A GA-based Floorplanning method for Topological Constraint

Masaya Yoshikawa*, and Hidekazu Terai*

* Department of VLSI System Design, Ritsumeikan University, Japan
(Tel : +81-77-561-2615; E-mail: yosikawa@se.ritsumei.ac.jp)

Abstract: The floorplanning problem is an essential design step in VLSI layout design and it is how to place rectangular modules as density as possible. And then, as the DSM advances, the VLSI chip becomes more congested even though more metal layers are used for routing. Usually, a VLSI chip includes several buses. As design increases in complexity, bus routing becomes a heavy task. To ease bus routing and avoid unnecessary iterations in physical design, we need to consider bus planning in early floorplanning stage. In this paper, we propose a floorplanning method for topological constraint consisting of bus constraint and memory constraint. The proposed algorithms based on Genetic Algorithm(GA) is adopted a sequence pair. For selection control, new objective functions are introduced for topological constraint. Studies on floor planning and cell placement have been reported as being applications of GA to the LSI layout problem. However, no studies have ever seen the effect of applying GA in consideration of topological constraint. Experimental results show improvement of bus and memory constraint.

Keywords: VLSIFloorplanning, Genetic Algorithm, Topological Constraint and Sequence pair

1. INTRODUCTION

With the deep sub-micron (DSM) technology, IC density and design complexity are growing at an incredible speed. Beside, the development of system on a chip (SOC) and IP reuse make number of modules in a design more numerous than previous times. In such a DSM technology, Floor planning plays an important role in physical design of VLSI circuits. Floorplanning is to decide the positions of circuit blocks or IP blocks on a chip. It is the early stage of design and it determines the overall chip performance. The traditional floorplanning problems deal with the unconstrained two dimensional packing problem. In the case of the packing problem, a set of rectangular blocks has to place such that no blocks overlap each other. The area of the rectangle circumscribing all blocks has to be minimal, hence the optimal packing pattern is that with minimal waste inside the enveloping rectangle. As DSM technology advances, a chip become more congested even though more metal layers are used for routing. However, traditional floorplanners did not pay attention to bus routing. Usually, a chip includes several buses. As design increases in complexity, bus routing becomes a heavy task. Since buses have different widths and go through several module blocks, the positions of macro blocks greatly affect bus planning. To ease bus routing and avoid unnecessary iterations in physical design, it is important to consider bus planning in early floorplanning stage. In this paper, we proposed the novel GA [1] based floorplanning method for topological constraint. Studies on floorplanning [2], routing [3], and cell placement [4] have been reported as being applications of GA to the VLSI layout problem. However, no studies have ever seen the effect of applying GA in consideration of topological constraint.

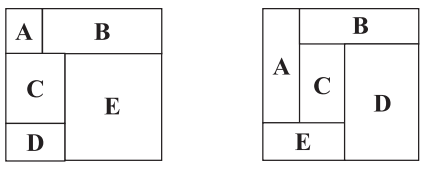
2. PRELIMINARIES

2.1 GA and Real-Coded GA

GAs are search algorithms based on the mechanics of natural selection and natural genetics. GA has the three operators of selection, crossover, and mutation. If an individual has a relatively high fitness value, it has more chances to obtain its children in the next generation by a fitness-based selection rule. This operator is an artificial version of natural selection. In the crossover phase, two individuals in a present population are randomly selected, and they exchange their bits from the crossing site determined by another random number. During the mutation process, every bit has as equal probability to replaced by its complement number. This is an imitation of natural crossover and is very rare in progressing GA as well. However, in order that GA adopts the bit string for coding, the continuity on solution space is not reflected. There is Real-Coded GA(RCGA) [5-8] as a technique for solving this problem. RCGA, in which each chromosome is encoded as a real number vector instead of the binary bit string, have turned out to be a promising approach for global optimization in continuous search space.

2.2 Floorplanning problem

The Floorplanning is a generalization of the placement problem in VLSI building block layout, and it determines the coarse placement for the given set of modules. A module set *M* consists of the set of hard modules and soft modules. A hard module is a sub-circuit whose shape is a rectangle with a given height and width. The soft module size is specified by an area and a range of acceptable aspect ratios. Soft modules are frequently assumed in the early stages of a floorplan because the module itself may not yet be fully designed so its final dimensions are unknown. A floorplan of a set of modules is a non-overlapping placement of given modules. The minimum bounding rectangle of a floorplan is called the chip. So, the traditional floorplanning problem is to find a floorplan of *M* onto a chip with the minimum area. Moreover, floorplans are classified as either slicing or nonslicing. A slicing floorplan is a floorplan that can be obtained by recursively cutting rectangles horizontally or vertically. A nonslicing floorplan is one that is not restricted to be slicing. Fig.1 shows an example of both floorplan. Nonslicing floorplans are a more general representation that can describe all kinds of packings. Therefore, the proposed algorithm adopts nonslicing floorplan.



(1) slicing (2) nonslicing

Fig. 1 Example of slicing and nonslicing

2.3 Sequence pair

The sequence pair [7] was proposed as a representation method of block placement to determine the densest possible placement of rectangular modules of floorplanning in VLSI layout design. The merit of using a sequence pair to solve the floorplanning problem are that it can represent arbitrary rectangle packing and each sequence pair always has its corresponding packing. A sequence pair is an ordered pair of Γ+ and Γ−, where each of Γ+ and Γ− is a permutation of the names of given n modules. Fig.2 shows floorplan and a relative position of each module of one. Given (Γ+, Γ−), one the optimal packing under the constraint can be obtained by applying the well known "longest path algorithm" for vertex weighted directed acyclic graphs (horizontal and vertical constraint graph) as shown in Fig.3.

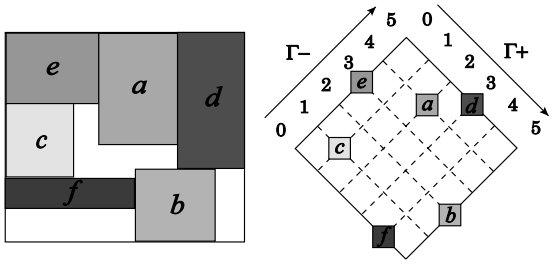


Fig. 2 Example of floorplan and a relative position

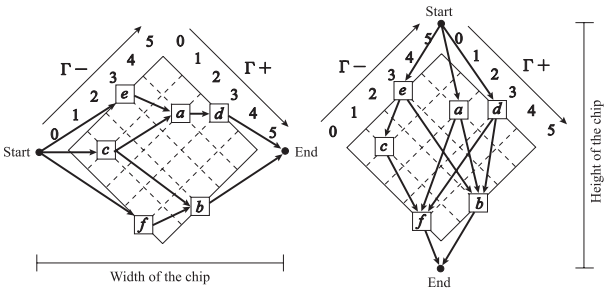


Fig. 3 Example of horizontal and vertical constraint graph

3. SELECTION OPERATION

For GAs it is important to set suitable evolution parameters for controlling the selection of individuals. In order to realize high quality floorplan, the proposed algorithm introduces a new multi-objective function in consideration of bus routing and memory. First, in order to satisfy the constraint for bus routing, the connectivity between modules was introduced. Concretely, the connectivity according to proximity constraint is defined as shown in Fig. 4 (1). Similarly, the connectivity according to symmetry constraint is also defined as shown in Fig. 4 (2). The connectivity is used as evaluation of topological constraint (layout constraint). It can be satisfied with minimizing this evaluation of the demand to floorplanning, such as length of bus routing, and the number of times of bending. Furthermore, a memory top has the constraint which cannot pass routing as shown in Fig. 5 (1). Therefore, when routability is considered, it is necessary to place a memory outside as much as possible. Then, the evaluation for the memory adopts the distance between an object module (memory) and the outer frame of the chip as shown in Fig. 5 (2). This value also should be small.

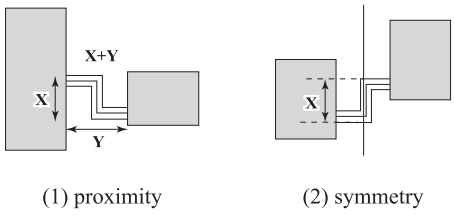


Fig. 4 Constraint for bus routing

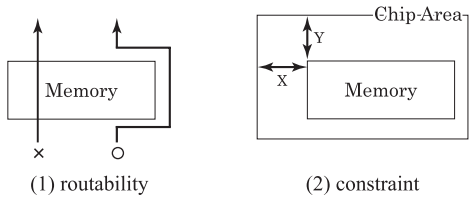


Fig. 5 Constraint for memory

4. CROSSOVER, AND MUTATION OPERATION

4.1 Crossover operation

For Real-coded Genetic Algorithms, there have been proposed many crossover operators. The blend crossover (BLX–α) [5,6] proposed by Eshelman and schaffer shows good search ability for separable fitness function. Therefore, we used a modified BLX–α in order to correspond to the proposed coding. Fig.6 shows the feasible children space for the BLX–α operator. In the BLX–α, children are generated as follows:

(1) Choose two parents x^1, x^2 randomly from the population.
(2) A value of each element x_i^c of the children vector x^c is chosen romdomly from the interval $[X_i^1, X_i^2]$ following the uniform distribution, where

$$\begin{aligned} X_i^1 &= \min(x_i^1, x_i^2) - \alpha d_i \\ X_i^2 &= \max(x_i^1, x_i^2) + \alpha d_i \\ d_i &= | x_i^1 - x_i^2 | \end{aligned} \quad (1)$$

and x_i^1 and x_i^2 are the i–th elements of x^1 and x^2, respectively, and α a positive parameter. For the value of α, we use 0.5.

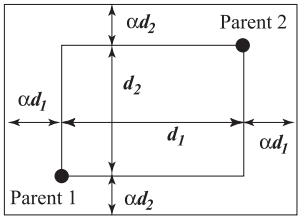


Fig. 6 Example of BLX-α

4.2 Mutation operation

The mutation in the proposed algorithm changes the aspect ratio of each module with a random number. However, the random number is the real number in the range of acceptable aspect ratios. Thus, efficient search is realized by limiting the random number to generate.

5. EXPERIMENTS AND DISCUSSION

In order to evaluate the effectiveness of the techniques proposed in this paper, we conducted the experiment using MCNC benchmark data. Parameters of GA are described as follows: The population size is 500, the tournament size is 4, and the mutation rate is 0.05%.

5.1 Experiment 1

The preliminary experiment was conducted in order that the soft block using RCGA might verify whether it is functioning effectively. The result of a preliminary experiment is shown in Fig.7 and Fig.8. The aspect ratios of all soft blocks differ as shown in these figures. Thus, the RCGA is effective in soft blocks.

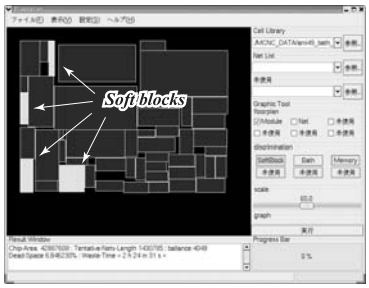


Fig. 7 Example of soft module 1

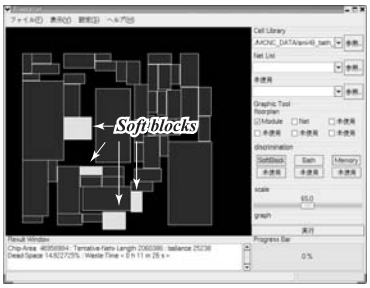


Fig. 8 Example of soft module 2

5.2 Experiment 2

Fig.9 shows the result of floorplan in consideration of topological constraints (bus routing and memory). Chip area is 45102792 and run time is 889 sec. Thus, the floorplan which does not almost have the dead-space was realized at high speed.

Topological constraints are considered. First, soft blocks became the blocks form which make area minimize in the acceptable aspect ratio as shown in Fig.9. Next, the memory constraint functioned effectively and the memory has been placed in a corner of chip area. Two topological constraints for bus routing were very effective. Thus, the floorplan in consideration of topological constraints could be realized, and the validity of the proposed algorithm was verified.

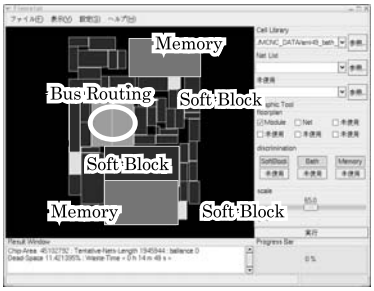


Fig. 9 Result of floorplan

6. CONCLUSION

In conclusion, we proposed a novel floorplan technique bas ed Real-Coded Genetic Algorithm. For selection control, new multi-objective function was introduced in order to improve th e performance floorplanning. The experiment verified that the proposed multi-objective functions are effective for bus routin g and memory.

In relation to future research, reducing the run time is the most important priority. We will also experiment using large scale.

ACKNOWLEDGMENTS

This research is partially supported by the Nakajima Foundation.

REFERENCES

[1] J.Holland, Adaptation in Natural Artificial Systems, the University of Michigan Press (Second edition ; MIT Press), 1992

[2] S.Nakaya, Koide, T. Wakabayashi, S, “An adaptive genetic algorithm for VLSI floor planning based on sequence-pair,” *Proc. IEEE ISCAS*, Vol.3, pp.65-68, 2000

[3] A. Kanasugi, et al., “A Greedy Router based on Maze Router and Genetic Algorithm,” *Proc. Int. Conf. on Electronics Packaging*, pp.427-430, 2001.

[4] L.Zhang, et al. “Analog Module Placement Realizing Symmetry Constraints Based on a Radiation Decoder,” *Proc. IEEE MWSCAS*, pp. I-481-484, 2004.

[5] L.J. Eshelman and J.D.Schaffer, “Real-Coded Genetic Algorithms and Interval Schemata,” *Foundations of Genetic algorithms 2*, pp.187-202, 1993.

[6] Takahashi, M.; Kita, H, “A crossover operator using independent component analysis for real-coded genetic algorithms,” *Proc. of CEC*, Vol.1, pp.643 - 649, 2001.

[7] H.kita, I.ono and S.kobayashi, “Multi-Parental Extension of Unimodal Normal Distribution Crossover for Real-Coded Genetic Algorithms,” *Proc. of the 2001CEC*, Vol.III, pp.646-651, 1999.

[8] I.ono and S.Kobayashi, A Real-Coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, Proc. 7th ICGA, pp.246-253, 1997