be of an even bit size, say 2*n* bits, where *n* is a positive integer. To avoid factoring $\mathbf{X}$ trivially into a product of itself with 1, we impose the restrictions $(\mathbf{Z_1} > 1)$ and $(\mathbf{Z_2} > 1)$. To avoid duplicate factorizations due to commutativity $(\mathbf{Z_1} * \mathbf{Z_2} = \mathbf{Z_2} * \mathbf{Z_1})$, we impose the additional restriction $(\mathbf{Z_1} \ge \mathbf{Z_2})$. Since $\mathbf{Z_2}$ can be as small as the integer 2, the integer $\mathbf{Z_1}$ can be as large as $(\mathbf{X}/2)$, and hence might occupy up to $(2n-1)$ bits. Since $(\mathbf{Z_1} \ge \mathbf{Z_2})$, the number $\mathbf{X}$ should satisfy $(\mathbf{X} \ge \mathbf{Z_2}^2)$, and hence $\mathbf{Z_2}$ might occupy up to n bits. The sizes of the integers $\mathbf{X}$, $\mathbf{Z_1}$, and $\mathbf{Z_2}$ are therefore $2n$, $(2n-1)$, and n bits, respectively. Using similar reasoning, we can show that if $\mathbf{X}$ has an odd bit size of $(2n-1)$ bits say, then $\mathbf{Z_1}$ and $\mathbf{Z_2}$ are of bit sizes $(2n-2)$ and n respectively. The triple (k, n_1, n_2) of bit sizes for $(\mathbf{X}, \mathbf{Z_1}, \mathbf{Z_2})$ can be either replaced by $(2n, (2n-1), n)$ or by $((2n-1), (2n-2), n)$.

We now start with an initial specification of the problem in the form of an equation

$$g_0(\mathbf{Z_1}, \mathbf{Z_2}) = 1 \quad (48)$$

with the function $g_0: \mathrm{B}^{n_1+n_2} \to \mathrm{B}$ constructed over the 'big' Boolean algebra $\mathrm{B} = \mathrm{FB}(\mathbf{X})$, *i.e.,* it is the free Boolean algebra with *k* generators **X**. The function g_0 is characterized by discriminants given for a specific value of $\mathbf{Z_1}$ and $\mathbf{Z_2}$ by

$$g_0(\mathbf{Z_1}, \mathbf{Z_2}) = \wedge_{i=1}(\mathrm{X}_i \odot \mathrm{X}_i(\mathbf{Z_1}, \mathbf{Z_2})) \quad (49)$$

Where

$$\mathrm{X}_i \odot \mathrm{X}_i(\mathbf{Z_1}, \mathbf{Z_2}) = \mathrm{X}_i^{\mathrm{X}_i(\mathbf{Z_1}, \mathbf{Z_2})} \quad (50)$$

is equal to X_i (uncomplemented) if $\mathrm{X}_i(\mathbf{Z_1}, \mathbf{Z_2}) = 1$ and equals $\overline{\mathrm{X}_i}$ (complemented) if $\mathrm{X}_i(\mathbf{Z_1}, \mathbf{Z_2}) = 0$. To complete the problem specifications, we need to replace g_0 by g given by

$$g(\mathbf{Z_1}, \mathbf{Z_2}) = g_0(\mathbf{Z_1}, \mathbf{Z_2})\, \mathrm{I}(\mathbf{Z_1} > 1)\, \mathrm{I}(\mathbf{Z_2} > 1)\, \mathrm{I}(\mathbf{Z_1} \geq \mathbf{Z_2})\, \mathrm{I}(\mathbf{X} \leq 2^{\mathrm{k}} - 1), \quad (51)$$

where the symbol $\mathrm{I}(\mathrm{event})$ is a Boolean indicator for that event, i.e., it is 1 if the event occurs and 0 if it does not occur. We have already discussed the necessity for the requirements $(\mathbf{Z_1} > 1)$, $(\mathbf{Z_2} > 1)$ and $(\mathbf{Z_1} \geq \mathbf{Z_2})$. The extra condition $\mathrm{I}(\mathbf{X} \leq 2^{\mathrm{k}} - 1)$ is needed to ensure that X is properly represented in k -bits. It is straightforward to note that

$$\mathrm{I}(\mathbf{Z_2} > 1)\, \mathrm{I}(\mathbf{Z_1} \geq \mathbf{Z_2}) \Rightarrow \mathrm{I}(\mathbf{Z_1} > 1) \quad (52)$$

and hence equation (52) is simplified to

$$g(\mathbf{Z_1}, \mathbf{Z_2}) = g_0(\mathbf{Z_1}, \mathbf{Z_2})\, \mathrm{I}(\mathbf{Z_2} > 1)\, \mathrm{I}(\mathbf{Z_1} \geq \mathbf{Z_2})\, \mathrm{I}(\mathbf{X} \leq 2^{\mathrm{k}} - 1), \quad (53)$$

Finally, our design task reduces to finding solutions of $g(\mathbf{Z_1}, \mathbf{Z_2}) = 1$, where $g(\mathbf{Z_1}, \mathbf{Z_2})$ is given by (53).

We already solved smaller versions of this problem when **X** has 3, 4, 5, or 6 bits [11,14]. Our solutions used comparatively large Karnaugh maps of up to 8-variable maps and we are currently investigating the solution of larger problems *via* variable-entered Karnaugh maps [39-44] or *via* an automated implementation of the present algorithm.

Table 1. Relating the numbers of generators, atoms, and elements for finite Boolean algebras

No. of generators	Possible atoms	Nullified atoms	Actual atoms	No. of elements	Comments
0	1	0	1	2	Two-valued Boolean algebra B_2
1	2	0	2	4	---
2	4	1	3	8	---
2	4	0	4	16	---
3	8	3	5	$2^5 = 32$	---
3	8	2	6	$2^6 = 64$	---
3	8	1	7	$2^7 = 128$	---
3	8	0	8	$2^8 = 256$	---
---	---	---	---	---	---
4	16	0	16	$2^{16} = 65536$	Problem in Rushdi and Zagzoog. [11]
---	---	---	---	---	---
5	32	0	32	$2^{32} \approx 4.3 \times 10^9$	---
---	---	---	---	---	---
6	64	0	64	$2^{64} \approx 1.8 \times 10^{19}$	Problem in Rushdi et al. [14]
---	---	---	---	---	---
$n \geq 1$	2^n	$\epsilon\, [0, 2^{n-1} - 1]$	$\mathrm{N} \epsilon\, [2^{n-1} + 1, 2^n]$	2^{N}	General case

5. CONCLUSIONS

This paper offered several novel contributions. First it gave full descriptions and demonstrations of methods to construct parametric solutions and compactly list particular solutions of 'big' Boolean equations. As an offshoot, the paper explained the necessity of imposing a consistency condition and the possible impact of such a condition on collapsing the underlying Boolean algebra to a strictly smaller subalgebra. As a result, the paper set the stage for many useful applications in digital design of arithmetic computer circuits.

The importance of this paper stems from the fact that it is a major step towards full utilization of the mathematics of 'big' Boolean algebras and 'big' Boolean equation-solving. The paper has two distinctive major contributions. It gives a detailed clarifying exposition of modern Boolean mathematics, and it outlines some of the potential applications that could rely on such mathematics. The theoretical development of Boolean mathematics has gone a very long way for the past two centuries, with some notable applications emerging occasionally. As we observed in [14], "it is clear now that Boolean mathematics have matured enough to find significant, diverse, and beneficial applications." Most prominent among the expected applications are those of integer factorization (a core step in cryptanalysis) as well as the solution of general Diophantine Equations.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFRENCES

1. Koppelberg S, Monk JD, Bonnet R. Handbook of Boolean Algebras. Amsterdam: North-Holland. 1989;384.
2. Mendelson E. Boolean algebra and switching circuits. Schaum's Outline Series, McGraw-Hill; 1970.
3. Givant S, Halmos P. Introduction to boolean algebras. Springer Science & Business Media; 2008.
4. Brown FM. Boolean reasoning: The logic of boolean equations. Kluwer Academic Publishers, Boston, USA; 1990.
5. Brown FM. Boolean reasoning: The logic of boolean equations, 2nd Ed. Dover Publications, Mineola, NY, USA; 2003.
6. Rushdi AM, Amashah MH. Parametric general solutions of Boolean equations via variable-entered Karnaugh maps. Journal of Qassim University: Engineering and Computer Sciences. 2010;3(1):59-71.
7. Rushdi AM, Amashah MH. Using variable–entered Karnaugh maps to produce compact parametric general solutions of Boolean equations. International Journal of Computer Mathematics. 2011;88(15):3136-3149.
8. Rushdi AM, Amashah MH. Purely-algebraic versus VEKM methods for solving big Boolean equations. Journal of King Abdulaziz University: Engineering Sciences. 2012;23(2):75-85.
9. Rushdi AMA, Al-Qwasmi MA. Formal derivation of a particular input of a single and (or) gate in terms of its output and other inputs. Journal of King Abdulaziz University: Engineering Sciences. 2015; 26(2):51-64.
10. Rushdi AMA, Ahmad W. Satisfiability in 'big' Boolean algebras via Boolean-equation solving. Journal of King Abdulaziz University: Engineering Sciences. 2016; 28(1):3-18.
11. Rushdi AMA, Zagzoog SS. Design of a digital circuit for integer factorization via solving the inverse problem of logic. Journal of Advances in Mathematics and Computer Science. 2018;26(3):1-14.
12. Rushdi AMA, Ahmad W. Digital circuit design utilizing equation solving over 'big' Boolean algebras. International Journal of Mathematical, Engineering and Management Sciences (IJMEMS). 2018;3(4).
13. Rushdi AMA. Handling generalized type-2 problems of digital circuit design via the variable-entered Karnaugh map. International Journal of Mathematical, Engineering and Management Sciences (IJMEMS). 2018;3(4).
14. Rushdi AMA, Zagzoog SS, Balamesh AS. Design of a hardware circuit for integer factorization using a big Boolean algebra. Journal of Advances in Mathematics and Computer Science. 2018;27(1):1-25.
15. Brown FM. Reduced solutions of Boolean equations. IEEE Transactions on Computers. 1970;C-19(10):976-981.
16. Rushdi AMA, Ahmad W. A novel method for compact listing of all particular solutions of a system of Boolean equations. British Journal of Mathematics & Computer Science. 2017;22(6):1-18.

17. Brown FM. On the suppression of variables in Boolean equations. Discrete Applied Mathematics. 2011;159(5):255-258.
18. Rushdi AM. A comparison of algebraic and map methods for solving general Boolean equations. Journal of Qassim University: Engineering and Computer Sciences. 2012;4(2):1-32.
19. Goodstein RL. Boolean algebra. Pergamon Press, Oxford; 1963.
20. Rushdi AM. Using variable-entered Karnaugh maps to solve boolean equations. International Journal of Computer Mathematics. 2001;78(1):23-38.
21. Abd-El-Barr M, El-Rewini H. Fundamentals of Computer Organization and Architecture. John Wiley & Sons. 2005;38.
22. Hennessy JL, Patterson DA. Computer Architecture: A Quantitative Approach. Elsevier; 2011.
23. Hwang K, Jotwani N. Advanced computer architecture. 3e. McGraw-Hill Education; 2011.
24. Kulisch UW, Miranker WL. Computer Arithmetic in Theory and Practice. Academic press; 2014.
25. Null L, Lobur J. The Essentials of Computer Organization and Architecture. Jones & Bartlett Publishers; 2014.
26. Rushdi AM, Al-Otaibi SO. On limitations of using scalar equations for analyzing synchronous Boolean networks. Journal of King Abdulaziz University: Engineering Sciences. 2008;19(2):41-49.
27. Schroeder M. Diophantine equations, in number theory in science and communications. Fifth Edition, Springer-Verlag, Berlin, Germany. 2009;(Chapter 7): 119-137.
28. Andreescu T, Andrica D. Diophantine equations, in number theory: Structures, examples and problems. Birkhäuser, Boston, USA. 2009;(Chapter 8):145-165.
29. Andreescu T, Andrica D, Cucurezeanu I. An introduction to diophantine equations: A problem-based approach. Birkhäuser, New York, USA; 2010.
30. Rushdi AMA, Alsogati AA. On reduced scalar equations for synchronous Boolean networks. Journal of Mathematics and Statistics. 2013;9(3):262-276.
31. Rushdi AMA. Derivation of reduced scalar equations for synchronous Boolean networks. Journal of King Abdulaziz University: Computer Science and Information Technology. 2015;4(2):39-68.
32. Menezes A, Oorschot P, Vanstone S. Handbook of applied cryptography. CRC Press Company, New York, NY, USA; 1997.
33. John AK, Shah S, Chakraborty S, Trivedi A, Akshay S. Skolem functions for factored formulas. In Proceedings of the 15th conference on formal methods in computer-aided design. FMCAD Inc. 2015; 73-80.
34. Fried D, Tabajara LM, Vardi MY. BDD-based Boolean functional synthesis. In International Conference on Computer Aided Verification. Springer International Publishing. 2016;402-421.
35. Rushdi AMA, Alsheikhy AA. A pedagogical multi-key multi-stage package to secure communication channels. Journal of Qassim University: Engineering and Computer Sciences. 2017;10(2).
36. Akshay S, Chakraborty S, John AK, Shah S. Towards parallel boolean functional synthesis. In international conference on tools and algorithms for the construction and analysis of systems. Springer, Berlin, Heidelberg. 2017;337-353.
37. Ahmad W, Rushdi AMA. A new cryptographic scheme utilizing the difficulty of big Boolean satisfiability. International Journal of Mathematical, Engineering and Management Sciences (IJMEMS). 2018; 3(1):47-61.
Available:www.ijmems.in/ijmems—volumes.html
38. Akshay S, Chakraborty S, Goel S, Kulal S, Shah S. What's hard about Boolean functional synthesis? arXiv Preprint arXiv:1804.05507; 2018.
Available:https://arxiv.org/pdf/1804.05507.pdf
39. Rushdi AM. Improved variable-entered Karnaugh map procedures, Computers and Electrical Engineering. 1987;13(1):41-52.
40. Rushdi AM, Al-Yahya HA. A boolean minimization procedure using the variable-entered Karnaugh map and the generalized consensus concept. International Journal of Electronics. 2000; 87(7):769-794.
41. Rushdi AM, Al-Yahya HA. Further improved variable-entered Karnaugh map procedures for obtaining the irredundant forms of an incompletely-specified switching function. Journal of King

Abdulaziz University: Engineering Sciences. 2001;13(1):111-152.

42. Rushdi AM. Prime-implicant extraction with the aid of the variable-entered Karnaugh map. Umm Al-Qura University Journal Science, Medicine and Engineering. 2001;13(1):53-74.
43. Rushdi AMA. Utilization of Karnaugh maps in multi-value qualitative comparative analysis. International Journal of Mathematical, Engineering and Management Sciences. 2018;3(1):28-46. Available:www.ijmems.in/ijmems—volumes.html
44. Rushdi RA, Rushdi AM. Karnaugh-map utility in medical studies: The case of fetal malnutrition. International Journal of Mathematical, Engineering and Management Sciences (IJMEMS). 2018;3(3):220-244. Available:www.ijmems.in/ijmems—volumes.html

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history/24676