Testbench for Sequential Circuit

Sequential circuit testbenches are similar to those for combinational circuits but include additional inputs like Clock and Reset. Clock signals require a separate process statement, while the reset signal can be configured as needed.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity up_down_counter is
    port (
        clk : in std_logic;
        rst : in std_logic;
        up_down : in std_logic;
        count : out std_logic_vector(3 downto 0)
    );
end up_down_counter;

architecture rtl of up_down_counter is
begin
    process(clk, rst)
    begin
        if rst = '1' then
            count <= "0000";
        elsif rising_edge(clk) then
            if up_down = '1' then
                count <= count + 1;
            else
                count <= count - 1;
            end if;
        end if;
    end process;
end rtl;
```

In this case, a synchronous up/down counter is tested with a testbench combining all three architecture models. The Clock uses a process statement, and Reset uses a simple assignment. Inputs are declared upfront as they do not change during the simulation.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity testbench is
end testbench;

architecture tb_arch of testbench is
    signal clk : std_logic := '0';
    signal rst : std_logic := '0';
    signal up_down : std_logic := '0';
    signal count : std_logic_vector(3 downto 0);

    component up_down_counter
      port (
        clk : in std_logic;
        rst : in std_logic;
        up_down : in std_logic;
        count : out std_logic_vector(3 downto 0)
      );
    end component;

begin
    UUT_inst : up_down_counter
      port map (
        clk => clk,
        rst => rst,
        up_down => up_down,
        count => count
      );

    -- Clock process
    clk_process : process
    begin
        clk <= not clk;
        wait for 10 ns;
    end process;
```

```
    -- Apply stimulus to the UUT
    up_down <= '1';
    wait for 10 ns;
    up_down <= '0';
    wait for 10 ns;
    rst <= '1';
    wait for 10 ns;
    rst <= '0';
    wait for 10 ns;
    wait;


 end tb_arch;
```