This is a fantastic line of thinking—especially as it spans signal design, encoding, memory management, error correction, and adaptable gates. Let's break down and expand on some of your key thoughts:

1. Multiple Signals and a Master Signal:

It is indeed possible to combine multiple signals, especially in a continuous wave system, similar to how we modulate different channels in telecommunications. Here, signals could differ in frequency, amplitude, or phase. You might even use frequency bands to encode distinct data streams (like RAM, program control, or user input). The concept of a "master signal" that can split into different channels or derive necessary signals reminds me of multiplexing, where multiple signals are combined into one for transmission and later demultiplexed.

- **Food for thought: Could we explore whether multiplexing principles (time division, frequency division, etc.) apply here? What would it mean to manage all control and data via a few, or even one, signal?**

2. Feedback and Error Correction:

Amplification and signal copying could certainly be used for feedback. The more common issue in analog systems is signal drift over time, which is where feedback loops can come in to compare a signal to a reference (a known correct value) and adjust accordingly.

- **Amplifiers: These might need to boost the signal at specific points, but copying and comparing signals could act as a built-in redundancy system.**

- **Error correction: Systems like Hamming codes in digital computation allow for the detection and correction of single-bit errors—analog systems might have equivalents, like comparing a degraded signal to the original waveform. Would you envision checking against an internal library of signal patterns?**

- **Food for thought: Could feedback loops be a form of "self-healing," where the signal corrects itself in real-time based on its redundancy or an inherent property of** the system?

3. Gates that Adapt and Nodes:

The idea of adaptable gates ties into machine learning (ML), where each "neuron" or node in a neural network adjusts based on incoming signals (learning) during training. In analog computing, an adaptable gate might operate similarly, adjusting its operation based on the input it receives. It could modulate itself to handle different types of logic or arithmetic operations dynamically.

- Quantum computing parallel: You’re right about quantum computing, where entangled qubits are interconnected in a highly interactive state. In your system, if gates could behave like interconnected nodes, you might have something similar, where certain gates can “learn” from others in complex, problem-solving scenarios.

- Food for thought: Could this adaptability apply not just to logic but also to how signals are routed? Would the system “optimize” itself based on the problem? Could we design the gates to reconfigure themselves like an FPGA (Field-Programmable Gate Array) but in an analog way?

4. Analog Memory:

Traditional memory would involve storing binary states (0s and 1s), but for analog memory, you're essentially storing continuous waveforms or signal states. Capacitors, as you mentioned, can hold a charge briefly, which mimics short-term memory. The challenge will be how to preserve this memory over time.

- State-capture: If we imagine memory as a snapshot of a signal’s energy signature, then replicating it means you would capture its waveform (amplitude, frequency, phase) at a given time and be able to reproduce it when necessary. But what if this “analog memory” could store more than just binary values? Could we explore storing different signal modulations, creating richer, continuous memory?

- **Food for thought: Could the system store memory as continuous values (e.g., voltage levels or wave frequencies), which could later be used to reconstruct state-dependent operations? Could this memory be used to "train" the system to recognize certain patterns?**

5. Scaling and Number Systems:

The idea of exploring different number systems is fascinating. Binary is great for digital, but in analog, where continuous values matter, something like base 10 could feel more natural. Base 10 scaling seems intuitive, but we might want to think about bases that align more closely with the properties of signals (e.g., logarithmic scales, harmonic frequencies).

- **Food for thought: What if the analog computer operated on a non-integer or fractional number system? Could we create a system where the “gates” process real numbers or fractions, reflecting the natural fluidity of analog signals?**

6. Quantum-Style Nodes:

If we treat each gate like a quantum node, we could explore how the nodes interact independently of the ALU. For example, instead of the ALU directing operations, could the gates operate more like a neural network, where the interaction of nodes leads to problem-solving? The ALU might just kickstart processes, but then the gates self-organize based on the problem.

- **Food for thought: Could we create a system where logic gates are like neurons that collaborate to solve problems without direct control from the ALU, leading to a more “organic” computing process?**