

Introducción al Diseño de Sistemas Embebidos usando Zynq

Zynq
Vivado 2018.1

Objetivos

➤ Al completar este módulo, el alumno será capaz de:

- Definir un Zynq All Programmable SoC (AP SoC) processor component
- Enumerar los aspectos claves del sistema de procesamiento Zynq AP SoC
- Describir el flujo de diseño embebido
- Entender la función de la herramienta IP Integrator
- Indicar como el diseño de hardware es relacionado al ambiente de desarrollo de software

Temario

- ***Componente Procesador Embebido***
- Reseña de Vivado para diseño de sistemas embebidos
- Flujo de diseño de sistemas embebidos
- Creación de plataforma de hardware
- Plataforma de software SDK
- Resumen

Arquitectura del Diseño Embebido en Zynq

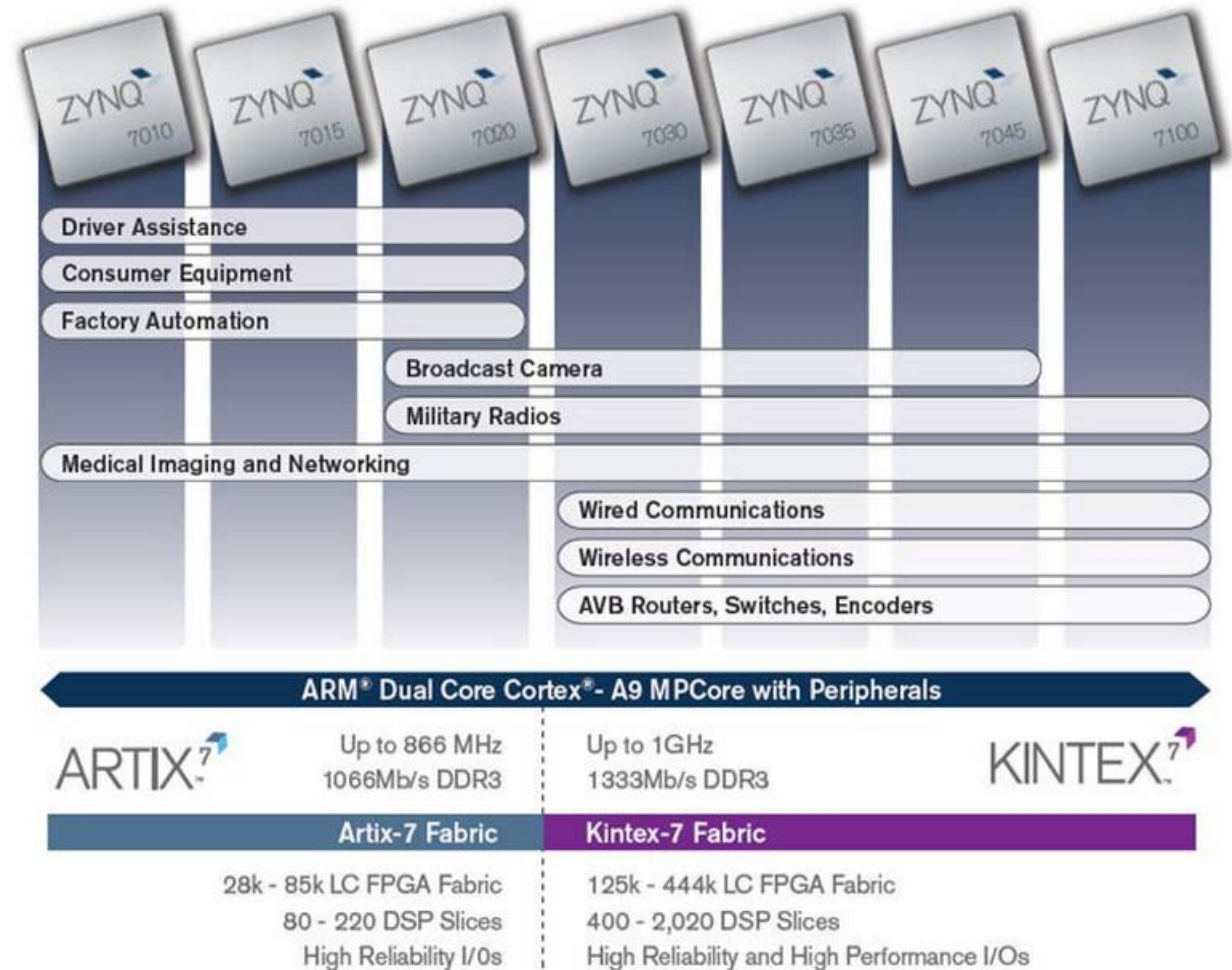
➤ El diseño embebido con Zynq está basado en:

- **Procesador y periféricos**
 - Dual ARM® Cortex™ -A9 processors of Zynq-7000 AP SoC
 - Interconexión AXI
 - Periféricos compatibles AXI
 - Reset, clocking, puertos de depuración
- **Plataforma de software para el sistema de procesamiento**
 - Aplicaciones bare-metal o Sistemas Operativos (ej: Linux, FreeRTOS)
 - Soporte de lenguaje C
 - Servicios de procesador
 - Drivers en C para el hardware
- **User application**
 - Rutinas para servicio de interrupciones (opcional)

PS (Sistema de Procesamiento) y PL (Lógica Programable)

➤ The Zynq-7000 AP SoC architecture consists of two major sections

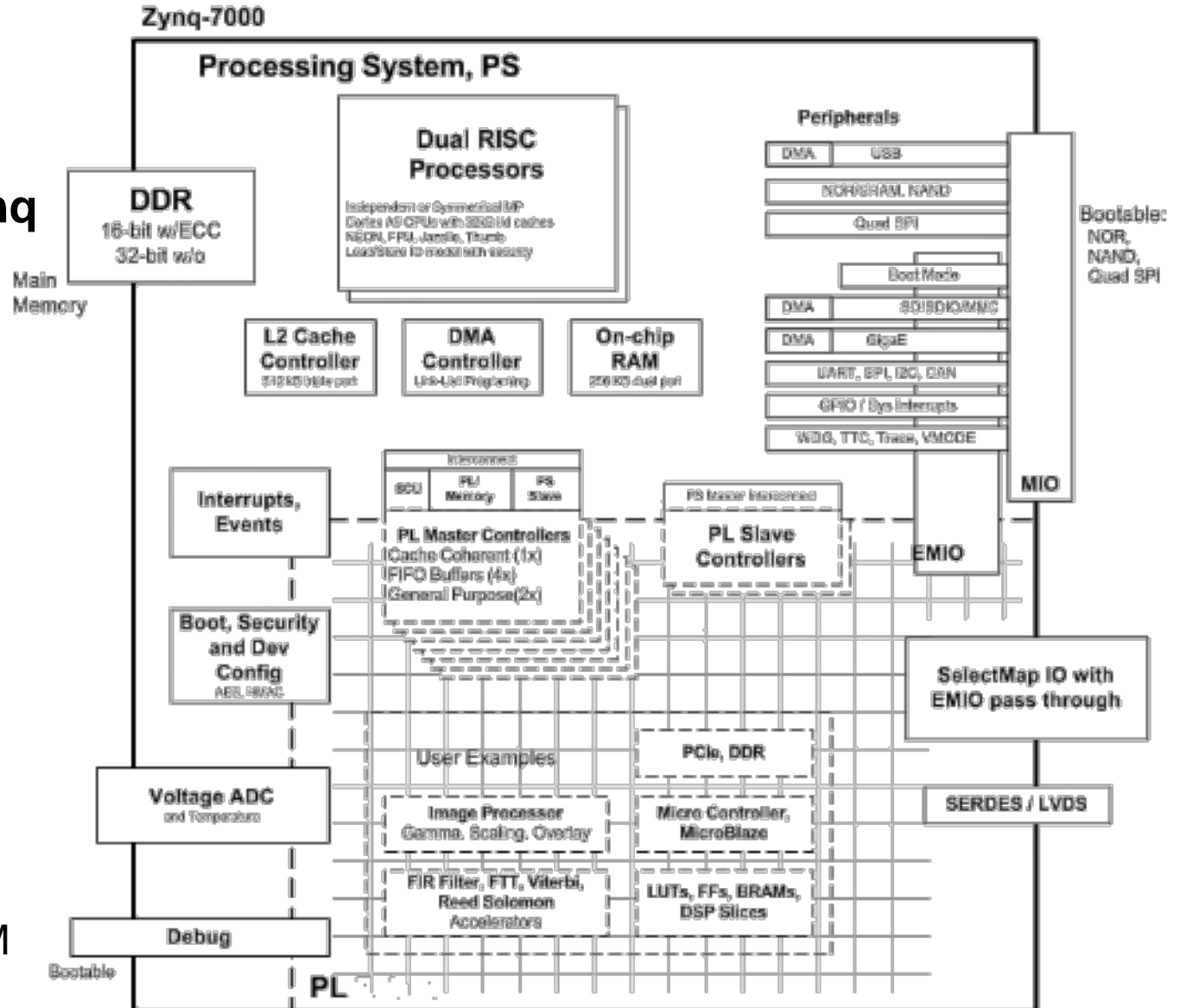
- PS: Sistema de procesamiento
 - Procesador dual basado en ARM Cortex-A9
 - Múltiples periféricos
 - Hard core de silicio
- PL: Lógica programable
 - Usa la misma lógica programable de la serie 7
 - Dispositivos basados en Artix™: Z-7010, Z-7015 and Z-7020 (high-range I/O banks only)
 - Dispositivos basados en Kintex™: Z-7030, Z-7035, Z-7045, and Z-7100 (mix of high-range and high-performance I/O banks)



PS Components

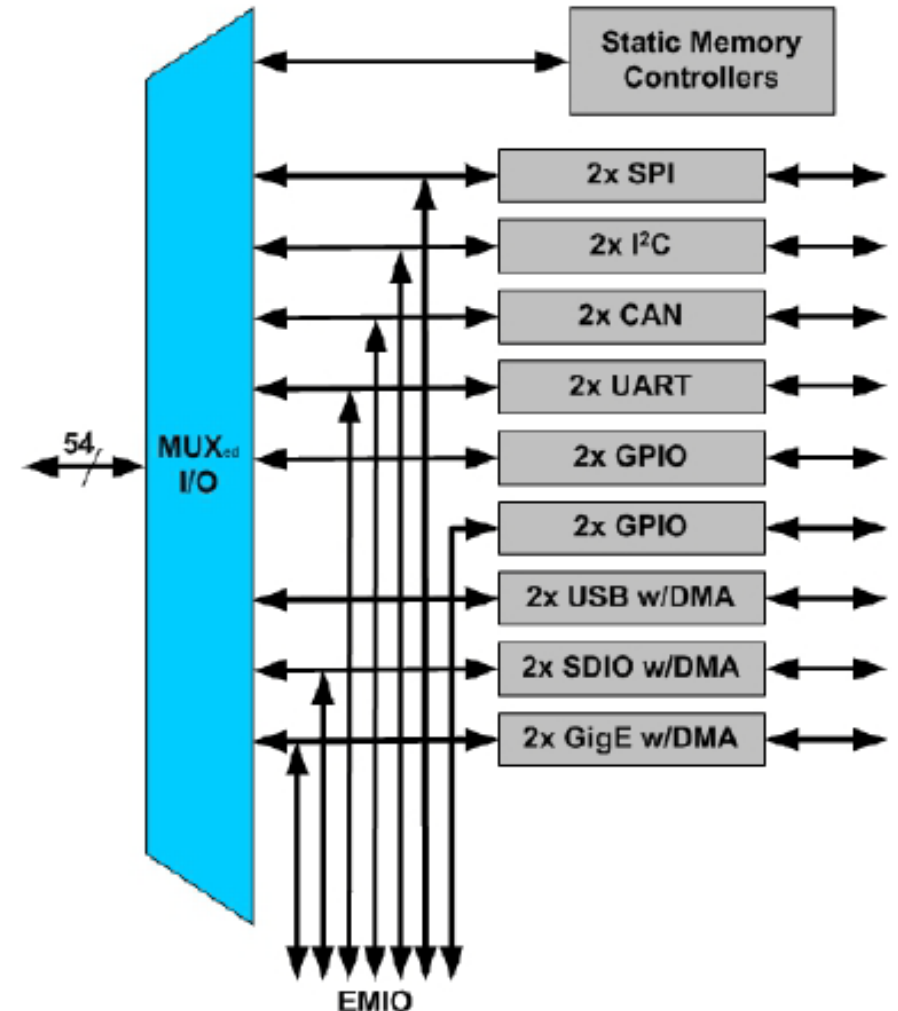
➤ El sistema de procesamiento del Zynq AP SoC está conformado por los siguientes bloques:

- Application processing unit (APU)
- Periféricos I/O (IOP)
 - I/O Multiplexado (MIO), I/O multiplexado extendido (EMIO)
- Interfaces de memoria
- Interconexiones del PS
- DMA
- Timers
 - Público and privado
- General interrupt controller (GIC)
- On-chip memory (OCM): ROM and RAM
- Controlador de depuración: CoreSight



Periféricos incluidos en la Arquitectura Zynq

- Dos USB 2.0 OTG/Device/Host
- Dos Tri- Mode GigE (10/100/1000)
- Dos interfaces SD/SDIO
 - Memoria, I/O y combo cards
- Dos CAN 2.0Bs, SPIs , I2Cs, UARTs
- Cuatro bloques GPIO de 32bit
 - 54 disponibles a través de MIO; otros a través de EMIO
- Entrada/Salida multiplexada (MIO)
 - Pinout multiplexado de periféricos y memoria estática
- MIO extendida
 - Mapea los puertos de los periféricos del PS al PL



Temario

- Componente Procesador Embebido
- ***Reseña de Vivado para diseño de sistemas embebidos***
- Flujo de diseño de sistemas embebidos
- Creación de plataforma de hardware
- Plataforma de software SDK
- Resumen

Vivado

➤ Qué son Vivado, IP Integrator y SDK?

- Vivado es un conjunto de herramientas para el diseño sobre FPGAs de Xilinx e incluye capacidades para el diseño de sistemas embebidos
 - IP Integrator, es parte de Vivado y permite el diseño a nivel de sistema de la parte de hardware de un sistema embebido
 - Está integrado dentro de Vivado
 - Vivado incluye todas las herramientas, IP, y documentación que son requeridas para diseñar sistemas con el Zynq-7000 AP SoC hard core y/o con MicroBlaze, el soft core de 32 bits de Xilinx
 - Vivado + IPI reemplaza a ISE/EDK
- SDK es un ambiente de desarrollo de software basado en Eclipse
 - Permite la integración de componentes de hardware y software
 - Se ejecuta desde Vivado

➤ Vivado es el administrador del proyecto completo y es usado para el desarrollo de hardware no-embebido y para instanciar sistemas embebidos

- Vivado/IP Integrator flow es lo recomendado para desarrollar sistemas embebidos con Zynq

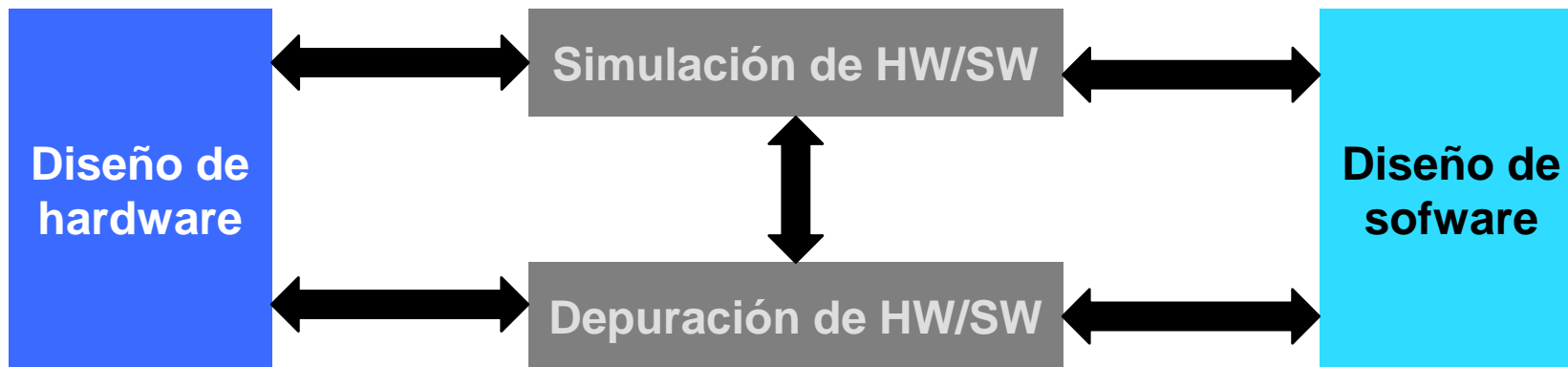
Componentes de Vivado

➤ Vivado/IP Integrator

- Ambiente de diseño para la configuración de PS, y el diseño de hardware para PL
- Plataforma de hardware (xml)
- Simulación de la plataforma, el software y los periféricos
- Integración de un analizador lógico

➤ Software Development Kit (SDK)

- Workspace del proyecto
- Definición de la plataforma de hardware
- Board Support Package (BSP)
- Aplicación de software
- Depuración de software



Herramientas para el Sistema Embebido: Hardware

➤ Hardware and software development tools

- IP Integrator
- IP Packager
- Generación de una netlist de hardware
- Generación de un modelo de simulación
- Xilinx Microprocessor Debugger (XMD)
- Depuración de hardware usando el analizador lógico (Vivado logic analyzer)

Herramientas para el Sistema Embebido: Software

➤ IDE basado en Eclipse, Software Development Kit (SDK)

- Creación de Board support package
- Herramientas GNU de desarrollo de software
- Compilador C/C++ para los procesadores MicroBlaze y ARM Cortex-A9 (gcc)
- Depurador para los procesadores MicroBlaze y ARM Cortex-A9 (gdb)

➤ Board support packages (BSPs)

- Stand-alone BSP
 - Drivers para dispositivos básicos y utilidades de Xilinx
 - NOT an RTOS

Vista de Vivado

➤ Paneles configurables

➤ A: Administración de proyecto

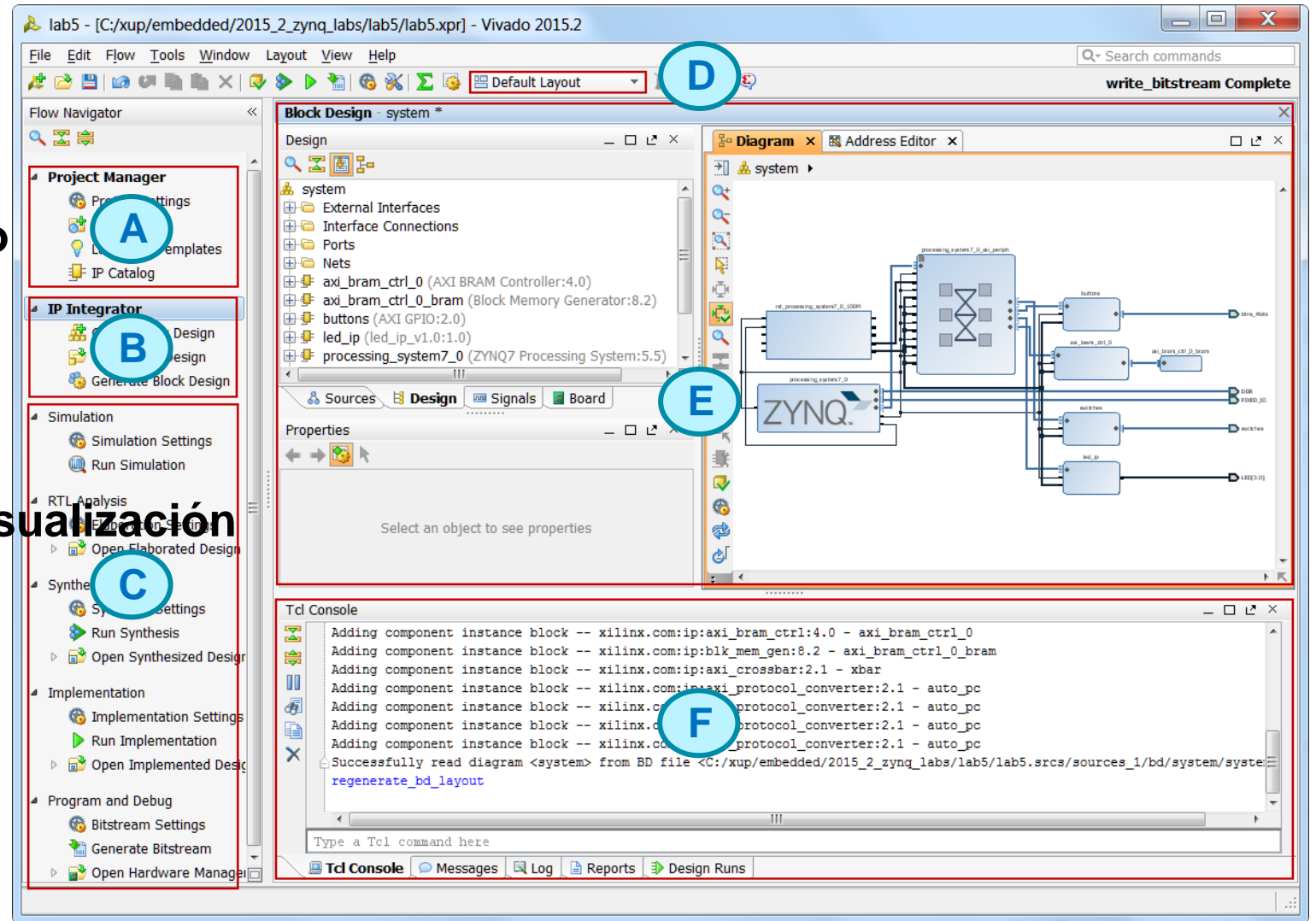
➤ B: IP Integrator

➤ C: FPGA Flow

➤ D: Selección de Layout

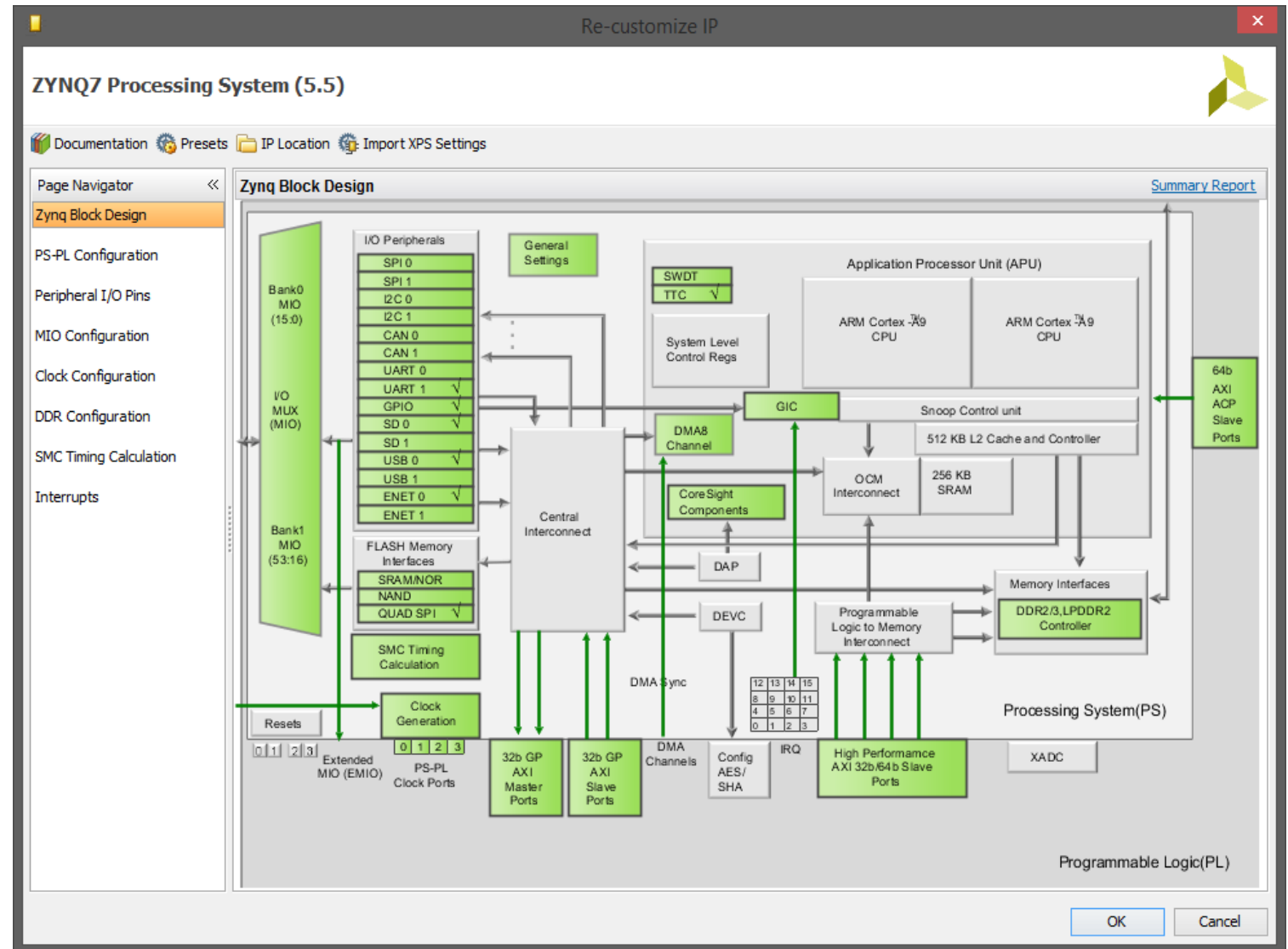
➤ E: Project view/Panel de previsualización

➤ F: Consola, Mensajes, Logs

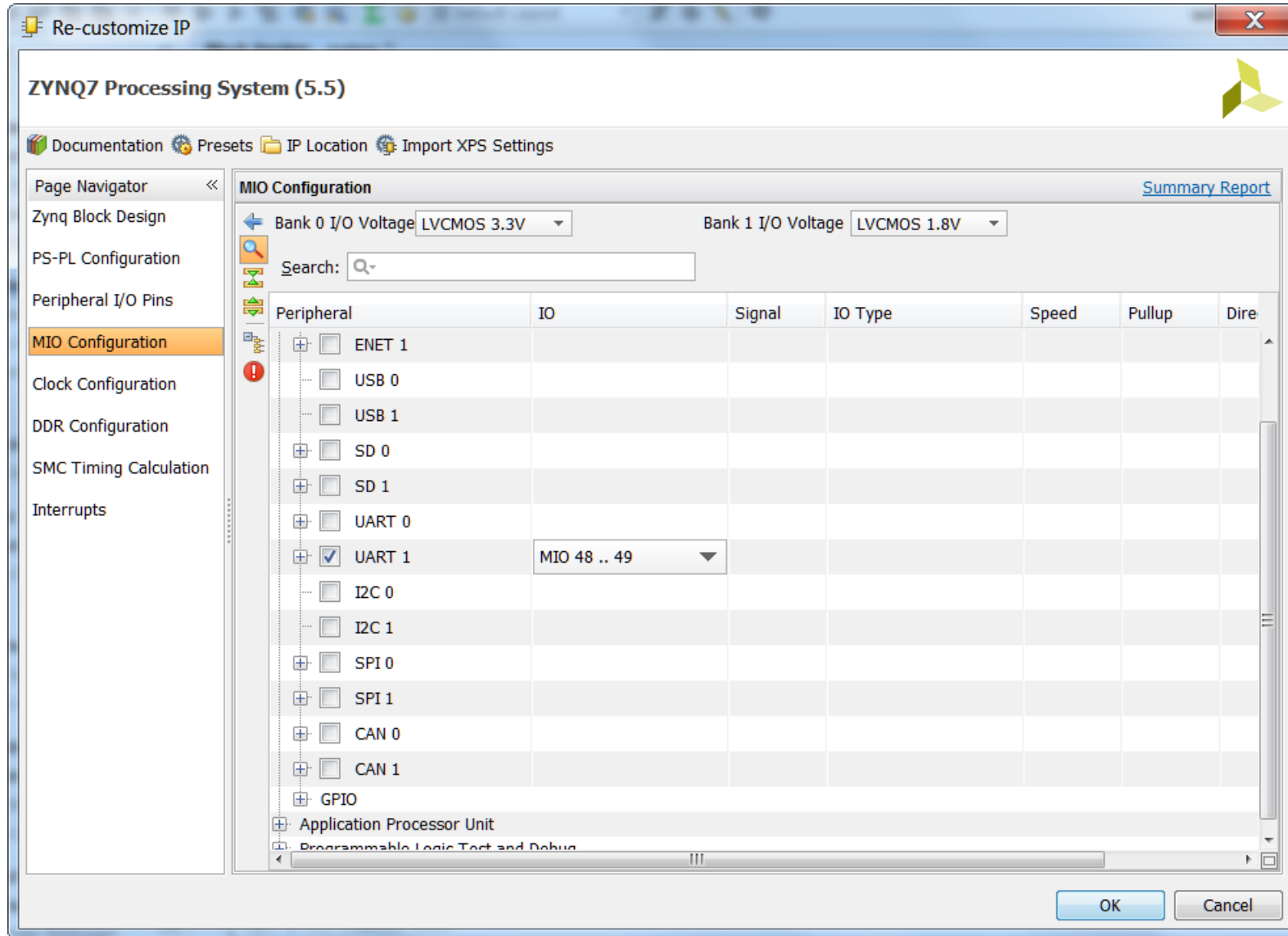


Zynq Customization Processing System

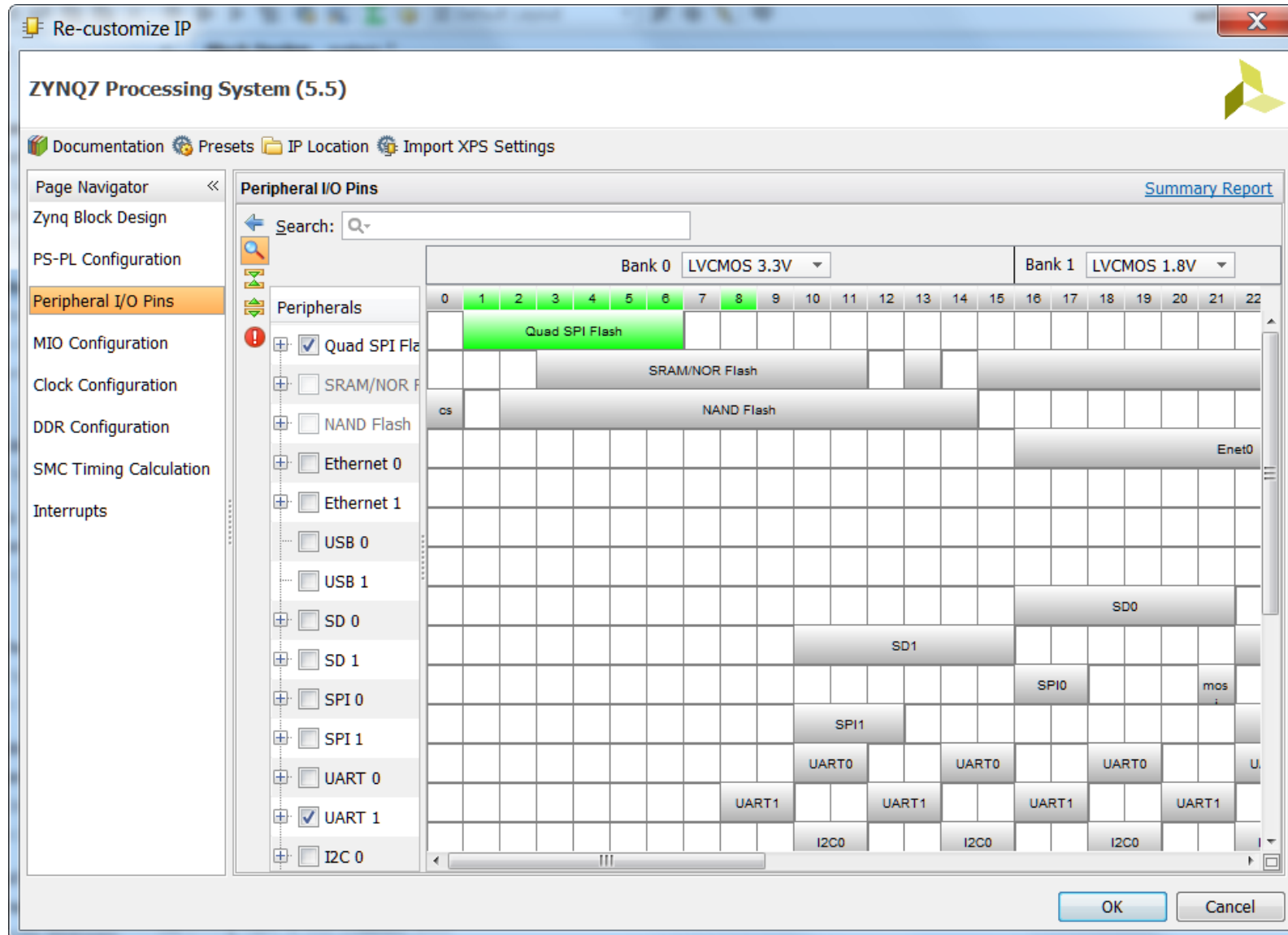
- Zynq Block Design
- Configuración de la interfaz PS-PL
- Pines de E/S de los periféricos
- Configuración MIO/Vista Tabla
- Configuración del reloj
- Configuración de la DDR
- Cálculo de tiempos SMC
- Interrupciones



Configuración MIO



Pines de E/S de los Periféricos



Archivos de Proyecto

➤ Top level Directory

- .xpr Vivado Project File (xml file), log files, journal

➤ .srcs

- Archivos fuente del proyecto, archivos del IP Integrator

➤ .sim

- Archivos relacionados con la simulación

➤ .runs

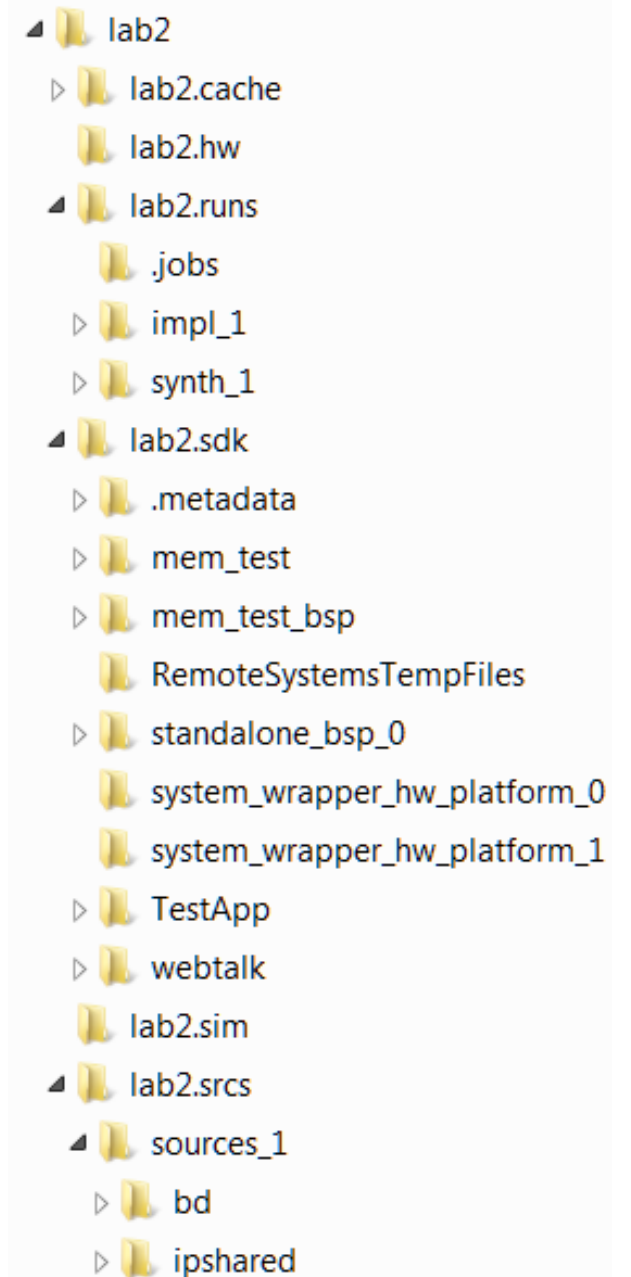
- Síntesis, corridas de implementación

➤ .sdk

- Directorio SDK Export, Plataforma de Hardware (xml)

➤ .cache

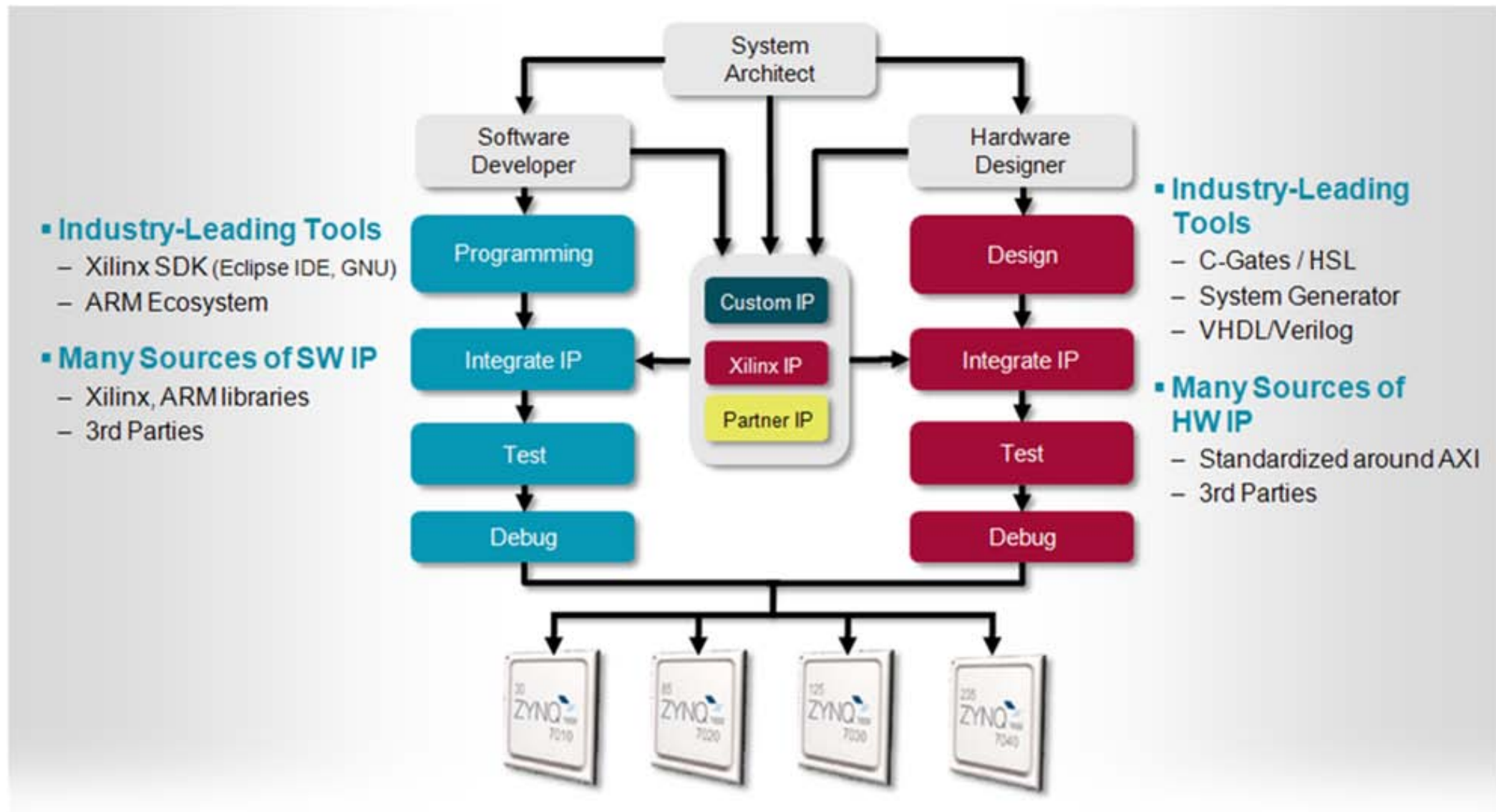
- Archivos temporales



Temario

- Componente Procesador Embebido
- Reseña de Vivado para diseño de sistemas embebidos
- ***Flujo de diseño de sistemas embebidos***
- Creación de plataforma de hardware
- Plataforma de software SDK
- Resumen

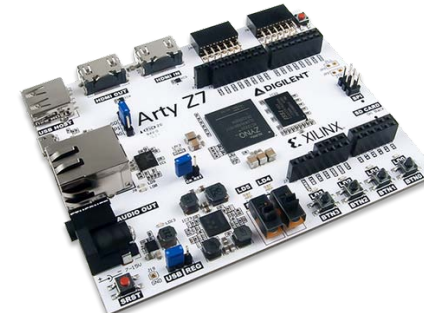
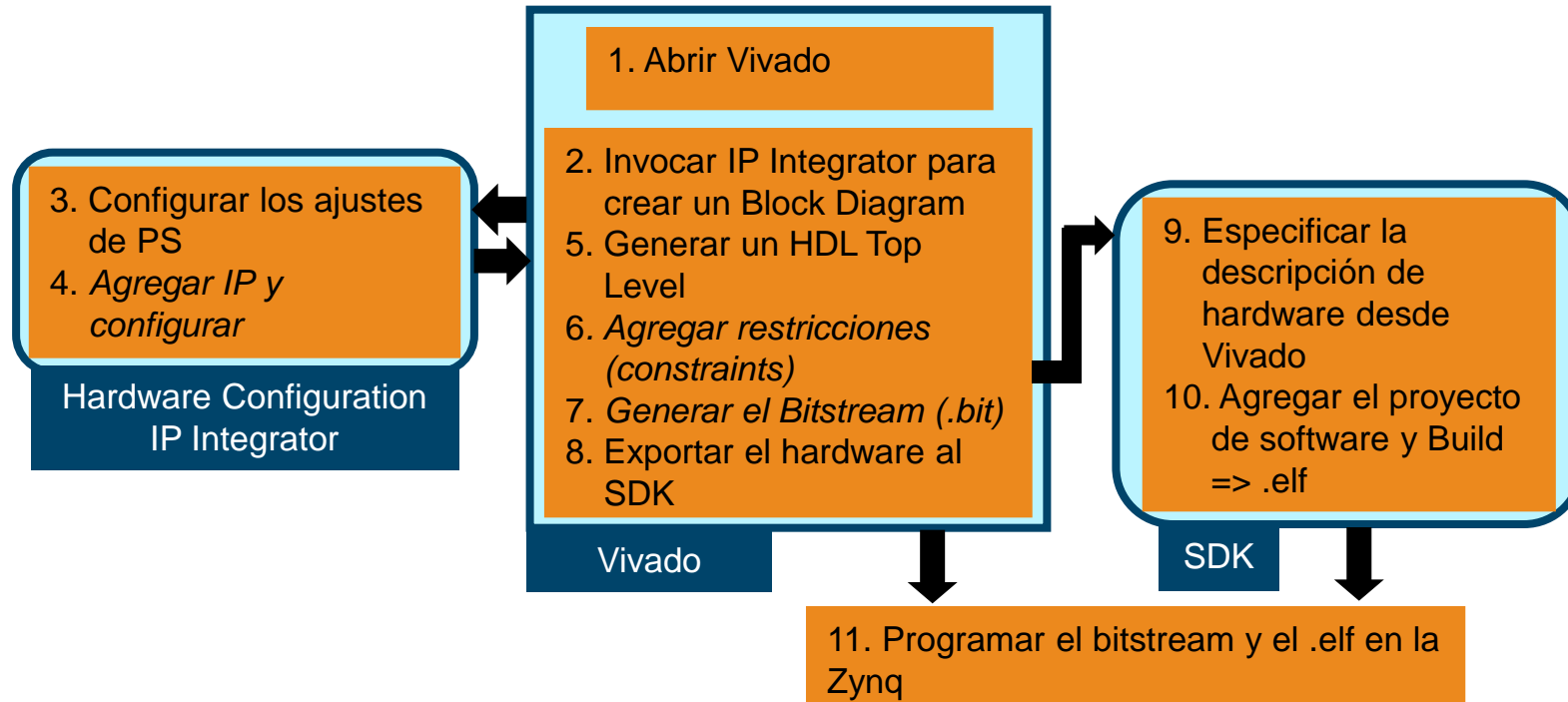
Flujo de Diseño de un Sistema Embebido para Zynq-7000 AP SoC



Diseño de un Sistema Embebido usando Vivado

- Crear un nuevo proyecto en Vivado, o abrir uno existente
- Invocar a IP Integrator
- Construir (modificar) la porción de hardware del diseño embebido
- Crear (Actualizar) el top level HDL wrapper
- [opcional] Sintetizar cualquier componente no-embebido e implementarlo en Vivado
- Exportar la descripción de hardware, y lanzar el SDK
- Crear un nuevo software board support package y proyectos de aplicación en el SDK
- Compilar el software con el cross-compilador GNU en SDK
- [opcional] Descargar el bitstream de la lógica programable
- Usar el SDK para descargar el programa (el archivo .ELF)

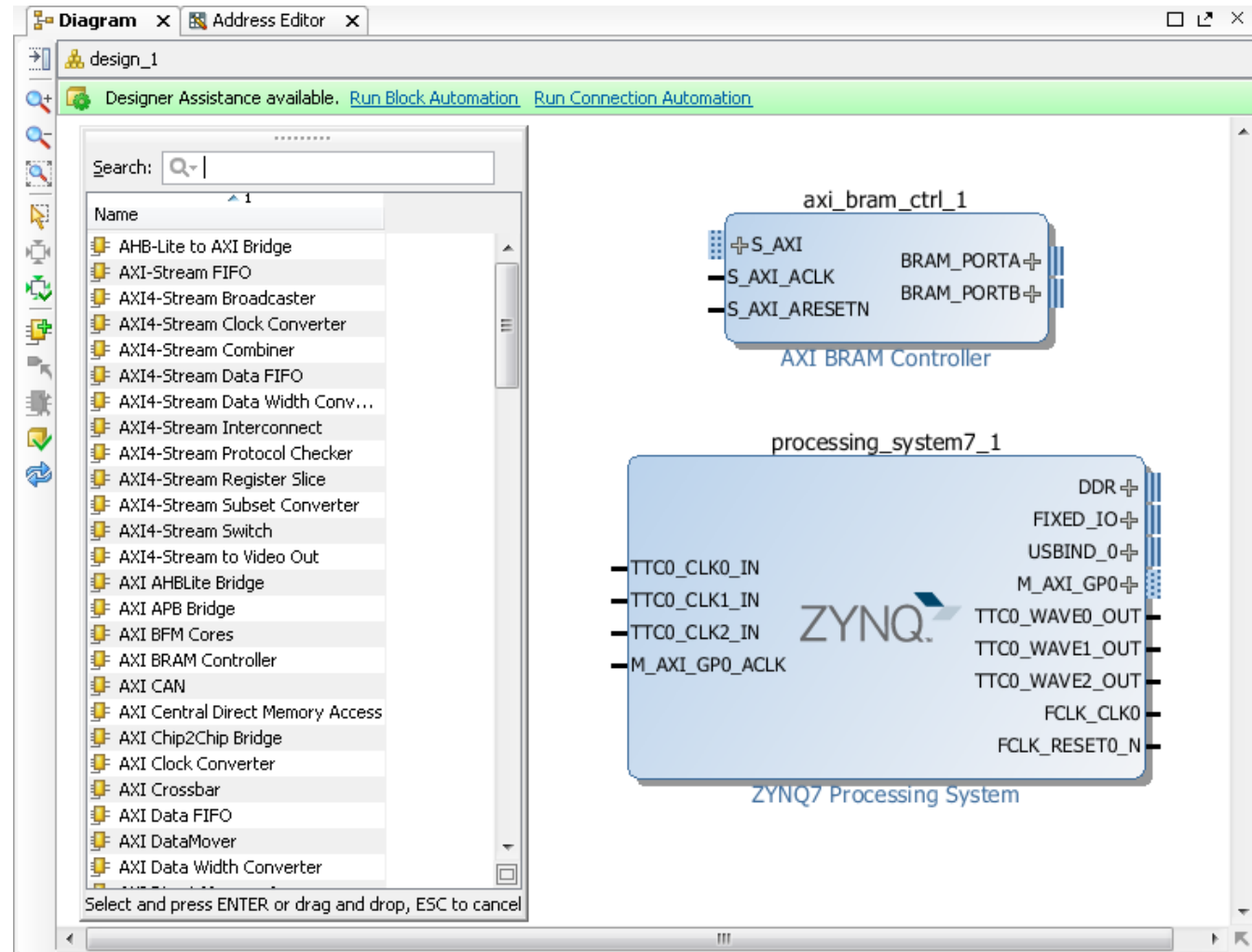
Diseño de un Sistema Embebido usando Vivado



Arty Z7-10

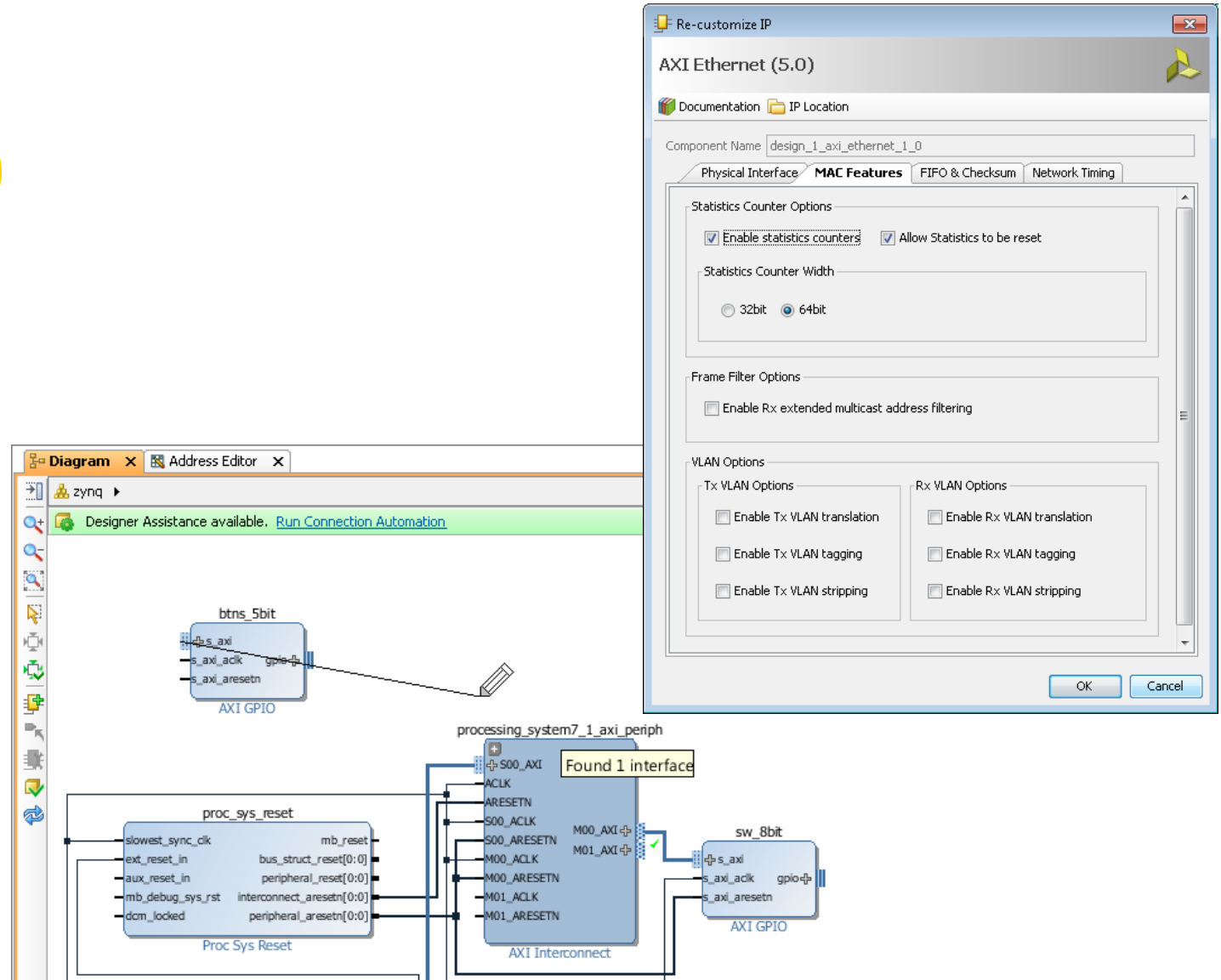
Integrator Block Diagram

- IP Integrator Block Diagram abre una ventana en blanco
- IP puede ser agregado desde el catálogo de IP
- Interfaz Drag and drop
- Ambiente de Diseño Inteligente
 - Asistencia de diseño
 - Automatización de conexiones
 - Resaltado de conexiones válidas
 - Agrupamiento, creación de bloques jerárquicos
- Puede crear e importar IP propia usando IP Packager



Configurando Hardware en el IP Integrator

- **Doble click sobre los bloques para acceder a las opciones de configuración**
- **Arrastrar el puntero para realizar conexiones**
 - Se resaltan las conexiones válidas
- **Conexiones automáticas**
 - Conecta automáticamente las interfaces reconocidas
- **Redibujo del sistema de manera automática**



Exportando al SDK

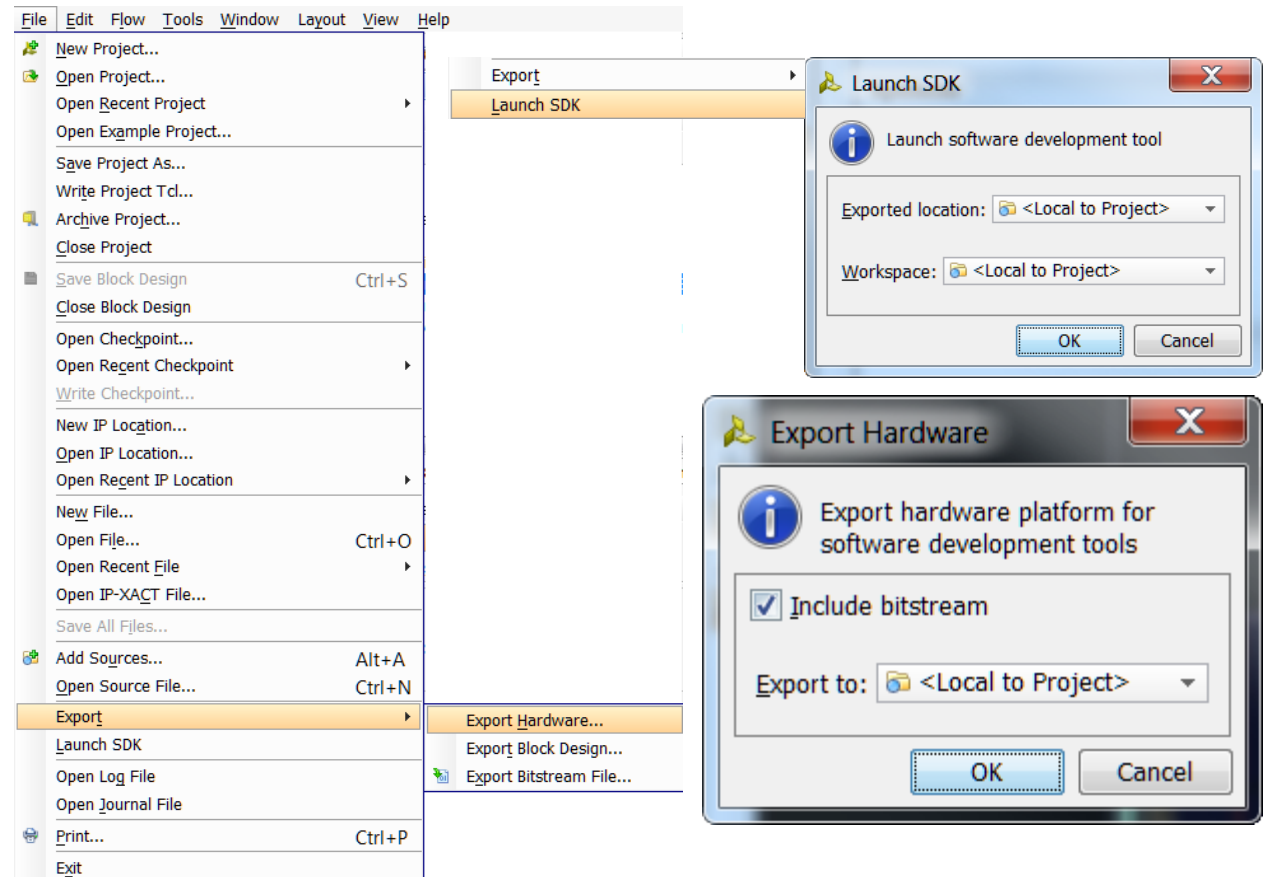
➤ Primero exportar hardware

- El archivo de formato, Hardware Description File (hdf), que contiene toda la información relevante será creado y ubicado en el directorio *.sdk
- Se incluye el bitstream, si es que fue generado

➤ Lanzar SDK

- El desarrollo de software se realiza con la herramienta SDK (Software Development Kit) de Xilinx

➤ La herramienta de SDK asociará entonces los proyectos de software de usuario al hardware



Flujo de Desarrollo de Software

➤ Crear un proyecto de plataforma de hardware

- Ejecutado automáticamente cuando la herramienta SDK es lanzada desde un proyecto de Vivado

➤ Crear BSP

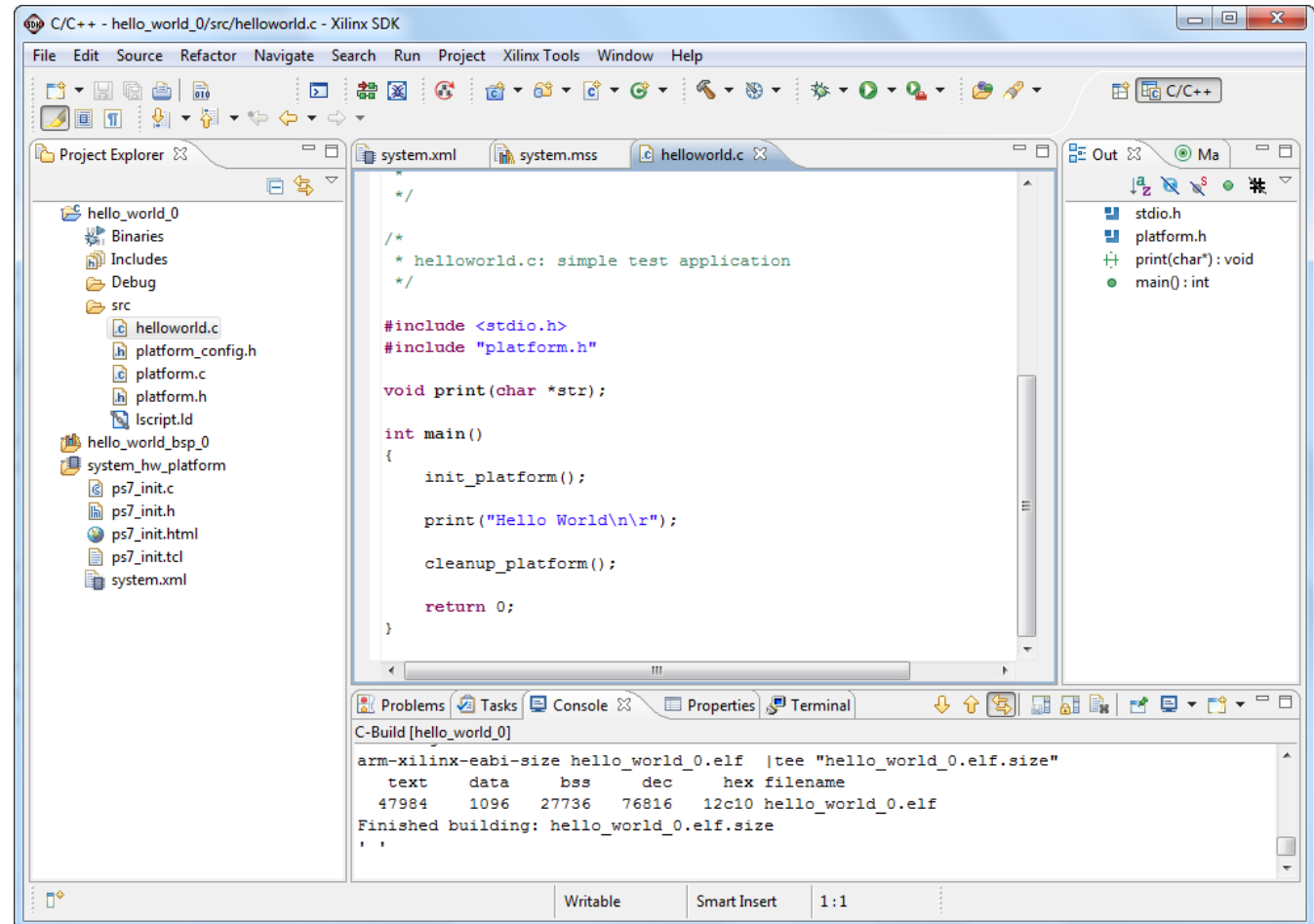
- Software de sistema, paquete de soporte de placa (board support package)

➤ Crear aplicación de software

➤ Crear el linker script

➤ Compilar el proyecto

- compilar, ensamblar, linquear el archivo de salida `<app_project>.elf`



Configurando la FPGA y Descargando una Aplicación

➤ Descarga del bitstream

- Sólo si es usado el PL (lógica programable)
- Archivo de entrada *<top_name>.bit*

➤ El bitstream puede ser descargado desde

- Vivado
- SDK

➤ Requiere que el cable de download esté conectado

Temario

- Componente Procesador Embebido
- Reseña de Vivado para diseño de sistemas embebidos
- Flujo de diseño de sistemas embebidos
- ***Creación de plataforma de hardware***
- Plataforma de software SDK
- Resumen

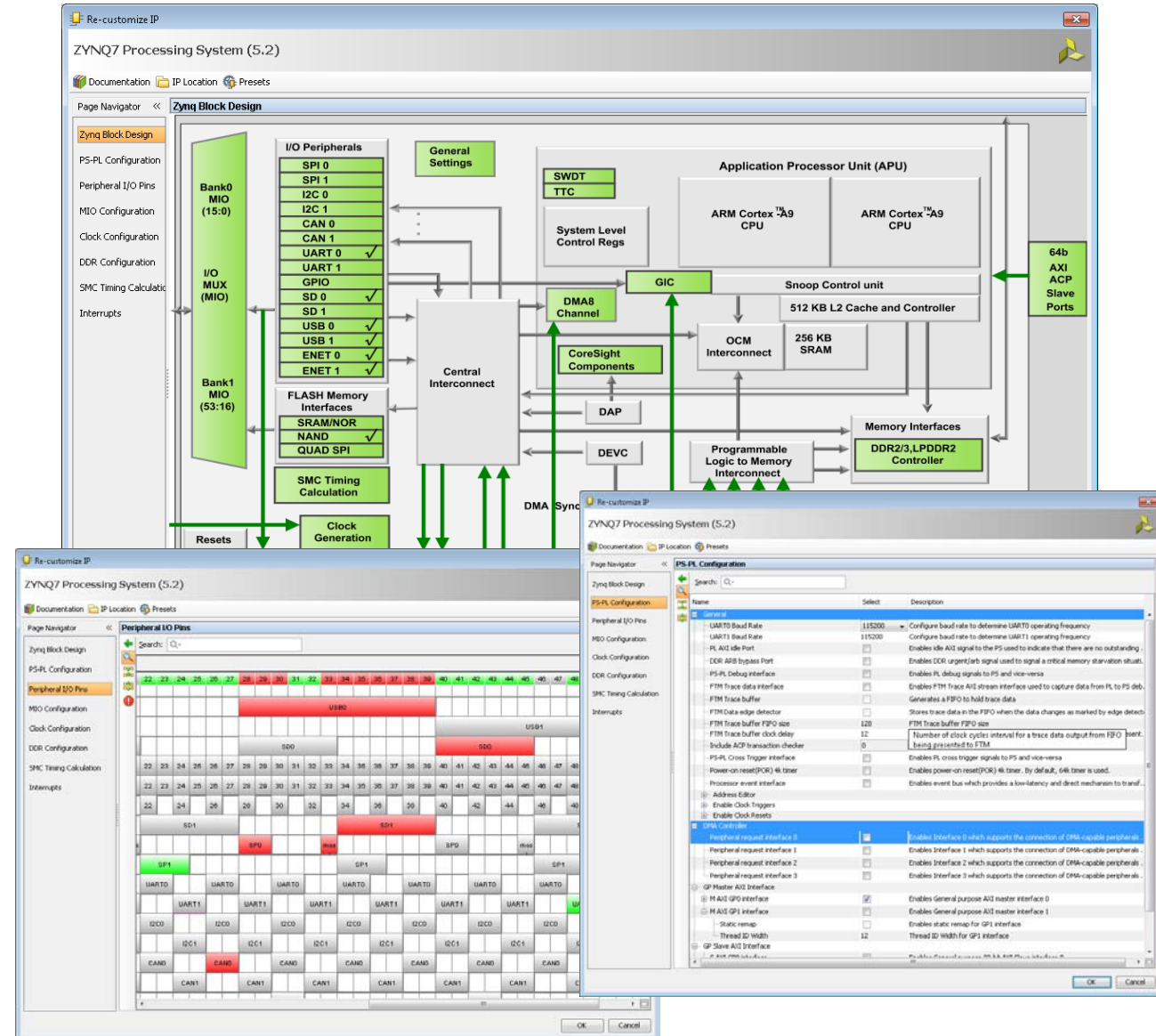
Configuración de la GUI de Zynq

➤ Provee una vista gráfica del PS para configurar

- los núcleos de ARM
- los periféricos de E/S
- el controlador DDR
- los sistemas de memoria

➤ Reparto de E/S entre pines dedicados de PS y E/S de lógica programable

➤ El Zynq-7000 AP SoC PS es configurado a través de un conjunto de registros de configuración mapeados en memoria



Configuración del Reloj

➤ Configuración del Reloj

- La frecuencia de entrada puede ser establecida
 - Processor, DDR
- Todas las frecuencias de reloj IOP pueden ser establecidas
- Los relojes de fábrica de la PL pueden ser habilitados y configurados
- Establecimiento de Timers

Page Navigator <<

- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration**
- DDR Configuration
- SMC Timing Calculation
- Interrupts

Clock Configuration [Summary Report](#)

Basic Clocking Advanced Clocking

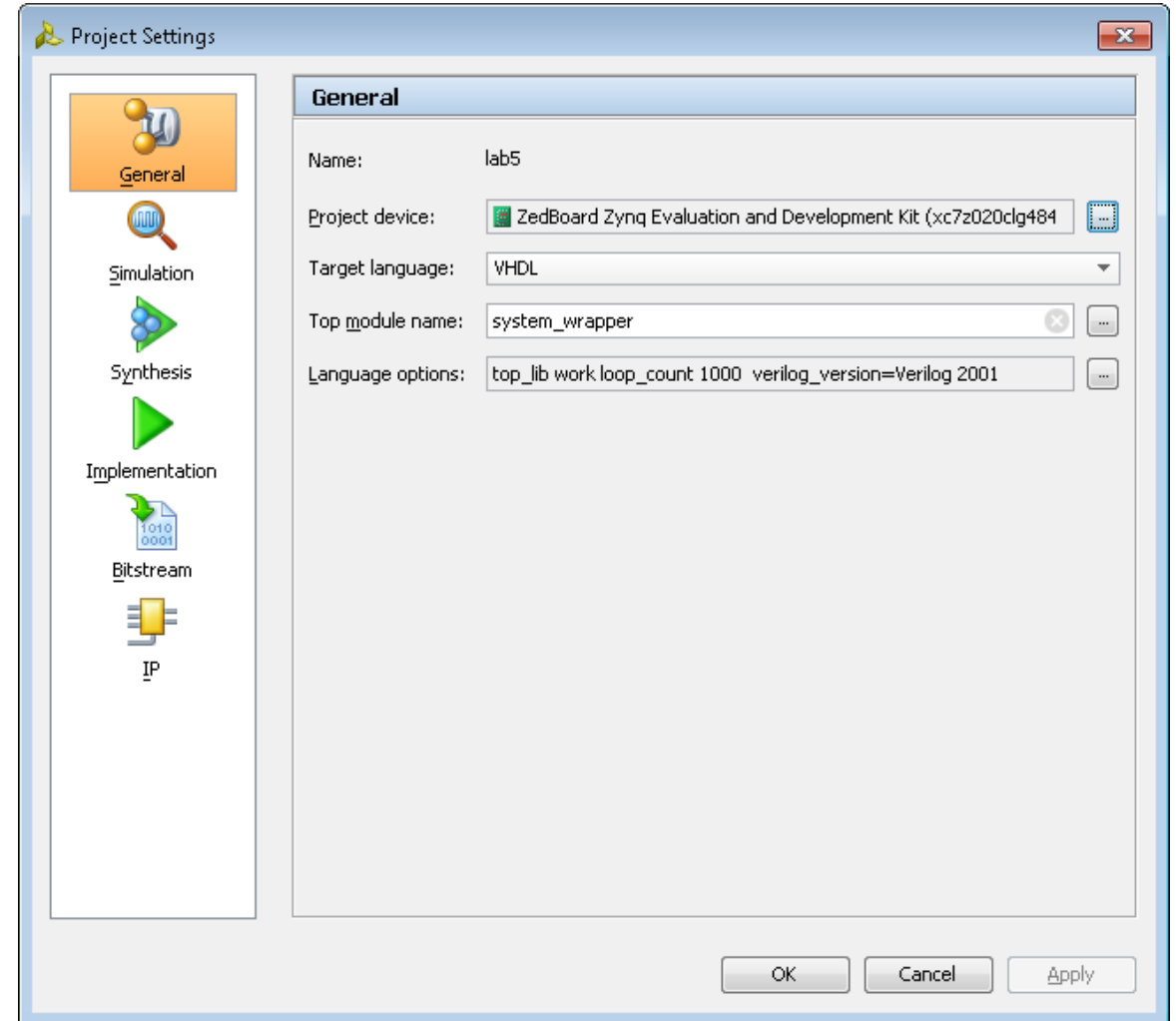
Input Frequency (MHz) 33.333333 CPU Clock Ratio 6:2:1

Search: Q

Component	Clock Source	Requested Frequ...	Actual Frequency...	Range(MHz)
Processor/Memory Clocks				
CPU	ARM PLL	666.666667	666.666687	50.0 : 667.0
DDR	DDR PLL	533.333313	533.333374	200.000000 : 534.0000...
IO Peripheral Clocks				
SMC	IO PLL	100	10.000000	10.000000 : 100.000000
QSPI	IO PLL	200.000000	200.000000	10.000000 : 200.000000
ENET0	IO PLL	1000 Mbps	125.000000	
ENET1	IO PLL	1000 Mbps	10.000000	
SDIO	IO PLL	50	50.000000	10.000000 : 125.000000
SPI	IO PLL	166.666666	10.000000	0.000000 : 200.000000
PL Fabric Clocks				
<input checked="" type="checkbox"/> FCLK_CLK0	IO PLL	100.000000	100.000000	0.100000 : 250.000000
<input type="checkbox"/> FCLK_CLK1	IO PLL	150.000000	142.857132	0.100000 : 250.000000
<input type="checkbox"/> FCLK_CLK2	IO PLL	50.000000	50.000000	0.100000 : 250.000000
<input type="checkbox"/> FCLK_CLK3	IO PLL	50	50.000000	0.100000 : 250.000000
System Debug Clocks				
Timers				

Ajustes del Proyecto

- Se accede desde el *flow navigator*
- La configuración por defecto es la típicamente usada
- Especificar/cambiar el dispositivo destino
 - Arquitectura, tamaño, encapsulado, grado de velocidad (speed grade)
- Simulación, Síntesis, Implementación, opciones de Bitstream
- Directorio del repositorio de IP
 - Provee acceso a IP personalizada que no está presente en la estructura de directorios del proyecto actual



Temario

- Componente Procesador Embebido
- Reseña de Vivado para diseño de sistemas embebidos
- Flujo de diseño de sistemas embebidos
- Creación de plataforma de hardware
- ***Plataforma de software SDK***
- Resumen

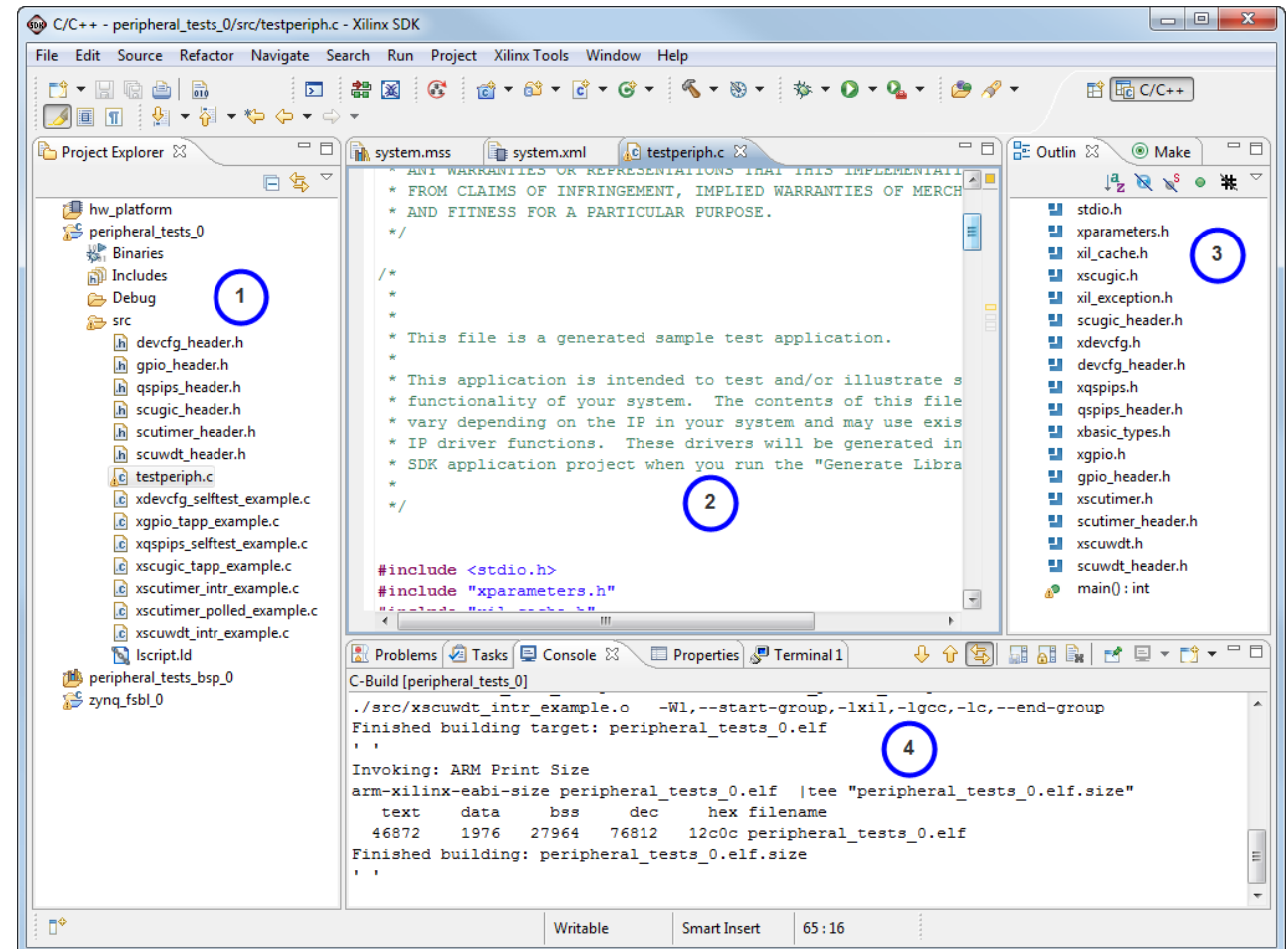
Software Development Kit (SDK)

- Entorno de diseño de software con funciones completas
- Herramienta separada de Vivado – puede ser instalada de manera independiente para equipos de software
- Basado en el popular IDE open-source Eclipse
- Usado sólo para aplicaciones de software; el diseño de hardware y sus modificaciones son realizadas en Vivado
- Entorno integrado para depuración fluida de targets embebidos
- Entorno de diseño de software sofisticado con muchas opciones y funciones con soporte para:
 - Múltiples procesadores
 - Múltiples plataformas de software
 - Múltiples aplicaciones de software
- Editor de código C/C++ con funciones completas y navegador de error



Vistas de los Workbenches del SDK

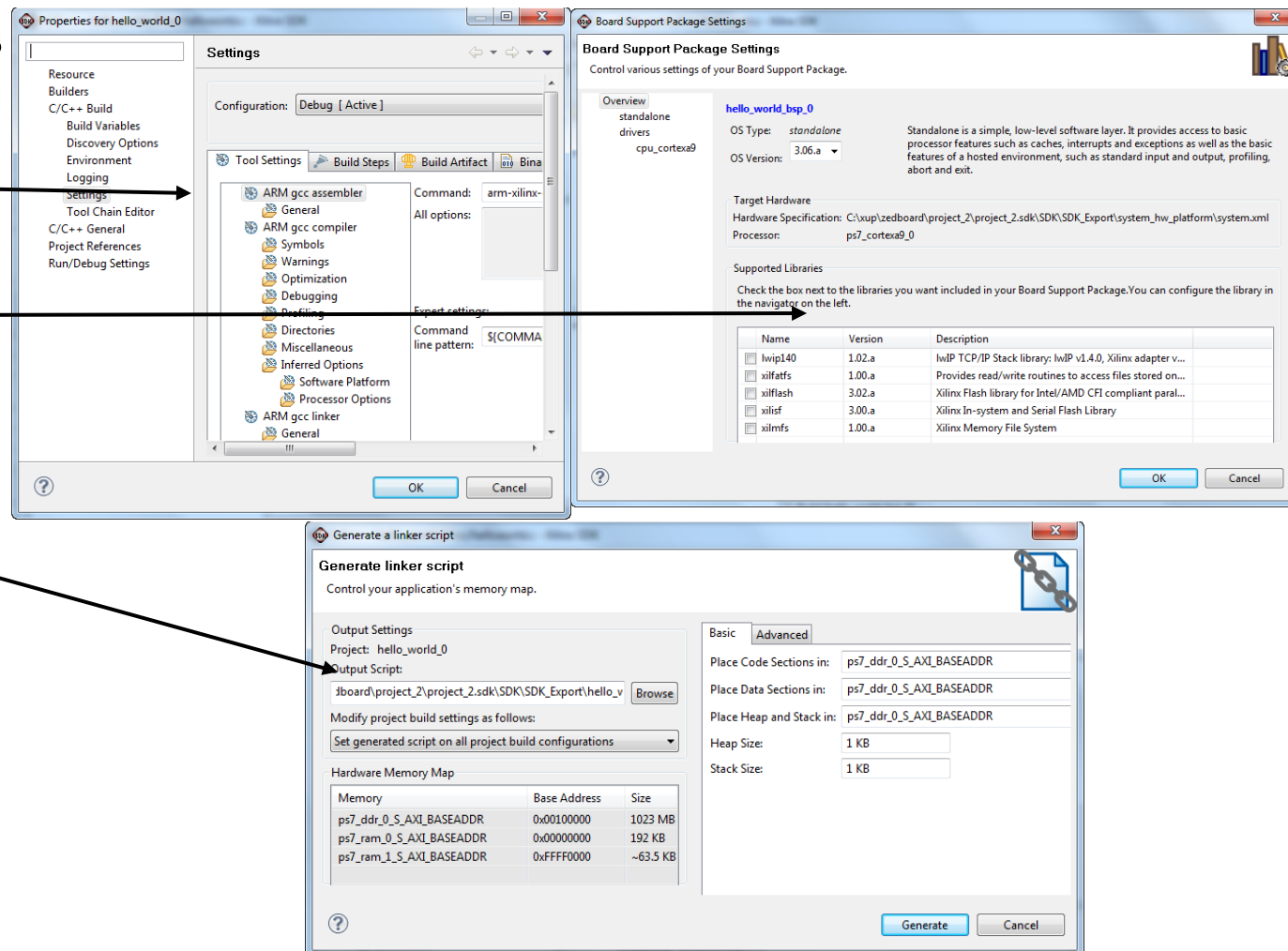
1. C/C++ project outline muestra los elementos de un proyecto con iconos para una sencilla identificación
2. Editor C/C++ para la creación de software integrado
3. Code outline muestra los elementos del archivo de software bajo desarrollo, con iconos para una sencilla identificación
4. Problemas, Consola, vista de la información de salida asociada con el flujo de desarrollo del software



Ajuste del Administrador del Software

➤ El software es manejado en tres áreas principales

- Opciones de Compilador/Linker
 - Programa de aplicación
- Configuraciones de la plataforma de Software
 - Board support package
- Generación del Linker Script
 - Asignando software a recursos de memoria



➤ Cubierto en más detalle luego

Temario

- Componente Procesador Embebido
- Reseña de Vivado para diseño de sistemas embebidos
- Flujo de diseño de sistemas embebidos
- Creación de plataforma de hardware
- Plataforma de software SDK
- ***Resumen***

Resumen

- **Vivado incluye todas las herramientas, documentación, e IP necesaria para construir sistemas embebidos**
- **IPI es una herramienta de diseño a nivel de sistema que incrementa la productividad, permitiendo que los diseños se completen más rápidamente**
- **El Software Development Kit (SDK) es un entorno de desarrollo de software completo para aplicaciones de software**
- **Un embedded processing system component está construido con IP provista en el catálogo de IP. Los diseñadores pueden también agregar su propia IP a este catálogo**
- **El asistente de configuración del Sistema de Procesamiento (PS Configuration wizard) permite acceso a varias características configurables del PS**