Proceedings of the 4th WSEAS/IASME Int. Conf. on System Science and Simulation in Engineering, Tenerife, Spain, December 16-18, 2005 (pp156-162)

Floorplanning Algorithm for multiple clock domains

Changhong Zhao, Jian Chen, Dian Zhou, Xiaofang Zhou, Xuan Zeng
ASIC & System State Key Lab. Fudan University
Rd. Zhangheng No.825 Shanghai P.R.China 201203

Abstract: This paper presents the problem of floorplanning with multiple clock domains and the corresponding algorithms based on simulated annealing algorithm and sequence pair representation. The main contribution of this algorithm is to solve the floorplanning problem with multiple clock domains without increasing the complexity. Experimental results show that good results can be obtained for the floorplanning with multiple clock domains.

Key word: floorplanning

NOMENCLATURE

*Parameter:

N : number of blocks

N_i : number of blocks in clock domain i

K : number of clock domains

h_i : the height of module i

w_i : the width of module i

a_i : the area of module i

$b(i)$: the ith block

$C(b(i))$: the clock domain of block $b(i)$

τ_i : the aspect ratio of block i

b_{ij} : the jth block in clock domain i

$clock_i$: The ith clock domain

$node(i)$: The ith clock domain in the constraint graph of clock domains

V : *node set* in the constraint graph of clock domains

X : the first sequence of the sequence pair

Y : the second sequence of the sequence pair

1 Introduction

Floorplanning is a key research point in recent years and most of the work is focused on the aspects of area minimization, total wire length minimization and minimization of the timing delay of the critical path. With the development of VLSI technology, more and more circuits of different clocks are integrated into one chip. So, a chip with several or even more clocks is becoming common. Therefore, it is necessary to study the floorplanning with multiple clock domains .

2 Preliminaries

2.1 Problem Formulation of the Traditional

* This research is supported by the Hi-Tech Research and Development (863) Program of China under Grant (2003AA1Z1120) and is also supported by NSF grants CCR-0306298, partly by NSFC research project 90307017, 60176017, partly by Cross-Century Outstanding Scholar's fund of Ministry of Education of China, partly by National 863 plan projects 2004AA1Z1050, Shanghai Science and Technology committee Key project 04JC14015 and Shanghai AM R&D fund 0418.

Floorplanning

Let $B = \{b_1, b_2, ..., b_N\}$ be a set of n rectangular modules. $b_i \in B$ is associated with a three tuple $\{h_i, w_i, a_i\}$, where h_i, w_i and a_i denote the width, height, and area of b_i respectively. The aspect ratio of $b_i \in B$ denoted by $\tau_i = h_i / w_i$ varies from $\tau_{i,\min}$ to $\tau_{i,\max}$. A floorplanning is an assignment of rectangular modules with the coordinates of their bottom-left corners being assigned to (x_i, y_i) 's so that there are no overlaps between any two modules overlap with $\tau_{i,\min} \le \tau_i \le \tau_{i,\max}$. The object of placement/floorplanning is to minimize a specified cost metric such as the area of the chip, the total wire length and the combination of both area and total wire length.

2.2 Sequence pair[1]

In this paper, sequence pair is adopted. A sequence pair is a pair of combinations of module names. It represents the relationship between any two modules a and b as follows:

- If a is ahead of b in both sequences, module b is on the right of module a. The corresponding sequence pair is $(\cdots a...b..., ...a...b...)$.
- If a is ahead of b in the first sequence while behind b in the second sequence, module a is above module b . The corresponding sequence pair is $(\cdots a...b..., ...b...a...)$.

With the sequence pair, we apply the following seven operations to perturb the solution space.

Rotation: Rotate a module.

Swap blocks: Swap two blocks in X.

Swap blocks: Swap two blocks in Y.

Swap blocks: Swap two blocks in both X and Y.

Swap clock domains: Swap all blocks of two clock domains in X.

Swap clock domains: Swap all blocks of two clock domains in Y.

Swap clock domains: Swap all blocks of two clock domains in X and Y.

3 Floorplanning with multiple clock domains

For a system with multiple clock domains, its floorplanning is more complex than that of a system with only one. Relationships of blocks within the same clock domain and among different clock domains must be considered simultaneously. Usually, blocks within the same clock domain should be placed together and blocks of different clock domains should be placed according to the number of nets connected between them. In order to represent the relationship of blocks of a system with multiple clock domains, several cases are given below.

Given a sequence pair (X, Y) and two clock domains $D_{clcok_i} = \{b_{i1}, b_{i2}, ..., b_{im}\}$ and $D_{clock_j} = \{b_{j0}, B_{j1}, ..., B_{jn}\}$. It is assumed that each macro block has only one clock. , i.e. $D_{clock_i} \cap D_{clock_j} = \phi$ and $| D_{clock_i} \cup D_{clock_j} | = m + n$ are true.

It is also assumed that the sequence pair of

$D_{clock_i} \cup D_{clock_j}$ is

$(X, Y) = (...b_{i_{i_1}}...b_{i_{i_2}}.....b_{i_{i_m}}..., ...b_{j_{j_1}}...b_{j_{j_2}}.....b_{j_{j_n}}...)$,

The subsequence pair that only includes the blocks of $D_{clock_i} \cup D_{clock_j}$ is $(X', Y') = (b_{0i_1} b_{0i_2} ... b_{0i_m} b_{1j_1} b_{1j_2} ... b_{1j_n})$. We define the position of $b(i)$ in X' and Y' as $p_{X'}(b(i)) = i$ and $p_{Y'}(D_i) = i$ respectively.

With the assumption above, the following inadmissible cases can be obtained.

Case 1:

If $\exists b_{ik}, b_{il} \in D_{clock_i} (\forall 0 < k, l \le |D_{clock_i}|)$

$p_{X'}(b_{ik}) \le p_{Y'}(b_{ik})$ and $p_{X'}(b_{il}) \le p_{Y'}(b_{il})$

are true,

$\exists b_{jm} \in D_{clock_j} 0 < m \le |D_{clock_j}|$

$$\begin{cases} p_{X'}(b_{ik}) < p_{X'}(b_{jm}) \\ p_{X'}(b_{il}) > p_{X'}(b_{jm}) \\ p_{Y'}(b_{ik}) < p_{Y'}(b_{jm}) \\ p_{Y'}(b_{il}) > p_{Y'}(b_{jm}) \end{cases}$$

are true

Case 2

if $\exists b_{ik}, b_{il} \in D_{clock_i}$ ($\forall 0 < k, l \le |D_{clock_i}|$)

$p_{X'}(b_{ik}) \le p_{Y'}(b_{ik})$ and

$p_{X'}(b_{il}) \le p_{Y'}(b_{il})$ are true,

$\exists b_{jm} \in D_{clock_j} 0 < m \le |D_{clock_j}|$

$$\begin{cases} p_{X'}(b_{ik}) < p_{X'}(b_{jm}) \\ p_{X'}(b_{il}) > p_{X'}(b_{jm}) \\ p_{Y'}(b_{ik}) > p_{Y'}(b_{jm}) \\ p_{Y'}(b_{il}) < p_{Y'}(b_{jm}) \end{cases}$$

are true.

Case 1 and case 2 are not admissible.

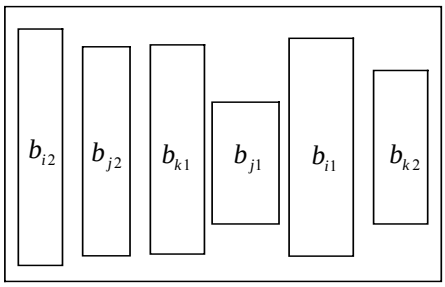


Figure 1: example of case 1

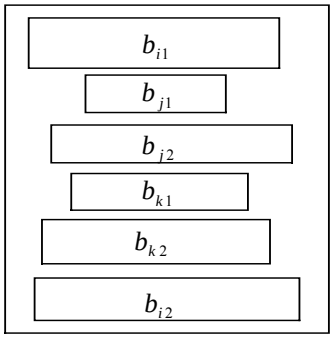


Figure 2: example of case 2

Definition 1: a floorplanning with multiple clock domains is admissible iff there are no case 1 and case 2 in it.

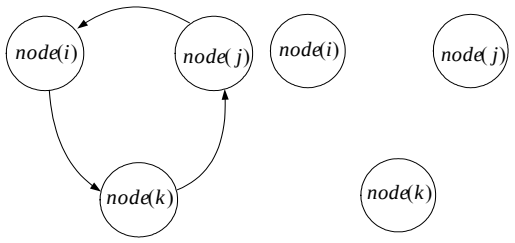
In order to identify the cases that are not admissible, we introduce the horizontal and vertical constraint graphs of clock domains.

Definition: Horizontal constraint graph of clock domains:

If a block of $clock_j$ is to the left of a block of $clock_k$, there exists a directed edge from $node(j)$ to $node(k)$.

Definition: Vertical constraint graph of clock domains:

If a block of $clock_k$ is above a block of $clock_j$, there exists a directed edge from $node(j)$ to $node(k)$.

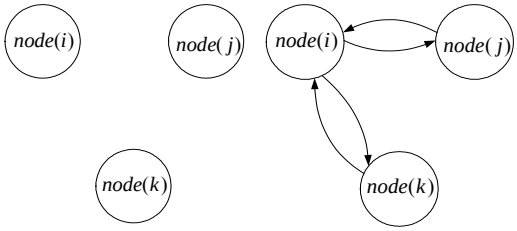


(a) (b)

(a) Horizontal constraint graph

(b) Vertical constraint graph

Figure 3: the constraint graphs of figure 1



(a) (b)

(a) Horizontal constraint graph

(b) Vertical constraint graph

Figure 4: the constraint graphs of figure 2

Theorem 1: A floorplanning with multiple clock domains is admissible iff there are no circles in its horizontal and vertical constraint graph of clock domains.

Proof:

First we prove that there are no circles in the horizontal and vertical constraint graph of an admissible floorplanning.

Proof:

For a floorplanning with K clock domains, if it is assumed that there is a circle in the horizontal constraint graph of an admissible floorplanning and the circle is from $node(1)$ to $node(2)$ to $...node(n)$, finally back to $node(1)$ without loss of generality.

According to the definition of constraint graph, the following formula can be obtained.

$$\exists\, b_{1i} \in clock_1,\ b_{2j} \in clock_2$$

$$\begin{cases} p_{X'}(b_{1i}) < p_{X'}(b_{2j}) \\ p_{Y'}(b_{1i}) < p_{Y'}(b_{2j}) \end{cases} \quad (1)$$

$$\exists\, b_{2k} \in clock_2,\ b_{3l} \in clock_2$$

$$\begin{cases} p_{X'}(b_{2k}) < p_{X'}(b_{3l}) \\ p_{Y'}(b_{2k}) < p_{Y'}(b_{3l}) \end{cases} \quad (2)$$

Because it is assumed that the floorplanning is admissible, the following can be obtained.

$$\forall\, b_{1i} \in clock_1,\ b_{2j} \in clock_2$$

$$\begin{cases} p_{X'}(b_{1i}) < p_{X'}(b_{2j}) \\ p_{Y'}(b_{1i}) < p_{Y'}(b_{2j}) \end{cases} \quad (3)$$

$$\forall\, b_{2k} \in clock_2,\ \forall b_{3l} \in clock_2$$

$$\begin{cases} p_{X'}(b_{2k}) < p_{X'}(b_{3l}) \\ p_{Y'}(b_{2k}) < p_{Y'}(b_{3l}) \end{cases} \quad (4)$$

Similarly $\forall\, b_{ik} \in clock_i \quad \forall b_{(i+1)l} \in clock_{i+1}$

$(1 \le i < n)$

$$\begin{cases} p_{X'}(b_{ik}) < p_{X'}(b_{(i+1)l}) \\ p_{Y'}(b_{ik}) < p_{Y'}(b_{(i+1)l}) \end{cases} \quad (5)$$

According to (5), (6) can be obtained.

$$\forall\, b_{1i} \in clock_1,\ b_{nj} \in clock_2$$

$$\begin{cases} p_{X'}(b_{1i}) < p_{X'}(b_{nj}) \\ p_{Y'}(b_{1i}) < p_{Y'}(b_{nj}) \end{cases} \quad (6)$$

According the assumption, there is a directed

edge from $node(n)$ to $node(1)$, so

$$\exists b_{1i} \in clock_1\ b_{nj} \in clock_2$$

$$\begin{cases} p_{X'}(b_{1i}) > p_{X'}(b_{nj}) \\ p_{Y'}(b_{1i}) > p_{Y'}(b_{nj}) \end{cases} \quad (7)$$

Since there is a contradiction ((6) and (7)), therefore the assumption is rejected and there are no circles in the horizontal and vertical constraint graphs of an admissible floorplanning.

The proof that a floorplanning is admissible if there is no circle in its horizontal and vertical constraint graphs of clock domains can be obtained from the definition of the admissible floorplanning in this paper.

The algorithm to convert an inadmissible floorplanning to an admissible one is given below.

Algorithm I

While V *is not empty{*

For $i = 1$ *to N*

Calculate the in-degree and out-degree in both horizontal and vertical constraint graph

Find min in-degree $\min_{in}$ *and min out-degree* $\min_{out}$

If($\min_{in} = 0$ *) or* $\min_{out} = 0$

Remove the corresponding node

Else

Find the node v with max-degree

Extract blocks of the corresponding clock domain and put them to the last of sequences with the relative position in this clock domain unchanged

End for

}

From theorem 1, it is clear that the process to transform the constraint graphs with circles to ones without circles is the process to transform inadmissible floorplannings to admissible ones.

Another method is to avoid inadmissible placement from the beginning. Details of the method are given below.

Algorithm II

begin

Let both of the initial sequences of X *and* Y *be*

$$X_{initial} = Y_{initial} = (b_{11}, b_{12}, ..., b_{1N1}, b_{21}, b_{22}, ..., b_{2N2}, ..., b_{K1}, b_{K2}, ..., b_{KNK})$$ Let

$$b(i) = X_{initial}(i)$$

Produce two random integers $\mathrm{elem1}$ *and* $\mathrm{elem2}$ *between* $[0, \mathrm{N})$ *.i.e.* $0 \le \mathrm{elem1}, \mathrm{elem2} < \mathrm{N}$.

If($rand(0,1) < change\_ratio$ *)*

TENT-CHANGE($h_{elem\,1}\ w_{elem1}$ *)*

If ($rand(0,1) < rotate\_ratio$ *)*

TENT-ROTATE $b(elem1)$

Else if $(rand(0,1) < swap\_ratio)$

If $clock(b(elem1)) = clock(b(elem2))$

TENT-EXCHGE $b(elem1), b(elem2)$ *in X or Y or both X and Y according to FAST-SP*[4].

Else

TENT-EXCHGE(C(b(elem1)),C(b(elem2))).

$\Delta\mathrm{cost} = \mathrm{cost(new\_place)} - \mathrm{cost(place)}$

if ($\Delta\mathrm{cost} > 0$ *)*

$place = new\_place$

else if($rand(0,1) > e^{-\Delta cost/T}$)

$place = new\_place$

$times = times + 1$ *od*

od

soft block adjustment

end

In this paper, algorithm II is adopted.

The construction method[6].is adopted for the adjustment of width and height of each soft block, i.e. there are 11 candidate widths and heights for each soft block. Each candidate width is calculate as follows:

$$w_{i1} = \sqrt{a_i / \underline{\tau_i}}$$

$$w_{i2} = 0.9 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.1 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i3} = 0.8 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.2 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i4} = 0.7 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.3 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i5} = 0.6 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.4 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i6} = 0.5 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.5 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i7} = 0.4 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.6 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i8} = 0.3 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.7 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i9} = 0.2 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.8 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i10} = 0.1 \cdot \sqrt{a_i / \underline{\tau_i}} + 0.9 \cdot \sqrt{a_i / \overline{\tau_i}}$$

$$w_{i11} = \sqrt{a_i / \overline{\tau_i}} \qquad (8)$$

The candidate height corresponding to each candidate width is calculate as

$$h_{i,j} = a_i / w_{ij} \quad (0 < j \le 1) \qquad (9)$$

4 Experimental Results

Based on the algorithm II, we implement our algorithm in c++ language in a 3.8G Dell PC with 1G bytes memory.

We test our algorithm by ami49 and a big benchmark test196 which is generated from four ami49s. The information of the benchmark is shown in table 1.. For each block, $0.33 \le \tau_i \le 3$

The number of iteration times in each temperature is 400. The cost function is $\lambda_1 \text{cost}(area) + (1-\lambda_1)\text{cost}(wire)$ Experiment results are shown in table 2 and figure 5 and 6.

Table 1:

circuit	Number of blocks	Number of clocks	Total area of all blocks
ami49	49	5	35.45
Test196	196	10	141.8

Figure 5 Floorplanning result of ami49 ($\lambda_1 = 1$)

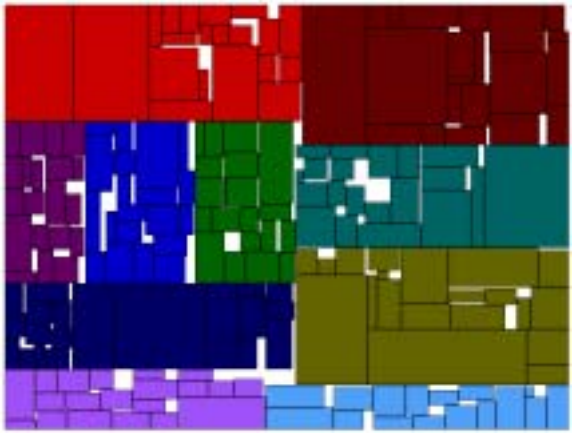


Figure 6: Floorplanning result of test196 ($\lambda_1 = 1$)

In figure 5 and 6, different clock domains are represented by different colors.

Experiment results show that blocks within the same clock domain are placed together and blocks of different clock domains are divided clearly by the algorithm in this paper.

5 Conclusion

This paper presents the problem of the floorplanning with multiple clock domains and the corresponding algorithms. The motivation of this work was that more and more clock domains exist in an SOC. Experimental results show that we can get reasonable results by the algorithm.

In the experiment, ten clock domains were packed efficiently in a reasonable time. With the development of new VLSI technologies, more and more clocks domains will appear in a single chip. Therefore, our algorithm will be more significant than before.

References:

[1] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 1518–1524, Dec. 1996.

[2] X. Tang and D. F. Wong, "FAST-SP: A fast algorithm for block placement based sequence pair," *Proc. ASPDAC*, pp. 521–526, 2001.

[3] Hua Xiang, Xiaoping Tang, and Martin D. F. Wong Bus-Driven Floorplanning *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, VOL. 23, NO. 11, pp. 1522-1530 NOVEMBER 2004

[4] D.F.Wong and C.L.Liu, "A New Algorithm for Floorplan Designs" *Proc. 23rd ACM/IEEE Design Automation Conf.* pp.101-107, 1986

[5] 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 pp1518-1524, Dec 1996

[6]Jae-Gon Kim, Yeong-Dae Kim A Linear Programming-Based Algorithm for Floorplanning in VLSI Design *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems,* Vol. 22, No. 5, MAY 2003 pp 584-592

Table 2:

circuits	area(mm^2) / total wire length(mm)			
	$\lambda_1 = 1$		$\lambda_1 = 0.5$	
	best	average	best (area)	average
ami49	36.58/1493.06	37.38/1471.105	38.10/755.91	39.02/849.72
Test196	150.28/9004.22	155.07/9544.16	158.99/5921.901	168.49/5541.84