UDC 004.67:621.3.049.77

Mozhzhukhina A. V. LSI element placement method based on deep reinforcement learning

Mozhzhukhina A. V.
National Research University of Electronic Technology, Moscow, Russia,
SIC «Technological Center», Moscow, Russia

Abstract. *In connection with the increasing complexity of large integrated circuits (LSI), the problem of improving computer-aided design systems (CAD) is acute. In particular, fast solutions are currently required at the stage of topological design of microcircuits, especially in domestic CAD systems. This article discusses the possibilities of using modern artificial intelligence capabilities in the field of microcircuit design. It substantiates the need to use and brief descriptions of some technologies and sections of artificial intelligence, such as deep reinforcement learning (DRL) in connection with the formalization of the task in the form of a Markov decision process, neural networks in connection with the representation of LSI in the form of a weighted graph with feature vectors. Additionally, an analysis of the most common architectures of neural networks is given. In addition, a method for placing LSI elements at the stage of topological design based on DRL using graph neural networks (GNN) is presented with a brief description of the stages: processing the initial LSI design and the characteristics of the elements to be placed and the basic matrix crystal (BMC), training or using an agent, asserting the placement and maintaining related information.*

Keywords: *LSI, topology, EDA-system, GNN, reinforcement learning, deep learning.*

DOI 10.54092/25421085_2022_2_31

Рецензент: Сагитов Рамиль Фаргатович, кандидат технических наук, доцент, заместитель директора по научной работе в ООО «Научно-исследовательский и проектный институт экологических проблем», г. Оренбург

Introduction

More and more areas of science and industry are striving to introduce the achievements of the field of artificial intelligence into their processes. The field of electronics was no exception, where many processes are associated with automation [1, 2]. In particular, at present, a lot of research is being carried out on the introduction of artificial intelligence in the process of designing microcircuits [3-7].

Since the technological processes for the production of microcircuits are decreasing, and the number of elements is increasing, the development process becomes more complicated and more expensive. Previously, this was compensated by the use of CAD on standard algorithms. However, in the current realities, standard algorithms, even when using parallelization, turn out to be too long and inefficient.

In the field of domestic production of LSI, an improvement in CAD using modern methods and tools is also required. One of the stages requiring changes is the stage of topological design, in particular, the operation of placing LSI elements on the BMC [8].

Purpose of the research

The most efficient, but the slowest and most costly, in finding a solution to the problem of placing LSI elements are greedy algorithms, which boil down to enumeration of all possible options. The main part of the standard algorithms used is their modification or parallelized version. With an increase in the number of elements in microcircuits, such algorithms become extremely inefficient.

Thus, new modern methods and means are needed. Taking into account the achievements in the field of artificial intelligence [9, 10] and the first studies on its application in the field of electronics, it was decided to analyze the possibility of using them for the problem of placing LSI elements.

Materials and Methods

For the correct selection of methods and means for solving the problem of placing LSI elements, it is necessary to formalize it.

Used sets:

1. $E = \{e_1, e_2, \dots, e_n \dots, e_N\}$ – set of elements of the microcircuit for placement;
2. $P = \{p_1, p_2, \dots, p_m, \dots, p_M\}$ – a fixed set of positions in which elements from the set E must be placed.

The main condition in this case:

$$\sum_{n=1}^{N} S_{e_n} \leq \sum_{m=1}^{M} S_{p_m}.$$

Minimum required metrics for evaluating the final placement of LSI elements:

1. estimation of the total distance between connected elements;
2. calculation of delays with a margin to assess the progress of the signal while maintaining the requirements for the microcircuit;
3. assessment of the microcircuit filling density with elements in order to avoid excessive concentration of elements at a specific point on BMC.

In the general case, the placement problem is reduced to a Markov decision-making process [11]. It can be solved by using reinforcement learning to approximate the optimal policy through learning reward maximization. To work with a complex structure - a LSI project, neural networks will be required [12].

Some of the most common neural network architectures are: Fully Connected Neural Network (FCNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Graph Neural Network (GNN). They can work with all possible representations of information, however, the problem is how well they are able to hold a representation of the relationship of information elements to each other. The following are brief characteristics and most common uses of these neural network architectures.

FCNN. Used for a variety of types of information from simple numbers to images and sounds. However, to keep the structure, i.e. "Remember" about the connection of the elements at the input to each other, can not in principle. In addition, this neural network is capable of receiving only the initially specified size of information.

CNN. Mainly used for image processing. Cannot hold structure and is unable to accept information of different sizes.

RNN. Most often used in the processing of textual information. Can hold simple links between pieces of information for a short time. The size of the information may vary. However, the longer the information supplied to the input, the worse the relationship between the first and the last units of information is retained in the network memory.

GNN. Works with graphs that can represent different types of coded information. In addition, this network holds the structure of the graph and can accept graphs of different sizes, provided that the characteristics of the vertices and connections are the same.

Thus, a GNN was chosen.

Formalization of the problem using Markov decision processes is as follows [13-15]:

$$J(\theta, G\{E \to P\}) = \mathbb{E}_{\pi_\theta}[R_{p,G}] = \sum_{p} \pi_\theta(p|G)\, q_{p,G},$$

where θ are the weights of the approximating function, i.e. neural network, π_θ is the agent's policy, G is a graph representing the placement of elements of the set E in the cells of the set P, $R_{p,G}$ is the reward, $q_{p,G}$ is the value function.

The neural network will be used to evaluate placement. To do this, it is required to present the LSI graph in a special way (fig. 1) - to select the necessary characteristics that are important for evaluation.

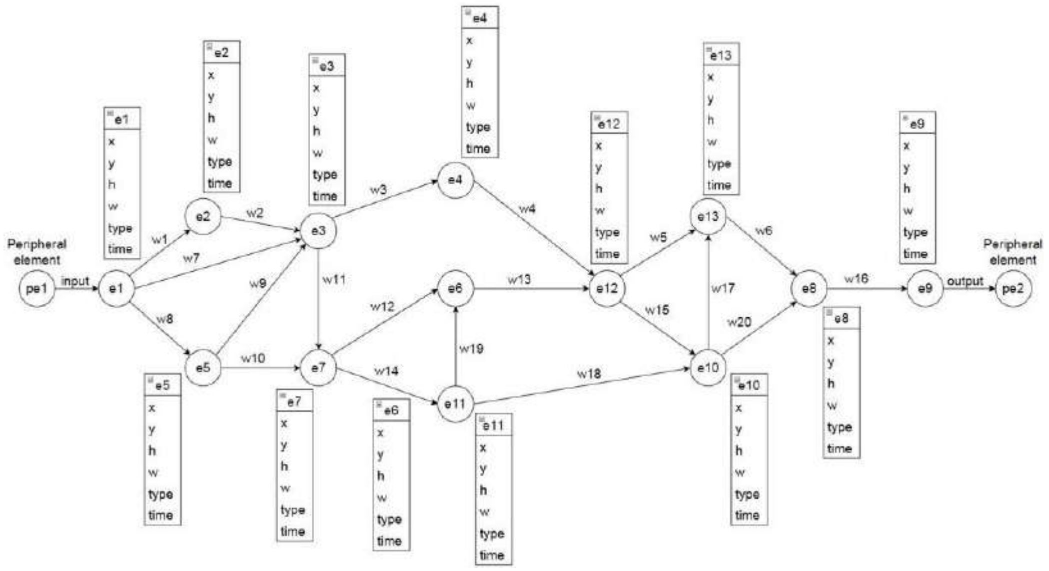

Fig.1. Representation of a part of the LSI in the form of a graph structure with vectors of characteristics of nodes and weights of connections

Results

The placement of LSI elements according to the developed method is carried out in stages (Fig. 2). Before the start of placement, preparatory work is carried out: the collection of information about the LSI, the connections of the elements, the peripheral elements and their placement, the design and area of the BMC.

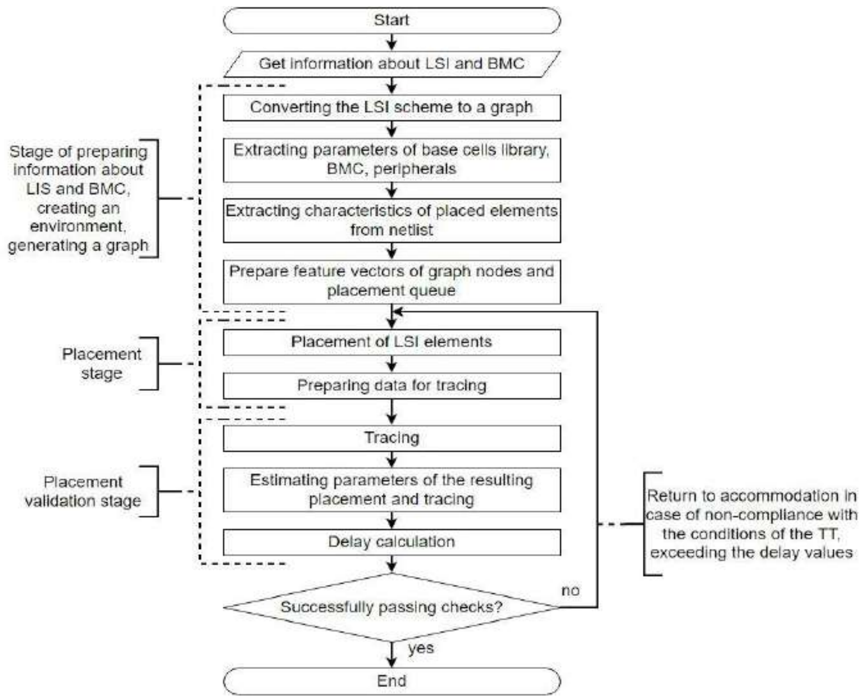
Start
Get information about LSI and BMC
Converting the LSI scheme to a graph
Extracting parameters of base cells library, BMC, peripherals
Extracting characteristics of placed elements from netlist
Prepare feature vectors of graph nodes and placement queue
Stage of preparing information about LIS and BMC, creating an environment, generating a graph
Placement of LSI elements
Preparing data for tracing
Placement stage
Tracing
Estimating parameters of the resulting placement and tracing
Delay calculation
Placement validation stage
Return to accommodation in case of non-compliance with the conditions of the TT, exceeding the delay values
Successfully passing checks?
no
yes
End

Fig.2. LSI Element Placement Method Using Reinforcement Deep Learning Agent

At the first stage, the received information is converted into working data structures, including information about the connections of elements is converted into a matrix form, the necessary variables, the agent and the neural network are initialized, and the queue of placed elements is formed.

The next step is directly step-by-step placement of LSI elements. At this moment, the agent gets access to the queue of elements to be placed, the graph structure and the representation of the BMC. The placement takes place in several steps, repeating in a circle one after another until the queue of elements for placement is empty. The first step is to prepare a package from several placements, one element at a time, and calculate the agent's goal. The second step calculates the estimate of the neural network of the current placement, obtained on the basis of a graph structure that changes over time from the placement of LSI elements on the BMC, checking some metrics that can be checked throughout the process, and calculating rewards.

At the placement approval stage, the placement is evaluated according to basic metrics, then, if the placement is found to be satisfactory, it is submitted for tracing. The average metrics from the database for similar chips will be used to evaluate the final placement. After the trace, the usual general checks will be performed, including the calculation of delays. If all tests are passed, then placement is considered successful. In case of problems, return to the previous steps.

During the training, the methodology will additionally introduce the stage of documentation, i.e. saving the learning history of the neural network weights, their changes and agent parameters to the database, reports on placement, queues and changes in the LSI graph structure, calculations of evaluation metrics and awards, both at the end and during the entire placement.

Discussion

This method of placing elements is designed to reduce the time of the physical stage of LSI design. Based on the results of scientific research related to the implementation of artificial intelligence in related fields, as well as the analysis of various methods and tools, the use of deep reinforcement learning will lead to significant improvements.

The main direction of future research will be related to the development of a test module for implementation in the microcircuits development process of CAD "Kovcheg" for testing and comparing results with existing methods for placing LSI elements.

References

1. Гагарина Л.Г., Колдаев В.Д. Алгоритмы и структуры данных. М.: Финансы и статистика, 2009. 482 с.

2. Макушин М.В., Фомина А.В. Искусственный интеллект и рентабельность как движущие факторы развития САПР // Электроника: наука, технология, бизнес. №4 (185) С. 90–100.

3. Иванова Е. Synopsys: тренды, решения, мифы // Электроника: наука, технология, бизнес. №7 (188). С. 16–21.

4. Абасов, Р. К., Силла А. Анализ методов искусственного интеллекта САПР технологических процессов производства электронной аппаратуры. // Молодой ученый. № 24 (128). С. 41–48.

5. Познер М. «Переплетение» трендов создает фантастические возможности для микроэлектроники // Электроника: наука, технология, бизнес. №10 (181). С. 12–16.

6. G. Lu, S. Areibi An island-based GA implementation for VLSI standard-cell placement // Genetic and Evolutionary Computation Conference. Seattle, WA, USA, Proceedings, Part II, Июнь 26-30, 2004. С. 1138–1150.

7. Goldie A., Mirhoseini A. Placement Optimization with Deep Reinforcement Learning. // Proceedings of the International Symposium on Physical Design, 2020. С. 2–7.

8. Гаврилов С.В., Денисов А.Н., Коняхин В.В. [под ред. Чаплыгина Ю.А.] Система автоматизированного проектирования «Ковчег 2.1». М.: Микрон-Принт, 2001. 210 с.

9. Дейтел П., Дейтел Х. Python. Искусственный интеллект, большие данные и облачные вычисления. СПб.: Питер. 2020. 864 с.

10. Эндрю Гласснер. Глубокое обучение без математики. Том 2. Практика. М.: ДМК Пресс, 2020. 610 с.

11. Richard S. Sutton, Andrew G. Barto. Reinforcement Learning. An Introduction. Second edition. The MIT Press, Cambridge, Massachusetts, London, England, 2020. 548 с.

12. Yao Ma, Jiliang Tang. Deep learning on Graphs. Cambridge University press, 2021. 339 с.

13. Судхарсан Равичандиран Глубокое обучение с подкреплением на Python. OpenAI Gym и TensorFlow для профи. СПб.: Питер. 2020. 320 с.

14. Лапань М. Глубокое обучение с подкреплением. AlphaGo и другие технологии. СПб.: Питер. 2020. 496 с.

15. Miguel Morales. Grokking Deep Reinforcement Learning. MANNING, Shelter Island, 2020. 472 с.