4-bit Universal shift using VHDL and implement on FPGA:

VHDL Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_arith.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
entity usr is
   Port ( Clk,rst,sin : in  STD_LOGIC;
       m : in  STD_LOGIC_VECTOR (1 downto 0);
       pin : in  STD_LOGIC_VECTOR (3 downto
0);
        sout : out  STD_LOGIC;
        pout : out  STD_LOGIC_VECTOR (3
downto 0));
end usr;
architecture Behavioral of usr is
signal p1,p2:std_logic_vector(3 downto 0);
signal s1:std_logic;
signal temp:std_logic_vector(22 downto 0);
signal sclk:std_logic;
begin
p11: process(clk,rst)
           begin
           if(rst='1')then temp<=(others=>'0');
                     elsif(clk'event and clk='1')
then
                     temp<= temp + 1;
                     end if;
           end process;
           sclk<=temp(22);
p22:process(sclk,rst,pin,sin,m)
begin
if(rst='1') then
    p1<="0000";
    p2<="0000";
           s1<='0';
elsif (sclk'event and sclk='1') then
           if(m="00") then     --SISO
           p1(0)<=sin;
           p1(1)<=p1(0);
           p1(2)<=p1(1);
           p1(3)<=p1(2);
           s1<=p1(3);
elsif (m="01") then   --SIPO
          p1(0)<=sin;
           p1(1)<=p1(0);
           p1(2)<=p1(1);
           p1(3)<=p1(2);
             p2<=p1;
elsif(m="10") then   --PIPO
             p1<=pin;
             p2<=p1;
else              --PISO
      p1<=pin;
      s1<=p1(3);
end if;
end if;
end process;
pout<=p2;
sout<=s1;
end Behavioral;
```

UCF File:

NET "CLK" LOC = "L15";

onBoardLeds

NET "pout<0>" LOC = "U18";

NET "pout<1>" LOC = "M14";

NET "pout<2>" LOC = "N14";

NET "pout<3>" LOC = "L14";

NET "Sout" LOC = "M13";

onBoard SWITCHES

NET "pin<0>" LOC = "A10";

NET "pin<1>" LOC = "D14";

NET "pin<2>" LOC = "C14";

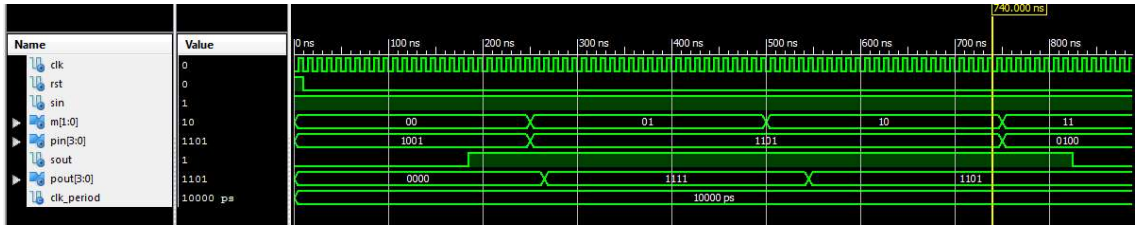
NET "pin<3>" LOC = "P15";

NET "sin" LOC = "P12";

NET "rst" LOC = "R5";

NET "m<0>" LOC = "T5";

NET "m<1>" LOC = "E4



Impletation on FPGA Board: