ALU refactor

Task: Update the input and output ports of your assigned module to use simple, **non-enum and non-record types (both in VHDL and Verilog)**. This will make the design easier for beginners to understand and trace signal flow.

If you have any questions while working on it, or once you're done, please let me know so I can review your work and give feedback.

Note: You're welcome to use your own naming conventions—just try to keep them descriptive. If you can simplify or shorten the names I've used, it's even better—but no worries if not!

Inputs:

32-bits :	A, B
3-bits	f3
7-bits	f7

Outputs:

32-bits :	result
1-bit:	Z, V, C, N

Notes for Refactor:

- Use **constants** for `funct3` and `funct7` values, similar to my implementation. This makes it easier to identify which operation is being performed and improves readability.
- You can **remove `res_temp.operation`** — it was mainly used for debugging, and I plan to handle that in the testbench.
- For **VHDL**, remember that all `case` and `if` statements must be placed **inside a `process` block**.
- Make sure **all relevant input signals are included in the sensitivity list** of each process to avoid simulation mismatches or unintended latching behavior.

Note: This ALU is a refined version of the one I initially developed for my pipelined CPU. It has been updated and optimized based on feedback and is now being used in my Superscalar CPU project. For more details, refer to the documentation in my Superscalar repository: https://github.com/NoridelHerron/SUPERSCALAR_CPU/tree/main