A Parametric Based Technique for Detection of Total and Multiform Symmetric Switching Functions in Logic Synthesis

Subhajit Guha
Department of Computer Science
BarrackporeRastraguruSurendranath College
West Bengal, INDIA

Kritanta Saha
Department of Computer Science
BarrackporeRastraguruSurendranath College
West Bengal, INDIA

Nilkantha Sen
Department of Computer Science
BarrackporeRastraguruSurendranath College
West Bengal, INDIA

ABSTRACT
An innovative as well as illuminating approach for detection of total and multiform symmetric switching functions is proposed. This method is based on modulo-2 sum between existent parameters rather than using maps, charts or large tables. The invariant properties are being revealed by the set of true minterms in accordance with the logical construction of existent parameters resulting in the reduction of complexity in time-space domain.

General Terms
Total Symmetric Function, Multiform Symmetric Functions and Invariant.

Keywords
Invariant set, Reduced Invariant Set, Existent Parameter and Displacement.

1. INTRODUCTION
The symmetric functions are dyed-in-the-wool to proclaim itself as an important class of switching functions for logic synthesis. Detection of a symmetric function is a fascinating problem, because a function can demonstrate symmetry in all possible polarities of its input variables. To be more precise, to detect the symmetry of an n variable switching function it is necessary to explore 2^n possible literals of the input variable's permutation. The dire necessity is to know when a function is totally or partially symmetric, as there are special economical synthesis procedures for integrated circuits that realize such functions. Several methods exist to detect symmetry in switching functions. All of these existing methods employ maps, truth tables, charts, matrices etc. The document presents here is based on modulo-2 sum between existent parameters. The method discussed here, is relevant on reduction of search space for total symmetry and multiform symmetry detection.

2. LITERATURE REVIEW
In 1938, a convenient way of writing a symmetric function is explored from the theorem of Shannon [1], in his paper "A symbolic analysis of relay and switching circuits". Firstly, S.H.Caldwell [3] in 1954 has described a method for detecting total symmetry of switching functions with the help of extended Karnaugh maps, though it is difficult for functions of larger than four variables. Furthermore, the identification method proposed by Marcus [6], McCluskey[5] in 1956, are more well-planned and can be applied to large number of variables. In 1963, A. Mukhopadhyay [11] formulated a technique for detecting symmetry using a chart based method; popularly it is known as decomposition chart. Apart from this method in 1971, S.R. Das and C.L. Sheng [13] applied an approach known as residue test by numerical method to detect symmetry. Tsai et al. [15] suggested generalized Reed-Muller forms for detecting groups of symmetric literals in completely specified switching functions, in 1996. Kravets and Sakallah [16], in 2000 and further J.S. Zhang et al. [19] in 2006; used circuit based method that makes uses of structural analysis, integrated simulation and Boolean SAT for fast and scalable detection of symmetries switching function. Peter M Maurer [22] introduced a method for detecting Total Symmetry by using hyper linear structure. Generalized symmetries in binary decision diagram (BDD)are found by Kettle et al. [20] in the year 2008. Z.Kohavi [14] advocated an approach for finding symmetric switching functions based on invariant under any permutation of literals. In 2012, S. Guha et al. [21] modified the method of Z.Kohavi by means the true minterms are bi-partitioned into two subsets depending upon the polarity of a particular variable to reduce the search space.

3. PRELIMINARIES
Definition 1 A switching function F of n variables $\{x_{n-1}, x_{n-2},....., x_0\}$is said to be totally symmetric if and only if any permutation of these n variables leaves the function invariant. Therefore, any permutation of variables may be obtained by successively interchanging only two variables at one time.

Example: $F(x_2,x_1,x_0) = \bar{x}_2\bar{x}_1x_0+x_2\bar{x}_1\bar{x}_0+\bar{x}_2x_1\bar{x}_0$ is totally symmetric because any permutation of these 3 variables leaves the function invariant.

Conjecture 1 By definition of total symmetry of switching functions, it can be concluded that if $x_i \sim x_j$(x_i invariant x_j, i.e., x_i and x_j can be interchanged without altering the outcome of the function) $\forall$ i=0 to n-1,$\forall$ j=0 to n-1and $i\neq j$,(i.e., for all permutation of x_i and x_j); then the function is totally symmetric.

Definition 2 A switching function of n variables$\{x_{n-1}, x_{n-2},\ldots.., x_0\}$ may be symmetric in a subset of m variables,$2 \leq m \leq n$, in many different forms. A function F of n variables exhibiting symmetry in a subset of m variables in all possible forms is said to be multiform symmetric in those m variables of the subset [13].

For example, $F(x_1,x_0) = x_1\bar{x}_0+\bar{x}_1x_0$ is multiform symmetric in $\{x_1,x_0\}\{\bar{x}_1,\bar{x}_0\}and\{x_1,\bar{x}_0\}\{\bar{x}_1,x_0\}$.

In general a symmetric function consists of mixed variables of symmetry if some of the variables are unprimed and rests of them are primed. The variables of symmetry are said to be multiform in nature when the function exhibits multiform symmetry in these variables.

Theorem 1 The number of total symmetric functions of n variables is $2^{(n+1)}$ where n is the number of literals.

Definition 3 Invariant Set (R) includes all possible invariantsof literals such that, $x_i \sim x_j$;∀ i=0 to n-1,∀ j=0 to n-1andi ≠ j

belong to R.

Definition 4 Reduced Invariant Set (R') includes $x_i \sim x_j$, where i=0 to n-1, j=(i+1) mod n and $x_0 \sim x_{n-1}$, where i=0, which is a subset of R. If all the invariants of R' holds for a given function then all other invariants which does not belong to R'but belongs to R is also stay the course by using transitive rule.

Definition 5 The i^{th}Existent parameter (E^i_{pq}) is a decimal number, it corresponds to a binary number having k bits where $k = 2^{n-2}$and each bit in the binary number denotes the existence of a particular term (either minterm or maxterm) of n literals. If the given switching function consists of those terms then the corresponding bits of E^i is set to 1 and 0 otherwise; where *p* and *q* are the values (either 0 or 1) of i^{th} bit and $((i+1) \bmod n)^{th}$ bit of those terms whose existence are denoted by existent parameters and i=0 to n-1.

The Existence of minterms which contains 0 at both i^{th} bit and $((i+1) \bmod n)^{th}$ bit are denoted by E^i_{00}.

The Existence of minterms which contains 1 at i^{th} bit and 0 at $((i+1) \bmod n)^{th}$ bit are denoted by E^i_{01}.

The Existence of minterms which contains 0 at i^{th} bit and 1 at $((i+1) \bmod n)^{th}$bit are denoted by E^i_{10}.

The Existence of minterms which contains 1 at both i^{th} bit and $((i+1) \bmod n)^{th}$bit are denoted by E^i_{11}.

Initially $E^i_{00}= E^i_{01}= E^i_{10}= E^i_{11}= 0$.

4. OBJECTIVE

The overall work can be conveniently studied into two sections. Firstly, to develop a new method for detection of totally and Multiform symmetric functions that is efficient than the existing methods in time-space domain. Secondly, to enumerate that the number of total symmetric switching function of n variables is $2^{(n+1)}$.

5. PROPOSED METHOD

In order to detect total symmetry and multiform symmetry, it is not necessary to test for all the invariants in the Invariant Set *R*; it suffices that if all the invariants in the Reduced invariant set R'holds for the given function. Therefore, the existent parameters are to be constructed first.

5.1 Construction of Existent parameters

At first the mintermshas to be found∀ i=0 to (n-1) whose existence in the given function are denoted by the MSBs of existent parametersE^i_{00}, E^i_{01}, E^i_{10} and E^i_{11}. The minterms can be found by using the following four theorems for different existent parameters.

Theorem 2 The MSB of E^i_{00} denotes the existence of the minterm, $M^i_{00_{k-1}}$having i^{th} bit 0 and $((i+1) \bmod n)^{th}$bit 0 and all the remaining bits are 0, i.e., $M^i_{00_{k-1}}=0$.

Proof Let the minterm $M^i_{00_{k-1}} = \{ x_{n-1} \ldots x_{i+1} x_i \ldots x_1 x_0 \}$; by definition of E^i_{00} each bit denotes the existence of those minterms contain 0 at both i^{th} bit and $((i+1) \bmod n)^{th}$bit, ∀ i=0 to n-1. MSB of E^i_{00}denote the minterm that have all bits 0; i.e., $\{x_{n-1} \ldots\ldots x_{i+1} x_i \ldots. x_1 x_0\} = \{0\ldots00\ldots00\}$.

Therefore, the minterm

$$M^i_{00_{k-1}} = x_{n-1}*2^{n-1}+\ldots+x_{i+1}*2^{i+1}+ x_i*2^i+\ldots+ x_0*2^0$$

$$= 0*2^{n-1}+\ldots+ 0*2^{i+1}+ 0*2^i +\ldots+ 0*2^0= 0$$

Theorem 3 The MSB of E^i_{01} denotes the existence of the minterm, $M^i_{01_{k-1}}$ having i^{th}bit 1 and $((i+1) \bmod n)^{th}$ bit 0 and all the remaining bits are 0, i.e.,$M^i_{01_{k-1}} = 2^i$.

Proof Let the minterm $M^i_{01_{k-1}} = \{ x_{n-1} \ldots x_{i+1} x_i .. x_1 x_0 \}$; by definition of E^i_{01}each bit denotes the existence of those minterms contains 1 at i^{th} bit and 0 at $((i+1) \bmod n)^{th}$bit, ∀ i = 0 to n-1. MSB of E^i_{01} denotes the minterm that have$x_i = 1$ and all other bits are 0; i.e.,$\{x_{n-1} x_{n-2} \ldots x_{i+2}\} = \{00\ldots0\}$ and $\{x_{i-1} .. x_1 x_0\} = \{0\ldots00\}$. Therefore, the minterm

$$M^i_{01_{k-1}} = x_{n-1}*2^{n-1}+\ldots+x_{i+1}*2^{i+1}+x_i*2^i+ \ldots+x_0*2^0$$

$$= 0*2^{n-1}+\ldots+ 0*2^{i+1}+ 1*2^i+\ldots+ 0*2^0$$

$$= 2^i$$

Theorem 4 The MSB of E^i_{10} denotes the existence of the minterm, $M^i_{10_{k-1}}$having i^{th} bit 0 and $((i+1) \bmod n)^{th}$bit 1 and all the remaining bits are 0, i.e., $M^i_{10_{k-1}}=2^{(i+1) \bmod n}$.

Proof Let the minterm $M^i_{10_{k-1}} = \{ x_{n-1} \ldots x_{i+1} x_i .. x_1 x_0 \}$; by definition of E^i_{10}each bit denotes the existence of those minterms contains 0 at i^{th} bit and 1 at $((i+1) \bmod n)^{th}$bit, ∀ i= 0 to n-1. MSB of E^i_{10}denotes the minterm that have$x_{(i+1) \bmod n}= 1$ and all other bits are 0; $\{x_{n-1} x_{n-2} \ldots x_{i+2}\} = \{00\ldots\ldots0\}$and $\{x_i .. x_1 x_0\} = \{0...00\}$.

So the minterm

$$M^i_{10_{k-1}} = x_{n-1}*2^{n-1}+\ldots+x_{i+1}*2^{i+1}+x_i*2^i+ \ldots+ x_0*2^0$$

$$= 0*2^{n-1}+\ldots+ 1*2^{i+1}+ 0*2^i+\ldots+0*2^0$$

$$= 2^{i+1}$$

As i= 0 to n-1 so when i= n-1, (i+1)= n but the minterm does not have n^{th} bit,i.e., for i= n-1 , (i+1) may exceed the range. ((i+1)mod n) keep it in range. Therefore, the minterm is $M^i_{10_{k-1}} = 2^{(i+1) \bmod n}$

Theorem 5 The MSB of E^i_{11} denotes the existence of the minterm, $M^i_{11_{k-1}}$having i^{th} bit 1,$((i+1) \bmod n)^{th}$bit 1 and all the remaining bits are 0, i.e., $M^i_{11_{k-1}} = 2^{(i+1) \bmod n} + 2^i$.

Proof Let the minterm $M^i_{11_{k-1}} = \{x_{n-1} \ldots x_{i+1} x_i .. x_1 x_0\}$; by definition of E^i_{11} each bit denotes the existence of those minterms contains 1 at i^{th} bit and 1 at $((i+1) \bmod n)^{th}$ bit, $\forall$ i=0 to n-1. MSB of E^i_{11} denotes the minterm that have $x_{(i+1) \bmod n}=1$, $x_i=1$ and all other bits are 0; $\{x_{n-1}x_{n-2} \ldots x_{i+2}\} = \{00 \ldots 0\}$ and $\{x_{i-1} \ldots x_1 x_0\} = \{0..00\}$.

So the minterm

$$M^i_{11_{k-1}} = x_{n-1}*2^{n-1}+\ldots+x_{i+1}*2^{i+1}+x_i*2^i+\ldots+x_0*2^0$$

$$= 0*2^{n-1}+\ldots+1*2^{i+1}+1*2^i+\ldots+0*2^0$$

$$= 2^{i+1}+2^i$$

As i = 0 to n-1, when i = n-1,(i+1) = n but the minterm does not have n^{th} bit, i.e., for i = n-1,(i+1) may exceed the range. ((i+1) mod n) keep it in range. Hence, the minterm is $M^i_{11_{k-1}} = 2^{(i+1) \bmod n}+2^i$.

Once the minterms are found, the next set of minterms whose existence in the given function is denoted by the remaining bits of the existent parameters can be generated by using the following theorems.

Theorem 6 The minterms that are denoted by the remaining bits of E^i_{00}, E^i_{01}, E^i_{10} and E^i_{11} are generated by using $M^i_{00_{k-1}}$, $M^i_{01_{k-1}}$, $M^i_{10_{k-1}}$ and $M^i_{11_{k-1}}$ respectively. The i^{th} bit and $((i+1) \bmod n)^{th}$ bit of the said minterm, i.e., $\{x_{i+1}x_i\}$ divides the minterms into partitions, i.e., $\{x_{n-1}x_{n-2} \ldots x_{i+2}\}$ and $\{x_{i-1} \ldots x_1 x_0\}$; where $\{x_{n-1}x_{n-2} \ldots x_{i+2}\}$ contains $((n-1)-(i+2)+1)=(n-i-2)$ number of bits, that consists 2^{n-i-2} combinations of literals. For each of this combinations $\{x_{i-1} \ldots x_1 x_0\}$ can have 2^i possible combinations of literals.

Case (a) For each combination of literals in $\{x_{n-1}x_{n-2} \ldots x_{i+2}\}$ the next 2^i minterms are generated by adding 1 with the previously generated minterm; i.e.,

$M^i_{00_{k-(d+1)}} = M^i_{00_{k-d}}+1$; $M^i_{01_{k-(d+1)}} = M^i_{01_{k-d}}+1$;

$M^i_{10_{k-(d+1)}} = M^i_{10_{k-d}}+1$; $M^i_{11_{k-(d+1)}} = M^i_{11_{k-d}}+1$;

Where d=1 to k-1

Case (b) To get the next combination of literals in $\{x_{n-1}x_{n-2} \ldots x_{i+2}\}$ add displacement D^i with the minterm that have the present combination of literals in $\{x_{n-1}x_{n-2} \ldots x_{i+2}\}$ and $\{x_{i-1} \ldots x_1 x_0\} = \{0 \ldots 00\}$, where $D^i = 2^{(i+2) \bmod n}$.

Proof By previous theorems (Theorem 2, Theorem 3, Theorem 4 and Theorem 5) it is already found that $M^i_{00_{k-1}} = 0$, $M^i_{01_{k-1}} = 2^i$, $M^i_{10_{k-1}} = 2^{(i+1) \bmod n}$ and $M^i_{11_{k-1}} = 2^{(i+1) \bmod n}+2^i$ respectively. In each of the minterms $\{x_{n-1}x_{n-2} \ldots x_{i+2}\} = \{00 \ldots 0\}$ and $\{x_{i-1} \ldots x_1 x_0\} = \{0 \ldots 00\}$. $\{x_{i-1} \ldots x_1 x_0\}$ has 2^i possible combinations. The next combinations of literals in $\{x_{i-1} \ldots x_1 x_0\}$ are {0…01}, {0…10}, {0…11}, …,{1…00}, …, {1…10}, {1…11}. It is observed that the above combinations are done by adding 1 with the previous one. Hence, the following minterms must be found by adding 1 with the previous minterms; i.e.,

$M^i_{00_{k-(d+1)}} = M^i_{00_{k-d}}+1$; $M^i_{01_{k-(d+1)}} = M^i_{01_{k-d}}+1$;

$M^i_{10_{k-(d+1)}} = M^i_{10_{k-d}}+1$; $M^i_{11_{k-(d+1)}} = M^i_{11_{k-d}}+1$;

Hence, case (a) is proved.

When the 2^i possible combinations of literals in $\{x_{i-1} \ldots x_1 x_0\}$ are considered, the next combination in $\{x_{n-1}x_{n-2} \ldots x_{i+2}\}$ will be taken. To achieve the next combination of literals in $\{x_{n-1}x_{n-2} \ldots x_{i+2}\}$, the value of $\{x_{n-1}x_{n-2} \ldots x_{i+2}\}$ incremented by 1. This can be done by adding $2^{(i+2) \bmod n}$ with the minterm

$M^i_{00_{k-1}} = \{x_{n-1}x_{n-2} \ldots x_{i+1}x_i \ldots x_1 x_0\}$,

where $\{x_{i-1} \ldots x_1 x_0\} = \{0 \ldots 00\}$,

$M^i_{01_{k-1}} = \{x_{n-1}x_{n-2} \ldots x_{i+1}x_i \ldots x_1 x_0\}$,

where $\{x_{i-1} \ldots x_1 x_0\} = \{0 \ldots 00\}$,

$M^i_{10_{k-1}} = \{x_{n-1}x_{n-2} \ldots x_{i+1}x_i \ldots x_1 x_0\}$,

where $\{x_{i-1} \ldots x_1 x_0\} = \{0 \ldots 00\}$,

$M^i_{11_{k-1}} = \{x_{n-1}x_{n-2} \ldots x_{i+1}x_i \ldots x_1 x_0\}$,

where $\{x_{i-1} \ldots x_1 x_0\} = \{0 \ldots 00\}$.

The $2^{(i+2) \bmod n}$ is considered as displacement and denoted by D^i. For i= n-1, or n-2,(i+2)> n-1. Therefore, it exceeds its range and (i+2) mod n makes it in range, i.e., $(n-1) \leq (i+2) \bmod n \leq 0$. Hence Case (b) is satisfied.

Whenever a minterm is generated by using previous theorems; existence of those minterms in the given function has to be checked and if the generated minterm belong to the given function then the corresponding bit of the existent parameter for which the minterm is generated is set to 1. Otherwise, the bit will be set to 0. This can be done by using the following theorem.

Theorem 7 Each bit of E^i_{00}, E^i_{01}, E^i_{10} and E^i_{11} can be assigned a value (either 0 or 1) as, if $M^i_{pq_{k-d}} \in F$ then $E^i_{pq} = (E^i_{pq}$ Left shift by 1) OR 1 otherwise $E^i_{pq} = (E^i_{pq}$ Left shift by 1) OR 0 where d=1 to n-1, F is the Given Switching function and $M^i_{pq_{k-d}}$ are the minterms denoted by $(k-d)^{th}$ bit of E^i_{pq}.

Proof By definition of the Existent parameters initially $E^i_{00} = E^i_{01} = E^i_{10} = E^i_{11} = 0$. The bits of E^i_{00}, E^i_{01}, E^i_{10} and E^i_{11} are generated from MSB to LSB. The minterm $M^i_{00_{k-d}}$ denoted by $(k-d)^{th}$ bit of E^i_{00} is first generated and if this minterm belongs to the function F then $(k-d)^{th}$ bit has to be set 1 otherwise 0. Therefore, if minterm belongs to the function perform $E^i_{00} = (E^i_{00}$ Left shift by 1) OR 1, otherwise $E^i_{00} = (E^i_{00}$ Left shift by 1) OR 0; which set or reset the LSB of E^i_{00}. After that $(k-d)$ bits is to be set or reset. So the Left shift operation on E^i_0 is performed $(k-d)$ times that eventually moves the LSB of E^i_{00} to (k-d)thbit. By using the Left shift and bitwise OR operator the value of $b^i_{l_{k-d}}$ can be assigned. Similarly each bit of E^i_{01}, $E^i_{10}$and E^i_{11} can be assigned a value using Left Shift and bitwise OR operator. Hence the theorem follows.

Use of Theorem 7 eliminates the need of handling the bits of existent parameters which may be time consuming. After the construction of existent parameters E^i_{00}, E^i_{01}, E^i_{10}, and E^i_{11}

1. $x_i \sim x_{(i+1) \bmod n}$ and $\overline{x}_i \sim \overline{x}_{(i+1) \bmod n}$

if $E^i_{01} \oplus E^i_{10} = 0$ or $E^i_{01} = E^i_{10} \forall$ i=0 to n-1.

2. $x_i \sim \overline{x}_{(i+1) \bmod n}$and $\overline{x}_i \sim x_{(i+1) \bmod n}$

if $E^i_{00} \oplus E^i_{11} = 0$or $E^i_{00} = E^i_{11} \forall$ i= 0 to n-1.

5.2 Detection of Total Symmetry

In order to detect total symmetry it is necessary to check whether $E^i_{01} \oplus E^i_{10} = 0, \forall$ i= 0 to n-1. As all the invariants in the reduced invariant set R' holds, where all $x_i$and $x_{(i+1) \bmod n}$are all primed or all unprimed. If $E^i_{01} \oplus E^i_{10} \neq 0$ for a particular value of i, then the function is not totally symmetric.Hence for detection of total symmetry only$E^i_{01}$and E^i_{10} need to be constructed. For any i, if the condition, i.e., $E^i_{01} \oplus E^i_{10} = 0$ does not hold for the given function, then the function is not totally symmetric.

Proposed Algorithm for Detection of Total Symmetric Functions

Input: A Switching Function of n variables

Output: Total Symmetry or Not

1. For i=0 to n-1 do

A. Call Construct_ E^i_{01} and E^i_{10} ;

B. If $E^i_{01} \oplus E^i_{10} \neq 0$ then

i. Print "the function is not Totally Symmetry";

ii. Exit;

C. End if

2. End for

3. Print "the function is Totally Symmetry";

4. End

ProcedureConstruct_ E^i_{01}andE^i_{10}

Input: number of literals in the given function, i.e.,n

Output: The Existent parameter E^i_{01} and E^i_{10}

1. Set $k=2^{n-2}$;

2. Set $M^i_{01_{k-1}}=2^i$ and $M^i_{10_{k-1}}=2^{(i+1) \bmod n}$;

3. Set $prvMin_{01}=M^i_{01_{k-1}}$ andnxtMin_{01}=prvMin_{01};

4. Set $prvMin_{10}=M^i_{10_{k-1}}$ andnxtMin_{10}=prvMin_{10};

5. For every pass=1 to ($k/2^{i \bmod (n-1)}$)

A. For every p=1 to $2^{i \bmod (n-1)}$

i. Ifnxt$Min_{01} \in$ F then

$E^i_{01}=(E^i_{01}$ Left shift by 1) OR 1;

ii. Else

$E^i_{01}=(E^i_{01}$ Left shift by 1) OR0;

iii. If nxt$Min_{10} \in$ F then

$E^i_{10}=(E^i_{10}$ Left shift by 1) OR 1;

iv. Else

E^i_{10}= (E^i_{10} Left shift by 1) OR0;

v. Set nxtMin_{01}=nxtMin_{01}+1 ;

vi. Set nxtMin_{10}=nxtMin_{10}+1 ;

B. End loop

C. Set$D^i=2^{(i+2) \bmod n}$;

D. Set prvMin_{01}=prvMin_{01}+D^i ;

E. Set prvMin_{10}=prvMin_{10}+D^i ;

F. Set nxtMin_{01}=prvMin_{01}

G. Set nxtMin_{10}=prvMin_{10}

6. End loop

7. End

Example 1.To identify total symmetry, consider the five variable switching function

$F(x_4, x_3, x_2, x_1, x_0) = \Sigma(7,11,13,14,19,21,22,25,26,28)$.

The reduced invariant set

$$R' = \{x_0 \sim x_1, x_1 \sim x_2, x_2 \sim x_3, x_3 \sim x_4, x_4 \sim x_0\}$$

Where x_i is either primed or unprimed.

Now examine this five invariants independently to check whether all the invariants in R'holds for the given function or not.

To check whether the first invariant $x_0 \sim x_1$ holds or not, it is necessary to construct the existent parameters $E^0_{01}$and E^0_{10} first**.** If$E^0_{01} \oplus E^0_{10}=0$, then $x_0 \sim x_1$and $\overline{x}_0 \sim \overline{x}_1$.Here in given function the number of variables is n=5, and the number of bits in each of the existent parameter is $k=2^{n-2}=2^{5-2}=8$.

Construction of $E^0_{01}$and E^0_{10} , where i=0

According to Definition 5, initially $E^0_{01}= 0$ and $E^0_{10}= 0$. The MSB of E^i_{01}denotes the existence of the minterm$M^i_{01_{k-1}}=2^i$ according to Theorem 3. So, the MSB of E^0_{01} denotes the existence of the minterm$M^0_{01_3}=2^0=1$. Similarly from Theorem 4, the MSB of E^i_{01} denote the existence of the minterm $M^i_{10_{k-1}}=2^{(i+1) \bmod n}$.Therefore, the MSB of E^0_{10}denotes the existence of the minterm $M^0_{10_3}=2^{(0+1) \bmod 4}=2^1=2$.

Here $(k/2^{i \bmod (n-1)})= (8/2^{0 \bmod 4})= 8$ number of passes are required and in each pass $2^{i \bmod (n-1)}=1$number of iterations are needed. For each pass displacement $D^i=2^{(i+2) \bmod n}$ is neededto generate next set of minters according to Theorem 6. Here $D^0 = 2^{(0+2) \bmod 5} = 4$.

$prvMin_{01} = M^0_{01_3} = 1$; $nxtMin_{01} = prvMin_{01} = 1$;

$prvMin_{10} = M^0_{10_3} = 2$; $nxtMin_{10} = prvMin_{10} = 2$.

Pass 1.Iteration 1.

As $1 \notin F$; $E^0_{01} = \left(\text{left shift } E^0_{01} \text{by } 1\right)$ OR 0=0 OR 0 = 0

As $2 \notin F$; $E^0_{10} = \left(\text{left shift } E^0_{10} \text{by } 1\right)$ OR 0 =0 OR 0 =0

For next pass: $prvMin_{01} = prvMin_{01} + D^0 = 1 + 4 = 5$

$prvMin_{10} = prvMin_{10} + D^0 = 2 + 4 = 6$

$nxtMin_{01} = prvMin_{01} = 5$; $nxtMin_{10} = prvMin_{10} = 6$

Pass 2.Iteration 1.

As $5 \notin F$; $E^0_{01} = \left(\text{left shift } E^0_{01} \text{by } 1\right)$ OR 0=0 OR 0 =0

As $6 \notin F$; $E^0_{10} = \left(\text{left shift } E^0_{10} \text{by } 1\right)$ OR 0 =0 OR 0 =0

For pass 3: $prvMin_{01} = prvMin_{01} + D^0 = 5 + 4 = 9$

$prvMin_{10} = prvMin_{10} + D^0 = 6 + 4 = 10$

$nxtMin_{01} = prvMin_{01} = 9$; $nxtMin_{10} = prvMin_{10} = 10$

Pass 3.Iteration 1.

As $9 \notin F$; $E^0_{01} = \left(\text{left shift } E^0_{01} \text{by } 1\right)$ OR 0 =0 OR 0 =0

As $10 \notin F$; $E^0_{10} = \left(\text{left shift } E^0_{10} \text{by } 1\right)$ OR 0 =0 OR 0 =0

For pass 4: $prvMin_{01} = prvMin_{01} + D^0 = 9 + 4 = 13$

$prvMin_{10} = prvMin_{10} + D^0 = 10 + 4 = 14$

$nxtMin_{01} = prvMin_{01} = 13$; $nxtMin_{10} = prvMin_{10} = 14$

Pass 4.Iteration 1.

As $13 \in F$; $E^0_{01} = \left(\text{left shift } E^0_{01} \text{by } 1\right)$ OR 1=0 OR 1 =1

As $14 \in F$; $E^0_{10} = \left(\text{left shift } E^0_{10} \text{by } 1\right)$ OR 1 =0 OR 1=1

For pass 5: $prvMin_{01} = prvMin_{01} + D^0 = 13 + 4 = 17$

$prvMin_{10} = prvMin_{10} + D^0 = 14 + 4 = 18$

$nxtMin_{01} = prvMin_{01} = 17$; $nxtMin_{10} = prvMin_{10} = 18$

Pass 5.Iteration 1.

As $17 \notin F$; $E^0_{01} = \left(\text{left shift } E^0_{01} \text{by } 1\right)$ OR 0=2 OR 0 =2

As $18 \notin F$; $E^0_{10} = \left(\text{left shift } E^0_{10} \text{by } 1\right)$ OR 0 =2 OR 0=2

For pass 6: $prvMin_{01} = prvMin_{01} + D^0 = 17 + 4 = 21$

$prvMin_{10} = prvMin_{10} + D^0 = 18 + 4 = 22$

$nxtMin_{01} = prvMin_{01} = 21$; $nxtMin_{10} = prvMin_{10} = 22$

Pass 6.Iteration 1.

As $21 \in F$; $E^0_{01} = \left(\text{left shift } E^0_{01} \text{by } 1\right)$ OR 1=4 OR 1 =5

As $22 \in F$; $E^0_{10} = \left(\text{left shift } E^0_{10} \text{by } 1\right)$ OR 1 =4 OR 1=5

For pass 7: $prvMin_{01} = prvMin_{01} + D^0 = 21 + 4 = 25$

$prvMin_{10} = prvMin_{10} + D^0 = 22 + 4 = 26$

$nxtMin_{01} = prvMin_{01} = 25$; $nxtMin_{10} = prvMin_{10} = 26$

Pass 7.Iteration 1.

As $25 \in F$; $E^0_{01} = \left(\text{left shift } E^0_{01} \text{by } 1\right)$ OR 1=10 OR 1 =11

As $26 \in F$; $E^0_{10} = \left(\text{left shift } E^0_{10} \text{by } 1\right)$ OR 1 =10 OR 1=11

For pass 8: $prvMin_{01} = prvMin_{01} + D^0 = 25 + 4 = 29$

$prvMin_{10} = prvMin_{10} + D^0 = 26 + 4 = 30$

$nxtMin_{01} = prvMin_{01} = 29$; $nxtMin_{10} = prvMin_{10} = 30$

Pass 8.Iteration 1.

As $29 \notin F$; $E^0_{01} = \left(\text{left shift } E^0_{01} \text{by } 1\right)$ OR 0= 22 OR 0 =22

As $30 \notin F$; $E^0_{10} = \left(\text{left shift } E^0_{10} \text{by } 1\right)$ OR 0 = 22 OR 0=22

Hence, E^0_{01} and E^0_{10} are obtained. As $E^0_{01} \oplus E^0_{10} = 0$; $x_0 \sim x_1$ and $\overline{x}_0 \sim \overline{x}_1$ holds. Now continuing the process and find E^1_{01} and E^1_{10} to check whether $x_1 \sim x_2$ or not.

Construction of $\mathbf{E^1_{01}}$ and $\mathbf{E^1_{10}}$, where i=1

Similarly, $M^1_{01_3} = 2^1 = 2$; $M^1_{10_3} = 2^{(1+1) \bmod 5} = 2^2 = 4$.

Here $(k/2^{i \bmod (n-1)}) = (8/2^{1 \bmod 4}) = 4$ number of passes are required. In each pass $2^{i \bmod (n-1)} = 2$ number of iterations are required and displacement $D^i = 2^{(i+2) \bmod n}$ helps to generate next set of minterms as in Theorem 6. Here, $D^1 = 2^{(1+2) \bmod 5} = 8$. In every iteration the next minterm is generated by adding 1 with its previous value. Initially $E^1_{01} = 0$ and $E^1_{10} = 0$. Here,

$prvMin_{01} = M^1_{01_3} = 2$; $nxtMin_{01} = prvMin_{01} = 2$;

$prvMin_{10} = M^1_{10_3} = 4$; $nxtMin_{10} = prvMin_{10} = 4$.

Pass 1.

Iteration 1.

As $2 \notin F$; $E^1_{01} = \left(\text{left shift } E^1_{01} \text{by } 1\right)$ OR 0=0 OR 0 =0

As $4 \notin F$; $E^1_{10} = \left(\text{left shift } E^1_{10} \text{by } 1\right)$ OR 0 =0 OR 0 =0

For next iteration : $nxtMin_{01} = nxtMin_{01} + 1 = 2 + 1 = 3$

And $nxtMin_{10} = nxtMin_{10} + 1 = 4 + 1 = 5$

Iteration 2.

As $3 \notin F$; $E^1_{01} = \left(\text{left shift } E^1_{01} \text{by } 1\right)$ OR 0=0 OR 0=0

As $5 \notin F$; $E^1_{10} = \left(\text{left shift } E^1_{10} \text{by } 1\right)$ OR 0 =0 OR 0=0

For pass 2: $prvMin_{01} = prvMin_{01} + D^1 = 2 + 8 = 10$

$prvMin_{10} = prvMin_{10} + D^1 = 4 + 8 = 12$

$nxtMin_{01} = prvMin_{01} = 10$; $nxtMin_{10} = prvMin_{10} = 12$

Pass 2.

Iteration 1.

As $10 \notin F$; $E^1_{01} = \left(\text{left shift } E^1_{01} \text{by } 1\right)$ OR 0 =0 OR 0 =0

As $12 \notin F$; $E^1_{10} = \left(\text{left shift } E^1_{10} \text{by } 1\right)$ OR 0 =0 OR 0 =0

For next iteration: $nxtMin_{01}=nxtMin_{01}+1=10+1=11$ and

$nxtMin_{10}=nxtMin_{10}+1=12+1=13$

Iteration 2.

As $11 \in F$; $E^1_{01} = \left(\text{left shift } E^1_{01} \text{by } 1\right)$ OR 1=0 OR 1 = 1

As $13 \in F$; $E^1_{10} = \left(\text{left shift } E^1_{10} \text{by } 1\right)$ OR 1=0 OR 1 = 1

For pass 3: $prvMin_{01}=prvMin_{01}+D^1=10+8=18$

$prvMin_{10}=prvMin_{10}+D^1=12+8=20$

$nxtMin_{01}=prvMin_{01}=18$; $nxtMin_{10}=prvMin_{10}=20$

Pass 3.

Iteration 1.

As $18 \notin F$; $E^1_{01} = \left(\text{left shift } E^1_{01} \text{by } 1\right)$ OR 0 =2 OR 0 =2

As $20 \notin F$; $E^1_{10} = \left(\text{left shift } E^1_{10} \text{by } 1\right)$ OR 0 =2 OR 0 =2

For next iteration: $nxtMin_{01}=nxtMin_{01}+1=18+1=19$ and

$nxtMin_{10}=nxtMin_{10}+1=20+1=21$

Iteration 2.

As $19 \in F$; $E^1_{01} = \left(\text{left shift } E^1_{01} \text{by } 1\right)$ OR 1=4 OR 1 = 5

As $21 \in F$; $E^1_{10} = \left(\text{left shift } E^1_{10} \text{by } 1\right)$ OR 1=4 OR 1 = 5

For pass 4: $prvMin_{01}=prvMin_{01}+D^1=18+8=26$

$prvMin_{10}=prvMin_{10}+D^1=20+8=28$

$nxtMin_{01}=prvMin_{01}=26$; $nxtMin_{10}=prvMin_{10}=28$

Pass 4.

Iteration 1.

As $26 \in F$; $E^1_{01} = \left(\text{left shift } E^1_{01} \text{by } 1\right)$ OR 1 =10 OR 1 =11

As $28 \in F$; $E^1_{10} = \left(\text{left shift } E^1_{10} \text{by } 1\right)$ OR 1 =10 OR 1 =11

For next iteration: $nxtMin_{01}=nxtMin_{01}+1=26+1=27$

and$nxtMin_{10}=nxtMin_{10}+1=28+1=29$

Iteration 2.

As $27 \notin F$; $E^1_{01} = \left(\text{left shift } E^1_{01} \text{by } 1\right)$ OR 0=22 OR 0 = 22

As $29 \notin F$; $E^1_{10} = \left(\text{left shift } E^1_{10} \text{by } 1\right)$ OR 0=22 OR 0 = 22

Hence E^1_{01}, E^1_{10} are obtained. As $E^1_{01} \oplus E^1_{10}=0$ so $x_1 \sim x_2$ and $\overline{x}_1 \sim \overline{x}_2$ holds for the given function.

So, the process can be continued and E^2_{01}, E^2_{10} have to be constructed where i=2 to check whether $x_2 \sim x_3$ hold or not. Similarly E^2_{01}, E^2_{10} can also be constructed as previous and here in this case $E^2_{01}=22$ and $E^2_{10}=22$ are found. As $E^2_{01} \oplus E^2_{10}=0$; $x_2 \sim x_3$ and $\overline{x}_2 \sim \overline{x}_3$ holds.

So, the process can be continued to find E^3_{01}, E^3_{10} where i=3 to check whether $x_3 \sim x_2$ hold or not. Similarly the construction of E^3_{01}, E^3_{10} can be carried out and $E^3_{01}=22, E^3_{10}=22$ are found. As $E^3_{01} \oplus E^3_{10}=0$; $x_3 \sim x_4$ and $\overline{x}_3 \sim \overline{x}_4$.

So, the process can be continuedto find E^4_{01}, E^4_{10} where i=4 to check whether $x_4 \sim x_0$ hold or not. Similarly, $E^4_{01}=22$ and $E^4_{10}=22$ can be found. As $E^4_{01} \oplus E^4_{10}=0$; therefore $x_4 \sim x_0$ and $\overline{x}_4 \sim \overline{x}_0$ satisfy.

As $\{x_0 \sim x_1, x_1 \sim x_2, x_2 \sim x_3, x_3 \sim x_4, x_4 \sim x_0\}$ holds; hence the function $F(x_4, x_3, x_2, x_1, x_0)=\Sigma(7,11,13,14,19,21,22,25,26,28)$

is totally symmetric.

5.3 Detection of Total and Multiform Symmetry

For detection of Total as well as multiform symmetry the existent parameters E^i_{00}, E^i_{01}, E^i_{10} and E^i_{11} has to be generated $\forall$ i=0 to n-1.

If $E^i_{01} \oplus E^i_{10}=0$ then $x_i \sim x_{(i+1) \bmod n}$ and $\overline{x}_i \sim \overline{x}_{(i+1) \bmod n}$.

If $E^i_{00} \oplus E^i_{11}=0$ then $x_i \sim \overline{x}_{(i+1) \bmod n}$ and $\overline{x}_i \sim x_{(i+1) \bmod n}$.

For any of i, if both the conditions i.e. $E^i_{01} \oplus E^i_{10}=0$ and $E^i_{00} \oplus E^i_{11}=0$ do not hold for the given function, i.e., x_i is not invariant with $x_{(i+1) \bmod n}$ where each of them can be primed or unprimed, then the function is not total or multiform symmetry. Hence the process of finding the next set of existent parameters shall be terminated and it can be concluded that function is not total or multiform symmetry.

Proposed Algorithm for Detection of Total and Multiform Symmetric Functions

Input: A Switching Function of n variables

Output: Total or multiform Symmetry or Not

1. For i=0 to n-1 do

A. Call Construct_ E^i_{00}_E^i_{01}_E^i_{10}_E^i_{11} ;

B. If $E^i_{01} \oplus E^i_{10} \neq 0$ then

I. If $E^i_{00} \oplus E^i_{11} \neq 0$ then

Print "The function is not Symmetry";

Exit;

II. End if

C. End if

2. End for

3. print "the function is Symmetry";

4. End;

Procedure Construct_ E^i_{00}_E^i_{01}_E^i_{10}_E^i_{11}

Input: number of literals in the given function i.e. n

Output: The Existent parameters E^i_{00}, E^i_{01}, E^i_{10}, E^i_{11}

1. Set $k=2^{n-2}$;

2. Set $M^i_{00_{k-1}}=0$ and $M^i_{11_{k-1}}= 2^{(i+1) \bmod n}+2^i$;
3. Set $M^i_{01_{k-1}}=2^i$ and $M^i_{00_{k-1}}=2^{(i+1) \bmod n}$;
4. Set $prvMin_{00}=M^i_{00_{k-1}}$ and $nxtMin_{00}=prvMin_{00}$;
5. Set $prvMin_{01}=M^i_{01_{k-1}}$ and $nxtMin_{01}=prvMin_{01}$;
6. Set $prvMin_{10}=M^i_{10_{k-1}}$ and $nxtMin_{10}=prvMin_{10}$;
7. Set $prvMin_{11}=M^i_{11_{k-1}}$ and $nxtMin_{11}=prvMin_{11}$;
8. For every pass=1 to $(k/2^{i \bmod (n-1)})$
   a. For every p=1 to $2^{i \bmod (n-1)}$
      i. If $nxtMin_{00} \in F$, then
         $E^i_{00}=(E^i_{00}$ Left shift by 1) OR 1;
      ii. Else
         $E^i_{00}=(E^i_{00}$ Left shift by 1) OR0;
      iii. If $nxtMin_{01} \in F$, then
         $E^i_{01}=(E^i_{01}$ Left shift by 1) OR 1;
      iv. Else
         $E^i_{01}=(E^i_{01}$ Left shift by 1) OR0;
      v. If $nxtMin_{10} \in F$, then
         $E^i_{10}=(E^i_{10}$ Left shift by 1) OR 1;
      vi. Else
         $E^i_{10}=(E^i_{10}$ Left shift by 1) OR0;
      vii. If $nxtMin_{11} \in F$,then
         $E^i_{11}=(E^i_{11}$ Left shift by 1) OR 1;
      viii. Else
         $E^i_{11}=(E^i_{11}$ Left shift by 1) OR0;
      ix. Set $nxtMin_{00}=nxtMin_{00}+1$;
      x. Set $nxtMin_{01}=nxtMin_{01}+1$;
      xi. Set $nxtMin_{10}=nxtMin_{10}+1$;
      xii. Set $nxtMin_{11}=nxtMin_{11}+1$;
   b. End loop
   c. Set $D^i=2^{(i+2) \bmod n}$;
   d. Set $prvMin_{00}=prvMin_{00}+ D^i$;
   e. Set $prvMin_{01}=prvMin_{01}+D^i$;
   f. Set $prvMin_{10}=prvMin_{10}+D^i$;
   g. Set $prvMin_{11}=prvMin_{11}+D^i$;
   h. Set $nxtMin_{00}=prvMin_{00}$;
   i. Set $nxtMin_{01}=prvMin_{01}$;
   j. Set $nxtMin_{10}=prvMin_{10}$;
   k. Set $nxtMin_{11}=prvMin_{11}$;
9. End loop
10. End;

Example 2.To detect total symmetry and multiform symmetry the four variable switching function is given below:

$$F(x_3,x_2,x_1,x_0)=\Sigma(0,1,3,5,8,10,11,12,13,15);$$

The reduced invariant set

$$R' = \{x_0{\sim}x_1,x_1{\sim}x_2,x_2{\sim}x_3,x_3{\sim}x_0\}$$

Where x_i is either primed or unprimed. Nowexamine this four invariants to check whether all the invariants in R' holds for the given function or not. Here in given function the number of variables is n=4, and the number of bits in each of the existent parameter is $k= 2^{n-2}= 2^{4-2}= 4$.

To check whether the first invariant $x_0{\sim}x_1$ holds or not the existent parameters E^0_{00}, E^0_{01}, E^0_{10} and E^0_{11} has to be constructed. If $E^0_{01} \oplus E^0_{10}= 0$, then $x_0{\sim}x_1$ and $\overline{x}_0{\sim}\overline{x}_1$. If $E^0_{00} \oplus E^0_{11}= 0$, then $x_0{\sim}\overline{x}_1$ and $\overline{x}_0{\sim}x_1$. The construction of existent parameters E^0_{00}, E^0_{01}, E^0_{10} and E^0_{11} can be carried out as it has been done in Example 1.For the given switching function $E^0_{00}= 11$, $E^0_{01}= 13$, $E^0_{10}= 2$ and $E^0_{11}= 11$.

As $E^0_{00} \oplus E^0_{11}= 0$, $x_0{\sim}\overline{x}_1$ and $\overline{x}_0{\sim}x_1$ holds, continue the process and find E^1_{00}, E^1_{01}, E^1_{10} and E^1_{11} for i=1to check whether $x_1{\sim}x_2$ holds or not. After the construction of the existent parameters $E^1_{00}=14$, $E^1_{01}=7$, $E^1_{10}= 7$ and $E^1_{11}= 1$.Again as $E^1_{01} \oplus E^1_{10}= 0$, $x_1{\sim}x_2$ and $\overline{x}_1{\sim}\overline{x}_2$ holds. Continue the process and find E^2_{00}, E^2_{01}, E^2_{10} and E^2_{11} for i = 2to check whether $x_2{\sim}x_3$ satisfies or not. After the construction of the existent parameters $E^2_{00}=13$, $E^2_{01}= 4$, $E^2_{10}= 11$, and $E^2_{11}= 13$ will be found. As $E^1_{00} \oplus E^1_{11}= 0$; $x_2{\sim}\overline{x}_3$ and $\overline{x}_3{\sim}x_2$ holds and continue the process to find the last set of existent parameters E^3_{00}, E^3_{01}, E^3_{10} and E^3_{11}; for i = 3 to check whether $x_3{\sim}x_0$ holds or not. After the construction of the existentparameters $E^3_{00}= 8$, $E^3_{01}= 14$, $E^3_{10}= 14$ and $E^3_{11}= 7$ will be found.As $E^3_{01} \oplus E^3_{10}= 0$, $x_3{\sim}x_0$ and $\overline{x}_3{\sim}\overline{x}_0$ holds.

As $\{ x_0{\sim}\overline{x}_1,\overline{x}_0{\sim}x_1,x_1{\sim}x_2,\overline{x}_1{\sim}\overline{x}_2, x_2{\sim}\overline{x}_3 , \overline{x}_3{\sim}x_2, x_3{\sim}x_0 \text{ and } \overline{x}_3{\sim}\overline{x}_0\}$ holds for the given function; hence the given function is multiform symmetric.

6. EXPERIMENTAL RESULTS AND DISCUSSIONS

Table 1: Results for total symmetry identification

SI.No.	# literals	Delivered switching functions in SOP	Decomposition chart (ms)[11]	Residue Method (ms)[13]	Kohavi Method (ms)[14]	Proposed Method (ms)
1	3	Σ(3,5,6)	0.134	0.156	0.2	0.163
2	3	Σ(1,2,3,4,5,6)	0.152	0.143	0.206	0.175
3	4	Σ(3,5,6,9,10,12,15)	0.26	0.192	0.291	0.236

4	4	Σ(1,2,4,7,8,11,13,14,15)	0.188	0.21	0.261	0.193
5	4	Σ(0,3,5,6,7,9,10,11,12,13,14,15)	0.456	0.487	0.563	0.326
6	5	Σ(0,15,23,27,29,30,31)	0.38	0.409	0.398	0.341
7	5	Σ(1,2,4,8,15,16,23,27,29,30)	0.3	0.379	0.431	0.37
8	5	Σ(0,1,2,4,8,15,16,23,27,29,30)	0.43	0.461	0.489	0.411
9	5	Σ(7,11,13,14,15,19,21,22,23,25,26,27,28,29,30)	0.63	0.653	0.745	0.591
10	5	Σ(1,2,3,4,5,6,8,9,10,12,15,16,17,18,20,23,24,27,29, 30)	0.486	0.56	0.51	0.327
11	5	Σ(3,5,6,7,9,10,11,12,13,14,17,18,19,20,21,22,24,25,26,28)	0.515	0.526	0.623	0.55
12	6	Σ(0,5,7,11,25,28,30,55,58,63)	0.202	0.237	0.269	0.187
13	6	Σ(0,2,4,9,10,23,28,37,50,59,61)	0.217	0.131	0.28	0.16
14	6	Σ(0,1,5,13,14,19,23,28,32,44,46,51,55,56,58,59)	0.209	0.36	0.184	0.21
15	7	Σ(0,1,8,10,13,18,36,54,57,69,90,106,117)	0.177	0.174	0.214	0.169
AVERAGE			0.316	0.339	0.378	0.294

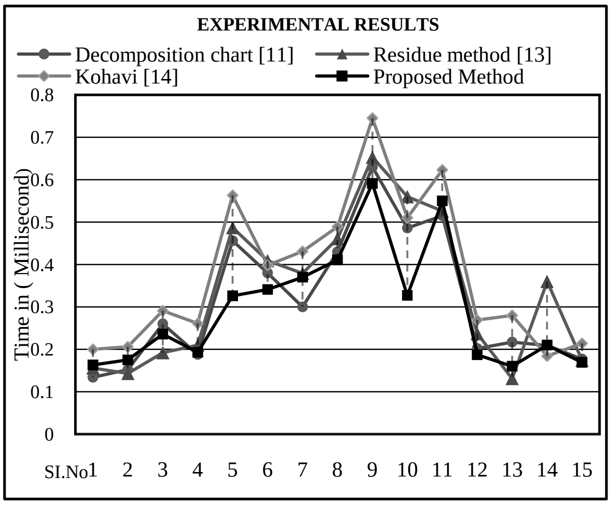

Fig. 1. Performance chart of Table 1

Table 2. Enumeration of switching functions

# literals	# Generated Functions	# Totally symmetric Functions	Decomposition chart (sec)[11]	Residue method (sec)[13]	Kohavi Method (sec)[14]	Proposed Method (sec)
1	4	4	0	0	0	0
2	16	8	0	0	0	0
3	256	16	0.0035	0.006	0.008	0.0029
4	65536	32	0.09	0.12	0.48	0.07
5	4294967296	64	12901.37	14895.26	19562.49	6537.13

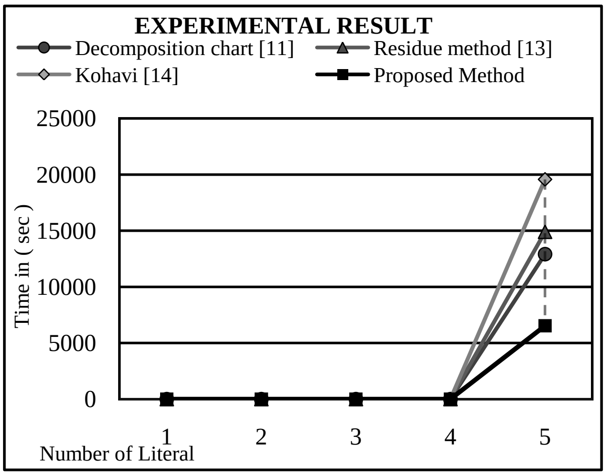

Fig. 2. Performance Chart of Table 2

Table 3. Results for total and multi symmetry identification

SI.No.	# literals	Delivered switching functions in SOP	Decomposition chart (ms)[11]	Residue Method (ms)[13]	Modified Kohavi Method (ms)[21]	Proposed Method (ms)
1	3	Σ(0,1,3,4,6,7)	0.188	0.201	0.179	0.185
2	4	Σ(0,3,5,6,9,10,12,15)	0.22	0.243	0.192	0.256
3	4	Σ(1,2,3,4,5,6,8,9,10,12)	0.189	0.275	0.321	0.22
4	4	Σ(3,5,6,7,9,10,11,12,13,14,15)	0.362	0.486	0.43	0.271
5	5	Σ(0,1,3,4,6,7,8,10,11,14)	0.381	0.317	0.301	0.283
6	5	Σ(0,9,10,12,17,18,20,24,27,29,30)	0.398	0.478	0.497	0.422

7	5	Σ(3,4,9,10,15,17,18,23,24,27,29,30)	0.26	0.194	0.324	0.243
8	5	Σ(0,1,3,4,6,7,8,10,11,14,17,20,21,23,24,25,27,28,30,31)	0.653	0.692	0.654	0.572
9	6	Σ(21,37,49,52,55,61)	0.352	0.45	0.405	0.43
10	6	Σ(19,35,49,50,51,55,59)	0.251	0.287	0.365	0.158
11	6	Σ(0,1,5,13,14,19,21,27,32,44,46,51,55,56,58,63)	0.38	0.403	0.332	0.292
12	7	Σ(26,43,52,59,68,75,84,101)	0.595	0.562	0.575	0.515
13	7	Σ(0,1,8,10,13,18,36,54,57,69,90,106,117)	0.43	0.465	0.549	0.402
14	7	Σ(0,1,5,13,19,21,27,44,49,56,57,72,83,95,110,116)	0.88	0.932	1.03	0.681
15	8	Σ(0,6,19,27,54,68,96,102,117,128,139,145)	0.71	0.823	0.761	0.642
AVERAGE			0.416	0.453	0.461	0.372

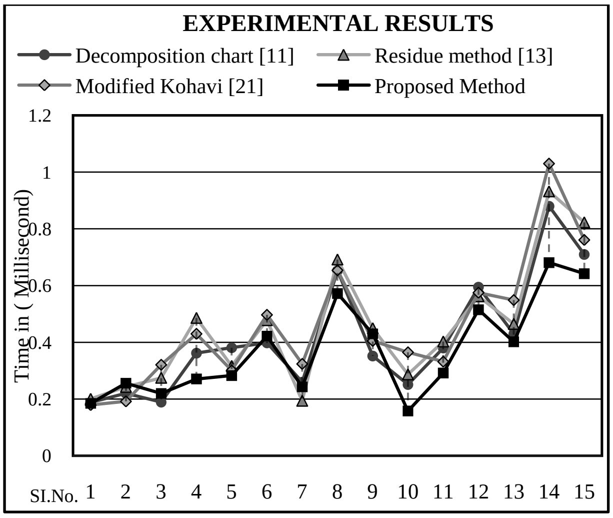

Fig. 3.Performance Chart of Table 3

The proposed approach of symmetry detection has been compared with various existing methods of symmetry detection. The methods have been implemented in C language on UNIX platform. For comparing the outcomes, the results of some existing standard symmetry detection methods have been tabulated along with the proposed method. As it turns out the proposed approach takes less time in most of the instances when compared to the other existing methods. Hence it can be concluded that the proposed method is optimum when viewed in time-space domain.

7. CONCLUSION

Detection of total and multiform symmetries cannot be viewed only as an NP complete problem, testing for any of the well-known switching functions of n literals leads to coNP complete. Various methods are available that give acceptable performance in practice, even though their worst case complexity is exponential in terms of number of input literals. These algorithms are generally used with relatively few numbers of terms, and in function representations that can be exponentially large compared to the corresponding switching variables. If all these algorithms were executed on switching expressions with large number of inputs, their exponential behaviour would be much more evident. The document shows a novel approach for identification of totally and multiform symmetric functions, where the set of true minterms shows their invariant properties in terms of modulo-2 addition between existent parameters. The suggested algorithm can be used for both completely and incompletely specified switching functions. The attempt is worth studying, efficient approach for detection of group and rotational symmetry isthe own envy. The next attempt is to build a catalogue of symmetries under function description and work with other types of special switching functions, which will be a new attachment to the existing state-of-the-art technology.

8. REFERENCES

[1] C. E. Shannon, “A symbolic analysis of relay and switching circuits,”AIEE Trans., vol. 57, pp. 713-723, 1938.

[2] G. Birkhoff and S. MacLane, “A Survey of Modern Algebra”, New York: Macmillan, 1953.

[3] S. H. Caldwell, “The recognition and identification of symmetric switching functions,” Trans. AIEE (Communication and Electronics), vol. 73, pt. 1,pp. 142-146, May 1954

[4] A. A. Mullin and W. G. Kellner, “A residue test for Boolean functions,” unpublished teaching memorandum, Dept. of Elec. Engrg., M.I.T., Cambridge, Mass., 1955.

[5] E. J. McCluskey, “Detection of group invariance or total symmetry of a Boolean function,” Bell Sys. Tech. J., vol. 35, pp. 1445-1453, November 1956.

[6] M. P. Marcus, “The detection and identification of symmetric switching functions with the use of tables of combinations,” IRE Trans. on Electronic Computers(Correspondence), vol. EC-5, pp. 237-239, December 1956.

[7] T. Singer, “Some uses of truth tables”, Proc. Int. Symp. on the Theory of Switching (Part I), pp. 125-133, April 1959.

[8] A. K. Choudhury and M. S. Basu, “On detection of group invariance or total symmetry of a Boolean function”, Indian J. Phys., vol. 36, pp.31-42, January 1962.

[9] S. R. Das, “Detection of invariance, total symmetry and partial symmetry of switching functions”, Indian J. Phys., vol. 37, pp. 219-232,April 1963.

[10] R. F. Arnold and M. A. Harrison, “Algebraic properties of symmetric and partially symmetric Boolean functions,” IEEE Trans. Electronic Computers, vol. EC-12, pp. 244-251, June 1963.

[11] A. Mukhopadhyay, “Detection of total or partial symmetry of a switching function with the use of decomposition charts,” IEEE Trans. on Electronic Computers(Correspondence), vol. EC-12, pp. 553-557, October 1963.

[12] C. L. Sheng, “Detection of totally symmetric Boolean functions,” IEEE Trans. Electronic Computers (Short Notes), vol. EC-14, pp.924-926, December 1965.

[13] S. Das and C. Sheng, “On detecting total or partial symmetry of switching functions”, IEEE Trans, on

Computers, 352-355, 1971.DOI: 10.1109/PROC.1970.7777

[14] Z.Kohavi, Switching and Finite Automata Theory, Tata McGraw-Hill publishers, Second Edition, pp.126-127,1978

[15] C.C. Tsai and M. MarekSadowska, "Generalized Reed Muller forms as a tool to detect symmetries," IEEE Trans. Comput., vol. 45, pp. 33-40, Jan. 1996. DOI: 10.1109/ISCAS.1994.408811

[16] V. N. Kravets, K. A. Sakallah, "Generalized Symmetries in Boolean Functions," International Conference on ComputerAided Design (ICCAD'00), 526, 2000. DOI: 10.1109/ICCAD.2000.896526

[17] T. Sasao, "A new expansion of symmetric functions and their application to non-disjoint functional decompositions for LUT type FPGAs," IEEE International Workshop on Logic Synthesis, pp. 105-110, 2000.

[18] Peter M Maurer, "Using Conjugate Symmetries to Enhance Gate-Level Simulations", Dept. of Computer Science, Baylor University #97356 WACO, TX-76798-7356, 2006

[19] J.S. Zhang, A. Mishchenko, R. Brayton and M.Chrzanowska-Jeske, "Symmetry detection for large Boolean functions using circuit representation, simulation, and Satisfiability". 43rd ACM/IEEE Design Automation Conference, USA, CA,ISBN:1-59593-381-6, pp. 510-515, 2006. DOI: 10.1109/DAC.2006.229269

[20] N. Kettle, A. King, "An Anytime Algorithm for Generalized Symmetry Detection in ROBDDs", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, pp. 764 – 777, Volume: 27 Issue: 4, April 2008.

[21] SubhajitGuha, SatrajitGhosh, ParthaPratimDas, "A Novel Approach for Identification of Symmetric Switching Function", Recent Advances in Information Technology (RAIT),1st International Conference, RAIT, India, IEEE,ISBN: 978-1-4577-0694-3,pp.731-736,March2012. DOI:10.1109/RAIT.2012.6194545

[22] Peter M Maurer, "Using Conjugate Symmetries to Enhance Gate-Level Simulations", Dept. of Computer Science, Baylor University #97356 WACO, TX-76798-7356, 2006