COMPARATIVE STUDY OF CIRCUIT PARTITIONING ALGORITHMS

***Zoltan Baruch*[1], *Octavian Creţ*[2], *Kalman Pusztai*[3]**

[1] PhD, Lecturer, *Technical University of Cluj-Napoca, Romania*

[2] Assistant, *Technical University of Cluj-Napoca, Romania*

[3] PhD, Professor, *Technical University of Cluj-Napoca, Romania*

Abstract. Circuit partitioning is a fundamental problem in VLSI design in general and FPGA design in particular. In this paper we present the experiments performed in order to compare two partitioning algorithms: a modified Kernighan-Lin algorithm and a simulated annealing algorithm. Both algorithms use the same cost function, which includes the cut size of the partition and the distribution of interconnections within the two parts of the partition. We used three criteria to compare these algorithms: an estimation of the network area for the circuit, the execution time and the cost function.

1. INTRODUCTION

Circuit partitioning is an important problem in many areas of VLSI design, such as layout, placement and multiple-FPGA partitioning. At the layout level, partitioning is used to find strongly connected components that can be placed together in order to minimize the layout area and propagation delay. In the design process with FPGA circuits, partitioning is used in the placement step, which assigns each node of the circuit network to a specific logic block in the FPGA device. Partitioning also plays an important role in rapid prototyping with multiple FPGA circuits.

We consider the problem of bipartitioning a circuit into two balanced components that minimizes the number of crossing connections. This problem was shown to be NP-complete. Because of its importance, many heuristic algorithms have been proposed to solve the bipartitioning problem. They include Kernighan and Lin type iterative improvement methods, simulated annealing approaches, network flow, eigenvector decomposition, etc.

The bipartitioning algorithm proposed by Kernighan and Lin randomly starts with two subsets, and pairwise swapping is iteratively applied on all pairs of nodes [4]. Simulated annealing [2] is another method based on iterative improvement. The objective function in simulated annealing is analogous to energy in a physical system, and each move is analogous to changes in the energy of the system. The maximum-flow-minimum-cut algorithm transforms the minimum cut problem into the maximum flow problem [5]. In order to separate a pair of nodes into two subsets, the minimum number of crossing edges is equal to the maximum amount of flow from one node to the other. In eigenvector decomposition [1], connections are represented in a matrix. The eigenvectors of the matrix define the locations of all components and thus derive partitioning results.

In this paper, we present an evaluation of a modified Kernighan-Lin partitioning algorithm and the simulated annealing algorithm for a set of benchmark circuits. The goal is to obtain a better view of the values and limitations of each algorithm. In Section 2, we give a concise description of the compared algorithms. Section 3 presents the results of our experiments. Section 4 contains the conclusion.

2. DESCRIPTION OF THE ALGORITHMS

2.1. The Modified Kernighan-Lin Algorithm

The Kernighan-Lin (KL) algorithm [4] starts from a random initial partition (A, B), and improves the current partition by interchanging subsets X an Y, where $X \subset A$, $Y \subset B$, and $|X| = |Y|$. Each step of the algorithm consists of interchanging two nodes, one from each side of the partition. The complexity of the algorithm is $O(n^2 \log n)$, where n is the number of nodes.

Subsequently, many improvements have been made to this method. Fiduccia and Mattheyses improved the algorithm by reducing time complexity to $O(p)$ with respect to the number of pins p, and Krishnamurthy [3] further added in lookahead ability. Schweikert and Kernighan proposed the use of a net cut model so that the algorithm can handle multipin net cases.

We implemented a modified KL bipartitioning algorithm, which reduces the cut size of the partition, and in the same time evenly distributes the connections among them. In the original algorithm, the only metric in the cost function is the cut size. However, the cut size alone is not a good metric for circuits with limited routing resources, such as FPGA circuits. In order to use this algorithm in the design process for FPGA circuits, we take into account not only the cut size, but also the distribution of interconnections within the two parts of the partition.

2.2. The Simulated Annealing Algorithm

The simulated annealing (SA) algorithm [2] is a widely used iterative technique for solving general optimization problems. It is an adaptive heuristic and belongs to the class of non-deterministic algorithms. The algorithm is based on the analogy to the annealing process, which consists of carefully cooling molten metals in order to obtain a good crystal structure. The attainment of good crystal structure is analogous to the attainment of global optimum.

The main advantage of the SA algorithm consists of its ability to avoid local optima by allowing an occasional uphill move. This is done under the influence of a random number generator and a control parameter called the *temperature*. The Metropolis Monte Carlo method is used to decide whether a move is accepted. Whenever the algorithm encounters an uphill move ($gain < 0$), it accepts this move with a probability $e^{-gain/T}$, where T is the temperature.

A typical implementation of the SA algorithm uses two additional parameters: a cooling ratio α, and a temperature length M. Temperature is initialized to a value T_0, and is slowly reduced in a geometric progression using the cooling ratio. The temperature length M indicates the number of solutions examined at a given temperature. The amount of time spent in annealing at a temperature is gradually increased as temperature is lowered. This is done using a parameter $\beta > 1$.

In our implementation of the SA algorithm, the cost function used is similar to that used in the modified KL algorithm. It includes the cut size and the distribution of interconnections within the two parts of the partition.

3. PERFORMANCE ANALYSIS

We realized a comparative study of the modified KL algorithm and the SA algorithm. The three criteria used in this analysis are the following:

- An estimation of the network area for the circuit;
- The execution time;
- The cost function.

The algorithms were tested on a set of benchmark circuits.

The first criterion is the *network area*. This is an estimation of the implementation area obtained after the placement of the circuit. This area is estimated by calculating the Manhattan distance for each pair of pins and cumulating it for all connections. The experimental results are shown in Table 1.

Table 1. Estimation of the network area for the modified KL algorithm and the SA algorithm.

Circuit	Number of nodes	Network area	
		KL	SA
Actlow	18	66	74
Regfb	21	67	67
Moore	25	102	106
Mealy	37	180	189
Sequence	49	248	283
Dmux1t8	60	373	433
Cntbuf	64	389	437
Decade	71	393	510
Binbcd	101	866	979

The graphical representation of the results obtained is presented in Figure 1.

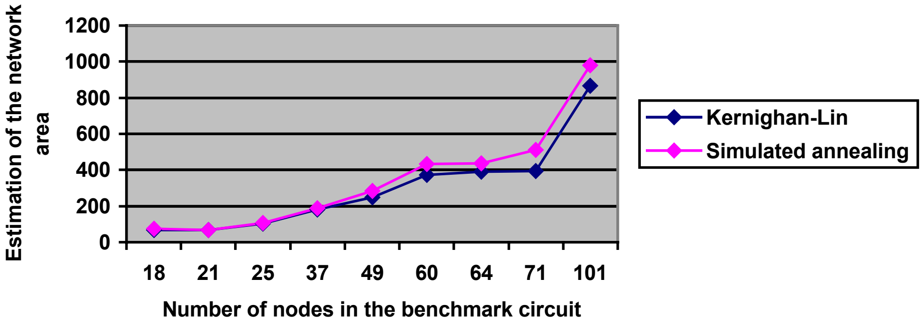

Figure 1. The network area for the modified KL and SA algorithms.

For a small number of nodes, the difference between results is almost negligible, but when the number of nodes increases, the difference becomes significant. The results suggest that the solutions obtained by the modified KL algorithm are better than those obtained by the SA algorithm, for the set of parameters used.

The second criterion is the *execution time*. For a small number of nodes, there are no significant differences between the results of the two algorithms, but for a higher number of nodes, the execution time grows for the SA algorithm. For this algorithm, the tests were performed using the following parameters: $T = 10$, $\alpha = 1.9$, $M = 20$, $\beta = 1.5$. The results are shown in Figure 2.

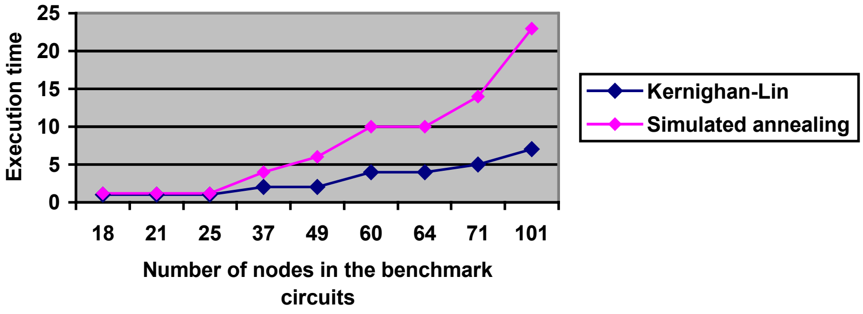

Figure 2. The execution time for the modified KL and SA algorithms.

The third criterion is the *cost function*. In Table 2 we show the components of the cost function used by the partitioning algorithms.

Table 2. Components of the cost function for the modified KL and SA algorithms.

		Kernighan-Lin				**Simulated annealing**			
		Initial partition		**Final partition**		**Initial partition**		**Final partition**	
Circuit	**Nodes**	T_i	E_i	T_f	E_f	T_i	E_i	T_f	E_f
Actlow	18	14	4	4	0	14	4	6	0
Moore	21	19	2	7	0	19	2	9	0
Regfb	25	15	1	4	0	15	1	4	0
Mealy	37	34	0	12	0	34	0	14	0
Sequence	49	42	5	11	0	42	5	23	0
Dmux1t8	60	52	3	15	0	52	3	26	1
Cntbuf	64	54	1	17	0	54	1	23	2
Decade	71	72	5	19	0	72	5	37	0
Binbcd	101	101	10	31	0	101	10	59	0

In Table 2, T_i and T_f represents the initial cut size and the final cut size, respectively. E_i and E_f represents the initial and the final balance number, indicating the difference between the number of connections in the two parts of the partition. The cost function F_c is computed according to the following formula:

$$F_c = I_t \cdot T_f + I_e \cdot E_f$$

where I_t indicates the relative importance of reducing the cut size, and I_e indicates the relative importance of balancing the number of connections. We used the following values for I_t and I_e: $I_t = 0.5$, $I_e = 0.5$. This means that both criteria have the same importance. Notice that $I_t + I_e = 1$.

In Figure 3 we present the variations of the cost function for the two algorithms. The modified KL algorithm produces better results than the SA algorithm. For the SA algorithm, the tests were performed using the same parameters presented before. By changing the values of the parameters, the results obtained for this algorithm can be improved, but the execution time grows significantly.

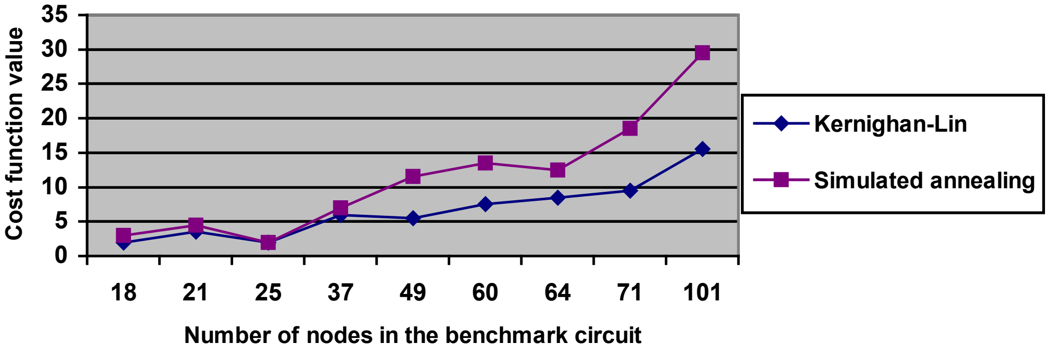

Figure 3. Representation of the cost function for the modified KL and SA algorithms.

4. CONCLUSIONS

In this paper we presented the experiments performed in order to compare two partitioning algorithms: a modified Kernighan-Lin algorithm and a simulated annealing algorithm. Both algorithms use the same cost function, which includes the cut size of the partition and the distribution of interconnections within the two parts of the partition. The experiments performed are based on three criteria: an estimation of the network area for the circuit, the execution time and the cost function.

The results show that the modified KL algorithm produces the best results when we consider the execution time and the cost function. From the point of view of the estimated network area, the differences are not significant, so both algorithms can be used for the placement of FPGA circuits.

REFERENCES

[1] Hadley, S.W., and Mark, B.L.: "An Efficient Eigenvector Approach for Finding Netlist Partitions", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 11, No. 7, July 1992, pp. 885-892.

[2] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P.: "Optimization by Simulated Annealing", Science, No. 220, May 1983, pp. 671-680.

[3] Krishnamurthy, B.: "An Improved Min-Cut Algorithm for Partitioning VLSI Networks", IEEE Transactions on Computers, Vol. 33, No. 5, May 1984, pp. 438-446.

[4] Sait, S.M., and Youssef, H.: "VLSI Physical Design Automation", McGraw-Hill Book Company, 1995.

[5] Yang, H.H., and Wong, D.F.: "Efficient Network Flow Based Min-Cut Balanced Partitioning", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 15, No. 12, December 1996, pp. 1533-1540.