Umm Al-Qura Univ. J. Sci. Med. Eng. Vol 15, No.2, pp. 59-79 (2003)
23-03- 029 © The University Press
www.ugu.edu.sa

Low-Cost Design of Multiple-Output Switching Circuits Using Map Solutions of Boolean Equations

ALI M. RUSHDI[1] AND OMAR M. BA-RUKAB[2]

[1]Department of Electrical and Computer Engineering, King Abdulaziz University,
P. O. Box: 80204, Jeddah 21589, Saudi Arabia
email: arushdi@kaau.edu.sa

[2]College of Telecomm. and Electronics , P. O. Box 2816, Jeddah 21461, Saudi Arabia
email: obarukab@hotmail.com

التصميم الأقل تكلفة لدوائر تبديلية عديدة المخارج باستخدام الحلول الخريطية للمعادلات البولانية

تقدم ورقة البحث هذه إجراءً مبتكراً لمعالجة مسألة تصميم دائرة تبديلية متعددة المخارج منخفضة التكلفة. يبدأ الإجراء بصياغة هذه المسألة كمسألة إيجاد حل لنظام من المعادلات البولانية، حيث يُراعى أن يكون الحل المطلوب فيها حلاً خاصاً يُستمد من حل احتوائي عام بحيث يتمتع بصيغة أصغرية. تُصاغ حلول صريحة لمخارج الدائرة سواءً أكانت الدوال الابتدائية كاملة التحديد أم لا. ويتضح أن شرط التواؤم المطلوب لحل المعادلات البولانية يتحقق في صورة متطابقة. يتم إعطاء مثالين مفصلين لتوضيح الإجراء المقترح ولبيان كفايته وفعاليته في إنشاء دائرة تبديلية أصغرية متعددة المخارج. إن الإجراء المقترح صالح للتطبيق في حالة كون الجبر البولاني المستخدم عاماً وليس ثنائي القيمة بالضرورة، ومن ثم فإن من الميسور تطويع هذا الإجراء للاستخدام مع المنطق متعدد القيم.

This paper presents a novel procedure for tackling the problem of designing a low-cost multiple-output switching circuit. The procedure starts by formulating this problem as that of solving a system of Boolean equations. The solution sought herein is a particular solution, derived from a general subsumptive solution, in such a way that it possesses a minimal form. Explicit expressions for the circuit outputs are obtained whether the initial functions are completely specified or not. The consistency condition needed in the Boolean-equation solution is shown to be identically satisfied. Two detailed examples are given to illustrate the proposed procedure and demonstrate its efficiency and power in the construction of a minimal multiple-output switching circuit. The proposed procedure is applicable when the underlying Boolean algebra is general and not necessarily 2-valued, and hence it is readily adaptable for use with multiple-valued logic.

Key words: Recursive multiple-output minimization, low cost, Boolean equations, general and particular solutions, consistency condition.

INTRODUCTION

The design of a compact combinational switching network with multiple outputs is an important problem that frequently arises in logic design practice (Bartee 1961; Muroga 1979; Sarje 1986; Chen & Twu 1995; Unger 1997). The first step in the design process is that of specification, which is the statement of a relation between the inputs $\boldsymbol{X}$ and outputs z. Such a relation can be presented in a variety of forms including truth tables, Karnaugh maps, decimal notation or Boolean equations. The specification is solved by finding a vector switching function $\boldsymbol{F} = (F_1, F_2, \ldots, F_m)$ such that the system $z = \boldsymbol{F}(\boldsymbol{X}, z)$ implies the specification. Note that the output values z may depend not only on the values of the inputs $\boldsymbol{X}$ but also on those of the outputs themselves. This recursive definition of z must be chosen in a way to avoid feedback, eliminate possible oscillation and ensure stability. The total cost of the formula expressing $\boldsymbol{F}$ should be as small as possible. Brown (1990) proposed a procedure that produces a least-cost recursive solution of a consistent tabular specification in the normal form $\overline{g}(z, X) = 1$, or equivalently, $g(z, X) = 0$. This procedure formulates the problem of multiple-output combinational circuit design within the framework of Boolean reasoning or Boolean-equation-solving techniques. It leads to the construction of a network in which feedback and redundant variables are excluded and for which the gate-input count is used as a cost measure. The procedure is implemented through the utilization of an automatic Boolean-equation solver.

The present paper is a further improvement of the aforementioned procedure of Brown (1990) that relies on an initial application of the Boolean equation-solving technique of Rushdi (2001). The key contribution of the present paper is to note that when the problem of multiple-output logic minimization is formulated as one of solving a system of Boolean equations, the solution is not a general solution, but is a particular solution that satisfies certain minimality requirements. Therefore, we deliberately avoid the construction of a general solution, thereby eliminating the need for a subsequent effort to locate a least-cost particular solution. Instead, a rigorous mathematical development of the minimal particular solution is directly presented herein, leading to explicit or closed-form expressions of the various outputs **z**. One of the outputs is shown to be expressed by its individual or separate specification. Furthermore, the consistency condition needed for solving a system of Boolean equations is shown to be identically satisfied. The paper addresses also the case when the initial specification is incomplete. Two significant examples are worked out in detail to illustrate the present procedure, and to demonstrate its efficiency and power in the construction of a minimal multiple-output switching circuit.

The procedure is applicable when the underlying Boolean algebra is a general one, and is not restricted to the case of the 2-valued or switching algebra which has been used in the examples for illustrative purposes only. This means that the procedure can be readily adapted to the design of multiple-valued logic (MVL) networks (Triumalai & Butler 1991) by employing techniques for converting the *partial order* in a Boolean algebra to the *total order* of MVL (Rudeanu 1974; Woods & Casinovi 2001).

SUBSUMPTIVE GENERAL SOLUTION

Given a system of *m* equations specifying (possibly incompletely) the required *m* outputs.

$$z_k = F_k(\boldsymbol{X}), \qquad 1 \leq k \leq m, \tag{1}$$

transform this system of m equations into the following equivalent system consisting of a single Boolean equation:

$$g(z, \boldsymbol{X}) = \bigvee_{j=1}^{m} (z_j \oplus F_j(\boldsymbol{X})) = 0, \tag{2}$$

$$\boldsymbol{X} = [X_1 \; X_2 \; \cdots \; X_n]^T, \tag{3a}$$

$$\boldsymbol{z} = [z_1 \; z_2 \; \cdots \; z_m]^T. \tag{3b}$$

Note that $g = B_2^{m+n} \to B_2$ where $B_2 = \{0,1\}$ can be viewed as a function of $\boldsymbol{z}$ only of the form $f(z) = g(z,X)$, where $f = B^m \to B$, and B is a “big” Boolean carrier consisting of the 2^n functions of the n variables $\boldsymbol{X}$, which is partially ordered in accordance with the inclusion operator ($\leq$) (Brown 1990). Now, construct a subsumptive general solution of the Boolean equation

$$g(\boldsymbol{z}, \boldsymbol{X}) \equiv f(\boldsymbol{z}) = 0, \tag{4}$$

Such a solution is obtained via a well-known classical technique, which is formally proved in Rudeanu (1974), and whose details are available in many references (see, e.g., Hammer and Rudeanu (1968); Brown (1990); Rushdi (2001)). A summary of this technique now follows. First, we construct the eliminants

$$f_m(z_1, z_2, \ldots, z_m), \ldots, f_k(z_1, z_2, \ldots, z_{k-1}, z_k), \ldots,$$
$$f_2(z_1, z_2), f_1(z_1), f_0, \tag{5}$$

by setting $f_m = f$ and using the recursion

$$f_{k-1}(z_1, z_2, \ldots, z_{k-1}) = (f_k / \bar{z}_k) \wedge (f_k / z_k),$$
$$k = m, m-1, \ldots, 1. \tag{6}$$

Note that f_{k-1} is the conjunctive eliminant of f_k with respect to the singelton $\{z_k\}$(Brown 1990). This means that f_{k-1} is a conjunction of the two ratios or restrictions

$$f_k / \bar{z}_k = f_k(z_1, z_2, \ldots, z_{k-1}, 0), \tag{7a}$$

$$f_k / z_k = f_k(z_1, z_2, \ldots, z_{k-1}, 1), \tag{7b}$$

obtained from f_k by setting or restricting z_k to 0 and to 1, respectively. For short, these two ratios will be denoted by $f_k(0)$ and $f_k(1)$, respectively.

The classical method for producing a subsumptive general solution proceeds by successive elimination of variables, a technique transforming the problem (4) of solving a single equation of m variables to that of solving m equations of one variable each. The solution requires a separate consistency condition.

$$f_0 = 0, \tag{8a}$$

plus expressing each of the pertinent variables as an interval of functions of the preceding variables, namely:

$$s_k(z_1, z_2, \ldots, z_{k-1}) \le z_k \le t_k(z_1, z_2, \ldots, z_{k-1})$$
$$k = 1, 2, \ldots, m. \tag{8b}$$

where the s_k and t_k functions can be expressed as completely specified Boolean functions. However, to leave room for further simplifications, they are expressed as incompletely specified Boolean functions in the interval form (Brown 1990)

$$f_k(0)\overline{f}_k(1) \leq s_k \leq f_k(0), \tag{9a}$$

$$\overline{f}_k(1) \leq t_k \leq f_k(0) \vee \overline{f}_k(1). \tag{9b}$$

The form of the general solution above allows all the particular solutions of (4), and nothing else, to be generated as a tree.

MINIMAL PARTICULAR SOLUTION

The solution expressed by (8) and (9) is a complete subsumptive solution. To obtain such a solution in a compact form, Rushdi (2001) proposed that the representation of the incompletely-specified Boolean functions s_k and t_k be switched from the interval form of (9) to the don't-care form

$$s_k = f_k(0)\overline{f}_k(1) \vee d(f_k(0)), \tag{10a}$$

$$t_k = \overline{f}_k(1) \vee d(f_k(0)). \tag{10b}$$

Since we are now not interested in the general solution as a whole, but are rather interested in a particular solution that satisfies certain minimality requirements, we switch the interval-form expression of z_k in (8b) into the don't-care form

$$z_k = s_k \vee d(t_k), \tag{11}$$

which, when substituted in by equations (10), reduces to

$$z_k = f_k(0)\overline{f}_k(1) \vee d(f_k(0) \vee \overline{f}_k(1)). \tag{12}$$

Note that while the asserted part of z_k in (12) is a completely-specified switching function, the don't-care part is not unique (Rushdi 2001). Out of the many possible choices of this don't-care part, we opted for the "largest" possible don't-care part, namely, a don't-care part that represents the complete sum of the asserted part ORed with any acceptable form of the don't-care part. This choice is very useful for our minimization objectives (Rushdi & AL-Yahya 2000). As a bonus, it yields the simplest possible expression for z_k. Now we state three new theorems which facilitate the construction of solutions for z_k, $k = m, m-1, \ldots, 1$. Proofs of these theorems are given in the Appendix.

THEOREM 1

$$f_k = \overset{k}{\underset{j=1}{\vee}} (z_j \oplus F_j(\mathbf{X})), \quad k = m, m-1, \ldots, 1, 0. \tag{13}$$

THEOREM 2

$$\overline{f}_k(1) \leq f_k(0), \qquad k = m, m-1, \ldots, 1. \tag{14}$$

THEOREM 3

$$z_k = \overline{f}_k(1) \vee d(f_k(0)), \qquad k = m, m-1, \ldots, 1. \tag{15}$$

In passing, we note that according to Theorem 1

$$f_1 = z_1 \oplus F_1(\mathbf{X}), \tag{16}$$

so that

$$f_1(0) = \overline{f}_1(1) = F_1(\mathbf{X}), \tag{17}$$

and according to Theorem 3, z_k is minimally expressed by

$$z_1 = \overline{f}_1(1) \vee d(f_1(0)) = F_1(\mathbf{X}), \tag{18}$$

which means that the present solution for z_1 is exactly its solution obtained when it is treated separately from the other outputs.

Furthermore, we observe that the consistency condition

$$f_0 = f_1(0) f_1(1) = F_1(\mathbf{X}) \overline{F_1}(\mathbf{X}) = 0, \tag{19}$$

is identically satisfied, and there should be no worry about it. The results in (18) and (19) are satisfied by all the computer-generated solutions obtained by Brown (1990). However, neither of these results has been noted before, because the earlier work of

Brown (1990) opted for an immediate use of an automatic Boolean-equation solver, thereby losing the insight that can be provided by some far-reaching preliminary analysis.

THE CASE OF INCOMPLETELY SPECIFIED FUNCTIONS

If the functions $F_k(X)$ in (1) are incompletely specified, they may be written in the form

$$F_k(\mathbf{X}) = G_k(\mathbf{X}) \vee d(H_k(\mathbf{X})), \tag{20}$$

where $G_k(X)$ and $H_k(\mathbf{X})$ are the asserted and don't care parts of $F_k(\mathbf{X})$, respectively (Rushdi 2001).

The complement of $F_k(\mathbf{X})$ is given by

$$\overline{F}_k(X) = \overline{G}_k(X)\overline{H}_k(X) \vee d(\overline{G}k(X)). \tag{21}$$

The solution z_k in (15) can now be written as

$$z_k = \text{Asserted part of } \overline{f_k}(1) \vee d(f_k(0)), \tag{22}$$

where the don't-care part of $\overline{f_k}(1)$ is absorbed in $d(f_k(0))$, by virtue of (14). The functions $\overline{f_k}(1)$ and $f_k(0)$ are obtained as an offshoot of the proof of Theorem 1 in the Appendix as:

$$\overline{f}k(1) = F_k(X) \wedge \bigwedge_{j=1}^{k-1}(z_j F_j(X) \vee \overline{z}j\overline{F}j(X)), \tag{23}$$

$$f_k(0) = F_k(X) \vee \bigvee_{j=1}^{k-1}(\overline{z}j\, F_j(X) \vee z_j \overline{F}j(X)). \tag{24}$$

Direct substitution of (23) and (24) into (22) produces the following expression for the solutions z_k

$$z_k = G_k(\boldsymbol{X}) \wedge \bigwedge_{j=1}^{(k-1)} (z_j G_j(\boldsymbol{X}) \vee \bar{z}_j \overline{G}_j(\boldsymbol{X}) \overline{H}_j(\boldsymbol{X})) \vee$$

$$d(G_k(\boldsymbol{X}) \vee H_k(\boldsymbol{X}) \vee \bigvee_{j=1}^{(k-1)} (\bar{z}_j (G_j(\boldsymbol{X}) \vee H_j(\boldsymbol{X})) \vee z_j \overline{G}_j(\boldsymbol{X}))). \quad (25)$$

VEKM IMPLEMENTATION

Fig. 1 shows a VEKM solution of the Boolean equations stated by (1) – (4) in accordance to the procedure of Rushdi (2001). This amounts to a statement of the solutions (15) for the completely specified case. In comparison, Fig. 2 demonstrates the solutions (25) for the incompletely specified case. Both Figs. 1 & 2 are given for m = 3, and their extensions to higher values of m are obvious.

EXAMPLE 1

We utilize the present method to solve the multiple-output logic minimization problem posed in example 9.5.1 (pp. 238 – 239) of Brown (1990). Here, we have a system of three completely specified equations

$$z_1 = F_1 = X_1 \vee \overline{X_2}\,\overline{X_3} \vee X_2 X_3, \quad (26a)$$

$$z_2 = F_2 = \overline{X_1} X_2 \vee \overline{X_1} X_3, \quad (26b)$$

$$z_3 = F_3 = \overline{X_1} X_2 X_3. \quad (26c)$$

We need to substitute the above values of F_1, F_2, and F_3 in Fig. 1, so as to obtain VEKM representations for z_2 and z_3. To this end, we perform the following preliminary computations.

$$\overline{F_1} = \overline{X_1}(X_2 \overline{X_3} \vee \overline{X_2} X_3),$$

$$\overline{F_2} = X_1 \vee \overline{X_2}\,\overline{X_3},$$

$$\overline{F_3} = X_1 \vee \overline{X_2} \vee \overline{X_3},$$

$$\overline{F_1}F_2 = \overline{X_1}(X_2\overline{X_3} \vee \overline{X_2}\,X_3),$$

$$F_1F_2 = \overline{X_1}X_2X_3,$$

$$F_1 \vee F_2 = 1,$$

$$\overline{F_1} \vee F_2 = \overline{X_1}X_2 \vee \overline{X_1}X_3,$$

$$\overline{F_1}\,\overline{F_2}\,F_3 = 0\,,$$

$$\overline{F_1}\,F_2F_3 = 0\,,$$

$$F_1\,\overline{F_2}\,F_3 = 0,$$

$$F_1F_2F_3 = \overline{X_1}X_2X_3\,,$$

$$F_1 \vee F_2 \vee F_3 = 1\,,$$

$$F_1 \vee \overline{F}2 \vee F_3 \;=\; X_1 \vee \overline{X}2\overline{X}3 \vee\; X_2X_3\,,$$

$$\overline{F_1} \vee F_2 \vee F_3 = \overline{X_1}X_2 \vee \overline{X_1}X_3,$$

$$\overline{F_1} \vee \overline{F_2} \vee F_3 = 1\,.$$

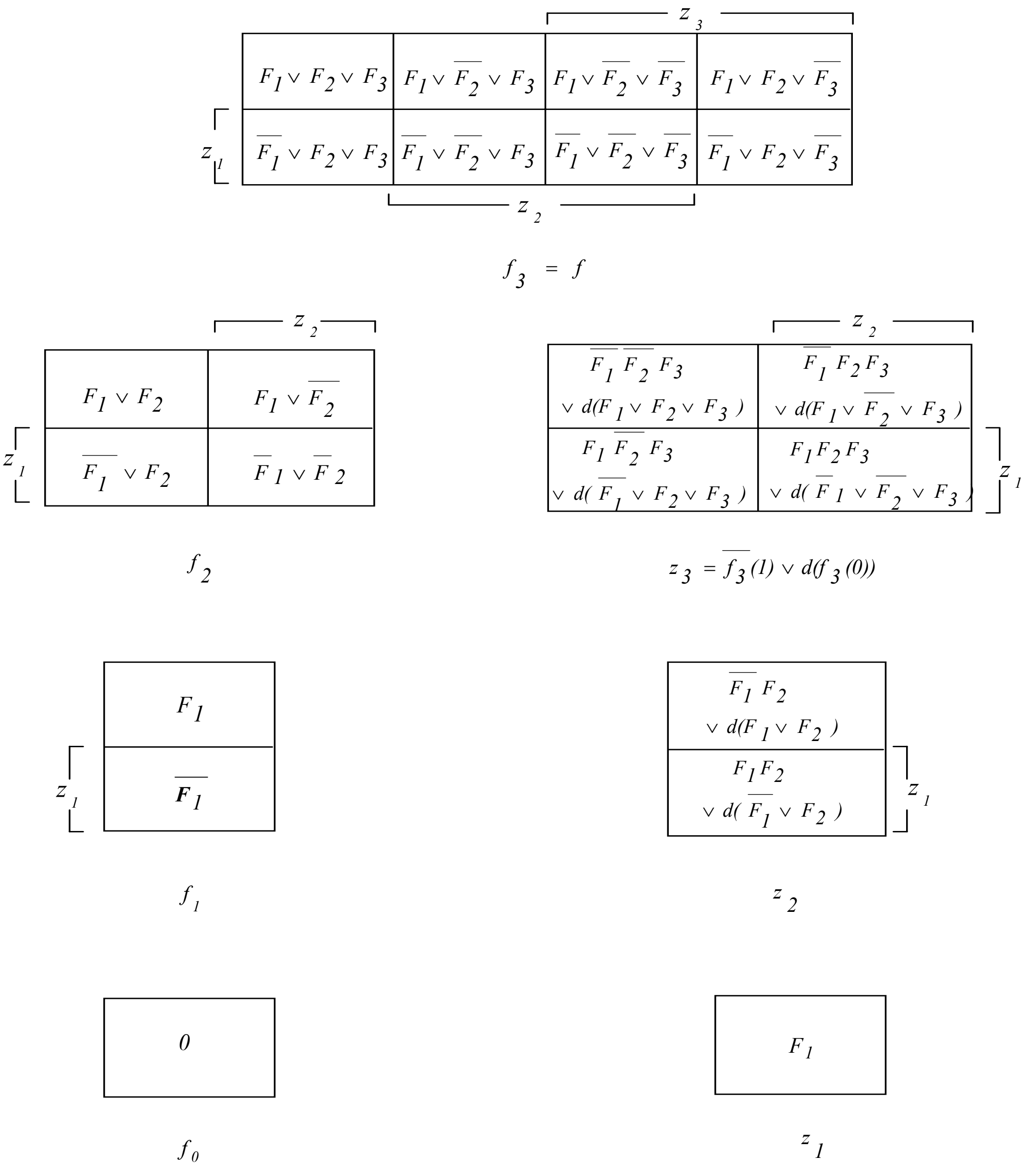
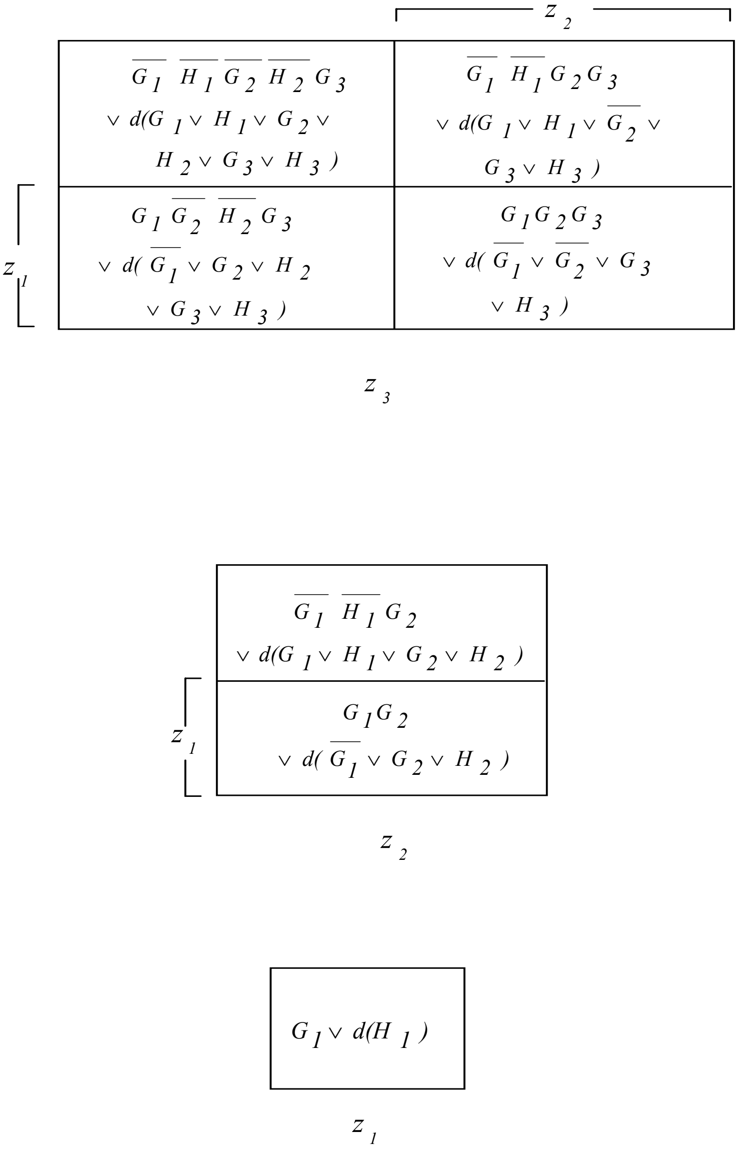


Fig. 1. VEKM solution of the multiple-output minimization problem for m = 3.



		z_2
	$\overline{G_1}\,\overline{H_1}\,\overline{G_2}\,\overline{H_2}\,G_3 \vee d(G_1 \vee H_1 \vee G_2 \vee H_2 \vee G_3 \vee H_3)$	$\overline{G_1}\,\overline{H_1}\,G_2 G_3 \vee d(G_1 \vee H_1 \vee \overline{G_2} \vee G_3 \vee H_3)$
z_1	$G_1 \overline{G_2}\,\overline{H_2}\,G_3 \vee d(\overline{G_1} \vee G_2 \vee H_2 \vee G_3 \vee H_3)$	$G_1 G_2 G_3 \vee d(\overline{G_1} \vee \overline{G_2} \vee G_3 \vee H_3)$

z_3

	$\overline{G_1}\,\overline{H_1}\,G_2 \vee d(G_1 \vee H_1 \vee G_2 \vee H_2)$
z_1	$G_1 G_2 \vee d(\overline{G_1} \vee G_2 \vee H_2)$

z_2

$G_1 \vee d(H_1)$

z_1

Fig. 2. A demonstration of the solution (25) for the incompletely specified case.

We now obtain the VEKM representations for z_2 and z_3 shown in Fig. 3. Employing the VEKM minimizaiton procedure of (Rushdi & Al-Yahya 2001), we obtain the following expressions for z_2 and z_3

$$z_2 = \overline{z_1} \vee \overline{X_1} X_2 , \tag{27a}$$

$$z_3 = z_1 z_2 , \tag{27b}$$

Note that we did not bother to find an expression for z_1 since it is simply given by its initial specification (26a). Neither did we check the consistency condition since it is identically satisfied in the present case. In passing, we note that the search for a solution with a minimal input count requires the repeated solution of the problem with various permutations of the variables (z_1, z_2, z_3). For example, if these variables are rearranged as (z_3, z_2, z_1), then the following solution is obtained

$$z_3 = \overline{X_1} X_2 X_3 , \tag{28a}$$

$$z_2 = \overline{X_1} X_2 \vee \overline{X_1} X_3 , \tag{28b}$$

$$z_1 = \overline{z_2} \vee z_3 , \tag{28c}$$

which has a better input count than the solution in (27).

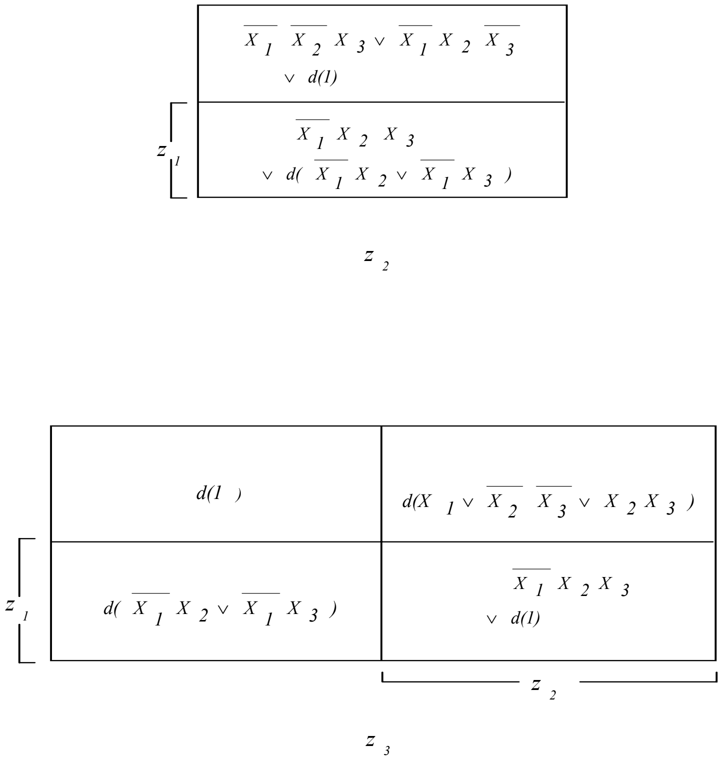


Fig. 3. VEKM representations for z_2 and z_3 in example 1.

EXAMPLE 2

Now, we apply the present method to the problem given in example 9.6.4 (pp. 235-237) of Brown (1990), and which was solved therein via a computer code based on a sophisticated algorithm. A tabular specification is given for three output functions, which can be rephrased here in the decimal notation

$$Z_1(A,B,C,D) = G_1 \vee d(H_1) = \sum(0,3,4) \vee d(7,9,14,15), \qquad (29a)$$

$$Z_2(A,B,C,D) = G_2 \vee d(H_2) = \sum(0,3,5,6,13) \vee d(7,9,10,12,14,15), \qquad (29b)$$

$$Z_3(A,B,C,D) = G_3 \vee d(H_3) = \sum(1,2,4,8,9,10,13) \vee d(3,7,11,12,14,15). \cdot \qquad (29c)$$

Direct substitutions of the decimal notation expressions for G_1, H_1, G_2, H_2, G_3 and H_3 given in (29), in the general VEKM representations for z_2 and z_3 given in Fig. 2, produces their specific VEKM representations for this example shown in Fig. 4, wherein the VEKM entries are given in decimal notation. Their representations easily translate into the ones shown in Figs. 5 and 6, in which the VEKM entries are now given in conventional Karnaugh-map forms and algebraic forms, respectively. Minimal expressions for z_2 and z_3 turn out to be

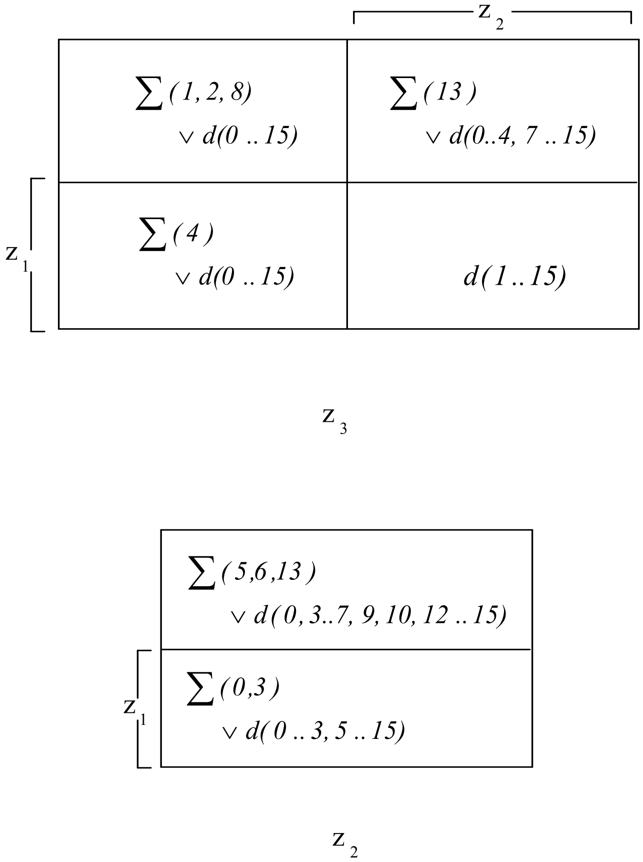
$$z_2 = \overline{B}z_1 \vee B\overline{z_1}, \tag{30a}$$

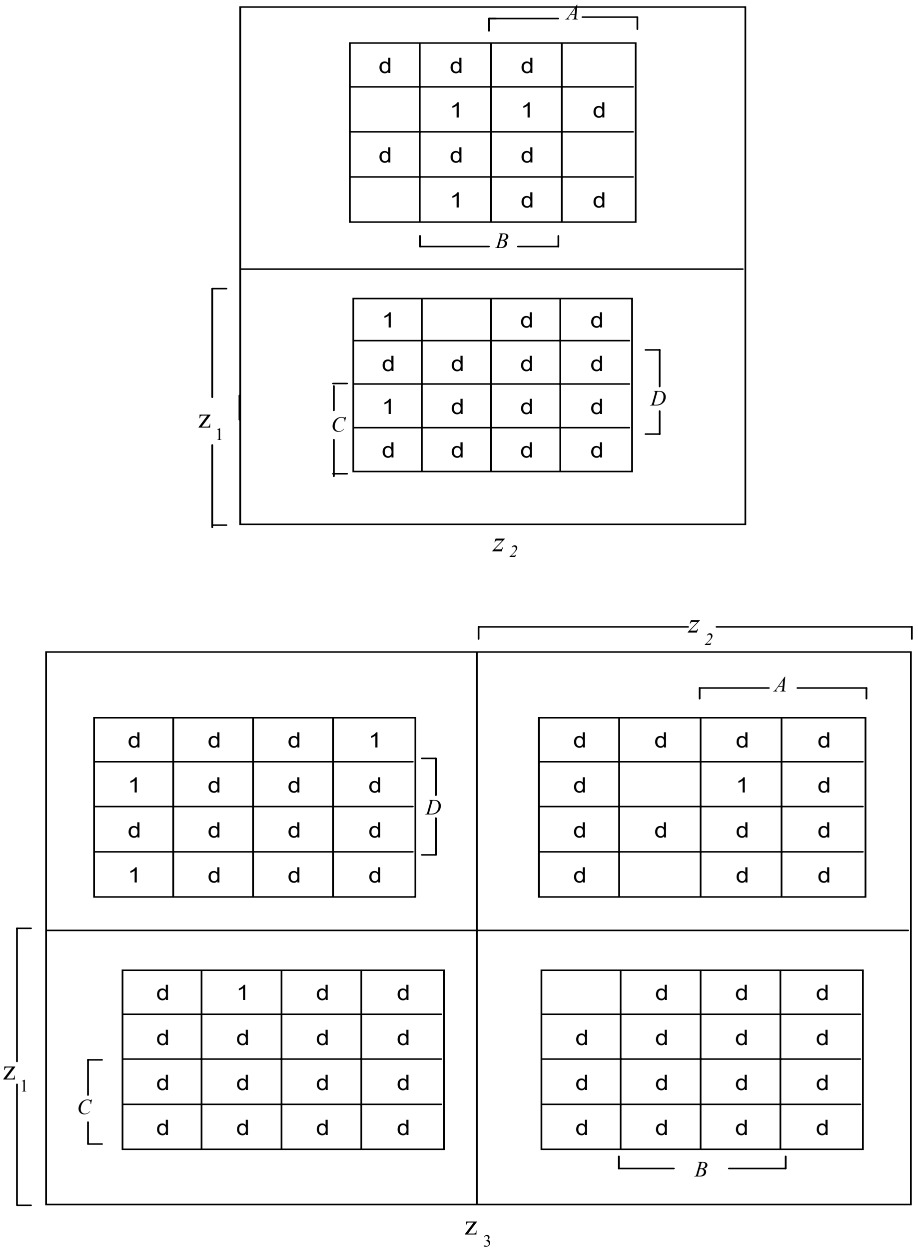
$$z_3 = \overline{z_2} \vee A, \tag{30b}$$

and can be obtained either through the application of conventional Karnaugh-map minimizaiton (Muroga 1979) to the maps in Fig. 5, or via the application of VEKM minimization (Rushdi & Al-Yahya 2000; Rushdi & Al-Yahya 2001) to the maps in Fig. 6. The minimal expression for z_1 (separately obtained by casting its decimal notation in (27a) into the map of Fig. 7) is

$$z_1 = \overline{A}\,\overline{C}\,\overline{D} \vee \overline{A}\,C\,D. \tag{30c}$$

Fig. 4. VEKM representations for z_2 and z_3 in example 2, with entries expressed in decimal notation.





d	d	d	
	1	1	d
d	d	d	
	1	d	d

1		d	d
d	d	d	d
1	d	d	d
d	d	d	d

d	d	d	1
1	d	d	d
d	d	d	d
1	d	d	d

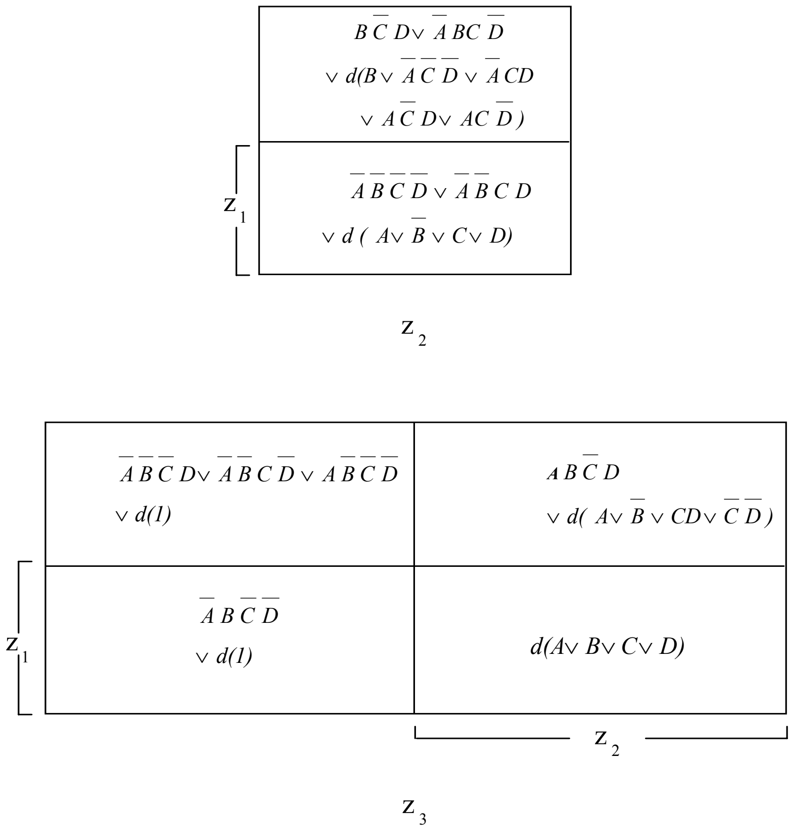
d	d	d	d
d		1	d
d	d	d	d
d		d	d

d	1	d	d
d	d	d	d
d	d	d	d
d	d	d	d

	d	d	d
d	d	d	d
d	d	d	d
d	d	d	d

Fig. 5. VEKM representations for Z_2 and Z_3 in example 2, with entries in conventional Karnaugh-map

Fig. 6. VEKM representations for Z_2 and Z_3 in example 2, with entries expressed algebraically.



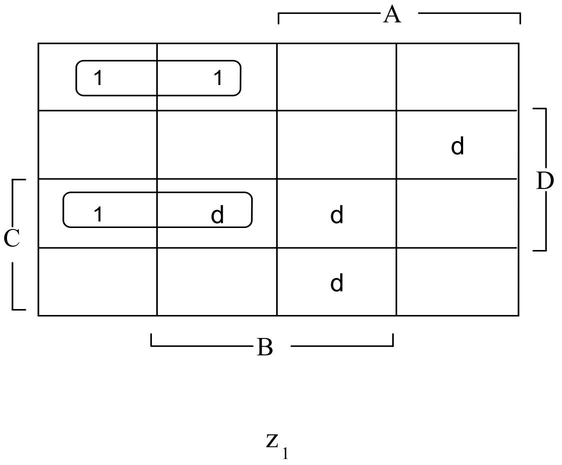


Fig. 7. Individual Karnaugh-map representation for z_1.

CONCLUSIONS

Boolean-equation solving is a very powerful tool in switching theory. It is very useful both as an analysis or simulation tool (Wood & Casinovi 2001) and also as a synthesis or design tool (Brown 1990). The present paper has added a new dimension to the employment of Boolean-equation solving in the design of multiple-output switching circuit. The key contribution of the paper is to avoid the brute-force application of a Boolean-equation solver to produce a general solution that is equivalent to a multitude of particular solutions, with the subsequent need to search among these for the one solution satisfying minimality requirements. Instead, the paper proposes some preliminary work in which a minimal particular solution is targeted from the outset. The outcome of this work is gratifying, since explicit or closed-form solutions for the desired outputs are obtained. As a bonus, it is shown that one of the output is expressible via its individual or separate specification, and it is proved that the consistency condition required in Boolean-equation solving is automatically satisfied.

The procedure presented herein is of definite pedagogical and pictorial advantages for manual use with medium-sized problems. We have not computer implemented this procedure yet. Such an implementation is desirable to allow the use of the procedure with large problems and to assess its computational complexity in comparison with existing algorithms.

APPENDIX
PROOF OF THEOREM 1

We use a downward version of mathematical induction. We label (13) as T_k and note that we can prove it for $k = m, m-1, \ldots, 1, 0$ by proving:

(a) the limiting case T_m, and

(b) the inductive case $T_k \rightarrow T_{k-1}$

For (a) we note that T_m is true since by definition (equations (2) and (4))

$$f_m = f = \bigvee_{j=1}^{m} (z_j \oplus F_j(\boldsymbol{X})), \tag{A1}$$

For (b) assume T_k is true, i.e., assume f_k is given by (13) and obtain

$$f_{k-1} = f_k(0) f_k(1), \tag{A2}$$

where

$$f_k(0) = (f_k)_{z_k=0} = A \vee B \tag{A3}$$

$$f_k(1) = (f_k)_{z_k=1} = \overline{A} \vee B \tag{A4}$$

where

$$A = F_k(\boldsymbol{X}) \tag{A5}$$

$$B = \bigvee_{j=1}^{(k-1)} (z_j \oplus F_j(\boldsymbol{X})) \tag{A6}$$

so that f_{k-1} finally becomes

$$f_{k-1} = (A \vee B)(\overline{A} \vee B) = B = \bigvee_{j=1}^{(k-1)} (z_j \oplus F_j(\boldsymbol{X})), \tag{A7}$$

which is T_{k-1}.

PROOF OF THEOREM 2

From (A3) and (A4), the complements of $f_k(0)$ and $f_k(1)$ are

$$\overline{f}_k(0) = (\overline{A \vee B}) = \overline{A}\,\overline{B}, \tag{A8}$$

$$\overline{f}_k(1) = (\overline{\overline{A} \vee B}) = A\,\overline{B}, \tag{A9}$$

which, when multiplied together produce

$$\overline{f}_k(1)\overline{f}_k(0) = (\overline{A}\,\overline{B})(A\,\overline{B}) = 0. \tag{A10}$$

Equation (A10) is equivalent to (14) by virtue of the very definition of the $\leq$ operator, viz. $(\{C \leq D\} \Leftrightarrow \{C\,\overline{D} = 0\})$. (Brown 1990; Rushdi 2001).

PROOF OF THEOREM 3

Other versions of defining the ($\leq$) operator in (14) are

$$f_k(0)\overline{f}_k(1) = \overline{f}_k(1), \tag{A11}$$

$$f_k(0) \vee \overline{f}_k(1) = f_k(0), \tag{A12}$$

by virtue of which the expression (12) for z_k reduces to (15).

REFERENCES

Bartee, T. C. 1961. Computer design of multiple output logical networks. *IEEE* Transactions on Electronic Computers EC-10 (1): 21-30.

Brown, F. M. 1990. Boolean reasoning: The logic of Boolean equations. Kluwer Academic Publishers, Boston, 273 pp.

Chen, L. & Twu, H. 1995. Synthesis of multilevel NAND gate circuits for incompletely specified multi-output Boolean functions and CAD using permissible cubes and PCRM graphs. International Journal of Electronics 78 (2): 303-316.

Hammer, P. L. & Rudeanu, S. 1968. Boolean methods in operations research and related areas. Springer verlag, Berlin, Germany, 329 pp.

Muroga, S. 1979. Logic design and switching theory. Wiely, New York, 617 pp.

Rudeanu, S. 1974. Boolean functions and equations. North-Holland Publ. Co. & American Elsevier, Amsterdam, The Netherlands, 442 pp.

Rushdi, A. M. & Al-Yahya, H. A. 2000. A Boolean minimization procedure using the variable-entered Karnaugh map and the generalized consensus concept. International Journal of Electronics 87 (7): 769-794.

Rushdi, A. M. & Al-Yahya, H. A. 2001. Further improved variable-entered Karnaugh map procedures for obtaining the irredundant forms of an incompletely-specified switching function. Journal of King Abdulaziz University: Engineering Science 13 (1): 111 – 152.

Rushdi, A. M. 2001. Using variable-entered Karnaugh maps to solve Boolean equations. International Journal of Computer Mathematics 78 (1): 23-38.

Sarje, A. K. 1986. Minimizaiton of multiple-output switching circuits. International Journal of Electronics 60 (5): 617-623.

Triumalai, P. P, & Butler, J. T. 1991. Minimization algorithms for multiple-valued programmable logic arrays. *IEEE* Transactions on Computers 40 (2): 167- 177.

Unger, S. H. 1997. The essence of logic circuits, (2nd edn). *IEEE* Press, New York, 340 pp.

Woods, **S. & Casinovi**, **G. 2001.** Multiple-level logic simulation algorithm. *IEE* Proceeding on Computer and Digital Techniques 148 (3): 129-137.

Received 29/3/1423; 10/6/2002, accepted 18/11/1423; 21/1/2003