E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

DOI: 10.32377/cvrjst1710

VLSI Implementation of Seed Transistor for Super Gate Design based on Grid based Transistor Network Generation

T. Subha Sri Lakshmi
Asst. Professor, CVR College of Engineering/ECE Department, Hyderabad, India
Email: rupashubha@gmail.com

Abstract: **In VLSI digital design, the Propagation Delay, Power Dissipation and Area of circuits are strongly related to the number of transistors which are present on an IC. Hence transistor optimization is special interest when designing digital integrated circuits. Therefore, efficient algorithms are used to generate optimized transistor networks, which are quite useful for designing digital integrated circuits (ICs). Several methods have been proposed for generating and optimizing transistor networks. Most Traditional solutions are based on factoring Boolean expressions in which only Chain - Parallel (CP) arrangement of transistors can be obtained from factored forms. Whereas grid-based methods are able to find CP and also non - CP (NCP) arrangements with potential reduction of transistor count. This method is an effective way of improvising VLSI circuits. In this an efficient algorithm is proposed i.e. Novel (Seed) method. It is automatically generates networks with minimal transistor count starting with irredundant sum of products (ISOPs) as inputs. Novel method is able to deliver both CP switching networks and NCP switching network arrangements, which improves VLSI circuit's performance in terms of area, power and delay. All the network circuits are implemented in ASIC Cadence by using 45nm and 90nm technology with GPDK libraries. By using Cadence Virtuoso tool, it can obtain schematics of design and its test bench, power analysis, SPICE simulation and its simulation waveform.**

Index Terms: **Transistor network, CMOS circuits, NMOS circuits, Super Gate design, Seed method, ISOP, CP and NCP.**

I. Introduction

Logic synthesis plays a major role in design automation. A logic function can be represented by a binary decision diagram (BDD). Here, a technique is proposed to construct a BDD whose nodes can be implemented by CMOS logics and Pass-Transistor Logics (PTL) in a cell library. The conventional synthesis flow needs three cell libraries: CMOS cell library, Pass Transistor Logics cell library, and CMOS remapping pattern. To simplify the synthesis flow, the logic function is decomposed into two kinds of functions and map them to Pass Transistor Logics and CMOS cells, respectively. The cell library contains high speed cells and low power cells. The experimental results in better performance and use less area than conventional CMOS technology mappings [1]. Dynamic circuits are widely used in today‘s high-performance microprocessors for obtaining timing goals that are not possible using static CMOS circuits. Currently, no commercial tools are able to synthesize dynamic circuits and therefore their design is either completely done by hand or aided by proprietary in-house design tools. This paper describes methodologies and tools for the design and synthesis of dynamic circuits, including general monotonic circuits, which consist of alternating low-skew and high-skew logic gates that may both contain functionality. Synthesis results show standard domino, dynamic-static domino, monotonic static CMOS, zipper CMOS, and footless domino and clock-delayed domino circuits to have average speed improvements of 1.57,1.66,1.47,1.71 and 1.60 times over static CMOS, respectively [1]. This paper is organized as follows. Section II gives an overview of existing system and its arrangements. Section III presents the Novel Method which minimizes the transistor count. Section IV presents the super gate design with few examples in different technologies. Section V presents and demonstrates the efficiency of Novel Method by providing experimental results in terms of transistor count, area minimization and power dissipation of the network. Finally the conclusion is drawn in Section VI.

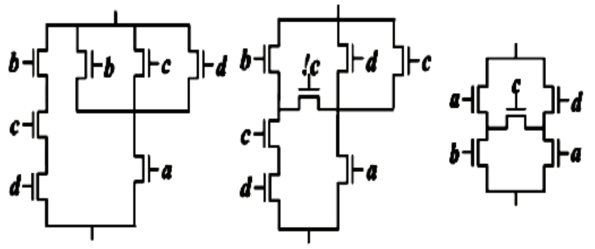
II. Existing System

To generate an optimized switch network circuits various methods are designed and implemented. In which the frequent traditional way of approach to achieve Chain-Parallel (CP) associations of transistors by using factorization process and CP net can be obtained from factored form. In few more cases grid based methods are used to implement the CP and also Non-CP (NCP) arrangements [10, 6] to get least transistor count. Even with the defects of earlier literature work, there are so many improvement techniques to get an optimized switch net which gets a minimum no. of transistors [2]. The main disadvantage of earlier work is usage of more number of transistors, more power and more area.

III. Novel Method

In this an efficient algorithm is proposed i.e., Novel Method (Seed). It automatically generates networks with minimal transistor count starting with irredundant sum of products (ISOPs) as inputs. Novel method is able to deliver both CP switching networks [3] and NCP switching network arrangements, which improves VLSI circuit's performance in terms of area, power and delay. All the network circuits are implemented in ASIC Cadence by using 45nm & 90nm technology with GPDK libraries. By using Cadence Virtuoso tool, can obtain a schematics of design and its test bench, power analysis, SPICE simulation and its simulation. This method introduces a new algorithm for Boolean factoring. The proposed approach is based on a novel synthesis paradigm, functional composition, which performs synthesis by associating simpler sub-solutions with minimum cost. The method constructively controls characteristics of final and intermediate functions, allowing the adoption of secondary criteria other than the number of literals for optimization. This multi objective factoring algorithm presents interesting features and advantages when compared to previous works [3]. For the mentioned below Boolean expression, CP network is designed using factorization method shown in Fig. 1(a) which consist of 7 switches. Active grid analysis techniques, consecutively, endow with the NCP solution shown in Fig. 1(b), and also with 7 switches. However, the finest network design consists of only 5 switches as shown in Fig.1(c), is not available by any of these techniques. [4]

Example: F = a ・ b + a ・ c + a ・ d + b ・ c ・ d



(a) (b) (c)

Figure 1. Transistor networks corresponding to (1). (a) CP solution from factored form. (b) NCP from existing graph-based generation methods (c) Optimum NCP solution.

A Switch is a component which is designed by having one control input and two contact inputs. Control input specifies the relation or connection between two contact inputs as shown in Fig. 2(a). In general switch represents an ideal MOS transistor. In this paper switch is replaced with transistor. When two transistors are connected in chain the flow of electrons is shown in Fig. 2(b), represents an AND operation, whereas parallel network seen in Fig. 2(c), which leads to OR operation. Example of CP network is shown in Fig. 2(d). An NCP transistor network cannot be designed by placing the transistors or contacts either in chain and/or in comparable. [3,4]

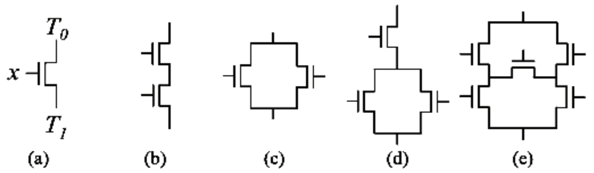


Figure 2. Represents of (a) MOS Transistor (b) chain (c) Comparable (d) CP (e) NCP.

The Novel (Seed) Method gives a reduced switch network which consists of very less number of transistors with the help of using Sum of Products technique. This method comprises 2 blocks i.e., Seed finding and Switch net circuit. Kernels are mainly used as a grid structures and they are help in designing well-organized CP and NCP transistor networks. Finally from the first block design get, inequitable nets which carry out switch sharing, and results in a solitary net which represents F. The obtained solitary net circuit consists of very less number of transistors compared to general Boolean expression. Various Boolean expression are taken into consideration and tested them to achieve few improvements towards area and power Flow chart of Novel (Seed) method is shown in Fig.3.

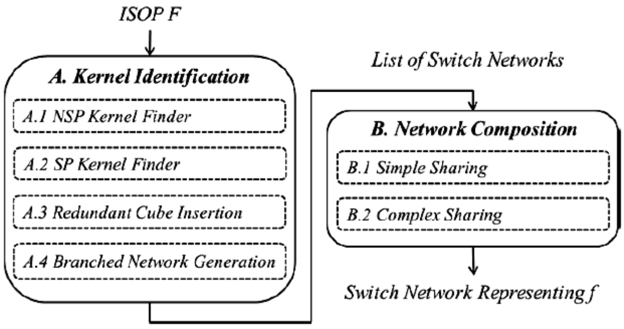


Figure 3. Execution Flow of the Proposed Method

A. *Seed Identification*

The identification module of Seed is divided into four steps. In the process of kernel identification module, the possible CP and NCP networks are searched by using Seed, which is known as intermediate data structure.If Ui ^ Uj_= Ø, then only existence of edge (Ui,Uj) Є G, i_ = j is possible. Here, edge g is labeled as Ui^Uj. To perform logic sharing using Seed structure, the determination of relationship among cubes of F is possible. All steps of Seed identification module are used to extract Seeds from F because of which transistor count optimization is possible.[5]

In Seed identification module, each step is used to find transistor networks which are used to represent the sub-functions of the target function f. To obtain the optimized NCP networks from the input ISOP F, the NCP kernel finder steps are used. Depending on the availability of transistor network, the circuit is achieved which are removed from F by using cubes. This process leads to a simpler ISOP F1.In CP kernel finder step method; the CP networks can be searched by using the input F1.In the same way, CP network cubes are eliminated from F which results in F2. To generate NCP or CP networks, rest of the cubes of F2 were not useful. So, to find NCP arrangements with repeated paths, and finally the repeated paths are added into the kernels. Hence the repeated paths are eliminated from F2 which results in F3. The final step generates group of transistor networks, which consists of comparable paths corresponding to cubes from F3. The final output of the Seed identification module generates a list of transistor networks, which is treated as output. [5]

Non Chain-Comparable Seed Finder – The ISOP form of a Boolean function is assumed to be f which is given by F = c1+c2+…….+cn, where the number of dices F is denoted by n. The combinations of 4 dices are taken simultaneously among n dices to identify NCP Kernels. By adding those 4 dices, an ISOP L can be obtained, which is known as sub-function of f. L is used to get a Seed with 4 vertices. To make sure that the generated Seed results in NCP transistor network, [5] 2 rules need to be verify i.e.,

First Rule: The set of edges which are connected to the vertex u Є U is assumed to be G_v. The cube is also known as vertex. The literals from u should be shared through edges g Є G_v for each cube u Є U.

Second Rule: The obtained kernel from L should be isomorphic of same form as shown in Figure 4.The type of grid stencil is known as NCP Kernel. To map NCP Seed to a transistor network, the edge swapping technique can be applied over Seed 3 edges. The generic NCP Seed is assumed as shown in Fig.4. The edge is shifted to g_4 place, g_4 is shifted to g_3 place, g_3 is shifted to g_2 place in order to map the kernel into a network. In order to get the network shown in below figure 4, such type of rearrangement process is applied. This procedure is helpful to make sure that each part of the network denotes a Dice from the sub function l. [5]

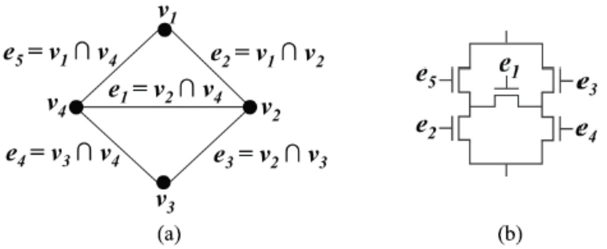


Figure 4. (a) NCP Seed template (b) Resulting Transistor network

A dice combination produces set of all possible 4 combinations of dices. Let it be'd'. Assume an ISOP which has n dices. The time complexity of get dice combinations is assumed to be O (n4). Each Seed contains one of 4 dices possible combinations. The resultant kernel sub routine's time complexity function is assumed to be O (n^2p). As 4 dices are combining simultaneously, the resultant function is assumed to be O (p). [5]

First Rule: It can be tested by using complexity of time and also it is similar to the obtained Seed subroutine. The grid with 5 edges only obeys second rule. At time O (1) this type of list will be done. The edge reordering subroutine process will be executed only if both rules are satisfied. This process should be done at a constant time to obtain transistor networks; the found NCP transistor network will be added. This process is done in constant time O (n4). As shown in Fig. 5, the Seed is going to be obtained for every combination of 4 dices. The resultant Seed sub routines time complexity function is assumed to be O (n^2p), where p is no. of variables in F. As 4 dices are taken into consideration, the complexity function can be reduced to O (p).

Chain-Comparable Seed Finder – let ISOP form F1 which is used to represent the dices, which were not used to generate transistor network in the NCP Seed finder step. The identification process of CP Seeds will be done by combining 4 n dices from F1. Then 4 vertices Seed is obtained. To check whether the resultant Seed is a valid CP network or not, both rules (first and third rule) must be satisfied. The resultant Seed must be same form to the grid. This type of grid stencil is termed as CP Seed. To obtain the transistor network, some changes should be performed on Seed by the CP Seed finder step. The first step includes that the kernel edges should be mapped to a supplementary template grid as shown in figure 6. Later by applying edge ordering sub routine one supplementary template grid in order to get a transistor network as shown in Fig. 6. [5]

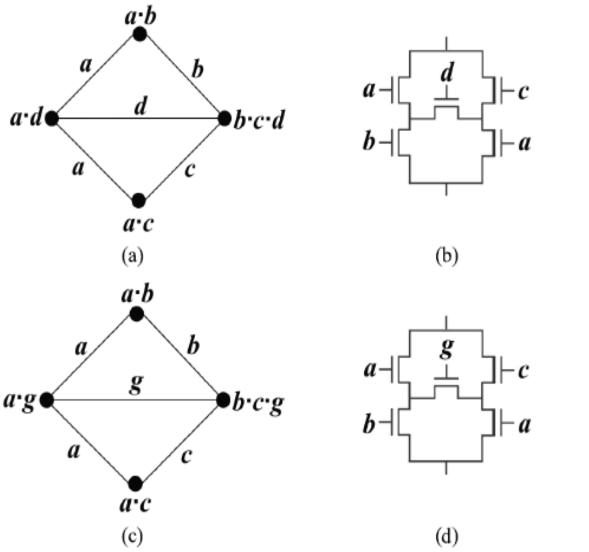


Figure 5. NCP Seeds (a) M_2 and (c) M_3, obtained from (4). Corresponding transistor networks (b) T_2 and (d) T_3

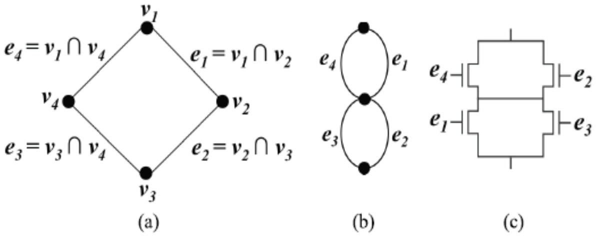


Figure 6. (a) CP Seed template (b) Supplementary Template grid (c) Resulting Transistor network.

Third Rule: Obtained Seed need to be isomorphic to the grid as shown in the Fig. 6(a). Such grid template is defined as CP Seed. To obtain the transistor network CP seed finder step should perform some transformations over the seed. Seed edges are given to supplementary template which results in transistor network by pertaining the edge reordering subroutine ended the supplementary template grid as shown in Fig. 6(c).

Superfluous dice Insertion – In most of the scenario NCP structures are arranged with the help of superfluous dice method in place of using CP structures as shown in the Fig. 7. Even dice is not present in NCP and CP networks; the superfluous dice insertion step tries to construct NCP Seeds by adding the left over dices. Let S be an ISOP which represents the Boolean function s. A dice d is superfluous if S + d = s. Transistor network which represents an ISOP function s is taken into the consideration [7]. Designing or Functioning of superfluous dice d in a circuit defined as superfluous path. Superfluous path gives best performance in logic sharing.

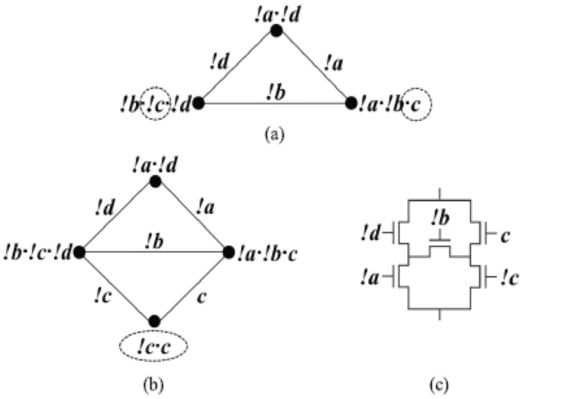


Figure 7. (a) Grid D1 (b) NCP Seed with Superfluous dice (c) Resulting Transistor network.

Stem Net Creation – During a net is created or found the dices are removed from ISOP S. Logic Sharing plays a prominent role, even though dices are not present in the designed nets. The left over dices in S3 forms as a unique transistor net [7]. Therefore stem net creation step converts each remained dice S3 into a stem of transistors which are connected in chain. Three left over dices are present to implement. Final net is shown in Fig. 8.

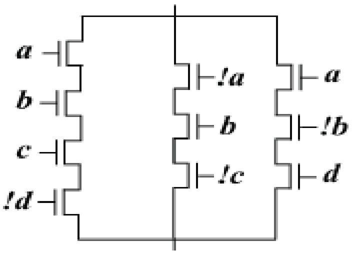


Figure 8. Stem Transistor Net

B. Network Composition

During Seed Identification module, net composition modules obtain the function F and a record of biased transistor networks S is produced. By executing logic sharing on the nets which are derived from S in a repeated and continuous process. In the target net to remove superfluous transistors, trouble-free and intricate sharing is used. The trouble-free sharing and intricate sharing are discussed in the below section.

Trouble-Free Sharing – Edge sharing technique is realized via trouble-free sharing method. Mostly the method negotiates the transistor net exploring for equivalent transistors, and all the transistors are organized with the help of literals. Net is reconstructed in which 1 universal knot is present between the corresponding transistors are available [8]. In general case, the corresponding transistors must be interchanged in the nets in order to utilise the universal knot. In this case only one transistor is utilised and used which leads to a superfluous transistors. After doing trouble-free sharing, the circuit performance of the net will be verified to make sure for a precise target function. This process is performed continuously over the net until there are no possible transistors left over for carve up the universal knot.

Intricate Sharing – Intricate sharing step gets an in-complete net which is provided by the stem net creation step and strives to do further optimizations. As explained in earlier cases i.e., trouble-free sharing step, when a universal knot [8] is available the transistor starts to enable its sharing mechanism to operate continuously with the subsequent knots. When the net is too large it will be difficult to find the universal knot which leads to further some more iterative steps. Therefore, in bid to perk up these sharing mechanisms, straightforward CP transistor compressions are used as shown in Fig. 9.

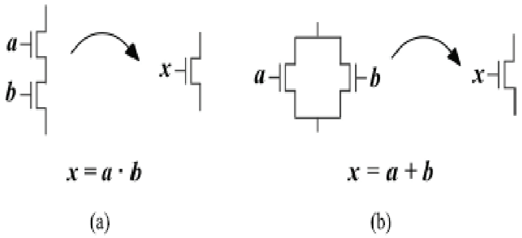


Figure 9. (a) Chain Transistor Compression (b) Comparable Transistor Compression

IV. Super Gate Design Using Seed Method

Four Boolean expressions are considered up to 6 literals. All the four Boolean expressions are designed using three different logic styles i.e., CMOS logic style, NMOS logic style and finally with Super Gate logic style by using Novel (Seed) Method. The transistor level circuits are designed and implemented using Cadence Design Suite 6.1.6 version with GPDK 45nm and 90nm technology libraries. SPICE simulations are carried out for the expressions in all the logic styles for the expression. Post-Layout Simulations are carried out during implementation phase. The below Four Boolean expressions are represented with P and N net's and only N net's.

Expression 1: F = w.x + w.y + w.z + x.y.z

This expression contains 4 literals with logic AND & logic OR operation. The expression is designed with a) CMOS style which contains both P and N net's in the circuit b) NMOS style which contains only N net's c) Super Gate Design style (using Seed Method) which contains only N net's with reduced path design & number of transistors compared to NMOS and CMOS style. Fig. 10 gives schematic structures in three styles and simulation waveform.

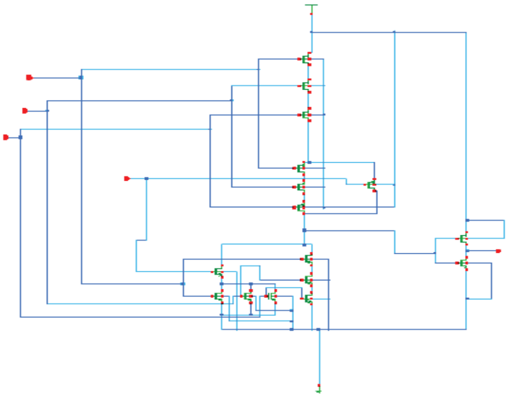


Figure 10. (a) Schematic structure of expression 1 in CMOS design style

CMOS design style of the expression 1 contains 8 nmos and pmos transistors for the reduced network by using Boolean algebra.

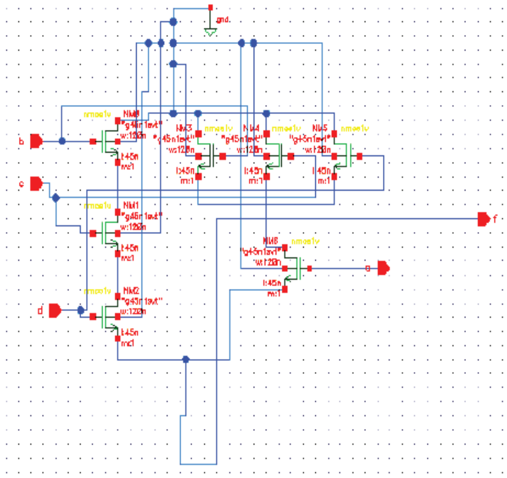


Figure 10. (b) Schematic structure of expression 1 in NMOS design style

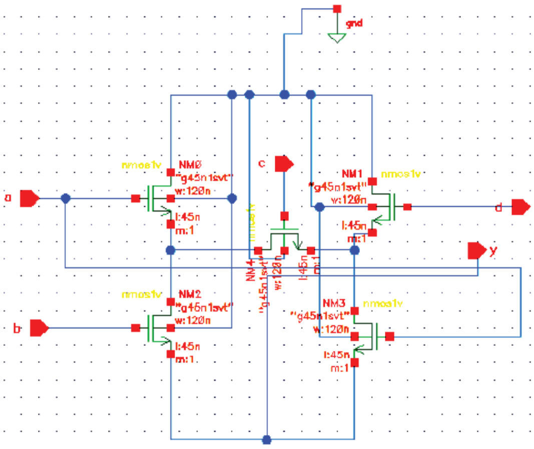


Figure 10. (c) Schematic structure of expression 1 in Super Gate style

Expression 1 by designing using Super Gate design consists very minimum number of transistors in terms of SP network.

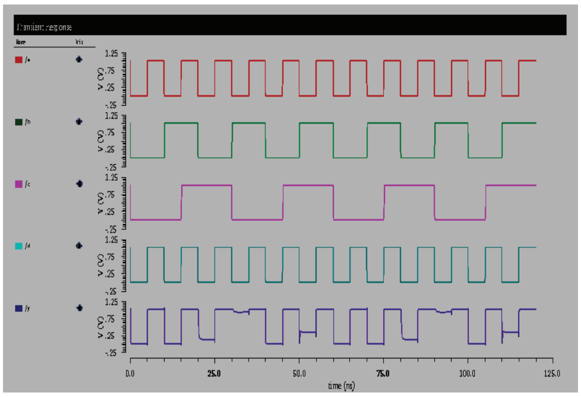


Figure 10. (d) Simulation waveform of Expression 1.

In the above simulation x-axis is defined by time in terms of nano seconds (ns) and y-axis defined by voltage in terms of volts (v).

Expression 2: F = w.x + w.y.s + z.s +x.y.z

This expression contains 5 literals namely w, x, y, z, s. Fig. 11 gives schematic structures in three styles and simulation waveform.

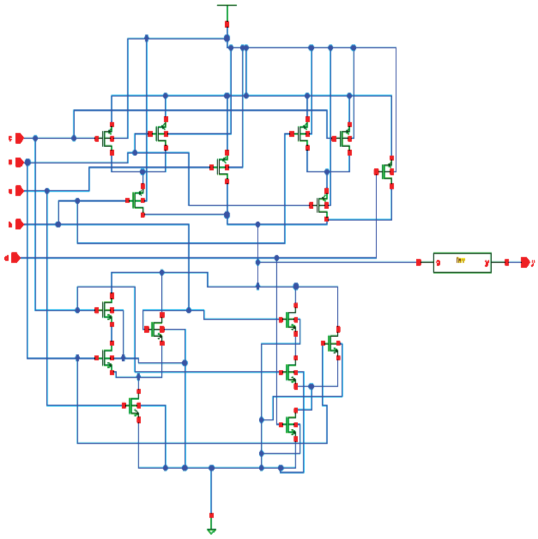


Figure 11. (a) Schematic structure of expression 1 in CMOS design style

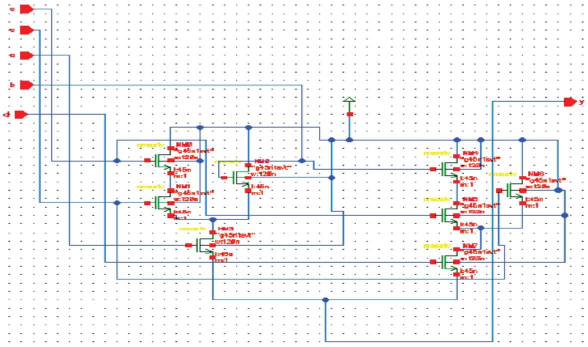


Figure 11. (b) Schematic structure of expression 1 in NMOS design style

Expression 2 by designing using CMOS design consists of very maximum number of transistors in terms of Boolean algebra and it contains 9 nmos and pmos transistors. In NMOS design style it only consists of 7 nmos transistors.

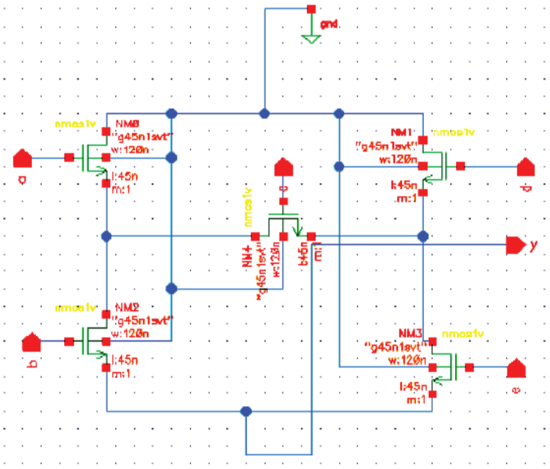


Figure 11. (c) Schematic structure of expression 2 in Super Gate design style

Expression 1 by designing using Super Gate design consists very minimum number of transistors in terms of SP network.

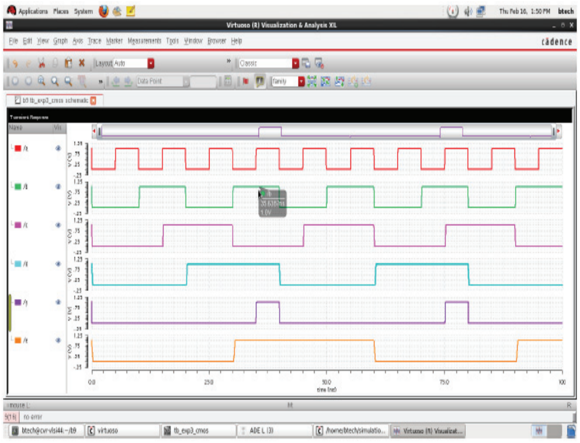

Applications Places System
Thu Feb 16, 1:50 PM btech
Virtuoso (R) Visualization & Analysis XL
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cadence
Transient Response
0.0
250
500
750
100
time (ns)

Figure 11. (d) Simulation waveform of Expression 2.

In the above simulation x-axis is defined by time in terms of nano seconds (ns) and y-axis defined by voltage in terms of volts (v).

Expression 3: F = w.’x’.y’.z’ + w’.x.z + w’.x.y +w.x’.y + w.x’.z + w.x.y’.z’ + x.y.z

This expression contains 4 literals namely w, x, y, z. The function is expressed in terms of its complements finally inverter is needed to be added at output side. The entire expression is minimized by using SOP or POS form, the reduced Boolean expression is designed by using various design styles. Fig. 12 gives schematic structures in three styles and simulation waveform.

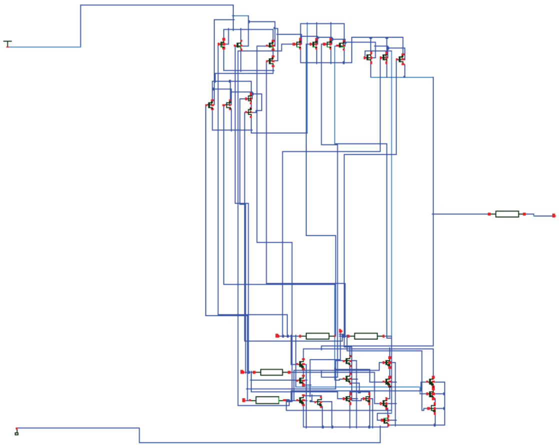


Figure 12. (a) Schematic structure of expression 3 in CMOS design style

Expression 3 CMOS design contains 5 inverters, 15 pmos and 15 nmos transistors. Complexity wise it has some difficulties to perform layout design.

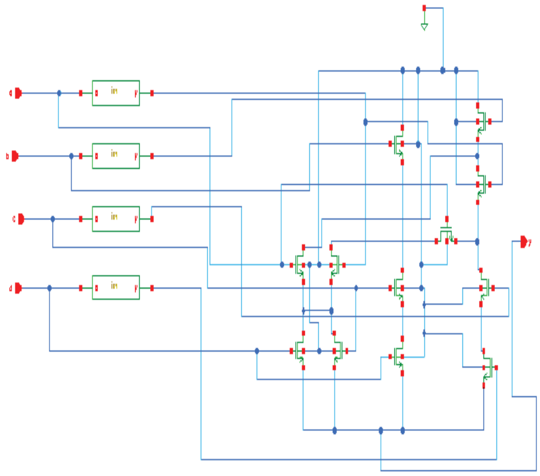


Figure 12. (b) Schematic structure of expression 3 in NMOS design style

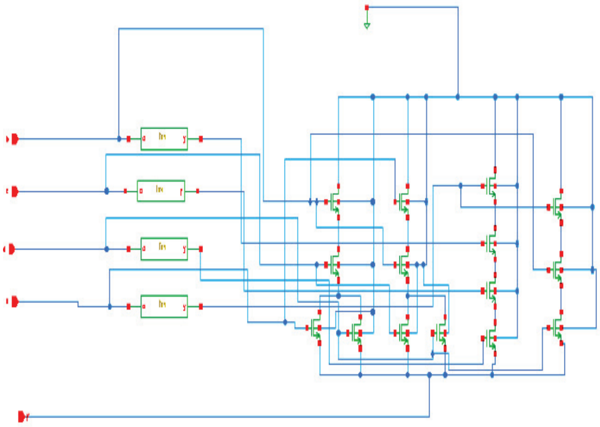


Figure 12. (c) Schematic structure of expression 3 in Super Gate design style

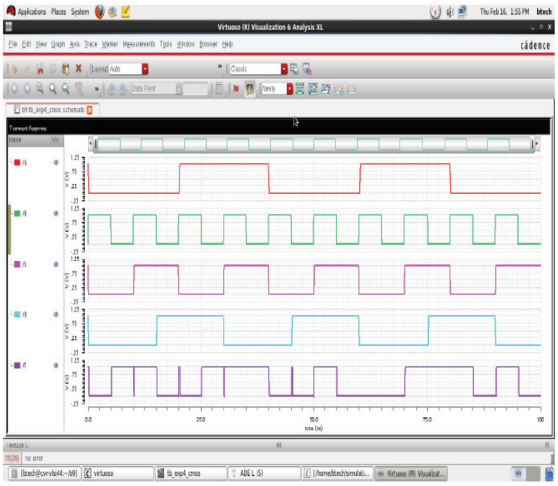


Figure 12. (d) Simulation waveform of Expression 3.

In the above simulation x-axis is defined by time in terms of nano seconds (ns) and y-axis defined by voltage in terms of volts (v).

Expression 4: F = w.x + w.y + w.z +x.y.z + x.y.v +y.s + w.v

This expression contains 6 literals namely w, x, y, z, s, v. The Fig. 13 gives schematic structures in three styles and simulation waveform. Boolean algebra (Demorgans law) is applied for the above expression so finally at the output side inverter is designed to retain the original output. CMOS design style of the expression 4 contains 11 nmos and pmos transistors and 1 inverter for the reduced network by using Boolean algebra for 6 literals i.e., w, x, y, z, s, v.

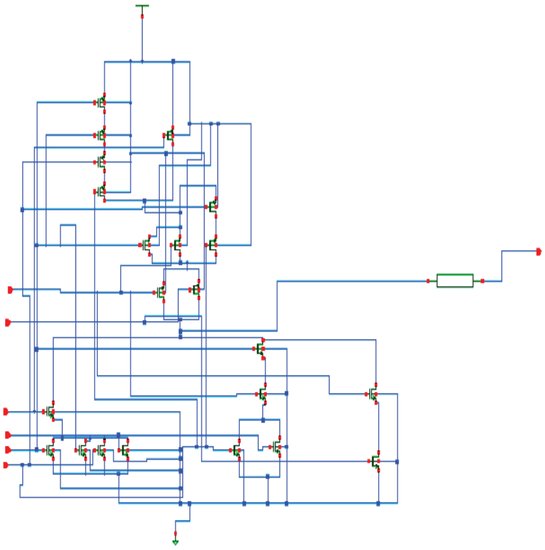


Figure 13. (a) Schematic structure of expression 4 in CMOS design style

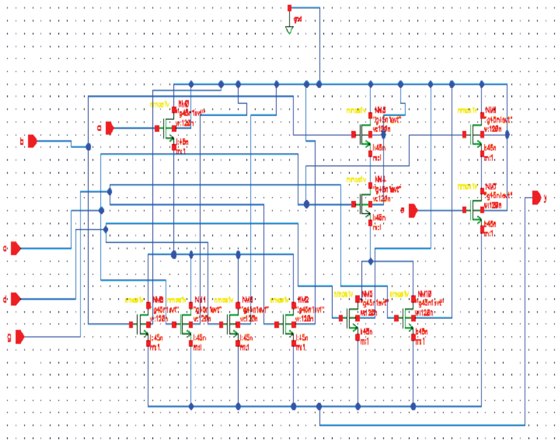


Figure 13. (b) Schematic structure of expression 4 in NMOS design style

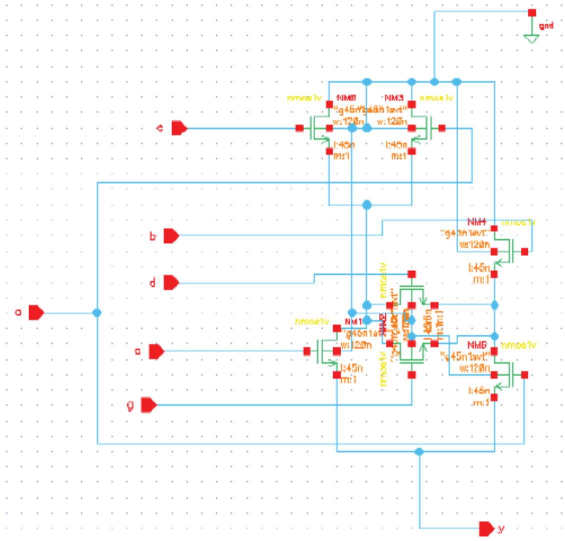


Figure 13. (c) Schematic structure of expression 4 in Super gate style

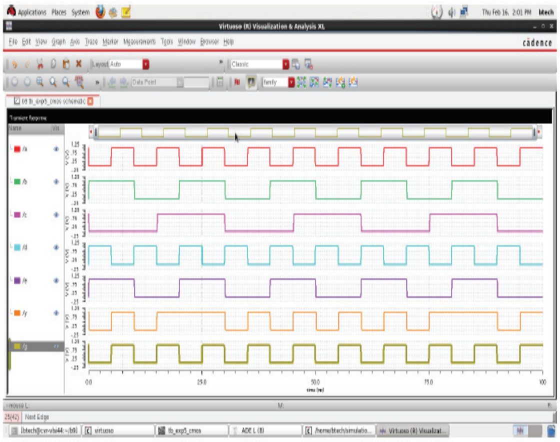


Figure 13. (d) Simulation waveform of Expression 4.

In the above simulation x-axis is defined by time in terms of nano seconds (ns) and y-axis defined by voltage in terms of volts (v).

V. Experimental Results

For evaluation and verification, the proposed Novel (Seed) Method is applied over four different Boolean expressions and compared with CMOS and NMOS logic style in three different aspects i.e., Transistor count [9] in Table I, Power Dissipated in nano watts of the network is shown in Table II and Total Area occupancy of the network in Table III.

TABLE I.
Transistor Count Minimization

	CMOS	NMOS	Super Gate
Expression 1	16	7	5
Expression 2	18	8	5
Expression 3	42	23	20
Expression 4	24	11	7

TABLE II.
Power Dissipation

	CMOS	NMOS	Super Gate
Expression 1	191.1 E-9	765.8 E-12	400.4 E-12
Expression 2	6.274 E-6	364.9 E-12	300.9 E-12
Expression 3	36.95 E-6	59.41 E-9	34.55 E-9
Expression 4	1.566 E-6	475.6 E-12	481.5 E-12

TABLE III.
Area Occupancy

	CMOS	Super Gate
Expression 1	171.797µm	18.473 µm
Expression 2	311.1797 µm	41.8914 µm
Expression 3	214.01 µm	15.30 µm
Expression 4	1072.78 µm	614.47 µm

VI. Conclusions

This paper describes an efficient super gate design method which is related on various combinations of Boolean functions in which to get a good relationship with the previous results. Four various combinations of Boolean functions were taken into consideration to get complete analysis of area, power and timing 1) deposit of four-input P-class Boolean functions 2) deposit of handcrafted arrangements that do not present transistors in CP associations 3) a given Boolean function with eleven variables a more complex type 4) the set of 5 – input transistor network is designed. A Novel (Seed) Method is designed and implemented in this paper, mainly used to generate optimized transistor networks. Novel (Seed) Method approach generates more general arrangement of network than the usual CP associations. The results give a significant and drastic change in the transistor count to implement the logic network or circuit compared to the existing approach. In CMOS technology the performance (timing), Power Dissipation and Area of Digital IC's are improved with the help of transistor count minimization concept. In general, Novel (Seed) method which is proposed in this paper reduces the power dissipated in the circuit, total Area and Number of Transistors used. These things even can be achieved by using STSCL logic and GTI Technique with few limitations.

References

[1] Y.T. Lai, Y.C.Jiang and H-M Chu, "BDD decomposition for Mixed CMOS/PTL logic circuit synthesis", in Proc. IEEE Int. Symp Circuits Syst (ISCAS), vol.6, May 2005,pp,5649-5652

[2] H.Al Hertani, D.Al Khalili and C Rozon,"Accurate total static leakage current estimation in transistor stacks," in proc. IEEE Int.Conf.Comput Syst.Appl,Mar.2006,pp.262-265

[3] D.Kagaris and T Haniotakis,"A methodology for transistor efficient super gate design," IEEE Trans. Very large Scale Integration Syst., Vol.15, no.4, pp,488-492, Apr.2007.

[4] T.J.Thorp, G.S.Yee and C.M.Sechen, "Design and synthesis of dynamic circuits,"IEEE trans. VLSI Syst., vol.11, no.1, pp, 141-149, Feb.2003.

[5] A.I.Reis and O.C.Andersen,"Library Sizing,"U.S.Patent 8015517, June 5, 2009.

[6] E.M Sentopvich et.al, "SIS: A system for sequential circuit synthesis," Dept., Elect. Eng. Comput. Sci., Univ. California, Berkley, Ca, USA, Tech. Rep, UGB/ERL M92/41, May1992.

[7] M.Rostami and K.Mohanram,"Dual-vth independent gate FinFETs for low power logic circuits," IEEE Trans.Comput-Aided Design Integration Circuit Systems., vol.30, no.3, pp. 337-349, Mar-2011.

[8] V.N.Possani,R S de Souza, J s Domingues, Jr. l V Agostini, F S Marques, and L S da Rosa, Jr. "optimizing transistor networks using a graph based network techniques", J analog Integration Circuits Signal Processing., vol. 73,no. 3,pp. 841-850, De. 2012.

[9] L S da Rosa, Jr F s Marques, F R Schneider, R P Ribas and A I Reis and "A Comparative study of cMOS gates with minimum stacks," Proc. 20th Annu., Conf. Integ. Circuit Systems Design (SBCCI), Sep.2007, pp. 93-98..

[10] M C Golumbic, A Mintz and U Rotics,"An improvement on the complexity of factoring read once Boolean functions", Discrete Appl. Math vol. 15