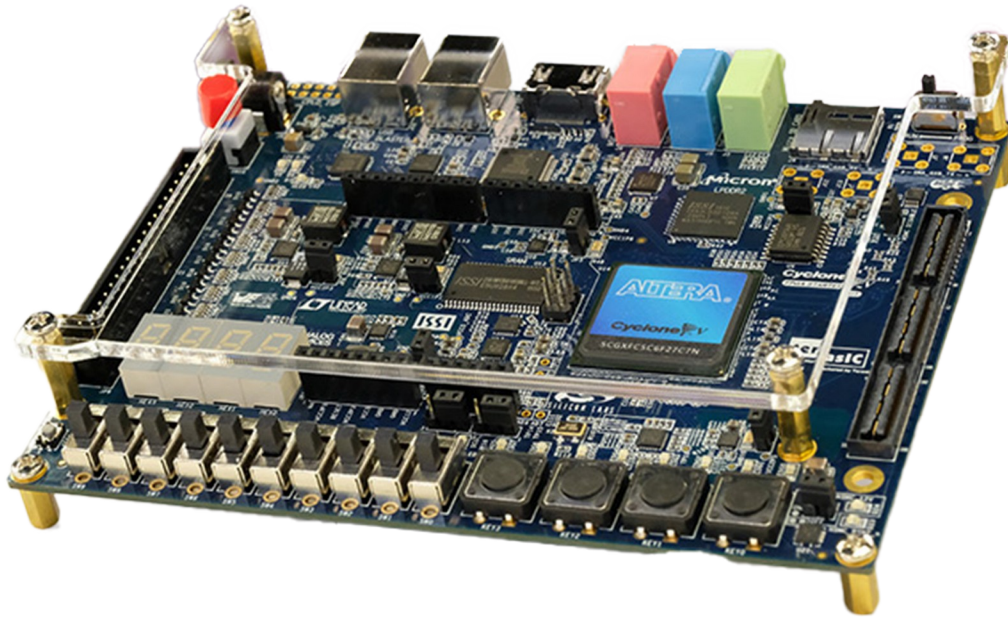


DISEÑO AUTOMÁTICO DE SISTEMAS FIABLES

Práctica 4



Marcos Gago García
David Gregorio Muñoz

ÍNDICE

Objetivos	3
Conceptos	3
Diseño	4
Otras posibles funcionalidades	8

OBJETIVOS:

En esta cuarta práctica se va a hacer el diseño de un procesador utilizando un microcontrolador Microblaze y diferentes componentes que ofrecen Vivado y la placa Nexys 4 DDR, como una memoria, un reloj o el módulo UART.

CONCEPTOS:

Para esta práctica es necesario definir y entender determinados conceptos que se aplicarán a lo largo de la misma. Estos conceptos son los siguientes:

MicroBlaze: es un microprocesador de 32 bits, con arquitectura de tipo RISC (conjunto de instrucciones reducido) y de tipo Harvard (memorias de datos e instrucciones separadas). Está diseñado y distribuido por Xilinx, y debido a su estructura y requisitos y alta configurabilidad lo hacen ideal para implementarse en una FPGA.

SoC: es el acrónimo de *System On a Chip*. Se les llama así a los chips que dentro de ellos integran múltiples componentes diferentes como memorias, procesadores, GPUs o sistemas de entrada/salida. Juntos forman un sistema electrónico completo.

UART: es el acrónimo de *Universal Asynchronous Receiver/Transmitter*. Es un circuito integrado utilizado para enviar y recibir datos en serie de manera asíncrona entre diferentes componentes y/o dispositivos. Es un estándar muy utilizado en el diseño hardware para la transmisión de datos.

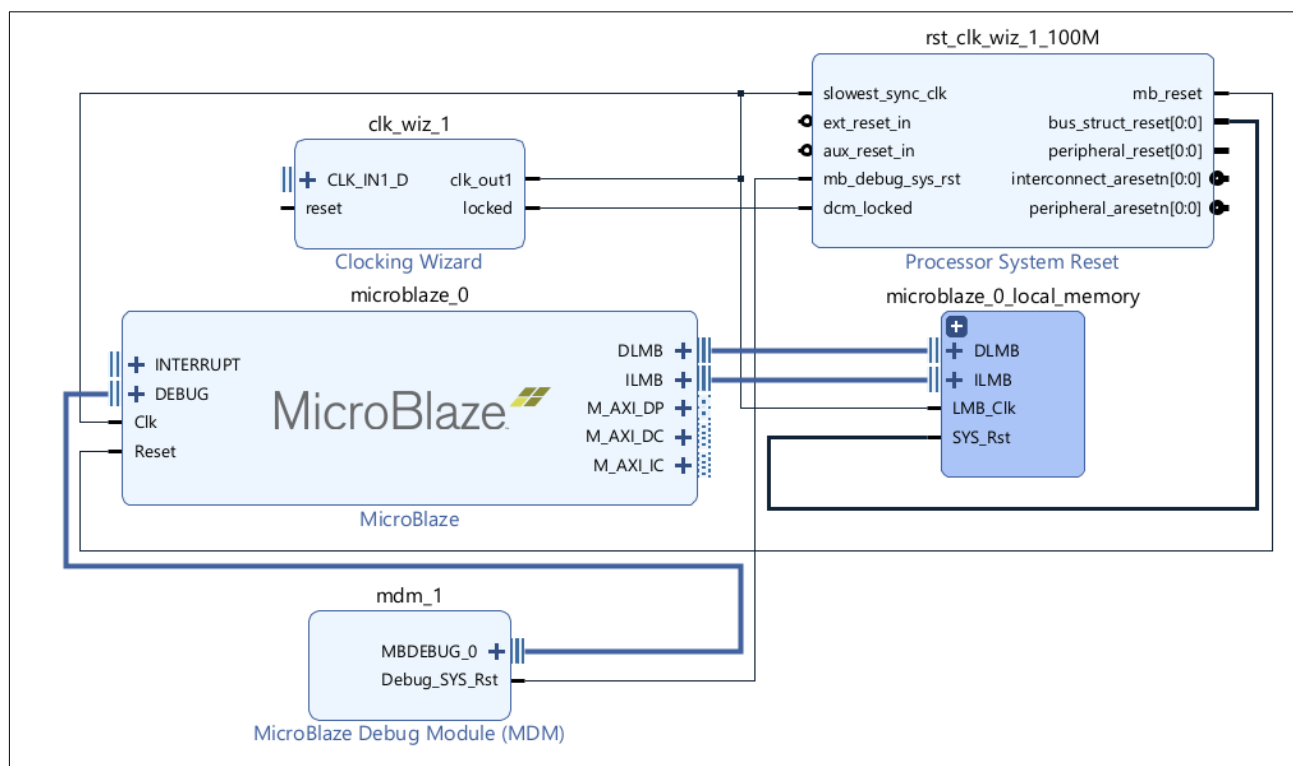
GPIO: es un conjunto de pines de entrada/salida de propósito general que se encuentran en multitud de dispositivos, que al no tener una función específica pueden aportar soluciones cuando se requieren pines adicionales en un diseño específico, y pueden ser programados para llevar a cabo funciones como la de sensores o LEDs.

MONTAJE DEL PROCESADOR

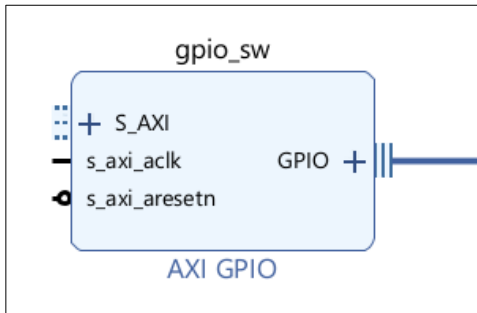
Creado el proyecto con la placa *Nexys 4 DDR*, se comienza haciendo un nuevo diagrama de bloques. Una vez hecho, lo primero para el diseño del procesador es añadir el microcontrolador Microblaze.



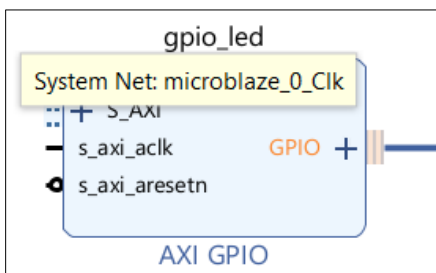
Se hace *Run Block Automation* y le asigna una memoria local de 32KB y una caché de 16KB, que se traduce en el siguiente diagrama. El propio Vivado ha introducido los módulos del reloj y de la memoria, aparte del de debug. También se va a modificar el bloque del reloj con la siguiente configuración donde, a las entradas de reloj y reset, se les asigna el reloj y el reset de la FPGA. Además se crean dos frecuencias diferentes, de 100 y 200MHz, y se pone el reset a *Active Low*.



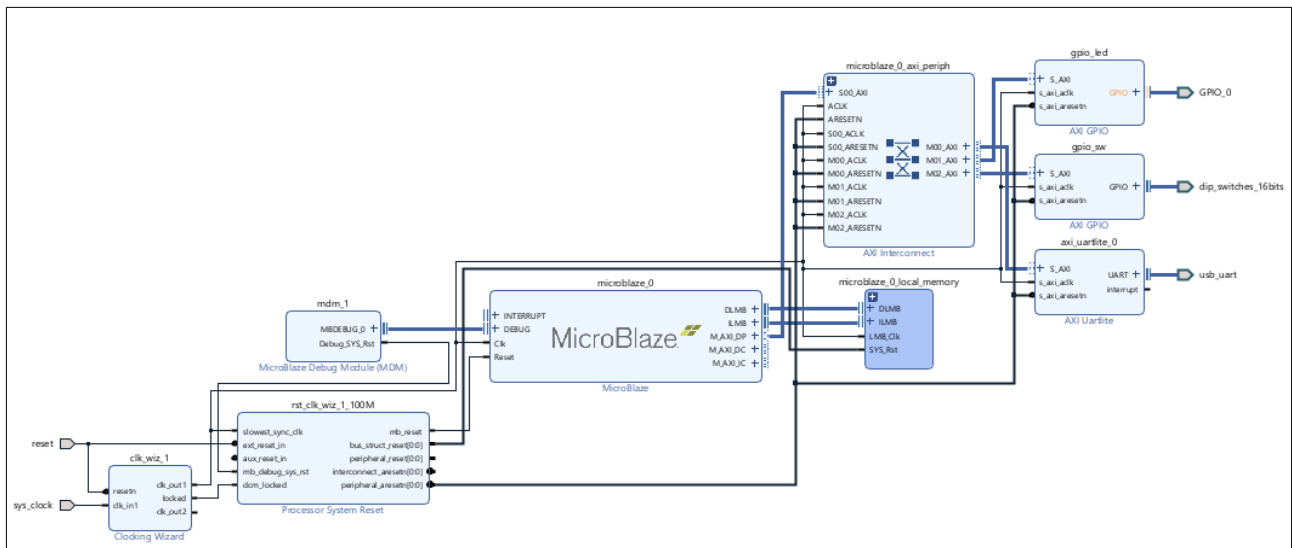
Se añaden ahora dos nuevos bloques: el de los switches y el de los LEDs.



Al bloque de los LEDs hay que asignarle los LEDs de la placa a la salida, y hacerla externa.



Tras esto, se añade el bloque de la interfaz UART (AXI Uartlite), se hace el Run Connection Automation para todos los bloques y se obtiene el siguiente diagrama (después del regenerate layout).



The diagram shows a block labeled **mig_7series_0**. On the left side, there are inputs: a 32-bit bus (indicated by a vertical dotted line) for **S_AXI**, a single-bit input **sys_rst**, a single-bit input **sys_clk_i**, and a single-bit input **aresetn**. On the right side, there are outputs: a 64-bit bus (indicated by a vertical dotted line) for **DDR2**, and single-bit outputs **ui_clk_sync_rst**, **ui_clk**, **ui_addn_clk_0**, **mmcm_locked**, and **init_calib_complete**.

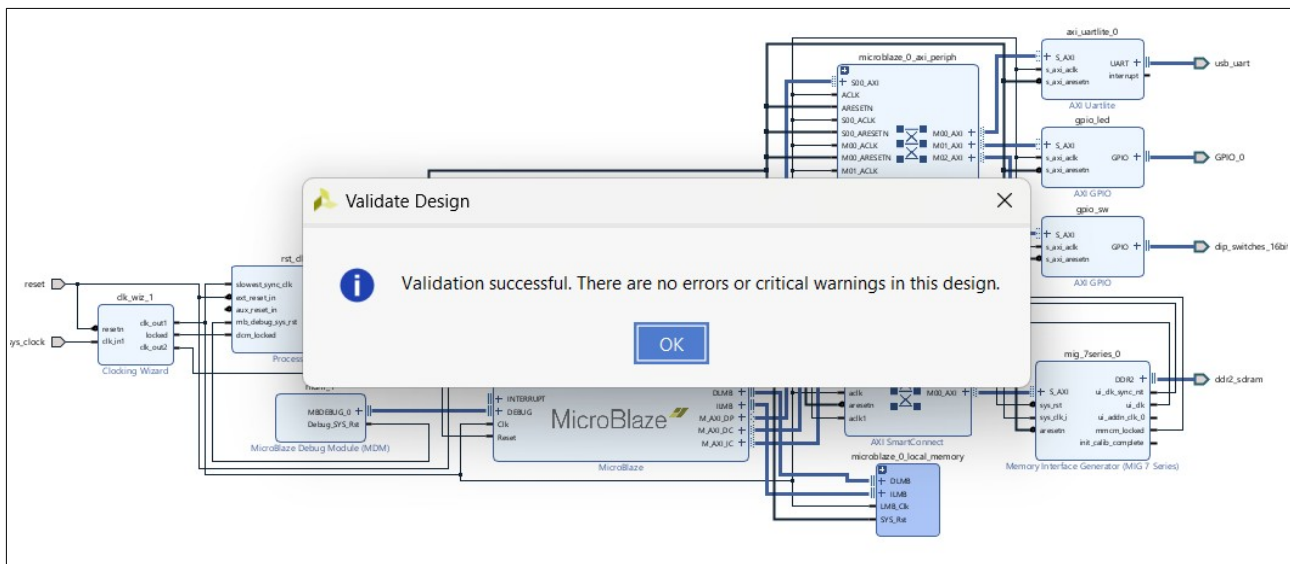
[illegible]

[illegible]

This block diagram illustrates the system reset logic for a MicroBlaze processor. The central component is the **MicroBlaze** processor, which is connected to various system components:

- Processor System Reset:** Two blocks (one labeled `rst_clk_wiz_1_100M` and another `rst_mig_7series_0_81M`) provide reset signals to the MicroBlaze. These blocks receive inputs from `reset` and `sys_clock` and output `slowest_sys_clk`, `rst_reset_in`, `rst_reset`, `rst_reset_n`, `mb_debug_sys_rst`, `interconnect_ar_reset(0:0)`, `peripheral_ar_reset(0:0)`, and `don't_lock` signals.
- MicroBlaze Debug Module (MDM):** The `mdm_1` block is connected to the MicroBlaze and provides `INTERRUPT + DEBUG`, `Clk`, and `Reset` signals.
- MicroBlaze Local Memory:** The `microblaze_0_local_memory` block is connected to the MicroBlaze and provides `DUMB + LUMB`, `Clk`, and `sys_rst` signals.
- AXI SmartConnect:** The `axi_smc` block is connected to the MicroBlaze and provides `S00_AXI`, `S01_AXI`, `ar_en`, `ar_en_n`, and `ar_en_n` signals.
- AXI Interconnect:** The `microblaze_0_axi_periph` block is connected to the MicroBlaze and provides `S00_AXI`, `S01_AXI`, `ar_en`, `ar_en_n`, and `ar_en_n` signals.
- AXI UARTlite:** The `axi_uartlite_0` block is connected to the MicroBlaze and provides `uart` and `interconnect` signals.
- AXI GPIO:** The `axi_gpio` block is connected to the MicroBlaze and provides `gpio` and `gpio_sw` signals.
- AXI LED:** The `axi_led` block is connected to the MicroBlaze and provides `gpio` and `gpio_sw` signals.
- AXI DIP Switches:** The `axi_dip_switches` block is connected to the MicroBlaze and provides `gpio` and `gpio_sw` signals.
- AXI MIG:** The `mig_7series_0` block is connected to the MicroBlaze and provides `ddr2_sdram` and `ddr2_sdram` signals.

The diagram shows the complex interconnections between these components, ensuring proper system reset and initialization.



OTRAS POSIBLES FUNCIONALIDADES:

La Nexys 4 DDR incluye multitud de módulos con diferentes funcionalidades y propósitos. Por citar algunos, tiene una salida VGA de 12 bit, gracias a la cuál podría mandar señales de vídeo a una pantalla conectada a la placa. Contiene también un sensor de temperatura, con el cual poder implementar multitud aplicaciones comerciales. Otros módulos que podrían ser interesantes que existen en la placa son un puerto Ethernet y un conector USB.