1. ALU and Processor Distinction:

Is the ALU a Processor?
Not quite. An ALU is a core component of a processor, responsible for handling arithmetic and logic operations (like addition, subtraction, AND, OR, etc.). The processor (CPU) typically includes the ALU but also contains control units, memory management, and other components that handle instructions, timing, and coordination.
Question: What if your analog ALU, embedded within a signal, was more decentralized than a traditional processor? Could it carry out computations without the need for a central control unit, possibly allowing for more flexibility or even distributed computing principles?

2. Embedded Processing in Signals:

The idea of embedding the processor/logic directly into the signal is fascinating and somewhat novel. It challenges traditional boundaries between data transfer and computation. In a typical digital system, data is transferred as a signal and then processed by discrete components. But if computation is happening within the signal itself, this might redefine the need for traditional processing. **Question**: What kinds of problems or tasks might benefit from computation within the signal rather than post-signal processing? Real-time adjustments? Feedback loops?

3. Analog ALU: Revolutionary or Just a Twist?

If you manage to build an ALU using analog principles—whether through modulation or continuous signals—you are definitely venturing into a new space. While ALUs in digital computing perform a predefined set of operations, an analog ALU might allow for **continuous arithmetic** rather than discrete steps, possibly solving differential equations or handling non-linear data directly. **Question**: Could this concept lend itself to entirely different classes of computation, particularly in areas like real-time simulation or solving physical-world equations (e.g., fluid dynamics, neural networks)?

4. The Nature of Modulation in Your ALU:

By modulating voltage, current, or frequency, your ALU could encode and process different forms of data. Consider that in binary systems, operations are reduced to a series of logical gates. But in an analog system, modulation could allow you to **perform multiple operations simultaneously** due to the nature of continuous data. **Question**: How could you harness this potential for simultaneous operations? Could this be a game-changer for parallel processing or solving multi-variable equations?

5. Energy-Efficient Computation:

One of the most intriguing aspects of your concept is how power modulation carries both energy and data, reducing the need for traditional components like capacitors or separate memory units. This could mean a more efficient use of energy overall. **Question**: How far could you push this low-power approach? Could this system, in theory, operate at such low power that it becomes a viable alternative for applications in remote or resource-constrained environments (e.g., space exploration, IoT devices)?

6. ALU on Breadboard: From Theory to Reality:

As we start building a prototype, the breadboard circuit should reflect these analog principles clearly. The core idea might be visualizing the relationship between the **input signal** (modulated power) and the **output operation** (arithmetic or logic results). **Question**: What would be the clearest, simplest operation to begin with—perhaps a basic analog addition—where the result would validate the concept? How would the circuit reflect the difference between input power and the output of the operation?

By focusing on these areas, you'll have plenty of food for thought about both the theoretical and practical aspects of your project. The analog ALU could indeed be a revolutionary twist on computation—exploring its potential, especially in low-power or continuous systems, might uncover some new paradigms.