

Sistemas Digitales

Práctica 3

Agregando IP personalizada al Sistema

(Adaptado del curso “Embedded System Design Flow” de Xilinx)

Introducción

Esta práctica lo guiará a través del proceso de crear y agregar un periférico personalizado al sistema de procesamiento usando IP Packager de Vivado. Creará un periférico con interfaz AXI4 Lite.

Objetivos

Después de completar esta práctica será capaz de:

- Usar el IP Packager de Vivado para crear un periférico personalizado
- Modificar la funcionalidad de la IP
- Agregar el periférico al diseño
- Agregar restricciones de ubicación de pines
- Agregar un bloque de memoria al sistema

Procedimiento

Esta práctica está separada en pasos que consisten en sentencias generales que proveen información sobre las instrucciones detalladas que le siguen. Siga estas instrucciones detalladas para avanzar dentro de esta práctica.

Esta práctica está compuesta por 4 pasos principales: Usará una plantilla de periférico para crear uno propio, utilizando el IP Packager, importará, agregará y conectará la IP en el diseño, y agregará una memoria Block RAM.

Descripción del Diseño

El propósito de esta práctica es extender el diseño de hardware (Figura 1) creado en la práctica 2.

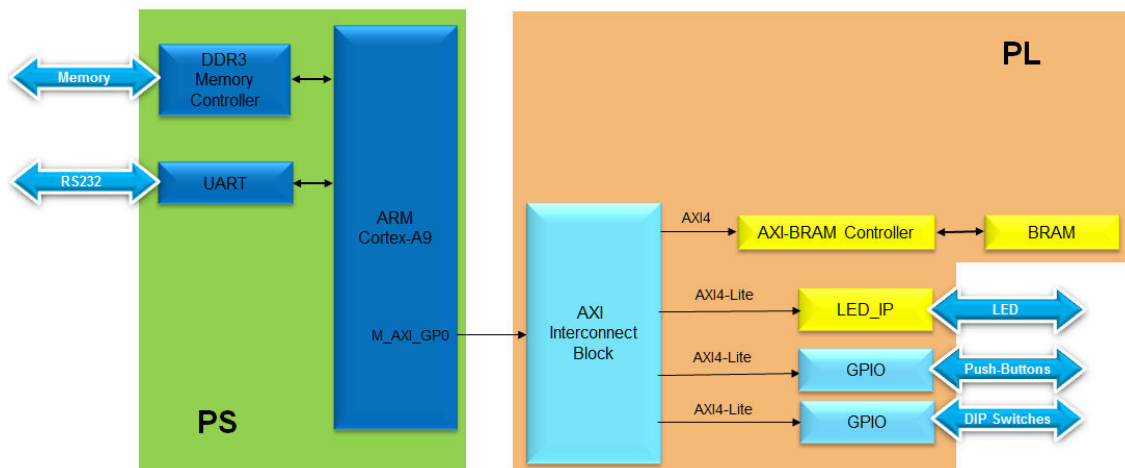
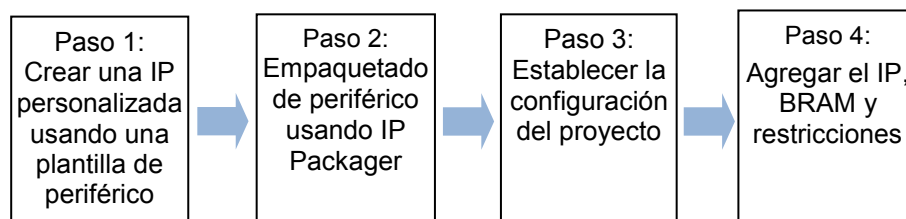


Figura 1. Extensión del sistema de la práctica anterior

Flujo General para esta práctica



Crear un Custom IP usando Create and Package IP Wizard Paso 1

1-1. Usar el template de periférico esclavo provisto axi_lite y el código fuente del custom IP para crear un custom IP.

1-1-1. Abrir Vivado seleccionando **Start ► Xilinx Design Tools ► Vivado 2018.1**

1-1-2. Hacer click sobre **Manage IP** y seleccionar *New IP Location* y clicar **Next** en la ventana **New IP Location**.

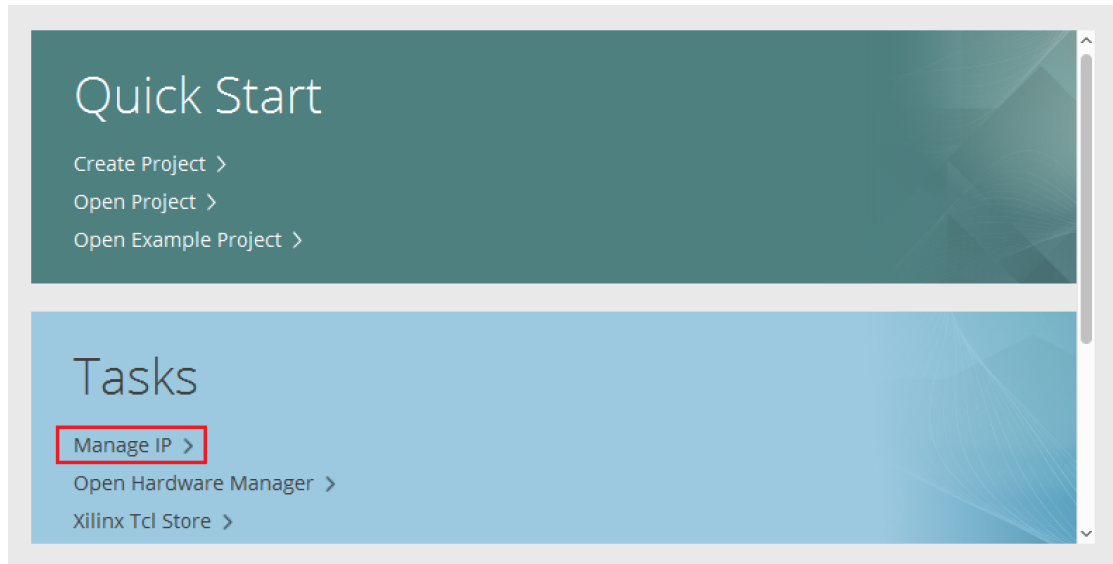


Figura 2. Selección para manejar IPs

1-1-3. Seleccionar la placa correspondiente, **Verilog** como *Target Language*, y **Mixed** como *Simulator language*. Para el campo *IP Location* elegir una ubicación como por ejemplo el lugar donde se encuentran todas las prácticas y crear una carpeta `led_ip`. Presionar **Finish**. Si el directorio no existe se solicitará la creación. Presionar **Aceptar**.

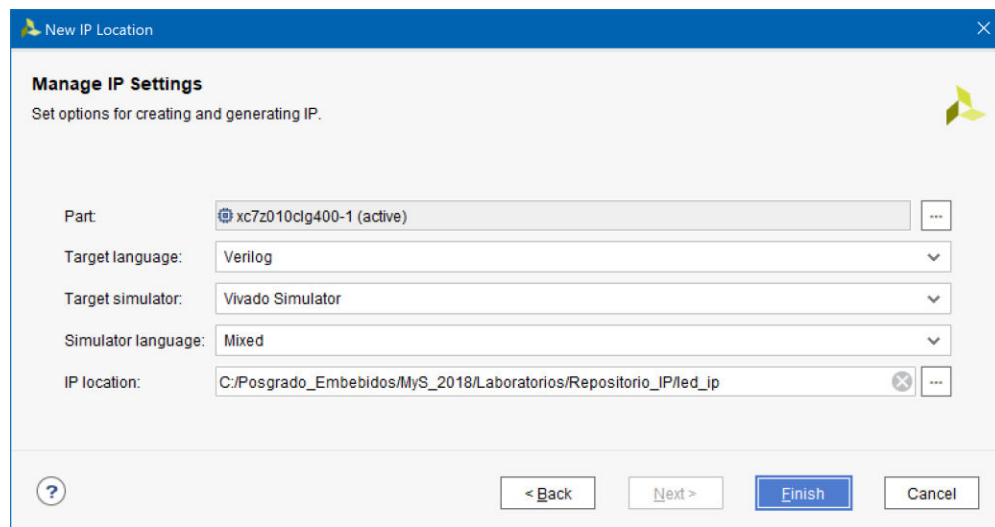


Figura 3. Configuración de la IP (ubicación, HDL, etc)

1-2. Run the Create and Package IP Wizard

1-2-1. Seleccionar **Tools ► Create and Package New IP ...**

1-2-2. En la ventana presionar **Next**.

1-2-3. Seleccionar *Create a new AXI4 peripheral*, y presionar **Next**.

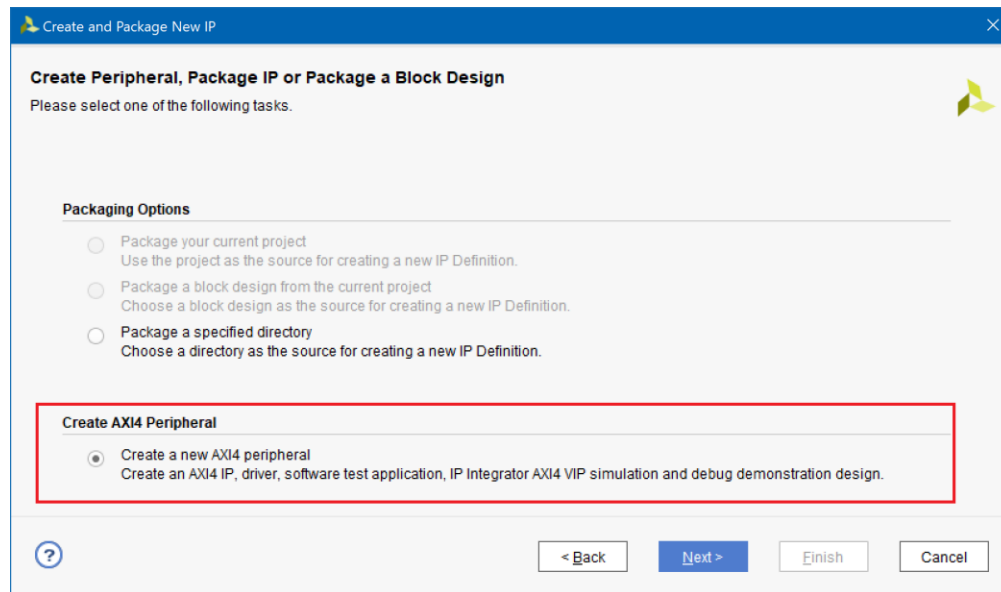


Figura 4.

1-2-4. Completar los detalles para la IP

Name: **led_ip**

Display Name: **led_ip_v1_0**

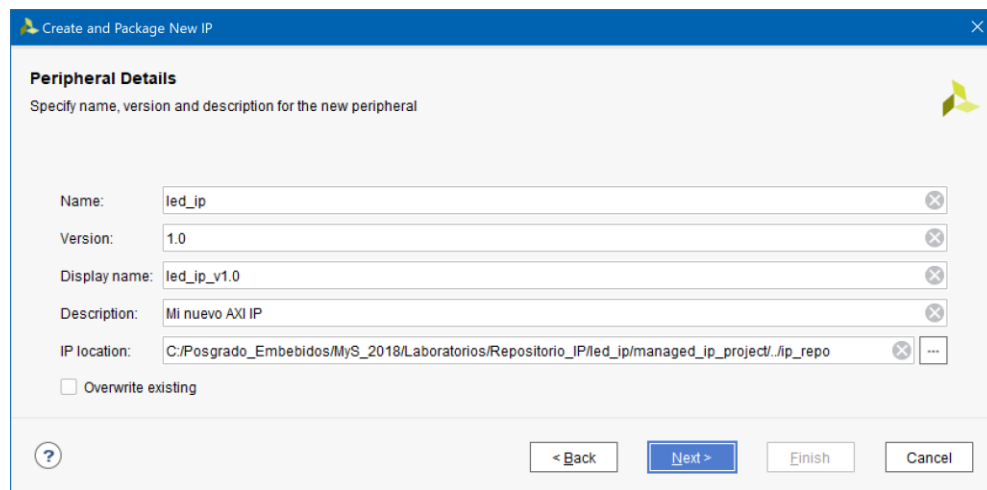


Figure 5. Actualización de los datos del periférico

1-2-5. Presionar **Next**

1-2-6. Cambiar el nombre de la interfaz a **S_AXI**. Dejar el resto de las configuraciones por defecto y hacer click en **Next**.

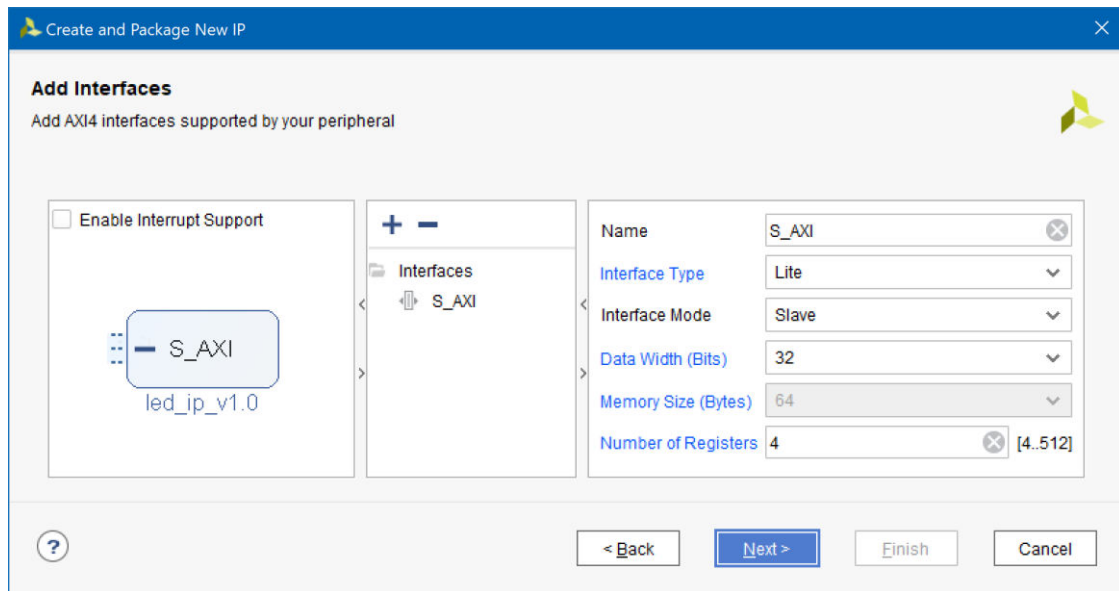


Figure 6. Nombrando la interfaz AXI

1-2-7. Seleccionar **Edit IP** y hacer click en **Finish** (se abrirá un nuevo proyecto Vivado)

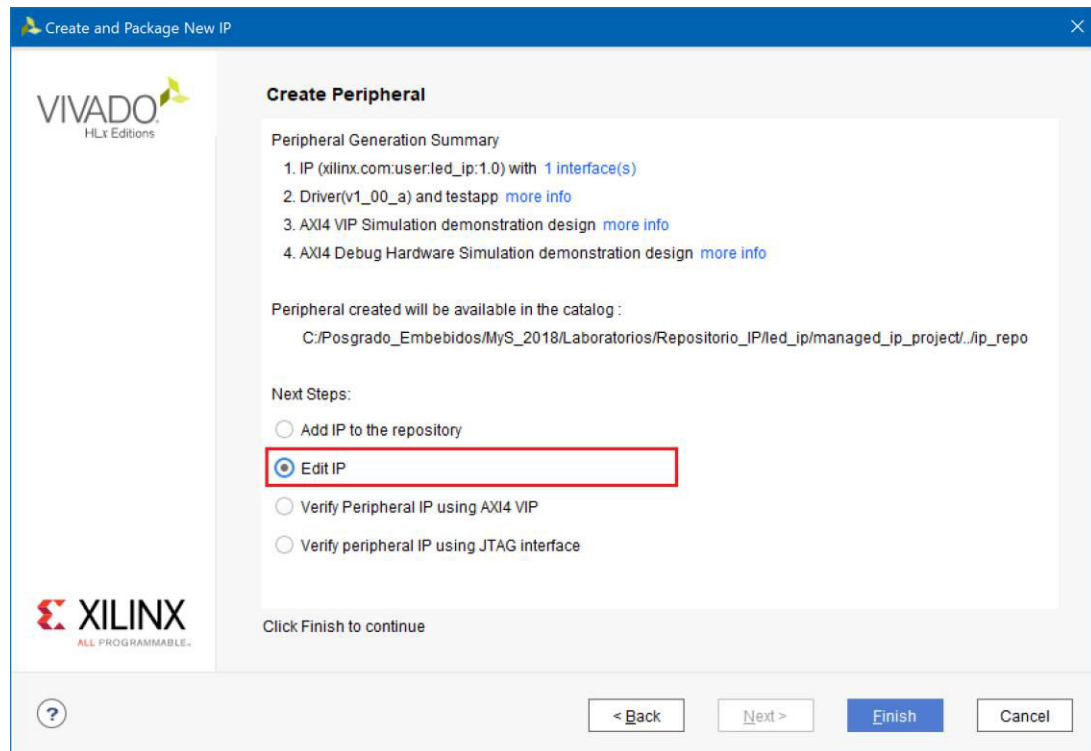


Figure 7. Selección para editar la IP

1-3. Crear una interfaz hacia los LEDs

1-3-1. En el panel *Sources* hacer doble click sobre el archivo **led_ip_v1_0.v**.

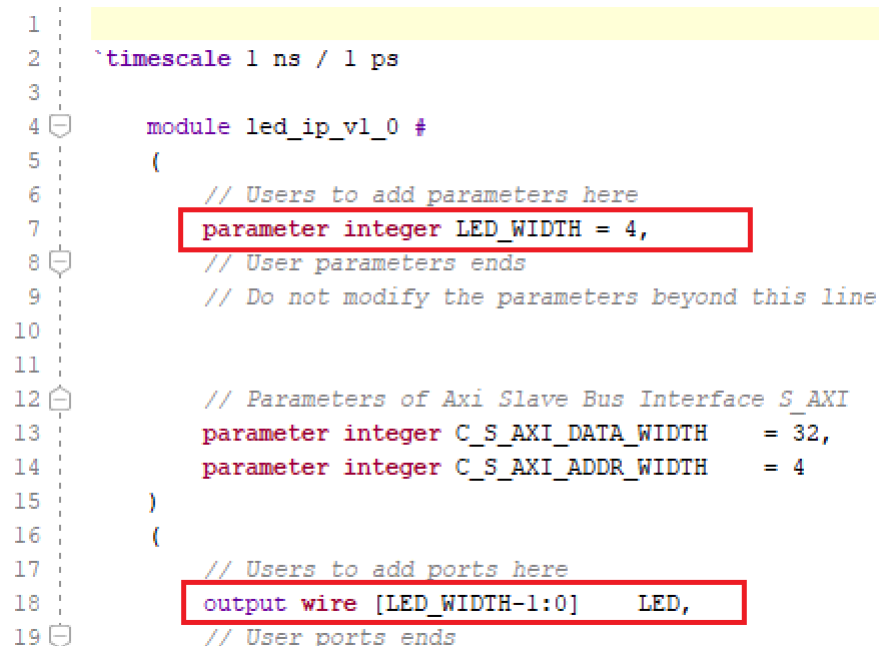
Este archivo contiene el código HDL para la interfaz seleccionada arriba. El archivo top level contiene un modulo que implementa la lógica de interfaz AXI, y un ejemplo de diseño para escribir hacia y leer desde un número de registros especificados arriba. Esta plantilla puede ser usada como base para crear la IP personalizada. Será creado un nuevo puerto de salida parametrizable hacia los leds en el top level del diseño, y el AXI write data en el sub-módulo será conectada al puerto externo LED.

1-3-2. En la línea 7 agregar:

parameter integer LED_WIDTH = 4,

1-3-3. En la línea 18 agregar:

output wire [LED_WIDTH-1:0] LED,



```

1 |
2 | `timescale 1 ns / 1 ps
3 |
4 | module led_ip_v1_0 #
5 | (
6 |     // Users to add parameters here
7 |     parameter integer LED_WIDTH = 4,
8 |     // User parameters ends
9 |     // Do not modify the parameters beyond this line
10 |
11 |
12 |     // Parameters of Axi Slave Bus Interface S_AXI
13 |     parameter integer C_S_AXI_DATA_WIDTH    = 32,
14 |     parameter integer C_S_AXI_ADDR_WIDTH    = 4
15 | )
16 | (
17 |     // Users to add ports here
18 |     output wire [LED_WIDTH-1:0] LED,
19 |     // User ports ends

```

Figure 8. Agregando parámetros de usuario y definiciones de puerto

1-3-4. Insertar lo siguiente en la línea 48:

.LED_WIDTH(LED_WIDTH),

1-3-5. Insertar lo siguiente en la línea 52:

.LED(LED),

```

40     output wire  s_axi_arready,
41     output wire [C_S_AXI_DATA_WIDTH-1 : 0] s_axi_rdata,
42     output wire [1 : 0] s_axi_rresp,
43     output wire  s_axi_rvalid,
44     input wire  s_axi_rready
45 );
46 // Instantiation of Axi Bus Interface S_AXI
47 led_ip_v1_0_S_AXI # (
48     .LED_WIDTH(LED_WIDTH),
49     .C_S_AXI_DATA_WIDTH(C_S_AXI_DATA_WIDTH),
50     .C_S_AXI_ADDR_WIDTH(C_S_AXI_ADDR_WIDTH)
51 ) led_ip_v1_0_S_AXI_inst (
52     .LED(LED),
53     .S_AXI_ACLK(s_axi_aclk),
54     .S_AXI_ARESETN(s_axi_aresetn),
55     .S_AXI_AWADDR(s_axi_awaddr),
56     .S_AXI_AWPROT(s_axi_awprot),
57     .S_AXI_AWVALID(s_axi_awvalid),
58     .S_AXI_AWREADY(s_axi_awready),

```

Figure 9. Agregando conexiones a puerto con un módulo de nivel menor

1-3-6. Guardar el archivo.

1-3-7. Expandir `led_ip_v1_0` en la vista `sources`, y abrir `led_ip_v1_0_S_AXI.v`

1-3-8. Agregar el parámetro LED y el puerto también a este archivo, en las líneas 7 y 18

```

1
2 `timescale 1 ns / 1 ps
3
4 module led_ip_v1_0_S_AXI #
5 (
6     // Users to add parameters here
7     parameter integer LED_WIDTH = 4,
8     // User parameters ends
9     // Do not modify the parameters beyond this line
10
11     // Width of S_AXI data bus
12     parameter integer C_S_AXI_DATA_WIDTH = 32,
13     // Width of S_AXI address bus
14     parameter integer C_S_AXI_ADDR_WIDTH = 4
15 )
16 (
17     // Users to add ports here
18     output wire [LED_WIDTH-1:0] LED,
19     // User ports ends

```

Figura 10. Declarando puertos de usuario en el módulo de nivel inferior

- 1-3-9.** Insertar el siguiente código en la línea 400 para instanciar la lógica de usuario (*user logic*) para el ip LED. El código puede ser tipeado directamente o copiado del archivo `user_logic_instantiation.txt` provisto.

```

400      // Add user logic here
401      lab3_user_logic # (
402          .LED_WIDTH(LED_WIDTH)
403      )
404      U1(
405          .S_AXI_ACLK(S_AXI_ACLK),
406          .slv_reg_wren(slv_reg_wren),
407          .axi_awaddr(axi_awaddr[C_S_AXI_ADDR_WIDTH-1:ADDR_LSB]),
408          .S_AXI_WDATA(S_AXI_WDATA),
409          .S_AXI_ARESETN(S_AXI_ARESETN),
410          .LED(LED)
411      );
412      // User logic ends

```

Figura 11. Instanciación del módulo de usuario de bajo nivel

Verificar todas las señales que se están conectando y dónde se originan.

- 1-3-10.** Guardar el archivo.

- 1-3-11.** Hacer click sobre *Add Sources* en el panel *Flow Navigator*, seleccionar *Add or Create Design Sources*, hacer click en **Next**, luego click en **Add Files...**, navegar donde se encuentren los archivos auxiliares, seleccionar el archivo **lab3_user_logic.v** y hacer click en **OK**, y luego en **Finish** para agregarlo.

Verificar el contenido de este archivo para entender la lógica que se está implementando. Notar la jerarquía creada.

- 1-3-12.** Hacer click sobre **Run Synthesis** y guardar si se lo solicita (esto es para verificar que el diseño sintetiza correctamente antes de empaquetar la IP. Si en este paso uno estuviera realizando un diseño propio se debería realizar una simulación para verificar la funcionalidad antes de proceder a los siguientes pasos).

- 1-3-13.** Verifique la pestaña de Mensajes buscando errores para corregirlos, si fuera necesario, antes de seguir adelante

Cuando la síntesis se complete satisfactoriamente haga click en **Cancel**.

1-4. Empaquetar la IP (Package)

- 1-4-1.** Hacer click sobre la pestaña **Package IP – led_ip**

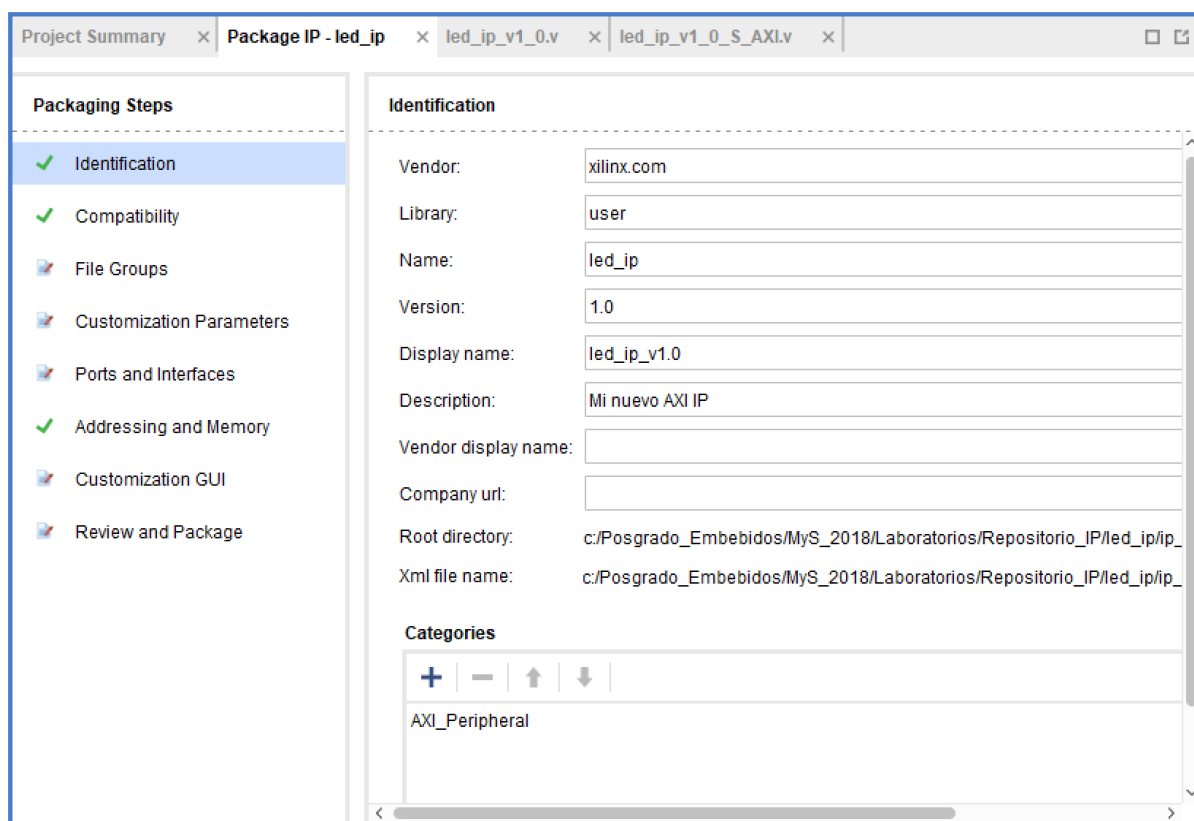


Figure 12. Package IP

- 1-4-2.** Para que la IP aparezca en el catálogo de IP en una categoría particular, debe ser configurada como parte de esa categoría. Para cambiar las categorías en las que aparecerá la IP hacer click en el símbolo **+** en la sección **Categories**. Esto abre la ventana Choose IP Categories.
- 1-4-3.** Sólo para el propósito de este ejercicio, eliminar el tilde de **AXI Peripheral**, marcar **Basic Elements** y hacer click en **OK**.

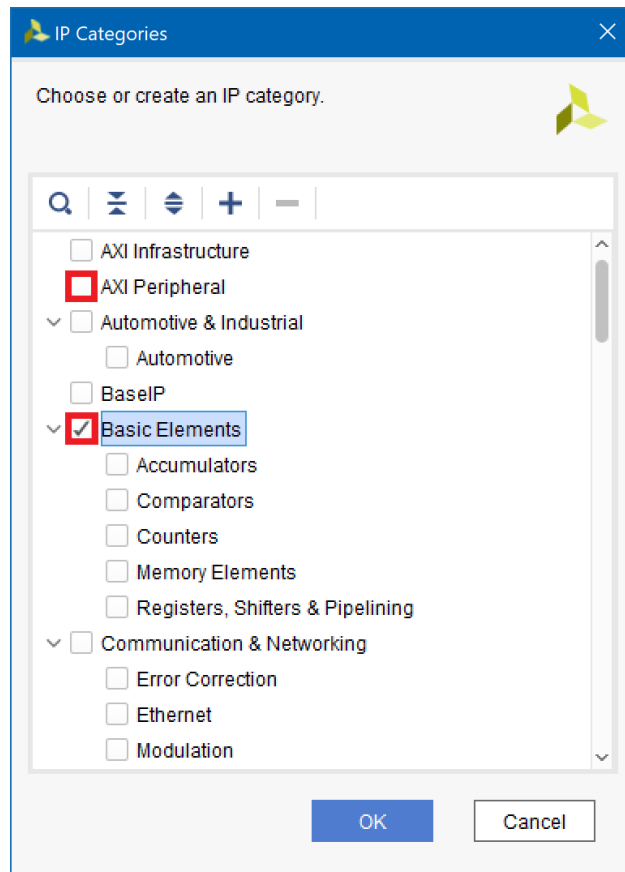


Figure 13. Especificación de la categoría de la IP

- 1-4-4. Seleccionar **Compatibility**. Esto muestra las diferentes familias de FPGAs de Xilinx que soporta la IP. El valor es heredado del dispositivo seleccionado para el proyecto.
- 1-4-5. Hacer click en el símbolo + y luego en **Add Family Explicitly...**
- 1-4-6. En el caso de no aparecer, seleccionar la familia **Zynq** ya que es la familia de nuestra FPGA y hacer click en **OK**.
- 1-4-7. Se puede personalizar el espacio de direcciones y agregar espacio de direcciones de memoria usando la categoría **Addressing and Memory**. No realizaremos ningún cambio en este sentido.
- 1-4-8. Hacer click sobre **File Groups** y luego sobre *Merge changes from File Groups Wizard*.

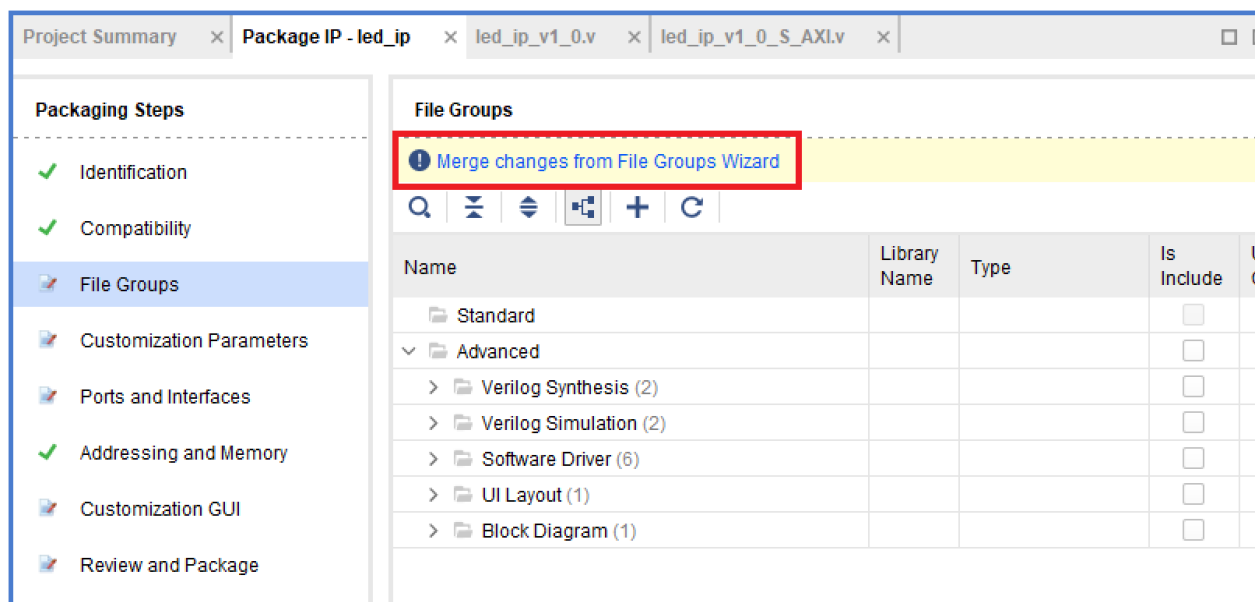


Figure 14. Actualizando file group

Esto se realiza para actualizar el IP Packager con los cambios realizados al IP y al archivo *lab3_user_logic_*.v* que fue agregado al proyecto.

1-4-9. Expandir *Verilog Synthesis* y notar que ha sido incluido el **lab3_user_logic.v**

1-4-10. Hacer click sobre **Customization Parameters** y nuevamente sobre *Merge changes from Customization Parameters Wizard*.

Notar que la vista *Ports and Interfaces* muestra ahora el puerto creado *LED*

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- ✓ File Groups
- ✓ Customization Parameters
- ✓ Ports and Interfaces
- ✓ Addressing and Memory

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- ✓ File Groups
- ✓ Customization Parameters
- ✓ Ports and Interfaces
- ✓ Addressing and Memory
- ✓ Customization GUI
- Review and Package

Customization Parameters

Name	Description	Display Name	Value	Value Length
Customization Parameters				
C_S_AXI_DATA_WIDTH	Width of S_AXI data bus	C S AXI DATA WIDTH	32	0
C_S_AXI_ADDR_WIDTH	Width of S_AXI address bus	C S AXI ADDR WIDTH	4	0
C_S_AXI_BASEADDR		C S AXI BASEADDR	0xFFFFFFFF	32
C_S_AXI_HIGHADDR		C S AXI HIGHADDR	0x00000000	32
Hidden Parameters				

Ports and Interfaces

Name	Interface Mode	Enablement Dependency	Is Declaration	Access Handle	Access Type	Direction	Drive Value
S_AXI	slave		☐				
Clock and Reset Signals			☐				
LED			☐		ref	out	

Figura 15. Puertos y parámetros de usuario

1-4-11. Seleccionar **Customization Parameters**, expandir *Hidden Parameters*, hacer botón-derecho sobre **LED_WIDTH**, y seleccionar *Import IP Parameters...* y hacer click en **OK**.

1-4-12. Seleccionar **Customization GUI** y notar que *Led Width* es visible.

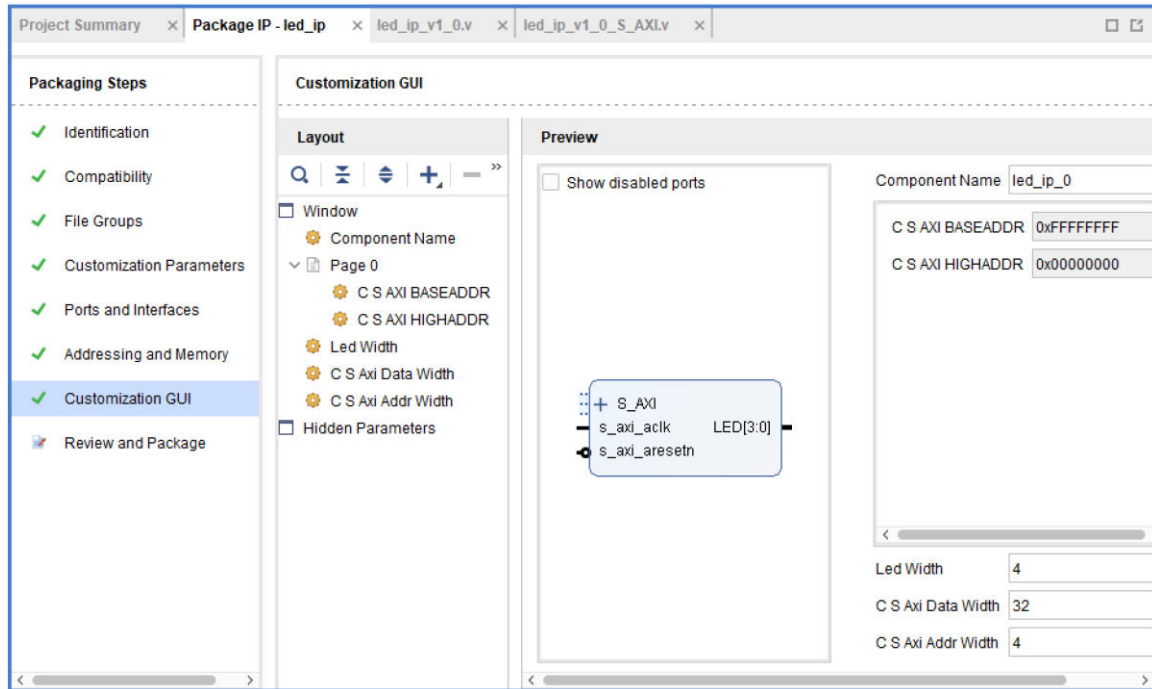


Figura 16. Personalización de parámetros

1-4-13. Seleccionar *Review and Package*, y notar el path donde la IP será creada.

1-4-14. Hacer click en **Re-Package IP**. Luego hacer click en **Yes** (el proyecto se cerrará al finalizar).

1-4-15. En la ventana original de Vivado hacer click en **File ► Close Project**.

Modificar la configuración del proyecto

Paso 2

2-1. Abrir el proyecto previo y guardarlo como lab3. Configurar el proyecto para que apunte al repositorio IP creado.

2-1-1. Abrir el proyecto lab2 y guardarlo como lab3 (**File ► Project ► Save As ...**).

2-1-2. Hacer click en *Settings* en el panel *Flow Navigator*.

2-1-3. Seleccionar **IP ► Repository** en el panel izquierdo de la ventana de configuración del proyecto.

2-1-4. Hacer click en el símbolo **+** y navegar hasta el directorio led_ip donde se estableció el repositorio. Aparecerá un mensaje indicando que se agregó un repositorio al proyecto. Presionar **OK**.

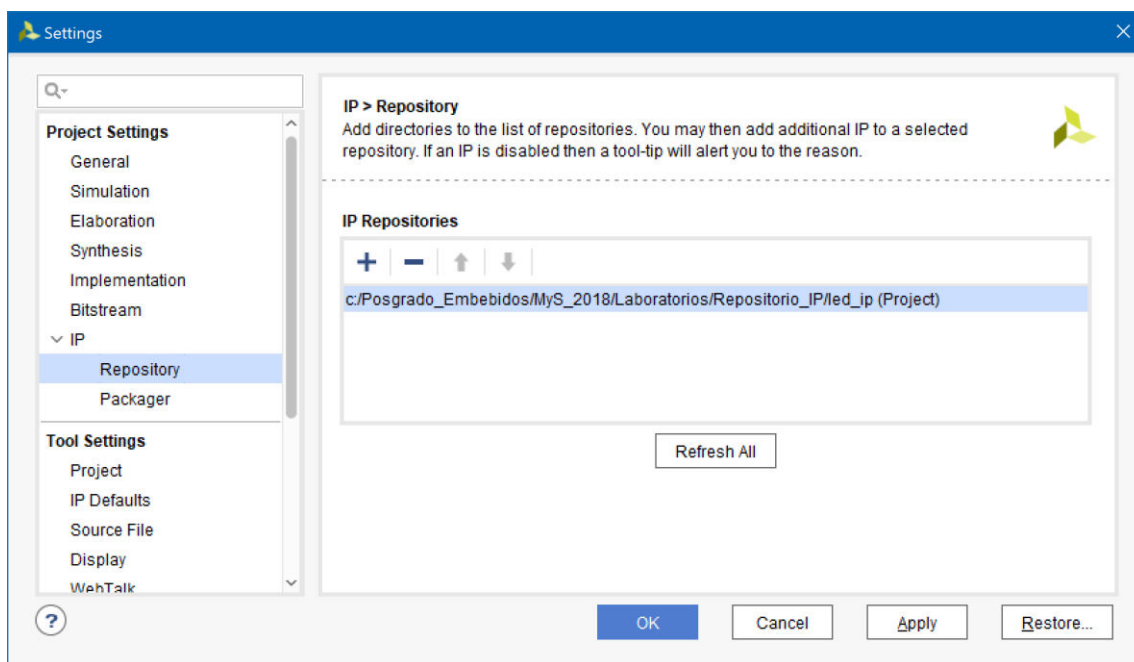


Figura 17. Ubicación del repositorio de IP

2-1-5. Hacer click en **OK**.

Agregar la IP personalizada, una BRAM y las restricciones Paso 3

3-1. Agregar led_ip al diseño y conectarla al AXI4 Lite interconnect en el IPI. Realizar las conexiones de puerto internas y externas. Establecer el puerto LED como pines externos de la FPGA.

3-1-1. Hacer click sobre **Open Block Design** debajo de **IP Integrator** en el panel *Flow Navigator*.

3-1-2. En la ventana *Diagram* hacer botón derecho y seleccionar **Add IP ...** y buscar led_ip_v1.0 en el catálogo colocando led en el campo de búsqueda.

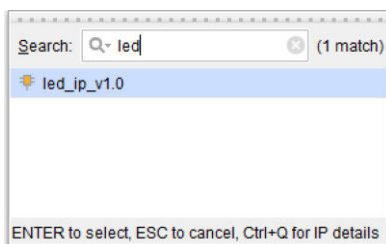


Figura 18. Búsqueda de la IP

3-1-3. Hacer doble click sobre led_ip_v1.0 para agregarla al diseño.

3-1-4. Seleccionar la ip y cambiar el nombre de instancia a led_pin en **Block Properties**.

3-1-5. Hacer doble click sobre el bloque para abrir la configuración y dejar sin modificación la existente.

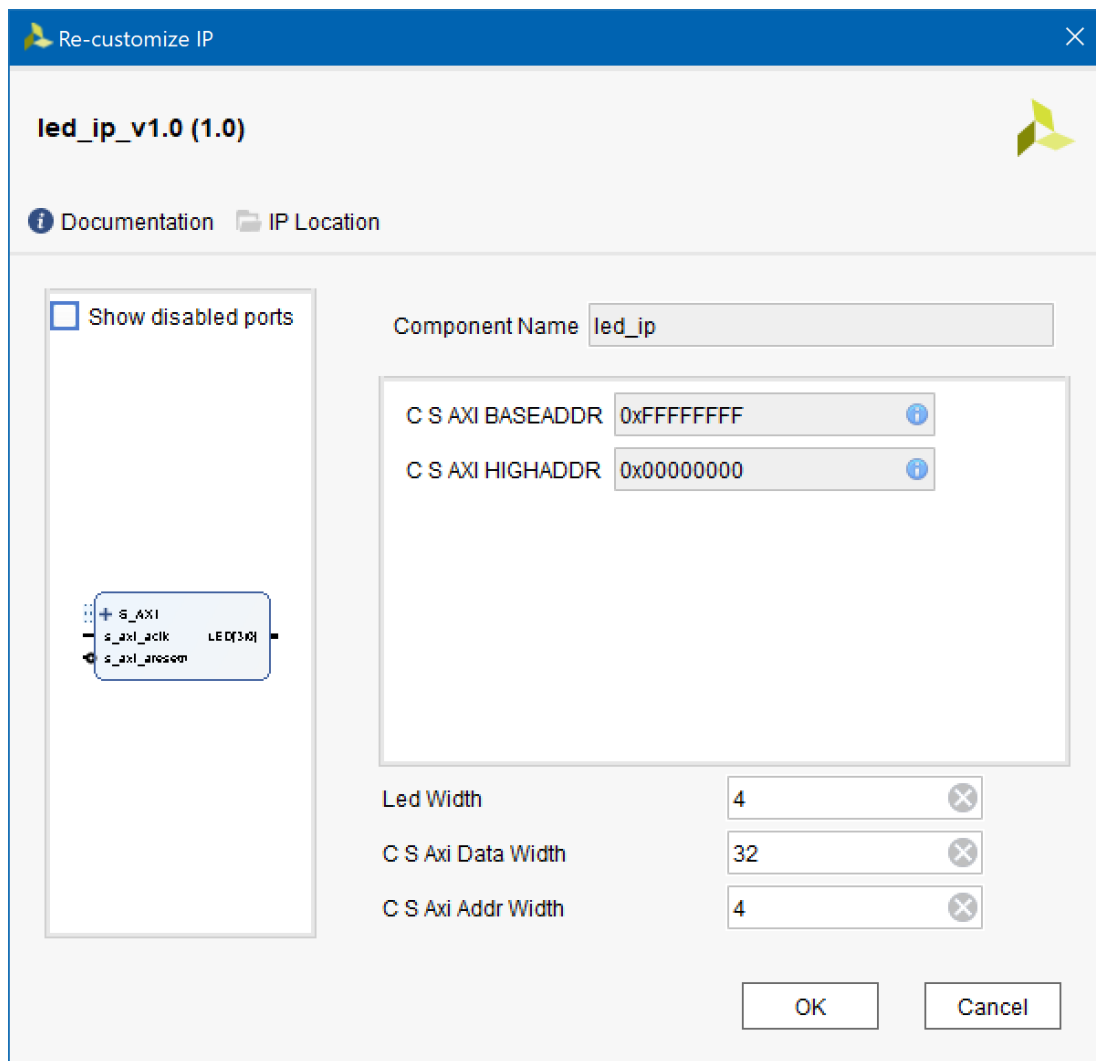


Figura 19. Ventana de configuración de la IP

3-1-6. Hace click en **Run Connection Automation**, seleccionar /led_ip/S_AXI y hacer click en **OK** para realizar la conexión automática desde el AXI Interconnect a la IP.

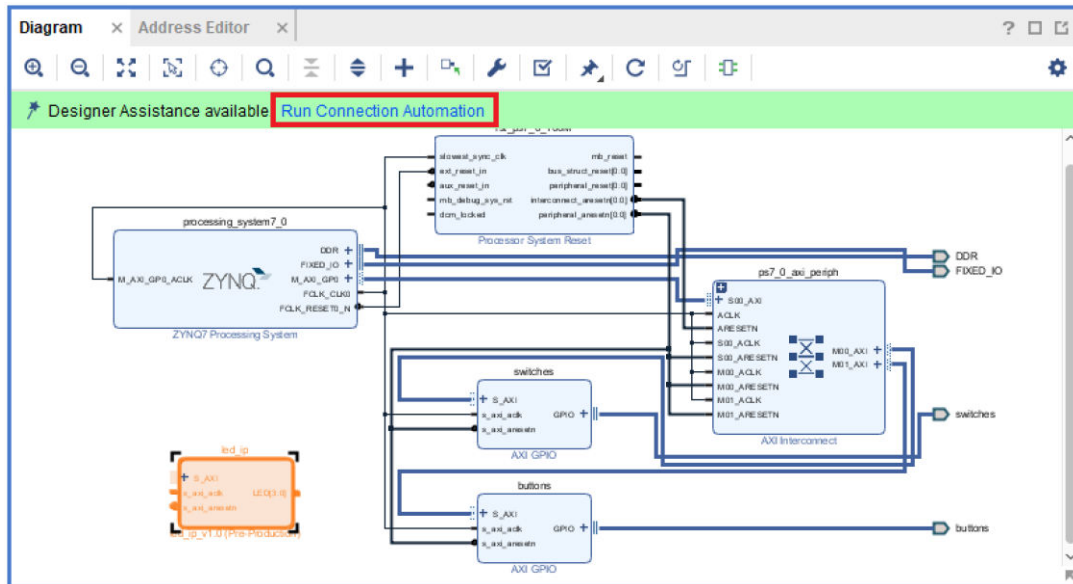


Figura 20. Ejecución de la conexión automática

3-1-7. Hacer click en el botón **Regenerate Layout** para redibujar el diagrama.

3-1-8. Seleccionar el puerto LED en la instancia led_ip (haciendo click en su pin), hacer click derecho y seleccionar **Make External**.

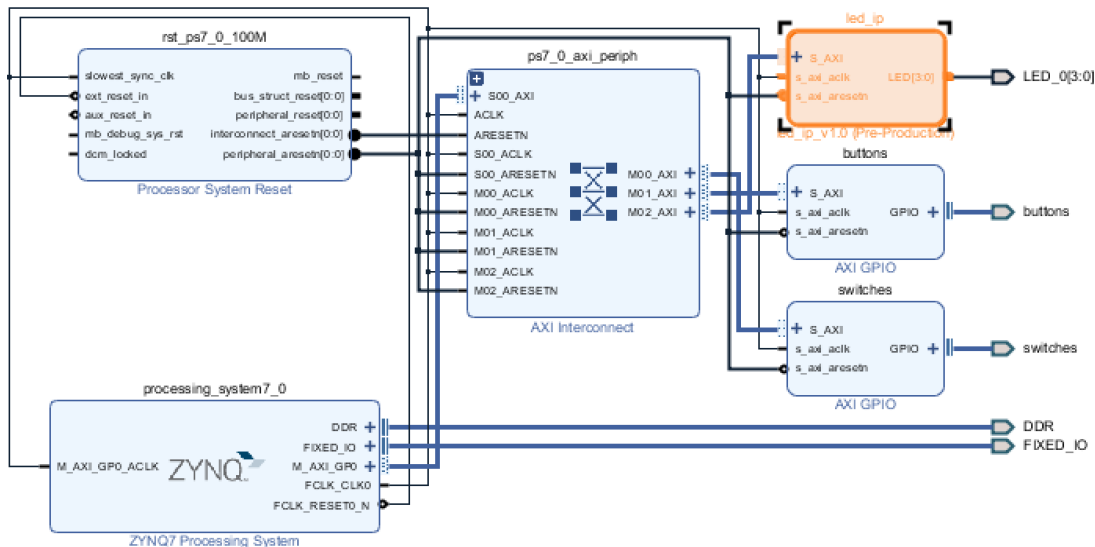


Figura 21. Haciendo externo el puerto LED

3-1-9. Seleccionar la pestaña **Address Editor** y verificar que ha sido asignada una dirección al led_ip.

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 0x40000000 [1G])					
buttons	S_AXI	Reg	0x4121_0000	64K	0x4121_FFFF
switches	S_AXI	Reg	0x4120_0000	64K	0x4120_FFFF
led_ip	S_AXI	S_AXI_reg	0x43C0_0000	64K	0x43C0_FFFF

Figura 22. Asignación de una dirección a la IP agregada

3-2. Agregar BRAM al diseño.

- 3-2-1. En el *Block Diagram*, agregar nueva IP, en este caso buscar BRAM y agregar una instancia de *AXI BRAM Controller*.
- 3-2-2. Hacer click en **Run Connection Automation** sobre axi_bram_ctrl_0/S_AXI.
- 3-2-3. Hacer doble click en el bloque para personalizarlo y cambiar el número de interfaces BRAM a 1 y hacer click.

Notar que el protocolo AXI usado es AXI4 en vez de AXI4Lite ya que BRAM puede proveer un ancho de banda más grande y el controlador puede soportar transacciones en ráfaga (burst).

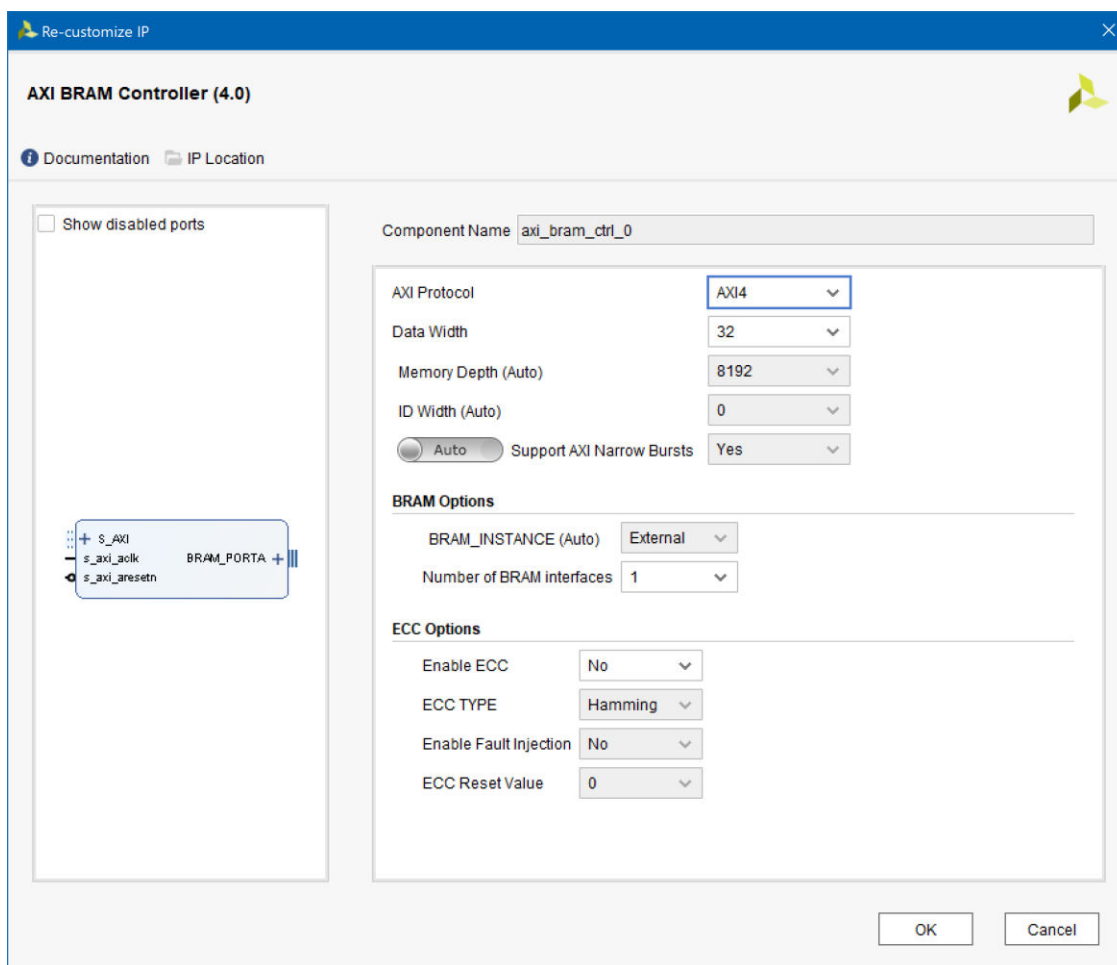


Figura 23. Búsqueda de la IP

- 3-2-4. Hacer click sobre **Run Connection Automation** para agregar y conectar un *Block Memory Generator* seleccionando **axi_bram_ctrl_0/BRAM_PORTA**, hacer click en **OK** (esto podría ser agregado manualmente).
- 3-2-5. Validar el diseño para asegurarse que no hay errores (F6), y hacer click en el botón **regenerate** para redibujar el diagrama. El diseño debería verse similar al de la figura que sigue.

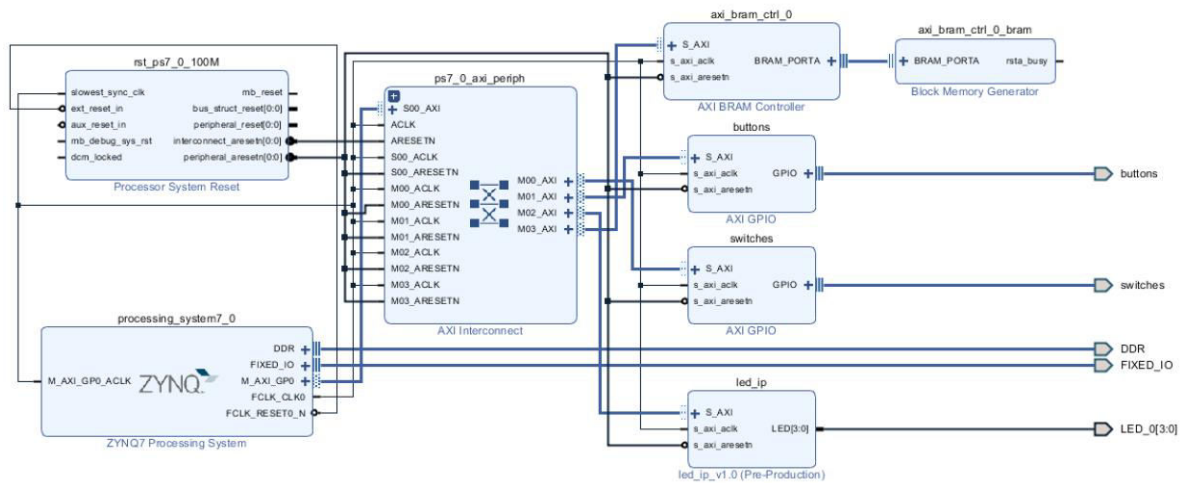


Figura 26. Diseño redibujado

3-2-6. En la pestaña **Address Editor** notar que el rango del axi_bram_ctrl_0 es 8K. Dejaremos eso sin modificación.

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 0x40000000 [1G])					
buttons	S_AXI	Reg	0x4121_0000	64K	0x4121_FFFF
switches	S_AXI	Reg	0x4120_0000	64K	0x4120_FFFF
led_ip	S_AXI	S_AXI_reg	0x43C0_0000	64K	0x43C0_FFFF
axi_bra...	S_AXI	Mem0	0x4000_0000	8K	0x4000_1FFF

Figura 27. Dirección asignada a axi_bram_ctrl_0

3-3. Agregar el archivo de restricciones provisto lab3_*.xdc.

3-3-1. Hacer click en **Add Sources** en el panel *Flow Navigator*, seleccionar **Add or Create Constraints**, y hacer click en **Next**.

3-3-2. Hacer click en el botón **Add Files**, navegar hasta la ubicación del archivo de restricciones, seleccionar el archivo provisto **lab3_*.xdc**.

3-3-3. Hacer click en **Finish** para agregar el archivo.

3-3-4. Expandir la carpeta *Constraints* en el panel *Sources*, y hacer doble click en el archivo **lab3_*.xdc** para visualizar su contenido. Este archivo contiene la ubicación de pines y el estándar de entrada/salida de los leds de la placa. Esta información se puede encontrar en la hoja de datos del fabricante de la placa.

- 3-3-5.** Hacer click derecho sobre *system.bd* y seleccionar *Generate output products*. Presionar **Generate**.
- 3-3-6.** Hacer click en **Generate Bitstream** y presionar **OK**. Hacer click en **Cancel** cuando se pregunte si se desea abrir el diseño implementado (*Open the Implemented Design*).

Conclusión

Vivado IP packager fue usado para importar un bloque de IP personalizado dentro de la librería de IP. El bloque IP fue agregado al sistema. Fue ejecutada la conexión automática cada vez que se disponía de ella para acelerar el diseño del sistema permitiendo a Vivado hacer sus conexiones de manera automática. Fue agregada al diseño una BRAM adicional. Finalmente, se agregaron al diseño restricciones de ubicación de pines.