测试程序的模板

■ 测试程序的模板如下：

```
module testbench_name();

    //  信号定义

    // 模块实例化

    // 时钟、复位和输入数据初始化，复位信号生成（通过initial语句）

    // 产生时钟（通过always语句）

    // 测试激励产生（通过initial语句，配合事件触发@和时间延迟#）

    // 显示输出结果（可以不添加任何显示打印语句，只生成波形图即可）

endmodule
```

时序逻辑电路模块

Sequential Building Blocks

时序逻辑电路的测试程序

```
module reg_D (input   clk, rst_n,
                      input   [3:0]  D,
                    output logic [3:0]  Q);
  always_ff @(posedge clk) begin
            if (!rst_n) Q <= 4'b0;
            else Q <= D;
  end
endmodule
```

待测模块DUT

打印输出结果：

```
At time    0ns: rst_n = 0, D =  0,  Q =  x
At time    5ns: rst_n = 0, D =  0,  Q =  0
At time 100ns: rst_n = 1, D =  0, Q =  0
At time 115ns: rst_n = 1, D =  1, Q =  0
At time 125ns: rst_n = 1, D =  3, Q =  1
At time 135ns: rst_n = 1, D =  3, Q =  3
```

```
`define CLK_PERIOD 10
module reg_D_tb ( );
  logic                clk, rst_n;
  logic    [3 : 0]     D, Q;
  reg_D dut (.clk(clk), .rst_n(rst_n), .D(D), .Q(Q));
// 时钟、复位、输入数据初始化，复位信号生成
  initial begin
    clk     <= 1'b0;    rst_n   <= 1'b0;    D <= 4'd0;
    #100;               rst_n   <= 1'b1;
  end
  always #(`CLK_PERIOD/2) clk = ~clk;                    // 产生时钟
  initial begin                                          // 产生测试激励
    @(posedge rst_n);          @(posedge clk);
    @(posedge clk);            D <= 4'd1;
    @(posedge clk);            D <= 4'd3;
    #30;                       $finish;
  end
  initial begin
    $timeformat(-9, 0, "ns", 5);
    $monitor("At time %t: rst_n = %b, D = %d, Q = %d", $time, rst_n, D, Q);
  end
endmodule
```