‘timescale 1ns/1ns  
module tb();  
 // Generates a 50MHz clock.  
 logic clk;  
 initial clk = 1’b0; // Clock starts at 0  
 always #10 clk = ~clk; // Wait 10ns, flip the clock, repeat forever

logic reset;  
 logic [7:0] dut\_x, dut\_y;  
 logic [15:0] dut\_result;  
 logic dut\_go, dut\_done;  
 ArrayMultiplier DUT  
 (  
 .clk(clk), .reset(reset), .i\_x(dut\_x), .i\_y(dut\_y),  
 .o\_result(dut\_result), .i\_go(dut\_go), .o\_done(dut\_done)  
 );

initial begin  
 dut\_go = 1’b0; // Start with the go signal off  
 reset = 1’b1; // Start with reset on  
 @(posedge clk);  
 reset = 1’b0; // Leave it on for a clock cycle and then turn it off

for (integer i = 0; i < 256; i++) begin  
 for (integer j = 0; j < 256; j++) begin  
 dut\_x = i; // Set up multiplier inputs  
 dut\_y = j;  
 dut\_go = 1’b1; // Activate go signal  
 @(posedge clk);  
 dut\_go = 1’b0; // Turn it off after 1 cycle

wait(dut\_done); // Wait for adder to finish - result is valid now

logic [15:0] realresult;

realresult = i \* j;

If (dut\_result !== realresult) begin

$display(“Mismatch! %d \* %d should be %d, got %d instead”, i, j, realresult, dut\_result);

$stop();

@(posedge clk); // Wait 1 cycle before sending next inputs  
 end  
 End

$display(“Test passed!”);

$stop();  
end

endmodule