

P3

Introducción a Vivado IDE, parte II: Implementación en FPGA

Objetivos

En la tercera práctica, el alumno entenderá las funcionalidades básicas del ambiente de desarrollo Vivado, para el desarrollo de firmware para FPGAs en el lenguaje VHDL, incluyendo la captura del código fuente, simulación, síntesis, verificación y programación del FPGA.

Planteamiento

En un programa de concursos de televisión, hay un juego en el que participan 3 personas. Cada participante cuenta con un interruptor; los interruptores cerrados (posición "ON") generan un "uno" lógico y los interruptores abiertos (posición "OFF") generan un "cero" lógico. Por otra parte, hay 4 indicadores (LEDs) en un tablero: A, B, C y D. Un indicador enciende si recibe un "uno" lógico. Tomar en cuenta las siguientes consideraciones:

- El indicador A debe encender sólo cuando todos los jugadores posicionen en "ON" sus interruptores.
- El indicador B debe encender si y sólo si A está en "OFF".
- El indicador C debe encender si y sólo si dos o más jugadores posicionan en "ON" sus interruptores.
- El indicador D debe encender si y sólo si uno o ninguno de los jugadores posicionan en ON sus interruptores.

Se generó la siguiente tabla de verdad y se encontraron las ecuaciones booleanas correspondientes:

P1	P2	P3	A	B	C	D
0	0	0	0	1	0	1
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	0	1	0	1	1	0
1	1	0	0	1	1	0
1	1	1	1	0	1	0

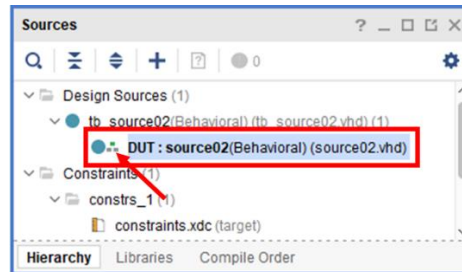
- $A = P1 \cdot P2 \cdot P3$
- $B = P1' + P2' + P3'$
- $C = P2 \cdot P3 + P1 \cdot P3 + P1 \cdot P2$
- $D = P1' \cdot P2' + P1' \cdot P3' + P2' \cdot P3'$

Resolver el problema implementando un programa en VHDL que resuelva la lógica necesaria para obtener las respuestas en las cuatro salidas a partir de las posibles combinaciones de entradas.

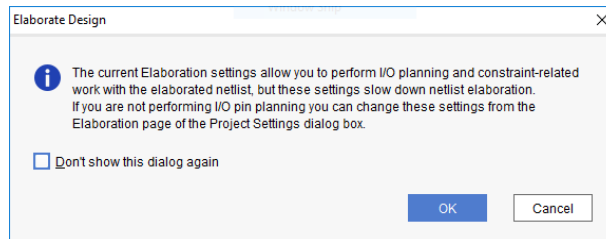
Procedimiento

Una vez completada la simulación de tu diseño, puedes proceder a la implementación en el FPGA.

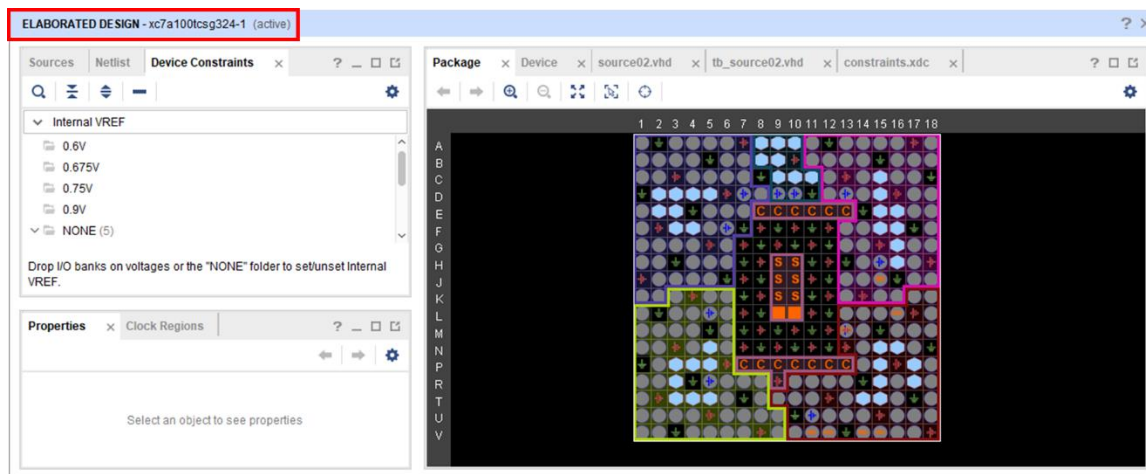
1. Es necesario deshabilitar el testbench programado para realizar la simulación del diseño. Para esto, en la ventana Sources, haz click derecho en tu DUT y selecciona la opción "Set as top". Debes ver el ícono de tres puntos para reconocer que el DUT ha sido asignado al tope de la jerarquía.



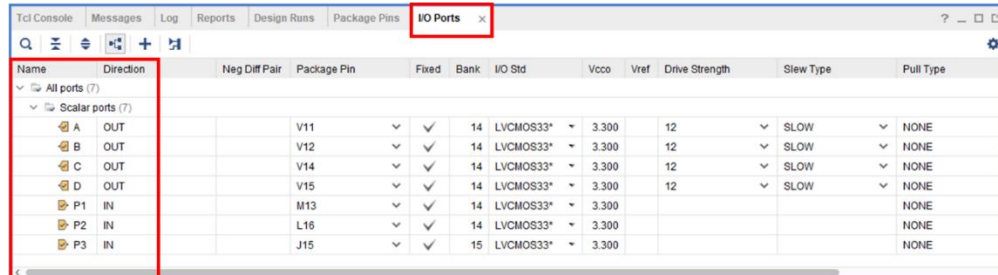
2. Bajo la pestaña de RTL ANALYSIS en el Flow Navigator, selecciona "Open Elaborated Design". Da click en "Ok" a la ventana emergente que aparecerá:



3. Se abrirá la ventana "ELABORATED DESIGN" para el FPGA que queremos programar, que es el xc7a100tcsg324-1:

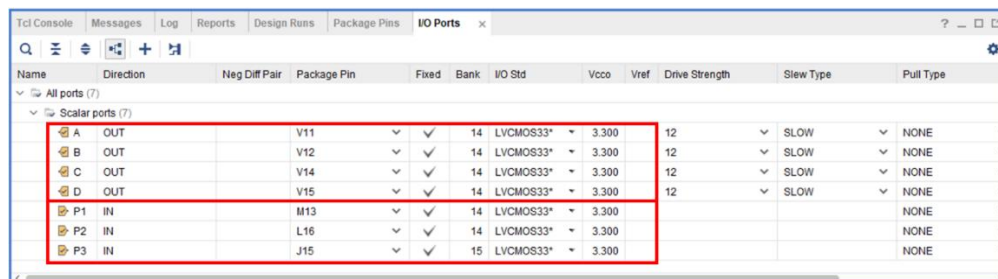


- En el menú "Layout" selecciona la opción I/O Planning. Se agregará la pestaña "I/O Ports" a tu espacio de trabajo, y en esta pestaña, encontrarás las entradas y salidas programadas en tu diseño, así como su direccionamiento:



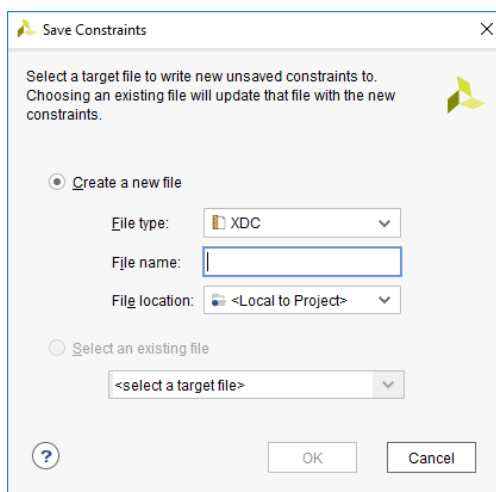
Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (7)											
Scalar ports (7)											
A	OUT		V11	✓		14	LVCMOS33*	3.300	12	SLOW	NONE
B	OUT		V12	✓		14	LVCMOS33*	3.300	12	SLOW	NONE
C	OUT		V14	✓		14	LVCMOS33*	3.300	12	SLOW	NONE
D	OUT		V15	✓		14	LVCMOS33*	3.300	12	SLOW	NONE
P1	IN		M13	✓		14	LVCMOS33*	3.300			NONE
P2	IN		L16	✓		14	LVCMOS33*	3.300			NONE
P3	IN		J15	✓		15	LVCMOS33*	3.300			NONE

- En esta ventana debemos mapear estas entradas y salidas del diseño, hacia la interfaz de interacción implementada en la tarjeta de desarrollo. En este caso, haremos uso de tres switches conectados a las entradas P1, P2 y P3, para dar entrada de datos; y usaremos 4 LEDs conectados a A, B, C y D, para ver la salida generada. En la columna "Package Pin", introduce V11, V12, V14 y V15, para mapearlos a las salidas A, B, C y D, respectivamente.
- También asigna los pines M13, L16 y J15 para mapearlos a las entradas P1, P2 y P3, respectivamente.
- Por último, asigna a todos los puertos el valor "LVCMOS33" en la columna "I/O Std". La información completa debe verse similar a la figura siguiente:

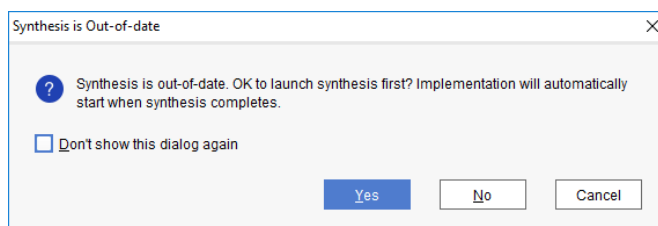


Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (7)											
Scalar ports (7)											
A	OUT		V11	✓		14	LVCMOS33*	3.300	12	SLOW	NONE
B	OUT		V12	✓		14	LVCMOS33*	3.300	12	SLOW	NONE
C	OUT		V14	✓		14	LVCMOS33*	3.300	12	SLOW	NONE
D	OUT		V15	✓		14	LVCMOS33*	3.300	12	SLOW	NONE
P1	IN		M13	✓		14	LVCMOS33*	3.300			NONE
P2	IN		L16	✓		14	LVCMOS33*	3.300			NONE
P3	IN		J15	✓		15	LVCMOS33*	3.300			NONE

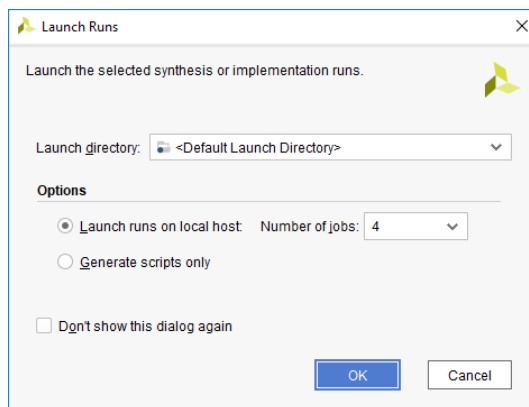
- Guardar el proyecto y aparecerá la ventana "Save Constraints" que pedirá guardar un nuevo archivo. En este archivo es donde se guardará la información del mapeo que se realizó en la pestaña I/O Ports. Asigna un nombre al archivo y haz click en "Ok".



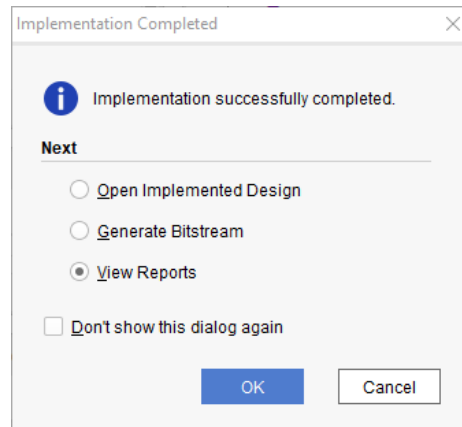
9. Una vez guardado el archivo de "constraints". Haz click en la opción "Run Implementation" bajo la pestaña IMPLEMENTATION del Flow Navigator. Si aparece la ventana "Synthesis is Out-of-Date", simplemente selecciona "Ok" para actualizar la síntesis.



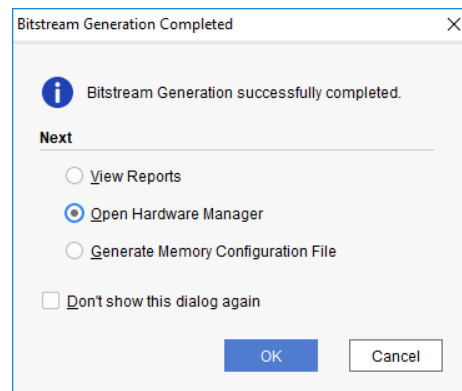
10. En la ventana Launch Runs, también haz click en "Ok" sin hacer ninguna modificación.



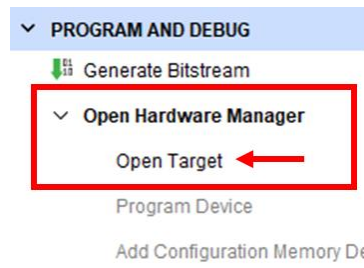
11. Espera unos segundos para completar la implementación. Al finalizar, deberás ver la notificación de que la implementación fue completada satisfactoriamente.



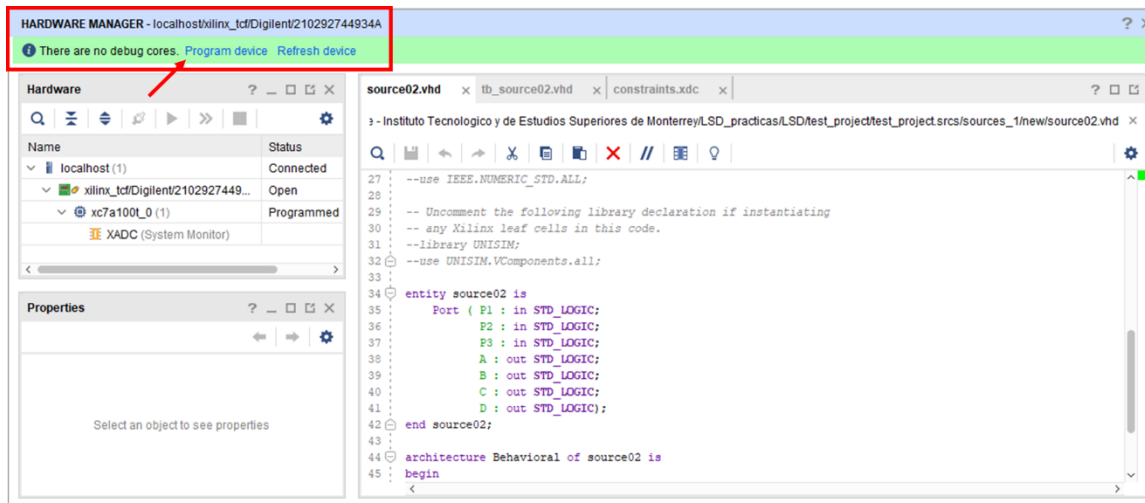
12. Bajo la pestaña "Program and Debug" en el Flow Navigator, selecciona "Generate Bitstream", nuevamente, selecciona "Ok" en la ventana "Launch Runs" que aparecerá. Espera unos segundo para completar la generación del bitstream. Al finalizar el proceso, deberás ver una ventana de notificación de generación de bitstream satisfactoria. Selecciona la opción "Open Hardware Manager" y haz click en "Ok".



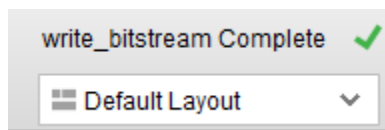
13. Bajo la misma pestaña de "Program and Debug", selecciona la opción "Open Target" y selecciona "Auto connect" en el menú emergente.



14. Finalmente, en la ventana HARDWARE MANAGER, tendrás la opción "Program device". Selecciona esta opción para programar el FPGA con tu diseño y selecciona el bitstream creado.



15. En la parte superior derecha de la pantalla, aparecerá el mensaje “write_bitstream Complete”, y el FPGA estará ya programado con tu diseño.



Entregables

Los entregables deben ser enviados al final de la parte 2 de la práctica. Se entrega **un solo reporte** en incluya tu trabajo tanto de la parte 1 como de la parte 2 de la práctica. Los entregables son:

- Código/ fuente en VHDL (GitHub)
- Demostración de implementación en FPGA (Video)
- Reporte técnico (PDF en Canvas)
 - Esquemático RTL (paso 15 de la parte 1)
 - Evidencias de simulación con capturas de pantalla para las diferentes combinaciones de entradas.
 - Enlaces a repositorio en GitHub y video demostrativo