

## Implementación de una compuerta lógica AND en FPGA mediante el Software Quartus Prime – Lite Edition.

Diego G. Nuñez<sup>a,b</sup>, Fernando Botterón<sup>a,b</sup>, Roberto E. Carballo<sup>a,b\*</sup>

<sup>a</sup> Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.

<sup>b</sup> GID-IE, FI-UNaM, Oberá, Misiones, Argentina.

e-mails: [ndiego208@gmail.com](mailto:ndiego208@gmail.com), [botteron@fio.unam.edu.ar](mailto:botteron@fio.unam.edu.ar), [robertocarballo@fio.unam.edu.ar](mailto:robertocarballo@fio.unam.edu.ar)

---

### Resumen

*Este trabajo tiene como objetivo divulgar los pasos a realizar en la configuración de una FPGA Altera DE0-Nano-SoC con el Software Quartus Prime – Lite Edition, cuya finalidad es lograr la implementación de circuitos lógicos combinacionales y secuenciales, los cuales pueden servir para implementar desde sistemas digitales básicos, hasta mapear algoritmos de control a hardware. El ejemplo particular que se describe en este trabajo es implementar una compuerta AND de dos entradas mediante el lenguaje de descripción de Hardware Verilog.*

**Palabras Clave** – FPGA, Implementación de hardware, Verilog.

### 1 Introducción

Una FPGA (Field-Programmable Gate Array) es un tipo de chip que se puede programar y reprogramar para realizar diferentes tareas de procesamiento. Está compuesta por bloques lógicos configurables y conexiones interconectadas que permiten crear circuitos digitales personalizados utilizando lenguajes de descripción de hardware como VHDL o Verilog [1]. A diferencia de los circuitos integrados fijos, las FPGAs son versátiles y se utilizan en una variedad de aplicaciones, como electrónica de consumo, sistemas embebidos y procesamiento de señales.

Para programar una FPGA, se emplea un lenguaje de descripción de hardware (HDL) como VHDL o Verilog, o herramientas de diseño visual proporcionadas por el fabricante. Una vez programada, la FPGA ejecuta la función deseada. En resumen, las FPGAs son chips programables que ofrecen flexibilidad para crear circuitos digitales específicos y son útiles en numerosos campos tecnológicos.

Este trabajo tiene como objetivo divulgar la experiencia llevada a cabo y el conocimiento adquirido durante la configuración de una FPGA en el Software Quartus Prime – Lite Edition [2], con el propósito de extender su aplicación a los cursos de Arquitectura de Computadoras y Fundamentos de Informática de la carrera de Ingeniería en Computación, así como también a investigaciones sobre control aplicado a Electrónica de Potencia llevadas adelante por el GIDE de la Facultad de Ingeniería.

## 2 Proyecto

Como primer paso, se debe ingresar al software, cuya interfaz se presenta a continuación.

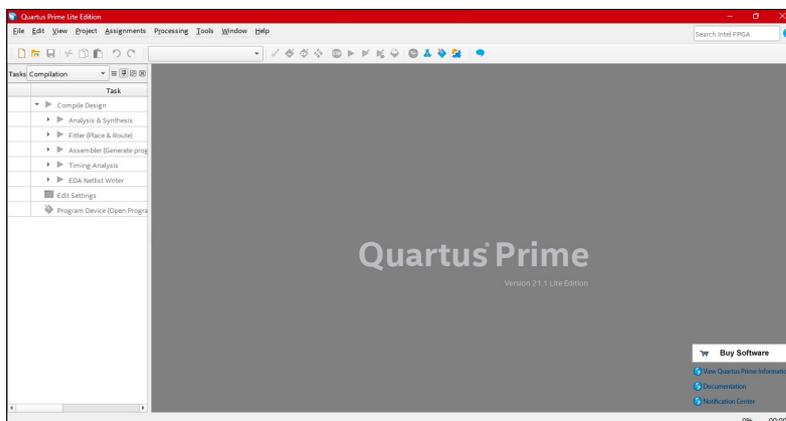


Fig. 1. Interfaz del Software Quartus Prime - Lite Edition.

Una vez ingresado al software, se procede a crear un proyecto, ingresando en **File** → **New Project Wizard**, tal como se aprecia en la Fig. 2.

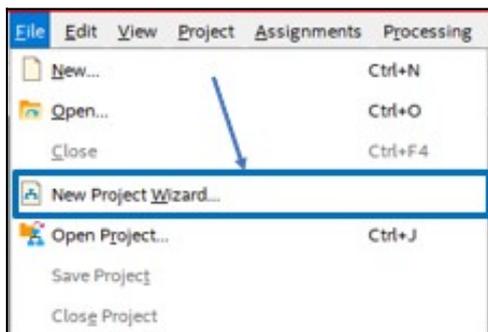
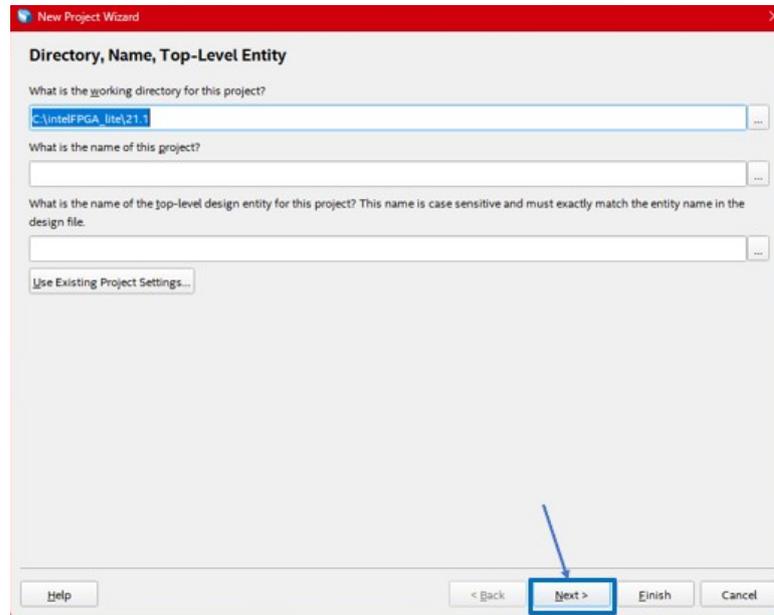


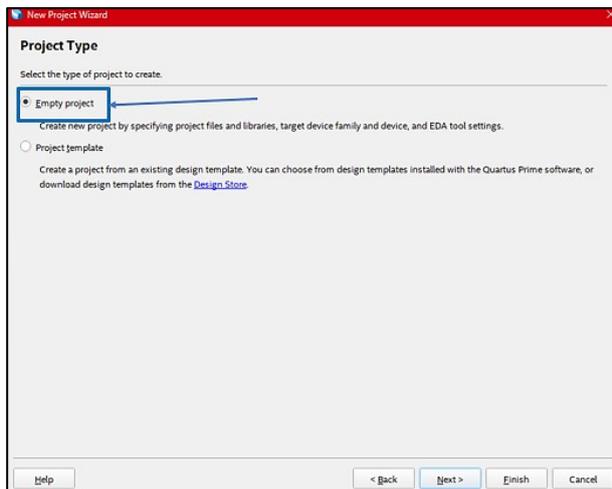
Fig. 2. Pasos para crear un proyecto.

Luego de ingresar en **New Project Wizard**, se deberá indicar la ubicación y el nombre del proyecto. Se recomienda que la ubicación del mismo no se encuentre dentro de la carpeta que se sugiere por defecto, dado que, al momento de cargar el programa a la FPGA, podría producir error; por otra parte, para el nombre del archivo se debe respetar la convención de nomenclatura de archivos, como, por ejemplo, utilizar letras mayúsculas y minúsculas, evitar espacios en blanco, caracteres especiales, entre otras.

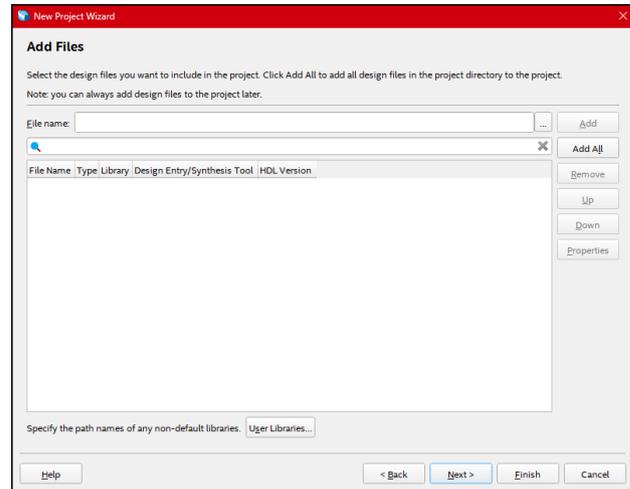


**Fig. 3. Ventana para crear el proyecto.**

Una vez indicada la ubicación del proyecto y establecido el nombre del mismo, se presiona el botón **Next**, como se muestra en la **Fig. 3**. Posteriormente, se deberá aclarar el tipo de proyecto, es decir, si se desea crear uno nuevo o basarse en otro previamente creado. Por lo general, se debe dejar marcado en la opción que aparece por defecto (**Fig. 4**), que es crear un nuevo proyecto, luego pasar de cuadro.



**Fig. 4. Selección del tipo de proyecto.**



**Fig. 5. Cuadro para agregar archivos al proyecto.**

En el cuadro de la **Fig. 5**, se puede agregar archivos al proyecto si así se desea, de lo contrario avanzar de cuadro.

Al pasar al siguiente cuadro, se deberá especificar la familia del dispositivo y la versión.

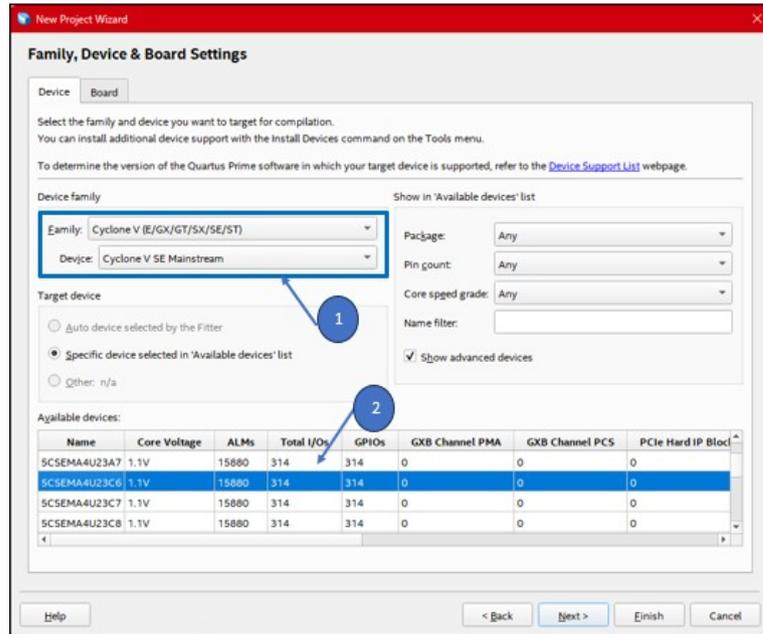


Fig. 6. Cuadro para configurar la familia y la versión del dispositivo.

Como se observa en la Fig. 6, primeramente, se debe seleccionar la familia “Cyclone V(E/GX/GT/SX/SE/ST)” y el dispositivo “Cyclone V SE Mainstream”. Al mismo tiempo, en **Available devices (Dispositivos disponibles)**, se debe buscar y seleccionar el nombre SCSEMA4U23C6, luego avanzar de cuadro.

Una vez avanzado de cuadro, se podrá indicar algunas herramientas de simulación si así se lo desea, tal como se muestra en la Fig. 7, de lo contrario avanzar.

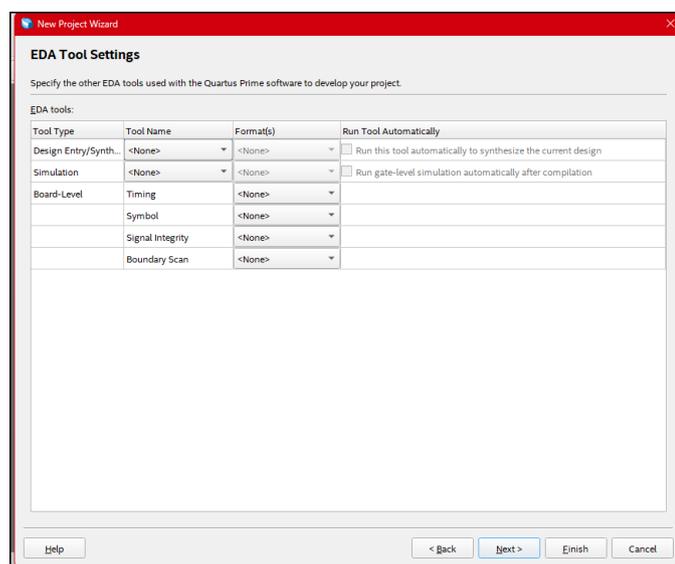
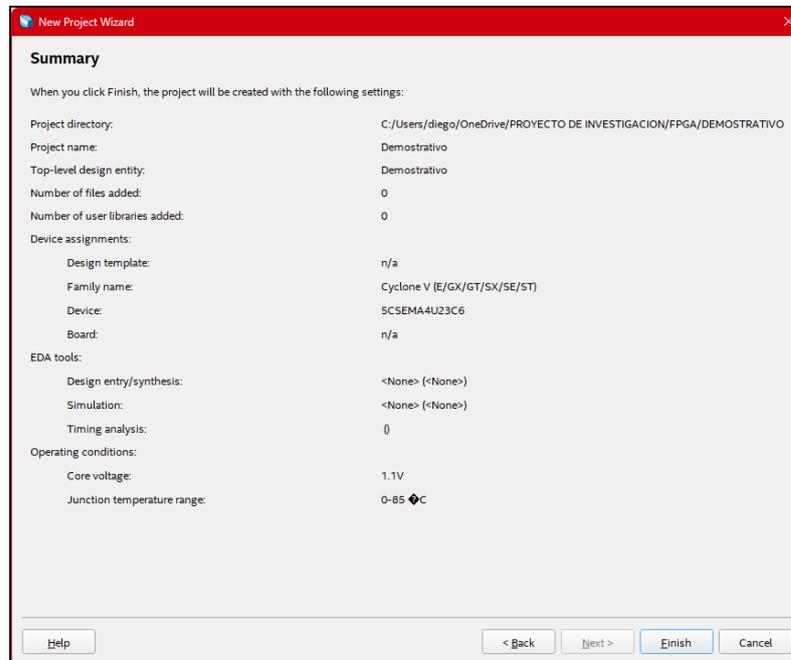


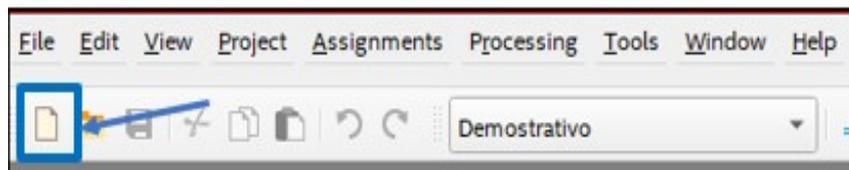
Fig. 7. Cuadro para agregar herramientas de simulación.

Por último, aparecerá un resumen de todas las configuraciones que se a realizado previamente, revisar que todo concuerde con lo establecido y finalizar la creación del proyecto (**Fig. 8**).

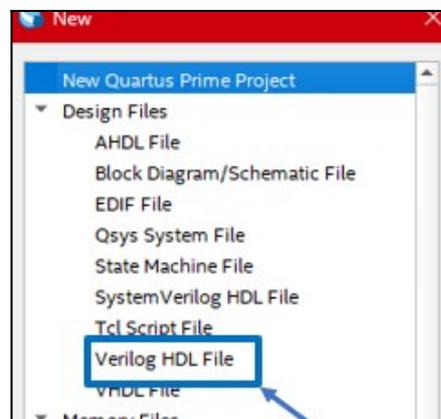


**Fig. 8. Resumen de las configuraciones establecidas del proyecto.**

A continuación, se debe crear un archivo en base al lenguaje de descripción de hardware que se desea utilizar, para este caso, se utilizó Verilog. Para ello se deben seguir los siguientes pasos: en las pestañas de herramientas, seleccionar el símbolo de nuevo archivo (Fig. 9) y luego en **Design Files (Diseño de archivos)**, seleccionar **Verilog HDL File** (Fig. 10).

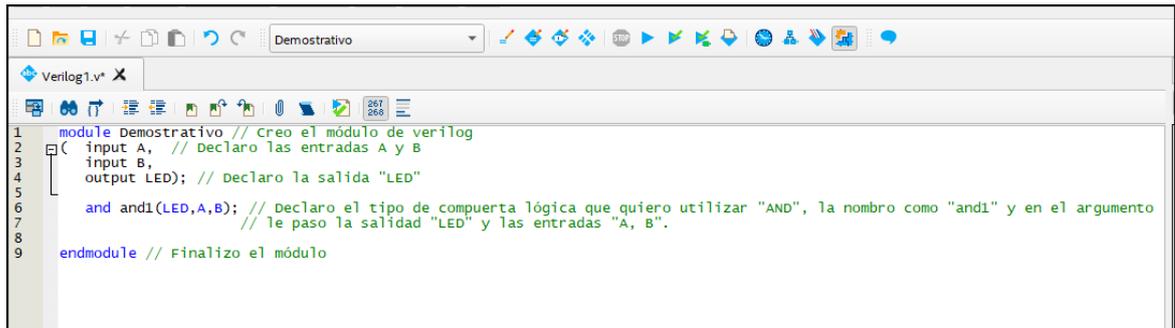


**Fig. 9. Pestaña para seleccionar nuevos archivos.**



**Fig. 10. Ubicación del archivo Verilog.**

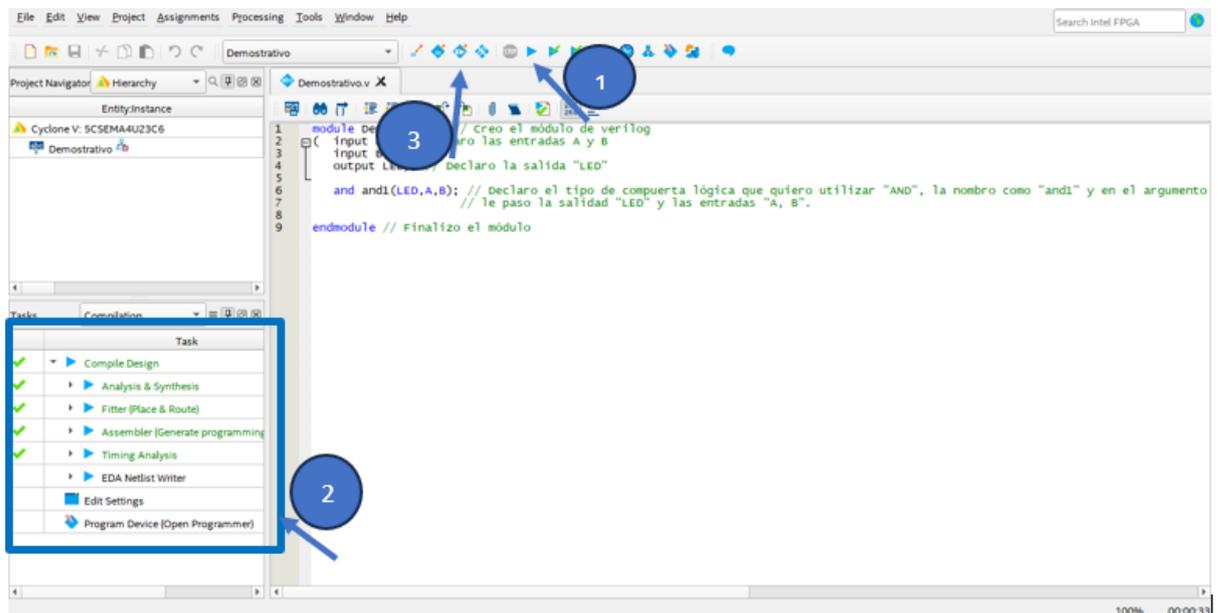
Consecuentemente se abrirá el archivo Verilog y se procede a escribir la descripción de hardware que, para una **AND**, es la que se observa en la **Fig. 11**. Es importante aclarar que el nombre del módulo debe ser el mismo que el proyecto, al igual que el nombre del archivo Verilog, caso contrario, producirá error y no compilará.



```
1 module Demostrativo // Creo el módulo de verilog
2 ( input A, // Declaro las entradas A y B
3   input B,
4   output LED); // Declaro la salida "LED"
5
6   and and1(LED,A,B); // Declaro el tipo de compuerta lógica que quiero utilizar "AND", la nombre como "and1" y en el argumento
7                       // le paso la salida "LED" y las entradas "A, B".
8
9 endmodule // Finalizo el módulo
```

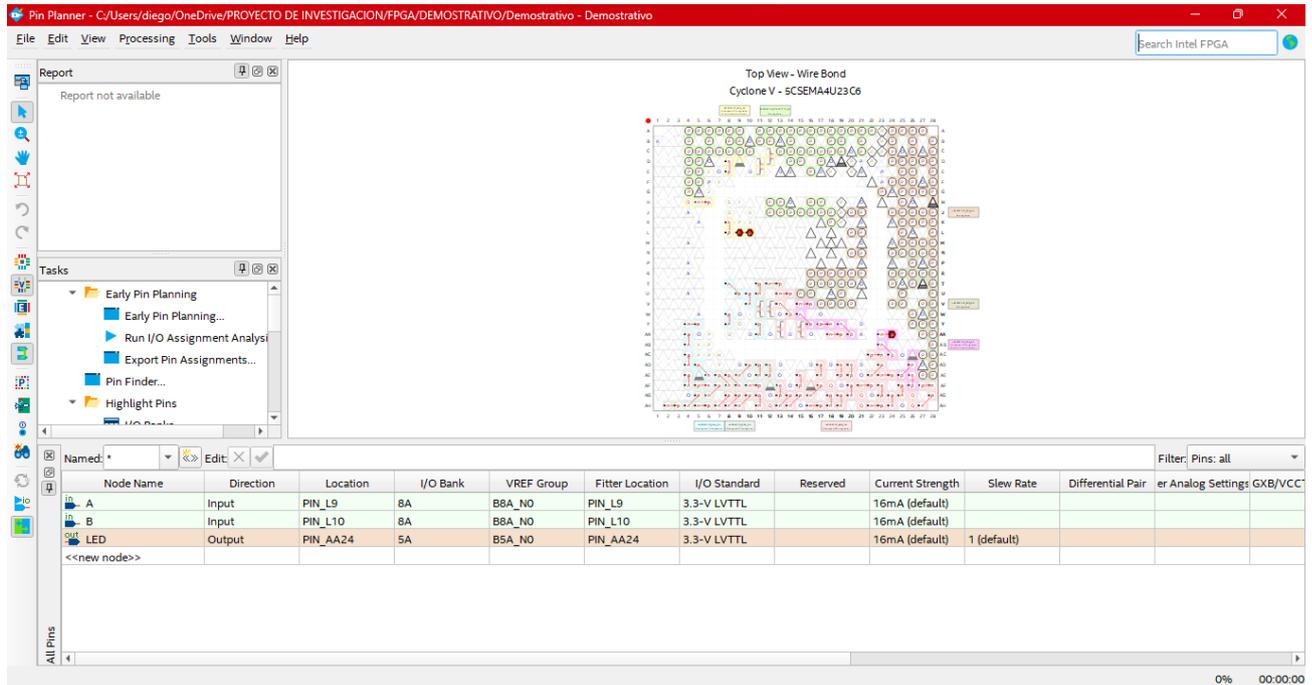
**Fig. 11. Descripción de Hardware para una compuerta AND.**

Una vez escrito el código se procede a guardar. Verificar que el módulo y el nombre del archivo Verilog sean el mismo que la del proyecto y la ubicación de éste se encuentre en el mismo directorio que el proyecto. Luego pulsar el botón que se indicar en el círculo "1" la **Fig. 12** (Start Compilation), y comenzará el proceso de compilado, tras unos segundos, sino hay errores se podrá observar el cuadro de tareas (**Task**) que indicará el compilado se completó de manera exitosa.



**Fig. 12. Proceso de compilado.**

Posteriormente que se haya completado el proceso de compilado, se procede a seleccionar el botón que se indica por el círculo 3, que se denomina **“Pin Planner (Planificador de Pines)”** con el cual, se deberá especificar los pines de la FPGA que se utilizarán. Para ello, es necesario basarse en el manual de la FPGA [3].



**Fig. 13. Planificador de Pines**

Como se aprecia en la **Fig. 13**, en la columna **“Location”** se debe ingresar el nombre de los pines que se utilizará en el ensayo, tanto las entradas como las salidas. Para ello, se puede observar la Fig. 14, cuya salida de la compuerta AND corresponde al **LED1** y Fig. 15 con las entradas **SW[0]** y **SW[1]**, obtenidas de la página 25 del manual.



**Fig. 14. Conexiones entre los LED y el FPGA.**

Signal Name	FPGA Pin No.	Description	I/O Standard
SW[0]	PIN_L10	Slide Switch[0]	3.3V
SW[1]	PIN_L9	Slide Switch[1]	3.3V
SW[2]	PIN_H6	Slide Switch[2]	3.3V
SW[3]	PIN_H5	Slide Switch[3]	3.3V

Fig. 15. Asignación de pines de interruptores deslizantes

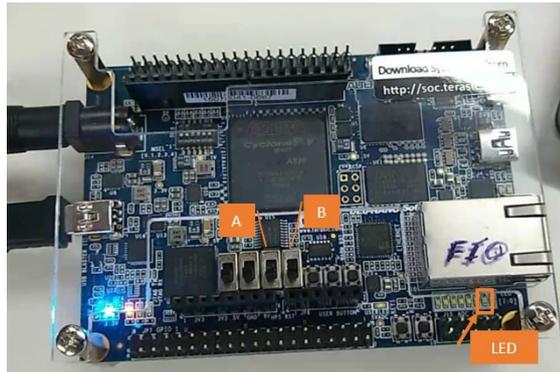


Fig. 16. Pines de entradas y salidas de la compuerta AND.

En la **Fig. 16** se puede observar las entradas, siendo **A** el **Switch 1** y **B** el **Switch 0** y salida el led que se indica en la imagen. Luego en **I/O Standard** de la **Fig. 13** se debe verificar que esté configurado en 3,3 V LVTTTL.

Una vez configurado los pines, se procede a cerrar la ventana “**Pin Planner**” y a hacer clic derecho en **Program Device (Open Programmer)** ubicada en el último renglón del cuadro “**Task**” como se indica en la **Fig. 12** y pulsar la opción **Open**, la cual, abrirá una nueva ventana, tal como se muestra en la **Fig. 17**.

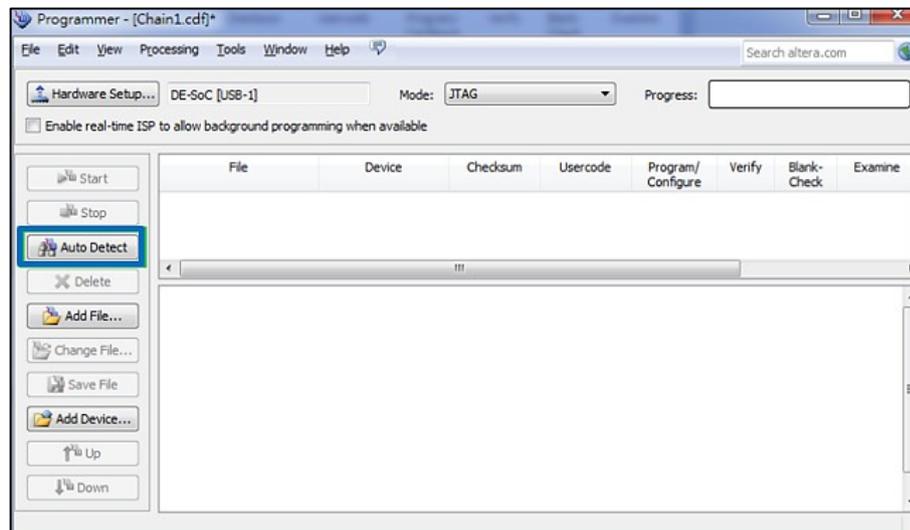


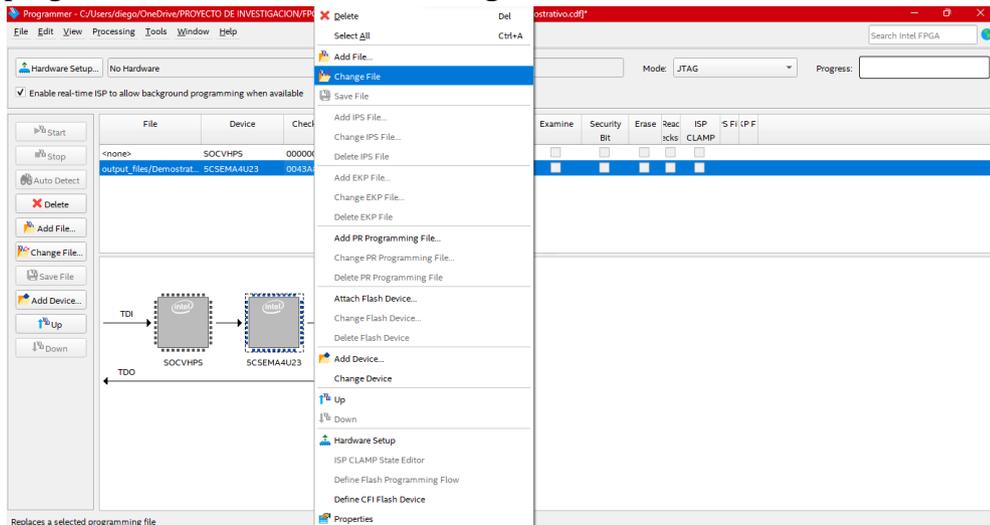
Fig. 17. Detectar dispositivo FPGA en modo JTAG.

Abierta la ventana, se debe seleccionar el dispositivo detectado asociado con la placa, como se indica en un círculo de la **Fig. 18**.

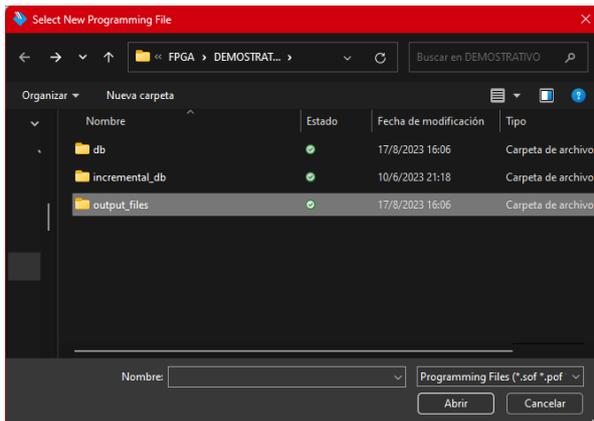


**Fig. 18. Seleccionar el dispositivo 5CSEMA4.**

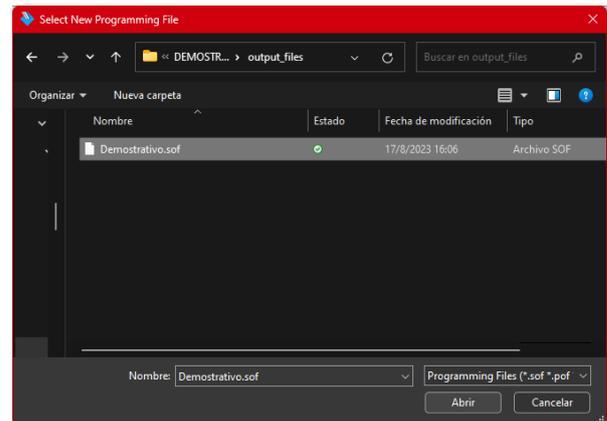
Luego, se debe seleccionar clic derecho en el dispositivo FPGA y proceder a abrir el archivo “.sof” para programarlo, como se resalta en la **Fig. 19**.



**Fig. 19. Abrir el archivo “.sof” para programarlo en el dispositivo FPGA.**



**Fig. 20. Seleccionar "output\_files".**



**Fig. 21. Seleccionar el archivo ".sof".**

Por último, se deben seguir los pasos de la Fig. 22, para que se programe la FPGA con el archivo “.sof”, si no se produce ningún error, la barra **Progress** (indicada por el círculo N°3), indicará mediante el color verde el 100 % de la carga del archivo a la FPGA.

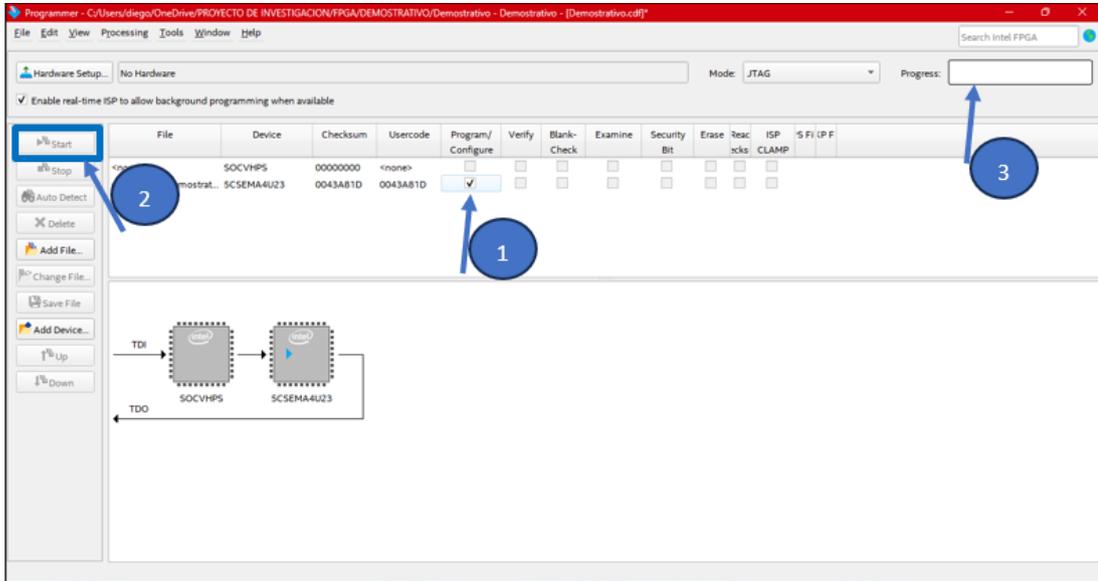


Fig. 22. Programar el archivo “.sof” en el dispositivo FPGA.

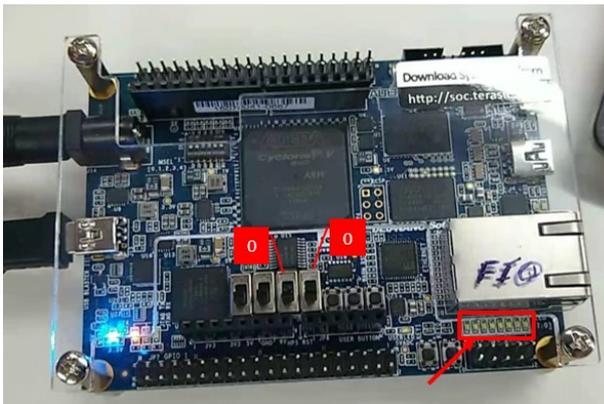


Fig. 23. Estado lógico, 0|0|0.

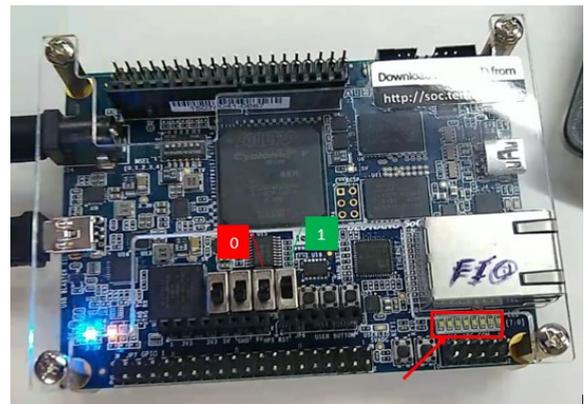
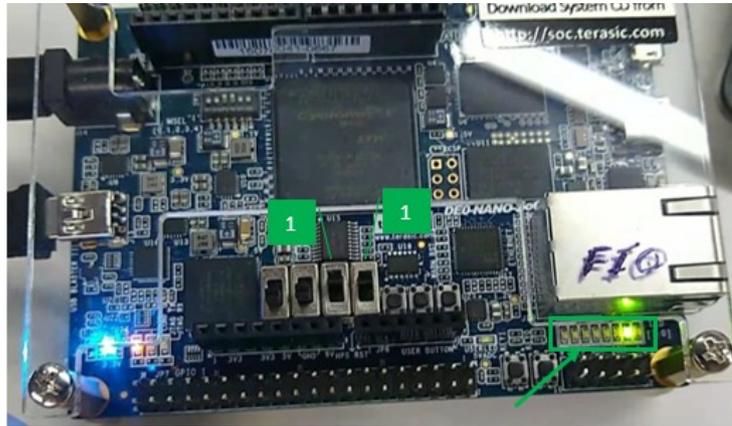


Fig. 24. Estado lógico, 0|1|0.

A continuación, se presentan los resultados obtenidos al configurar la FPGA con Quartus Prime – Lite Edition usando como lenguaje de descripción de hardware Verilog para implementar una compuerta AND de dos entradas.

En la Fig. 23 se muestra el estado lógico para las entradas aplicadas A=0 y B=0, obteniendo LED=0. En la Fig. 24 se observa que la entrada A se mantiene en 0 y B asume el estado 1, mientras que la salida LED permanece apagada.



**Fig. 25. Estado lógico 1|1|1.**

Sin embargo, en la Fig. 25, A y B son iguales a 1, por lo que la salida también es 1, y se aprecia que el LED1 se enciende.

### 3 Conclusión

Se detalla de manera didáctica el proceso de implementación de una compuerta AND en una FPGA, empleando el software Quartus Prime - Lite Edition y el lenguaje de descripción de hardware Verilog. Los pasos esenciales para programar una FPGA se presentan de forma coherente, estableciendo una base para proyectos más complejos que requieran tanto hardware como software avanzado. Esta herramienta se prevé útil no solo en cursos prácticos de programación de FPGAs, sino también en las investigaciones actuales de la FIO.

### 4 Referencias

- [1] Libro – Reconfigurable computing. The theory and practice of FPGA – Based computation – Scott Hauck and André Dehon.
- [2] <https://www.intel.com/content/www/us/en/software-kit/773998/intel-quartus-prime-lite-edition-design-software-version-22-1-1-for-windows.html>
- [3] <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=941&PartNo=4#contents>