

Research and development of methods for technology-independent power optimization of integrated circuits

Anastasiya Kurganskaya
ISP RAS
MIEM, HSE
Moscow, Russia
askurganskaya_1@edu.hse.ru

Alexander Kamkin
ISP RAS
HSE
PRUE
MSU
MIPT
Moscow, Russia
kamkin@ispras.ru

Sergey Smolov
ISP RAS
PRUE
Moscow, Russia
smolov@ispras.ru

Alexey Yagzhov
ISP RAS
MIEM, HSE
Moscow, Russia
aayagzhov@edu.hse.ru

Abstract—Increasing power consumption has become a major issue in the integrated circuit industry. Solving this problem will lead to a reduction in packaging costs. The main challenge in this problem is to generate a netlist that minimizes power dissipation. This paper proposes an algorithm for technology-independent power optimization of combinational circuits. The proposed algorithm is based on a probabilistic estimate of power consumption. The main idea is to rearrange the inputs of sub-circuits of the initial circuit to minimize the power consumption. The developed algorithm reduces the switching activity by 0.5% in logic circuits built in any basis without limitation.

Keywords— *integrated circuit, switching activity, power consumption, technology-independent optimization*

I. INTRODUCTION

In the era of advanced technology and increasing demands for high-performance electronic devices, the design and manufacturing of integrated circuits (IC) play an increasingly important role in digital systems. ICs is a group of electronic circuits placed on a metal plate designed with semiconductor materials. An IC is the fundamental building block for all modern electronic devices.

The complexity and functionality of ICs continue to increase, so does their power consumption. The increase in power will lead to an increase in heat dissipation cost and packaging cost [1, 2]. Reducing power consumption in control systems, especially digital circuits, is an urgent scientific and technical issue that has recently attracted increasing attention from scientists and engineers [3]. It is clear that estimating and minimizing power during the design phase is crucial to avoid costly redesigns during manufacturing.

The IC design process contains several steps: 1) system specification, 2) architectural design step, 3) RTL model design, 4) logic synthesis, 5) physical synthesis [4]. Logic synthesis transforms a cycle-level functional design into a gate-level representation [5]. It makes up the gap between the technology-independent and technology-dependent stages. The result of logic synthesis is an optimal network composed of standard cells in a given technology library. In most design systems, the logical synthesis process contains at least two important stages: technologically independent

optimization and technological mapping [6]. It is necessary to apply algorithms to optimize power consumption at all steps of the IC design process.

Logic synthesis has a waste optimization area. The technology-independent power optimization is very important, because it has a cumulative effect [7].

This paper is aimed to explore the problem of technology-independent power optimization of ICs. This paper firstly provides an overview of existing solutions for technology-independent power optimization of ICs. Then the power estimation model on a technology-independent phase is considered in section 3. After that the heuristic approach is proposed in section 4. The current results are described in section 5. Section 6 concludes the paper.

II. RELATED WORK

Power optimization of logic circuits involves techniques and methodologies to reduce power consumption while improving circuit performance (functionality remains unchanged). There are several methods for power optimization of integrated circuits. Power optimization problem is NP-complete [8]. There is no polynomial algorithm to solve this problem. In this way, all algorithms are based on different heuristics.

The technology-independent power optimization there is a sequence of logic transformations on a logic function to reduce the power model. Regarding power optimization methods in the technology-independent phase of logic synthesis, there are approaches for logic functions in different bases. Logic functions can be based on Reed-Muller (RM) logic (XNOR/OR or XOR/AND based) and traditional Boolean (TB) logic (AND/OR/NOT based) [9].

First, a number of power optimization techniques for RM circuits based on XOR/AND have been proposed, including exhaustive approaches [10], genetic algorithms (GAs) [11, 12], particle swarm optimization (PSO) [13], and others [14, 15]. Additionally, there are numerous papers available for XNOR/OR-based RM circuits. In [16] an approach is proposed to optimize power using an adaptive PSO algorithm. In [17] a binary differential evolution algorithm is described. In [18], Al Jassani et al. analyzed the characteristics of Boolean logic circuits and proposed a genetic algorithm for optimizing expressions. In [19] Yuhao Zhou et al. proposed a method based on the multilevel adaptive memetic algorithm, matrix

decomposition strategy, and parallel polarity conversion algorithm.

Secondly, CAD tools like Espresso [20], SIS [21], and ABC [22] have been developed for logic synthesis and optimization based on TB logic. In references [23] and [24], attempts were made to conceal signals with high switching probability within look-up tables (LUTs) in FPGAs.

All these methods are limited by their reliance on a specific basis (RM or TB). This article presents a basis-independent approach for optimizing logic circuits.

III. ESTIMATION MODEL

A. Sources of dissipated power

The power dissipation of CMOS digital circuits includes two components: dynamic one and static one. Reverse currents of p - n junctions, resistive load, and leakage currents are the cause of static power. Static power is dissipated when the logic element is in a fixed logic state ("0" or "1"). The reasons for the dynamic power are processes of charging and discharging of the circuit node capacitances. This energy is dissipated when the signals at the outputs of the circuit nodes are switched. And this component of power dissipation is dominant [25] in comparison with static power.

The average level of dynamic power dissipation for a single gate is estimated by the following approximation [26]:

$$P_{avg} = 0.5 \cdot C \cdot \frac{V^2}{T} \cdot E_{sw} \quad (1)$$

where C — load capacitance of the gate; V — supply voltage; T — clock cycle time; E_{sw} — average number of switchings per cycle at the output of the gate (i.e. *switching activity*).

The only multiplier in the formula that can be changed is E_{sw} (rest of them are constants for the optimized circuit). As a result, to minimize power consumption, the switching activity should be minimized.

B. Power estimation model

Power estimation methods are classified [27, 28]:

- methods based on modeling
- statistical methods
- probabilistic methods

The modeling is the simplest method and the most accurate method. The specified sets are fed to the input of the circuit. Then the behavior of the circuit is simulated and as a result, a power value is obtained. This method is costly.

The idea of statistical estimation is to repeatedly simulate the operation of the circuit. In this case, random sets are fed to the input of the circuit. The simulation takes place until the average power value has not been obtained. This method is less accurate than the previous one. A stopping criterion based on statistics is required for this method.

The probabilistic method is the fastest. This method is based on calculating the probability of switching each node of the circuit [25].

For estimation of the switching activity of integrated circuits a probabilistic model was chosen. In this idea, a signal probability is used to evaluate the switching activity. Signal probability can be calculated by the signal's input probabilities depending on the logical function of the cell. Figure 1 illustrates the formulas for calculating signal probabilities for several Boolean functions: inversion, logical AND, logical OR, logical XOR. Signal probabilities for three-input, four-input functions etc. are calculated using a similar idea.

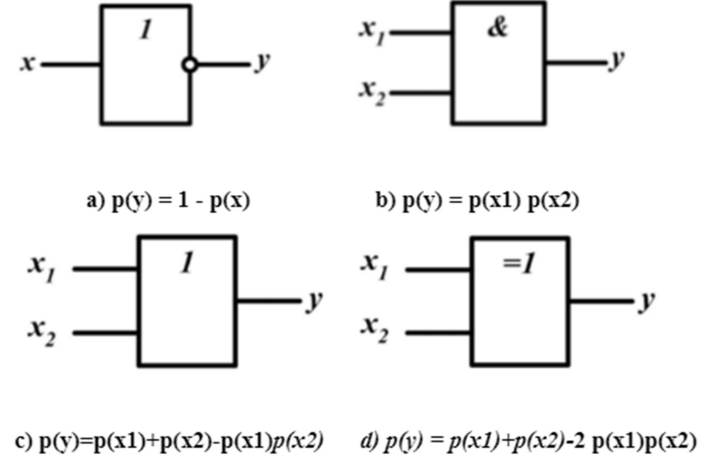


Figure 1. Formulas for calculating signal probabilities for a) inverter; b) two-input AND; c) two-input OR; d) two-input XOR.

The switching activity for gate i is the sum of the probabilities of switching from 0 to 1 and from 1 to 0. The probability of switching from 1 to 0 is equal to the product of the probabilities that the gate is in state 1 in this clock cycle and in state 0 in the next clock cycle. The formula for the probability of switching from 1 to 0:

$$p_{sw}^{1 \rightarrow 0} = p(1) \cdot p(0) \quad (2)$$

Similarly, the formula for the probability of switching from 0 to 1:

$$p_{sw}^{0 \rightarrow 1} = p(0) \cdot p(1) \quad (3)$$

Consequently, the formula for the switching activity:

$$E_{sw} = p_{sw}^{1 \rightarrow 0} + p_{sw}^{0 \rightarrow 1} = 2 p(0) \cdot p(1) \quad (4)$$

Finally, $p(0) = 1 - p(1)$. The formula for the switching activity for gate i can be calculated as follows:

$$E_{sw} = 2 p(1) \cdot (1 - p(1)) \quad (5)$$

where $p(1)$ — the output signal probability that the gate is in state 1 for g node.

IV. PROPOSED METHOD

The purpose of this article is to develop an algorithm of technology-independent power optimization of combinational CMOS circuits. According to the selected power estimation method, switching activity should be

minimized for power optimization. The proposed algorithm is based on heuristic optimization.

The input data for the algorithm is a graph of the logical circuit on an arbitrary basis. Logic gates are nodes, inputs and outputs of gates are connected by edges. The developed method is based on the local resynthesis of subcircuits. The algorithm consists of two stages: 1) select a set of subcircuits to optimize; 2) select from the received set of intersecting subgraphs a subset of non-intersecting ones.

The algorithm to select a set of subcircuits is described by the following pseudo-code:

```

for i {1, ..., L} do
  subGraph := builtSubGraph(cell_i);
  if isAssociativeFunc(subGraph) then
    newSubGraph := reorderingInputs(subGraph);
    if estimation(newSubGraph) <
estimation(SubGraph) then
      addNewSubGraph(newSubGraph);
    end
  end // if isAssociativeFunc
end // for i

```

In this pseudo-code L is depth of the initial graph.

First of all, two-level subgraphs are selected from the initial graph. To select a subgraph, proceed as follows. All nodes are analyzed. The for loop iterates through all nodes. By considering a certain node, a two-level subgraph is constructed, where the selected node is the output (the function builtSubGraph, where the current node is a parameter). For example, in Figure 2 subgraph is created for the lowest node.

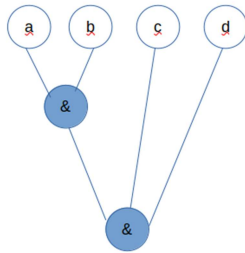


Figure 2. Subcircuit

Each subgraph is a Boolean function. If this Boolean function is associative (it is checked using the function isAssociativeFunc), it is checked for optimization. The associative function for a subgraph here is a subgraph in which the inputs can be swapped and the function remains logically equivalent. Subgraph in Figure 2 is associative. Fig. 3 shows that there are six unique subgraphs in terms of switching activity for the Boolean function presented in Figure 2. All of these subgraphs are equivalent functions.

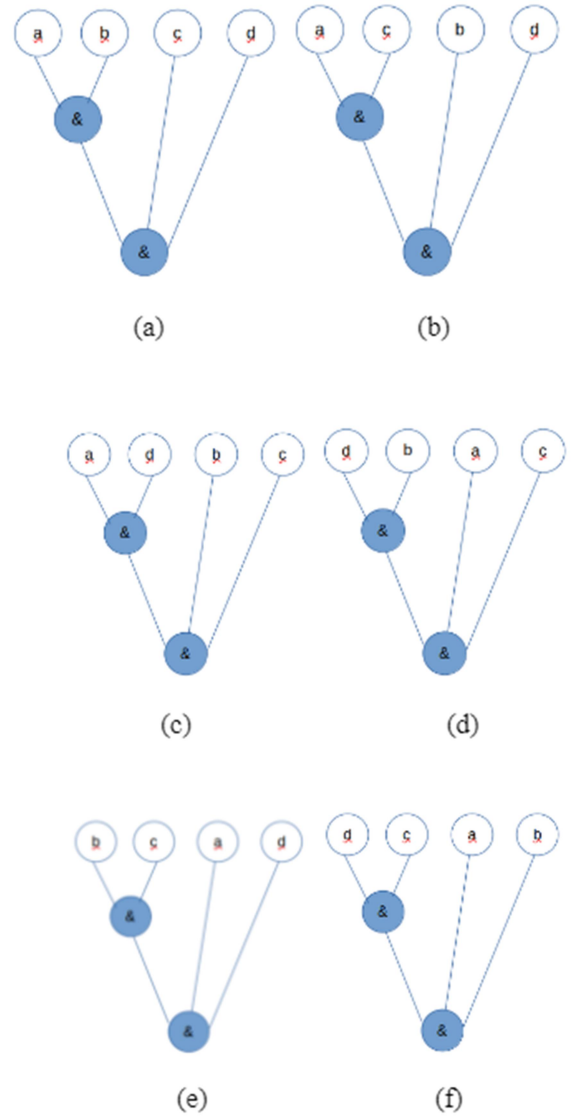


Figure 3. Example

This example shows how the overall switching activity of the subgraph can change when the inputs are rearranged. The overall switching activity for subgraph a:

$$E_{sw}^a = 2p(a)p(b) * (1 - p(a)p(b)) + 2p(a)p(b) * p(c) * p(d) * (1 - p(a) * p(b) * p(c) * p(d)) \quad (6)$$

for subgraph b:

$$E_{sw}^b = 2p(a)p(c) * (1 - p(a)p(c)) + 2p(a)p(b) * p(c) * p(d) * (1 - p(a) * p(b) * p(c) * p(d)) \quad (7)$$

for subgraph c:

$$E_{sw}^c = 2p(a)p(d) * (1 - p(a)p(d)) + 2p(a)p(b) * p(c) * p(d) * (1 - p(a) * p(b) * p(c) * p(d)) \quad (8)$$

for subgraph d:

$$E_{sw}^d = 2p(b)p(d) * (1 - p(b)p(d)) + 2p(a)p(b) * p(c) * p(d) * (1 - p(a) * p(b) * p(c) * p(d)) \quad (9)$$

for subgraph e:

$$E_{sw}^e = 2p(b)p(c) * (1 - p(b)p(c)) + 2p(a)p(b) * p(c) * p(d) * (1 - p(a) * p(b) * p(c) * p(d)) \quad (10)$$

for subgraph f:

$$E_{sw}^f = 2p(d)p(c) * (1 - p(d)p(c)) + 2p(a)p(b) * p(c) * p(d) * (1 - p(a) * p(b) * p(c) * p(d)) \quad (11)$$

In this example, the inputs are swapped in the subgraph, but the switching activity of the inputs and outputs for different subcircuits remains unchanged. However the overall switching activity has changed.

The function `reorderingInputs` in pseudo-code rearranges the inputs as described in the example and selects the subgraph with the minimum value of the switching activity. And if that value of the switching activity smaller than the switching activity of initial subgraph (it is checked using the function estimation), the subgraph has been added to the desired set (function `addNewSubGraph` in pseudo-code).

Then the task is to select from the received set of intersecting subgraphs a set of non-intersecting ones. Additionally, the solution to such a task should provide the greatest possible gain in optimizing the total

switching activity of the circuit. The problem is depicted as a graph. The graph represents subgraphs that need to be replaced as nodes, and edges indicate intersections between subgraphs. Each node has a weight. It is optimization gain that can be obtained by replacing. This problem is equivalent to the maximum weighted independent set problem. It is widely acknowledged that this is an NP-hard problem. To solve it, a greedy algorithm is used.

When the non-intersecting set is selected, subcircuits are replaced. The algorithm can be applied multiple times to the same circuit to achieve the best result.

In this method the preliminary estimation of optimization can be obtained before replacement. And this algorithm can be applied to circuits on every basis.

V. RESULTS

The algorithm was implemented with C++ language and was applied on circuits from OpenABCD [29] benchmark. The input signal probability values have been set to 0.5. GraphML files have been converted to an internal representation. Then the presented algorithm has been applied to circuits.

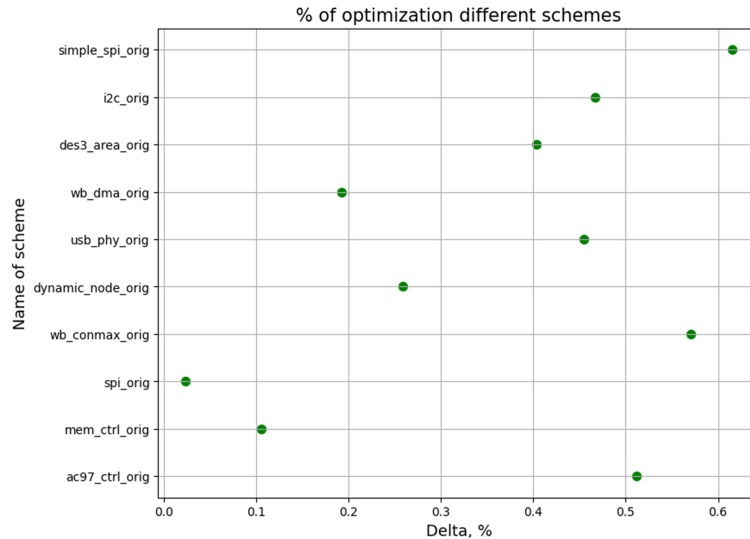


Figure 4. Results of optimization

The graphic on Figure 4 represents names of test circuits on the axis of ordinates and the optimization of the switching activity on the abscissa axis. Optimization is calculated by the following formula:

$$Delta = \frac{E_{sw}^{old} - E_{sw}^{new}}{E_{sw}^{old}} * 100\% \quad (12)$$

where E_{sw}^{old} — the switching activity of the initial circuit, E_{sw}^{new} — the switching activity of the circuit after optimization.

It can be seen that the reduction of switching activity is very small, no more than 0,7% . The gain is small because there were few subcircuits that could be optimized in original circuits. The algorithm will be enhanced with new heuristics to achieve optimal results.

VI. CONCLUSION

Power optimization of logic circuits is a crucial concern in computer engineering. This paper presents a technology-independent approach to minimize the power consumption of integrated circuits. The optimization method uses various heuristics to minimize the switching activity of a logic circuit (model of the initial circuit on the technology-independent stage of logic synthesis).

For future work, the results of refining the power minimization method suggest that the circuit will be optimized not only at the technology-independent stage, but will also contribute to technology-dependent optimization. Additionally, further research prospects

include multi-criterion optimization with other parameters, such as delay.

REFERENCES

- [1] M. Yaseen, S. Abd, and I. Mansoor, 'Critical factors affecting the adoption of open source software in public organizations', *ijcsm*, pp. 29–37, Jul. 2020, doi: 10.52866/ijcsm.2020.01.02.005.
- [2] K. Kaur and A. Noor, 'CMOS Low Power Cell Library for Digital Design', *VLSICS*, vol. 4, no. 3, pp. 43–51, Jun. 2013, doi: 10.5121/vlsic.2013.4305
- [3] W. Nebel and J. P. Mermet, Low power design in deep submicron electronics: proceedings of the NATO advanced study institute on low power design in deep submicron electronics, Il Ciocco, Lucca, Italy, 20-30 August, 1996. in NATO ASI series, no. 337. Dordrecht Boston London: Kluwer academic publ. in cooperation with NATO scientific affairs division, 1997.
- [4] A. B. Kahng, J. Hu, J. Lienig, and I. L. Markov, *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Dordrecht: Springer Science+Business Media B.V, 2011.
- [5] S. Zou, J. Zhang, B. Shi, and G. Luo, 'PowerSyn: A Logic Synthesis Framework With Early Power Optimization', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 43, no. 1, pp. 203–216, Jan. 2024, doi: 10.1109/TCAD.2023.3297069.
- [6] R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli, 'Multilevel logic synthesis', *Proc. IEEE*, vol. 78, no. 2, pp. 264–300, Feb. 1990, doi: 10.1109/5.52213
- [7] J. Lamoureux and S. J. Wilton, "On the interaction between power-aware FPGA CAD algorithms," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, 2003, pp. 701–708.
- [8] S.-Y. Lee, H. Riener, A. Mishchenko, R. K. Brayton, and G. De Micheli, "A simulation-guided paradigm for logic synthesis and verification," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 8, pp. 2573–2586, Aug. 2022.
- [9] H. Liang, Y. Xia, L. Qian, and C. Huang, "Low power 3-input AND/XOR gate design," *J Comput.-Aided Des Comput Graphics*, vol. 27, no. 5, pp. 940–945, 2015.
- [10] X. Wang, Y. Lu, Y. Zhang, Z. Zhao, T. Xia, and L. Xiao, "Power optimization in logic synthesis for mixed polarity Reed-Muller logic circuits," *Te Computer Journal*, vol. 58, no. 6, pp. 1306–1313, 2015.
- [11] M. Yang and J. Lai, "Optimisation of mixed polarity Reed-Muller functions," *Journal of Software*, vol. 8, no. 11, pp. 2770–2774, 2013.
- [12] F. Sun, P. Wang, and H. Yu, "Best polarity searching for ternary FPRM logic circuit area based on whole annealing genetic algorithm," in *Proceedings of the 10th International Conference on ASIC (ASICON)*, IEEE, Shenzhen, China, October, 2013.
- [13] W. Raju, S. Pawan, S. Charu, G. Terje, and P. Chittaranjan, "Bio-inspired hybrid BFOA-PSO algorithm-based reactive power controller in a standalone wind-diesel power system," *International Journal of Electr Energy*, vol. 31, no. 3, Article ID e12778, 2021.
- [14] A. Das and S. N. Pradhan, "Area-power-temperature aware AND-XOR network synthesis based on shared mixed polarity reed-muller expansion," *International Journal of Intelligent Systems and Applications*, vol. 10, pp. 35–46, 2018.
- [15] Z. He, T. Xia, S. Su et al., "An efficient and fast polarity optimization approach for mixed polarity Reed-Muller logic circuits," *Frontiers of Computer Science*, vol. 11, no. 4, pp. 728–742, 2017.
- [16] H. Yu, P. Wang, and H. Zhang, "Optimization of MPRM circuits based on ternary diversity particle swarm optimization," *Acta Electronica Sinica*, vol. 45, no. 7, pp. P1601–P1607, 2017.
- [17] Z. He, G. Qin, L. Xiao, F. Gu, and Z. Huo, "An efficient polarity optimization approach for fixed polarity reed-muller logic circuits based on novel binary differential evolution algorithm," *Lecture Notes in Computer Science*, vol. 10578, pp. 118–121, 2017.
- [18] B. Al Jassani, N. Urquhart, and A. E. A. Almaini, "Manipulation and optimization techniques for Boolean logic," *IET Computers and Digital Techniques*, vol. 4, no. 3, pp. 227–239, 2010.
- [19] Y. Zhou et al., 'Power Optimization for Mixed Polarity Reed-Muller Circuits Based on Multilevel Adaptive Memetic Algorithm', *International Journal of Intelligent Systems*, vol. 2023, p. 3510001, Mar. 2023, doi: 10.1155/2023/3510001.
- [20] Mcgeer P C, Sanghavi J V, and Brayton R K, *IEEE Transactions on Very Large Scale Integration Systems*, 1, p.432 (1993).
- [21] Brayton R K, Rudell R and Sangiovanni-Vincentelli A, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 6, p.1062-1081 (1987).
- [22] Berkeley Logic Synthesis and Verification Group, <https://bitbucket.org/alanmi/abc>. [15] M. Yang and J. Lai, "Optimisation of mixed polarity Reed-Muller functions," *Journal of Software*, vol. 8, no. 11, pp. 2770–2774, 2013.
- [23] L. Cheng, D. Chen, and M. D. Wong, "GlitchMap: An FPGA technology mapper for low power considering glitches," in *Proc. Design Autom. Conf. (DAC)*, 2007, pp. 318–323.
- [24] R. Dobson and K. Steinhöfel, "SA based power efficient FPGA LUT mapping," in *Proc. 15th Annu. Conf. Compan. Genet. Evol. Comput. (GECCO)*, 2013, pp. 1545–1552.
- [25] I. A. Murashko, 'METODY OCENKI RASSEIVAEMOJ MOSHCHNOSTI V CIFROVYH KMOP SKHEMAH, *Doklady BGUIR*, no. 1(17), pp. 100–108, 2007. (in Russian)
- [26] Chi-Ying Tsui, M. Pedram, and A. M. Despain, 'Technology Decomposition and Mapping Targeting Low Power Dissipation', *30th ACM/IEEE Design Automation Conference*, Dallas, TX, USA, pp. 68–73, 1993, doi: 10.1109/DAC.1993.203921.
- [27] M. Pedram, 'ACM Transactions on Design Automation of Electronic Systems', *Power minimization in {IC} design: principles and applications*, vol. 1, no. 1, pp. 3–56, Jan. 1996.
- [28] F. N. Najm, 'A survey of power estimation techniques in VLSI circuits', *IEEE Trans. VLSI Syst.*, vol. 2, no. 4, pp. 446–455, Dec. 1994, doi: 10.1109/92.335013.
- [29] 'GitHub - NYU-MLDA/OpenABC: OpenABC-D is a large-scale labeled dataset generated by synthesizing open source hardware IPs. This dataset can be used for various graph level prediction problems in chip design.' Accessed: Feb. 01, 2024. [Online]. Available: <https://github.com/NYU-MLDA/OpenABC>