

Constraint-Driven Floorplanning based on Genetic Algorithm

MASAYA YOSHIKAWA, HIDEKAZU TERAJ
 Department of VLSI System Design,
 Ritsumeikan University
 1-1-1, Nojihigashi, Kusatsu, Shiga, 525-8577
 JAPAN

Abstract: - With recent advances of Deep Sub Micron technologies, the floorplanning problem is an essential design step in VLSI layout design and it is how to place rectangular modules as densely as possible. In this paper, we propose a novel constraint driven floorplanning technique based on Genetic Algorithm (GA). Many works have done for the floorplanning problem using GA. However, no studies have ever seen the effect of applying GA in consideration of bus routing constraint and position constraint. Experimental results show improvement of bus routing constraint and position constraint, keeping the chip area and total wire length.

Key-Words: - Genetic Algorithm, Bus routing, Floorplanning, Constraint-driven, LSI layout

1 Introduction

With increasing circuit integration and downsizing, chips become more congested even though more metal layers are used for 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[1]. To ease bus routing and avoid unnecessary iterations in physical design, we need to consider bus planning in early floorplanning stage. However, almost previous algorithms of floorplanning perform interconnect optimization on individual nets and are unable to take advantage of busing signals with related timing, topology and parasitic requirements. All of these factors not only increase the iterations, but also result in poor global optimization of the floorplan[2].

In this paper, we propose a novel floorplanning technique based on Genetic Algorithm (GA)[3],[4] in consideration of bus routing constraint and position constraint. The GA is one of the most powerful optimization methods based on the mechanics of natural evolution. Studies on floorplanning based on GA have been reported[5],[6]. However, the goal of these previous works is minimizing a chip area and total wire length. No studies have ever seen the effect of applying GA in consideration of bus routing constraint and position constraint.

2 Preliminaries

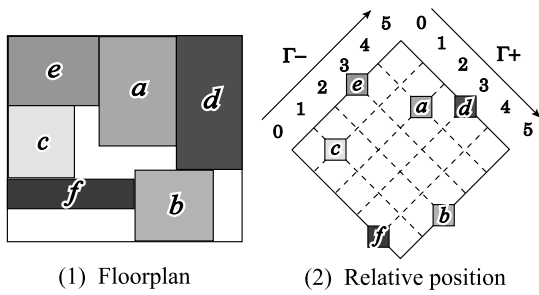
2.1 Floorplanning Problem

In the floorplanning problem discussed in this paper, we assume that there are n rectangular blocks, B_1, B_2, B_3, B_n , which consist of hard blocks, with fixed widths and heights, and soft blocks, with variable-length sides. The smallest rectangle that surrounds all of the blocks is called the chip. Furthermore, each block is assumed to have multiple terminals on the perimeter or in its interior, with the terminals connected to the routing layer above the block based on a net-list, which is provided as the input. A net is the set of terminals connected by a single signal wire. The wire length of a net is evaluated by 1/2 of the perimeter of the smallest rectangle that contains all of the terminals of that net.

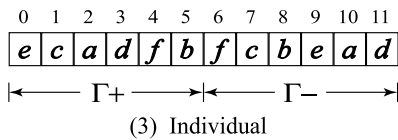
2.2 Sequence Pair

The sequence pair[7] was proposed as a representation method of block placement to determine the densest possible placement of rectangular blocks 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 blocks. Fig.1 shows floorplan and a relative position of each block 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.2.

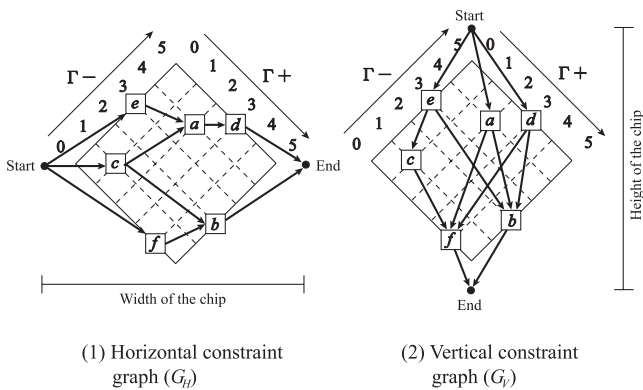


(1) Floorplan (2) Relative position



(3) Individual

Fig.1 Example of sequence pair



(1) Horizontal constraint graph (G_H) (2) Vertical constraint graph (G_V)

Fig.2 Example of constraint graphs

3 Base Algorithm

3.1 Coding

The proposed algorithm (Constraint-driven Floorplanning using Genetic algorithm: CFG) expresses the form of a soft block with an aspect ratio. Therefore, an individual is composed of the sequence pair $(\Gamma+, \Gamma-)$, the orientations of the blocks, denoted by θ , and the aspect ratios of the blocks as shown in Fig.3. Here, the block orientation considers as four kinds (0,90,180,270 degrees), and the aspect ratio of each hard block is set to 0. By this coding, the floorplanning in which hard blocks and soft blocks are intermingled can be dealt with.

3.2 Selection

For GA, it is important to set suitable evaluation parameters for controlling the selection of individuals.

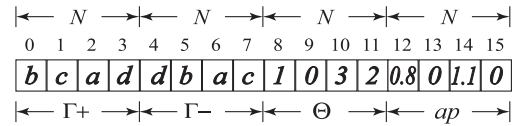


Fig.3 Example of coding

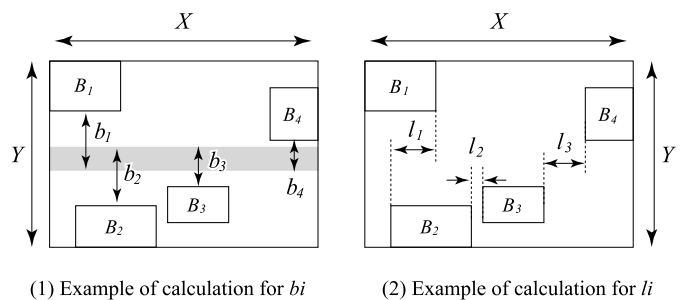
In this research, the evaluation for bus routing constraint and position constraint are introduced in addition to conventional evaluation (chip area and total wire length).

3.2.1 Bus routing constraint

As regards bus routing, it is necessary to take bit width into consideration. The demands for bus routing are as follows. (1) Lessen the number of times of bending. (2) Shorten length. Therefore, two objective functions are introduced in order to satisfy these demands of bus routing. One is the objective function (b_s) which divides bus routing into a trunk and branches line, and the other is the objective function (b_n) which considers the number of times of bending of bus routing. As shown in Fig.4, the sum total of the length (b_i) of each branch line and the sum total of the distance during each block (l_i) used for the objective function b_s . And then, b_n is the total of b_i which is not zero in each branch line. The value of b_s and b_n should be small.

3.2.2 Position constraint

Usually, it is necessary to place interface blocks (AD/DA converters, USB module, etc.) on the outside of a chip. Moreover, the routing is usually prohibited on the memory as shown in Fig.5(1). In such the case, the memory also should be placed on the outside of a chip from a viewpoint of routability. Therefore, the evaluation for the constraint blocks adopts the distance between the object block and the outer frame of the chip as shown in Fig.5(2). This value also should be small.



(1) Example of calculation for b_i (2) Example of calculation for l_i

Fig.4 Bus routing constraint

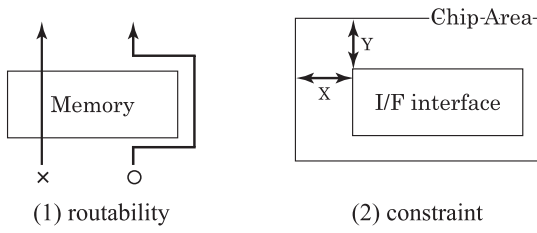


Fig.5 Position constraint

3.3 Crossover and Mutation

In CFG, two kinds of crossover operators are adopted in order to realize a floorplan in which hard blocks and soft blocks are intermingled. One uses CTPX[5] as a crossover operator for sequence pair. The other uses BLX- α [4] as a crossover operator for the aspect ratio. CTPX is an elite crossover, which preserves the common characteristics of parents. In CTPX, first, the longest common subsequences (LCSs) of both $\Gamma+$ and $\Gamma-$ of parents are found. Then, blocks in the LCSs are preserved on the same loci. Other blocks are exchanged so that the order of them in $\Gamma+$ and $\Gamma-$ and θ of one child is the same as the parent of the other child. The example of CTPX is shown in Fig.6.

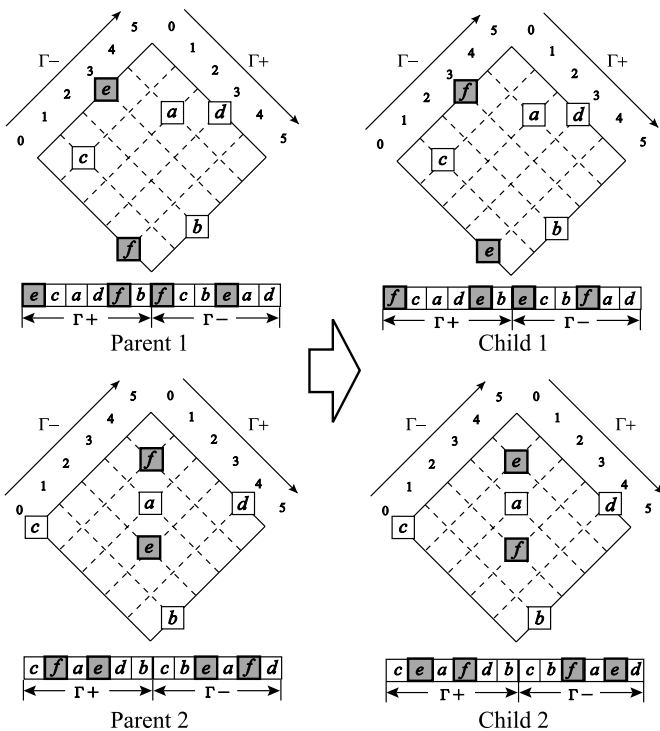


Fig.6 Example of CTPX

Since chromosome of the aspect ratio is encoded as a real number vector, CTPX cannot be used for the aspect ratio. BLX- α shows good search ability for real-coded GA. Therefore, we used a modified BLX- α in order to correspond to the proposed coding. Fig.7 shows the feasible children space for the BLX- α operator. With the sequence pair, the floorplan is determined by $\Gamma+$, $\Gamma-$, θ (block orientation), and the aspect ratio.

Mutations must therefore be introduced into these four elements to maintain diversity in the population. In addition, the floorplan must be drastically modified by mutation to avoid local optima. For this reason, CFG employs six types of mutations: mutations on the four elements (i.e., $\Gamma+$, $\Gamma-$, orientation, and aspect ratio mutations) and those that drastically modify the floorplan ($\Gamma+\Gamma-$ mutation and pair exchange mutation). The example of $\Gamma+$ mutation is shown in Fig.8 and that of pair exchange mutation is shown in Fig.9.

3.4 Local Search

GA is good at searching globally, because the GA is a multi-point searching algorithm. However, GA is not suitable for local searches in order to execute the searches by crossover. On the other hand, one-point searching algorithm (LS: Local search) is suitable for the local searches. That is, a hybrid method of GA with LS is good at searching not only globally but also locally. In CFG, the hybrid GA performs LS operation on the elite individual. The elite individual will be replaced if LS can improve the evaluation value. Moreover, CFG introduced three types of LS. The first LS changes relative positions of blocks on the critical path of horizontal- and vertical-constraint graphs. And then, the second LS changes the orientation of blocks on the critical path of horizontal- and vertical-constraint graphs. The last LS changes the aspect ratio of soft blocks on the critical path of horizontal- and vertical-constraint graphs as shown in Fig.10.

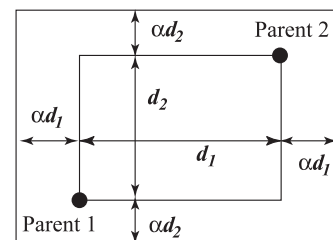


Fig.7 Example of BLX- α

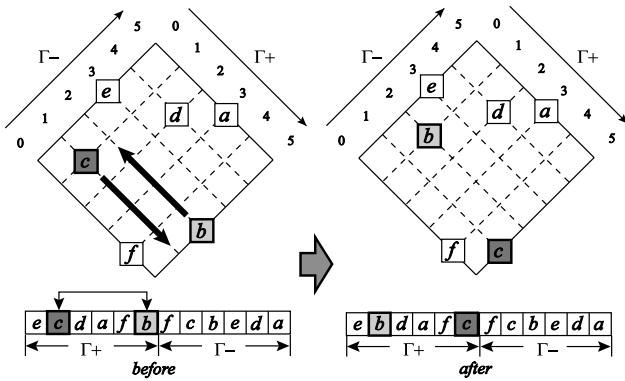


Fig.8 Example of $\Gamma+$ mutation

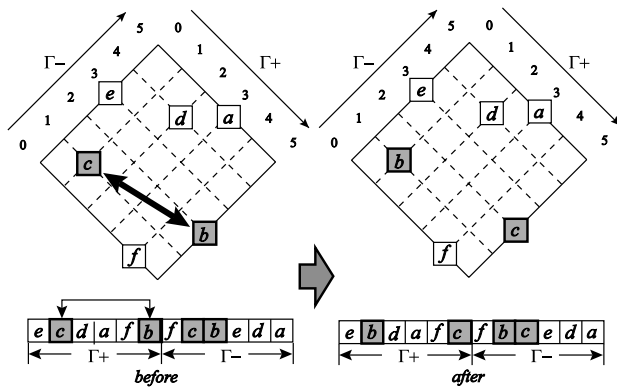


Fig.9 Example of pair exchange mutation



(1) Before of LS for aspect ratio (2) After of LS for aspect ratio

Fig.10 Example of LS for aspect ratio

An aspect ratio will be enlarged if the soft block is on horizontal constraint graph. On the contrary, that will be made small if the block is on vertical constraint graph. Thus, LS is applied to three elements which determine a floorplan.

4 Experiments and Discussion

In order to evaluate the effectiveness of the techniques proposed in this paper, we conducted the experiment using MCNC benchmark data (ami33 and ami49). The experiment consisted of 10 trials, with the software run on a platform of a Pentium IV 3.6GHz CPU. Parameters of GA are described as follows: The population size is 50, generations are 1,000, the tournament size is 4, and the mutation rate is 0.05%.

The experimental results are shown in Table.1 and Table.2. Each value in both tables expresses the average value of 10 trials. CFG improved the quality for bus routing and constraints blocks, keeping the chip area and total wire length, in comparison with conventional floorplanning technique. Especially, CFG realized bus routing without bending.

Table.1 Experimental result of ami33

	Conventional Method	CFG
bs (mm) (bus length)	977.4	84.4
bn (#bending of bus)	7.2	0
pl (mm) (position const.)	762.1	0
area (mm ²)	1.300	1.257
length (mm)	64.82	69.12
run time (s)	38	48

Table.2 Experimental result of ami49

	Conventional Method	CFG
bs (mm) (bus length)	6762.6	103.6
bn (#bending of bus)	9.2	0.2
pl (mm) (position const.)	2918.3	0
area (mm ²)	39.51	40.72
length (mm)	1293.5	1179.3
run time (s)	83	184

It became clear that Genetic Algorithm could improve floorplanning performance.

Lastly, result of floorplanning using proposed technique is shown in Fig.11.

5 Conclusion

In conclusion, we proposed a novel floorplanning technique based on Hybrid Genetic Algorithm. The proposed algorithm was used to search the solution based on sequence pairs, and dealt with the floorplanning in which hard blocks and soft blocks were intermingled. For selection control, new objective functions were introduced for bus routing constraint and position constraint. The experiments verified that the proposed objective functions are effective for improvement of the quality for bus routing and constraint blocks, keeping the chip area and total wire length.

Future research includes the reduction of wire congestion and power consumption.

References:

[1] Hua Xiang, Xiaoping Tang, Wong, M.D.F., Bus-driven floorplanning, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.23, No.11, pp.1522-1533, 2004.

[2] Rafiq, F.; Chrzanowska-Jeske, M.; Yang, H.H.; Jeske, M.; Sherwani, N., Integrated floorplanning with buffer/channel insertion for bus-based designs, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.22, No.6, pp.730-741, 2003.

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

[4] Takahashi, M.; Kita, H, A crossover operator using independent component analysis for real-coded genetic algorithms, *Proc. of IEEE Congress on Evolutionary Computation*, Vol.1, pp.643 - 649, 2001.

[5] Nakaya, S., Koide, T., Wakabayashi, S., An adaptive genetic algorithm for VLSI floorplanning based on sequence-pair, *Proc. of The 2000 IEEE International Symposium on Circuits and Systems*, Vol.3, pp.65-68, 2000.

[6] Hung, W.-L., Xie, Y., Vijaykrishnan, N., Addo-Quaye, C., Theocharides, T., Irwin, M.J., Thermal-aware floorplanning using genetic

algorithms, *Proc. of International Symposium on Quality of Electronic Design*, pp.634-639, 2005.

[7] H.Murata, K.Fujiyoshi, S.Nakatake and Y.Kajitani , VLSI module placement based on rectangle-packing by the sequence-pair, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.15, No.12, pp.1518-1524, 1996.



Fig.11 Result of floorplanning