INSTITUTO TECNOLOGICO DE ESTUDIOS SUPERIORES DE MONTERREY

LABORATORIO DE SISTEMAS DIGITALES AVANZADOS

PROF. EMILIO CABALLERO

HECTOR JAVIER PEQUEÑO CHAIREZ

A01246364

Registro de corrimiento 8 bits bidireccional 74LS194.

LINKS GITHUB

FUENTE: REGISTRO 4 BITS
https://github.com/HectorPequeno06/HectorPequeno06/blob/master/ShiftRegister8/shiftregister4bits.vhdl

TEST BENCH REGISTRO 4 BITS

https://github.com/HectorPequeno06/HectorPequeno06/blob/master/ShiftRegister8/shiftregister4bitsTB.vhdl

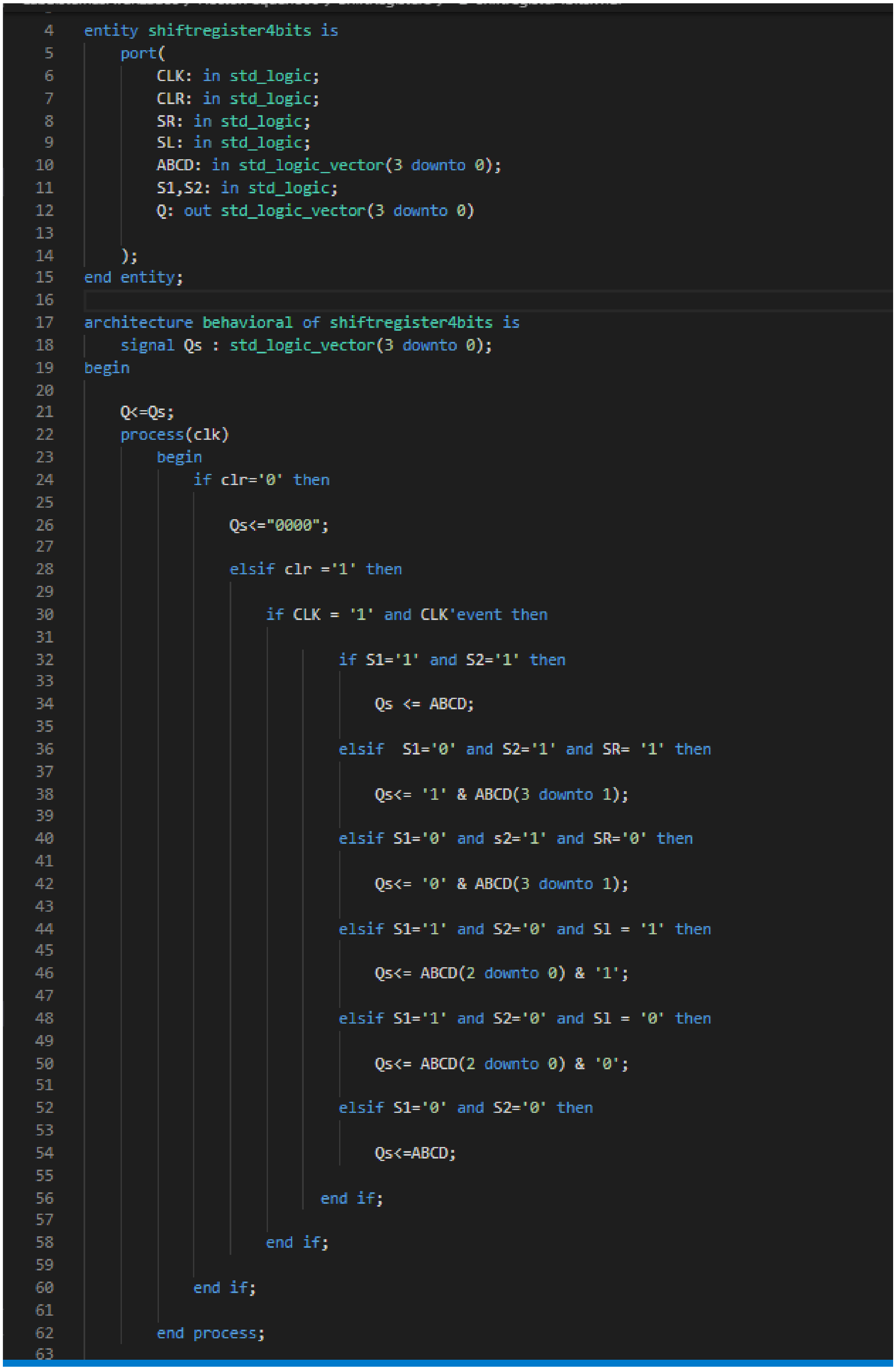
FUENTE: REGISTRO DE 8 BITS

https://github.com/HectorPequeno06/HectorPequeno06/blob/master/ShiftRegister8/shiftregister8bits.vhdl

TEST BENCH REGISTRO 8 BITS

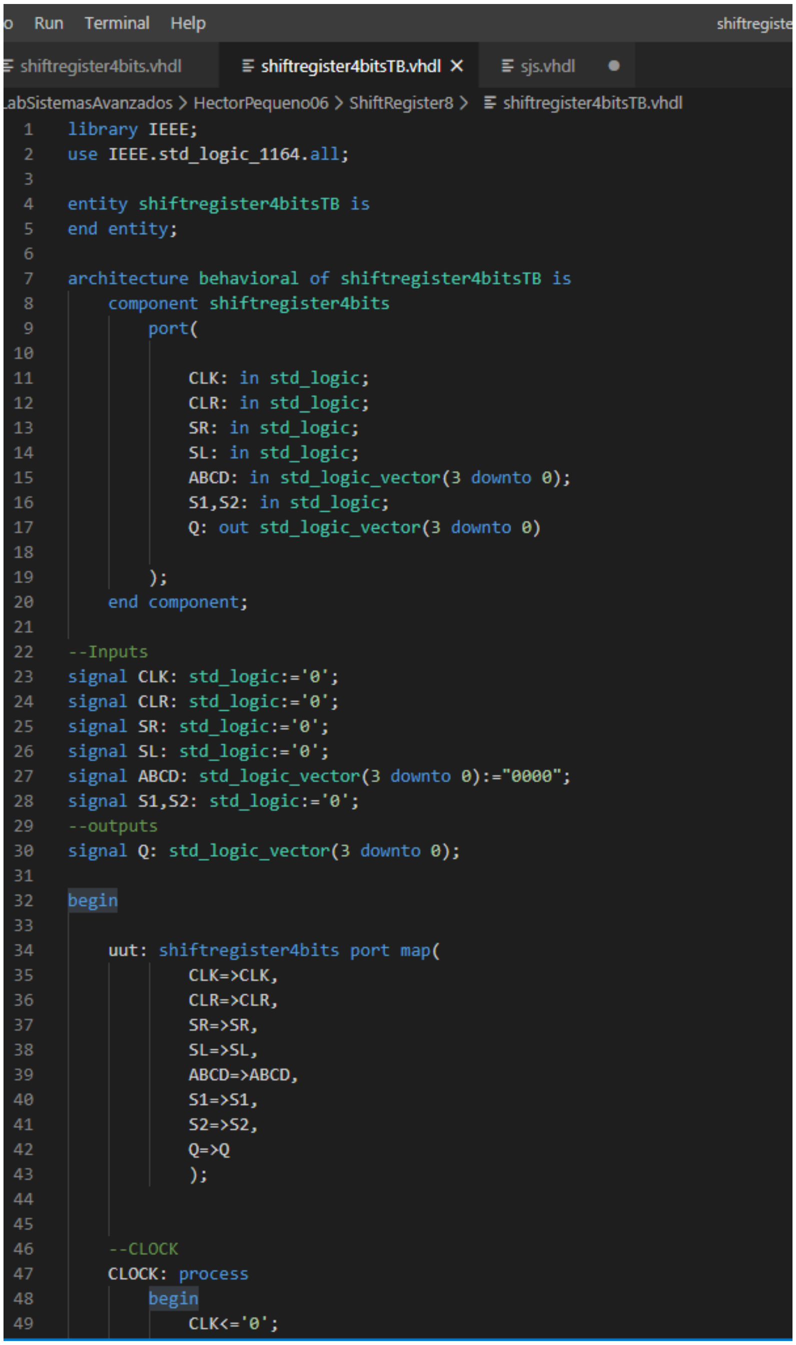
https://github.com/HectorPequeno06/HectorPequeno06/blob/master/ShiftRegister8/shiftregister8bitsTB.vhdl

Código Fuente Registro de 4 Bits:



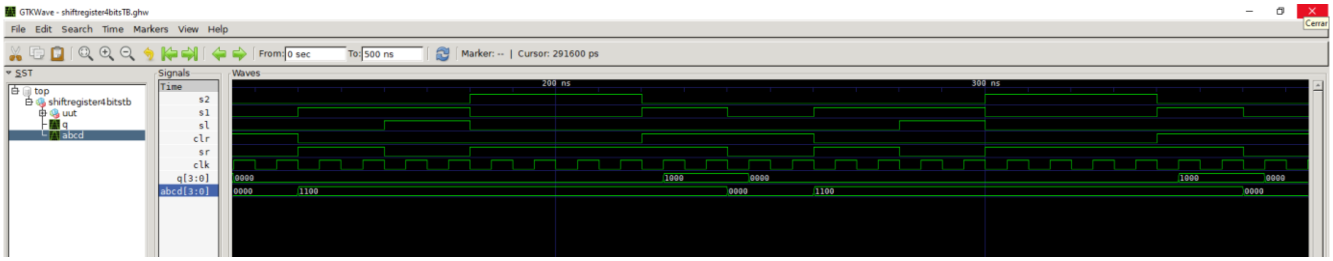
```
entity shiftregister4bits is
    port(
        CLK: in std_logic;
        CLR: in std_logic;
        SR: in std_logic;
        SL: in std_logic;
        ABCD: in std_logic_vector(3 downto 0);
        S1,S2: in std_logic;
        Q: out std_logic_vector(3 downto 0)

    );
end entity;

architecture behavioral of shiftregister4bits is
    signal Qs : std_logic_vector(3 downto 0);
begin

    Q<=Qs;
    process(clk)
        begin
            if clr='0' then

                Qs<="0000";

                elsif clr ='1' then

                    if CLK = '1' and CLK'event then

                            if S1='1' and S2='1' then

                                Qs <= ABCD;

                            elsif  S1='0' and S2='1' and SR= '1' then

                                Qs<= '1' & ABCD(3 downto 1);

                            elsif S1='0' and s2='1' and SR='0' then

                                Qs<= '0' & ABCD(3 downto 1);

                            elsif S1='1' and S2='0' and Sl = '1' then

                                Qs<= ABCD(2 downto 0) & '1';

                            elsif S1='1' and S2='0' and Sl = '0' then

                                Qs<= ABCD(2 downto 0) & '0';

                            elsif S1='0' and S2='0' then

                                Qs<=ABCD;

                          end if;

                    end if;

            end if;

        end process;

```

Código TESTBENCH Registro de 4 Bits:

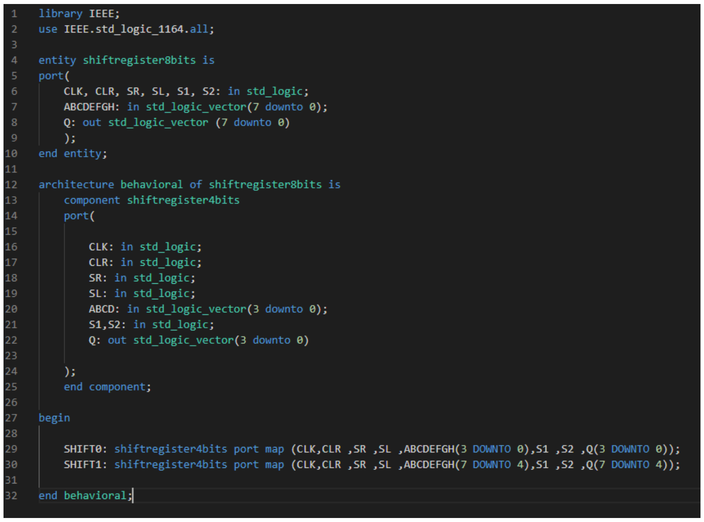

o Run Terminal Help
shiftregiste
shiftregister4bits.vhdl
shiftregister4bitsTB.vhdl
sjs.vhdl
LabSistemasAvanzados > HectorPequeno06 > ShiftRegister8 > shiftregister4bitsTB.vhdl

```
library IEEE;
use IEEE.std_logic_1164.all;

entity shiftregister4bitsTB is
end entity;

architecture behavioral of shiftregister4bitsTB is
    component shiftregister4bits
        port(

            CLK: in std_logic;
            CLR: in std_logic;
            SR: in std_logic;
            SL: in std_logic;
            ABCD: in std_logic_vector(3 downto 0);
            S1,S2: in std_logic;
            Q: out std_logic_vector(3 downto 0)

        );
    end component;

--Inputs
signal CLK: std_logic:='0';
signal CLR: std_logic:='0';
signal SR: std_logic:='0';
signal SL: std_logic:='0';
signal ABCD: std_logic_vector(3 downto 0):="0000";
signal S1,S2: std_logic:='0';
--outputs
signal Q: std_logic_vector(3 downto 0);

begin

    uut: shiftregister4bits port map(
            CLK=>CLK,
            CLR=>CLR,
            SR=>SR,
            SL=>SL,
            ABCD=>ABCD,
            S1=>S1,
            S2=>S2,
            Q=>Q
            );


    --CLOCK
    CLOCK: process
        begin
            CLK<='0';
```

Simulación del registro de corrimiento de 4 bits:

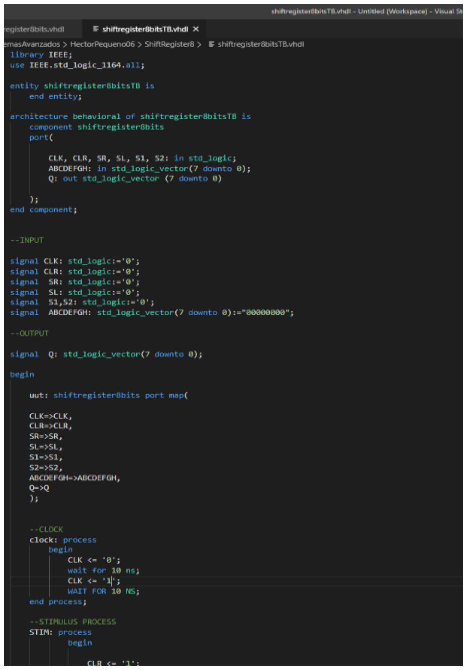


Código Fuente Registro de 4 Bits (UTILIZANDO INSTANCIAMIENTO):



```
library IEEE;
use IEEE.std_logic_1164.all;

entity shiftregister8bits is
port(
    CLK, CLR, SR, SL, S1, S2: in std_logic;
    ABCDEFGH: in std_logic_vector(7 downto 0);
    Q: out std_logic_vector (7 downto 0)
    );
end entity;

architecture behavioral of shiftregister8bits is
    component shiftregister4bits
    port(

        CLK: in std_logic;
        CLR: in std_logic;
        SR: in std_logic;
        SL: in std_logic;
        ABCD: in std_logic_vector(3 downto 0);
        S1,S2: in std_logic;
        Q: out std_logic_vector(3 downto 0)

    );
    end component;

begin

    SHIFT0: shiftregister4bits port map (CLK,CLR ,SR ,SL ,ABCDEFGH(3 DOWNTO 0),S1 ,S2 ,Q(3 DOWNTO 0));
    SHIFT1: shiftregister4bits port map (CLK,CLR ,SR ,SL ,ABCDEFGH(7 DOWNTO 4),S1 ,S2 ,Q(7 DOWNTO 4));

end behavioral;
```

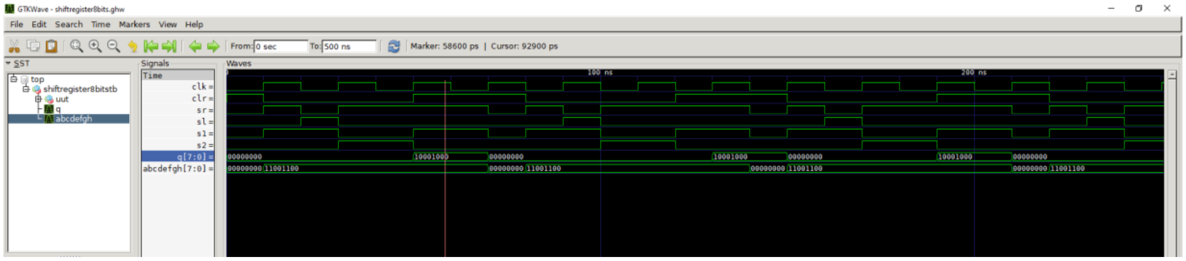
Código TESTBENCH Registro de 4 Bits (UTILIZANDO INSTANCIAMIENTO):



shiftregister8bitsTB.vhdl - Untitled (Workspace) - Visual St
register8bits.vhdl
shiftregister8bitsTB.vhdl ×
emasAvanzados > HectorPequeno06 > ShiftRegister8 > shiftregister8bitsTB.vhdl

```
library IEEE;
use IEEE.std_logic_1164.all;

entity shiftregister8bitsTB is
    end entity;

architecture behavioral of shiftregister8bitsTB is
    component shiftregister8bits
    port(

        CLK, CLR, SR, SL, S1, S2: in std_logic;
        ABCDEFGH: in std_logic_vector(7 downto 0);
        Q: out std_logic_vector (7 downto 0)

    );
end component;


--INPUT

signal CLK: std_logic:='0';
signal CLR: std_logic:='0';
signal  SR: std_logic:='0';
signal  SL: std_logic:='0';
signal  S1,S2: std_logic:='0';
signal  ABCDEFGH: std_logic_vector(7 downto 0):="00000000";

--OUTPUT

signal  Q: std_logic_vector(7 downto 0);

begin

    uut: shiftregister8bits port map(

    CLK=>CLK,
    CLR=>CLR,
    SR=>SR,
    SL=>SL,
    S1=>S1,
    S2=>S2,
    ABCDEFGH=>ABCDEFGH,
    Q=>Q
    );


    --CLOCK
    clock: process
        begin
            CLK <= '0';
            wait for 10 ns;
            CLK <= '1';
            WAIT FOR 10 NS;
    end process;

    --STIMULUS PROCESS
    STIM: process
            begin

                CLR <= '1';
```

Simulación del registro de corrimiento de 8 bits:



From: 0 sec To: 500 ns | Marker: 58600 ps | Cursor: 92900 ps