6T SRAM Verilog Code and Testbench

# SRAM 6T Verilog Code:

module sram\_6t\_cell (  
 input wire clk,  
 input wire wl, // Word line (control signal to access cell)  
 input wire write\_en, // Write enable  
 input wire bl, // Bit line  
 input wire blb, // Bit line bar (complement)  
 output reg q, // Stored bit  
 output reg qb // Complement of stored bit  
);  
  
always @(posedge clk) begin  
 if (wl) begin  
 if (write\_en) begin  
 q <= bl;  
 qb <= blb;  
 end  
 end  
end  
  
endmodule

# Testbench for SRAM 6T Cell:

`timescale 1ns / 1ps  
  
module tb\_sram\_6t\_cell;  
  
 reg clk;  
 reg wl;  
 reg write\_en;  
 reg bl;  
 reg blb;  
 wire q;  
 wire qb;  
  
 // Instantiate the SRAM cell  
 sram\_6t\_cell uut (  
 .clk(clk),  
 .wl(wl),  
 .write\_en(write\_en),  
 .bl(bl),  
 .blb(blb),  
 .q(q),  
 .qb(qb)  
 );  
  
 // Clock generation: 10ns period  
 always #5 clk = ~clk;  
  
 initial begin  
 // Initialize  
 clk = 0;  
 wl = 0;  
 write\_en = 0;  
 bl = 0;  
 blb = 1;  
  
 // Wait for a few cycles  
 #10;  
  
 // Write '1' to cell  
 wl = 1;  
 write\_en = 1;  
 bl = 1;  
 blb = 0;  
 #10;  
  
 // Hold value  
 wl = 0;  
 write\_en = 0;  
 #10;  
  
 // Read (no change in bl/blb, but check output stability)  
 wl = 1;  
 write\_en = 0;  
 #10;  
  
 // Write '0' to cell  
 bl = 0;  
 blb = 1;  
 write\_en = 1;  
 #10;  
  
 // Hold again  
 wl = 0;  
 write\_en = 0;  
 #10;  
  
 // Read again  
 wl = 1;  
 write\_en = 0;  
 #10;  
  
 $finish;  
 end  
  
endmodule