

**Instituto Tecnológico y De Estudios Superiores de
Monterrey**
Laboratorio - Arquitectura de Computadoras TE-2031.1



Lab 04 - Making Assignments

Entrega: 07 / 10 / 2022

Equipo 07:

Héctor Javier Pequeño Chairez A01246364

Gabriela Jazmín Álvarez Espinoza A00825719

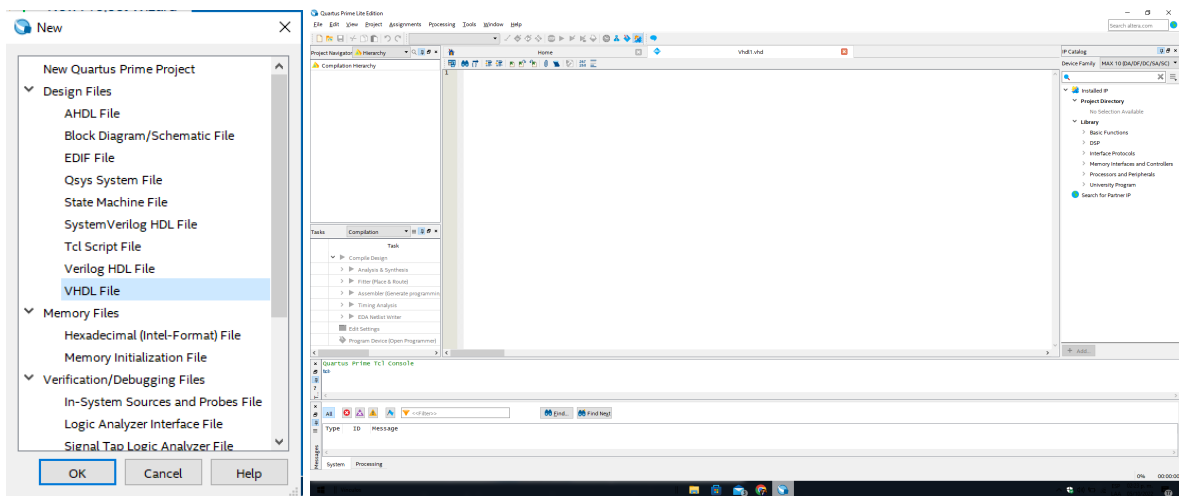
Introducción

En la siguiente práctica nos enfocaremos en sintetizar un código escrito en VHDL para ser enviado hacia nuestro FPGA, esto con el objetivo de observar y probar el funcionamiento de la aplicación Quartus Prime y testear en un objeto físico la síntesis de nuestro programa desarrollado en VHDL.

Desarrollo

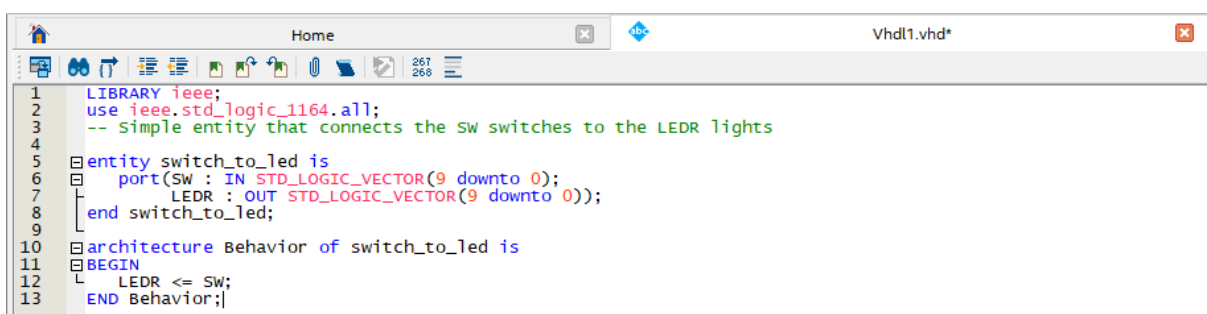
Crear un archivo:

En File se seleccionó la creación de un archivo de tipo VHDL



Archivo tipo VHDL creado.

Añadimos el código presentado en el manual:



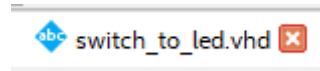
Código insertado en la ventana del archivo tipo VHDL.

BRAIN EXERCISE: Check your syntax carefully! Can you explain what this circuit does?

El código anterior define una entidad que tiene como entrada el switch SW usando 10 bits del 0 al 9 (Los 10 switches del FPGA), además define como salida el LEDR donde de

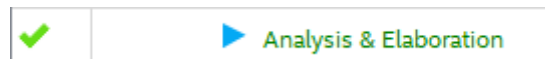
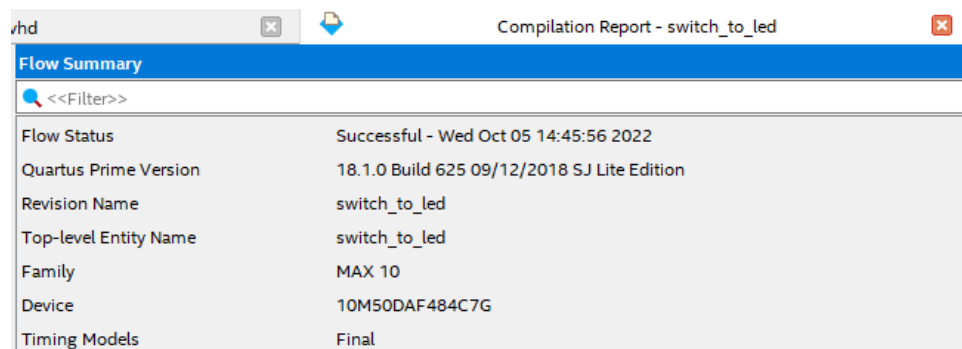
igual manera tenemos 10 bits del 0 al 9 (los 10 leds rojos del FPGA), en si lo que ejecuta el programa es que el LEDR o los 10 leds tomará el valor que tenga el SW o el switch.

Una vez escrito el código, guardamos el archivo con el mismo nombre de la “entity”.



Nuestro programa con el nombre correspondiente a la entidad.

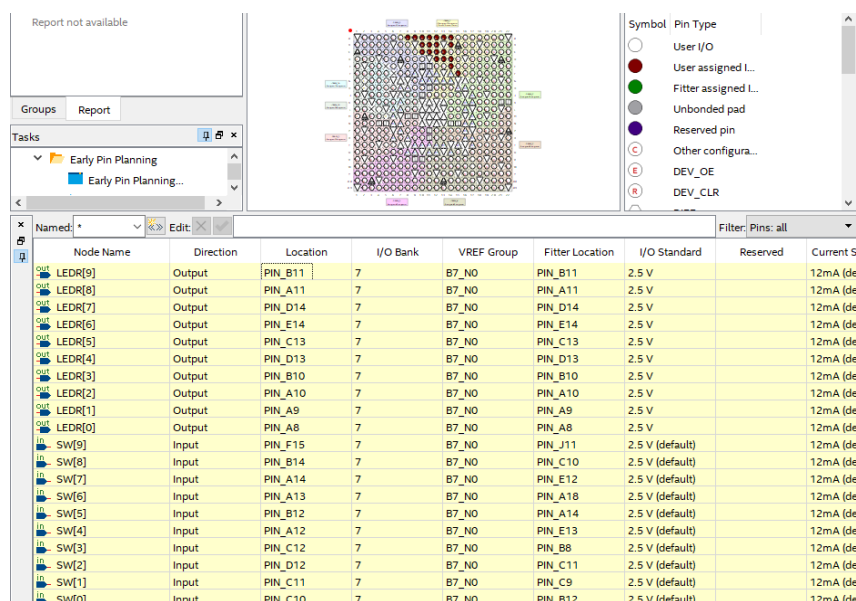
Ejecutamos Analysis and Elaboration:



Analysis and Elaboration ejecutados correctamente.

Asignación de pines por Pin Planner

Debemos ingresar al “Pin planner” en donde indicaremos a los leds, la locación de los pines que están conectados a ellos, lo que nos deja con el siguiente mapa:



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current S
LEDR[9]	Output	PIN_B11	7	B7_NO	PIN_B11	2.5 V		12mA (de
LEDR[8]	Output	PIN_A11	7	B7_NO	PIN_A11	2.5 V		12mA (de
LEDR[7]	Output	PIN_D14	7	B7_NO	PIN_D14	2.5 V		12mA (de
LEDR[6]	Output	PIN_E14	7	B7_NO	PIN_E14	2.5 V		12mA (de
LEDR[5]	Output	PIN_C13	7	B7_NO	PIN_C13	2.5 V		12mA (de
LEDR[4]	Output	PIN_D13	7	B7_NO	PIN_D13	2.5 V		12mA (de
LEDR[3]	Output	PIN_B10	7	B7_NO	PIN_B10	2.5 V		12mA (de
LEDR[2]	Output	PIN_A10	7	B7_NO	PIN_A10	2.5 V		12mA (de
LEDR[1]	Output	PIN_A9	7	B7_NO	PIN_A9	2.5 V		12mA (de
LEDR[0]	Output	PIN_A8	7	B7_NO	PIN_A8	2.5 V		12mA (de
SW[9]	Input	PIN_F15	7	B7_NO	PIN_J11	2.5 V (default)		12mA (de
SW[8]	Input	PIN_B14	7	B7_NO	PIN_C10	2.5 V (default)		12mA (de
SW[7]	Input	PIN_A14	7	B7_NO	PIN_E12	2.5 V (default)		12mA (de
SW[6]	Input	PIN_A13	7	B7_NO	PIN_A18	2.5 V (default)		12mA (de
SW[5]	Input	PIN_B12	7	B7_NO	PIN_A14	2.5 V (default)		12mA (de
SW[4]	Input	PIN_A12	7	B7_NO	PIN_E13	2.5 V (default)		12mA (de
SW[3]	Input	PIN_C12	7	B7_NO	PIN_B8	2.5 V (default)		12mA (de
SW[2]	Input	PIN_D12	7	B7_NO	PIN_C11	2.5 V (default)		12mA (de
SW[1]	Input	PIN_C11	7	B7_NO	PIN_C9	2.5 V (default)		12mA (de
SW[0]	Input	PIN_C10	7	B7_NO	PIN_B12	2.5 V (default)		12mA (de

Pines asignados a LEDs y Switches.

Asignación de pines por consola

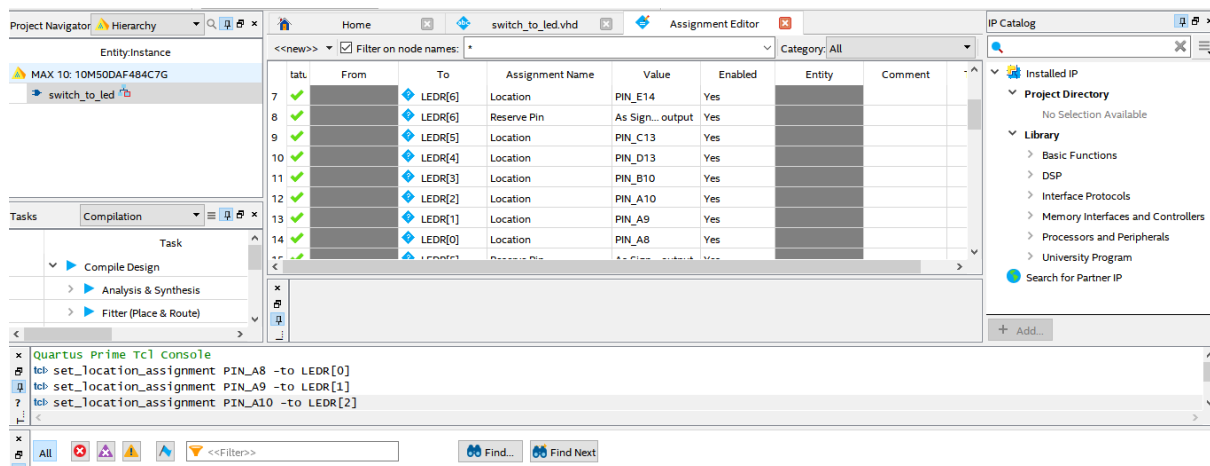
Una opción que tenemos es también asignar nuestros inputs y outputs mediante consola, esto asignando al pin y nombre que le daremos a ese objeto en específico.

Aquí declaramos acorde al pin que corresponda cada uno de los leds:

```
tcl> set_location_assignment PIN_A8 -to LEDR[0]
tcl> set_location_assignment PIN_A9 -to LEDR[1]
tcl> set_location_assignment PIN_A10 -to LEDR[2]
tcl> set_location_assignment PIN_B10 -to LEDR[3]
tcl> set_location_assignment PIN_A12 -to SW[4]
tcl> set_location_assignment PIN_D13 -to LEDR[4]
tcl> set_location_assignment PIN_C13 -to LEDR[5]
tcl> set_location_assignment PIN_E14 -to LEDR[6]
tcl> set_location_assignment PIN_D14 -to LEDR[7]
tcl> set_location_assignment PIN_A11 -to LEDR[8]
tcl> set_location_assignment PIN_B11 -to LEDR[9]
tcl> set_location_assignment PIN_B11 -to LEDR[9]
tcl> set_location_assignment PIN_B11 -to LEDR[9]
tcl> set_location_assignment PIN_C10 -to SW[0]
tcl> set_location_assignment PIN_C11 -to SW[1]
tcl> set_location_assignment PIN_D12 -to SW[2]
tcl> set_location_assignment PIN_C12 -to SW[3]
tcl> set_location_assignment PIN_B12 -to SW[5]
tcl> set_location_assignment PIN_A13 -to SW[6]
tcl> set_location_assignment PIN_A14 -to SW[7]
tcl> set_location_assignment PIN_B14 -to SW[8]
tcl> set_location_assignment PIN_F15 -to SW[9]
tcl>
```

Aquí declaramos acorde al pin que corresponda cada uno de los switches:

Confirmamos con ayuda de la assignment editor window que sí se hayan asignado acorde al pin que declaramos en los comandos:



Compilación

Para esto necesitamos pulsar el boton de “play” de color azul, el cual ejecutara la compilación y despues de 1 minuto desplegara en una ventana y en consola si la compilación fue correcta.

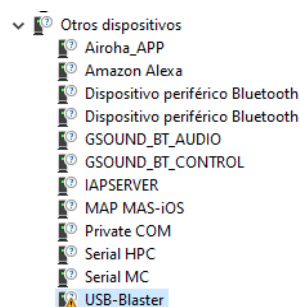
Flow Summary	
<<Filter>>	
Flow Status	Successful - Wed Oct 05 18:09:44 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	switch_to_led
Top-level Entity Name	switch_to_led
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	1 / 49,760 (< 1 %)
Total registers	0
Total pins	20 / 360 (6 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

Type	ID	Message
①	332140	No Setup paths to report
①	332140	No Hold paths to report
①	332140	No Recovery paths to report
①	332140	No Removal paths to report
①	332140	No Minimum Pulse width paths to report
①	332102	Design is not fully constrained for setup requirements
①	332102	Design is not fully constrained for hold requirements
> ①		Quartus Prime Timing Analyzer was successful. 0 errors, 5 warnings
①	293000	Quartus Prime Full Compilation was successful. 0 errors, 14 warnings

Se observa que en el Flow Summary y en la terminal, se compilo correctamente.

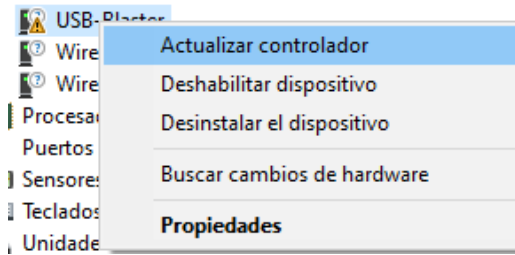
Instalación USB Blaster para el diseño de FPGA

Abrimos el Device Manager de la computadora e identificamos el USB blaster.



USB-Blaster conectado en los dispositivos.

Actualizamos el controlador, dando click derecho y pulsando en “Actualizar controlador”.



Ventana emergente con la opción para actualizar.

Seleccionamos la opción de buscar el controlador en el PC

→ [Examinar mi PC en busca de controladores](#)
[Busca e instala un controlador manualmente.](#)

Opción a elegir.

Y colocamos la ruta hacia nuestro controlador, en este caso es la siguiente:

[Buscar controladores en el equipo](#)


Buscar controladores en esta ubicación:

Examinar...

☒ Incluir subcarpetas

Ruta hacia el controlador.

Esto nos desplegará el siguiente mensaje:

 Actualizar controladores - Altera USB-Blaster

Ya están instalados los mejores controladores para el dispositivo

Windows determinó que el mejor controlador para este dispositivo ya está instalado. Puede haber controladores mejores en Windows Update o en el sitio web del fabricante del dispositivo.

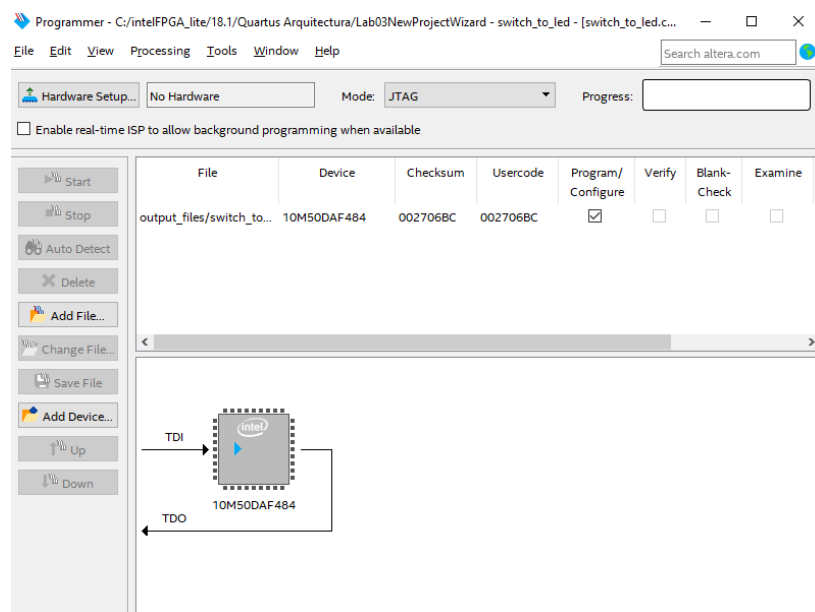


Altera USB-Blaster

Los drivers han sido instalados correctamente.

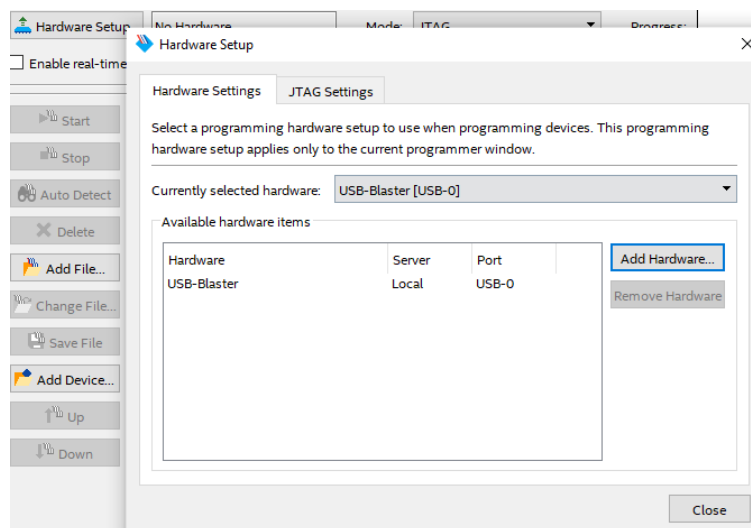
Programación del diseño del FPGA

Ahora cargaremos la imagen hacia nuestro FPGA, por lo que deberemos seleccionar la opción de “Programmer”, en donde nos cargara la siguiente ventana.



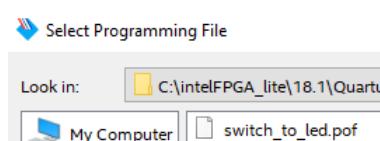
Ventana del programmer en Quartus.

Seleccionamos Hardware setup y seleccionamos USB blaster en la ventana emergente y lo cerramos.



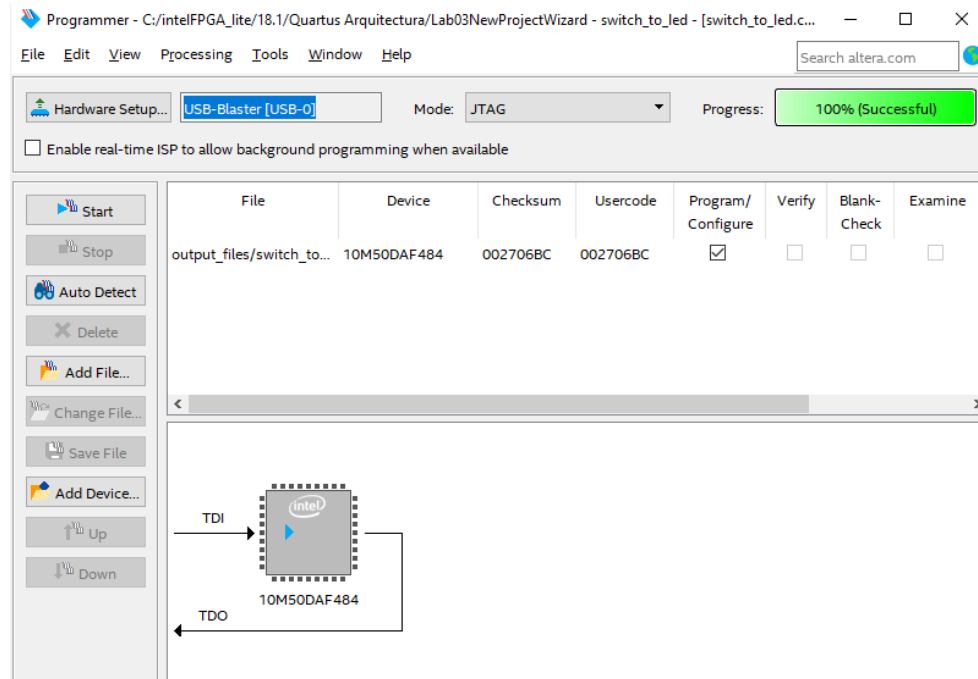
Currently selected Hardware = USB- Blaster.

Seleccionamos Add file y en la carpeta de Output, seleccionamos nuestro objeto.



Archivo.pof en carpeta output.

Ejecutamos pulsando el botón "Start" y la ventana en la esquina superior izquierda debe marcar "Successful".



Ventana con el succesfull de la carga de nuestro archivo al FPGA.

Problemas y soluciones

USB-Blaster driver no encontrado, debido a que ocurrio este error fue necesario dirigirnos a la siguiente carpeta: C:\intelFPGA_lite\18.1\quartus\drivers, donde ejecutamos el DPInst.exe como administrador para instalar los drivers que faltaban.

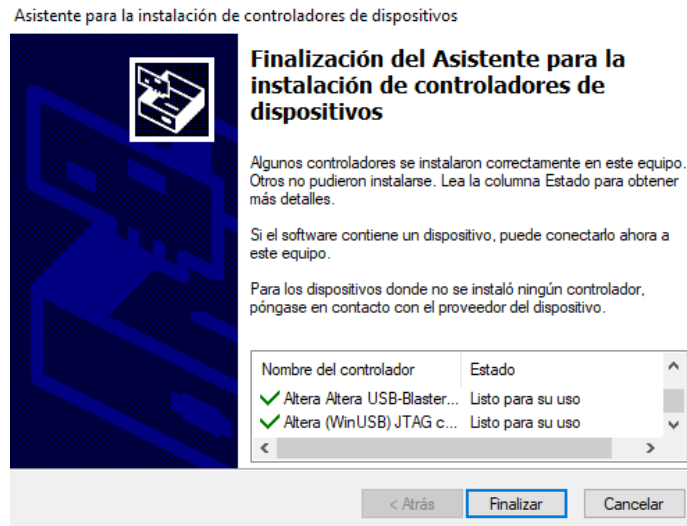
Windows no pudo instalar su USB-Blaster

Windows no puede encontrar controladores para el dispositivo.

Si sabes quién fabricó el dispositivo, busca información acerca de los controladores para descargar en la sección de Soporte técnico del sitio web del fabricante.

Error marcado por windows al asignar el driver:

Al ejecutar el DPInst.exe nos aparece la siguiente ventana una vez haya finalizado el proceso con el Altera USB- Blaster descargado.



Instalador con los drivers faltantes listos.

Reflexión Individual

Héctor: En la práctica pude recordar un poco sobre la programación en VHDL y de su estructura como programa, además de que pude conocer cómo asignar puertos a los diferentes pines del FPGA por medio de la consola y de manera gráfica. Además pude explorar una nueva forma de instalar drivers a un dispositivo en concreto debido al error que nos surgió y entender bien cómo es que la programación en VHDL funciona en un dispositivo después de haber sido sintetizado el programa.

Gabriela: Fue de gran ayuda para poder ver las diferentes maneras que podemos declarar el uso de inputs y outputs, pues puede ser muy útil para facilitar la conexión de cada uno de nuestros I/O, además hubo algunos errores que tuvimos, incluyendo el elegir mal la placa que generó algunos errores, por lo que sí fue de gran utilidad hacerla para familiarizarnos con el dispositivo físico con quartus.

Video

<https://drive.google.com/file/d/1Np9aOruvRBdGpgtP-kJ3-g7QIRaLRiBr/view?usp=sharing>

Bibliografía

1. Manual DE10-lite :

https://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=1021&FID=a13a2782811152b477e60203d34b1baa

2. Guia Practica lab 04:

https://docs.google.com/document/d/1cDhqNXdcL0ft5W1FUYTI_XAr3HVN1pJWEjeVxCYhBio/edit