

Sumador completo de un bit - Versión VHDL Sintético

Federico Paredes¹, Daniel Durán¹

¹ Becarios de investigación

Importante:

- Verifique la instalación del **Webpack 7.1** y el **Modelsim XE 7.1 SE**. Ambos programas se pueden bajar desde www.xilinx.com.
- Verifique también la licencia del **Modelsim** a través del programa **Licensing Wizard** provisto con la instalación.
- Cuando comience a trabajar en el proyecto recuerde grabar periódicamente.

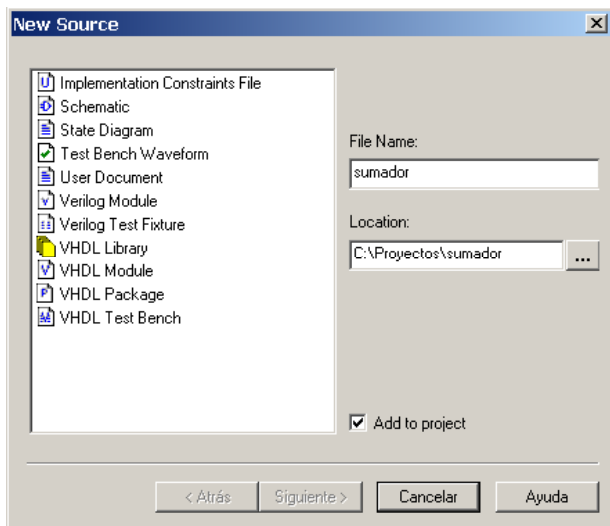
- 1) Abra el **Project Navigator**.
- 2) Sobre el menú **File** haga clic en **New Project**.
- 3) Complete la primera ventana con la información mostrada y haga clic en **Siguiente**.

- 4) Rellene la información de la segunda ventana de acuerdo a lo mostrado debajo y haga clic en **Siguiente**.

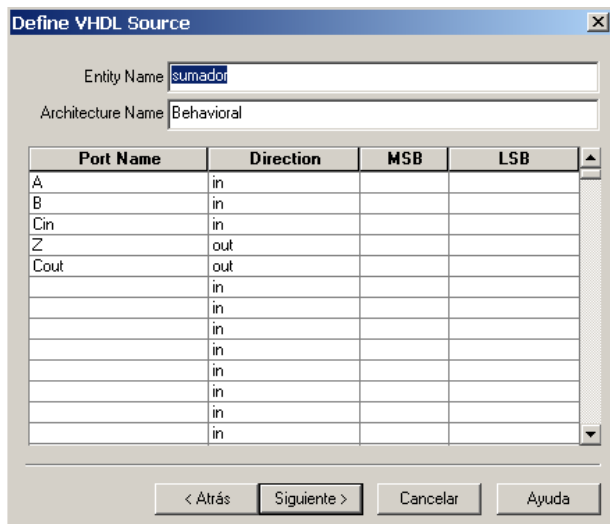
Property Name	Value
Device Family	xC9500 CPLDs
Device	xc9572
Package	PC44
Speed Grade	-7
Top-Level Module Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	Modelsim
Generated Simulation Language	VHDL

Navigation buttons at the bottom: '< Atrás', 'Siguiente >', 'Cancelar', and 'Ayuda'.

- 5) Pulse sobre **Siguiente** en las dos ventanas posteriores a la de arriba y **Finalizar** en la última.
- 6) Haga clic sobre el menú **Project** y luego en **New Source**.
- 7) En la ventana desplegada complete el nombre del archivo, seleccione **VHDL module** de la lista de la izquierda y haga clic en **Siguiente**.



8) Luego defina las entradas y salidas del sumador de acuerdo a lo mostrado en la ventana de abajo y haga clic en **Siguiente** para después hacer clic en **Finalizar**.



9) Observará a continuación el código generado automáticamente más las líneas agregadas que hacen al sumador. Copie esas dos líneas a su proyecto (delimitadas aquí entre líneas de asteriscos) y guarde el archivo.

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

-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
-- library UNISIM;
-- use UNISIM.VComponents.all;

entity sumador is
    Port ( A : in std_logic;
          B : in std_logic;
          Z : out std_logic;
          Cin : in std_logic;
          Cout : out std_logic);
end sumador;
```

architecture Behavioral of sumador is
begin

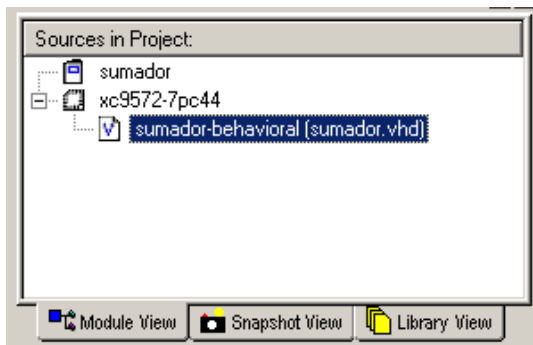
```
-- Insertar las siguientes dos líneas en el código
-- *****
  Z <= A XOR B XOR Cin;                                -- Calcula la salida
  Cout <= (A AND B) OR (Cin AND (A OR B));              -- Calcula el acarreo
-- *****

end Behavioral;
```

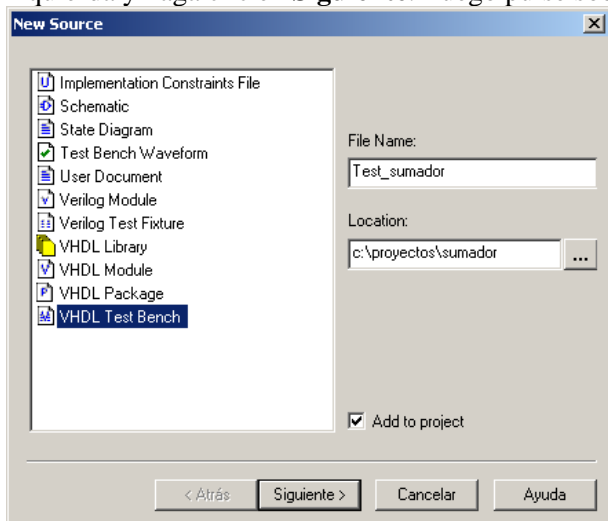
Simulación

Se utilizara un archivo fuente **VHDL TestBench** para la simulación del archivo sintetizable **VHDL module** anterior.

10) A continuación haga clic sobre el archivo **sumador** de la ventana **Sources in Project**, y luego clic con el botón derecho para seleccionar **New Source**.



11) En la ventana desplegada complete el nombre del archivo, seleccione **VHDL TestBench** de la lista de la izquierda y haga clic en **Siguiente**. Luego pulse sobre **Siguiente** y por último en **Finalizar**.



12) Observará a continuación el código generado automáticamente más las líneas agregadas que hacen al testbench. Copie esas líneas a su proyecto (delimitadas aquí entre líneas de asteriscos) y guarde el archivo.

```
-- VHDL Test Bench Created from source file sumador.vhd -- 00:31:49 04/26/2006
```

```

--
-- Notes:
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test.  Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guarantee that the testbench will bind correctly to the post-implementation
-- simulation model.
--
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
-- *****
USE ieee.std_logic_arith.ALL;
-- *****

ENTITY sumador_Test_sumador_vhd_tb IS
END sumador_Test_sumador_vhd_tb;

ARCHITECTURE behavior OF sumador_Test_sumador_vhd_tb IS

    COMPONENT sumador
    PORT(
        A : IN std_logic;
        B : IN std_logic;
        Cin : IN std_logic;
        Z : OUT std_logic;
        Cout : OUT std_logic
    );
    END COMPONENT;

    SIGNAL A : std_logic;
    SIGNAL B : std_logic;
    SIGNAL Cin : std_logic;
    SIGNAL Z : std_logic;
    SIGNAL Cout : std_logic;

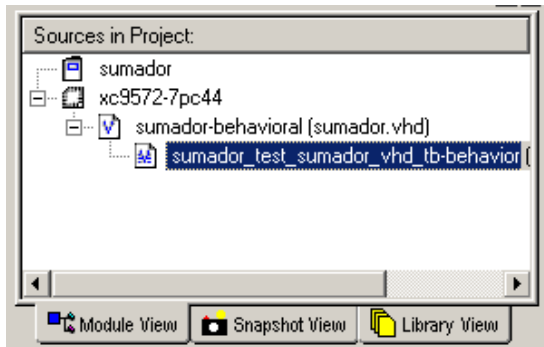
BEGIN

    uut: sumador PORT MAP(
        A => A,
        B => B,
        Cin => Cin,
        Z => Z,
        Cout => Cout
    );

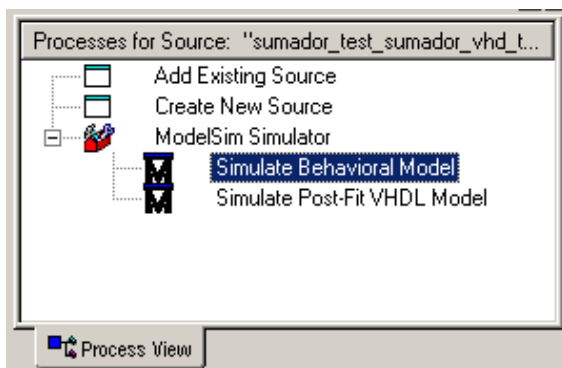
-- *** Test Bench - User Defined Section ***
tb : PROCESS
BEGIN
-- Bucle de prueba
-- *****
    for X in 0 to 7 loop
        A <= CONV_STD_LOGIC_VECTOR(X,3)(0); -- Bucle que genera una tabla
        B <= CONV_STD_LOGIC_VECTOR(X,3)(1); -- de tres variables, A, B y Cin
        Cin <= CONV_STD_LOGIC_VECTOR(X,3)(2);
        wait for 10 ns;
    end loop;
-- *****
    wait; -- will wait forever
END PROCESS;
-- *** End Test Bench - User Defined Section ***

END;
```

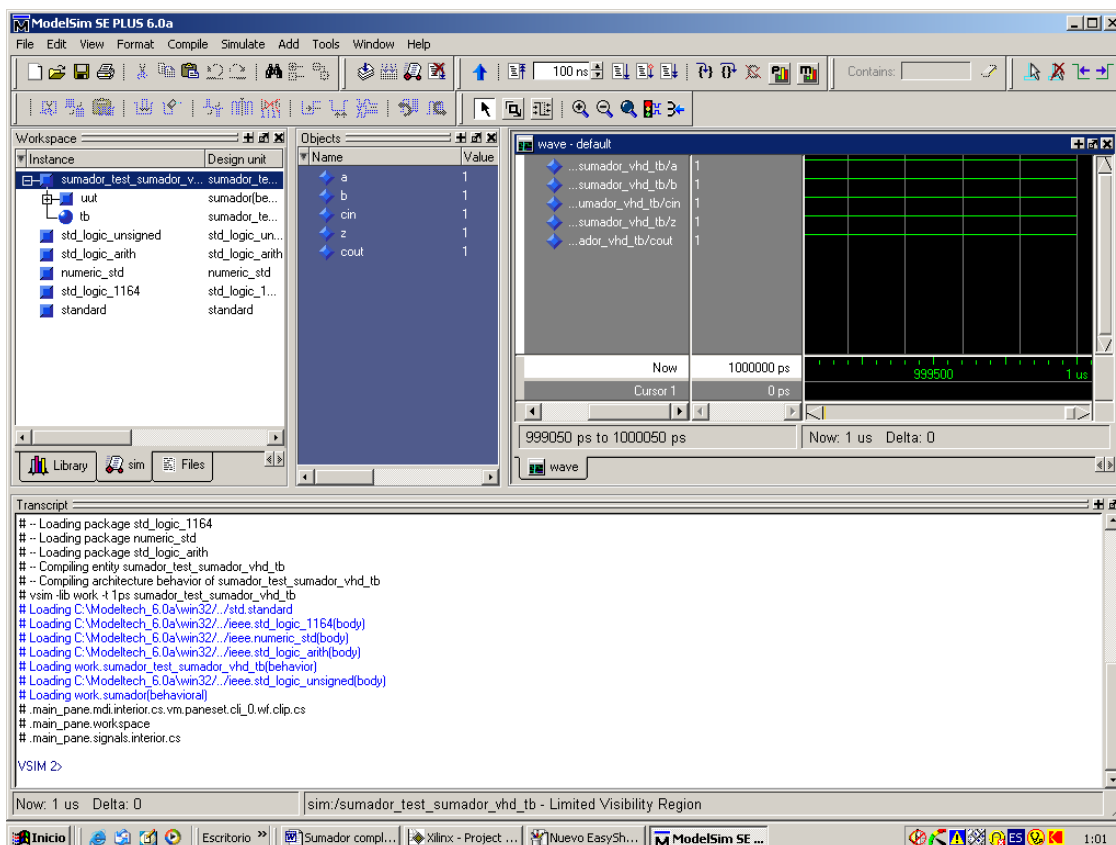
13) Haga clic sobre el archivo del banco de pruebas en la ventana **Sources in Project**.



14) Luego pulse sobre **Simulate Behavioral Model** en la ventana **Processes for Source**.

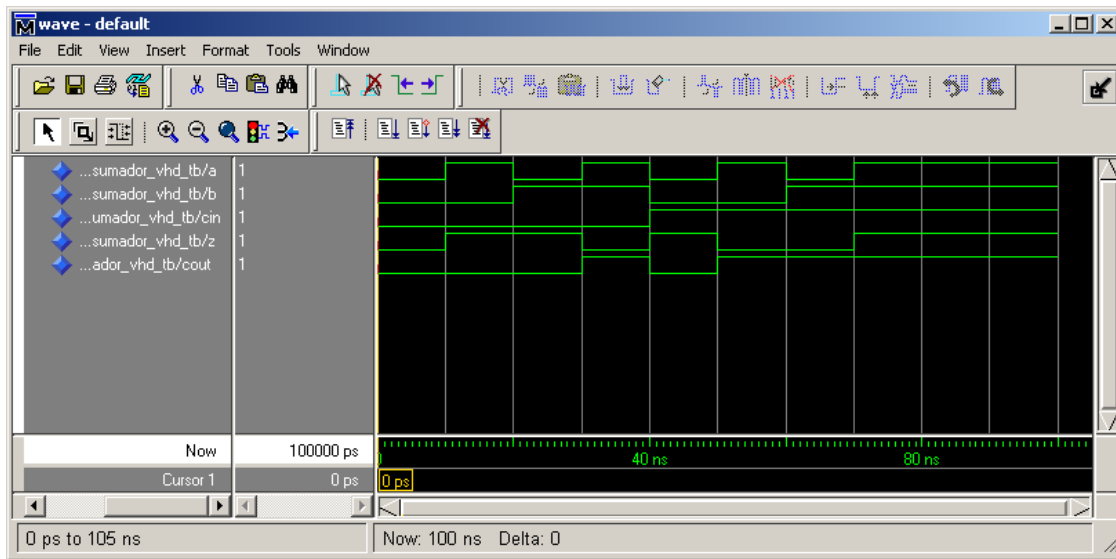


15) Observará la apertura del simulador Modelsim:



16) Haga clic la ventana **wave** situada en la zona media-superior derecha.

17) Pulse sobre el ícono de la lupa oscura (**Zoom Full**), situado junto al ícono del semáforo y utilice los botones de zoom junto a la barra de desplazamiento horizontal para obtener una imagen apreciable del funcionamiento del sumador, como se muestra a continuación.



18) De no lograr el objetivo esperado verifique los pasos realizados según lo establece este sencillo tutorial.