
Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL



Escuela Técnica Superior de Ingeniería
Departamento de
Ingeniería Electrónica, Sistemas Informáticos y Automática

Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

Introducción

Una de las aplicaciones del lenguaje VHDL es la simulación del comportamiento de los circuitos descritos en tal lenguaje. Esta simulación funcional, de tipo «virtual» (sobre la pantalla del computador) permite comprobar, sin construirlo físicamente, si el circuito cumple las especificaciones o requisitos que eran el objetivo propio de su diseño.

La simulación funcional de un circuito digital, en el contexto del lenguaje VHDL, se realiza a través de un nuevo fichero VHDL en el cual se «inserta» el circuito a simular dentro de una nueva entidad (una nueva «caja») sin terminales exteriores. La estructura de este fichero de test se resume en los siguientes puntos:

- Se crea **una entidad sin puertos** (podemos denominarla *nombre_del_circuitoTest.vhd*).
- En la arquitectura de dicha entidad (antes del begin) se declara como componente el circuito a simular y una señal para cada puerto del circuito (que puede tener el mismo nombre que el puerto al que se va a conectar).
- En la misma arquitectura (después del begin) se «instancia» el circuito a simular, conectando cada uno de sus puertos a la señal correspondiente (de las declaradas antes).
- También en la misma arquitectura se describen los vectores de test o formas de onda que darán valores a las entradas del circuito a lo largo del tiempo de simulación.

Consideremos como ejemplo simple de aplicación la función:

$$f = ab + c.$$

Definimos la función en VHDL en el entorno de diseño ISE WebPACK, según se observa en la siguiente diapositiva.

Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

The screenshot displays the ISE Project Navigator interface. The main editor shows the VHDL code for 'ejercicio2.vhd'. The code defines an entity 'ejercicio2' with three input ports (a, b, c) and one output port (y). The architecture is behavioral and implements the logic $y = (a \text{ AND } b) \text{ OR } c$.

```
1  -----  
2  -- Company:  
3  -- Engineer:  
4  -----  
5  library IEEE;  
6  use IEEE.STD_LOGIC_1164.ALL;  
7  use IEEE.STD_LOGIC_ARITH.ALL;  
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;  
9  
10 entity ejercicio2 is  
11     Port ( a : in  STD_LOGIC;  
12           b : in  STD_LOGIC;  
13           c : in  STD_LOGIC;  
14           y : out STD_LOGIC);  
15 end ejercicio2;  
16  
17 architecture Behavioral of ejercicio2 is  
18  
19 begin  
20     y <= (a and b) or c;  
21  
22 end Behavioral;  
23  
24
```

The left pane shows the project hierarchy with 'ejercicio2 - Behavioral (ejercicio2.vhd)' selected. The bottom pane shows the 'Design Summary (Synthesized)' tab, indicating that the synthesis process completed successfully.

Console Output:

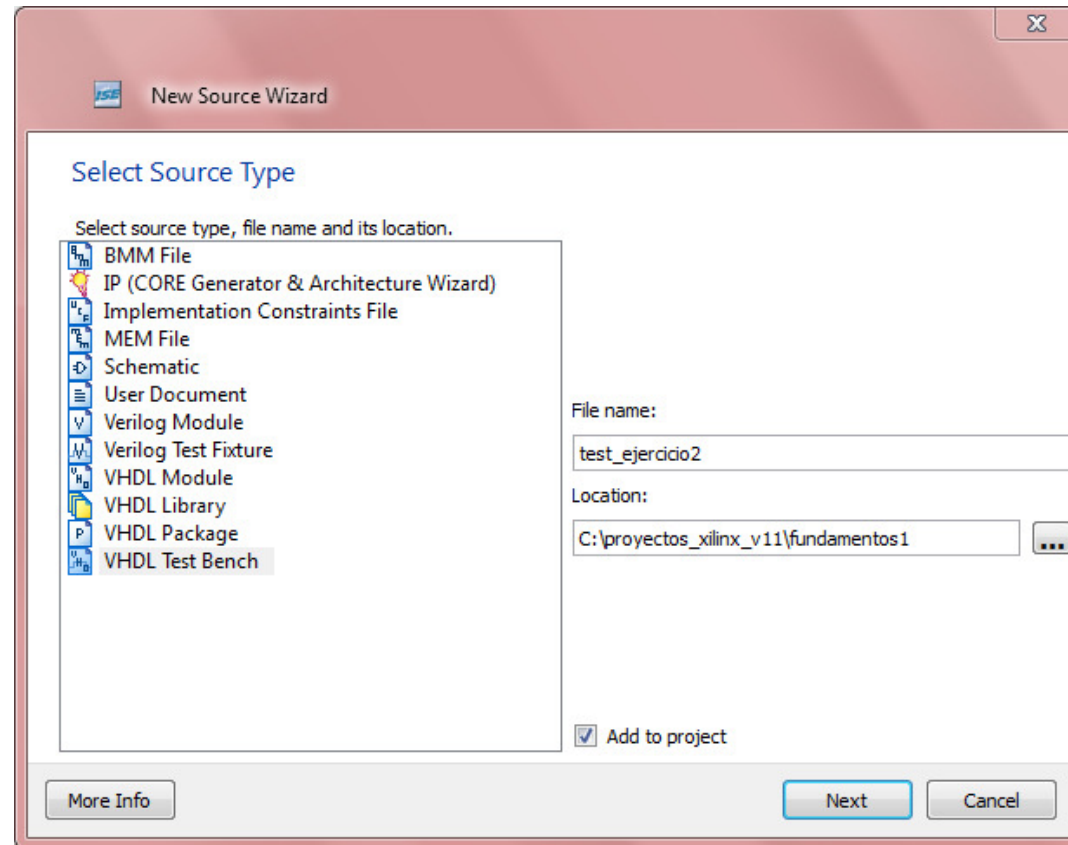
```
Maximum output required time after clock: No path found  
Maximum combinational path delay: 6.209ns  
  
-----  
  
Process "Synthesis" completed successfully
```

The status bar at the bottom indicates the current line is 20, column 23, in the VHDL file.

Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

Añadimos, mediante la opción “**New Source**”, un fichero del tipo **VHDL Test Bench** al proyecto.



Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

Edición de la plantilla generada por el entorno de diseño

Editamos la plantilla generada, dejándola como se muestra en la figura. Hemos eliminado algunas líneas de definición del reloj del sistema, el cual no se utiliza en los diseños de tipo combinacional como es el caso de nuestro ejemplo. En este fichero pueden verse las distintas partes que lo forman:

- Entidad sin puertos
- Declaración como componente (component) del diseño de la función
- Señales de conexión, con el mismo nombre que los puertos originales del diseño a simular.
- Instanciamos el circuito a simular (**Unit Under Test - UUT**) en la arquitectura, tras el "begin". Conectamos los puertos del circuito con las señales del mismo nombre, usando los "port maps".

Nos quedan por definir los estímulos, es decir, indicar en el fichero de test cómo deben cambiar las entradas para que el simulador nos proporcione las salidas.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;

ENTITY test_ejercicio2 IS
END test_ejercicio2;

ARCHITECTURE behavior OF test_ejercicio2 IS
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT ejercicio2
    PORT (
        a, b, c : IN  std_logic;
        y : OUT  std_logic
    );
    END COMPONENT;
    --Inputs
    signal a, b, c: std_logic := '0';
    --Outputs
    signal y : std_logic;

    BEGIN
        -- Instantiate the Unit Under Test (UUT)
        uut: ejercicio2 PORT MAP (
            a => a,
            b => b,
            c => c,
            y => y
        );

        -- Stimulus process
        stim_proc: process
        begin
            -- insert stimulus here

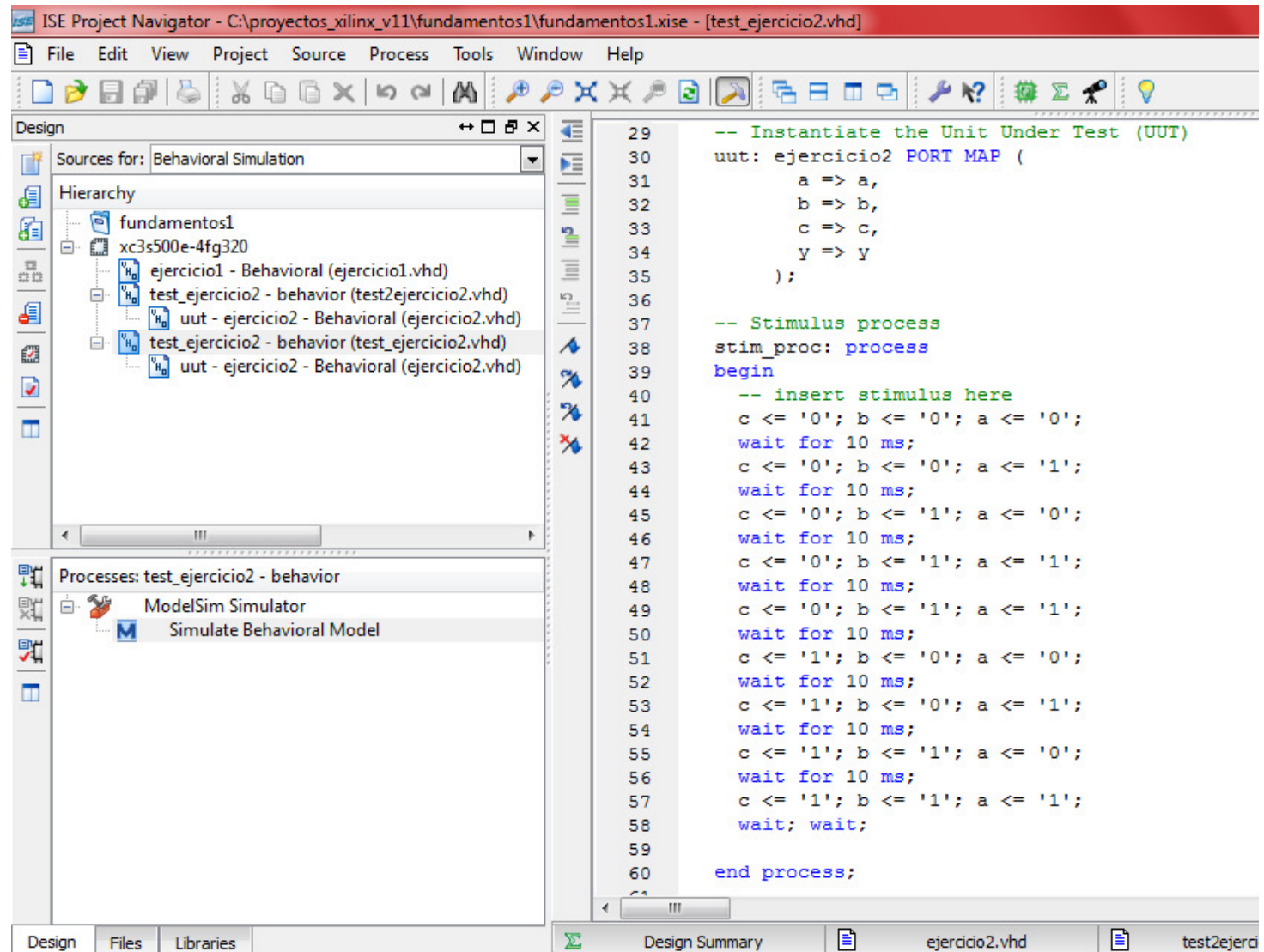
        end process;
    END;
```

Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

Definición de

estímulos: El archivo de test lo podemos ver en "**Behavioral Simulation**".

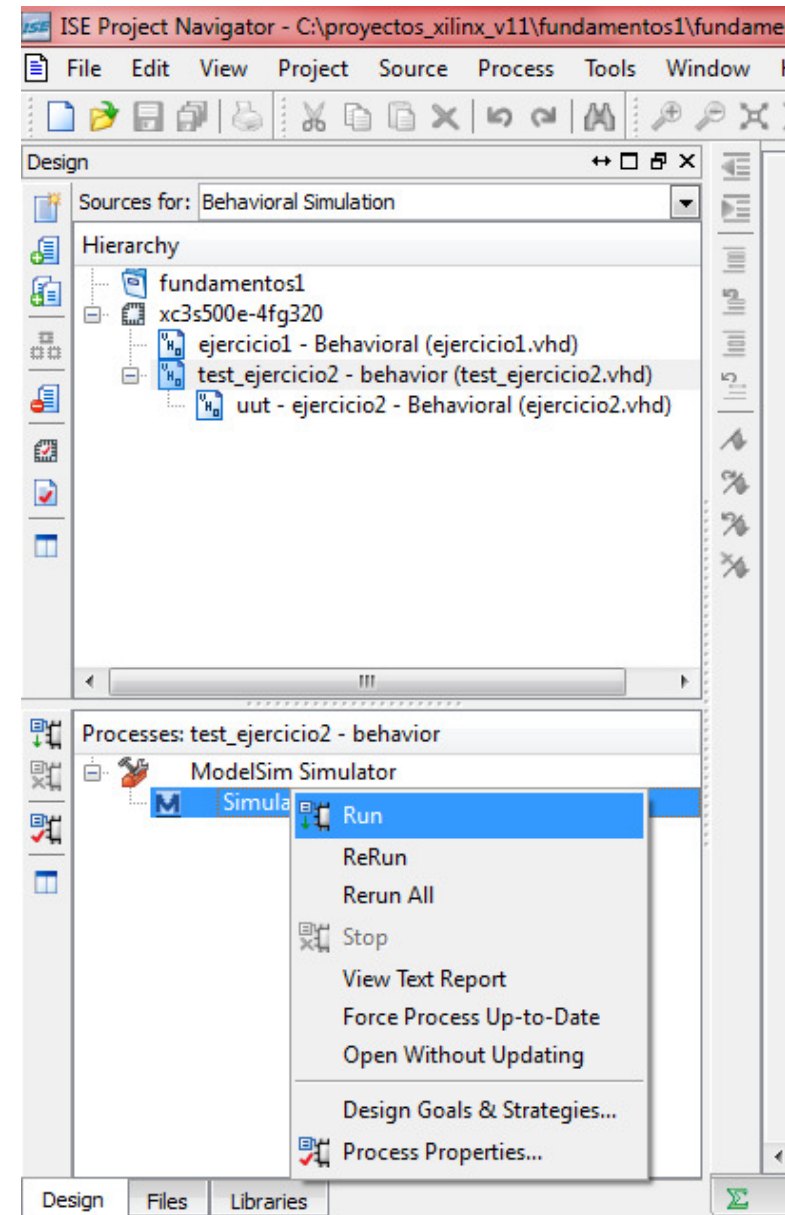


Tema 4: Sistemas Combinacionales

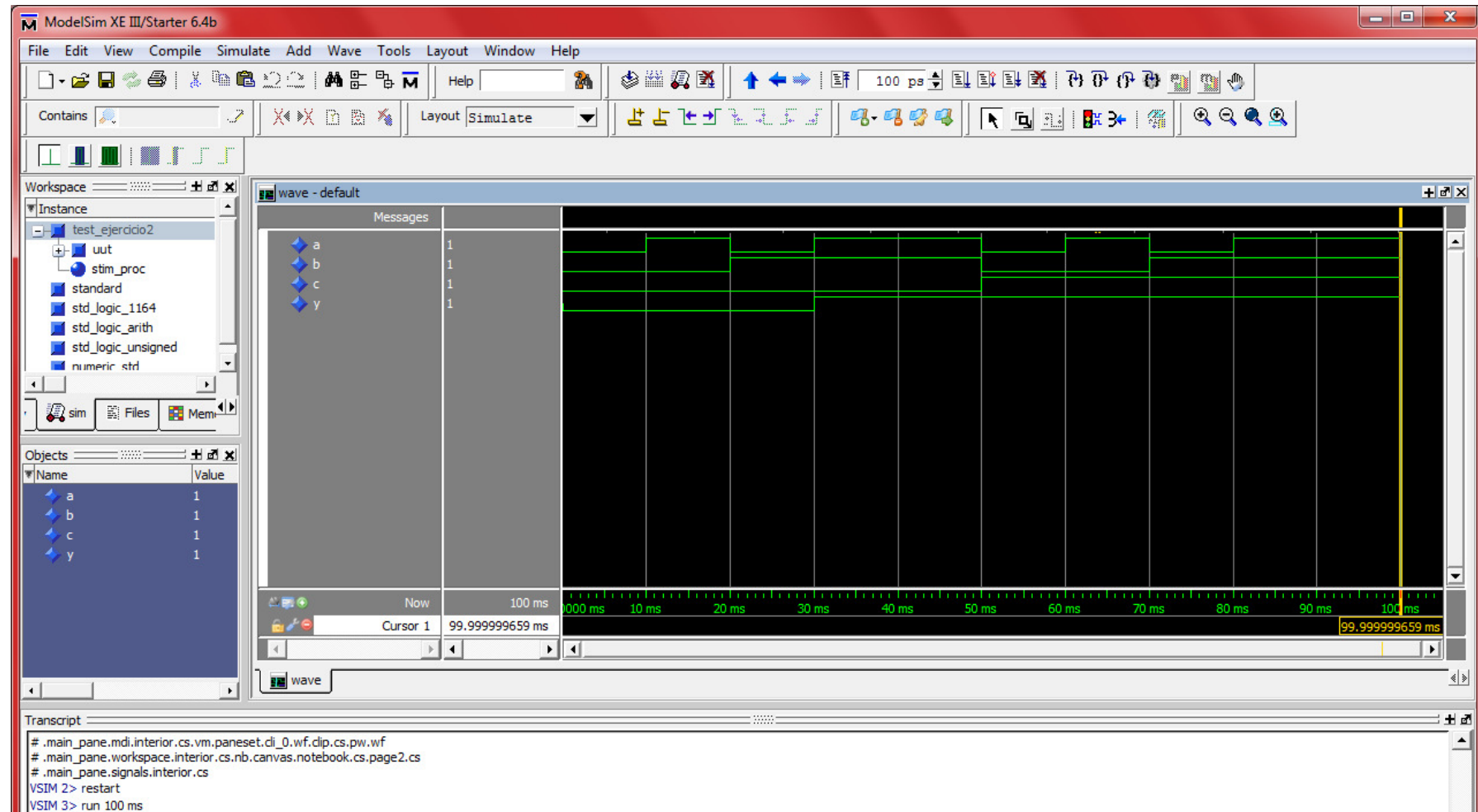
Anexo I: Introducción a la simulación usando VHDL

Elejimos “**Simulate Behavioral Model -> Run**”.

Eso ejecutará el simulador (ModelSim), según vemos en la siguiente diapositiva. Hemos de decirle al simulador que ejecute una cantidad de tiempo determinada. Para ello, en el prompt del simulador escribimos **restart** y pulsamos enter. Esto reiniciará desde el principio (instante 0 ms) el simulador. Luego simulamos durante 100 ms mediante la orden “**run 100 ms**”.



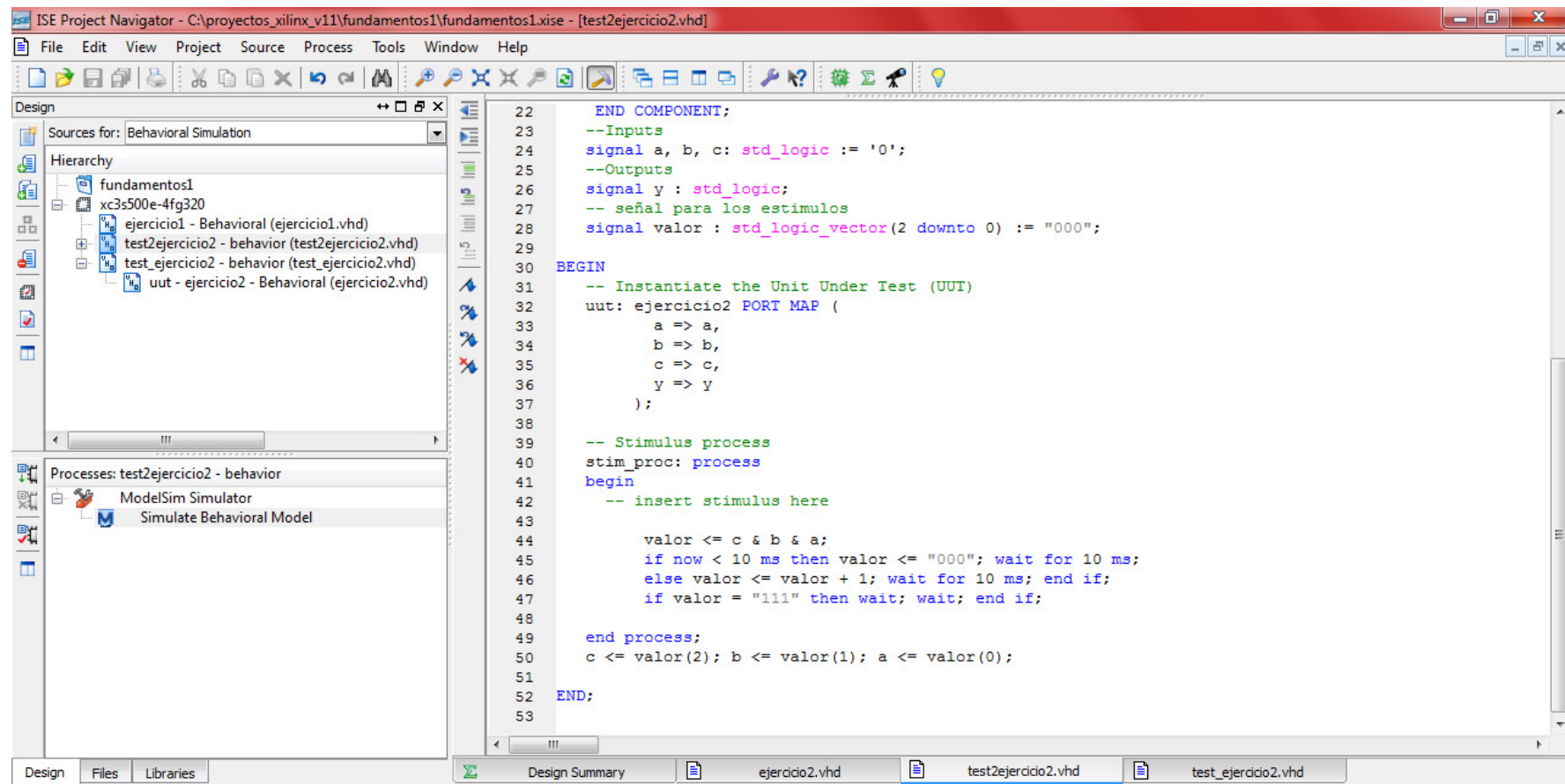
Anexo I: Introducción a la simulación usando VHDL



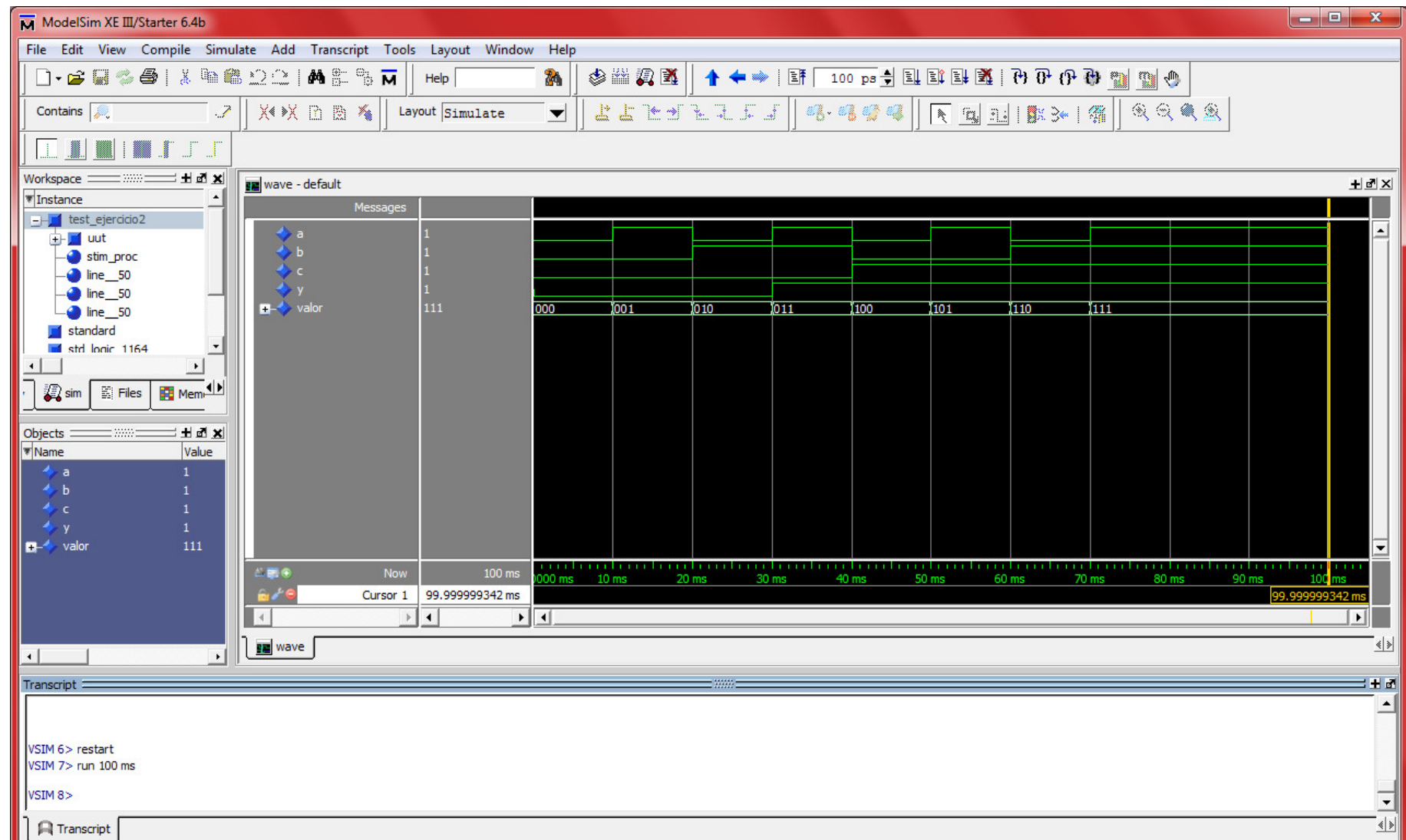
Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

También podemos definir los estímulos de una forma más compacta tal y como se observa en la figura. Obsérvese como se declara una nueva señal (valor), que irá tomando los valores desde **000** hasta **111**. En la siguiente diapositiva observamos el resultado de la simulación.



Anexo I: Introducción a la simulación usando VHDL



Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

Otro ejemplo de simulación de un circuito combinatorial

A continuación se muestra el fichero VHDL que define la función lógica: $f_1 = \sum_4(0, 2, 6, 7, 12, 14)$

```
-----  
--  
-- Create Date:      11:37:53 12/01/2008  
-- Module Name:      funcion1 - descripcion_f1  
-- Revision 0.01  
--  
-----  
  
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;  
  
entity funcion1 is  
    port (d, c, b, a: in std_logic;  
          f1 : out std_logic);  
end funcion1;  
  
architecture descripcion_f1 of funcion1 is  
    signal entrada: std_logic_vector(3 downto 0);  
    signal ent_int: integer range 0 to 15;  
begin  
    entrada <= d & c & b & a;  
    ent_int <= conv_integer(entrada);  
    with ent_int select  
        f1 <= '1' when 0|2|6|7|12|14,  
              '0' when others;  
end descripcion_f1;
```

Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

El fichero de estímulos para la simulación de la función es el siguiente:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
USE ieee.std_logic_arith.all;
USE ieee.numeric_std.ALL;

ENTITY funcion1_test2 IS
END funcion1_test2;

ARCHITECTURE behavior OF funcion1_test2 IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT funcion1
    PORT(
        d, c, b, a : IN  std_logic;
        f1 : OUT  std_logic
    );
    END COMPONENT;

    --Inputs
    signal d : std_logic := '0';
    signal c : std_logic := '0';
    signal b : std_logic := '0';
    signal a : std_logic := '0';
    signal entrada : std_logic_vector(3 downto 0) := "0000";
    signal entrada_int: integer range 0 to 15;
    --Outputs
    signal f1 : std_logic;
```

(Continúa en la siguiente diapositiva)

Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

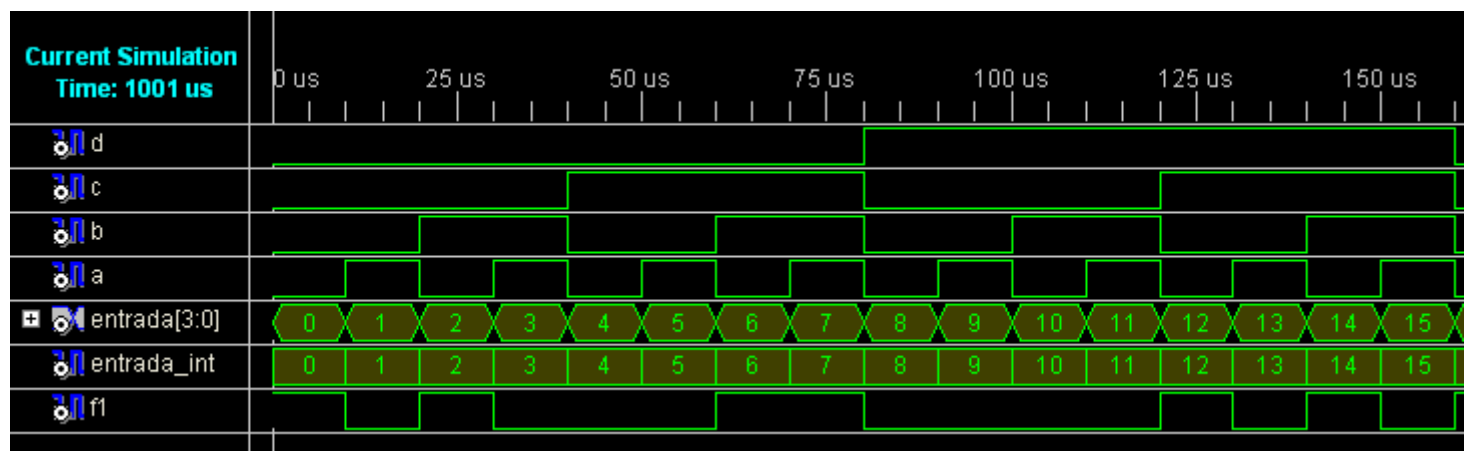
Fichero de estímulos para la simulación de la función (continuación).

```
BEGIN
  -- Instantiate the Unit Under Test (UUT)
  uut: funcion1 PORT MAP (
    d => d,
    c => c,
    b => b,
    a => a,
    f1 => f1
  );
  -- Stimulus process
  estimulos: process
  begin
    for i in 0 to 15 loop
      entrada_int <= i; wait for 10 us;
    end loop;
    wait; wait;
  end process;
  entrada <= conv_std_logic_vector(entrada_int, 4);
  a<=entrada(0); b<= entrada(1); c<= entrada(2); d<=entrada(3);
END;
```

Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

La simulación funcional puede observarse a continuación.



Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

Otra forma de definir los estímulos, sin usar el bucle FOR:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;

ENTITY funcion1_test IS
END funcion1_test;

ARCHITECTURE behavior OF funcion1_test IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT funcion1
    PORT(
        d : IN  std_logic;
        c : IN  std_logic;
        b : IN  std_logic;
        a : IN  std_logic;
        f1 : OUT std_logic
    );
    END COMPONENT;

    --Inputs
    signal d : std_logic := '0';
    signal c : std_logic := '0';
    signal b : std_logic := '0';
    signal a : std_logic := '0';
    signal entrada : std_logic_vector(3 downto 0) := "0000";
    --Outputs
    signal f1 : std_logic;
```

(Continúa en la siguiente diapositiva)

Tema 4: Sistemas Combinacionales

Anexo I: Introducción a la simulación usando VHDL

Otra forma de definir los estímulos, sin usar el bucle FOR (continuación):

```
BEGIN

    -- Instantiate the Unit Under Test (UUT)
    uut: function1 PORT MAP (
        d => d,
        c => c,
        b => b,
        a => a,
        f1 => f1
    );

    -- Stimulus process
    stim_proc: process
    begin

        if now < 10 us then entrada <= "0000"; wait for 10 us;
        else entrada <= entrada + 1; wait for 10 us; end if;
        if now > 170 us then entrada <= "0000"; wait; wait; end if;

    end process;
    a<=entrada(0); b<= entrada(1); c<= entrada(2); d<=entrada(3);
END;
```