

EMMAN ACE G. MENION
BSCPE 3A

VHDL CODE FOR JK FLIP FLOP WITH ASYNCHRONOUS RESET: (Master Slave JK Flip-Flop)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity practice is
    Port (
        j  : in STD_LOGIC;
        k  : in STD_LOGIC;
        clk : in STD_LOGIC;
        reset : in STD_LOGIC; -- Active-low reset
        Q   : out STD_LOGIC
    );
end practice;

architecture Behavioral of practice is
    signal counter : unsigned(22 downto 0) := (others => '0');
    signal slow_clk : STD_LOGIC := '0';
    signal q_reg : STD_LOGIC := '0';
    signal jk_input : STD_LOGIC_VECTOR(1 downto 0); -- For case statement
begin

    -- Clock Divider
    process(clk)
    begin
        if rising_edge(clk) then
            counter <= counter + 1;
        end if;
    end process;

    slow_clk <= counter(22); -- Slow down clock

    -- JK Flip-Flop Logic
    process(slow_clk, reset)
    begin
        if reset = '0' then
            q_reg <= '0'; -- Asynchronous reset
        elsif rising_edge(slow_clk) then
            jk_input <= j & k; -- Combine JK into 2-bit vector
            case jk_input is
                when "00" => null; -- No change
                when "01" => q_reg <= '0'; -- Reset
                when "10" => q_reg <= '1'; -- Set
                when "11" => q_reg <= not q_reg; -- Toggle
                when others => null;
            end case;
        end if;
    end process;
```

```

end if;
end process;

```

```

Q <= q_reg;

```

```

end Behavioral;

```

The screenshot displays the Quartus Prime IDE interface. The main window shows the VHDL code for 'practice.vhd'. The code implements a behavioral model of a JK flip-flop. It includes a clock divider process and a JK flip-flop logic process. The compilation messages window at the bottom shows the results of the compilation, including a warning about the number of processors and a successful full compilation.

VHDL Code (practice.vhd):

```

14 architecture Behavioral of practice is
15     signal counter : unsigned(22 downto 0) := (others => '0');
16     signal slow_clk : STD_LOGIC := '0';
17     signal q_reg : STD_LOGIC := '0';
18     signal jk_input : STD_LOGIC_VECTOR(1 downto 0); -- For case statement
19 begin
20     -- Clock Divider
21     process(clk)
22     begin
23         if rising_edge(clk) then
24             counter <= counter + 1;
25         end if;
26     end process;
27     slow_clk <= counter(22); -- slow down clock
28
29     -- JK Flip-Flop Logic
30     process(slow_clk, reset)
31     begin
32         if reset = '0' then
33             q_reg <= '0'; -- Asynchronous reset
34         elsif rising_edge(slow_clk) then
35             jk_input <= j & k; -- Combine JK into 2-bit vector
36             case jk_input is
37                 when "00" => null; -- No change
38                 when "01" => q_reg <= '0'; -- Reset
39                 when "10" => q_reg <= '1'; -- Set
40                 when "11" => q_reg <= not q_reg; -- Toggle
41             end case;
42         end if;
43     end process;
44     Q <= q_reg;
45 end Behavioral;

```

Compilation Messages:

Type	ID	Message
Information	1	Running Quartus Prime EDA Netlist Writer
Information	2	Command: quartus_eda --read_settings_files=off --write_settings_files=off practice -c practice
Warning	18236	Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment
Information	204019	Generated file practice.vo in folder "c:/intelFPGA_lite/23.1std/simulation/questa/" for EDA simulation tool
Information	293000	Quartus Prime EDA Netlist writer was successful. 0 errors, 1 warning
Information	293000	Quartus Prime Full compilation was successful. 0 errors, 16 warnings