SCIENCE CHINA
Information Sciences

• MOOP •

December 2025, Vol. 68, Iss. 12, 224301:1–224301:3
https://doi.org/10.1007/s11432-025-4630-7

Toward mobile communication baseband circuit auto-design: a Bayesian model approach

Chuan ZHANG[1,2*], Changhan LI[1,2], Yunwei MAO[1,2], Yuwei ZENG[1,2], Tao HAN[1,2], You YOU[1,2], Yongming HUANG[1,2] & Xiaohu YOU[1,2*]

[1]*Laboratory of Efficient Architectures for Digital-communication and Signal-processing (LEADS), National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China*
[2]*Purple Mountain Laboratories, Nanjing 211111, China*

Received 4 April 2025/Revised 9 August 2025/Accepted 16 October 2025/Published online 19 November 2025

Citation Zhang C, Li C H, Mao Y W, et al. Toward mobile communication baseband circuit auto-design: a Bayesian model approach. Sci China Inf Sci, 2025, 68(12): 224301, https://doi.org/10.1007/s11432-025-4630-7

Mobile communication networks represent the next-generation ubiquitous, intelligent, and converged information infrastructure. To meet exponentially evolving demands, mobile networks must provide different capabilities spanning four key dimensions: enhanced bandwidth scalability, massive machine-type connectivity, ultra-reliable network stability, and deterministic ultra-low latency—requiring comprehensive cross-scenario adaptability [1]. Baseband (BB) circuits, serving as the central component in both base stations and end-user devices, represent the most technology-dense domain of mobile communication systems. Conventional development approaches rely on fragmented processes requiring specialized expertise, heavy R&D investment, and prolonged timelines, struggling to enable rapid customization capabilities. Therefore, forging an alternative pathway, this research proposes a Bayesian model-driven methodology for BB circuit auto-design—a paradigm shift in wireless system design automation.

Algorithm unification: A Bayesian model-based unified framework has been developed for heterogeneous BB processing modules, enabling cross-module computational consistency. As shown in Figure 1(a), it is observed that different BB modules share a unified system model with

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{y}$ is the received (RX) signal, $\mathbf{H}$ and $\mathbf{x}$ are transformation matrix and transmitted (TX) signal, and $\mathbf{n}$ is the system noise. Therefore, solutions of these BB modules can be derived from a unified Bayesian expression as

$$\begin{cases} R_{ij}(\mu_k) = \max\limits_{\mathbf{x}:x_j=\mu_k} (\log_{10}(\mathbf{F}_i(\mathbf{x})) + \sum\limits_{l\neq j} Q_{li}) \\ \qquad\qquad - \max\limits_{\mathbf{x}:x_j=\mu_0} (\log_{10}(\mathbf{F}_i(\mathbf{x})) + \sum\limits_{l\neq j} Q_{li}), \\ Q_{ji}(\mu_k) = \sum\limits_{l\neq i} R_{lj}(\mu_k), \end{cases} \tag{2}$$

where R_{ij} and Q_{ij} represent the messages sent from different directions in each module, and μ_k is the k-th possible entry of x_j [2]. The configurable local function $\mathbf{F}(\mathbf{x})$ bridges unified Bayesian BB algorithms and hardware architectures.

Architecture standardization: Building upon this Bayesian framework, a homogeneous reconfigurable architecture has been established for BB circuit implementations, achieving high-level hardware reuse across functional blocks. The architecture implements message-updating solutions for different BB modules (including channel estimation, MIMO detection, channel decoding, NOMA detection, and FFT/iFFT) by configuring the processing elements (PEs) corresponding to the local functions [3].

Automated design: Most critically, an intelligent design automation methodology has been pioneered through architectural homogenization, demonstrating considerable acceleration in BB system development cycles. We represent BB modules as parametric formulas [4]:

$$C(\mathcal{P}) = \textstyle\sum_{\mathcal{P}_i\in D(\mathcal{P})} \left(\prod_{j\in\mathcal{S}^i} s_j(M_j^1, M_j^2, \ldots, M_j^{n_j})\right) \cdot \delta(\mathcal{P}=\mathcal{P}_i), \tag{3}$$

where $C(\mathcal{P})$ represents an Auto-Generator circuit template of parameters $\mathcal{P}$, $\mathcal{P}_i$ and $D(\mathcal{P})$ denote a parameter set and its domain of definition, s_j and $\mathcal{S}^i$ stand for the j-th sub-module and all sub-modules under parameter set $\mathcal{P}_i$, $\prod$ represents the cascading of sub-modules, M_j is the basic module in the circuit library, and $\delta(\cdot)$ is the Dirac delta function. With the formulaic representation, we construct the parametric design space, enabling automatic design space exploration towards the optimal BB circuit.

The proposed approach involves three distinct yet collaborative steps: (1) Auto-Generation, (2) Auto-Estimation, and (3) Auto-Optimization—collectively forming the Auto^3 framework that enables intelligent self-adaptation with order-of-magnitude reduction in manual interventions. Auto-Generation constructs a parametric BB circuit design space, enabling auto-estimation to rapidly evaluate each candidate solution, which then guides auto-optimization to identify optimal multi-objective trade-offs.

* Corresponding author (email: chzhang@seu.edu.cn, xhyu@seu.edu.cn)

© Science China Press 2025

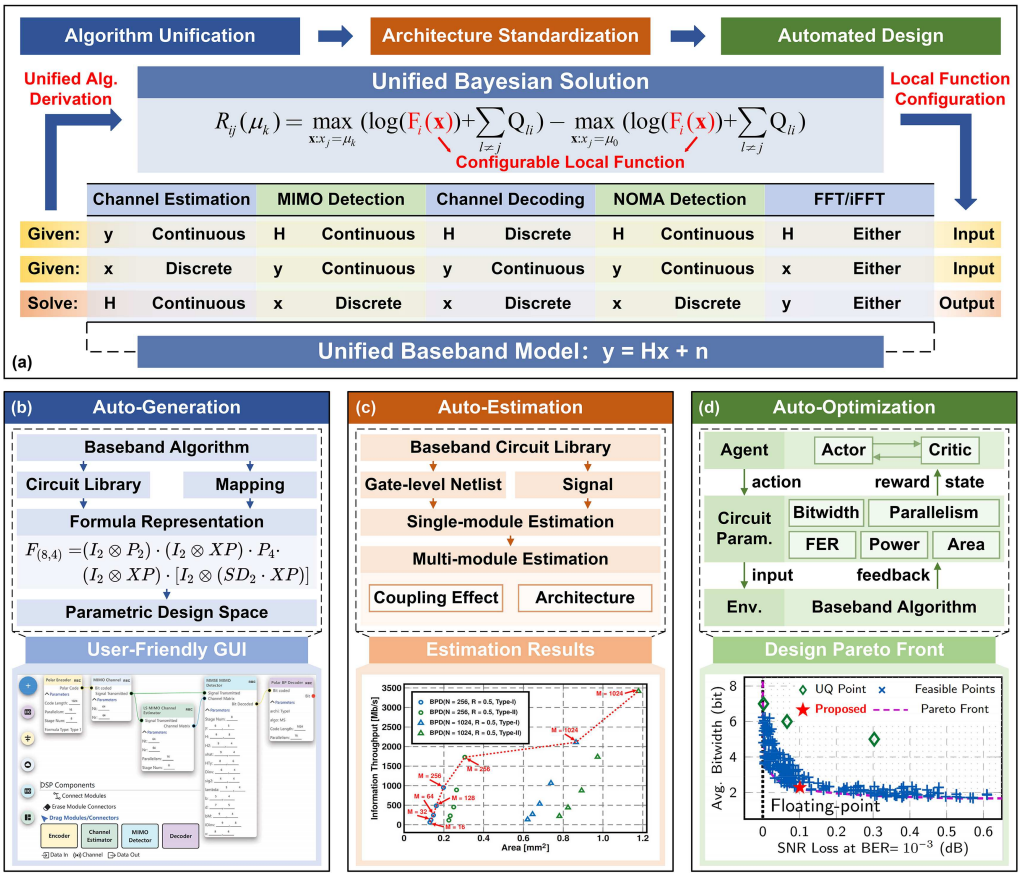

Figure 1 (Color Online) (a) Proposed Auto-Design framework; (b) Auto-Generation workflow and user-friendly GUI; (c) Auto-Estimation workflow and estimation results; (d) Auto-Optimization workflow and design Pareto front.

Auto-Generation. Based on the unified Bayesian model, we are able to construct a parametric design space without compromising flexibility. As shown in Figure 1(b), our approach comprises three key steps.

- Developing a low-level circuit library for Bayesian BB signal processing.
- Expressing BB circuits as non-parametric formulas based on the circuit library.
- Mapping BB algorithms to parametric formulas.

A low-level circuit library tailored for Bayesian BB signal processing serves as a base for Auto-Generation. We design circuits for the shared basic operations in the unified Bayesian expression as part of the low-level circuit library. The library aims to cover all necessary components to construct the calculation cores of BB acceleration circuits.

Non-parametric formulas are derived for BB circuits as intermediate representations. Based on the circuit library, complex BB circuits are expressed as inter-connected modules with formulaic representations. Using the established mapping rules between formulas and circuits, we developed a register-transfer level (RTL) code generator for circuit formulas. Therefore, the feasible design space for a specified Bayesian BB algorithm is depicted by mapping the algorithm to different formulas.

High-level mapping methods enable efficient construction of parametric design space. To facilitate high-level mapping, general design assumptions are taken, for instance, using spatial arrays for regular matrix operations, and using piecewise polynomial approximation for non-linear operations. Under such assumptions, mapping methods including space-time transformation (STT) for systolic arrays are adopted. Using the mapping methods, we develop parametric formulas for high-level arithmetic, where different parameter sets reduce it to different non-parametric formulas. Following the above scheme, we map Bayesian BB algorithms to parametric formulas. The design space is therefore given by the domain of definition for formula parameters.

Ji et al. [5] developed an Auto-Generator for belief propagation (BP) polar decoders, the parametric formula is

$$\begin{aligned} \mathrm{BP}D(N,M) = \mathrm{BCB}_0^{\otimes} \cdot \prod_{k=1}^{n-2} \left[W_k^{\otimes} \cdot \mathrm{BCB}_1^{\otimes} \right] \cdot P_M \\ \times \mathrm{BCB}_0^{\otimes} \cdot P_M^r \cdot \prod_{k=1}^{n-2} \left[\mathrm{BCB}_1^{\otimes} \cdot T_k^{\otimes} \right], \end{aligned} \quad (4)$$

where N is code length, M is parallelism, $\mathrm{BCB}_0^{\otimes}$ and $\mathrm{BCB}_1^{\otimes}$ represent two types of duplicates of basic computational blocks, $W_k^{\otimes}$ and $T_k^{\otimes}$ denote duplicates of different permutation blocks, and P_M and P_M^r denote permutation blocks. The auto-generator can generate RTL codes within seconds. With design space exploration, Ref. [5] achieved better energy- and area-efficiency than prior art.

Following the proposed approach, we have developed auto-generators for various BB modules. By configuring fundamental parameters, corresponding RTL codes can be generated within one second. As shown in Figure 1(b), we have also developed a user-friendly graphical user interface (GUI)

for these auto-generators, allowing users to easily configure parameters and generate RTL codes.

Auto-Estimation. Based on Auto-Generation, Auto-Estimation enables fast and accurate prediction of key metrics for circuits. Auto-Estimation comprises two key steps.

- Building prediction models for modules in the library.
- Conducting a multi-module estimation for hardware.

Single-module estimation serves as the foundation for large-scale circuit analysis. As depicted in Figure 1(c), to construct prediction models for each module in the library, we proposed an abstract syntax tree (AST) explorer for Verilog hardware description language (HDL) to facilitate the feature extraction of gate-level netlist and signal. Subsequently, machine learning was utilized to train prediction models based on these features and synthesized PPA metrics. When estimating the entire circuit, we decompose the overall architecture into submodules and obtain the hardware parameters of each submodule. Pre-trained models are then employed to predict the key metrics of submodules.

A multi-module estimation approach for the entire circuit enables a more accurate mapping of hardware parameters to key performance metrics. The approach considers both the performance metrics of each submodule and the coupling effects and hardware characteristics that collectively impact the overall performance. To capture the hardware characteristics, we employed an enhanced control and data flow graph to analyze the temporal behavior of submodules and their interconnect relationships. To mitigate the impact of coupling effects, we proposed a novel graph neural network (GNN) to extract submodule coupling features from the hardware architecture. By accounting for the performance metrics of each submodule and the coupling effects, we can provide a reliable prediction of the entire circuit performance.

In previous research, Zhong et al. [6] analyzed the submodules in proposed polar code encoders and estimated the complexity of their hardware implementations. Encoder type I comprises $M \times \log_2 \frac{N}{\sqrt{M}}$ XOR gates, $N - M$ D flip-flops, and $M \times \log_2 N/M$ 2-to-1 multiplexers (MUXs), where N is the code length and M signifies parallelism. Encoder type II includes $M/2 \times \log_2 N$ XOR gates, $N - M$ D flip-flops, and $M \times \log_2 N/M$ 2-to-1 MUXs. Encoder type III integrates $\log_2 M$ XOR gates, N D flip-flops, and N 2-to-1 MUXs. The analysis established the relationship between performance metrics and hardware parameters.

Auto-Optimization. After identifying the solution space, Auto-Optimization aims to fine-tune circuit parameters, achieving optimal trade-offs among key performance metrics. To achieve automated optimization, it is imperative first to identify the specific parameters within the hardware modules that require optimization. Based on the mapping obtained by Auto-Estimation, it is feasible to further construct the constraints in Auto-Optimization straightforwardly according to performance demands.

Deep reinforcement learning (DRL) is a promising technique to automatically specialize parameters for different modules. Figure 1(d) presents a framework of Auto-Optimization based on DRL. The Pareto front represents the set of optimal solutions that cannot be improved in one objective without degrading another objective. Feasible points are those that satisfy the constraints of the optimization problem. In performance-constrained scenarios, the reward function incorporates performance loss (e.g., frame error rate) as a hard constraint, defined as

$$r_t = \begin{cases} \theta_1 \exp(-\theta_2 \sum_i w_i f(v_i)), & \Delta\ell \leqslant \delta, \\ -\rho, & \Delta\ell > \delta, \end{cases} \tag{5}$$

where $\Delta\ell$ is performance loss, δ is the predefined threshold, θ_1 and θ_2 are hyper-parameters controlling the reward shape, w_i is the weight for each optimized variable v_i, $f(\cdot)$ is the estimation function obtained in Auto-Estimation, and ρ is the penalty coefficient. Given that BB digital signal processing (DSP) algorithms are sensitive to quantization noise, the action space $\mathcal{A}$ is restricted to $\{a_t | -L_a \leqslant a_t \leqslant L_a\}$ to prevent divergence, where L_a is the maximum action. This bounded action space helps to reduce the risk of severe performance degradation. The state space incorporates the embeddings of both module and agent information, including variable index i, element size $size_i$, and last action a_{t-1}.

As shown in Figure 1(d), Ge et al. [7] achieved auto-optimization for quantization policies and attained an optimal trade-off between average bitwidth and BER performance loss in MIMO detectors. Compared to traditional unified quantization (UQ), the same detector utilizing the method in [7] demonstrates 2.97× area efficiency and 1.24× energy efficiency with 17.92 Gb/s peak throughput.

Conclusion. This work introduces a Bayesian model-driven automated design methodology for BB circuits, which is different from the existing high-level synthesis (HLS) [8] approach, by establishing an algorithm-architecture co-optimization framework. It achieves 6.15× higher design efficiency than conventional template-based approaches. Experimental validation demonstrates 1.24× energy efficiency and 1.70× throughput gains over state-of-the-art manual implementations in 6G scenarios. Initial deployment in distributed MIMO systems reveals promising potential for automated BB design. The framework's extensibility spans diverse DSP domains, including adaptive filters, machine learning accelerators, and autonomous driving systems through its unified Auto^3 framework.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62331009, 62401642), Natural Science Foundation of Jiangsu Province (Grant Nos. BG2024004, BM2023016), and Fundamental Research Funds for the Central Universities.

Supporting information Videos and other supplemental documents. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 ITU-R. Framework and overall objectives of the future development of IMP for 2030 and beyond. M.2160. https://www.itu.int/rec/R-REC-M.2160/en
2 Zhang Y, Zhou W, Zhang Y, et al. BayesBB: a 9.6-Gb/s 1.61-ms configurable all-message-passing baseband-accelerator for B5G/6G cell-free massive-MIMO systems. IEEE J Solid-State Circuits, 2025, doi: 10.1109/JSSC.2025.3571647
3 Ji H, Zhang Y, Sun Y, et al. UniDec: a unified factor-graph-based decoder fully compatible with 5G NR LDPC/Polar codes. IEEE Trans Circuits Syst I, 2025, 72: 4235–4247
4 Mao Y W, You Y, Tan X S, et al. FLAG: formula-LLM-based auto-generator for baseband hardware. In: Proceedings of International Symposium on Circuits and Systems (ISCAS), Singapore, 2024. 1–5
5 Ji C, Shen Y, Zhang Z, et al. Autogeneration of pipelined belief propagation polar decoders. IEEE Trans VLSI Syst, 2020, 28: 1703–1716
6 Zhong Z W, Gross W J, Zhang Z C, et al. Polar compiler: auto-generator of hardware architectures for polar encoders. IEEE Trans Circuits Syst I, 2020, 67: 2091–2102
7 Ge Y, Ji Z, Huang Y, et al. Automatic hybrid-precision quantization for MIMO detectors. IEEE Trans Signal Process, 2023, 71: 1039–1052
8 Coussy P, Gajski D D, Meredith M, et al. An introduction to high-level synthesis. IEEE Des Test Comput, 2009, 26: 8–17