گزارشکار آزمایش ۶

: shift_register

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity shift_register is
  port(
    parallel_in : in std_logic_vector(3 downto 0);
    LR : in std_logic_vector(1 downto 0);
    load : in std_logic;
    rst : in std_logic;
    clock : in std_logic;
    output : out std_logic_vector(3 downto 0)
    );
end shift_register;
architecture Behavioral of shift_register is
signal qout : std_logic_vector ( 3 downto 0);
begin
  process(clock,rst)
  begin
  if rst = '1' then
    qout <= "0000";
   end if;
  if (clock'event and clock='1') then
    if load = '1' then
       qout <= parallel_in;
    end if;
    if load = '0' then
      case LR is
        when "00" =>
          qout <= qout;
```

```
      when "01" => --righ shift arithmetic
        qout(2) <= qout(3);
        qout(1) <= qout(2);
       qout(0) <= qout(1);
      when "10" => -- left shift
        qout(3) <= qout(2);
       qout(2) <= qout(1);
       qout(1) <= qout(0);
       qout(0)<='0';
      when others => -- right shift logical
        qout(2) <= qout(3);
        qout(1) <= qout(2);
       qout(0) <= qout(1);
       qout(3)<='0';
      end case;
   end if;
 end if;
end process;
 output <= qout;
end Behavioral;
```

:delay

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity delay is
          generic ( freq : integer :=2;
           uni : integer := 1);
   port    ( clk_in   : in   std_logic;
           clk_out  : out  std_logic;
```

```
          rst     : in  std_logic);
end delay;
architecture Behavioral of delay is
   signal tmp : std_logic;
   signal cnt : integer range 0 to freq/(uni*2);
begin
          process (clk_in, rst)
          begin
            if rst = '1' then
               cnt <= 0;
               tmp <= '0';
        elsif ( clk_in'event and clk_in = '1') then
               if (cnt < freq/(uni*2)) then
                    cnt <= cnt + 1;
          else
                    tmp <= not tmp;
                    cnt <= 0;
              end if;
           end if;
    end process;
clk_out <= tmp;
end Behavioral;
```

:seven_seg

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity seven_seg is
Port ( inp : in STD_LOGIC_VECTOR (3 downto 0);
clk : in  STD_LOGIC;
```

```
o : out STD_LOGIC_VECTOR (6 downto 0));
end seven_seg;
architecture Behavioral of seven_seg is
 begin
process (clk)
        begin
    if (clk'event and clk='1') then
            case inp is
                when "0000" =>
                        o<= "1111110"; -- 0
                when "0001" =>
                        o <= "0110000"; -- 1
                when "0010" =>
                        o <= "1101101"; -- 2
                when "0011" =>
                        o <= "1111001"; -- 3
                when "0100" =>
                        o <= "0110011"; -- 4
                when "0101" =>
                        o <= "1011011"; -- 5
                when "0110" =>
                        o <= "1011111"; -- 6
                when "0111" =>
                        o <= "1110001"; -- 7
                when "1000" =>
                        o <= "1111111"; -- 8
                when "1001" =>
                        o <= "1111011"; -- 9
                when "1010" =>
```

```
                o <= "1111101"; -- a
            when "1011" =>
                o <= "0011111"; -- b
            when "1100" =>
                o <= "1001110"; -- c
            when "1101" =>
                o <= "0111101"; -- d
            when "1110" =>
                o <= "1001111"; -- e
            when "1111" =>
                o <= "1000111"; -- f
            when others =>
                o <= "0000000"; --null
        end case;
    end if;
end process;
end Behavioral;
```

:main

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity main is
    Port ( clk : in  STD_LOGIC;
           rst : in  STD_LOGIC;
           din : in  STD_LOGIC_VECTOR(3 downto 0);
           lr : in std_logic_vector(1 downto 0);
           load : in  STD_LOGIC;
           res : out  STD_LOGIC_VECTOR(6 downto 0));
end main;
```

```
architecture Behavioral of main is
            component delay is
    port    ( clk_in   : in   std_logic;
              clk_out  : out  std_logic;
              rst      : in  std_logic);
             end component;
             component shift_register is
Port ( parallel_in : in std_logic_vector(3 downto 0);
         LR : in std_logic_vector(1 downto 0);
         load : in std_logic;
         rst : in std_logic;
        clock : in std_logic;
        output : out std_logic_vector(3 downto 0));
             end component;
             component seven_seg is
    Port ( inp : in STD_LOGIC_VECTOR (3 downto 0);
             clk : in  STD_LOGIC;
             o : out STD_LOGIC_VECTOR (6 downto 0));
             end component;
             Signal tmpClk: STD_LOGIC;
             Signal tmpData: STD_LOGIC_VECTOR(3 downto 0);
             begin
             shift_reg:shift_register port map (din,lr,load,rst,clk,tmpData);
             del: delay port map (clk, tmpClk, rst);
             display: seven_seg port map (tmpData, tmpclk, res);
end Behavioral;
```

: main_tb

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY main_tb IS
END main_tb;
ARCHITECTURE test OF main_tb IS
   -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT main
   PORT(
        clk : IN  std_logic;
        rst : IN  std_logic;
        din : IN  std_logic_vector(3 downto 0);
        lr : IN  std_logic_vector(1 downto 0);
        load : IN  std_logic;
        res : OUT  std_logic_vector(6 downto 0)
       );
   END COMPONENT;
  --Inputs
  signal clk : std_logic ;
  signal rst : std_logic ;
  signal din : std_logic_vector(3 downto 0) ;
  signal lr : std_logic_vector(1 downto 0) ;
  signal load : std_logic ;
          --Outputs
  signal res : std_logic_vector(6 downto 0);
BEGIN
          -- Instantiate the Unit Under Test (UUT)
  uut: main PORT MAP (
        clk => clk,
        rst => rst,
```

```
        din => din,
        lr => lr,
        load => load,
        res => res
      );
din <= "1010";
 clk <= '0' ,'1' after 25 ns ,'0' after 50 ns , '1' after 75 ns ,
  '0'  after 100 ns ,'1' after 125 ns, '0' after 150 ns ,'1' after 175 ns ,'0' after 200 ns ,'1' after 225 ns ,'0' after 250 ns , '1' after 275 ns , '0' after 300 ns ,'1' after 325 ns ,'0' after 350 ns ,'1' after 375 ns,0' after 400 ns ,'1' after 425 ns ,'0' after 450 ns ,'1' after 475 ns;
 rst <= '1' ,'0' after 25 ns;
 lr <= "00"  , "01" after 475 ns;
 load <='0' ,'1' after 225 ns ,'0' after 250 ns;
END;
```

خروجی شبیه سازی main_tb :