1. Ablation (bugs introduced into your RTL)

I intentionally introduced **three separate logical bugs** across multiple files so the fix requires multiple reasoning steps and edits across modules. Each bug compiles and simulates but yields incorrect ***functional*** behavior.

🔹 Bug I — Arithmetic inversion in `add_sub`

Change: swapped ADD/SUB selection so `inst[5]` does the opposite operation.

In `add_sub`:

```
assign add_sub_out = inst_5 ? (A + bus) : (A - bus);
```

Instead of subtraction when `inst_5==1`, it incorrectly adds.

Effect: `ADD` executes as subtraction and `SUB` executes as addition. Subtle because arithmetic still produces 8-bit outputs and no syntax errors.

🔹 Bug II — Register `C` load broken

removed/commented-out the assignment that writes `bus` into register `C` when its load bit is active.

In `micro`:

```
mux8_2to1 bus_to_c (LD[3], bus, C, C1);
dff dC (clk, C1, C);
```

Comment this block out.

Effect: Register `C` never updates (MOV/IN to C silently fails).

🔹 Bug III — Tri-state buffer for `A` swapped

In `micro`:

```
tribuff_8_2to1 from_A (OE[1], A, bus);
```

Change it to:

```
tribuff_8_2to1 from_A (OE[1], 8'bz, bus);
```

Effect: When trying `OUT A`, the bus is high-impedance instead of A's value.

Why these bugs are good

Each bug is independent and in a different area (ALU, register-load logic, bus/mux wiring). A single failing test will not point to all faults at once. A junior engineer must run the testbench, inspect waveforms/monitors, trace signals across modules, and apply fixes in multiple modules.