

Cómo importar/exportar un DCP en Vivado

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2025/01/21/como-importar-exportar-un-dcp-en-vivado/>

Blog: <https://soceame.wordpress.com/>

GitHub: <https://github.com/DRubioG>

Fecha última modificación: 13/02/2025

<https://soceame.wordpress.com/2025/01/21/como-importar-exportar-un-dcp-en-vivado/>

Parte de la explicación de esta entrada viene de esta entrada anterior, pero lo recupero.

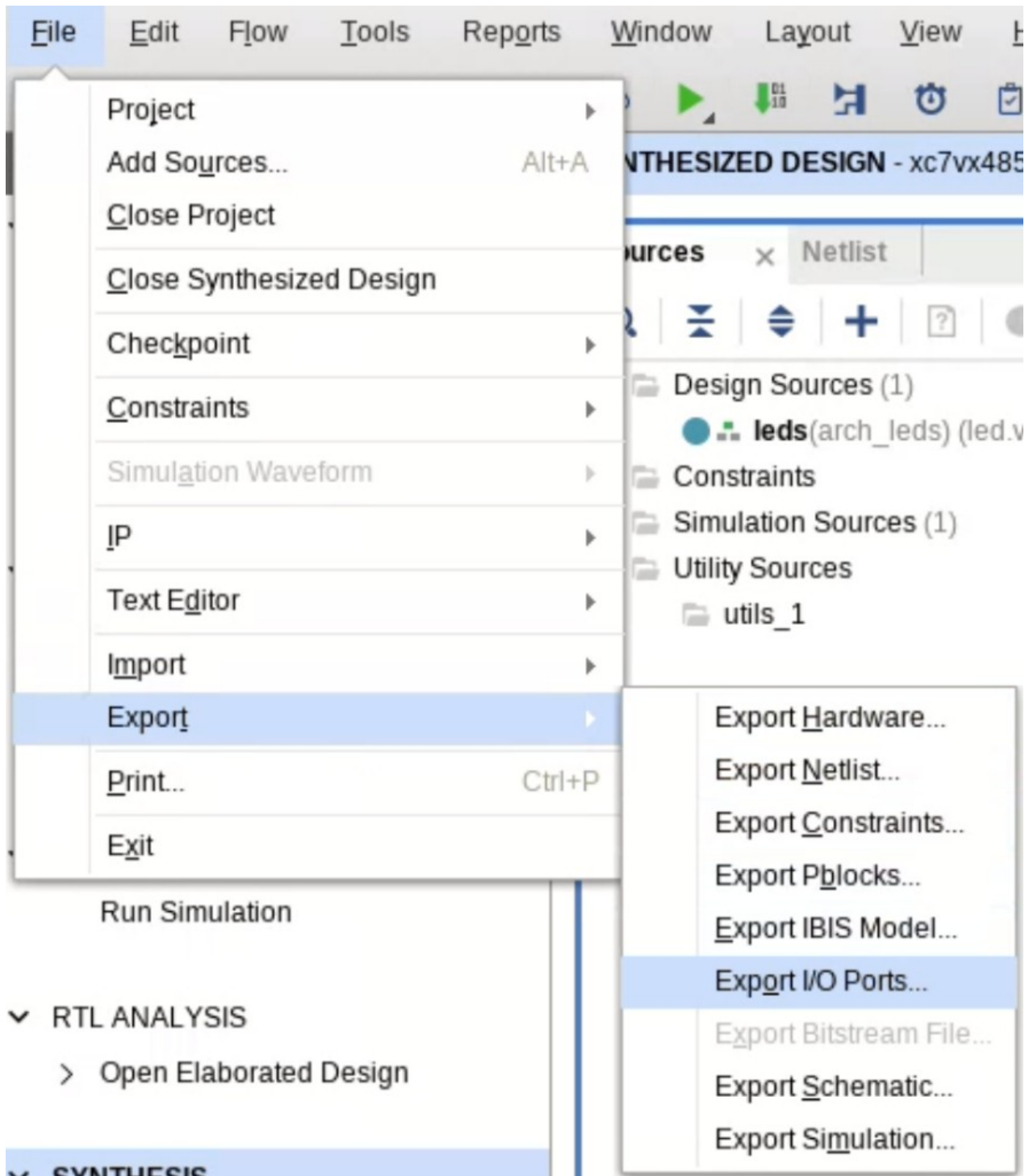
<https://soceame.wordpress.com/2024/05/26/como-compartir-un-proyecto-firmware-sin-compartir-el-codigo-fuente-parte-2/>

Exportar DCP

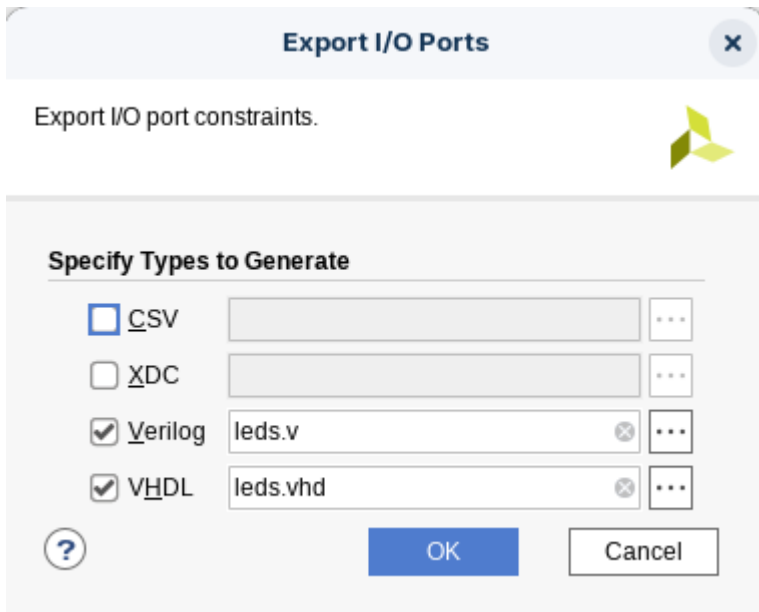
Para exportar lo único que necesitas es sintetizar un diseño con Vivado. Esto genera ya el fichero DCP en la carpeta *synth_1* de nuestro proyecto.



Ahora lo siguiente que necesitamos es una máscara para nuestros puertos, o lo que es lo mismo un fichero en VHDL o Verilog que tenga los puertos que tiene el DCP. Para ello en File > Export > Export I/O Ports.



Esto nos abre una pestaña que nos permite elegir si queremos un fichero en VHDL o Verilog.



NOTA: es posible que la máscara de puertos en Verilog no funcione, pero solo con la máscara de VHDL se puede trabajar.

La máscara es muy básica.

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

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

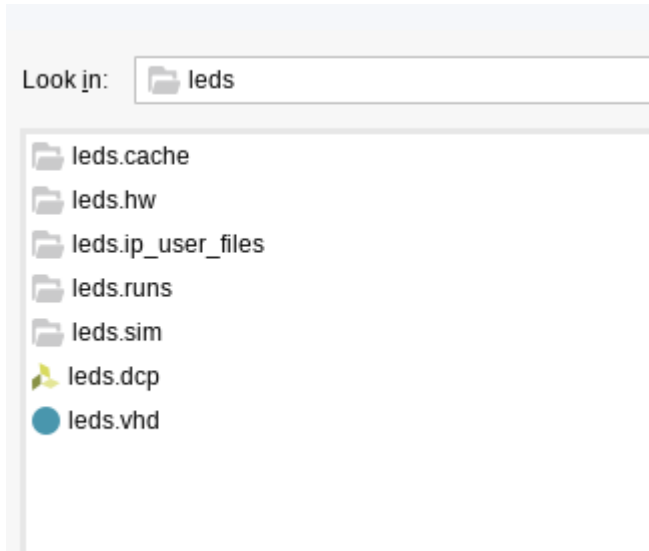
entity leds is
  Port (
    led : out STD_LOGIC_VECTOR ( 2 downto 0 );
    clk : in STD_LOGIC;
    rst_n : in STD_LOGIC
  );
end leds;

architecture Behavioral of leds is
begin

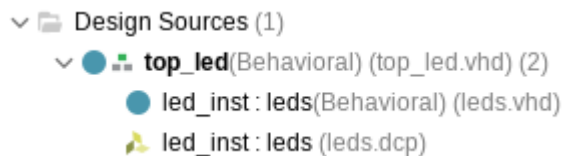
end Behavioral;
```

Importar DCP

Para importar un DCP tenemos que irnos al proyecto en el que queremos importar el DCP y decirle que añada el DCP con la máscara que ha creado.



Ahora Vivado nos pone al mismo nivel el DCP con la máscara.



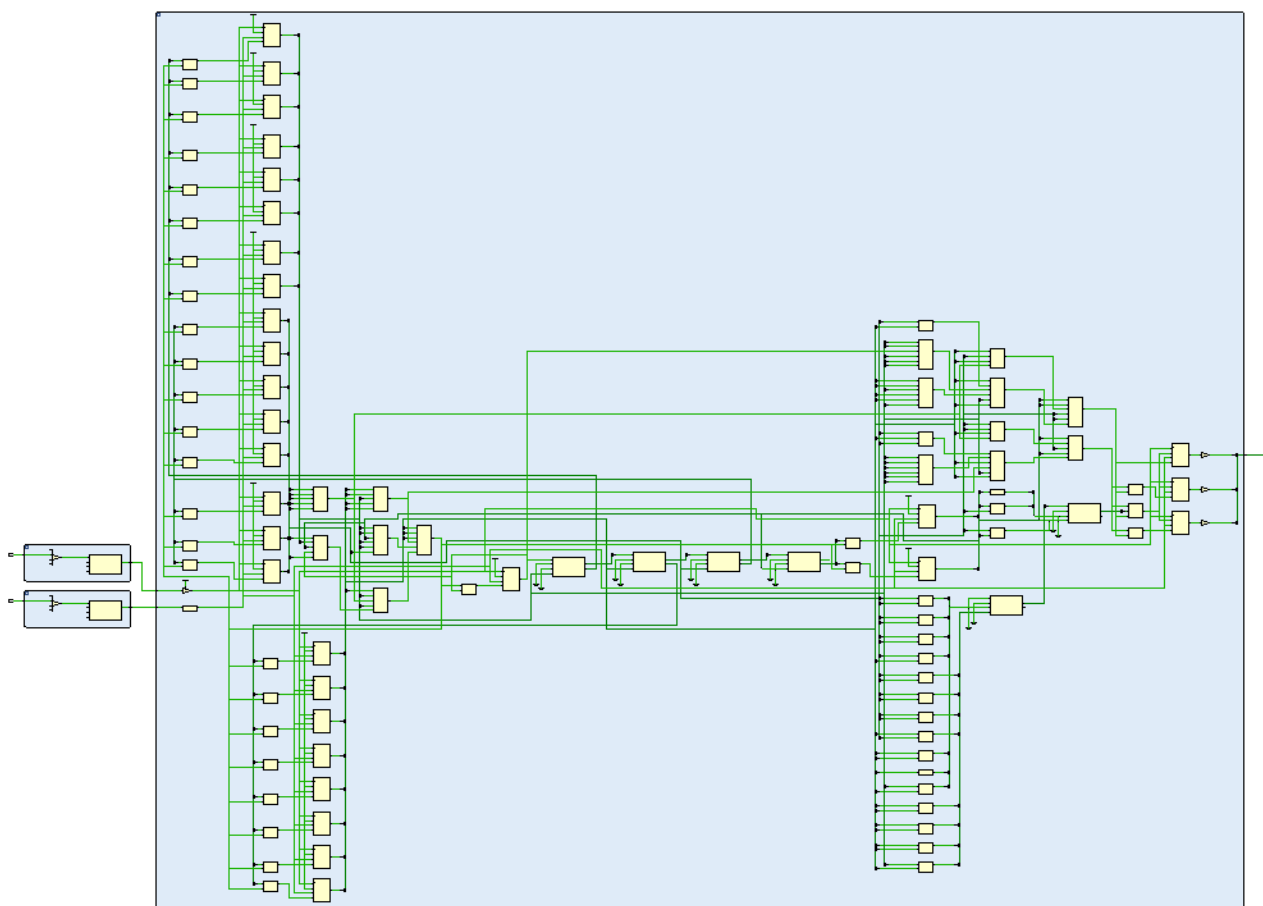
Para instanciar el bloque se hace utilizando la máscara como referencia.

```
component leds is
  Port (
    clk : in std_logic;
    rst_n : in std_logic;
    led : out std_logic_vector(2 downto 0)
  );
end component;

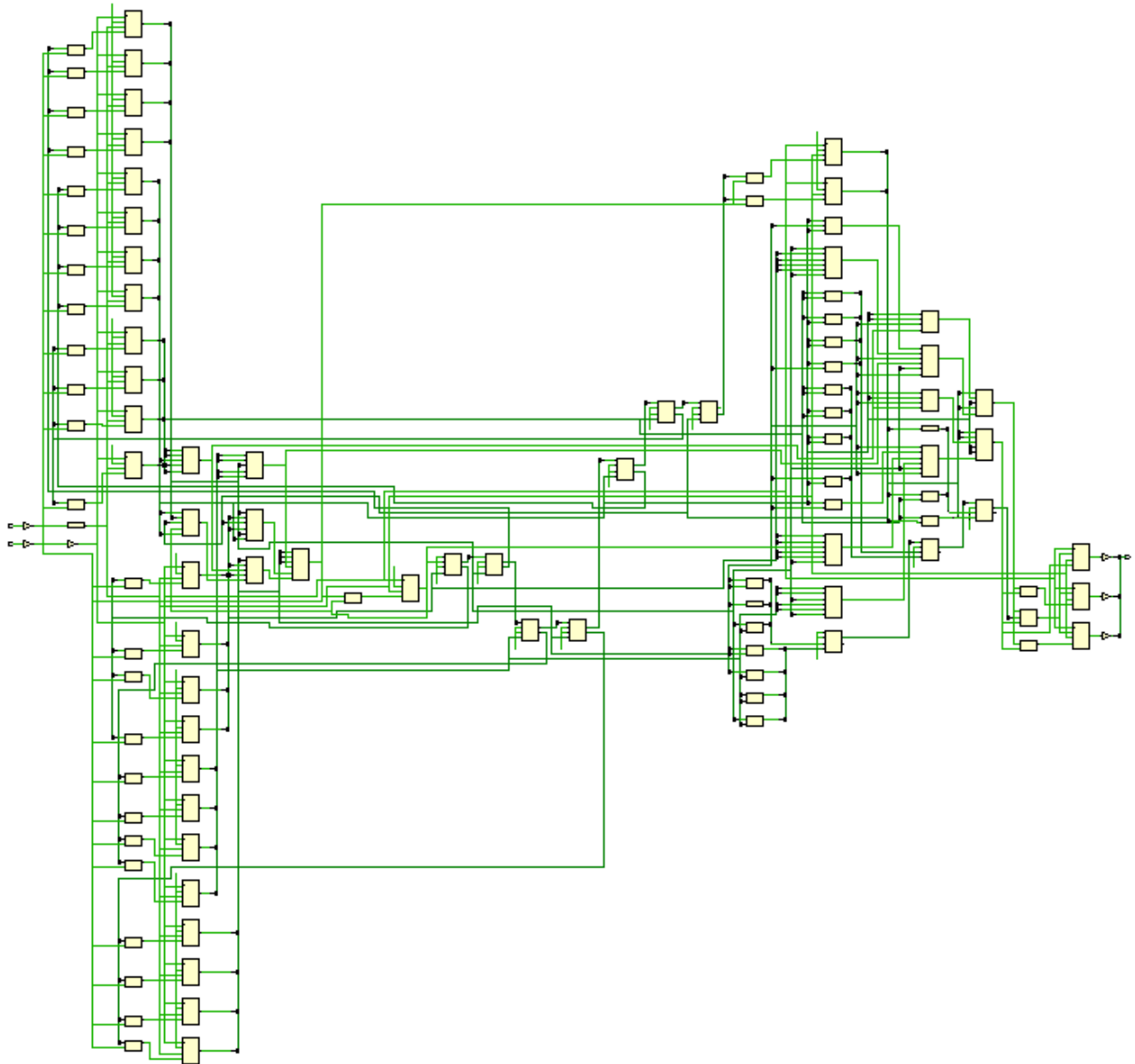
begin

led_inst : leds
  Port map (
    clk => clk,
    rst_n => rst_n,
    led => led_1
  );
```

Entonces, si miramos el modelo sintetizado.



Y lo comparamos con el modelo sintetizado del que se ha generado el DCP.



Podemos ver que son iguales, con cambios en las ubicaciones de algunas puertas, pero con las mismas puertas.

Entonces, el modelo se ha transferido perfectamente mediante un DCP.

Nota final

La máscara no es necesaria para transferir el DCP, solo que si el DCP no se instancia exactamente con los puertos que tiene dará problemas al sintetizarlo.

▼ Design Sources (1)
▼ top_led(Behavioral) (top_led.vhd) (1)
 led_inst : leds (leds.dcp)

<https://soceame.wordpress.com/2025/01/21/como-importar-exportar-un-dcp-en-vivado/>

Entonces, es mejor tener algo, como una máscara de puertos, que permita conocer cuáles son los puertos para instanciarlos correctamente.