A New Generalized Polish Expression for VLSI Floor plan Problem

Bichitra Kalita
Assam Don Bosco University School of Technology, Guwahati-17, Azara.

Rongdeep Pathak
Dept. of Computer Applications, Assam Engineering College, Jalukbari, Guwahati, Assam.

Abstract: *The floor-plan design is one of the important problem and this problem can be modeled as a combinatorial optimization problem.Today,VLSI technology has taken a fundamental role in developing most of the high-tech electronic circuits. Even though VLSI design is renowned for its smaller size, lower cost, lower power, high reliability and high functionality, the design process takes long time and produces high risk. In this paper, a new expression NGEP has been proposed to representation VLSI floor plan having rectangular I-shape and as well as L-shape module. An algorithm has been discussed where input of the algorithm is an NGPE and corresponding output is the shape and layout/placement of the module of the floor plan. Two operations have been discussed for any modification of VLSI floor plan having rectangular and L-shape module.*

I.   Introduction:

Floor-planning is one of the most important problems in VLSI circuit design. The main concern is placement of rectangular modules of arbitrary size and shape in such a way that the total area covered by the modules and interconnections should be minimum. It has been found that a floor-plan can be classified into two categories – Slicing structure and Non Slicing structure. A slicing structure floor-plan can be obtained by repetitively cutting the floorplan horizontally or vertically, where as a Non slicing structure floorplan cannot. A binary tree can be used to represent a slicing floorplan on the basis of the slicing property. The advantages of slicing representation are generally found smaller encoding cost and solution space for faster runtime for packing. Besides, it is flexible to deal with hard, preplaced, soft and rectangular blocks. But the solution space of slicing structure may not be the optimal solution. On the other hand, in Non slicing structure the optimal solution might be achieved but it needs more evaluating runtime for packing than slicing approach.

There are two popular approaches of floorplanning, Simulated annealing and Analytical formulation, typically used to solve the floorplanning problem. Simulated annealing based floorplanning relies on the representation of the geometric relationship among the modules and analytical approaches usually capture the absolute relationship among the modules directly. In simulated annealing approach, first has to encode a floorplan as a solution, called a floorplan representation, which model the geometric relation of the modules in a floorplan.

It has been found that Floorplan representation is an important issue in designing a floor plan as it has a great impact on the flexibility and complexity of the floorplan. The slicing structure was first proposed by Otten [1]. Wong et al has been proposed a mechanism that transforms a binary tree in post order, called Polish expression to represent a Slicing floorplan [10]. It has been found that there are various floorplan representation of Non-slicing structure such as Normalized Polish expression [10], Sequencing Pair [9], O- tree [2], Corner Block list [3], B^+ - tree [4], transitive closure graph [5]. Chang Tzu Lin et al has been proposed a new representation of VLSI floorplan, which can efficiently reuse some area that cannot be utilized by Normalized Polish expression and also able to represent Non slicing structure of floorplan [6]. In Polish expression, a floorplan is encoded by a sequence including module names and two relational operators + and * for vertical and horizontal cut respectively. D.F. Wong and C.L. Liu had discussed about the inclusion of L- shaped module in a floorplan for minimizing the total area of a floorplan [7]. The New generalized Polish expression has been used to represent a Slicing and Non slicing floorplan containing only rectangular modules. The Graph theoretical concepts and algorithms have also been applied to the floorplan problem. The properties of oriented graphs had been used to solve rectangular dual transformation problem [12]. It has been found that every rectangular dual graph of a rectangular floorplan is a Planar triangulated graph [11]. Also it has been discussed that any PTG can be converted to a floorplan with I- shaped and L – shaped modules [8]. Kimura Yosuke et al[16] have been proposed a new genetic algorithm for the non-slicing structure problem. The proposed method is compared with existing methods using well-known benchmark problems. Different contributions on VLSI

design, about 52 papers on the design of VLSI using optimization are reviewed here. Accordingly, VLSI design optimization is analyzed through different bio-inspired algorithms and the performance measures of different VLSI experimentations are compared. Further, various improvements on Self-Adaptive Particle Swarm Optimization (SA-PSO) and VLSI design optimization, without the adoption of bio-inspired algorithms, are examined. Additionally, the floor planning problem of VLSI is considered and reviewed. Eventually, this paper provides the diverse research gaps and challenges in VLSI design, which may be helpful for the authors and the philosophers to contribute for further research[17].

II.   Proposed Work:

In this paper, a new representation NGPE has been proposed for a Slicing and Non slicing floorplan having rectangular as well as L – shaped modules. In this NGPE the concepts of Polish expression and four new operators have been used to represent the floorplan. An algorithm has been proposed to determine the shape and the layout of the modules in the floorplan. Previously it has been discussed that a Polish expression can be used to represent a floorplan which contain only rectangular modules, but the NGPE discussed here can be applied in case of I – shape module as well as L- shape modules. We have also discussed two operations which can be performed on the NGPE- tree of the corresponding floorplan for any modification of the given floorplan.

2.1.A New Generalized Polish Expression (NGPE):

A floorplan of a circuit dual graph with n modules where n>3, consists only I – shaped and L – shaped modules [8]. An L- shaped module may have four different forms in the floorplan. Therefore, a floorplan may consist of five different types of module, one of them is rectangular in shaped and rest of them are four different form of L – shaped modules. The following figure 1 shows five different types of modules that may be present in a floorplan
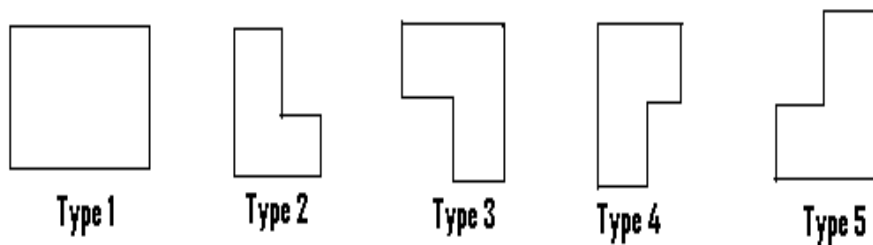

Figure 1

The NGPE is a sequence of module, including modules name and the operators where the operators arrange a module or a super module according to their properties. When an operator is scanned, the operator will combine previously scanned modules or super module according to its property. By scanning the NGPE from left to right a NGPE- tree can be constructed, where modules are the leaf and the operators are the internal nodes of the tree. The NGPE can also be obtained from the NGPE – tree by traversing it in post order.

In the NGPE, four corner points of a rectangular module and five corner points of an L- shaped module have been considered for accurate representation of a floorplan. The four corner point of a rectangular module can be represented by 4-tuples (top left; top right; bottom left; bottom right) .The four tuple of the module X is $(x_1; x_2; x_3; x_4)$ which is shown in figure 2.
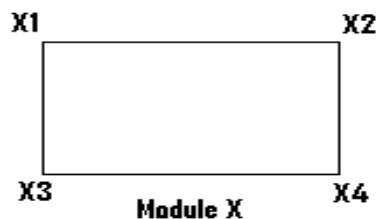

Figure 2

An L- shaped module has four outermost boundaries, two of them are horizontal and remaining two are vertical edges. We consider the smaller length of the horizontal and vertical edges of the L-shaped module. For the module A which is shown in figure 3, X1 and X2 are the outermost horizontal edges and Y1 and Y2 are the outermost vertical edges.
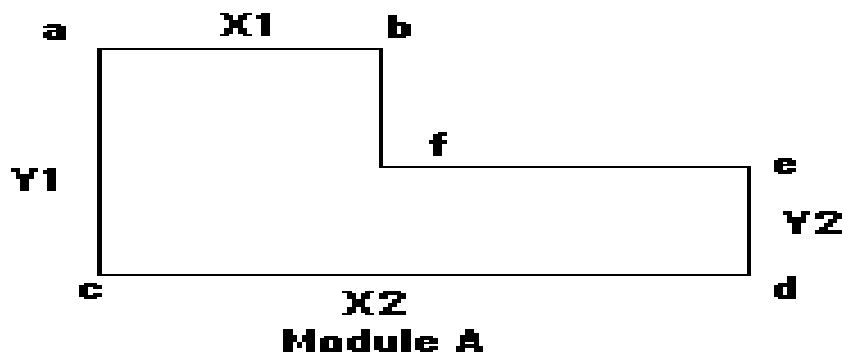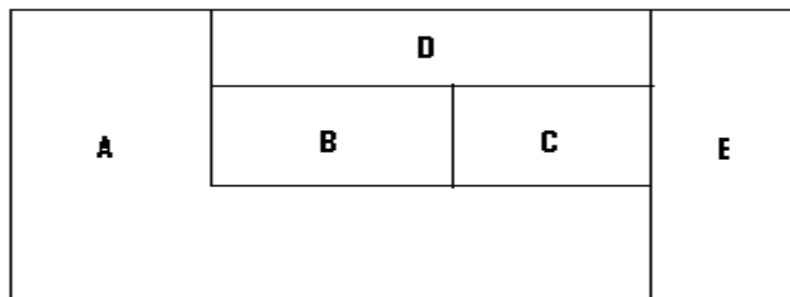
Figure 3

Here, the length of X1 is less than the length of X2 and the length of Y2 is less than the length of Y1. Therefore, we now consider the edge X1 and Y2 as shown in figure 3, where (a, b) and (d, e) are the two end points of X1 and Y2 respectively. We now consider these four points as the four corner points of the L –shaped module. The fifth corner point is the middle corner of the corner region formed by the L- shaped module. Therefore, the five corner points of the module A, can be represented by the 5- tuple (a; b; e; d; f). The five corner points of type 2 , type 3, type 4 and type 5 module can be represented as (top horizontal left ; top horizontal right ; right vertical top ; right vertical bottom ; middle corner), (left vertical top ; left vertical bottom ; bottom horizontal left ; bottom horizontal right ; middle corner), (bottom horizontal left ; bottom horizontal right ; right vertical top ; right vertical bottom ; middle corner), and (top horizontal left ; top horizontal right ; left vertical top ; left vertical bottom ; middle corner) respectively. These tuple information of rectangular and L – shaped module gives the neighbor module information touching the corner points of the respective module. If there is no neighbor module for a corner point then this can be specified as null or blank.

Figure 4

The figure 4 is a floorplan X consisting of one L-shaped (type2) module A and four rectangular (type 1) modules B, C, D and E respectively. The 5- tuple of module A can be represented as (---; D; C,E ; E ; B). In the floorplan X , the top horizontal left corner point of module A has no neighbor module , therefore it is specified as blank and the right vertical top has two neighbor module C and E. Therefore, it is specified as C, E in the tuple representation. These five tuple and 4- tuple of L-shaped and rectangular module are used in the floorplan representation for accurate placement of modules.
 In the NGPE, six operators +, *, $@_1$, $@_2$, $@_3$ and $@_4$ has been used to represent a floorplan. The operator $@_1$ is used to place a module or super module into the corner region formed by type 2 module. The operator $@_2$ is used to place a module or super module into the corner region formed by type 3 module. The operator $@_3$ is used to place a module or super module into the corner region formed by type 4 module. The operator $@_4$ is used to place a module or super module into the corner region formed by type 5 module.
Therefore, operators $@_1$, $@_2$, $@_3$ and $@_4$, has a relationship with type 2, type 3, type 4 and type 5 module respectively in the NGPE for a floorplan. The operators + and * has a relation with rectangular module. The operator + is used to place a type 1 module on the top of another module or super module and the operator * is used to place a type 1 module to the right of another module or super module. For accurate placement of modules in the floorplan, the 5- tuple and the 4- tuple information is provided along with the operators in the

NGPE. When an operator is scanned a module or a super module is placed into the previously placed module or a super module according to the type of the operator and the tuple information.
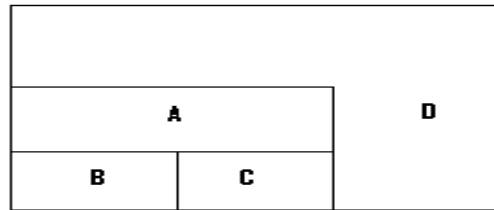

Figure 5

Let us consider, X is a super module consisting of module A, B and C as shown in figure 5. If we want to place a super module X   in the corner region of module D, then the NGPE would be $XD@_{2\,(-\,-\,-;\,A;\,C;\,-\,-\,-;\,A)}$

III. An Algorithmic Approach:

An algorithm has been developed to determine the shape and the placement of modules from a given NGPE. In this algorithm an ARRAY is used to store the NGPE and a STACK is used to store the modules and the super modules. Here, PUSH operation is used to insert a module into the top of the STACK and POP operation is used to remove a module from the top of the STACK. The NGPE is scanned character by character from left to right.

3.1.An Algorithm of determining the shape and placement of modules for a NGPE:
INPUT: X – a NGPE
OUTPUT: Y – floorplan corresponding to the NGPE X with the shape of the modules.
While there are input characters left
      Read the next character from the input
      If the character is a module or a super module
          PUSH the character into the STACK
      Else if the character is an operator
          Case 1: if the character is '+ '
               A = POP STACK
               B = POP STACK
               A is placed on top of B and PUSH into the STACK
          Case 2: if the character is '* '
               A = POP STACK
               B = POP STACK
               A is placed to the right of B and PUSH into the STACK
          Case 3: if the character is '$@_1$ '
               A = POP STACK
               B = POP STACK
               (B is type 2 module)
               A is placed in to the corner region of B and PUSH into the STACK
          Case 4: if the character is '$@_2$ '
               A = POP STACK
               B = POP STACK
               (A is type 3 module)
               B is placed into the corner region of A and PUSH into the STACK
          Case 5: if the character is '$@_3$ '
               A = POP STACK
               B = POP STACK
               (B is type 4 module)
               A is placed into the corner region of B and PUSH into the STACK
          Case 6: if the character is '$@_4$ '
               A = POP STACK
   B = POP STACK
               (A is type 5 module)
               B is placed on the corner region of A and PUSH into the STACK

IV.   Example :

Let us consider an example as shown in figure 6. The floorplan X which is shown in figure 6, consisting of eight modules A, B, C, D, E, F, G and H. The module A and G are of type 2, F is of type 3 and B,C,D,E and Hare of type 1 as discussed previously in figure 1. Now we are going to discuss how the floorplan X can be expressed into the new generalized Polish expression (NGPE), with the help of following considerations. The floorplan X in the NGPE form, we first consider module D and E where module E on the right of module D. Therefore, this can be expressed as DE*. The super module formed by D and E is in the corner of the module F and this can be represented as $DE*F@_4$.
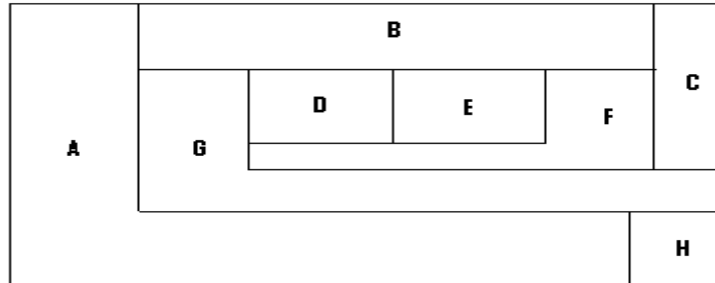
Floorplan X
Figure 6

Now, the modules D, E and F are in the corner region of module G and this is expressed as $GDE*F@_4@_1$. Again, module B is on the top of the super module consisting of modules G, D, E and F. Therefore, this can be represented as $GDE*F@_4@_1B+$. After the placement of module B, the shape of this super module is of type 2 and module C need to be placed on the corner region of the super module. Therefore, this can be expressed as $GDE*F@_4@_1B+C@_1$. Now, this super module is in the corner region of module A and this can be represented as $AGDE*F@_4@_1B+C@_1@_1$. Finally, the module H is on the corner region of the super module which is formed by the modules A, B, C, D, E, F and G. Therefore, the final NGPE for the floorplan X would be $AGDE*F@_4@_1B+C@_1@_1H@_3$.

V.   Addition and Deletion operations on the floorplan:

It has been found that three kinds of operation can be performed to perturb a GPE – tree [20]. Complement operation is performed to complement a chain of none zero length. Rotate operation is performed to rotate a module. Swap operation is performed to swap two leaves/modules, swap one leaf and one sub tree and swap two sub trees. Here, two operations Addition and Deletion on the NGPE- tree to modify a floorplan have been proposed. The addition operation can be performed by adding a new leaf or sub tree on the NGPE – tree. The deletion operation can be performed by deleting a new leaf or sub tree on the NGPE – tree.

For **addition operation** the following steps need to be performed:
Step 1: Locate the leaf/node (L), where the new leaf/node or sub tree is to be placed.
Step 2: Locate the parent node P of the leaf/node (L)
Step 3: Remove the leaf/node (L) from the parent node (P)
Step 4: Construct the NGPE tree with the removing leaf/node (L) and the new leaf or sub tree according to the type of arrangement
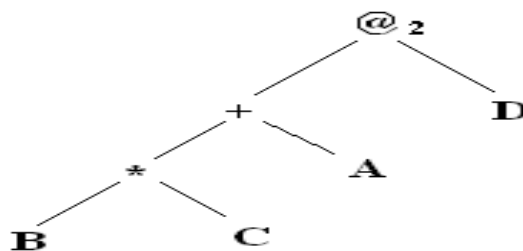Step 5: The Root (R) of the newly constructed tree become child of the parent node (P).

Figure 7

The figure 7 is the NGPE tree of the floorplan of figure 5. If we want to add a new module E to the right of the module A, then by following the steps as discussed above the new modified floorplan and the NGPE tree has been found which are shown in figure 8a and 8b respectively.
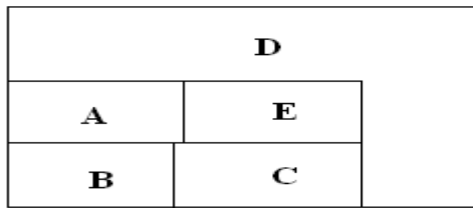

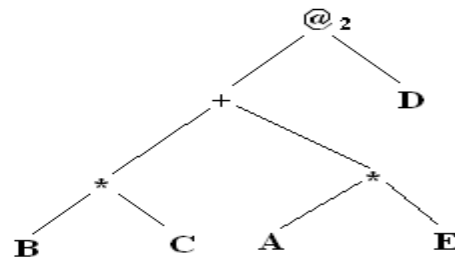**Figure 8a**   **Figure 8b**

For **deletion operation** the following steps need to be performed:

Step 1: Locate the deleting leaf/node (L) or root (R) of the sub tree

Step 2: Locate the parent node (P) of the deleting leaf/node (L) or the root of the subtree

Step 3: Locate the parent of the parent node (PP)

Step 4: Locate the sibling (S) of the deleting node/leaf (L) or the root of the sub tree

Step 5: Sibling node (S) become the child of the parent of the parent (PP) of the deleting leaf/node (L) or root of the sub tree (R)

Step 6: If there is no parent of the parent node (PP) then remove the leaf/node or the sub tree along with the parent (P) of the deleting leaf/node or the sub tree

Step 7: The sibling (S) of the deleting leaf/node (L) or the sibling of the root of the deleting sub tree become the root of the new NGPE tree.
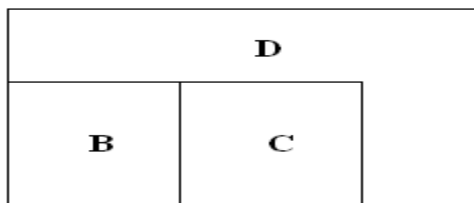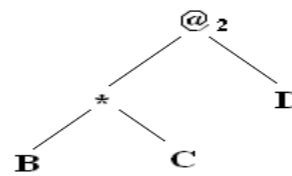

Figure 9a   **Figure 9b**

If we want to delete a super module formed by module A and E from the floorplan which is shown in figure 8a .Then by the following the steps as discussed above a new modified floorplan and a NGPE tree has been found, which are shown in figure 9a and 9b respectively..

VI.   Conclusion:

It has been found that a floorplan representation is an important issue in designing a floorplan as it has a great impact on the flexibility and complexity of the floorplan. A new generalized Polish expression (NGPE) has been introduced to represent a Slicing and Non Slicing floorplan having rectangular as well as L shaped module. This new representation can efficiently be reused some area which cannot be utilized by vertical and horizontal operators defined by Polish expression. In this new representation the concepts of Polish expression and a set of new operators have been used to represent a floorplan. An algorithm has been developed to determine the shape and the layout the modules in the floorplan. Two operations for any modification of a floorplan have also been discussed.

References

[1].   R. H. J. M. Otten, "Automatic floorplan design", Proc. DAC, pp. 261-267, 1982
[2].   Y. Pang, C. K. Cheng and T. Yoshimura, " An enhance perturbing algorithm for floorplan design using O tree representation", Proc. ISPD, pp. 168-173, 2000.
[3].   X. Hong, G. Huang, Y. Cai, S. Dong, C. K. Cheng and J. Gu ," Corner block list: an effective and efficient topological representation of non slicing floorplan", Proc. ICCAD, pp. 8-12, 2000.
[4].   Y. C. Chang, Y. W. Chang, G. M. Wu and S. W. Wu, "B* trees: a new representation for non slicing floorplans", Proc. DAC, pp. 458-463, 2000.

[5]. J. M. Lin and Y. W. Chang," TCG: A transitive closure graph based representation for non slicing floorplan", Proc. DA, pp. 764-769, 2001.

[6]. C. T. Lin, D. S. Chen and Y. W. Wang," GPE: a new representation for VLSI Floorplan problem", Proc. ICCD, PP. 42-44, 2002.

[7]. D. F. Wong and C. L. Liu," Floorplan design of VLSI circuits", Algorithmica, Vol. 4, 1989, pp 263-291.

[8]. BornaliGogoi, Bichitra Kalita, "Algorithm for Designing VLSI Floor plan using Planar Triangulated Graph", International Journal of Information and Communication Technology Research(ISSN 2223-4985),Vol.2 No.7, pp. 564-572, July 2012.

[9]. Nakaya S., Koide T. and Wakabayashi S. " An adaptive genetic algorithm for VLSI floorplanning based on Sequence pair", Proc. ISCAS , pp. 65-68, 2000.

[10]. D. F. Wong and C. L. Liu ," A new algorithm for floorplan design", Proc. DAC, pp. 101-107, 1986.

[11]. M. Sarrafzadeh and C. K. Wong " An Introduction to VLSI Physical Design", McGraw Hill International edition, 1996.

[12]. Hui Tang and Waikai Chen, " Generation of Rectangular Duals of a Planar Triangulated graph by elementary Transformation", IEEE 1990, pp. 2857-2860.

[13]. Y. Lai and S. M. Leinwand" Algorithms for Floorplan design via rectangular dualization", IEEE Trans. On Computer Aided Design, 7 (12): 1278-1289, 1988.

[14]. Sadiq M. Sait and Habib Youssef, " VLSI physical design automation: theory and practice', IEEE press, vol 3 , 1995 pp 90-92.

[15]. Kalita B. " Partitioning of special circuit", IJMIE, vol. 2 issue 2, February 2012, pp. 70-77.

[16]. Yosuke Kimura, Kenichi Ida, " Improved genetic algorithm for VLSI floorplan design with non-slicing structure", Computers & Industrial Engineering 50, 528–540,2006.

[17]. S.B. Vinay Kumar , P.V. Rao, H.A. Sharath, B.M. Sachin, U.S. Ravi, B.V. Monica, "Review on VLSI design using optimization and self-adaptive particle swarm optimization", Journal of King Saud University - Computer and Information Sciences, Volume 32, Issue 10, Pages 1095-1107, December 2020.

[18]. Vinod Kumar,KrishnendraShekhawat, "A transformation algorithm to construct a rectangular floorplan", Theoretical Computer Science 871,94–106,2021.