



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2343 — Arquitectura de Computadores 2019-2

Tutorial – Vivado – Crear Simulaciones

Tutorial para crear una simulación en Vivado

Introducción:

Un software como Vivado permite que convertir un circuito descrito en un lenguaje como VHDL sea convertido en conexiones reales ejecutadas por una placa. Para lograr esto, Vivado debe sintetizar, implementar y compilar lo que se describe, para que sea leído por una máquina, y todo el proceso puede tardar varios minutos (hasta 15 minutos).

Cuando tenemos una descripción sin ningún tipo de error, del cual estamos seguros que funcionará y cumplirá el objetivo deseado, pues esperar unos cuantos minutos no es un problema. Pero cuando se trata de ir construyendo una pieza de hardware, de la cuál no se tiene certeza de que pueda funcionar, podría ser un gran problema, ya que la probabilidad de error es alta, lo que se resume en compilar código descriptivo varias veces.

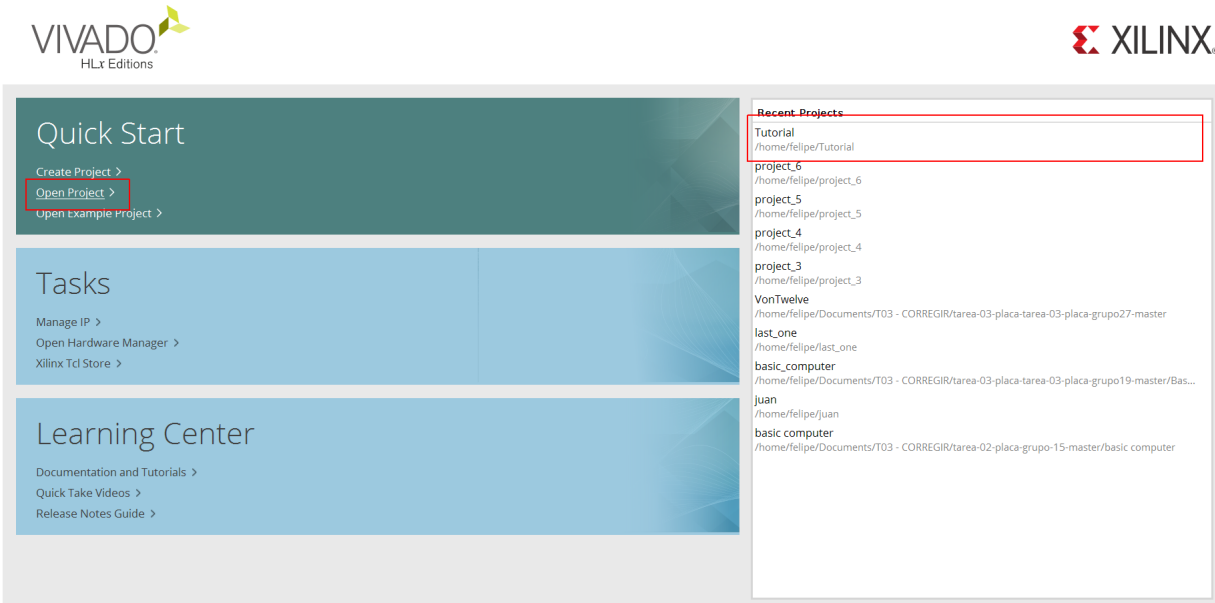
El proceso descrito anteriormente podría tardar horas. Es por ello que Vivado además de pasar lo escrito en VHDL a un lenguaje de máquina que pueda procesar una placa, también permite simular el comportamiento de un circuito, lo que es una herramienta bastante útil cuando se trata de la construcción de hardware.

En este tutorial se enseñará las nociones básicas de como simular código VHDL en Vivado. Para seguir este, se recuerda que **es necesario** haber completado los tres tutoriales pasados (disponibles en el *syllabus* del curso).

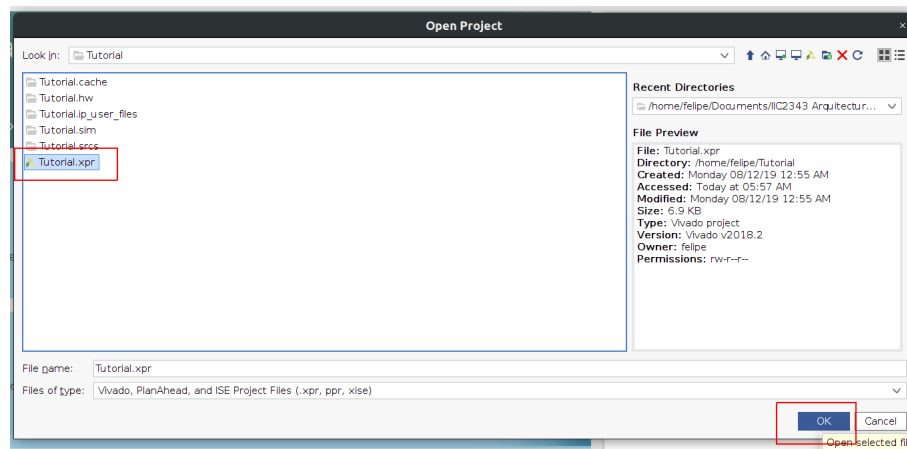
1 Preparar archivo a ser simulado

1.1 Abrir un proyecto:

Asumiendo que ya se tiene un proyecto creado (en caso de no ser así se puede crear uno siguiendo las instrucciones del tutorial de creación de proyecto) debemos abrir el proyecto para eso tenemos dos opciones:

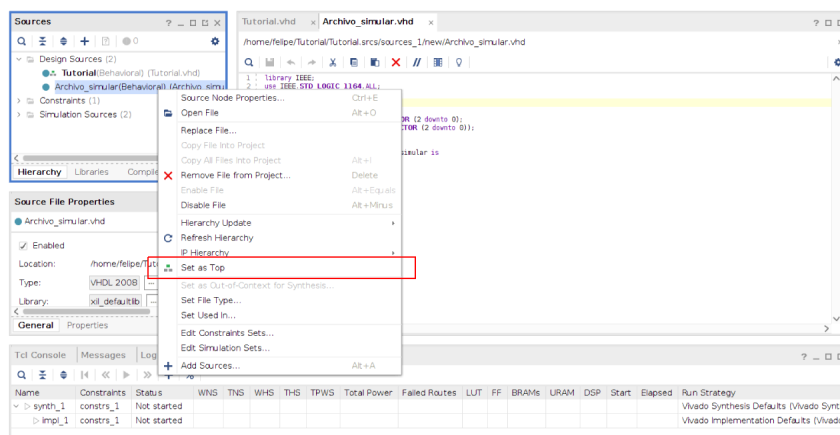


La más rápida es ir a la columna izquierda y apretar en *Recent Projects*, lo que nos abrirá inmediatamente nuestro proyecto. En caso de no tener esa opción se puede presionar *Open Project*. De ser así se debe buscar en su directorio la carpeta donde está el proyecto, y seleccionar el archivo de extensión **.xpr** como se muestra a continuación:



1.2 Seleccionar archivo a ser simulado:

Una vez dentro del proyecto seleccionaremos el archivo VHDL que queremos simular y daremos click derecho dentro de la ventana *Sources*. Se desplegará una serie de opciones, de la cuál daremos a la que dice *Set as Top* como se muestra en la imagen:

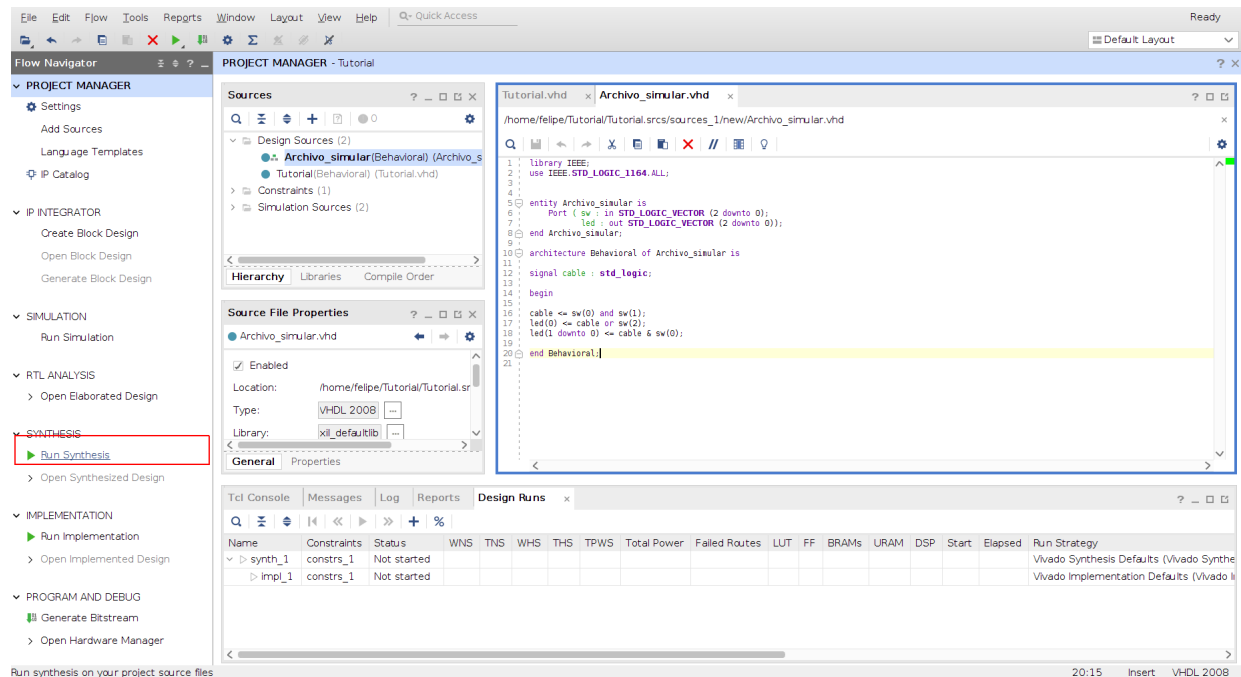


NOTA 1: Si el proyecto solo consta de un solo archivo, este paso se omite, pues por defecto Vivado ya habrá hecho esto.

NOTA 2: En caso de usar un multiplexor con el *statement with/select* es **necesario** que se termine siempre con **others;** para que Vivado pueda simularlo. Esto se aplica para el archivo a ser simulado como también a los todas las entidades que lo compongan. En caso de no usar selectores con este *statement* ignorar esta nota.

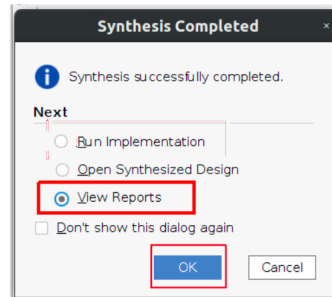
1.3 Correr la síntesis:

Una vez hecho lo anterior, debemos apretar ir a la columna lateral izquierda, y seleccionar la opción *Run Synthesis*, como se muestra en la figura:



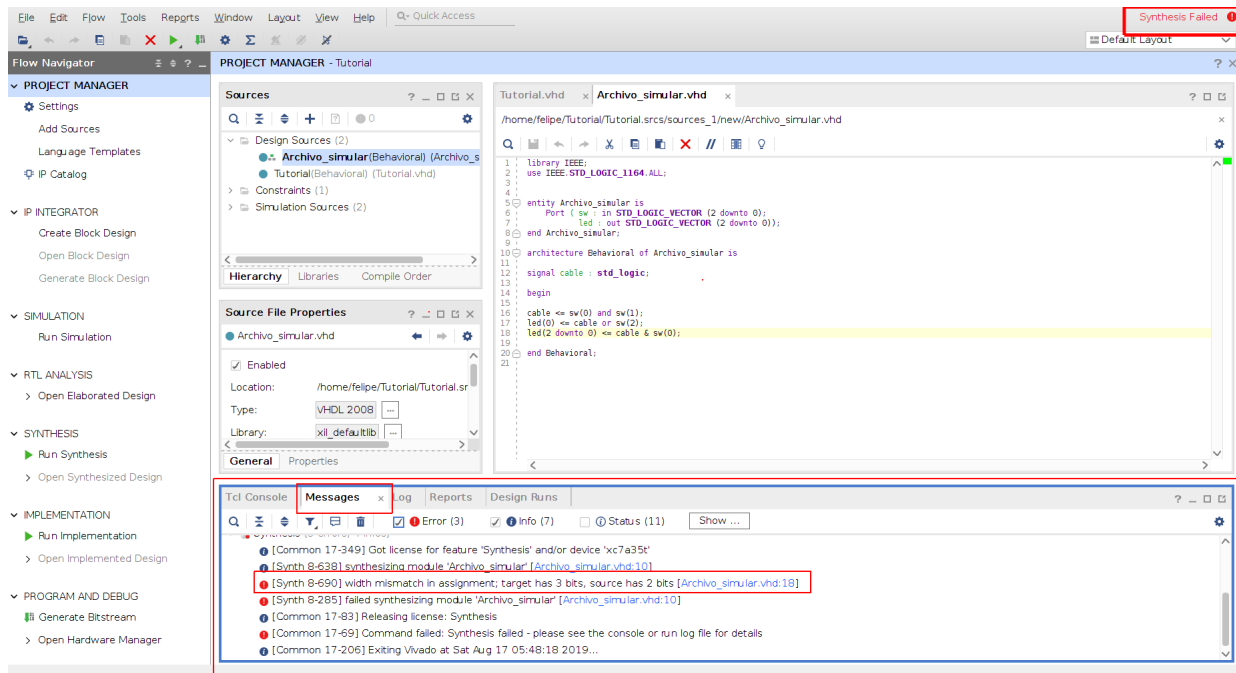
Esto activará un proceso de revisión sintaxis de VHDL (como por ejemplo no falte un ";") y una revisión semántica (como por ejemplo que no existan dos conexiones hacia un mismo puerto).

Una vez terminado dicho proceso exitosamente, debería aparecer una ventana como la siguiente:



Aquí solo se selecciona *View Reports* y luego se presiona en *OK*.

En caso de no haber una falla, se mostrará un mensaje similar al siguiente:

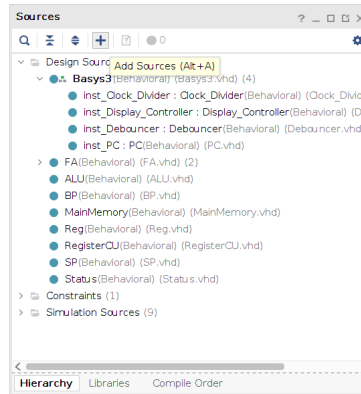


En el panel inferior en la ventana de *Messages* Vivado nos indicará cual es la línea que falla y la razón. En este caso se intento conectar una señal de 2 bits a otra de 3 bits. Una vez solucionado todos los errores, se repite este paso hasta lograr una síntesis exitosa.

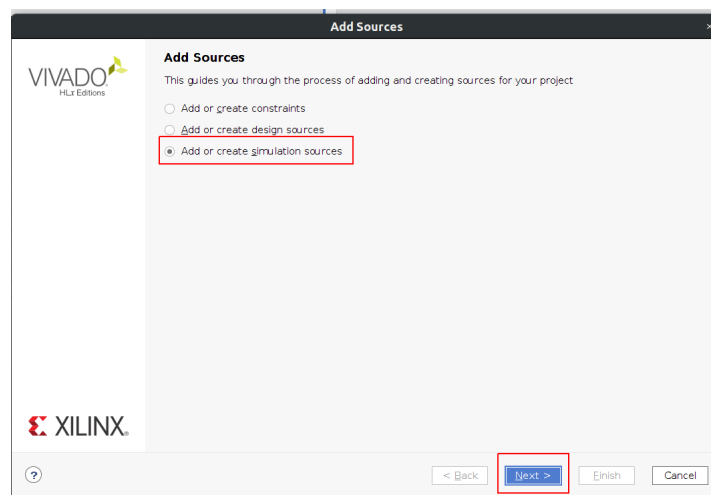
2 Crear archivo de simulación

2.1 Crear el recurso:

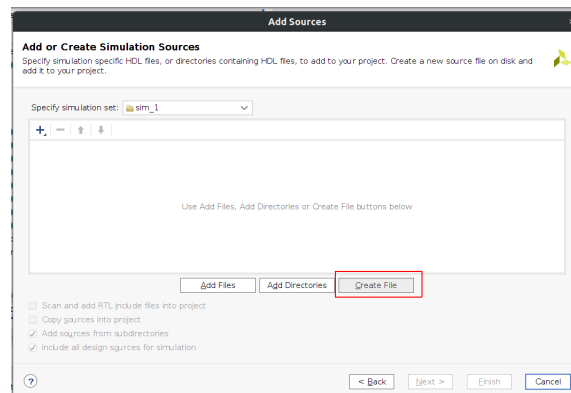
Primeramente se debe apretar en la ventana de *Sources* añadir *Add Source* (también se encuentra en la columna lateral derecha, el llamado *Project Manager*):



En segundo lugar, seleccionar *Add or create simulation sources* y luego apretar siguiente:

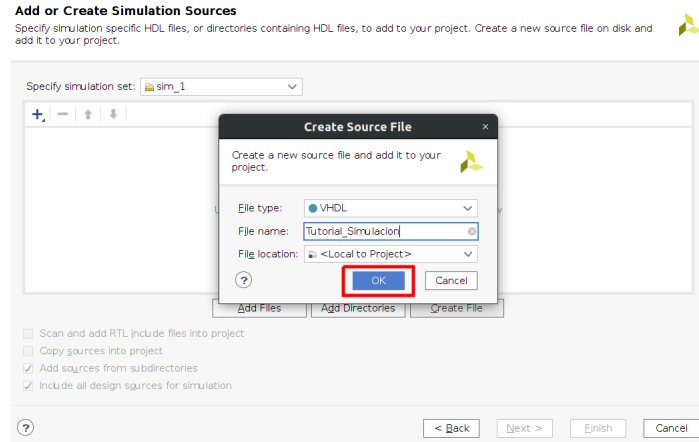


Una vez hecho eso, seleccionar la opción *Create File*:

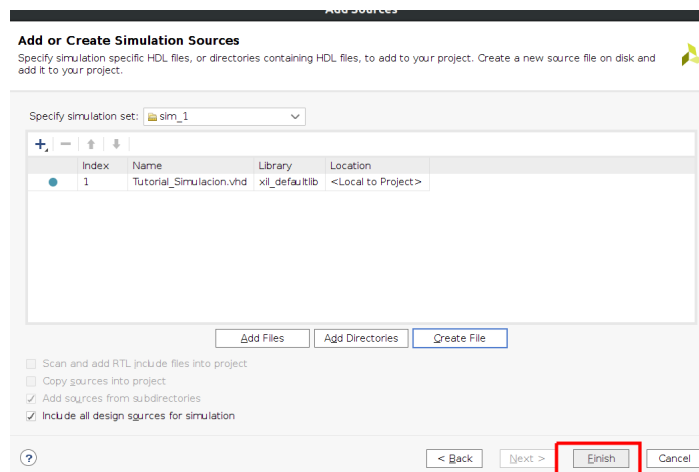


2.2 Dar el nombre al archivo de simulación:

Notar que el nombre del archivo a la simulación será **Tutorial_Simulacion**, distinto al **Archivo_simular**. El primero es que Vivado ocupará para hacer la simulación y el segundo es el que describe el archivo que podría ser compilado para ser ocupado en la placa. No confundir.

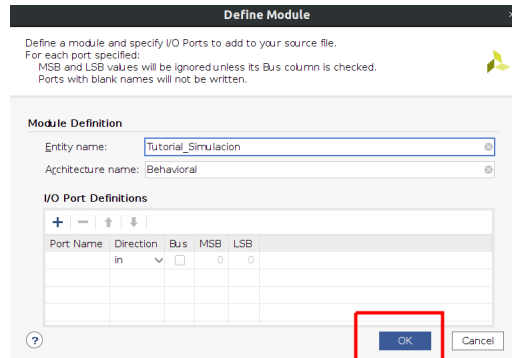


Una vez dado el nombre se presiona *Finish* como se muestra en la imagen:

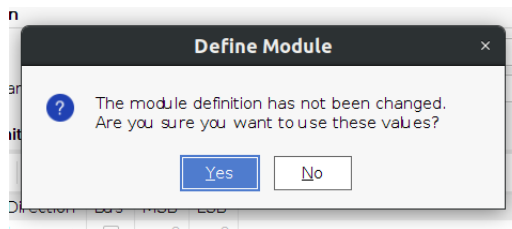


2.3 Definir el modulo:

Una vez terminado el paso anterior, aparecerá una ventana de definición de modulo similar a la de esta imagen:



A diferencia del tutorial anterior, se ignorará el ingreso de puertos, no se escribirá nada y solo se presionará *OK*. Por esta misma acción aparecerá un recuadro como el siguiente:



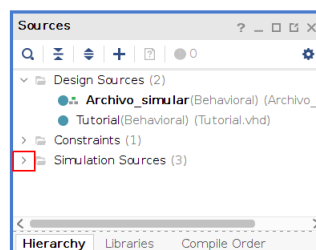
Se presionará *Yes*. Se tomarán estas decisiones, debido a que se escribirá el archivo directamente como se mostrará más adelante en el documento.

3 Preparar archivo de simulación

Similar a la primera sección de este escrito, debemos preparar el archivo a simular para que Vivado lo seleccione correctamente.

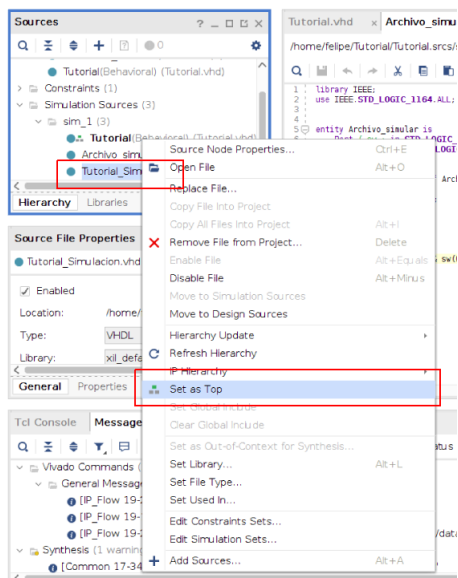
3.1 Ubicar archivo

Para esto primero se redirige a la ventana de **Sources** y se expande la carpeta llamada *Simulation Sources*:



3.2 Seleccionar archivo de simulación

Luego expandir la carpeta `sim_1` donde nos mostrará todas las entidades definidas. Una vez hecho eso al igual que en la primera parte de este tutorial, se selecciona el archivo que anteriormente creado, se presiona click derecho y se selecciona la opción *Set as Top*

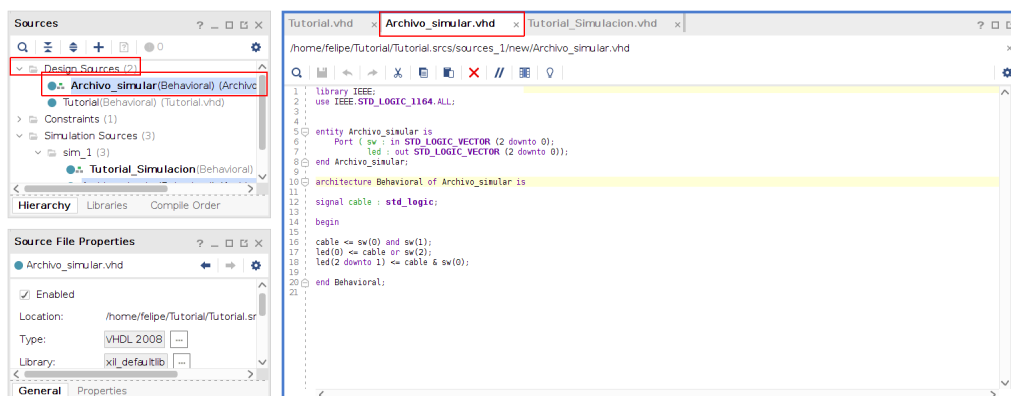


4 Escribir archivo de simulación

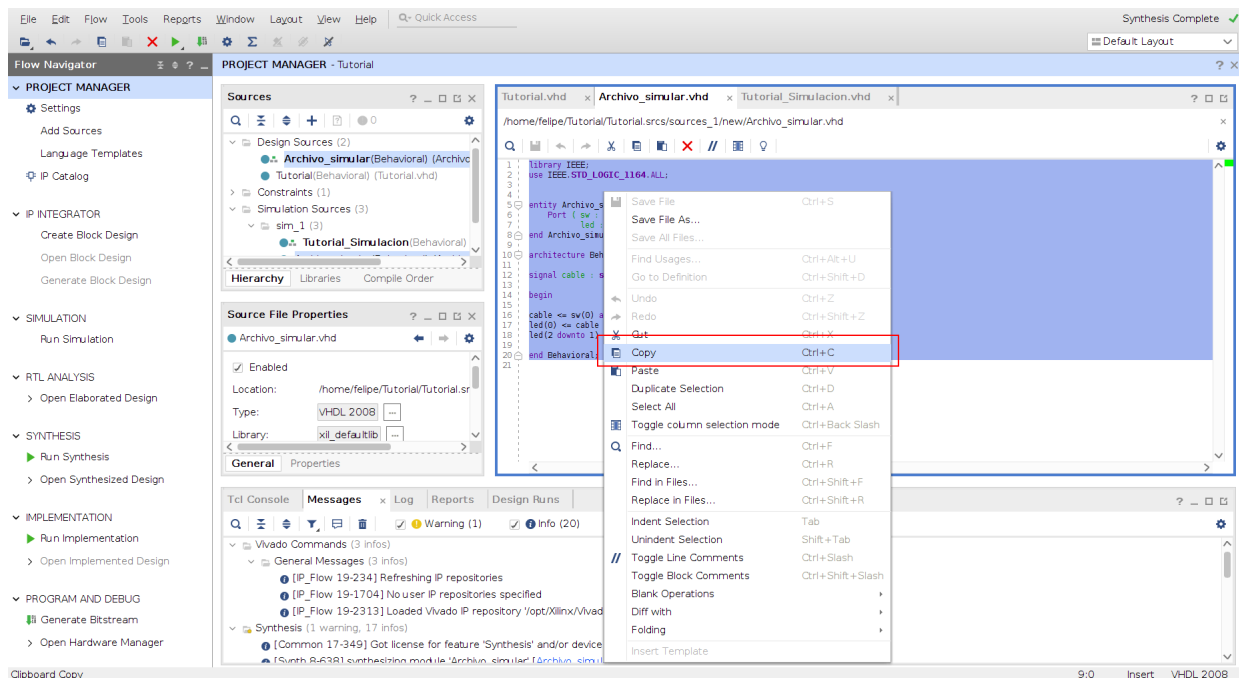
La escritura de un archivo de simulación se divide en dos partes primeramente se debe instanciar la entidad a ser simulada, y en segundo lugar se debe entregar todas las entradas que se deseen simular para que luego Vivado las despliegue. Para la primera parte se ocupará un sitio web que proporcionará todo lo necesario para la primera parte, cosa de solo tener que preocuparse de la segunda parte.

4.1 Copiar el archivo a ser simulado

Volviendo a la sección *Source*, dentro de la carpeta *Design Source*, seleccionar el archivo a ser simulado:



Luego seleccionar el archivo completo, y copiarlo como se muestra en la imagen:

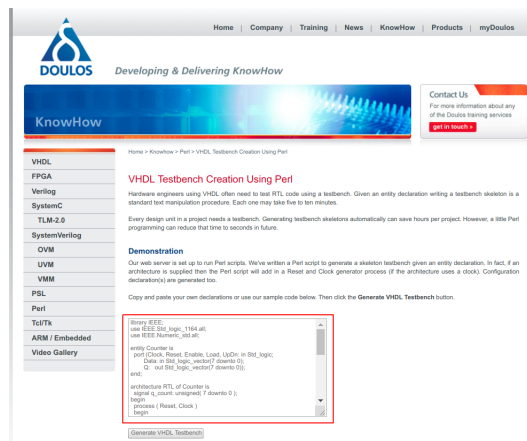


4.2 Redirigirse al sitio web Doulos

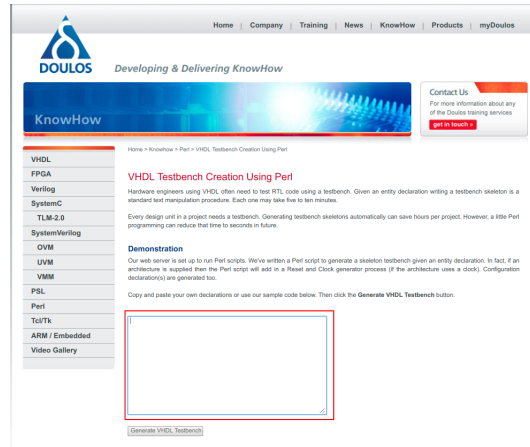
Doulos es un sitio web que proporciona una herramienta para generar código base para simular. Ir al siguiente URL.

4.3 Descartar código de ejemplo:

Dentro del sitio habrá un código de muestra, como se ve en la imagen:

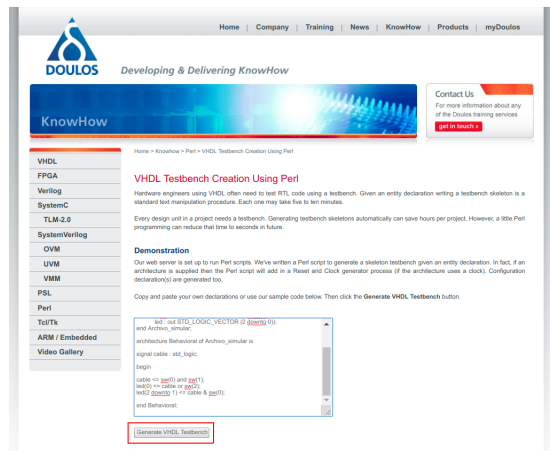


Se selecciona todo y se descarta hasta que quede de esta forma:



4.4 Pegar archivo a ser simulado:

Se pega el archivo a ser simulado que y luego se presiona en *Generate VHDL Testbench*:



NOTA: El archivo a ser simulado no debe tener ningún input, output o variable con el nombre **clock**.

4.5 Copiar archivo base de simulación:

Se abrirá una ventana similar a la siguiente:

```

VHDL Testbench

No reset code was detected. This may be because no architecture was supplied or because a supplied architecture does not contain code such as:
if Reset = '1' then
  elsif rising_edge( Clock ) then
    ...
end if;

No clock code was detected. This may be because no architecture was supplied or because a supplied architecture does not contain code such as:
rising_edge( clock ) then
  ...
end if;

-- Testbench created online at:
-- www.doulos.com/knownhow/perl/testbench_creation/
-- Copyright Doulos Ltd
-- 50, 63 November 2002

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity Archivo_simular_tb is
end;

architecture bench of Archivo_simular_tb is
  component Archivo_simular
    Port ( pw : in STD_LOGIC_VECTOR (2 downto 0);
          led : out STD_LOGIC_VECTOR (2 downto 0));
  end component;

  signal pw: STD_LOGIC_VECTOR (2 downto 0);
  signal led: STD_LOGIC_VECTOR (2 downto 0);

begin

  uut: Archivo_simular port map ( pw => pw,
                                led => led );

  stimulus: process
  begin
    -- Put initialization code here

    -- Put test bench stimulus code here

    wait;
  end process;

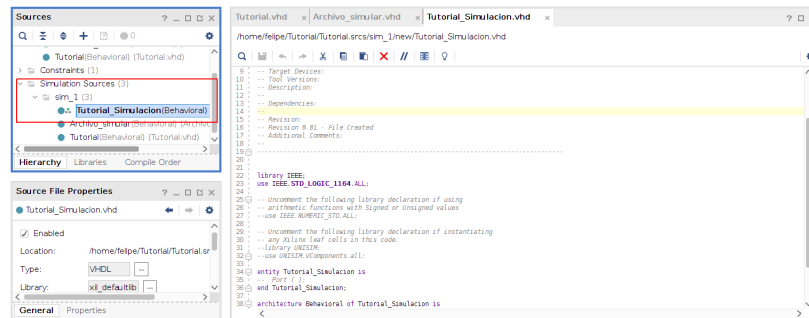
end;

```

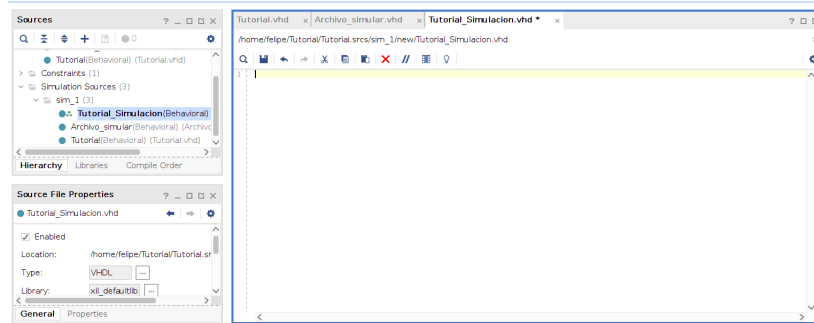
De aquí solo se copiará desde la línea **library IEEE;** hasta la línea **end;**. El resto del texto será ignorado:

4.6 Pegar archivo base de simulación:

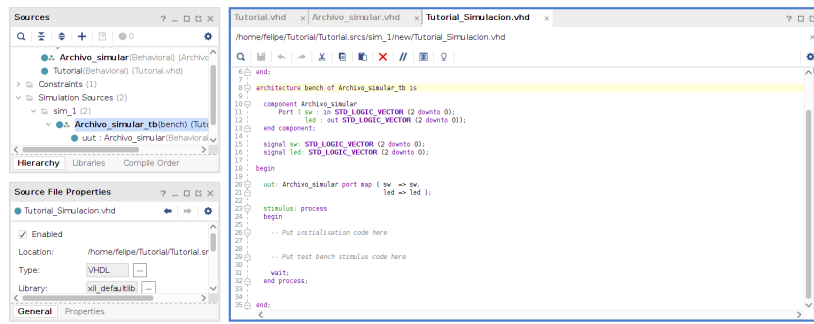
Ahora es cosa de volver a Vivado, ir a la ventana *Source*, dentro de la carpeta *Simulation Source* y dentro de ella a la carpeta *sim_1* ubicar el archivo de simulación (en este caso **Tutorial_Simulacion**)



Se selecciona el archivo y se descarta todo lo que tiene escrito:



Finalmente pegar el archivo base dado por Doulos:



Luego se guarda apretando Ctrl+S.

NOTA: Puede que aparezca una ventana mencionando que el código es inválido. Simplemente apretar OK con las opciones por defecto, y Vivado lo solucionará de manera automática.

4.7 Escribir simulación:

Dentro del código base tendremos dos líneas de comentarios que comienzan con la palabra *Put*. Dentro de esas líneas escribiremos nuestra simulación. Para ello se escribir el comportamiento exacto de todos los input y un tiempo de espera. En este caso solo tenemos un input (los *switches*) por lo que solo necesitamos escribir una línea de código indicando su comportamiento. Sin embargo, si nuestra entidad tuviese 5 entradas, pues deberíamos escribir una línea por cada entrada, y luego el tiempo de espera.

Para este ejemplo se simulará que los *switches* tendrán valores de cero hasta tres, con una espera de 100 nanosegundos por cada estado, como se muestra a continuación:

```
begin

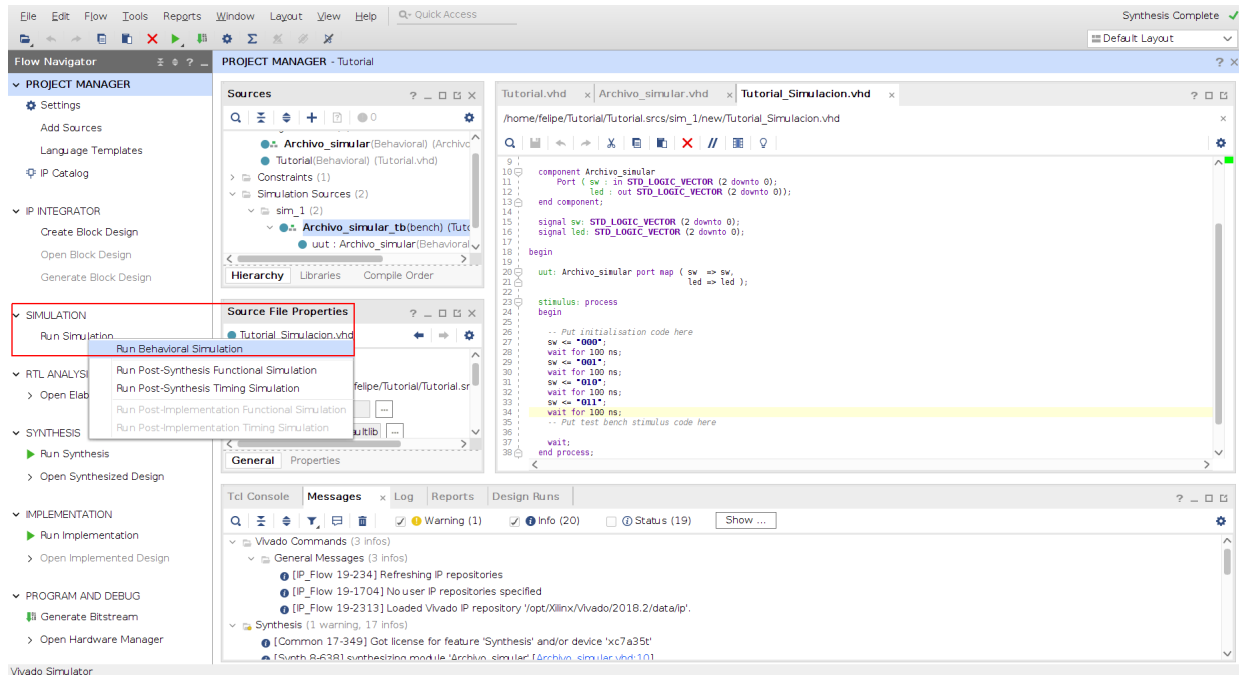
    -- Put initialisation code here
    SW <= "000";
    wait for 100 ns;
    SW <= "001";
    wait for 100 ns;
    SW <= "010";
    wait for 100 ns;
    SW <= "011";
    wait for 100 ns;
    -- Put test bench stimulus code here

    wait;
end process;
```

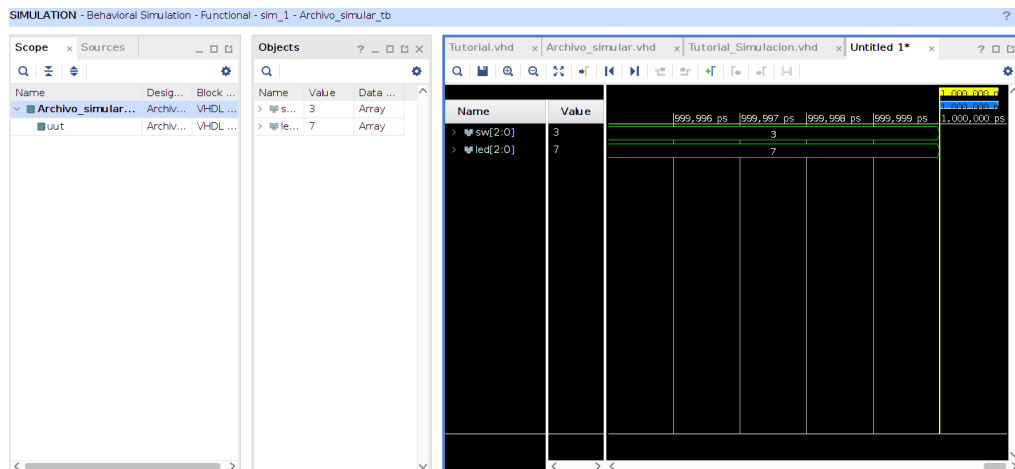
Luego se guarda el archivo que esta listo para ser simulado.

5 Correr simulación:

Dentro del **Project Manager**, seleccionar **Run Simulation**, y luego **Run Behavioral Simulation** como se muestra en la imagen:



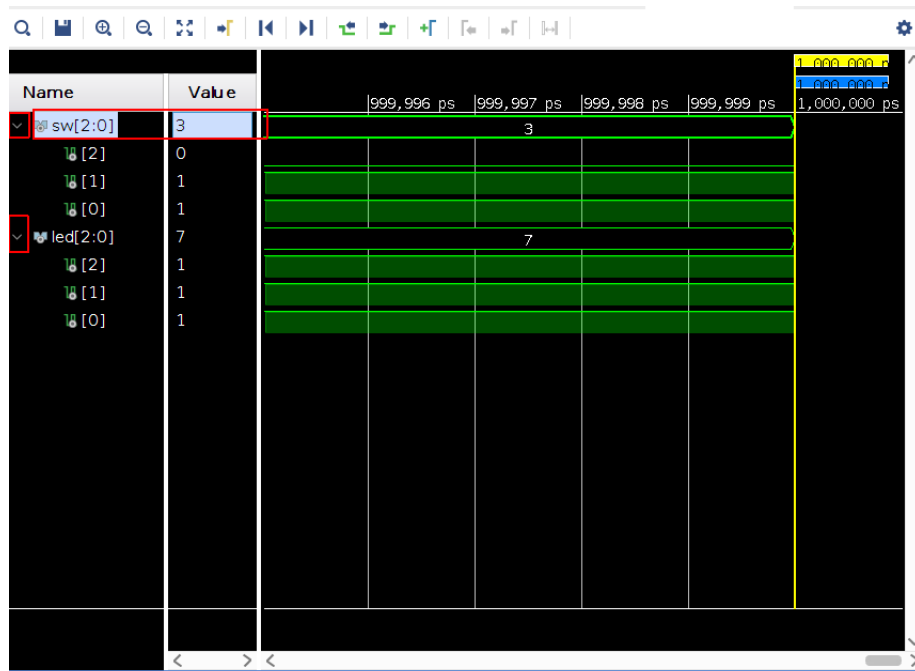
Luego esperar unos minutos a que cargue. Hasta llegar a una pantalla similar a esta:



6 Uso de la simulación:

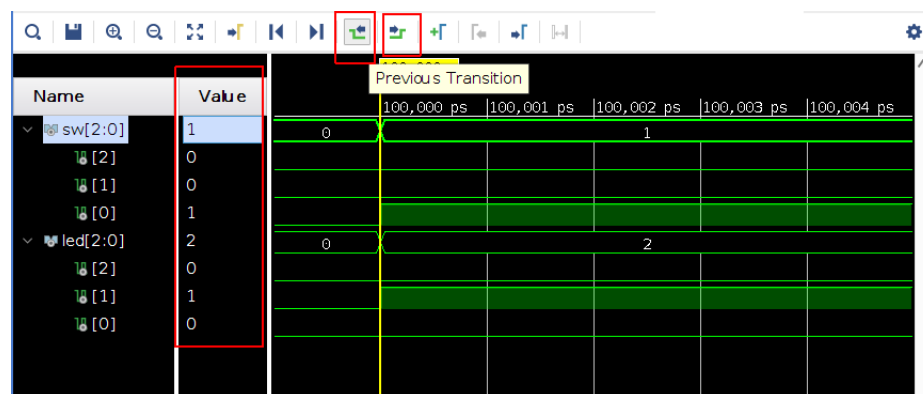
6.1 Expandir las señales y seleccionar entradas

Al centrarse solo en la pantalla negra se verá que en la columna *Name* están los nombres de las entradas y salidas, el cual podemos expandir para ver sus valores individuales. Mientras que en la columna *Value* nos dará la información sobre el valor en hexadecimal del conjunto de bits. Para ver el comportamiento seleccionaremos el input **sw** como se muestra en la imagen:



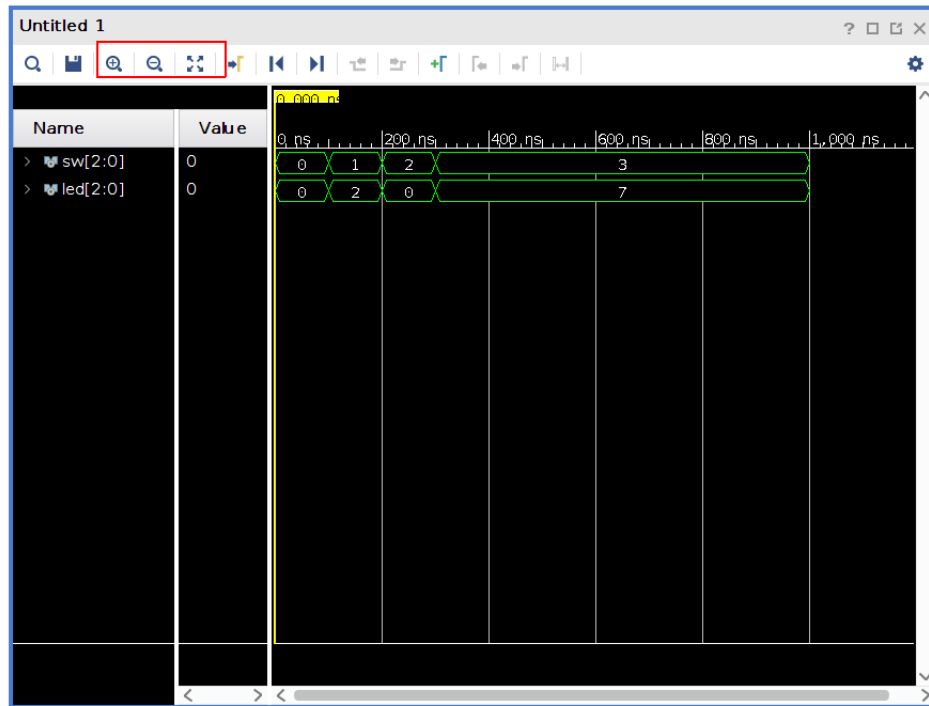
6.2 Revisión de estados

Presionar los botones de *Previous Transition* y *Next Transition* (ver imagen) para visualizar como cambian las señales, como cambian sus valores y así verificar si el circuito se comporta como se espera:



6.3 Expandir visión

A través de los botones de lupa para acercar y alejarse, o haciendo uso de la tecla Ctrl más la rueda de su *mouse* (o el scroll de su laptop), pueden tener una visión más de cerca o más alejada de las señales indicando sus valores en cada momento, como se muestra en la figura:

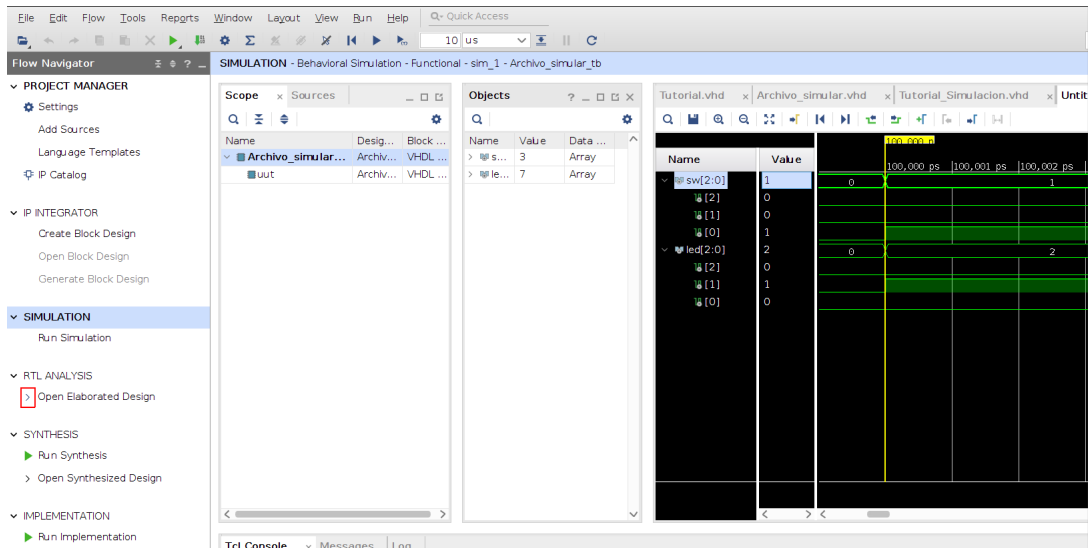


7 Bonus: Uso de Schematic:

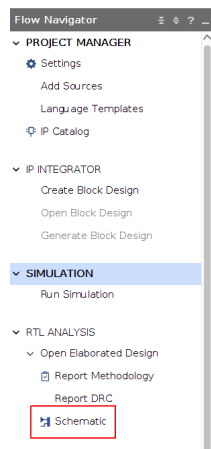
Si acaso con simular no basta para entender el funcionamiento del circuito diseñado, Vivado provee otra opción llamada **Schematic**, que permite visualizar el circuito en forma de diagrama.

7.1 Acceso a Schematic:

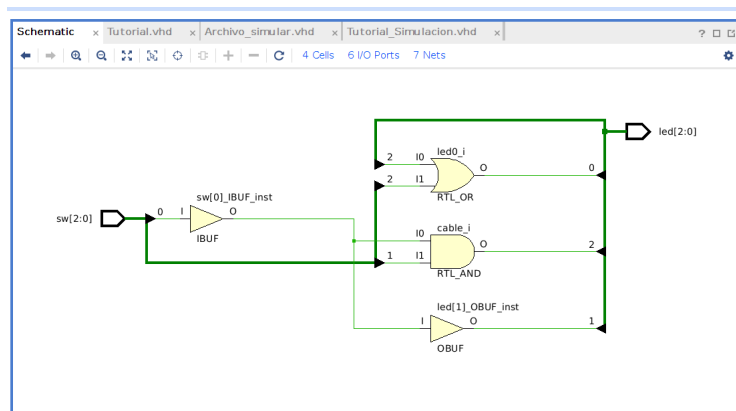
Dirigirse a la columna lateral izquierda en el *Program Manager*, ubicar la sección **RTL ANALYSIS**, y dentro de dicha sección expandir *Open Elaborate Desing* como se muestra en la imagen:



Luego seleccionar **Schematic**:



Luego esperar unos minutos y aparecerá un diagrama completo del archivo a simular:



Bibliografía:

Para más información revisar los siguientes enlaces

- https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_1/ug937-vivado-design-suite-simulation.pdf
- <https://www.youtube.com/watch?v=ShjXQdKdxsE>