

An Interconnectivity based Efficient Partitioning Algorithm of Combinational CMOS Circuits

Milon Mahapatra
SRM University
Department of Electronics
and Communication

M.Malathi
SRM University
Department of Electronics
and Communication

B.Srinath
SRM University
Department of Electronics
and Communication

ABSTRACT

In this new technology era, circuit partitioning is a fundamental problem in very large-scale integration (VLSI) physical design automation. In this brief, we present a new interconnection oriented clustering algorithm for combinational VLSI circuit partitioning. The proposed clustering method focuses on capturing clusters in a circuit, i.e., the groups of cells that are highly interconnected in a VLSI circuit. Therefore, the proposed clustering method can reduce the size of large-scale partitioning problems without losing partitioning solution qualities. The performance of the proposed clustering algorithm is evaluated on a standard set of partitioning benchmarks—ISCAS85 benchmark suite. The experimental results show that the proposed algorithm yields results comparable to that of the rajaraman-wong optimum delay clustering approach with a faster execution time.

General Terms

Clustering, Benchmark, Algorithms.

Keywords

Clustering, Benchmark, Algorithms, Partitioning.

1. INTRODUCTION

Integrated circuits belong to the most important products of today's industry, addressing a huge market. Therefore, the competitors make great efforts to produce the best chip with the lowest cost in the shortest possible time. For achieving high quality, the tools perform more and more work on each design task. VLSI circuit partitioning is an important part of physical design automation of VLSI chips and multi-chip systems. Partitioning is used to sub-divide large VLSI circuits into smaller sub-circuits that can be designed separately and efficiently. The circuit-partitioning problem can be expressed in graph theoretic terms by modeling each circuit as a hyper graph with circuit components (cells) represented by vertices, and the nets connecting the components represented by hyper edges. A general objective of the circuit partitioning is to minimize the *net cut* or the number of nets that span the partitions. In the past two decades, partitioning problems have been studied by many researchers and various heuristic algorithms for partitioning have been developed [1]–[4]. In this paper, we present an efficient algorithm that clusters any combinational network such that the maximum delay through the network is minimized and the clustering is used to obtain a good initial partition.

Circuit layout and VLSI design procedures are too complex for any single processing technique to handle in isolation. Circuit partitioning is used to reduce the complexity of large VLSI circuits [13]. The circuit partitioning problem is formulated as follows: given a net list of m modules and n nets connecting

these modules together, partition these modules into k blocks in such a way to minimize number of nets cut between different blocks. The partitioning may be also required to maintain certain size constraints for each block, i.e., the number of modules assigned to each block may have an upper, lower limit or both limits. The partitioning problem is an NP (non-polynomial) complete problem [14]. Some heuristic and graphical techniques have been developed to obtain reasonable solutions (partitions) for the problem [13] [15]. Simulated annealing has been also used and proved to give near optimum results [16], but it is too slow to be used for large problems.

2. CLUSTERING ALGORITHM REVIEW

Clustering algorithms play a crucial role in VLSI circuit partitioning. The partitioning algorithms are based on clustering a large circuit into a sequence of smaller circuits. Clustering is an important factor in determining the performance of partitioning algorithms, where different clustering approaches can lead to different partitioning solutions. A large number of clustering algorithms have been proposed in the past decade. In [6] and [7], edge clustering, hyper edge clustering, modified hyper edge clustering, and First Choice clustering schemes are presented. The edge Clustering scheme groups two nodes that are highly connected to form a cluster. Hyper edge clustering and modified hyper edge clustering group the nodes belonging to a hyper edge to form a cluster. The First Choice clustering scheme groups the nodes such that each node in a cluster is highly connected with at least one other node in the same cluster. In [8], heavy-edge matching that clusters pairs of cells with highest connectivity is used. In [10], the clustering is done based on *edge separability*. This technique uses a graph model and exploits the global connectivity information for clustering. In [12], a fine granularity clustering (FGC) algorithm is presented that uses a modified FM (Fiduccia-Mattheyses) algorithm to refine initial clusters. In this technique, first the entire circuit is clustered using a greedy algorithm, then a modified FM algorithm is used on the whole circuit to refine initial clusters. In [11], partitioning is performed at each clustering level where only the cells in the same partition can be clustered. In this method, the clustering solution is a byproduct of the partitioning. Among various clustering algorithms, edge clustering is one of the most commonly used clustering approaches and has been successfully used in many approaches.

3. PROBLEM FORMULATION

A combinational network can be represented as an undirected acyclic graph $G = (V, E)$, where V is the set of nodes, and E is the set of directed edges. Each node in V represents a gate in the network and each edge (u, v) in E represents an interconnection between gates u and v in the network. The *fan-in* of a node is

the number of edges incident into it, and the *fan-out* is the number of edges incident out of it. A *primary input* (PI) is a node with fan-in 0, and a *primary output* (PO) is a node with fan-out 0. We represent the set of PI's by PI , and the set of PO's by PO . Each node has a *weight* and a *delay* associated with it. We denote the weight function by $w: V \rightarrow \mathbb{R}^+$ and the delay function by $\delta: V \rightarrow \mathbb{R}^+$, where \mathbb{R}^+ denotes the set of nonnegative reals. A cluster is a set of nodes $U \in V$ of the network.

Definition 1: A clustering of a network $G = (V, E)$ is a triple (H, ϕ, Σ) , where

- 1) $H = (V', E')$ is a directed acyclic graph,
- 2) ϕ is a function mapping V' to V such that

- a) for every edge $(u', v') \in E'$, $[\phi(u'), \phi(v')] \in E$,
- b) for every node $v' \in V'$ and edge $[u, \phi(v')] \in E$, there exists a unique $u' \in V'$ such that $\phi(u') = u$ and $(u', v') \in E'$, and for every PO node $v \in V$, there exists a unique $v' \in V'$ such that $\phi(v') = v$, and
- c) Σ is a partition of V' .

Let $\Gamma = [H = (V', E'), \phi, \Sigma]$ be a clustering of G . For $v \in V$, $v' \in V'$, if $\phi(v') = v$, we call v' a copy of v . The set V' consists of all the copies of the nodes in V that appears in the clustering. Moreover, $v' \in V'$ is a PI (resp., PO) of a clustering (H, ϕ, Σ) if $\phi(v')$ is a PI (resp., PO) of G . It follows from the definition of ϕ that H is logically equivalent to G .

The weight (resp., delay) of any node v' in V' is the weight (resp., delay) of $\phi(v')$. The weight of any cluster $C \in \Sigma$, denoted by $W(C)$, is the sum of the weights of the nodes in C . We now present a notion of delay of a clustering of a network using the general delay model. The delay of an edge $(u', v') \in E$ is D if u and v are in different elements of C , and zero otherwise. The delay along a path $(v_1', v_2' \dots v_k')$ in a clustering Γ is the sum of the delays of the gates $v_1', v_2' \dots v_k'$ and the delays on the interconnection edges $(v_1', v_2') \dots (v_{k-1}', v_k')$. The delay of a node v' in clustering Γ , denoted by $d_r(v')$, is the maximum delay along a path from a PI to v' . The delay of a clustering Γ is the maximum delay of a PO in Γ .

Definition 2: Given a combinational network $G = (V, E)$ with weight function $w: V \rightarrow \mathbb{R}^+$, weight capacity M , and delay function $\delta: V \rightarrow \mathbb{R}^+$:

- 1) A clustering $\Gamma = (H, \phi, \Sigma)$ of G is feasible if for every cluster $C \in \Sigma$, $W(C)$ is at most M .
- 2) The circuit clustering problem is to compute a feasible clustering Γ of G such that the delay of Γ is minimum among all feasible clustering of G .

Fig. 1 shows an instance of the circuit clustering problem. The combinational circuit is represented by the acyclic graph in Fig. 1(a) and the clustering solution is represented by the Fig. 1(b).

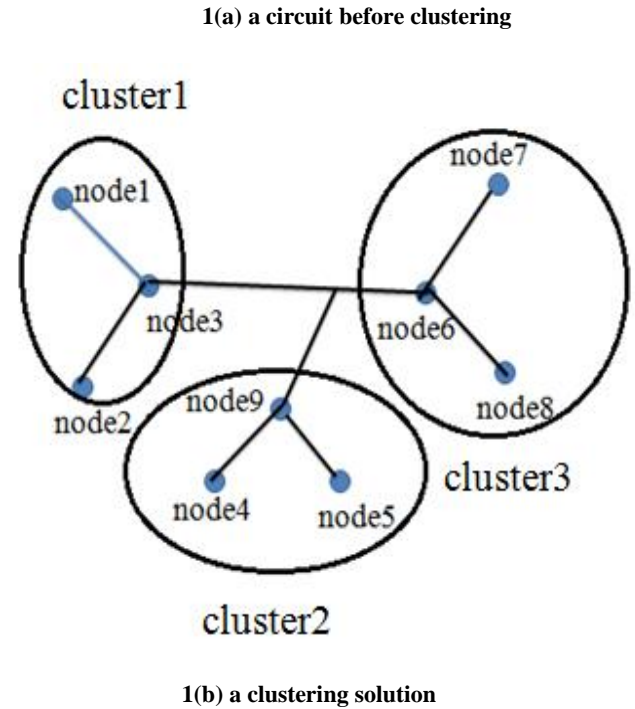
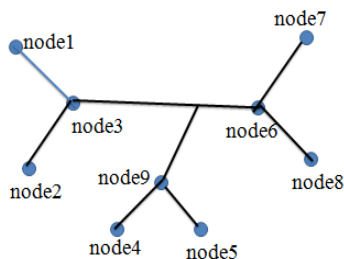


Fig 1: An instance of the circuit clustering problem

4. THE PROPOSED CLUSTERING ALGORITHM APPROACH

Clustering technique is a well known heuristic technique used mainly in image processing and pattern recognition. The main advantage of clustering algorithm is its small execution time which makes it the first candidate for real time image processing. The proposed clustering algorithm starts by assuming each module to be a cluster. In this work, the clusters are generated directly from benchmark circuit. So, this is an automatic process of cluster generation with faster execution time. The algorithm steps are as follows:

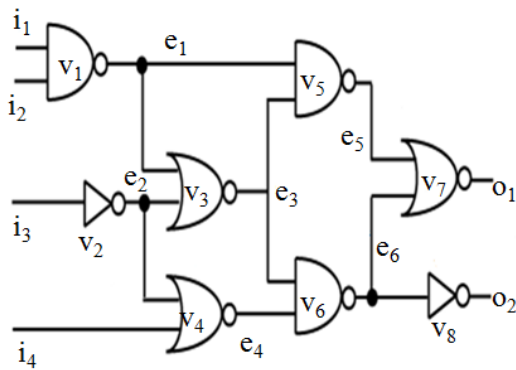
1. Visit all the gates in benchmark circuit.
2. Mark all the gates as a node in topological order.
3. Read the netlist from benchmark circuit.
4. Again visit all the nodes and Find the number of nets which are interconnected between two or more than two gates/nodes. These nets are called as matching nets.
5. List the number of matching nets.
6. Again visit the nodes and group the nodes which are interconnected through a common net. That group of nodes is called as cluster for the corresponding common nets.
7. Likewise visit all the nodes for all matching nets and list the total number of group. These groups represent the total number of cluster in benchmark.

This approach can be explained with mathematical expression. Let assume a graphical representation of an electrical circuits as shown in Fig. 2. A combinational network can be represented as an undirected acyclic graph $G = (V, E)$, Where V is the set of nodes, and E is the set of directed edges. Each node in V represents a gate in the network and each edge (u, v) in E represents an interconnection between gates u and v in the network. Also $V = (v_1, v_2, v_3, \dots, v_k)$ and $E = (e_1, e_2, e_3, \dots, e_n)$. So, there are n and k number of edges and vertices respectively.

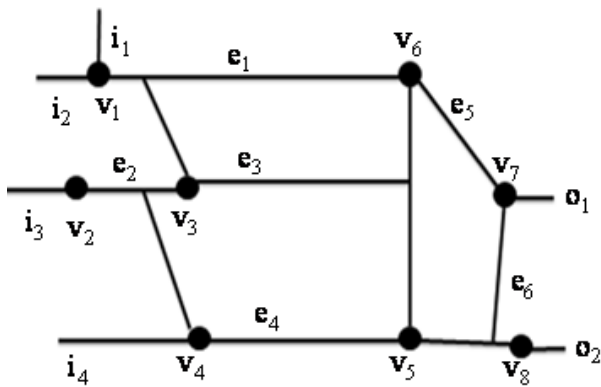
Visit v_1 : Note that v_1 is contained in e_1 . Then we have to find the neighbor of v_1 (the number of gates which are connected to e_1). Here neighbor of $(v_1) = \{v_5, v_3\}$. So the cluster for interconnect e_1 is $C_1 = \{v_1, v_5, v_3\}$. Next we have to visit vertices v_2 and find the cluster C_2 .

Visit v_2 : Note that v_2 is contained in e_2 . So, here neighbor of $(v_2) = \{v_3, v_4\}$. So, the cluster here is $C_2 = \{v_2, v_3, v_4\}$.

Likewise we have to visit all the vertices and we have to find all the corresponding clusters. Here we have to remember that the clusters are formed according to net connection only. This point is very important for this approach.



(a)



(b)

Fig 2: (a) an electrical circuit and (b) its graphical representation

Example network	Number Of Nodes	After Clustering Number Of Clusters	Execution Time In Proposed method(second)	Execution Time in Rajaraman-Wong method(second)
C432	196	50	1.057	1.5
C499	243	25	1.315	2.0
C880	443	50	3.777	4.0

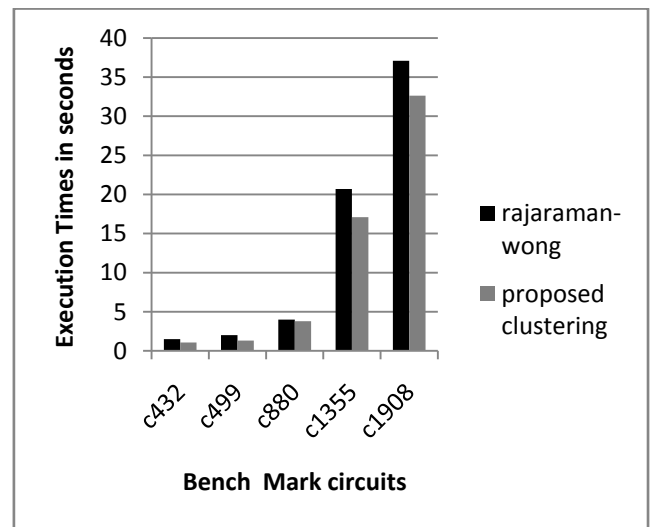
5. EXPERIMENTAL RESULTS

We have implemented the Clustering algorithm in C++ coding language and compiled with GNU GCC compiler and run on a Code Blocks integrated development environment. The algorithm was tested on some ISCAS combinational networks. Results for five ISCAS Circuits are shown in Table 1. The results are compared with rajaraman-wong optimum delay clustering algorithm. Also the comparison of Rajaraman-wong [17] and proposed clustering results has been plotted in fig. 3.

Table 1. Simulation Results

C1355	587	66	17.1	20.7
C1908	913	119	32.621	37.1

Fig 3: Experimental results



6. CONCLUSIONS

A clustering algorithm for circuit partitioning has been proposed. Comparison with other algorithm shows that the proposed algorithm obtains good solution at a faster execution time. Our method can be applied to the most general clustering problem. Moreover, the algorithm can be generalized for any clustering problem. Our algorithm generates minimum number of clusters in the solution within short period of time. Future work may involve developing an independent multilevel partitioner based on proposed clustering algorithm.

7. REFERENCES

- [1] C. J. Alpert and A. B. Kahng, "Recent directions in netlist partitioning: A survey," *VLSI J. Integr.*, vol. 19, pp. 1–81, 1995.
- [2] S. Dutt and W. Deng, "Probability-based approaches to vlsi circuit partitioning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits*, vol. 19, pp.534–549, 2000.
- [3] C. M. Fiduccia and R. M. Mattheyses, "A linear time heuristic for improving network partitions," in *Proc. ACM/IEEE DAC*, 1982, pp.175–181.
- [4] L. W. Hagen, D. J. Huang, and A. B. Kahng, "On implementation choices for iterative improvement partitioning methods," in *Proc. Eur. DAC*, 1995, pp. 144–149.

- [5] C. J. Alpert, "The ISPD98 circuit benchmark suite," in *Proc. ACM/IEEE ISPD*, 1998, pp. 80–85.
- [6] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Application in VLSI domain," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 1, pp. 69–79, Jan. 1999.
- [7] G. Karypis and V. Kumar, "Multilevel k-way hypergraph partitioning," in *Proc. ACM/IEEE DAC*, 1999, pp. 343–348.
- [8] C. J. Alpert, D. Huang, and A. B. Kahng, "Multilevel circuit partitioning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits*, vol. 17, no.8, pp. 655–667, Aug. 1998.
- [9] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Improved algorithms for hypergraph bi-partitioning," in *Proc. Asia South Pacific DAC*, 2000, pp. 661–666.
- [10] J. Cong and S. Lim, "Edge separability-based circuit clustering with application to multilevel circuit partitioning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits*, vol. 23, no. 3, pp. 346–357, 2004.
- [11] Y. Saab, "An effective multilevel algorithm for bisecting graphs and hypergraphs," *IEEE Trans. Comput.*, vol. 53, no. 6, pp. 641–652, Jun. 2004.
- [12] B. Hu and M. Marek-Sadowska, "Fine granularity clustering for large scale placement problems", in *Proc. ACM/IEEE ISPD*, 2003, pp. 67–74.
- [13] B.W. Kernighan, and S.Lin, "An Efficient Heuristic Procedure for Partitioning Graphs", *The Bell Systems Technical Journal*, 49/2 (1970), pp. 291-307.
- [14] M.R. Garey and D.S.Johnson, *Computers and Intractability*, Freeman, San Fransisco, CA. 1979.
- [15] L.A. Sanchis, "A Multiple- Way Network Partitioning", *IEEE Transactions on Computer*, vol. 38, No.1, (1989), pp. 62-81.
- [16] C. Reeves, "Modern Heuristic Techniques for Combinatorial Problems", John Wiley & Sons, Inc., New York, 1983.
- [17] Rajmohan Rajaraman and D. F. Wong, "Optimum clustering for delay minimization," *IEEE Trans. on Comput.-Aided Des. Integr. Circuits*, vol. 14 no. 12, December 1995.