

AdC 2023

- Patterson, Hennesy, “**Computer Organization and Design - ARM edition**”, 2017.
- Stuart Sutherland, Simon Davidmann y Peter Flake, “**SystemVerilog For Design - A Guide to Using SystemVerilog for Hardware Design and Modeling**”, Second Edition, 2006.
- Hennesy, Patterson, “**Computer architecture - A quantitative approach**”, Sixth Edition, 2019.
- Harris, Harris, “**Digital Design and Computer Architecture - ARM edition**”, 2016.

Fechas de parciales

- Parcial 1: Viernes 22 de septiembre
- Parcial 2: Miércoles 8 de noviembre
- Recuperatorios: Miércoles 22 de noviembre

FIELD PROGRAMMABLE GATE ARRAYS (FPGAs)

Arquitectura de Computadoras 2023

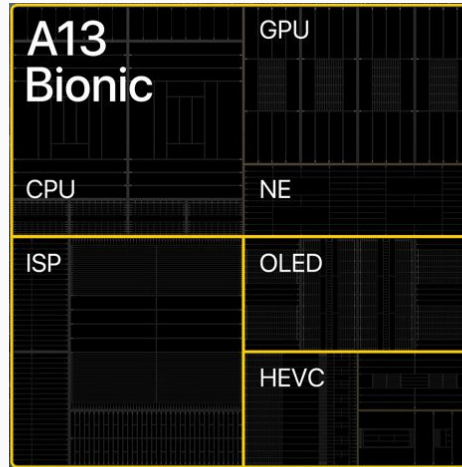
FPGAs: ¿Qué son?

“Arreglos de Compuertas Programables en Campo”

Son circuitos integrados digitales que contienen bloques lógicos programables junto con interconexiones configurables entre dichos bloques.

Microprocesadores, ASIC, FPGA

- Microprocesadores: el hardware es fijo. Las funciones se realizan en software.
- ASIC (Application Specific IC): diseñado para implementar una función lógica particular. Son “hechos a medida”. Son más rápidos que las FPGA, consumen menos y, fabricados en gran escala, son más baratos.
- FPGA: las funciones se realizan en hardware. No son hechas a medida, por lo que el usuario puede configurarlas de acuerdo a sus necesidades.



ASIC: A13 Bionic chip

FPGAs: ¿Cómo se configuran?

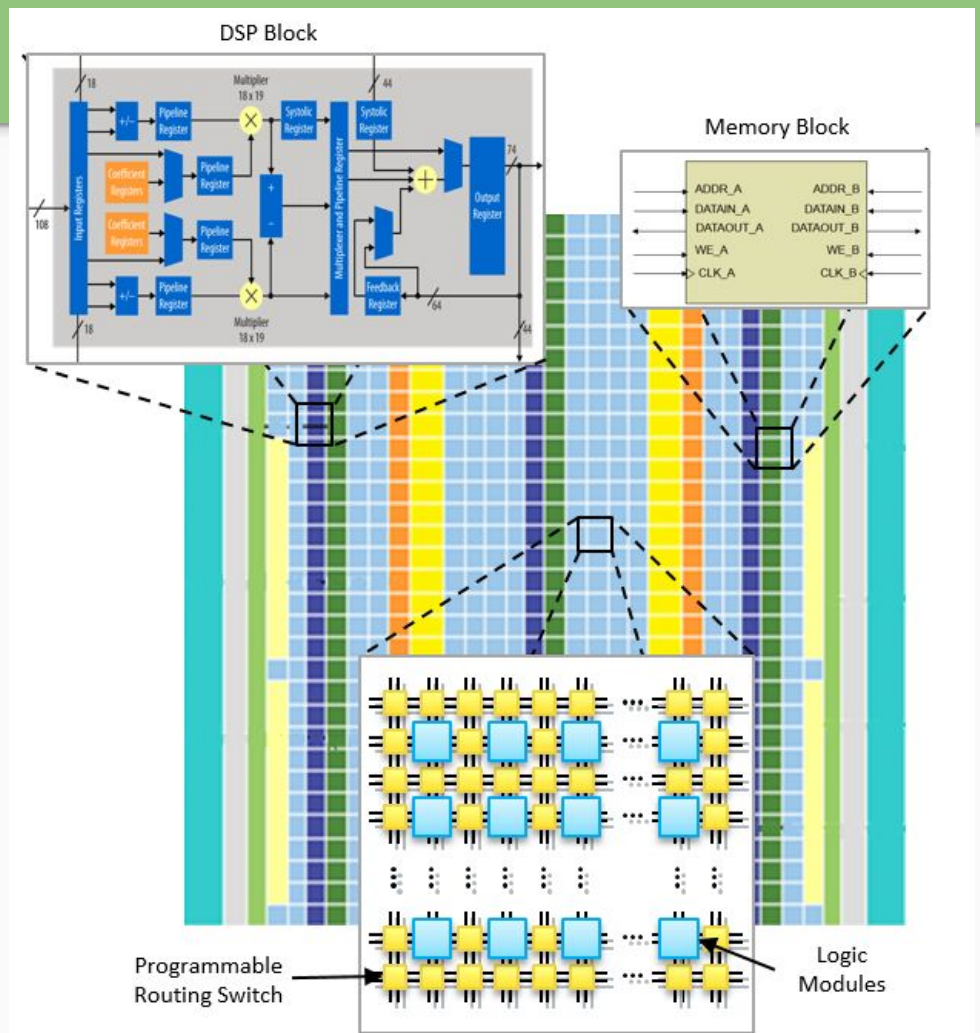
- Algunas FPGA permiten ser programados una sola vez (OTP One Time Programmable).
- Otras pueden ser programadas múltiples veces “in the field”, es decir, no las programa el fabricante, sino el desarrollador.
- Si un dispositivo puede ser programado mientras embebido en un sistema mayor, se dice que es ISP (In System Programmable).

FPGAs: Elementos básicos

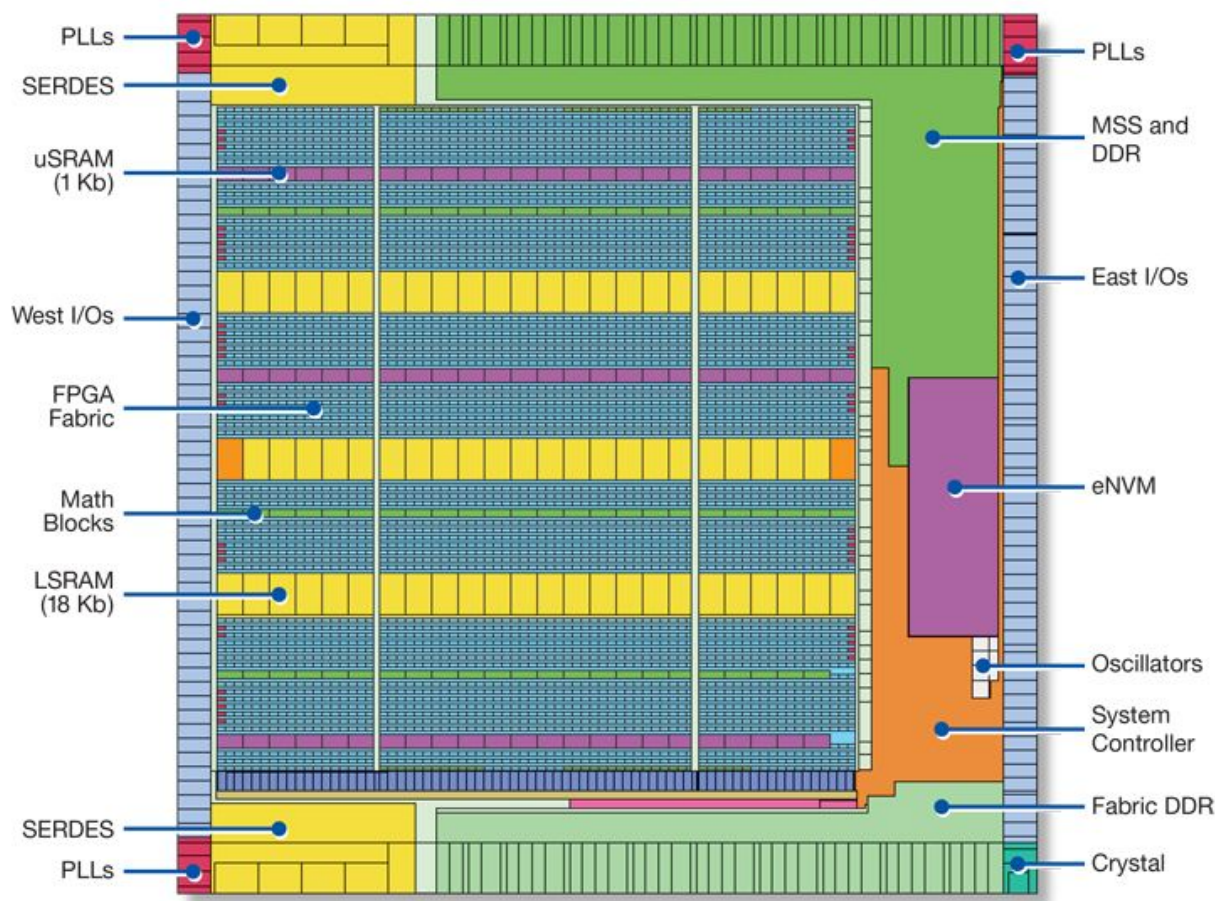
- Elementos lógicos
- Recursos de memoria
- I/O configurables
- Recursos de ruteo
- Recursos adicionales

FPGAs:

Arquitectura genérica

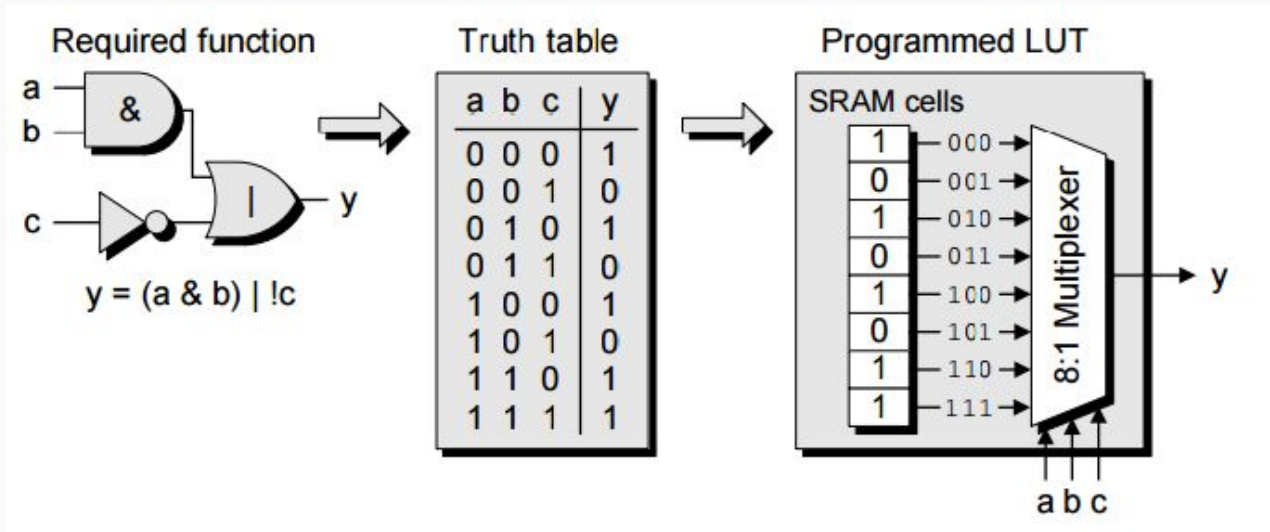


FPGAs: Arquitectura genérica



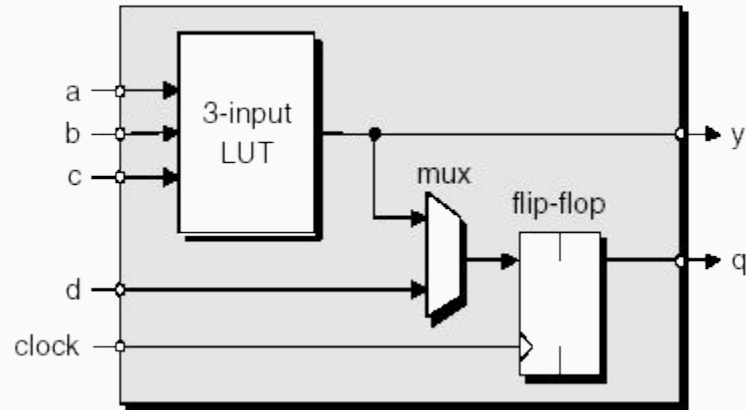
FPGAs: Elementos lógicos (LUT)

La función lógica se almacena en una tabla de verdad de 16x1 (para las LUTs de 4 entradas). La columna de valores de salida de la función combinacional son los valores que realmente se almacenan en la LUT.



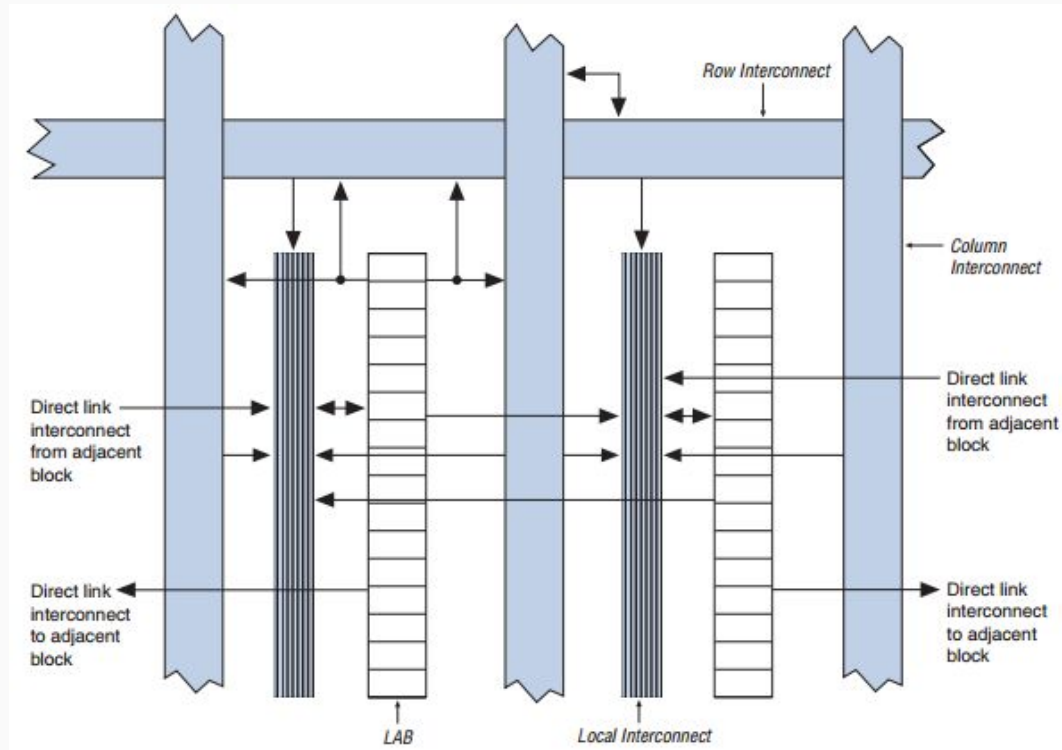
FPGAs: Elementos lógicos

Todas las FPGA se basan en arrays de pequeños elementos de lógica digital. Para usar un determinado dispositivo, los problemas de lógica digital deben ser descompuestos en circuitos lógicos que puedan ser mapeados en una o más de estas “celdas lógicas”.



Logic Array Blocks

Contienen grupos de LEs:



FPGAs: Recursos de memoria

Para aplicaciones que requieren acceso a memoria, las FPGAs cuentan con bloques de memoria disponibles. La cantidad de bloques disponibles depende del tamaño de la FPGA.

Se pueden conectar diferentes RAMs en cascada, ya sea para tener un mayor ancho de la palabra de datos, para tener un mayor tamaño de la memoria o ambos.

Un mismo bloque de memoria puede ser configurado para que funcione como RAM, ROM, FIFO (First Input First Output), convertidor de ancho de palabra, registro de desplazamientos, etc.

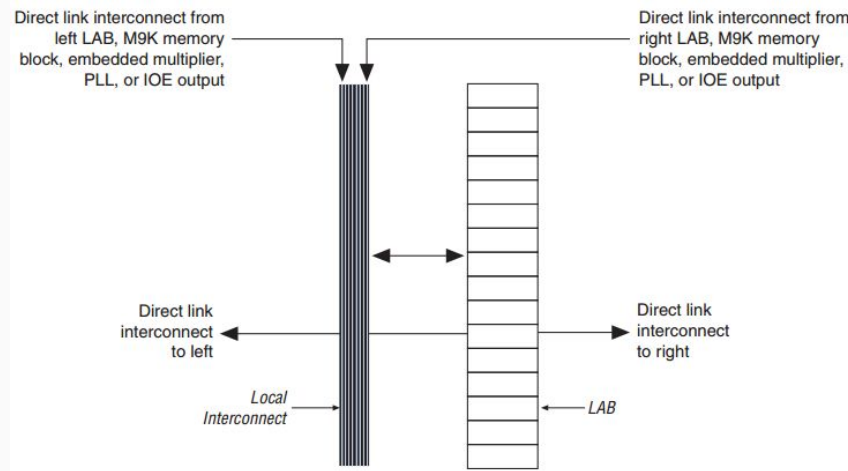
FPGAs: I/O Configurables

Para poder recibir y transmitir señales digitales, las FPGAs disponen de un complejo bloque de E/S que posibilita su uso en muy diversos rangos de tensiones, frecuencias de trabajo, estándares de señales digitales, etc., lo que las hace muy adaptables a las necesidades del sistema del que forman parte.

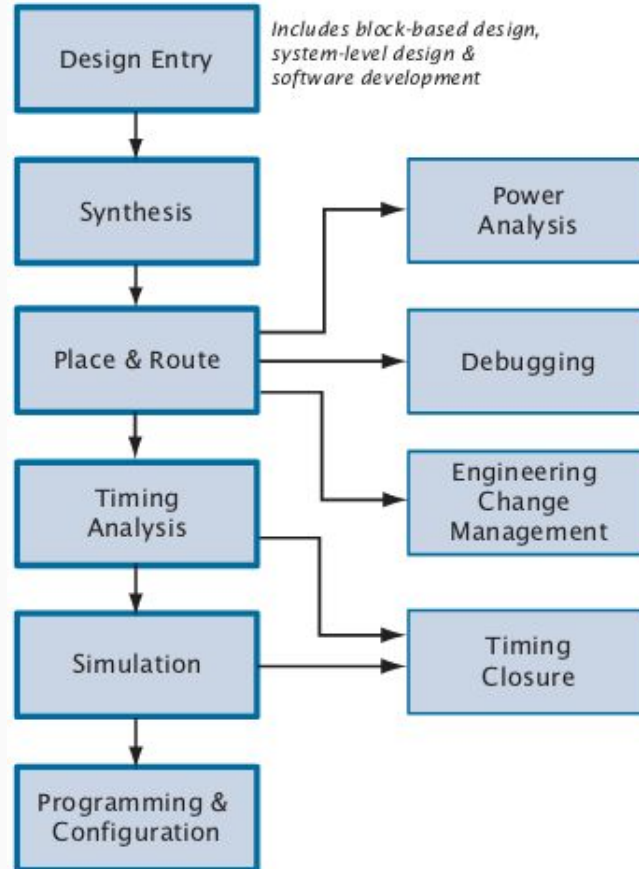
Existe un bloque E/S por cada terminal de la FPGA, por lo que cada una puede ser configurada como entrada, como salida o bidireccional.

FPGAs: Recursos de ruteo

Las conexiones entre LEs dentro de un LAB se realizan mediante columnas y filas internas (Local Interconnect). Las conexiones con otros LABs, PLLs, memorias y multiplicadores adyacentes también pueden realizarse a través de la línea Local Interconnect (Direct link interconnect), minimizando el uso de filas y columnas externas y proporcionando un mayor rendimiento.



Flujo de diseño



FPGAs: Compilación del circuito

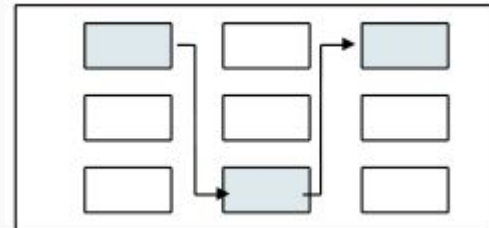
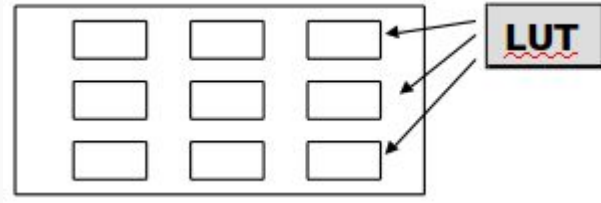
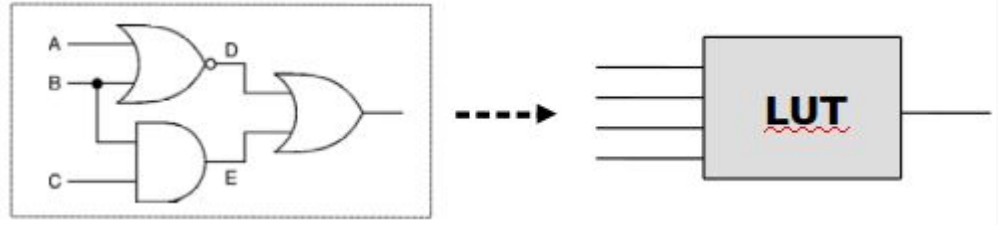
Technology Mapping



Placement



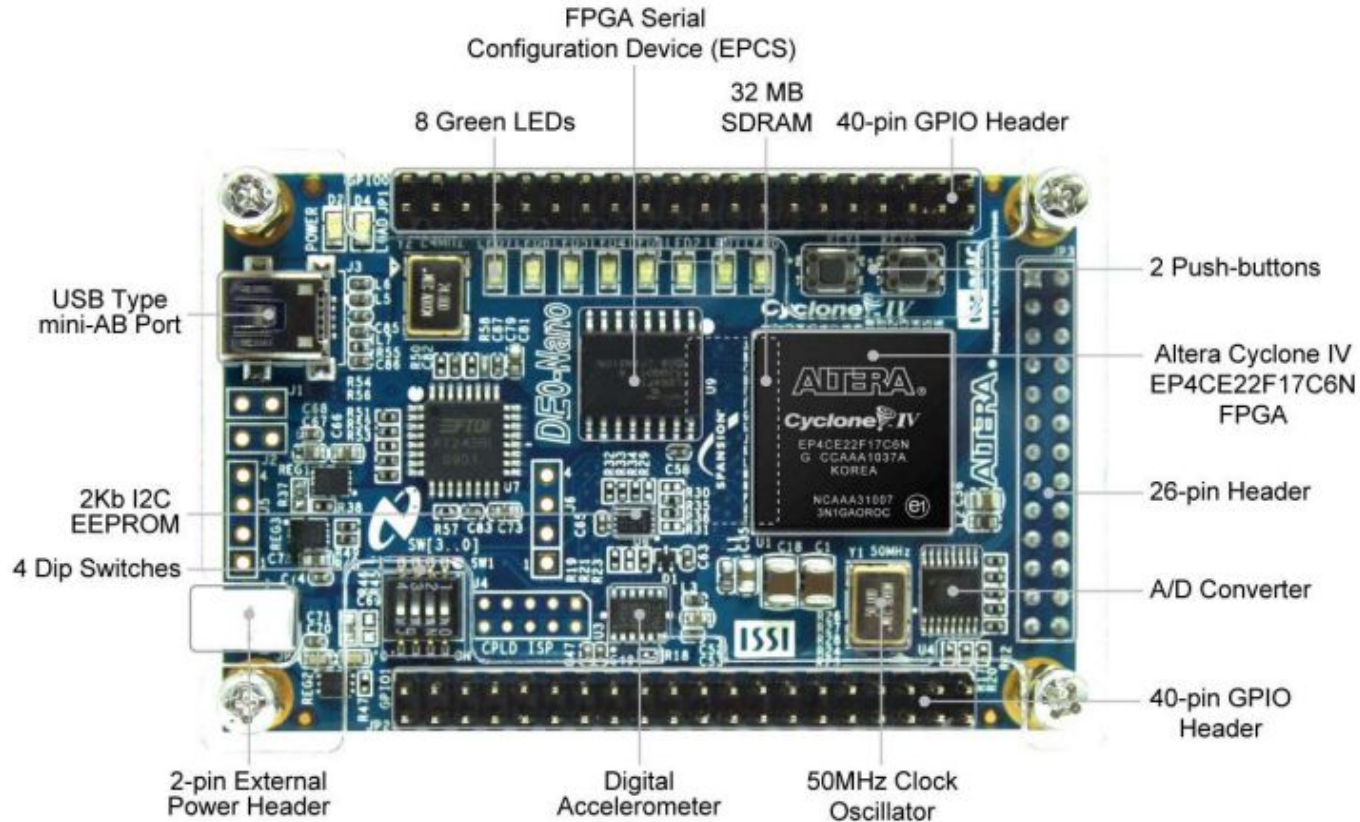
Routing



Flujo de diseño

- **Design Entry:** El circuito deseado es especificado mediante un diagrama esquemático o utilizando algún lenguaje de descripción de Hardware como SystemVerilog.
- **Synthesis:** A partir del diseño ingresado, se infiere la lógica correspondiente y se sintetiza a un circuito utilizando los elementos lógicos (LEs) del chip FPGA.
- **Functional Simulation:** Se verifica la funcionalidad del diseño sintetizado mediante simulación.
- **Fitting:** La herramienta *Fitter* determina la ubicación de los LEs del diseño en los LEs disponibles en el chip FPGA y elige las interconexiones entre ellos.
- **Configuration:** El circuito diseñado es implementado físicamente en el chip FPGA.

DE0-Nano - Altera Cyclone IV FPGA starter board

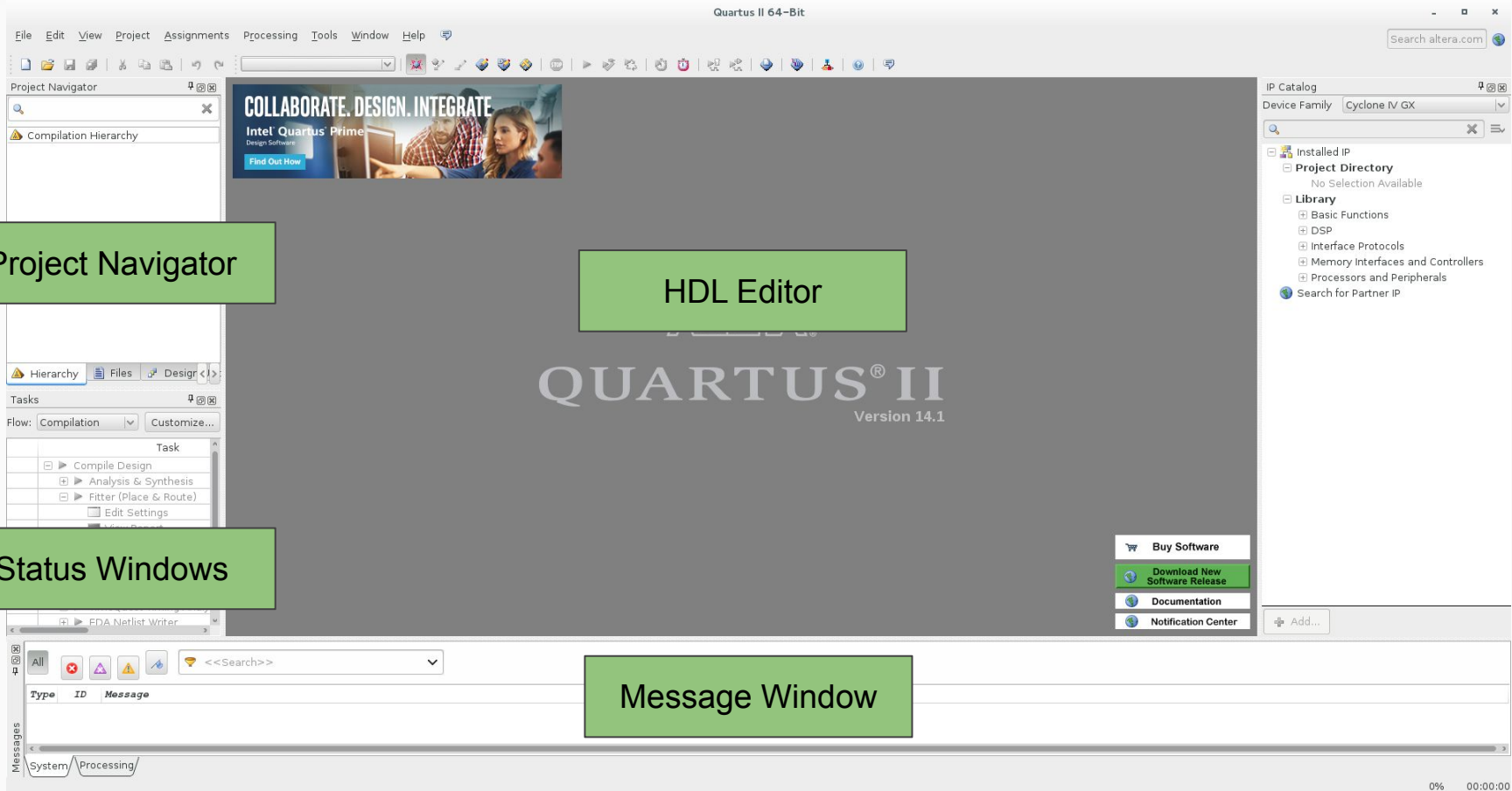


Bibliografía de consulta

- G. Güichal. “Dispositivos Lógicos Programables (FPGAs)”. 2010.
- C. Sisterna. “Field Programmable Gate Arrays (FPGAs)”.
- Altera Corporation. “Cyclone IV Device Handbook”. 2009.
- D. M. Harris, S. L. Harris. “Digital Design and Computer Architecture - Arm Edition”.

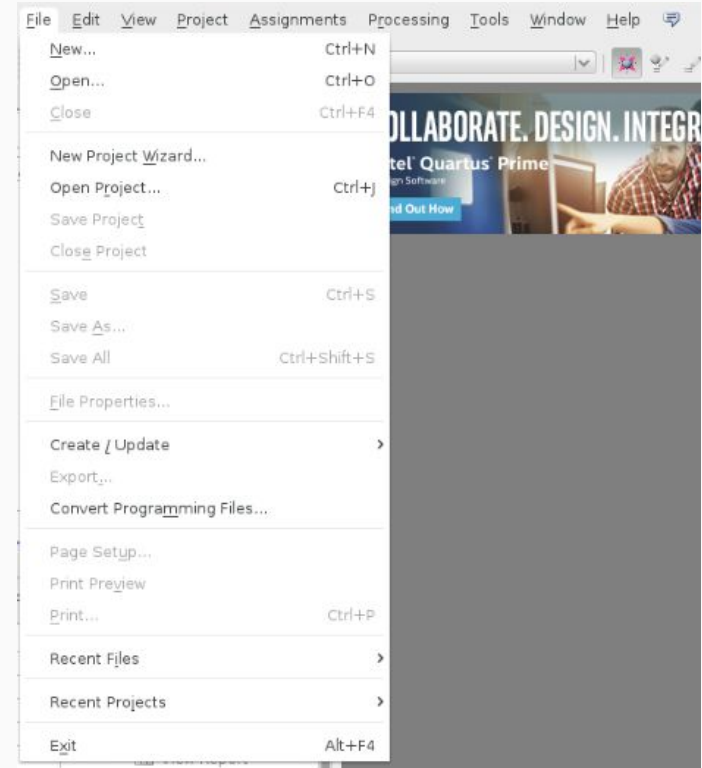
Introducción a la herramienta Quartus Prime

Quartus Prime: GUI



Quartus Prime: Crear un proyecto

- Click en File
- Seleccionar “New Project Wizard”
- Esto abrirá una nueva ventana donde se especificará la información del proyecto



Quartus Prime: Crear un proyecto

- Seleccionar el directorio donde guardar el proyecto
- Seleccionar el nombre del proyecto
- Automáticamente se completa el nombre de la entidad top (No modificar)

New Project Wizard

Directory, Name, Top-Level Entity

What is the working directory for this project?

/home/gvordanovic/Desktop

What is the name of this project?

sillyfunction

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

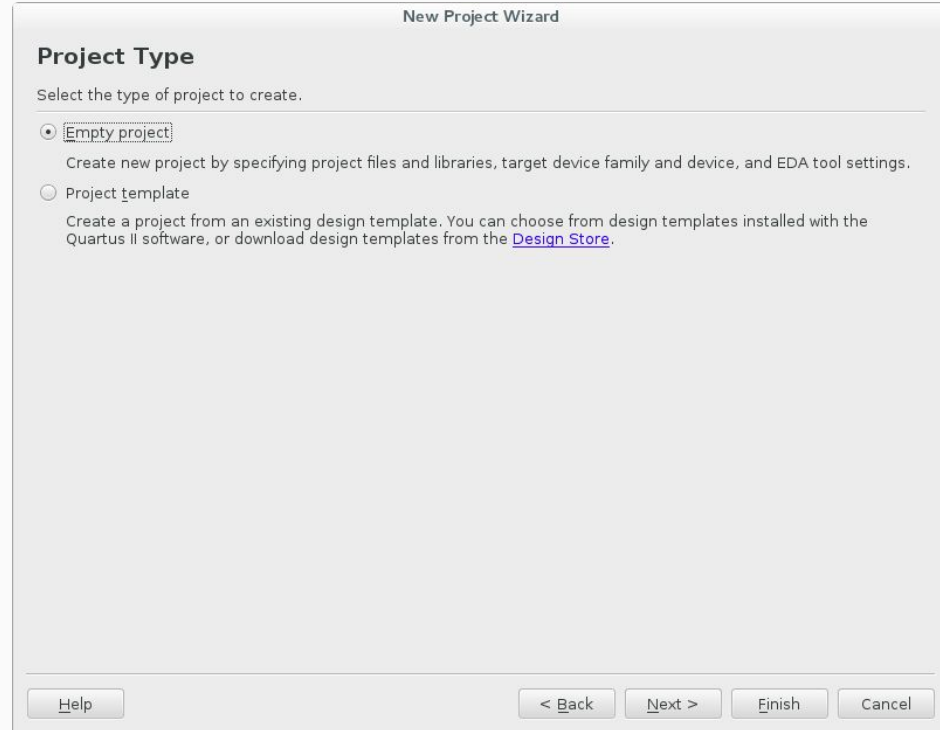
sillyfunction

Use Existing Project Settings...

Help < Back Next > Finish Cancel

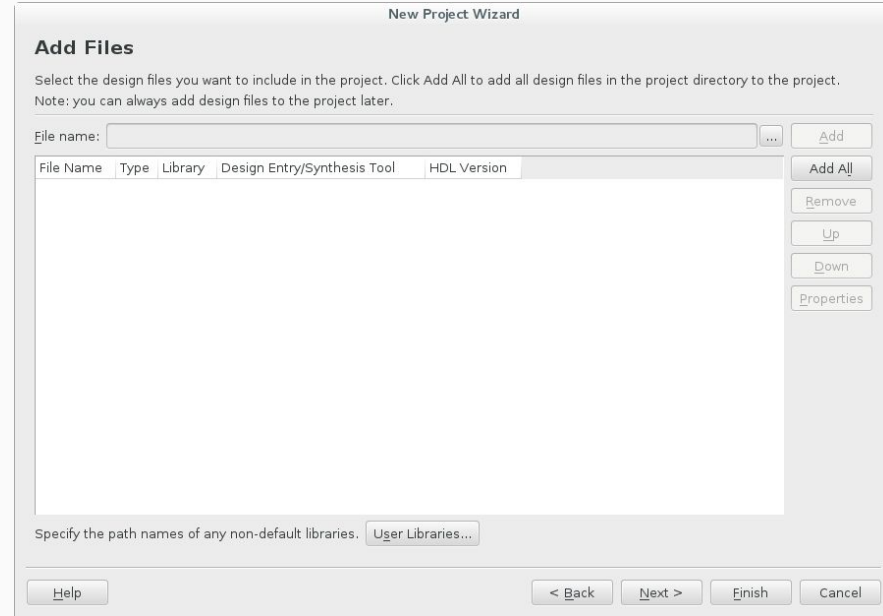
Quartus Prime: Crear un proyecto

- Elegir la opción “*Empty Project*”



Quartus Prime: Crear un proyecto

- Agregar archivos ya existentes al proyecto, en el caso en que existan



Quartus Prime: Crear un proyecto

- Elegir el dispositivo a utilizar:
Cyclone IV: EP4CE22F17C6

The screenshot shows the 'New Project Wizard' dialog box in Quartus Prime, specifically the 'Family, Device & Board Settings' tab. The 'Device' sub-tab is selected. The interface includes instructions on how to select a device and a table of available devices. The 'Cyclone IV E' family and 'EP4CE22F17C6' device are selected. The 'Name filter' is set to 'EP4CE22F17C6', and the 'Show advanced devices' checkbox is checked.

Family, Device & Board Settings

Device Board

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
Device: All

Show in 'Available devices' list

Package: Any
Pin count: Any
Core speed grade: Any
Name filter: EP4CE22F17C6
☒ Show advanced devices

Target device

☐ Auto device selected by the Fitter
☒ Specific device selected in 'Available devices' list
☐ Other: n/a

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elemen
EP4CE22F17C6	1.2V	22320	154	154	608256	132

Help < Back Next > Finish Cancel

Quartus Prime: Crear un proyecto

- Verificar que la herramienta de simulación sea ModelSim Altera y el formato SystemVerilog HDL
- Seleccionar “*Finish*”

New Project Wizard

EDA Tool Settings

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

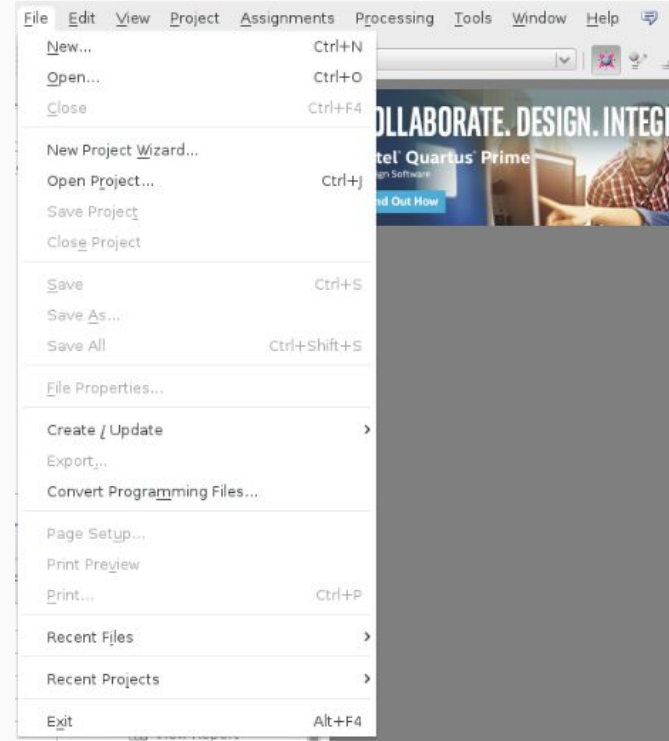
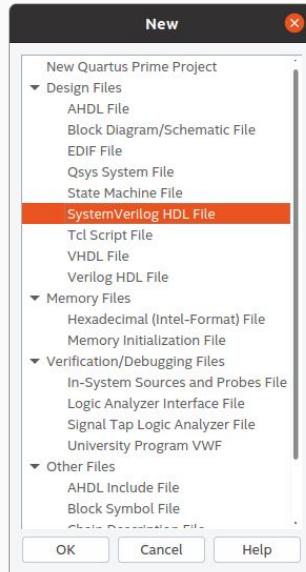
EDA tools:

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

Help < Back Next > Finish Cancel

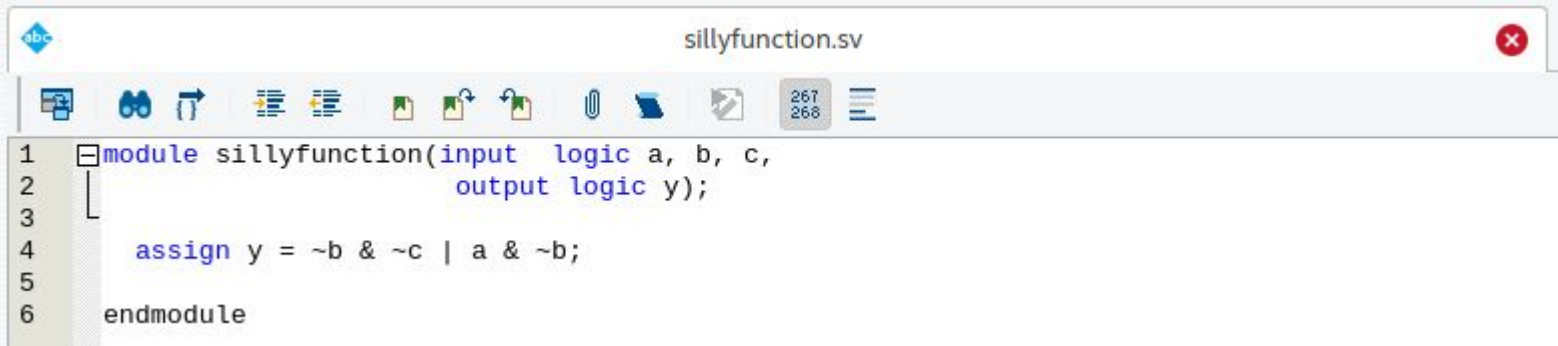
Quartus Prime: Crear un archivo hdl

- Click en File
- Seleccionar “New”
- Esto abrirá una nueva ventana donde se especificará el tipo de archivo, en este caso: Design Files / SystemVerilog HDL File



Quartus Prime: Crear un archivo hdl

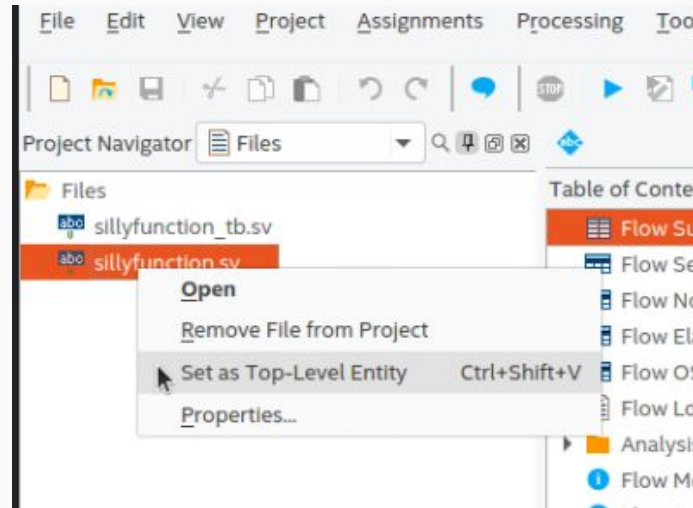
- Escribir el diseño
- Guardar el archivo como `sillyfunction.sv` (respetando el nombre de la entidad!)



```
1 module sillyfunction(input logic a, b, c,  
2                       output logic y);  
3  
4     assign y = ~b & ~c | a & ~b;  
5  
6 endmodule
```

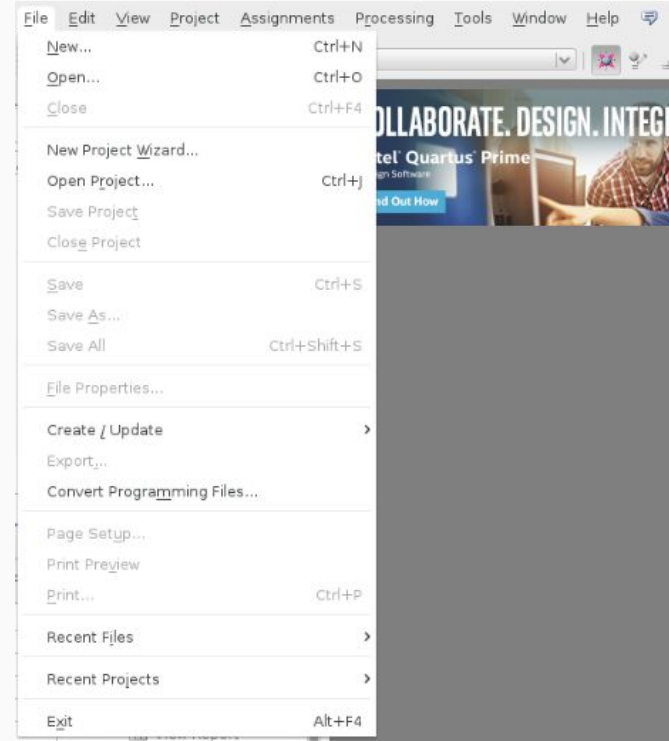
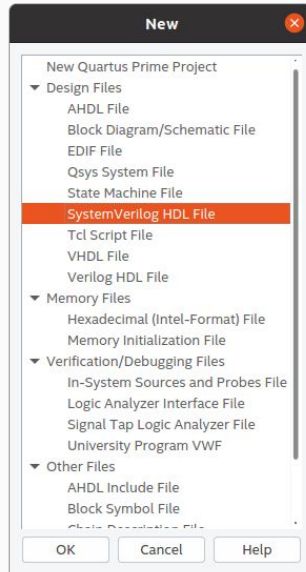
Quartus Prime: Configurar como “Top-level entity”

Al archivo HDL de mayor jerarquía es necesario configurarlo como entidad top-level



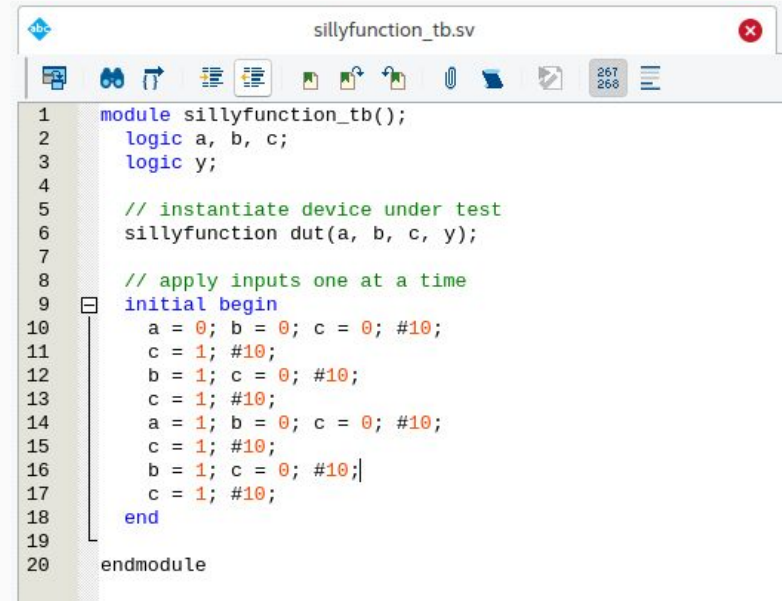
Quartus Prime: Crear un archivo de simulación

- Click en File
- Seleccionar “New”
- Esto abrirá una nueva ventana donde se especificará el tipo de archivo, nuevamente seleccionar: Design Files / SystemVerilog HDL File



Quartus Prime: Crear un archivo de simulación

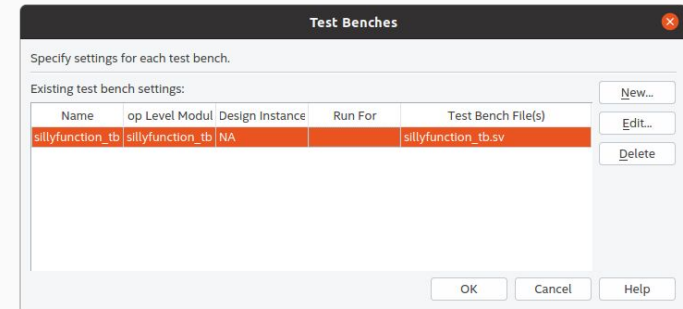
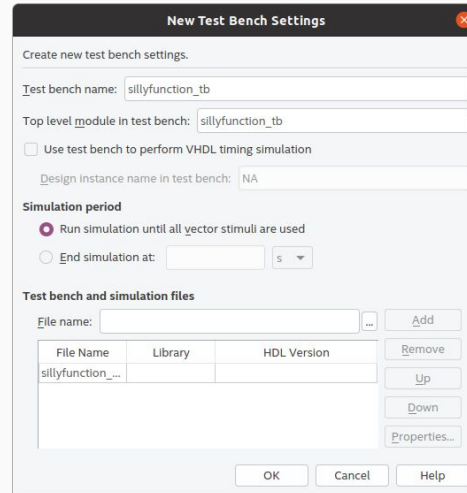
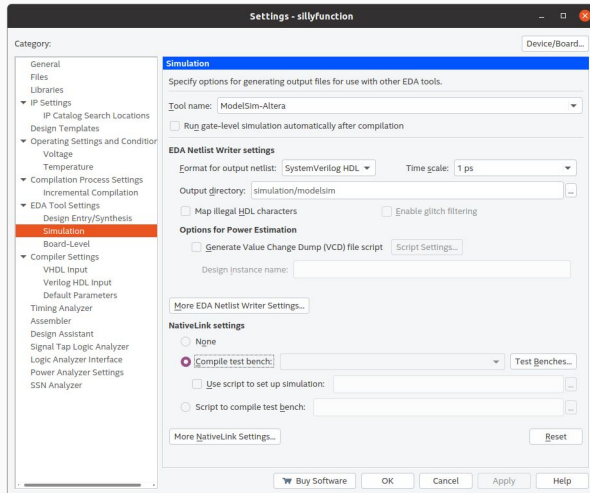
- Escribir el testbench
- Guardar el archivo como sillyfunction_tb.sv



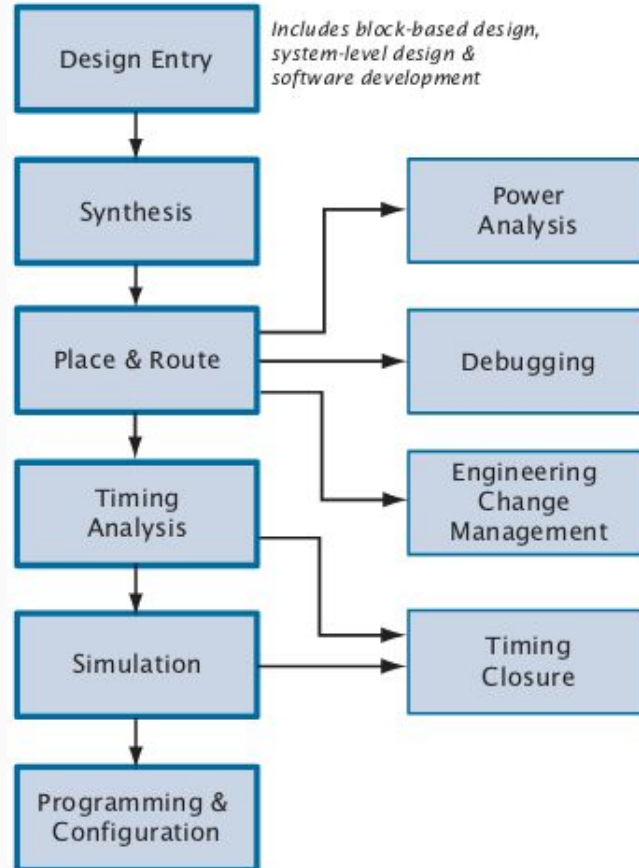
```
1 module sillyfunction_tb();
2     logic a, b, c;
3     logic y;
4
5     // instantiate device under test
6     sillyfunction dut(a, b, c, y);
7
8     // apply inputs one at a time
9     initial begin
10         a = 0; b = 0; c = 0; #10;
11         a = 1; #10;
12         b = 1; c = 0; #10;
13         c = 1; #10;
14         a = 1; b = 0; c = 0; #10;
15         c = 1; #10;
16         b = 1; c = 0; #10;
17         c = 1; #10;
18     end
19
20 endmodule
```

Quartus Prime: Seleccionar el archivo de simulación

- Seleccionar Assignments/Settings...
- Luego en la Categoría Simulation elegir la opción “Compile testbench”
- Clickear la opción “Test Benches...”, luego “New” e indicar el nombre del testbench, en este caso: “sillyfunction_tb” y en la opción “File name”, indicar el nombre del archivo “sillyfunction_tb.sv”



Flujo de diseño

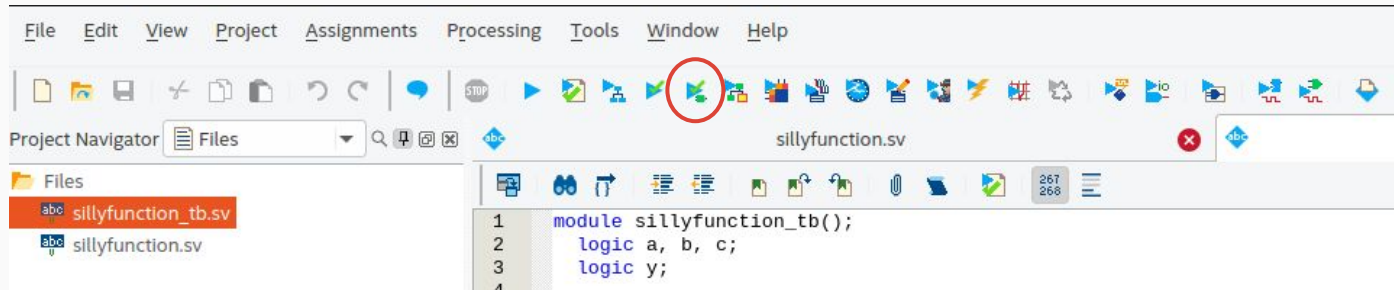


Flujo de diseño

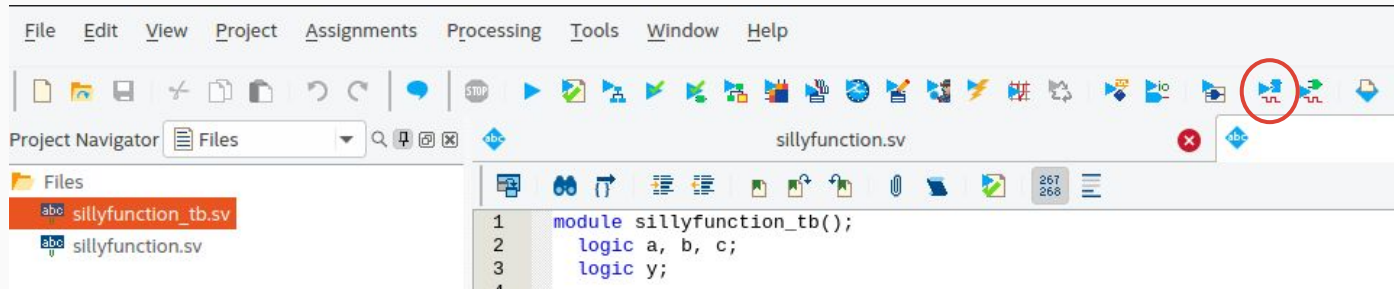
- **Design Entry:** El circuito deseado es especificado mediante un diagrama esquemático o utilizando algún lenguaje de descripción de Hardware como VHDL.
- **Synthesis:** A partir del diseño ingresado, se infiere la lógica correspondiente y se sintetiza a un circuito que consiste de elementos lógicos (LEs) previstos por el chip FPGA.
- **Functional Simulation:** Se verifica la funcionalidad del diseño sintetizado mediante simulación.
- **Fitting:** La herramienta *Fitter* determina la ubicación de los LEs del diseño en los LEs disponibles en el chip FPGA y elige las interconexiones entre ellos.
- **Configuration:** El circuito diseñado es implementado físicamente en el chip FPGA.

Quartus Prime: Realizar la síntesis y simulación

- Síntesis: Start Analysis and Synthesis 



- Simulación: RTL Simulation 



Quartus Prime: Análisis de resultado

ModelSim - INTEL FPGA STARTER EDITION 10.5b

File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help

Layout simulate ColumnLayout default

Search:

sim - Default

Instance

- sillyfunction_tb
- dut
- std
- #vsim_capacity#

Objects

Name	Value	Type
a	1	Regi... Inter...
b	0	Regi... Inter...
c	0	Regi... Inter...
y	1	Regi... Inter...

Wave - Default

Msgs

- /sillyfunction_tb/a
- /sillyfunction_tb/b
- /sillyfunction_tb/c
- /sillyfunction_tb/y

Procesos (Active)

Name	Type (filtered)	Status
------	-----------------	--------

Library sim

Transcript

VSIM 2>
VSIM 2>
Now: 80 ps Delta: 1 ns sim:/sillyfunction_tb

Nombre de las Señales

Visor de Señales

Mensajes y Errores