International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 392–403, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-028

Handling Generalized Type-2 Problems of Digital Circuit Design via the Variable-Entered Karnaugh Map

Ali Muhammad Ali Rushdi
Department of Electrical and Computer Engineering, Faculty of Engineering
King Abdulaziz University, P. O. Box 80204, Jeddah 21589, Saudi Arabia
E-mail: arushdi@kau.edu.sa

(Received July 13, 2017; Accepted September 28, 2017)

Abstract
This paper offers a novel treatment of the generalized Type-2 problem, a prominent fundamental problem of digital circuit design. We adapt the input-domain constraining technique *via* utilization of Variable-Entered Karnaugh Maps (VEKMs) together with careful employment of modern don't-care notation. Our analysis covers the cases when an honest translator is possible or a sneaky translator is warranted, and is effective whether side inputs are absent or present, and for scalar or vectorial outputs.

Keywords- Digital circuit design, Generalized type-2 problems, Honest and sneaky translators, 'Big' Boolean algebras.

1. Introduction

Certain classical problems of digital circuit design can be described by the general layout presented in Fig. 1. This layout was first suggested, without the shady sneaky path, by Ledley (1959, 1960), and later used by Bell (1968), and Brown (1975). Ledley's layout involves a combinational logic circuit C that is comprised of two subnetworks A and B. The final output can be viewed as output $\boldsymbol{s}(\boldsymbol{Y},\boldsymbol{Z})$ of network B or output $\boldsymbol{t}(\boldsymbol{X},\boldsymbol{Y})$ of network C. Here, the five quantities $\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{s}$ and $\boldsymbol{t}$ are binary or Boolean vectors of lengths k, m, n, l, and l representing basic inputs, side inputs, intermediary values, output for subnetwork B, and output for the overall network C, respectively. This means that the quantities $\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{s}$ and $\boldsymbol{t}$ belong to $B_2^k, B_2^m, B_2^n, B_2^l$, and B_2^l, respectively, where B_2 stands for the two-valued Boolean algebra B_2, also known as the switching algebra or propositional algebra. Consequently, three 'elementary problems of digital circuit design' emerge (Rushdi & Ahmad, 2018), namely

Type-1 problem: Given $\boldsymbol{Z}(\boldsymbol{X})$ and $\boldsymbol{s}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$, find $\boldsymbol{t}(\boldsymbol{X},\boldsymbol{Y})$.
Type-2 problem: Given $\boldsymbol{Z}(\boldsymbol{X})$ and $\boldsymbol{t}(\boldsymbol{X},\boldsymbol{Y})$ find $\boldsymbol{s}(\boldsymbol{Y},\boldsymbol{Z})$.
Type-3 problem: Given $\boldsymbol{s}(\boldsymbol{Y},\boldsymbol{Z})$ as well as $\boldsymbol{t}(\boldsymbol{X},\boldsymbol{Y})$, find $\boldsymbol{Z}(\boldsymbol{X})$.

Brown (1975) slightly altered the structure of Ledley's network by adding the shady sneaky path in Fig. 1, and, as a result, replacing $\boldsymbol{s}(\boldsymbol{Y},\boldsymbol{Z})$ by $\boldsymbol{s}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$. Network B is called an 'honest translator' or a 'sneaky translator' depending on whether the shaded path in Fig. 1 can be omitted or it is needed (or used). Brown (1975) also proposed a solution of the modified type-2 problem for $l = 1$, which can now be restated as: Given $\mathbf{Z}(\boldsymbol{X})$ and $t(\boldsymbol{X},\boldsymbol{Y})$ find $s(\boldsymbol{Y},\boldsymbol{Z})$ if an honest translator is possible, but otherwise find $s(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$ for a sneaky translator. Brown (1975) formulated his solution via an interval-based condition of the form $g_l \leq g \leq g_u$.

This paper proposes a novel solution of the generalized type-2 problem that relies on replacing the interval-based condition by one based on the don't-care notation of the form $g = g_l \vee$

$d(g_u)$ (Reusch, 1975; Rushdi, 1987, 2001, 2018). Such a notation allows useful utilization of the Variable-Entered Karnaugh Map (VEKM) (Rushdi, 1983, 1985, 1987, 2018; Rushdi and Al-Yahya, 2000, 2001, 2002), which acts as the natural map for 'big' Boolean functions (Brown, 1990; Rushdi and Amashah, 2010, 2011). The advantages of our novel method compared to the earlier method of Brown (1975) include conceptual clarity, high speed, aggregation of tasks, and better control on outcomes. Such advantages do not pertain only to pedagogical issues but might also be of significant benefit to contemporary practical digital design (Brown and Vranesic, 2014). Therefore, our contribution herein should be considered a warranted revival of a problem of everlasting interest (albeit being an old one, indeed).

This paper is a part of our continuing effort to modernize techniques for handling fundamental problems of digital circuit design via 'big' Boolean algebraic techniques. Type-1 problems can be solved by direct substitution, but Type-2 or Type-3 problems are challenging, indeed. While we devote this paper to handling generalized Type-2 problems, we dedicate our sister publication (Rushdi and Ahmad, 2018) to Type-3 problems, wherein we rely solely and successfully on the methodology of equation solving in 'big' Boolean algebras. The methodology of equation solving in 'big' Boolean algebras is also applicable to the current generalized Type-2 problems, and it involves solving an inverse problem of logic in which knowledge of the vectorial function $\boldsymbol{Z}(\boldsymbol{X})$ is utilized to produce its inverse vectorial function $\boldsymbol{X}(\boldsymbol{Z})$. We did not opt for using equation solving herein, since it seems to involve certain unwarranted steps as it produces parametric solutions with the parameters disappearing in the final stage of problem solving.

The organization of this paper is as follows. This introductory section is followed by Section 2 which reviews Brown's formulation of the generalized Type-2 problem, and then introduces our novel formulation and method. Section 3 discusses a design example demanding a sneaky translator, while Sec. 4 explores another example allowing an honest translator. Both examples ignore the presence of side inputs $\boldsymbol{Y}$, and are therefore supplemented by a third example in Section 5 in which side inputs $\boldsymbol{Y}$ are present. Section 6 concludes the paper. The paper is made self-contained by the inclusion of a supplementary appendix about expressing interval functions *via* modern don't-care notation.

2. Formulation and Method

The constrained-input problem of Brown (1975) is to find a general solution for s in the relation

$$\{h = 0\} \Rightarrow \{s = t\} \tag{1}$$

Brown (1975) showed that s has the interval solution

$$\bar{h}t \leq s \leq h \vee t \tag{2}$$

Our procedure is to rewrite the double-inequality (2) in the don't-care notation (see Appendix A)

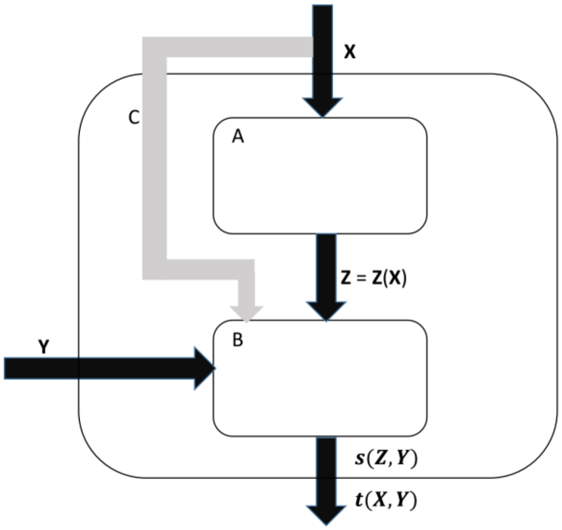
X
C
A
Z = Z(X)
B
Y
$s(\boldsymbol{Z},\boldsymbol{Y})$
$t(\boldsymbol{X},\boldsymbol{Y})$

Fig. 1. Layout of generalized problems of digital circuit design (If the sneaky (dotted) path is needed, the function $\boldsymbol{s}(\boldsymbol{Z}.\boldsymbol{Y})$ is replaced by $\boldsymbol{s}(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y})$)

$$s = \bar{h}t \vee d(h \vee t) \ = \bar{h}t \vee d(h) \ = gt \vee d(\bar{g} \ \vee t) = gt \vee d(\bar{g}) \tag{3}$$

Since the side inputs $\boldsymbol{Y}$ do not play a crucial role in the constrained-input problem, we temporarily assume that they are not present, and then relax this assumption at a later stage of our analysis. The basic information conveyed by the function $\boldsymbol{Z}(\boldsymbol{X})$ is equivalent to a single Boolean equation of the form $h(\boldsymbol{X},\boldsymbol{Z}) = 0$ or $g(\boldsymbol{X},\boldsymbol{Z}) = 1$ where $h(\boldsymbol{X},\boldsymbol{Z}) = \bar{g}(\boldsymbol{X},\boldsymbol{Z})$, and

$$g(\boldsymbol{X},\boldsymbol{Z}) = \ \wedge_{i=1}^{n}(Z_i \odot Z_i(\boldsymbol{X})) = \wedge_{i=1}^{n} Z_i^{Z_i(\boldsymbol{X})} = 1 \tag{4}$$

The factor $(Z_i \odot Z_i(\boldsymbol{X})) = \ Z_i^{Z_i(\boldsymbol{X})}$ in (4) is equal to Z_i if $Z_i(\boldsymbol{X}) = 1$ and is equal to $\bar{Z}_i$ if $Z_i(\boldsymbol{X}) = 0$. The function $g(\boldsymbol{X},\boldsymbol{Z})$ is understood to be $g\colon B_2^{k+n} \rightarrow \ B_2$. Since we need to solve (at least implicitly) for $\boldsymbol{X}$ in terms of $\boldsymbol{Z}$, it is unavoidable to resort to ‘big’ Boolean algebras which are finite (atomic) Boolean algebras larger than the two-valued Boolean algebra (Brown, 1990; Rushdi and Amashah, 2010, 2011). This is done by reinterpreting the aforementioned $g(\boldsymbol{X},\boldsymbol{Z})$ as $g(\boldsymbol{X})$ where $g\colon B_{2^N}^{k} \rightarrow B_{2^N}$, where B_{2^N} is the free Boolean algebra $FB(Z_1, Z_2 \ldots \ldots . Z_n)$, which has $N = \ 2^n$ atoms, and 2^N elements.

The natural map or VEKM for $g(\boldsymbol{X})$ has k map variables and hence 2^k cells. This natural map is equivalent to expressing $g(\boldsymbol{X})$ by its Minterm Canonical Form (MCF) (Brown, 1990)

$$g(\boldsymbol{X}) \ \equiv \ \vee_{\mathbf{A} \in \{0,1\}^k} \ g(\boldsymbol{A}) \, \boldsymbol{X}^{\boldsymbol{A}} \tag{5}$$

For $\boldsymbol{X} = [\, X_1 \, X_2 \ldots \ X_k \,]^T \in \ B_{2^N}^{k}, \ \boldsymbol{A} = [\, a_1 \, a_2 \ldots \ a_k \,]^{\boldsymbol{T}} \ \in \ \{0,1\}^k$, the symbol $\boldsymbol{X}^{\boldsymbol{A}}$ is defined as

$$\boldsymbol{X}^{\boldsymbol{A}} = \ X_1{}^{a_1} \ X_2{}^{a_2} \ldots \ X_k{}^{a_k} \tag{6}$$

where ${X_i}^{a_i}$ takes the value $\bar{X}_i$ (complemented literal) if $a_i = 0$, and takes the value X_i (uncomplemented literal) if $a_i = 1$. For $\mathbf{A} \in \{0, 1\}^k$, the symbol $\boldsymbol{X}^{\boldsymbol{A}}$ spans the minterms of $\boldsymbol{X}$, which are the 2^k elementary or primitive products

$$\overline{X_1}\,\overline{X_2} \ldots \overline{X_{k-1}}\,\overline{X_k}, \quad \overline{X_1}\,\overline{X_2} \ldots \overline{X_{k-1}}\,X_k, \quad . . . , \quad X_1\,X_2 \ldots X_{k-1}\,X_k \tag{7}$$

The constant values $g(\boldsymbol{A})$ in equation (5) are elements of B_{2^N} called the discriminants of $g(\boldsymbol{X})$, and constitute the entries of the natural map of $g(\boldsymbol{X})$ which has an input domain $\{0, 1\}^k \subseteq B_{2^N}^k$. The Boolean algebra $B_{2^N} = FB(Z_1, Z_2 \ldots\ldots Z_n)$ has generators Z_i $(1 \leq i \leq n)$ which look like variables (In fact, they were originally variables before we changed their roles to generators). Therefore, we can accept the name assigned (for historical reasons) to the natural map of $g(\boldsymbol{X})$, namely the name of the Variable-Entered Karnaugh Map (VEKM). We now observe that the minterms of $\boldsymbol{Z}$, which are the $2^n = N$ elementary or primitive products

$$\overline{Z_1}\,\overline{Z_2} \ldots \overline{Z_{n-1}}\,\overline{Z_n}, \quad \overline{Z_1}\,\overline{Z_2} \ldots \overline{Z_{n-1}}\,Z_n, \quad . . . ., \quad Z_1\,Z_2 \ldots Z_{n-1}\,Z_n \tag{8}$$

are exactly the atoms of the underlying Boolean algebra (Rushdi and Amashah, 2011). For convenience, we call these atoms T_i $(0 \leq i \leq (N-1))$. Now, we compare (4) and (5) to obtain

$$g(\boldsymbol{A}) = \wedge_{i=1}^{n}(Z_i \odot Z_i(\boldsymbol{A})) = \wedge_{i=1}^{n} Z_i^{Z_i(\boldsymbol{A})} \tag{9}$$

Equation (9) reveals that every cell entry in the natural map of $g(\boldsymbol{X})$ is a particular atom T_i of the underlying Boolean algebra B_{2^N}. Consequently, every cell entry in the natural map of $\bar{g}(\boldsymbol{X})$ is a disjunction of all atoms except the particular atom T_i appearing in the corresponding cell for $g(\boldsymbol{X})$. The natural map for $s(\boldsymbol{X})$ can be constructed from those of $g(\boldsymbol{X})$, $\bar{g}(\boldsymbol{X})$, and $t(\boldsymbol{X})$ by rewriting (3) as

$$s(\boldsymbol{X}) = g(\boldsymbol{X})t(\boldsymbol{X}) \vee d(\bar{g}(\boldsymbol{X})) \tag{10}$$

where $s, g, \bar{g}: B_{2^N}^k \rightarrow B_{2^N}$, and $t: B_2^k \rightarrow B_2$. For $\mathbf{A} \in \{0, 1\}^k$, $\{t(\boldsymbol{A}) = 0\}$ implies $\{s(\boldsymbol{A}) = d(\bar{g}(\boldsymbol{A}))\}$ and hence $s(\boldsymbol{A})$ is a don't-care disjunction of all atoms except one, *i.e.*, cell $\boldsymbol{A}$ of the map for $s(\boldsymbol{A})$ has a single negatively-asserted atom with each of the rest of atoms added don't-care. Alternatively, $\{t(\boldsymbol{A}) = 1\}$ implies $\{s(\boldsymbol{A}) = g(\boldsymbol{A}) \vee d(1)\}$ and hence $s(\boldsymbol{A})$ is equal to a single atom ORed with a don't-care, *i.e.*, cell $\boldsymbol{A}$ of the map for $s(\boldsymbol{A})$ has a single positively-asserted atom with each of the rest of atoms added don't-care.

The atoms of the underlying Boolean algebra B_{2^N} differ in their modes of appearance in the cells of $s(\boldsymbol{X})$. They can be classified into four sets:

1. **Purely don't-care atoms:** which are never asserted (positively or negatively) but appear don't-care in every map cell,
2. **Positively asserted atoms:** which are positively-asserted in some map cells, and are don't-care in the rest (not negatively-asserted in any cell),
3. **Negatively asserted atoms:** which are negatively-asserted in some map cells, and are don't-care in the rest (not positively-asserted in any cell),

4. **Atoms of mixed assertion:** which are positively-asserted in some map cells**,** negatively-asserted in some others, and are don't-care in the rest.

An honest translator is possible if and only if the fourth set (Atoms of mixed assertion) is empty. In this case s is independent of $\boldsymbol{X}$ and is a disjunction of atoms of the second set (positively-asserted atoms) and possibly some of the atoms of the first set (purely don’t-care atoms), If the fourth set is not empty, a sneaky translator must be used and s is dependent on $\boldsymbol{X}$ as well as $\boldsymbol{Z}$. An expression of s can be obtained *via* VEKM-reading techniques (Rushdi, 1983, 1985, 1987, 2001, 2004, 2018).

In summary, we use an auxiliary function $g(\boldsymbol{X})$ to encode the information given by the vectorial input function $\boldsymbol{Z}(\boldsymbol{X})$. For $\mathbf{A} \in \{0,1\}^k$, the discriminant $g(\boldsymbol{A})$ is a single atom of the underling Boolean algebra $FB(\boldsymbol{Z})$. This atom is specified as $\boldsymbol{Z}^{\boldsymbol{Z}(\boldsymbol{A})} = \wedge_{i=1}^{n}(Z_i \odot Z_i(\boldsymbol{A})) = \wedge_{i=1}^{n} Z_i^{Z_i(\boldsymbol{A})}$. Now, depending on the value of $t(\boldsymbol{A})$**,** the desired $s(\boldsymbol{A})$ inherits as a value the single-atom constituting $g(\boldsymbol{A})$**:** positively asserted if $t(\boldsymbol{A}) = 1$**,** and negatively asserted if $t(\boldsymbol{A}) = 0$**.** Apart from this single assertion in $t(\boldsymbol{A})$**,** all the remaining atoms are included in its sum-of-products expression don't-care.

In the analysis above, we ignored the presence of the side inputs $\boldsymbol{Y}$. Straightforward modifications are needed if these side inputs are present. Both t and s will now have an extra $\boldsymbol{Y}$ dependence, which means that the $\boldsymbol{Y}$ variables will now appear as extra entered variables in the natural maps of t and s. The underlying Boolean algebra will now be $FB(\boldsymbol{Z},\boldsymbol{Y}) = FB(Z_1, Z_2 \ldots Z_n, Y_1, Y_2 \ldots Y_m)$, and hence will have $(n+m)$ generators, $NM = 2^{(n+m)}$ atoms, and 2^{NM} elements.

In the analysis above and in the examples to follow, we used scalar quantities t and s for simplicity. Generally, we could use vectors $\boldsymbol{t}$ and $\boldsymbol{s}$ of the same length $l > 1$. In this latter case, the single equation (3) is simply replaced by l independent equations of the same form, namely

$$s_i = g t_i \vee d(\bar{g}), \quad 1 \leq i \leq l \tag{3a}$$

Therefore, the task of constructing an s function from a t function is repeated l times, albeit with a common auxiliary function g encoding the common knowledge of $\boldsymbol{Z}(\boldsymbol{X})$. Each of the functions s_i $(1 \leq i \leq l)$ could be an honest translator or a sneaky one independently of the others.

3. Design Example Demanding a Sneaky Translator

This example was solved earlier by Ledley (1960) and Brown (1975). It does not have side inputs Y, and it is specified by:

$$t(\boldsymbol{X}) = X_1X_2 \vee X_1\bar{X}_3 \vee X_2\bar{X}_3 \vee \bar{X}_2X_3 \tag{11}$$

$$Z_1(\boldsymbol{X}) = \bar{X}_1(X_2X_3 \vee \bar{X}_2\bar{X}_3) \vee X_1\bar{X}_2X_3 \tag{12a}$$

$$Z_2(\boldsymbol{X}) = \bar{X}_3(X_1 \vee X_2) \vee \bar{X}_1\bar{X}_2X_3 \tag{12b}$$

We rewrite (12) in equational form as

$$g(\boldsymbol{X},\boldsymbol{Z}) = \big(Z_1 \odot (\bar{X}_1(X_2X_3 \vee \bar{X}_2\bar{X}_3) \vee X_1\bar{X}_2X_3)\big) \ \big(Z_2 \odot (\bar{X}_3(X_1 \vee X_2) \vee \bar{X}_1\bar{X}_2X_3)\big) = 1 \tag{13}$$

Fig. 2 presents a sequence of five maps, three conventional Karnaugh maps for $Z_1(\boldsymbol{X})$, $Z_2(\boldsymbol{X})$, and $t(\boldsymbol{X})$ and two natural maps (VEKMs) for $g(\boldsymbol{X})$ and $s(\boldsymbol{X})$. This sequence of maps illustrates the steps of our method of solution. Note that we view $g(\boldsymbol{X})$ as a function $g\colon B_{16}^3 \rightarrow B_{16}$ instead of as $g\colon B_2^5 \rightarrow B_2$, where $B_{16} = FB(Z_1, Z_2)$ has four atoms, namely $\bar{Z}_1\,\bar{Z}_2$, $\bar{Z}_1\,Z_2$, $Z_1\bar{Z}_2$, and Z_1Z_2. One of these atoms, *viz.*, $Z_1\bar{Z}_2$ has a mixed assertion in the natural map for $s(\boldsymbol{X})$ in Fig. 2. In fact, it is positively asserted in cell $X_1\bar{X}_2X_3$ and negatively-asserted in cells $\bar{X}_1\bar{X}_2\bar{X}_3$ and $\bar{X}_1X_2X_3$. Therefore, an honest translator is impossible. Using VEKM-reading methodology (Rushdi, 1987; Rushdi and Al-Yahya, 2001), we recover the minimal expression for $s(\boldsymbol{X}, \boldsymbol{Z})$ obtained by Brown (1975), namely

$$s = X_1 \vee Z_2 \tag{14}$$

The reader who is unfamiliar with VEKM-reading methodology can be convinced simply of the correctness of (14) as follows. It is possible to include the term Z_2 in a sum-of-products expression of s since the term Z_2 appears don't-care in every cell of the map of s (Fig. 2(e)). It is also possible to include the term X_1 in such an expression since the term 1 appears don't-care in every cell of the loop X_1 of that map (Fig. 2(e)). On the other hand, all instances of the terms entered asserted in that map ($\bar{Z}_1\,Z_2$, $Z_1\bar{Z}_2$, and $\bar{Z}_1\,\bar{Z}_2$) are covered either by the loop of coverage 1 and depth Z_2 or the loop of coverage X_1 and depth 1.

			X_1	
	1	**0**	**0**	**0**
X_3	**0**	**1**	**0**	**1**
		X_2		

$(a)Z_1(\boldsymbol{X})$

			X_1	
	0	**1**	**1**	**1**
X_3	**1**	**0**	**0**	**0**
		X_2		

$(b)Z_2(\boldsymbol{X})$

			X_1	
	$\boldsymbol{Z_1\bar{Z}_2}$	$\boldsymbol{\bar{Z}_1\,Z_2}$	$\boldsymbol{\bar{Z}_1\,Z_2}$	$\boldsymbol{\bar{Z}_1\,Z_2}$
X_3	$\boldsymbol{\bar{Z}_1\,Z_2}$	$\boldsymbol{Z_1\bar{Z}_2}$	$\boldsymbol{\bar{Z}_1\,\bar{Z}_2}$	$\boldsymbol{Z_1\bar{Z}_2}$
		X_2		

$(c)g(\boldsymbol{X})$

			X_1	
	0	**1**	**1**	**1**
X_3	**1**	**0**	**1**	**1**
		X_2		

$(d)t(\boldsymbol{X})$

			X_1	
	$d(\bar{Z}_1 \vee Z_2)$ $\vee\, n(Z_1\bar{Z}_2)$	$\bar{Z}_1\,Z_2 \vee d(1)$	$\bar{Z}_1\,Z_2 \vee d(1)$	$\bar{Z}_1\,Z_2 \vee d(1)$
X_3	$\bar{Z}_1\,Z_2 \vee d(1)$	$d(\bar{Z}_1 \vee Z_2)$ $\vee\, n(Z_1\bar{Z}_2)$	$\bar{Z}_1\,\bar{Z}_2 \vee d(1)$	$Z_1\bar{Z}_2 \vee d(1)$
		X_2		

$(e)s(\boldsymbol{X})$

Fig. 2. Maps illustrating the evolution of the solution of the Example in Sec. 3

4. Design Example Allowing an Honest Translator

This example was solved earlier by Brown (1975). Again, it does not have side inputs $\boldsymbol{Y}$, and it is specified by:

$$t(\boldsymbol{X}) = \bar{X}_1\bar{X}_2X_3 \ \vee\ X_1X_2 \vee\ X_1\bar{X}_3 \tag{15}$$

$$Z_1(\boldsymbol{X}) = \bar{X}_1X_2 \vee \bar{X}_2X_3 \tag{16a}$$

$$Z_2(\boldsymbol{X}) = X_1X_2\,X_3 \vee \bar{X}_1\bar{X}_2 \vee\ \bar{X}_2\,\bar{X}_3 \tag{16b}$$

$$Z_2(\boldsymbol{X}) = \bar{X}_1\,X_3\ \vee\ X_1\,\overline{X}_3\ \vee X_2 \tag{16c}$$

We rewrite (16) in equational form as

$$g(\boldsymbol{X},\boldsymbol{Z}) = \big(Z_1 \odot (\bar{X}_1X_2 \vee \bar{X}_2X_3)\big)\ \ (Z_2 \odot (X_1X_2\,X_3 \vee \bar{X}_1\bar{X}_2 \vee\ \bar{X}_2\,\bar{X}_3))\big(Z_3 \odot (\bar{X}_1\,X_3\ \vee\ X_1\,\overline{X}_3\ \vee X_2)\big) = 1 \tag{17}$$

Fig. 3 presents a sequence of five maps, three conventional Karnaugh maps for $Z_1(\boldsymbol{X})$, $Z_2(\boldsymbol{X})$, and $t(\boldsymbol{X})$ and two natural maps (VEKMs) for $g(\boldsymbol{X})$ and $s(\boldsymbol{X})$. This sequence of maps illustrates the steps of our method of solution. Note that we view $g(X)$ as a function $g\colon B_{256}^3 \rightarrow\ B_{256}$ instead of as $g\colon B_2^6 \rightarrow\ B_2$, where $B_{256} = FB(Z_1,Z_2,Z_3)$ has eight atoms. We note that an honest translator is possible since none of these eight atoms has a mixed assertion in Fig. 3(e). In fact, there are three atoms that are asserted positively ($\bar{Z}_1\,\bar{Z}_2\,Z_3$, $\bar{Z}_1\,Z_2\,Z_3$, and $Z_1Z_2Z_3$), three atoms that are asserted negatively ($\bar{Z}_1\,Z_2\,\bar{Z}_3$, $Z_1\bar{Z}_2\,Z_3$, and $Z_1\,\overline{Z}_2\,\bar{Z}_3$), and two atoms that are asserted neither way ($Z_1Z_2\bar{Z}_3$, and $\bar{Z}_1\,\bar{Z}_2\,\bar{Z}_3$). Hence, s can be made an honest translator as a disjunction of the three positively-asserted atoms together with (possibly) any of the two purely don’t-care atoms. This precisely expresses s as

$$\begin{aligned} s(\boldsymbol{X}) = \bar{Z}_1\,\bar{Z}_2\,Z_3\ \vee\ \bar{Z}_1\,Z_2\,Z_3 \vee\ Z_1Z_2Z_3\ \vee d(Z_1Z_2\bar{Z}_3\ \vee \bar{Z}_1\,\bar{Z}_2\,\bar{Z}_3\,) \\ = \bar{Z}_1\,Z_3\ \vee\ Z_2Z_3\ \vee d(Z_1Z_2 \vee \bar{Z}_1\,\bar{Z}_2\,) \end{aligned} \tag{18}$$

and hence produces three minimal honest solutions for $s(\boldsymbol{X})$, namely

$$s(\boldsymbol{X}) =\ \bar{Z}_1\,Z_3\ \vee\ Z_2Z_3 \tag{19a}$$

$$s(\boldsymbol{X}) =\ Z_2Z_3\ \vee\ \bar{Z}_1\,\bar{Z}_2 \tag{19b}$$

$$s(\boldsymbol{X}) =\ \bar{Z}_1\,Z_3\ \vee\ \ Z_1Z_2 \tag{19c}$$

Using VEKM-reading methodology, we can come up with a sneaky translator of the form

$$s(\boldsymbol{X}) =\ X_1Z_3\ \vee\ X_3Z_2 \tag{20}$$

The reader who is unfamiliar with VEKM-reading methodology can simply verify (20) as follows. It is possible to include the term X_1Z_3 in a sum-of-products expression of s since the term Z_3

appears at least don't-care in every cell of the loop X_1 of the map of s (Fig. 3(e)). It is also possible to include the term X_3Z_2 in such an expression since the term Z_2 appears at least don't-care in every cell of the loop X_3 of that map (Fig. 3(e)). On the other hand, all instances of the terms entered asserted in that map ($Z_1Z_2Z_3$, $\bar{Z}_1\bar{Z}_2Z_3$, $\bar{Z}_1Z_2Z_3$, and $\bar{Z}_1Z_2Z_3$) are covered either by the loop of coverage X_1 and depth Z_3 or the loop of coverage X_3 and depth Z_2. An alternative sneaky translator is given by the products-of-sums form

$$s(\boldsymbol{X}) = Z_3(X_1 \vee \bar{X}_2) \tag{21}$$

5. Design Example Including Side Inputs

This example was solved earlier by Ledley (1960) and Brown (1975) and it differs from the earlier two examples, since it has side inputs $\boldsymbol{Y}$. The example is specified by:

$$t(\boldsymbol{X},\boldsymbol{Y}) = X_1X_2 \vee Y_1\bar{X}_2X_3 \vee Y_2(X_1\bar{X}_3 \vee X_2) \tag{22}$$

$$Z(\boldsymbol{X}) = \begin{bmatrix} X_1\bar{X}_2 \vee \bar{X}_1X_3 \\ \bar{X}_1X_2 \vee X_1\bar{X}_3 \\ X_1X_2 \vee \bar{X}_2X_3 \end{bmatrix} \tag{23}$$

We rewrite (19) in equational form as

$$g(\boldsymbol{X},\boldsymbol{Z}) = \big(Z_1 \odot (X_1\bar{X}_2 \vee \bar{X}_1X_3)\big)\big(Z_2 \odot (\bar{X}_1X_2 \vee X_1\bar{X}_3)\big)\big(Z_3 \odot (X_1X_2 \vee \bar{X}_2X_3)\big) = 1 \tag{24}$$

Instead of representing $t(\boldsymbol{X},\boldsymbol{Y})$ by a conventional map, we represent it by the VEKM of Fig. 4, which has entered variables Y_1 and Y_2. The function $s(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$ is constructed via (10) and then viewed as $s(\boldsymbol{X})\colon B^3 \to B$, where $B = FB(Z_1, Z_2, Z_3, Y_1, Y_2)$ has 32 atoms. None of these atoms has a mixed assertion in the map of s, indicating the possibility of an honest translator $s(\boldsymbol{Z},\boldsymbol{Y})$ that is independent of $\boldsymbol{X}$. This translator is the disjunction of fourteen positively-asserted atoms, together with two out of eight purely don't care atoms, and is given by

$$s(\boldsymbol{Z},\boldsymbol{Y}) = \bar{Z}_1Z_3 \vee Y_1Z_3 \vee Y_2Z_2 \tag{25}$$

			X_1	
	0	1	0	0
X_3	1	1	0	1
		X_2		

$(a) Z_1(\boldsymbol{X})$

			X_1	
	1	0	0	1
X_3	1	0	1	0
		X_2		

$(b) Z_2(\boldsymbol{X})$

			X_1	
	0	1	1	1
X_3	1	1	1	0
		X_2		

$(c) Z_3(\boldsymbol{X})$

			X_1	
	$\bar{Z}_1 Z_2 \bar{Z}_3$	$Z_1 \bar{Z}_2 Z_3$	$\bar{Z}_1 \bar{Z}_2 Z_3$	$\bar{Z}_1 Z_2 Z_3$
X_3	$Z_1 Z_2 Z_3$	$Z_1 \bar{Z}_2 Z_3$	$\bar{Z}_1 Z_2 Z_3$	$Z_1 \bar{Z}_2 \bar{Z}_3$
		X_2		

$(c) g(\boldsymbol{X})$

			X_1	
	0	0	1	1
X_3	1	0	1	0
		X_2		

$(d) t(\boldsymbol{X})$

			X_1	
	$d(Z_1 \vee \bar{Z}_2 \vee Z_3) \vee n(\bar{Z}_1 Z_2 \bar{Z}_3)$	$d(\bar{Z}_1 \vee Z_2 \vee \bar{Z}_3) \vee n(Z_1 \bar{Z}_2 Z_3)$	$\bar{Z}_1 \bar{Z}_2 Z_3 \vee d(1)$	$\bar{Z}_1 Z_2 Z_3 \vee d(1)$
X_3	$Z_1 Z_2 Z_3 \vee d(1)$	$d(\bar{Z}_1 \vee Z_2 \vee \bar{Z}_3) \vee n(Z_1 \bar{Z}_2 Z_3)$	$\bar{Z}_1 Z_2 Z_3 \vee d(1)$	$d(\bar{Z}_1 \vee Z_2 \vee Z_3) \vee n(Z_1 \bar{Z}_2 \bar{Z}_3)$
		X_2		

$(e) s(\boldsymbol{X})$

Fig. 3. Maps illustrating the evolution of the solution of the Example in Sec. 4

			X_1	
	0	Y_2	**1**	Y_2
X_3	Y_1	Y_2	**1**	Y_1
		X_2		

$t(\boldsymbol{X}, \boldsymbol{Y})$

Fig. 4. VEKM representation for $\boldsymbol{t}(\boldsymbol{X}, \boldsymbol{Y})$ in the Example in Sec. 5

6. Conclusions

This paper handles the generalized Type-2 problem, a prominent fundamental problem of digital circuit design. The paper adapts the input-domain constraining technique *via* utilization of Variable-Entered Karnaugh Maps (VEKMs) together with careful employment of modern don't-care notation. An auxiliary function is used to encode the information content of the vectorial input function, by using single atoms of the underling Boolean algebra as its discriminants . The desired output has discriminants that consist each of all atoms don't-care except a single inherited atom that could be positively or negatively asserted.

Our novel method outperforms the conventional method both in speed and clarity. Three demonstrative examples are worked out herein, which obtain solutions that agree with (or occasionally are more compact than) solutions obtained conventionally. The advantages of our method are not restricted to pedagogical aspects but extend to practical ones as well.

The paper serves as an additional illustration of a thesis forwarded by Brown (1990) stating that 'big' Boolean algebras (atomic Boolean algebras with more than two elements) are *unavoidable*, especially for digital-design purposes. The paper also demonstrates how representations and manipulations in these algebras can be easily tackled pictorially and semi-algebraically with the aid of natural maps (VEKMs).

Appendix A: On Expressing Interval Functions *via* Don't-Care Notation

An interval Boolean function defined by the double-inequality

$$A \leq f \leq B \tag{A.1}$$

can conveniently be depicted by the don't-care notation introduced by Reusch (1975) and developed further by Rushdi (1987, 2001, 2004, 2018). According to Reusch (1975), the notation

$$f = A \vee d(B) \tag{A.2}$$

is rigorously equivalent to the conjunction of the two implications

$$\{A = 1\} \rightarrow \{f = 1\} \tag{A.3a}$$
$$\{A = 0\} \wedge \{B = 0\} \rightarrow \{f = 0\} \tag{A.3b}$$

and is, therefore, exactly equivalent to expressing the function f *via* the double-inequality (A.1). Note that (A.3) keeps silent about the value of f when

$$\{A = 1\} \wedge \{B = 1\} \tag{A.4}$$

This means that the function f becomes a partially-defined or an incompletely-specified one. In fact, it stands for several functions rather than a unique one. An implicit purpose of the notation (A.1) or (A.2) is to mathematically handle these several functions as if they were a single one. The notation (A.2) is understood (in the case of several variables $\boldsymbol{X}$) (Rushdi and Al-Yahya, 2001) to mean

$$f(\boldsymbol{X}) = A(\boldsymbol{X}) \vee (\mathrm{d}(\boldsymbol{X}) \wedge B(\boldsymbol{X})) \tag{A.5}$$

but while the arbitrary function $d(\boldsymbol{X})$ is clearly understood to be dependent on $\boldsymbol{X}$, this dependency is *deliberately hidden* or made implicit to facilitate manipulations. Hiding the dependence of the don't-care symbol d on the pertinent variables is a common practice in digital-design circles, in general, and in Karnaugh-map applications, in particular. The complement $\bar{f}$ of the function f given by (A.2) is given by the sum-of-products formula (Rushdi and Albarakati, 2014)

$$\bar{f} = \bar{A}\bar{B} \vee d(\bar{A}) \tag{A.6}$$

The function A in (A.2) is a completely-specified function and is called the (positively-) asserted part of f, while the function B in (A.2) is an incompletely-specified function and is called the don't-care part of f (Rushdi, 1987). Note that the functions A and B might conveniently be assumed to be disjoint (albeit they do not necessarily have to be so). Therefore, many variants of (A.2) might exist. If the functions A and B are disjoint, these variants include one in which the don't-care part is minimized, namely

$$f = A \vee d(\bar{A}B) \tag{A.7}$$

and also include another in which the don't-care part is maximized, namely

$$f = A \vee d(A \ \vee \ B) \tag{A.8}$$

The positively-asserted part $\bar{A}\bar{B}$ of $\bar{f}$ is also called the negatively-asserted part of f, and is occasionally needed explicitly in the expression of f (Rushdi, 2018). In such a case (A.2) is rewritten as

$$f = A \vee d(B) \vee n(\bar{A}\bar{B}) \tag{A.9}$$

References

Bell, N. R. (1968). A map method for the teaching of the fundamental concepts of compound-input logic circuits. *IEEE Transactions on Education*, *E-11*(3), 173-177.

Brown, F. M. (1975). The constrained-input problem. *IEEE Transactions on Computers*, *C-24*(1), 102-106.

Brown, F. M. (1990). *Boolean reasoning: the logic of Boolean equations*. Kluwer Academic Publishers, Boston, USA.

Brown, S. D., & Vranesic, Z. (2014). *Fundamentals of digital logic with Verilog design. Third Edition*, McGraw-Hill, New York, NY, USA.

Ledley, R. S. (1960). *Digital computer and control engineering.* McGraw-Hill, New York, NY, USA.

Ledley, R. S. (1959). Boolean matrix equations in digital circuit design. *IRE Transactions on Electronic Computers*, *EC-8*(2), 131-139.

Reusch, B. (1975). Generation of prime implicants from subfunctions and a unifying approach to the covering problem. *IEEE Transactions on Computers*, *C-24*(9), 924-930.

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 Sciences*, *13*(1), 111-152.

Rushdi, A. M., & Amashah, M. H. (2010). Parametric general solutions of Boolean equations via variable-entered Karnaugh maps. *Journal of Qassim University: Engineering and Computer Sciences*, *3*(1), 59-71.

Rushdi, A. M., & Amashah, M. H. (2011). Using variable–entered Karnaugh maps to produce compact parametric general solutions of Boolean equations. *International Journal of Computer Mathematics*, *88*(15), 3136-3149.

Rushdi, A. M. (1983). Symbolic reliability analysis with the aid of variable-entered Karnaugh maps. *IEEE Transactions on Reliability*, *R-32* (2), 134-139.

Rushdi, A. M. (1985). Map derivation of the minimal sum of a switching function from that of its complement. *Microelectronics and Reliability*, *25*(6), 1055-1065

Rushdi, A. M. A., & Ahmad, W. (2018). Digital circuit design utilizing equation solving over 'big' Boolean algebras*. International Journal of Mathematical*, *Engineering and Management Sciences* (Accepted).

Rushdi, A. M. A., & Albarakati, H. M. (2014). Prominent classes of the most general subsumptive solutions of Boolean equations. *Information Sciences*, *281*, 53-65.

Rushdi, A. M. A. (2018). Utilization of Karnaugh maps in multi-value qualitative comparative analysis. *International Journal of Mathematical, Engineering and Management Sciences*, *3*(1), 28-46.

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. (2002). Variable-entered Karnaugh map procedures for obtaining the irredundant disjunctive forms of a switching function from its complete sum. *Journal of King Saud University: Engineering Sciences*, *14*(1), 13-27.

Rushdi, A. M. (1987). Improved variable-entered Karnaugh map procedures. *Computer and Electrical Engineering*, *13*(1), 41-52.

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

Rushdi, A. M. (2004). Efficient solution of Boolean equations using variable-entered Karnaugh maps. *Journal of King Abdulaziz University: Engineering Sciences*, *15*(1), 105-121.