ALU Module Explained

ALU stands for **Arithmetic Logic Unit**.
It is the primary building block in any CPU for doing calculations—addition, subtraction, and various logic operations such as AND and OR.

Name	Dir.	Width	Description
a	input	32-bit	First operand
b	input	32-bit	Second operand
aluop	input	3-bit	Selects the operation to perform
y	output	32-bit	Result of the selected operation
zero	output	1-bit	True if result `y` is zero (`y == 0`)

Operation Selection

The `aluop` input chooses which calculation or logic operation the ALU should perform on inputs `a` and `b`:

aluop	Operation	Meaning
000	y = a + b	ADD
001	y = a - b	SUB
010	y = a & b	AND (bitwise AND)
011	y = a | b	OR (bitwise OR)
other	y = 0xDEAD_BEEF	Default/error case

1. **Combinational Logic**
   The result `y` is computed continuously based on the current inputs—no waiting for a clock; it's combinational.
2. **Zero Flag**
   The `zero` output is set to `1` (true) if the result `y` equals zero. This is commonly used for branch decisions (`beq`, `bne`, etc.).

Where does the ALU fit in the CPU?

- **EXE Stage:**
  The ALU is the main component in the Execute (EXE) pipeline stage. It receives operands (numbers to operate on) and control signals (which operation to perform) from previous pipeline stages.
- **Arithmetic & Logic:**
  All arithmetic (`add`, `sub`) and logical (`and`, `or`) operations in the CPU are performed here.
- **Branch Decisions:**
  The `zero` output helps the CPU quickly check if two values are equal (used by branch instructions).