Cómo configurar un pin de FPGA en modo pull-up-pull-down con Vivado

Creador: David Rubio G.

Entrada: https://soceame.wordpress.com/2024/10/28/como-configurar-un-pin-de-fpga-en-modo-pull-up-pull-down-con-vivado/

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

GitHub: https://github.com/DRubioG

Fecha última modificación: 23/02/2025

Si has llegado aquí, ya sabrás cuál es el funcionamiento de una resistencia de *pull-up* y/o de *pull-down*. Bien pues ahora te voy a comentar como se consigue configurar los pines de una FPGA para que estén por defecto en *pull-up* o en *pull-down*, y así puedas ahorrarte colocar resistencias.

Configuración

Para configurar un pin en modo *pull-up/pull-down*, hay dos métodos:

- ***Mediante restricciones (constraints)*** en el XDC**.** Para ello se tiene que utilizar la restricción de ***PULLTYPE*** como *PULLUP* o *PULLDOWN*.

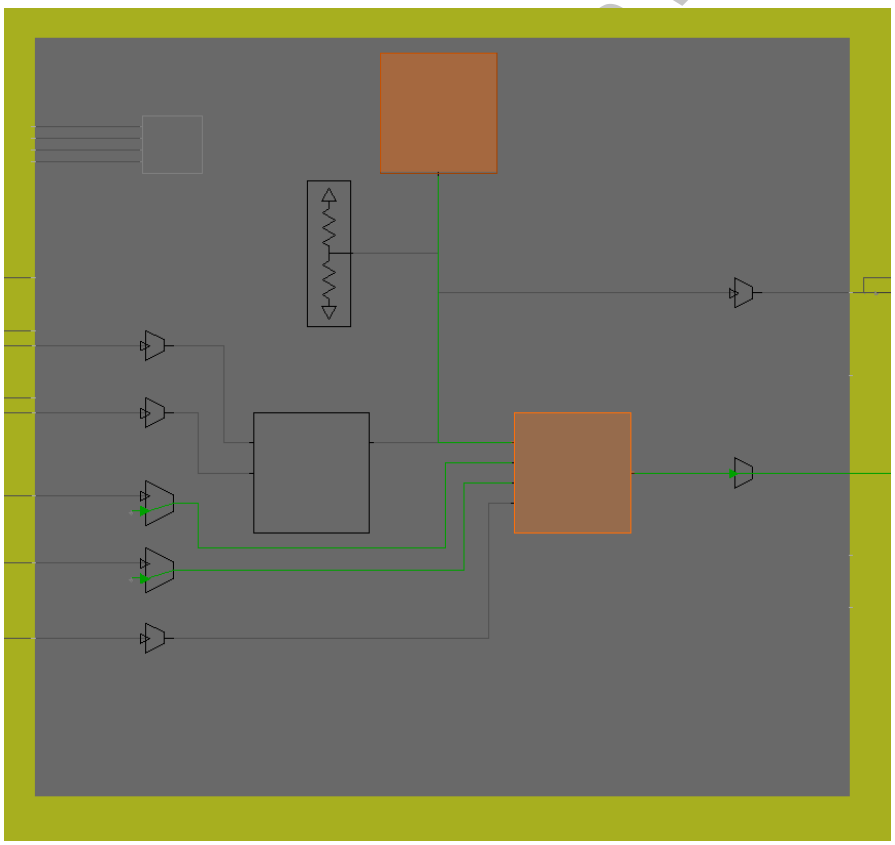
```
set_property PULLTYPE <PULLUP o PULLDOWN> [get_ports puerto];
```

- O mediante ***atributos en el código***. Para ello se utiliza el atributo PULLTYPE.

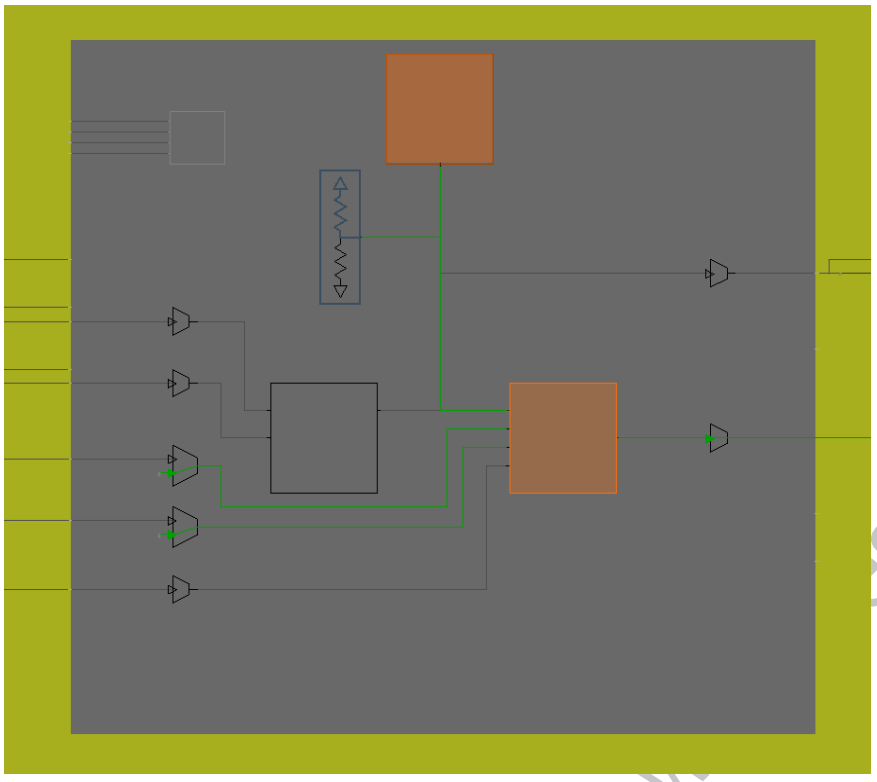
```
attribute PULLTYPE: string;
attribute PULLTYPE of puerto : signal is "<PULLUP o PULLDOWN>";
```

NOTA: si se quiere hacer colisionar una restricción en el XDC con un atributo en el código, **gana la restricción.**

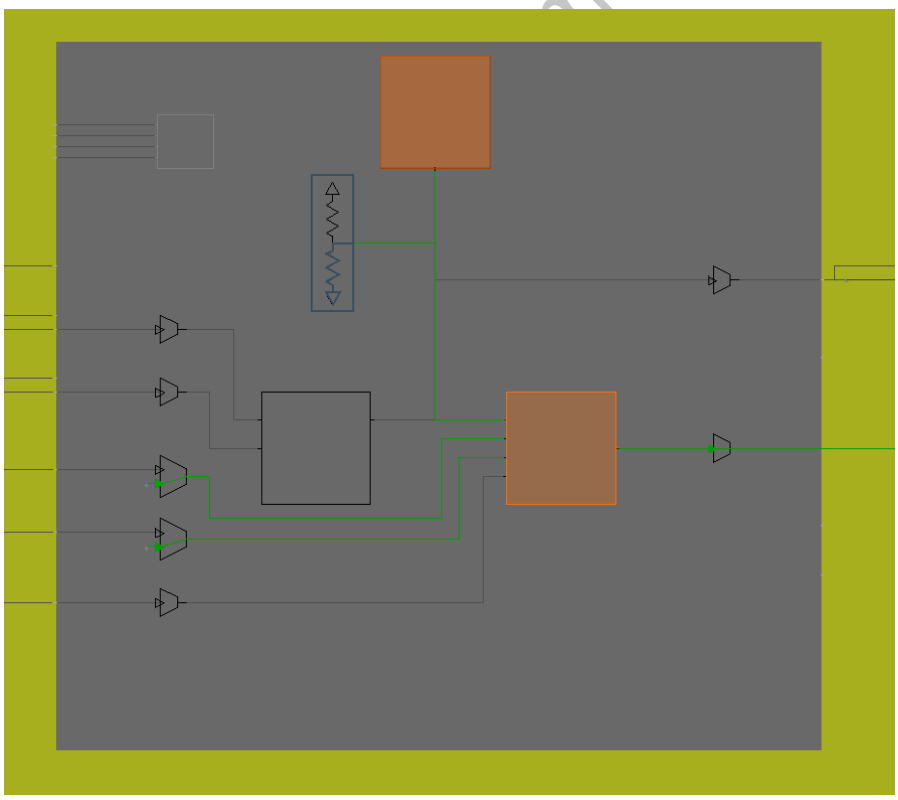
Si el pin no se configura como *pull-up* o *pull-down*, queda de la siguiente forma. Donde se puede ver que hay dos resistencias desactivadas.



Si se configura como **PULL-UP,** Vivado solo configura la resistencia superior. Se puede ver que la marca en azul.



Y si se configura como **PULL-DOWN**, configura la resistencia en azul.



NOTA: solo por curiosidad, el bloque cuadrado que no marca Vivado es el buffer de salida, el que marca al lado es el de salida, y el de arriba es el pin.

Ejemplo

Hacemos la demostración con un pequeño ejemplo.

Imaginemos que queremos leer el valor de varios pines de entrada con un ILA, pero los queremos configurados la mitad en pull-up y la otra mitad en pull-down *(al estar en los pines en vacío o en el aire, se puede ver mejor su funcionamiento)*.

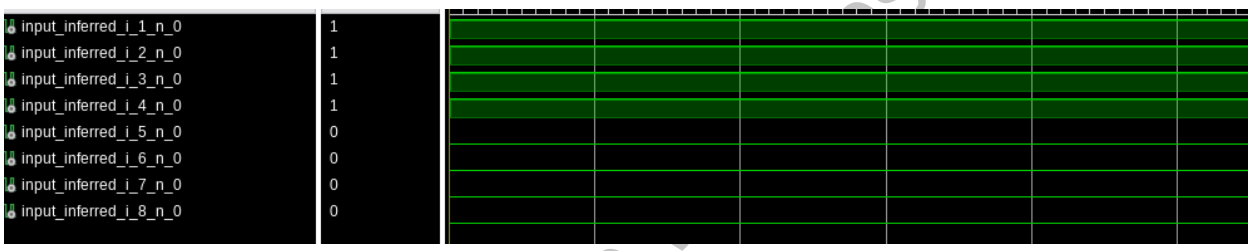
Para ello con este pequeño código. *(comentada está la otra opción para configurar un pull-up)*

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

entity test is
    Port (
        clk : in std_logic;
        input : in std_logic_vector(7 downto 0)
    );
end test;

architecture Behavioral of test is


COMPONENT ila_0

PORT (
    clk : IN STD_LOGIC;

    probe0 : IN STD_LOGIC_VECTOR(7 DOWNTO 0)
);
END COMPONENT  ;


--attribute PULLTYPE: string;
--attribute PULLTYPE of input : signal is "PULLUP";

begin
  ila : ila_0
PORT MAP (
    clk => clk,

    probe0 => input
);

end Behavioral;
```

Después configuramos el siguiente XDC, donde la mitad está en *pull-up* y la otra mitad en *pull-down*.

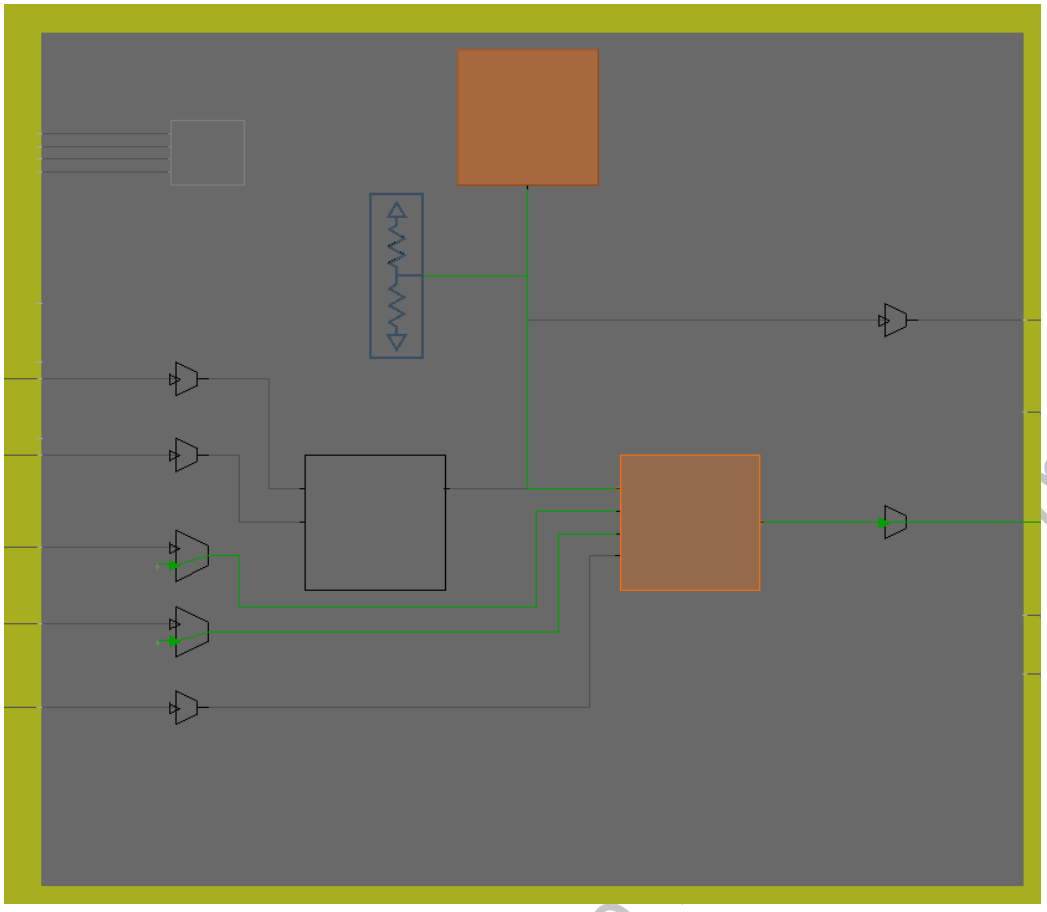
```
set_property -dict { PACKAGE_PIN D5 IOSTANDARD LVCMOS33 } [get_ports {input[0]}];
set_property -dict { PACKAGE_PIN G5 IOSTANDARD LVCMOS33 } [get_ports {input[1]}];
set_property -dict { PACKAGE_PIN G7 IOSTANDARD LVCMOS33 } [get_ports {input[2]}];
set_property -dict { PACKAGE_PIN G8 IOSTANDARD LVCMOS33 } [get_ports {input[3]}];
set_property -dict { PACKAGE_PIN E5 IOSTANDARD LVCMOS33 } [get_ports {input[4]}];
set_property -dict { PACKAGE_PIN E6 IOSTANDARD LVCMOS33 } [get_ports {input[5]}];
set_property -dict { PACKAGE_PIN D6 IOSTANDARD LVCMOS33 } [get_ports {input[6]}];
set_property -dict { PACKAGE_PIN G6 IOSTANDARD LVCMOS33 } [get_ports {input[7]}];

set_property PULLTYPE PULLDOWN [get_ports input[0]];
set_property PULLTYPE PULLDOWN [get_ports input[1]];
set_property PULLTYPE PULLDOWN [get_ports input[2]];
set_property PULLTYPE PULLDOWN [get_ports input[3]];
set_property PULLTYPE PULLUP [get_ports input[4]];
set_property PULLTYPE PULLUP [get_ports input[5]];
set_property PULLTYPE PULLUP [get_ports input[6]];
set_property PULLTYPE PULLUP [get_ports input[7]];
```

Si ahora generamos el bitstream, lo grabamos en la FPGA y leemos el valor que lee el ILA. Se puede ver que la mitad de los pines el ILA los lee a nivel alto (*los que están en pull-up*) y la otra mitad a nivel bajo (*los que están en pull-down*)

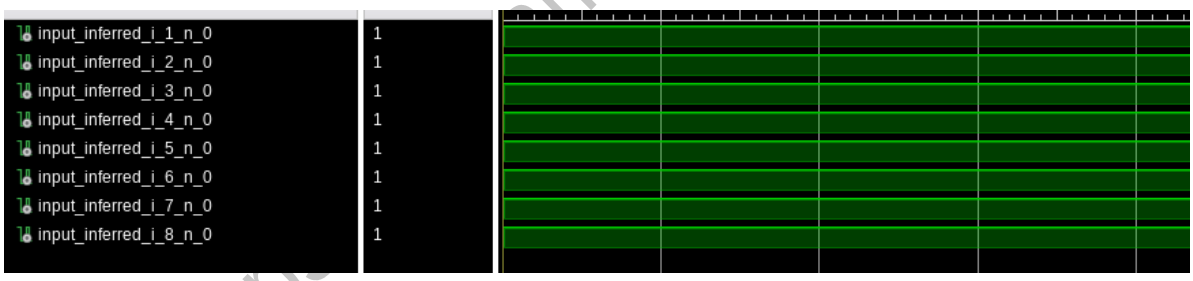


NOTA Final

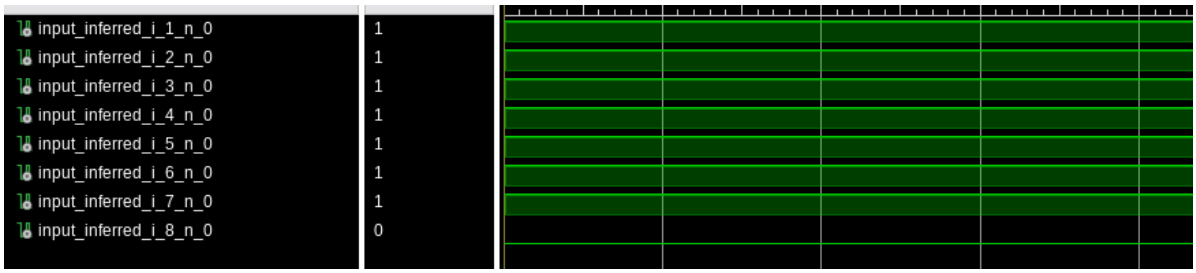
Existe un tercer estado que es el estado KEEPER («*mantenedor*«), que permite dejar configurado el pin en el estado en estado anterior. Para ello Vivado lo que hace es activar las dos resistencias a la vez.



Lo que ocurre con este estado es que para empezar a funcionar la línea está a nivel alto.

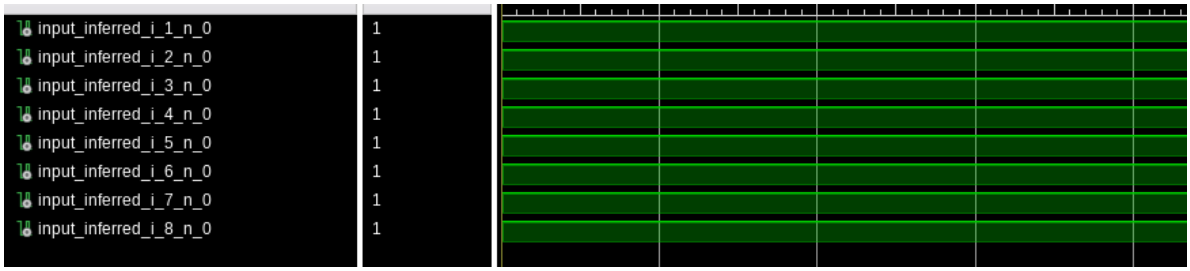


Si una de las líneas se pone a nivel bajo, la línea cambia a nivel bajo y se queda a nivel bajo, no vuelve a nivel alto como si fuera un *pull-up*.



Y si esa misma línea se vuelve a poner a nivel alto, la línea se mantiene a nivel alto, aunque quites el nivel alto de la entrada. No vuelve a nivel bajo como si fuera un *pull-down*.

https://soceame.wordpress.com/2024/10/28/como-configurar-un-pin-de-fpga-en-modo-pull-up-pull-down-con-vivado/



Para configurar se hace como un *pull-up* o un *pull-down*, pero utilizando el atributo **KEEPER**.

```
attribute PULLTYPE: string;
attribute PULLTYPE of input : signal is "KEEPER";

set_property -dict { PACKAGE_PIN D5 IOSTANDARD LVCMOS33 } [get_ports {input[0]}];
set_property -dict { PACKAGE_PIN G5 IOSTANDARD LVCMOS33 } [get_ports {input[1]}];
set_property -dict { PACKAGE_PIN G7 IOSTANDARD LVCMOS33 } [get_ports {input[2]}];
set_property -dict { PACKAGE_PIN G8 IOSTANDARD LVCMOS33 } [get_ports {input[3]}];
set_property -dict { PACKAGE_PIN E5 IOSTANDARD LVCMOS33 } [get_ports {input[4]}];
set_property -dict { PACKAGE_PIN E6 IOSTANDARD LVCMOS33 } [get_ports {input[5]}];
set_property -dict { PACKAGE_PIN D6 IOSTANDARD LVCMOS33 } [get_ports {input[6]}];
set_property -dict { PACKAGE_PIN G6 IOSTANDARD LVCMOS33 } [get_ports {input[7]}];

set_property PULLTYPE KEEPER [get_ports input[0]];
set_property PULLTYPE KEEPER [get_ports input[1]];
set_property PULLTYPE KEEPER [get_ports input[2]];
set_property PULLTYPE KEEPER [get_ports input[3]];
set_property PULLTYPE KEEPER [get_ports input[4]];
set_property PULLTYPE KEEPER [get_ports input[5]];
set_property PULLTYPE KEEPER [get_ports input[6]];
set_property PULLTYPE KEEPER [get_ports input[7]];
```

Parte 2

https://soceame.wordpress.com/2025/02/01/como-configurar-un-pin-de-fpga-en-modo-pull-up-pull-down-con-vivado-part-2/