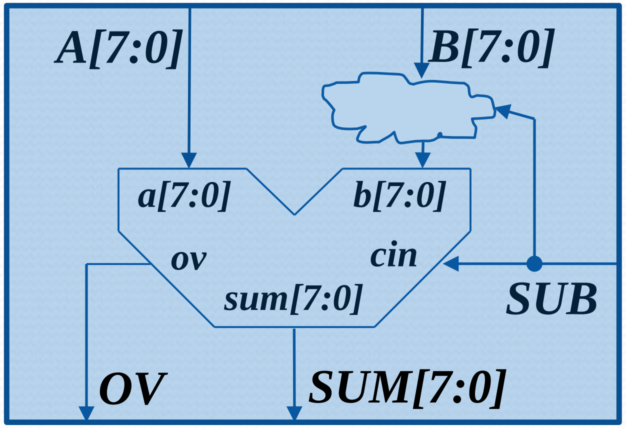
Exercise 4 (Testbench for Arithmetic Block):

- 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

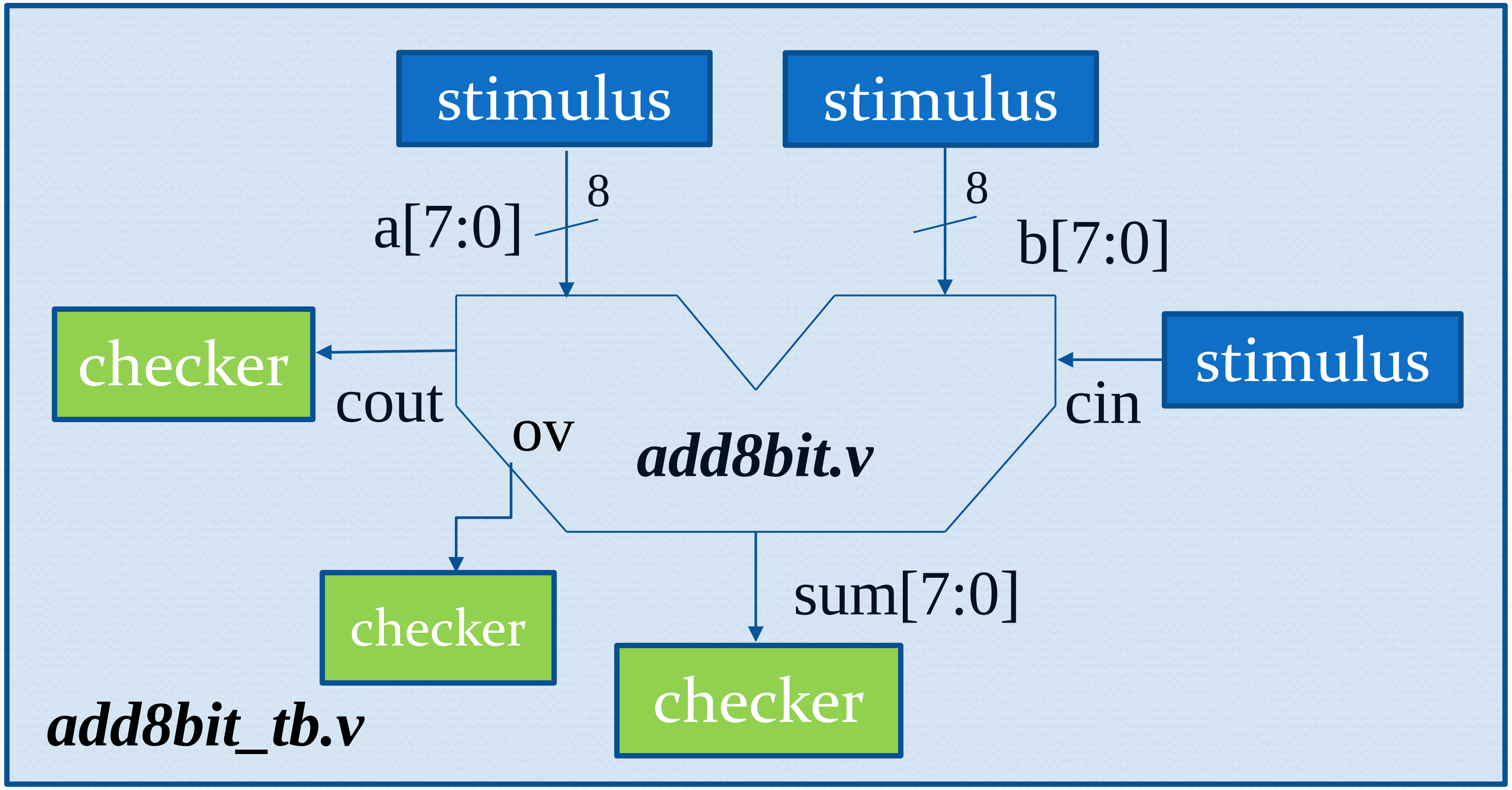
  - Arithmetic unit also outputs an overflow signal if the result is either to positive or too negative to represent as an 8-bit signed number.

  - Code for DUT **(arith.v**) is provided in Exercise04 folder on Moodle



arith.v

Test Benches with Verilog



- Test bench is implemented in Verilog too
  - Instantiates DUT
  - Applies Stimulus
  - Good test bench also checks results

Test Bench Example:

```
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 4 (Testbench for Arithmetic Block):

- Your testbench is for **arith.v** (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.
  - 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.
  - Submit **arith_tb.v,** and waveforms showing you ran it to the dropbox