```
module tb_synchronous_ram;
  // Testbench parameters
  parameter ADDR_WIDTH = 4;
  parameter DATA_WIDTH = 8;

  // Testbench signals
  reg clk;
  reg we;
  reg [ADDR_WIDTH-1:0] addr;
  reg [DATA_WIDTH-1:0] din;
  wire [DATA_WIDTH-1:0] dout;

  // Instantiate the synchronous RAM
module
  synchronous_ram #(
    .ADDR_WIDTH(ADDR_WIDTH),
    .DATA_WIDTH(DATA_WIDTH)
  ) uut (
    .clk(clk),
    .we(we),
    .addr(addr),
    .din(din),
    .dout(dout)
);

// Clock generation
initial clk = 0;
always #5 clk = ~clk; // 10ns clock period

// Test procedure
initial begin
    // Test initialization
    we = 0; addr = 0; din = 0;

    // Write to address 0
    @(posedge clk);
    we = 1; addr = 4'h0; din = 8'hAA;

    // Write to address 1
    @(posedge clk);
    we = 1; addr = 4'h1; din = 8'hBB;
```

```
        // Read from address 0
        @(posedge clk);
        we = 0; addr = 4'h0;

        // Read from address 1
        @(posedge clk);
        we = 0; addr = 4'h1;

        // Finish simulation
        @(posedge clk);
        $finish;
    end

    // Monitor the signals
    initial begin
        $monitor("Time = %0t | WE = %b | Addr = %h | Din = %h | Dout = %h", $time, we, addr, din, dout);
    end
endmodule
```