Microprocessor Design Space Exploration via Space Partitioning and Bayesian Optimization

Zijun Jiang and Yangdi Lyu†
Microelectronics Thrust, The Hong Kong University of Science and Technology (Guangzhou)
†Corresponding author: yangdilyu@hkust-gz.edu.cn

***Abstract*—Design space exploration (DSE) has long been a very important topic in electronic design automation (EDA), but the growing diversity of applications and the complexity of integrated circuits make conventional DSE frameworks less effective and efficient. Therefore, an exploration algorithm that can find the optimal designs with fewer samples is demanded. This paper proposes a DSE framework for microprocessors that integrates a novel optimization algorithm with EDA flows. The proposed optimization algorithm utilizes space partitioning and Bayesian optimization to explore diverse and high-dimensional design spaces in microprocessors efficiently. Using the framework, we explore the design space of VexRiscv CPUs for TinyML workloads, where our proposed optimization algorithm obtains more Pareto-optimal designs and higher hypervolume with fewer samples.**

I. Introduction

Design space exploration is critical in microprocessor designs to achieve optimal power, performance, and area (PPA) when choosing various design parameters and implementations of microarchitectures. There has been extensive research in design space exploration using intelligent algorithms and exploration strategies, including evolutionary algorithms [1], multi-objective Bayesian optimization (MOBO) [2] and active learning [3]. Nonetheless, sub-optimal designs could be generated due to the inaccurate high-level PPA estimation and the simplified design spaces. To address these problems, an exploration algorithm that can explore the high-dimensional design space with fewer times of accurate simulations is needed.

The contributions of this paper include:

- A flexible and full-scale microprocessor design space exploration framework, which can be further extended to be integrated with any workload, hardware design, and EDA flow through a group of files with unified formats.
- A multi-objective optimization algorithm that utilizes space partitioning to apply Bayesian optimization only in promising sub-spaces, enabling efficient exploration of the microarchitecture design space with minimal samples and iterations. The algorithm successfully obtains hypervolume improvement in early iterations, and achieves better Pareto fronts with higher hypervolume gain (1.63×) and more dominant designs.

II. Design Space Exploration Framework

The overall diagram of the our framework is shown in Fig. 1.

This work is partly supported by the Guangzhou-HKUST(GZ) Joint Funding Program (No. 2023A03J0156) and the Guangzhou Municipal Science and Technology Project (Municipal Key Laboratory Construction Project, Grant No. 2023A03J0013).

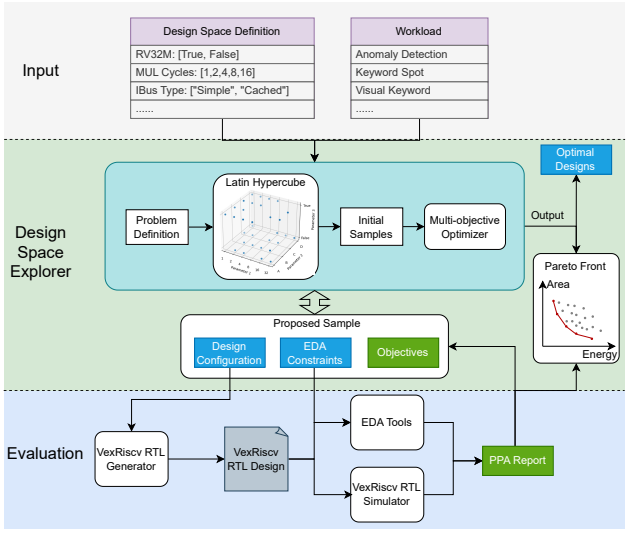

Fig. 1: The overall diagram of proposed framework

In this paper, We select Vexriscv [4], a micro-controller-level RISC-V microprocessor as the design to explore. Vexriscv has an extremely huge design space with over 40 parameters, which includes detailed implementations of the components, such as pipeline stages, bypassing types, branch prediction types, and multiply/division latency. This enables us to perform a fine-grained DSE that offers more precious trade-offs, especially for resource-constrained edge devices.

Input: The framework takes two files as inputs: 1) a design space definition file that describes the parameters and their options, and 2) a workload (e.g., an ML model) to run on the microprocessor.

Design Space Explorer: The design space explorer is the core component of the framework, it proposes samples, interacts with evaluation processes, and performs multi-objective optimizations. As shown in the middle part of Fig. 1, by parsing the design space definition into an input vector, the design space explorer builds up the problem definition. Then, the explorer generates the initial samples from the problem definition, passes them to the optimization algorithms, and gets new designs proposed by the optimization algorithms iteratively.

Evaluation: To evaluate the proposed designs, we use EDA tools to get the synthesis reports which contain the PPA information, and an RTL simulator to benchmark the designs on the given workloads. The metrics are then fed back to the explorer and the optimization algorithms.

979-8-3503-4859-0/DATE24/© 2024 EDAA

III. Optimization Algorithm

We propose a novel algorithm that learns the space partition of the design space via Monte Carlo tree search and performs localized Bayesian optimization only in promising spaces. Fig. 2 shows the 4 main steps in the proposed algorithm.

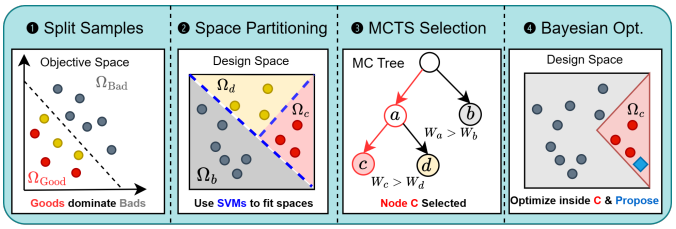

Fig. 2: Algorithm Procedure

The original idea of MCTS-based space partitioning comes from LA-MCTS [7]. As shown in Fig. 2, a tree is built to partition the space into good and bad regions. Each tree node represents a region Ω in the space and the samples in the region. To convert the single-objective optimization in [7] into a multi-objective optimization, we use the number of dominant points against the target point as the *metric* to distinguish the good regions and the bad regions. In other words, good points are dominated by fewer points, while bad points are dominated by more points, as shown in step 1 of Fig. 2.

Once the number of samples in a node reaches a certain threshold θ, using the mean value of the metric as the standard, the samples in the node are split into good and bad samples. Then, a support vector machine (SVM) based classifier is fit to learn the boundary of the good samples and the bad samples in the design space, as shown in step 2 of Fig. 2.

After the split of the tree nodes, as shown in step 3 of Fig. 2, the selection process of MCTS starts from the root node. To guide the selection of MCTS, each node holds a weight calculated from the statistics of the node. Inspired by the method proposed by MO-MCTS [8], we formulate the weight of node a, $W(a)$ based on hypervolume, as shown in Fig. 3.

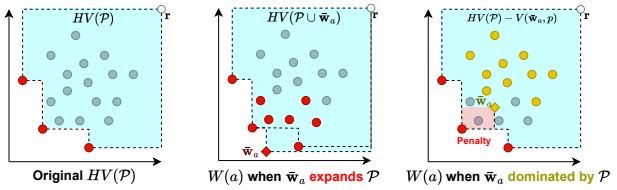

Fig. 3: visualization of $W(a)$. $\bar{\mathbf{w}}_a$ represents an optimistically predicted point of the selected region. The blue area represents the hypervolume of the current Pareto front. When it is extended by $\bar{\mathbf{w}}_a$, $W(a)$ is increased. Otherwise, $W(a)$ is reduced by the red area as a penalty.

From the root node, the succeeding node with the largest weight is selected on each branch, until a leaf node that cannot be split is selected. After the leaf node is selected by the MCTS algorithm, the Bayesian optimization using predictive entropy search (PES) acquisition [6] is performed only in the region bounded by the SVM classifier of the selected node, and the new sample with the highest acquisition value is proposed for PPA evaluation.

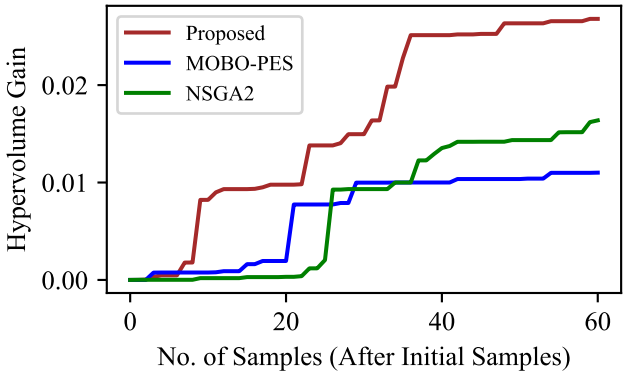

Fig. 4: The average hypervolume gain after the initial samples

IV. Evaluation Results

We use Vivado to perform the synthesis of RTL designs and get the PPA results on the FPGA(`xcau25p-sfvb784-1-i`), and use MLPerf™ Tiny benchmark as the workloads of VexRiscv designs. We set two objectives to be optimized by the framework: area (LUT+FF utilization) and energy (total energy consumed to run workloads). Starting from the same set of 40 initial samples, we compared the proposed algorithm with NSGA2 [5] and MOBO [2].

Hypervolume: Fig. 4 shows the average hypervolume gain of different optimization algorithms on all 4 TinyML benchmarks. As shown in the figure, the proposed algorithm expands the hypervolume earlier than other algorithms and finally achieves the highest after 100 total samples.

Pareto front: Our algorithm finds more optimal designs as well, it contributed 47.4% of the designs in the merged Pareto front of the 3 algorithms, while NSGA2 and MOBO only contributed 26.3% and 36.8% In benchmark KWS, the Pareto front explored by NSGA2 is entirely dominated by the one explored by the proposed algorithm.

V. Conclusions

In this paper, we proposed a design space exploration framework for microprocessors with a novel exploration algorithm. Experimental results demonstrated that our framework outperforms other DSE algorithms with more Pareto-optimal designs and higher hypervolume even with a small number of samples.

References

[1] Y. Liao *et al.*, "Efficient system-level design space exploration for high-level synthesis using pareto-optimal subspace pruning," in *ASPDAC*, 2023.

[2] B. Reagen *et al.*, "A case for efficient accelerator design space exploration via Bayesian optimization," in *ISLPED*, 2017.

[3] C. Bai *et al.*, "BOOM-Explorer: RISC-V BOOM Microarchitecture Design Space Exploration Framework," in *ICCAD*, 2021.

[4] "VexRiscv: A FPGA friendly 32 bit RISC-V CPU implementation." github.com/SpinalHDL/VexRiscv.

[5] K. Deb *et al.*, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, 2002.

[6] D. Hernandez-Lobato *et al.*, "Predictive entropy search for multi-objective bayesian optimization," in *Proceedings of The 33rd International Conference on Machine Learning*, PMLR 48:1492-1501, 2016.

[7] L. Wang *et al.*, "Learning search space partition for black-box optimization using monte carlo tree search," in *NeurIPS*, 2020.

[8] W. Wang and M. Sebag, "Multi-objective Monte-Carlo tree search," in *ACML*, 2012.