



Analog Design Experiments With AI—Part 2

In the previous article [1], we assessed the analytical abilities of ChatGPT for basic circuits. In this article, we extend our effort toward circuit design by asking artificial intelligence (AI) to predict performance trends. As in the previous article, we give a score from zero to four to AI's answers.

The Analog Design Pyramid

To create a cohesive framework for our studies, we first ponder a general question: How can AI become a competent analog designer? We postulate that AI must climb a pyramid of five levels, as depicted in Figure 1. At the foundation, AI must analyze circuits correctly. As explained below, this ability appears to be maturing in some AI tools, especially if it simply requires writing and solving equations. The second level relates to intuition: Can AI understand a structure well enough to predict its behavior as we vary certain parameters? Possessed by good analog engineers, this key skill guides the design process, distinguishing those engineers from “SPICE monkeys.”

At the third level, AI must be able to design a given circuit topology according to certain specifications. We see below that the task of training AI for this purpose presents formidable challenges. The fourth level extends this ability to multiple topologies for the same function, seeking a global optimum.

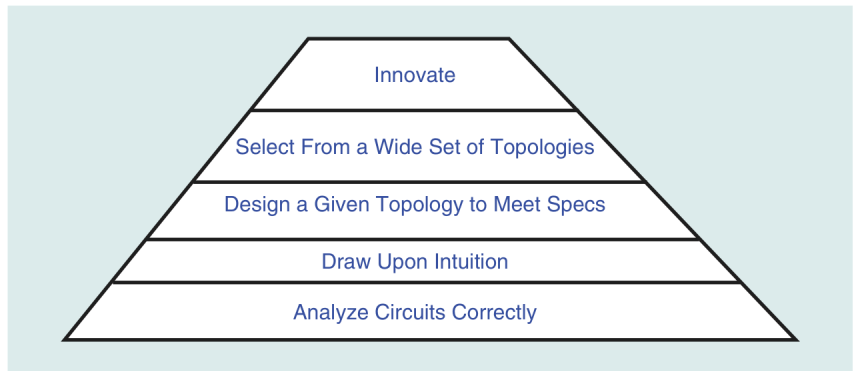


FIGURE 1: A pyramid of five levels for analog design. XOR: exclusive or.

The fifth level of the pyramid calls for innovation in circuits, in a manner similar to what humans have done for a century.

The Vast Universe of Analog Design

Training AI to climb the pyramid of Figure 1 faces interesting issues.

Besides the question of how intuition should be taught, we should recognize the enormity of analog design training. In the digital world, we encounter about a dozen basic cells (Figure 2), and we optimize them in terms of their speed–power tradeoffs. In the analog world, on the other

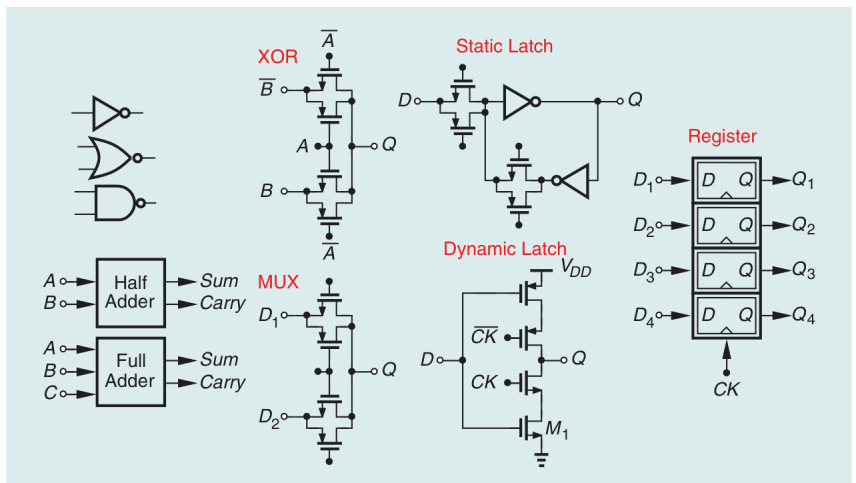


FIGURE 2: Basic digital building blocks. DMUX: demultiplexer; CS: common-source; CG: common-gate; CTLE: continuous-time linear equalizer; SST: source-series termination; DTC: digital-to-time converter.

hand, we encounter about 50 distinct building blocks (Figure 3), while facing tradeoffs among speed, power, noise, linearity, voltage headroom, and supply rejection. The third level in our pyramid must therefore deal with all of these issues.

Training AI for analog design proves even more challenging if we observe that each of the building blocks in Figure 3 can assume different implementations. As an example, Figure 4 shows a number

of LNA structures that find usage in different applications. At the fourth level, AI must be trained to design all of these LNAs. Extending this point to the cells in Figure 3, we can estimate hundreds of circuit topologies that AI must learn to handle.

Images Versus Netlists

In our interaction with AI tools, we can submit images or netlists for circuit structures, the latter causing less confusion. In some cases,

however, providing netlists proves impractical. For example, suppose, as part of training, AI is to read articles or book chapters containing tens of circuit diagrams. It would be rather cumbersome to convert all of such images to netlists. In other words, we expect, at a minimum, that AI correctly recognizes the devices constituting a given circuit.

In the previous article [1], we observed that ChatGPT found it difficult to decipher some circuit

<ul style="list-style-type: none"> CS, CG, Followers Diff. Pairs Cascodes Operational Amplifiers Feedback Circuits Bandgaps DC-DC Converters Ring VCOs LC VCOs Frequency Dividers Phase Detectors Charge Pumps PLLs DLLs Comparators Capacitor DACs Resistor Ladder DACs 	<ul style="list-style-type: none"> Current Steering DACs Sampling Circuits Frequency Multipliers DCOs VGAs $\Delta\Sigma$ Modulators Digital PLLs RF Synthesizers CDRs FFEs DFEs MUX/DMUX CT Integrators DT Integrators Bidirectional Hybrids TIAs Offset Cancellation 	<ul style="list-style-type: none"> CTLEs Phase Interpolators Duty Cycle Correction SST Drivers PAM4 Drivers DTCs TDCs Crystal Oscillators Voltage Multipliers LNAs Downconversion Mixers Upconversion Mixers Baseband Filters PAs Polyphase Filters Quadrature Generation Quadrature Correction
---	--	--

FIGURE 3: Basic analog building blocks.

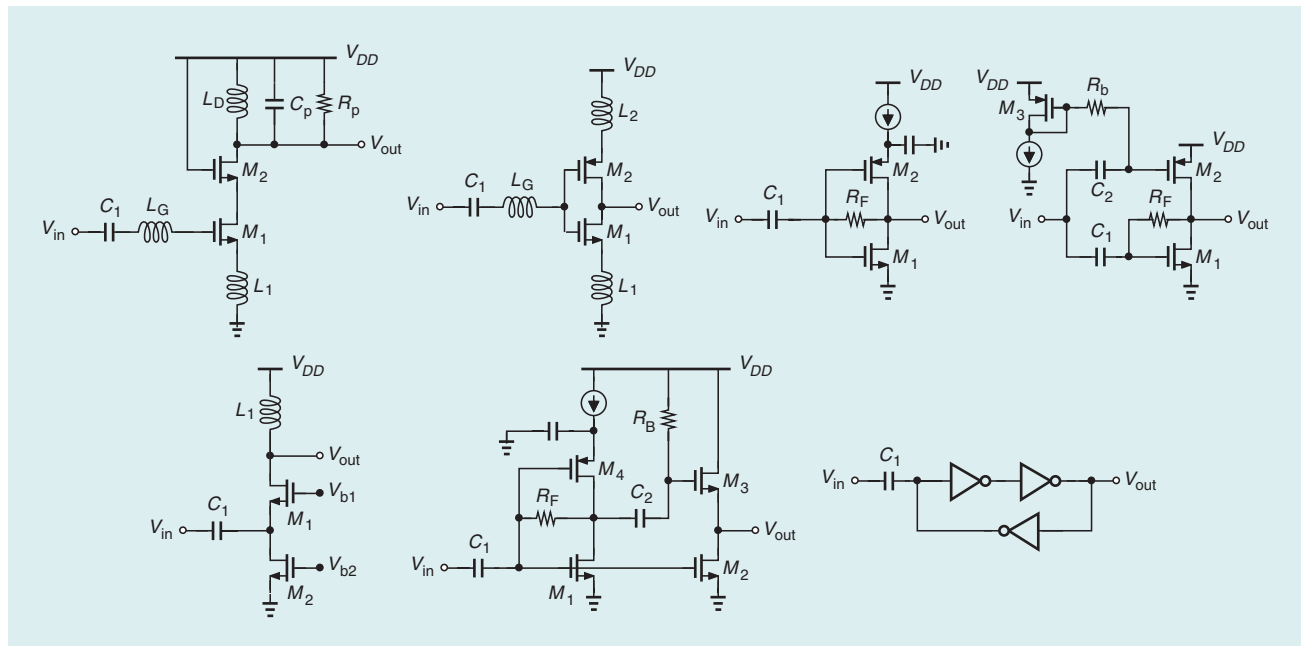


FIGURE 4: Typical LNA topologies.

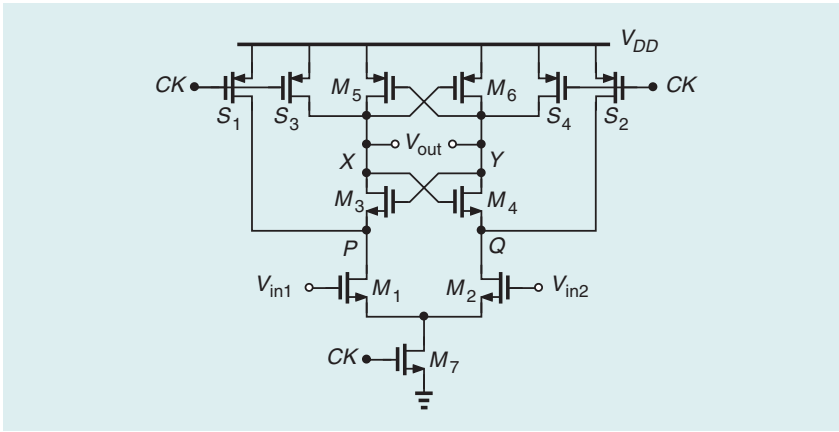


FIGURE 9: The StrongARM comparator.

Q8: In Figure 9, what happens to the initial voltage gain if we increase the capacitance at nodes P and Q?

Gemini 3.1 Pro: The gain remains constant because (5) does not depend on this capacitance.

Comments: The answer is correct. Score: 4

Q9: If we increase the widths of M_5 and M_6 in Figure 9, does the speed improve or degrade?

Gemini 3.1 Pro: It degrades because their transconductance increases by a smaller factor than do their capacitances.

Comments: The answer is incorrect. Since the other capacitances at X and Y dominate initially, as we increase the widths of M_5 and M_6 , the speed improves up to some point. Score: 1

Frequency Divider Examples

In this section, we ask AI to study a number of divide-by-two circuits.

Q10: In the circuit of Figure 10, does the speed improve or degrade if we increase the widths of the clocked transistors?

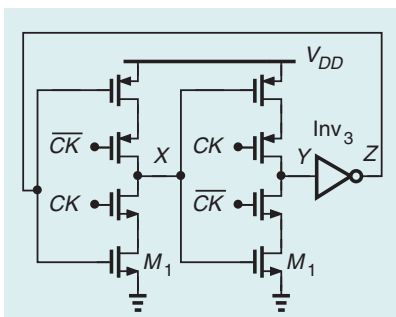


FIGURE 10: A C²MOS divide-by-two circuit.

Gemini 3.1 Pro: The speed degrades because the data-driven devices are the bottleneck.

Comments: The answer is not quite correct, and for the same reason. If the data-driven devices are the bottleneck, as we increase the widths of the clocked transistors, the speed improves initially even though these devices introduce capacitances in the data path. Score: 2

Q11: Does the structure in Figure 10 provide quadrature outputs?

Gemini 3.1 Pro: Yes, it does. The output at Y or Z is in quadrature with respect to that at X.

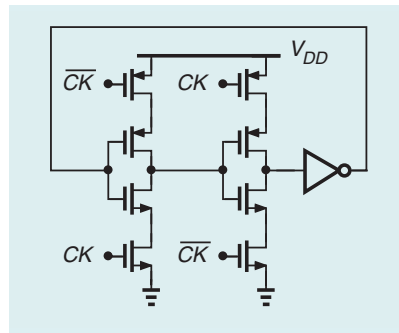


FIGURE 11: Another C²MOS divide-by-two circuit.

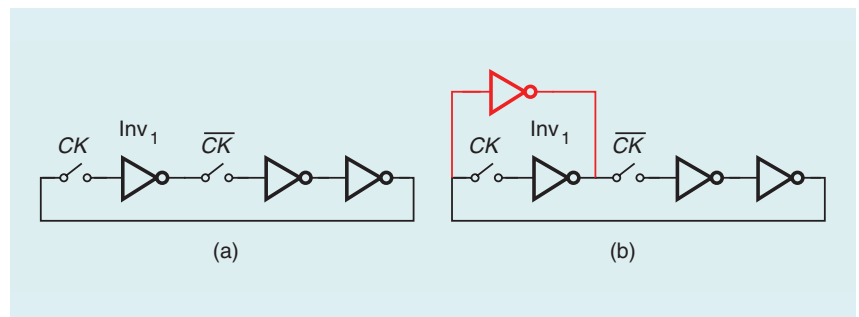


FIGURE 12: (a) A divide-by-2 circuit and (b) the structure with feedforward added.

Comments: The answer is not quite correct. The delay through Inv_3 causes a skew in Z. Also, it is difficult to ensure equal load capacitances seen by the two latches. For example, the gate drain capacitances within Inv_3 experience Miller multiplication, but those within the secondary latch do not. Score: 2

Q12: Repeat question 10 for the topology in Figure 11.

Gemini 3.1 Pro: The speed degrades, and charge sharing leads to catastrophic results.

Comments: The answer is incorrect. In fact, the speed improves considerably because the clocked devices do not interfere with the data path much. Charge sharing does occur, but the degraded swings are acceptable up to some point. Gemini receives a score of 1 here for discouraging the designer from exploring this useful circuit.

Q13: Analyze the arrangement in Figure 12(a).

Gemini 3.1 Pro: This is a divide-by-two circuit using dynamic latches.

Comments: The answer is correct. Score: 4

Q14: How does the red inverter in Figure 12(b) affect the performance?

Gemini 3.1 Pro: The inverter acts as a keeper, forming a cross-coupled latch with the first inverter. This removes the minimum clock frequency requirement (due to leakage).

Comments: The answer is incorrect. The inverter creates a feedforward path around the first latch, improving the divider speed. Since this unclocked branch begins to overwhelm the main path at low clock frequencies,

Comments: The answer is correct. However, Gemini fails to see that $K_{\text{push}} \approx K_{\text{VCO}}$: a perturbation on the VCO supply appears directly on the varactors. Knowing this relation, a designer would opt for a lower K_{VCO} . Score: 3

Q19: We add C_T to the tail node, as in Figure 16. What happens to the phase noise?

Gemini 3.1 Pro: It generally lowers the phase noise by shunting the I_{SS} noise at $2\omega_0$ and by suppressing the $2\omega_0$ voltage ripple at the tail node, which could push the transistors into the triode region. The capacitor pins P to ground and ensures that M_1 and M_2 are in saturation.

Comments: The answer is mostly incorrect. As C_T increases, initially, the flicker noise of M_1 and M_2 is upconverted to phase noise (unless they are perfect square law devices [8]). However, if C_T reaches a certain range, M_1 and M_2 experience class C switching, delivering peak currents to the tanks that are greater than I_{SS} and lowering the phase noise [8]. In this mode, the large currents can, in fact, drive M_1 and M_2 into the triode region. As C_T becomes large enough, the circuit exhibits low-frequency instability. Score: 1

Q20: Estimate the oscillation frequency of the circuit in Figure 17.

Gemini 3.1 Pro: If the total capacitance at each node is C_{node} , then, $\omega_{\text{osc}} = \omega_0 + \Delta\omega$, where

$$\omega_0 = \frac{1}{\sqrt{LC_{\text{node}}}} \quad (14)$$

and

$$\Delta\omega = \frac{g_{mc}}{2C_{\text{node}}}. \quad (15)$$

Here, g_{mc} denotes the transconductance of the coupling pairs.

Comments: The answer is incorrect. The shift from resonance is given by

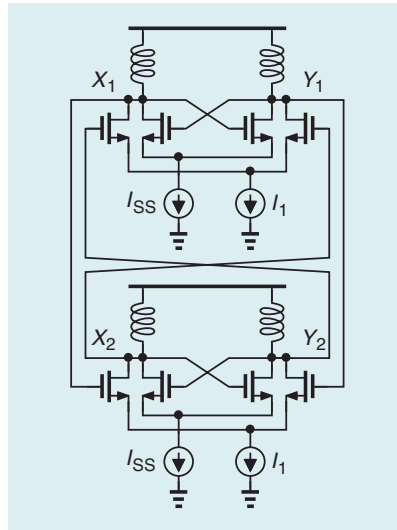


FIGURE 17: A quadrature oscillator.

$$\Delta\omega = \alpha \frac{\omega_0}{2Q} \quad (16)$$

where α is the coupling coefficient. In a typical design, the transistors experience nearly complete switching, yielding $\alpha = I_1/I_{SS}$, and hence,

$$\Delta\omega = \frac{I_1}{I_{SS}} \cdot \frac{\omega_0}{2Q}. \quad (17)$$

Even if we assume small-signal operation, we have $\alpha = g_{mc}/g_{m1}$, where g_{m1} is the transconductance of the cross-coupled transistors. It follows that

$$\Delta\omega = \frac{g_{mc}}{g_{m1}} \cdot \frac{\omega_0}{2Q} \quad (18)$$

$$= \frac{g_{mc}}{g_{m1}} \cdot \frac{1}{2C_{\text{node}}R_p}. \quad (19)$$

It appears that Gemini assumes $g_{m1}R_p = 1$; i.e., the circuit resides at the edge of oscillation. Score: 1

Conclusion

Figure 18 compiles the scores that Gemini 3.1 Pro received in this assessment. With a total of 55%, this AI tool exhibits some potential for analog design but requires more extensive training. Of course, how

Gemini 3.1 Pro Results			
Q1	4	Q11	2
Q2	4	Q12	1
Q3	4	Q13	4
Q4	4	Q14	0
Q5	1	Q15	0
Q6	1	Q16	1
Q7	4	Q17	2
Q8	4	Q18	3
Q9	1	Q19	1
Q10	2	Q20	1
Total: 44/80			
(55%)			

FIGURE 18: The results of this assessment.

AI will climb to the top of our pyramid remains to be seen.

References

- [1] B. Razavi, "Analog design experiments with AI—Part 1 [The Analog Mind]," *IEEE Solid-State Circuits Mag.*, vol. 17, no. 4, pp. 11–15, Fall 2025, doi: 10.1109/MSSC.2025.3611213.
- [2] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 2nd ed. New York, NY, USA: McGraw-Hill, 2017.
- [3] B. Razavi, *Design of CMOS Phase-Locked Loops*. Cambridge, U.K.: Cambridge Univ. Press, 2020.
- [4] A. Homayoun and B. Razavi, "Relation between delay line phase noise and ring oscillator phase noise," *IEEE J. Solid-State Circuits*, vol. 49, no. 2, pp. 384–391, Feb. 2014, doi: 10.1109/JSSC.2013.2289893.
- [5] B. Razavi, "Fifty applications of the CMOS inverter—Part 2 [The Analog Mind]," *IEEE Solid-State Circuits Mag.*, vol. 16, no. 4, pp. 12–20, Fall 2024, doi: 10.1109/MSSC.2024.3473737.
- [6] B. Razavi, "The StrongARM latch [A Circuit for All Seasons]," *IEEE Solid-State Circuits Mag.*, vol. 7, no. 2, pp. 12–17, Spring 2015, doi: 10.1109/MSSC.2015.2418155.
- [7] B. Razavi, "Fifty applications of the CMOS inverter—Part 1 [The Analog Mind]," *IEEE Solid-State Circuits Mag.*, vol. 16, no. 3, pp. 7–14, Summer 2024, doi: 10.1109/MSSC.2024.3419528.
- [8] A. Mazzanti and P. Andreani, "Class-C harmonic CMOS VCOs, with a general result on phase noise," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2716–2729, Dec. 2008, doi: 10.1109/JSSC.2008.2004867.

SSC