

Objetivo General.

Realizar una práctica introductoria sobre la síntesis de código HDL en un FPGA.

Objetivos específicos.

- Familiarizar al estudiante con el ambiente integrado de desarrollo de XILINX.
- Presentar al estudiante un ejemplo de código HDL para un ALU muy sencillo que prende los leds en una secuencia determinada.

Propuesta del problema.

El objetivo de experimento consiste en sintetizar una ALU muy sencilla.

Esta ALU despallará un número en binario en los LEDS de la tarjeta de desarrollo.

Este número se incrementará en una unidad aproximadamente cada segundo.

El código para resolver este problema será proporcionado por el profesor, el estudiante no tendrá necesidad de implementarlo para este experimento.

Set de Instrucciones.

El ALU para el experimento 1 solo tiene 6 instrucciones.

El formato de las instrucciones es el siguiente

(4 bits)	(8 bits)	(8 bits)	(8 bits)
Operación: Op	Destino: Dst	Fuente1: Src1	Fuente2: Src2

Operación: Op	Destino: Dst	Valor inmediato: Imm
----------------------	---------------------	-----------------------------

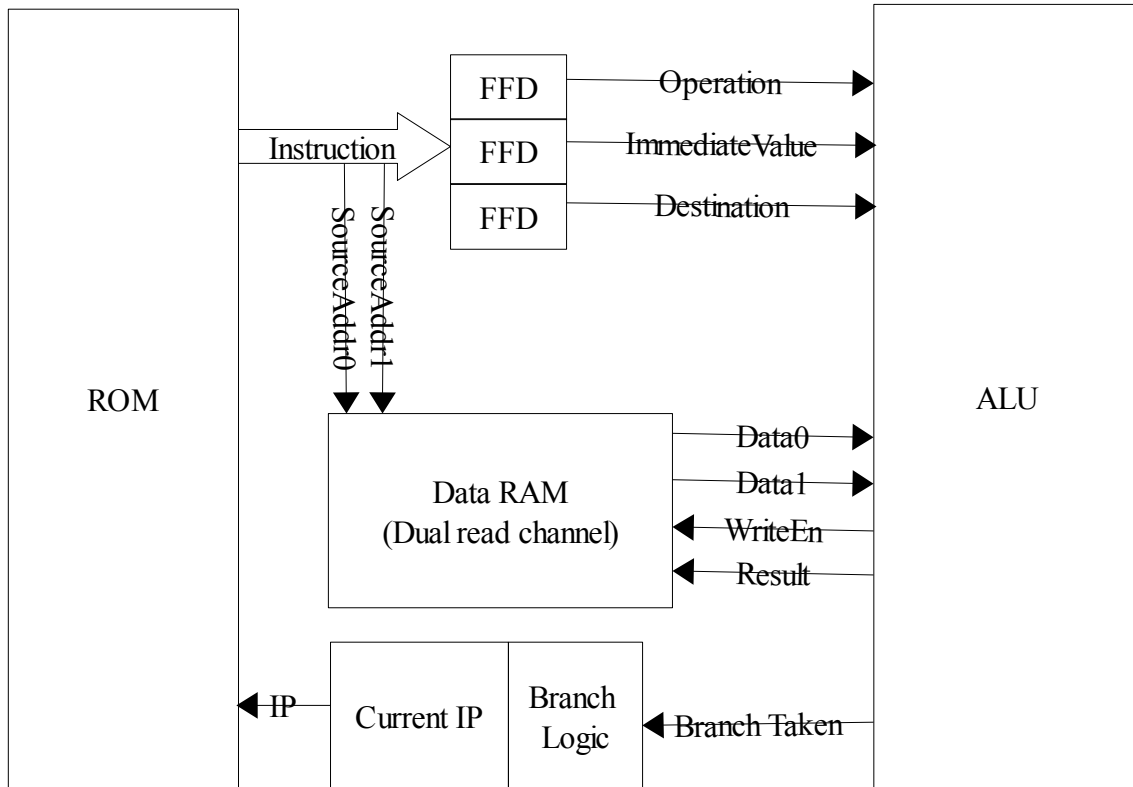
Tanto la fuente como los destinos puede ser uno de los 8 registros de 8 bits R1 hasta R7.

El set de instrucciones es el siguiente:

Nombre	Ciclos	Descripción
NOP	2	No hay operación
LED	2	Manda el contenido de Src1 a los LEDS. LEDS = Src1
ADD	2	Suma los contenidos de Src1 y Src2 y los guarda en Dst . Dst = Src1 + Src2
STO	2	Guarda el valor inmediato en Dst . Dst = Imm
JMP	2	Salta incondicionalmente a Dst . Próximo IP = Dst.
BLE	2	Salta a Dst si Src1 <= Src2

Arquitectura.

La arquitectura del sistema es la siguiente



Procedimiento.

1. Crear el proyecto de XILINX, "Project Navigator", seleccione File → New Project.

New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name: MiniAlu

Location: C:\Users\diego\Desktop\MiniAlu

Description:

Select the type of top-level source for the project

Top-level source type: HDL

More Info Next Cancel

Pulse Next

Aparece una ventana de dialogo con las propiedades de la tarjeta. Seleccionar las

- siguientes opciones y clic next.
- • Family: Spartan3E
- • Device: xc3s500E
- • Package: fg320
- • Speed Grade: -4
- • Synthesis Tool: XST (VHDL/Verilog)
- • Simulator: ISE Simulator (VHDL/Verilog)
- • Preferred Language: Verilog o VHDL. En este caso verilog.

New Project Wizard

Device Properties

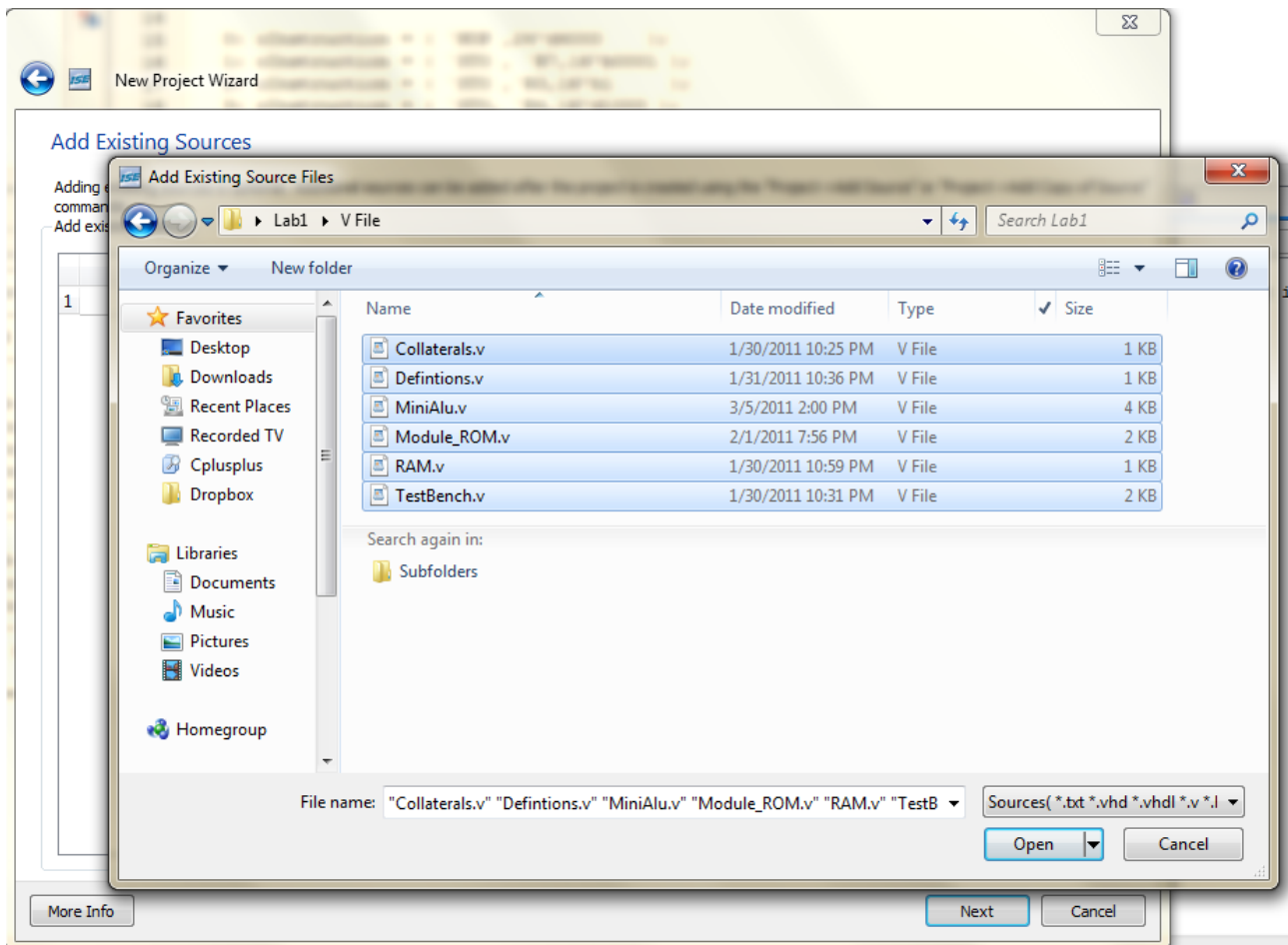
Specify device and project properties.
Select the device and design flow for the project

Property Name	Value
Product Category	General Purpose
Family	Spartan3E
Device	XC3S500E
Package	FG320
Speed	-4
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Manual Compile Order	<input type="checkbox"/>
Enable Enhanced Design Summary	<input checked="" type="checkbox"/>
Enable Message Filtering	<input type="checkbox"/>
Display Incremental Messages	<input type="checkbox"/>

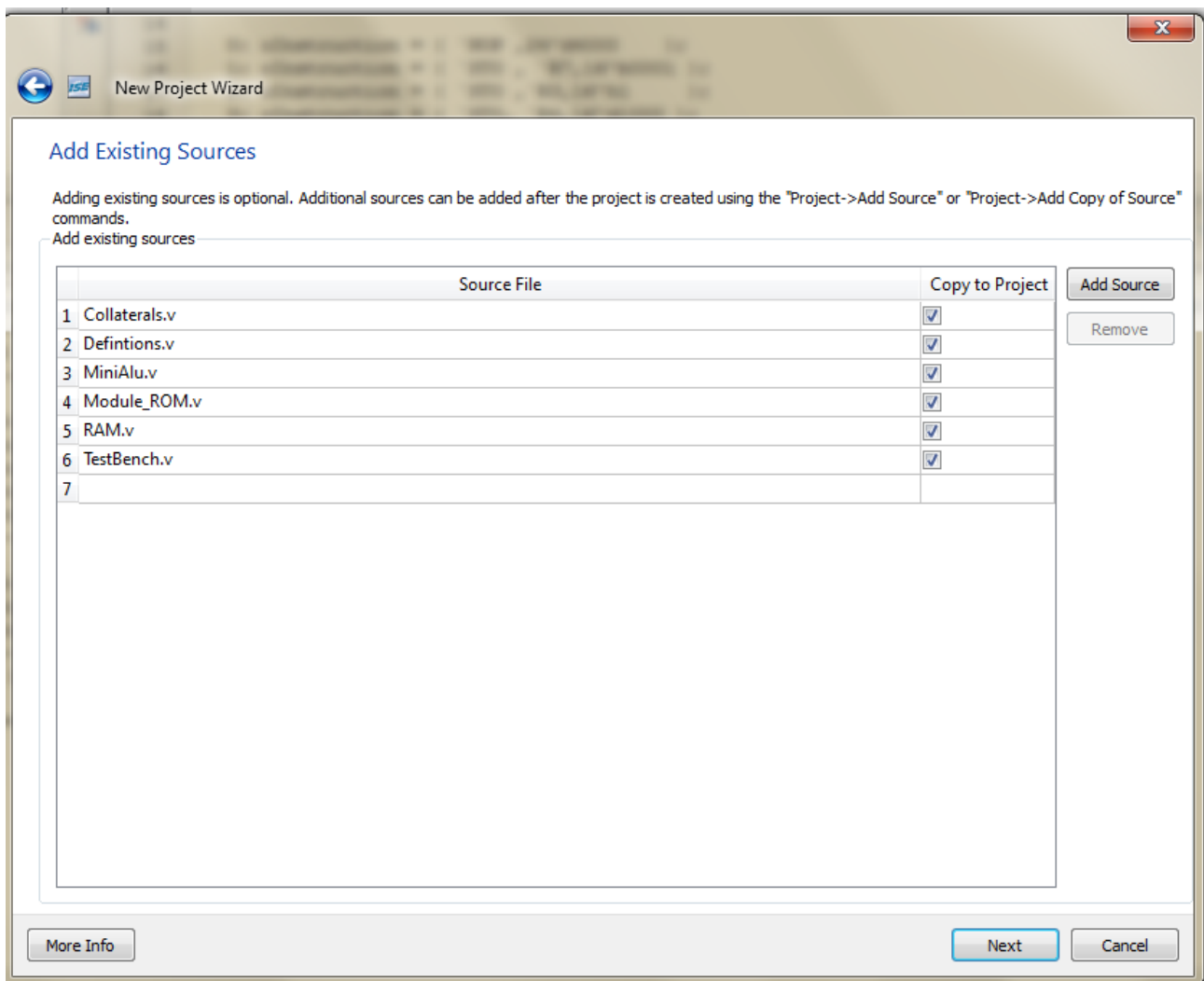
More Info Next Cancel

En la ventana de Create New Source, no se debe crear ninguna fuente nueva, pulsar Next.

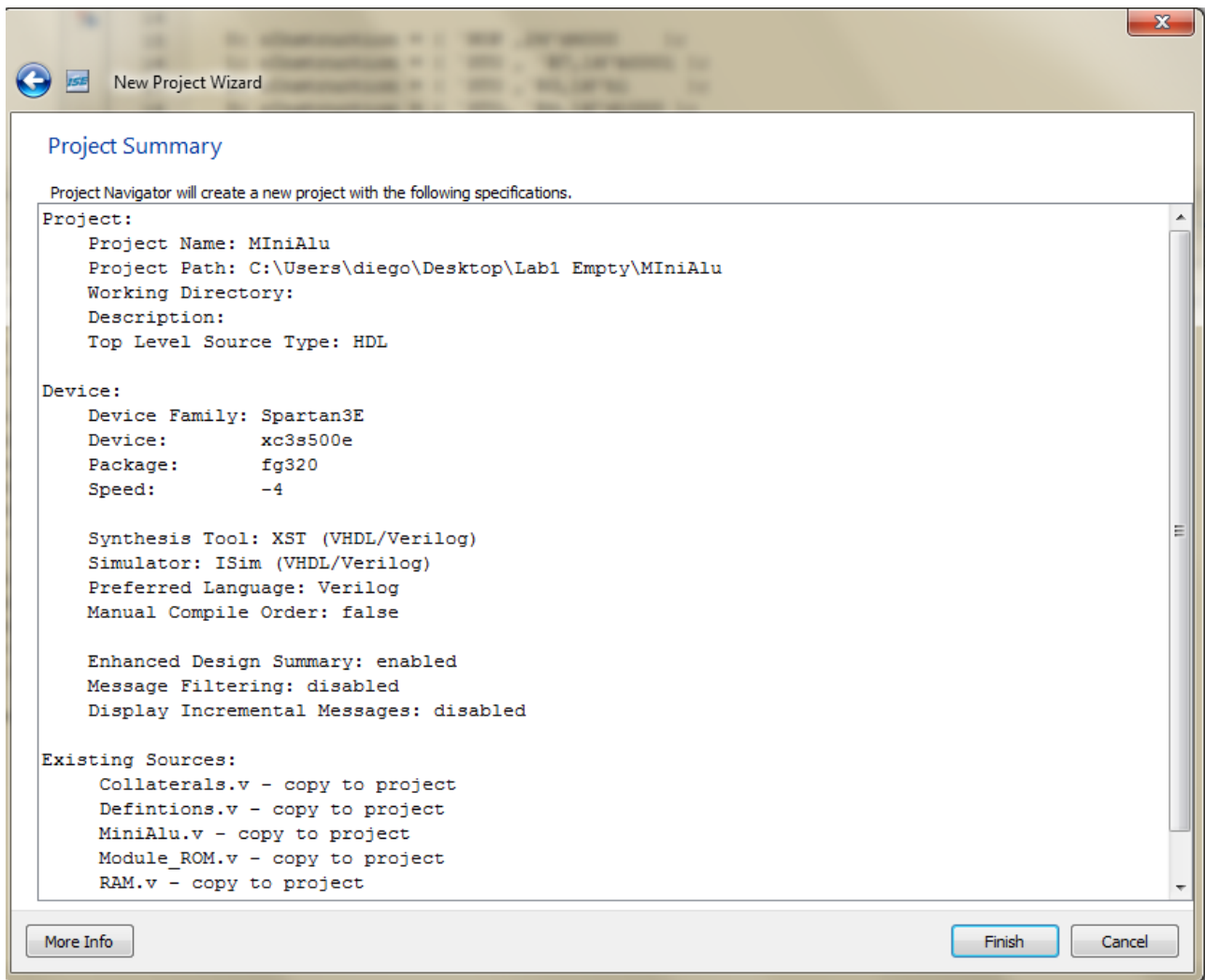
En la ventana de add existing source se deben agregar los archivos fuente *.v como se muestra



la siguiente ventana muestra las fuentes agregadas.

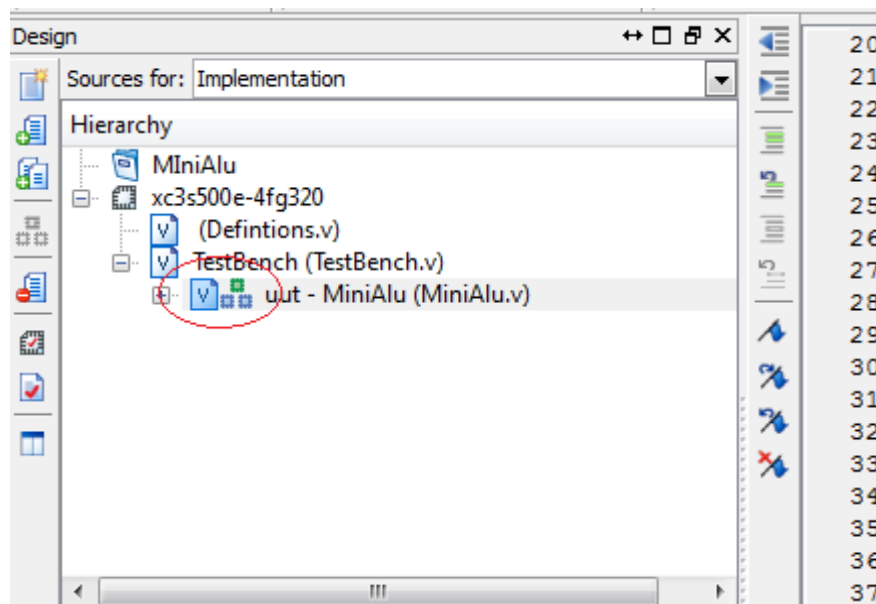


Pulse Next.



La ventana anterior muestra un dialogo de confirmación.
Pulsar Finish

Asegúrese de que las fuentes están seleccionadas para implementación y que el **“top module”** es **MiniAlu.v** como se muestra a continuación. Además recuerde que si su proyecto tiene un archivo .ucf, este debe encontrarse bajo la jerarquia de MiniAlu.v y **no bajo la jerarquía de testbench** (Árbol Hierarchy a la izquierda).



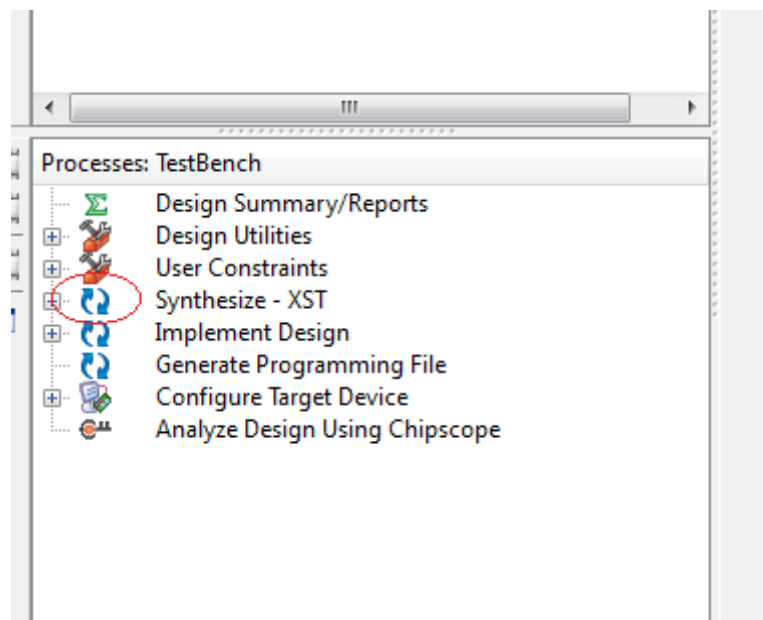
Finalmente, se hará uso de la tarjeta Spartan 3E. Se utilizará los interruptores y los leds. La asignación será de la siguiente forma:

Nombre	Localización	Pull	Drive
Clock	C9	Default	Default
Reset	L13	LVTTL/PULLDOWN	Default
oLed[0]	F12	LVTTL	8
oLed[1]	E12	LVTTL	8
oLed[2]	E11	LVTTL	8
oLed[3]	F11	LVTTL	8
oLed[4]	C11	LVTTL	8
oLed[5]	D11	LVTTL	8
oLed[6]	E9	LVTTL	8
oLed[7]	F9	LVTTL	8

Use de la herramienta “Floorplan IO” del User constraints. En esta ventana se digita las propiedades de los pines IO asociados a la tarjeta Spartan 3E, en este caso el de los interruptores y leds. Debe verse como a continuación.

I/O Ports									
Name	Dir	Neg Diff Pair	Location	Bank	I/O Std	Drive Strength	Slew Type	Pull Type	
All ports (10)									
oLed (8)	Output				0 LVTTTL*	8 SLOW			
oLed[0]	Output		F12		0 LVTTTL*	8* SLOW			
oLed[1]	Output		E12		0 LVTTTL*	8* SLOW			
oLed[2]	Output		E11		0 LVTTTL*	8* SLOW			
oLed[3]	Output		F11		0 LVTTTL*	8* SLOW			
oLed[4]	Output		C11		0 LVTTTL*	8* SLOW			
oLed[5]	Output		D11		0 LVTTTL*	8* SLOW			
oLed[6]	Output		E9		0 LVTTTL*	8* SLOW			
oLed[7]	Output		F9		0 LVTTTL*	8* SLOW			
Scalar ports (2)									
Reset	Input		L13		1 LVTTTL*	12 SLOW		PULLDOWN	
Clock	Input		C9		0 LVCMOS25	12 SLOW			

Sintetise el código dando doble click en el ícono de Synthetize como se muestra:



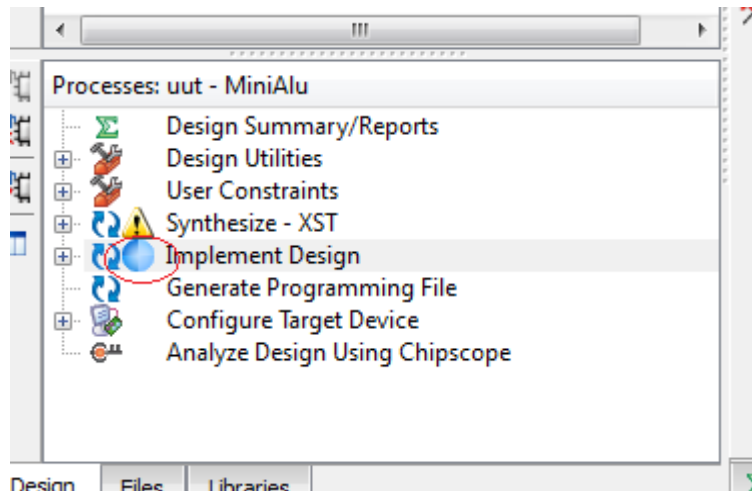
Note en el reporte como los diferentes elementos del diseño son reconocidos por la herramienta de síntesis, la ROM, la RAM de doble canal, el “Up Counter”, etc.

=====			
Advanced HDL Synthesis Report			
Macro Statistics			
RAMs	:	2	
9x16-bit dual-port distributed RAM	:	2	
ROMs	:	1	
16x28-bit ROM	:	1	
Adders/Subtractors	:	2	
16-bit adder	:	2	
Counters	:	1	
16-bit up counter	:	1	
Registers	:	72	
Flip-Flops	:	72	
Comparators	:	1	
16-bit comparator lessequal	:	1	

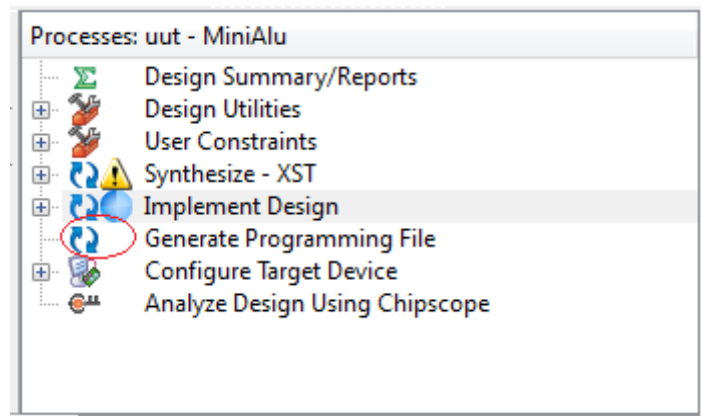
=====

Algunos warnings van a aparecer, esto es normal y no afecta el resultado final, sin embargo es su tarea más adelante resolver estos warnings.

Implemente el diseño dando doble click en “Implement Design” como se muestra seguidamente:



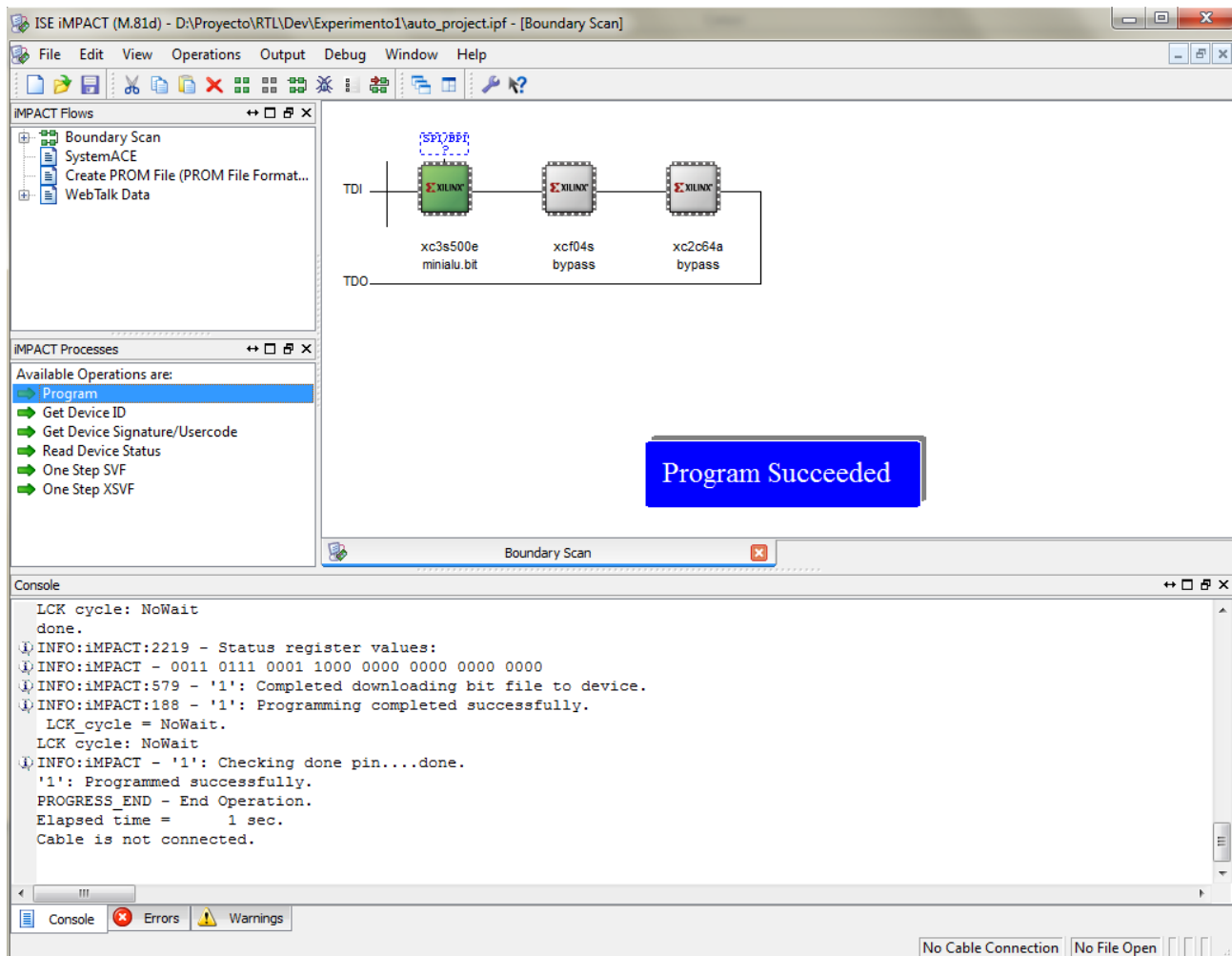
Finalmente, de doble click en “Generate Programming File” para generar el archivo .bit con el que programará la tarjeta.



Conecte la tarjeta Spartan 3E a la computadora por medio del cable USB.

- Hacer doble clic en la pestaña “Configure Device (iMPACT)”
- Clic en el botón “Finished” de la ventana Welcome to iMPACT.
- Asignar el archivo .bit al dispositivo correspondiente. (asignar bypass a los otros dispositivos)
- Programar el FPGA, haciendo clic derecho sobre el icono del dispositivo correspondiente.

- Esperar el mensaje de “Program Succeeded” como se muestra en la siguiente figura:



Si todo salió bien, en este momento usted deberá ver como los leds se prenden con un secuencia numérica.

Ejercicio 1 (Obligatorio):

Puede notar que el número que se despliega en los LEDS cambia muy lentamente. Modifique el código en la ROM para que este cambio sea más rápido.

Ejercicio 2 (Obligatorio):

La ALU cuenta con una operación ADD para sumar, usted ahora debe implementar una operación SUB para restar siguiendo el mismo formato de ADD.

SUB Destino Fuente1 Fuente2.

Modifique el código en la ROM para restar 2 números y desplegar el resultado en los LEDS.

Ejercicio 3 (Opcional):

Utilice el visualizador de ondas de Xilinx para ver si el ALU cuenta con un pipeline.

Si es así, ¿que sucede cuando hay una condición Read-after-Write? ¿Puede usted arreglarlo?

Ejercicio 4 (Opcional):

El diseño tiene una serie de warnings, ¿puede usted arreglarlos?