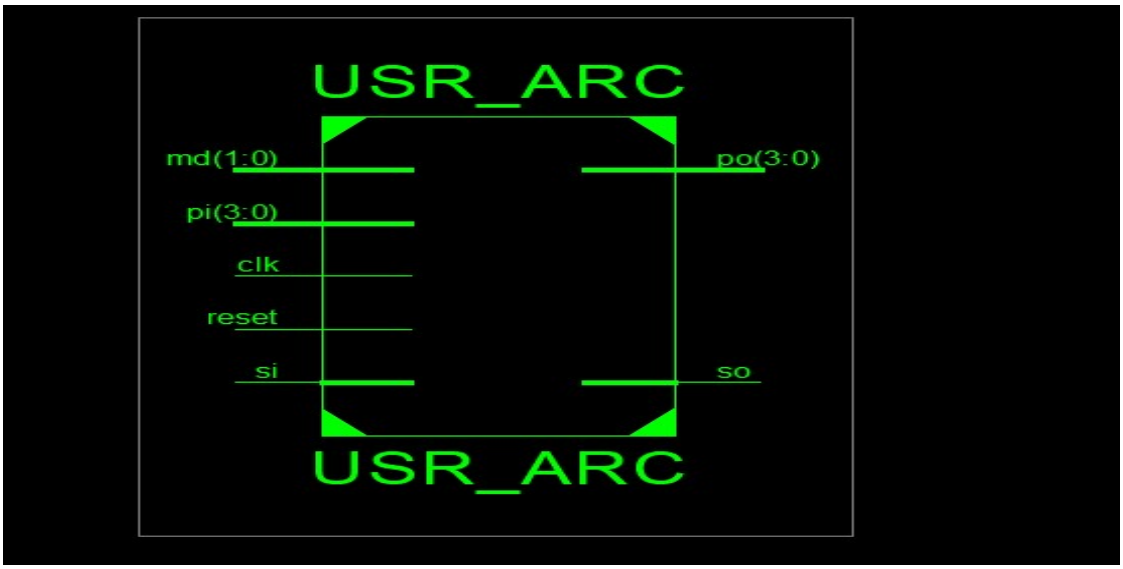
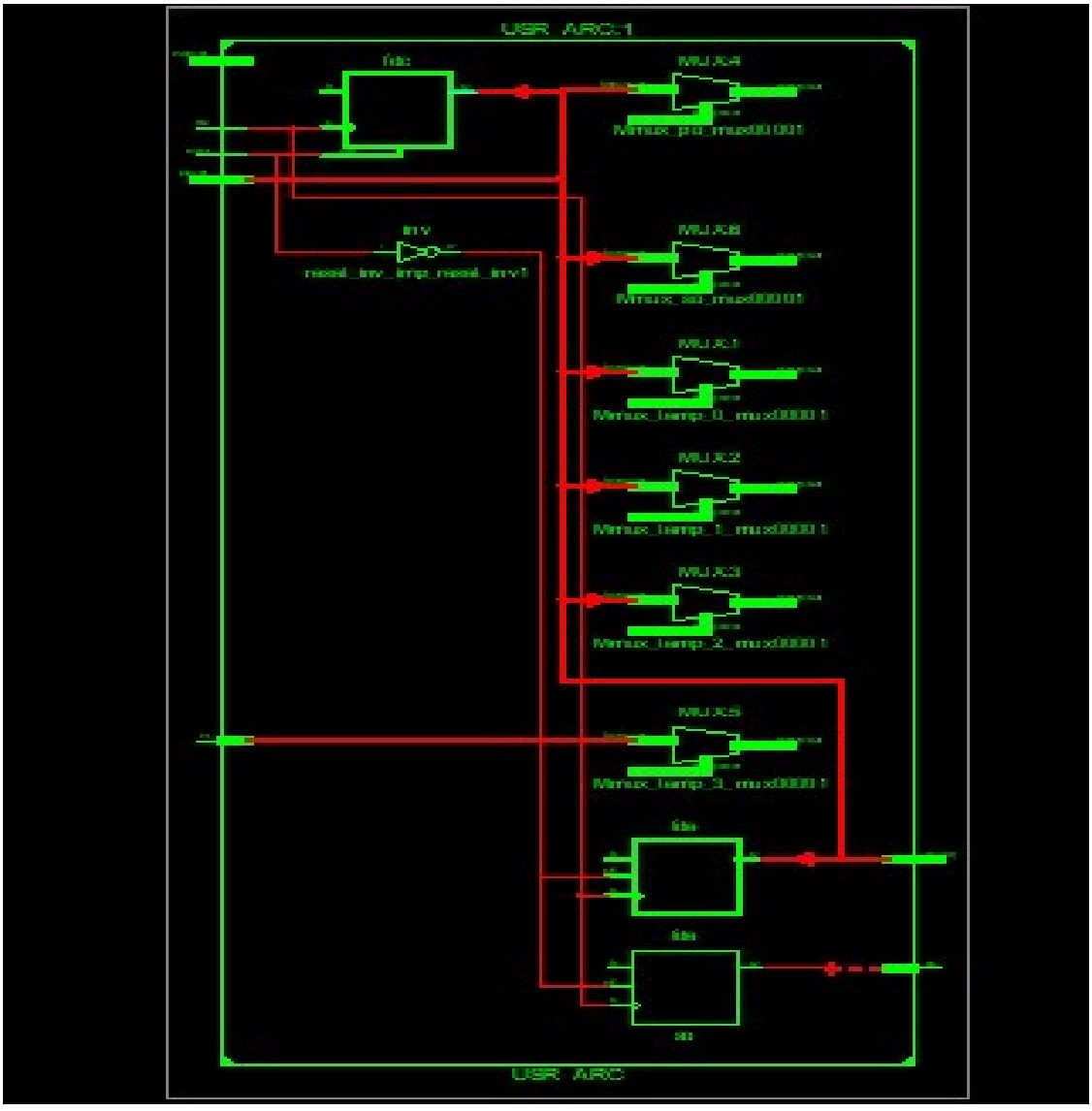
Practical No -: 2

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

entity USR_ARC is
        Port ( si : in     STD_LOGIC;
                    clk : in     STD_LOGIC;
                    pi : in     STD_LOGIC_VECTOR (3 downto 0);
                    md : in     STD_LOGIC_VECTOR (1 downto 0);
                    reset : in     STD_LOGIC;
                    so : out     STD_LOGIC;
                    po : out STD_LOGIC_VECTOR (3 downto 0));
end USR_ARC;

architecture Behavioral of USR_ARC is
signal temp:std_logic_vector(3 downto 0);
begin
process(clk,reset,md)
begin
if(reset='1')then
temp<="0000";
elsif (clk'event and clk='1')then
```

```
case md is
when"00"=> temp(3)<=si;
temp(2 downto 0)<=temp(3 downto 1);
so<=temp(0);

when"01"=> temp(3)<=si;
temp(2 downto 0)<=temp(3 downto 1);
po<=temp;

when "10"=>po<=pi;
when "11"=>temp<=pi;
temp(2 downto 0)<=temp(3 downto 1);
so<=temp(3);

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


USR_ARC
md(1:0)
pi(3:0)
clk
reset
si
po(3:0)
so
USR_ARC


USR_ARC:1
fdc
MUX4
inv
MUX8
MUX1
MUX2
MUX3
MUX5
fdc
fdc
USR_ARC

TESTBENCH -:

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

entity TB_USR_ARC is
end TB_USR_ARC;

architecture behavior of TB_USR_ARC is
    -- Component Declaration
    component USR_ARC
        Port ( si    : in  STD_LOGIC;
               clk   : in  STD_LOGIC;
               pi    : in  STD_LOGIC_VECTOR (3 downto 0);
               md    : in  STD_LOGIC_VECTOR (1 downto 0);
               reset : in  STD_LOGIC;
               so    : out STD_LOGIC;
               po    : out STD_LOGIC_VECTOR (3 downto 0));
    end component;
    -- Signals
    signal si_tb    : STD_LOGIC := '0';
    signal clk_tb   : STD_LOGIC := '0';
    signal pi_tb    : STD_LOGIC_VECTOR (3 downto 0) := (others => '0');
    signal md_tb    : STD_LOGIC_VECTOR (1 downto 0) := "00";
    signal reset_tb : STD_LOGIC := '0';
    signal so_tb    : STD_LOGIC;
    signal po_tb    : STD_LOGIC_VECTOR (3 downto 0);
```

```
    constant clk_period : time := 10 ns;

begin
    -- Instantiate the Unit Under Test (UUT)
    uut: USR_ARC
        Port map (
            si    => si_tb,
            clk   => clk_tb,
            pi    => pi_tb,
            md    => md_tb,
            reset => reset_tb,
            so    => so_tb,
            po    => po_tb
        );
    -- Clock generation
    clk_process : process
    begin
        while true loop
            clk_tb <= '0';
            wait for clk_period/2;
            clk_tb <= '1';
            wait for clk_period/2;
        end loop;
    end process;
    -- Stimulus
    stim_proc: process
    begin
```

```
-- Reset
reset_tb <= '1';
wait for clk_period;
reset_tb <= '0';
wait for clk_period;

-- Mode 10: Parallel Load pi = 1010
pi_tb <= "1010";
md_tb <= "10";
wait for clk_period;

-- Mode 00: Shift right with si
md_tb <= "00";
si_tb <= '1';
wait for clk_period;
si_tb <= '0';
wait for clk_period;
si_tb <= '1';
wait for clk_period;

-- Mode 01: Shift right with parallel output
md_tb <= "01";
si_tb <= '0';
wait for clk_period*3;

-- Mode 11: Load then shift left
pi_tb <= "1100";
```

```
        md_tb <= "11";
        wait for clk_period*4;
        -- End simulation
        wait;
    end process;
end behavior;
```

Time Division -:

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

```
entity USR_ARC is
        Port ( si : in     STD_LOGIC;
                    clk : in     STD_LOGIC;
                    pi : in     STD_LOGIC_VECTOR (3 downto 0);
                    md : in     STD_LOGIC_VECTOR (1 downto 0);
                    reset : in     STD_LOGIC;
                    so : out     STD_LOGIC;
                    po : out STD_LOGIC_VECTOR (3 downto 0));
end USR_ARC;

architecture Behavioral of USR_ARC is
signal temp:std_logic_vector(3 downto 0);
signal count:STD_LOGIC_VECTOR (20 downto 0);
signal clk1:STD_LOGIC;
begin
process(clk)
begin
if clk'event and clk='1' then
count<=count+"0001";
end if;
clk1<=count(20);
end process;
```

```
begin
process(clk1,reset,md)
begin
if(reset='1')then
temp<="0000";
elsif (clk1'event and clk1='1')then
case md is
when"00"=> temp(3)<=si;
temp(2 downto 0)<=temp(3 downto 1);
so<=temp(0);

when"01"=> temp(3)<=si;
temp(2 downto 0)<=temp(3 downto 1);
po<=temp;

when "10"=>po<=pi;
when "11"=>temp<=pi;
temp(2 downto 0)<=temp(3 downto 1);
so<=temp(3);

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