Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 240-250

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

BDD based Optimization Algorithm for High Performance PTL Synthesis

[1]Thangavel Bhuvaneswari, [2]V.C.Prasad, [3]Ajay Kumar Singh

[1,3]Multimedia University, Faculty of Engineering and Technology, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia
[2]DayalBagh Educational Institute ,[2]Department of Electrical Engineering Agra 282005, Uttar Pradesh India

ARTICLE INFO

Article history:
Received 23 June 2015
Accepted 25 August 2015
Available online 2 September 2015

Keywords:
BDDs, Algorithms, Mapping, LEAP cells, MUX and PTL Synthesis

ABSTRACT

Background: BDDs are the state-of-the-art data structure for the representation and manipulation of Boolean functions. Realization of Binary Decision Diagrams (BDDs) using *LEA*n integration of *P*ass transistor logic synthesis (LEAP) cells is of great interest since BDD node can be directly mapped to the pass transistor logic (PTL) cell. LEAP cells (Y3, Y2, and Y1 cells) are multiplexers implemented in PTL logic. The total number of Y cells need to be minimized for an efficient logic synthesis. By minimizing the Y cells, the numbers of inverters used are reduced. A systematic mapping of BDD nodes with LEAP cells is necessary to achieve this. By increasing the number of Y3 cells (grouping of three BDD nodes), the overall cell count can be reduced. **Objective:** In this paper, an algorithm is proposed to increase the number of Y3 cells and reduce the overall cell count. The algorithm gives a global maximal set of Y3 cells. The proposed algorithm can also be used to get maximum number of Y2 cells (grouping of two BDD nodes) for the remaining nodes to achieve an efficient logic synthesis. **Results:** Simulations results of the benchmark circuits based on the proposed algorithm are encouraging in terms of area, output capacitance and total power consumption compared to the existing techniques. **Conclusion:** The detailed discussion on the pseudo code and the complexity analysis of the proposed algorithm are performed. The proposed algorithm is illustrated with the Benchmark circuit CM152. Simulations results are encouraging, and high performance PTL synthesis is achieved.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: Thangavel Bhuvaneswari, V.C.Prasad, Ajay Kumar Singh., BDD based Optimization Algorithm for High Performance PTL Synthesis. ***Aust. J. Basic & Appl. Sci.,*** *9(27): 240-250, 2015*

INTRODUCTION

The increasing speed and complexity of today's designs implies that CAD tools play a major role in the design development. Designers rely on the tools for every aspect of the development cycle starting with specification, design entry, logic synthesis and optimization. BDDs are the state-of-the-art data structure for the representation and manipulation of Boolean functions. BDDs are mainly graph representation of Boolean functions. The efficient manipulations of BDDs were developed by Bryant (1986). The BDD size depends on the variable order. Finding the variable order that determines the BDD size (linear or exponential) is an important research topic The depth first approach is used in algorithms by Malik *et al*., 1988. Determining an optimal variable ordering is an NP-hard problem (Justin *et al*., 2001). Friedman and Supowit (1990) proposed algorithm leads to minimal BDD size with the restriction on the number of inputs. Another factor that influences the size of the BDD is the decomposition rules. The Basic Shannon's decomposition can be replaced by positive davio and negative davio decomposition rules proposed by kebschull *et al*., 1992.

Various types of BDDs and their applications in VLSI CAD are explained by Rolf Drechsler *et al*., 2001. BDDs have been used in the synthesis of logic because they allow aspects of circuit synthesis and technology mapping (Lai *et al*., 2005; Shelar *et al*., 2001, 2005; Buch *et al*., 1997; Kao (2011); Bhuvaneswari *et al*., 2013). Several case studies Zimmerman *et al*., 1997; Yano *et al*., 1996,1990 have shown that for most functions, PTL implementation can be done with less number of transistors compared to static CMOS and offers reduction in overall capacitance, increased access time and lower power. The one-to-one correspondence of BDD and PTL synthesis is well known, and that makes the BDD based PTL circuit synthesis important.

Another aspect for logic synthesis and optimization is the technology mapping. Paul *et al*., (2010) quoted that through technology mapping, getting the best optimization is NP- hard. Mapping techniques have got high impact for BDD based logic synthesis (Chaudhry *et al*., 1998; Hsiao *et al*.,

Corresponding Author: Thangavel Bhuvaneswari, Multimedia University, Faculty of Engineering and Technology, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia.
E-mail: t.bhuvaneswari@mmu.edu.my , +606 2523038

2002 and Mallik *et al*.,2004).Yano *et al*. (1996) have proposed the LEAP cell library consists of Y1, Y2 and Y3 cells with different drive inverters. The LEAP cell library is based on the top down design flow for pass-transistor logic called LEAP (LEAn integration of *P*ass transistor logic synthesis). Choudhry *et al*. (1998) developed BDD algorithms for automated PTL synthesis targeting minimum area using LEAP cells. Decomposition techniques were used in PTL to optimize area, delay and power (Buch *et al*., 1997). However, most of the techniques use greedy approach. A cutset based BDD decomposition is proposed by Yen-Tai Lai *et al*., 2005.

Zhuang *et al*. (1999) proposed a logic synthesis tool designed for mapping pass-transistor logic. Pass transistor mapper (PTM) integrates both synthesis and logic optimization in one step, so that it can handle large functions. BDD variable order optimization is achieved through genetic algorithm with dynamic parameters. Synthesis of PTL circuits using the BDD decomposition technique has been presented by Shelar *et al*., 2000 and solved by using the max flow min-cut technique. Shelar *et al*. (2005) performed BDD decomposition by recursive bipartition scheme that uses the max flow min-cut technique. For NAND intensive arithmetic logic circuits and control circuits, the delay reduction is small and inconsistent, compared to the static CMOS circuits.

Geun Rae cho *et al*., 2004 and Lai *et al*., 1997 applied genetic algorithm to find the best mixture of PTL and static CMOS cells for low power applications. To simplify the logic synthesis flow, Yen-Tai Lai *et al*. (1997) proposed the BDD based technique which decompose the logic function into two and mapped them to PTL and CMOS cells accordingly. This approach uses less area than CMOS technology mapping and has better performance. An efficient synthesis algorithm has been proposed by Chi-Chou Kao (2011) based on mixed CMOS/PTL cells. According to algorithm by constructing BDD, which nodes can be implemented by CMOS and PTL cells in a library, performance of the circuit can be improved. Wille and Drechsler (2009) discussed the effect of BDD optimization on synthesis of reversible and quantum logic and later extended the idea for solving large functions.

Paul *et al*. (2010) proposed modified Y3 cells. It also discussed a BDD based technology mapping technique using modified Y cells. But this BDD mapping technique for maximizing Y3 cells and minimizing the total cell count allows logic duplication which increases the transistor count. In this paper, an algorithm is proposed to maximize the Y3 cells (both Y3 and modified Y3 are considered) and reduce the overall cell count in a systematic way without duplication. The proposed algorithm, (algorithm for global maximal set) will always give the global maximal set of Y3 cells. In this paper, LEAP cell library is explained first. Then the proposed algorithm, pseudo code, complexity analysis and illustration are covered in detail. Then the proposed algorithm based simulation of benchmark circuits is discussed. Comparisons of the simulated results with the existing results are also included and finally, conclude the paper with its contributions.

Leap Cell Library:

In pass transistor logic (PTL), transistors are used to pass the logic from input to output. The number of transistors to implement any logic can be reduced by eliminating redundant transistors. A multiplexer or MUX is a combinational circuit which allows only one input to output at one time. A multiplexer can be implemented using CMOS or PTL. A 2:1 MUX has the same function as a BDD node where the outgoing and incoming edges of a BDD node are similar to the input and output of a MUX. The equivalent of a BDD node, 2:1 MUX and PTL implementation of a MUX are shown in Fig 1 (Lai *et al*., 1997).

The BDD-based PTL implementation is quite compact due to: (i) sneak-path-free implementation (for any input combinations in BDDs, only one path is active); (ii) BDD based design is benefited by using various variable reordering techniques to get the optimal BDDs. Better performance and low power dissipation in PTL logics are mainly due to the use of NMOS pass transistors only (Buch *et al*., 1997).

The logic scheme LEAP cell is very popular cell based design library which has three essential logic cells (Y1, Y2, and Y3) and four inverters of different drive capabilities. Y1 cell has to select one of the inputs A or B (Fig 2(a)) inputs depending on the control signal at the gate and appears as the letter Y. In order to minimize the use of number of inverters, the complementary control signals are shared by many cells. Y2 cell has two stacked Y1 cells and Y3 cell has three stacked Y1 cells as shown in Fig 3(a) and 4(a). For the Y3 cell as shown in Fig 4(a), the control signal **C** and its complement are shared by 2 MUX (first MUX with input **A, B** with control signal **C** and $\overline{C}$, second MUX with input **E**,**F** with same control signal **C** and $\overline{C}$ (Chaudhry al., 1998).

A modified Y3 cell is shown in Fig 5(a). By allowing each MUX to have its own control signal and its complement grouping of three BDD nodes at any level is possible (first MUX with input **A**, **B** with control signal **C** and $\overline{C}$, second MUX with input **E**, **F**, with control signal **G** and $\overline{G}$ (Paul *et al*., 2010).

The LEAP cell library makes use of the redundant pass-transistor rules. If the inputs of the pass transistors are 1 and 0 for the control signals A and $\overline{A}$, the reduced expression is A. If the inputs of the pass transistors are 0 and 1 for the control signals

A and $\overline{A}$, the reduced expression is $\overline{A}$. If both of the inputs are same (B,B) for the control signals A and $\overline{A}$, the reduced expression is B(Yano *et al*., 1996).

Realization of the BDD using LEAP cells is achieved by grouping BDD nodes. The BDD nodes have to be grouped into groups of three nodes; or two nodes, or single node as shown in Fig 2(a), 3(a), 4(a) and 5(a). Let Y_{2n} denotes a pair of nodes of a BDD as shown in Fig 3(a). Similarly, let Y_{3n} denotes a group of three nodes of a BDD (Fig 4(a) or 5(a)). Single node is replaced by Y1 cell as shown in Fig 2(b). Y_{2n} is replaced by Y2 cell (Fig 3 (b)). Similarly, Y_{3n} can be replaced by Y3 cell as shown in Fig 4(b) or modified Y3 cell (Fig 5 (b)).

A BDD node, Y_{1n} (Fig 2(a)) is equivalent to Y1 cell. A pair of BDD nodes, Y_{2n} (Fig 3(a)) is equivalent to Y2 cell. A group of three BDD nodes; Y_{3n} (Fig 4(a)) is equivalent to Y3 cell. A group of three BDD nodes; Modified Y3n (Fig 4(a)) is equivalent to modified Y3 cell.

The BDDs generated from the open source blif files are reversed (terminal nodes at the top and the root at the bottom) and are called Reversed BDDs (Bhuvaneswari *et al*., 2013).The reversed BDDs is served as input for the proposed algorithm since it is easy to do the mapping in this way.

Proposed Algorithm:

a. Significance of the proposed algorithm:

In order to reduce the overall count, grouping of BDD nodes can be done in many ways. Here, it is referred to $Y_{3n,}$ $Y_{2n,}$ and Y_{1n} groups. First, make the Y_{3n} group as many as possible followed by Y_{2n} groups and the remaining nodes Y_{1n} cells. This will reduce area, delay and interconnection wires. Nodes of a BDD must be formed into groups such that each node lies in one group only. If the nodes are selected randomly, they may or may not result in maximum number of $Y_{3n,}$ Y_{2n} groups. Even if they resulted in maximum number, there is no guarantee that it is the maximum number of $Y_{3n,}$ Y_{2n} groups that one can achieve. There is no systematic approach to get the maximum number of Y3, and Y2 cells. In this paper, an algorithm is proposed to reduce the overall cell count and maximize the Y3 cells in a systematic way.

The Benchmark circuit CM152 is taken as example shown in Fig 6. It is a single output function with 11 inputs. The starting reversed BDD has 11 levels. All the non-terminal nodes are numbered as (n1-n15). Random selection of Y_{3n} groups may result in less number of Y_{3n} groups, followed by less number of Y_{2n} groups. For example, group of (n9, n10, n13) nodes as $Y_{3n1,}$ group of (n11, n12, n14) nodes as Y_{3n2} groups, then no other Y_{3n} groups possible. Now proceed to Y_{2n} grouping, which is not possible to group Y_{2n}, the remaining nodes are treated as Y_{1n} group. It results in two Y_{3n} and nine Y_{1n} groups. Grouping should be done following the Fig 2(a), 3(a), 4(a) and 5(a). For example, (n3, n5, n1) nodes cannot be grouped as they don't follow the structure of any of figures discussed above.

Selecting the (n3, n7, n9) nodes as $Y_{3n1,}$ (n1, n5, n10) nodes as $Y_{3n2,}$ (n4, n8, n11) nodes as $Y_{3n3,}$ (n13, n14, n15) nodes as $Y_{3n4,}$ and (n2, n12) nodes as Y_{2n1} groups, the remaining node n6 is Y_{1n1} group. This resulted in fourY_{3n} groups, one Y_{2n} group and one Y_{1n} group. Like this grouping, the nodes of BDD can be grouped in different ways. A systematic approach of grouping is explained in the following section.

b. The Terminologies Related to the Proposed Algorithm:

The terminologies related to the proposed algorithm are defined as follows:

1. Two Y_{3n} groups are *touching* if they have at least one common node of BDD.
2. A set of *non-touching* Y_{3n} groups is maximal if it is not possible to add another Y_{3n} group without touching at least one Y_{3n} group of the set.
3. A set of Y_{3n} groups will be called as a *global maximal set* if its size is not less than the size of any other maximal set.

For a given BDD, as many Y_{3n} groups as possible are formed. A node can lie in more than one Y_{3n} group (by allowing overlapping). From these groups, a maximum number of groups can be selected such that each node lies in one Y_{3n} group only. Pairs of touching Y_{3n} groups are formed and a Boolean expression F is derived from these pairs. In the proposed algorithm, the Boolean expression is simplified in Sum of Product (SOP) form. The smallest SOP term is used to get a global maximal set.

Problem statement: To maximize the number of non-touching Y3 cells.

c. Proposed Algorithm for Global Maximal Set:

- For a given BDD, Y_{3n} groups as many as possible are formed by allowing overlapping of BDD nodes between them.
- Let, the set S contains all the Y_{3n} groups of the BDD. The Y_{3n} groups may or may not have common nodes, i.e. touching and non-touching. If the Y_{3n} groups are touching, then one Y_{3n} group is considered for selection.
- Suppose ($Y_{3n1,}$ Y_{3n2}) are touching Y_{3n} group, then only one Y_{3n} group is considered for selection but not both (Y_{3n1} OR Y_{3n2}). The non-touching Y_{3n} group is considered as such.
- Form the Boolean function F of all pairs of touching Y_{3n} groups in POS form.

Suppose ($Y_{3n1,}$ Y_{3n2}), ($Y_{3n3,}$ Y_{3n4}), etc. are the pairs of touching Y_{3n} groups,

$F = (Y_{3n1} + Y_{3n2}).(Y_{3n3} + Y_{3n4}) \ldots 1(a)$

Or

F = Y_{3n1} Y_{3n3} +Y_{3n2} Y_{3n3}+Y_{3n1} Y_{3n4} + Y_{3n2} Y_{3n4}… (1b)

- Simplifying the equation (1b) using Boolean algebra, the product term P can be obtained with minimum Y_{3n} groups, i.e. look for the product term with minimum number of literals where P is the product term of Sum of Product form (SOP) of F .
- Since $P \cup \overline{P} = S$, $\overline{P}$ (complement of P) denotes all Y_{3n} group of S not lying in P and is a maximal set of non-touching Y_{3n} group of S .
- This is true for every product term of F . Conversely every maximal set of non-touching Y_{3n} groups is a product term of F .

The pseudo code for the proposed algorithm is:

Begin
Form the set S of all Y3 cells
Determine all pairs of touching Y3 cells
For each touching pair of Y3 cells
Form an OR expression of its Y3 cells.
(Each OR expression is a factor)
End
Form F.
(F is a product of all these factors)
Store the first factor in F1
For each of the remaining factors
Multiply F1 with the next factor.
Expand and simplify using Boolean Rules
X + X = X, X.X = X , X (Y + Z) = XY + XZ, X + XY = XY
Store the resulting expression back in F1.
End
For each product term P in F1
Form $\overline{P}$ (P bar) which contains all Y3 cells of S not contained in P.
($\overline{P}$ (P bar) is a maximal set of non-touching Y3 cells)
End
End

d. Complexity Analysis:

Given a Boolean function of n variables, the number of nodes in the BDD can be exponential. The complexity of the algorithm depends on the number of nodes in the BDD. Consequently the algorithm cannot be executed in polynomial time in terms of the number of Boolean variables. However, it can be executed in a polynomial time in terms of number of BDD nodes, as the following analysis shows.

Let there be m nodes in the BDD. Every BDD node has two children. Every Y3 cell has one parent node and two children. Consequently, there can be at most one Y3 cell for each node. Therefore, there are O(m) Y3 cells. Let P1, P2, P3… Pm be these cells. Every Y3 cell can touch every other Y3 cell in the worst case. Therefore, every Y3 cell can give rise to O(m) factors. Since there are O(m) cells, there can be $O(m^2)$ factors in all. Let F be the POS (Product of Sums) of all these factors. Then F can have at most $O(m^2)$ factors. Multiply, expand and simplify these factors one by one. Let F_s denote the simplified sum of products obtained in this process.

Initially F_s is null. Place one factor in it and then multiply it with another factor. Expand and simplify and store it back in F_s. At any given time, there cannot be more than $O(m^2)$ product terms in it after it is simplified. This can be seen as follows: If a product form, say, P1 P2 P3, is a minimal term, it means that P1 P2, P2 P3, P1 P3, P1, P2 and P3 are not product terms in F_s. In view of this, the number of product terms in F_s cannot be more than the number of pairs of Y3 cells, which is $O(m^2)$. Since there are O (m^2) factors in F, the final F_s will be obtained after **$O(m^4)$** work giving at most $O(m^2)$ minimal sets.

e. Illustration of the Proposed Algorithm for Global Maximal Set:

The proposed algorithm is illustrated with the benchmark circuit CM152 as shown in Fig 7. All the non-terminal nodes are numbered as (n1-n15). The grouping procedure starts from top to bottom approach.

Reversed BDDs are mapped by LEAP cells, so that the number of cells can be minimized by increasing the Y_{3n} groups as many as possible, followed by Y_{2n} groups. The increase in the number of Y_{3n} groups is achieved by allowing overlapping of BDD nodes between the groups and using proposed algorithm to select the Y_{3n} groups.

In this example, group (n3, n7, n9) nodes as $Y_{3n1,}$ (n1, n5, n10) nodes as $Y_{3n2,}$ (n4, n8, n11) nodes as $Y_{3n3,}$ (n2, n6, n12) nodes as $Y_{3n4,}$ (n9, n10, n13)nodes as Y_{3n5}, (n11, n12, n14)nodes as Y_{3n6} and (n13, n14, n15)nodes as $Y_{3n7.}$ For the easy representation, these Y_{3n} groups can be called as $a_1, a_2, a_3, a_4, a_5, a_6, a_7$ groups. The maximum possible Y_{3n} groups are seven. The Y_{3n} groups $(a_1 a_5)$ are touching as they are sharing the node n9. Other touching Y_{3n} groups are $(a_2 a_5)$, $(a_3 a_6)$, $(a_4 a_6)$, $(a_5 a_7)$ and $(a_6 a_7)$.

By step 1 of the proposed algorithm for global maximal set, the set S is formed.

$$S = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\} \quad (2)$$

By step 2, the Boolean function of the Y_{3n} groups of all pairs of touching Y_{3n} groups in Product of Sum form (POS) is formed. The POS equation is represented as:

$$F = (a_1 + a_5).(a_2 + a_5).(a_3 + a_6).(a_4 + a_6).(a_5 + a_7).(a_6 + a_7) \quad (3)$$

Simplification of equation (3) by step 3, resulted in one product term with 6 literals, one product term with 5 literals, six product terms with 4 literals, and one product term with 2 literals. The simplified equation is given as:

$$F = a_1a_2a_3a_4a_5a_6 + a_5a_3a_4a_6 + a_2a_6a_4a_5 + a_5a_6 + a_5a_6a_1a_2 + a_1a_2a_3a_4a_7 + a_5a_3a_4a_7 + a_2a_6a_4a_7 + +a_1a_2a_6a_7 \quad (4)$$

The product term with less number of literals is selected from the eqn.(4) The smallest product term (with less number of literals) is a_5a_6. So,

$$P = \{a_5a_6\} \quad (5)$$

By step 4, the global maximal set $\overline{P}$ is obtained by removing the selected product term P from the set S so that

$$\overline{P} = \{a_1, a_2, a_3, a_4, a_7\} \quad (6)$$

The selected Y_{3n} groups are removed and the same algorithm is again applied to get optimal Y_{2n} groups. After removing Y_{2n} groups, the remaining nodes are Y_{1n} group. In this example, since all the nodes are grouped, so the algorithm terminates here without proceeding to Y_{2n} group selection.

RESULTS AND DISCUSSION

The input is the source file of the benchmark circuit in Berkeley logic interchange format (BLIF) (1992).The input source file is optimized using the SIS-1.2 Synthesis tool developed by Ellen *et al*., 1992. For optimization, rugged script command is used, where the process runs from the script library. The optimized circuit is applied as an input for the BDD generation package CUDD-2.4.2 developed by Somenzi (2009) and by using symmetric sifting to convergence variable ordering technique, BDDs and the reversed BDDs are obtained.

The selected benchmark circuits (combinational circuits) for conducting the experiments are MUX, 9SYM, PARKER, MAJORITY, XOR5, PM1, CM138, CM152, CM82, C17, PCLE, CM42 and RD53. The selected benchmark circuits BLIF file is applied as input to SIS tool. Table-1(a) gives the literal counts of the inputs, outputs, the sum of product (SOP) in the original circuit and the optimized combinational benchmark circuits. The optimal BDD size includes the non-terminal nodes, terminal nodes and a root node. From the optimal BDDs, reversed BDDs are obtained.

By applying the redundant pass transistor rules to the reversed BDD and simplified BDD are obtained and the results are given in Table-1(b).The levels are reduced at least 20% (rd 53 circuit) and at the maximum of 77% (cm152 circuit). When the levels get reduced, node count and transistor count will also reduce accordingly. Since one BDD node is equivalent to 2:1MUX and implemented by NMOS PTL circuits only, we need 3NMOS and 1PMOS transistor for 1 BDD node. For example, the Benchmark circuit CM152 has 15 nodes (2nd column, Table-1(b), results in 45NMOS+15PMOS (15x3NMOS and 15x1PMOS transistors) for (3rd column, Table-1(b)) transistor count.

Group the BDD nodes as many as possible by allowing overlapping into Y_{3n}, Y_{2n} groups. By applying the proposed algorithm the Y_{3n}, Y_{2n} groups are selected. Then Y_{3n}, Y_{2n} and Y_{1n} groups are mapped with LEAP cells. LEAP cell realization is done by using DSCH2 CAD tool. One inverter for each MUX cell is provided to make the modified Y cells mapping easier. The layouts are generated using Microwind tool for the 65 nm feature size. The simulations are carried out using BSIM4 model parameters and are analyzed for area, delay and power. Comparison of the implementation technique with reversed BDD benchmark circuits with only Y1 cells and other existing techniques is performed.

Comparison of the multiple output functions benchmark circuit PCLE with Y1 cells and LEAP cells implementation with the proposed algorithm for 65nm and 90nm feature size is carried out and the result is shown in Fig-8. It is clear that the output capacitance after LEAP cells implementation with the proposed algorithm is smaller compared to the actual circuit (with Y1 cells).It is also observed that the number of output nodes as well as the node count is reduced.

A comparison of the single output function benchmark circuits with Y1 cells and LEAP cells implementation with the proposed algorithm in terms of area and output capacitance is given in Table-2(a) and in terms of total power dissipation and delay in Table-2(b). The maximum improvement on area is 50% for the circuits CM152 and MUX. On the average 5-30% area improvement is observed for the remaining circuits. We also observed 75% improvement in output nodal capacitance for 9sym circuit. For the circuits like CM152, XOR5, PM1, PCLE, MAJORITY, C17, and CM42 both total power dissipation (50-90%) and delay are improved compared to Y1 cells. For some circuits like RD53 and CM138, power improvement is observed at the expense of increase in delay. The main reason for the area improvement is the reversed BDD technique and the proposed algorithm. It results in compact Y3 cells. The reduction in the output capacitance is due the less number of transistors connected to the node. The improvement in power dissipation is due the reduction in the output capacitance and switching activities.

Comparison of the reversed BDD based LEAP cells implementation based on the proposed algorithm with other existing techniques is

performed and the results in terms of area, delay and capacitance are tabulated. Table-3(a) gives the experimental results of transistor count and Table-3(b) gives the delay value for the benchmark circuits which is implemented using reversed BDD technique and technique reported by Marik *et al*., 2005 in the same environment (0.25µm technology).

It is observed that the transistor count by LEAP cells implementation with the proposed algorithm is certainly better than CMOS implementation. 50% improvement with mixed nY and cY techniques for most circuits. Reduction in levels ultimately reduces the transistor count. The propagation delay of the optimization technique shows about 80% improvement for all the circuits except PCLE (Table-3(b)) due to reduced critical path and reduction in transistor count and area. For the PCLE circuit the reduction is around 30% for all the three techniques. The delay improvement is observed because of the reversed BDD technique and the proposed algorithm compared to mixed CMOS/PTL implementation technique.

Table-4 gives the experimental results of delay value for the benchmark circuits RD53 and 9SYM implemented using LEAP cells with the proposed algorithm and Shelar *et al*., 2005 technique in the same environment for 0.13µm technology. With the proposed implementation procedure, it is observed that the level reduction and fewer nodes count which results in reduced critical path. Compared to forward BDD PTL implementation technique, the delay improvement is observed because of the reversed BDD technique and the proposed algorithm.

Table-5 gives the experimental results of the capacitance value for the benchmark circuit RD53 and XOR5 implemented using proposed implementation procedure and the technique reported by Zhuang *et al*., 1999 for 5V/0.6µm technology. Improvement around 90% is observed for both the circuits. Reduction in the transistor count and reversed BDD technique results in lower capacitance value.

Table-6(a) illustrates the area comparison of the proposed method and the existing methods Paul *et al*., 2010, Chaudhry *et al*., 1998 and Marik *et al*., 2004. The total number of Y cells of the X2 circuit using the proposed implementation procedure is lesser compared to the existing results. The reason for the reduction in total number of Y cells is due to the increase in Y3 cell count and which is achieved by the proposed algorithm.Table-6(b) shows that the improvement in power dissipation (mW) is more than 80% in all the 3 existing methods because of the reversed BDD based mapping algorithms and the reduction in the number of transistors. This reduces the capacitance, critical path and interconnects which in turn help the delay reduction (ps) of more than 90%. By using the proposed algorithm, the total number of the cells used can be reduced. This results in less area, less number of transistors and hence less parasitic capacitance. So Low Area/Power can be achieved. This is particularly suitable for large complex circuits where optimization is a must. The reduction in the total number of the cells used actually depends on number of Y_{3n}, Y_{2n} and Y_{1n} groups. By considering all possible groups, and by using the algorithm, the global optimization can be achieved.

Conclusion:

By increasing the number of Y3 cells, the overall cell count can be decreased. In this paper, an algorithm is proposed to increase the number of Y3 cells and reduce the overall cell count. The algorithm gives a global maximal set of Y3 cells. The proposed algorithm can also be used to get maximum number of Y2 cells (grouping of two BDD nodes) for the remaining nodes. The detailed discussion on the pseudo code of the proposed algorithm and complexity analysis of the algorithm are done. The proposed algorithm is illustrated with the Benchmark circuit CM152. Simulations results of the benchmark circuits are encouraging in terms of area, output capacitance and total power consumption compared to the existing techniques because of the reversed BDD technique and the proposed algorithm.

Table-1(a): Benchmark circuits before and after optimization using SIS-1.2 and BDD size for the Symmetrical sifting to convergence variable ordering technique

S.No	Benchmark Circuit	Input/ Output	Original(before optimization)	Optimized	Optimized BDD Size
			Literals	Literals	Number of nodes
1.	MUX	21/1	232	56	35
2.	9SYM	9/1	655	333	36
3.	PARKER	8/1	8	8	11
4.	MAJORITY	5/1	35	13	10
5.	XOR5	5/1	97	16	12
6.	PM1	16/13	158	52	59
7.	CM138	6/8	45	31	27

8.	CM152	11/1	54	32	18
9.	CM82	5/3	55	24	22
10.	C17	5/2	23	10	11
11.	PCLE	19/9	136	78	52
12.	CM42	4/10	46	34	32
13.	RD53	5/3	182	38	28

Table-1(b): Levels, Node count and Transistor count of Benchmark circuits before and after removing redundancies

S.No	Benchmark Circuit	Actual circuit			After removing redundancies		
		Levels	Node count	Transistor count	Levels	Node count	Transistor count
1.	MUX	21	32	96N+32P	5	16	48N+16P
2.	9SYM	9	33	99N+33P	8	31	93N+31P
3.	PARKER	8	8	24N+8P	7	7	21N+7P
4.	RD53	5	23	69N+23P	4	21	63N+21P
5.	MAJORITY	5	7	21N+7P	4	6	18N+6P
6.	XOR5	5	9	27N+9P	4	7	21N+7P
7.	PM1	16	44	132N+44P	12	39	117N+39P
8.	CM138	6	17	51N+17P	5	16	48N+16P
9.	CM152	11	15	45N+15P	3	7	21N+7P
10.	CM82	5	17	51N+17P	4	15	45N+15P
11.	CM42	4	20	60N+20P	3	18	54N+18P
12.	C17	5	7	21N+7P	4	5	15N+5P
13.	PCLE	19	41	123N+41P	10	32	96N+32P

Table-2(a): Area and output capacitance of the benchmark circuits with Y1 cells and LEAP cells implementation with proposed algorithm

S.No	Benchmark Circuit	With Y1 cells		LEAP cells implementation with our proposed algorithm	
		Area (μm^2)	Output Capacitance(fF)	Area (μm^2)	Output Capacitance(fF)
1.	MUX	100x19	6.45	50x19	3.49
2.	9SYM	103x11	6.14	94x11	1.48
3.	PARKER	26x8	4.47	23x9	2.50
4.	MAJORITY	23x9	2.46	20x9	2.12
5.	XOR5	29x9	2.77	23x10	2.56
6.	PM1	137x23	8.35	121x21	7.67
7.	CM138	53x11	4.13	50x12	4.03
8.	CM152	47x12	3.79	23x10	2.67
9.	CM82	53x10	3.93	47x13	3.72
10.	PCLE	127x22	7.94	100x21	6.60
11.	RD53	72x11	4.74	66x16	4.64

Table 2(b): Total power dissipation and delay of the benchmark circuits with Y1 cells and LEAP cells with proposed algorithm

S.No	Bench mark Circuit	With Y1 cells		LEAP cells with the proposed algorithms		% imp (power)
		Total power Dissipation	Delay (ps)	Total power Dissipation	Delay (ps)	
1.	CM152	0.551mW	35	0.826 µW	16	99.85
2.	RD53	0.225mW	22.5	5.752 µW	56.5	97.44
3.	XOR5	7.062 µW	49	1.419 µW	33.5	79.90
4.	9SYM	16.085µW	42.5	4.250 µW	45	73.5
5.	PM1	12.301µW	5006.5	3.274 µW	90	73.38
6.	PCLE	10.946µW	233.5	3.487 µW	90.6	68.14
7.	C17	5.290 µW	23.5	1.897 µW	18	64.13
8.	MAJORITY	3.555 µW	21.5	1.599 µW	21.5	55.02
9.	CM42	12.031µW	332	5.926 µW	34	50.77
10.	CM138	6.410 µW	21	4.03 µW	29.5	37.12

Table 3(a): Transistor count comparison of the proposed method with Marik *et al*., 2005

S.NO	Benchmark Circuit	Proposed	Marik *et al*., 2005 Transistor count #					
			CMOS (a)	(Mix) nY(b)	(Mix) cY(c)	%Imp (a)	%Imp (b)	%Imp (c)
1.	CM152	28	122	32	40	77	13	30
2.	CM82	60	84	90	90	29	33	33
3.	MUX	64	220	86	106	71	26	40
4.	PCLE	128	288	212	212	56	40	40

Table 3(b): Delay comparison of the proposed method with Marik *et al*., 2005

S.NO	Benchmark Circuit	Proposed	Marik *et al*., 2005 delay(ns)					
		Delay(ns)	CMOS (a)	(Mix) ny(b)	(Mix) cY(c)	%Imp (a)	%Imp (b)	%Imp (c)
1.	CM152	0.0945	1.02	0.64	0.67	91	85	86
2.	CM82	0.094	0.69	0.8	0.71	86	88	87
3.	MUX	0.24	1.8	1.42	1.53	87	83	84
4.	PCLE	1.51	2.35	2.16	2.2	36	30	31

Table 4: Delay comparison of the proposed method with Shelar *et al*., 2005

S.No	Bench mark Circuit	Proposed Delay(ps)	Shelar *et al*., 2005 delay (ps)			
		Delay (ps)	SPICE (delay) (a)	timing analysis (delay)(b)	% Imp (a)	% Imp (b)
1	RD53	87	196.10	205.54	56	58
2	9SYM	500	462.40	618.50	-8	19

Table 5: Capacitance comparison of the proposed method with Zhuang *et al*., 1999

S.No	Benchmark Circuit	Proposed	Zhuang, *et al*.,1999 C value (pf)			
		C value (pf)	PTM (a)	CMOS (b)	% Imp (a)	% Imp (b)
1	RD53	0.0273	0.7267	1.070	96.24	97.44
2	XOR5	0.0220	0.395	0.353	94.43	93.76

Table 6(a): Area comparison between the proposed method and the existing methods Paul *et al*., 2010, Chaudhry *et al*., 1998 and Marik *et al*., 2004 for the X2 benchmark circuit

S.No	Method	#Y3,#Y2,#Y1 Cells	%Imp
1	Proposed	7,6,3	-
2	Paul *et al*., 2010	3,11,8	27.27
3	Chaudhry *et al*., 1998	2,12,12	38.46
4	Marik *et al*., 2004	9,12,3	33.33

Table 6(b): Performance comparison between the proposed method and the existing methods Paul *et al*., 2010, Chaudhry *et al*., 1998 and Marik *et al*., 2004 for the X2 benchmark circuit

S.No	Method	Power (mW)	Delay (ps)	%Imp Power (mW)	%Imp Delay (ps)
1	Proposed	0.113	125	-	-
2	Paul *et al*., 2010	0.74	1250	84.72	90
3	Chaudhry *et al*., 1998	0.81	1140	86.04	89.03
4	Marik *et al*., 2004]	0.77	1420	85.32	91.19

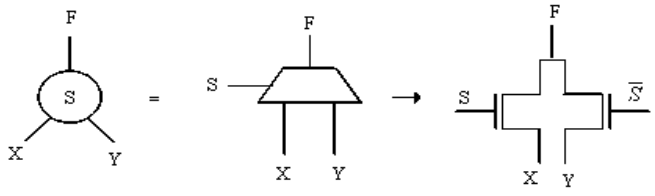

Fig. 1: A BDD Node as a Multiplexer and a PTL Circuit

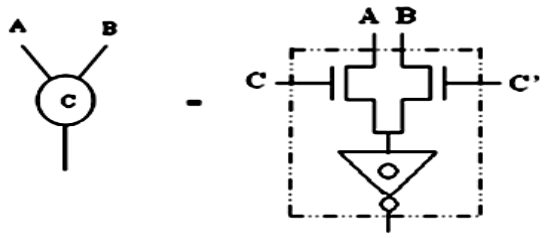

Fig. 2: (a) Y_{1n} group (b) Y1 Cell

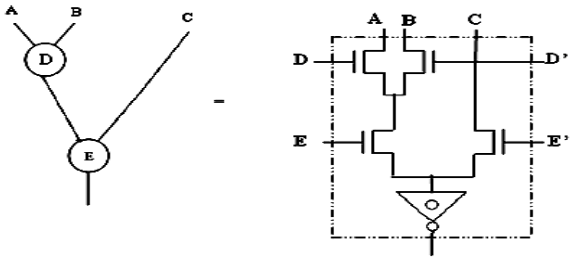

Fig. 3: (a) Y_{2n}group (b) Y2 Cell

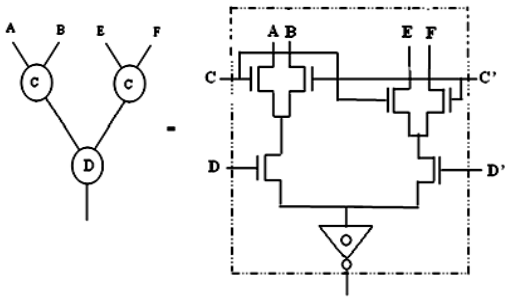

Fig. 4: (a) Y_{3n}group (b) Y3 Cell

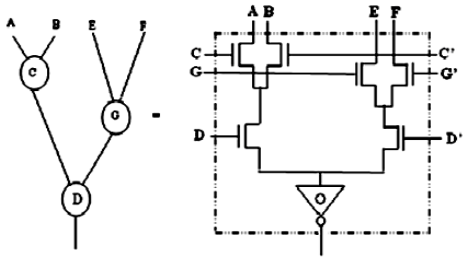

Fig. 5: (a) Modified Y_{3n}group (b) ModifiedY3 Cell

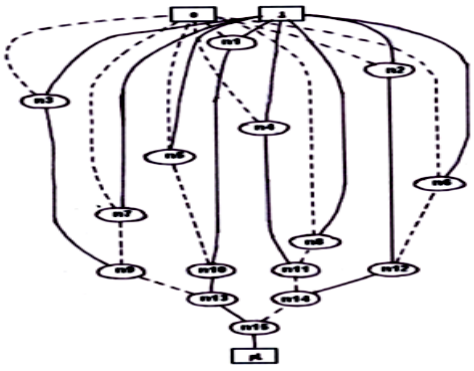

Fig. 6: CM152 circuit reversed BDD

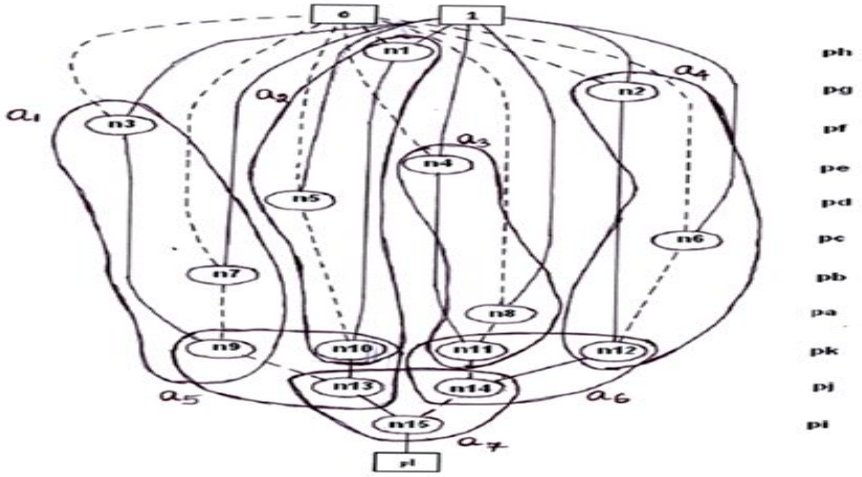

Fig. 7: CM152 circuit reversed BDD with Y_{3n} groups

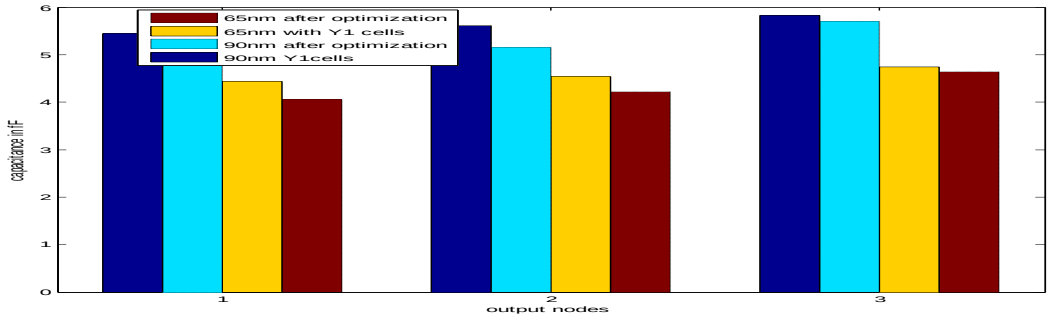

Fig. 8: Output nodal capacitance of the multiple output benchmark circuit PCLE

REFERENCES

Bryant, R.E., 1986. Graph Based Algorithms for Boolean Function Manipulation, IEEE Transactions on Computers, c35-8: 677-691.

Malik, S., A. Wang, R.K. Brayton, A. Sangiovanni-Vincentelli, 1998. Logic Verification using Binary Decision Diagrams in a Logic Synthesis Environment, IEEE International Conference on Computer-Aided Design, (ICCAD-88), 6-9, DOI: 10.1109/ICCAD.1988.122451.

Justin, E., Harlow, Franc Brglez, 2001. Design of Experiments and evaluation of BDD ordering Heuristics, International Journal of Software Tools for Technology Transfer, 3(2): 193-206.

Friedman, S.J., K.J. Supowit, 1990. Finding the Optimal Variable Ordering for Binary Decision Diagrams, IEEE Transactions on Computers, 39(5): 710-713.

Kebschull, U., E. Schubert, W. Rosenstiel, 1992. Multilevel Logic Synthesis Based on Functional Decision Diagrams, European Design Automation Conference, pp: 43-47.

Drechsler, R., D. Sieling, 2001. Binary Decision Diagrams in Theory and Practice, International Journal on Software Tools for Technology Transfer (STTT), pp: 112-136.

Lai, Y.T., Y.C. Jiang, H.M. Chu, 2005. BDD Decomposition for Mixed CMOS/PTL Logic Circuit Synthesis, IEEE International Symposium on Circuits and Systems (ISCAS 2005), 6: 5649-5652.

Shelar, R.S., S.S. Sapatnekar, 2001. Recursive Bipartitioning of BDDs for Performance Driven Synthesis of Pass Transistor Logic Circuits, IEEE/ACM International Conference on Computer Aided Design, pp: 449-452.

Shelar, R.S., S.S. Sapatnekar, 2005. BDD Decomposition for Delay Oriented Pass Transistor Logic Synthesis, IEEE Transactions on Very Large Scale Integration Systems (VLSI), 13(8): 957-970.

Buch, P., A. Narayanan, A.R. Newton, A. Sangiovanni-Vincentelli, 1997. Logic synthesis for Large Pass Transistor Circuits, Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp: 663-670.

Kao, C.C., 2011. BDD-based Synthesis for mixed CMOS/PTL Logic, International Journal of Circuit Theory and Applications, 39(9): 923–932, DOI: 10.1002/cta.676

Thangavel Bhuvaneswari, Vishnuvajjula Prasad, Ajay Kumar Singh, Chinnaiyan Senthilpari, 2013. Performance Analysis of Reversed Binary Decision Diagrams Pass-Transistor logic Synthesis, International Journal of Circuit Theory and Applications, 41(8): 844-853, DOI: 10.1002/cta822

Zimmermann, R. and W. Fichtner, 1997. Low Power Logic Styles: CMOS versus Pass Transistor Logic, IEEE Journal of Solid State Circuits, 31(6): 1079-1090.

Yano, K., Y. Sasaki, K. Rikinoand, K. Seki, 1996. Top-Down Pass Transistor Logic Design, IEEE Journal of Solid-State Circuits, 31: 792-803.

Yano, K., T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi and A. Shimizu, 1990. A 3.8ns CMOS 16x16-b Multiplier using Complementary Pass-Transistor Logic, IEEE Journal of Solid-State Circuits, 25(2): 388-393.

Paul, G., A. Pal, B.B. Battacharya, 2010. A Novel Technology Mapping Technique for BDD-based Circuits using LEAP Cells, 2010 International Symposium on Electronics System Design, 135-140, DOI 10.1109/ISED.2010.34

Chaudhry, R., T-H. Liu, A. Aziz, J. Burns, 1998. Area-oriented Synthesis for Pass-Transistor Logic, International Conference on Computer Design (ICCD'98), 160-167.

Hsiao, S-F., J-S. Yeh, D-Y. Chen, 2002. High-performance Multiplexer-based Logic Synthesis Using Pass-transistor Logic, VLSI Design, 15(1): 417-426.

Marik, M., A. Pal, 2004. Energy Aware Logic Synthesis and Technology Mapping for MUX based FPGAs, 17th International Conference on VLSI Design, pp: 73-78.

Zhuang, N., M.V. Scotti, P.Y.K. Cheung, 1999. PTM: Technology Mapper forPass-Transistor Logic, IEE Proceedings–Computers and Digital Techniques, 146(1).

Cho, G.R. and T. Chen, 2004. Synthesis of Single/Dual-rail Mixed PTL/Static Logic for Low-Power Applications, IEEE Transactions on Computer-Aided Design of Integrated Circuits and System, 23(2): 229-24.

Wille, R., R. Drechsler, 2009. BDD-based Synthesis of Reversible Logic for Large Functions, 46th Annual Design Automation Conference, (DAC'09), DOI:10.1145/1629911.162998413-19

University of California Berkeley: Berkeley Logic Interchange Format (BLIF) (July28, 1992) http://www.ece.cmu.edu/~ee760/760docs/blif.pdf

Ellen, M., Sentovich, Kanwar Jit Singh, Luciano Lavagno, Cho Moon, Rajeev Murgai, Alexander Saldanha, Hamid Savoj, Paul R. Stephan, Robert K. Brayton, Alberto Sangiovanni-vincentelli,1992."SIS: A System for Sequential Circuit Synthesis", www.eecs.berkeley.edu/IPRO/Software/.../sis1.2.html

Somenzi, F., 2009. "CUDD:CUDecision Package Release 2.4.2", University of Colorado at Boulder.
http://vlsi.colorado.edu/~fabio/CUDD/node1.html
DOI: 2009-02-20