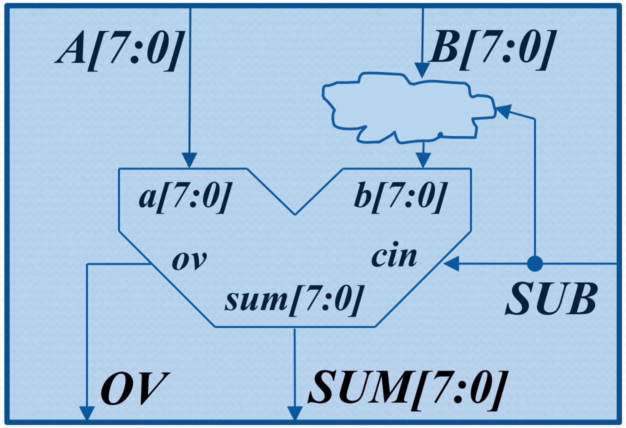
Exercise 5 (Testbench for Arithmetic Block):

- You will create a testbench for an 8-bit arithmetic block.
  - Arithmetic unit can perform addition or subtraction.

SUM = A + B if SUB is low
SUM = A – B if SUB is high

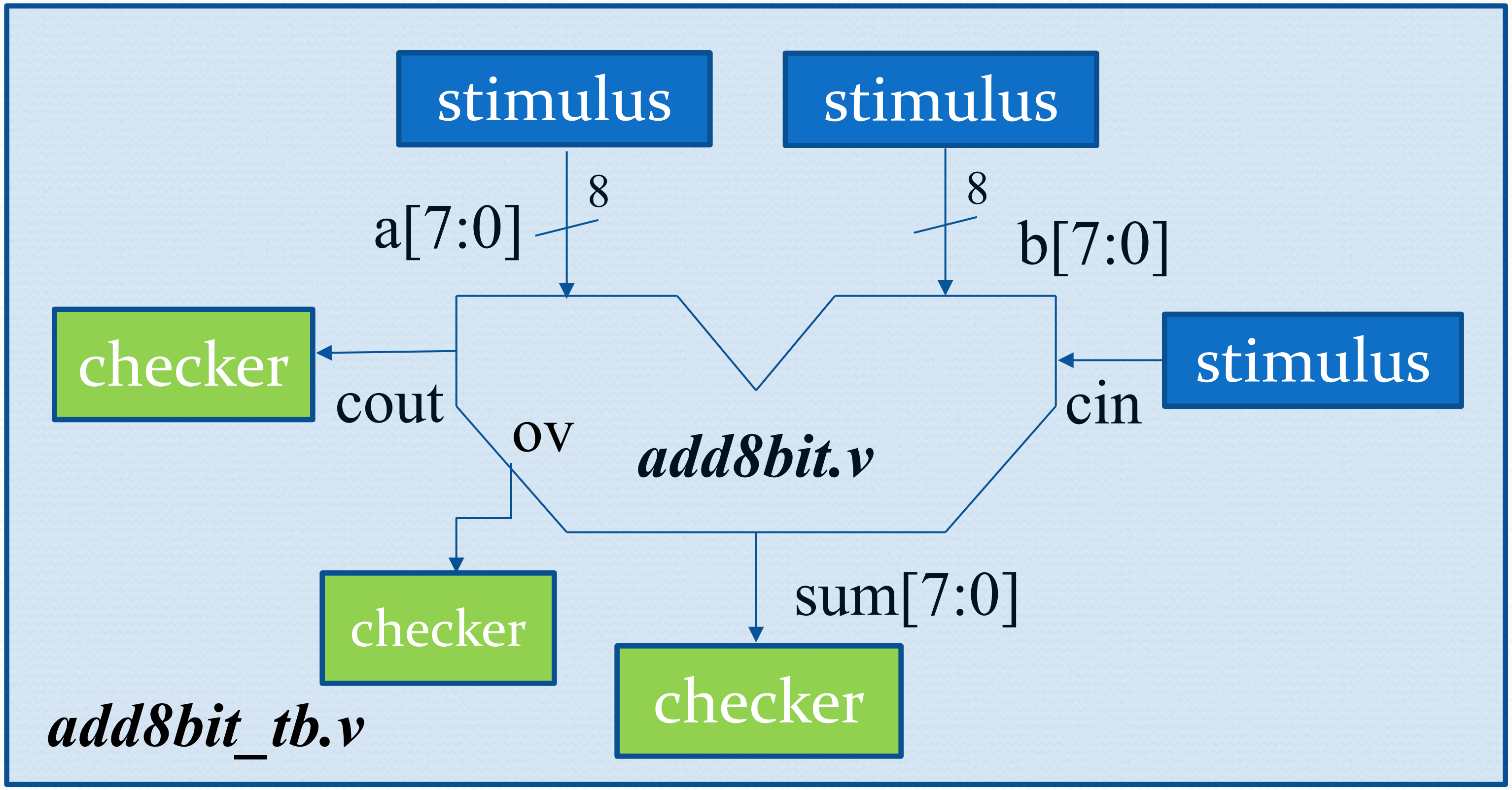
  - Arithmetic unit also outputs an overflow signal if the result is either too positive or too negative to represent as an 8-bit **signed** number.

  - Code for DUT **(arith.sv**) is provided.



arith.sv

Test Benches with Verilog



- Elements of a test bench.
    - Instantiate DUT
    - Apply Stimulus
    - Good test bench also checks results

Test Bench Example:

I use type **reg** for stimulus and type **wire** for monitoring outputs because I am an old man who can’t change his ways. You use type **logic** for both.

```
module add8bit_tb();

reg [7:0] stm_a,stm_b;   // used as stimulus for a,b
reg stm_cin;             // used as stimulus for cin

wire [7:0] sum_mon;      // used to monitor sum
wire cout_mon,ov_mon;    // used to monitor cout & ov

add8bit iDUT(.a(stm_a), .b(stm_b), .cin(stm_cin),
             .cout(cout_mon), .sum(sum_mon),
             .ov(ov_mon));

initial begin
  stm_a = 8’hA5;
  stm_b = 8’h5A;
  stm_cin = 0;           // result should be FF
  #5;                    // wait 5 time units
  …                      // I expect more stimulus
end

endmodule
```

Exercise 5 (Testbench for Arithmetic Block):

- Your testbench is for **arith.sv** *(what I showed was for an adder only block)*
  - The stimulus you apply should test both addition and subtraction
  - The stimulus you apply should test for overflow conditions.
  - **arith.sv** contains a bug in the **ov** logic. See if your testbench can find it.
  - Your testbench should include commenting telling what you are trying to test. It does not have to be self checking. We will work on that more in the future.
  - You can use AI/LLM for generating the template of the testbench, but it is best if you think of the corner conditions to test on your own.
  - Submit **arith_tb.sv,** and waveforms showing you ran it to the dropbox
  - Submit a fixed version of **arith.sv** *(fixing the bug in the **ov** logic)*