

Controlar IP en PL desde PS

Introducción

En esta práctica se agregará IP instanciada en el bloque PS y la misma se controlará por software desde el PS.

Objetivos

- Agregar IP en el bloque PL de la FPGA.
- Configurar el GP Master port del bloque PS para conectarlo a la IP en el bloque PL
- Ejecutar una aplicación de test para probar el funcionamiento de la IP agregada

Procedimiento

Crear el nuevo Proyecto

1. Abrir el proyecto anterior y guardarlo como **Lab2PS** para ello seleccionar el menu **File -> Project -> Save As...** Verificar que la opción **Create Project Subdirectory** este seleccionada. Presionar **OK**. Se creará el nuevo proyecto **Lab2PS** en su correspondiente directorio.
2. Abrir el diseño de hardware **Flow Navigator -> IP Integrator -> Open Block Design**.

Agregar tres instancias del bloque IP GPIO

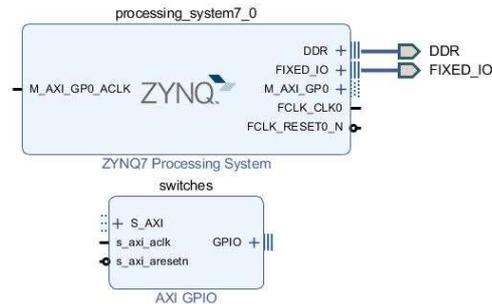
1. Dar doble click en el bloque Zynq block para abrir la ventana de configuración.
2. Seleccionar el menu **PS-PL Configuration** (ubicado a la izquierda de la ventana).
3. Expandir la opción **AXI Non Secure Enablement -> GP Master AXI Interfaces**, dar click en el recuadro **M_AXI_GP0 interface** para habilitar el puerto AXI GP0.
4. Expandir la opción **General -> Enable Clock Resets** y seleccionar la opción **FCLK_RESET0_N**.
5. Seleccionar el menu **Clock Configuration** (ubicado a la izquierda de la ventana). Expandir la opción **PL Fabric Clocks** y seleccionar la opción **FCLK_CLK0** (con **requested clock frequency** de 100.000000 MHz) y presionar **OK**.
6. En el bloque Zynq se agrego la interfaz **M_AXI_GPO** y las señales **M_AXI_GPO_ACLK**, **FCLK_CLK0**, and **FCLK_RESET0_N**



Bloque Zynq con las interfaces AXI y CLK

8. Presionar el boton derecho del raton en el diagrama y seleccionar **Add IP**. Buscar el bloque **AXI GPIO** en el catalogo.

- Dar doble click en el bloque **AXI GPIO** para agregarlo al sistema. En el diagrama aparecera el nuevo bloque. Ubicarlo en un lugar conveniente del diagrama (p.ej. debajo del bloque Zynq).
- Presionar sobre el bloque **AXI GPIO** para seleccionarlo. En el panel **Block Properties** cambiar el nombre de **axi_gpio_0** a **Switches**

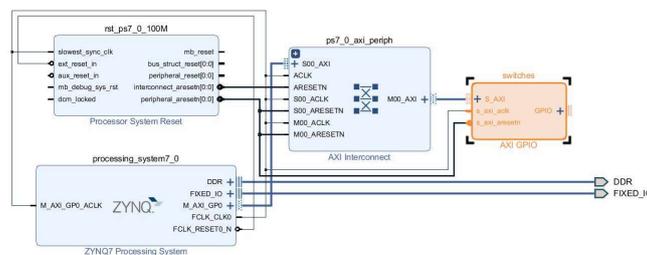


Bloque AXI GPIO "Switches"

- Dar doble click en el bloque **AXI GPIO** para abrir la ventana de configuración **Re-customize IP**.
- En el menu desplegable **Board Interface** correspondiente a **GPIO**, seleccionar **sws 2bits** (la placa Arty Z7-20 tiene dos interruptores).
- Cambiar a la pestaña **IP configuration** y verificar que **GPIO Width** esta configurado en "2", que coincide con la cantidad de interruptores de la placa.

El periférico puede configurarse para dos canales, pero dado que se utilizará un solo canal sin interrupciones, dejar sin chequear las opciones **Enable Dual Channel** y **Enable Interrupt**.

- Presionar **OK** para guardar la configuración y cerrar la ventana.
- En la parte superior de la ventana se activo la opcion **Designer assistance available**. Presionar en **Run Connection Automation** y seleccionar **/switches/S_AXI**
- Presionar **OK** para conectar la interfaz.



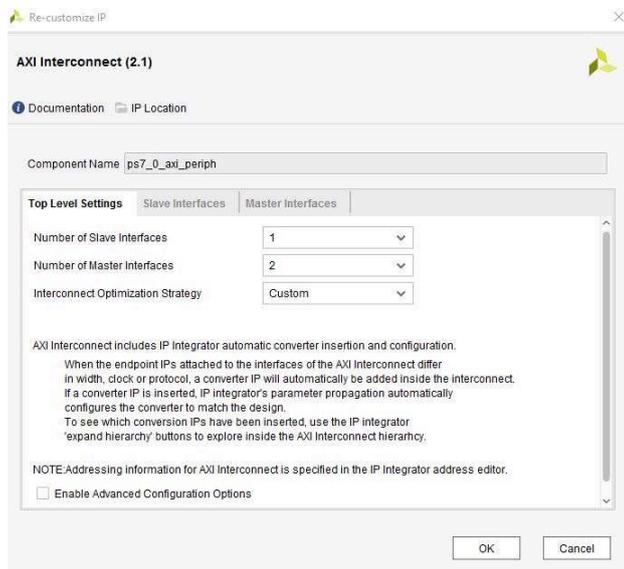
Sistema con los interruptores conectados

En el diagrama se agregaron dos bloques adicionales: **Processor System Reset** y **AXI Interconnect**. Se puede reordenar la ubicación de los bloques para mayor claridad en las conexiones.

- Agregar otra instancia del periférico **GPIO** (Mediante **Add IP**). Cambiar el nombre a **buttons**
- Dar doble click en el nuevo bloque **buttons**, seleccionar en **Board Interface** la opción **btns_4bits**. Presionar **OK**.

El conexionado se puede hacer en forma automatica como en el bloque **switches** o en forma manual. En este caso lo haremos en forma manual.

- Dar doble click en el bloque **AXI Interconnect** (su nombre es **ps7_0_axi_periph**) y cambiar el numero en **Master Interfaces** de 1 a 2 y presionar **OK**.



Agregar otro master port al bloque AXI Interconnect

21. Presionar sobre la conexión **S_AXI** del bloque AXI GPIO **buttons** y arrastrar el lapiz al bloque **AXI Interconnect** (no soltar el boton del raton).

Aparecera el mensaje '**Found 1 interface**' y un tilde verde al lado de la conexión **M01_AXI** del bloque **AXI Interconnect** indicando que esa es una conexión valida. Arrastrar el lapiz a esa conexión y soltar el boton del raton. Debera crearse una conexión entre el bloque **AXI Interconnect** y el bloque GPIO **buttons** .

22. De la misma manera hacer las siguientes conexiones:

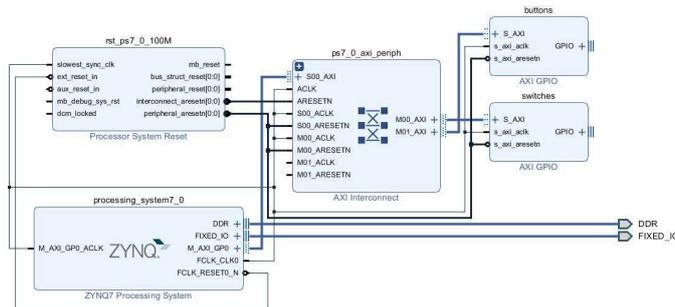
buttons / s_axi_aclk -> Zynq7 Processing System / FCLK_CLK0

buttons / s_axi_arenstn -> Processor System Reset / peripheral_arenstn

AXI Interconnect / M01_ACLK -> Zynq7 Processing System / FCLK_CLK0

AXI Interconnect / M01_ARESETN -> Processor System Reset / peripheral_arenstn

The block diagram should look similar to this:



Sistema despues de agregar el segundo bloque AXI GPIO

23. Agregar un tercer bloque AXI GPIO de la misma manera que el bloque **buttons**, cambiar el nombre a **Leds**.
24. Dar doble click en el nuevo bloque **leds**. Seleccionar la pestaña **IP configuration**, seleccionar **All Outputs** y en **GPIO Width** cambiar de 32 a 6. Presionar **OK**
25. Dar doble click en el bloque **AXI Interconnect** y agregar una interfaz master adicional (cambiar el numero en **Master Interfaces** de 2 a 3). Presionar **OK**.
26. Hacer las siguientes conexiones:

leds / S_AXI -> AXI Interconnect / M02_AXI

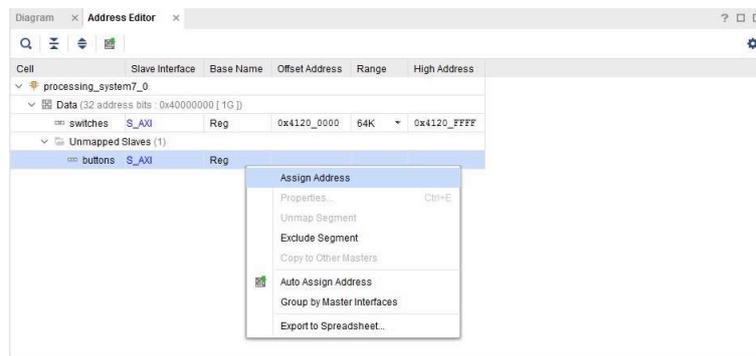
leds / s_axi_aclk -> Zynq7 Processing System / FCLK_CLK0

leds / s_axi_aresetn -> Processor System Reset / peripheral_aresetn

AXI Interconnect / M02_ACLK -> Zynq7 Processing System / FCLK_CLK0

AXI Interconnect / M02_ARESETN -> Processor System Reset / peripheral_aresetn

27. En el panel del diagrama, presionar en la pestaña **Address Editor** y expandir **processing_system7_0 -> Data -> Unmapped Slaves**.
28. El bloque **switches** tiene asignado un rango de direcciones en forma automática, pero los bloques **buttons** y **leds** no, ya que se conectaron manualmente. Presionar sobre la primera entrada debajo de **Unmapped Slaves** para seleccionarla; presionar el botón derecho del ratón y seleccionar la opción **Assign Address**.

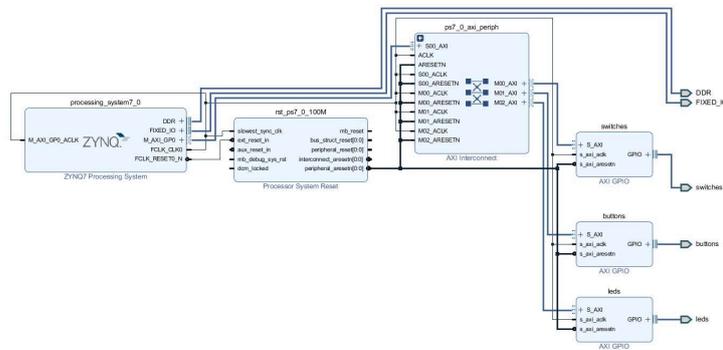


Periféricos mapeados a memoria

Conexión externa de los periféricos GPIO

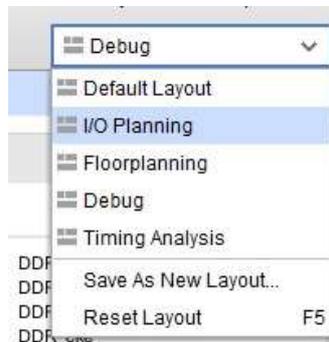
Las instancias **buttons**, **switches** y **leds** se deben conectar a pines de I/O asociados a los correspondientes elementos de la placa. Esto se puede hacer manualmente o mediante el asistente *Designer Assistance*. La información para saber a que pines de I/O están conectados los elementos de la placa está en el manual de la placa..

1. En la vista del diagrama, la opción **Designer Assistance** está disponible, igualmente vamos a conectar los puertos manualmente.
2. Presionar el botón derecho del ratón sobre el puerto **GPIO** del bloque **buttons** y seleccionar la opción **Make External** para crear la conexión externa. Esto creará un puerto externo denominado **GPIO_0** conectado al bloque. Como el entorno de desarrollo tiene identificada la placa utilizada, la asignación de pines de I/O de la FPGA se hace automáticamente.
3. Cambiar el nombre **GPIO_0** a **buttons** en el panel **External Interface Properties** (pestaña **General**).
4. Para el bloque **leds** presionar el enlace **Run Connection Automation**.
5. En la ventana seleccionar **leds -> GPIIP** y en la parte de **Options** seleccionar **custom**. Presionar **OK**.
6. Cambiar el nombre **gpio_rtl** a **leds**.
7. Cambiar el nombre **sws_2bits** a **switches** en el bloque **switches**.
8. Ejecutar la validación de diseño (**Tools -> Validate Design**) y verificar que no haya errores. El sistema debería verse similar a la siguiente figura.



Sistema Completo

9. En el panel **Flow Navigator** presionar en **Run Synthesis**. (presionar el boton **Save** si aparece una ventana emergente para guardar el proyecto) y cuando finaliza el proceso en la ventana emergente cambiar a la opcion **Open Synthesized Design** y presionar **OK** (no se puede elegir **Run Implementation** porque no estan todos los pines de I/O asignados)
10. En el menu de atajos (arriba a la derecha) seleccionar **I/O Planning**



Seleccionar la vista IO planning

3. En la pestaña **I/O ports** (parte inferior de la pantalla), expandir las tres entradas GPIO
4. Al expandir **switches_tri_i** se observara que los puertos han sido asignados automaticamente, con ubicaciones físicas (M19 y M20) y I/O Std *LVC MOS33*
5. Al expandir **buttons_tr_i** se observara que los puertos han sido asignados automaticamente, con ubicaciones físicas (L19, L20, D20 y D19) y I/O Std *LVC MOS33*
6. Al expandir **leds_tri_o** se observa que no hay pines asignados. Asignar

leds_tri_o[5] -> Package Pin M15 -> I/O Std LVC MOS33

leds_tri_o[4] -> Package Pin N15 -> I/O Std LVC MOS33

leds_tri_o[3] -> Package Pin M14 -> I/O Std LVC MOS33

leds_tri_o[2] -> Package Pin N16 -> I/O Std LVC MOS33

leds_tri_o[1] -> Package Pin P14 -> I/O Std LVC MOS33

leds_tri_o[0] -> Package Pin R14 -> I/O Std LVC MOS33

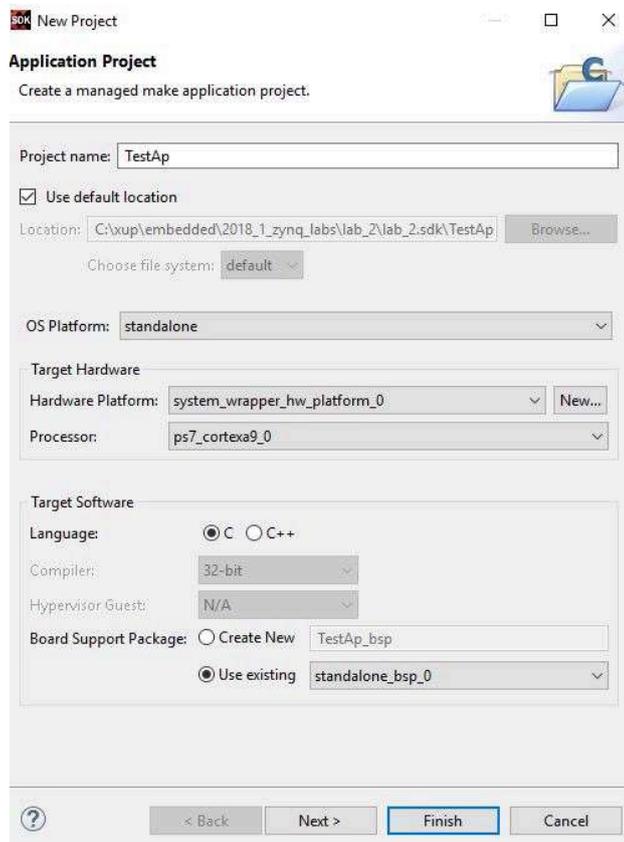
Generar el archivo de configuración Bitstream y exportar a SDK

1. En **PROGRAM AND DEBUG** seleccionar **Generate Bitstream**. presionar el boton **Save** si aparece una ventana emergente para guardar el proyecto. Presionar **OK** si

- aparece una ventana emergente informando que el diseño esta desactualizado (**Out of Date Design**)
2. En la ventana Emergente **Save Constraints** colocar el nombre de archivo **Constraints**. Presionar **OK**
 3. Aparecera una ventana emergente avisando que la sintesis esta desactualizada, presionar el boton **Yes**
 4. Cuando aparezca la ventana emerente **Bitstream Generation Complete** presionar **Cancel**
 5. Presionar el enlace **reload** en la pestaña de la parte superior de la pantalla
 6. Exportar el diseño de hardware mediante el menu **File -> Export -> Export Hardware**. Verificar que este chequeado el recuadro **Include Bitstream** y presionar **OK**. En este caso hay hardware instanciado en el bloque PL, por lo que se generó un bitstream que debe ser incluido en la exportación a SDK.
 7. En la ventana emergente presionar **Yes** para sobrescribir el archivo exportado (este archivo habia sido previamente creado en el Proyecto Lab1).
 8. Iniciar la herramienta SDK presionando **File > Launch SDK**. Presionar **OK**.

Generar la aplicación TestApp en SDK

1. En la herramienta SDK, presionar el boton derecho del raton en el Proyecto **mem_test** y seleccionar la opción **Close Project**
2. Hacer lo mismo para los proyectos **mem_test_bsp** y **system_wrapper_hw_platform_0**
3. En el menu de la parte superior seleccionar **File > New > Board Support Package**
4. Presionar el boton **Finish** con las opciones por defecto (Board Support Package OS **standalone OS**, Hardware Platform **system_wrapper_hw_platform_1** y nombre por defecto del Proyecto **standalone_bsp_0**)
5. Presionar **OK** en la ventana emergente con las opciones por defecto.
6. En el menu de la parte superior seleccionar **File > New > Application Project**
7. El nombre del proyecto es **TestApp**, en la seccion **Board Support Package** (al final), seleccionar **Use existing (standalone_bsp_0)** y presionar **Next**.



Configuración del Proyecto Application Project

8. En la ventana emergente seleccionar **Empty Application** y presionar **Finish**
 Esto creara un nuevo Proyecto **TestApp** usando el Proyecto BSP creado.
9. El gestor de librerías crea el archivo **xparameters.h** in el directorio `lab22.sdk_bsp_07_cortexa9_0`
 . Este archivo contiene las definiciones de direcciones de los periféricos.
10. Expandir el proyecto **TestApp**. Presionar el boton derecho del rato en la carpeta **src** y seleccionar **Import**
11. En la ventana emergente expandir la categoría **General** y dar doble click en la opción **File System**.
12. En la ventana emergente, presionar el boton **Browse** y navegar hasta la carpeta donde se encuentra el archivo **lab2IO.c**. Presionar el boton **Aceptar**
13. Seleccionar el archivo **lab2IO.c** y presionar **Finish**.

Prueba de funcionamiento en el Hardware

1. En el menu de la parte superior seleccionar **Xilinx -> Program FPGA**
2. Presionar el boton **Program** para configurar la FPGA. Cuando el proceso finalice, el led **Done** de la placa se iluminara en color verde. Seleccionar nuevamente la pestaña **SDK Terminal**
3. Seleccionar la pestaña **SDK Terminal** en la parte inferior de la pantalla
4. Presionar el boton + y verificar los parametros de puerto COM y velocidad de transmision si es necesario (normalmente estan los parametros del Lab1). Presionar **OK**. En la pantalla del terminal aparecera el texto **Connected to COMxx at 115200**

5. En el panel **Project Explorer** select el proyecto **TestApp** in Project Explorer.
Presionar el boton derecho del raton y elegir del menu desplegable la opcion **Run As**
-> **Launch on Hardware** (System Debugger) para ejecutar la aplicación.
6. Seleccionar nuevamente la pestaña **SDK Terminal**. Se debe ver un texto similar al siguiente (dependiendo de los botones e interruptores activos)

```
DIP Switch Status 6  
Push Buttons Status 8  
DIP Switch Status 6  
Push Buttons Status 8
```

Salida de SDK Terminal

Al presionar un pulsador o interruptor, se encendera el led correspondiente y se actualizara el valor en el SDK Terminal. Tener en cuenta que puede haber un pequeño retraso porque el programa se ejecuta una vez por segundo.

7. Cerrar la herramienta SDK y Vivado mediante la opcion de menu **File -> Exit** en ambas herramientas

8. **Conclusión**

En la practica se agregaron periféricos GPIO obtenidos del catalogo de IP. Estos se conectaron al PS a través de la interfaz *Master GPO*. Los periféricos se configuraron en tamaño de datos y se conectaron a pines externos de la FPGA. Se creo una aplicación de prueba y se verifico la funcionalidad en el hardware.