第26卷 第1期 半 导 体 学 报 Vol.26 No.1
2005年1月 CHINESE JOURNAL OF SEMICONDUCTORS Jan.,2005

A New Clustering-Based Partitioning Method for VLSI Mixed-Mode Placement *

Lü Yongqiang, Hong Xianlong, Yang Changqi, Zhou Qiang, and Cai Yici

(Deptartment of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: An efficient partitioning algorithm for mixed-mode placement, extended-MFFC-based partitioning, is presented. It combines the bottom-up clustering and the top-down partitioning together. To do this, designers can not only cluster cells considering logic dependency but also partition them aiming at min-cut. Experimental results show that extended-MFFC-based partitioning performs well in mixed-mode placement with big pre-designed blocks. By comparison with the famous partitioning package HMETIS, this partitioning proves its remarkable function in mixed-mode placement.

Key words: mixed-mode placement; extended MFFC; HMETIS; clustering; partitioning
EEACC: 2570 **CCACC:** 7410D
CLC number: TN402 **Document code:** A **Article ID:** 0253-4177(2005)01-0022-07

1 Introduction

At present, in spite of so many great achievements made in VLSI/ULSI placement area, such as quadratic placements[1,2], detailed placements[3,4], incremental placements[5~7], placement flow designs[3,8] and so on, the advanced technology has brought out a great challenge to standard cell placement. Designers are required to integrate from ten thousands to hundred thousands even millions of standard cells onto one single chip. Furthermore, to improve design process and time-to-market, the concept of design hierarchy and reuse is becoming very important. Some critical parts of circuits are designed separately and then integrated onto the chips as pre-designed macros, such as datapath, memory and intellectual property (IP), and so on. It is very difficult to place standard cells and those pre-designed macros together directly in mixed mode. Firstly, though some placement methods, for example Q-place, are good at placing normal cells, we may not use them directly due to the big pre-designed blocks. Secondly, though floorplaning methods are good at placing big blocks, we may not use it directly either due to the normal cells outside the pre-designed blocks. Consequently, the hierarchical mixed-mode placement (HMMP)[3] process is employed to meet this requirement. HMMP includes a partitioning, a floorplaning, and a cell placement process. As a part of HMMP, partitioning takes an important role in the entire process. It acts as a preprocessor to decrease the problem scale. Also it joins the floorplaning and the cell

* Project supported by National Natural Science Foundation of China (Nos. 60121120706, 60076016), National Natural Science Foundation of USA (No. CCR-0096383), State Key Development Program for Basic Research of China (No. G1998030403), and National Hi-Tech Research & Development Program of China (No. 2002AA1Z1460)

Lü Yongqiang male, was born in 1980, PhD candidate. He is majoring in VLSI layout design. Email: luyj01@mails.tsinghua.edu.cn

Hong Xianlong male, was born in 1940, professor. His research focuses on VLSI design automation.

Received 22 April 2004, revised manuscript received 24 June 2004

placement process together. The quality of partitioning affects much the final placement result.

However, people are accustomed to using clustering or partitioning techniques in the beginning of hierarchical methods, such as maximum fan-out free cone (MFFC) clustering[9], first choice clustering[10], ratio-cut partitioning[11], spectrum-based partitioning[12], the generalization of the FM-algorithm with look-ahead scheme[13], and some two-way or multilevel partitioning methods. In fact, we have noticed that the conventional bottom-up clustering methods try to group the inner nets as many as possible but ignore reducing the interconnection among clusters; while the conventional top-down partitioning methods try to reduce the interconnected nets among blocks but ignore the logic dependency of cells in blocks. In this paper, a new method unites the advantages of the two by selecting the model of extended-MFFC-based (EMFFC) partitioning, which is specially designed for mixed-mode placement with big pre-designed macros, not only to group cells into clusters with logic dependency as close as possible, but also to partition clusters into blocks with interconnections among them as few as possible. It first creates clusters surrounding the macros by the connection, then partitions the clustered circuits into blocks with the min-cut objective. The EMFFC-based partitioning is combined into HMMP in this paper, replacing the former ratio-cut partitioning[11]. Experimental results on five DEF/LEF benchmarks show that the EMFFC-based partitioning is very effective in the mixed-mode placement.

2　HMMP process

Wu and Hong first proposed HMMP process[3], and they achieved very good experimental results. Its flow begins with a circuit partitioning, which is followed by a block-based floorplaning. Then a quadratic placement in each block and a final detailed placement on entire chip will be employed.

In this paper, the partitioning part of the process executes the extended-MFFC-based partitioning. It is adopted to gather standard cells with closest connections into virtual soft blocks and convert the mixed circuit into the block circuit. Then a simulated annealing optimization procedure is employed as floorplaning[3,8] to place the blocks onto the chip. After floorplaning, a Q-place[1,2] (global placement) is done in each virtual block respectively to obtain the cells' global placement. Finally, the process adopts the detailed placement[3,4] to eliminate the overlap of global-placed cells and allocate the cells into the defined rows, aligned with sites. We can see that the partitioning makes it available for floorplaning and Q-place to work together.

The HMMP in this paper differs from that of Ref. [3] just in the partitioning stage; the former uses the EMFFC-based partitioning, and the latter uses the ratio-cut partitioning method[11].

3　EMFFC-based partitioning

Actually, the extended-MFFC-based partitioning is composed of the extended-MFFC clustering and the HMETIS partitioning. Before the introduction of extended-MFFC clustering, the MFFC clustering will be presented firstly.

3.1　MFFC clustering

The MFFC clustering[9] includes decomposition technique and splitting technique. The splitting technique is used when a cluster decomposed is too large. It splits the cluster into small ones. Due to the length restriction, only the decomposition technique will be described here.

The decomposition always starts with a set of cells, and traces all in-direction connections of each cell in set to cluster those whose all outputs are in the set. For example, a circuit as Fig. 1 shows will be decomposed into A, B, C, D, E five clusters. The cluster A is decomposed from the set with the only cell V, and it absorbs cells V1 because the output of the cell V, i. e. V1, is in the set.

While the cell S cannot be clustered into the set because there exists one output of it, i.e. the cell V2, being outside the set.

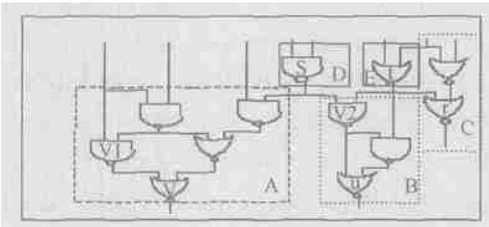

Fig. 1 MFFC decomposition

3.2 Extended-MFFC clustering for mixed-mode

The extended-MFFC clustering is on the basis of the extended-MFFC decomposition technique, and it is specially designed for the placement of circuits with big predesigned macros. The extended-MFFC decomposition technique will be introduced first.

3.2.1 Extended-MFFC decomposition technique

Extended-MFFC decomposition technique differs from the conventional MFFC decomposition technique in initial decomposition-from cells selection. The extended-MFFC traces from a set of cells; while the conventional MFFC traces from one single primary output (PO) or a certain cell. Given initial set $S = \{c_1, c_2, c_3, \cdots, c_n\}$, we are required to decompose a new cluster from S using extended-MFFC. Before this, we define:

input(c): a cell ingoing net of c connects.

output(v): a cell outgoing edge of v connects.

{output(v)}: the set of all the output(v)s.

{output(v)}| S: the output(v)s that are in the set of S.

{output(v)}|^S: the output(v)s that are not in the set of S.

| S| : the number of the elements in S.

So, {output(v)} = {output(v)}| S + {output(v)}|^S, and {output(v)}| S is exclusive to {output(v)}|^S.

$F(v)$:| {output(v)}| S| / | {output(v)}|^S| , is the factor which indicates whether the cell v should be clustered into set S or not. The maximum is 1, and the larger it is, the more connections there are between v and S. In this paper, we ignore those output(v)s, which are not in the set S but have been clustered by other sets.

In extended-MFFC decomposition, a clustering factor will be specified before clustering begins. When we judge whether a cell v would be clustered into the set or not, we only need to check $F(v)$. If it is greater than the clustering factor specified, the cell v will be clustered into S, otherwise not. So, when the clustering factor is set to 1, the extended-MFFC clustering is just the same as MFFC clustering.

Suppose we are checking one input(c_i), and it is the cell v. We check the $F(v)$ to determine whether or not to absorb v. Then we go on this process from the next input(c_i) until all cells in S are checked.

3.2.2 Extended-MFFC clustering from hard blocks

The problem of clustering in circuits with big pre-designed blocks (hard blocks) becomes subtle because of the existence of hard blocks. If we just cluster as if those hard blocks did not exist, the result must be not as good as taking them into account. The extended-MFFC clustering is an overall clustering solution of the standard cell placement with big hard blocks. It begins with hard blocks, creating clusters surrounding them. These clusters are created by the connections between clusters and hard blocks. By controlling the clustering factor mentioned in section 3.2.1, we can achieve the clusters surrounding hard blocks ring by ring.

For example, as Fig. 2 demonstrates, we are now clustering from the hard block. First, we divide the hard block into four parts 1, 2, 3, 4, and cluster the cells in those parts into sets. Then we decompose from them to trace new clusters one by one under the current clustering factor. After that, we go on decomposing from those new-created clusters under a new smaller clustering factor until the clustering factor goes out the proper value. So we obtain the clusters ring by ring just as the area A in Fig. 2. The nearest to the hard block are the closest in connection to the cells in it, and the clusters in the same fill (in Fig.

2) are created under the same clustering factor. If not all cells are clustered into sets after the clustering factor goes out proper value, we will cluster them from primary outputs (POs) of circuits with the algorithm of extended-MFFC clustering. So, even if in the placement of circuits with hard blocks that have little connections with other normal cells, our algorithm can still work well. The algorithm can be done as following algorithm pseudo codes.

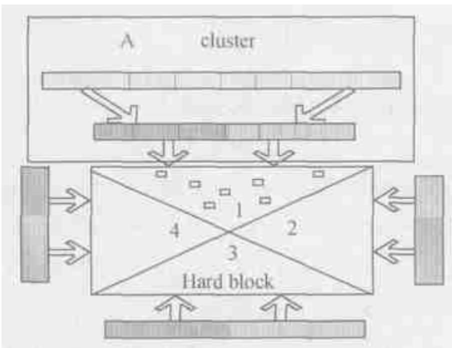

Fig. 2　Extended MFFC clustering from a hard block

Algorithm extended-MFFC clustering for mixed-mode placement

(1) create the root clusters from hard blocks
(2) while (clustering factor ≥ proper value)
(3) for each root cluster
(4) decompose_new_cluster_from_root;
(5) set new cluster as new root;
(6) end for
(7) decrease the clustering factor properly;
(8) end while
End algorithm

The process decompose_new_cluster_from_root in the algorithm works in the way as mentioned in section 3.2.1.

After the clustering has done, HMETIS will be invoked to partition clusters into a certain number of partitions.

3.3 HMETIS package

HMETIS[14] is a library downloaded from http://www-users.cs.umn.edu/~karypis/metis. It can partition a hyper-graph into partitions with min-cut among them. It can also take the standard deviation of areas of blocks into account. We employ it to partition the clustered circuits in this process. All options used in this paper are default as described in Ref. [14].

3.4 Determine the parameters

In this partitioning, some parameters must be determined before it executes. In this paper, we are required to determine these parameters: the clustering factor, the number of blocks, and the blocks' area deviation.

3.4.1 Clustering factor

This parameter is introduced in section 3.2 and is used for gaining the clusters around hard blocks. It is more independent of later process, as it is to improve the logic dependency of clusters, so we decide to determine its value first. It is set to 1.0 initially, and stop decreasing at a certain value between 0.1 and 1.0. The less it is stopped at, the fewer blocks are inputted to HMETIS, and then the running time is cut down; while the cut number may increase after partitioning. So we need to determine the minimum value of cluster factor by experimenting much. Shown in Fig. 3, this is done on the circuit of two macros, and the other two parameters are set to four blocks and standard area deviation. We find out that we get the shortest total wire length when the minimum clustering factor is set to 0.6. So it is used as default in later experiments.

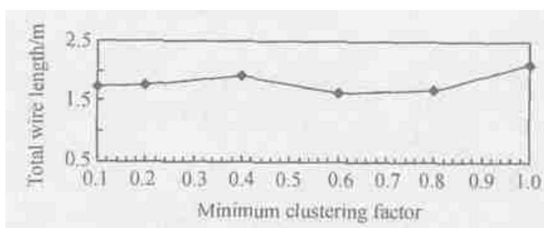

Fig. 3　Clustering factor and total wire length

3.4.2 Number of blocks

How many blocks (soft blocks) the normal cells in circuits will be partitioned into is a conventional problem.

In current research ,how to determine the number is difficult. Due to the different applications ,the block number specification may be different. There may be a certain relation between the optimal block number and the circuit logic dependency ,but we cannot find it yet.

In our HMMP process ,the block number specification is also subjected to the time complexity of the later floorplaning. Under average running time ,we find out the total number of soft blocks and hard blocks is proper in the range of 6～30. Because the number also affects the global placement ,which places cells in each block ,we cannot give a proof to the optimal number. We have no choice but to experiment on it. However ,one single block cannot contain too many cells in it because of the complexity of the Q-place. So this parameter may range with the corresponding scale of circuits. Figure 4 gives the diagram of the number of soft blocks and the final placement result (total wire length) on the same circuits as the final experimental results. Due to length restriction ,here we give one circuit with two pre-designed blocks in Fig. 4(a) and one with six in Fig. 4(b) ,under the clustering factor from the result of section 3.4.1 and standard area deviation. From the data ,we can find out that the optimal block number is 4.

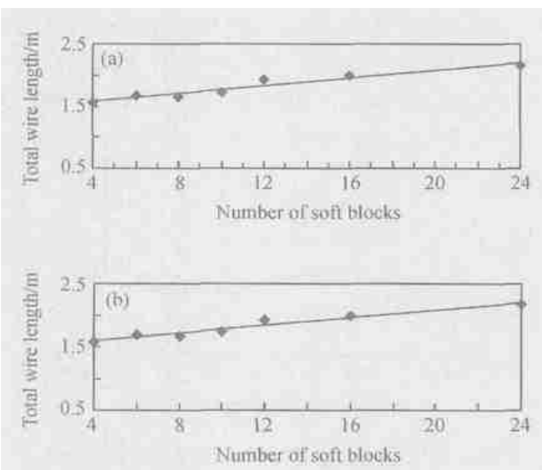

Fig. 4 Number of blocks and total wire length (a) With two hard blocks; (b) With six hard blocks

3.4.3 Controlling the areas of blocks

In conventional partition , a standard deviation of blocks' areas may be preferred. But in the EMFFC-based partitioning of the mixed-mode process ,blocks with a certain area deviation may be more fit for floorplaning. In fact ,floorplan is good at placing such blocks. However ,the areas of blocks ,indicating the number of cells in blocks , will influence on the Q-place in global placement ,because the number of cells can affect the final solution accuracy of it. So it is also subjected to two conflictive aspects. For simplicity, the standard deviation is used in all experiments in this paper.

4 Experimental results

The proposed extended-MFFC-based partitioning algorithm is implemented in C++ language and works on Sun Workstation V880. Intel Corp. provides the test circuits whose characteristics are shown in Table 1 ,including cell number ,macro number ,net number ,and total area of macros versus total area of chip in percentage.

Table 1 Characteristics of test circuits

Case	# cell	# macro	# net	Macro area/ %
Block2	7094	2	10049	37
Block6	5996	6	10049	47
Block8	5662	8	10049	50
Block9	5895	9	10049	53
Block10	5151	10	10049	57

Table 2 gives the value of the parameters used in extended-MFFC-based partitioning. These parameters are determined as section 3.4 describes.

Table 2 Parameters

Parameter	Min clustering factor	# block	Area deviation
Value	0.6	4	Standard

Table 3 gives the experimental results of extended-MFFC-based HMMP with comparison to the results of HMMP[3]. TWL is the abbreviation of total wire length , and its unit is μm.

We can see that the extended-MFFC-based HMMP outperforms the former HMMP on all test circuits. The improvement ratio ranges between 7.4 % and 43.6 %.

Table 3　TWL of EMFFC-based HMMP and Ref. [3]

Circuit	TWL of Ref. [3]	TWL of EMFFC-based	Improve ratio/ %
Block2	1692002	1566458	7.4
Block6	2029252	1492078	26.5
Block8	2022614	1481309	26.8
Block9	2345725	1948917	16.9
Block10	2130945	1201478	43.6

Table 4 gives the running time of our algorithm. We can see in Table 4 that the running time of partitioning (col. partition) is only a small part of total running time (col. total), at most 2.5 %. It is very efficient for us to gain so good total wire length.

Table 4　Running time of EMFFC-based partitioning

Circuit	Partition / s	Total / s	Ratio of total/ %
Block2	8	322	2.5
Block6	7	381	1.8
Block8	6	426	1.4
Block9	7	516	1.4
Block10	5	290	1.7

We also give the comparison between the extended-MFFC-based partitioning and the HMETIS partitioning in mixed-mode placement. HMETIS uses all default options. We still employ the process of HMMP, but use extended-MFFC-based partitioning and HMETIS as partitioning method respectively. From Table 5, we can find out that the extended-MFFC-based partitioning outperforms HMETIS by the ratio ranging from 7.6 % to 33.7 % on all circuits with about 50 % time conservation. Our algorithm is more suitable for large-scale placement than HMETIS.

Table 5　Comparison between EMFFC-based portioning and HMETIS

Case	HMETIS		Our algorithm		TWL improving ratio/ %
	TWL	Time/ s	TWL	Time/ s	
Block2	1.70×10^6	14	1.57×10^6	8	7.6
Block6	1.67×10^6	14	1.49×10^6	7	10.5
Block8	1.68×10^6	13	1.48×10^6	6	11.9
Block9	2.94×10^6	14	1.95×10^6	7	33.7
Block10	1.31×10^6	13	1.20×10^6	5	8.0

5　Conclusion

The method of extended-MFFC-based partitioning can help improve the quality of mixed-mode placement with big pre-designed blocks. In the next stage, timing, congestion, and other mixed-mode kinds of placement such as circuits with various heights of big cells will be taken into account.

References

[1] L ü Yongqiang, Hong Xianlong, Hou Wenting, et al. An efficient partitioning method in quadratic placement. Chinese Journal of Semiconductors, 2004, 25(3): 272

[2] Hou Wenting, Yu Hong, Hong Xianlong, et al. A new congestion-driven placement algorithm based on cell inflation. Chinese Journal of Semiconductors, 2001, 22(3): 275

[3] Wu Weimin, Hong Xianlong, Cai Yici. A hierarchical mixed mode placement algorithm. Journal of Software, 2001, 12(12): 1752

[4] Yang Changqi, Hong Xianlong, Cai Yici, et al. Data-path placement based on regularity extraction and implementation. Chinese Journal of Semiconductors, 2004, 25(8): 925

[5] Li Zhuoyuan, Wu Weimin, Hong Xianlong. Incremental placement algorithm for timing and routability optimization. Chinese Journal of Semiconductors, 2004, 25(2): 158

[6] Li Zhuoyuan, Wu Weimin, Hong Xianlong. Incremental placement algorithm for standard cell layout. Chinese Journal of Semiconductors, 2002, 23(12): 1338

[7] Li Zhuoyuan, Wu Weimin, Hong Xianlong. New incremental placement algorithm based on integer programming for reducing congestion. Chinese Journal of Semiconductors, 2004, 25(1): 30

[8] Yang Changqi, Hong Xianlong, Yang Hannah Honghua, et al. Recursively combine floorplan and Q-place in mixed mode placement based on circuit's variety of block configuration. Proceeding of ISCAS, 2004

[9] Cong J, Xu Dongmin. Exploiting signal flow and logic dependency in standard cell placement. Proceedings on Asia Pacific Design Automation Conference, Mukuhari, Japan, 1995: 399

[10] Karypis G, Kumar V. Multilevel K-way hypergraph partitioning. Proc Design Automation Conf, 1998: 343

[11] Wei Y C, Cheng C K. Ratio cut partitioning for hierarchical designs. IEEE Trans CAD, 1991, 10(7): 911

[12] Chan P, Schlag M, Zien J. Spectral K-way ratio-cut partitioning and

clustering. Proc 30th ACM/ IEEE Design Automation Conf ,1993

[13] Sanchis L. Multiple-way network partitioning. IEEE Trans Computers ,1989 ,38 :62

[14] Karypis G, Kumar V. HMETIS, a hypergraph partitioning package version 1.5.3 ,Manuscript ,1998

一种新的基于结群的混合模式划分方法 *

吕勇强 洪先龙 杨长旗 周 强 蔡懿慈

(清华大学计算机科学与技术系, 北京 100084)

摘要: 提出一种专用于带有预设计模块的混合模式布局的划分算法. 它基于扩展的 MFFC 结群算法,结合自下而上的结群和自上而下的划分为一体进行混和模式下的划分. 这样不仅可以使划分能够考虑电路本身的逻辑依赖,而且可以得到很好的"最小割"划分结果. 实验结果表明,这种划分算法在层次式混合模式布局流程里起到了显著的作用. 将此算法和当今国际上著名的划分包 HMETIS 进行比较,结果表明此算法有一定的优势.

关键词: 混合模式布局; 扩展 MFFC; HMETIS; 结群; 划分

EEACC: 2570 **CCACC:** 7410D

中图分类号: TN402 **文献标识码**: A **文章编号**: 0253-4177(2005)01-0022-07

* 国家自然科学基金(批准号:60121120706 和 60076016),美国国家自然科学基金(批准号:CCR-0096383),国家高技术研究发展计划(批准号:2002AA1Z1460)和国家重点基础研究发展规划(批准号:G1998030403)资助项目

吕勇强 男,1980 年出生,博士研究生,研究方向为大规模集成电路布图布局.

洪先龙 男,1940 年出生,教授,研究领域为大规模集成电路设计自动化.

2004-04-22 收到,2004-06-24 定稿