Verilog Testbench

How to Write Testbench?

- Create a test module template
  - Declare inputs to the module-under-test (MUT) as "reg", and the outputs as "wire".
  - Instantiate the MUT.
- Initialization
  - Assign some known values to the MUT inputs.
- Clock generation logic
  - Various ways to do so.
- May include several simulator directives
  - $monitor, $finish.

Association of identifiers

Testbench	Module

module t_circuit;
  reg t_A, t_B;
  wire t_C;
  parameter stop_time =100;
 	
circuit M (t_C, t_A, t_B);
 	
//stimulus generation for t_A, t_B goes here
 	
initial # stop_time **$finish**
endmodule
	module circuit (C,A,B)
	
	input A,B;
	
	output C;
	
	//Module body
	
	endmodule

Example

```
initial
  begin
    A = 0; B = 0;
    #10 A = 1;
    #20 A = 0; B = 1;
  end
```

- At time 0, A and B are set to 0.
- Ten time units later, A is changed to 1.
- Twenty time units after (at t=30), A is changed to 0 and B to 1.

Example

```
initial
  begin
  D = 3' b000 ;
  repeat (7)
  #10 D = D + 3' b001 ;
  end
```

2 to 1 line mux - testbench

```
module mux_21 (m_out, A, B, select);
   output m_out;
   input A, B;
   input select;

   assign m_out = (select)? A:B;
endmodule


module mux_21_tb;
  wire mux_out_tb;
  reg A_tb, B_tb;
  reg select_tb;
  parameter stop_time = 120;
```

2 to 1 line mux - testbench

```
mux_21 M1 (mux_out_tb, A_tb, B_tb, select_tb);

 initial #stop_time $finish;

 initial
   begin
     select_tb = 1; A_tb = 0; B_tb = 1;
     #10 A_tb = 1; B_tb = 0;
     #10 select_tb = 0;
     #10 A_tb = 0; B_tb = 1;
 end

 initial
   begin
     $monitor("time =%0d ",$time, "select = %b A = %b B = %b Out = %b", select_tb,
A_tb, B_tb, mux_out_tb);
     end
endmodule
```