Engineering Applications of Artificial Intelligence

MACROMODELING FOR VLSI PHYSICAL DESIGN AUTOMATION PROBLEMS

Roman Bazylevych, Marek Pałasiński, Lubov Bazylevych

Abstract: *The paper summarizes the authors methodology for solving the intractable combinatorial problems in physical design of electronic devices: VLSI, SOC, PCB and other. The Optimal Circuit Reduction (OCR) method has proved to be an efficient and effective tool to identify the hierarchical clusters' circuit structure. The authors review the applicability of this method for solving of some problems, including hierarchical clustering, partitioning, packaging, and placement. Developed approach based on multilevel decomposition with the recursive use of global and local optimization algorithms at it every level for unique, not very large size subproblems. At every step we receive some initial solutions which are improved by optimization algorithms. Experiments confirm the efficiency of developed approaches. For some well-known test cases the optimal results were achieved for the first time, while for many other cases improved results were obtained.*

Keywords: *VLSI, SOC and PCB physical design, hierarchical clustering, partitioning, packaging, placement*

ACM Classification Keywords: *B.7.2 Design Aids*

Introduction

Many of the most difficult in Design Automation are intractable combinatorial problems. They appear in physical design – partitioning, packaging, placement, routing as well as testing and other areas. Optimization are especially important for VLSI, SoC and PCB design. Rapid growth of electronic circuit complexity requires a further search for new robust, efficient and effective approaches to solve them with high quality. From mathematic point of view they belong to the very large-scale intractable combinatorial NP-class problems – nowadays chips have a few billions of transistors.

Many of these problems have identical input data. The ideas to solve the large-scale problems are to transfer the full mathematical model to the aggregate mathematical notation that could significantly decrease the number of arguments and to operate by the not very large number of hierarchically built macromodels instead of original elements, the number of which is extraordinarily high. This enables us, to decrease the size of the problem in every step of decision making, to reduce the calculation consumption, to improve the quality of solution, as well as to easier trap into the zone of the global optimum. Basic approaches and algorithms were developed for

- hierarchical circuit clustering:
  - free clustering,
  - partially enforced clustering,
  - enforced clustering;
- partitioning and decomposition :

- serial, parallel-serial and dichotomy partitioning,
- initial partitioning,
- partitioning optimization;

- packaging:
  - serial and parallel-serial packaging,
  - initial packaging,
  - packaging optimization;
- placement:
  - hierarchical initial placement by multilevel macromodels,
  - multilevel placement optimization by scanning area method with macromodels.

For these problems also were developed special algorithms for escaping from the local extreme.

The proposed algorithms have such properties:

- can be efficient for choosing the appropriate number of partitions to divide the circuit;
- arbitrary division ratio can be chosen for partitioning;
- many same procedures can be used for initial solution and their optimization;
- close to linear computational complexity;
- provide good quality of solutions;
- are appropriate for large and very large-scale problems.

Most likely, the first proposal to use the free hierarchical clustering for partitioning was in [Bazylevych, 1975]. It was further developed in [Bazylevych, 1981] and used for packaging and placement [Bazylevych, 2000, 2002, 2007] with good results. More lately hierarchical clustering, especially enforced, was used for hyper graph partitioning [Garbers, 1990], [Cong, 1993], [Dutt, 1996], [Karypis, 1997], [Saab, 2000] and for others problems.

For all test cases investigated, the results are not worse, and in many cases they are better comparatively with obtained by other known methods. For some cases, the optimal results were received for the first time.

Main stages for solving the problems by hierarchical clustering

For solving the large-scale intractable combinatorial problems at the first step we must perform aggregation. We divide large problems into the set of small ones that are simulated by macromodels. Every macromodel include the fixed number of initial circuit elements. The number of macromodels in aggregated circuit is also very important. It is possible to create multilevel model in such way that the number of macromodels and numbers of their elements at very level of decomposition must be not very large to receive good quality solution. We build multilevel system of hierarchically built macromodels. In such system every subproblem could be solved with the high quality for not large CPU time. The main decisions that we must make in such approach are:

- how to chose the number of elements of basic subproblems that we can solve with high quality in a reasonable running time,
- what must be the number of level in macromodeling,
- what method is desirable to use for solving the basic subproblems,
- how merge the partial solutions of subproblems into one solution of whole problem,
- do we need to use additional optimization (refinement) algorithms or not,
- how to escape from the local extreme at every level of macromodeling ?

One of the first problems that appear here is to create the hierarchical macromodel of initial system. Thereto we must receive multilevel aggregation of circuit. One way is to reveal hierarchical built clusters. For this reason it is

possible in electrical system to use the Optimal Circuit Reduction method [Bazylevych, 1975, 1981]. By this method the problem solving is divided into the following steps (Figure 1):

- the bottom-up free hierarchical circuit clustering;
- the mathematical description of clusters by macromodels;
- the top-down multilevel solving with receiving global initial solutions and theirs local optimizations with macromodels at every level of decomposition.

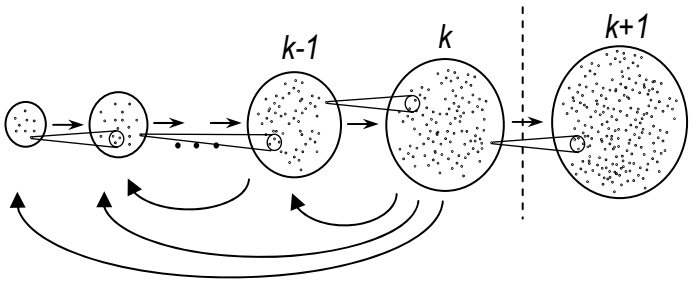

Fig. 1. Bottom-up (left arrows) hierarchical clustering and top-down (right arrows) problem solving

The main feathers of developed approach are:

- the problem size as a whole (the number of variables) increases step by step during the solution process from substantially reduced, initially (level 1) to real (level k);
- the number of tasks which are to be solved increases on each recursive level. However, all of them would be not large, and are properly hierarchically inserted one into the other, and thus can be solved by the same basic procedure with high quality.

The idea was to operate not by original elements, the number of which is extraordinarily high, but by the hierarchically built clusters of arbitrary sizes (not large) that could be mathematically described by macromodels. The (*k+1*)-level (Figure 1) shows that simulating the problem by 0-1 models (binary programming) will significantly increase the number of variables. This case can not simplify the problem solving.

This enables us:

- to essentially decrease the size of the problem, facilitating a solution and reducing the calculation consumption, the large size problem is reduced to recursive solving of small unique tasks;
- to improve the quality of the solution by more easier trapping into the zone of the global optimum. The number of local extrema is significantly smaller.

The Optimal Circuit Reduction Method

The Optimal Circuit Reduction (OCR) method builds the Optimal Reduction Tree T^R (Figure 2). It is a rooted (generally *n*-ary) tree which leaves (level *1*) correspond to the set $P = \{p_1, \dots, p_n\}$ of circuit elements and a root (level *H*) - to all aggregated circuit.

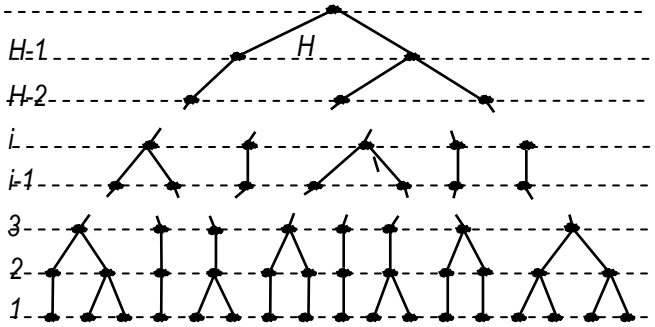

Fig. 2. The Optimal Reduction Tree

The main steps of T^R generation are:

- consider the set C_i of all clusters in the level *i*. At the first level we consider a set of all initial elements $P = \{p_1, \ldots, p_n\}$;
- form a set for all pairs of adjacent clusters for every cluster of the set C_i;
- calculate the merging criterion values for all pairs of adjacent clusters;
- create the ordered list $L(\eta)$ of pairs of adjacent clusters by the chosen merging criterion;
- form the new set C_{i+1} of clusters of the $(i + 1)$-th level. There are several possibilities. In the best case - free clustering - we merge only the maximum number of independent pairs of adjacent clusters with the best value of the merging criterion. It could cause a large tree's height and consequently takes a lot of CPU time. The one possible way to reduce the running time is to take all independent pairs with ε given decreasing (increasing) of the best criterion value. The second way is to merge the first λ of all possible independent pairs, where λ ($0 < \lambda \leq 1$) - is a reduction parameter. It is partially enforced clustering. In the last case when $\lambda = 1$ we merge all clusters. It is enforced clustering. Here a height of the ORT is a minimal and therefore it takes the minimal running time but results could be worse. This case corresponds to the enforced circuit reduction that might not generate good natural clusters, because at every level we must merge together some clusters that do not have good criterion's value;
- form the new $(i + 1)$-th level of the tree T^R by including a set of the new clusters, defined by merging and the rest clusters from the previous level that are not merged.

We must draw attention that we do not have to build binary Reduction Tree obviously. If, for example, one element creates two or more pairs with the same criterion's value, it is possible to join three or more elements together at one step. It reduces the height of tree and, of course, the CPU time. It is not easy to choose the criteria for clusters merging to receive the best clusters. There are many possible merging criteria. The main of them are:

- maximization the full number of the internal clusters' nets;
- minimization the full number of the external clusters' nets;
- maximization the full number of the subtraction of the of internal and external nets.

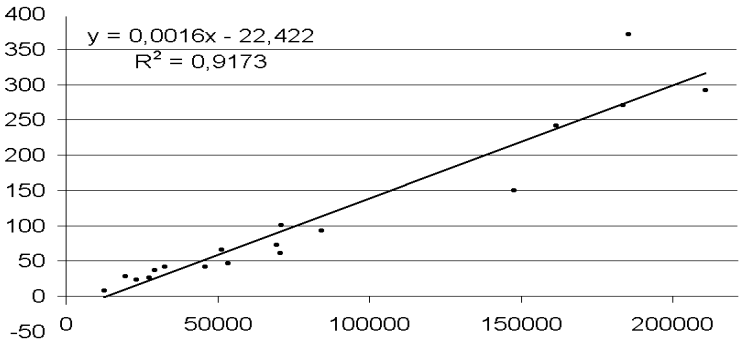

Fig.3. Circuit size vs time for TR building

There are possibilities of exploiting the various mixed merging criteria with the weight coefficients for the individual nets, the element numbers, the sizes of clusters, the time delay, etc. Very important is also the dependency circuit size vs time. As our experiments show, this dependency is close to linear by using the OCR method (Figure 3). Experiments are conducted with the library of the IBM01-IBM18 [Alpert, 1998].

Figure 4 shows the example of the Reduction Tree T^R and the dependency of the cluster external nets'number vs reduction steps starting from element 11 for some circuit with 17 elements. It can help to receive the better dividing the circuit into partitions. The cutting χ_8 at eighth level shows that partitions (1, 2, 3, 4, 5, 6, 6, 7, 8, 9. 10) and (11, 12, 13, 14, 15,16, 17) create the minimal cut with 5 nets. Other cuttings have more external nets. No other partitioning method has such possibility.

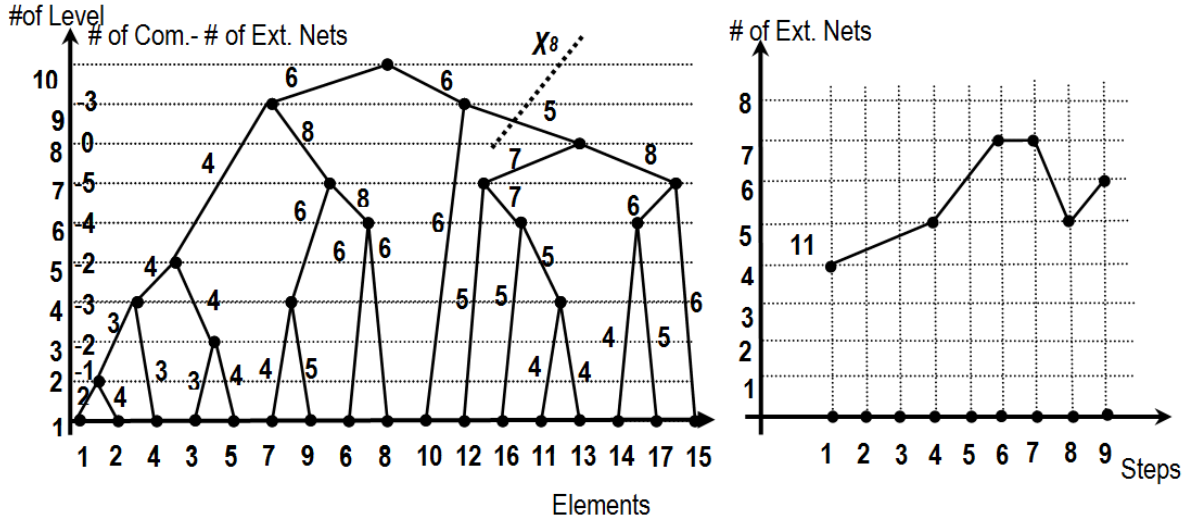

Fig. 4. Example of the Reduction Tree and the diagram dependency of the external nets clusters' number vs reduction steps

Partitioning

It is necessary to obtain the partitioning $P^* = \{P_1, \ldots, P_k\}$ for the set of elements $P = \{p_1, \ldots, p_n\}$ so that the quality function is optimized:

$$Q(P^*) \rightarrow opt\, Q\left(\widetilde{P}\right), \quad \widetilde{P} \in D,$$

while satisfying such or some other constraints:

$$(\forall P_i, P_j \in P^*)\ [|P_i| \approx |P_j|].$$

Set $\widetilde{P}$ is the arbitrary partitioning in the feasible region *D*, *k* – the number of partitions. The solution should also satisfy the following additional conditions:

$$(\forall P_i \in P^*)\,[P_i = \{p_{i1}, \ldots, p_{ini}\},\ p_{ij} \in P;\ i = 1, \ldots, k;\ j = 1, \ldots, n_i];$$

$$(\forall P_i \in P^*)(P_i \neq \varnothing);$$

$$(\forall (P_i, P_j) \in P^*)[P_i \cap P_j = \varnothing].$$

The n_i is the number of elements of *i-th* partition.

By the OCR method we recommend to solve partitioning problem in the two stages: initial partitioning and partitioning optimization.

Initial partitioning

Using the constructive method, it is desirable to find an initial solution at the first stage, which must be improved by the iterative method at the second stage. The important peculiarity of the approach developed is that it is recommended to use the hierarchical circuit clustering, obtained by the OCR method at the both stages. For initial partitioning it is possible to use following algorithms: serial, parallel-serial, and dichotomy.

By the serial algorithm on the Reduction Tree T^R the vertex is found, whose number of elements is equal to or greater than the desired value. If the number of elements is what we desire, we create the first partition and move forward to the next partitions. If this number is greater then desired, the problem is to remove the necessary number of elements. The problem recursively continues to final solution.

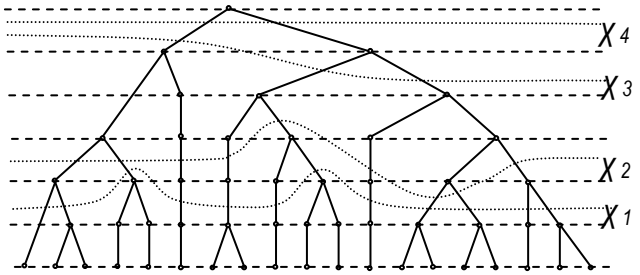

Fig. 5. Cuts in the Reduction Tree

Any cut of the Reduction Tree T^R by an arbitrary line forms subtrees (forest), the set of initial vertices (leaves) of each subtree can be considered as the set of elements of a single partition. For example, cut χ_1 at the Figure 5 can be used when it is necessary to split the circuits into the minimum number of partitions with a number of elements not greater than 2. Seven partitions are formed directly. The five elements remain ungrouped. For their assignment it is possible to construct the Reduction Tree in the subsequent repetition, and so on until the completion the problem. Cut χ_2 creates the three groups for three elements. To form the remaining partitions it is necessary to continue the process as in the previous case. Cut χ_3 gives good initial solution for three partitions (with 6, 5 and 8 elements). The first subcircuit can be directly incorporated into the solution as one partition, and then it would be necessary to transfer one element from the third to the second subcircuit. We obtain the partitions with 6, 6 and 7 elements. Cut χ_4 dived the circuit into two partitions with 6 and 13 elements. For receiving two approximately equal partitions we need to transfer three elements from larger partition into smaller one.

Dichotomy algorithm performs the top-down circuit division with constraint on the number of elements that should be equally divided to the desired number of elements at one partition. In the first step we consider the two highest vertices. This determines the number of possible partitions that can be formed from each vertex and the

number of elements in the remainders. The next step is to transfer the remaining elements from one piece to the other in the optimal way. The problem is reduced to the two new problems of the same type but of lesser size. In both cases, we use identical procedures to transfer the small number of elements from one piece to another, procedures which are performed recursively on the sets that decrease from the step to the step.

Partitioning optimization

For partitioning optimization at the first step we build the separate ORT for two partitions T^{R1}, T^{R2} and use the following procedures:

- *P1.* The exchange: arbitrary element from one partition and arbitrary element (cluster) of other partition.
- *P2*. The exchange: arbitrary clusters between two partitions.
- *P3*. The exchange: arbitrary sets of clusters between two partitions.
- *P4*. The transference: arbitrary element (cluster) from one partition to another.
- *P5*. The transference: arbitrary set of clusters and elements from one partition to another.

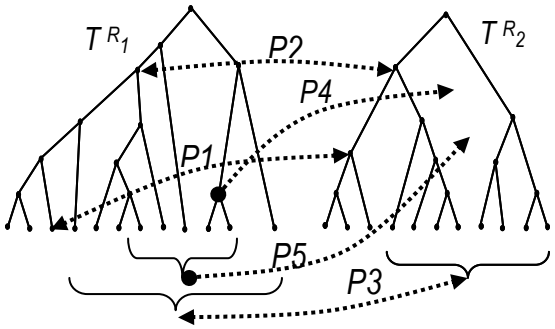

Fig. 6. Procedures for partitioning optimization

Some experimental bipartition results of test-case IBM01 [Alpert, 1998] by using our approach are shown at Table 1. For 100 randomly generated initial solutions we perform optimization. For escaping from the local extreme we use the perturbations by replacing clusters with the smallest value of the solution's worsening. The number of such perturbation is presented at the first column. For all initial solution we received the cut with180 nets that we think it is an optimal result (our conjecture), as was received from other investigators by using another approaches [Karypis, 1997], [Saab, 2000]. Forth column shows the number of the best solutions that were received for all randomly generated initial solutions. The fifth-eighth columns show the numbers of solutions that have 1, 2, 5 and 10 % deviation from the best solution. Last columns show the average solution, average number of iteration and average runtime.

Table 1. Partitioning results for IBM 01

# steps of perturbations	Maximal solution	Minimal solution	# of optimal solutions	Deviation from the best solution				Average solution	Average # of iterations	Average runtime, s
				<= 1 %	<= 2 %	<= 5 %	<= 10 %			
0	707	180	1	6	27	38	41	307	22	174
1	699	180	1	25	57	61	61	236	33	204
2	699	180	3	43	65	67	67	225	43	233
3	699	180	4	52	68	68	68	222	53	261
5	699	180	8	60	68	68	68	222	73	318

Packaging

It is necessary to obtain the partitioning $P^* = \{P_1, \ldots, P_k\}$ for the set of elements $P = \{p_1, \ldots, p_n\}$ so that the total number of partitions is minimized:

$$k \rightarrow min,$$

while satisfying the given constraints:

$$(\forall P_i \in P^*)\,[(n_i \le n_{i\,max}) \,\&\, (m_i^{ex} \le m_{i\,max}^{ex})].$$

Here n_i and m_i^{ex} are the numbers of elements and external nets (IO terminals) in each partition P_i that can not exceed the upper bounds $n_{i\,max}$ and $m_{i\,max}^{e}$. It should also satisfy the same additional constraints as for partitioning.

By the OCR method we also recommend to solve the packaging problem in two stages: initial packaging and packaging optimization.

Initial packaging

The algorithm begins to operate on the cluster of the Reduction Tree T^R, which appears first in violation of the constraint on the number of elements. From this cluster we form the first partition with as many as possible elements without violation on constraint on the number of external nets. Two strategies are used: to remove the minimal number of elements and to identify the best cluster without violation on constraints. The next step consists of the addition of the maximum number of elements. The experiments reveal the advantage of simultaneous combination of both strategies that perform iterative removal and addition of elements and clusters. The partitions separated first have a good density; but the final ones – bad. This is caused first of all by the "greedy" partitioning by serial strategy. As a result, the number of partitions can be greater than the optimal.

Table 2: Packaging Results for FPGAs

Circuit	# of CLBs	# of Nets	Numbers of FPGAs				
			[Kuznar, 1993]	[Nan-Chi Chou,1994	[Bazylevych, 2000]		Theoretic optimum
					Initial	Opt.	
a) with 64 CLB and 58 IO (Xilinx XC2064)							
C499	74	123	2	-	3	2	2
C1355	74	123	2	-	2	2	2
C1908	147	238	3	-	4	3	3
C2670	210	450	6	-	6	6	4
C3540	373	569	6	6	8	6	6
C5315	531	936	11	12	12	10	9
C7552	611	1057	11	11	12	10	10
C6288	833	1472	14	14	14	14	14
b) with 320 CLB and 144 IO (Xilinx XC3090)							
s15850	842	1265	4	3	4	3	3
s13207	915	1377	7	6	6	4	3
s38417	2221	3216	12	10	10	8	7
s38584	2904	3884	17	14	14	10	10

Packaging optimization

The partitions with the number of elements lesser than the constraint merge into one or several without violating it. The next step is the optimization on the set of all partitions that allows to increase the number of elements, but not to exceed of constraints on the first group of partitions, which were not subject for merging. Often such optimization substantially decreases the number of external nets of final partitions up to the desired value. If this is impossible to obtain, then the new final partition is divided into two smaller ones. The first partition should be without violations on constraints; the second may exhibit the violation on the number of external nets, if it is not possible to create it without violation, and so forth, up to the completion of the problem.

The experiments confirm the high efficiency of this approach on the set of some well-known test-cases. We merged only 20% of the better independent pairs at the every level of the Reduction Tree T^R generation. The test results (#FPGAs) are shown at the Tables 2. We used the 64 CLBs and 58 IOs constraints (FPGA Xilinx XC2064) for the tests with Table 2 and 320 CLBs and 144 IOs (FPGA Xilinx XC3090) for the tests with Table 3. As one could see from the tables the obtained results are not worse, and in the 5 cases from 12 they are the best among the known and are optimal. If our results are not being theoretically optimal, they are close to the optimal solutions and differ from them minimally, i.e. only by one partition (circuits c5315, s13207, and s38417) or two partitions (circuit c2670).

Floorplanning and placement

Combined hierarchical clustering and decomposition can be used for floor planning and placement. Such an approach is especially effective for large and very large-scale problems. The problem is solved in several stages:

- the bottom-up free hierarchical circuit clustering;
- the mathematical description of clusters by macro models at every level of decomposition;
- top-down multilevel placement with global and local optimization at every level of decomposition by using macromedels.

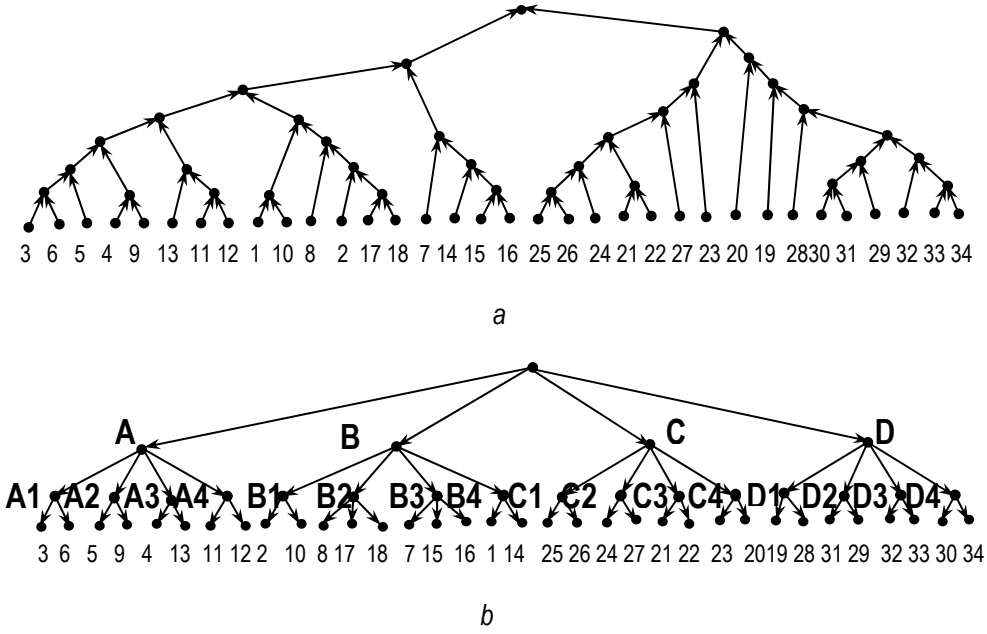

Fig. 7. Bottom-up hierarchical circuit clustering (a) and top-down 3-level of decomposition for Steinberg test-case

Figures 7 and 8 show the results of exploiting the developed approach for the Steinberg placement test-case [Steinberg, 1961]. For the first step (*a*) we build the ORT, for the next (*b*) – the tree level of decomposition (with 4, 16 macro models and 34 initial elements at the lowest level). At every level of decomposition we received some initial solution and performed it optimization using macro models (Figure 8 *a, b* and *c*) by Scanning-area method [Bazylevych, 1981, 1997]. We got the results of L_e = 4119,7 (the summary length of all connections with the Euclidian metric), which is the best comparatively with the other known solutions.

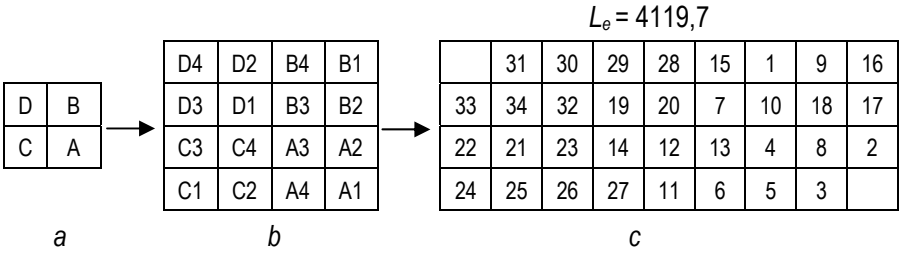

Fig. 8. Multilevel placement for Steinberg test-case

Conclusions

Hierarchical circuit clustering is a good precondition for solving the physical design problems of large and very large-scale electronic devices - VLSI, SOC and for PCB. For hierarchical clustering we developed the OCR method. Basic algorithms were proposed for partitioning, packaging, floorplanning and placement problems. They were used to obtain the initial solutions with not very large number of macromodels, as well as for their optimization. The proposed algorithms have some new properties, for example, they can be efficient in choosing the most appropriate number of partitions into which it is necessary to divide the circuit; arbitrary division coefficient can be chosen for partitioning; the same procedures can be used for initial solution and their optimization. The suggested algorithms have near linear computational complexity and provide good quality of results. For all test-cases investigated, the results are not worse, and in many cases they are better comparatively with obtained by other known methods. For some cases, the optimal results were received for the first time.

Bibliography

[Bazylevych, 1975] R.P.Bazylevych, S.P. Tkachenko. The partitioning problem solving by the parallel reduction method. In: Vychuslitelnaia technika: Materialy konferencii po razvitiiu technicheskich nauk po avtomatizirovannomu proektirovaniiu, V. 7, Kaunas, 1975, pp. 295-298 (In Russian).

[Bazylevych, 1981] R.P. Bazylevych. Decomposition and topological methods for electronic devices Physical Design Automation, Lviv: Vyshcha shkola, 1981, 168 P, (In Russian).

[Bazylevych, 1997] Roman P. Bazylevych, Taras M. Telyuk. VLSI and PCB elements placement optimizing using hierarchical scanning area method. 42 Internationales Wissenschaftliches Kolloquium. Technische Universitat Ilmenau. Ilmenau. 1997, pp. 594-599.

[Bazylevych, 2000] R.P. Bazylevych, R.A. Melnyk, O.G. Rybak. Circuit partitioning for FPGAs by the optimal circuit reduction method. In: VLSI Design, Vol. 11, No 3, pp. 237-248, 2000.

[Bazylevych, 2002] Bazylevych R.P. The optimal circuit reduction method as an effective tool to solve large and very large size intractable combinatorial VLSI physical design problems. In: 10-th NASA Symp. on VLSI Design, March 20-21, 2002, Albuquerque, NM, USA, pp. 6.1.1-6.1.14.

[Bazylevych, 2002] Bazylevych R. P., Podolskyy I.V., Bazylevych P.R. Hierarchical clustering – efficient tool to solve nonpolinomial combinatorial problems of high sizes. In: Shtuchnyy intelekt, Ukraine NAN, No. 3, 2002, pp. 447-483, (In Ukrainian).

[Bazylevych, 2007] R. Bazylevych, I. Podolskyy and L.Bazylevych. Partitioning optimization by recursive moves of hierarchically built clusters. In: Proc. of 2007 IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems. April, 2007, Krakow, Poland, pp. 235-238.

[Garbers, 1990] Jorn Garbers, Hans Jurgen Promel, Angelika Steger. Finding Clusters in VLSI Circuits. In: Proc. of IEEE/ACM Intern. Conf. on Computer-Aided Design, Santa Clara, 1990, pp. 520-523.

[Cong, 1993] Jason Cong and M'Lissa Smith. A Parallel Bottom-up Clustering Algorithm with Applications to Circuit Partitioning in VLSI Design. In: Proc. 30th ACM/IEEE DAC, 1993, pp. 755-760.

[Dutt, 1996] S.Dutt, W.Deng. VLSI circuit partitioning by cluster-removal using iterative improvement techniques. In: ICCAD'96, pp. 194-200.

[Karypis, 1997] G.Karypis, R.Aggarwal, V.Kumar, and S.Shekhar, Multilevel hypergraph partitioning: Application in VLSI domain. In: DAC 97, pp. 526-529.

[Saab, 2000] Yousseb Saab. A new multy-level partitioning algorithm. In: VLSI Design, Vol 11, No 3, pp. 301-310, 2000.

[Alpert, 1998] C.J. Alpert. The ISPD-98 Circuit Beanchmark Suit. In: Proc. ACM/IEEE Intern. Symposium on Physical Design, April 1998, pp. 80-85.

[Kuznar, 1993] Kuznar, R., Brglez, F. and Kozminski, K. Cost minimizationof partitions into multiple devices. Proc. Of IEEE/ACM 30th Design Automation Conference, 1993, pp. 315 - 320.

[Nan-Chi Chou, 1994] Nan-Chi Chou, Lung-Tien Liu, Chung-Kuan Cheng, Wei-Jin Dai and Rodney Lindelof. Circuit partitioning for huge logic emulation systems. Proc. 31 ACM/IEEE Design Automation Conference, 1994, pp. 244-249.

[Steinberg,1961] Steinberg L. The backboard wiring problem a placement algorithm. SIAM Review, 1961, v.3, ¹1, pp.37-50.

Authors' Information

Roman Bazylevych *– Prof. dr.hab., University of Information Technology and Management in Rzeszow; 1/42 Dovbusha str., Lviv, 79008, Ukraine;*

email: rbazylevych@wsiz.rzeszow.pl;

Major Fields of Scientific Research: combinatorial optimization, computer-aided design, algorithms

Marek Pałasiński *– Prof. nadzw. dr hab., Chair of Mathematics and Computer Science Foundations, University of Information Technology and Management in Rzeszow;*

email: mpalasinski@wsiz.rzeszow.pl;

Major Fields of Scientific Research: theoretical computer science, theory of algorithms, graph theory, data mining and algebraic logic

Lubov Bazylevych *– senior scientist, Institute of Mechanical and Mathematical Applied Problems of the Ukrainian National Academy of Sciences;*

email:lbaz@iapmm.lviv.ua;

Major Fields of Scientific Research: applied mathematics and mechanics, combinatorial optimization, computer science problems.