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

entity usr1 is
   Port (
        clk : in  STD_LOGIC;
        clk_div : inout std_logic;
        rst : in  STD_LOGIC;
        load: in  STD_LOGIC;
        mode : in  STD_LOGIC_VECTOR(1 downto 0);  -- mode selection input
        si : in  STD_LOGIC;  -- serial input
        pi : in  STD_LOGIC_VECTOR(3 downto 0);  -- parallel input
        so : out  STD_LOGIC;  -- serial output
        po : out  STD_LOGIC_VECTOR(3 downto 0)  -- parallel output
    );
end usr1;

architecture Behavioral of usr1 is
    signal shift_reg : STD_LOGIC_VECTOR(3 downto 0);
    signal counter: std_logic_vector(26 downto 0):=( others=>'0');
begin
    process(clk_div, rst,load)
    begin
        if rst = '1' then
            shift_reg <= (others => '0');
        elsif clk_div'event and clk_div= '1' then
            case mode is
                when "00" =>  -- SISO mode
                    shift_reg( 3 downto 1) <= shift_reg(2 downto 0);
                    shift_reg(0)<=si;
                    so <= shift_reg(3);
                when "01" =>  -- SIPO mode
                    shift_reg( 3 downto 1) <= shift_reg(2 downto 0);
                    shift_reg(0)<=si;
                     po <= shift_reg;
               when "10" =>  -- PISO mode
                       if(load='0') then
                                 shift_reg <= pi;
                        else
                            shift_reg( 3 downto 1) <= shift_reg(2 downto 0);
                        end if;
                        so <= shift_reg(3);
                when "11" =>  -- PIPO mode
                    po<= pi;

                when others =>
                    null;
            end case;
        end if;
    end process;
```

```
   -- so <= shift_reg(3);
   -- po <= shift_reg;

   process(clk)
    begin
        if clk'event and clk = '1' then
            if rst = '1' then
                counter <= (others => '0');
            else
                counter <= counter + '1';
            end if;
        end if;
    end process;

clk_div<= counter(0);

end Behavioral;
```