

DISEÑO DIGITAL
PRIMERA SESIÓN DE LABORATORIO (PB01)
SEMESTRE ACADÉMICO 2020-2

Horario: TODOS

Duración 230 minutos

Elaborado por: Mario Raffo

ADVERTENCIAS:

- Todo dispositivo electrónico (teléfono, tableta, computadora u otro) deberá permanecer apagado durante la evaluación.
- Coloque todo aquello que no sean útiles de uso autorizado durante la evaluación en la parte delantera del aula, por ejemplo, mochila, maletín, cartera o similar, y procure que contenga todas sus propiedades. La apropiada identificación de las pertenencias es su responsabilidad.
- Si se detecta omisión a los dos puntos anteriores, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- Es su responsabilidad tomar las precauciones necesarias para no requerir la utilización de servicios higiénicos: durante la evaluación, no podrá acceder a ellos, de tener alguna emergencia comunicárselo a su jefe de práctica.
- En caso de que el tipo de evaluación permita el uso de calculadoras, estas no podrán ser programables.
- Quienes deseen retirarse del aula y dar por concluida su evaluación no lo podrán hacer dentro de la primera mitad del tiempo de duración destinado a ella.

TEMA:

**INTRODUCCIÓN AL LABORATORIO Y CIRCUITOS
COMBINACIONALES I**

OBJETIVOS:

- Familiarizarse con el uso de las herramientas y dispositivos que serán empleados a lo largo de todas las sesiones de laboratorio.
- Conocer el flujo de diseño.
- Aprender a utilizar las diferentes herramientas del software QUARTUS PRIME LITE EDITION.
- Familiarizarse con la codificación del lenguaje de descripción de hardware VHDL.
- Implementar circuitos digitales en un FPGA.

Actividades a Realizar

Nº	DESCRIPCIÓN	DURACIÓN APROXIMADA	PUNTOS
1	Introducción.	20 min.	0.0 Pts
2	Explicación del hardware y software a utilizar en el laboratorio.	20 min.	0.0 Pts
3	Experiencia práctica (EP): Uso del software Quartus Prime Lite Edition.	175 min	15.0 Pts
4	Prueba de salida (PS).	15 min.	5.0 Pts

Descripción de Circuitos Digitales: FPGAs, Flujo de Diseño, Descripción, Simulación, Herramientas Quartus y ModelSIM

Los FPGAs (FIELD PROGRAMMABLE GATE ARRAY) presenta una arquitectura compuesta de LEs (Logical Elements) que puede ser configurada de acuerdo a las necesidades del diseñador, logrando así una gran diversidad de aplicaciones, ya sean simples o complejas.

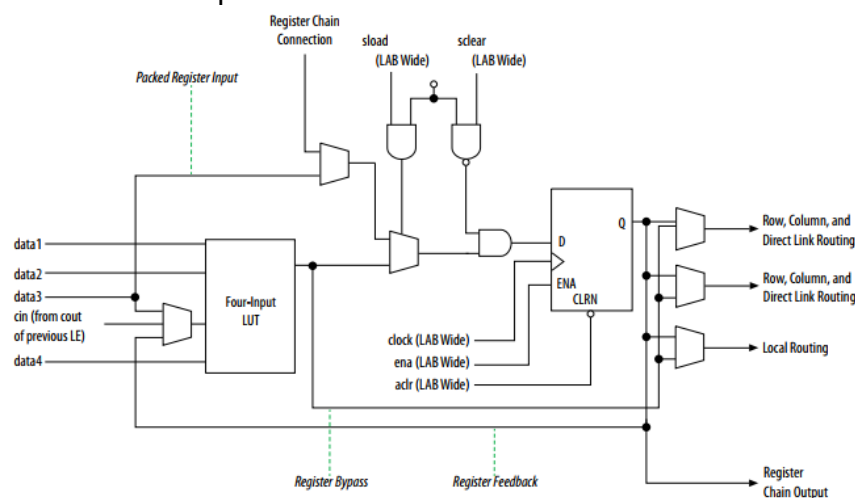
El almacenamiento de estos diseños se realiza por lo general en una SRAM (Static Random Access Memory), pero también presenta otras tecnologías de configuración como la EEPROM/FLASH y la antifuse .



Los LEs (Logical Elements), del FPGA MAX10 [1], están compuestos de:

- Un LUT (Look-Up Table), que sirve para describir una función booleana de 4 entradas
- Un registro programable
- Interconexiones
- Registros Bypass y Feedback
- Modo de funcionamiento normal y aritmético

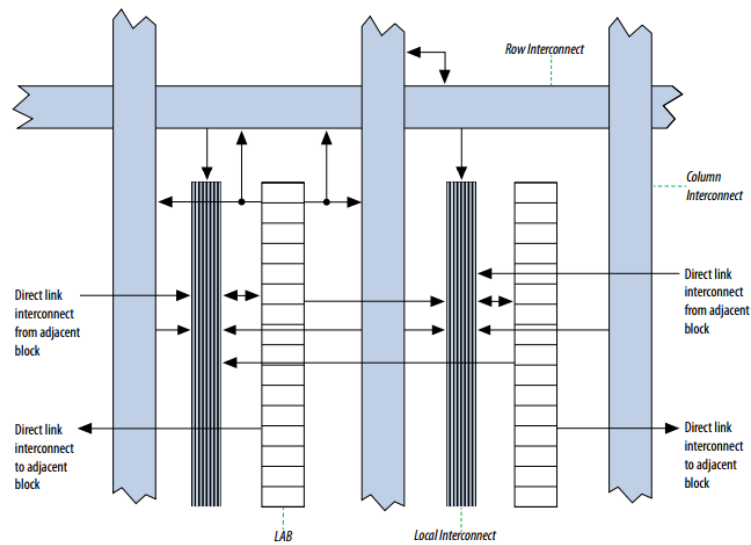
A continuación se ilustra un LE para un modo normal del FPGA MAX10



Los LABs (Logic Array Blocks), del FPGA MAX10 [7], están compuestos de:

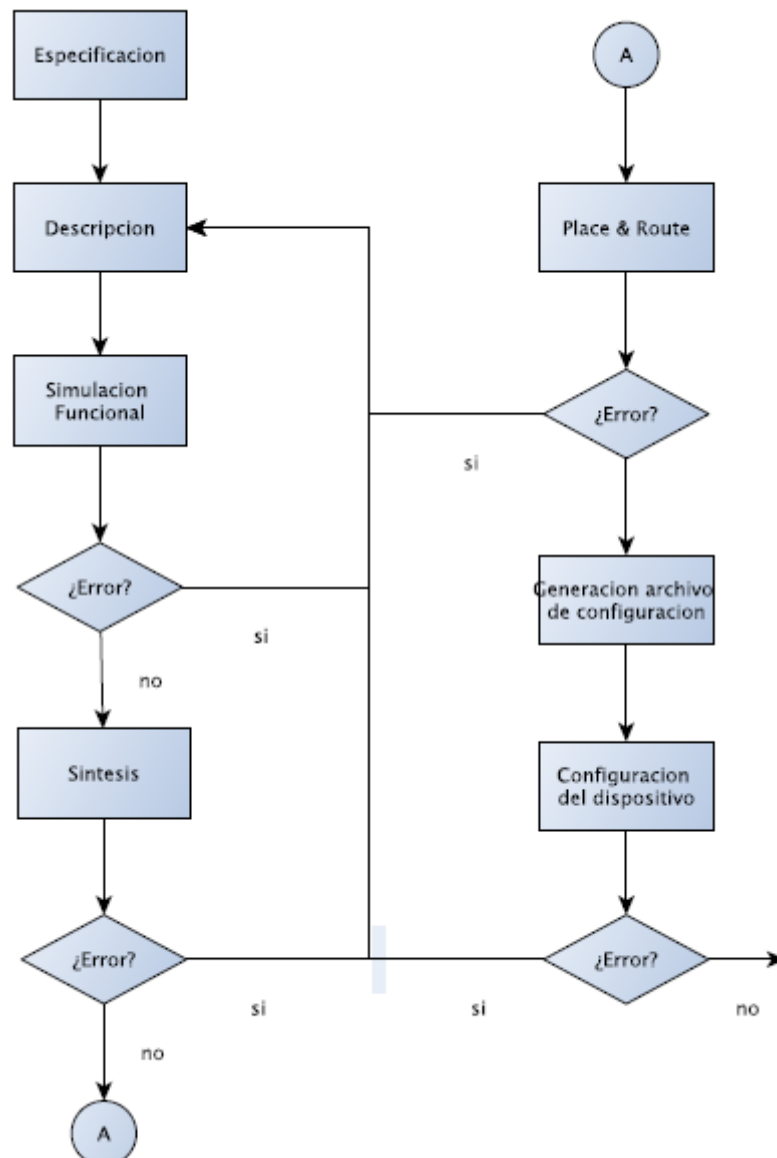
- 16 LEs
- Señales de control de LAB
- Interconexiones

A continuación se presenta la estructura de LABs, del FPGA MAX10



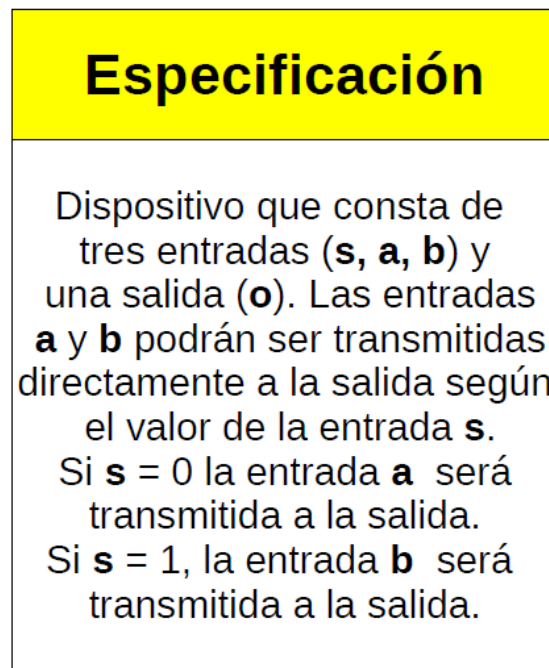
Flujo de Diseño

La figura a continuación presenta el Flujo de Diseño típico para FPGAs [2]



Especificación

La especificación es la etapa del Flujo de Diseño en la cual se indica los requisitos y detalles del sistema digital requerido. La siguiente figura muestra, con un ejemplo, el concepto de especificación.



Descripción

En esta etapa se efectúa la descripción de la especificación. Para este fin se pueden emplear dos de los más utilizados lenguajes de descripción de hardware (HDL), el VHDL y el Verilog HDL.

La figura a continuación presenta, con un ejemplo, el concepto de especificación

Descripción	
VHDL	Verilog HDL
<pre>1 library ieee; 2 use ieee.std_logic_1164.all; 3 4 entity lab0 is 5 port (signal s : in std_logic; 6 signal a : in std_logic; 7 signal b : in std_logic; 8 signal o : out std_logic); 9 end lab0; 10 11 architecture behavior of lab0 is 12 13 begin 14 15 o <= (not(s) and a) or (s and b); 16 17 end behavior;</pre>	<pre>1 `timescale 1 ns / 1 ns 2 3 module lab0 4 (5 input wire s, 6 input wire a, 7 input wire b, 8 output wire o); 9 10 assign o = (~s & a) (s & b); 11 12 endmodule</pre>

En curso de Diseño Digital se emplea el lenguaje de descripción de Hardware VHDL.

VHDL es un acrónimo que representa la combinación de VHSIC y HDL.

VHSIC = V ery High Speed Integrated Circuits.

HDL = Hardware Description Language.

VHDL= VHSIC Hardware Description Language.

VHSIC: Grupo creado por el Departamento de Defensa (DoD) de los Estados Unidos.

El VHDL ha sido estandarizado por la IEEE a través del:

- i. Primera versión del VHDL: VHDL 1987.
- ii. Segunda versión del VHDL : VHDL 1993.
- iii. Tercera versión del VHDL: VHDL 2002.
- iv. Cuarta versión del VHDL : VHDL 2008.

Básicamente el VHDL es utilizado para describir circuitos a nivel académico, mientras que el Verilog HDL es empleado en la industria.

La siguiente figura muestra una descripción efectuada por medio del VHDL y en ella se pueden apreciar las tres partes de un código básico en VHDL, además de un encabezado que es bueno tenerlo como documentación del circuito.

- i. Declaraciones de Librerías (Library). Contienen las lista de librerías a ser utilizadas en el diseño. Por ejemplo: ieee, std, work, etc.
- ii. Entidad (Entity). Especifica las entradas y salidas del circuito.
- iii. Arquitectura (Architecture). Contiene la descripción del comportamiento del circuito.

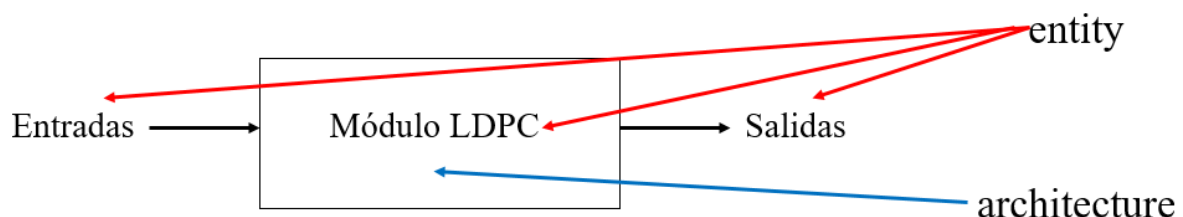
```
1  -----
2  -- Archivo: ex1.vhd
3  -----
4  -- Autor:      Mg. Ing. Mario Raffo
5  -- Email:      mraffo@pucp.edu.pe
6  -- Entidad:    Pontificia Universidad Católica del Perú (PUCP)
7  -- Facultad:   Estudios Generales Ciencias (EE.GG.CC)
8  -- Curso:      IIEE04 - Diseño Digital
9  -----
10 -- Historia de Versión:
11 -- Versión 1.0 (10/04/2020) - Mg. Ing. Mario Raffo
12 -----
13 -- Descripción:
14 -- Separata de Ejercicios Clase 1, 2, 3
15 -- Circuito correspondiente al Ejercicio 1, el cual es una función booleana
16 -- Entradas: a, b, c
17 -- Salida: f
18 -----
19
20 library ieee;
21 use ieee.std_logic_1164.all;
22
23 entity ex1 is
24     port(signal a : in std_logic;
25           signal b : in std_logic;
26           signal c : in std_logic;
27           signal f : out std_logic);
28 end ex1;
29
30 architecture structural of ex1 is
31
32 begin
33
34     f <= not(a) or (b and(a or not(c)));
35
36 end structural;
```

Encabezado del circuito

Declaración de librerías

Entidad

Arquitectura

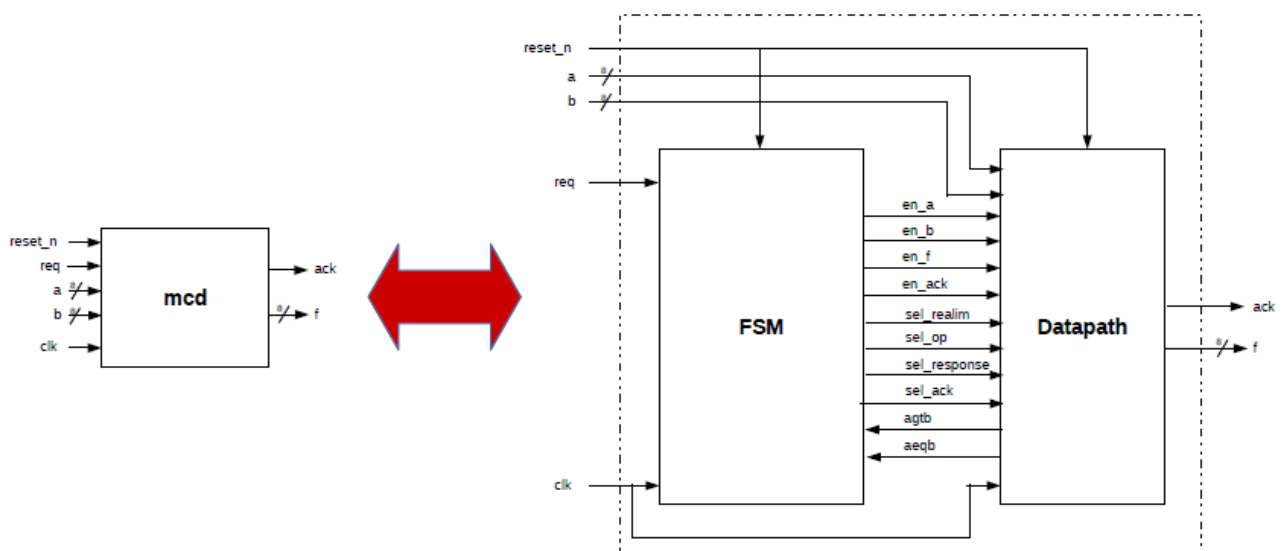


Las metodologías de descripción de hardware que se usarán en el curso son la comportamental y la secuencia

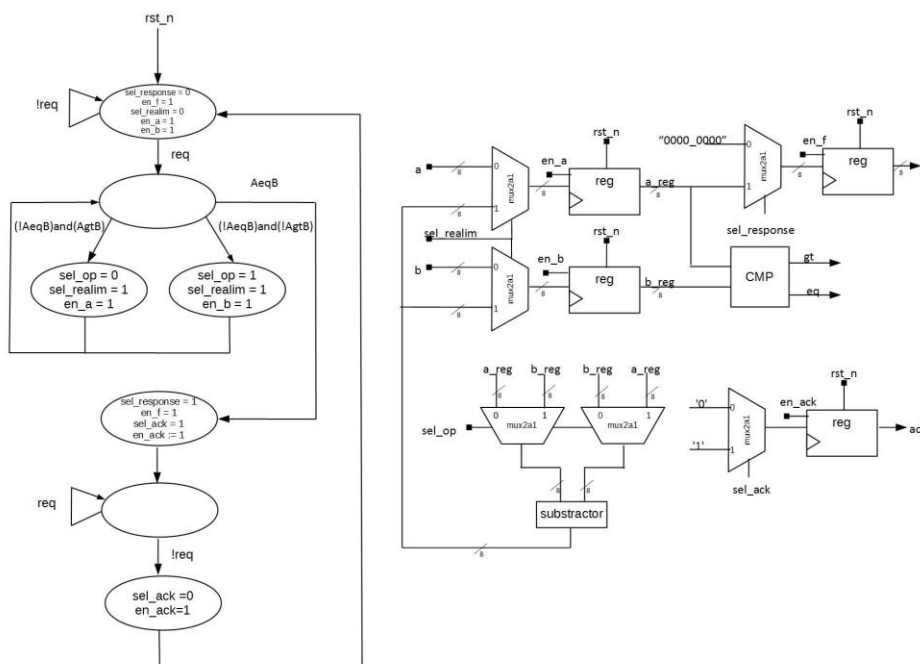
La comportamental, como su nombre lo sugiere, describe el comportamiento del circuito. La siguiente figura ilustra, con un ejemplo, el concepto de descripción comportamental.

Descripción Comportamental		
Nivel operadores lógicos	Nivel concurrente	Nivel secuencial
<pre> 1 library ieee; 2 use ieee.std_logic_1164.all; 3 4 entity lab0 is 5 port (signal s : in std_logic; 6 signal a : in std_logic; 7 signal b : in std_logic; 8 signal o : out std_logic); 9 end lab0; 10 11 architecture behavior of lab0 is 12 13 begin 14 15 o <= (not(s) and a) or (s and b); 16 17 end behavior; </pre>	<pre> 1 library ieee; 2 use ieee.std_logic_1164.all; 3 4 entity lab0 is 5 port (signal s : in std_logic; 6 signal a : in std_logic; 7 signal b : in std_logic; 8 signal o : out std_logic); 9 end lab0; 10 11 architecture behavior of lab0 is 12 13 begin 14 15 o <= a when (s='0') else 16 b when (s='1') else 17 '-'; 18 19 end behavior; </pre>	<pre> 1 library ieee; 2 use ieee.std_logic_1164.all; 3 4 entity lab0 is 5 port (signal s : in std_logic; 6 signal a : in std_logic; 7 signal b : in std_logic; 8 signal o : out std_logic); 9 end lab0; 10 11 architecture behavior of lab0 is 12 13 begin 14 15 MR: process(s,a,b) 16 begin 17 case(sel) is 18 when '0' => o <= a; 19 when '1' => o <= b; 20 when others => o <= '-'; 21 end case; 22 end process MR; 23 24 end behavior; </pre>

La estructural, como su nombre lo sugiere, divide la descripción del circuito en más de un sub-modulo, para así hacer más simple el diseño de un circuito complejo. La figura a continuación ilustra, con un ejemplo, el concepto de descripción estructural.

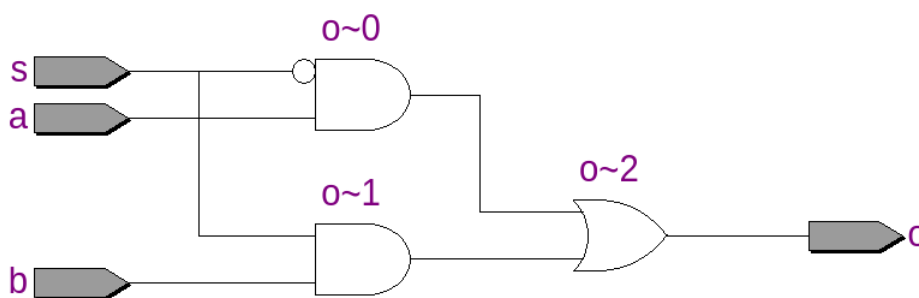


La figura a continuación muestra mayores detalles del circuito descrito estructuralmente.



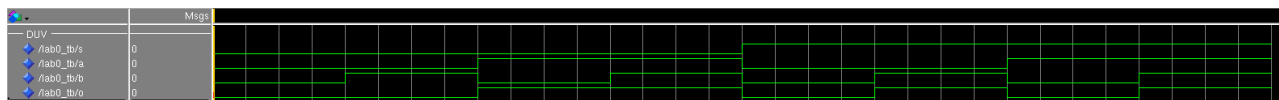
Vista RTL

La vista RTL sirve para poder ver la relación de la salida respecto de las entradas del circuito. Por lo general, permite ver si a nivel VHDL no se ha usado una entrada definida. Un ejemplo presenta la figura a continuación.



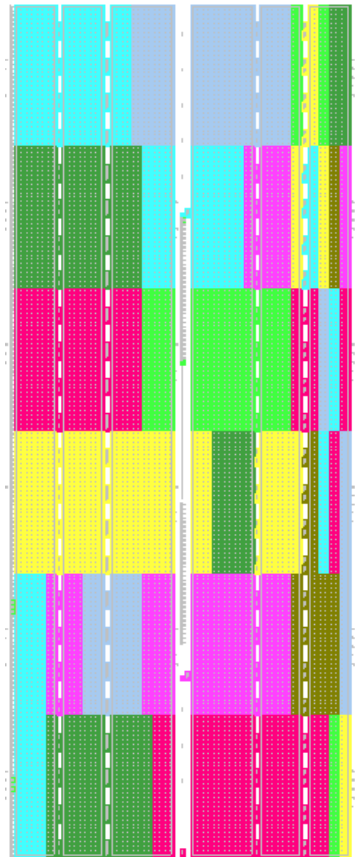
Simulación Funcional

En esta etapa se verifica el funcionamiento de la descripción del proyecto de acuerdo al comportamiento que el mismo debe presentar. La siguiente figura presenta una simulación funcional de un circuito, empleando la herramienta ModelSim-Altera



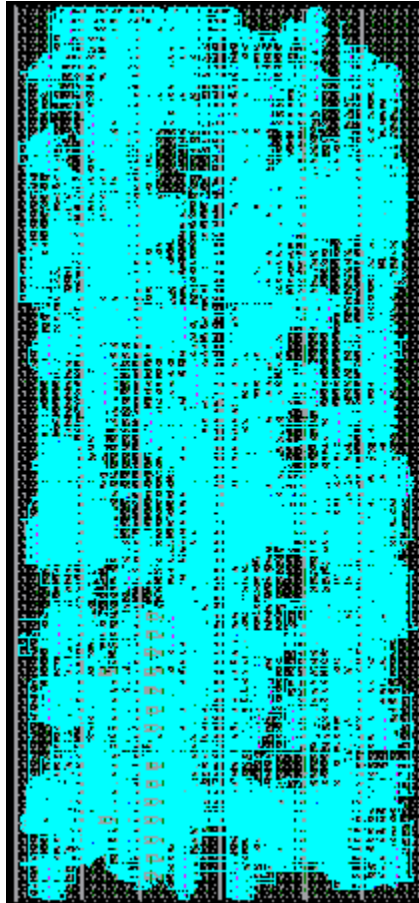
Floorplanning

El Floorplanning está referida a la síntesis física y empieza con la delimitación del área (LEs o CLBs) en la que cada submódulo del proyecto va a estar delimitado. La figura a continuación [3] muestra este concepto.



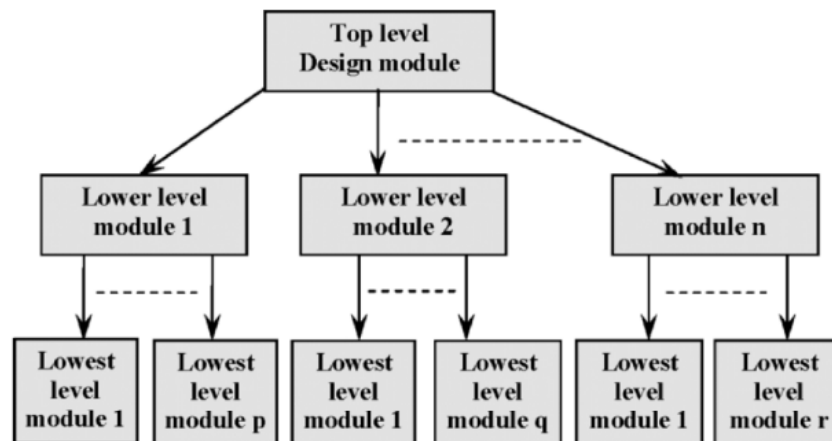
Place and Route

La etapa del Place and Route, después de haber delimitado el área (LEs o CLBs) en la que cada submódulo del proyecto, continúa con el posicionamiento físico del circuito en un determinado lugar (celdas est_andar, i/o, etc) además de efectuar la síntesis de los caminos de la señal de clock y de reset (Placement), para después proceder a conectar cada elemento del circuito (Routing). La siguiente figura [3] muestra este concepto.



Metodología de Diseño Top-Down

De todas las metodologías de diseño de sistemas digitales, la que se utilizará en el curso 1IEE04 - Diseño Digital es la Top-Down, ilustrada por la figura a continuación [6]. La misma está basada en dividir el circuito en más de un nivel. Cada nivel estará compuesto de módulos. Hay que realizar simulaciones funcionales para verificar del diseño en cada nivel. Un problema es que no se tiene métricas de los submódulos (tiempos de retardo, consumo de procesamiento (hot spot), etc) ya que el layout no es conocido hasta el fin del flujo de diseño.



Código VHDL

Un código debe tener un encabezado. El nombre de la entidad debe ser el nombre del archivo. Si el código es para testbench, la entidad debe tener un sufijo " tb" y si es para un package, debe tener el sufijo " pkg". El VHDL no distingue entre mayúsculas y minúsculas, por ello debe limitar el uso de las mayúsculas al uso de parámetros genéricos y constantes. La siguiente figura resume esto. Para mayores detalles, revise la referencia [7].

```
1 .....
2 -- Archivo: multiplexer2to1.vhdl
3 .....
4 -- Autor:      MSc. Ing. Mario Raffo
5 -- Email:      mraffo@pucp.edu.pe
6 -- Entidad:    Pontificia Universidad Católica del Perú (PUCP)
7 -- Facultad:   Estudios Generales Ciencias (EE.GG.CC)
8 -- Curso:      IEE146 - Laboratorio de Circuitos Digitales
9 .....
10 -- Historia de Versión:
11 -- Versión 1.0 (03/05/2016) - Mario Raffo
12 .....
13 -- Descripción:
14 -- Circuito multiplexor de dos entradas (in_0 e in_1) a una salida (o) de una cantidad
15 -- genérica de bits, por medio de una entrada selectora (sel).
16 .....
17
18 Library ieee;
19 use ieee.std_logic_1164.all;
20
21 entity multiplexer2to1 is
22   generic (WIDTH: natural :=2);
23   port (signal sel : in std_logic;
24         signal in_0 : in std_logic_vector(WIDTH-1 downto 0);
25         signal in_1 : in std_logic_vector(WIDTH-1 downto 0);
26         signal o : out std_logic_vector(WIDTH-1 downto 0));
27 end multiplexer2to1;
28
29 architecture rtl of multiplexer2to1 is
30
31 begin
32
33   with sel select
34     o <= in_0 when '0',
35         in_1 when '1',
36         (others => '-') when others;
37
38 end rtl;
```

Referencias

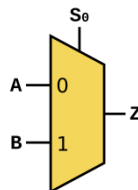
- [1] https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/max-10/m10_handbook.pdf
- [2] Eduardo Augusto Bezerra, Djones Vinicius Lettnin. Synthesizable VHDL Design for FPGAs. Capítulo 1
- [3] Mario Raffo: Desenvolvimento de um Sistema Dinamicamente Reconfigurável baseado em Redes Intra-Chip e Ferramenta para o Posicionamento de Módulos. Tese de Mestrado. Universidade de Sao Paulo, 2010.
- [4] I. Herrera-Alzu and M. López-Vallejo. Design Techniques for Xilinx Virtex FPGA Configuration Memory Scrubbers.
- [5] <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=218&No=1021>
- [6] Dr. S. Ramachandran: Digital VLSI Systems Design A Design Manual for Implementation of Projects on FPGAs and ASICs Using Verilog. Capítulo 6.
- [7] Pablo Cisterna: Design and Coding Style Guidelines for Synthesizable VHDL-FPGA Code.

Experiencia N°1: Uso del Software Quartus Prime Lite Edition

Circuito Multiplexor 2 a 1

Desarrollo de un circuito multiplexor en Quartus así como de simulación y configuración/programación en el dispositivo MAX10.

Un circuito multiplexor permite canalizar varias entradas hacia una salida. Un multiplexor de 2 a 1 dispone de dos entradas, una salida y un selector tal como se muestra en la figura.



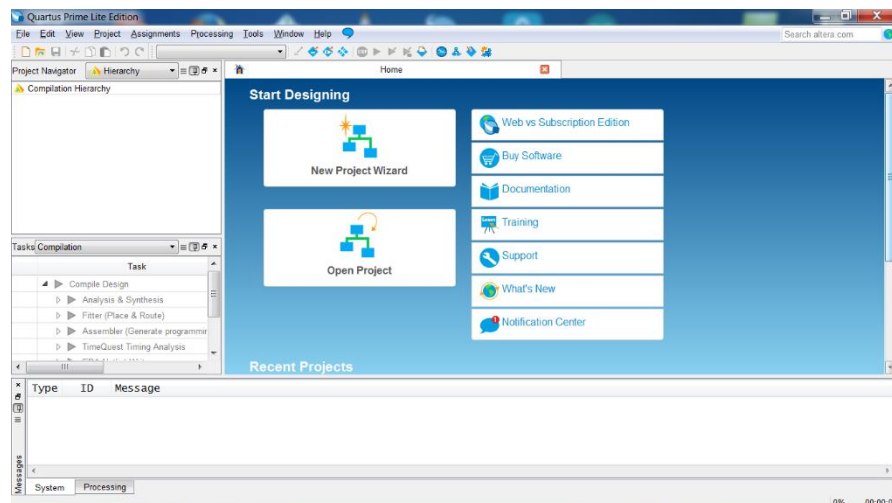
La salida Z será igual a A si es que S es igual a '0' y será igual a B si es que S es igual a '1'. La tabla de verdad es:

N° fila	S	A	B	Z
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

La función simplificada es $Z = S'.A + S.B$

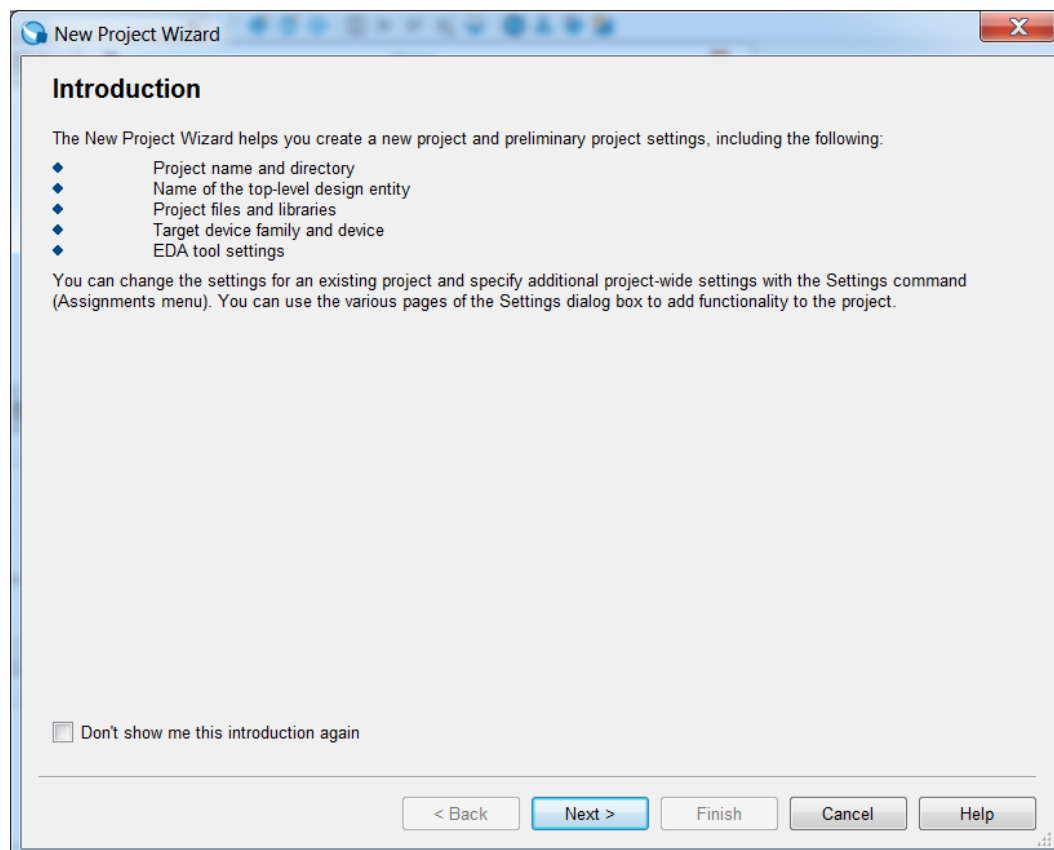
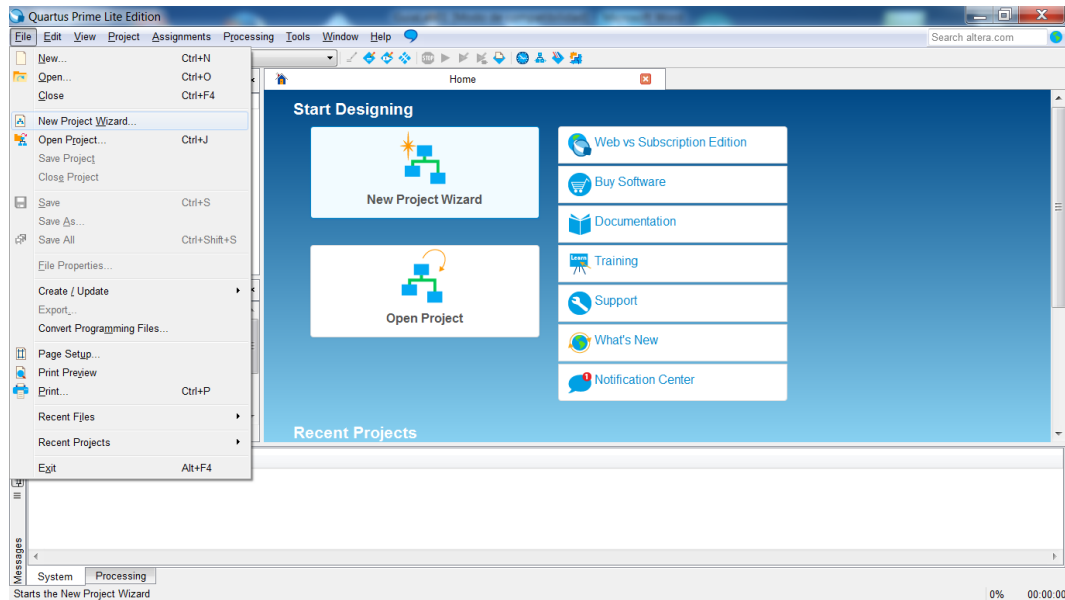
PROCEDIMIENTO

EJECUTAR EL PROGRAMA QUARTUS PRIME LITE EDITION



CREAR UN NUEVO PROYECTO

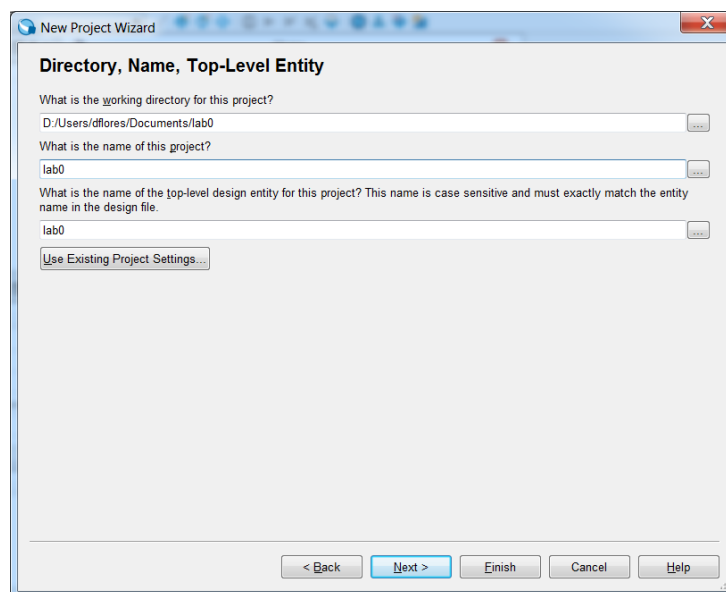
Creación de un proyecto: File -> New Project Wizard. En la ventana hacer click en next



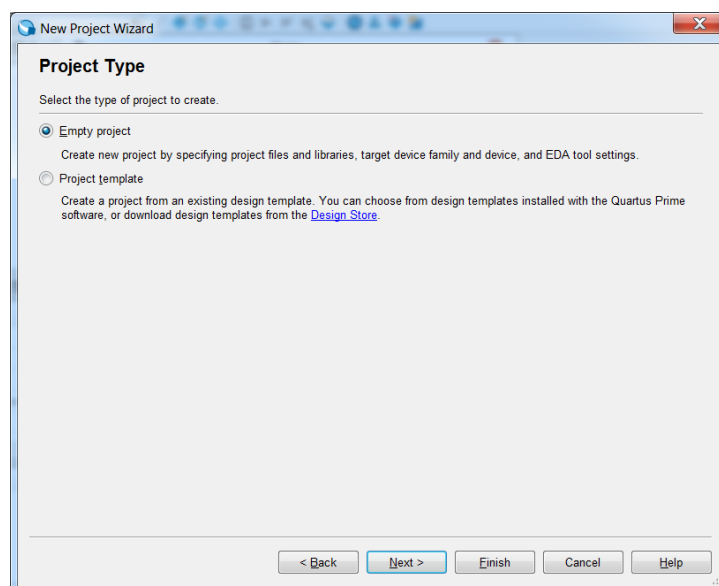
Aquí se debe ingresar 3 datos:

- Ruta de trabajo (c:/temp/1IEE04/h44xy/ex1), donde h44xy se reemplaza por el horario al que pertenece.
- El nombre del proyecto (lab0)
- El nombre de la Entidad Top (nombre del archivo principal), lab0

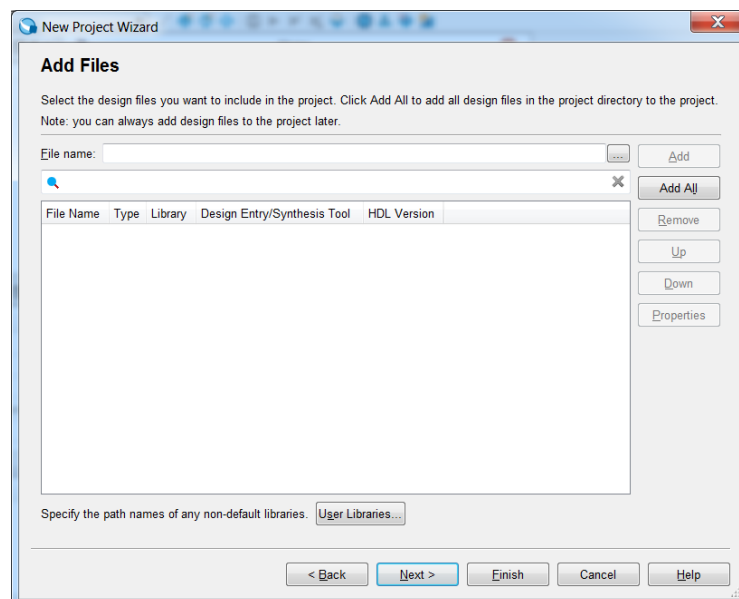
La ruta y el directorio, no se debe tener espacio en blanco ni caracteres en Castellano como la tilde, la ñ. Considere que el nombre del proyecto será igual al nombre del circuito (nombre de la entidad). Este nombre deberá tener las restricciones aplicadas anteriormente. Después de completar esto, dar click en next.



Seleccionar Empty project y dar click en next



Como no se tiene aún el circuito descrito con VHDL, dar click en next.

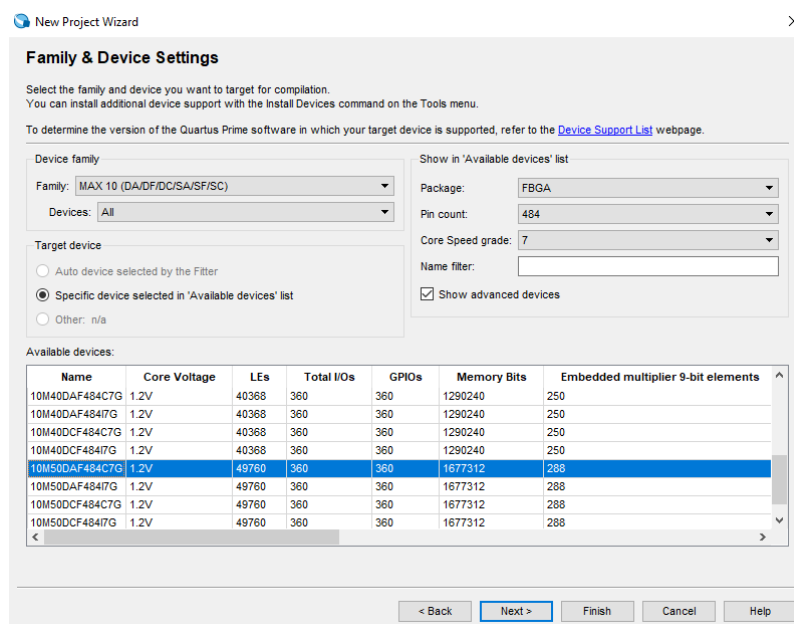


Para realizar el diseño del circuito en VHDL, se debe usar la herramienta Quartus II 15.1, en la cual se elegirá como FPGA:

- Familia: MAX 10
- Package: FBGA
- Pin count 484
- Core Speed grade 7
- Dispositivo: 10M50DAF484C7G

Dar click en Next

Los pasos en letra azul son opcionales, pero ayudan a reducir las opciones entre las que se buscará el Dispositivo



Para realizar el diseño del circuito en VHDL, se debe usar la herramienta Quartus II 13.01 (solamente los alumnos cuya Laptop/PC sea a 32 bits), en la cual se elegirá como FPGA:

- Familia: Cyclone IV E
- Package: FBGA
- Pin count 780
- Core Speed grade 7
- Dispositivo: EP4CE115F29C7

Dar click en next

Family & Device Settings

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: Cyclone IV E

Devices: All

Target device

☐ Auto device selected by the Filter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: FBGA

Pin count: 780

Core Speed grade: 7

Name filter:

☒ Show advanced devices

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elements
EP4CE40F29C7	1.2V	39600	533	533	1161216	232
EP4CE40F29I7	1.2V	39600	533	533	1161216	232
EP4CE55F29C7	1.2V	55856	375	375	2396160	308
EP4CE55F29I7	1.2V	55856	375	375	2396160	308
EP4CE75F29C7	1.2V	75408	427	427	2810880	400
EP4CE75F29I7	1.2V	75408	427	427	2810880	400
EP4CE115F29C7	1.2V	114480	529	529	3981312	532
EP4CE115F29I7	1.2V	114480	529	529	3981312	532

< Back Next > Finish Cancel Help

Aparecerá la siguiente ventana, verifique que estén seleccionado exactamente lo mismo y de click en next

EDA Tool Settings

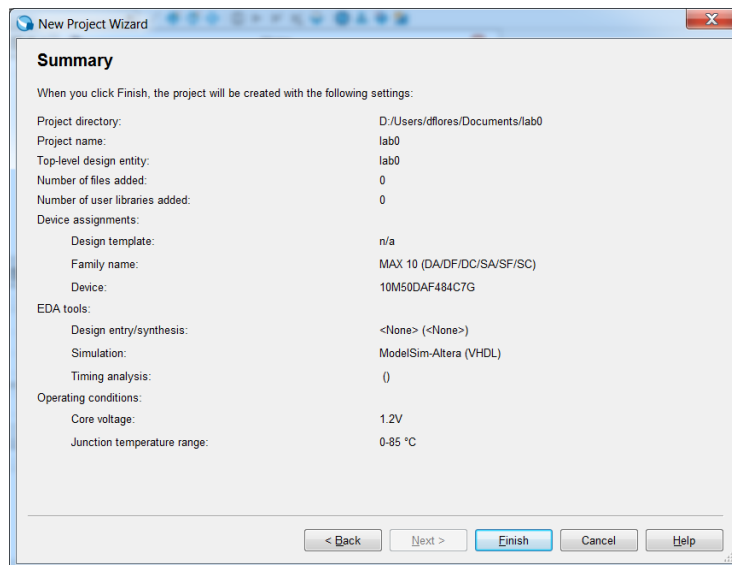
Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	VHDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back Next > Finish Cancel Help

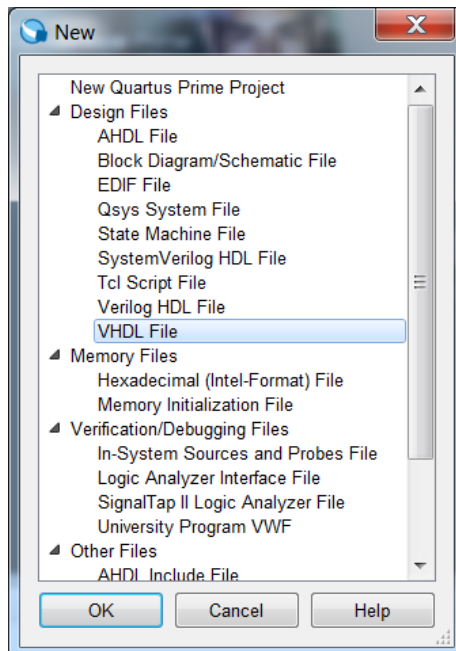
Dar click en Finish



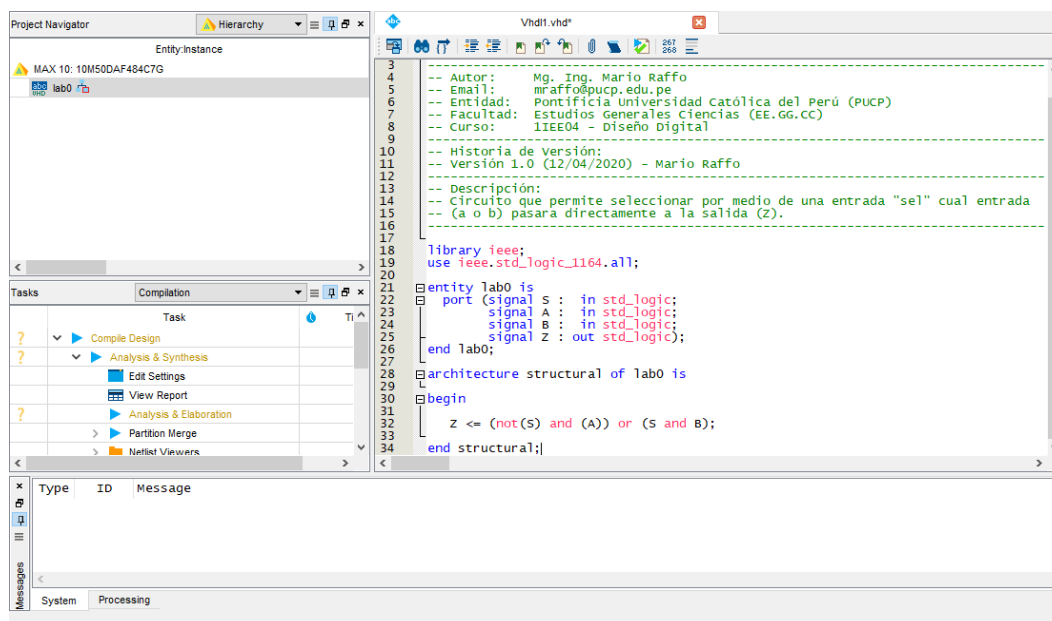
Ahora se ha creado el proyecto denominado **lab0**.

CREAR UN NUEVO ARCHIVO PARA EL CIRCUITO DE TRABAJO

Ahora ya con el proyecto creado, deberá dar click en File -> New y se abrirá la ventana mostrada, donde elegirá VHDL File.



A continuación, escribirá la descripción VHDL en la ventana de texto, tal como se muestra a seguir. Favor, adicione el respectivo encabezado.



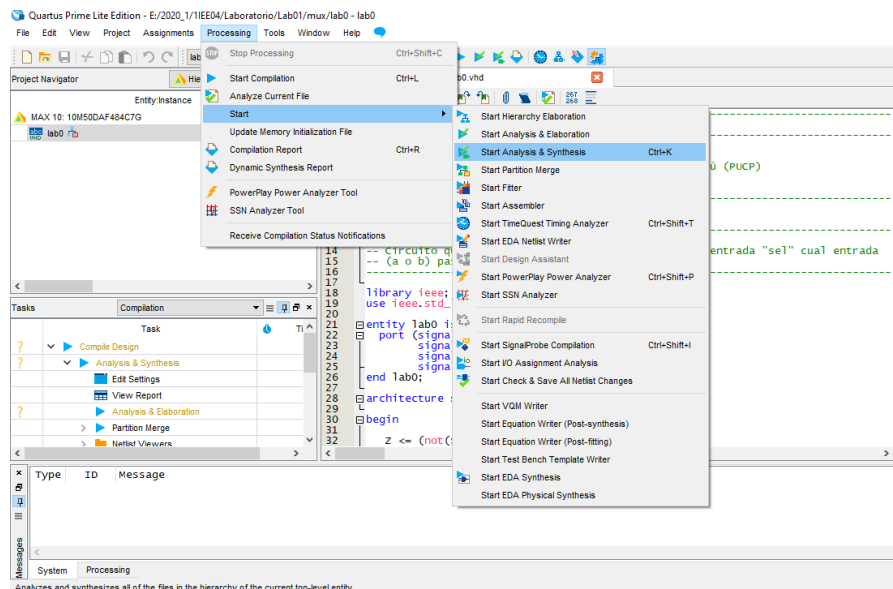
Luego de escribir deberá guardar el archivo haciendo clic en File -> Save.

Acepte el nombre propuesto **lab0** haciendo clic en Save.

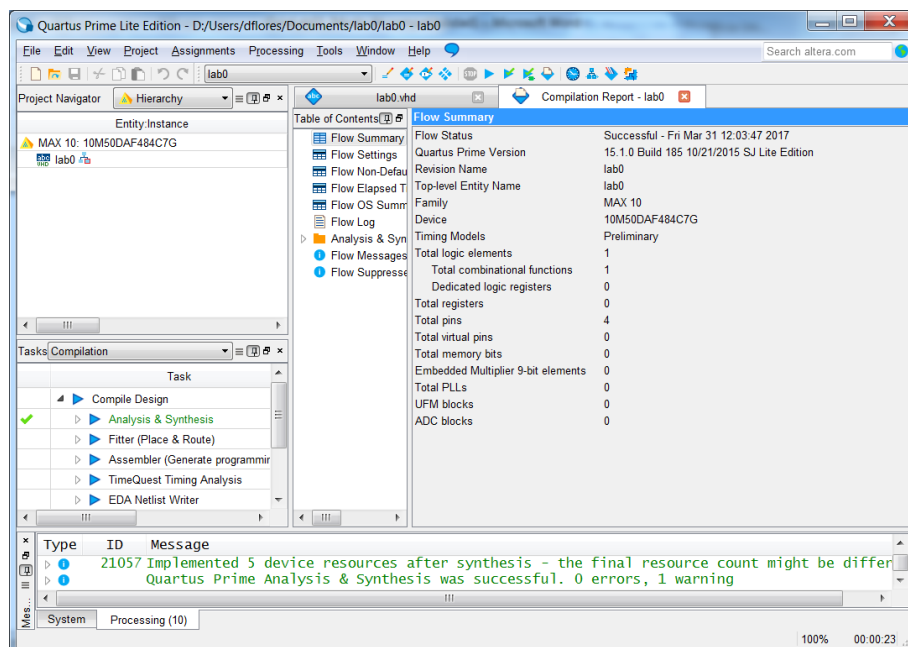
Notar que el nombre del archivo **lab0** coincide con el nombre de la entidad (entity).

COMPILAR EL ARCHIVO DEL CIRCUITO DE TRABAJO

A continuación, se procederá a realizar la síntesis lógica del diseño, para lo cual debe hacer clic en Processing -> Start -> Start Analysis and Synthesis.

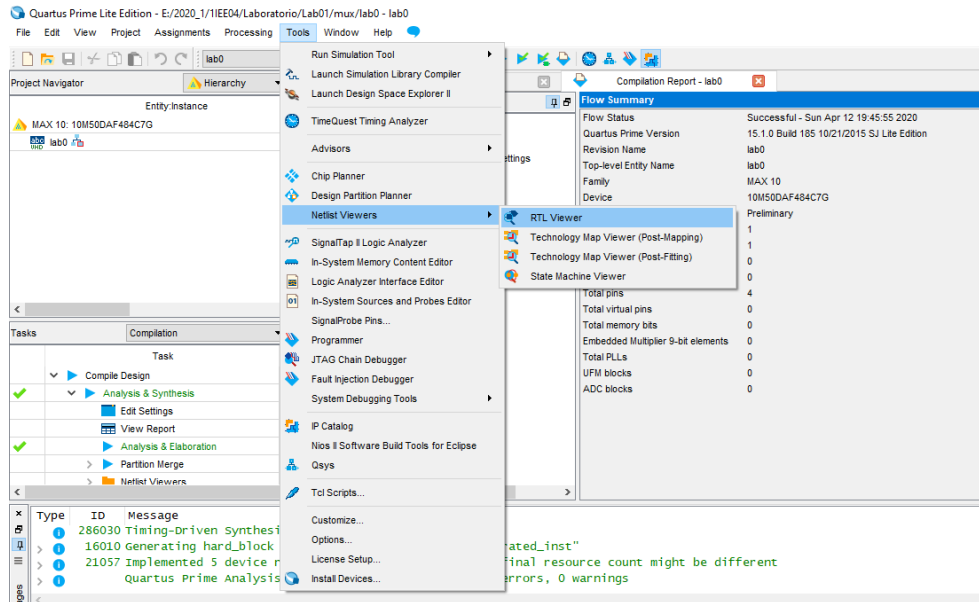


El programa compilará el proyecto, si encuentra errores deberá revisarlos y corregirlos. Si fue exitosa la compilación, aparecerá el siguiente mensaje:

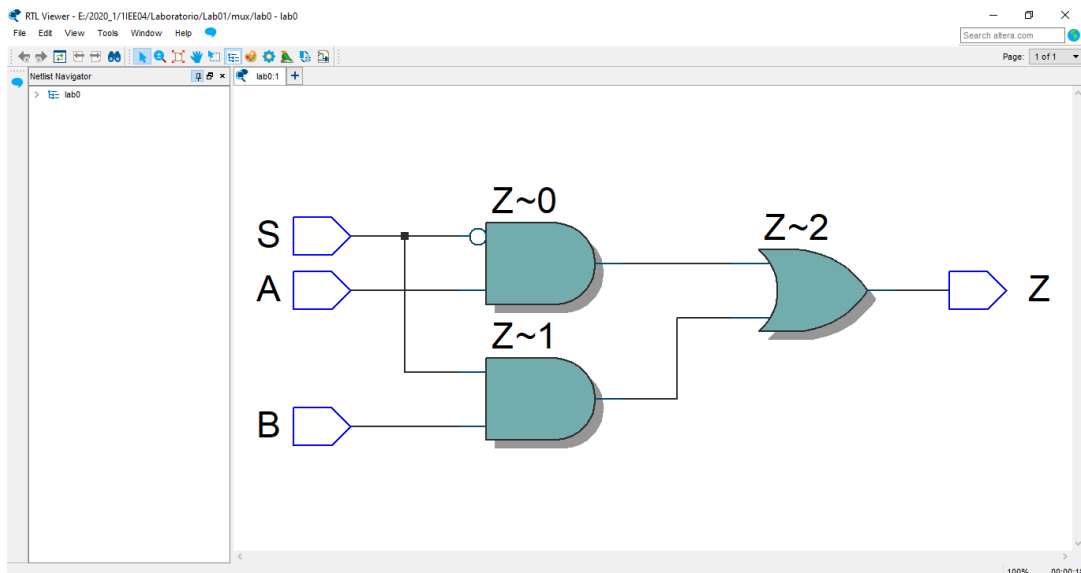


En el caso de haber error, el mensaje le indicará la línea, donde está el error. Es bueno aclarar que dependiendo del tipo de error, hay ocasiones que indica la línea aproximada, siendo muchas veces la línea superior.

Es siempre conveniente ejecutar Tools → Netlist Viewer → RTL Viewer, para poder ver la relación de la salida respecto de las entradas del circuito. Por lo general, permite ver si a nivel VHDL no se ha usado una entrada definida.

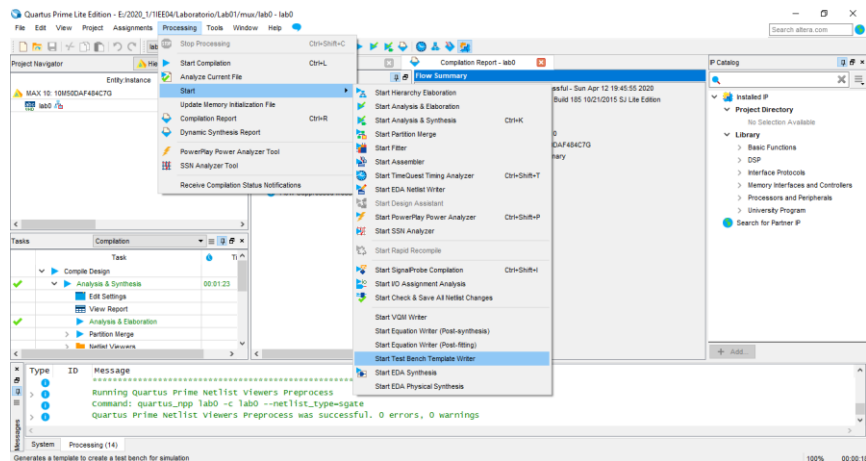


El RTL Viewer no permite ver que una salida no dependa de una entrada, pero si permitirá ver si una entrada no ha sido usada, al realizar la descripción en VHDL.



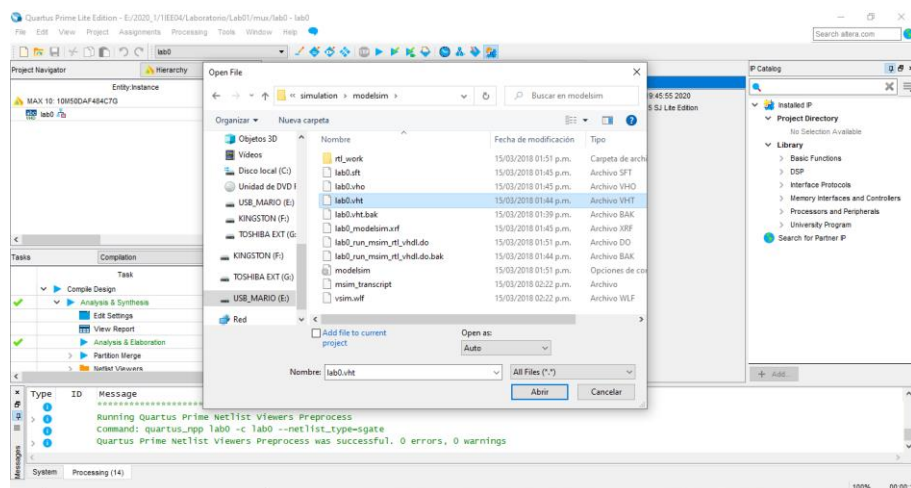
SIMULAR EL ARCHIVO DE TRABAJO (Introducción al TestBench y ModelSim-Altera)

La generación del testbench se efectúa en:
Processing -> Start -> Start Test Bench Template Writer.

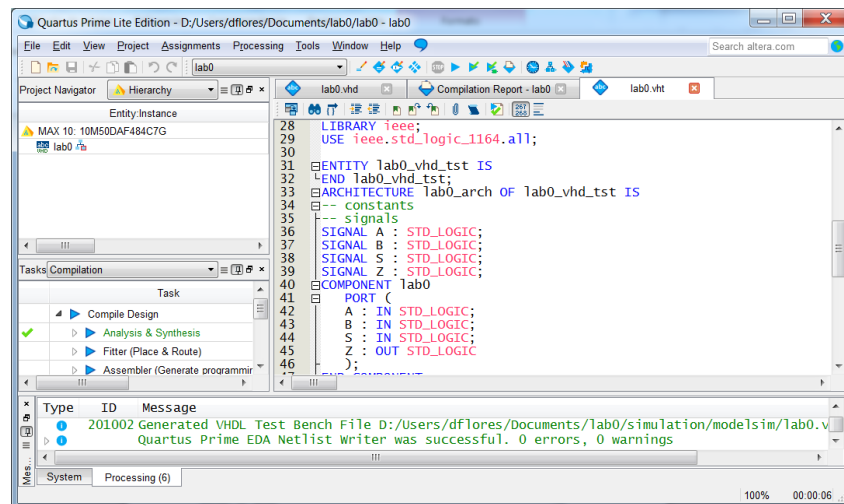


El programa Quartus II indica, en la ventana de mensajes, que generó el archivo de testbench bajo el nombre lab0.vht, en la siguiente ruta, dentro del directorio del proyecto. /simulation/modelsim/

Se procederá a abrir el archivo ex1.vht, por medio de File -> Open
En la ruta no se verá el archivo de extensión vht, a menos que se seleccione en la opción encima de “abrir” la opción “All Files”.



El archivo .vht deberá ser editado para cumplir con los requerimientos de simulación según sea necesario.



Debe ubicar la línea que dice: always: PROCESS, unas pocas líneas más abajo estará escrito BEGIN. Debajo de BEGIN Tendrá que adicionar los estímulos al circuito lab0 que será verificado.

Esos estímulos corresponderán a los casos de la tabla de verdad y se les dará un tiempo de 10 ns para poder visualizar las 8 respuestas del circuito. Es decir la simulación corresponderá a 80 ns.

Recuerde que la tabla de verdad del multiplexor 2a1 es:

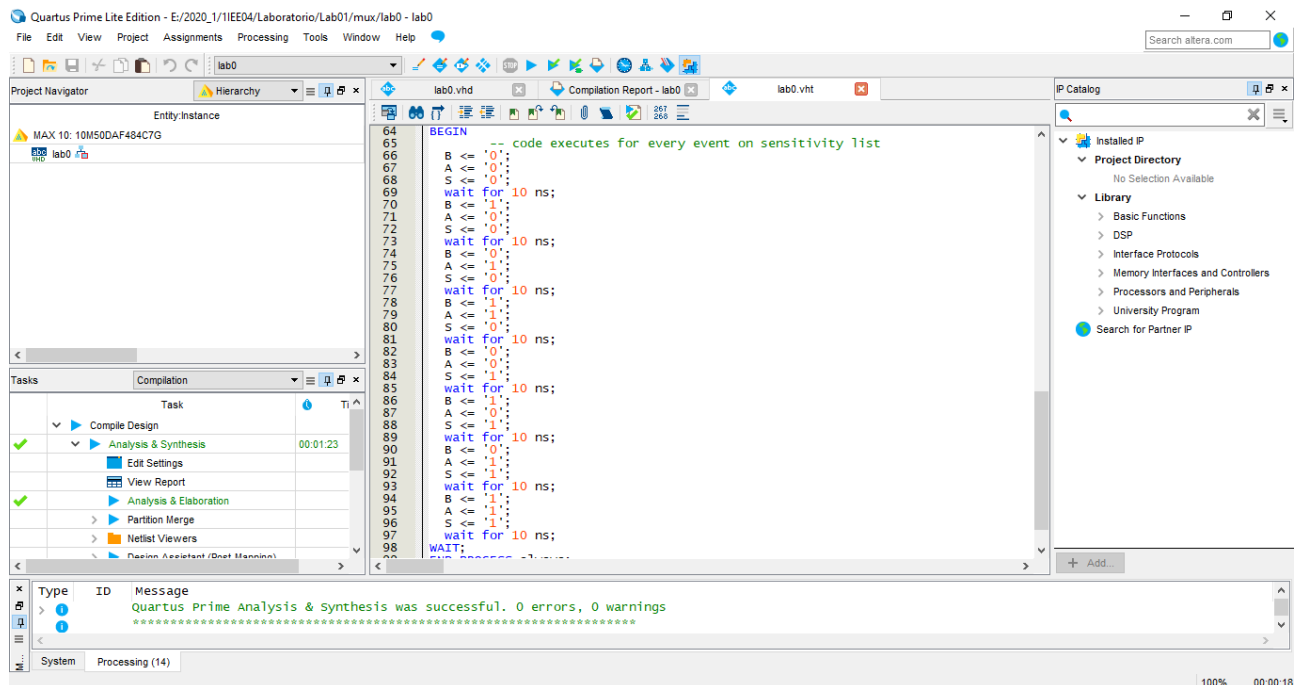
N° fila	S	A	B	Z
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Ejemplo de los estímulos

```
B <= '0';  
A <= '0';  
S <= '0';  
wait for 10 ns;
```

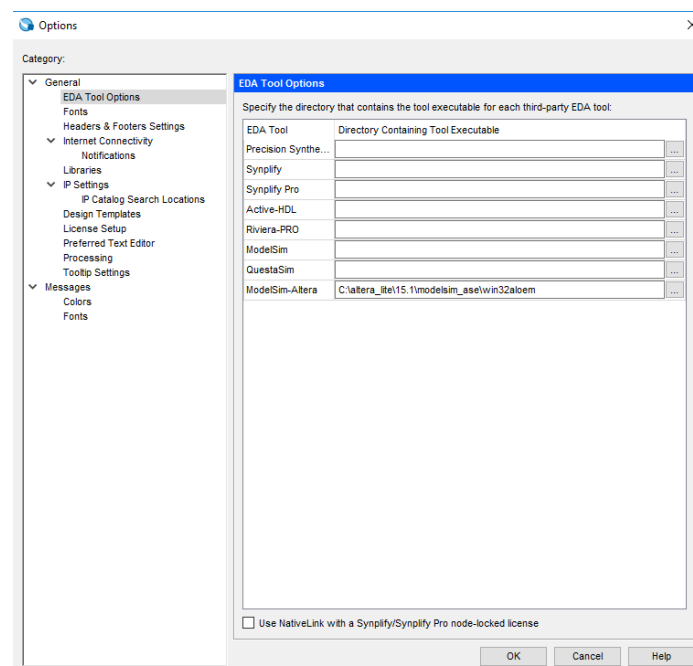
```
B <= '1';  
A <= '0';  
S <= '0';  
wait for 10 ns;
```


En la sección always : PROCESS se deben agregar unas líneas en donde se generan los diferentes estímulos de entrada que se desean aplicar al módulo diseñado.

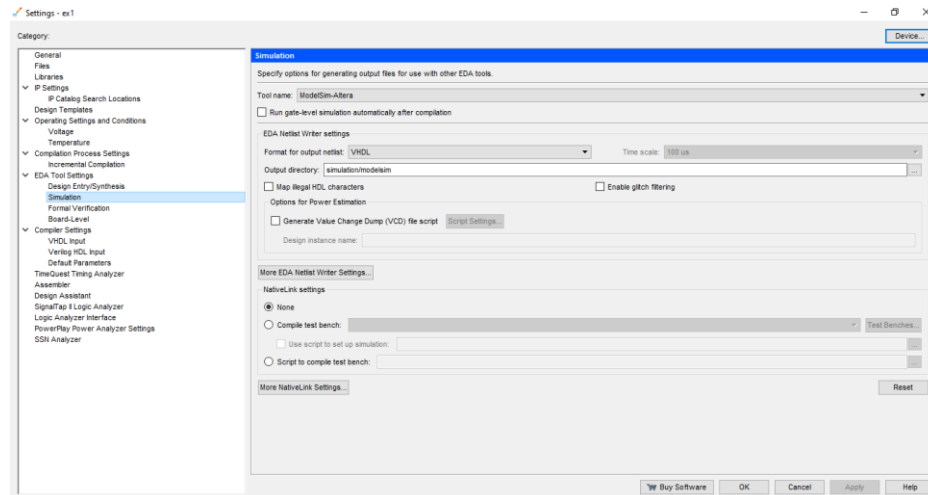


En este punto el testbench ya está listo.

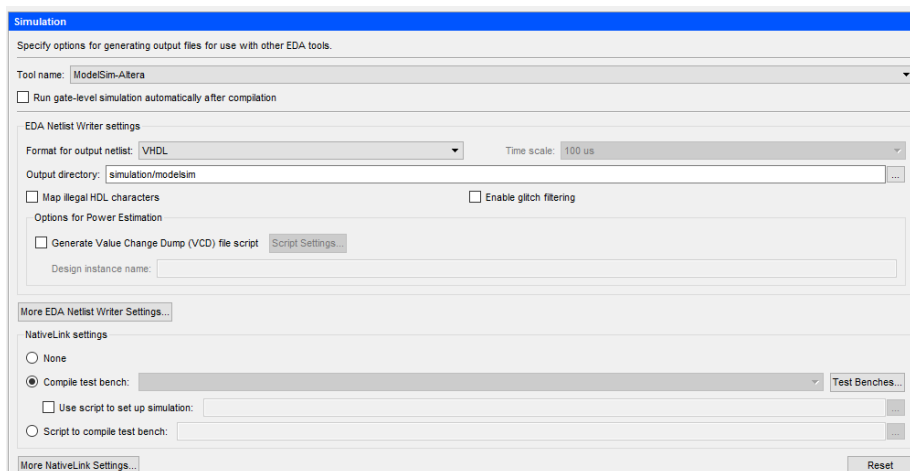
Previo a la simulación del archivo, se debe verificar que el software esté vinculado a la ruta donde se encuentra el simulador. Se deberá hacer clic en Tools -> Options -> EDA Tools Options. Se debe verificar que la ruta del ítem ModelSim-Altera sea C:\altera_lite\15.1\modelsim_ase\win32aloem



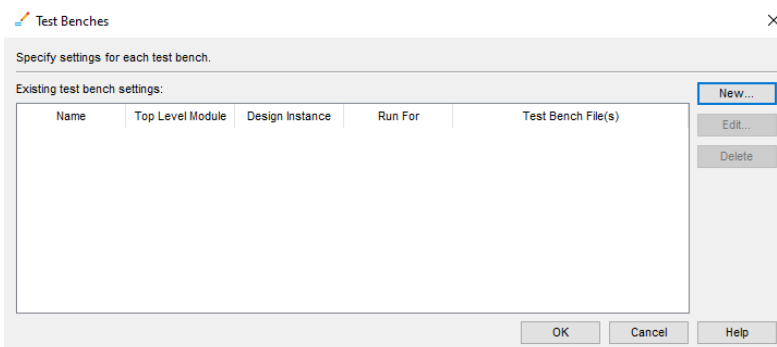
Si bien el testbench ya esta listo, hay que configurar al programa Quartus II para que interaccione con la herramienta ModelSIM Altera. Para ello debe dar click en Assignments -> Settings. Luego en la ventana que se abre, dentro de Category, en EDA Tool Settings, debe seleccionar Simulation



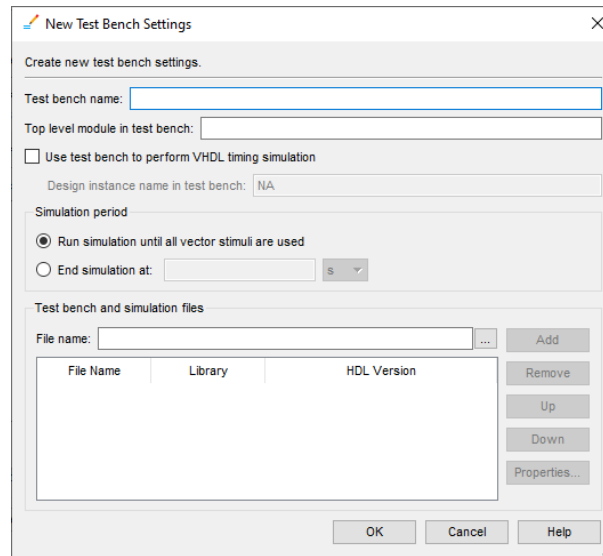
En la parte de simulation debe seleccionar Compile test bench y debe dar click en Test Benches



En la nueva ventana, dar click en New



Se obtendrá la siguiente ventana



New Test Bench Settings

Create new test bench settings.

Test bench name:

Top level module in test bench:

☐ Use test bench to perform VHDL timing simulation

Design instance name in test bench:

Simulation period

☒ Run simulation until all vector stimuli are used

☐ End simulation at: s

Test bench and simulation files

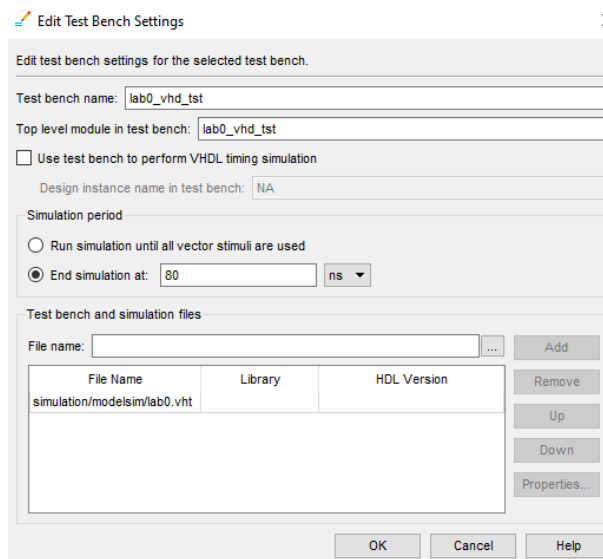
File name: ...

File Name	Library	HDL Version

En la nueva ventana, llenar lo siguiente:

- Test bench name: lab0_vhd_tst (autocopiara en Top level module in test bench)
- Seleccionar End Simulation y colocar 80 ns
- En file name debe seleccionar el archivo lab0_vht (danto click en e lboton que está al lado izquierdo de Add), para luego añadirlo.

Debe haber obtenido lo mostrado en la figura, luego de comprobarlo deberá dar click en ok



Edit Test Bench Settings

Edit test bench settings for the selected test bench.

Test bench name:

Top level module in test bench:

☐ Use test bench to perform VHDL timing simulation

Design instance name in test bench:

Simulation period

☐ Run simulation until all vector stimuli are used

☒ End simulation at: ns

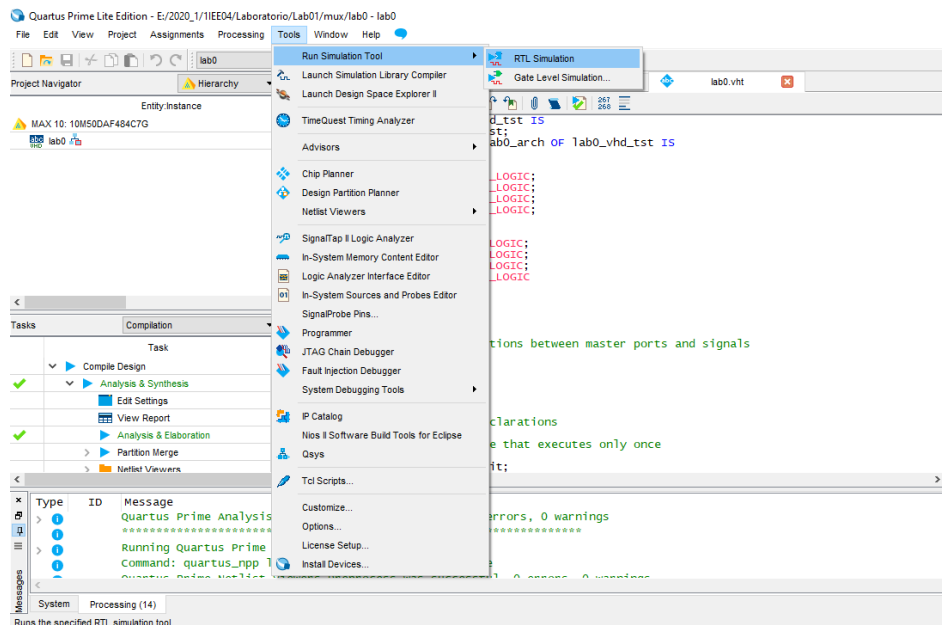
Test bench and simulation files

File name: ...

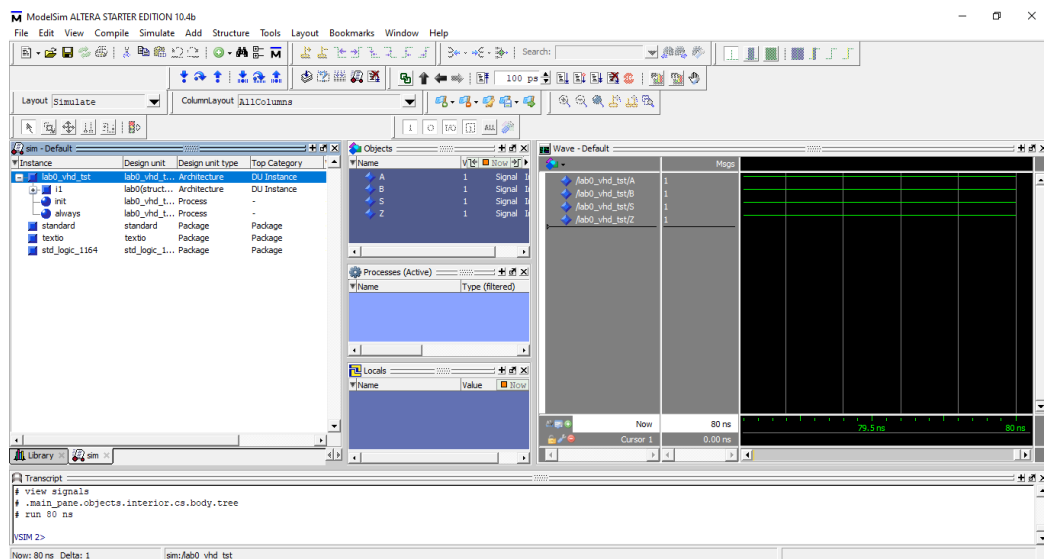
File Name	Library	HDL Version
simulation/modelsim/lab0.vht		

Una vez seleccionado, dar OK a todas las opciones.

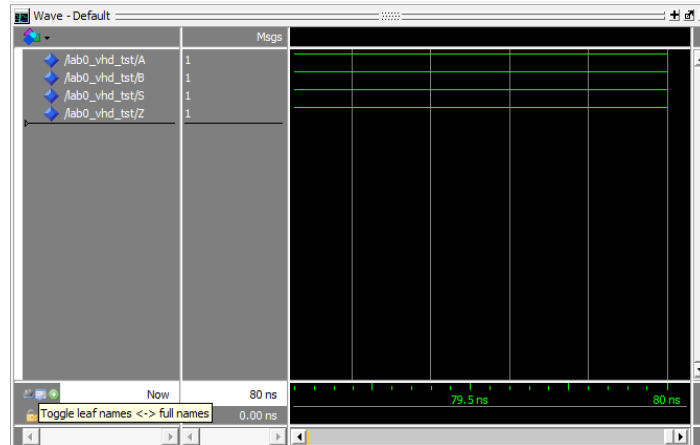
Para que funcione el ModelSIM Altera debe dar click en Tools → Run Simulation Tool → RTL Simulation



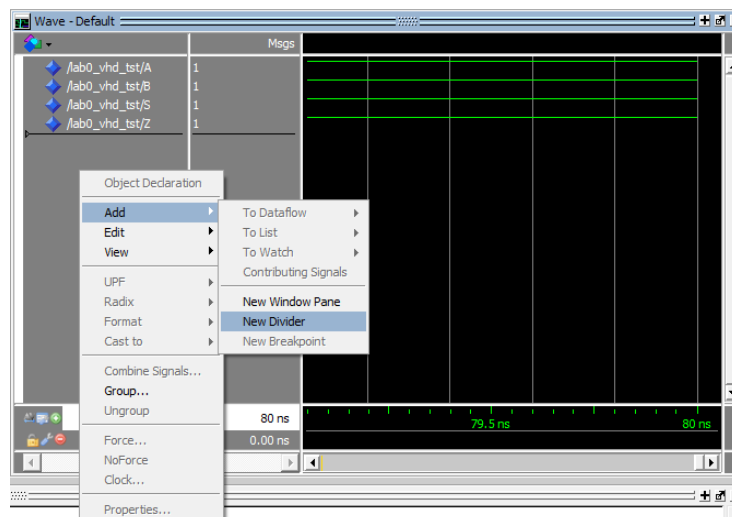
De no haber habido errores en el testbench, el ModelSIM Altera se vera como muestra la figura a continuación.



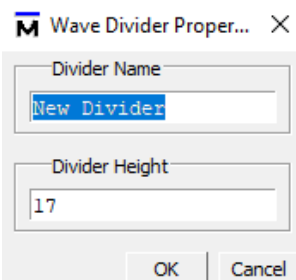
En la parte inferior de donde dice Wave – Default, encima del símbolo del candado hay un símbolo oscuro de nombre “Toggle leaf names”. Debe dar click en ese símbolo



En Wave – Default, debajo de los nombres de las entradas y salidas, Debe dar click derecho y seleccionar Add → New Divider

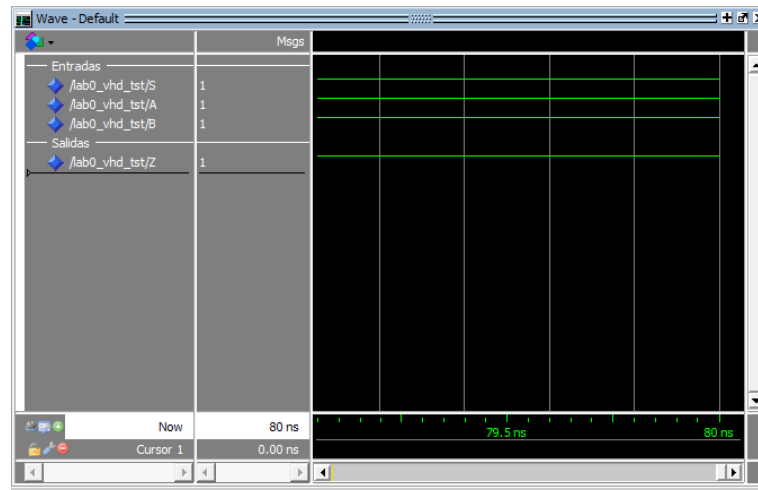


En la nueva ventana deber poner de nombre en Divider Name, el nombre Entradas y click en ok

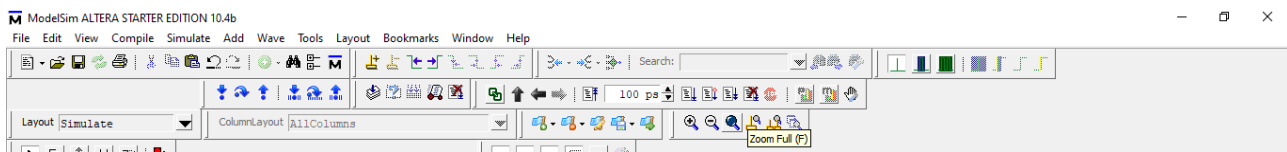


Repita el paso y genere un Divider de nombre Salidas.

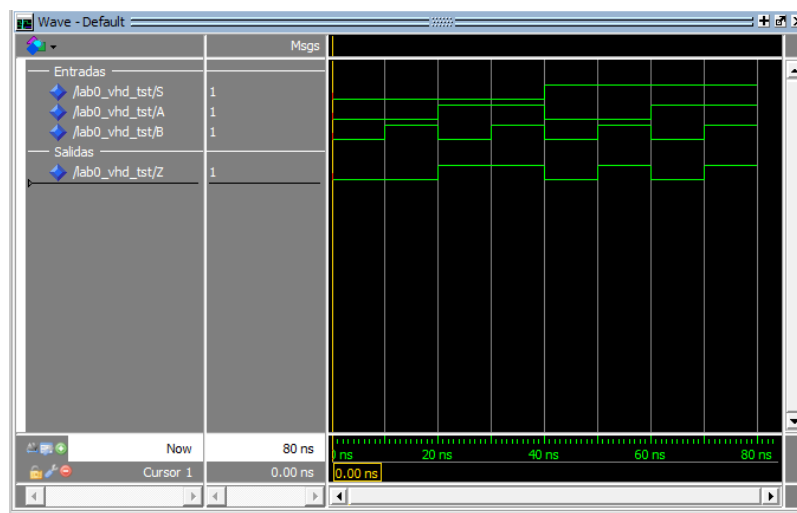
En wave seleccione el divider Entradas (mantenga apretado el click izq) y arrástrelo sobre las señales s,a,b, repita lo mismo para el divider Salida, pero arrástrelo sólo sobre la señal z. Al final debe obtener la vista mostrada en la figura, a continuación.



Seleccione la lupa oscura y podrá ver en wave todo el intervalo de la simulación



La simulación se presenta a continuación



Se ve que la simulación es consecuente con la tabla de verdad. También es necesario hacer la siguiente comparación.

Dado el comportamiento del circuito:

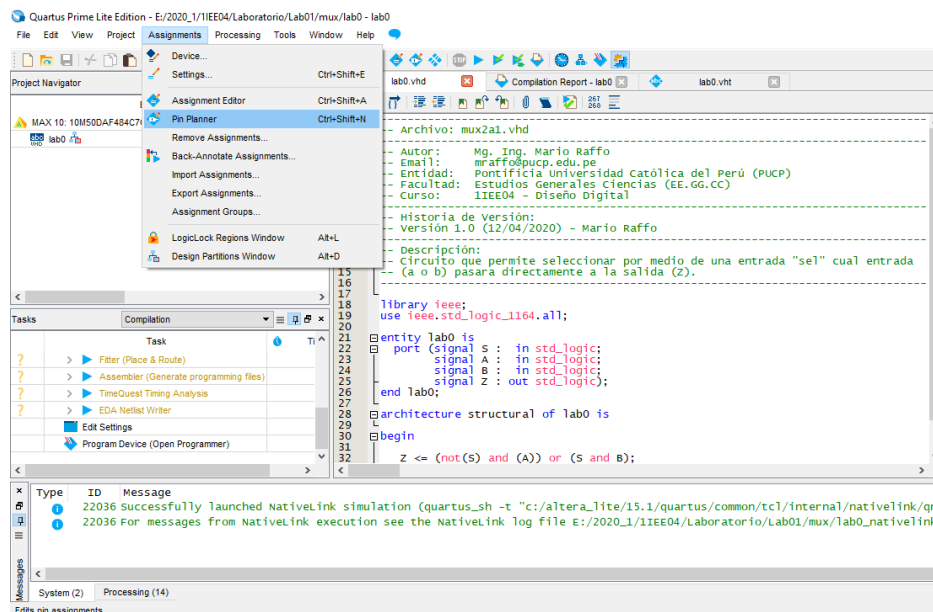
$S = 0$, entonces $Z = A$;

$S = 1$, entonces $Z = B$;

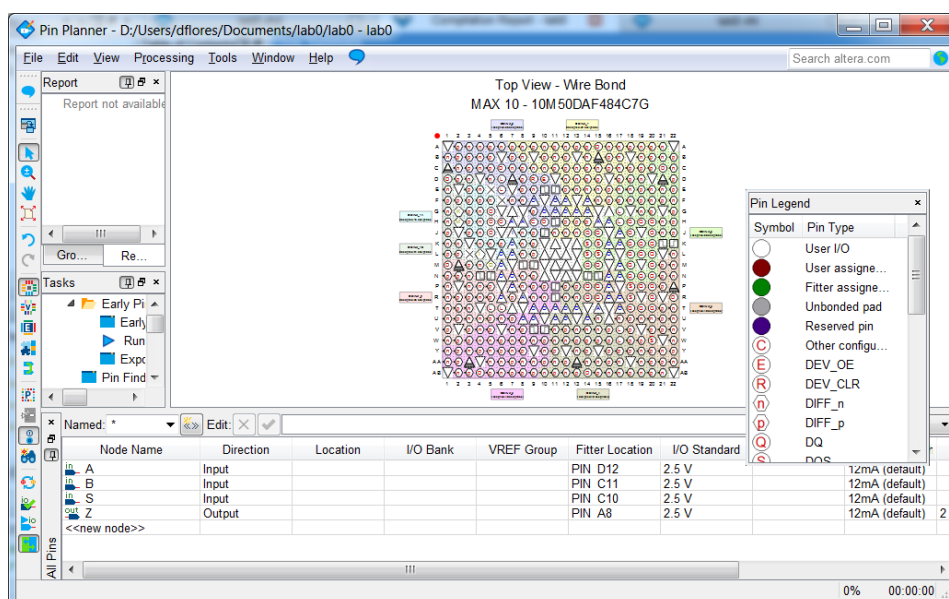
CONFIGURACIÓN DEL FPGA (Asignación de Pines y Programación del FPGA)

Se usará la tarjeta DE10-Lite que dispone del dispositivo 10M50DAF484C7G, cuyo manual deberá revisar. (En el caso de los que usen el Quartus 13.01 deberá usar la tarjeta DE2-115, la cual dispone del dispositivo EP4CE115F29C7, cuyo manual deberá revisar)

En primer lugar, se asignan los pines físicos del dispositivo que será empleado. Para asignar pines, hacer clic en: Assignments -> Pin Planner.



Se deberá mostrar una ventana similar a la que se muestra a continuación.



Se nota que las entradas y salidas del circuito declaradas en la descripción VHDL aparecen en una tabla en la parte inferior de la ventana. Dicha tabla tendrá varios campos, de los cuales son de interés los siguientes:

Node name: nombre de la entrada/salida. En este campo deben aparecer todos los nombres de las entradas y salidas declaradas en la descripción.

Direction: función del nodo. Se indica si es entrada o salida al circuito.

Location: pin asignado a la entrada/salida. Al hacer clic en una celda correspondiente a este campo, se abrirá una lista desplegable de todos los pines disponibles del FPGA, de la cual se puede escoger el pin que se quiere asignar a la entrada o salida del circuito.

Para asignar los pines deberá hacer lo siguiente:

- Para las entradas S, A y B deberá usar interruptores (switches)
- Para la salida Z, deberá usar un led.

Para conseguir hacer esto, usted deberá revisar el manual de la DE10-Lite o e de la DE2-115 (según la versión del Quartus II que esté usando):

- Para la DE10-Lite, revisar las tablas 3.4 y 3.5 (interruptores y leds) correspondientes a las páginas 26 y 27.
- Para la DE115, revisar las tablas 4.1 y 4.3 (interruptores y leds) correspondientes a la página 35.

A continuación es mostrada la asignación de pines para los que usan la DE10-Lite:

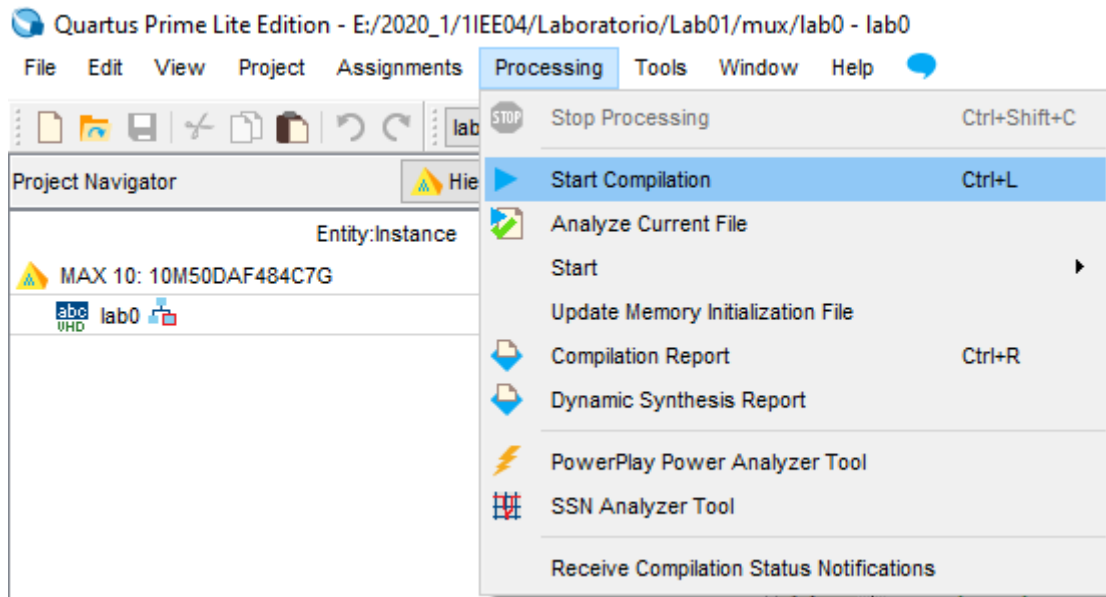
Top View - Wire Bond
MAX 10 - 10M50DAF484C7G

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	R _{pin}
in_A	Input	PIN_C10	7	B7_N0	2.5 V (default)	
in_B	Input	PIN_C11	7	B7_N0	2.5 V (default)	
in_S	Input	PIN_F15	7	B7_N0	2.5 V (default)	
out_Z	Output	PIN_B11	7	B7_N0	2.5 V (default)	

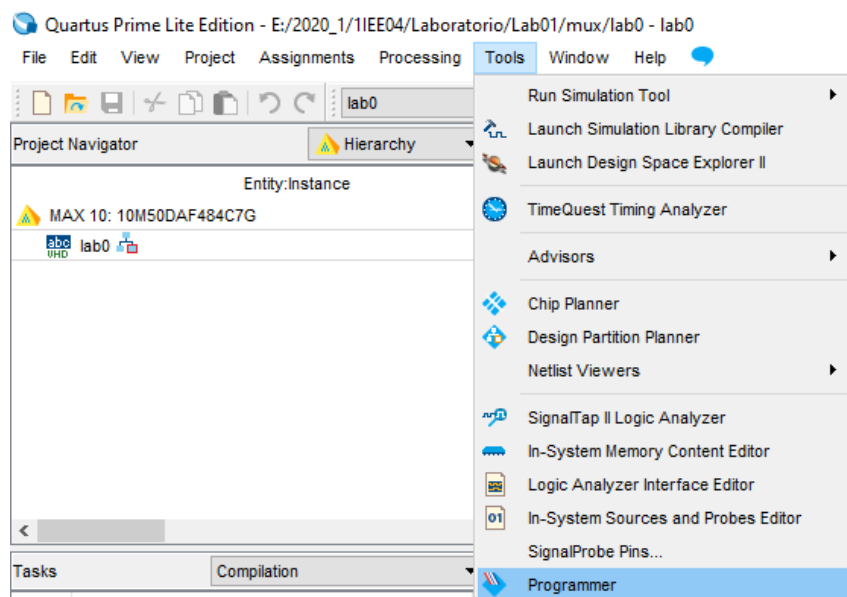
Pin Legend

Symbol	Pin Type
○	User I/O
●	User assigned I/O
○	Fitter assigned I/O
○	Unbonded pad
○	Reserved pin
○	Other configuration
○	DEV_OE
○	DEV_CLR
○	DIFF_n
○	DIFF_p
○	DQ
○	DQS
○	DQSB
○	CLK_n

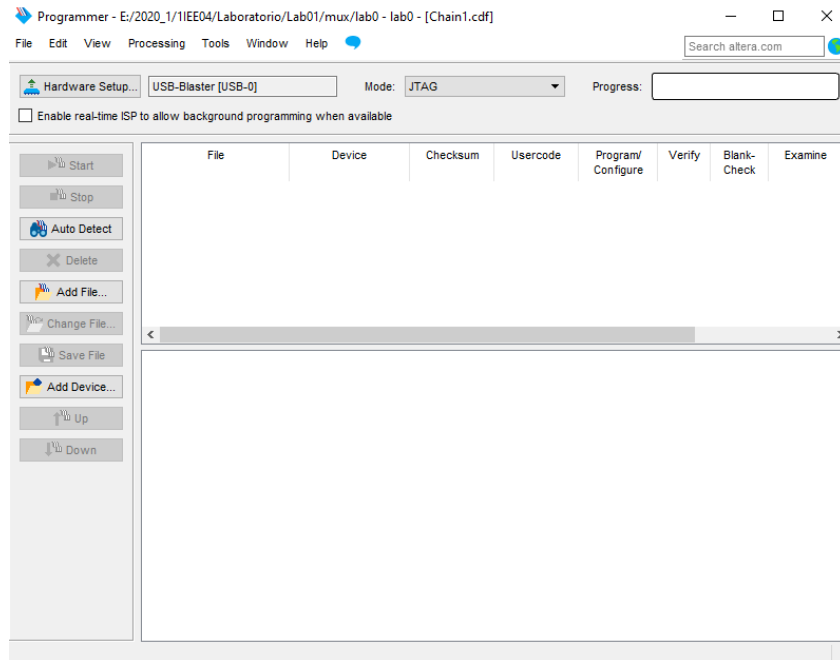
Una vez asignados los pines, deberá realizarse el proceso de compilación completo. Para ello, hacer clic en Processing -> Start Compilation.



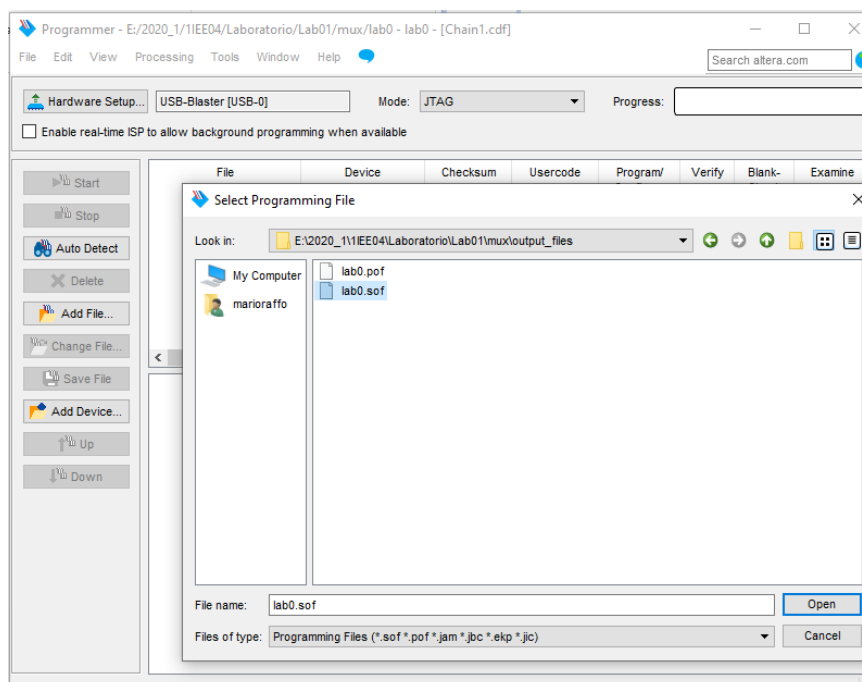
Finalmente, se pasará a configurar el FPGA. Se le solicita conectar la tarjeta DE10-Lite / DE115 mediante el cable USB a la computadora. Luego debe hacer clic en Tools -> Programmer.



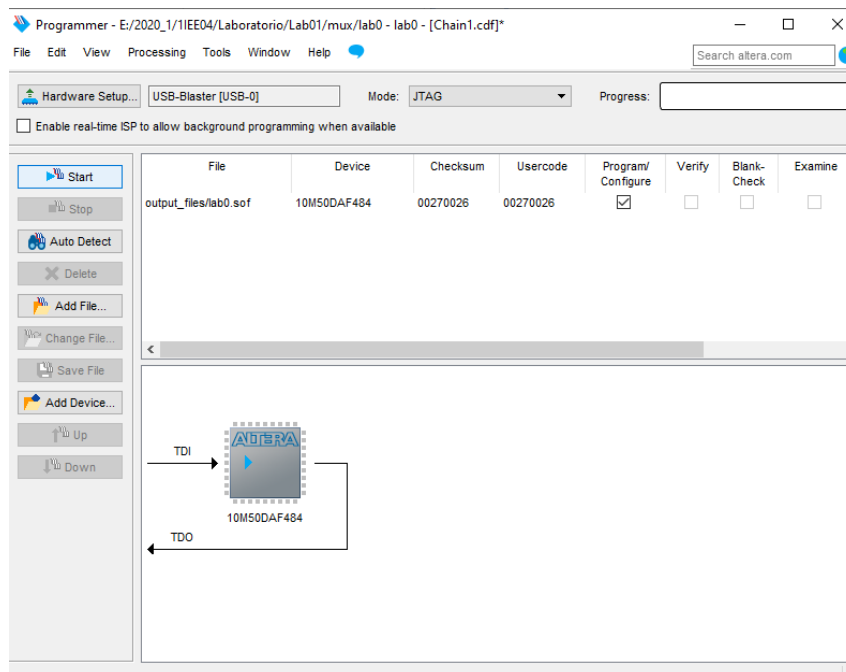
Verificar que la computadora haya detectado el dispositivo (USB-Blaster) tal como se muestra. Caso contrario dar click en Hardware Setup y buscar y seleccionar el USB Blaster¹



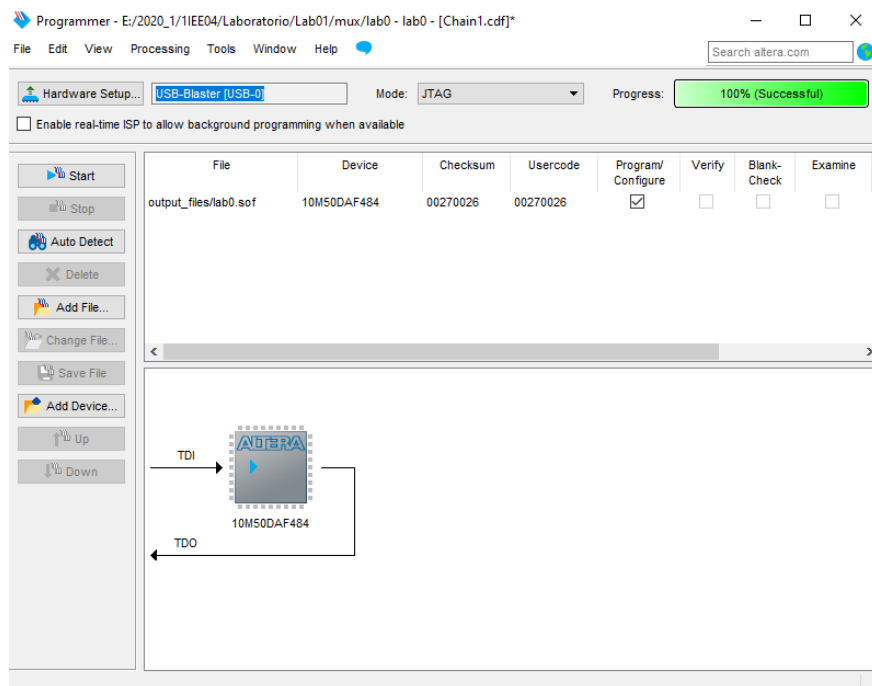
Hacer clic en Add file, seleccionar en la carpeta output_files el archivo lab0.sof y hacer clic en Open



¹ Si esto no resulta es que no se configuró el USB Blaster en la Laptop/PC, para ello vea el manual: Configuración_DE10_Lite_Laptop (independientemente de tener la DE10-Lite o la DE2-115, este manual permitirá configurar el USB Blaster)



Hacer click en Start

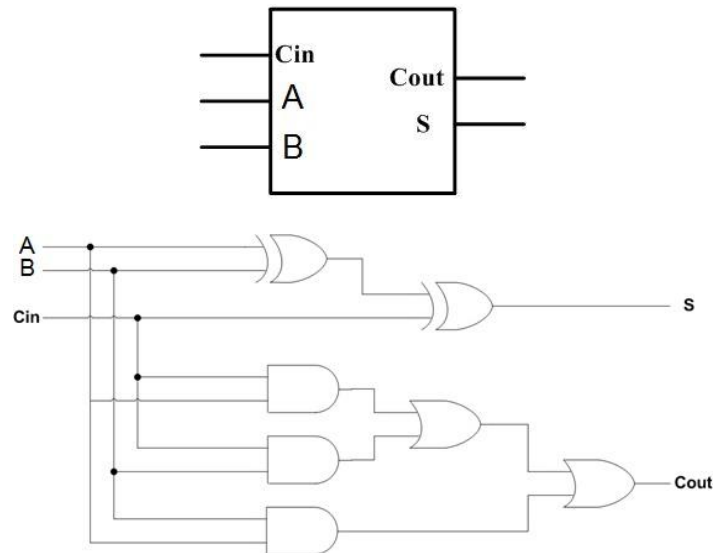


Luego de concluida la configuración, el dispositivo estará implementado con el circuito descrito.

A continuación, deberá verificar la funcionalidad física del circuito en la tarjeta DE10-Lite manipulando los interruptores y observando el LED correspondiente. Comprobar que se cumple la tabla de verdad.

Ejercicio 2 (15 puntos) Ruta de trabajo (c:/temp/1IEE04/h44xy/Lab01/ex2), donde h44xy se reemplaza por el horario al que pertenece.

Dado el circuito fulladder ilustrado por la figura a continuación. Considere que las entradas del circuito son Cin, A, B, mientras que las salidas son Cout, S



N°	entradas			Acarreo	Suma
Filas	Cin	A	B	Cout	S
0	0	0	0		
1	0	0	1		
2	0	1	0		
3	0	1	1		
4	1	0	0		
5	1	0	1		
6	1	1	0		
7	1	1	1		

Se solicita que desarrolle lo siguiente (Experiencia Práctica Parcial (EPP)):

- La expresión lógica booleana. (1.5 puntos)
- La tabla de verdad. (1.5 puntos)
- Proyecto con la descripción del circuito fulladder empleando VHDL (fulladder.vhd) (6 puntos)
- La simulación del circuito fulladder por medio de un Ambiente de Verificación (Testbench) y analice la gráfica generada con la tabla de verdad del mismo (5 puntos).
- Luego proceda a asignar pines para implementar el circuito en la tarjeta DE10-Lite / DE115 según sea el caso (1 punto).

Usted deberá sustentar su desarrollo ante el JP que le haya sido asignado (se le efectuarán preguntas), para tener el puntaje de la Experiencia Práctica (EP). El JP asignará un porcentaje (que puede ser del 0 al 100%) y la nota será:

$$EP = EPP * Porcentaje$$

La nota de la presente sesión de laboratorio será:

$$Pb01 = EP + PS$$

Donde Pb será llevada al entero superior si su parte fraccionaria es mayor o igual a 0.5, caso contrario la parte fraccionaria se volverá 0.