A HIERARCHICAL PLACEMENT PROCEDURE WITH A SIMPLE BLOCKING SCHEME

Shinichi Murai, Hiroo Tsuji, Morio Kakinuma,
Kazumichi Sakaguchi and Chiyoji Tanaka

Mitsubishi Electric Corporation
Kamakura, Japan

The outline of a hierarchical placement procedure utilizing a simple blocking scheme is described with the results of the application to the DSA-MOS gate arrays. Indirect clustering value is introduced for the blocking, i.e. grouping of modules under block size restriction. The system including the procedure has been successfully applied to the design of MOS gate arrays with effectively no manual assistance.

Introduction

Module placement problem as well as wire routing problem in the layout design of LSI chips or printed circuit boards is one of the problems which the heaviest effort to apply computer has been made upon. Automatic wire routing has been quite successful in the practical environment to such an extent that the result of automatic routing is better than or at least as good as the manual routing especially when the physical constraints such as the shape of the routing area or the way of routing are simple or regular[1]. Automatic placement, on the other hand, does not seem to be as successful as automatic wire routing in the sense that the result of the automatic placement can usually be improved by manual effort in the practical environment.

A well accepted and therefore a quite reasonable solution to this is the interactive approach backed by the graphics capabilities[7 et al] where manual intervention plays very important roles especially when the physical constraints such as module shapes are complex.

Hierarchical placement is an effective approach to automatically obtain better placement, but it requires considerable amount of computation time with the complex partitioning or blocking scheme[2].

Although it incorporates extensive manual intervention features such as "seeding", HPS is a hierarchical placement subsystem with a simple blocking scheme, whose objective is to completely automate the placement procedure with the performance comparable to manual placement in the environment where the physical constraints are basically regular as in the masterslice LSI's.

The next section describes the outline of HPS with the idea behind it, followed by the definition of terms and constraints. The fourth section includes the more detailed discussion of each procedure, especially blocking, of HPS. The last section shows the results of the actual use of HPS and discusses the characteristics of HPS appeared in the results.

Concepts and Outline of HPS

The straightforward placement algorithm which would give the optimum solution has not yet been found in the practical environment, and most of the placement systems appeared in the open literature employ the heuristic iterative placement improvement techniques usually with the simple constructive initial placement procedure. This HPS (Hierarchical Placement Subsystem) too employs the iterative improvement procedure as one of the schemes to obtain better results, but the key techniques of the HPS are blocking and the better initial placement algorithm.

According to the experiments of Hanan et al[2] the better initial placement gives better final results even in the case where the iterative improvement procedure plays the central role. With this in mind, HPS employs the initial placement procedure based on clustering[3] and two dimensional top down placement procedure[6] which gives the better initial placement than the ordinary constructive initial placement procedure[4].

The motives of the other key technique of HPS i.e. blocking or hierarchical placement is the observation that the simultaneous interchange of more than two modules in iterative improvement procedure is often effective in obtaining better placement[5], and the experimental results in 2. In order to reduce the computation time, which is otherwise quite serious disadvantage of hierarchical placement, and yet obtain better placement, a new simple blocking algorithm,which is later described in detail, is devised and implemented for HPS.

Thus, the outline of HPS is as shown in Fig.1. The first step of HPS is blocking i.e. grouping modules into blocks based on the interconnection strength between modules. After blocking HPS continues grouping (clustering) until whole circuit is grouped into a cluster. After the completion of blocking and clustering HPS first obtains the initial placement of the blocks by the two dimensional top down, placement procedure, followed by the iterative block placement improvement by FDPR technique[4]. After the block placement finished, HPS obtains the module level placement again by the two dimensional top down placement procedure, followed by the module level FDPR procedure.

Definition of Terminology

The definition of the terms used in this paper basically follows 2. Described is the placement problem in terms of placing internal and external modules into internal and external slots respectively. A cluster is a group of internal and external modules. The size of a cluster is the number of modules in the cluster. The cluster of size 1 is a module. A bench is a collection of consecutive internal slots. A block is a special type of cluster which consists only of internal modules and fits into a bench.

The arrangements of slots which HPS can handle is shown in Fig.2. The internal slots are arranged in a matrix fashion where the distance between the adjacent rows or columns may vary with each other. The external slots are arranged along the four sides of the rectangular routing area within which all the internal slots are contained.

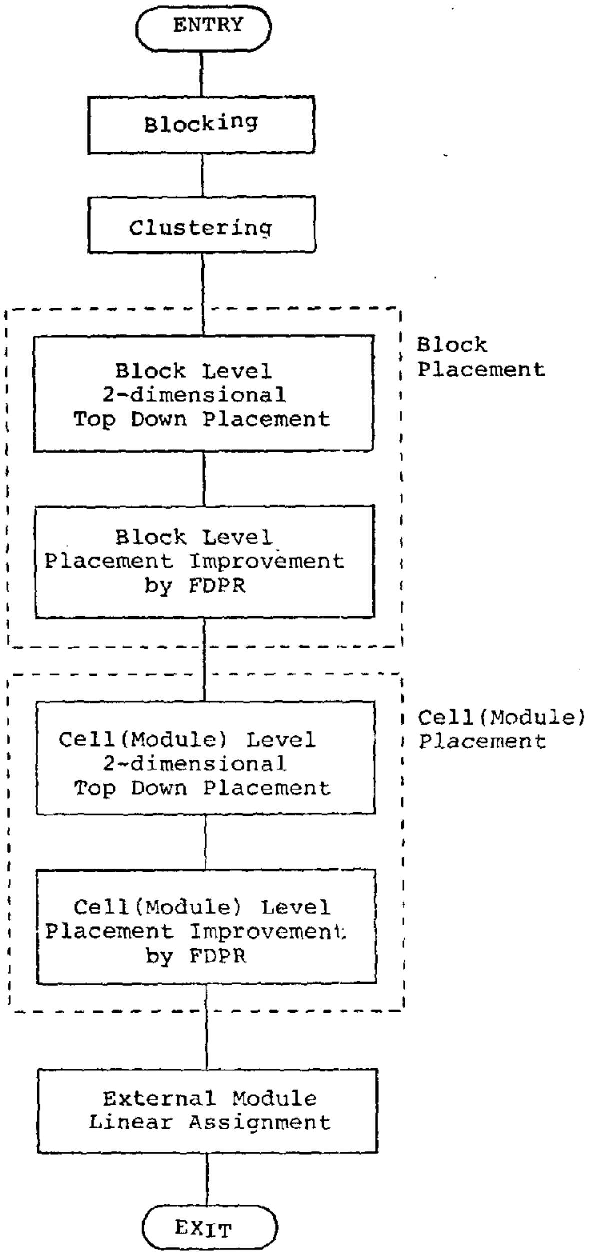
ENTRY
Blocking
Clustering
Block Level 2-dimensional Top Down Placement
Block Level Placement Improvement by FDPR
Block Placement
Cell(Module) Level 2-dimensional Top Down Placement
Cell(Module) Level Placement Improvement by FDPR
Cell(Module) Placement
External Module Linear Assignment
EXIT

Fig. 1. General Flow of HPS

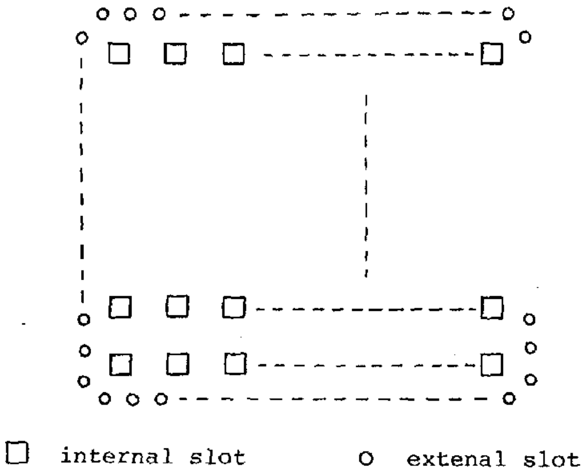

Fig. 2. Arrangement of Slots

Placement Procedure

Blocking

As desicribed before placement is done in two levels, block level and module level. In order to carry out block level placement in the same manner as the module level placement, it is necessary to form blocks whose sizes are equal to or smaller than the bench size, and for all the benches to have the same bench size. This restriction makes the blocking procedure different from the later clustering procedure.

The outline of the blocking procedure is shown in Fig.3. As shown in the figure, blocking consists of two procedures, blocking 1 and blocking 2. Blocking 1 is executed when the number of blocking cores is smaller than the number of benches where a "blocking core" is a special cluster whose size is larger than 1 and around which a block is formed. The number of blocking cores is always smaller than or equal to the number of benches. Blocking 2 is carried out when the number of the blocking cores is equal to the number of benches. Preprocess deals with the manual blocking or clustering and manual block placement. Unless the number of blocking cores has already reached the number of benches by manual specification, blocking always starts from blocking 1 procedure.

In blocking 1 a pair of clusters whose clustering value is the strongest and the sum of the sizes of the both clusters is not larger than the bench size is searched for and grouped to form a new blocking core. The clustering value CV_{ij} between elements i and j is given by:

$$CV_{ij} = f(S_i)\frac{CN_{ij}}{T_i - CN_{ij}} + f(S_j)\frac{CN_{ij}}{T_j - CN_{ij}}$$

where:

CN_{ij} = connection strength between element i and j

T_i = total connection strength to element i

T_j = total connection strength to element j

$f(S_i)$ = some function of the size of element i

$f(S_j)$ = some function of the size of element j

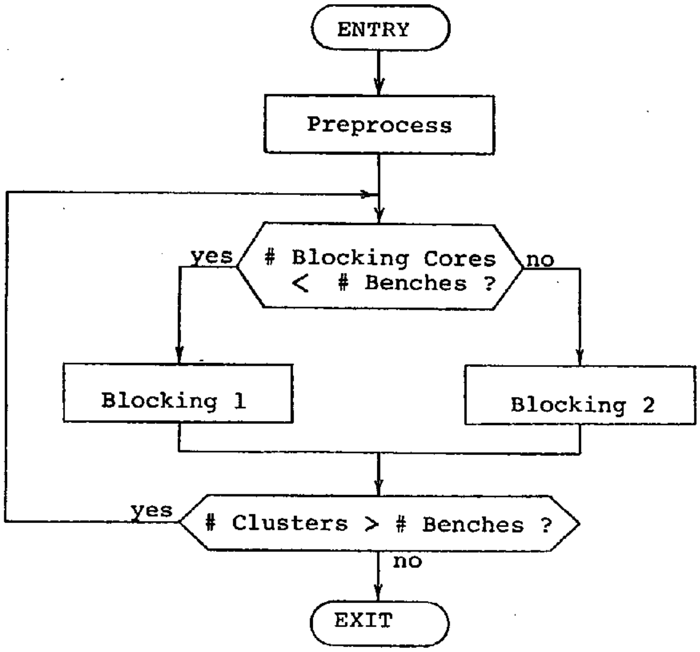

Fig. 3. General Flow of Blocking

The connection strength CN_{ij} is defined as follows:

$$CN_{ij} = \quad P_k \cdot g(K_{ij})$$

where:

P_k = weight of signal (net) k

I = a set of signals connected to element i

J = a set of signals connected to element j

$g(K_{ij})$ = weight associated with edge K_{ij} between elements i and j

There exists the following three cases in this procedure;

i) two blocking cores are grouped to form a new blocking core (the number of blocking cores is reduced by one)
ii) a module is grouped to a blocking core (the number of blocking cores does not change)
iii) two modules form a new blocking core (the number of blocking cores is increased by one).

When the number of blocking cores reaches the number of benches, blocking 2 procedure is executed. In blocking 2 clustering is allowed only between blocking cores, or at least one element of a pair must be a blocking core. In other words the above case iii) is not allowed.

Thus blocking 1 and blocking 2 procedures are repeated until all the clusters become blocking cores, which are now blocks.

In the course of blocking, especially at the very last stage, it occasionally occurs that there is no pair with a positive clustering value, mainly because of the block size restriction. If this happens, the cluster with the maximum indirect clustering value is grouped into the corresponding blocking core. The cluster with the maximum indirect clustering value is such cluster as shown in Fig.4.

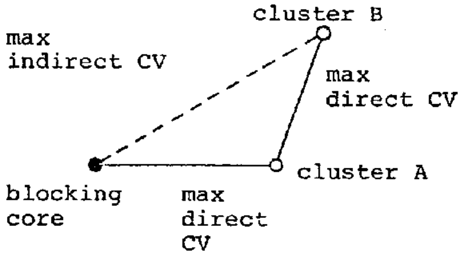

Fig. 4. Indirect Clustering Value

Clustering

After blocking is completed, pairwise clustering[3] is executed between the blocks just formed and also the external modules which are not blocked during the previous blocking phase until all the blocks and external modules are drawn to a single cluster. Clustering procedure is the same as the blocking 1 procedure except that the external modules are also clustered together, and no restriction is imposed on the sizes of the resultant clusters even though some size function appears in the formula of the clustering value.

If the circuit consists of plural independent subcircuits where the clustering value between them would be zero, clustering between the subcircuits is forced for the ease of placement procedure.

Blocking and clustering process is expressed as a clustering tree as shown in Fig.5.

Block Placement

When blocking and clustering is finished, block placement is executed in two phases. The first phase is the initial block placement, and the second phase is the iterative block placement improvement.

2-dimensional top down placement. Although the linear placement algorithm proposed in 3 would give quite satisfactory results to such linearly structured devices as discussed in 8, it requires some adjustment for the ordinary 2-dimensionally structured devices. One way of the adjustment is first obtain linear ordering of the modules and then fold it into 2-dimensional structure as shown in Fig.6. This is simple and fast but has obvious shortcomings for the ordinary 2-layers rectangular routing scheme.

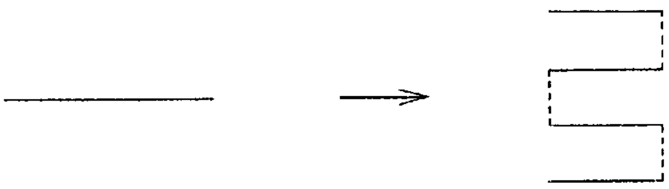

Fig. 6. Transformation from Linear Ordering to 2-dimensional Placement

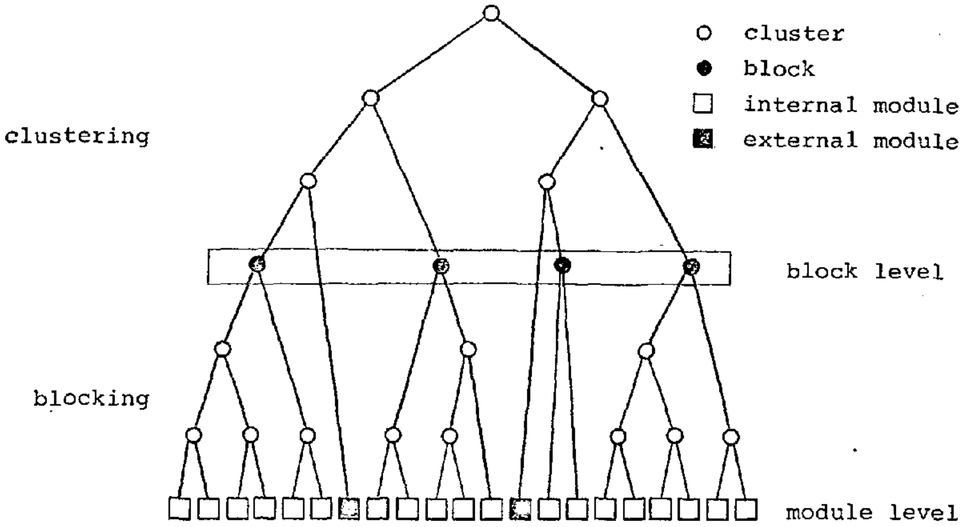

Fig. 5. Blocking and Clustering Process

Another way of obtaining 2-dimensional placement is the straightforward 2-dimensional top down placement procedure which is a natural expansion of the top down linear placement procedure proposed in 3, and is adopted in HPS. As in the top down linear placement procedure the 2-dimensional top down procedure proceeds from the top vertex down the clustering tree until all the blocks and external modules have been processed. There are four possible arrangements for the two subtrees of each vertex in the clustering tree as shown in Fig.7, and the arrangement that gives the shortest interconnection length is chosen. The process would look like Fig.8.

Iterative Improvement by FDPR. The above 2-dimensional top down procedure cannot be free from the same drawbacks as the one with the linear top down procedure, that is, deciding the locations of each subtree without knowing the local placement in the subtree. In order to make up this weakness the iterative procedure based on FDPR technique[4] is used to improve the block placement.

Module Placement

After the final block placement by FDPR procedure is obtained. the initial local module placement within

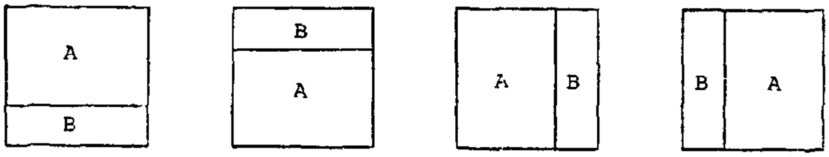

Fig. 7. Four Possible Arrangement of the Subtrees

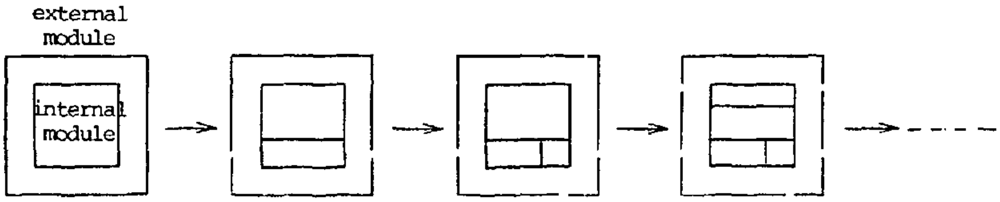

Fig. 8. 2-dimensional Top Down Placement

each block is obtained in exactly the same way as in the initial block placement. Then the final module placement is obtained again by FDPR procedure executed on all the modules without bench boudaries.

Linear Assignment Procedure for the External Modules

The linear assignment procedure or the Steinberg's algorithm is implemented for the optimum placement of the external modules. This procedure can be executed after:

i) the initial block placement,
ii) the block level FDPR,
iii) the initial module placement,
iv) the module level FDPR

Results

HPS has been successfully used for more than a year as the placement subsystem of MARS-MII, the second version of MARS-M [6], which is the automatic chip layout design system for the DSA-MOS gate arrays [9]. No manual assistance has been necessary for the actual layout design of the gate arrays except for the external pin assignment which is sometimes necessary for the compatibility purposes, etc.

As Table 1 suggests, the final results seems to be better than the manual design in terms of the maximum channel capacity required which is the most important criterion for gate array design, even though the manual design has always the possibility to be improved if the designer takes more time.

Table 2 shows the effect of blocking in terms of channel capacity required and total wire length. In case of Circuit A, blocking with the bench size 5 yields the best result in terms of both channel capacity and wire length. In Circuit B, however, no blocking is slightly better than blocking with bench size 5. This shows that the most suitable bench size varies from circuit to circuit, and it has to be found by observation of the interconnection condition, or by experiments.

Computation time of blocking is about 5 minutes for the typical circuits of 600 gates (internal modules) by MELCOM-COSMO Model 700*, which is about 30 to 10% of the total placement execution time for the same size circuits.

Table 1. Comparison between Manual and Automatic Placement

	Circuit A(578 gates)			Circuit B(563 gates)		
	MP/MR	MP/AR	AP/AR	MP/MR	MP/AR	AP/AR
max # tracks required /horizontal channel	14	14	14	16	15	12
ave # tracks required /horizontal channel	11.9	11.7	9.0	11.6	10.3	8.8
ave wire dencity of horizontal channel		6.6	4.2		4.6	3.9

MP: manual placement
AP: automatic placement
MR: manual routing
AR: automatic routing

Table 2. Effect of Blocking

circuit	bench size	# unconnected nets	hypothetical total wire length	max # tracks required /horizontal channel	ave # tracks required /horizontal channel
A	no blocking	1	55357 (105%)	(16)	(9.8)
A	5	0	52817 (100%)	14	9.0
A	13	0	54011 (102%)	14	10.2
A	26	14	56503 (107%)	(16)	(10.5)
B	no blocking	0	48609 (100%)	12	8.8
B	5	0	48733 (100%)	14	9.5
B	13	0	52386 (107%)	16	9.8
B	26	0	56458 (116%)	15	11.5

(*Comparable to IBM 370/145)

Conclusion

The outline of HPS consisting of block level placement and module level placement procedures is described. HPS has been actually used for more than a year for the automatic layout design of the DSA-MOS gate arrays with effectively no manual intervention.

Acknowledgements

The authers wish to thank H. Matsumoto for his support and encouragement to this work. They also wish to thank T. Shiraishi for the implementation of the initial placement procedures including blocking and clustering.

References

1. A. Hashimoto and J. E. Stevens, "Wire Routing by Optimizing Channel Assignment within Large Apertures", Proc. 8th Design Automation Workshop pp. 155-169, June 1971.

2. M. Hanan, P. K. Wolff and B. J. Agule, "A Study of Placement Techniques", Journal of Design Automation & Fault-Tolerant Computing, Vol. 1, No. 1, pp. 28-61, Oct. 1976.

3. D. M. Schuler and E. G. Ulrich, "Clustering and Linear Placement, Proc. 9th Design Automation Workshop pp. 50-56, June 1972.

4. M. Hanan and J. M. Kurtzberg, "Placement Techniques", Chap. 5 in Design Automation of Digital Systems: Theory and Techniques, Vol. 1 (Ed. M. A. Breuer), Prentice-Hall, N. J., pp.213-282, 1972.

5. B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs", Bell System Tech. J. pp. 291-307, Feb. 1970.

6. C. Tanaka, S. Murai et al, "CAD Oriented 920 Gate DSA-MOS Masterslice LSI", Proc. 3rd USA-Japan Computer Conference pp. 401-406, Oct. 1978.

7. D. G. Schweikert, "A 2-dimensional Placement Algorithm for the Layout of Electrical Circuits", Proç. 13th Design Automation Conference pp. 408-416, June 1976.

8. H. Yoshizawa, H. Kawanishi and K. Kani, "A Heurist "A Heuristic Procedure for Ordering MOS Arrays", Proc. 12th Design Automation Workshop pp. 384-393, June 1975.

9. I. Ohkura, O. Tomisawa et al, "A Multi-level Metalized DSA-MOSMasterslice", to be published in the J. Solid State Circuits, IEEE.