

## Output

### Universal Shift Register

#### Program for UNIVERSAL SHIFT

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity uni_shift1 is
    port(reset, si, clk, load : in STD_LOGIC;
    pi :in STD_LOGIC_VECTOR ( 3 downto 0);
    po: out STD_LOGIC_VECTOR ( 3 downto 0);
    mode : in STD_LOGIC_VECTOR ( 1 downto 0);
    so : out STD_LOGIC
    );
end uni_shift1;
architecture Behavioral of uni_shift1 is
    signal temp: std_logic_vector(3 downto 0):="0000";
    signal clk_s: std_logic_vector(23 downto 0):=(others=>'0');
begin
    --Count<= clk_s(22);
    Process(clk)
    Begin
        If rising_edge (clk) then
            Clk_s<=clk_s+ '1';
        End if;
    End process;--
    Process(clk_s(22), mode, si, load, reset)
    Begin
        If reset='1' then temp<="0000"; po<="0000"; so<='0';
        Elsif rising_edge(clk_s(22)) then
            Case mode is
                --SISO
```

When "00"=> temp(3 downto 1) <= temp(2 downto 0);

Temp(0)<=si;

so<=temp(3);

--SIPO

When "01"=> temp(0)<=si;

temp(3 downto 1) <= temp(2 downto 0);

po<=temp;

--PIPO

When "10"=> if load='1' then

temp<=pi;

else

po<= temp;

end if;

--PISO

When "11"=> if(load='1') then

Temp(3 downto 0)<=pi(3 downto 0);

Else

so<=temp(3);

temp(3 downto 1)<= temp(2 downto 0);

temp(0)<='0';

end if;

when others=> null;

end case;

end if;

end process;

end behavioral;

## UCF Shift reg :

NET "clk" LOC= "p181";

NET "load" LOC= "p57";

NET "mode<0>" LOC= "p52";

NET "mode<1>" LOC= "p51";

NET "pi<0>" LOC= "p43";

NET "pi<1>" LOC= "p42";

NET "pi<2>" LOC= "p40";

NET "pi<3>" LOC= "p37";

NET "po<0>" LOC= "p80";

NET "po<1>" LOC= "p79";

NET "po<2>" LOC= "p78";

NET "po<3>" LOC= "p77";

NET "reset" LOC= "p50";

NET "si" LOC= "p34";

NET "so" LOC= "p58";

