

```

-----
-- Test Bench for JK flip-flop (ESD 2.3.1)
--
-- we use another process to offer the concurrent clock signal
-----

library ieee;
use ieee.std_logic_1164.all;

entity jkff_TB is
end jkff_TB;
-- entity declaration

-----

architecture TB of jkff_TB is

    signal T_J, T_K:    std_logic;
    signal T_clock:    std_logic;
    signal T_reset:    std_logic;
    signal T_Q, T_Qbar:    std_logic;

    component JK_FF is
    port (
        clock:    in std_logic;
        J, K:    in std_logic;
        reset:    in std_logic;
        Q, Qbar:    out std_logic
    );
    end component;

begin

    U_JKFF: JK_FF port map (T_clock, T_J, T_K, T_reset, T_Q, T_Qbar);

    -- concurrent process to offer clock signal
    process
    begin
        T_clock <= '0';
        wait for 5 ns;
        T_clock <= '1';
        wait for 5 ns;
    end process;

    process

        variable err_cnt: integer := 0;

    begin

        T_reset <= '1';
        wait for 25 ns;

        T_reset <= '0';
        wait for 10 ns;

        -- case 1
        T_J <= '0';
        T_K <= '1';

```

```

wait for 15 ns;
assert (T_Q='0') report "Error1!" severity error;
if (T_Q/='0') then
    err_cnt := err_cnt + 1;
end if;

-- case 2
wait for 5 ns;
T_J <= '1';
T_K <= '0';
wait for 15 ns;
assert (T_Q='1') report "Error2!" severity error;
if (T_Q/='0') then
    err_cnt := err_cnt + 1;
end if;

-- case 3
wait for 5 ns;
T_J <= '1';
T_K <= '1';
wait for 15 ns;
assert (T_Q='0') report "Error3!" severity error;
if (T_Q/='0') then
    err_cnt := err_cnt + 1;
end if;

-- summary of all the tests
if (err_cnt=0) then
    assert false
    report "Testbench of Adder completed successfully!"
    severity note;
else
    assert true
    report "Something wrong, try again"
    severity error;
end if;

wait;

end process;
end TB;

-----
configuration CFG_TB of jkff_TB is
    for TB
        end for;
end CFG_TB;
-----

```