

```

module test_ff; // testbench
reg CLR,CLK,D;
wire Q;

// instantiate device

dff my_FF(.clr(CLR), .q(Q), .clock(CLK), .d(D));

initial
    CLR=0;

initial // define stimuli
begin
    D=0;
    wait(CLK);
    D=1; // input for 1st clock
    wait(!CLK);
    #4 CLR = 1;
    wait(CLK);
    #1 D=0; // input for 2nd clock
    wait(!CLK)
    #2 CLR=0;
    wait(CLK);
    #1 D=0;
    wait(!CLK);
    wait(CLK);
    D=1;
    wait(!CLK);
    wait(CLK);
    D=1;
end

initial
begin
    #43 force my_FF.Q = 1'b0; // "Q" declared as net in this file
    #2 release my_FF.Q; // "Q" is the flip-flop output
end

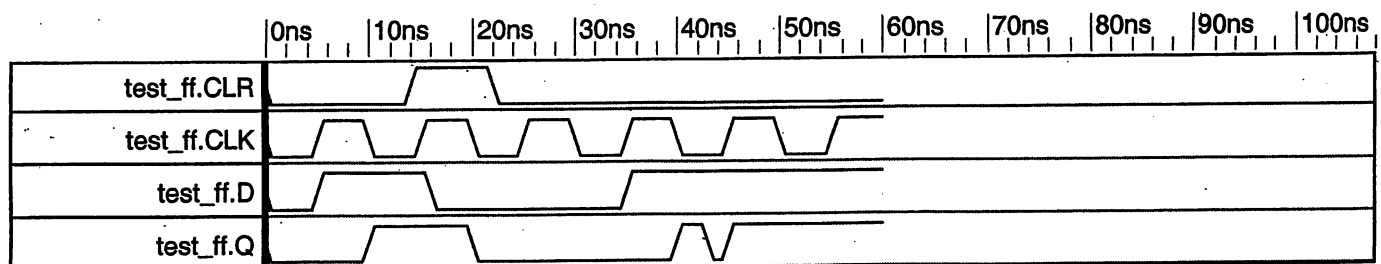
initial
    CLR=0;

always // generate clock signal
    #5 CLK= ~CLK;

initial
    #60 $finish;

endmodule

```



```

module test_ff; // testbench
reg CLR,CLK,D;
wire Q;

// instantiate device

dff my_FF(.clr(CLR), .q(Q), .clock(CLK), .d(D));

initial
    CLR=0;

initial // define stimuli
begin
    D=0;
    wait(CLK);
    D=1; // input for 1st clock
    wait(!CLK);
    #4 CLR = 1;
    wait(CLK);
    #1 D=0; // input for 2nd clock
    wait(!CLK);
    #2 CLR=0;
    wait(CLK);
    #1 D=0;
    wait(!CLK);
    wait(CLK);
    D=1;
    wait(!CLK);
    wait(CLK);
    D=1;
end

initial
begin
    #43 force my_FF.q = 1'b0; // "q" declared as reg in D FF template
    #2 release my_FF.q;
end

initial
    CLK=0;

always // generate clock signal
    #5 CLK= ~CLK;

initial
    #60 $finish;

endmodule

```

