

Partitioning of VLSI Circuits on Subcircuits with Minimal Number of Connections Using Evolutionary Algorithm*

Adam Słowik and Michał Białko

Department of Electronics and Computer Science, Technical University of Koszalin,
ul. Śniadeckich 2, 75-453 Koszalin, Poland
aslowik@ie.tu.koszalin.pl

Abstract. In this paper, we present an evolutionary algorithm application to partitioning VLSI circuits on subcircuits with minimal number of connections between them. The algorithm is characterized by a multi-layer chromosome structure. Due to this structure, the partition of circuits is possible without applying a repair mechanism in the algorithm. The test circuits chosen from literature and created randomly are partitioned using proposed method. Results obtained by this method are compared with results obtained using a traditional Kernighan-Lin algorithm.

1 Introduction

With the quick development of VLSI integrated circuits, it is possible to place larger and larger numbers of electronic elements in a single integrated circuit. The problem of the circuit partitioning on subcircuits is still important, especially in early phases of layout design of integrated circuits [1]. The VLSI integrated circuits are partitioned into subcircuits in order to fulfill constraints connected with: a) the number of external connections of the integrated circuit, b) signal delays in the circuit, c) electro-thermal interactions between elements, d) the size of the area for the circuits, and e) the testability of integrated circuits. The first constraint is essential, because the large number of external connections dramatically increases the overall expense of producing the circuit [2]. The reduction of the number of connections between subcircuits is a primary objective of partitioning algorithms. It is important from a few reasons. First, electric signal are delayed during transmission through external connections; second, external connections occupy large place on circuit printed boards; and third, external connections decrease reliability of the system [2]. The problem of the circuit partitioning into k subcircuits with minimal number of connections between them is NP-complete [3] thus, the typical approach to partitioning problems is an application of heuristic algorithms which allow to find suboptimal solutions. The

* This work was supported by the Polish Ministry of Scientific Research and Information Technology (MNiI) under Grant No. 3 T11B 025 29.

most popular heuristics is Kernighan-Lin algorithm [4] and its modification proposed by Fiduccia-Mattheyses [5]. Recently, evolutionary algorithms are applied to partitioning problems of electronic circuits. This comes out of the fact that evolutionary algorithms are the very effective tools for solving difficult combinatorial problems. A few partitioning algorithms based on evolutionary algorithms are described in literature, including Raman [6], Vemuri [7], and Kozieł [1]. In the Kozieł paper [1], he has presented an algorithm for multiobjective partitioning of VLSI circuits based on evolutionary algorithm. This algorithm was based on single-layer chromosome representation of each individual (potential solution). However, this approach caused that in the case when the circuit was to be partitioned into k equal-number subcircuits, then a typical crossover operator created solutions that did not satisfy the constraints. In this article, we introduce the evolutionary algorithm with multi-layer chromosomes (similar to that applied in [10]) to partitioning of electronic circuits. Here, the circuit partitioning into k equal-number subcircuits is possible without applying additional repair algorithms, because each created solution is acceptable. Suitable genetic operators were proposed to this data structure. This algorithm (described in section 3) is named MLCEA-PP (*Multi Layer Chromosome Evolutionary Algorithm for Partitioning Problem*). Results obtained by this algorithm were compared with the results obtained using Kernighan-Lin algorithm.

2 Problem of VLSI Circuits Partitioning

Each electronic circuit can be modelled using the $G(V, E)$ graph, where V represents a set of graph nodes, that is the set of electronic elements, and E corresponds to a set of graph edges, that is the set of connections between electronic elements in the circuit. In Fig. 1 an example of a digital circuit composed of 8 gates, and 9 connections between them, is presented.

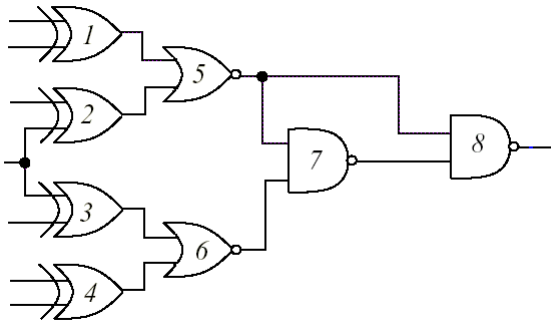


Fig. 1. Example of the digital circuit

In Fig. 2 the graph corresponding to the circuit shown in Fig. 1 is presented. This graph has 8 nodes, and 9 edges between them.

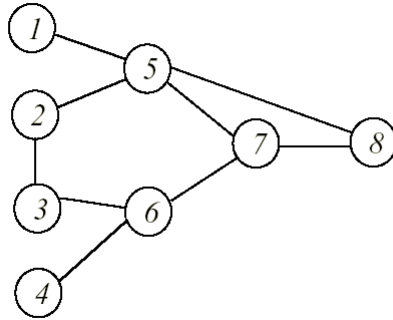


Fig. 2. Graph corresponding to the digital circuit of Fig. 1

The described problem corresponds to such partition of the set V of circuit nodes into k subsets (subcircuits) to optimize given objective function including possible constraints. It usually depends on minimization of the connection number between particular subcircuits (subsets). In Fig. 3 two examples of the circuit partitioning into two equal-number subcircuits ($k=2$) are presented.

Additionally, the graphs corresponding to the circuits of Fig. 3 are shown in Fig. 4.

The circuit (graph) partitioning presented in Fig. 3a (Fig. 4a) possesses four connections between subcircuits (subgraphs), however the partitioning shown in Fig. 3b (Fig. 4b) has only two connections between subcircuits (subgraphs). So, we can say that partitioning presented in Fig. 3b (Fig. 4b) is better. Generally an objective function in circuit partitioning problem is the sum of products of external graph edge, and weight value assigned to it. In the case when all weight values of the graph edges are equal to one, then the objective function is the number of the connections between particular subcircuits (subgraphs).

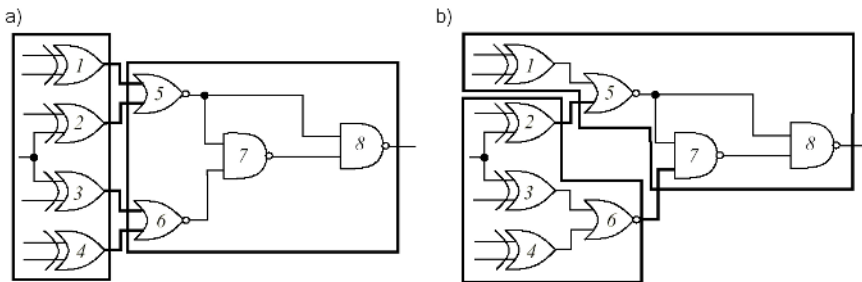


Fig. 3. Two examples of circuit partitioning into two equal-number subcircuits

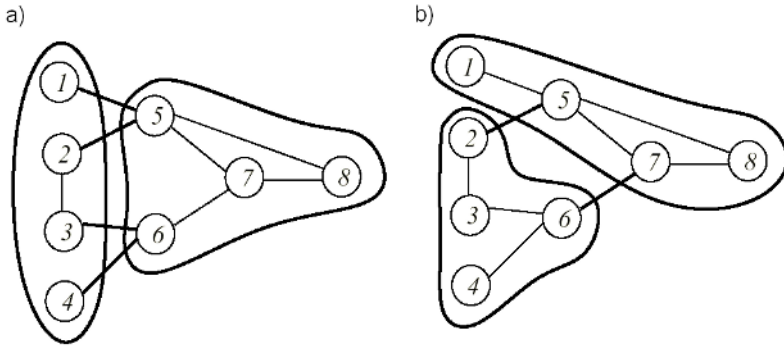


Fig. 4. Graphs partitioning corresponding to the circuits partitioning into two equal-number subcircuits from Fig. 3

3 MLCEA-PP Algorithm

3.1 Representation of Individuals

Each individual is represented by a multiple-layer chromosome which structure is shown in Fig. 5.

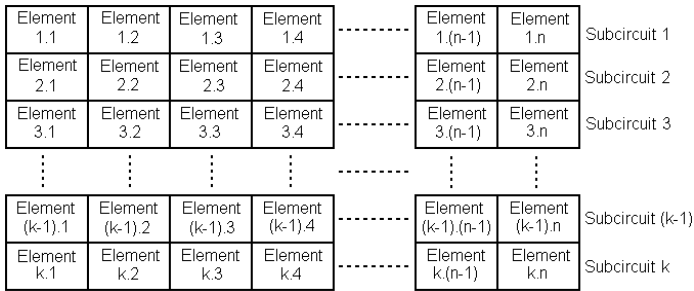


Fig. 5. Structure of the chromosome

The particular chromosome consists of k layers where each layer represents a single subcircuit (subset). With this representation, the circuit partitioning into k equal-number subcircuits each including n elements is possible without applying additional repair algorithms. Each solution is acceptable solution, because the constraints is "build-in" the proposed data structure. It is necessary to point out that the introduced structure is not limited only and exclusively to the circuit partitioning into k equal-number subcircuits. Introducing so called dummy nodes, that are not connected with any other node, enables a graph partitioning into k subgraphs with different number of elements. In Fig. 6, an example circuit partitioned into three subcircuits is presented.

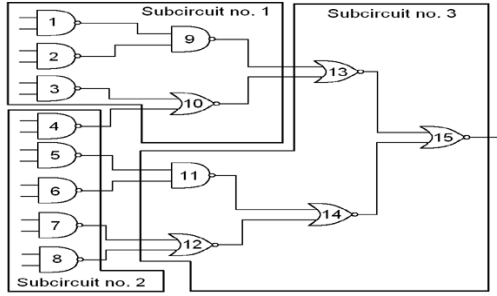


Fig. 6. Example circuit partitioned into three subcircuits

In Fig. 7 the graph corresponding to the circuit form Fig. 6 is shown.

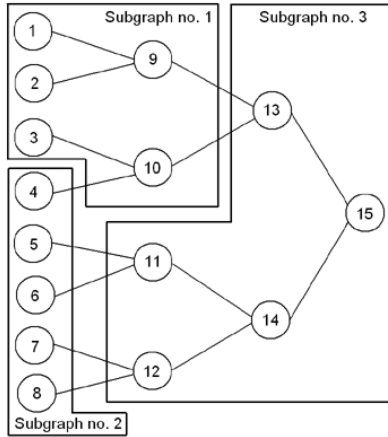


Fig. 7. Graph corresponding to the circuit from Fig. 6

In Fig. 8 the multi-layer chromosome representing the partitioned circuit (graph) from the Fig. 6 (Fig. 7) is presented.

1	2	3	9	10	Subcircuit no. 1 (Subgraph no. 1)
4	5	6	7	8	Subcircuit no. 2 (Subgraph no. 2)
11	12	13	14	15	Subcircuit no. 3 (Subgraph no. 3)

Fig. 8. Multi-layer chromosome representing the partitioned circuit

3.2 Used Genetic Operators

In the described algorithm, we use the well known PMX [11] crossover; however, we have modified it to make it suitable for operation on the multi-layer chromosome. This crossover operator has one important advantage: the child individuals that are obtained using it always represent acceptable solutions and additional repair algorithms are not required. In Fig. 9 the PMX [11] crossover procedure is presented.

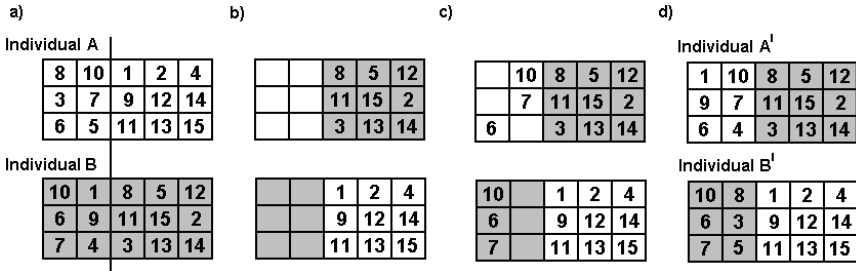


Fig. 9. PMX crossover operation

The crossover consists of three steps. In the first step, after choosing parent chromosomes and determining the point of the crossover for them (in the Fig. 9a crossing is after 2 gene) the code sequences after the cross-point are exchanged. As a result of this operation a pair of individuals presented in Fig. 9b is obtained. In the second step, we insert "not-colliding" genes from the parental individual into the blank places, (not-colliding genes, that is not existing still in the newly created child individuals). In the case of the individual A such genes are: "10", "7", and "6"; while for individual B the genes "10", "6", and "7" are inserted. After this operation the second step of crossing is finished, what is shown in Fig. 9c. In the third step, remaining vacancies are filled in the chromosomes according to the following rule. In the first gene of the first layer of the created individual A' the gene "8" would be inserted from parental individual A. However, it exists already in the newly created individual; therefore we exchange the genes according to the following rule: we are looking which gene from the created individual B' corresponds to the gene "8" from the created individual A'. The responding gene is "1" which does not exist in the individual A', therefore it is inserted to it. Following this way the genes: "9", and "5" are inserted to the individual A', and genes: "8", "3", "5" to individual B'. This completes the crossover, and as the effect of this operation is the pair of child individuals A', and B' presented in Fig. 9d. Besides the crossover, two mutation operators are introduced in the algorithm. After selection of a given gene for the mutation, the kind of mutation is determined randomly with equal probability. The first mutation operator chooses randomly two genes from the parental individual which are exchanged during the mutation. In the relation to the electronic circuit (graph) this mutation causes exchange of one element between two subcircuits (subgraphs). This mutation operation is shown in Fig. 10.

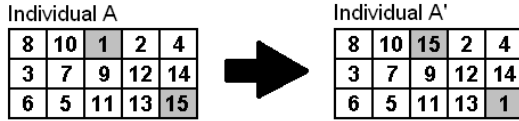


Fig. 10. First mutation operation

The second mutation operation causes a circulation of the column (chosen randomly). In reference to the electronic circuit (graph) it causes the exchange of all subcircuits (nodes in subgraphs). The second mutation operation is shown in Fig. 11.

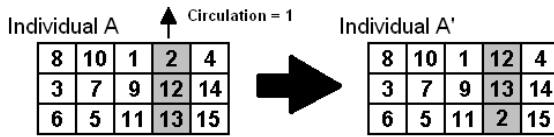


Fig. 11. Second mutation operation

3.3 Objective Function

The following objective function is used in the algorithm:

$$f_c = \frac{NC}{NOC} \tag{1}$$

where: *NC*-number of all connections in the graph (the constant parameter for a given graph), *NOC*-number of all connections between created subgraphs (external connections). During operation of the algorithm the objective function is maximized. It is necessary to notice, that during the maximization of the objective function, the minimization of the number of external connections occurs (i.e. connections between created subgraphs, what is the main objective of this problem).

3.4 Description of the Algorithm

The following evolutionary algorithm was applied to circuit partitioning into *k* subcircuits with the minimal number of connections between them. At the start the initial population was created. Then, we check whether the algorithm converged (lack of change of the best solution). If the algorithm converged, the result represented by the best individual in the population is printed out and algorithm is stopped. If the algorithm did not converge, the following operations are used: crossover, mutation, calculation of value of the objective function for particular individuals, and the fan selection [8]. Then the algorithm convergence is checked again and the whole process is repeated.

4 Description of Experiments

During performed experiments the parameters of evolutionary algorithm were: population size=100, crossover probability=0.5, mutation probability=0.05, parameter of fan selection $a=0.7$. Results obtained by MLCEA-PP algorithm were compared with results obtained using the Kernighan-Lin (K-L) algorithm (Tab. 1). The circuits were partitioned into k equal-number subcircuits. A few test circuits were chosen to the experiments from [9] (they are marked with the symbol "*" in the Tab. 1); also randomly created graphs with different number of nodes and connections between were tested. It is assumed that each connection is equally important (i.e., in the graph all edges weights have value "1"). The experiments were repeated 10 times, and results obtained as an average values from 10 starts are shown in Tab. 1. The algorithm has converged and stopped after 100 generations for 10, 12, 24, 32 nodes graphs, after 400 generations for 48 nodes graphs, and after 1000 generations for 96 nodes graphs. In Tab. 1 the symbols used are as follows: NN -number of nodes, NC -number of connections (edges), k -the number of subcircuits (subsets), NOC -number of connections between subcircuits (external connections).

Table 1. Results of graph partitioning for algorithms: MLCEA-PP (M-PP), and Kernighan-Lin (K-L)

			M-PP	K-L	—				M-PP	K-L	—				M-PP	K-L
NN	NC	k	NOC	NOC	—	NN	NC	k	NOC	NOC	—	NN	NC	k	NOC	NOC
10*	21*	2	5	6.2	—	24	96	3	45.5	48.8	—	48	384	4	236	245
10*	21*	2	7	7	—	24	96	4	53.8	56.5	—	48	384	6	272.9	281.6
32*	54*	2	7.4	7.4	—	24	96	6	64.8	68.2	—	48	384	8	292.7	300.7
32*	54*	4	12.9	18.4	—	24	96	8	73	74.9	—	48	768	2	339.8	343
12	24	2	8.11	8.9	—	48	96	2	19.7	22.7	—	48	768	3	465	472.4
12	24	3	11	12.1	—	48	96	3	30.8	34.1	—	48	768	4	533	539.8
12	24	4	13.2	14.2	—	48	96	4	37.8	42.3	—	48	768	6	606.5	613.7
12	48	2	22	22	—	48	96	6	45.8	51.5	—	48	768	8	649	652.4
12	48	3	31	31.1	—	48	96	8	51.2	56.6	—	96	192	2	35.5	36
12	48	4	36	36.1	—	48	192	2	55.9	58.4	—	96	192	3	58.5	62.7
24	48	2	12.6	13.4	—	48	192	3	83.8	89.4	—	96	192	4	73.5	78.5
24	48	3	18.5	19.9	—	48	192	4	97.7	105.6	—	96	192	6	95.1	93.3
24	48	4	21.6	24.6	—	48	192	6	116.1	122.4	—	96	768	2	270.2	271.1
24	48	6	27.6	29.3	—	48	192	8	128	134	—	96	768	3	390.1	397.7
24	48	8	31.5	32.6	—	48	384	2	146.2	149.8	—	96	768	4	459.5	459.4
24	96	2	30.3	32.7	—	48	384	3	202.5	210.6	—	96	768	6	506	530.7

5 Conclusions

Results obtained for the algorithm MLCEA-PP are comparable or better than results obtained using the K-L algorithm. It is necessary to add, that in relation

to the algorithm of paper [1], in the presented algorithm MLCEA-PP any repair mechanisms were not used. Due to application of multi-layer chromosome, the crossover operator does not create unacceptable solutions or infringe the constraints.

References

1. Kozieł S., "Evolutionary algorithms and their applications to the optimisation and modelling of the analogue electronic circuits", Ph. D. Thesis, Technical University of Gdańsk, Department of Electronics, Telecommunications, and Computer Science, 1999,
2. Sait S. M., Youssef H., "VLSI physical design automation. Theory and practise", IEEE Press, New York, 1995,
3. Chen Y. P., Wang T. Ch., Wong D. F., "A Graph partitioning problem for multiple-chip design", Proceedings ISCAS 1993, pp. 1778-1781, 1993,
4. Kernighan B. W., Lin S., "An efficient heuristic procedure for partitioning graphs", Bell System Technical Journal, 49(2), pp. 291-307, 1970,
5. Fiduccia C. M., Mattheyses R. M., "A linear time heuristic for improving network partitions", Proceedings of the ACM/IEEE Design Automation Conference, pp. 175-181,
6. Raman S., Patnaik L. M., "Performance-Driven MCM Partitioning Through an Adaptive Genetic Algorithm", IEEE Transaction on VLSI Systems, Vol. 4, No. 4, December 1996,
7. Vemuri R., "Genetic algorithms for MCM partitioning", Electronic Letters, Vol. 30, No. 16, pp. 1270-1272, 1994,
8. Słowik A., Białko M., "Modified Version of Roulette Selection for Evolution Algorithm - The Fan Selection", Proceedings of 7th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2004, LNAI Vol. 3070/2004, pp. 474-479, Springer-Verlag,
9. Rutkowski J., Zieliński L., "Using evolutionary techniques for chosen optimization problems related to analog circuits design", In Proceedings of ECCTD 2003 Conference,
10. Słowik A., Białko M., "Design and Optimization of Combinational Digital Circuits Using Modified Evolutionary Algorithm", Proceedings of 7th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2004, LNAI Vol. 3070/2004, pp. 468-473, Springer-Verlag,
11. Goldberg D. E., Lingle R., "Alleles, Loci, and the TSP", Proceedings of the First International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, Hillsdale, NJ, 1985, pp. 154-159