

Introducción a VHDL

Federico G. Zacchigna

FIUBA

13/01/2017

Outline

HDLs

Tipos de datos en VHDL

Descripción de una entidad

Testbench

Lenguajes HDL

Es un lenguajes utilizaado para una descripción formal del hardware.

- ▶ Esquemático.
- ▶ Arquitectura.
- ▶ Micro-arquitectura.
- ▶ Comportamiento.

Ejemplos

VHDL, Verilog, SystemC, SystemVerilog, AHDL.

Bibliotecas y tipos de datos

Bibliotecas típicas

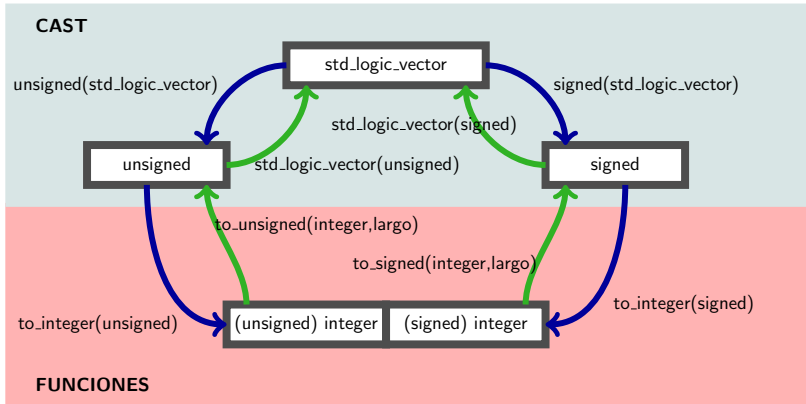
```
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.numeric.all;
```

Tipos de datos

```
std_logic  
std_logic_vector(1 downto 0)  
signed  
unsigned  
integer
```

Conversiones entre datos

Para aquellas señales que tengan más de un bit:



Es necesario incluir las bibliotecas:

```
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.numeric.all;
```

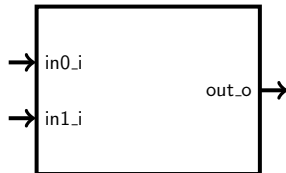
Estructura de un archivos

Se utilizará un archivo por cada bloque de diseño.

```
1  -----
2  -- bibliotecas
3  library ieee;
4  use ieee.std_logic_1164.all;
5  use ieee.numeric_std.all;
6  -----
7
8  -----
9  -- se definen el nombre los puertos de la ENTIDAD
10 entity andgate is
11     -- puertos de la arquitectura
12
13 end entity andgate;
14 -----
15
16 -----
17 -- se define la ARQUITECTURA del bloque
18 architecture rtl of andgate is
19     -- declaracion de variables internas
20 begin
21     -- comportamiento/estructura del bloque
22
23 end architecture rtl;
24 -----
```

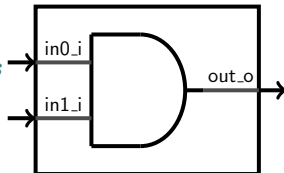
Puertos de entrada y salida

```
1  -- entidad
2  entity andgate is
3      port (
4          in0_i : in  std_logic;
5          in1_i : in  std_logic;
6          out_o  : out std_logic
7      );
8  end entity andgate;
```



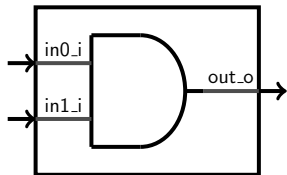
Descripción del comportamiento

```
1  -- arquitetura
2  architecture rtl of andgate is
3      -- declaracion de senales,
4      -- variables y constantes internas
5  begin
6      -- descripcion de la arquitectura
7      out_o <= in0_i and in1_i;
8  end architecture rtl;
```



Bloque AND completo

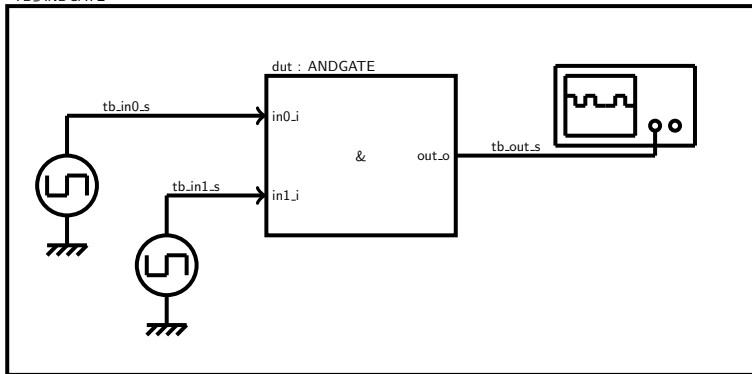
```
1  -- bibliotecas
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5
6  -- entidad
7  entity andgate is
8      port (
9          in0_i : in  std_logic;
10         in1_i : in  std_logic;
11         out_o  : out std_logic
12     );
13 end entity andgate;
14
15 -- arquitectura
16 architecture rtl of andgate is
17     -- declaracion de senales,
18     -- variables y constantes internas
19 begin
20     -- descripcion de la arquitectura
21     out_o <= in0_i and in1_i;
22 end architecture rtl;
```



Testbench para compuerta AND

- ▶ Para testear otros bloques.
- ▶ Se generan señales de pruebas que se conectan a las entradas del DUT.
- ▶ Se observan las salidas del DUT.

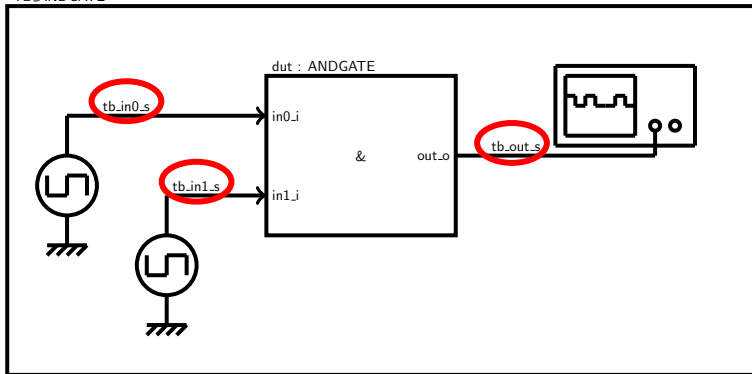
TB_ANDGATE



Testbench para compuerta AND

- ▶ Para testear otros bloques.
- ▶ Se generan señales de pruebas que se conectan a las entradas del DUT.
- ▶ Se observan las salidas del DUT.

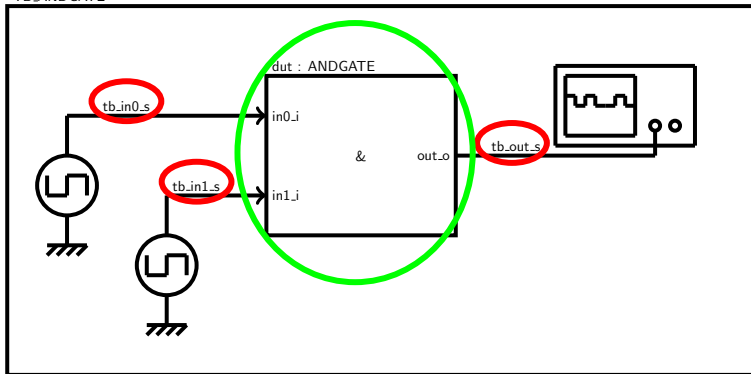
TB_ANDGATE



Testbench para compuerta AND

- ▶ Para testear otros bloques.
- ▶ Se generan señales de pruebas que se conectan a las entradas del DUT.
- ▶ Se observan las salidas del DUT.

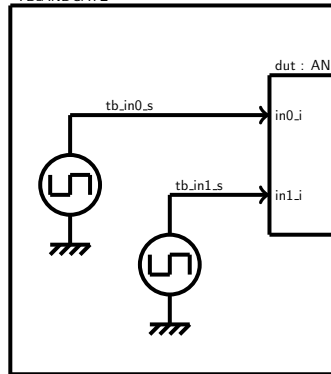
TB_ANDGATE



Testbench para compuerta AND

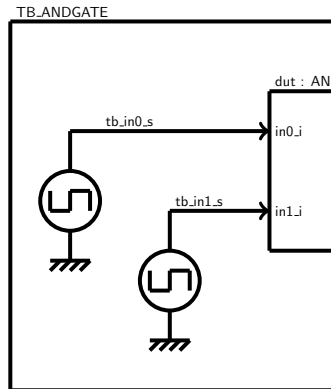
```
1  -- bibliotecas
2  library ieee;
3  use ieee.std_logic_1164.all;
4
5  -- entidad
6  entity tb_andgate is
7  end entity tb_andgate;
8
9  -- arquitetura
10 architecture rtl of tb_andgate is
11     component andgate is
12     port (
13         in0_i : in  std_logic;
14         in1_i : in  std_logic;
15         out_o  : out std_logic
16     );
17     end component andgate;
18     signal tb_in0_s : std_logic;
19     signal tb_in1_s : std_logic;
20     signal tb_out_s : std_logic;
21 begin
```

TB_ANDGATE



Testbench para compuerta AND

```
21 begin
22
23     dut : andgate
24     port map(
25         in0_i => tb_in0_s,
26         in1_i => tb_in1_s,
27         out_o => tb_out_s
28     );
29
30     -- asignacion de senales de entrada
31     -- senal 0
32     tb_in0_s <= '0',
33                '1' after 10 ns,
34                '0' after 23 ns;
35
36     -- senal 1
37     tb_in1_s <= '1',
38                '0' after 17 ns,
39                '1' after 27 ns;
40
41 end architecture rtl;
```

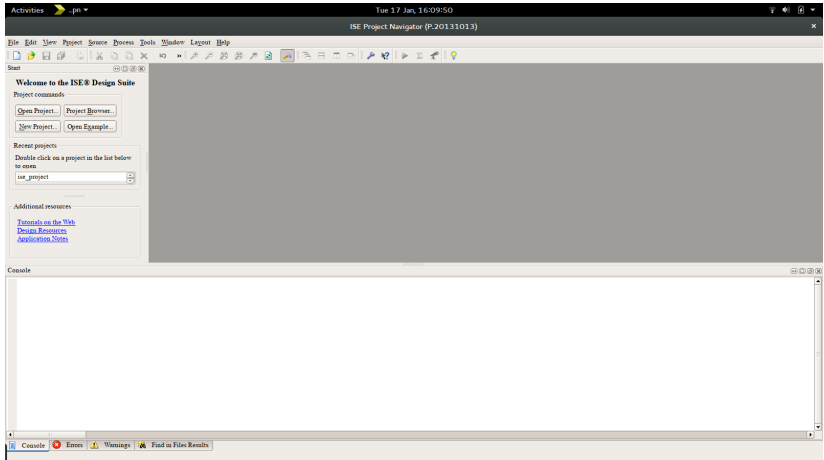


Testbench para compuerta AND

```
1  -- bibliotecas
2  library ieee;
3  use ieee.std_logic_1164.all;
4
5  -- entidad
6  entity tb_andgate is
7  end entity tb_andgate;
8
9  -- arquitectura
10 architecture rtl of tb_andgate is
11     component andgate is
12     port (
13         in0_i : in  std_logic;
14         in1_i : in  std_logic;
15         out_o  : out std_logic
16     );
17 end component andgate;
18 signal tb_in0_s : std_logic;
19 signal tb_in1_s : std_logic;
20 signal tb_out_s : std_logic;
21
22 begin
23     -- device under test
24     dut : andgate
25     port map(
26         in0_i => tb_in0_s,
27         in1_i => tb_in1_s,
28         out_o => tb_out_s
29     );
30
31     -- asignacion de senales de entrada
32     -- senal 0
33     tb_in0_s <= '0',
34                '1' after 10 ns,
35                '0' after 23 ns;
36
37     -- senal 1
38     tb_in1_s <= '1',
39                '0' after 17 ns,
40                '1' after 27 ns;
41
42 end architecture rtl;
```

Simulación del testbench

Abrimos el ise.

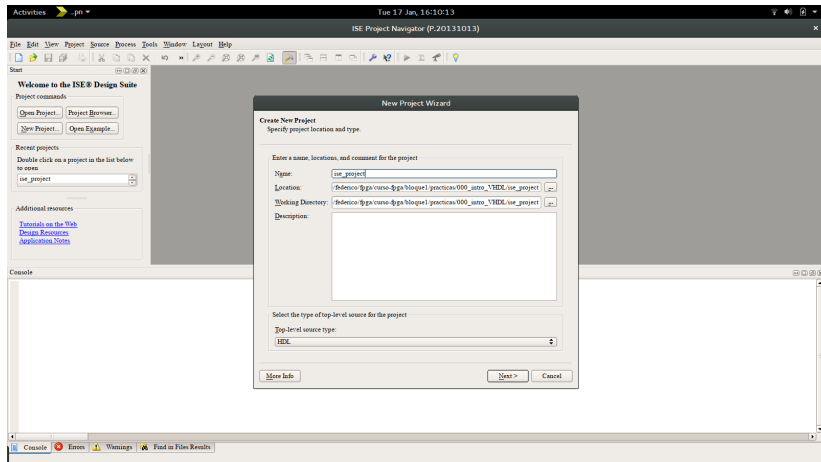


Simulación de testbench

Luego “File → New Project ...”

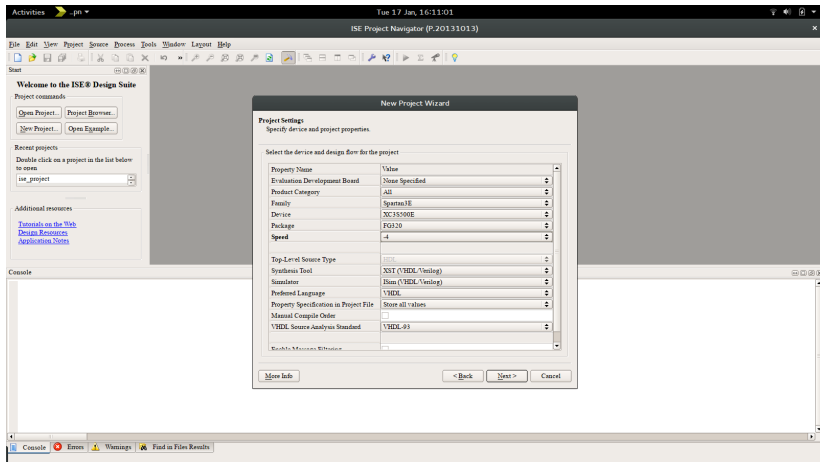
Elegimos directorio de trabajo

Elegimos nombre del proyecto



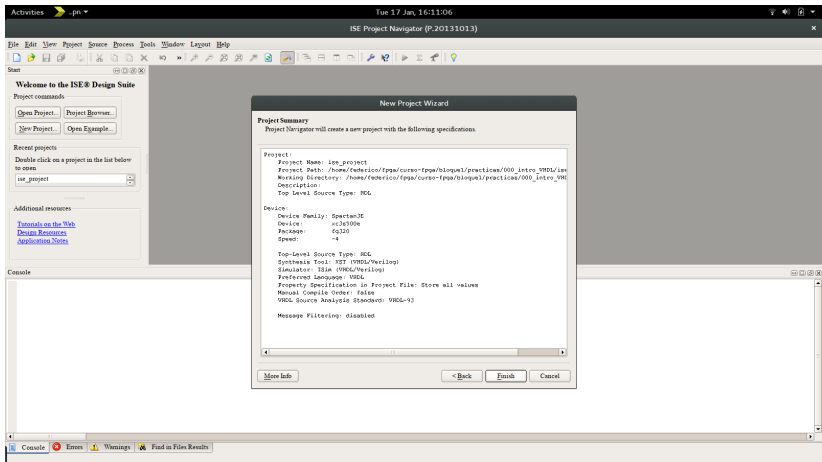
Simulación del testbench

Elegimos el kit (FPGA), lenguaje y herramientas



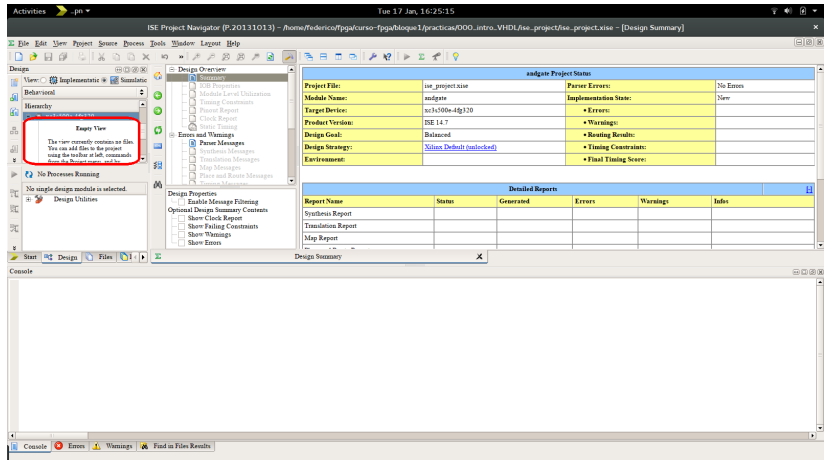
Simulación del testbench

“Finish”



Simulación del testbench

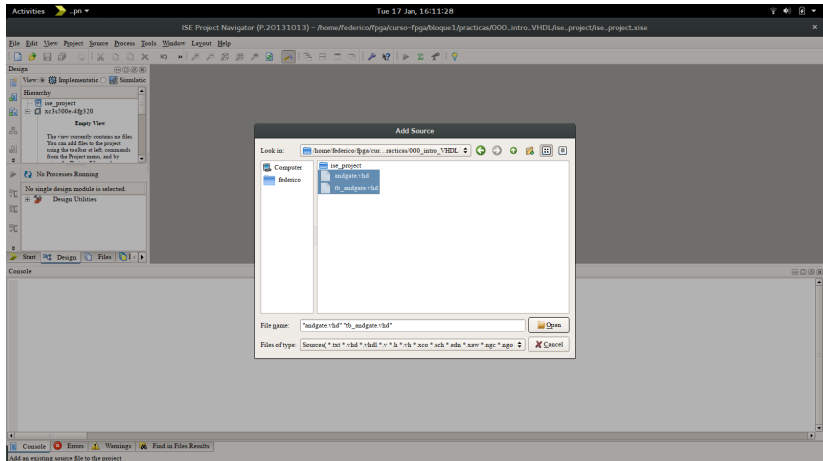
Clic derecho sobre la zona marcada y “Add Source ...”.



Simulación del testbench

Elegimos los archivos fuentes de nuestros archivos.

NOTA: El “tb_andgate.vhd”, agregarlo solo para la simulación



Simulación del testbench

Clic sobre “Simulación”

The screenshot shows the Xilinx ISE Project Navigator interface. The 'Design' tab is active, and the 'Simulation' button is highlighted with a red rectangle. The 'Design Overview' pane on the right shows the project hierarchy, including 'ise_project', 'src1500e-4dg320', and 'tdt_andgate_vhdl'. The 'Design Properties' pane on the left shows various options for enabling message filtering and showing design summary contents. The 'Design Summary' pane on the right displays the 'andgate Project Status' and 'Detailed Reports'.

andgate Project Status

| Project File: | ise_project.xise | Parser Errors: | No Errors |
|------------------|---------------------------|-----------------------|-----------|
| Module Name: | andgate | Implementation State: | New |
| Target Device: | xc3s100e-4dg320 | • Errors: | |
| Product Version: | ISE 14.7 | • Warnings: | |
| Design Goal: | Balanced | • Routing Results: | |
| Design Strategy: | Xilinx Default (unlocked) | • Timing Constraints: | |
| Environment: | | • Final Timing Score: | |

Detailed Reports

| Report Name | Status | Generated | Errors | Warnings | Infos |
|--------------------|--------|-----------|--------|----------|-------|
| Synthesis Report | | | | | |
| Translation Report | | | | | |
| Map Report | | | | | |

Console

```
INFO: HDLCompiler:1061 - Parsing VHDL file "/home/federico/fpga/cursos-fpga/bloque1/practicas/000_intro_VHDL/andgate.vhd" into library work
INFO: ProjectMgt - Parsing design hierarchy completed successfully.
INFO: HDLCompiler:1061 - Parsing VHDL file "/home/federico/fpga/cursos-fpga/bloque1/practicas/000_intro_VHDL/andgate.vhd" into library work
INFO: ProjectMgt - Launching Design Summary/Report Viewer.
INFO: HDLCompiler:1061 - Parsing VHDL file "/home/federico/fpga/cursos-fpga/bloque1/practicas/000_intro_VHDL/andgate.vhd" into library work
INFO: ProjectMgt - Parsing design hierarchy completed successfully.
```

Simulación del testbench

Clic sobre el archivo a simular: "tb_andgate.vhd"

The screenshot shows the Xilinx ISE Project Navigator interface. The 'Design Overview' pane on the left displays the project hierarchy, with 'tb_andgate.vhd' highlighted. The 'Design Summary' window on the right provides a comprehensive overview of the project status and reports.

Design Overview

- Summary
 - I/O Properties
 - Module-Level Utilization
 - Timing Constraints
 - Power Report
 - Clock Report
 - Static Timing
- Errors and Warnings
 - Parser Messages
 - Synthesis Messages
 - Translation Messages
 - Map Messages
 - Place and Route Messages

Design Properties

- Enable Message Filtering
- Optional Design Summary Contents
 - Show Clock Report
 - Show Failing Constraints
 - Show Warnings
 - Show Errors

Design Summary

andgate Project Status

| | | | |
|------------------|---------------------------|-----------------------|-----------|
| Project File: | ise_project.xise | Parser Errors: | No Errors |
| Module Name: | andgate | Implementation State: | New |
| Target Device: | xc3s100e-dg120 | • Errors: | |
| Product Version: | ISE 14.7 | • Warnings: | |
| Design Goal: | Balanced | • Routing Results: | |
| Design Strategy: | Xilinx Default (unlocked) | • Timing Constraints: | |
| Environment: | | • Final Timing Score: | |

Detailed Reports

| Report Name | Status | Generated | Errors | Warnings | Infos |
|--------------------|--------|-----------|--------|----------|-------|
| Synthesis Report | | | | | |
| Translation Report | | | | | |
| Map Report | | | | | |

Console

```
INFO: HDLCompiler:1061 - Parsing VHDL file "/home/federico/fpga/cursos-fpga/bloque1/practicas/000_intro_VHDL/ise_project/ise_project.xise" into library work
INFO: ProjectMgt - Parsing design hierarchy completed successfully.
INFO: HDLCompiler:1061 - Parsing VHDL file "/home/federico/fpga/cursos-fpga/bloque1/practicas/000_intro_VHDL/ise_project/ise_project.xise" into library work
INFO: ProjectMgt - Parsing design hierarchy completed successfully.
```

Simulación del testbench

Doble clic sobre "Behavioral Check Syntax"

Doble clic sobre "Simulate Behavioral Model"

The screenshot shows the Xilinx ISE Project Navigator interface. The top window is the "Design Summary" for the project "ise_project.xise". It displays various project details and a table of report names.

Design Summary - [Design Summary]

| andgate Project Status | | | |
|------------------------|---------------------------|-----------------------|-----------|
| Project File: | ise_project.xise | Parser Errors: | No Errors |
| Module Name: | andgate | Implementation State: | New |
| Target Device: | xc3s100e-dg320 | • Errors: | |
| Product Version: | ISE 14.7 | • Warnings: | |
| Design Goal: | Balanced | • Routing Results: | |
| Design Strategy: | Xilinx Default (unlocked) | • Timing Constraints: | |
| Environment: | | • Final Timing Score: | |

| Detailed Reports | | | | | |
|--------------------|--------|-----------|--------|----------|-------|
| Report Name | Status | Generated | Errors | Warnings | Infos |
| Synthesis Report | | | | | |
| Translation Report | | | | | |
| Map Report | | | | | |

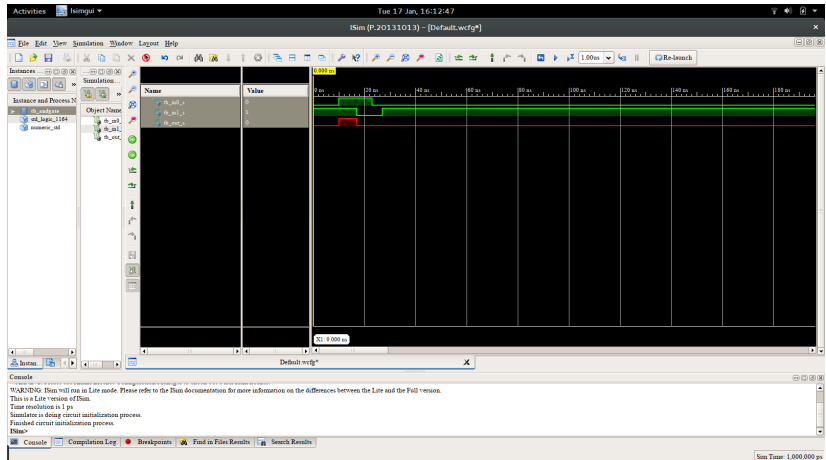
The "Run Simulator" process is highlighted with a red box in the "Processes" pane. The console window at the bottom shows the output of the "Behavioral Check Syntax" process, which completed successfully.

```
Started: "Behavioral Check Syntax"  
Determining files marked for global include in the design...  
Running vbcprep...  
Command Line: vbcprep work isia_temp -intstyle ise -prj /home/federico/fpga/cursos-fpga/bioguel/practicas/000_intro_VHDL/ise_project/tb_andgate_str_beh.prj  
Determining compilation order of VHDL files  
Parsing VHDL file "/home/federico/fpga/cursos-fpga/bioguel/practicas/000_intro_VHDL/ise_project/..andgate.vhd" into library isia_temp  
Parsing VHDL file "/home/federico/fpga/cursos-fpga/bioguel/practicas/000_intro_VHDL/ise_project/..tb_andgate.vhd" into library isia_temp  
Process "Behavioral Check Syntax" completed successfully
```


Simulación del testbench

Se abre el ISim.

Comprobamos que la salida de nuestra compuerta sea correcta
Solo está en '1', cuando las dos entradas están en '1'.



Compuerta AND completa de 1 y 4 entradas

```
1  -- bibliotecas
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5
6  -- entidad
7  entity andgate is
8
9      port (
10         in0_i : in  std_logic;
11
12         in1_i : in  std_logic;
13
14         out_o : out std_logic
15
16     );
17 end entity andgate;
18
19 -- arquitectura
20 architecture rtl of andgate is
21 begin
22     out_o <= in0_i and in1_i;
23 end architecture rtl;
```

```
1  -- bibliotecas
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5
6  -- entidad
7  entity andgate4 is
8
9      port (
10         in0_i : in  std_logic_vector(
11             3 downto 0);
12         in1_i : in  std_logic_vector(
13             3 downto 0);
14         out_o : out std_logic_vector(
15             3 downto 0)
16     );
17 end entity andgate4;
18
19 -- arquitectura
20 architecture rtl of andgate4 is
21 begin
22     out_o <= in0_i and in1_i;
23 end architecture rtl;
```

Compuerta AND completa de 4 y N entradas

```
1  -- bibliotecas
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5
6  -- entidad
7  entity andgate4 is
8
9      port (
10         in0_i : in  std_logic_vector(
11             3 downto 0);
12         in1_i : in  std_logic_vector(
13             3 downto 0);
14         out_o : out std_logic_vector(
15             3 downto 0)
16     );
17 end entity andgate4;
18
19 -- arquitectura
20 architecture rtl of andgate4 is
21 begin
22     out_o <= in0_i and in1_i;
23 end architecture rtl;
```

```
1  -- bibliotecas
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5
6  -- entidad
7  entity andgateN is
8      generic (N : integer);
9      port (
10         in0_i : in  std_logic_vector(
11             N-1 downto 0);
12         in1_i : in  std_logic_vector(
13             N-1 downto 0);
14         out_o : out std_logic_vector(
15             N-1 downto 0)
16     );
17 end entity andgateN;
18
19 -- arquitectura
20 architecture rtl of andgateN is
21 begin
22     out_o <= in0_i and in1_i;
23 end architecture rtl;
```