module asmd\_multiplier\_tb;

// Parameters

localparam word\_length = 4;

// Ports (inputs and outputs)

wire [2\*word\_length-1:0] product;

wire ready;

reg  [word\_length-1:0] word0;

reg  [word\_length-1:0] word1;

reg  start;

reg  clk = 0;

reg  reset;

reg  [2\*word\_length-1:0] model\_product; // Expected product for verification

// ASMD multiplier design-under-test (DUT)

asmd\_multiplier # (

  .word\_length(word\_length)

)

asmd\_multiplier\_inst (

  .product(product),

  .ready(ready),

  .word0(word0),

  .word1(word1),

  .start(start),

  .clk(clk),

  .reset(reset)

);

// Clock Generation

always #5  clk = !clk;

// Task to apply test cases

task apply\_test(input [word\_length-1:0] a, input [word\_length-1:0] b, input [2\*word\_length-1:0] expected\_result);

begin

    word0 = a;

    word1 = b;

    start = 1;

    #10 start = 0;

    // Wait for the ready signal

    wait (ready);

    #10;

    // Check the product against expected result

    if (product == expected\_result) begin

        $display("PASS: %0d \* %0d = %0d", a, b, product);

    end else begin

        $display("FAIL: %0d \* %0d, Expected = %0d, Got = %0d", a, b, expected\_result, product);

    end

end

endtask

// Task to verify the initial reset state

task verify\_reset\_state;

begin

    // Assert reset

    reset = 1;

    start = 0;

    word0 = 0;

    word1 = 0;

    #10;

    // Check if the system is in the idle state and ready

    if (ready == 1 && product == 0) begin

        $display("PASS: Reset works as expected.");

    end else begin

        $display("FAIL: Reset behavior is incorrect.");

    end

    // Deassert reset

    reset = 0;

    #10;

end

endtask

// Initial block to apply test cases

initial begin

    $dumpfile("asmd\_multiplier.vcd");

    $dumpvars(0, asmd\_multiplier\_tb);

    // ASMD\_MULT\_TP001: Verify interface

    // These checks can be done through compilation and synthesis, thus not simulated directly.

    // ASMD\_MULT\_TP002: Verify default parameters

    // Checking if word\_length is correctly set to 4 can be done during synthesis and inspection of the design.

    // ASMD\_MULT\_TP003: Reset Test

    verify\_reset\_state;  // Verifies that reset initializes everything correctly

    // ASMD\_MULT\_TP004: Test various word0 and word1 inputs and transitions

    // Test Case 1: Multiply 3 by 5 (simple case)

    apply\_test(4'b0011, 4'b0101, 8'b00001111); // Expected product = 15

    // Test Case 2: Multiply 0 by 5

    apply\_test(4'b0000, 4'b0101, 8'b00000000); // Expected product = 0

    // Test Case 3: Multiply 7 by 4

    apply\_test(4'b0111, 4'b0100, 8'b00111000); // Expected product = 28

    // Test Case 4: Multiply 15 by 15 (maximum value)

    apply\_test(4'b1111, 4'b1111, 8'b11100001); // Expected product = 225

    // Test Case 5: Multiply 9 by 9

    apply\_test(4'b1001, 4'b1001, 8'b01000001); // Expected product = 81

    #100;

    $finish();

end

endmodule