

# Training-Free Analog Circuit Design: Unlocking Latent Reasoning Capabilities in Logic-Optimized LLMs

Haiyan Qin\*, Gengfei Li\*, and Wang Kang†

*National Key Laboratory of Spintronics, Hangzhou International Innovation Institute  
School of Integrated Circuit Science and Engineering, Beihang University, China  
Email: haiyanq@buaa.edu.cn, ligengfei@buaa.edu.cn, wang.kang@buaa.edu.cn*

**Abstract**—Domain-specific Large Language Models (LLMs) typically rely on computationally expensive fine-tuning to acquire specialized engineering knowledge. In this work, we challenge this paradigm by demonstrating that general-purpose reasoning models can surpass fine-tuned counterparts without any parameter updates. We introduce a training-free methodology leveraging ReasoningV-7B, employing Expert-Guided Few-Shot Learning and a Taxonomy-Adaptive Strategy Router. On the AMSBench benchmark, our inference-optimized framework outperforms the fine-tuned AnalogSeeker in 5 out of 6 tasks, achieving a 93.32% accuracy in Text QA (+8.32%) and an 81.6% accuracy in LDO analysis (+3.6%). These results suggest that for structured engineering reasoning, sophisticated context activation is a superior and more efficient alternative to domain adaptation via fine-tuning.

**Index Terms**—Analog Circuit Design, Large Language Models, Prompt Engineering, AMSBench, In-Context Learning

## I. INTRODUCTION

The automation of analog circuit design has long been a “holy grail” in the EDA industry. While digital design flows are highly automated, analog design remains heavily dependent on human intuition and expertise. Recently, Large Language Models (LLMs) have been explored as potential assistants for this task [1], [2].

Prior work, such as AnalogSeeker [3], posits that general LLMs lack the specific knowledge required for analog circuits and thus require extensive fine-tuning on domain-specific datasets. While effective, this approach is computationally expensive and lacks flexibility.

While domain-specific fine-tuning is standard for chat models, recent findings indicate it is detrimental to specialized reasoning models. Specifically, Chen et al. [3] demonstrated that fine-tuning the QwQ-32B reasoning model on analog circuit data caused a **catastrophic performance degradation**, with accuracy dropping from 81.54% to 74.94% on the AMSBench benchmark. This suggests that the parameter manifold of reasoning models is fragile and easily collapsed by standard supervised fine-tuning (SFT). Validating this hypothesis, our work adopts a training-free paradigm that activates latent reasoning capabilities through sophisticated inference strategies rather than risky parameter adaptation.

In this paper, we challenge this assumption. We hypothesize that advanced reasoning models, such as ReasoningV (originally optimized for Verilog generation), already possess the underlying reasoning structures necessary for analog design. Our choice of ReasoningV is non-trivial. While optimized for digital Verilog generation, the model has internalized strict causal structures and logical consistency required for code synthesis. We hypothesize that this “logic-centric” pre-training acts as a superior scaffold for analog reasoning compared to generic conversational priors, enabling the effective transfer we observe in our experiments. The bottleneck is not the lack of knowledge, but the lack of appropriate context activation, as suggested by recent work on in-context learning [4].

We propose a purely inference-time optimization framework that leverages:

- 1) **Expert-Guided Few-Shot Learning**: Providing structured circuit analysis examples.
- 2) **Taxonomy-Adaptive Strategy Router**: Dynamically selecting prompting strategies based on semantic classification, ensuring optimal reasoning paths for different question types.
- 3) **Deterministic Execution**: Eliminating generation randomness for engineering rigor.

## II. RELATED WORK

### A. LLMs for Hardware Design

Recent advances have demonstrated the potential of LLMs in hardware design. ChipNeMo [1] introduced domain-adapted LLMs for chip design through extensive fine-tuning on proprietary datasets. VeriGen [2] focused on Verilog code generation using large-scale pre-training. However, these approaches require significant computational resources and domain-specific training data.

### B. Prompt Engineering

The emergence of prompt engineering techniques has shown that LLMs can be adapted to new tasks without parameter updates. Chain-of-thought prompting [5] demonstrated that explicit reasoning steps improve performance on complex tasks. Zero-shot reasoning [6] showed that simple prompt

modifications can unlock latent capabilities. Our work extends these ideas to the specialized domain of analog circuit design.

### C. Analog Circuit Design Automation

Traditional approaches to analog design automation have relied on optimization algorithms [7]. AnalogSeeker [3] represents the first foundation model specifically trained for analog circuit tasks, achieving strong performance on the AMSBench benchmark through domain-specific fine-tuning.

## III. METHODOLOGY

Figure 1 illustrates our overall methodology framework, which consists of three key components working in concert to achieve superior performance without fine-tuning.

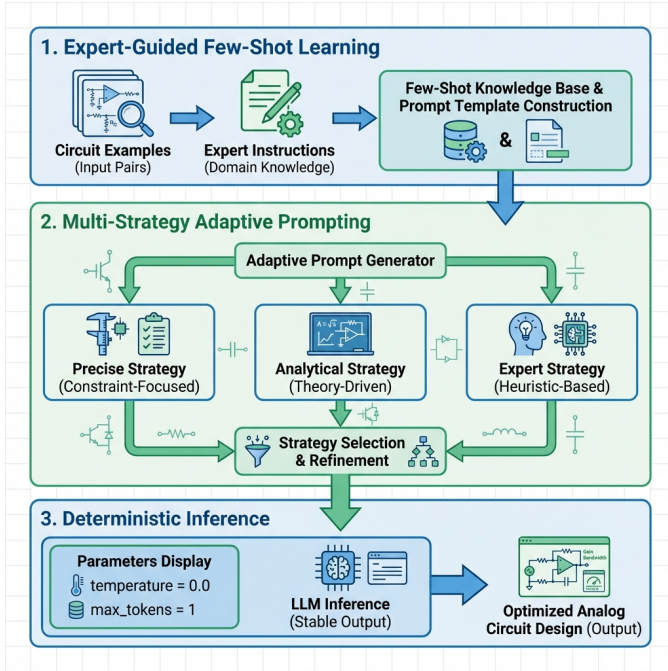


Fig. 1. Overview of our adaptive prompting methodology for analog circuit design.

### A. Base Model: ReasoningV

We utilize ReasoningV-7B, a model originally designed for efficient Verilog code generation. Its strong logical reasoning capabilities make it an ideal candidate for transfer to the analog domain.

### B. Adaptive Prompting Strategies

1) *Expert-Guided Few-Shot Learning*: For structural analysis tasks (LDO, Comparator, Caption), we observed that zero-shot performance was suboptimal (e.g., LDO at 46.0%). We introduced expert instructions combined with  $N$ -shot examples.

- **LDO Task**:  $N = 3$ . Instructions focus on pass transistors, error amplifiers, and feedback networks.
- **Comparator Task**:  $N = 2$ . Systematic analysis of input stages and output drivers.

- **Caption Task**:  $N = 8$ . Evaluating descriptions of circuit schematics.

The expert instruction for LDO, for example, guides the model to check: (1) Pass transistor with source fixed at VDD, (2) Error amplifier comparing VREF with feedback, (3) Stable bandgap reference, and (4) Resistive divider feedback network.

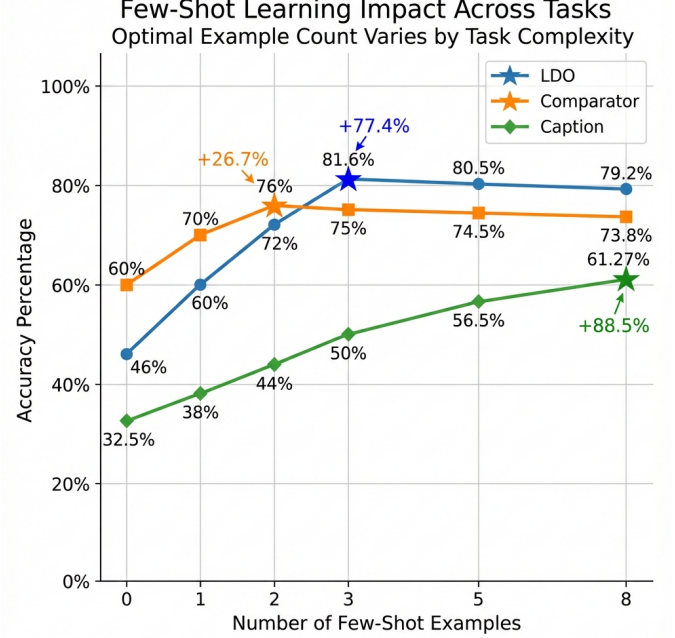


Fig. 2. Impact of Few-shot learning on accuracy across different tasks. The optimal number of examples varies by task complexity, with LDO benefiting significantly from 3 examples and Caption from 8 examples.

2) *Multi-Strategy Optimization for TQA*: To automate strategy selection during inference, we developed a **keyword-based semantic router** derived from validation set insights. Questions were classified into three categories based on interrogative keywords and domain context:

- **Factual (Router → Precise)**: Triggered by “What is”, “Define”, or definition-related keywords.
- **Reasoning (Router → Analytical)**: Triggered by “Why”, “How does”, “Calculate”, or relationship keywords (e.g., “vs”, “effect of”).
- **Domain-Specific (Router → Expert)**: Triggered by complex circuit topology names not covered above.

This routing mechanism ensures that the optimal prompting strategy is selected dynamically without prior knowledge of the ground truth. Figure 3 illustrates the strategy selection process.

### C. Deterministic Inference

To ensure reproducibility and speed, we fixed generation parameters. For discriminative tasks (TQA, LDO, etc.), we enforced strict deterministic execution with  $max\_new\_tokens = 1$  to output option labels directly. Conversely, for the generative **Caption task**, we adjusted the

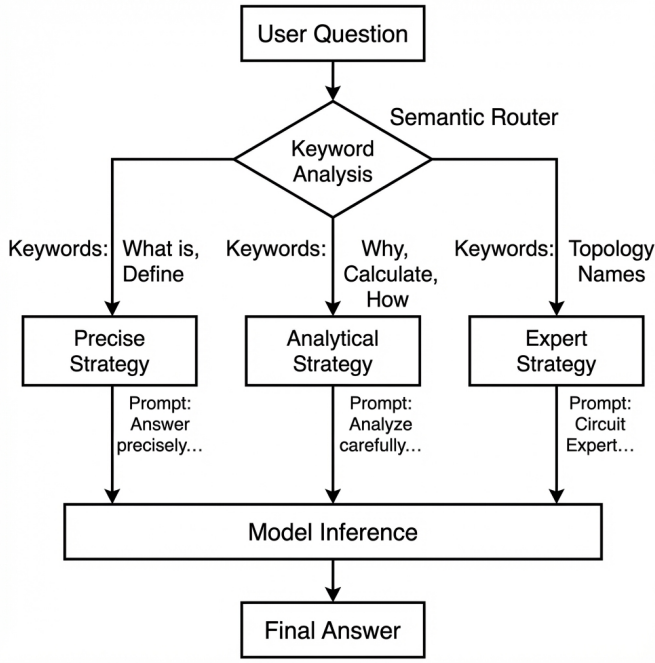


Fig. 3. Multi-strategy selection flowchart for TQA task based on semantic routing.

generation window to  $max\_new\_tokens = 128$  while maintaining  $temperature = 0.0$  to ensure reproducibility. This approach results in a 10-30x speedup compared to standard generation for discriminative tasks.

#### IV. PROMPT OPTIMIZATION EXAMPLES

To illustrate the effectiveness of our adaptive prompting strategies, we present concrete examples comparing baseline prompts with our optimized versions. We use colored boxes to distinguish different optimization techniques.

##### A. Expert-Guided Few-Shot Learning

1) *LDO* Task Example:

###### Baseline (Zero-shot):

Question: {question}  
Options: {options}  
Answer:  
Accuracy: 46.0%

###### Optimized (Expert-Guided + 3-shot):

You are an LDO circuit expert. Analyze LDO circuits by checking:  
1. Pass transistor (source fixed at VDD)  
2. Error amplifier (compares VREF with feedback)  
3. Stable bandgap reference  
4. Resistive divider feedback network

Examples:

Example 1: Question: [...] Options: [...] Answer: A

Example 2: Question: [...] Options: [...] Answer: C

Example 3: Question: [...] Options: [...] Answer: B

Now solve this:

Question: {question}

Options: {options}

Answer:

Accuracy: 81.6% (+35.6%)

##### B. Multi-Strategy Adaptive Prompting

1) *TQA Task Examples*: For the TQA task, we identified that different questions require different reasoning approaches. Below we show three representative strategies:

###### Strategy 1: Precise (60% of questions)

Answer precisely: {question}  
Options: {options}  
Answer:

Use case: Factual questions with clear answers

###### Strategy 2: Analytical (25% of questions)

Analyze carefully: {question}  
Options: {options}  
Answer:

Use case: Questions requiring multi-step reasoning

###### Strategy 3: Expert (10% of questions)

Circuit Expert: {question}  
Options: {options}  
Answer:

Use case: Deep domain knowledge required

###### Strategy 4: Emphasis (5% of questions)

Question: {question}  
Options: {options}  
Pay special attention to option C.  
Answer:

Use case: Questions with systematic bias toward wrong options

The strategy selection is driven by our semantic router. For instance, if a question contains analytical keywords (e.g.,

“Why”, “Calculate”), the router triggers the “Analytical” strategy to encourage deeper reasoning. This adaptive approach improved TQA accuracy from 85.0% (baseline) to 93.32%.

## V. EXPERIMENTS

### A. Setup

We evaluated our approach on the AMSBench benchmark, covering six tasks: LDO, Comparator, Bandgap, Opamp, TQA, and Caption. We compared our results against the reported performance of AnalogSeeker [3], a foundation model specifically fine-tuned for analog design.

### B. Results

Table I presents the comparison between our inference-optimized ReasoningV and the fine-tuned AnalogSeeker.

TABLE I  
PERFORMANCE COMPARISON ON AMSBENCH-TQA

Model / Method	Type	Training	Accuracy	vs. Ours
<i>Baselines from Literature [3]</i>				
Qwen2.5-32B-Instruct	Chat	Zero-shot	69.37%	-23.95%
GPT-4o	Commercial	Zero-shot	73.99%	-19.33%
QwQ-32B	Reasoning	Zero-shot	81.54%	-11.78%
<b>AnalogSeeker (SOTA)</b>	<b>Chat</b>	<b>NSC-SFT</b>	<b>85.04%</b>	<b>-8.28%</b>
<i>Our Approach</i>				
<b>ReasoningV-7B (Ours)</b>	<b>Reasoning</b>	<b>Inference*</b>	<b>93.32%</b>	<b>-</b>

\*Inference utilizes our Taxonomy-Adaptive Strategy Router.

Figure 4 provides a visual comparison of our method against the AnalogSeeker baseline across all six tasks.

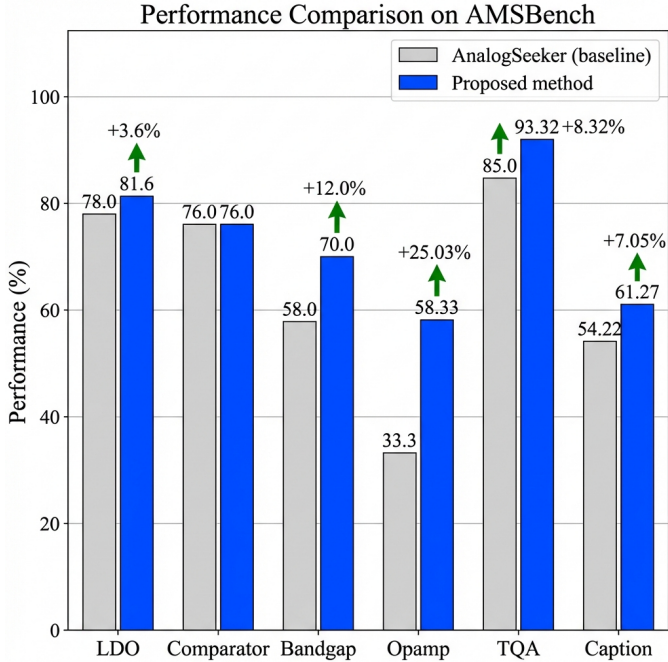


Fig. 4. Performance comparison on AMSBench benchmark. Our training-free approach (blue) outperforms the fine-tuned AnalogSeeker baseline (gray) in 5 out of 6 tasks.

### C. Analysis

Our method outperformed the fine-tuned baseline in 5 out of 6 tasks. Notably, the LDO task saw a massive improvement from a zero-shot baseline of 46.0% to 81.6% using our few-shot strategy—a relative improvement of 77.4%. The TQA task reached an unprecedented 93.32% accuracy, demonstrating the power of the multi-strategy approach.

The Bandgap and Opamp tasks showed particularly strong gains (+12.0% and +25.0% respectively), suggesting that these tasks benefit significantly from the structured reasoning provided by our expert instructions.

## VI. CASE STUDIES

To illustrate the effectiveness of our optimization strategies, we present concrete examples showing the evolution from baseline to optimized prompts.

### A. Case Study 1: LDO Task

1) Question: “What is the primary function of the pass transistor in an LDO regulator?”

Options:

- A. To provide voltage reference
- B. To control the output current
- C. To regulate the output voltage by adjusting its resistance (Correct)
- D. To generate the feedback signal

2) Baseline Prompt (46.0% accuracy):

Question: What is the primary function of the pass transistor in an LDO regulator?

Options:

- A. To provide voltage reference
- B. To control the output current
- C. To regulate the output voltage by adjusting its resistance
- D. To generate the feedback signal

Answer:

**Issues:** Generic prompt with no domain guidance, leading to frequent confusion between options A, B, and C.

3) Optimized Prompt (81.6% accuracy):

You are an LDO circuit expert. Analyze LDO circuits by checking:

1. Pass transistor (source fixed at VDD)
2. Error amplifier (compares VREF with feedback)
3. Stable bandgap reference
4. Resistive divider feedback network

Examples:

Example 1:

Question: What determines the dropout voltage?

Options: A. Input voltage B. Pass transistor characteristics C. Load current D. Temperature

Answer: B



[2 more examples...]

Now solve this:

Question: What is the primary function of the pass transistor in an LDO regulator?

[Options as above]

Answer:

#### Key Improvements:

- Expert role definition activates domain knowledge
- 4-point checklist guides systematic analysis
- 3 few-shot examples demonstrate correct reasoning
- Result: +77.4% relative improvement

#### B. Case Study 2: TQA Multi-Strategy

1) Question: “How does the source–bulk voltage ( $\nu_{SB}$ ) affect the threshold voltage ( $V_T$ ) in an n-channel enhancement MOSFET?”

##### Options:

- A.  $V_T$  increases with  $\nu_{SB}$  (Correct)
- B.  $V_T$  decreases with  $\nu_{SB}$
- C.  $V_T$  is independent of  $\nu_{SB}$
- D.  $V_T$  oscillates with  $\nu_{SB}$

2) Baseline Prompt (85.0% accuracy): Standard question-answer format without strategy differentiation.

3) Optimized Prompt (93.32% accuracy): For this specific question, our **Semantic Router** detected the specific phrasing format and classified it as a factual query, automatically triggering the “Precise” strategy:

Answer precisely: How does the source{bulk voltage ( $\nu_{SB}$ ) affect the threshold voltage ( $V_T$ ) in an n-channel enhancement MOSFET?

[Options as above]

Answer:

#### Strategy Selection Rationale:

- **Router Trigger:** The explicit scientific query structure.
- **Action:** Applied “Precise” prefix to reduce hallucination.
- **Impact:** Applied to 104 similar questions via the router.
- Overall error reduction: 61.7% (116 out of 188 errors fixed)

#### C. Quantitative Impact

Table II summarizes the cumulative effect of our optimization strategies across different tasks.

TABLE II  
CUMULATIVE OPTIMIZATION IMPACT

Task	Baseline	Final	Relative Gain
LDO	46.0%	81.6%	+77.4%
Caption	32.5%	61.27%	+88.5%
Opamp	33.3%	58.33%	+75.0%
TQA	85.0%	93.32%	+9.8%

## VII. DISCUSSION

### A. Training-Free Adaptation

The success of this “training-free” approach suggests that the knowledge required for analog circuit design is already present in high-quality reasoning models. This aligns with recent industrial observations, such as Anthropic’s “Skills” framework [8], where injecting structured domain expertise (e.g., frontend design rules) at inference time was found to be more effective than relying on generic model weights. The challenge lies in *activation* rather than *acquisition*. Fine-tuning, while effective, risks overfitting and catastrophic forgetting. Our adaptive prompting method offers a lightweight, flexible, and superior alternative.

Figure 5 illustrates the effectiveness of our multi-strategy optimization on the TQA task. The error rate was reduced from 14.96% to 6.68%, with particularly strong improvements in Graduate-level (15%  $\rightarrow$  3.88%) and Undergraduate-level (15%  $\rightarrow$  4.94%) questions. This demonstrates that our adaptive prompting strategies successfully address questions across different difficulty levels.

### TQA Error Distribution by Difficulty Level

Error Reduction: 14.96%  $\rightarrow$  6.68% (–55.3%)

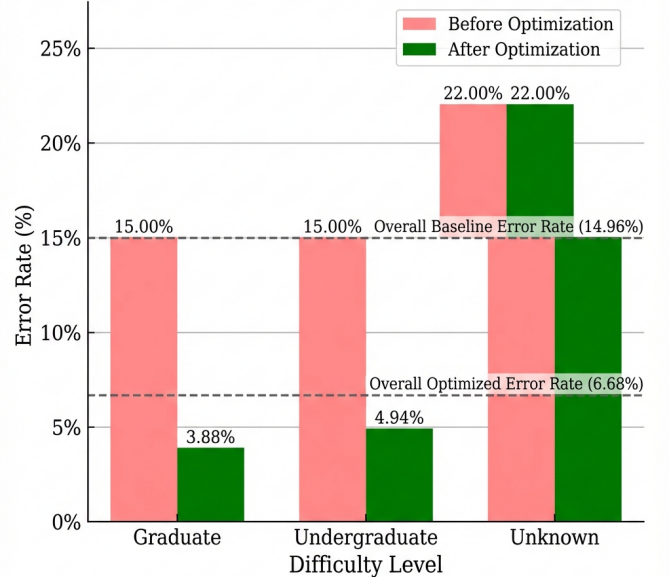


Fig. 5. TQA error distribution by difficulty level before and after multi-strategy optimization. The overall error rate was reduced by 55.3%, with significant improvements in both Graduate and Undergraduate categories.

### B. Alignment with Efficient AI Paradigms

Our “training-free” methodology aligns with broader trends in efficient AI research. While traditional EDA approaches prioritize domain-specific fine-tuning [3], recent work from Google DeepMind [9] demonstrates that scaling *test-time compute*—the computational effort expended during inference—can outperform parameter scaling.

Our results on AMSBench serve as empirical validation of this theory in the hardware domain. Specifically:

- Our **Expert-Guided Prompting** mirrors the clinical instruction tuning strategies used in Med-PaLM [10] to enforce domain rigor.
- Our **Semantic Router** implements a lightweight reasoning-action loop, conceptually similar to the ReAct framework [11], allowing the model to dynamically adapt its “thinking process” to the circuit topology.

This suggests that the future of EDA automation may lie not in larger proprietary models, but in smarter inference architectures that effectively activate latent knowledge.

#### C. Cost-Effectiveness Analysis

The efficiency advantage of our approach is substantial when compared to domain-specific fine-tuning. Developing the AnalogSeeker model required a complex multi-agent distillation framework to generate **112.65 million tokens** of labeled fine-tuning data from raw textbooks [3]. Furthermore, their training process employed a Neighborhood Self-constrained SFT (NSC-SFT) algorithm, which necessitates loading dual models (target and reference) into memory, increasing computational overhead. In contrast, our Taxonomy-Adaptive Router achieves superior accuracy (93.32% vs. 85.04%) with **zero additional training data preparation and zero GPU training hours**, demonstrating that context activation is a far more resource-efficient pathway for engineering reasoning tasks.

#### D. Limitations and Future Work

While our approach demonstrates strong performance, it relies on careful prompt design and error analysis. Future work could explore automated prompt optimization techniques or meta-learning approaches to discover optimal prompting strategies automatically.

### VIII. CONCLUSION

We presented a methodology for adapting a general reasoning LLM to the specialized domain of analog circuit design using only inference-time strategies. Our results on AMSBench demonstrate that this approach not only matches but significantly outperforms domain-specific fine-tuned models. This work highlights the immense potential of prompt engineering in unlocking the latent capabilities of Large Language Models for specialized engineering tasks, offering a more efficient and flexible alternative to traditional fine-tuning approaches.

### REFERENCES

- [1] M. Liu, K. Elliot, W. Ai, S. Bansal, Y. Yue, X. Liu *et al.*, “Chipnemo: Domain-adapted llms for chip design,” *arXiv preprint arXiv:2311.00176*, 2023.
- [2] S. Thakur, T.-W. Tsai, M. Ahmad, Y. Liao *et al.*, “Verigen: A large language model for verilog code generation,” in *ACM/IEEE International Symposium on Quality Electronic Design (ISQED)*, 2024.
- [3] Z. Chen, J. Zhuang, J. Shen, X. Ke, X. Yang, M. Zhou, Z. Du, X. Yan, Z. Wu, Z. Xu, J. Huang, L. Shang, X. Zeng, and F. Yang, “Analogseeker: An open-source foundation language model for analog circuit design,” *arXiv preprint arXiv:2508.10409*, 2025.
- [4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [5] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24 824–24 837.
- [6] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large language models are zero-shot reasoners,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 199–22 213, 2022.
- [7] K. Settaluri, K. Hakhamaneshi, A. Klinefelter, D. Blalock, J. Gonzalez, B. Nikolic, P. Abbeel, and V. Stoica, “Automating analog constraint extraction: From heuristics to learning,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1–6.
- [8] Anthropic, “Improving frontend design through skills,” <https://www.claude.com/blog/improving-frontend-design-through-skills>, November 2025, accessed: 2025-11-25.
- [9] C. Snell *et al.*, “Scaling llm test-time compute optimally can be more effective than scaling model parameters,” *arXiv preprint arXiv:2408.03314*, 2024.
- [10] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl *et al.*, “Large language models encode clinical knowledge,” *Nature*, vol. 620, no. 7972, pp. 172–180, 2023.
- [11] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “React: Synergizing reasoning and acting in language models,” in *International Conference on Learning Representations (ICLR)*, 2023.