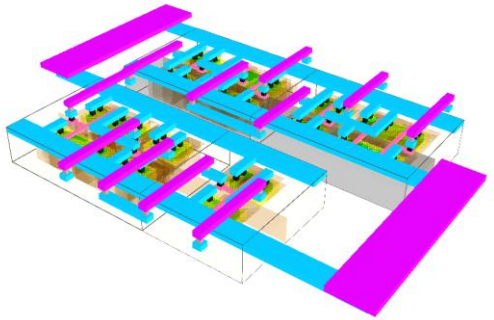
Electric VLSI Design System

Liliana Arias Torres.
Walter Alberto Pulido Chiguasuque.

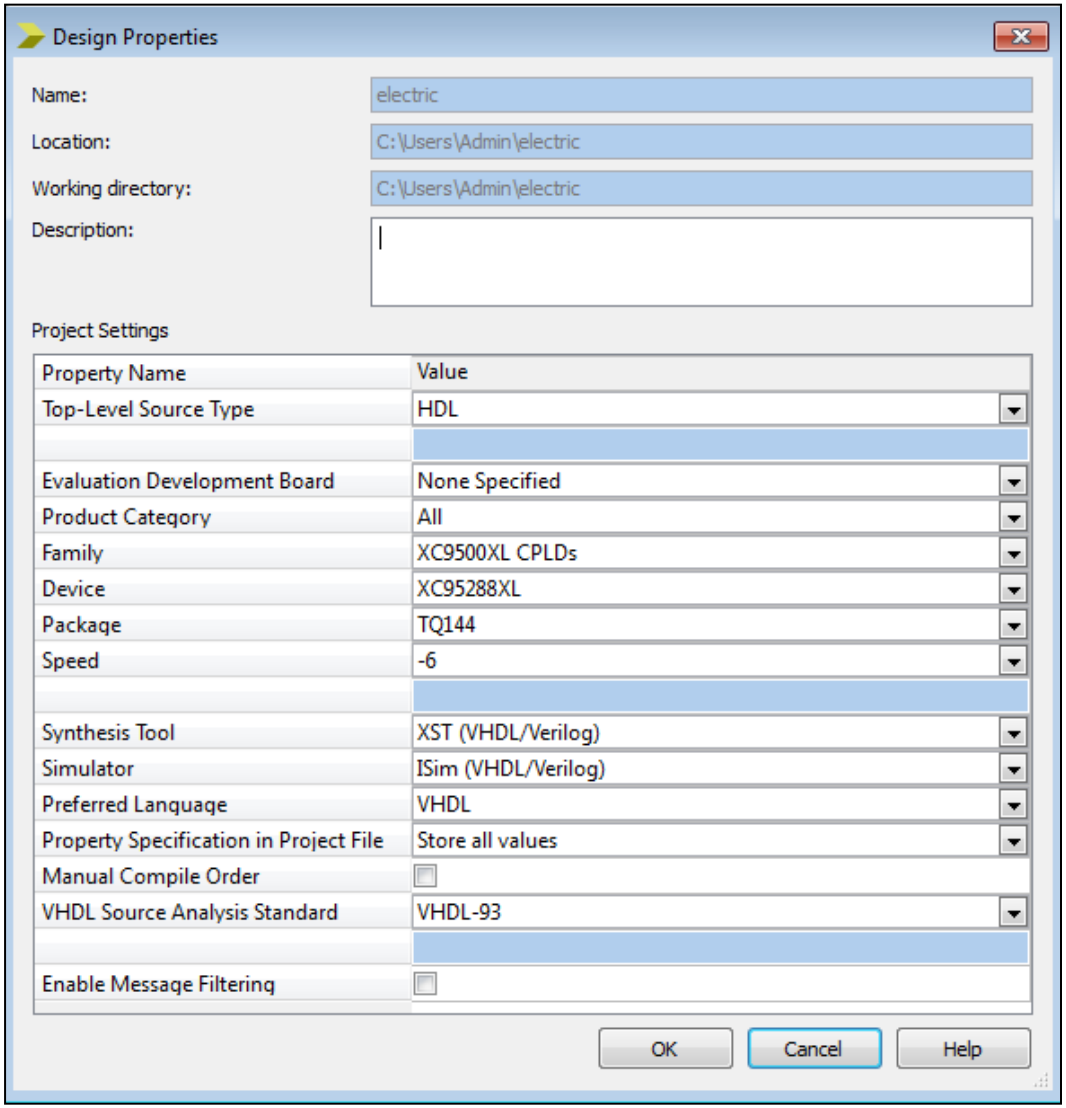
Director : Sebastian Eslava Garzón PhD.

Departamento de Ingeniería Eléctrica y Electrónica.
Universidad Nacional de Colombia



Diseño semi-custom de un módulo realizado en Xilinx con código HDL comportamental

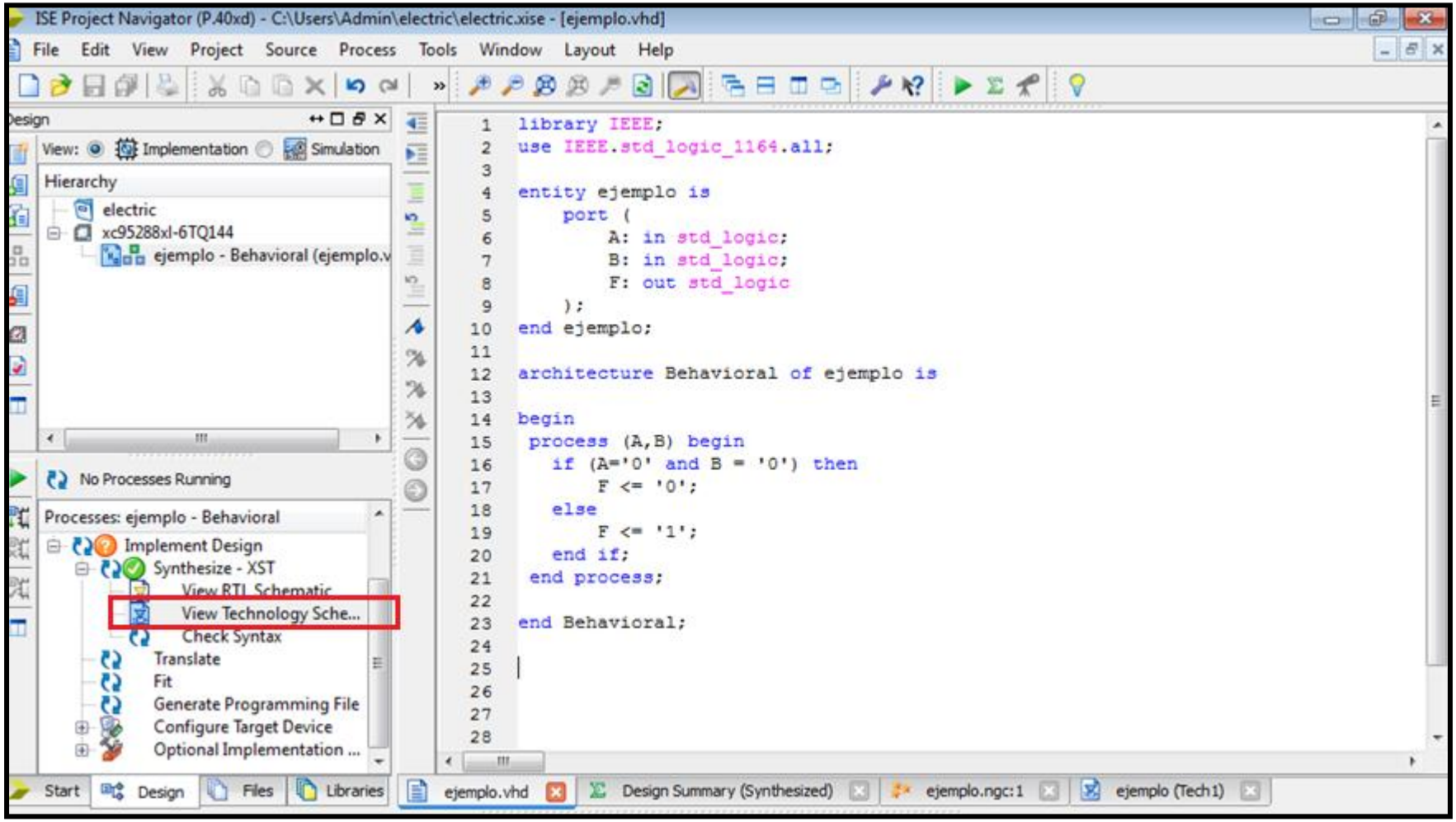
- Crear un proyecto nuevo en Xilinx con el nombre ejemplo.
- Asegúrese de elegir la familia CPLD para sintetizar el proyecto.



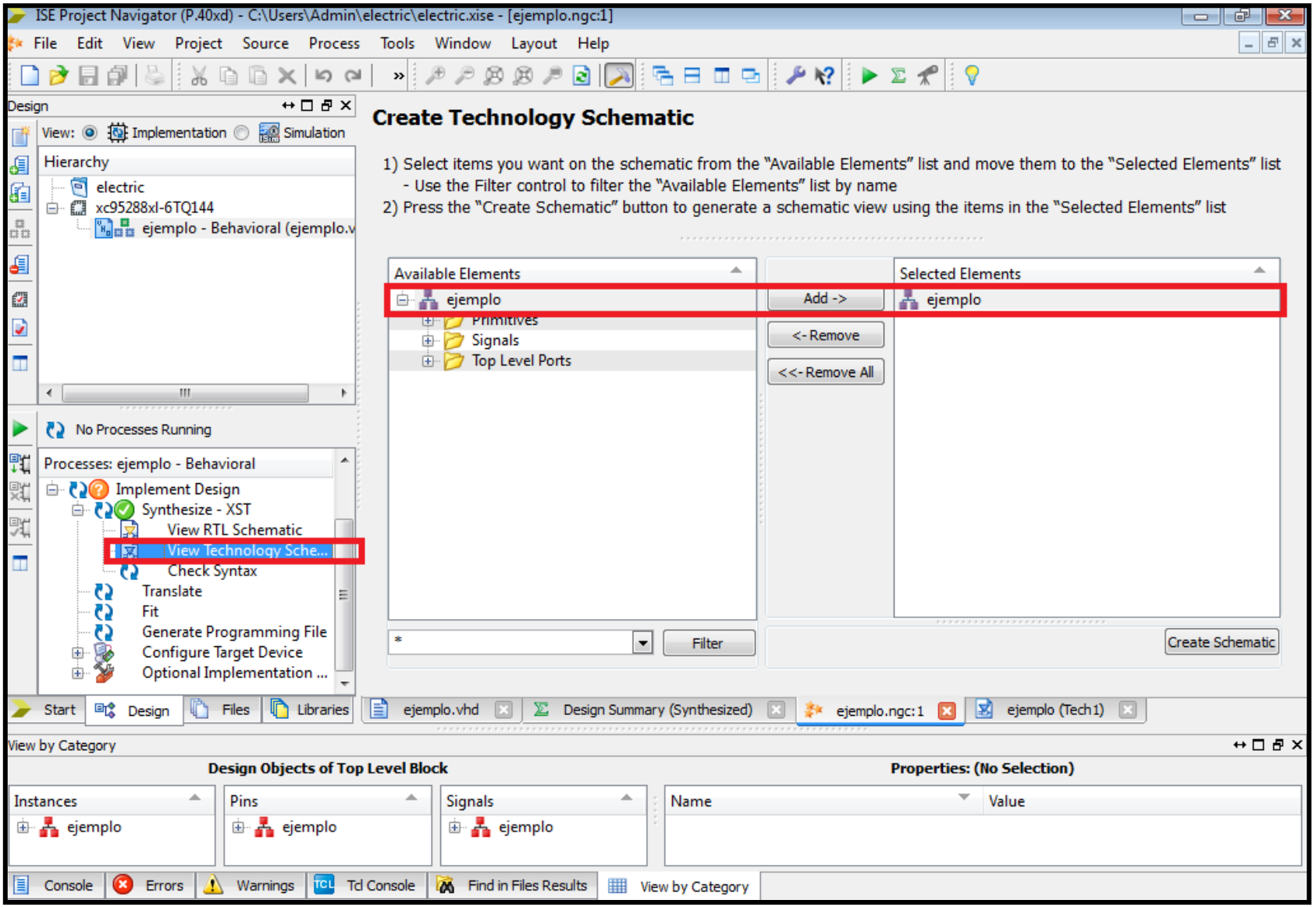
- Vamos a realizar como ejemplo una compuerta OR de 2 entradas utilizando el siguiente código.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity ejemplo is
    port (
        A: in std_logic;
        B: in std_logic;
        F: out std_logic
    );
end ejemplo;

architecture Behavioral of ejemplo is

begin
 process (A,B) begin
   if (A='0' and B = '0') then
       F <= '0';
   else
       F <= '1';
   end if;
 end process;

end Behavioral;
```

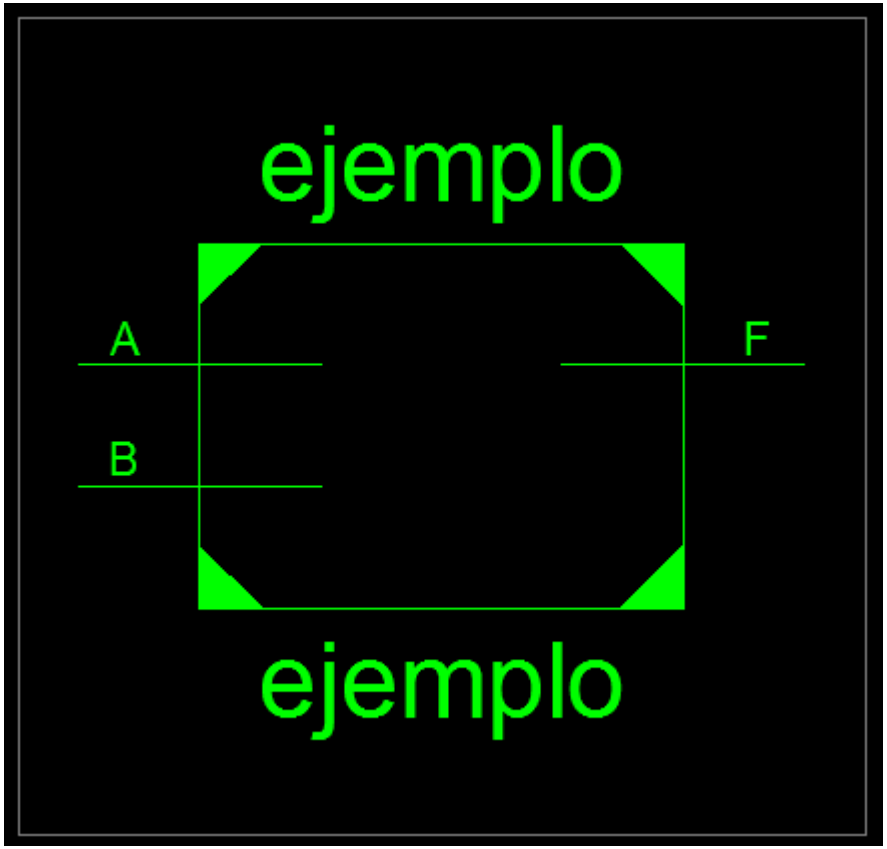
- Para obtener el RTL sintetizar el ejemplo y dar clic en la opción: **View Technology Sche..**



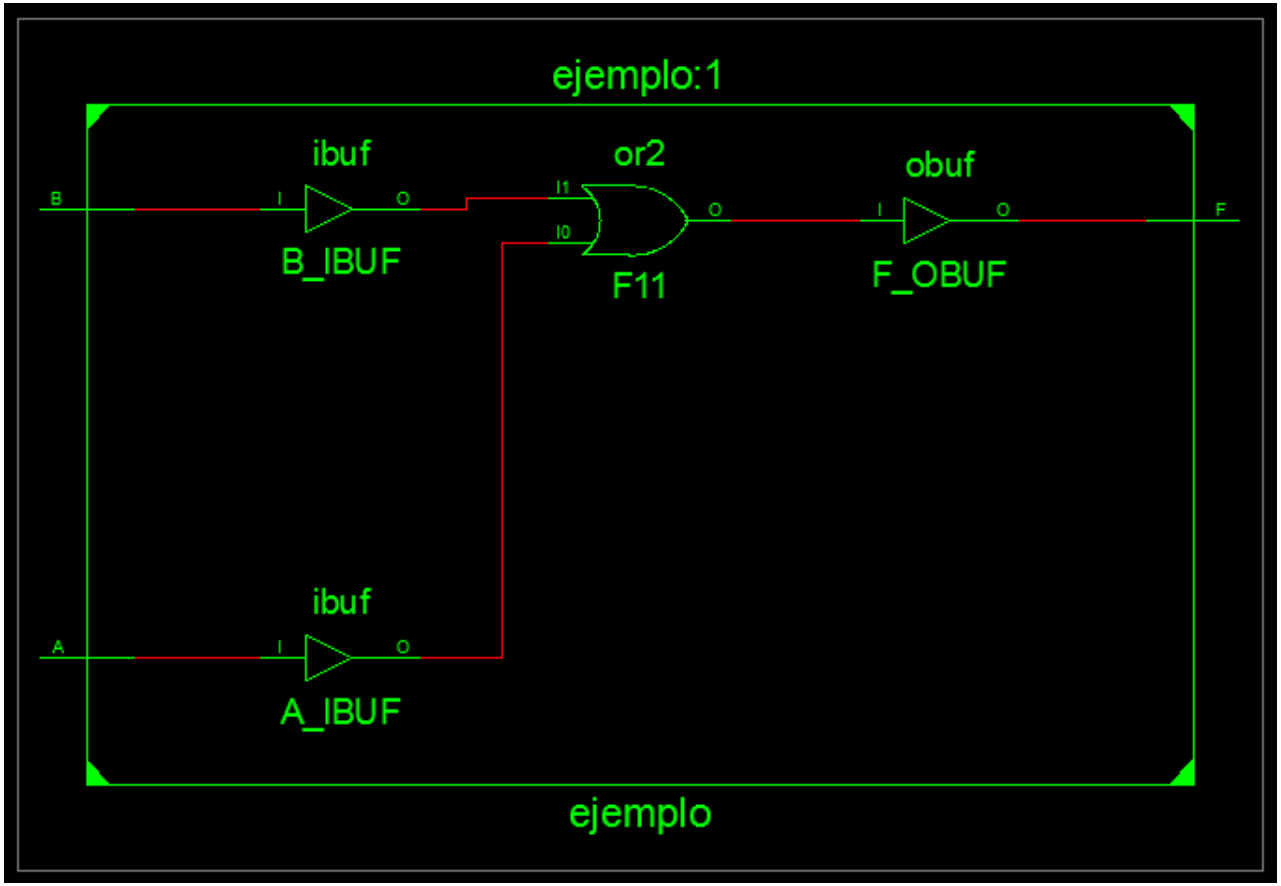
- Crear esquemático



- Se obtiene el diagrama RTL.



- Con la descripción RTL se pueden observar los componentes que deben ser instanciados como celdas estándar en la herramienta Electric VLSI.
- Realizar la lista de celdas



Lista de celdas

- Or 2
- Buffer

Conversión del código HDL comportamental a HDL estructural

- Abrir una Terminal en Linux ó para Windows:

  Inicio → Ejecutar → cmd.
- Ubicarse en el directorio donde está el archivo .ngc que se desea convertir.
  - En este caso ejemplo.ngc se encuentra en:
    C:\Users\Admin\Documents
- Para generar un archivo vhdl estructural escribir:

  netgen -ofmt vhdl mi_archivo.ngc
- Para generar un archivo verilog estructural escribir:

  netgen -ofmt verilog mi_archivo.ngc

Nota: Si el proyecto realizado en Xilinx está en código verilog comportamental se puede generar un archivo vhdl estructural.

- Fuente: http://hdl-fpga.blogspot.com/2011/10/mensaje-de-error-cant-find-net-or.html

- En caso de obtener un mensaje de error diciendo que no reconoce como ejecutable al programa 'netgen', deberá anteponerle el camino del directorio donde reside el ejecutable.

- En resumen los comandos que debe ejecutar son:

1. C:\Users\Admin>cd C:\Users\Admin\Documents
2. C:\Users\Admin\Documents>C:\Xilinx\14.3\ISE_DS\ISE\bin\nt\netgen -ofmt vhdl ejemplo.ngc

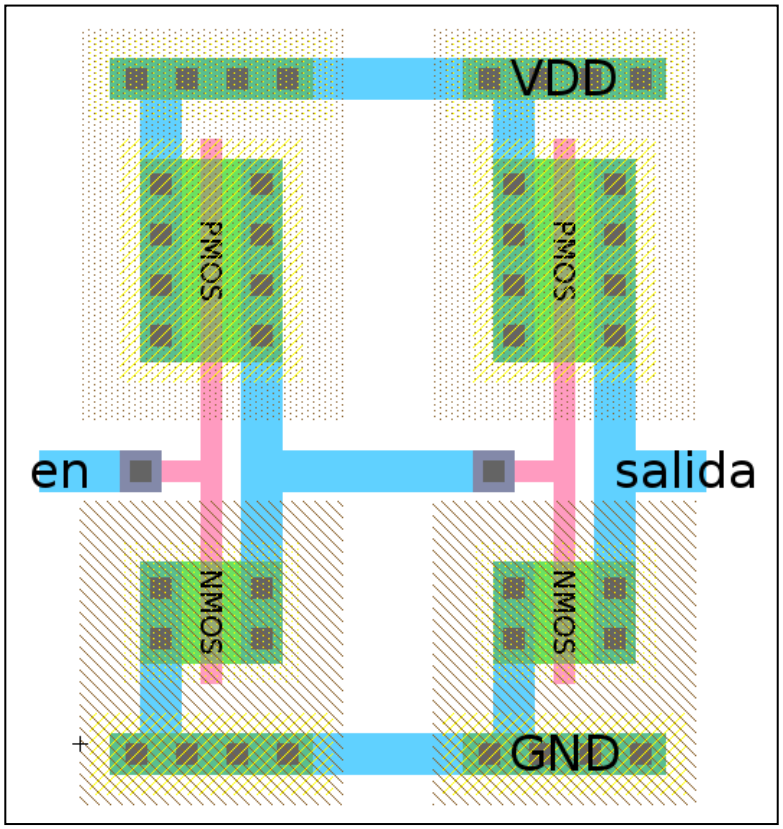
   Esto nos genera un archivo .nlf
3. C:\Users\Admin\Documents>C:\Xilinx\14.3\ISE_DS\ISE\bin\nt\netgen -ofmt ejemplo.ngc ejemplo.vhd

- En el código VHDL estructural que fue generado también podemos observar la lista de celdas IBUF y OBUF corresponden al mismo buffer.

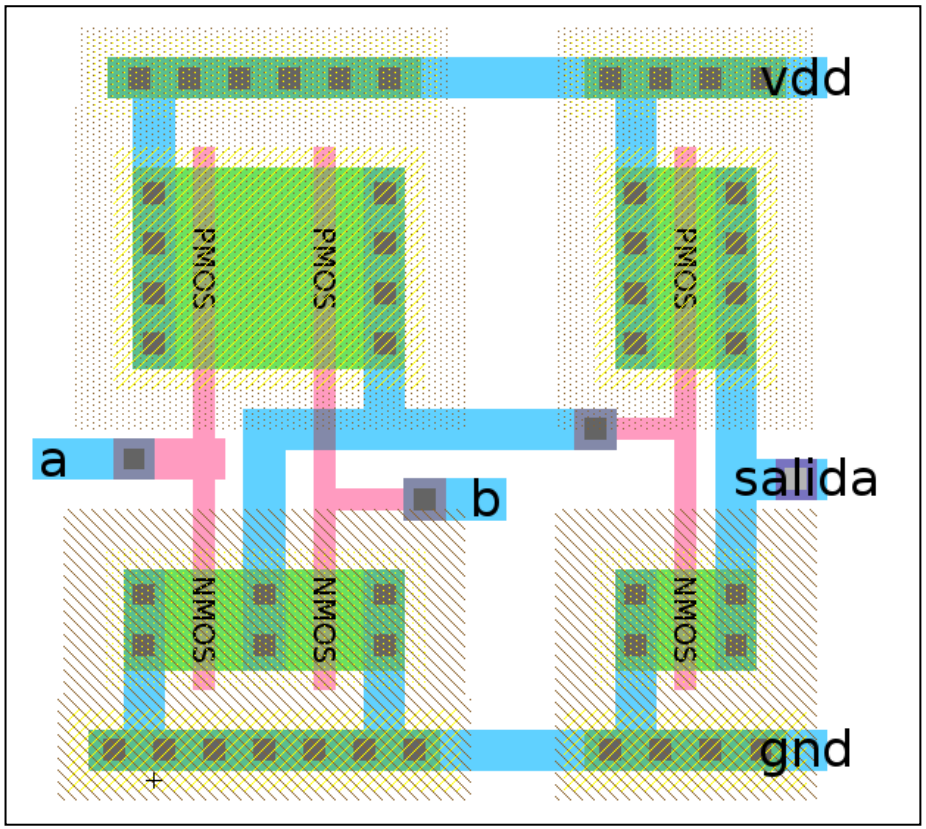
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
use UNISIM.VPKG.ALL;

entity ejemplo is
  port (
    A : in STD_LOGIC := 'X';
    B : in STD_LOGIC := 'X';
    F : out STD_LOGIC
  );
end ejemplo;

architecture STRUCTURE of ejemplo is
  signal A_IBUF_1 : STD_LOGIC;
  signal B_IBUF_3 : STD_LOGIC;
  signal F_OBUF_5 : STD_LOGIC;

begin
  F11 : OR2
    port map (
      I0 => A_IBUF_1,
      I1 => B_IBUF_3,
      O => F_OBUF_5
    );
  A_IBUF : IBUF
    port map (
      I => A,
      O => A_IBUF_1
    );
  B_IBUF : IBUF
    port map (
      I => B,
      O => B_IBUF_3
    );
  F_OBUF : OBUF
    port map (
      I => F_OBUF_5,
      O => F
    );

end STRUCTURE;
```

Elaboración de la librería de celdas estándar en la herramienta Electric.

- Realizar un proyecto con la librería de celdas estándar según la lista obtenida en el código HDL estructural.



BUFFER

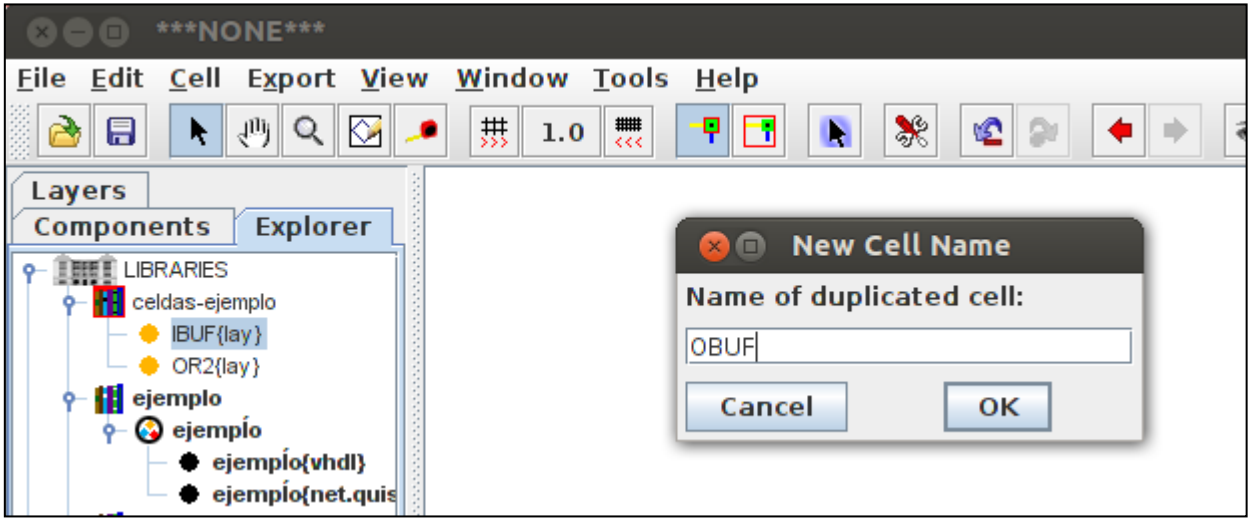


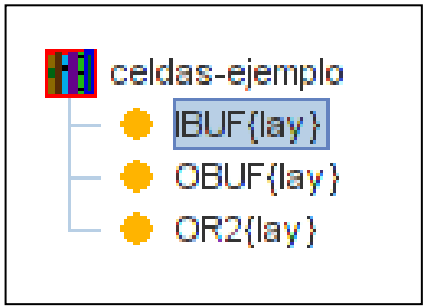
OR2

- Como las celdas IBUF y OBUF corresponden al mismo buffer debe duplicar la celda:

  Clic derecho Duplicate Cell

- Las celdas deben ser nombradas según son instanciadas por el código VHDL estructural.





Creación del proyecto VHDL en la herramienta Electric.

- Cree una nueva librería llamada ejemplo.
- Dentro de su librería cree una nueva celda tipo VHDL con el mismo nombre del proyecto en Xilinx *ejemplo*.
- Copie el código ejemplo.vhd generado desde la consola.

- Modifique el código según se indica a continuación:
  1. Elimine los caracteres escritos después de STD_LOGIC

```
entity ejemplo is
  port (
    A : in STD_LOGIC := 'X';
    B : in STD_LOGIC := 'X';
    F : out STD_LOGIC
  );
end ejemplo;
```

2. Instancie los componentes después de crear la arquitectura y antes del begin, tenga en cuenta que los componentes deben ser consecuentes con las entradas y salidas del layout de sus celdas estándar realizadas en Electric.

```
architecture STRUCTURE of ejemplo is

component IBUF port(en: in BIT; salida: out BIT);
    end component;
component OBUF port(en: in BIT; salida: out BIT);
    end component;
component OR2 port(a, b: in BIT; salida: out BIT);
    end component;

  signal A_IBUF_1 : STD_LOGIC;
  signal B_IBUF_3 : STD_LOGIC;
  signal F_OBUF_5 : STD_LOGIC;


begin
```

- Nota: Si tiene vectores debe instanciar cada posición como una entrada o señal independiente.

- Ejemplo

    b : in STD_LOGIC_VECTOR (2 downto 0)

Se debe convertir a:

    b0: in STD_LOGIC_VECTOR

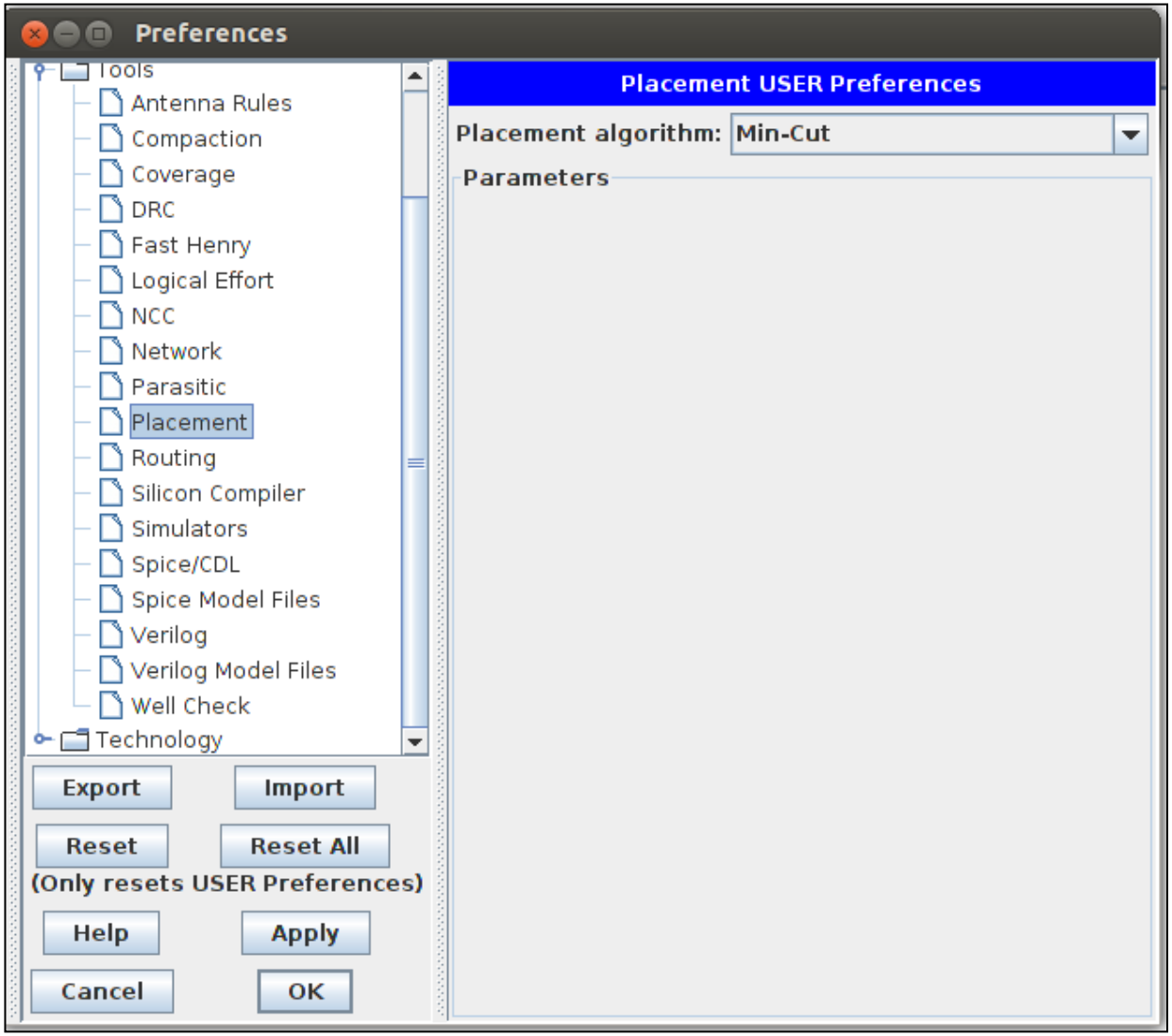
    b1: in STD_LOGIC_VECTOR

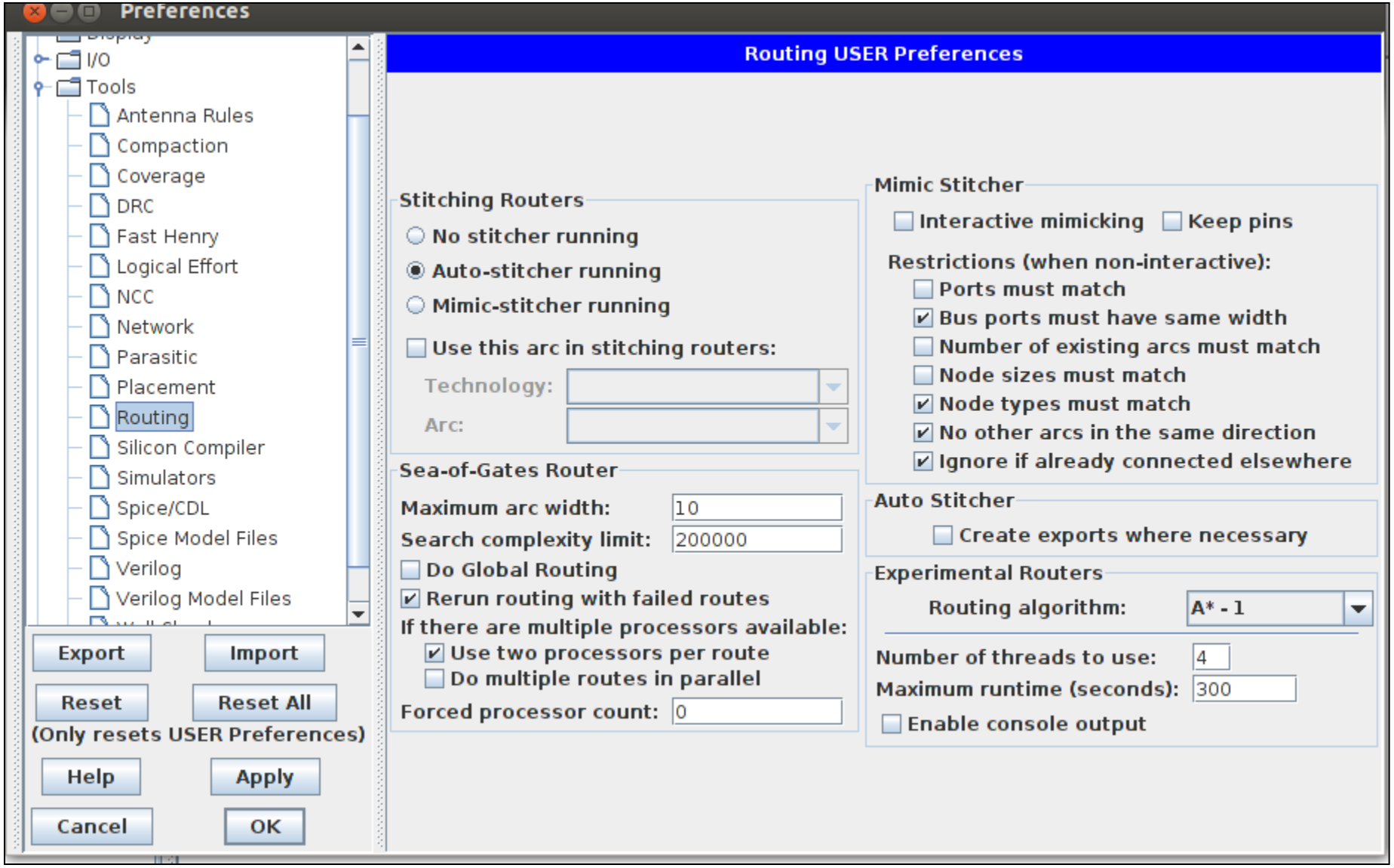
    b2: in STD_LOGIC_VECTOR

- Finalmente el código VHDL estructural que se debe implementar en Electric es:

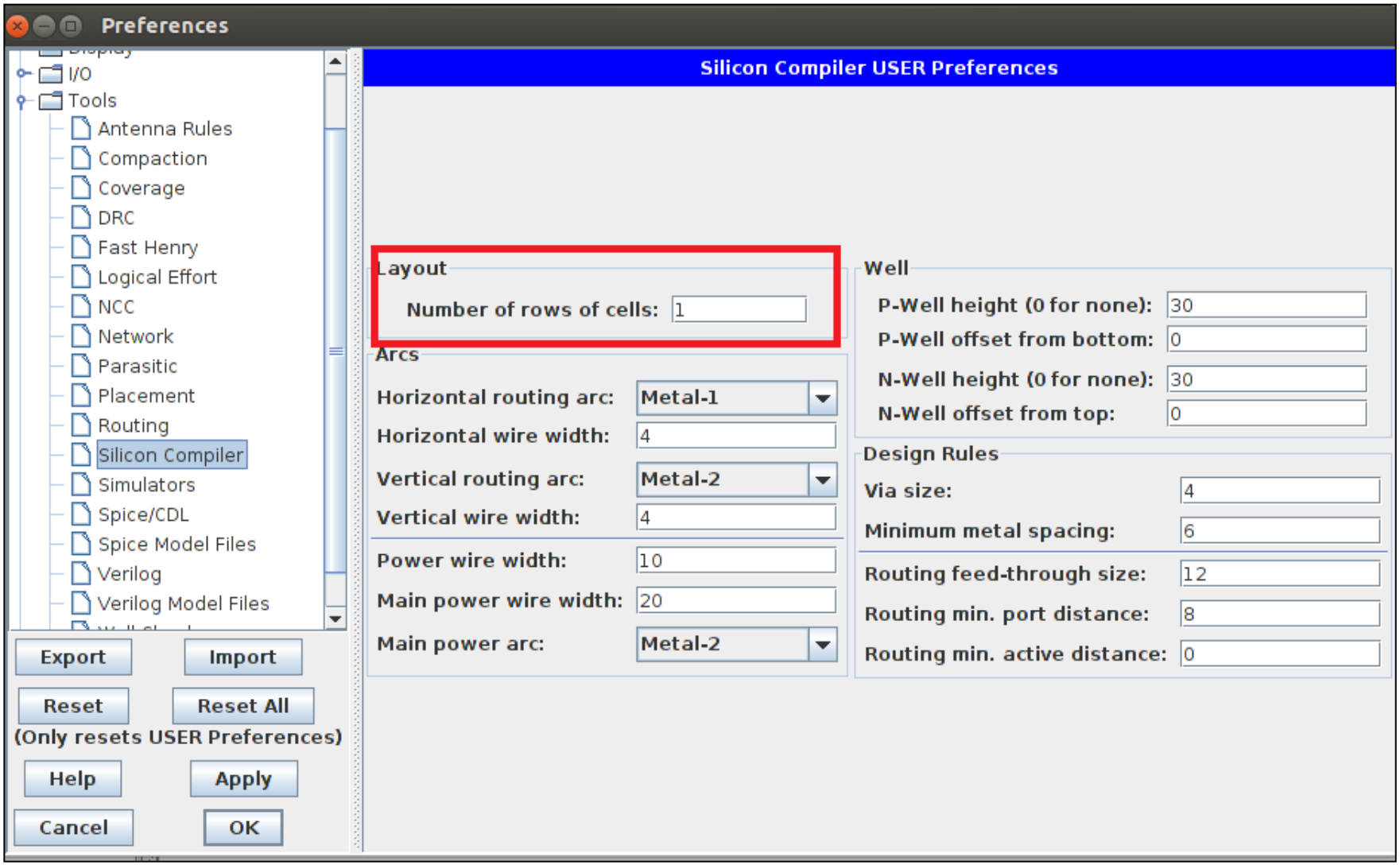
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
use UNISIM.VPKG.ALL;

entity ejemplo is
  port (

    A : in STD_LOGIC;
    B : in STD_LOGIC;
    F : out STD_LOGIC
  );
end ejemplo;


architecture STRUCTURE of ejemplo is


component IBUF port(en: in BIT; salida: out BIT);
    end component;
component OBUF port(en: in BIT; salida: out BIT);
    end component;
component OR2 port(a, b: in BIT; salida: out BIT);
    end component;

  signal A_IBUF_1 : STD_LOGIC;
  signal B_IBUF_3 : STD_LOGIC;
  signal F_OBUF_5 : STD_LOGIC;

begin

  F11 : OR2
    port map (
      I0 => A_IBUF_1,
      I1 => B_IBUF_3,
      O => F_OBUF_5
    );
  A_IBUF : IBUF
    port map (
      I => A,
      O => A_IBUF_1
    );
  B_IBUF : IBUF
    port map (
      I => B,
      O => B_IBUF_3
    );
  F_OBUF : OBUF
    port map (
      I => F_OBUF_5,
      O => F
    );

end STRUCTURE;
```

- Asegúrese de configurar la herramienta como se indica a continuación en **file → preferences →Tools.**




Preferences
I/O
Tools
Antenna Rules
Compaction
Coverage
DRC
Fast Henry
Logical Effort
NCC
Network
Parasitic
Placement
Routing
Silicon Compiler
Simulators
Spice/CDL
Spice Model Files
Verilog
Verilog Model Files
Export
Import
Reset
Reset All
(Only resets USER Preferences)
Help
Apply
Cancel
OK
Routing USER Preferences
Stitching Routers
No stitcher running
Auto-stitcher running
Mimic-stitcher running
Use this arc in stitching routers:
Technology:
Arc:
Sea-of-Gates Router
Maximum arc width: 10
Search complexity limit: 200000
Do Global Routing
Rerun routing with failed routes
If there are multiple processors available:
Use two processors per route
Do multiple routes in parallel
Forced processor count: 0
Mimic Stitcher
Interactive mimicking
Keep pins
Restrictions (when non-interactive):
Ports must match
Bus ports must have same width
Number of existing arcs must match
Node sizes must match
Node types must match
No other arcs in the same direction
Ignore if already connected elsewhere
Auto Stitcher
Create exports where necessary
Experimental Routers
Routing algorithm: A* - 1
Number of threads to use: 4
Maximum runtime (seconds): 300
Enable console output

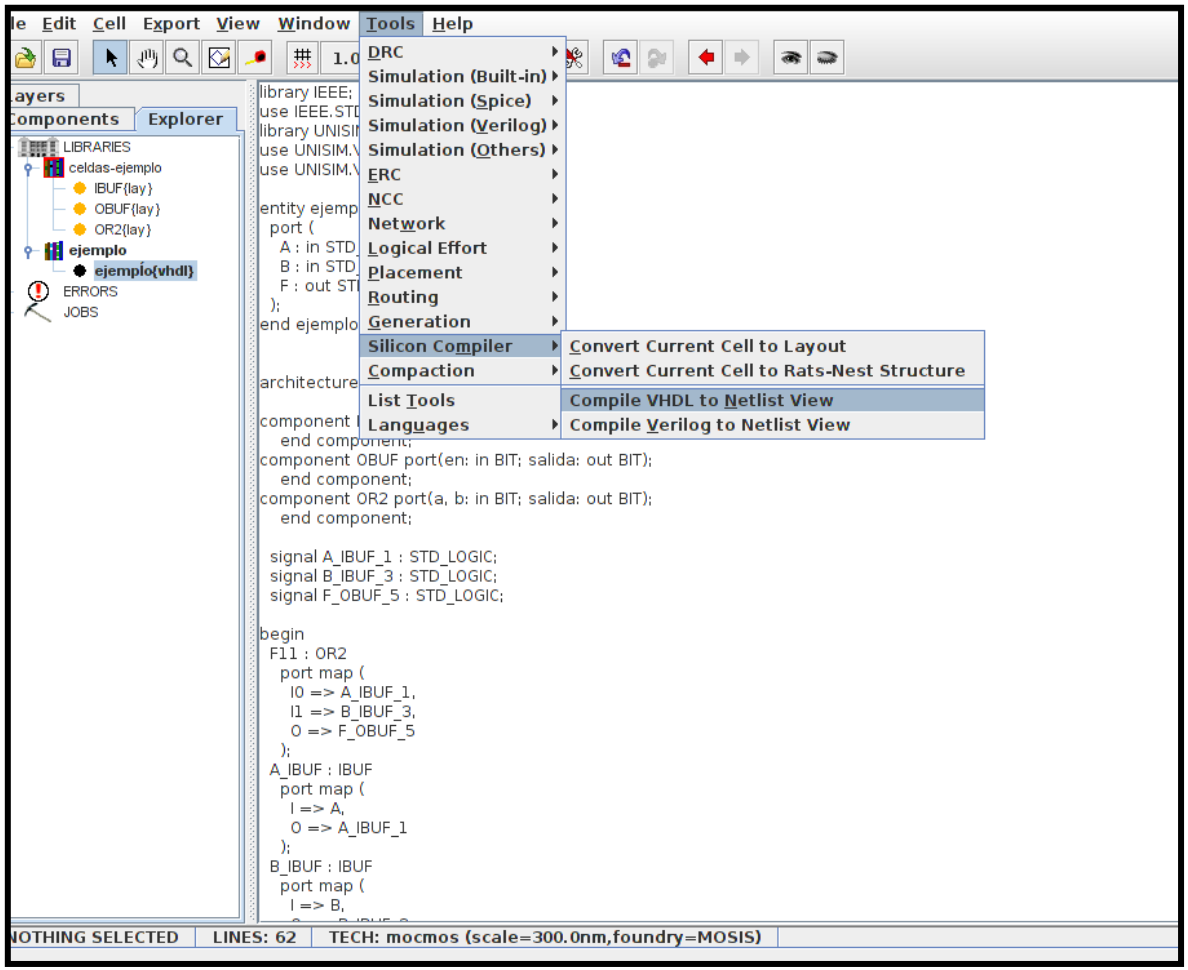
- Use una fila para cada módulo y 4 o más filas para el proyecto final, modificar donde señala el cuadro rojo.



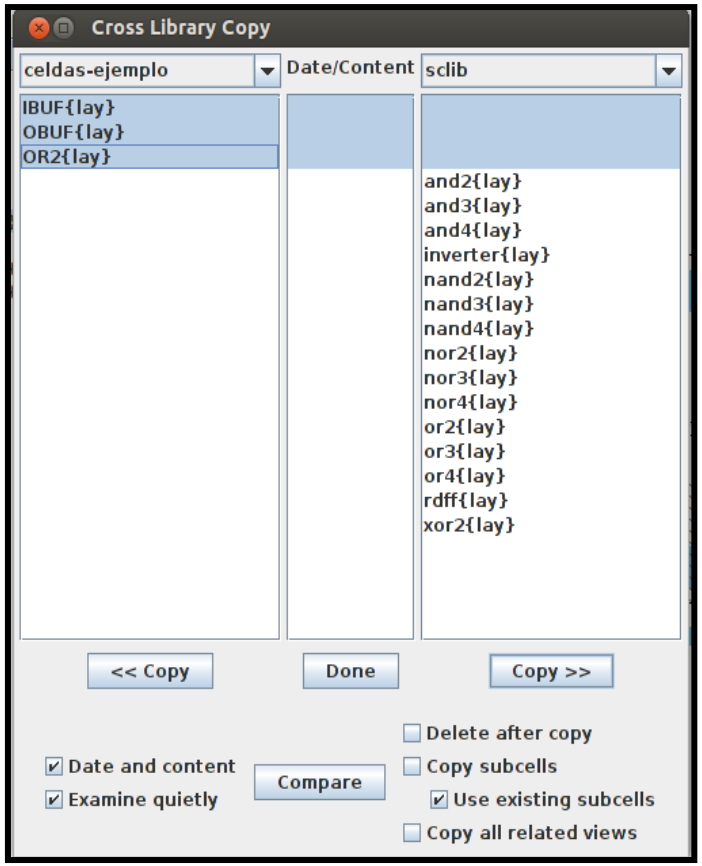
- Genere la netlist para esto vaya a:

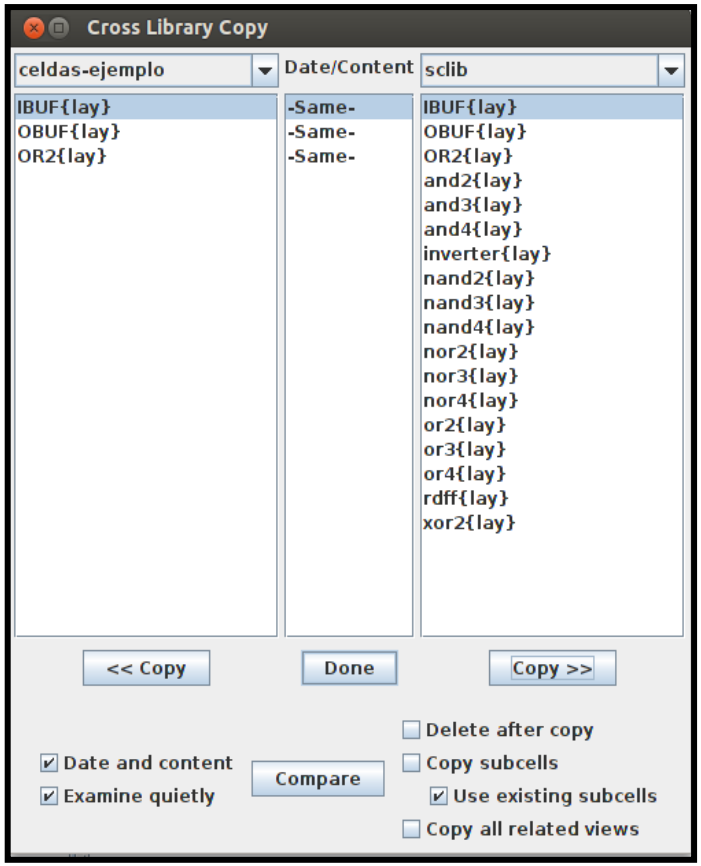
  Tools → Silicon Compiler → Compile VHDL to netlist View

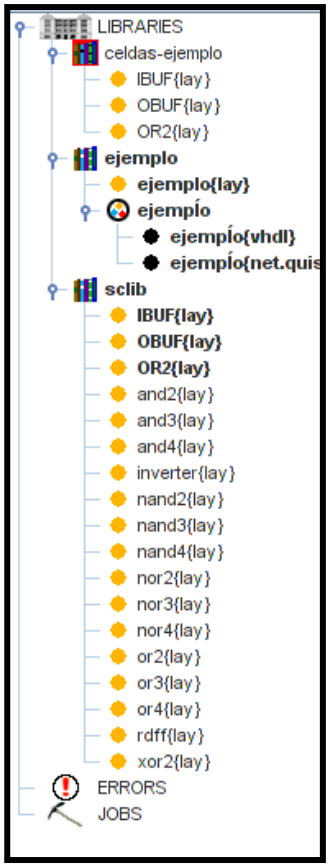
- Si no se genera la netlist es porque existe un error en el código.



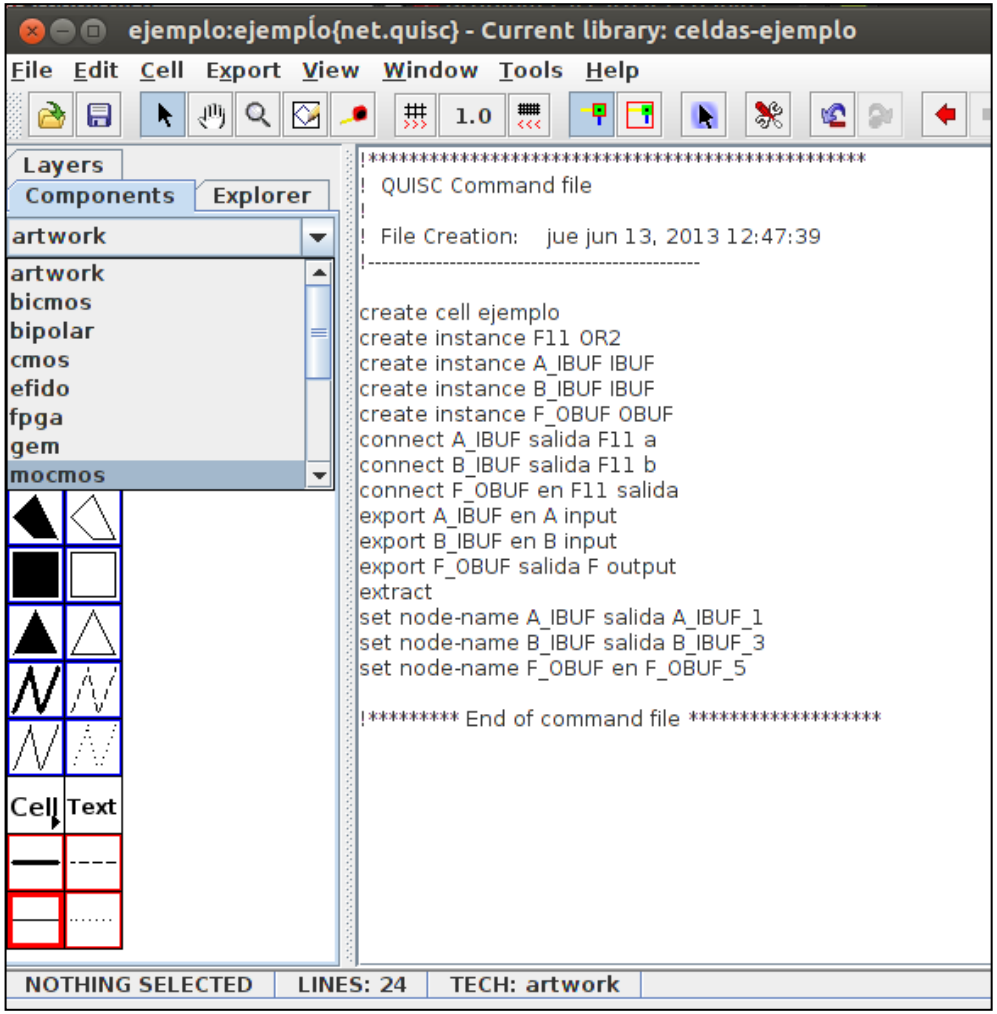
- Para que sus celdas sean tomadas debe copiarlas a la librería de sclib vaya a: **cell → Cross Library Copy**
- Copie los datos y contenido de todas las celdas de su proyecto**.**





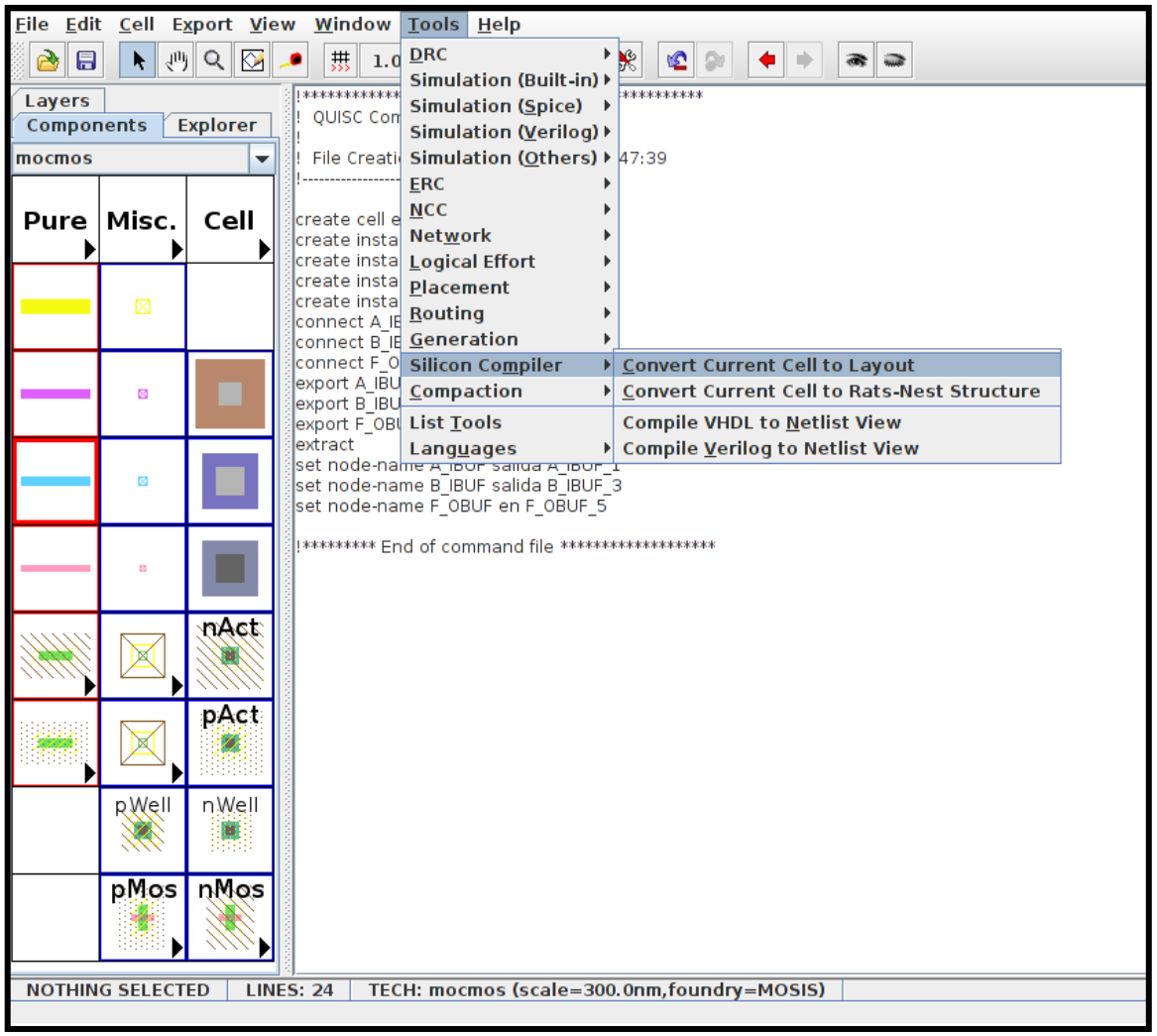


- Para generar el layout debe revisar que la netlist se ejecute con la tecnología mocmos.
- En components cambie artwork a mocmos.

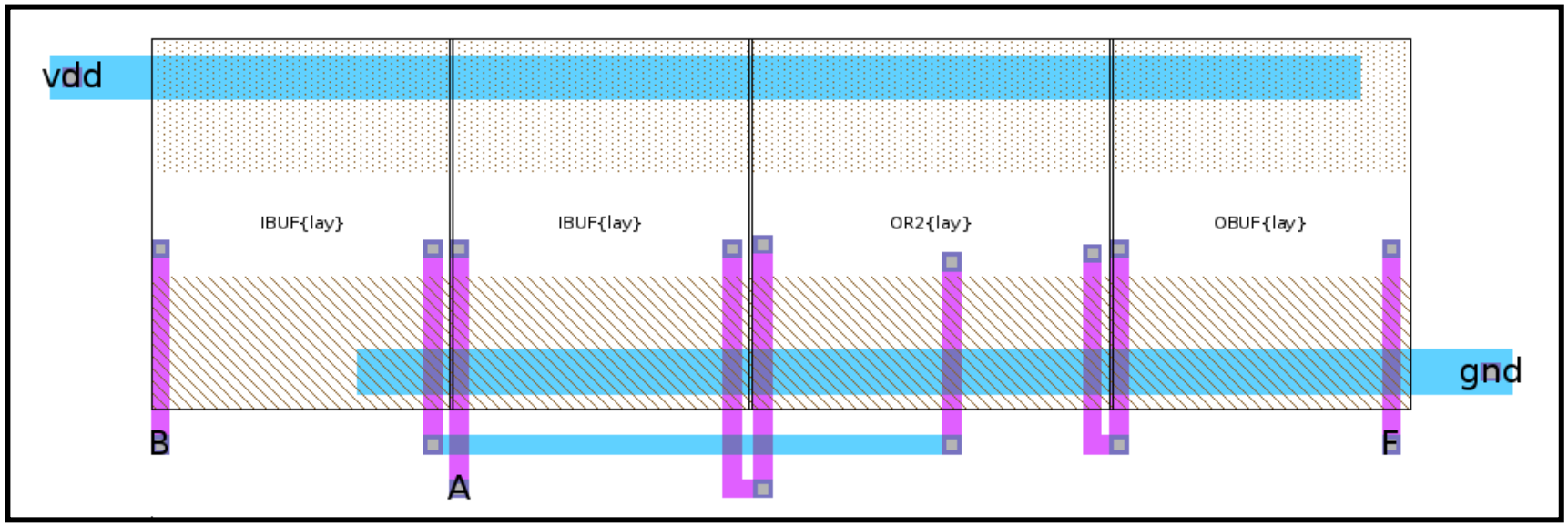


- Genere el layout para esto seleccione la netlist y vaya a:

  Tools → Silicon Compiler → Convert Current Cell to Layout



- Debe aparecer el layout.



- Con el icono en forma de *ojo* se pueden observar las celdas estándar implementadas.