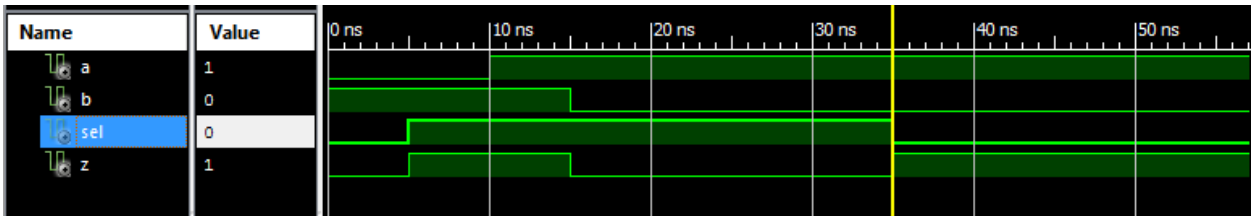
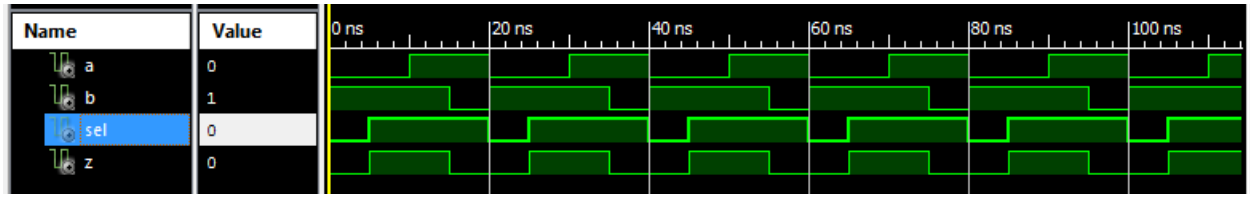
# Practica I

## 1.1.1




Como podemos apreciar en la simulación, el comportamiento del multiplexor se corresponde con el esperado.

## 1.1.2




Como podemos apreciar por la simulación, el comportamiento se corresponde con el esperado.

La principal diferencia es que los estímulos se añaden como procesos, evitando así tener que introducirlos de manera manual, como en la simulación anterior.

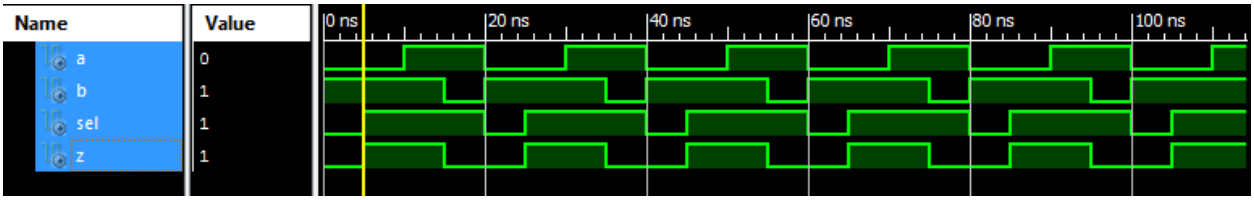
```
a_stimuli: process
    begin
   a <= '0', '1' after 10 ns;
   wait for 20 ns;
      end process;

b_stimuli: process
    begin
   b <= '1', '0' after 15 ns;
   wait for 20 ns;
      end process;

sel_stimuli: process
    begin
   sel <= '0', '1' after 5 ns;
   wait for 20 ns;
end process;
```

## 1.2.1

```
Parsing VHDL file "C:/Users/rboq/Documents/LCSE/mux2/mux_estructural.vhd" into library work
ERROR:HDLCompiler:104 - "C:/Users/rboq/Documents/LCSE/mux2/mux_estructural.vhd" Line 15: Cannot find <logic_components> in library <teach_logic_lib>.
ERROR:HDLCompiler:374 - "C:/Users/rboq/Documents/LCSE/mux2/mux_estructural.vhd" Line 17: Entity <mi_mux> is not yet compiled.
ERROR:HDLCompiler:69 - "C:/Users/rboq/Documents/LCSE/mux2/mux_estructural.vhd" Line 23: <inv> is not declared.
ERROR:HDLCompiler:69 - "C:/Users/rboq/Documents/LCSE/mux2/mux_estructural.vhd" Line 26: <or2> is not declared.
ERROR:HDLCompiler:69 - "C:/Users/rboq/Documents/LCSE/mux2/mux_estructural.vhd" Line 29: <nand2> is not declared.
ERROR:HDLCompiler:69 - "C:/Users/rboq/Documents/LCSE/mux2/mux_estructural.vhd" Line 32: <nand2> is not declared.
VHDL file C:/Users/rboq/Documents/LCSE/mux2/mux estructural.vhd ignored due to errors
```

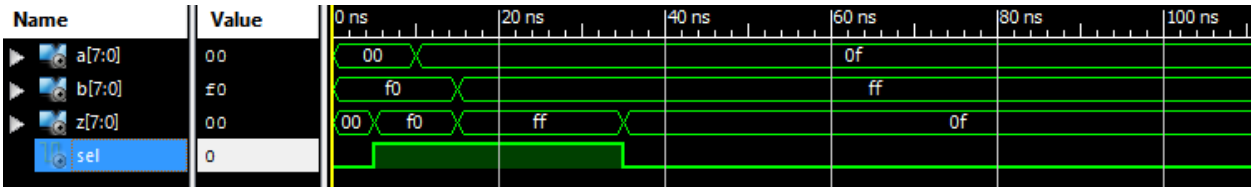
La primera causa del error, es que no se han añadido las librerías en las que se describen los componentes utilizados por el multiplexor. Para solucionarlo debemos añadir las librerías correspondientes.

## 1.2.2




Como podemos apreciar en la simulación, el funcionamiento es igual al del apartado 1.1.2

## 1.3.1




La simulación es diferente a las anteriores, dado que las entradas y la salida son grupos de bits descritos utilizando la librería std_logic_vector.

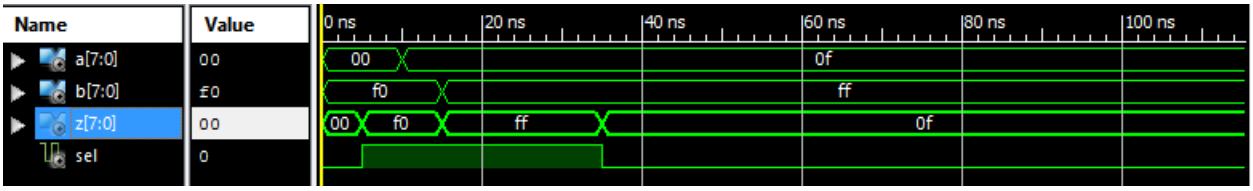
## 1.3.2

Para la construcción del multiplexor de 2 a 1 de 8 bits a partir de un multiplexor 2 a 1 de 1 bit, proponemos la siguiente solución:

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY mux2a1_de8 IS
    PORT(a: in std_logic_vector (7 downto 0);
    b: in std_logic_vector (7 downto 0);
    sel: in std_logic;
    z: out std_logic_vector (7 downto 0));
END;

architecture behavior of mux2a1_de8 is

component mi_mux
   port( a: in std_logic;
         b: in std_logic;
         sel: in std_logic;
         z: out std_logic);
end component;

begin
   gen: for i in 7 downto 0 generate
      mux:mi_mux port map (a(i), b(i), sel, z(i));
   end generate gen;

end behavior;
```

Esta simulación replica un multiplexor de 2 a 1 para un bus de 7 bits de ancho.

La simulación valida la solución propuesta:

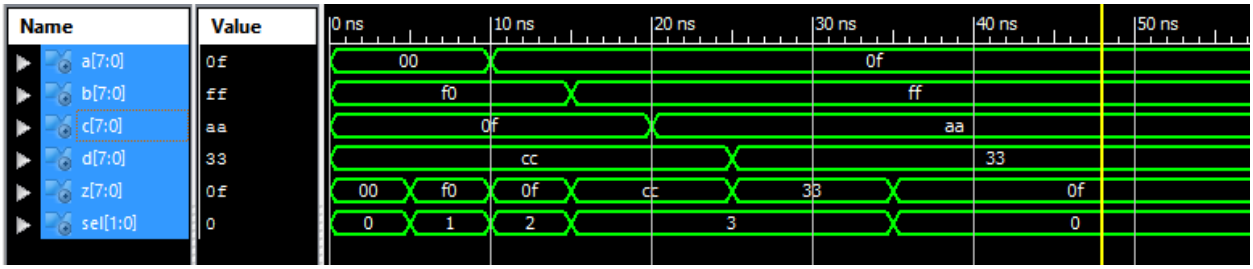



## 1.3.3

Realizamos el modelo pedido:

```
entity mux4a1_de8 is
    Port ( a : in  STD_LOGIC_VECTOR (7 downto 0);
           b : in  STD_LOGIC_VECTOR (7 downto 0);
           c : in  STD_LOGIC_VECTOR (7 downto 0);
           d : in  STD_LOGIC_VECTOR (7 downto 0);
           sel : in  STD_LOGIC_VECTOR (1 downto 0);
           z : out  STD_LOGIC_VECTOR (7 downto 0));
end mux4a1_de8;

architecture Behavioral of mux4a1_de8 is

begin
   process (a, b, c, d, sel)
      begin
         if sel = "00" then
            z <= a;
         elsif sel = "01" then
            z <= b;
         elsif sel = "10" then
            z <= c;
         else
            z <= d;
         end if;
   end process;

end Behavioral;
```

La simulación valida el modelo propuesto:




## 2.1.1

Se produce un error durante la compilación.

```
Elaborating entity <dec3to8> (architecture <behavior>) from library <work>.
ERROR:HDLCompiler:299 - "C:\Users\rboq\Documents\LCSE\decoder\decoder.vhd" Line 31: case statement does not cover all choices. 'others' clause is needed
Netlist dec3to8(behavior) remains a blackbox, due to errors in its contents
-->
```

Este error se debe a que para sintetizar el  HDL, es necesario que el modelo del multiplexor contemple el comportamiento a todas las entradas posibles.  Para esto, añadimos una respuesta a “others” que garantiza que la salida siempre tendrá un valor por defecto.

## 2.1.2

```
Elaborating entity <dec3to8> (architecture <behavior>) from library <work>.
INFO:HDLCompiler:679 - "C:\Users\rboq\Documents\LCSE\decoder\decoder.vhd" Line 40. Case statement is complete. others clause is never selected
```

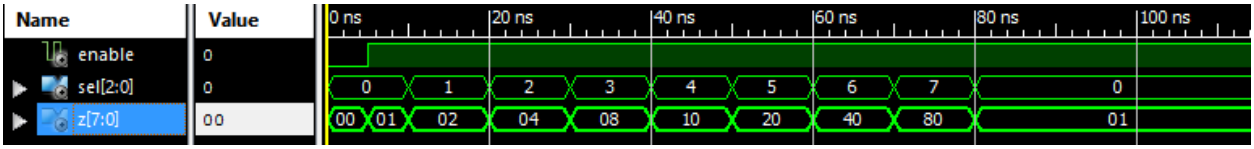
El mensaje lanza un warning advirtiendo que el estímulo por defecto nunca se ejecutará, esto tiene sentido dado que X no es un valor contemplado a la hora de realizar  la síntesis.

## 2.1.3

Los estímulos generados en el testbench son los siguientes:

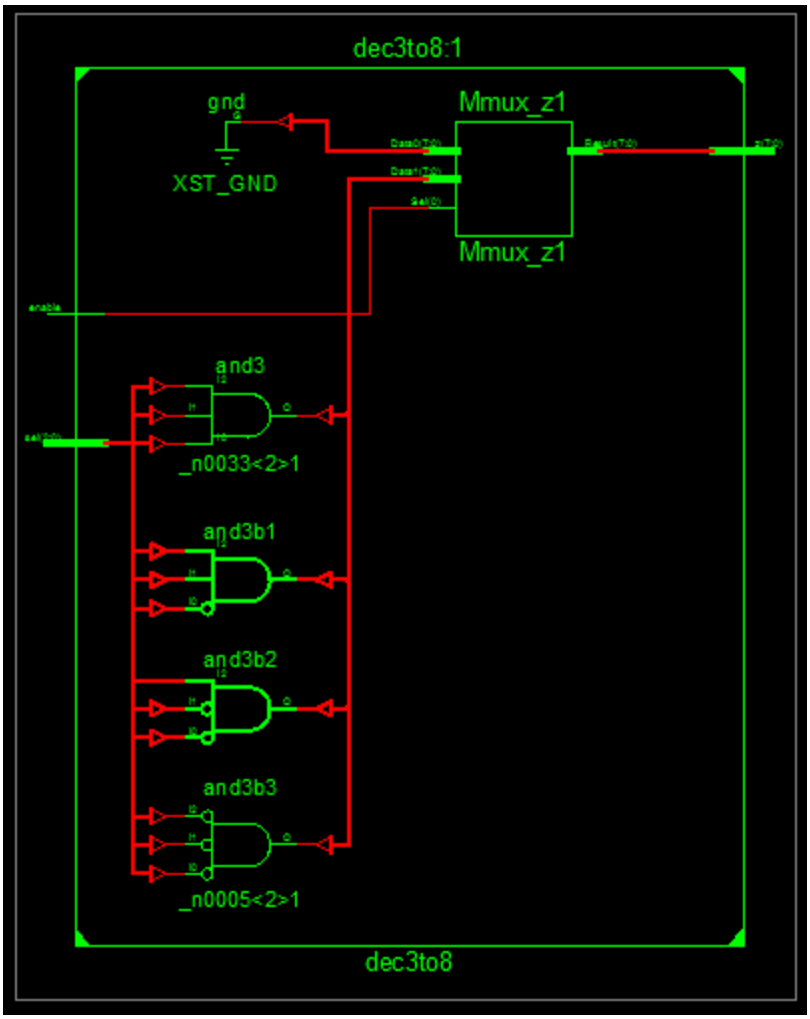
```
-------------------------------------------------------------------------------
--  estimulos
-------------------------------------------------------------------------------

   sel <= "000", "001" after 10 ns, "010" after 20 ns, "011" after 30 ns, "100" after 40 ns,
    "101" after 50 ns, "110" after 60 ns, "111" after 70 ns, "000" after 80 ns;
   enable <= '0', '1' after 5 ns;
```

Comprobamos que el decodificador funciona correctamente:




## 2.2.1

Al tratarse de un circuito combinacional, podemos escribir las salidas como un POS, que es precisamente lo que podemos ver en la síntesis. Por otro lado, la señal de enable multiplexa la salida de este POS a la salida cuando está activa, en caso contrario, nos da la salida por defecto.




2.2.2

Podemos ver el resumen de los recursos de la FPGA usados, en la siguiente captura

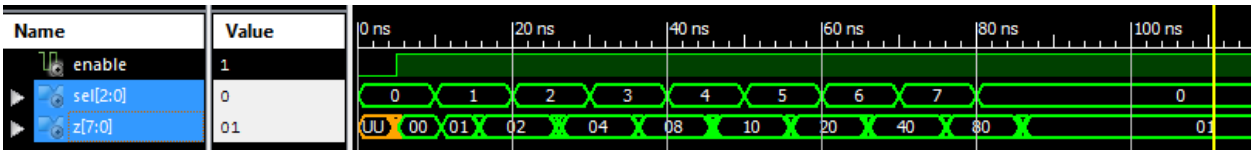
| Device Utilization Summary (estimated values) | | | [-] |
|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** |
| Number of Slice LUTs | 8 | 63400 | 0% |
| Number of fully used LUT-FF pairs | 0 | 8 | 0% |
| Number of bonded IOBs | 12 | 210 | 5% |

2.2.3

El circuito es combinacional, dado que no depende de un estado y puede escribirse como una función booleana dependiente solamente de las entradas.

2.2.5

El decodificador no se comporta de igual manera, dado que la simulación post place-route, tiene en cuenta los retardos producidos durante la implementación física en la FPGA.




2.3.1

El código escrito tiene el siguiente aspecto:-

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY dec4to16 IS
   PORT (
       sel : IN std_logic_vector (3 downto 0);
       enable : IN STD_LOGIC;
       z : OUT STD_LOGIC_VECTOR (15 downto 0)
   );
END dec4to16;

LIBRARY ieee;
USE ieee.std_logic_arith.all;

ARCHITECTURE behavior OF dec4to16 IS
BEGIN
   PROCESS (sel, enable)
   BEGIN
      z <= "0000000000000000";
      IF (enable = '1') THEN
        CASE sel IS
         WHEN "0000" => z(0) <= '1';
         WHEN "0001" => z(1) <= '1';
         WHEN "0010" => z(2) <= '1';
         WHEN "0011" => z(3) <= '1';
         WHEN "0100" => z(4) <= '1';
         WHEN "0101" => z(5) <= '1';
         WHEN "0110" => z(6) <= '1';
         WHEN "0111" => z(7) <= '1';
         WHEN "1000" => z(8) <= '1';
         WHEN "1001" => z(9) <= '1';
         WHEN "1010" => z(10) <= '1';
         WHEN "1011" => z(11) <= '1';
         WHEN "1100" => z(12) <= '1';
         WHEN "1101" => z(13) <= '1';
         WHEN "1110" => z(14) <= '1';
         WHEN "1111" => z(15) <= '1';
         WHEN others => z <="XXXXXXXXXXXXXXXX";
        END CASE;
      END IF;
   END PROCESS;
END behavior;
```