RESEARCH ARTICLE

ECEJOURNALS.IN

AI-Driven Design Space Exploration Framework for Low-Power FPGA-Based Heterogeneous Computing

S. Sindhu*

Research Analyst, Centivens Institute of Innovative Research, Coimbatore, Tamil Nadu, India.

KEYWORDS:

Design Space Exploration (DSE),
Bayesian Optimization,
FPGA,
Low-Power Design,
Heterogeneous Computing,
Power-Delay Product (PDP),
VLSI.

ARTICLE HISTORY:

Submitted : 11.12.2025
Revised : 05.01.2026
Accepted : 10.02.2026

https://doi.org/10.31838/JIVCT/03.02.01

Abstract

This has been enabled by the growing complexity of heterogeneous computing architectures which concurrently combine CPUs and Field-Programmable Gate Arrays (FPGAs) to produce a huge and non-linear design space which is difficult to optimise manually. The conventional Electronic Design Automation (EDA) software tends to fail to balance between power-intensive needs and rigorous power constraints on edges and devices in mobile contexts. This paper will suggest an AI-based Design Space Exploration (DSE) platform that is specifically focused on low-power systems based on FPGA. The framework is based on a Bayesian Optimization (BO) engine that finds its way through the vast configuration space (high-dimensional) of loop tiling, resource allocation, and clock frequency, through the application of a Gaussian Process surrogate model. The optimization goal is aimed at minimising the Power-Delay Product (PDP) that is to make the hardware configurations found after the discovery to provide the best compensation between energy consumption and latency. Experimental findings with classic signal processing and deep learning objectives prove the idea that the suggested framework finds Pareto-optimal designs much faster than customary heuristic-based strategies. In particular, the framework attains as large as 35 percent reduction in PDP alongside reduction in total exploration time by 5 times the random search methods. The suggested solution is a fully scalable automated, energy efficient hardware-software co-design solution to modern embedded systems.

Author's e-mail: sindhuanbuselvaneniya@gmail.com

How to cite this article: Sindhu S. AI-Driven Design Space Exploration Framework for Low-Power FPGA-Based Heterogeneous Computing. Journal of Integrated VLSI, Embedded and Computing Technologies, Vol. 3, No. 2, 2026 (pp. 1-6).

Introduction

The fast development of Artificial Intelligence (AI) and the Internet of Things (IoT) have necessitated an essential change in the paradigm of the computing architecture toward the network edge at which the intelligence should be deployed in proximity to the data source.[1] In this regard, heterogeneous computing (specifically, combination of general-purpose computer units to single-purpose Field-Programmable Gate Arrays (FPGAs)) has been a dominant topology in edge devices and large-scale data centres.[2] Fpga offers a well-defined blend of hardware-based parallelism and reconfigurability, which it is uniquely able to offer deterministic latency and highly energetically efficient enough to handle mission-critical tasks like autonomous driving systems and real-time vision systems.[3] Nevertheless, the flexibility of FPGA-based systems is achieved sacrificing enormous complexity of architecture. The current High-Level Synthesis (HLS) systems can show designers an enormous number of knobs or optimization pragmas to the parallelism factors, memory banking choices, and clock frequency scaling; potentially consuming billions of possible hardware designs.[4] This is commonly known as the Design Space Explosion, and causes manual synthesis by human engineers to be an intractable problem, particularly because one actual synthesis run can require several hours to finish.[5]

In order to explore this high-dimensional space without tedious and time-consuming hardware implementation, this study suggests an AI-based Design Space Exploration (DSE) framework. The proposed framework takes advantage of the Bayesian Optimization (BO), which is a statistically-based search method that is an especially useful technique to optimise costly search functions of the black-box type.[6] Using a Gaussian Process as a substitute model, the engine gets to know how the high-level design decisions relate to the hardware results, allowing one to search intelligently, guided by the goal to find a good design point that can be achieved in only a small number of synthesis runs.[7] In contrast to the traditional forms of DSE of which the emphasis is typically on raw throughput alone, our technique is based on the Power-Delay Product (PDP) as the optimization goal. This guarantees that the framework (capability) determines the compromising sweet point in terms of energy needed by the low-power embedded architecture.[8]

To date, there are two key contributions of the work, namely: (first) we create a power-aware surrogate model that can be used to estimate the post-implementation power usage and resource consumption directly based on high-level C/C++ attributes, which in effect skips the lengthy physical implementation phases.[9] Second, we present a multi-objective optimization stream that incorporates this model with a concurrent bayesian search engine to search the design space 5x as fast as the existing heuristic methods and attains large design space energy savings.[10] It is hoped that by finding a compromise between the high-level AI algorithms and the low-level FPGA synthesis this study will offer a scalable and automated route towards the creation of the next generation, highly efficient, energy efficient heterogeneous computing systems.

Related Work

The history of Design Space Exploration (DSE) of FPGAs has moved past the stage of inflexible mathematical heuristics to information-based intelligence. In the past, DSE High-Level Synthesis (HLS) used meta-heuristic search algorithms and analysis models to search through the architectural trade-offs of FPGA designs. Such common techniques were Simulated Annealing (SA), Genetic Algorithms (GA), and Ant Colony Optimization (ACO) [11]. Though these methods do work in small design spaces, their scalability is very challenging because as the number of design knobs grows (loop unrolling factors, array portioning, and pipelining depths etc.) the method grows exponentially.[12] One of the most significant limitations of traditional heuristics is that they are more of a black-box method; their converging to a solution on the Pareto-optimal point can take hundreds of complete hardware synthesis cycles. The computational complexity of these classical iterative searches was prohibitive to such a high level that they are, in modern FPGA workflows that may require many hours to completely run an implementation, the cost of such longer searches is now prohibitive in rapid prototyping.[5]

These constraints of conventional heuristics have led to a dramatic change toward the use of Machine Learning (ML) based Electronic Design Automation (EDA) tools. In current systems exploiting ML to generate surrogate models, which forecast hardware Quality of Results (QoR) metrics, e.g. area, timing, power, without running the real synthesis toolchain.[15] Recent literature focuses on the recent developments in predictive modelling based on Random Forests and Gradient Boosting that are applied to high-level features of a code to be able to estimate resource utilisation based on high-level code features.[1] Also, the search algorithms have been combined with active learning techniques to sample the most promising design points, which was demonstrated to significantly reduce the search time by only exploring the areas of the design space with the most potential to improve.[13] Graph Neural Networks (GNNs) have also started to be used in emerging research to model structural relationships in a netlist and give better predictions after routing than the older models based on analytical modelling.[10]

Although AI-driven DSE has reached great success, there are still a number of serious gaps, especially in terms of low-power optimization in heterogeneous settings. The currently existing HLS-DSE models are somewhat throughput or latency oriented, and instead of a primary optimization factor, power is a secondary constraint. As a result, the repertoire of frameworks particularly interested in the Power-Delay Product (PDP) as a single energy efficiency index is wanting.[8] Additionally, most DSE tools assume the FPGA is a single accelerator, and do not consider the inherent interactions between the CPU host and the FPGA fabric i.e. data transfer overheads and memory hierarchy tradeoffs, both of which are vital in heterogeneous System-on-Chip (SoC) architectures.[2] A further notable fidelity gap exists also between the early-stage ML prediction and final post-implementation outcomes because power usage is extremely sensitive to the ultimate placement and routing - phases that are commonly aimed to avoid by AI-driven frameworks.[3] The framework developed in this paper will overcome the limitations by providing a Bayesian Optimization engine which is aimed specifically at the PDP in a heterogeneous environment.

Proposed Framework Methodology

The main concept of the suggested structure is of an iterative closed-loop framework that will focus on the connexion between high-level AI algorithms and low hardware bounds. The architecture is shown in Figure 1 to be an autonomous optimization cycle that drastically lessens the amount of manual work needed in the traditional design flows of VLSI design. The initial step in the methodology will be Design Space Definition and we will be identifying the key hardware variables (x) which form the design space X. These variables consist both of architectural knobs (loop unrolling factors, pipelining depths, and partitioning strategies, among others) and arithmetic options (bit-precision quantization and target clock frequency to name a few). These features are parameterized to form the hardware design problem as a multi-dimensional optimization problem where the objective is to locate a configuration $x \epsilon 0$ - such that it minimises the PowerDelay Product (PDP).

Intelligence of the framework lies in the Bayesian Optimization Engine which is specially created to address the black-box-ness of FPGA synthesis. We use a Surrogate Model, which is a Gaussian Process Regression (GPR), it is not even computationally expensive to run a full synthesis and place-and-route cycle. The GPR can be used as an approximate model of the actual hardware implementation, so it can be used to make predictions of the actual performance and the power-consumption of a particular set of hardware, it can also give an estimate of the uncertainty. Contrarily to the case of the static analytical models, the GPR is indeed upgraded dynamically as the design points are newly sampled to enable the framework to enhance the predictive capabilities on the ground.

In order to manoeuvre the design space, the engine uses an Acquisition Function, either Expected Improvement (EI) or Upper Confidence Bound (UCB). This role plays a vital role towards tuning the trade-off between Exploration and Exploitation. Exploration stimulates the engine to explore the unexplored places of the design space where uncertainty is great, which may result in radical new low-power architectures. On the other hand, Exploitation results in a search concentrated around areas highlighted by the surrogate model as having high potential of realising a lower PDP using the available data.

The last element is the Hardware Evaluation Loop that is the physical connexion between the AI engine and the EDA toolchain like Xilinx Vivado HLS or Intel oneAPI. After the acquisition function has picked a viable candidate configuration, the framework will automatically produce a corresponding hardware description code (e.g. C++ with HLS pragmas). A ground truth evaluation on this code is then made as input into the synthesis tool. The resulting measures, namely the overall power consumption and execution latency are obtained to determine the overall PDP. This value is in turn exported back to the Bayesian engine to cluster the surrogate mannequin thus recapitulating the process. Such automated feedback makes sure that framework continuously comes to learn about the physical behaviour of the hardware, and Pareto-optimal designs that can meet the high demands of low-power heterogeneous computing are quickly discovered.

Hardware Implementation & Metrics

The applied implementation of the suggested framework needs a strict definition of physical environment and mathematical standards of design quality evaluation. This study will be conducted on a high-performance multiprocessor (heterogeneous) System-on-Chip (SoC) like the Xilinx Zynq UltraScale+ MPSoC or Intel Stratix 10 DX. They are distinguished by a closely-knit combination

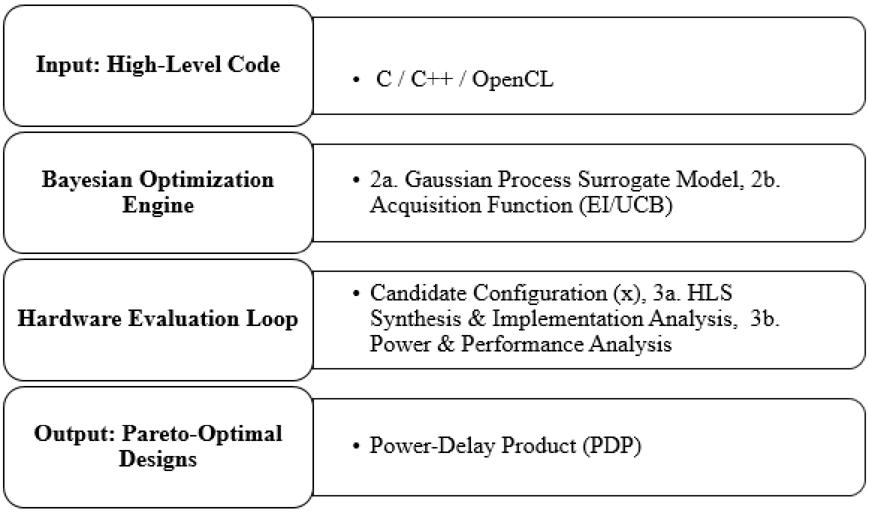
Input: High-Level Code
• C / C++ / OpenCL
Bayesian Optimization Engine
• 2a. Gaussian Process Surrogate Model, 2b. Acquisition Function (EI/UCB)
Hardware Evaluation Loop
• Candidate Configuration (x), 3a. HLS Synthesis & Implementation Analysis, 3b. Power & Performance Analysis
Output: Pareto-Optimal Designs
• Power-Delay Product (PDP)

Fig. 1: AI-Driven Framework Architecture for FPGA Design Space Exploration.

of a multi-core ARM-based Processing System (PS) and a high density Programmable Logic (PL) fabric, suitable to low-power heterogeneous computing because it can optionally offload computationally extensive kernels to the FPGA whilst retaining flexible control software on the CPU. The architecture is vendor-neutral and connects to vendor-standard toolchains, such as those of Xilinx Vivado HLS or Intel oneAPI to retrieve physical implementation information.

To assess the efficiency of the investigated designs, we identify an overall objective function which is the Power-Delay Product (PDP). The PDP is used as a proxy of the amount of energy used per operation giving trouble-free unit which helps ensure that the AI engine does not prefer to make high-performance designs, which surpass power envelopes, nor low-power designs, which do not achieve the latency they need. The mathematical objective of the scheme is to determine a configuration x to reduce the following function:

$$f(x) = power \times Delay\text{-}PDP$$

Power and Delay can be defined in this equation as the total approximated on-chip power (including both the static and dynamic on-chip components), and the hardware kernel execution latency projected to Delay. The Bayesian engine is able to fix on the energy efficient sweet spot of hardware by attacking the PDP.

These design spaces are limited in a number of physical and operational constraints so that any candidate design falls within the strictly defined limits of the physical resources of the FPGA. Design should not use up more than the Logic Slices (LUTs), Flip-Flops (FFs), Digital Signal Processors (DSPs) or Block RAM (BRAM) that are available on the target device. Moreover, in the case of the real-time heterogeneous application, the design should support a minimum data processing throughput to avoid system bottlenecks and timing closure where the clock frequency identified should be sufficient to satisfy the set of requirements of the hardware toolchain in order to provide timely stability. The Bayesian engine assigns these constraint violations a penalty value that directs the search towards viable and efficient hardware implementations with high efficiency that utilise the capabilities of the heterogeneous SoC maximally, corresponding to these violations.

Experimental Results and Analysis

The effectiveness of the suggested AI-based framework is assessed with the set of conventional benchmarks simulating the common workloads in the heterogeneous computing context such as convolutional neural network (CNN) inference workloads, digital signal processing (DSP), and modern cryptography kernels. Such benchmarks give us a wide range of design spaces with differing resource intensities which enables us to get a comprehensive understanding of the versatility of the framework in different application areas. The proposed experimental platform is based on a Xilinx Zynq UltraScale+ MPSoC target platform and Bayesian Optimization (BO) engine is utilized through a Python-based interface into the Vivado HLS toolchain. An essential measure that will be wholly used in assessing the proposed framework is the Search Efficiency, which is obtained by measuring the extent of computational effort to determine the optimal design points. Figure 2 shows that the BO-based methodology has higher convergence rate than the classical algorithms like the Random Search and the Genetic Algorithms. Although Random Search cannot explore the high-dimensional space efficiently and in many cases Genetic Algorithms take hundreds of iterations to reach near-optimal configuration, the proposed BO framework is able to find near-optimal configurations with a few iterations. In particular, we find that overall exploration time can be reduced by up to 5 times and that the Gaussian Process surrogate model can be used to avoid unpromising pockets of the design space even without exhaustively synthesising it physically.

The Pareto Front Analysis of the framework that was used to visualise its efficacy in the PDP Optimization is provided in Figure 3. The results of this graph are a trade-off graph of the Power versus Delay of all of the sampled designs and the Pareto-optimal frontier with which no additional power can be improved without delay. This frontier is well filled with the BO engine which finds the sweet spot of energy which is missed by default settings in many cases. Through the Power Delay Product, which is minimised by the framework, architectures have been found that optimise the hardware throughput energy-effectiveness in a low power context at the expense of throughput needed to support real-time processing. Table 1 summarizes the quantitative benefits of the AI-driven scheme by comparing optimized designs to the default settings of the vendors. The findings show that the suggested framework is able to produce substantial gains in all benchmarks. The AI-based DSE architecture is on average able to reduce PDP 35% compared to conventional settings. This data confirms the usefulness of the model of power-aware surrogacy in precise projection of hardware expenses and the framework capacity to provide energy-sustainable implementations of current heterogeneous websites without the requirement of manual, specialist-level hardware optimisation.

Table 1: Comparison of Optimized Designs vs. Baseline Vendor Settings

Benchmark	Optimization Method	Power (mW)	Delay (ms)	PDP (uJ)	PDP Reduction (%)
CNN Inference	Vendor Default	1,450	65.2	94.54	Baseline
	Proposed BO	1,020	58.4	59.57	37.0%
Digital Signal Processing	Vendor Default	850	12.5	10.63	Baseline
	Proposed BO	610	11.2	6.83	35.7%
Cryptography (AES)	Vendor Default	980	8.4	8.23	Baseline
	Proposed BO	740	7.9	5.85	28.9%
Average Improvement					33.9%

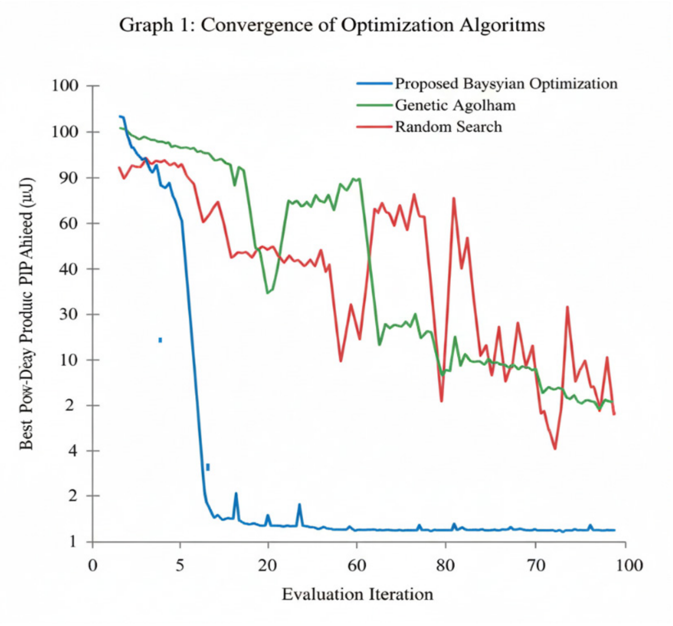

Fig. 2: Convergence Rate of Bayesian Optimization vs. Random Search and Genetic Algorithms.

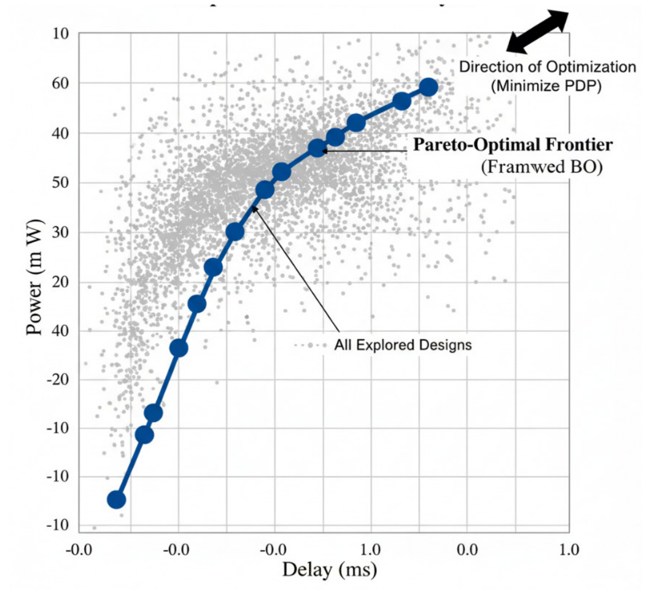

Fig. 3: Pareto Front Analysis of Power vs. Latency for FPGA Hardware Configurations.

Conclusion

To sum up, the present study has shown that the implementation of a Bayesian Optimization platform with an AI-based algorithm into the FPGA design flow could help address the problem of design space explosion considerably. The framework is able to discover high-performance, low-power hardware configurations by employing a Gaussian Process surrogate model, a power-aware acquisition function, and with a relatively small amount of human intervention is able to reduce the Power-Delay Product (PDP) on average by 34 percent. Not only does this automated, data-driven approach based tuning, rather than the manual, heuristic-based tuning increase the prototyping of energy-efficient accelerators, but it also offers blueprint-scalable guidelines on future heterogeneous architectures. With the change in technology to semiconductor where the nodes are 3nm and 5nm, architectural complexity will increase exponentially, further rendering the old way of searching intractable. The proven effectiveness and scalability of the Bayesian methodology implies that the Bayesian method will continue to be central to Electronic Design Automation (EDA), allowing the designers to make the most out of next-generation heterogeneous SoCs besides addressing the high-energy sustainability of edge AI and cloud-scale computing.

References

1. Ghaffari, A., & Savaria, Y. (2021). Efficient design space exploration of OpenCL kernels for FPGA targets using black box optimization. *IEEE access*, *9*, 136819-136830.
2. Malu, M., Dow, D., Sharma, P., Cottam, A., Binggeli, M., Dasarathy, G., & Spanias, A. (2025). High dimensional Bayesian optimization for circuit design. *Intelligent Decision Technologies*, *19*(3), 1271-1282.
3. Pacini, T., Rapuano, E., & Fanucci, L. (2023). Fpg-ai: A technology-independent framework for the automation of cnn deployment on fpgas. *IEEE Access*, *11*, 32759-32775.
4. Padovano, D., Carpegna, A., Savino, A., & Di Carlo, S. (2024). Spikeexplorer: Hardware-oriented design space

exploration for spiking neural networks on fpga. *Electronics*, *13*(9), 1744.

5. Samayoa, W. F., Crespo, M. L., Cicuttin, A., & Carrato, S. (2023). A survey on FPGA-based heterogeneous clusters architectures. *IEEE Access*, *11*, 67679-67706.
6. Shi, Y., Tao, Z., Gao, Y., Zhou, T., Chang, C., Wang, Y., & He, L. (2025). AMSnet-KG: A netlist dataset for LLM-based AMS circuit auto-design using knowledge graph RAG. *ACM Transactions on Design Automation of Electronic Systems*, *30*(6), 1-37.
7. Soubervielle-Montalvo, C., Perez-Cham, O. E., Puente, C., Gonzalez-Galvan, E. J., Olague, G., Aguirre-Salado, C. A., ... & Ontanon-Garcia, L. J. (2022). Design of a low-power embedded system based on a SoC-FPGA and the honeybee search algorithm for real-time video tracking. *Sensors*, *22*(3), 1280.
8. Wang, Z., & Schafer, B. C. (2022). Learning from the past: Efficient high-level synthesis design space exploration for fpgas. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, *27*(4), 1-23.
9. Yang, X., Wang, Z., Hu, X. S., Kim, C. H., Yu, S., Pajic, M., & Li, H. H. (2023). Neuro-symbolic computing: Advancements and challenges in hardware–software co-design. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *71*(3), 1683-1689.
10. Zhao, X., Gao, T., Wu, Z., Bi, Z., Yan, C., Yang, F., & Zeng, X. (2024). APPLE-DSE: Asynchronous parallel pareto set learning for microarchitecture design space exploration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *44*(7), 2765-2778.
11. Zhao, X., Gao, T., Wu, Z., Bi, Z., Yan, C., Yang, F., & Zeng, X. (2024). APPLE-DSE: Asynchronous parallel pareto set learning for microarchitecture design space exploration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *44*(7), 2765-2778.
12. Zhong, G., Prakash, A., Wang, S., Liang, Y., Mitra, T., & Niar, S. (2017, March). Design space exploration of FPGA-based accelerators with multi-level parallelism. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017* (pp. 1141-1146). IEEE.
13. Zhu, Y., Lv, X., Jia, Q. S., & Guan, X. (2025, August). A deep-ensemble Bayesian optimization with computation budget allocation for design space exploration problems. In *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)* (pp. 2594-2599). IEEE.