

Uso de Vivado para crear un Sistema Embebido

Introducción

En esta práctica se creará y probará un sistema básico en la plataforma Zynq instanciando los bloques de hardware y ejecutando un software de prueba.

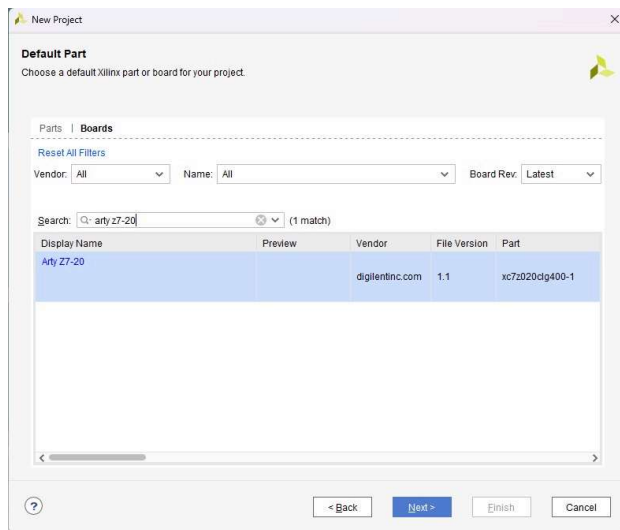
Objetivos

- Crear un proyecto en Vivado para la plataforma Zynq.
- Usar la herramienta IP Integrator para instanciar el hardware
- Usar la herramienta SDK(Software Development Kit) para crear un proyecto de software que realiza un test de la memoria
- Ejecutar la aplicación de test en la placa y verificar la funcionalidad del hardware

Procedimiento

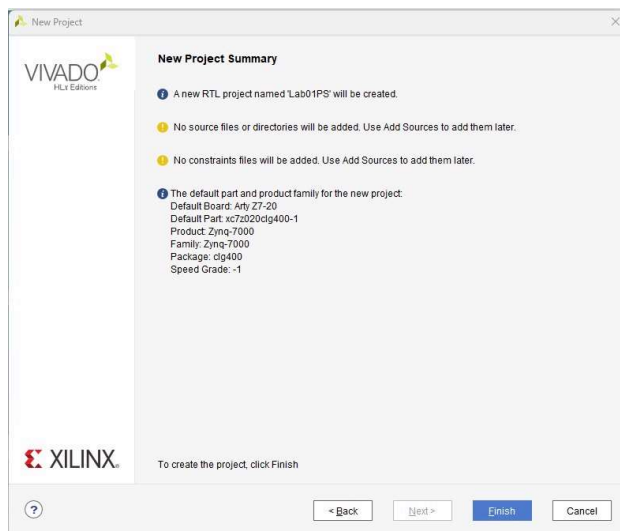
Crear un proyecto en Vivado

1. Ejecutar la herramienta **Vivado** (2018.3)
2. Presionar **Create Project** para iniciar el asistente. Aparecerá la ventana **Create a New Vivado Project**. Presionar **Next**.
3. Elegir la ubicación y guardar el proyecto como **Lab01PS**. Presionar **Next**.
4. En **Project Type** seleccionar **RTL Project**. Desmarcar la opción **Do not specify sources at this time**. Presionar **Next**.
5. En la ventana **Add Sources** en la opción **Target language** seleccionar **Verilog**. En la opción **Simulator language** seleccionar **Mixed**. Presionar **Next**.
6. Pasar la ventana **Add Constraints** sin cambios presionando **Next**.
7. En la ventana **Default Part window** seleccionar la pestaña **Boards**. En la opción **Search** buscar la placa **Arty Z7-20**. Presionar **Next**.



Ventana *Default Part*

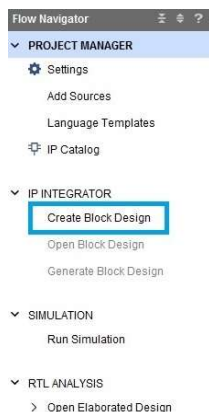
- Verificar el contenido de la ventana **Check the Project Summary** (debería ser similar a esta) y presionar **Finish** para crear el proyecto.



Ventana *Project Summary*

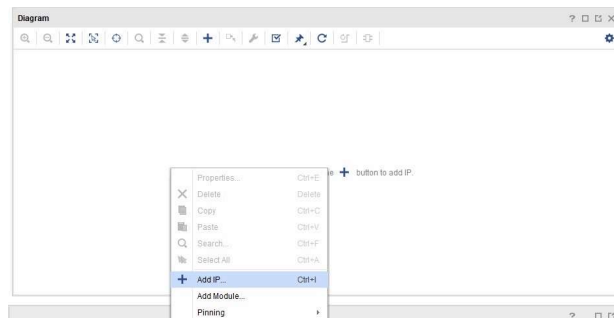
Instanciar el hardware utilizando la herramienta IP Integrator

- En la ventana **Flow Navigator**, dentro de **IP INTEGRATOR** presionar **Create Block Design**.



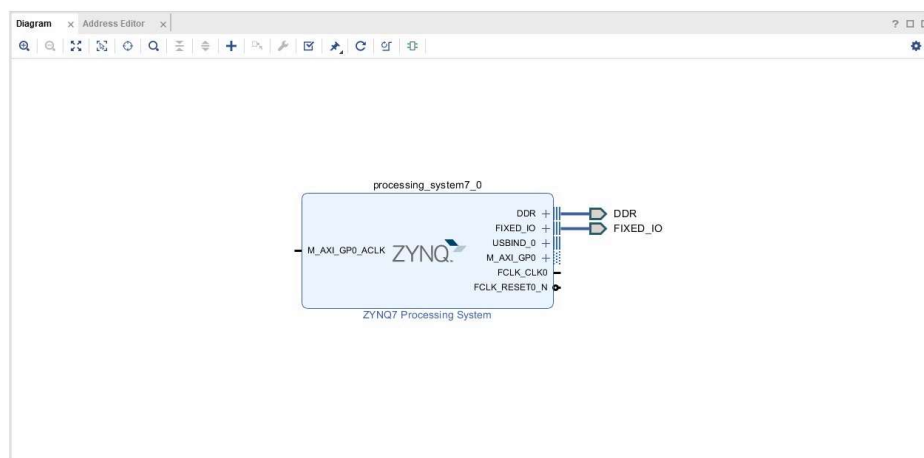
Opcion *Create Block Diagram*

2. En **Design Name** escribir **system** y presionar **OK**.
3. Presionar el boton derecho del mouse en cualquier lugar del area del diagrama y seleccionar **Add IP**.



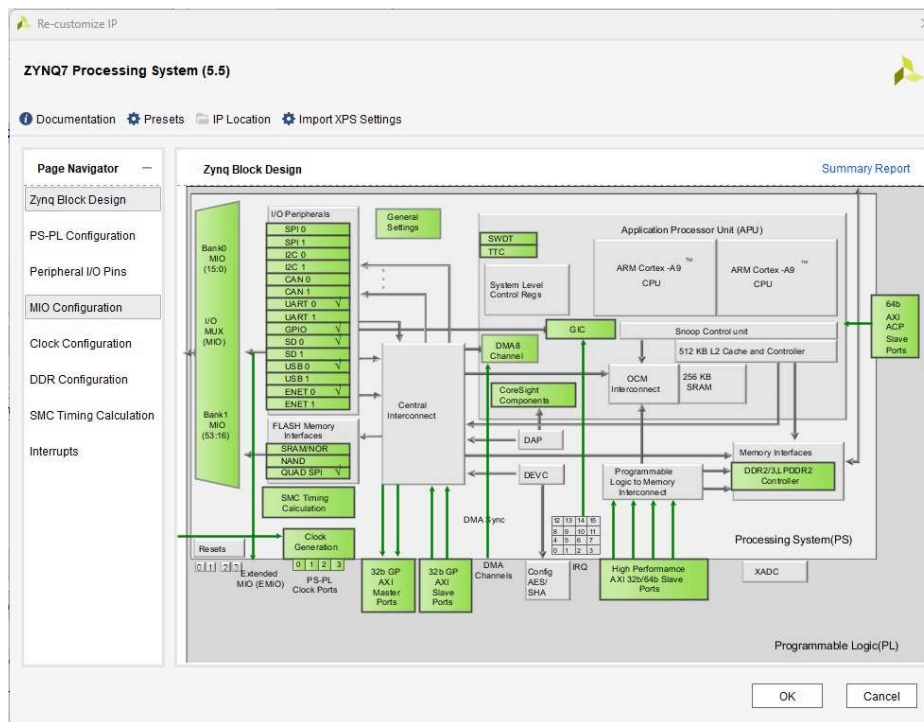
Opción de Menu *Add IP*

4. Cuando se abre la ventana **IP Catalog**, en la barra de busqueda **Search** escribir **“zyn”**. Aparecera la opción **ZYNQ7 Processing System**. Dar doble click para seleccionarla y agregarla al diagrama.
5. En la parte superior de la ventana de diagrama aparecera un mensaje en verde **Designer Assistance available**. Presionar **Run Block Automation**.
6. En la ventana **Run Block Automation** seleccionar **processing_system7_0**, dejar las opciones por defecto y presionar **OK**.
7. Una vez finalizada la operación, se agregaron en forma automática los puertos **DDR** y **Fixed IO**. El bloque debería verse así:



Zynq Block con puertos DDR y Fixed_IO

8. Dar doble click sobre el bloque para abrir la ventana **Re-customize IP**. Observar que la ventana muestra algunos perifericos con marcas de chequeo aplicadas en la parte de I/O Peripherals (UART 0, GPIO, SD 0, USB 0, ENET 0). Esta es la configuracion de perifericos por defecto del PS que se esta instanciando.



Ventana *Re-customize IP* con los periféricos por defecto

Configurar el PS con solo el periférico UART 0 habilitado.

9. En la ventana **Re-customize IP** los bloques marcados en verde son configurables. Al dar Click en uno de los periféricos en verde que están dentro de **I/O Peripherals** la ventana cambia al periférico correspondiente dentro de la opción **MIO Configuration**.
10. En la sección **Page Navigator** seleccionar **MIO Configuration** y deseleccionar todos los periféricos excepto **UART0**

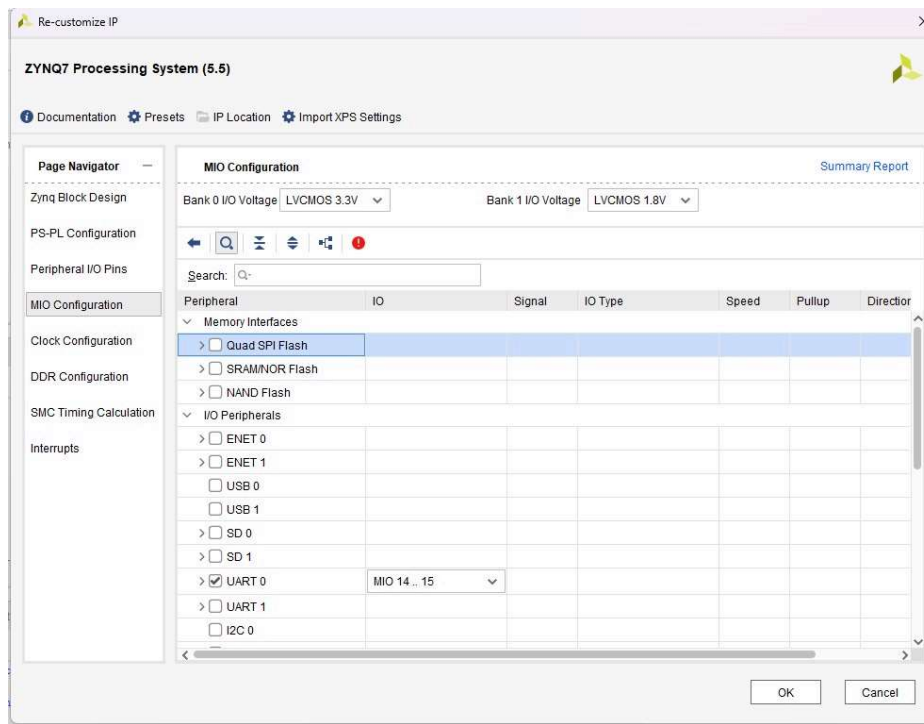
Se debe remover *ENET 0*

USB 0

SD 0

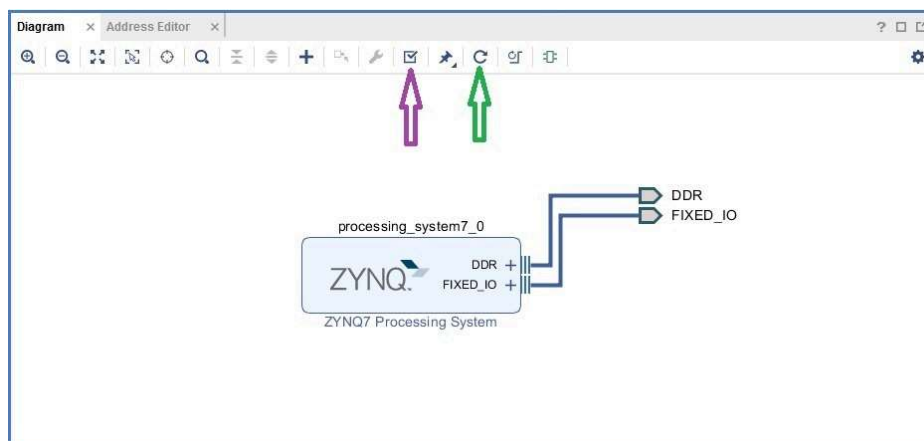
Expandir GPIO para deseleccionar GPIO MIO

Expandir Memory Interfaces to deselect Quad SPI Flash



Seleccionar solo UART 0

11. Seleccionar la opción **PS-PL Configuration** (a la izquierda de la ventana).
12. Expandir las opciones **AXI Non Secure Enablement > GP Master AXI interface** y deseleccionar **M AXI GP0 interface**.
13. Expandir **General > Enable Clock Resets** y deseleccionar **FCLK_RESET0_N**.
14. Seleccionar la opción **Clock Configuration** (a la izquierda de la ventana). Expandir **PL Fabric Clocks** y deseleccionar **FCLK_CLK0**. Presionar **OK**.
15. Presionar el boton **Regenerate Layout button** (flecha verde):



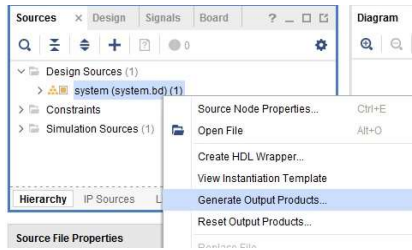
Regenerar y Validar el sistema

16. Presionar el boton **Validate Design button** (flecha violeta) y verificar que no hay errores.

Generar el archivo HDL Raiz y exportarlo al SDK

1. En el panel **Sources**, dar click al archivo **system.bd** para seleccionarlo, presionar el boton derecho del raton para desplegar el menu de opciones y seleccionar **Generate Output Products....** Presionar el boton **Generate** (Tambien se puede dar click en

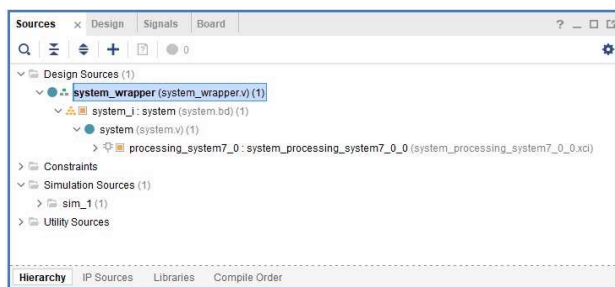
Generate Block Design en el panel **Flow Navigator** de la izquierda). Presionar **OK** cuando finaliza el proceso.



Proceso *Generate output products*

2. Presionar el boton derecho del raton nuevamente en **system.bd** y seleccionar **Create HDL Wrapper...** para generar el HDL raiz en Verilog. Dejar seleccionada la opción **Let Vivado manager wrapper and auto-update** y presionar **OK**.

El archivo **system_wrapper.v** se crea y se agrega al proyecto. Al dar doble-click en el archivo se puede ver la declaración del módulo.



Archivo HDL raiz agregado al proyecto

Notar que el archivo esta configurado como módulo principal, indicado con el icono correspondiente

3. Seleccionar en la parte superior izquierda la opción **File > Export > Export hardware** y presionar el boton **OK** (Guardar el proyecto si es necesario).

Observación: Dado que no hay hardware en el bloque PL, no es necesario generar un archivo de bitstream, por eso no es necesario seleccionar la opción **Include bitstream**.

1. Seleccionar en la parte superior izquierda la opción **File > Launch SDK** y presionar el boton **OK**

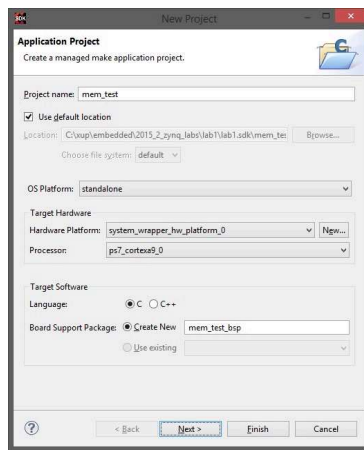
Se abra la herramienta SDK con la carpeta **system_wrapper_hw_platform_0** en el panel **Project Explorer** a la izquierda.

En el panel de vista de archivo (centro de la pantalla) se abra el archivo **system.hdf**.

En el archivo .HDF se puede ver cierta información del proyecto: Dispositivo, Mapa de memoria de los perifericos y bloques. Este archivo es utilizado por el entorno de software para saber que perifericos hay disponibles y su ubicación en memoria.

Generar la aplicación "Memory TestApp" en SDK

1. En **SDK** Seleccionar en la parte superior izquierda la opción **File > New > Application Project**
2. Nombrar al proyecto **mem_test** y presionar el boton **Next**

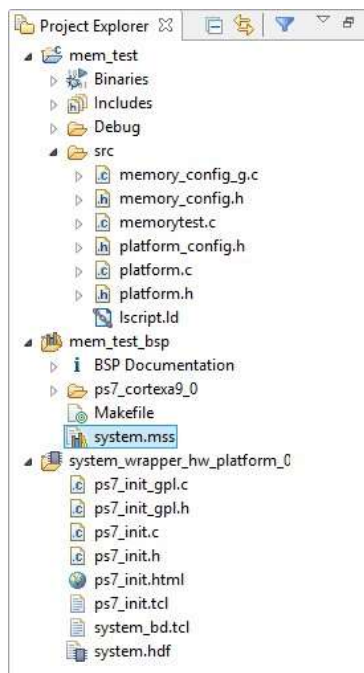


Ventana *SDK New Project*

3. Seleccionar el template **Memory Tests** de la lista y presionar el boton **Finish**.

El proyecto en lenguaje C **mem_test** y el proyecto BSP **mem_test_bsp** estaran disponibles en el panel **Project Explorer**.

4. El proyecto **mem_test** es la aplicación que se utilizara para verificar la funcionalidad del sistema. El proyecto **hw_platform** incluye la función **ps7_init** que inicializa el PS y el proyecto **mem_test_bsp** es el paquete de soporte (board support package). El **Project Explorer** con los proyectos expandidos deberia verse asi:



Project Explorer

5. Abrir el archivo **memorytest.c** del proyecto **mem_test** (dentro de la carpeta **src**) y examinar su contenido. Se vera que el archivo llama a funciones para verificar la memoria del sistema.

Verificar la funcionalidad en el hardware

Configurar la conexión

1. Conectar la placa Arty Z7-20 a la computadora
2. En la parte central abajo, seleccionar la pestaña **SDK Terminal**
3. Presionar el boton "**Connect**" (signo "+" en verde). Se abriran las propiedades de la conexion serie

4. Configurar el puerto apropiado y establecer la conexión a 115200 (En Microsoft Windows se puede usar el Windows Device Manager para ver el valor del puerto)

Ejecutar la aplicación **mem_test** y verificar su funcionalidad.

1. En SDK seleccionar el proyecto **mem_test** presionar el botón derecho del ratón y seleccionar **Run As > Launch on Hardware (System Debugger)** para programar la aplicación y ejecutarla.
2. En la pestaña **SDK Terminal** se debería ver la siguiente salida.

```
NOTE: This application runs with D-Cache disabled. As a result, cacheline requests will not be generated
Testing memory region: ps7_ddr_0
  Memory Controller: ps7_ddr
    Base Address: 0x00100000
    Size: 0x1ff00000 bytes
    32-bit test: PASSED!
    16-bit test: PASSED!
    8-bit test: PASSED!
Testing memory region: ps7_ram_1
  Memory Controller: ps7_ram
    Base Address: 0xffff0000
    Size: 0x0000fe00 bytes
    32-bit test: PASSED!
    16-bit test: PASSED!
    8-bit test: PASSED!
--Memory Test Application Complete--
```

Salida en **SDK Terminal**

Tener en cuenta que al lanzar la aplicación, la pestaña activa pasa a ser **Console**, cambiar a la pestaña **SDK Terminal**

3. Cerrar las herramientas SDK y Vivado seleccionando el menú **File > Exit** de cada programa.

Conclusión

Las herramientas **Vivado** e **IP Integrator** permiten diseñar y configurar un sistema PS. Una vez definido el sistema, el hardware se puede exportar a la herramienta SDK.

El desarrollo de software se realiza en la herramienta SDK, la cual provee distintas plantillas, una de ellas con una aplicación para testear la memoria. Se verificó el funcionamiento del hardware ejecutando una aplicación en la FPGA cuya salida fue visualizada en un terminal de la herramienta SDK.