Selection of output elements by minimum cost flow algorithm

Ruben, Adamyan
Yerevan State University
Yerevan, Armenia
e-mail: ruboam@yahoo.com

Stepan, Markosyan
Yerevan State University
Yerevan, Armenia

ABSTRACT

During the placement stage of VLSI (Very Large Scale of Integration) physical design phase it is needed to take into account the external connections of the placing elements. So later, it is possible to get better result in routing stage. Thus it is required to map external nets of a circuit to its elements such a way that the maximum number of nets corresponding to an element is minimal. In this article this problem is solved by reducing it to the problem of finding a minimum cost flow of a given value in a network.

1. INTRODUCTION

In general, integrated circuits are represented by the output terminals of the circuit, by its elements and nets connecting those elements and terminals. The nets connected to output terminals are called external nets. If we ignore terminals of the circuit then we can consider a net like some subset of the circuit elements.

In general case the nets of a circuit (and particularly external nets) contain two and more elements. Thus during VLSI physical design [1] it is required to choose one element from each external net (based on some criterion) which will be connected with output terminal of that net. Depending on the needs of physical design stage different criteria are used to choose those elements. For example one might want to minimize the number of those elements. Here we are considering a version of this problem where it is required to minimize the number of connections of an element having the most connections with output terminals.

2. PROBLEM FORMULATION

Let S be a circuit and $N = \{N_1, N_2, \ldots, N_m\}$ be the set of external nets of the circuit. Let's use the notation $E = \cup_{i=1}^{m} N_i$.

Let Φ be the set of all $f : N \to E$ mappings where $f(N_i) \in N_i$ for all $N_i \in N$. It is required to find a mapping $f_0 \in \Phi$ such that

$$\max_{e \in E} |f_0^{-1}(e)| = \min_{f \in \Phi} \max_{e \in E} |f^{-1}(e)|.$$

For each net $N_i; i = 1, \ldots, m$ $f(N_i)$ is the element from N_i that must be connected to the circuit terminal corresponding to that net. Thus it is required to find such a mapping that minimizes the number of connections of the element the most connected with circuit output terminals.

This problem is solved in [2] via $\log m$ consecutive application of maximum flow algorithm. This gives $O(m \sum_{i=1}^{m} |N_i| \log m)$ complexity [3, 4, 5] algorithm for this problem. Here we are reducing the problem to the problem of finding minimum cost flow having a given value. This reduction will also give $O(m \sum_{i=1}^{m} |N_i| \log m)$ complexity [3, 4, 5] algorithm for the problem.

3. SOLUTION OF THE PROBLEM

To solve the problem it will be reduced into the problem of finding minimum cost flow in a network.

Let's construct the following network (figure 1).

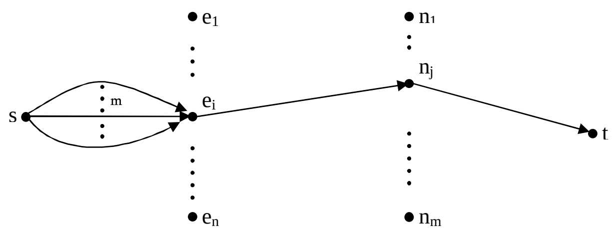

Figure 1

$\{s, t, e_1, \ldots, e_n, n_1, \ldots, n_m\}$ is the set of vertices of the network where $e_1, \ldots, e_n$ correspond to the elements of the set E and vertices $n_1, \ldots, n_m$ correspond to the nets $N_1, \ldots, N_m$. The vertex s is connected with each of $e_1, \ldots, e_n$ vertices by m parallel edges. Each of vertices $n_1, \ldots, n_m$ is connected with vertex t by one edge. There is an edge between vertices e_i and n_j if and only if $e_i \in N_j$. By the way later it will become clear that instead of $(s, e_i); i = 1, \ldots, n$ m parallel edges we can take just k_i parallel edges where k_i is the number of edges going from the vertex e_i to vertices $n_1, \ldots, n_m$. Doing so is important to get the algorithm complexity as mentioned in the problem formulation.

We are assigning 0 cost and 1 capacity to all (e_i, n_j) and (n_j, t); $i = 1, \ldots, n; j = 1, \ldots, m$ edges. For each vertex $e_i; i = 1, \ldots, n$ the capacity of all (s, e_i) m parallel edges is set to 1. The cost of those edges is defined in the following way: let's enumerate those edges by $1, \ldots, m$ numbers then assign R cost to the first edge and $R^j - R^{j-1}$ cost to the other $j = 2, \ldots, m$ edges. Here R is any number greater than m. By the way later it will become clear that instead of those costs we could take any other costs; just need to ensure the sum of costs of first k edges is more than m times greater than the sum of costs of first $k-1$ edges, for any k ($k = 2, \ldots, m$); i.e. we want the total cost of $k-1$ value flows passing by each $e_i; i = 1, \ldots, n$ vertices to be less than the cost of k value flow passing by just one e_i vertex. In case of costs assigned above we have that the sum of costs of first k edges is equal to R^k which is greater than mR^{k-1} as $R > m$.

In the constructed network we are finding minimum cost

$s - t$ F flow having value m. For finding such a flow we will convert the constructed network into an oriented graph by splitting $(s, e_i); i = 1, \ldots, n$ parallel edges via addition of new vertices, afterwards will use minimum cost flow algorithm given in [3] (or any other minimum cost flow finding algorithm). If in the obtained flow the flow passing via (e_i, n_j) edge has value 1 then we will take $f_0(N_j) = e_i$. In this way for each net N_j $(j = 1, \ldots, m)$ we will find exactly one vertex from N_j, as the capacity of (n_j, t) edges is 1 and the value of flow entering vertex t is m.

Let f_0 be the mapping constructed in the above described way and $\max_{e \in E} |f_0^{-1}(e)| = k$. So the cost of F flow will be greater or equal than R^k. If it is possible to construct a mapping $f' \in \Phi$ such that $\max_{e \in E} |f'^{-1}(e)| \leq k - 1$ then the flow constructed according to the mapping f' will have a cost smaller than mR^{k-1} which is smaller than R^k. Thus we got a flow having smaller cost than the cost of F flow which contradicts to the fact that F is minimum cost flow.

REFERENCES

[1] N.A. Sherwani, "Algorithms for VLSI Physical Design Automation", *Kluwer Academic Publishers*, Norwell, MA, USA, 1993.

[2] А.В. Петросян, С.Е. Маркосян и Ю.Г. Шукурян, Математические вопросы автоматизации проэктирования ЭВМ, Академия наук Армянкой ССР, 1977.

[3] E.A.Dinits, "The method of scaling and transportation problems", *Studies in Discrete Mathematics*, Moscow, pp. 46-57, 1973.

[4] H.N.Gabow, "Scaling algorithms for network problems", *24th Annual Symposium on Foundations of Computer Science*, New York, pp. 248-257, 1983.

[5] H.N.Gabow, "Scaling algorithms for network problems", *Journal of Computer and System Science* 31, pp.148-168, 1985.