CODIGO

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

entity state_machine is
    Port ( clk  : in    STD_LOGIC;
           x    : in    STD_LOGIC;
           z    : out   STD_LOGIC);
end state_machine;

architecture Behavioral of state_machine is
    type estados is (q0,q1,q2,q3,q4);
    signal edo_presente, edo_futuro: estados;
begin

    proceso_1: process (edo_presente, x)
    begin
        case edo_presente is
            when q0 => z <= '0';
                if x = '1' then
                    edo_futuro <= q1;
                else
                    edo_futuro <= q4;
                end if;
            when q1 => z <= '0';
                if x = '1' then
                    edo_futuro <= q4;
                end if;
            when q2 =>
                if x = '1' then
                    edo_futuro <= q3;
                    z <= '1';
                else
                    edo_futuro <= q4;
                    z <= '0';
                end if;
            when q3 => z <= '0';
                if x = '1' then
                    edo_futuro <= q3;
                else
                    edo_futuro <= q3;
                end if;
            when q4 => z <= '0';
                if x = '1' then
                    edo_futuro <= q1;
                else
                    edo_futuro <= q4;
                end if;
        end case;
    end process proceso_1;
```

```
proceso_2: process (clk)
begin
    if (clk'event and clk='1') then
        edo_presente <= edo_futuro;
    end if;
end process proceso_2;
```

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

entity clk_div is
    Port ( clk_in : in STD_LOGIC;
           rst : in STD_LOGIC;
           clk_out : out STD_LOGIC);
end clk_div;

architecture Behavioral of clk_div is
    signal temp: std_logic;
    signal contador: std_logic_vector(25 downto 0) := (others => '0');
begin
    divisor: process (rst, clk_in)
    begin
        if (rst = '1') then
            temp <= '0';
            contador <= (others => '0');
        elsif rising_edge(clk_in) then
            if (contador = 3) then
                temp <= not(temp);
                contador <= (others => '0');
            else
                contador <= contador + 1;
            end if;
        end if;
    end process;
    clk_out <= temp;

end Behavioral;
```

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

entity state_machine_top is
    Port ( clk  : in    STD_LOGIC;
           rst : in STD_LOGIC;
           x    : in    STD_LOGIC;
           z    : out   STD_LOGIC);
end state_machine_top;

architecture Behavioral of state_machine_top is
COMPONENT state_machine
    Port ( clk  : in    STD_LOGIC;
           x    : in    STD_LOGIC;
           z    : out   STD_LOGIC);
END COMPONENT;
COMPONENT clk_div
    Port ( clk_in : in STD_LOGIC;
           rst : in STD_LOGIC;
           clk_out : out STD_LOGIC);
END COMPONENT;
SIGNAL  clk_out : STD_LOGIC;
begin
    C_clock: clk_div
    Port map( clk_in => clk,
              rst => rst,
              clk_out => clk_out);
    C_machine: state_machine
    Port map( clk => clk_out,
              x => x,
              z => z);
end Behavioral;
```

TEST BENCH

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

entity state_machine_tb is
--  Port ( );
end state_machine_tb;

architecture Behavioral of state_machine_tb is
    COMPONENT state_machine_top
        Port (  clk  : in STD_LOGIC;
                rst : in STD_LOGIC;
                x    : in    STD_LOGIC;
                z    : out   STD_LOGIC);
    END COMPONENT;
    SIGNAL  clk  : STD_LOGIC := '0';
    SIGNAL  rst  : STD_LOGIC := '1';
    SIGNAL  x  : STD_LOGIC;
    SIGNAL  z  : STD_LOGIC;
begin
    DUT: state_machine_top
    Port map(   clk => clk,
                rst => rst,
                x => x,
                z => z);
    estimulos_clk: process

    begin
        clk <= '1';
        wait for 10 ns;
        clk <= '0';
        wait for 10 ns;
    end process;
    estimulos_rst: process
    begin
        wait for 40 ns;
        rst <= '0';
    end process;
    estimulos: process
    begin
        x <= '1';
        wait for 160 ns;
        x <= '0';
        wait for 160 ns;
        x <= '0';
        wait for 160 ns;
        x <= '1';
        wait for 160 ns;
        x <= '1';
        wait for 160 ns;
        x <= '1';
        wait for 160 ns;
        x <= '0';
        wait for 160 ns;

    end process;
end Behavioral;
```