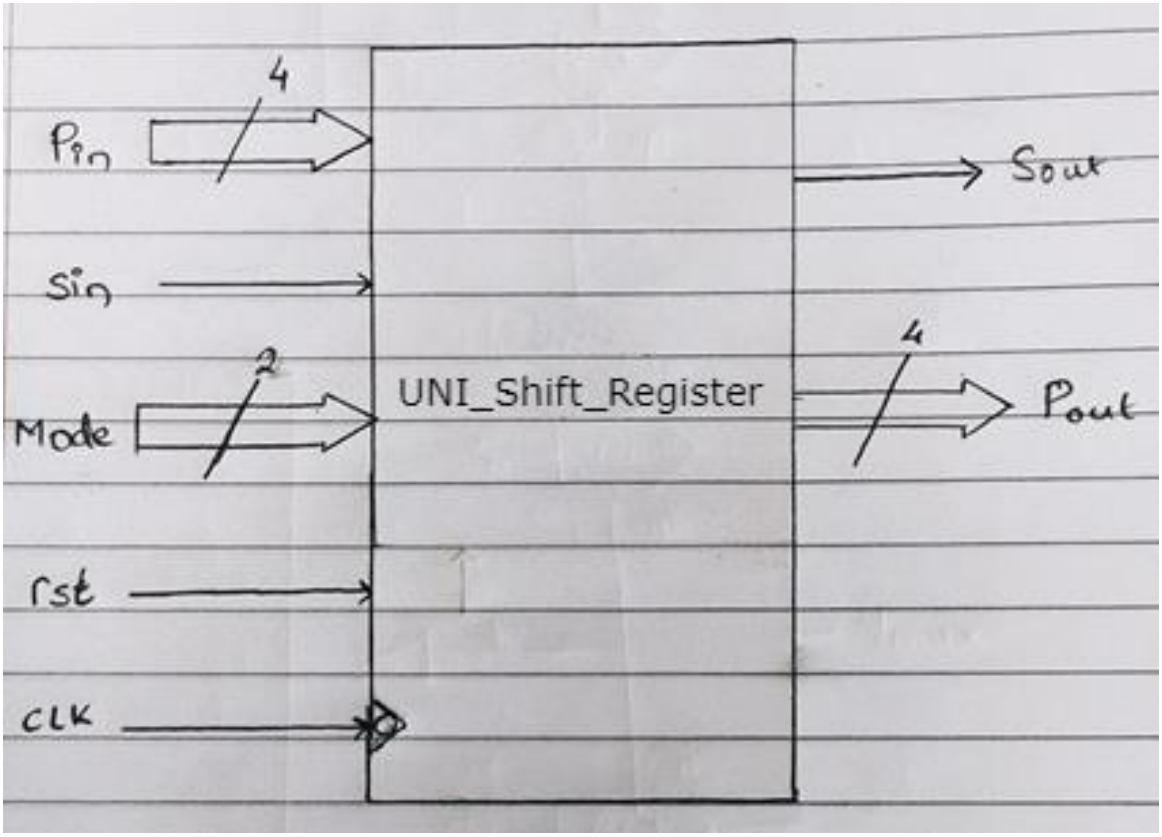
BLOCK DIAGRAM



FUNCTION TABLE

rst	clk	MODE		Output
		M(1)	M(0)	
1	X	X	X	X
0	↓	0	0	Serial In Serial Out (SISO)
0	↓	0	1	Serial In Parallel Out (SIPO)
0	↓	1	0	Parallel In Serial Out (PISO)
0	↓	1	1	Parallel In Parallel Out (PIPO)

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

entity USR_4b is
    Port ( rst : in  STD_LOGIC;
           clk : in  STD_LOGIC;
           mode : in  STD_LOGIC_VECTOR (1
downto 0);
           Sin : in  STD_LOGIC;
           Pin : in  STD_LOGIC_VECTOR (3
Downto 0);
           Sout : out  STD_LOGIC;
           Pout : out  STD_LOGIC_VECTOR
(3 downto 0));
end USR_4b;

architecture USR_4b_arch of USR_4b is
    SIGNAL temp : STD_LOGIC_VECTOR(3
DOWNTO 0):="0000" ;
    SIGNAL flag : STD_LOGIC:='0';

begin
        PROCESS(rst, clk, mode, Sin, Pin)
```

```
        BEGIN
                IF rst='1' THEN
                        Sout <= '0';
                        Pout <= "0000";
                        flag <= '0';

                ELSIF falling_edge(clk)
THEN
                        CASE MODE IS
                                WHEN "00"
=>

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

temp(0) <= Sin;

Sout <= temp(3);

Pout <="0000";


flag <= '0';
                                WHEN "01"
=>
```

```
temp(3 DOWNTO 1) <= temp(2 DOWNTO 0);

temp(0) <= Sin;

Pout<=temp;

Sout <= '0';

flag <= '0';
                                        WHEN "10"
=>


IF flag='0' THEN

  temp <= Pin;

  Pout <= "0000";

ELSE

  Sout <= temp(3);

  Pout <= "0000";
```

```
  temp(3 DOWNTO 1) <= temp(2 DOWNTO 0);
END IF;
flag <= '1';

                                        WHEN
OTHERS =>
temp <= Pin;
Pout <= temp;
Sout <= '0';
flag <= '0';
                            END CASE;
                    END IF;
          END PROCESS;

end USR_4b_arch;
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY UNI_4b_TB IS
END UNI_4b_TB;

ARCHITECTURE behavior OF UNI_4b_TB IS
     -- Component Declaration for the
Unit Under Test (UUT)

    COMPONENT UNI_4b
    PORT(
         Pin : IN  std_logic_vector(3
downto 0);
         Sin : IN  std_logic;
         Mode : IN  std_logic_vector(1
downto 0);
         RST : IN  std_logic;
         CLK : IN  std_logic;
         Sout : OUT  std_logic;
         Pout : OUT  std_logic_vector(3
downto 0)
        );
    END COMPONENT;
```

```
        --Inputs
   signal Pin : std_logic_vector(3 downto
0) := "0000";
   signal Sin : std_logic := '0';
   signal Mode : std_logic_vector(1
downto 0) := (others => '0');
   signal RST : std_logic := '0';
   signal CLK : std_logic := '1';
        --Outputs
   signal Sout : std_logic;
   signal Pout : std_logic_vector(3
downto 0);

   -- Clock period definitions
   constant CLK_period : time := 10 ns;

BEGIN
        -- Instantiate the Unit Under
Test (UUT)
   uut: UNI_4b PORT MAP (
          Pin => Pin,
          Sin => Sin,
          Mode => Mode,
          RST => RST,
          CLK => CLK,
```

```
        Sout => Sout,
        Pout => Pout
      );

-- Clock process definitions
CLK_process :process
begin
              CLK <= '0';
              wait for CLK_period/2;
              CLK <= '1';
              wait for CLK_period/2;
end process;

-- Stimulus process
stim_proc_Mode: process
begin

              Mode <= "00";
              wait for 80 ns;
              Mode <= "01";
              wait for 50 ns;
              Mode <= "10";
              wait for 50 ns;
              Mode <= "11";
              wait for 20 ns;
```

```
   end process;

        stim_proc_Sin: process
        begin
                Sin <= '1';
                wait for 10 ns;
                Sin <= '0';
                wait for 10 ns;
                Sin <= '1';
                wait for 10 ns;
                Sin <= '0';
                wait for 10 ns;

                ---Sin process for 0-40
ns
                                Sin <=
'0';
                wait for 40 ns;
                ---Sin process for 40-80
ns

                Sin <= '1';
                wait for 10 ns;
                Sin <= '0';
                wait for 10 ns;
```

```
                Sin <= '1';
                wait for 10 ns;
                Sin <= '0';
                wait for 10 ns;
                ---Sin process for 80-120
ns

         end process;

        stim_proc_RST: process
        begin
                wait for 122.5 ns;
                RST <= '0';
                wait for 5 ns;
                RST <= '1';
                wait;
        end process;
END;
```