Original Research Article

Logical Design of n-bit Comparators: Pedagogical Insight from Eight-Variable Karnaugh Maps

Abstract

An n-bit comparator is a celebrated combinational circuit that compares two n-bit inputs $\boldsymbol{Y}$ and $\boldsymbol{Z}$ and produces three orthonormal outputs: G (indicating that $\boldsymbol{Y}$ is strictly greater than $\boldsymbol{Z}$), E (indicating that $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are equal or equivalent), and L (indicating that $\boldsymbol{Y}$ is strictly less than $\boldsymbol{Z}$). The symbols 'G', 'E', and 'L' are deliberately chosen to convey the notions of 'Greater than,' 'Equal to,' and 'Less than,' respectively. This paper analyzes an n-bit comparator in the general case of arbitrary n and visualizes the analysis for $n = 4$ on a regular and modular version of the 8-variable Karnaugh-map. The cases $n = 3$, 2, and 1 appear as special cases on 6-variable, 4-variable, and 2-variable submaps of the original map. The analysis is a tutorial exposition of many important concepts in switching theory including those of implicants, prime implicants, essential prime implicants, minimal sum, complete sum and disjoint sum of products (or probability-ready expressions).

Key Words

Comparator, Karnaugh map, Prime implicant, Minimal sum, Complete sum, Probability-ready expression.

1. Introduction

Modern logic digital design handles real-life problems that involve very large numbers of variables, and hence are not amenable to solution *via* heuristic manual tools but are solvable only *via* computerized algorithms. However, there is one heuristic manual tool, namely, the Karnaugh map [1-23], that plays an indispensable role in logic design as it provides pictorial insight in demonstrating concepts, proving theorems, and understanding procedures by showing their details in small examples. The literature abounds with contributions that offer instructive and pedagogical expositions of the Karnaugh map and related logic design tools [24-30]. The purpose of this paper is to make yet another such contribution, as it provides a tutorial exposition of a regular and modular version of the Karnaugh map [31-34] and to utilize this version in presenting many important concepts of switching theory and logic design. This map version can be (theoretically) extended to an arbitrary large number of variables, and includes all maps of smaller sizes as special cases.

The Karnaugh map is an enhanced form of the truth table [9], in which two dimensions (rather than one dimension) are used, and in which reflected binary ordering or grey ordering (as opposed to direct binary ordering) is employed. The n-variable map consists of 2^n cells, such that every cell has n neighboring cells or logically adjacent cells. Two cells are (first) neighbors or (immediately) adjacent if their variable values except one are exactly the same. Such two cells are said to have a Hamming distance [35-43] of one or to differ in exactly one-bit position. The map is constructed such that any two logically adjacent cells are made also as visually adjacent as possible. In general, two logically adjacent cells appear as the mirror images with respect to boundary lines separating the internal and external domains of the variable in whose value the two cells differ (See Fig. 1).

Typically, the Karnaugh map is conveniently used up to six variables [4]. There are occasions in which Karnaugh maps of eight variables are used, in which the rectangular shape of cells is abandoned to a triangular shape [44-48]. In this paper, however, we will use the aforementioned regular and modular form of the Karnaugh map that appeared earlier in [31-34], and is such that

a) The rectangular shape of the cell is retained.
b) The internal domain of the $(n + 1)$st variable is introduced to be centered around the boundary lines of the $(n - 1)$st variable (See Fig. 2).

We note that the process outlined in (b) above can be, in theory, indefinitely continued. Hence, there is no theoretical upper bound on the size of the Karnaugh map constructed this way. However, as the number of variables increases, the size of the map increases exponentially, and its utility diminishes very quickly due to prohibitively increasing difficulty.

As a demonstration of the usefulness of the aforementioned version of the Karnaugh map, we present its case of eight variables herein. We use this map to explore the design of a well-known combinational circuit, namely an n-bit digital magnitude comparator [49-51]. Note that we deal herein only with digital (as opposed to analogue) comparators. A digital comparator typically uses two n-bit inputs **Y** and **Z**, and could possibly be

1. An **Identity Comparator**, which has a single output E such that $E = 1$ when $\boldsymbol{Y} = \boldsymbol{Z}$**,** *i.e.*, when the two inputs match bit for bit.

2. A **Magnitude Comparator**, which has three orthonormal outputs $\{G, E, L\}$, namely $G = 1$ when $\boldsymbol{Y} > \boldsymbol{Z}$, $E = 1$ when $\boldsymbol{Y} = \boldsymbol{Z}$ and $L = 1$ when $\boldsymbol{Y} < Z$.

Note that a magnitude comparator includes an identity comparator as a special case. The magnitude comparator is a redundant circuit in the sense that any of its three outputs might be readily obtained from the other two. Digital Comparators are used widely in Analogue-to-Digital Converters (ADC) and to perform a variety of arithmetic operations in the Arithmetic Logic Units (ALU) of a digital computer.

Karnaugh-map analysis of the digital magnitude comparator is employed herein to provide instructive and pedagogical exposition of many important concepts in logic design and switching theory including those of implicants, prime implicants, essential prime implicants, minimal sum, complete sum and disjoint sum of products (or probability-ready expressions).

The organization of the rest of this paper is as follows. Section 2 presents a mathematical description of an n-bit magnitude digital comparator. Section 3 derives expressions for the comparator outputs in minimal-sum or complete-sum form as well as in probability-ready form. Section 4 concludes the paper. To make the paper self-contained, five appendices are included. Appendix A explains basic concepts of Boolean minimization, Appendix B is about the complete sum. Appendix C defines probability-ready expressions. Appendix D briefly introduces the Boole-Shannon expansion. Appendix E deals with unate Boolean functions.

2. Mathematical Description of an n-bit Comparator

An n-bit comparator is a (combinational) circuit (shown in Fig. 3) that compares two n-bit inputs $\mathbf{Y} = (\mathrm{Y}_{n-1}\mathrm{Y}_{n-2} \ldots \mathrm{Y}_1\mathrm{Y}_0)_2$ and $\mathbf{Z} = (\mathrm{Z}_{n-1}\mathrm{Z}_{n-2} \ldots \mathrm{Z}_1\mathrm{Z}_0)_2$ such that

$$\mathbf{Y} = \sum_{k=0}^{n-1} \mathrm{Y_k}\, 2^k, \tag{1a}$$

$$\mathbf{Z} = \sum_{k=0}^{n-1} \mathrm{Z_k}\, 2^k. \tag{1b}$$

The comparator has three 1-bit outputs, namely

$$G = \{\mathbf{Y} > \mathbf{Z}\}, \tag{2a}$$

$$E = \{\mathbf{Y} = \mathbf{Z}\}, \tag{2b}$$

$$L = \{\mathbf{Y} < \mathbf{Z}\}. \tag{2c}$$

The three variables G, E, and L form an orthonormal set, or in other words, they are mutually exclusive and exhaustive, *i.e.*,

$$G \vee E \vee L = 1. \tag{3a}$$

$$GE = EL = GL = 0. \tag{3b}$$

Consequently, these three variables are inter-related by the following equations.

$$G = \bar{E}\bar{L}, \;\; \bar{G} = E \vee L, \tag{4a}$$

$$E = \bar{G}\bar{L}, \;\; \bar{E} = G \vee L, \tag{4b}$$

$$L = \bar{G}\bar{E}, \;\; \bar{L} = G \vee E. \tag{4c}$$

Figure 4 is a display of the results above for two single-bit inputs $\mathbf{Y} = Y_k$ and $\mathbf{Z} = Z_k$. For this case, we simply obtain

$$G = Y_k\overline{Z_k} = \{Y_k > Z_k\} = \overline{\{Y_k \leq Z_k\}} = \overline{\{Y_k \rightarrow Z_k\}} \tag{5a}$$

$$E = \overline{Y_k}\;\overline{Z_k} \vee Y_k Z_k = \{Y_k \odot Z_k\} = \{Y_k \equiv Z_k\} \tag{5b}$$

$$L = \overline{Y_k}\, Z_k = \{Y_k < Z_k\} = \overline{\{Z_k \leq Y_k\}} = \overline{\{Z_k \rightarrow Y_k\}} \tag{5c}$$

As seen from equations (5), the three variables $G, E,$ and L in the case of single-bit inputs are given by the functions $INHIBIT(Y_k,\, Z_k)$, $XNOR(Y_k,\, Z_k)$, and $INHIBIT(Z_k,\, Y_k)$.

3. Derivation of the Comparator Equations

In this section, we derive the explicit output equations for a 4-bit comparator, and then generalize our results to an n-bit one. by obtaining the output equations in terms of recursive relations and boundary conditions. Figure 5 is a flow chart that compares the bits Y_k to the bits Z_k (starting from the *most significant* bit and ending with the *least significant* one, *i.e.*, for $k = 3, 2, 1,$ and 0. As the flow chart indicates, the three outputs denoted as $G_4, E_4, and\ L_4$ are given by

$$\mathrm{G}_4 = Y_3\overline{Z_3} \vee (\overline{Y_3}\,\overline{Z_3} \vee Y_3Z_3)(Y_2\overline{Z_2} \vee (\overline{Y_2}\,\overline{Z_2} \vee Y_2Z_2)(Y_1\overline{Z_1} \vee (\overline{Y_1}\,\overline{Z_1} \vee Y_1Z_1)Y_0\overline{Z_0})) \tag{6a}$$

$$\mathrm{E}_4 = \wedge_{m=0}^{3} \ (\overline{Y_m}\,\overline{Z_m} \vee Y_mZ_m) \tag{6b}$$

$$\mathrm{L}_4 = \overline{Y_3}Z_3 \vee (\overline{Y_3}\,\overline{Z_3} \vee Y_3Z_3)(\overline{Y_2}Z_2 \vee (\overline{Y_2}\,\overline{Z_2} \vee Y_2Z_2)(\overline{Y_1}Z_1 \vee (\overline{Y_1}\,\overline{Z_1} \vee Y_1Z_1)\overline{Y_0}Z_0)) \tag{6c}$$

Equations (6) are demonstrated by the 8-variable Karnaugh map in Fig. 6, where the cells for which $G_4 = 1$ are entered by G and given a light blue color, while the cells for which $L_4 = 1$ are entered by L and given a pale red color, and the cells for which $E_4 = 1$ are entered by E and left uncolored. The single map in Fig. 6 is obtained by combining three maps for the orthonormal variables G_4, L_4, and E_4. Both the cells for the functions G_4 and L_4 are covered by disjoint (non-overlapping loops). For each of these two functions, there is one 64-cell loop, two 16-cell loops, four 4-cell loops, and eight 1-cell loops. These loops come in four consecutive stages, with the loops in a succeeding stage doubling in number and diminishing to quarter size, compared to the loops in the preceding stage. Remarkable symmetry could be observed with respect to the main diagonal of the map.

Figure 6 is, in a sense, a summary of the results of equations (6) (for the 4-bit comparator) demonstrated on an 8-variable Karnaugh map with inputs $\mathbf{Y} = (\mathrm{Y}_3\mathrm{Y}_2\mathrm{Y}_1\mathrm{Y}_0)_2$ and $\mathbf{Z} = (\mathrm{Z}_3\mathrm{Z}_2\mathrm{Z}_1\mathrm{Z}_0)_2$. Though the analysis is intended for $n = 4$ on the 8-variable map, the cases $n = 3, 2,$ and 1 appear as special cases on 6-variable, 4-variable, and 2-variable submaps of the original map. The top left quarter of this map is a 6-variable submap representing a 3-bit comparator with inputs $\mathbf{Y} = (\mathrm{Y}_2\mathrm{Y}_1\mathrm{Y}_0)_2$ and $\mathbf{Z} = (\mathrm{Z}_2\mathrm{Z}_1\mathrm{Z}_0)_2$. Again, the top left quarter of this submap is a 4-variable submap representing a 2-bit comparator with inputs $\mathbf{Y} = (\mathrm{Y}_1\mathrm{Y}_0)_2$ and

$\mathbf{Z} = (\mathrm{Z}_1\mathrm{Z}_0)_2$. Finally, the top left quarter of this latter submap is a 2-variable submap representing a 1-bit comparator with inputs $\mathbf{Y} = (\mathrm{Y}_0)_2$ and $\mathbf{Z} = (\mathrm{Z}_0)_2$.

The analysis above for $n = 4$ on the 8-variable map of Fig. 6 can also be extended to higher (encompassing) values of n. Figure 7 demonstrates the construction of the $2n$-variable map (for the n-bit comparator) from a $2(n-1)$-variable map (for the $(n-1)$-bit comparator). Theoretically, such a construction can be inductively continued without limit. Therefore, one can easily imagine how the maps for $n = 5, 6, 7 \ldots etc.$ look like. The $2n$-variable map might be viewed as a *map-entered map* [52-54] with two map variables Y_n and Z_n, and four major cells, each of which having the size of a $2(n-1)$-variable map. The middle point of this new map is taken for a center of symmetry. Initially, the major cell $\overline{Y_n}\,\overline{Z_n}$ is filled with the original $2(n-1)$-variable map as it is. Next, the major cell $Y_n Z_n$ is filled with the original $2(n-1)$-variable map reflected with respect to the center of symmetry, while the major cell $Y_n\overline{Z_n}$ is filled uniformly with a $'G'$ in each of its cells. Finally, the major cell $\overline{Y_n}\ Z_n$ is filled uniformly with an $'L'$ in each of its cells. In fact, one can start with a base case of the 2-variable map with inputs $\mathbf{Y} = (\mathrm{Y}_0)_2$ and $\mathbf{Z} = (\mathrm{Z}_0)_2$, and use the recursive step suggested by Fig. 7 repeatedly, so as to construct any desirable $2n$-variable map.

Equations (6) constitute probability-ready expressions [55-60] (See Appendix C), and hence, can be converted, on a one-to one basis, to the corresponding expectation expressions

$$\mathrm{E}\{G_4\} = E\{Y_3\}E\{\overline{Z_3}\} + (E\{\overline{Y_3}\}E\{\overline{Z_3}\} + E\{Y_3\}E\{Z_3\})(E\{Y_2\}E\{\overline{Z_2}\} + (E\{\overline{Y_2}\}E\{\overline{Z_2}\} + E\{Y_2\}E\{Z_2\})(E\{Y_1\}E\{\overline{Z_1}\} + (E\{\overline{Y_1}\}E\{\overline{Z_1}\} + E\{Y_1\}E\{Z_1\})E\{Y_0\}E\{\overline{Z_0}\})). \quad (7a)$$

$$E\{E_4\} = \wedge_{m=0}^{3}\ (E\{\overline{Y_m}\}E\{\overline{Z_m}\} + E\{Y_m\}E\{Z_m\}). \quad (7b)$$

$$\mathrm{E}\{L_4\} = E\{\overline{Y_3}\}E\{Z_3\} + (E\{\overline{Y_3}\}E\{\overline{Z_3}\} + E\{Y_3\}E\{Z_3\})(E\{\overline{Y_2}\}E\{Z_2\} + (E\{\overline{Y_2}\}E\{\overline{Z_2}\} + E\{Y_2\}E\{Z_2\})(E\{\overline{Y_1}\}E\{Z_1\} + (E\{\overline{Y_1}\}E\{\overline{Z_1}\} + E\{Y_1\}E\{Z_1\})E\{\overline{Y_0}\}E\{Z_0\})). \quad (7c)$$

A generalization of equations (6) is possible *via* the following recursive relations

$$\mathrm{G}_k = Y_k\overline{Z_k} \vee (\overline{Y_k}\,\overline{Z_k} \vee Y_k Z_k)\, G_{k-1}, \qquad 1 \le k \le n \quad (8a)$$

$$E_k = (\overline{Y_k}\,\overline{Z_k} \vee Y_k Z_k)\, E_{k-1}, \qquad 1 \le k \le n \quad (8b)$$

$$L_k = \overline{Y_k} Z_k \vee (\overline{Y_k}\,\overline{Z_k} \vee Y_k Z_k)\, L_{k-1}. \qquad 1 \le k \le n \quad (8c)$$

These recursive relations are used in conjunction with the boundary conditions

$$G_0 = Y_0\overline{Z_0}, \quad (9a)$$

$$E_0 = \overline{Y_0}\,\overline{Z_0} \vee Y_0 Z_0, \quad (9b)$$

$$L_0 = \overline{Y_0} Z_0. \quad (9c)$$

Equations (8) have a complete-sum version of the form

$$CS(G_n) = Y_k\overline{Z_k} \vee (Y_k \vee \overline{Z_k})\, CS(G_{k-1}), \qquad 1 \le k \le n \quad (10a)$$

$$CS(E_k) = (\overline{Y_k}\,\overline{Z_k} \vee Y_k Z_k)\, CS(E_{k-1}), \qquad 1 \le k \le n \quad (10b)$$

$$CS(L_k) = \overline{Y_k} Z_k \vee (\overline{Y_k} \vee Z_k)\, CS(L_{k-1}). \qquad 1 \le k \le n \quad (10a)$$

Equations (10) are used together with a complete-sum version of (9), namely

$$CS(G_0) = Y_0\overline{Z_0}, \quad (11a)$$

$$CS(E_0) = \overline{Y_0}\,\overline{Z_0} \vee Y_0 Z_0, \quad (11b)$$

$$CS(L_0) = \overline{Y_0} Z_0. \quad (11c)$$

Equations (8) and (9) are also probability-ready expressions [55-60] (See Appendix C) that are useful in signal-probability calculations [61-69] as they transform on a one-to-one basis to the probability domain, namely

$$E\{G_k\} = E\{Y_k\}E\{\overline{Z_k}\} + (E\{\overline{Y_k}\}E\{\overline{Z_k}\} + E\{Y_k\}E\{Z_k\})\, E\{G_{k-1}\}, \quad (12a)$$

$$E\{E_k\} = (E\{\overline{Y_k}\}E\{\overline{Z_k}\} + E\{Y_k\}E\{Z_k\})\, E\{E_{k-1}\}, \quad (12b)$$

$$E\{L_k\} = E\{\overline{Y_k}\}E\{Z_k\} + (E\{\overline{Y_k}\}E\{\overline{Z_k}\} + E\{Y_k\}E\{Z_k\})\, E\{L_{k-1}\}. \quad (12a)$$

$$E\{G_0\} = E\{Y_0\}E\{\overline{Z_0}\}, \quad (13a)$$

$$E\{E_0\} = (E\{\overline{Y_0}\}E\{\overline{Z_0}\} + E\{Y_0\}E\{Z_0\}), \quad (13b)$$

$$E\{L_0\} = E\{\overline{Y_0}\}E\{Z_0\}. \quad (13c)$$

Figure 8 displays all the prime implicants of G_4, each on a separate map. There are fifteen prime-implicant loops colored in dark blue to be distinguished from other asserted cells of G_4, which remain colored in light blue. Each of these loops (except the first) in an enlargement of one of the loops in Fig. 6 (entered with G), so as to allow overlapping with earlier loops. Note that all fifteen loops pass through the cell at row 0 and column 15, which is the all-0 cell for **Z** and the all-1 cell for **Y**. These prime-implicant loops are all essential. They come in four consecutive stages, with the loops in a succeeding stage doubling in number and diminishing to half size, compared to the loops in the preceding stage. The function G_4 is a unate function with positive polarity in Y_3, Y_2, Y_1 and Y_0 and with negative polarity in Z_3, Z_2, Z_1 and Z_0 (See Appendix E). The minimal sum (or complete sum) for G_4 is covered by one 64-cell loop, two 32-cell loops, four 16-cell loops, and eight 8-cell loops, and is given by

$$
\begin{aligned}
G_4 = {} & Y_3\overline{Z_3} \vee \\
& Y_3 Y_2 \overline{Z_2} \vee \overline{Z_3} Y_2 \overline{Z_2} \vee \\
& Y_3 \overline{Z_2} Y_1 \overline{Z_1} \vee Y_3 Y_2 Y_1 \overline{Z_1} \vee \overline{Z_3} Y_2 Y_1 \overline{Z_1} \vee \overline{Z_3}\,\overline{Z_2} Y_1 \overline{Z_1} \vee \\
& Y_3 \overline{Z_2}\,\overline{Z_1} Y_0 \overline{Z_0} \vee Y_3 \overline{Z_2} Y_1 Y_0 \overline{Z_0} \vee Y_3 Y_2 Y_1 Y_0 \overline{Z_0} \vee Y_3 Y_2 \overline{Z_1} Y_0 \overline{Z_0} \vee \overline{Z_3} Y_2 \overline{Z_1} Y_0 \overline{Z_0} \vee \overline{Z_3} Y_3 Y_2 Y_0 \overline{Z_0} \vee \\
& \overline{Z_3}\,\overline{Z_2} Y_1 Y_0 \overline{Z_0} \vee \overline{Z_3}\,\overline{Z_2}\,\overline{Z_1} Y_0 \overline{Z_0}
\end{aligned} \tag{14a}
$$

$$
\begin{aligned}
= {} & Y_3 \overline{Z_3} \vee \\
& (Y_3 \vee \overline{Z_3}) Y_2 \overline{Z_2} \vee \\
& (Y_3 \vee \overline{Z_3})(Y_2 \vee \overline{Z_2}) Y_1 \overline{Z_1} \vee \\
& (Y_3 \vee \overline{Z_3})(Y_2 \vee \overline{Z_2})(Y_1 \vee \overline{Z_1}) Y_0 \overline{Z_0}
\end{aligned} \tag{14b}
$$

Note that the factored expression (14b) might be obtained from (10a) and (11a). Similar analysis is possible for the function L_4.

4. Conclusions

This paper is a tutorial on the basic concepts of switching algebra, including Boolean minimization, the complete sum (Blake canonical form), probability-ready expressions, the Boole-Shannon expansion and unate Boolean functions. The topic explored in this tutorial is the design of a well-known combinational circuit, namely the n-bit digital magnitude comparator. The tool employed herein is a regular and modular version of the 8-variable Karnaugh-map, for which the case $n = 4$ of the n-bit comparator is explored. The cases $n = 3$, 2, and 1 appear as special cases on 6-variable, 4-variable, and 2-variable submaps of the original map. The analysis for $n = 4$ on the 8-variable map is shown to be extendible (theoretically without limit) to higher (encompassing) values of n.

Appendix A: Basic Concepts of Boolean Minimization

This Appendix summarizes notions and concepts employed in the minimization of Boolean functions. Additional information is available in Lee [3], Muroga [4], Rushdi [5-7], Hill and Peterson [8], and Roth and Kinney [14].

The two **literals** of a Boolean variable X_m are its complemented form $\bar{X}_m$ and its uncomplemented one X_m. A product (conjunction) of literals is called a **term** $T(\boldsymbol{X})$ if a literal for each variable appears in it at most once, *i.e.*, a term is an irredundant product (conjunction). A redundant product can be reduced to a term by eliminating repeated appearances of a literal through employment of idempotency of 'AND.' The constant 1 is the multiplication (ANDing) identity and is the product or term of no literals. The dual of a term is the irredundant sum (disjunction), called an **alterm**. The constant 0 is the addition (ORing) identity and is the sum or alterm of no literals. The constant 1 is not an alterm and the constant 0 is not a term. A term $T(\boldsymbol{X})$ is an **implicant** of a function $f(\boldsymbol{X})$ (denoted by $T(\boldsymbol{X}) \rightarrow f(\boldsymbol{X})$ or $T(\boldsymbol{X}) \leq f(\boldsymbol{X})$) if every $T(\boldsymbol{X})$ satisfying $\{T(\boldsymbol{X}) = 1\}$ also satisfies $\{f(\boldsymbol{X}) = 1\}$, while the converse is not necessarily true. A term/alterm $T_i(\boldsymbol{X})$ is said to subsume another term/alterm $T_j(\boldsymbol{X})$ if the set of literals of $T_j(\boldsymbol{X})$ is a *subset* of that of $T_i(\boldsymbol{X})$ (*i.e.*, the literals of $T_i(\boldsymbol{X})$ include those of $T_j(\boldsymbol{X})$).

A **prime implicant** $P(\boldsymbol{X})$ of a Boolean function $f(\boldsymbol{X})$ is an implicant of $f(\boldsymbol{X})$ such that no other term subsumed by it is an implicant of $f(\boldsymbol{X})$. An **irredundant disjunctive form** $IDF(f(\boldsymbol{X}))$ of a Boolean function $f(\boldsymbol{X})$ is a disjunction of some of its prime implicants that expresses $f(\boldsymbol{X})$ but ceases to do so upon the removal of one of these prime implicants. A **minimal sum** $MS(f(\boldsymbol{X}))$ (**minimal irredundant**

form $MIF(f(\boldsymbol{X}))$) of a Boolean function $f(\boldsymbol{X})$ is an irredundant disjunctive form for the function with the minimum number of prime implicants such that the total number of their literals is minimum.

An **essential (core)** prime implicant of $f(\boldsymbol{X})$ is a prime implicant that appears in every irredundant disjunctive form for $f(\boldsymbol{X})$. For every essential prime implicant, there exists an asserted minterm of $f(\boldsymbol{X})$ that subsumes it and does not subsume any other prime implicant. This means that the Karnaugh-map loop covering an essential prime implicant is the *only* loop that covers the cell of this asserted minterm. An **absolutely eliminable** prime implicant of $f(\boldsymbol{X})$ is a prime implicant that does not appear in any irredundant disjunctive form for $f(\boldsymbol{X})$. A **conditionally eliminable** prime implicant of $f(\boldsymbol{X})$ is a prime implicant that appears in some irredundant disjunctive form(s) for $f(\boldsymbol{X})$, but that does not appear in other irredundant disjunctive form(s) for $f(\boldsymbol{X})$.

Appendix B: The Complete Sum (Blake Canonical Form)

The **Complete Sum** $CS(f(\boldsymbol{X}))$ of a Boolean function $f(\boldsymbol{X})$ (also called its **Blake Canonical Form** $BCF(f(\boldsymbol{X}))$) is the disjunction (ORing) of *all* its prime implicants, and nothing else [70-78]. The complete sum is a closure, unique and canonical formula for $f(\boldsymbol{X})$. It is the minimal or absorptive special case of a *syllogistic* formula of $f(\boldsymbol{X})$, where a syllogistic formula is defined as a sum-of-products formula, whose terms include, but are not necessarily confined to, all the prime implicants of $f(\boldsymbol{X})$. Complete-sum construction usually requires the two operations of: (a) absorbing a term by another, and (b) generating the consensus of two ORed terms. A brief explanation of these operations follows.

B.1. Absorbing a Term by Another

If a term $T_1(\boldsymbol{X})$ subsumes (implies) another $T_2(\boldsymbol{X})$, then the disjunction $(T_1(\boldsymbol{X}) \vee T_2(\boldsymbol{X}))$ could simply be rewritten as $T_2(\boldsymbol{X})$, *viz.*

$$T_1(\boldsymbol{X}) \vee T_2(\boldsymbol{X}) = T_2(\boldsymbol{X}). \qquad \text{(B.1)}$$

The deletion of $T_1(\boldsymbol{X})$ in (B.1) is called absorption of the subsuming term $T_1(\boldsymbol{X})$ in the subsumed term $T_2(\boldsymbol{X})$. For example, the term $XY\bar{Z}W$ subsumes each of the sixteen terms $XY\bar{Z}W, Y\bar{Z}W$, $X\bar{Z}W, XYW$, $XY\bar{Z}, \bar{Z}W$, $YW, XW, Y\bar{Z}$, $X\bar{Z}, XY$, $W, \bar{Z}$,

Y, X, and 1. Hence, it could be deleted if it is ORed with any of them. The complete sum is an absorptive syllogistic formula, *i.e.*, it is a syllogistic formula in which no term subsumes another.

B.2. Generating the Consensus of Two ORed Terms

Two terms $T_1(\boldsymbol{X})$ and $T_2(\boldsymbol{X})$ have a consensus if and only if they have exactly one opposition, *i.e.*, exactly one variable that appears complemented ($\bar{X}_m$) in one term (say $T_1(\boldsymbol{X})$) and appears uncomplemented (X_m) in the other term. In such a case, the consensus is the ANDing of the remaining literals of the two terms, *i.e.*

$$consenus(T_1(\boldsymbol{X}), T_2(\boldsymbol{X})) = (T_1(\boldsymbol{X}) \,/\, \bar{X}_m) \wedge (T_2(\boldsymbol{X}) \,/\, X_m), \tag{B.2}$$

where (f/t) denotes the Boolean quotient of the function f by the term t, *i.e.*, the restriction of f when t is asserted [59, 70, 78], *viz.*

$$f/t = [f]_{t=1}. \tag{B.3}$$

When two terms have a consensus, their disjunction can be augmented by this consensus, *i.e.*

$$T_1(\boldsymbol{X}) \vee T_2(\boldsymbol{X}) = T_1(\boldsymbol{X}) \vee T_2(\boldsymbol{X}) \vee consenus(T_1(\boldsymbol{X}), T_2(\boldsymbol{X})). \tag{B.4}$$

For example, the terms $A\bar{B}$ and BC have a single opposition and are represented on the Karnaugh map by two disjoint loops sharing a border, and hence their disjunction can be augmented by their consensus $(A\bar{B}/\bar{B}) \wedge (BC/B) = AC$, which is a loop extending across the common border between the original loops and covering the part $A\bar{B}C$ of $A\bar{B}$ and the part ABC of BC. By contrast, the two terms A and BC have zero opposition, and consequently non-disjoint or overlapping loops, and possess zero or no consensus. The two terms $A\bar{B}$ and $\bar{A}B$ have more than one opposition, and consequently disjoint far-away loops, and again possess zero or no consensus [74].

The complete-sum formula $CS(f)$ may be generated by a two-step iterative-consensus procedure, in which (a) a syllogistic formula F for $f(\boldsymbol{X})$ is found by continually comparing terms and adding their consensuses (if any) to the current formula of $f(\boldsymbol{X})$, and (b) the resulting formula is converted to an absorptive one $ABS(F)$, again by continually comparing terms and deleting subsuming terms by absorbing them in their subsumed terms. A streamlined algorithmic version of iterative consensus is the Blake-Tison Method, which produces the complete sum by performing *explicit consensus generation* with respect to each bi-form variable, and following this by *absorption*. Alternatively, a *syllogistic formula* for the function might be produced (without explicit consensus generation) through multiplying out any suitable product-of-sums (pos) expression for the function to produce a sum-of-products (sop) expression [77].

Appendix C: Probability-Ready Expressions

A Probability-Ready Expression [55-60] is a random expression that can be directly transformed, on a one-to-one basis, to its statistical expectation (its probability of being equal to 1) by replacing all logic variables by their statistical expectations, and also replacing logical multiplication and addition (ANDing and ORing) by their arithmetic counterparts. A logic expression is a *PRE* if

a) all *ORed* terms are *disjoint* (*mutually exclusive*), and
b) all *ANDed* sums (alterms) are *statistically independent*.

Appendix D: The Boole-Shannon Expansion

An effective way for converting a Boolean formula into a *PRE* form is to (repeatedly) employ the Boole-Shannon Expansion [59, 70], which takes the following form when implemented *w.r.t.* a single variable X_{k}

$$f(\boldsymbol{X}) = (\bar{X}_k \wedge f(\boldsymbol{X}|0_{\mathrm{k}})) \vee (X_{\mathrm{k}} \wedge f(\mathbf{X}|1_{\mathrm{k}})), \tag{D.1}$$

This Boole-Shannon Expansion expresses the Boolean function $f(\boldsymbol{X})$ in terms of its two subfunctions $f(\boldsymbol{X}|0_{\mathrm{k}})$ and $f(\boldsymbol{X}|1_{\mathrm{k}})$. These subfunctions are equal to the Boolean quotients $f(\boldsymbol{X})/\bar{X}_k$ and $f(\boldsymbol{X})/X_{\mathrm{k}}$, and hence are obtained by restricting X_k in the expression $f(\boldsymbol{X})$ to 0 and 1, respectively. If $f(\boldsymbol{X})$ is a sop expression of n variables, the two subfunctions $f(\boldsymbol{X}|0_{\mathrm{k}})$ and $f(\boldsymbol{X}|1_{\mathrm{k}})$ are functions of at most $(n-1)$ variables. If the Boole-Shannon expansion is applied in sequence to the n variables of $f(\boldsymbol{X})$, the expansion tree is a complete binary tree (usually called a Binary Decision Diagram) of 2^{n} leaves. These leaves are functions of no variables, or constants, and equal the entries of a corresponding conventional Karnaugh map of $f(\boldsymbol{X})$ [79]. Sibling nodes (nodes at the same level) of this expansion tree constitute the entries of a variable-entered (or a map-entered) Karnaugh map of $f(\boldsymbol{X})$ [79].

Appendix E: Unate Boolean Functions

A Boolean function $f(\boldsymbol{X}) = f(X_1, X_2, \dots, X_{k-1}, X_k, X_{k+1}, \dots, X_n)$ is called *unate* if and only if it can be represented as a normal (sum-of-products or product-of-sums) formula in which no variable appears both complemented and un-complemented, *i.e.*, every variable is mono-form and no variable is bi-form. This Boolean function is called *positive* in its argument X_k, if there exists a normal representation of the function in which X_k does not appear complemented. This happens if and only if every occurrence of the literal $\bar{X}_k$ is redundant and can be eliminated, *i.e.*, if and only if there exist functions f_1 and f_2 (independent of X_k) such that [80-86]

$$f(\boldsymbol{X}) = X_k \ f_1(X_1, X_2, \dots, X_{k-1}, X_{k+1}, \dots, X_n) \vee \ f_2(X_1, X_2, \dots, X_{k-1}, X_{k+1}, \dots, X_n). \tag{E.1}$$

A Boolean function $f(\boldsymbol{X})$ is called *negative* in its argument X_k, if there exists a normal representation of the function in which X_k does not appear un-complemented. This happens if and only if every occurrence of the literal X_k is redundant and can be eliminated, i.e., if and only if there exist functions f_3 and f_4 (independent of X_k) such that [80-86]

$$f(\boldsymbol{X}) = \ f_3(X_1, X_2, \dots, X_{k-1}, X_{k+1}, \dots, X_n) \vee \ \bar{X}_k \ f_4(X_1, X_2, \dots, X_{k-1}, X_{k+1}, \dots, X_n). \tag{E.2}$$

If the function $f(\boldsymbol{X})$ is *positive* in its argument X_k, then its subfunctions are $f(\boldsymbol{X}|1_\mathrm{k}) = f(\boldsymbol{X})/X_k = \ f_1 \vee \ f_2$ and $f(\boldsymbol{X}|0_\mathrm{k}) = f(\boldsymbol{X})/\bar{X}_k = \ f_2$, which means that $f(\boldsymbol{X}|0_\mathrm{k}) \leq \ f(\boldsymbol{X}|1_\mathrm{k})$. Similarly, if the function $f(\boldsymbol{X})$ is *negative* in its argument X_k, then its subfunctions are $f(\boldsymbol{X}|1_\mathrm{k}) = f(\boldsymbol{X})/X_k = \ f_3$ and $f(\boldsymbol{X}|0_\mathrm{k}) = f(\boldsymbol{X})/\bar{X}_k = \ f_3 \vee \ f_4$, which means that $f(\boldsymbol{X}|1_\mathrm{k}) \leq \ f(\boldsymbol{X}|0_\mathrm{k})$.

All threshold (linearly-separable) functions are unate, but the converse is not true [87-91]. The function $X_1X_2 \vee X_3X_4$ is an example of a unate function that is not threshold. All the prime implicants of a unate function are essential, so that it has a single irredundant disjunctive form, which serves as both its (unique) minimal sum and its complete sum.

References

[1] Karnaugh, M. (1953). The map method for synthesis of combinational logic circuits. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, *72*(5), 593-599.

[2] Hurley, R. B., (1963). Probability maps, *IEEE Transactions on Reliability*, *R-12*(3): 39-44.

[3] Lee, S. C. (1978). *Modern Switching Theory and Digital Design*, Prentice-Hall, Englewood Cliffs, New Jersey, NJ, USA.

[4] Muroga, S., (1979). *Logic Design and Switching Theory*, John Wiley, New York, NY, USA.

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

[6] Rushdi, A. M. (1986). Map differentiation of switching functions. *Microelectronics and Reliability*, *26*(5), 891-907.

[7] Rushdi, A. M. (1987). Improved variable-entered Karnaugh map procedures. *Computers & electrical engineering*, *13*(1), 41-52.

[8] Hill, F. J., & Peterson, G. R. (1993). *Computer Aided Logical Design with Emphasis on VLSI*, 4th Edition, Wiley, New York, NY, USA.

[9] Rushdi, A. M. A., (1997). Karnaugh map, Encyclopedia of Mathematics, Supplement Volume I, M. Hazewinkel (Editor), Boston, Kluwer Academic Publishers, pp. 327-328. Available at http://eom.springer.de/K/k110040.html.

[10] 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.

[11] 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.

[12] Rushdi, A. M., & Al-Yahya, H. A. (2001). Derivation of the complete sum of a switching function with the aid of the variable-entered Karnaugh map. *Journal of King Saud University-Engineering Sciences*, *13*(2), 239-268.

[13] 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), 13-26.

[14] Roth, C. & Kinney, L., (2014). *Fundamentals of Logic Design,*7th Edition, Cengage Learning, Stamford, CT, USA.

[15] Rushdi, A. M. A., & Ghaleb, F. A. M. (2014). The Walsh spectrum and the real transform of a switching function: A review with a Karnaugh-map perspective. *Journal of Engineering and Computer Sciences, Qassim University*, *7*(2), 73-112.

[16] Fabricius, E. D. (2017). *Modern Digital Design and Switching Theory*. CRC Press.

[17] Cavanagh, J. (2017). *Digital Design and Verilog HDL Fundamentals*. CRC Press.

[18] Rushdi, A. M. A., & Badawi, R. M. S. (2017). Karnaugh-map utilization in Boolean analysis: The case of war termination. *Journal of Qassim University: Engineering and Computer Sciences*, *10*(1), 53-88.

[19] Deschamps, J. P., Valderrama, E., & Terés, L. (2017). *Digital systems: From Logic Gates to Processors*. Springer.

[20] Rushdi, A. M. A., & Badawi, R. M. S. (2017). Karnaugh map utilization in Coincidence Analysis, *Journal of King Abdulaziz University: Faculty of Computers and Information Technology*, *6*(1-2) 37-44.

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

[22] Rushdi, R. A., & Rushdi, A. M. (2018). Karnaugh-map utility in medical studies: The case of Fetal Malnutrition. *International Journal of Mathematical, Engineering and Management Sciences (IJMEMS), 3*(3), 220-244.

[23] Rushdi, A. M. A., & Badawi, R. M. S. (2019). Computer engineers look at Qualitative Comparative Analysis. *International Journal of Mathematical, Engineering and Management Sciences (IJMEMS), 4*(3).

[24] Roth, C. H. (1993, November). Computer aids for teaching logic design. In *Proceedings of IEEE Frontiers in Education Conference-FIE'93* (pp. 188-191). IEEE.

[25] Hacker, C., & Sitte, R. (2004). Interactive teaching of elementary digital logic design with WinLogiLab. *IEEE Transactions on Education*, *47*(2), 196-203.

[26] Solairaju, A., & Periyasamy, R. (2011). Optimal Boolean function simplification through K-map using object-oriented algorithm. *International Journal of Computer Applications*, *15*(7), 28-32.

[27] Baneres, D., Clarisó, R., Jorba, J., & Serra, M. (2014). Experiences in digital circuit design courses: A self-study platform for learning support. *IEEE Transactions on Learning Technologies*, *7*(4), 360-374.

[28] Battistella, P., & von Wangenheim, C. G. (2016). Games for teaching computing in higher education–a systematic review. *IEEE Technology and Engineering Education*, *9*(1), 8-30.

[29] Rushdi, A. M., Zagzoog, S. S. (2018). Derivation of all particular solutions of a 'big' Boolean equation with applications in digital design. . *Current Journal of Applied Science and Technology*. *27*(3), 1-16.

[30] Salhi, Y. (2018, September). Approaches for enumerating all the essential prime implicants. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications* (pp. 228-239). Springer, Cham.

[31] Halder, A. K. (1982). Karnaugh map extended to six or more variables. *Electronics Letters*, *18*(20), 868-870.

[32] Motil, J. M. (2017). Views of digital logic & probability via sets, numberings. Available at: http://www.csun.edu/~jmotil/ccSetNums2.pdf.

[33] Rushdi, A. M., Zagzoog, S. S. & Balamesh, A. S. (2019). Derivation of a scalable solution for the problem of factoring an n-bit integer, *Journal of Advances in Mathematics and Computer Science*, 30(1), 1-22.

[34] Rushdi, A. M. A., & Alsayegh, A. B., Reliability analysis of a commodity-supply multi-state system using the map method, , *Journal of Advances in Mathematics and Computer Science*, 31(2), 1-17.

[35] Fantauzzi, G. (1968, April). Application of Karnaugh maps to Maitra cascades. In *Proceedings of the April 30--May 2, 1968, spring joint computer conference* (pp. 291-296). ACM.

[36] Edwards, C. R., & Hurst, S. L. (1978). A digital synthesis procedure under function symmetries and mapping methods. *IEEE Transactions on Computers*, C-27(11), 985-997.

[37] Heiss, M. (1990). Error-detecting unit-distance code. *IEEE Transactions on Instrumentation and Measurement*, *39*(5), 730-734.

[38] Pomeranz, I., & Reddy, S. M. (1999). Pattern sensitivity: A property to guide test generation for combinational circuits. In *Proceedings Eighth Asian Test Symposium (ATS'99)* (pp. 75-80). IEEE.

[39] Tabandeh, M. (2012). Application of Karnaugh map for easy generation of error correcting codes. *Scientia Iranica*, *19*(3), 690-695.

[40] El-Maleh, A. H., & Oughali, F. C. (2014). A generalized modular redundancy scheme for enhancing fault tolerance of combinational circuits. *Microelectronics Reliability*, *54*(1), 316-326.

[41] Pezeshkpour, P., & Tabandeh, M. (2015). Data bits in Karnaugh map and increasing map capability in error correcting, pp. 1-8. *arXiv preprint arXiv:1502.02253*.

[42] Rushdi, A. M. A., & Ba-Rukab, O. M. (2017). Map calculation of the Shapley-Shubik voting powers: An example of the European Economic Community. *International Journal of Mathematical, Engineering and Management Sciences (IJMEMS)*, *2*(1), 17-29.

[43] Rushdi, A. M. A., & Ba-Rukab, O. M. (2017). Calculation of Banzhaf voting indices utilizing variable-entered Karnaugh maps. British Journal Mathematics and Computer Science, 20(4), 1-17.

[44] Rushdi, A. M., and Al-Khateeb, D. L., A review of methods for system reliability analysis: A Karnaugh-map perspective, *Proceedings of the First Saudi Engineering Conference, Jeddah, Saudi Arabia*, vol. 1, pp. 57-95, (1983).

[45] Rushdi, A. M., Overall reliability analysis for computer-communication networks, *Proceedings of the Seventh National Computer Conference*, Riyadh, Saudi Arabia, pp. 23-38, (1984).

[46] Rushdi, A. M., On reliability evaluation by network decomposition, *IEEE Transactions on Reliability*, R-33(5): 379-384, (1984), Corrections: *ibid*, R-34(4): 319 (1985).

[47] Rushdi, A. M. A., & Zagzoog, S. S. (2018). Design of a digital circuit for integer factorization via solving the inverse problem of logic. *Journal of Advances in Mathematics and Computer Science*, *26*(3), 1-14.

[48] Rushdi, A. M., Zagzoog, S. S., & Balamesh, A. S. (2018). Design of a hardware circuit for integer factorization using a big Boolean algebra. *Journal of Advances in Mathematics and Computer Science*, *27*(1), 1-25.

[49] Taylor, F. J., Gill, R., Joseph, J., & Radke, J. (1988). A 20 bit logarithmic number system processor. *IEEE Transactions on Computers*, *37*(2), 190-200.

[50] Thapliyal, H., Ranganathan, N., & Ferreira, R. (2010, August). Design of a comparator tree based on reversible logic. In *10th IEEE International Conference on Nanotechnology* (pp. 1113-1116). IEEE.

[51] Morrison, M., Lewandowski, M., & Ranganathan, N. (2012, August). Design of a tree-based comparator and memory unit based on a novel reversible logic structure. In *2012 IEEE Computer Society Annual Symposium on VLSI* (pp. 231-236). IEEE.

[52] Rushdi, A. M. (1987). Logic design of NAND (NOR) circuits by the entered-map-factoring method. *Microelectronics Reliability*, *27*(4), 693-701.

[53] Rushdi, A. M., & Ba-Rukab, O. M. (2007). A purely map procedure for two-level multiple-output logic minimization. *International Journal of Computer Mathematics*, *84*(1), 1-10.

[54] Rushdi, A. M., & Ba-Rukab, O. M. (2004). A map procedure for two-level multiple-output logic minimization. In *Proceedings of the Seventeenth National Computer Conference* (pp. 521-532).

[55] Rushdi, A. M., and Goda, A. S., Symbolic reliability analysis via Shannon's expansion and statistical independence, *Microelectronics and Reliability*, 25(6): 1041-1053, (1985).

[56] Rushdi, A. M., and AbdulGhani A. A., A comparison between reliability analyses based primarily on disjointness or statistical independence, *Microelectronics and Reliability*, 33: 965-978, (1993).

[57] Rushdi, A. M. A., & Hassan A. K. (2015). Reliability of migration between habitat patches with heterogeneous ecological corridors, *Ecological modelling*, 304, 1-10.

[58] Rushdi, A. M. A., & Hassan A. K. (2016). An exposition of system reliability analysis with an ecological perspective, *Ecological Indicators*, 63, 282-295.

[59] Rushdi, A. M., & Rushdi, M. A. (2017). *Switching-Algebraic Analysis of System Reliability*, Chapter 6 in M. Ram and P. Davim (Editors), *Advances in Reliability and System Engineering*, Management and Industrial Engineering Series, Springer International Publishing, Cham, Switzerland, 139-161.

[60] Rushdi, A. M., & Alturki, A. M. (2018). Novel representations for a coherent threshold reliability system: A tale of eight signal flow graphs. *Turkish Journal of Electrical Engineering & Computer Sciences*, *26*(1), 257-269.

[61] Parker, K. P., and E. J. McCluskey, Probabilistic treatment of general combinational networks, *IEEE Transactions on Computers*, C-24(6): 668-670, (1975).

[62] Ogus, R. C., The probability of a correct output from a combinational circuit, *IEEE Transactions on Computers*, C-24(5): 534-544, (1975).

[63] McCluskey, E. J., K. P. Parker, and J. J. Shedletsky, Boolean network probabilities and network design, *IEEE Transactions on Computers*, 27 (2): 187-189, (1978).

[64] Krishnamurthy, B., and I. G. Tollis, Improved techniques for estimating signal probabilities. *IEEE Transactions on Computers*, 38 (7): 1041-1045, (1989).

[65] Ercolani, S., M. Favalli, M. Damiani, P. Olivo, and B. Ricco, Estimate of signal probability in combinational logic networks, Proceedings *of the 1st IEEE European Test Conference*, 132-138, (1989).

[66] Jiang, Y., Y. Tang, Y. Wang, and Y. Savaria, Evaluating the output probability of Boolean functions without floating point operations, *IEEE Canadian Conference on Electrical and Computer Engineering*, 1: 433-437, (1999).

[67] Franco, D. T., M. C. Vasconcelos, L. Naviner, and J. F. Naviner, Signal probability for reliability evaluation of logic circuits, *Microelectronics Reliability*, 48 (8): 1586-1591, (2008).

[68] Han, J., H. Chen, E. Boykin, and J. Fortes, Reliability evaluation of logic circuits using probabilistic gate models, *Microelectronics Reliability*, 51 (2): 468-476, (2011).

[69] Qian, W., M. D. Riedel, H. Zhou, and J. Bruck, Transforming probabilities with combinational logic. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30 (9): 1279-1292 (2011).

[70] Brown, F. M., 1990. Boolean Reasoning: The Logic of Boolean Equations, Kluwer Academic Publishers, Boston, USA.

[71] Rushdi, AM, Ba-Rukab, OM. Map derivation of the closures for dependency and attribute sets and all candidate keys for a relational database, Journal of King Abdulaziz University: Engineering Sciences 2014; 25(2): 3-34.

[72] Rushdi, A. M. A., & Albarakati, H. M. (2014). Construction of general subsumptive solutions of Boolean equations via complete-sum derivation. *Journal of Mathematics and statistics*, *10*(2), 155-168.

[73] Rushdi, AM, Alshehri, TM, Zarouan, M, Rushdi, MA. Utilization of the Modern Syllogistic Method in the exploration of hidden aspects in engineering ethical dilemmas. Journal of King Abdulaziz University: Computers and Information Technology 2014, 3(1): 73-127.

[74] Rushdi, AM, Zarouan, M, Alshehri, TM, Rushdi, MA. The incremental version of the Modern Syllogistic Method. Journal of King Abdulaziz University: Engineering Sciences 2015; 26(1): 25-51.

[75] Rushdi, A. M., & Rushdi, M. A. (2015). Utilization of the modern syllogistic method in the service of academic advising. In *Proceedings of KAU Conference on Academic Advising in Higher Education* (pp. 228-241).

[76] Rushdi, AM, and Rushdi, MA. Switching-algebraic algorithmic derivation of candidate keys in relational databases, Proceedings of the IEEE International Conference on Emerging Trends in Communication Technologies (ICETCT-2016), 2016.

[77] Rushdi, A. M. & Rushdi, M. A. (2018). *Mathematics and Examples of the Modern Syllogistic Method of Propositional Logic*, Chapter 6 (pp. 123-167) in Ram, M. (Editor), *Mathematics Applied in Information Systems*, Bentham Science Publishers, Emirate of Sharjah, United Arab Emirates.

[78] Rushdi, A. M. A., & Ghaleb, F. A. M. (2019). Novel Characterizations of the JK Bistables (Flip Flops). *Journal of Engineering Research and Reports*, 4(3), 1-20

[79] Rushdi, A. M. A., & Ghaleb, F. A. M. (2016). A tutorial exposition of semi-tensor products of matrices with a stress on their representation of Boolean functions. *Journal of King Abdulaziz University: Faculty of Computers and Information Technology*, *5*(1-2), 3-41.

[80] McNaughton, R. (1961). Unate truth functions. *IRE Transactions on Electronic Computers*, (1), 1-6.

[81] Choudhury, A., Sarma, D., & Das, S. R. (1967). Minimal Third-order Expressions of Boolean Unate Functions. *International Journal of Control*, *6*(5), 447-459.

[82] Fisher, L. T. (1974). Unateness properties of AND-EXCLUSIVE-OR logic circuits. *IEEE Transactions on Computers*, *C-24*(2), 166-172.

[83] Thayse, A., & Deschamps, J. P. (1977). Logic properties of unate discrete and switching functions. *IEEE Transactions on Computers*, C-26(12), 1202-1212.

[84] Srivatsa, S. K., & Biswas, N. N. (1977). Karnaugh map analysis and synthesis of threshold functions. *International Journal of Systems Science*, *8*(12), 1385-1399.

[85] Hansen, P., & Simeone, B. (1986). Unimodular functions. *Discrete Applied Mathematics*, *14*(3), 269-281.

[86] Feigelson, A., & Hellerstein, L. (1997). The forbidden projections of unate functions. *Discrete Applied Mathematics*, *77*(3), 221-236.

[87] Muroga, S., Tsuboi, T., & Baugh, C. R. (1970). Enumeration of threshold functions of eight variables. *IEEE Transactions on Computers*, *C-19*(9), 818-825.

[88] Rushdi, A. M. (1990). Threshold systems and their reliability. *Microelectronics and Reliability*, *30*(2), 299-312.

[89] Zhang, R., Gupta, P., Zhong, L., & Jha, N. K. (2005). Threshold network synthesis and optimization and its application to nanotechnologies. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *24*(1), 107-118.

[90] Subirats, J. L., Jerez, J. M., & Franco, L. (2008). A new decomposition algorithm for threshold synthesis and generalization of Boolean functions. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *55*(10), 3188-3196.

[91] Rushdi, A. M. A., & Alturki, A. M. (2015). Reliability of coherent threshold systems. *Journal of Applied Sciences*, *15*(3), 431-443.

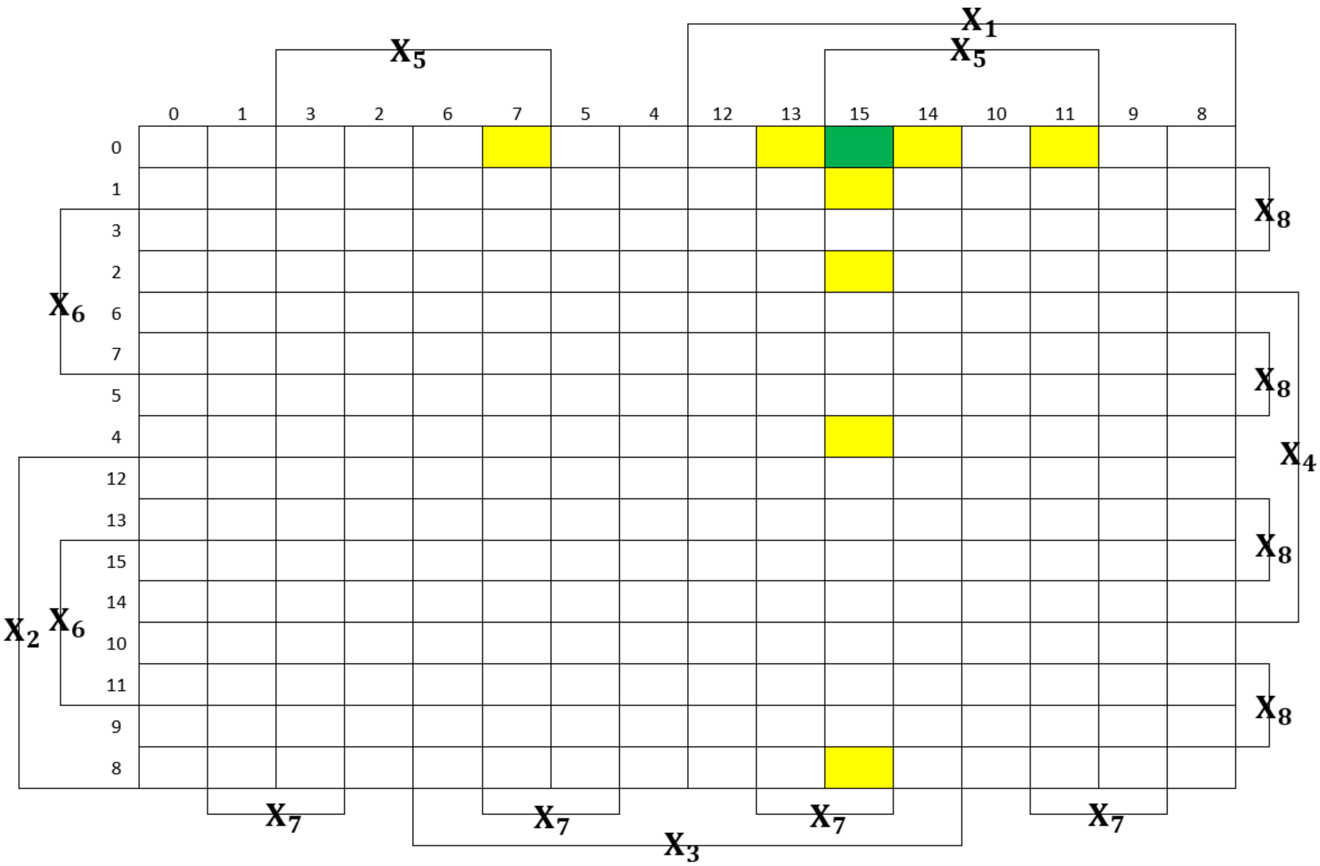


Fig. 1. The general layout of the eight-variable Karnaugh map used herein. The cell colored in green (column 15 and row 0) represents the minterm $X_1\overline{X_2}X_3\overline{X_4}X_5\overline{X_6}X_7\overline{X_8}$ or the bit sequence 10101010. Its eight logically adjacent or neighboring cells are highlighted in yellow. Only four of these cells are visually adjacent to the original cell when the map is viewed to lie on a torus.

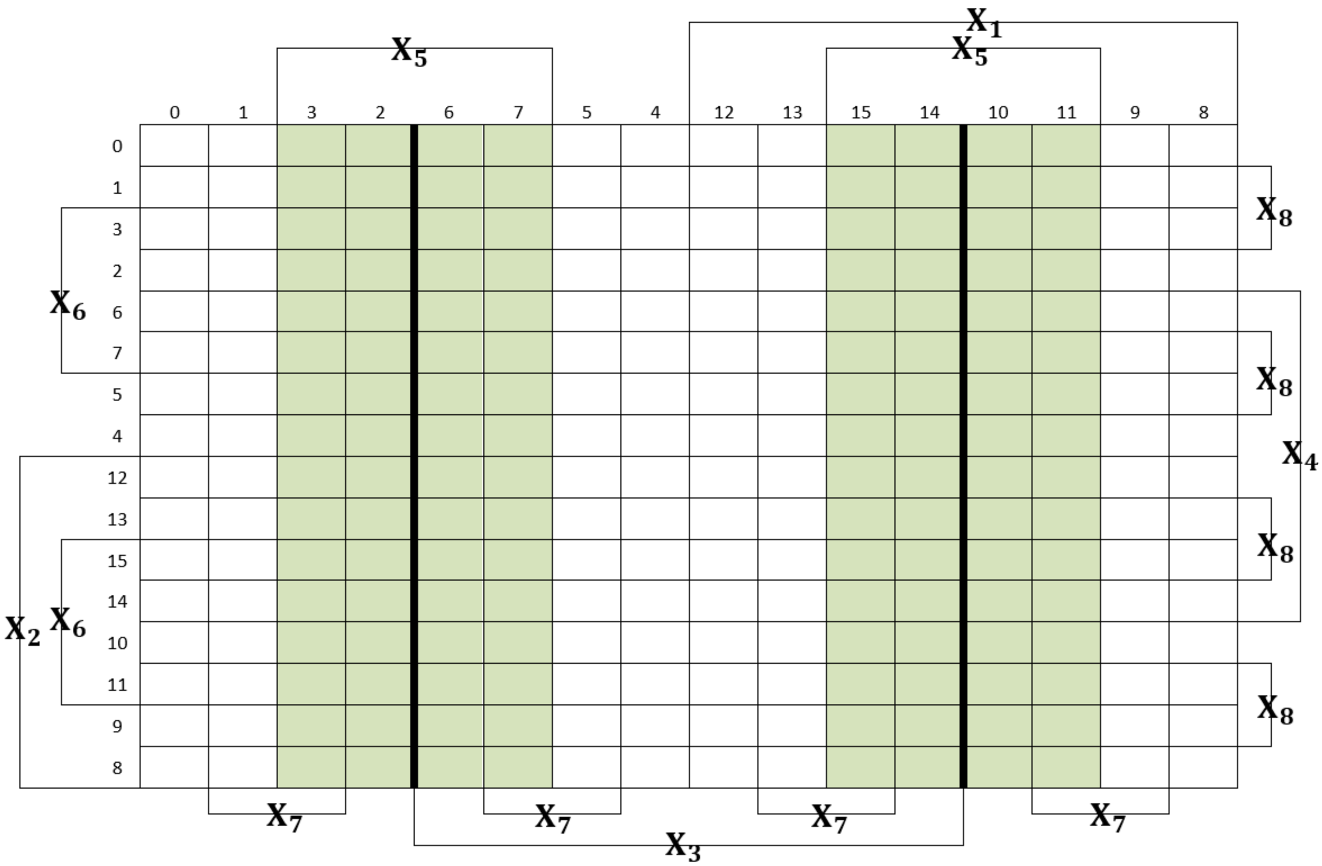


Fig. 2. The eight-variable Karnaugh map of Fig. 1. There are two borders of the variable X_3 (separating its internal domain ($X_3 = 1$) and external domain ($X_3 = 0$)), which are highlighted in bold. There are two internal regions for the variable X_5 (colored) which are centered around these borders.

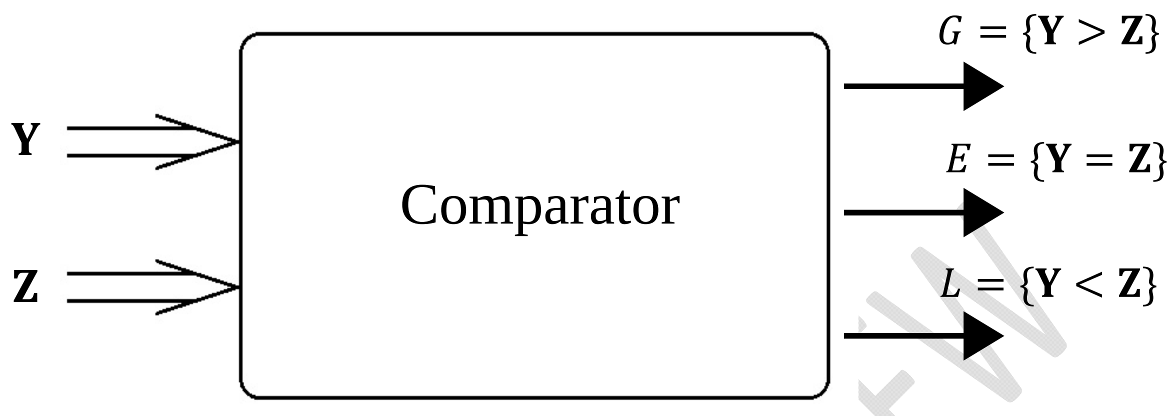


Fig. 3. A comparator is a combinational circuit that compares two n-bit inputs $\mathbf{Y}$ and $\mathbf{Z}$ and produces three orthonormal outputs $G = \{\mathbf{Y} > \mathbf{Z}\}$, $E = \{\mathbf{Y} = \mathbf{Z}\}$ and $L = \{\mathbf{Y} < \mathbf{Z}\}$ such that $G \vee E \vee L = 1$, $GE = EL = LG = 0$, and consequently $G = \bar{E}\bar{L}, E = \bar{G}\bar{L},$ and $L = \bar{G}\bar{E}$.

		$\mathrm{Y_k}$
	$\{\overline{Y_k}\,\overline{Z_k} = 1\} \rightarrow \{E = 1\}$	$\{Y_k\overline{Z_k} = 1\} \equiv \{G = 1\}$
$\mathrm{Z_k}$	$\{\overline{Y_k}Z_k = 1\} \equiv \{L = 1\}$	$\{Y_k Z_k = 1\} \rightarrow \{E = 1\}$

Fig. 4. Karnaugh map for two single-bit inputs Y_k and Z_k. Note that $\{E = 1\} \equiv \{\overline{Y_k}\,\overline{Z_k} = 1\} \vee \{Y_k Z_k = 1\} \equiv \{\overline{Y_k}\,\overline{Z_k} \vee Y_k Z_k = 1\}$.

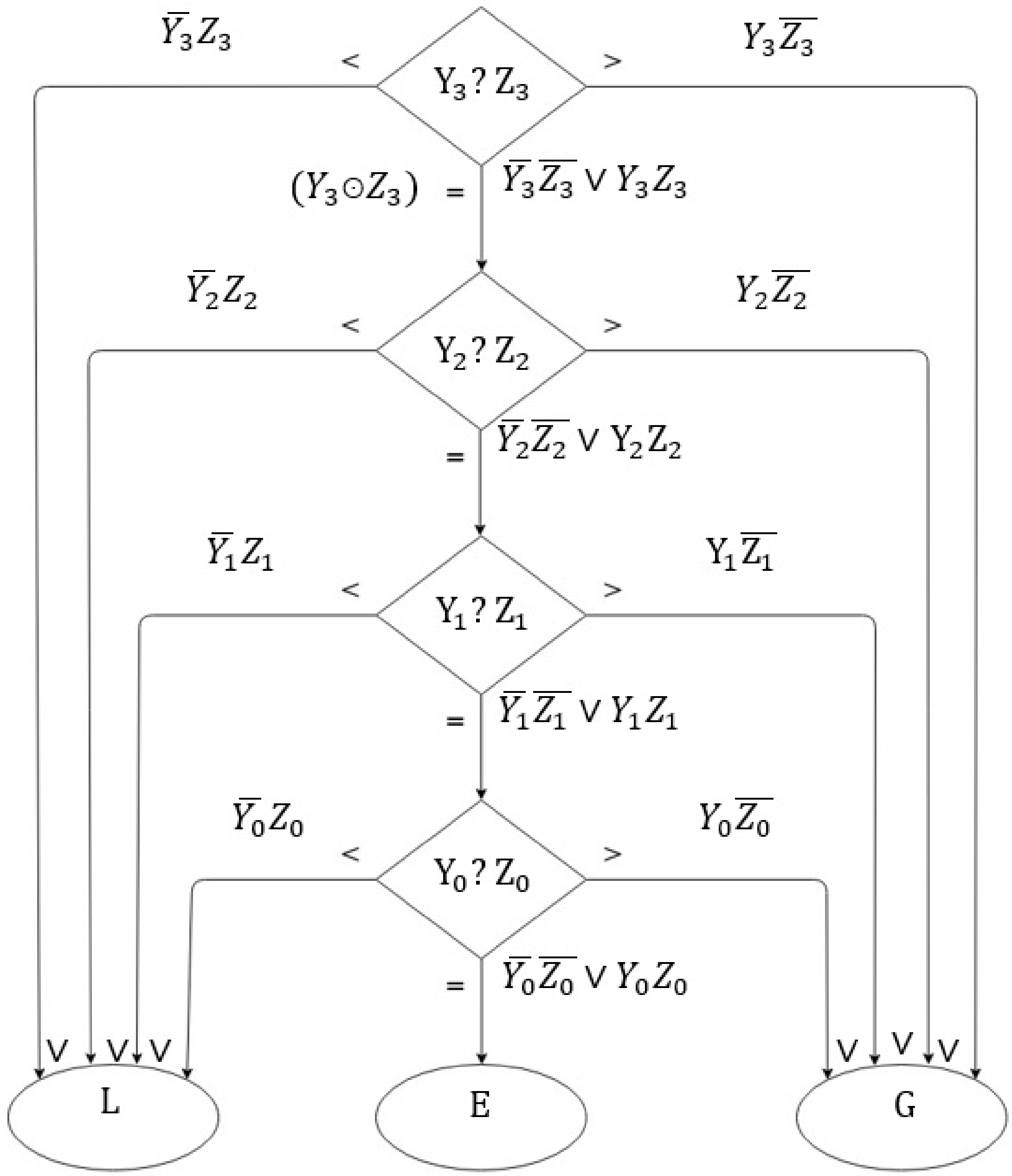


Fig. 5. A flow chart depicting the comparison of a four-bit input $\mathbf{Y} = (\mathrm{Y}_3\mathrm{Y}_2\mathrm{Y}_1\mathrm{Y}_0)_2$ to another four-bit input $\mathbf{Z} = (\mathrm{Z}_3\mathrm{Z}_2\mathrm{Z}_1\mathrm{Z}_0)_2$. The comparator starts by comparing the highest-order or most-significant bits (MSB) first. If equality exists ($\mathrm{Y}_3 = \mathrm{Z}_3$), then the comparator compares the next lower bits and so on until it reaches the lowest-order or least-significant bits (LSB). If equality still exists then the two numbers are defined as being equal ($\boldsymbol{Y} = \boldsymbol{Z}$). If inequality is detected at any stage (either $\mathrm{Y}_\mathrm{k} > \mathrm{Z}_\mathrm{k}$ *or* $\mathrm{Y}_\mathrm{k} < \mathrm{Z}_\mathrm{k}$) the relationship between the two numbers $\boldsymbol{Y}$ and $\boldsymbol{Z}$ is determined (respectively as $\boldsymbol{Y} > \boldsymbol{Z}$ *or* $\boldsymbol{Y} < \boldsymbol{Z}$) and no further comparison is needed.

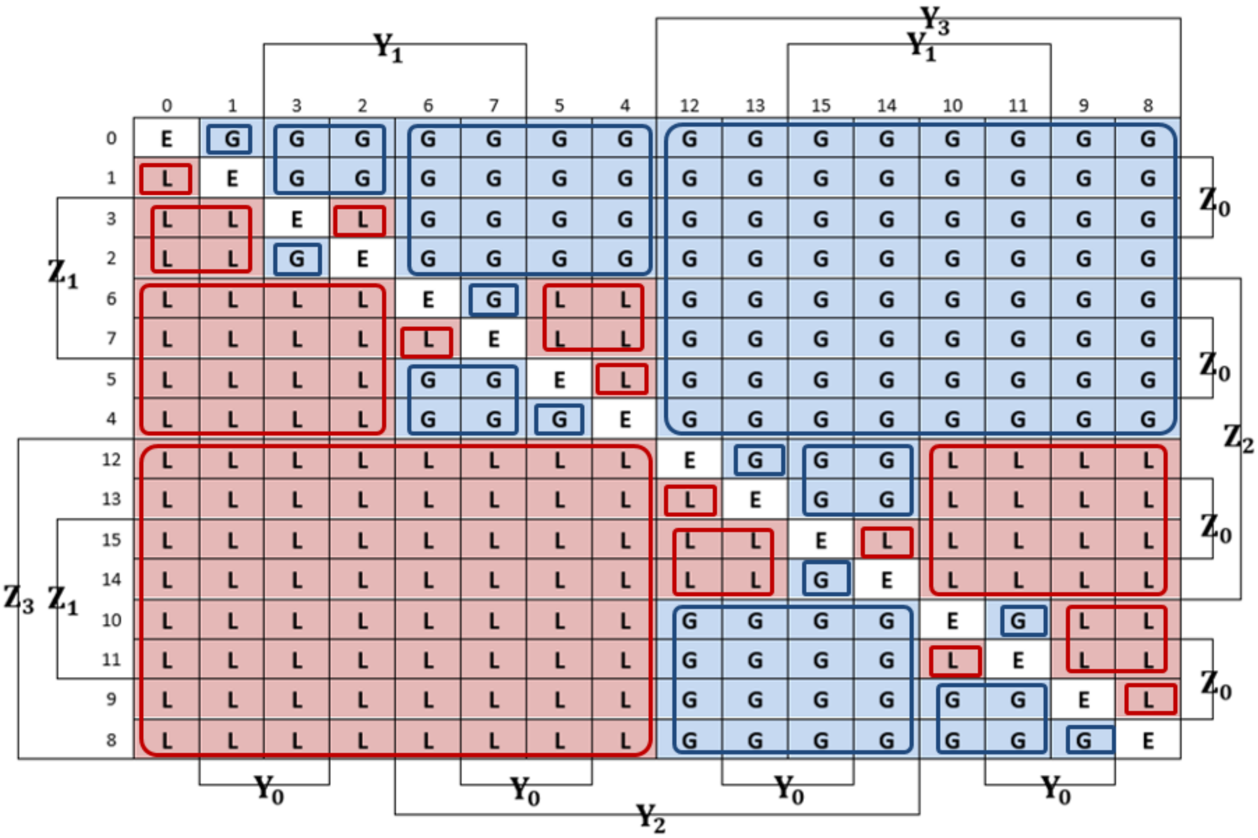


Fig. 6. A summary of the results of equations (6) (for the 4-bit comparator) demonstrated on an 8-variable Karnaugh map with inputs $\mathbf{Y} = (Y_3Y_2Y_1Y_0)_2$ and $\mathbf{Z} = (Z_3Z_2Z_1Z_0)_2$. The top left quarter of this map is a 6-variable submap representing a 3-bit comparator. Again, the top left quarter of this submap is a 4-variable submap representing a 2-bit comparator. Finally, the top left quarter of this latter submap is a 1-variable submap representing a 1-bit comparator. Remarkable symmetry could be observed with respect to the main diagonal of the map.

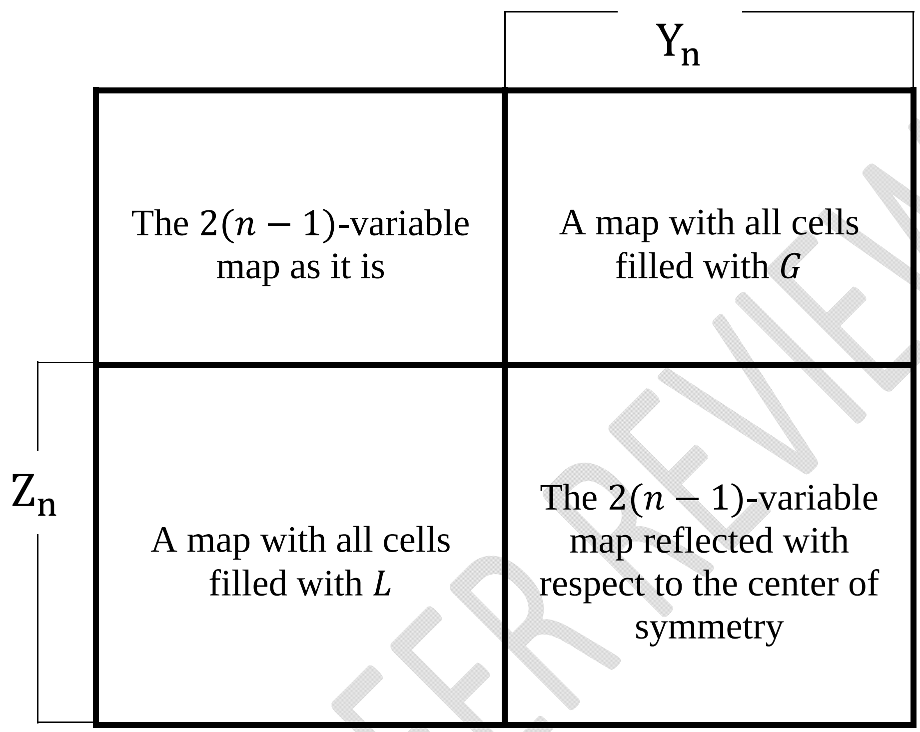
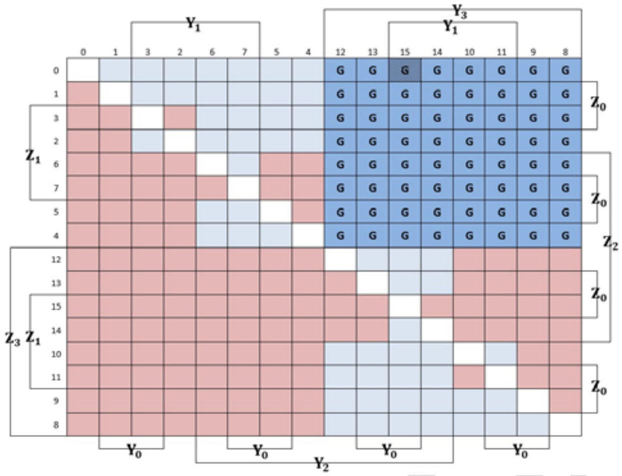
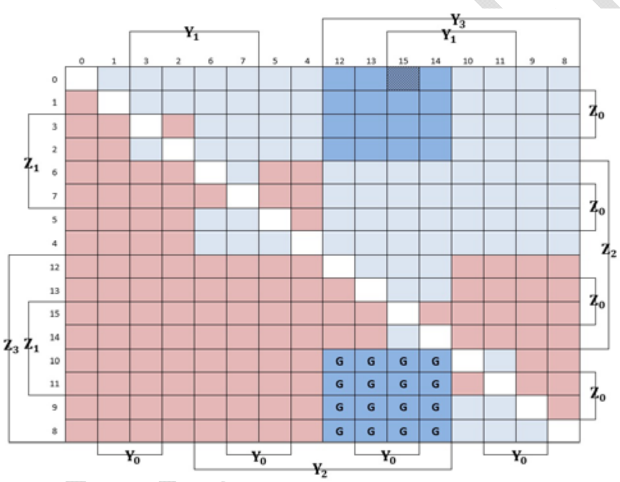


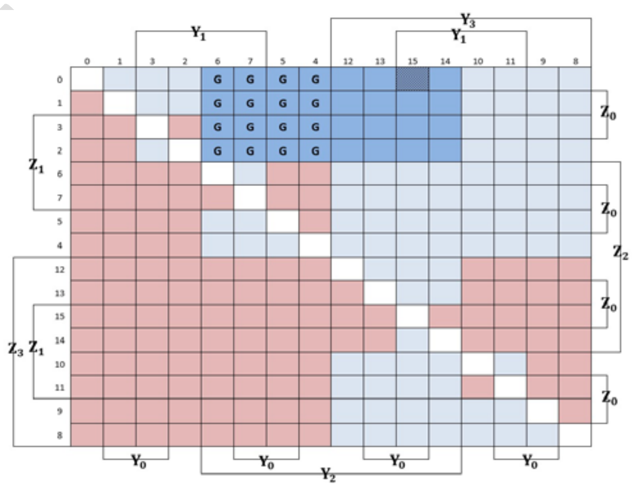
Fig. 7. Construction of the $2n$-variable map (for the n-bit comparator) from the $2(n-1)$-variable map (for the $(n-1)$-bit comparator). Theoretically, such a construction can be inductively continued without limit.



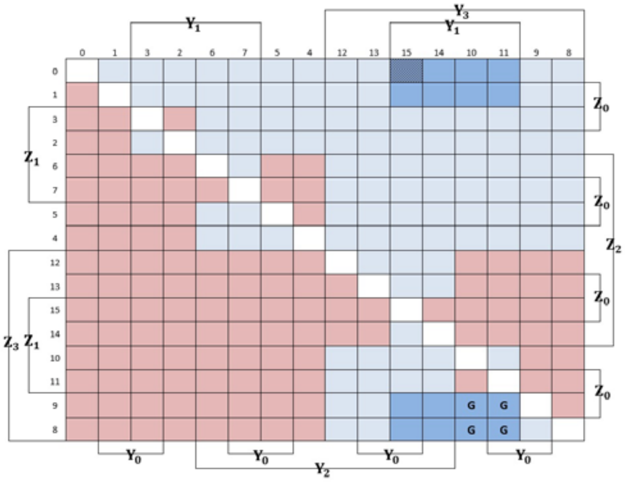
(a) $Y_3\bar{Z}_3$



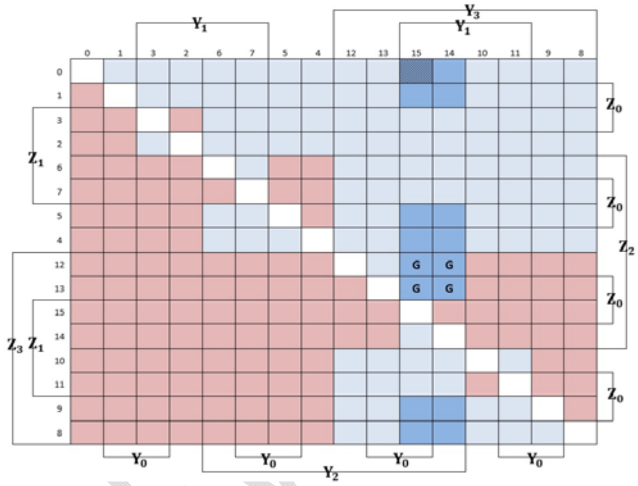
(b1) $Y_3Y_2\bar{Z}_2$



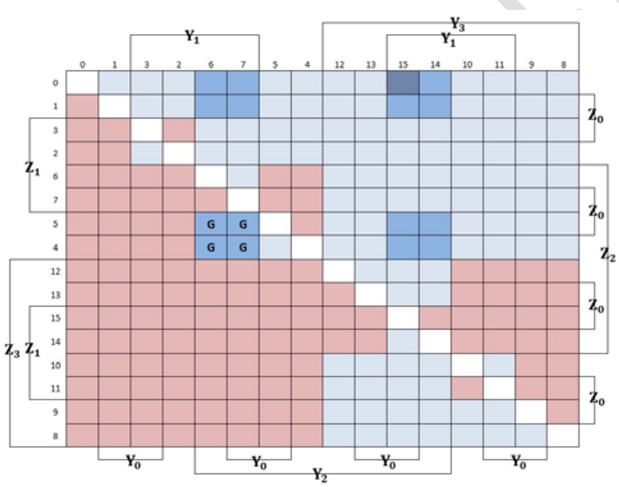
(b2) $\bar{Z}_3Y_2\bar{Z}_2$



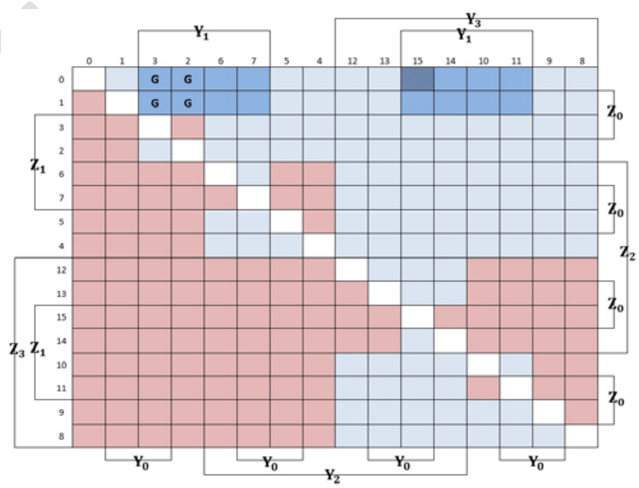
(c1) $\mathrm{Y}_3\bar{\mathrm{Z}}_2\mathrm{Y}_1\bar{\mathrm{Z}}_1$



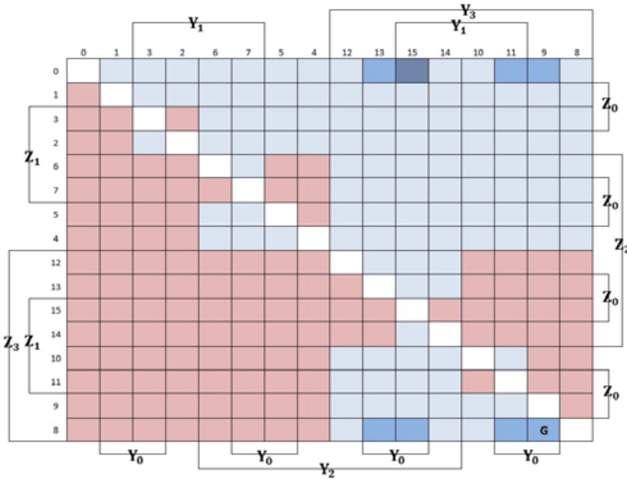
(c2) $\mathrm{Y}_3\mathrm{Y}_2\mathrm{Y}_1\bar{\mathrm{Z}}_1$



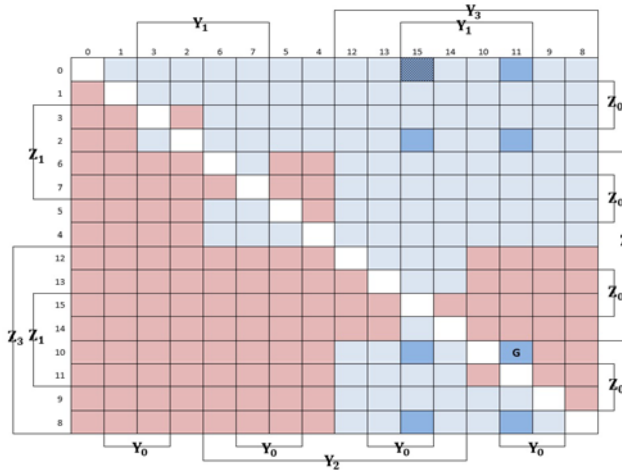
(c3) $\bar{\mathrm{Z}}_3\mathrm{Y}_2\mathrm{Y}_1\bar{\mathrm{Z}}_1$



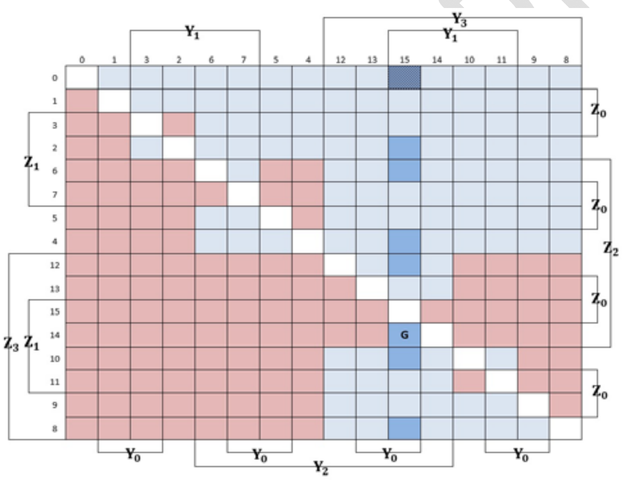
(c4) $\bar{\mathrm{Z}}_3\bar{\mathrm{Z}}_2\mathrm{Y}_1\bar{\mathrm{Z}}_1$



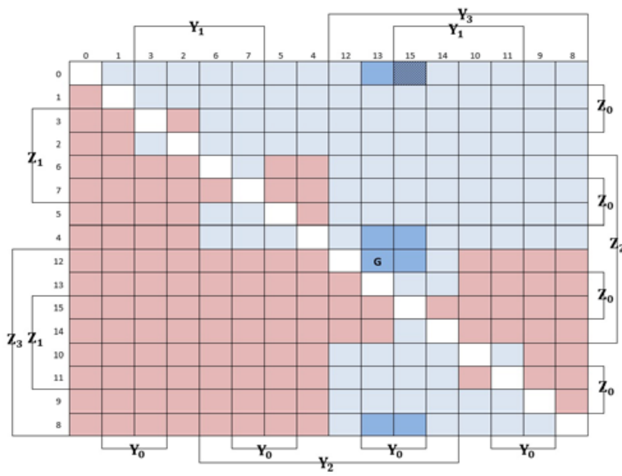
(d1) $Y_3\bar{Z}_2\bar{Z}_1Y_0\bar{Z}_0$



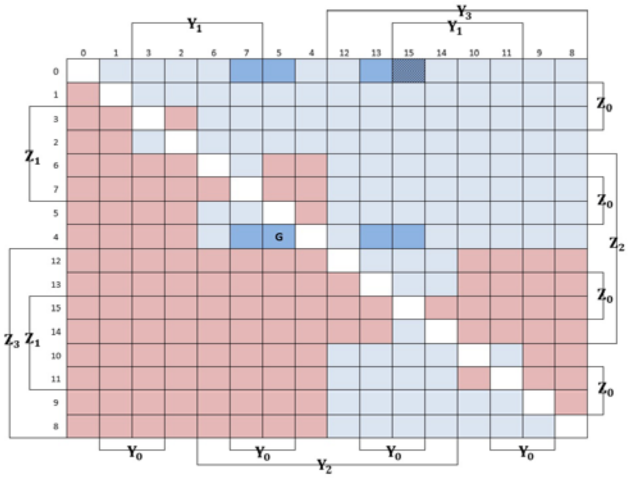
(d2) $Y_3\bar{Z}_2Y_1Y_0\bar{Z}_0$



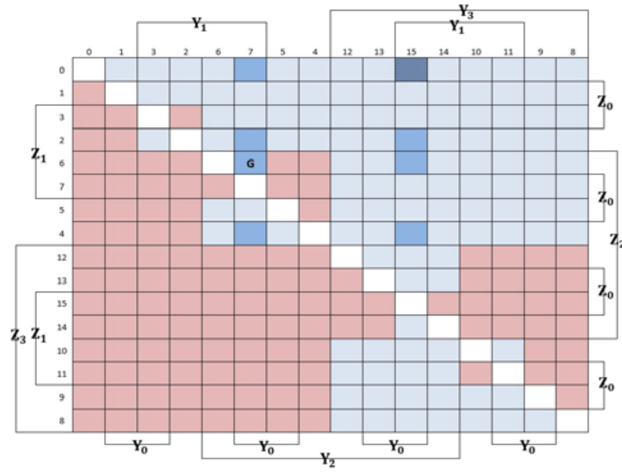
(d3) $Y_3Y_2Y_1Y_0\bar{Z}_0$



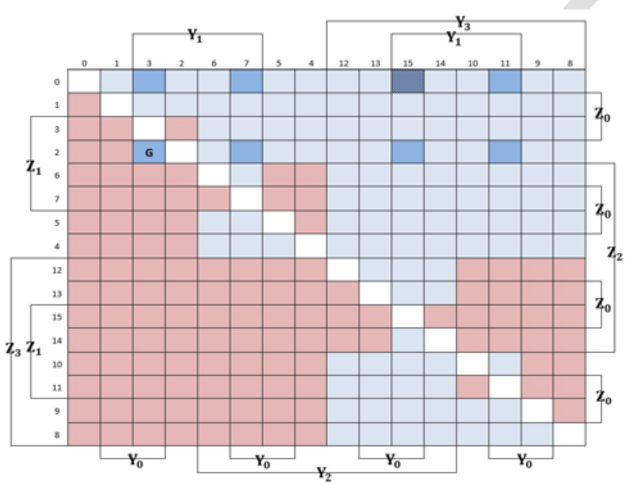
(d4) $Y_3Y_2\bar{Z}_1Y_0\bar{Z}_0$



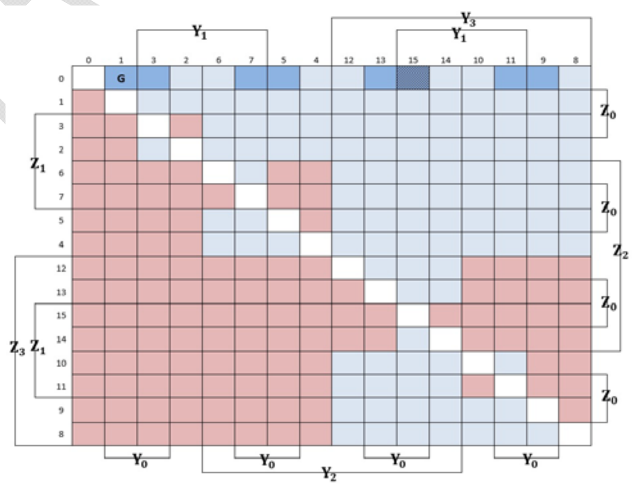
(d5) $\bar{Z}_3 Y_2 \bar{Z}_1 Y_0 \bar{Z}_0$



(d6) $\bar{Z}_3 Y_2 Y_1 Y_0 \bar{Z}_0$



(d7) $\bar{Z}_3 \bar{Z}_2 Y_1 Y_0 \bar{Z}_0$



(d8) $\bar{Z}_3 \bar{Z}_2 \bar{Z}_1 Y_0 \bar{Z}_0$

Fig. 8. A complete sum (and also a minimal sum) for G_4 given by loops on fifteen maps. Note that this coverage proves that G is a unate function (with a positive polarity in Y_k $(0 \leq k \leq 3)$ and a negative polarity in Z_k $(0 \leq k \leq 3)$. Note that all loops pass through the shaded cell $(Y_3Y_2Y_1Y_0Z_3Z_2Z_1Z_0) = (11110000)$, which is the all-1 cell for **Y** and the all-0 cell for **Z**. Each of the fifteen loops in this figure is an *essential* prime-implicant loop, since it is the only loop covering some of its cells. For example, the loop $\bar{Z}_3\bar{Z}_2\bar{Z}_1Y_0\bar{Z}_0$ in (d8) is the only PI loop covering

the cell $\overline{Y}_3\overline{Z}_3\overline{Y}_2\,\overline{Z}_2\overline{Y}_1\overline{Z}_1Y_0\overline{Z}_0$ (labelled with G). This cell has three asserted neighbors only, and if it could be covered by an 8-cell loop (which is the case herein), such a loop would be an essential PI loop.