

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>

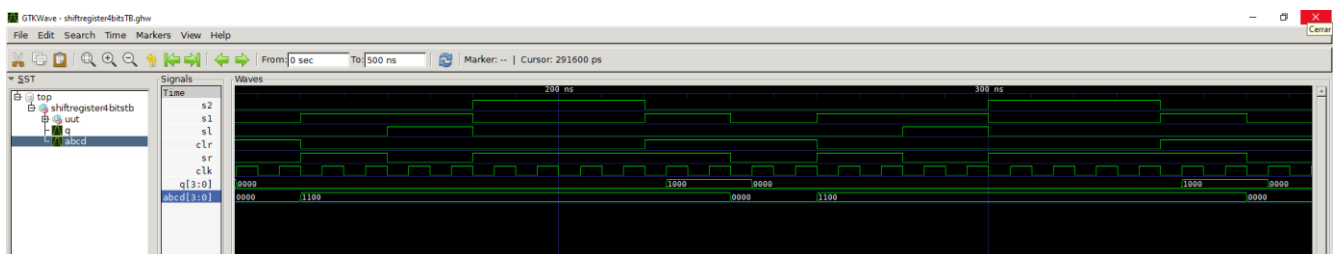
## Código Fuente Registro de 4 Bits:

```
4  entity shiftregister4bits is
5      port(
6          CLK: in std_logic;
7          CLR: in std_logic;
8          SR: in std_logic;
9          SL: in std_logic;
10         ABCD: in std_logic_vector(3 downto 0);
11         S1,S2: in std_logic;
12         Q: out std_logic_vector(3 downto 0)
13     );
14 end entity;
15
16
17 architecture behavioral of shiftregister4bits is
18     signal Qs : std_logic_vector(3 downto 0);
19 begin
20
21     Q<=Qs;
22     process(clk)
23     begin
24         if clr='0' then
25
26             Qs<="0000";
27
28         elsif clr='1' then
29
30             if CLK = '1' and CLK'event then
31
32                 if S1='1' and S2='1' then
33
34                     Qs <= ABCD;
35
36                 elsif S1='0' and S2='1' and SR= '1' then
37
38                     Qs<= '1' & ABCD(3 downto 1);
39
40                 elsif S1='0' and s2='1' and SR='0' then
41
42                     Qs<= '0' & ABCD(3 downto 1);
43
44                 elsif S1='1' and S2='0' and S1 = '1' then
45
46                     Qs<= ABCD(2 downto 0) & '1';
47
48                 elsif S1='1' and S2='0' and S1 = '0' then
49
50                     Qs<= ABCD(2 downto 0) & '0';
51
52                 elsif S1='0' and S2='0' then
53
54                     Qs<=ABCD;
55
56             end if;
57
58         end if;
59
60     end if;
61
62     end process;
63
```

Código TESTBENCH Registro de 4 Bits:

```
o Run Terminal Help shiftregister4bitsTB.vhdl
shiftregister4bits.vhdl shiftregister4bitsTB.vhdl x sjs.vhdl
labSistemasAvanzados > HectorPequeno06 > ShiftRegister8 > shiftregister4bitsTB.vhdl
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity shiftregister4bitsTB is
5 end entity;
6
7 architecture behavioral of shiftregister4bitsTB is
8     component shiftregister4bits
9     port(
10
11         CLK: in std_logic;
12         CLR: in std_logic;
13         SR: in std_logic;
14         SL: in std_logic;
15         ABCD: in std_logic_vector(3 downto 0);
16         S1,S2: in std_logic;
17         Q: out std_logic_vector(3 downto 0)
18     );
19 end component;
20
21 --Inputs
22 signal CLK: std_logic:='0';
23 signal CLR: std_logic:='0';
24 signal SR: std_logic:='0';
25 signal SL: std_logic:='0';
26 signal ABCD: std_logic_vector(3 downto 0):="0000";
27 signal S1,S2: std_logic:='0';
28 --outputs
29 signal Q: std_logic_vector(3 downto 0);
30
31 begin
32
33     uut: shiftregister4bits port map(
34         CLK=>CLK,
35         CLR=>CLR,
36         SR=>SR,
37         SL=>SL,
38         ABCD=>ABCD,
39         S1=>S1,
40         S2=>S2,
41         Q=>Q
42     );
43
44 --CLOCK
45
46 CLOCK: process
47     begin
48         CLK<='0';
49
```

Simulación del registro de corrimiento de 4 bits:



## Código Fuente Registro de 4 Bits (UTILIZANDO INSTANCIAMIENTO):

```
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity shiftregister8bits is
5 port(
6     CLK, CLR, SR, SL, S2: in std_logic;
7     ABCDEFGH: in std_logic_vector(7 downto 0);
8     Q: out std_logic_vector (7 downto 0)
9 );
10 end entity;
11
12 architecture behavioral of shiftregister8bits is
13     component shiftregister4bits
14     port(
15
16         CLK: in std_logic;
17         CLR: in std_logic;
18         SR: in std_logic;
19         SL: in std_logic;
20         ABCD: in std_logic_vector(3 downto 0);
21         S1,S2: in std_logic;
22         Q: out std_logic_vector(3 downto 0)
23     );
24     end component;
25
26 begin
27
28     SHIFT0: shiftregister4bits port map (CLK,CLR ,SR ,SL ,ABCDEFGH(3 DOWNTO 0),S1 ,S2 ,Q(3 DOWNTO 0));
29     SHIFT1: shiftregister4bits port map (CLK,CLR ,SR ,SL ,ABCDEFGH(7 DOWNTO 4),S1 ,S2 ,Q(7 DOWNTO 4));
30
31 end behavioral;
```

## Código TESTBENCH Registro de 4 Bits (UTILIZANDO INSTANCIAMIENTO):

```
shiftregister8bitsTB.vhdl - Untitled (Workspace) - Visual S
shiftregister8bitsTB.vhdl 1 shiftregister8bitsTB.vhdl X
masAvancadas > HectorPequeno06 > ShiftRegister > 1 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, 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:

