4. "Information Processing 1962" "Proceedings of International Federation for Information Processing", nugget [IFIP 747 62]

615
371
618
619
1528
+
1530

SOME THEOREMS USEFUL IN THRESHOLD LOGIC FOR ENUMERATING BOOLEAN FUNCTIONS

BFs

E. GOTO

Massachusetts Institute of Technology, Cambridge, Mass., USA

H. TAKAHASI

University of Tokyo, Japan

1. Introduction

Among the various classification methods of Boolean switching functions, the method used in the Harvard Table [1]) is the most common. The Harvard method may be called PN classification, because functions which coincide with one another by permutation and negation of variables are classified into one type of class. In another commonly used method, the negation of functions, is introduced for defining equivalence relations within a class, and this may be called NPN classification.

In this paper, a new method to be called SD (self-dual) classification is presented. Two extra operations, self-dualization and anti-self-dualization, are introduced for defining equivalence relationships of functions within a SD class.

SD classification is specially suitable for threshold logic. For example, all switching functions belonging to the same SD class can be realized with essentially the same threshold logical circuit. SD classification in threshold logic is very similar to PN classification in relay logic. It is well known that all functions within a PN class are realizable with essentially the same contact network.

In SD classification, the number of different types of function is considerably reduced, since more operations are included than in PN or NPN classification. Therefore, SD classification is believed to be helpful for studying threshold logic, especially for tabulative or enumerative types of work.

As self-dualization is an operation which induces a self-dual Boolean function of $n + 1$ variables, from a non-self-dual function of n variables, the explicit rule of self-dualization will provide a convenient method for designing threshold logical circuits including self-dual functions.

The evaluation of the number $N(n)$ of linear input functions, i.e., the number of different Boolean functions of up to n variables, realizable with a single threshold device, is one of the interesting problems in threshold logic. For example, $N(n)$ will give a measure of the complexity of linear input functions.

Self-Dualization and SD Classification

Definition 1.

Given an arbitrary Boolean function $b(x_i)$ of n variables $1 \leqslant i \leqslant n$, then $b^d(x_i)$ associated with $b(x_i)$ by

$$b^d(x_i) = \overline{(b(\bar{x}_i)} \; *$$

is said to be the *dual* of $b(x_i)$. If $b^d(x_i) \equiv b(x_i)$, the function $b(x_i)$ is said to be *self-dual*.

The Boolean function $B(x_i)$ of $n + 1$ variables x_i, $0 \leqslant i \leqslant n$, associated with $b(x_i)$ of n variables x_i, $1 \leqslant i \leqslant n$ by

$$B = x_0 b(x_i) + x_0 b^d(x_i) \ldots \text{(B) (Boolean)}$$

is called the *self-dualized* of b, where (B) indicates a Boolean expression.

It is to be understood that the self-dualized of a self-dual function means the function itself. Actually, if

$$b = b^d, \text{ then } B = x_0 b + \bar{x}_0 b^d = (x_0 + \bar{x}_0) b = b \ldots \text{(B)}$$

The Boolean function $b(x_i)$ of n variables, associated with a self-dual Boolean function $B(x_i)$ of $n + 1$ variables, by the relation

$$b(x_1, \ldots x_n) = B(x_0 = 1, x_1 \ldots x_n)$$

is called the *anti-self-dualized* of B.

It is to be understood that the anti-self-dualized function of a non-self-dual function b means the function b itself.

Consistency of definition 1.

A Boolean function B defined as $B = x_0 b + \bar{x}_0 b^d$ is always self-dual. The anti-self-dualized function b of a self-dual function B is not a self-dual function, unless x_0 is an idle variable. Hereafter, it will be understood that "function of m variables" means that all variables are non-idle. Conversely, "up to m variables" means that some of the variables may be idle. The following are well known properties of dual and self-dual functions.

1. The dual of the dual is the original:

i.e. $$(b^d)^d = b. \qquad \text{(B)}$$

2. The dual is unique:

i.e. if $b = (b_1)^d$ and $b = (b_2)^d$, then $b_1 = b_2$. (B)

3. The dual of the function h of functions $b_1 \ldots b_m$,

* The bar denotes negation.

equals the dual function h^d of the dual functions $b_1{}^d \dots b_m{}^d$:

i.e. $\quad (h(b_1, b_2 \dots b_m))^d = h^d(b_1{}^d, b_2{}^d, \dots b_m{}^d).$ (B)

4. The self-dual function H of the self-dual functions $B_1 \dots B_m$, is a self-dual function:

i.e. $(H(B_1 \dots B_m))^d = H^d(B_1{}^d \dots B_m{}^d) = H(B_1 \dots B_m).$ (B)

Notice that some of the B_i's may be the variables themselves or their negatives, since x_i and $\bar{x}_i$ are both self-dual.

The self-dualized has properties similar to the above, provided that none of the variables is idle:

Theorem 1.,

1.1 The self-dualized of the anti-self-dualized of a self-dual function is the original, and the anti-self-dualized of the self-dualized of a non-self-dual function is also the original.

1.2 The correspondence between the self-dualized B and the anti-self-dualized b is one-to-one.

1.3 The self-dualized of a function h of functions $b_1 \dots b_m$ is equal to the self-dualized function H of the self-dualized functions $B_1 \dots B_m$.

Proofs. 1.1 and 1.2 are almost self-evident by definition. To prove 1.3, let the variables be x_i, $i = 1, 2 \dots n$, and let x_0 be the variable introduced for self-dualization. The function h, having m Boolean functions $b_1 \dots b_m$ as its arguments, can be regarded as a Boolean function h_x of the x_i's:

$$h_x(x_i) = h(b_1(x_i) \dots b_m(x)_i). \qquad \text{(B)}$$

By definition, the self-dualized H_x of h_x is given by

$$H_x(x_i; x_0) = x_0 h_x(x_i) + \bar{x}_0(h_x(x_i))^d. \text{ (B)}$$

On the other hand, the self-dualized function of the self-dualized functions H_x, (regarded as function of the x_i's and x_0) is given by

$$H_x'(x_i; x_0) = H(B_1(x_i; x_0) \dots B_m(x_i; x_0); x_0), \text{ (B)}$$

where H, $B_1 \dots B_m$ are respectively the self-dualized's of h, $b_1 \dots b_m$, i.e.,

$$H(y_1 \dots y_m; x_0) = x_0 h(y_1 \dots y_m) + x_0 h^d(y_1 \dots y_m),$$

and

$$B_j(x_i; x_0) = x_0 b_j(x_i) + \bar{x}_0 b_j{}^d(x_i),\ 1 \leqslant j \leqslant m.$$

Putting $x_0 = 1$ or $x_0 = 0$ and using property (3) of dual functions, we obtain

$$H_x'(x_i; x_0 = 1) = h(b_1 \dots b_m) = h_x = H_x(x_i; x_0 = 1)$$

and

$$H_x'(x_i; x_0 = 0) = h^d(b_1{}^d \dots b_m{}^d) = (h_x(x_i))^d = \\ = H_x(x_i; x_0 = 0), \text{ (B)}$$

and therefore

$$H_x' = H_x.$$

Definition 2.

A Boolean function b_2, is said to belong to the same SD class as a Boolean function b_1, if b_1 coincides with b_2 by self-dualization, anti-self-dualization, and negation of the functions, and permutation and negation of the variables. (The sequence and the number of applications of these operations are not important.) These five operations will be called the self-dual class operations.

Table 1. Classification and Number of Types of Switching Functions

	Number of Variables n	0	1	2	3	4	5	6
	$2^n n!$	1	2	8	48	384	3,840	46280
	2^{2^n}	2	4	16	256	65,536	4.3×10^9	1.8×10^{19}
371	General Functions *	2	2	10	218	64,594	4.3×10^9	1.8×10^{19}
615	Linear Input Fcns.*	2	2	8	72	1,536	86,080	14,487,040
618	PN Class GC *	2	1	3	16	380	1,227,756	4.0×10^{14}
619	PN Class LIF *	2	1	2	5	17	92	994
1528	NPN Class GF *	1	1	2	10	208	615,904	~
1530	NPN Class LIF *	1	1	1	3	9	48	504
	SD Class** GF *	(0) + 1	(1) + 0	(0) + 2	(2) + 4	(4) + 76	(76) + 109, 875	(109, 875) + ~
	SD Class** LIF *	(0) + 1	(1) + 0	(0) + 1	(0) + 1	(1) + 4	(4) + 14	(14) + 114

* All numbers are given for *exactly* n variables.

** In SD Class, the number in parentheses indicates the number of different types of self-dual functions of n variables, and the number without parenthesis indicates the number of different types of non-self-dual functions of n variables.

3. Self-Dual Transformation and Threshold Logic

A single threshold device may best be characterized mathematically by the non-linear unit step function $D(x)$. (cf. table 3).

Definition 3.

A Boolean function f of n variables x_i, $1 \leqslant i \leqslant n$, f and x_i taking either of the values 0 or 1, is said to be a *linear input function* if and only if there exists a set of real numbers W_i and T, to be called weight and threshold, such that

$$f = D(\sum W_i x_i - T). \quad \text{(A)} \tag{3.1}$$

Definition 3 is in agreement with the most commonly used definition of linear input functions [5-7,10]. The general synthesis problem in threshold logic may be characterized as: "Express a given Boolean switching function $b(x_i)$ in terms of k linear input functions f_r, $l \leqslant r \leqslant k$."

It can easily be shown that a combinatorial threshold logical circuit, with *no feed back loops* and having x_i's as inputs and $b(x_i)$ as the output, is expressible in the following functional form.

$$f_{rx} = f_r(x_i, f_{lx}, \ldots f_{lr-1lx})$$

and

$$b(x_i) = f_{kx} \tag{3.2}$$

where f_j is a linear input function of at most $n + r - 1$ arguments, and the suffix x as in f_{rx} means that f_r is regarded as a function of the x_i's. From the absence of feedback loops it should be possible to number the functions (or the threshold devices) in such a way that any f_j has only functions with smaller suffix numbers as its arguments. In minimization problems, we are asked to minimize the number $N_f (= k$ in (3.2)) of linear input functions. The minimum of N_f is a characteristic number of the given function $b(x_i)$, e.g., min $N_f = 1$ obviously means a linear input function. Substitution of (3.1) into (3.2) will give an algebraic expression of D functions which take the required values $b(x_i) = 0$ or 1 on all switching vertices of the input switching cube $x_i = 0$ or 1.

The minimum number of non-linear step functions N_D in an algebraic expression having the above property will also be a characteristic number of $b(x_i)$. Min N_D may be regarded as a measure of *non-linearity* of the Boolean function $b(x_i)$. One might think that $N_D \equiv N_f$, but this is not always the case. Function B_3 (part or full sum of three inputs) of table 3 gives a counter example. B_3 is obviously not a linear input function: min $N_f = 2$, but it can be expressed in terms of *one* non-linear function D. One of the two D functions in the second expression of table 3 is replaceable by an identity function $I(x) \equiv x$ which is obviously linear. In general therefore, $N_f = N_D + N_I$ will hold instead of $N_f = N_D$, where N_I is the number of I functions.

The following theorem indicates the importance of self-dual classification in threshold logic.

Theorem 2.

All Boolean functions belonging to the same self-dual class can be expressed in terms of the same number of linear input functions N_f, non-linear step functions N_D, and I functions N_I. Using definition 2, this may be rewritten as: All of the self-dual class operations preserve the numbers N_f, N_D and N_I.

Proof. From (3.1) and (3.2)

$$f_{rx} = D/I(A_r + C_r),\ b(x_i) = f_{kx},$$

$$A_r = \sum_{i=1}^{n} W_{ir} x_i + \sum_{j=1}^{r=1} \overline{W}_{jr} f_{jx} - T_r, \tag{3.3}$$

where C_r is a constant which may have to be added in case D is replaced by an I function. The theorem is proved by finding for each of the self-dual class operations, explicit transformation rules which are applicable to (3.3) without changing N_D and N_I (and

$$N_f = N_D + N_I).$$

1. Permutation: Rename the variables.
2. Negation of a variable, say x_1: Replace x_1 by $1 - x_1$ in all A_r's, i.e., replace T_r by $T_r - W_{1r}$ and W_{ir} by $- W_{ir}$.
3. Negation of a function: Change the sign of A_r in the case of a D function, i.e., change the sign of all W_{ik}'s, $\overline{W}_{jk}$'s and T_k. In the case of I, also replace C_r by $1 - C_r$.
4. Anti-Self-Dualization. In case $b(x_i)$ is self-dual, rename one of the x_i, say x_1, as x_0 and set it to 1, i.e. set x_1 to 1.
5. Self-Dualization: Let the largest negative value of each A_r on the switching vertices $x_i = 0, 1$ be $- M_r$. M_r should be positive and non-zero since there are only finite vertices [6].

$$W_{0r} = \sum_{i=1}^{n} W_{ir} + M_r - 2T_r + \sum_{j=1}^{r=1} \overline{W}_{jr}. \tag{3.4}$$

and

$$T_r^d = \sum_{i=1}^{n} W_{ir} + \sum_{j=1}^{n} \overline{W}_{jr} + M_r - T_r. \tag{3.5}$$

Replace T_r by T_r^d and add a term $W_0 x_0$ in A_r. In case of I, put $M = 1$ and do the same.

While transformation rules (1) to (4) are quite obvious, (5) may need extra explanation. It has been shown [6]) that the dual of any linear input function can be obtained by changing only the threshold. In each f_r, change of T_r to T_r^d gives the dual f_r^d. Since the change of variable x_0 from 1 to 0 switches all linear input functions f_r's into f_r^d's, $b(x_i)$ is also switched into its dual $b^d(x_i)$. By definition this means self-dualization.

As a special case of theorem 2 in which

$$N_f = N_D + N_I = 1$$

we obtain: Functions belonging to the same self-dual class are either all linear input or all not linear input functions.

The following examples will indicate the usefulness of the explicit self-dualizing rules in deriving some practical circuits.

Example 1. Self-dualization of a parity check circuit of $2m$ inputs gives a parity check circuit of $2m + 1$ inputs.

Example 2. Self-dualization of a combinatorial binary counter circuit gives a reversible counter. Suppose we have a combinatorial circuit of $n+1$ inputs a and $x_1, x_2 \ldots x_n$, and of n outputs $y_1 \ldots y_n$, such that $Y = X + a \pmod{2^n}$, where

$$X = \sum_1^n 2^{i-1} x_i \text{ and } Y = \sum_1^n 2^{i-1} y_i.$$

By self-dualizing the circuit and putting $\bar{x}_0 = s$, we obtain a circuit such that $Y = X + a - s \pmod{2^n}$. This circuit not only counts in both additive (a) and subtractive (s) modes but also operates correctly even if a and s are applied simultaneously, i.e., $Y = X$ if $a = s = 1$.

4. A Lower Bound of $N(n)$

Theorem 3.

$N(n)$, the number of Boolean functions of up to n variables realizable with a single threshold element, is larger than $2^{0.25 n^2}$.

Proof. Let f be a linear input Boolean function of $2m$ variables $x_1 \ldots x_{2m}$, with integral weights W_i, W_j such that $W_i = 2^{i-1}$ for $1 \leqslant i \leqslant m$ and $1 \leqslant W_j \leqslant 2^m$ for $m+1 \leqslant j \leqslant 2m$ and with threshold $T = 2^m$, i.e.

$$f = D\left(\sum_1^m W_i x_i + \sum_{m+1}^{2m} W_j x_j - 2^m\right).$$

Suppose the two set of weights $W_j^{(1)}$ and $W_j^{(2)}$ differ at $j = k$, so that $W_k^{(1)} > W_k^{(2)} \geqslant 1$. Setting $x_k = 1$, $x_j = 0$ for $j \neq k$ and

$$2^m - 2 \geqslant \sum_1^m W_i x_i = 2^m - W_k^{(1)} > 0$$

(obviously giving unique values to x_i's) results in $f^{(1)} = 1$ but $f^{(2)} = 0$. Hence, different sets of W_j, $(2^m)^m$ in number, give different linear input functions. All the x_j are non-idle variables, since putting all $x_i = 1$ and all $x_j = 0$ gives $f = 0$, but all $x_i = 1$ and at least one $x_j = 1$, gives $f = 1$. Since, all possible negations of non-idle variables of a linear input function give different linear input functions, we obtain when $n = 2m$:

$$N(n) > 2^m (2^m)^m \geqslant 2^{0.25 n^2} \text{ for } m \geqslant 1.$$

When $n = 2m - 1$, putting $W_{2m} = 0$ instead of $1 \leqslant W_{2m} \leqslant 2^m$ and using the same argument as above, we obtain:

$$N(n) > 2^{m-1}(2^m)^m = 2^{(n^2+4n+1)/4} > 2^{0.25 n^2}$$

for $n \geqslant 3$. These two cases and $N(0) = 2$, $N(1) = 4$ prove the theorem. This lower bound is much larger than that given by Muroga [6]). The upper bound $U(n)$ of $N(n)$ given by Willis and Winder [8,9]),

$$U(n) = 2 \sum_{i=0}^{n} (2^n - 1)C^i,$$

behaves asymptotically like $2^{n^2}/n!$ for large n, and also satisfies

$$\lim_{n \to \infty} ((\log_2 U(n))/n^2) = 1.$$

The most interesting feature of the new lower bound is its similarity to $U(n)$ in its functional form. This leads to a conjecture that $N(n)$ would behave like

$$\lim_{n \to \infty} (\log_2 N(n))/n^2 = k \text{ or } N(n) \approx 2^{kn^2},$$

with k being a certain constant between 1/4 and 1.

It may be worth noting that, although 2^{kn^2} is much smaller than the total number 2^{2^n} of Boolean functions of up to n variables, it is much larger than $2^n n!$ which is the number of all possible ways of negating and permuting the n variables.

Since

$$\lim_{n \to \infty} (\log_2 2^n n!)/n^2 = 0,$$

negation and permutation of variables do not have any significance in any kind of argument which leads to bounds on k, or which would lead to the determination of the value of k.

5. References

1) Staff of the Harvard Computation Laboratory: "Synthesis of Electronic Computing Circuits," Harvard Univ. Press, 1951.
2) Slepian, D.: *On the Number of Symmetry Types of Boolean Functions of n Variables*, Can. J. Math., **5** (1953) 185.
3) Elspas, B.: *Self-Complementary Symmetry Types of Boolean Functions*, Trans. IRE. EC-9 (1960) 264.
4) Toda, I.: *On the Number of the Types of Self-Dual Logical Functions*, J. Inf. Processing Soc. Japan, **2** (1961) 21. (In Japanese, to be published in English.)
5) McNaughton, R.: *Unate Truth Functions*, Trans. IRE. EC-**10**, 1961.
6) Muroga, S., I. Toda and S. Takasu: *Theory of Majority Decision Elements*, J. Franklin Inst., **271**, (1961) 376.
7) Winder, R. O.: *Single Stage Threshold Logic*, AIEE Conference Paper, October 1960.
8) Winder, R. O.: *More about Threshold Logic*, AIEE Conference Paper, October 1961.
9) Willis, G.: Personal Communication. He showed the same upper bound $U(n)$ as Winder in 1959.
10) Minnick, R. C.: *Linear Input Logic*, Trans IRE, EC-**10**, (1961) 6.
11) Goto, E.: "Seminar Notes on Threshold Logic." The material of this paper has been covered at seminars given at M.I.T. during the period November to December 1961.

ABSTRACTS

A new method of classifying Boolean functions, called *self-dual* classification, and specially suited for threshold logic, is presented. Besides permutation and negation of variables, two other operations, *self-dualization* and *anti-self-dualization*, are introduced to define the equivalence of functions within a class. These operations preserve the number of non-linear threshold elements in combinatorial switching circuits. The self-dual classification considerably reduces the number of different types of switching function in threshold logic, e.g., for up to 4 variables, 83 instead of the 402 in the conventional classification.

Lower and upper bounds of the number $N(n)$ of linear input functions, i.e., functions realizable with a single threshold element, of up to n variables are given. Upper bounds show that $N(n)$ is smaller than 2^{n^2} and a lower bound shows that $N(n)$ is larger than $2^{0.25 n^2}$. From these bounds it is conjectured that for large n, $N(n)$ would behave asymptotically like 2^{kn^2} where k is a certain constant between ¼ and 1.