

Práctica 06

Introducción al lenguaje VHDL

Objetivo

- Aprender a programar en VHDL
- Observar la diferencia entre la programación gráfica y la programación en VHDL

Material y equipo

- Tarjeta TerAsic
- Equipo PC
- Compuertas AND, OR y NOT

Introducción¹

VHDL significa VHSIC Hardware Description Language, y a su vez VHSIC significa Very High Speed Integrated Circuit. Se trata de un lenguaje de descripción de hardware, esto significa que mediante él se puede describir la forma de comportarse de un circuito electrónico. El comportamiento puede ser llevado a algún dispositivo que dispondrá de sus propios componentes con los que lograr ese comportamiento deseado. La forma de comportarse es independiente del hardware donde se implementará.

El VHDL es un estándar llamado IEEE 1076-1993. Sus ventajas son:

- Una disponibilidad pública
- Independencia de dispositivos y fabricantes
- Reutilización
- Diseño jerárquico

Un proyecto de VHDL puede contener muchos ficheros. El código VHDL usualmente se encuentra en los ficheros con extensión *.vhd. La estructura típica de uno de estos ficheros es:

- Llamadas a librerías
- Entidad
- Arquitectura(s)

1 información obtenida de http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_VHDL

NOTA: para poder realizar la práctica es **necesario** que descargue el archivo "VHDL tutorial"

Decodificador

En la figura 1 se muestra el diagrama lógico, la tabla de verdad y las ecuaciones de un decodificador de 2 a 4.

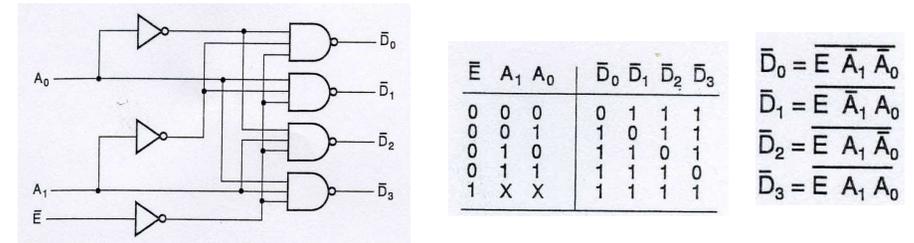


Fig. 1 Decodificador 2 a 4

La estructura en VHDL de éste decodificador es:

```
-- 2-to-4 Line Decoder: Structural VHDL Description -- 1
-- (See Figure 3-14 for logic diagram) -- 2
library ieee, lcdf_vhdl; -- 3
use ieee.std_logic_1164.all, lcdf_vhdl.func_prims.all; -- 4
entity decoder_2_to_4 is -- 5
  port(E_n, A0, A1: in std_logic; -- 6
        D0_n, D1_n, D2_n, D3_n: out std_logic); -- 7
end decoder_2_to_4; -- 8

architecture structural_1 of decoder_2_to_4 is -- 9
  component NOT1 -- 10
    port(in1: in std_logic; -- 11
          out1: out std_logic); -- 12
  end component; -- 13
  component NAND3 -- 14
    port(in1, in2, in3: in std_logic; -- 15
          out1: out std_logic); -- 16
  end component; -- 17
  signal E, A0_n, A1_n: std_logic; -- 18
begin -- 19
  g0: NOT1 port map (in1 => A0, out1 => A0_n); -- 20
  g1: NOT1 port map (in1 => A1, out1 => A1_n); -- 21
  g2: NOT1 port map (in1 => E_n, out1 => E); -- 22
  g3: NAND3 port map (in1 => A0_n, in2 => A1_n, -- 23
                     in3 => E, out1 => D0_n); -- 24
  g4: NAND3 port map (in1 => A0, in2 => A1_n, -- 25
                     in3 => E, out1 => D1_n); -- 26
  g5: NAND3 port map (in1 => A0_n, in2 => A1, -- 27
                     in3 => E, out1 => D2_n); -- 28
  g6: NAND3 port map (in1 => A0, in2 => A1, -- 29
                     in3 => E, out1 => D3_n); -- 30
end structural_1; -- 31
```

Procedimiento

Programa en VHDL el codificador 2 a 4 y compruebe su funcionamiento mediante la tarjeta de Altera.

Una vez que haya programado en lenguaje VHDL, programe el mismo codificador usando programación gráfica; comente las diferencias entre ambas formas de programar la tarjeta.

Una vez que ha programado el decodificador, programe los siguientes circuitos en VHDL:

1. Las compuertas NOT, AND y OR.
2. El circuito que se muestre en la figura 6 de la práctica 0.
3. El sumador medio y completo.
4. EL multiplexor, el codificador y el demux/deco de la práctica 5

Cuestionario

1. Reporte sus observaciones con respecto a la programación en VHDL