EXE_WB Stage Module Explained

The `EXE_WBstage` module is a **pipeline register** (latch) between the **Execute (EXE)** and **Write Back (WB)** stages in your CPU pipeline.

- Its job is to **store the outputs from the EXE stage** (specifically, the ALU result and register-control signals) so they're reliably available to the Write Back stage on the next clock cycle.
- It helps maintain correct, clocked data flow across the pipeline.

Signal	Dir	Width	Description
clk	in	1 bit	Clock signal (syncs the register)
rst	in	1 bit	Reset (clears the pipeline register)
alu_y_in	in	32 bits	Result output from the ALU (EXE stage)
rd_in	in	5 bits	Destination register number
regwrite_in	in	1 bit	Register write-back enable
alu_y_out	out	32 bits	ALU result forwarded to WB stage
rd_out	out	5 bits	Destination register for WB
regwrite_out	out	1 bit	Write-back enable for WB

How It Works — Internals

- **On the positive edge of the clock (rising edge):**
  - If **reset** (`rst`) is asserted, the outputs are cleared to zero — **safe startup/flush state**.
  - If **no reset**, the input values are copied to the outputs.
- These outputs are held constant until the next clock edge, just like a D flip-flop.

What values are moved through?

- The **ALU result**—the computation performed by the EXE stage (often an ADD, SUB, etc.).

- The **destination register number** (rd)—tells the write-back stage which register, if any, should be written.
- The **regwrite flag**—indicates whether the instruction's result should be written to the register file.

Example scenario:

Clock Edge	alu_y_in	rd_in	regwrite_in	===>	alu_y_out	rd_out	regwrite_out
@t	0x00000015	x3	1		0x00000015	x3	1

Role in the Pipeline

- ALU completes the operation in the EXE stage and produces the result (alu_y_in).
- EXE_WBstage latches the result and the control signaling about whether/where to write.
- On the next clock cycle (in the WB stage), the result is written to the specified register—but only if regwrite_out is 1.
- Without this buffer, values might "change underfoot" due to overlapping instructions in the pipeline. This module keeps results stable from EXE to WB, even as new instructions begin flowing through EXE.

In Summary

- **EXE_WBstage** is a simple, essential pipeline register.
- It acts as a reliable “hand-off” between the execute and write-back parts of your CPU.
- It ensures only valid data/control signals reach the write-back, prevents race conditions, and helps the pipeline move smoothly.