

# Introducción a ZYNQ

Daniel Temalatzi Mojica

Instituto Tecnológico de Puebla

16 de mayo de 2019

# Índices

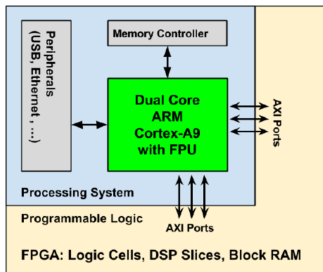
- 1 Introduccion
  - Conociendo ZYNQ
  - Conociendo VIVADO
- 2 Primeros pasos
  - Creando el hardware
  - Conociendo y personalizando el procesador
- 3 Programando
  - Software Development Kit (SDK)

# Qué es ZYNQ?

## Definición

ZYNQ es un SoC (System on a Chip) el cual posee un **Hard-Processor** y una **FPGA**, ambos en un mismo chip.

Figura: Arquitectura de ZYNQ



# Especificaciones

- Procesador ARM Dual-Core A9
- FPGA equivalente a la serie 7 de Xilinx
- 512 MB de RAM

Esas suelen ser las características estándar de un SoC ZYNQ, sin embargo hay diversas variaciones por los distintos modelos, llegando a existir algunos con un solo núcleo A9.

# Tarjetas más populares

Existen muchas tarjetas de desarrollo basadas en ZYNQ, sin embargo algunas sobresalen mas que otras.

Estas son algunas de ellas:

- ZYBO Z7 (Digilent)
- ARTY Z7 (Digilent)
- CORA Z7 (Digilent)
- ZEDBOARD (Avnet)

# Qué es VIVADO?

VIVADO es la herramienta que nos proporciona Xilinx para poder diseñar sobre sus FPGA, es una herramienta bastante completa, a diferencia de ISE, en VIVADO contamos ya con todo lo necesario para poder cargar los *bitstreams* a la FPGA.

VIVADO tambien nos ayuda a poder exportar nuestro hardware (.hdf) para poder trabajar con el **SDK**.

# Qué es el SDK?

Xilinx SDK (Software Development Kit) es el entorno de programación con el que podemos programar nuestros procesadores, ya sea los núcleos ARM de ZYNQ o el **Soft-Core** Microblaze.

El SDK cuenta ya con ejemplos prácticos para poder comenzar a programar, contiene todo lo necesario para empezar, solo necesita nuestro hardware (.hdf), el cual obtenemos de VIVADO.

# Creación del proyecto

Antes de poder comenzar a programar nuestro procesador ZYNQ es necesario configurarlo, para ello debemos crear un nuevo proyecto en VIVADO.

Figura: Nombre y ubicación.

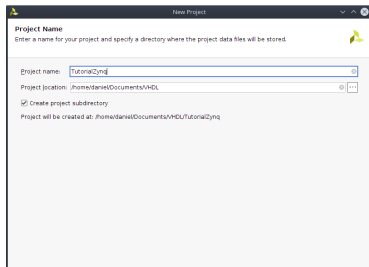
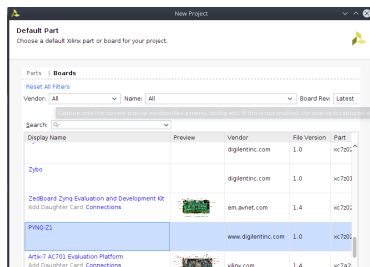


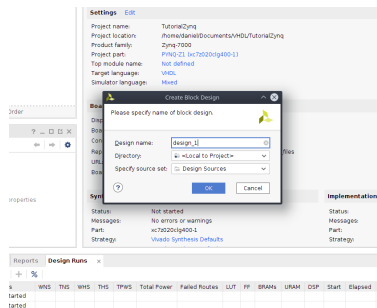
Figura: Tarjeta que usaremos.





## Creando el hardware

**Figura:** Creación de diseño de bloque.



**Figura:** Buscamos el procesador ZYNQ7

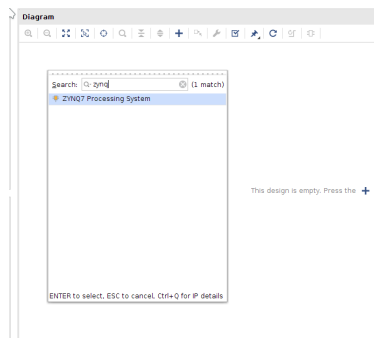


Figura: Corremos la automatización del bloque.

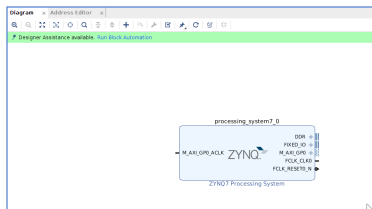
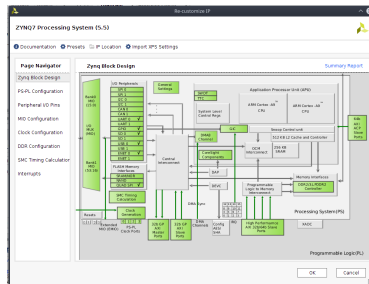


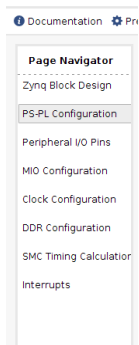
Figura: En la siguiente pantalla podemos personalizar nuestro procesador ARM.



# Lista de opciones

## Figura: Opciones de configuración

ZYNQ / Processing :



- Zynq Block Desing
- PS-PL Configuration
- Peripheral IO Pins
- MIO Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts

# Configurando ZYNQ

Existen muchas opciones de configuración para el procesador ZYNQ, sin embargo, en esta ocasión solo revisaremos algunas de ellas.

## Zynq Block Desing

Este menu nos sirve como acceso directo a los apartados que deseamos configurar, de igual manera podemos activar o desactivar los puertos y protocolos que usaremos.

## PS-PL Configuration

Aquí podremos configurar como interactua el PS (ARM) con la PL (FPGA).

## Peripheral IO Pins

En este menu configuramos los puertos de entrada y salida del procesador, incluidos los protocolos UART,SPI,I2C,etc.

## MIO Configuration

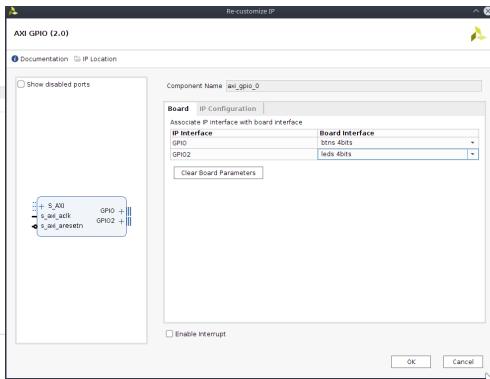
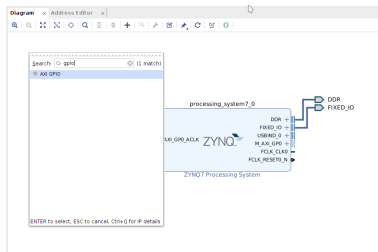
Esta sección es para configurar los puertos MIO, que nos indican a que lugar iran los IO, ademas de poder confiurar los timer internos del procesador.

# Puertos PL

En algunas tarjetas el procesador ARM tiene puertos propios, ya sean leds, botones o puertos GPIO, sin embargo lo normal es que todos los puertos GPIO estén conectados directo a la FPGA por lo que es necesario "conectarlos" con el procesador ARM.

Esto se hace posible gracias al protocolo AXI, el cual nos permite mandar información entre ambas partes del SoC (PS-PL, PL-PS)

Figura: IP AXI GPIO



## Conociendo y personalizando el procesador

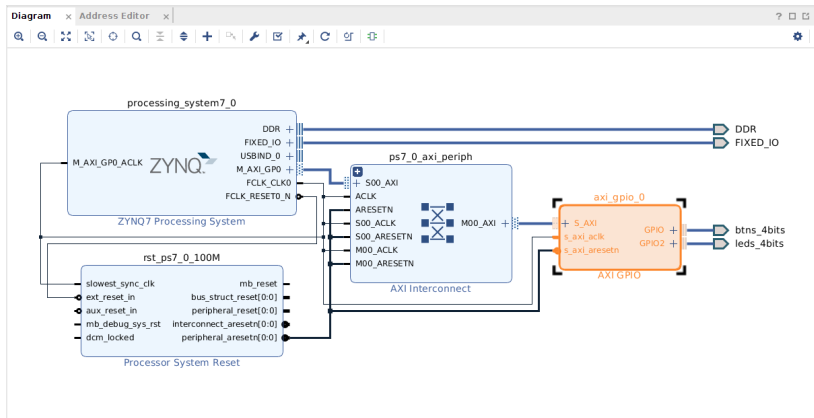


Figura: Vista final



# Finalizando el diseño

Tras terminar nuestro diseño tenemos que generar nuestro *bits-tream* para exportarlo al SDK.

# Primeros pasos

Una vez con el hardware exportado podemos comenzar a trabajar el software, en el SDK se programa en lenguaje C y C++.

A diferencia de *Microblaze* en ZYNQ se requiere de un **BSP (Board Support Package)**, este nos da todas las librerias necesairas para poder usar el procesador.

Tambien se necesita de un bootloader, en este caso es llamado **FSBL (First Stage Bootloader)**, lo que haces este archivo es programar la parte logica (FPGA) para que pueda interactuar con el procesador.

# BSP

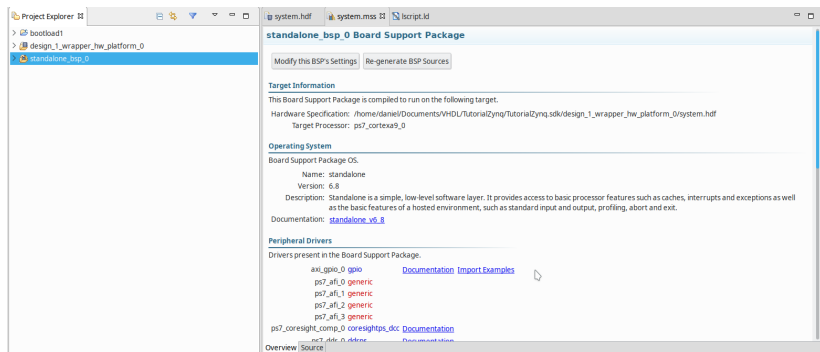
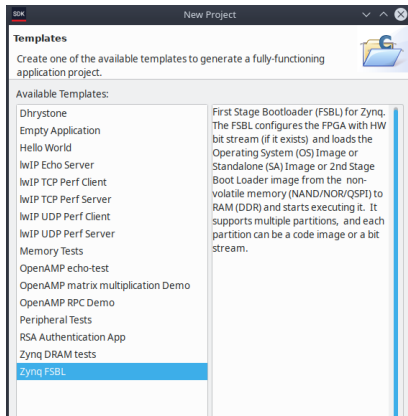


Figura: Board Support Package

# FSBL

Figura: First Stage Bootloader



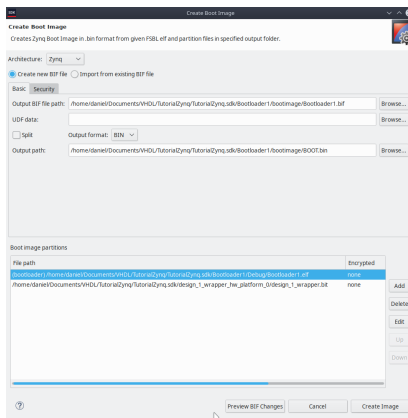
# Creando imagen BOOT.BIN

Lo siguiente será crear el BOOT.BIN, esta imagen contiene 2 archivos, el FSBL y el .bit

La imagen BOOT.BIN se puede guardar tanto en un Micro SD, como en la memoria QSPI que incluye la tarjeta, en este caso lo grabaremos en la memoria QSPI.

# BOOT.BIN

Figura: Archivos para crear la imagen



# Hola Mundo!

Figura: En la siguiente pantalla podemos personalizar nuestro procesador ARM.

Hola

