INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

**NOMBRE DE LOS INTEGRANTES:**

* Luciano Espina Melisa
* Medina Conde Gerson Levi
* Romero Martínez Edgar Adán

**Práctica 2.8 "Demultiplexor 1 a 4"** **con habilitación**

Grupo: 2CV2

**Unidad de Aprendizaje:**

Fundamentos de Diseño Digital

**Fecha de revisión:**

26 de octubre de 2016

[Desarrollo 3](#_Toc465470980)

[Simulación 5](#_Toc465470981)

[Código 6](#_Toc465470982)

[Foto 7](#_Toc465470983)

[Conclusiones 8](#_Toc465470984)

[Medina Conde Gerson Leví 8](#_Toc465470985)

[Luciano Espina Melisa 8](#_Toc465470986)

[Romero Martínez Edgar Adán 8](#_Toc465470987)

# Desarrollo

Un demultiplexor es un circuito combinacional que tiene una entrada de información de datos *d* y *n* entradas de control que sirven para seleccionar una de las 2n salidas, por la que ha de salir el dato que presente en la entrada. Esto se consigue aplicando a las entradas de control la combinación binaria correspondiente a la salida que se desea seleccionar. Por ejemplo, si queremos que la información que tenemos en la entrada *d*, salga por la salida *S4*, en la entrada de control se ha de poner, de acuerdo con el peso de la misma, el valor *100*, que es el 4 en binario.

En esta práctica desarrollaremos un demultiplexor 1:4 (1 entrada, 4 salidas), basándonos en la tabla de verdad característica y en su circuito lógico, así como el diagrama a bloques y el lenguaje VHDL.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| H | S2 | S1 | O0 | O1 | O2 | O3 |
| 0 | 0 | 0 | X | X | X | X |
| 0 | 0 | 1 | X | X | X | X |
| 0 | 1 | 0 | X | X | X | X |
| 0 | 1 | 1 | X | X | X | X |
| 1 | 0 | 0 | I | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | I | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | I | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | I |

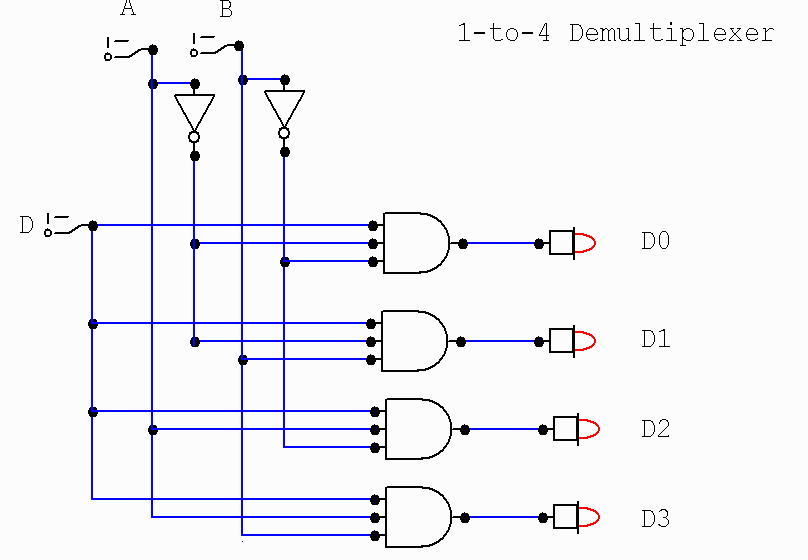
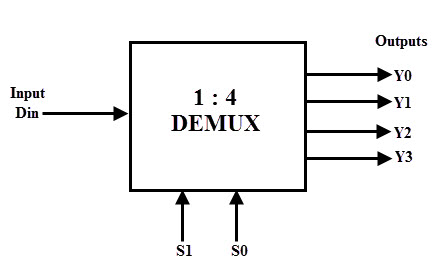
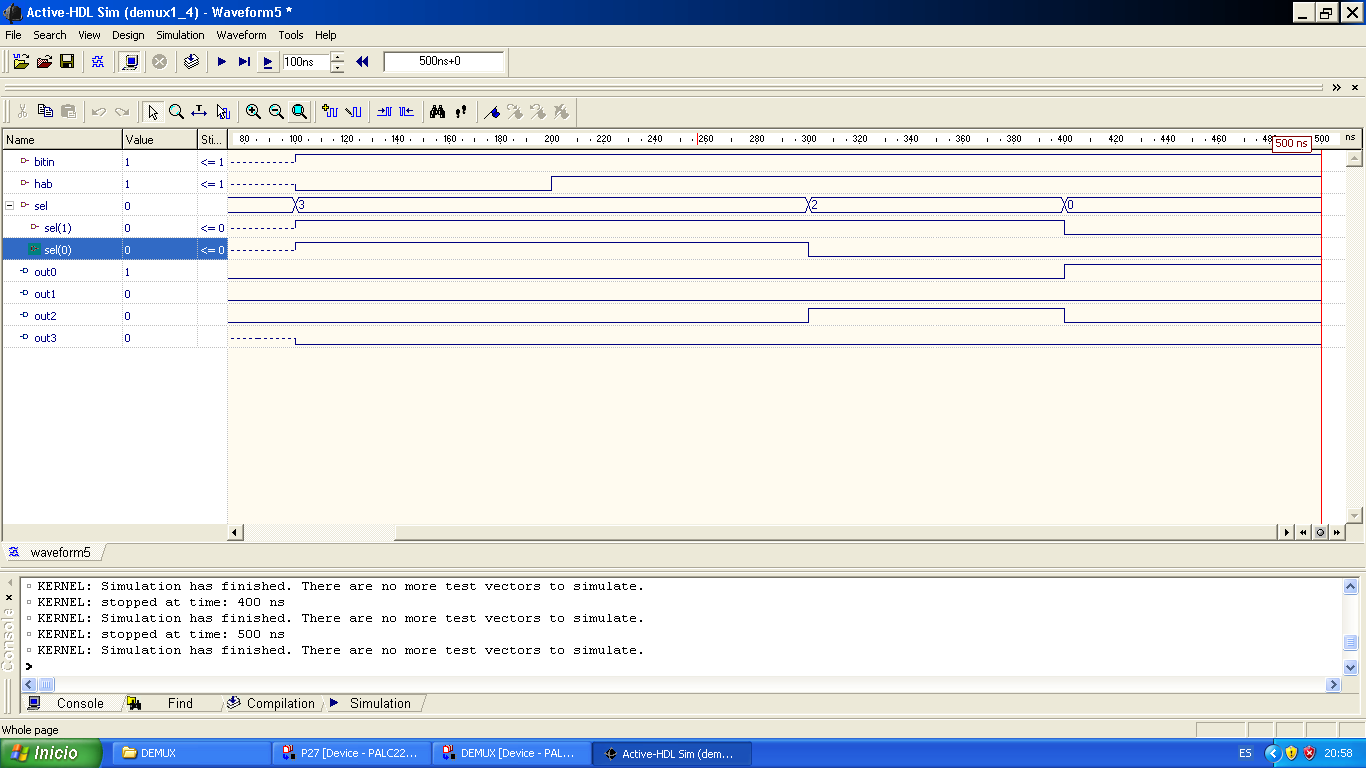
Donde H es el bit de habilitación, S2 y S1 son los bits de selección de salidas, y O0, O1, O2 y O3 son las salidas del demultiplexor.

Diagrama lógico del demultiplexor 1:4

Diagrama a bloques del demultiplexor 1:4

# Simulación



# Código

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity demux1\_4 is

port (

out0 : out std\_logic; --output bit

out1 : out std\_logic; --output bit

out2 : out std\_logic; --output bit

out3 : out std\_logic; --output bit

sel : in std\_logic\_vector(1 downto 0);

hab : in std\_logic;

bitin : in std\_logic --input bit

);

-- attribute pin\_numbers of demux1\_4: entity is

-- "bitin:2 hab:9 sel(0):10 sel(1):11 out0:17 out1:16 out2:15 out3:14";

end demux1\_4;

architecture Behavioral of demux1\_4 is

begin

process(bitin,sel)

begin

if(hab = '0') then

out0 <= '0'; out1 <= '0'; out2 <= '0'; out3 <= '0';

else

case sel is

when "00" => out0 <= bitin; out1 <= '0'; out2 <= '0'; out3 <='0';

when "01" => out1 <= bitin; out0 <= '0'; out2 <= '0'; out3 <='0';

when "10" => out2 <= bitin; out0 <= '0'; out1 <= '0'; out3 <='0';

when others => out3 <= bitin; out0 <= '0'; out1 <= '0'; out2 <='0';

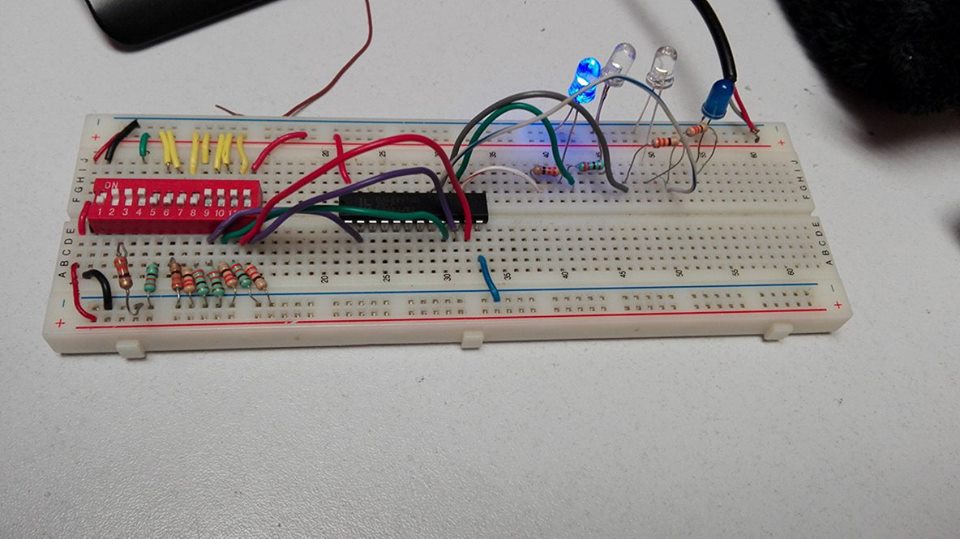
end case;

end if;

end process;

end Behavioral;

# Foto

****

# Conclusiones

### Medina Conde Gerson Leví

Los demultiplexores realizan la función inversa a la del multiplexor, es decir, una señal de entrada única, es obtenida en uno de los N canales de salida. El conmutador ahora selecciona el canal de salida por donde estará presente el dato de entrada.

### **Luciano Espina Melisa**

He comprendido que gracias al demultiplexor tenemos la posibilidad de desglosar una señal dada en varios canales, obteniendo sólo una de ellas, gracias a las combinaciones del selector.

### **Romero Martínez Edgar Adán**

Gracias a los demultiplexores podemos implementar el desarrollo de sistemas que manejen transferencia de bits, ya que se utiliza un algoritmo basado en álgebra de Boole, gracias a la cual podemos desglosar las señales dadas.