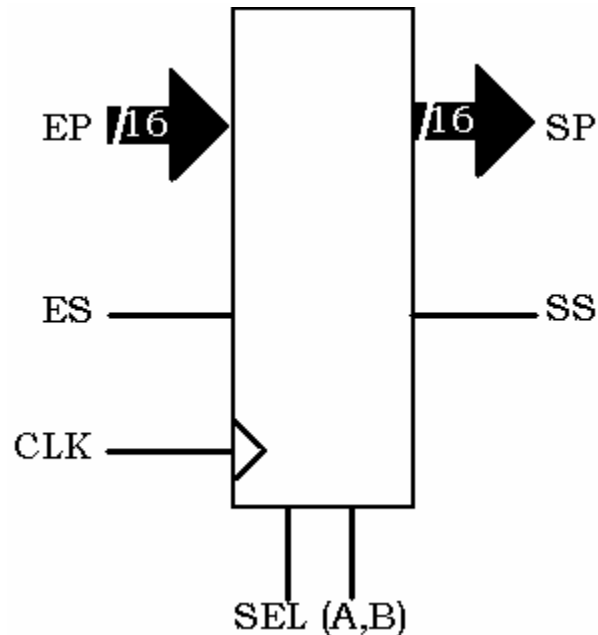


## REGISTRO DE DESPLAZAMIENTO DE 16 BITS CON ENTRADA PARALELA, SALIDA PARALELA, ENTRADA SERIAL, SALIDA SERIAL.

Primero diseñamos uno con las siguientes características, ya que no especifica nada el problema.



A	B	Resultado	Resultado colateral
0	0	EP → SP	SS = 'Z'
0	1	ES → SP	SS = 'Z'
1	0	EP → SS	SP = 'Z'
1	1	ES → SS	SP = 'Z'

Creamos el siguiente código fuente:

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

entity problemal_registro is
    Port ( ep : in std_logic_vector(15 downto 0);
          sp : out std_logic_vector(15 downto 0);
          es : in std_logic;
              selector:in bit_vector (1 downto 0);
          clk : in std_logic;
          ss : out std_logic);
```

```

end problemal_registro;

architecture algoritmo of problemal_registro is
begin
    process (clk)
        variable reg_temp: std_logic_vector (15 downto 0);
        variable cuenta1,cuenta2: integer;
        begin
            if clk='1' and clk'event then
                case selector is
                    when "00" =>
                        ss<='Z';
                        sp<=ep;
                        cuenta1:=0;
                        cuenta2:=0;

                    when "11" =>
                        ss<=es;

                        sp<="ZZZZZZZZZZZZZZZZZZ";
                        cuenta1:=0;
                        cuenta2:=0;

                    when "01" =>
                        if cuenta1 >=0 then
                            reg_temp(cuenta1):=es;
                            cuenta1:=cuenta1+1;
                            if cuenta1=16 then
                                cuenta1:=0;
                                sp<=reg_temp;
                            else
                                sp<="ZZZZZZZZZZZZZZZZZZ";
                            end if;
                        end if;
                        ss<='Z';
                        cuenta2:=0;

                    when "10" =>
                        cuenta1:=0;
                        if cuenta2=0 then
                            reg_temp:=ep;
                            sp<="ZZZZZZZZZZZZZZZZZZ";
                            ss<=reg_temp(cuenta2);
                            cuenta2:=cuenta2+1;
                            if cuenta2=16 then
                                cuenta2:=0;
                            end if;
                        end if;

                end case;
            end if;
        end process;
    end algoritmo;

    library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
    use IEEE.STD_LOGIC_ARITH.ALL;
    use IEEE.STD_LOGIC_UNSIGNED.ALL;

    entity problemal_registro is

```

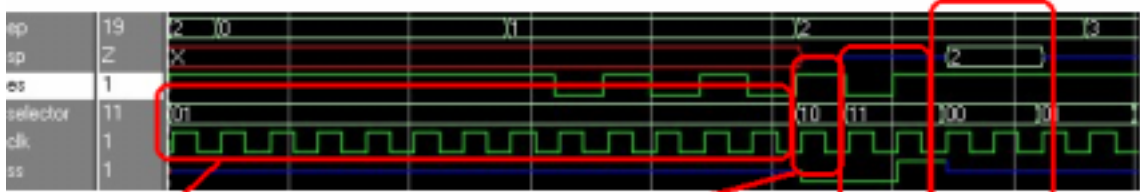
```

    Port ( ep : in std_logic_vector(15 downto 0);
          sp : out std_logic_vector(15 downto 0);
          es : in std_logic;
              selector:in bit_vector (1 downto 0);
          clk : in std_logic;
          ss : out std_logic);
end problemal_registro;

architecture algoritmo of problemal_registro is
begin
    process (clk)
        variable reg_temp: std_logic_vector (15 downto 0);
        variable cuenta1,cuenta2: integer;
    begin
        if clk='1' and clk'event then
            case selector is
when "00" =>
                ss<='Z';
                sp<=ep;
                cuenta1:=0;
                cuenta2:=0;
when "11" =>
                ss<=es;
                sp<="ZZZZZZZZZZZZZZZZZZ";
                cuenta1:=0;
                cuenta2:=0;
when "01" =>
                if cuenta1 >=0 then
                    reg_temp(cuenta1):=es;
                    cuenta1:=cuenta1+1;
                    if cuenta1=16 then      cuenta1:=0;
                    sp<=reg_temp;
                    else
                        sp<="ZZZZZZZZZZZZZZZZZZ";
                    end if;
                end if;
                ss<='Z';
                cuenta2:=0;
when "10" =>
                cuenta1:=0;
                if cuenta2=0 then
                    reg_temp:=ep;
                    sp<="ZZZZZZZZZZZZZZZZZZ";

                    ss<=reg_temp(cuenta2);
                    cuenta2:=cuenta2+1;
                    if cuenta2=16 then
                        cuenta2:=0;
                    end if;
                end if;
            end case;
        end if;
    end process;
end algoritmo;

```



opción "01" de entrada serie - salida en paralelo, acá la salida en paralelo cambiará de alta impedancia al valor cargado despues de 16 ciclos de reloj, acá se ve truncado

opción "10", entrada paralelo-salida serie, acá sólo sale el primer bit, ya que cambia de opción al siguiente ciclo

opción "11" entrada serie-salida serie; se ve como la salida en paralelo se pone en alta impedancia

opción "00" de carga paralela, vemos que la salida en serie se pone en alta impedancia