

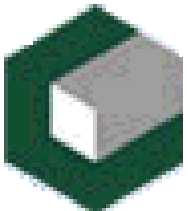
Software Development Kit SDK

Maestría en Sistemas Digitales

Alejandro J. Cabrera Sarmiento

Dpto. de Automática y Computación
Universidad Tecnológica de La Habana
“José Antonio Echeverría”
CUJAE

alex@automatica.cujae.edu.cu



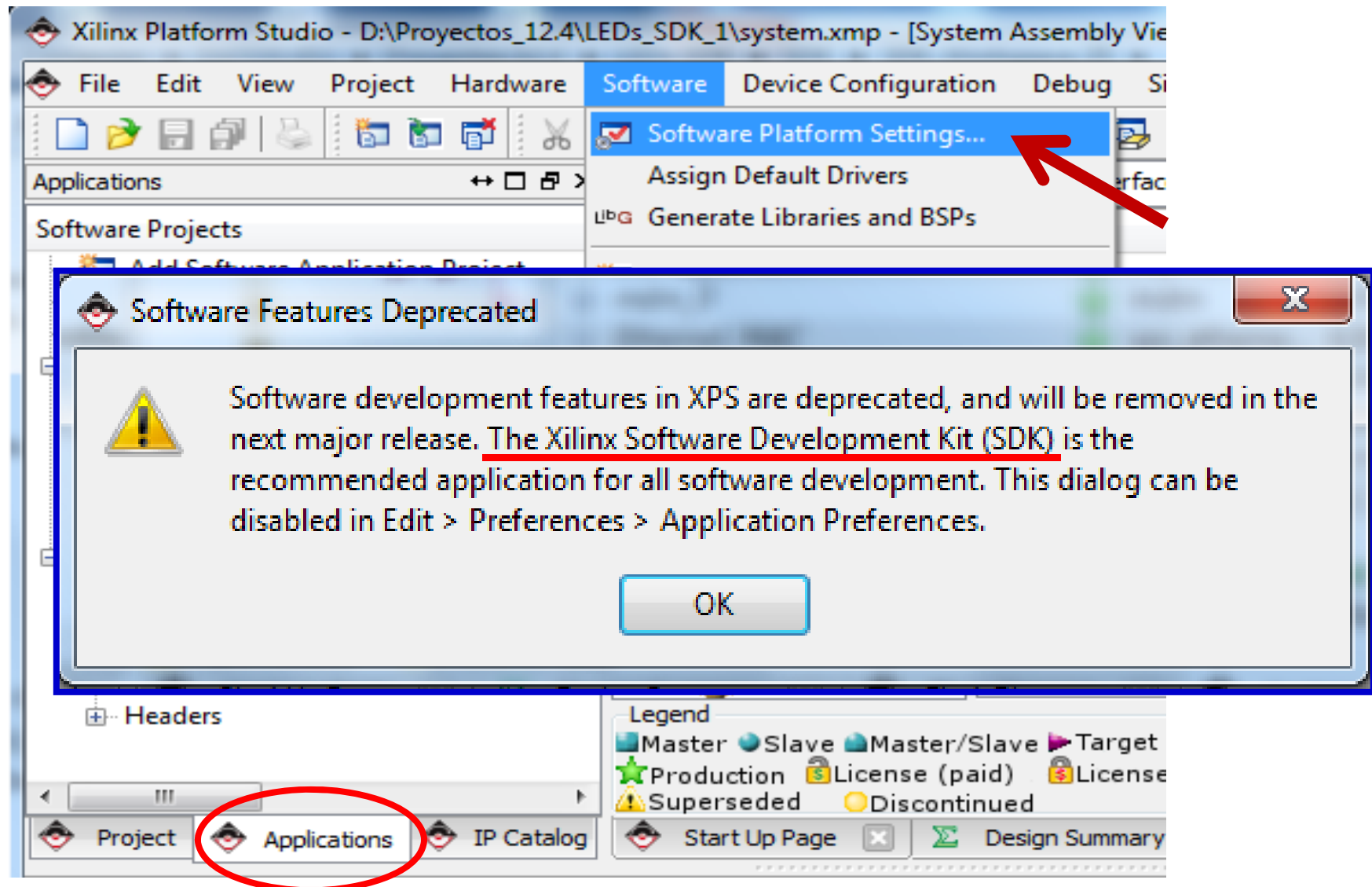
Sumario

- Características de SDK
- Desarrollo de aplicaciones con SDK
 - Exportar diseño HW
 - Configuración SW de la plataforma (BSP)
 - Creación de proyecto SW
 - Compilación y descarga al FPGA
- Depuración de aplicaciones con SDK



Desarrollo de aplicaciones

- Puede hacerse en XPS hasta versión 12.4



Software Development Kit



- Entorno de desarrollo de aplicaciones SW
- Basado en Eclipse
 - Programa de código abierto multiplataforma para desarrollo de aplicaciones
- Facilidades:
 - Configurar la plataforma SW
 - Crear y compilar aplicación
 - Generar *download.bit*
 - Descargar en el FPGA
 - Enlace con XMD

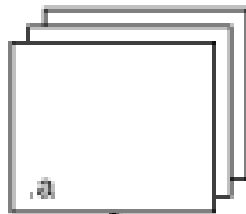
Flujo de desarrollo SW con SDK

Plataforma
SW

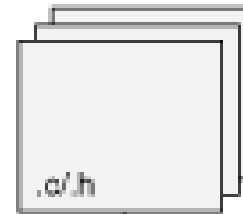


Libgen

Libraries

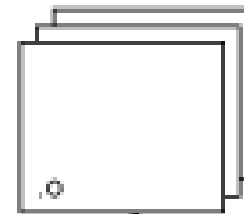


Aplicación
SW

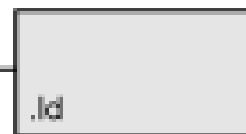


gcc/g++

Object Files



Linker Script



ld

Executable



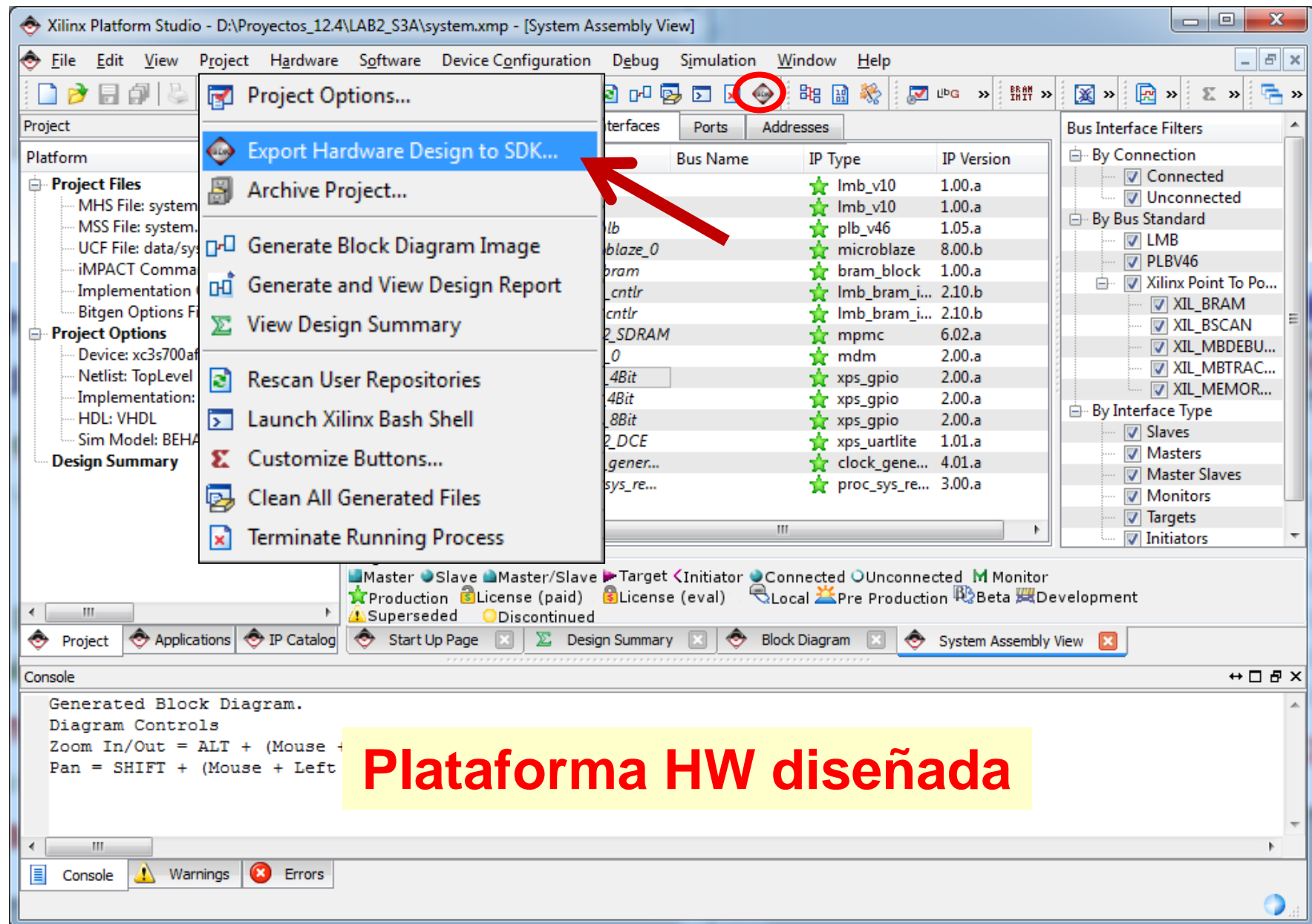
30110207



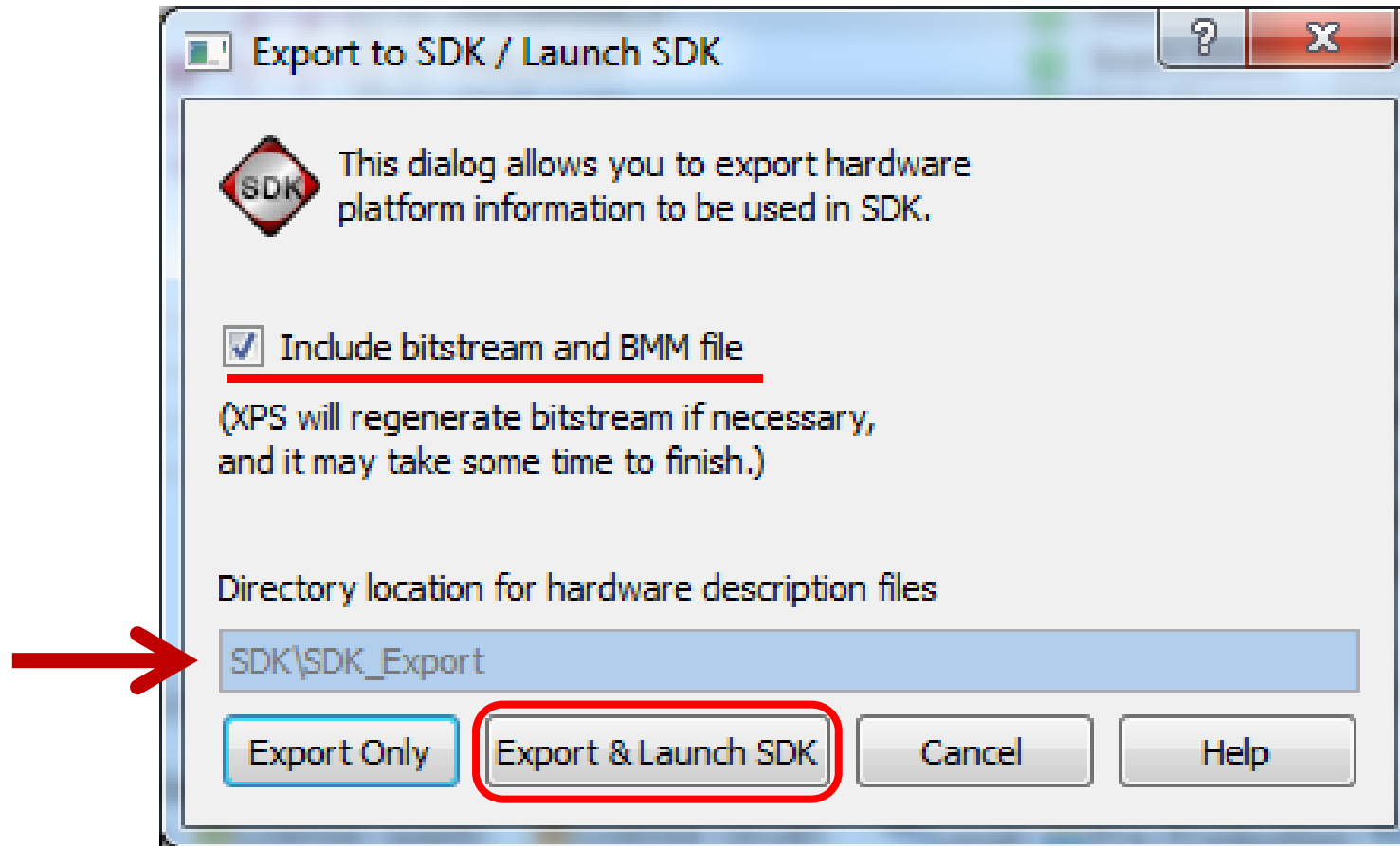
Flujo de diseño con SDK

1. Diseñar plataforma HW con **XPS**
2. Exportar diseño a **SDK**
 - Síntesis, implementación y generación de *system.bit*
 - Creación de espacio de trabajo (*Workspace*)
3. Configurar plataforma SW en **SDK**
 - **Board Support Package**: equivalente a *MSS*
 - Ejecuta *LibGen*: compilación de drivers, bibliotecas y SO
4. Crear proyecto con la aplicación SW de usuario
 - Contiene los **ficheros fuentes** de la aplicación
5. [Modificar el mapeo de memoria con **linker script**]
6. Implementar la aplicación SW
 - Compilación y enlace
7. Descargar el bitstream sobre el FPGA
 - Genera *download.bit*
8. Ejecutar y depurar el SW sobre el FPGA

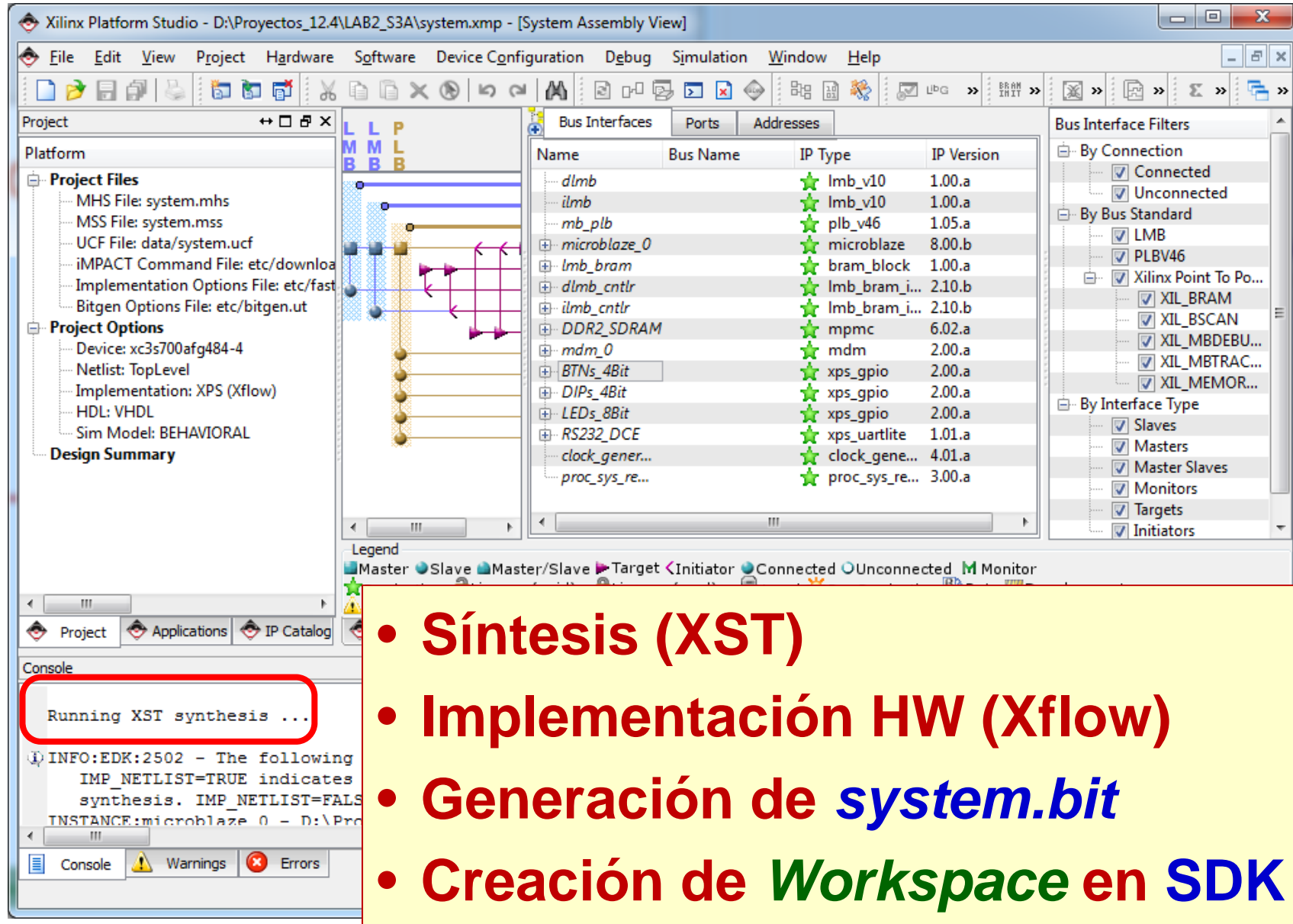
1. Diseño plataforma HW



2. Exportar HW a SDK



2. Exportar HW a SDK (cont.)



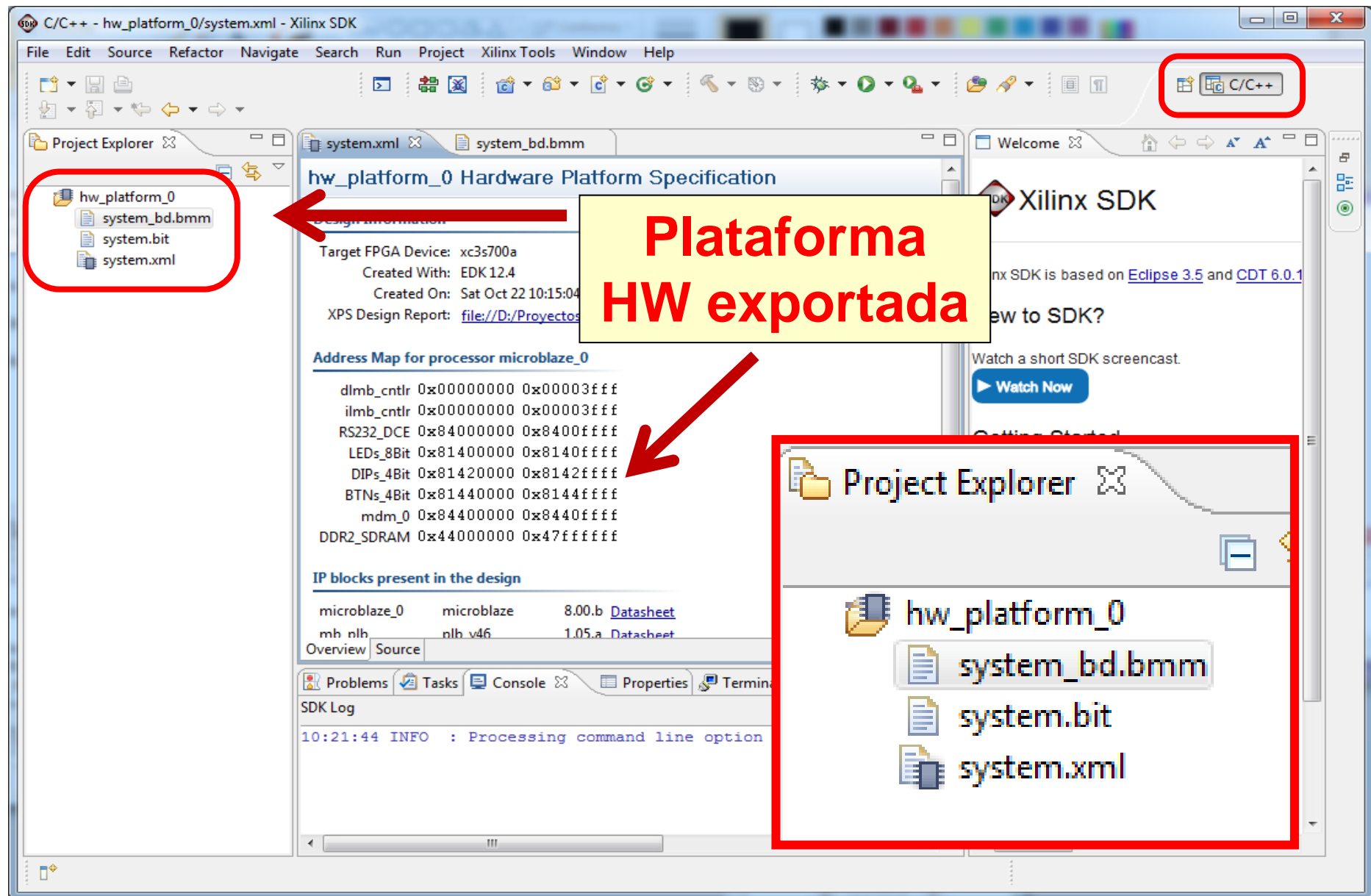
The screenshot displays the Xilinx Platform Studio (XPS) interface in the System Assembly View. The main window shows a block diagram of the hardware design. The right-hand pane lists the bus interfaces and their properties:

Name	Bus Name	IP Type	IP Version
dlmb		lmb_v10	1.00.a
ilmb		lmb_v10	1.00.a
mb_plb		plb_v46	1.05.a
microblaze_0		microblaze	8.00.b
lmb_bram		bram_block	1.00.a
dlmb_cntlr		lmb_bram_i...	2.10.b
ilmb_cntlr		lmb_bram_i...	2.10.b
DDR2_SDRAM		mpmc	6.02.a
mdm_0		mdm	2.00.a
BTNs_4Bit		xps_gpio	2.00.a
DIPs_4Bit		xps_gpio	2.00.a
LEDs_8Bit		xps_gpio	2.00.a
RS232_DCE		xps_uartlite	1.01.a
clock_gener...		clock_gene...	4.01.a
proc_sys_re...		proc_sys_re...	3.00.a

The bottom console pane shows the message "Running XST synthesis ...".

- Síntesis (XST)
- Implementación HW (Xflow)
- Generación de *system.bit*
- Creación de *Workspace* en SDK

Interfaz de SDK

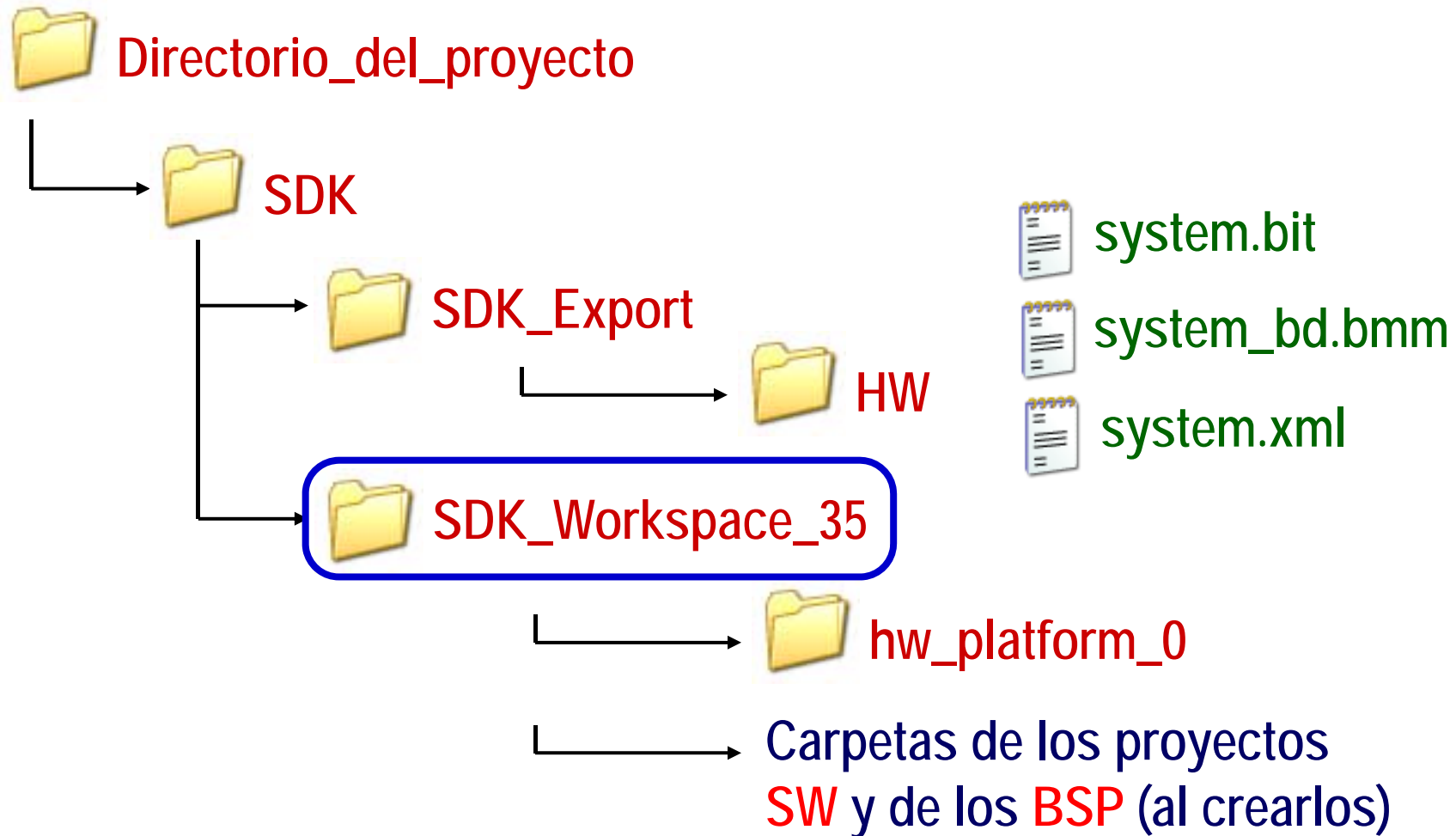


Workspace

- **Directorio donde se almacena la información de un proyecto SDK**
 - Se crea al abrir SDK
- **Incluye:**
 - Plataformas HW
 - Plataformas SW (Board Support Package)
 - Proyectos SW
- **Pueden crearse varios Workspace**



Estructura del proyecto

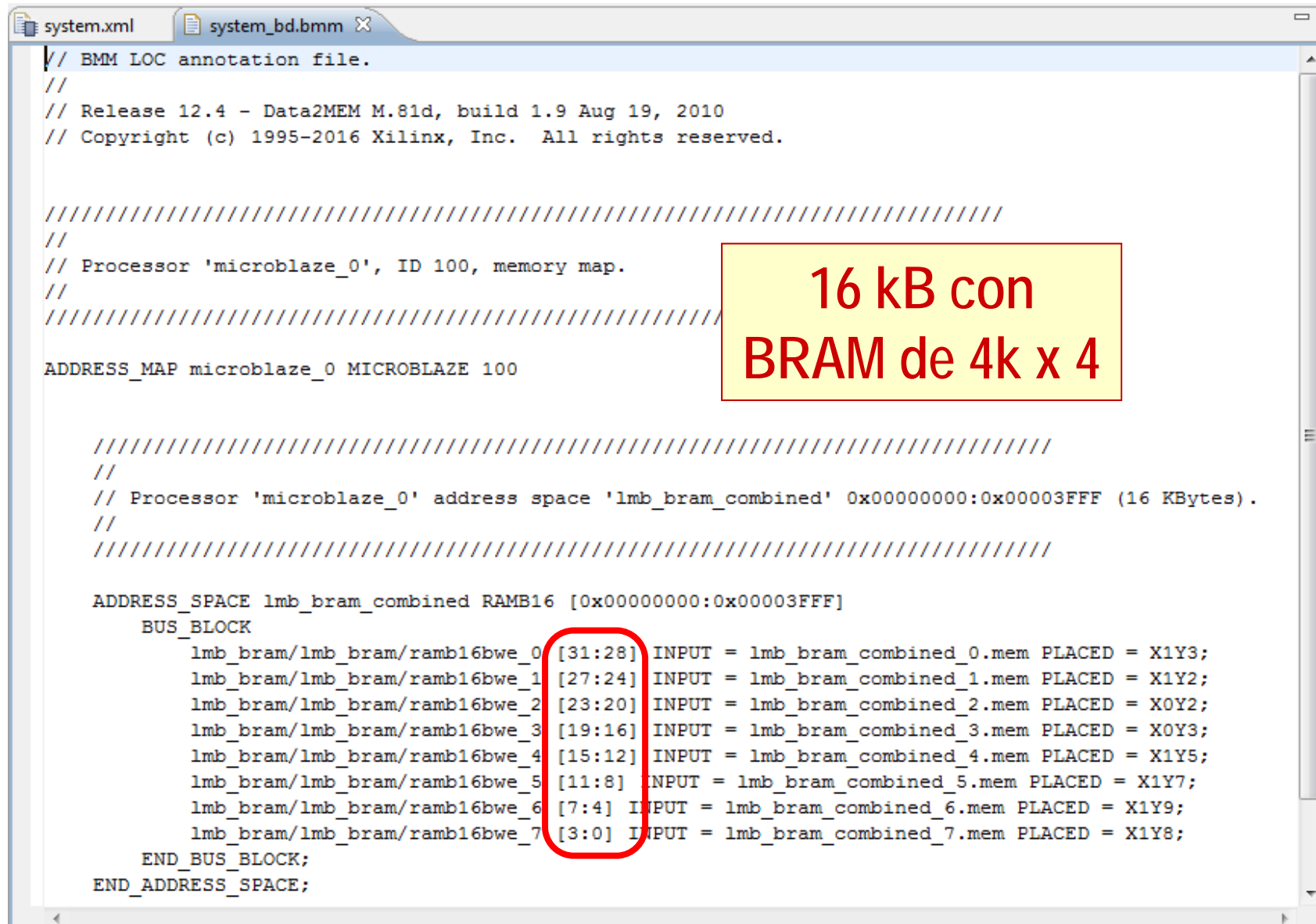


Plataforma HW

- **system-bd.bmm**: descripción de las conexiones de las **BRAM** utilizadas en la configuración HW.
 - Se utiliza para modificar el **system.bit** y obtener el **download.bit** en correspondencia con el programa.
- **system.bit**: interconexiones HW del sistema de procesamiento.
- **system.xml**: descripción HW del sistema de procesamiento resumido en un archivo **XML**
 - Equivalente a **.mhs**



Fichero *system_bd.bmm*



```
// BMM LOC annotation file.
//
// Release 12.4 - Data2MEM M.81d, build 1.9 Aug 19, 2010
// Copyright (c) 1995-2016 Xilinx, Inc. All rights reserved.

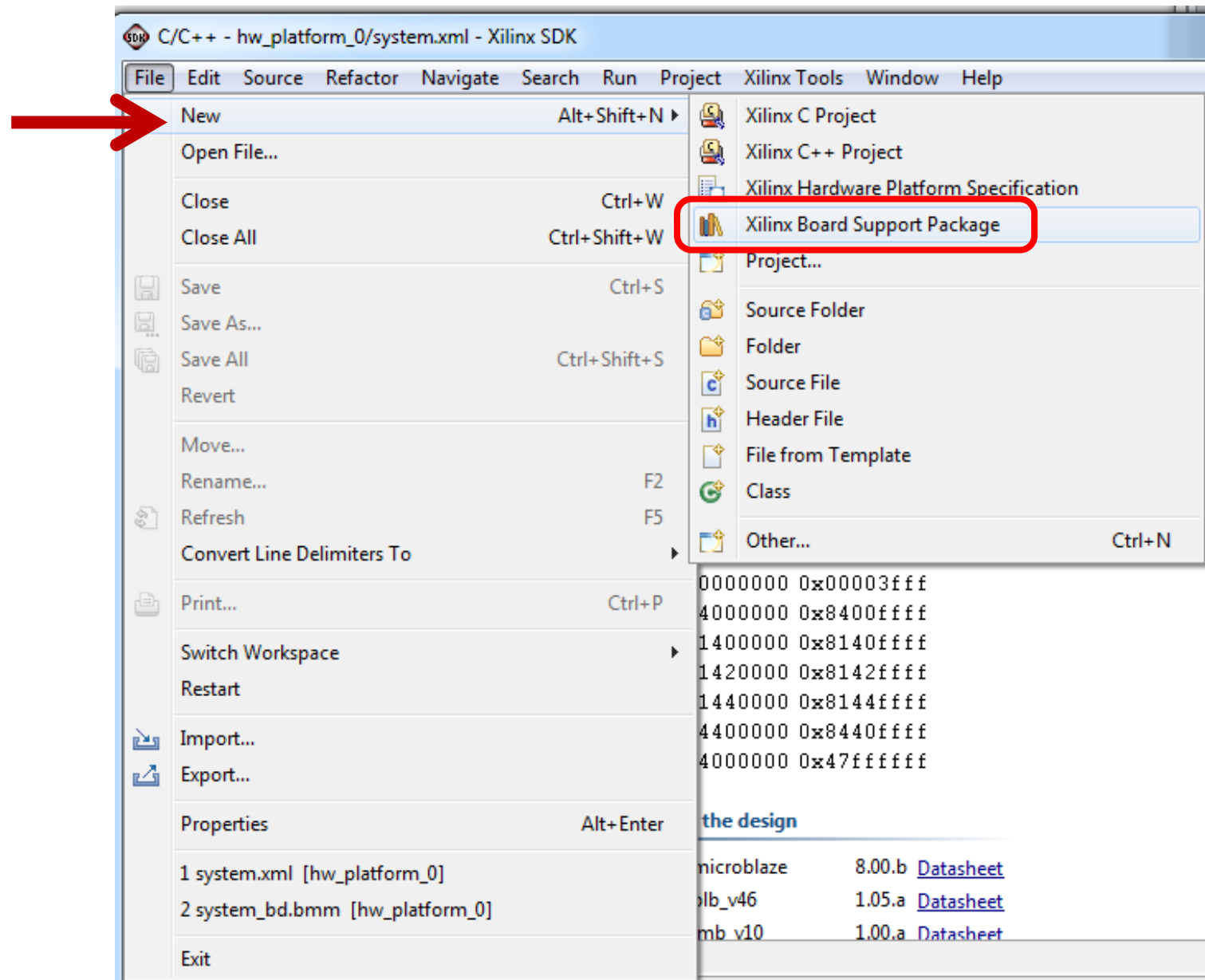
////////////////////////////////////
//
// Processor 'microblaze_0', ID 100, memory map.
//
////////////////////////////////////

ADDRESS_MAP microblaze_0 MICROBLAZE 100

////////////////////////////////////
//
// Processor 'microblaze_0' address space 'lmb_bram_combined' 0x00000000:0x00003FFF (16 KBytes).
//
////////////////////////////////////

ADDRESS_SPACE lmb_bram_combined RAMB16 [0x00000000:0x00003FFF]
  BUS_BLOCK
    lmb_bram/lmb_bram/ramb16bwe_0 [31:28] INPUT = lmb_bram_combined_0.mem PLACED = X1Y3;
    lmb_bram/lmb_bram/ramb16bwe_1 [27:24] INPUT = lmb_bram_combined_1.mem PLACED = X1Y2;
    lmb_bram/lmb_bram/ramb16bwe_2 [23:20] INPUT = lmb_bram_combined_2.mem PLACED = X0Y2;
    lmb_bram/lmb_bram/ramb16bwe_3 [19:16] INPUT = lmb_bram_combined_3.mem PLACED = X0Y3;
    lmb_bram/lmb_bram/ramb16bwe_4 [15:12] INPUT = lmb_bram_combined_4.mem PLACED = X1Y5;
    lmb_bram/lmb_bram/ramb16bwe_5 [11:8] INPUT = lmb_bram_combined_5.mem PLACED = X1Y7;
    lmb_bram/lmb_bram/ramb16bwe_6 [7:4] INPUT = lmb_bram_combined_6.mem PLACED = X1Y9;
    lmb_bram/lmb_bram/ramb16bwe_7 [3:0] INPUT = lmb_bram_combined_7.mem PLACED = X1Y8;
  END_BUS_BLOCK;
END_ADDRESS_SPACE;
```

3. Configurar plataforma SW



Board Support Package (BSP)

- Configuración de la plataforma SW
- Colección de bibliotecas y drivers que forman la **capa más baja** de la aplicación SW
 - Sistema operativo
 - Bibliotecas
 - Drivers de periféricos
 - E/S estándar
 - Periférico de depuración
- Equivalente a **MSS**
- Al salvarlo, ejecuta **LibGen**
- Crea carpeta en el **Workspace**
- Pueden existir **varios BSP** para un mismo HW



3a. Crear BSP

New Board Support Package Project

Xilinx Board Support Package Project

Create a Board Support Package.

Project name: LAB2_bsp_0

☒ Use default location

Location: D:\Proyectos_12.4\LAB2_S3A\SDK\SDK_Workspace_35\LAB2_bsp_0 Browse...

Hardware Platform: hw_platform_0

CPU: microblaze_0

xilkernel
standalone

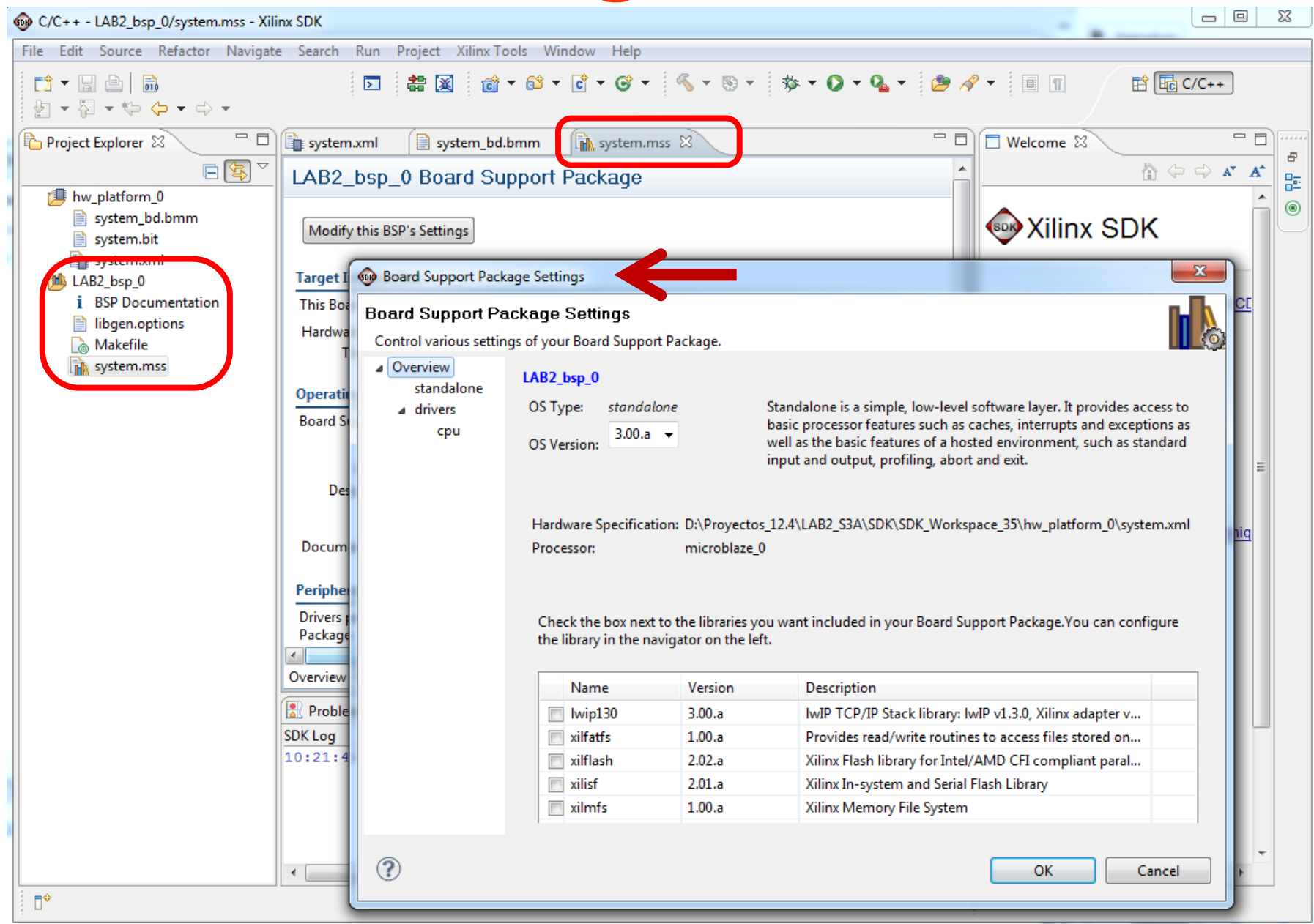
Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.

Finish Cancel

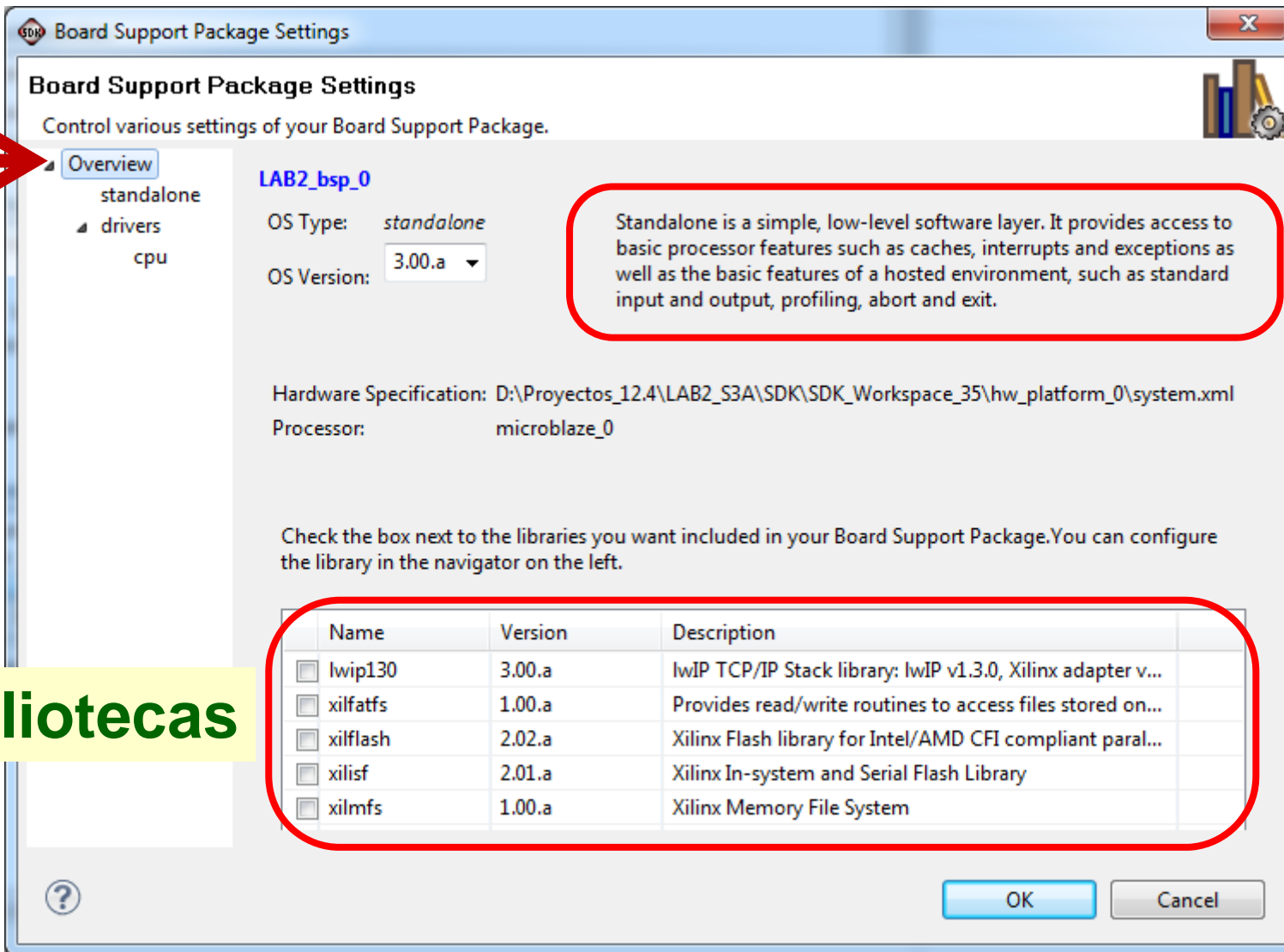
Especificar
plataforma HW

Seleccionar
SO

3b. Configurar BSP



3b. Configurar BSP (cont.)



Board Support Package Settings

Control various settings of your Board Support Package.

Overview

- standalone
- drivers
- cpu

LAB2_bsp_0

OS Type: *standalone*

OS Version: 3.00.a

Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.

Hardware Specification: D:\Proyectos_12.4\LAB2_S3A\SDK\SDK_Workspace_35\hw_platform_0\system.xml

Processor: microblaze_0

Check the box next to the libraries you want included in your Board Support Package. You can configure the library in the navigator on the left.

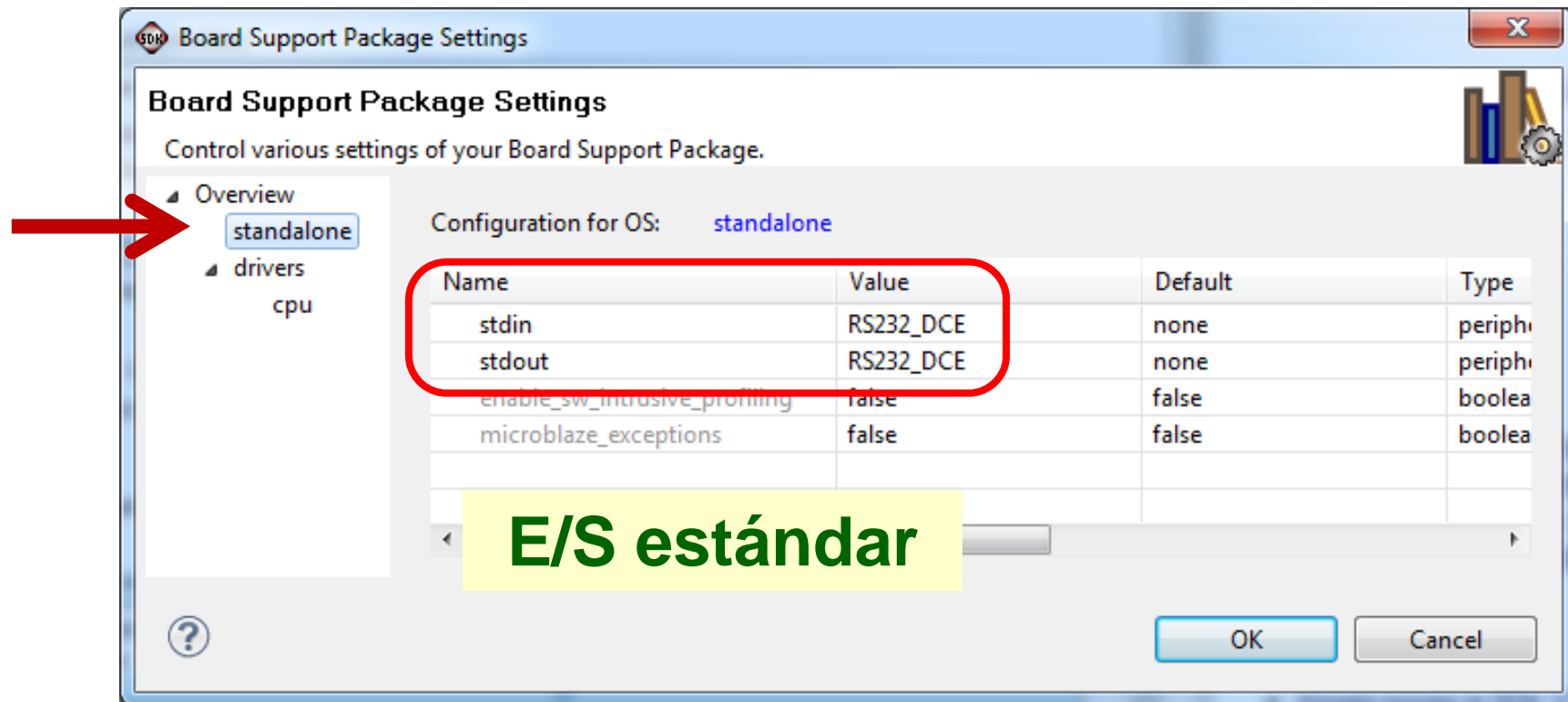
Name	Version	Description
<input type="checkbox"/> lwip130	3.00.a	lwIP TCP/IP Stack library: lwIP v1.3.0, Xilinx adapter v...
<input type="checkbox"/> xilfatfs	1.00.a	Provides read/write routines to access files stored on...
<input type="checkbox"/> xilflash	2.02.a	Xilinx Flash library for Intel/AMD CFI compliant paral...
<input type="checkbox"/> xilisf	2.01.a	Xilinx In-system and Serial Flash Library
<input type="checkbox"/> xilmfs	1.00.a	Xilinx Memory File System

Bibliotecas

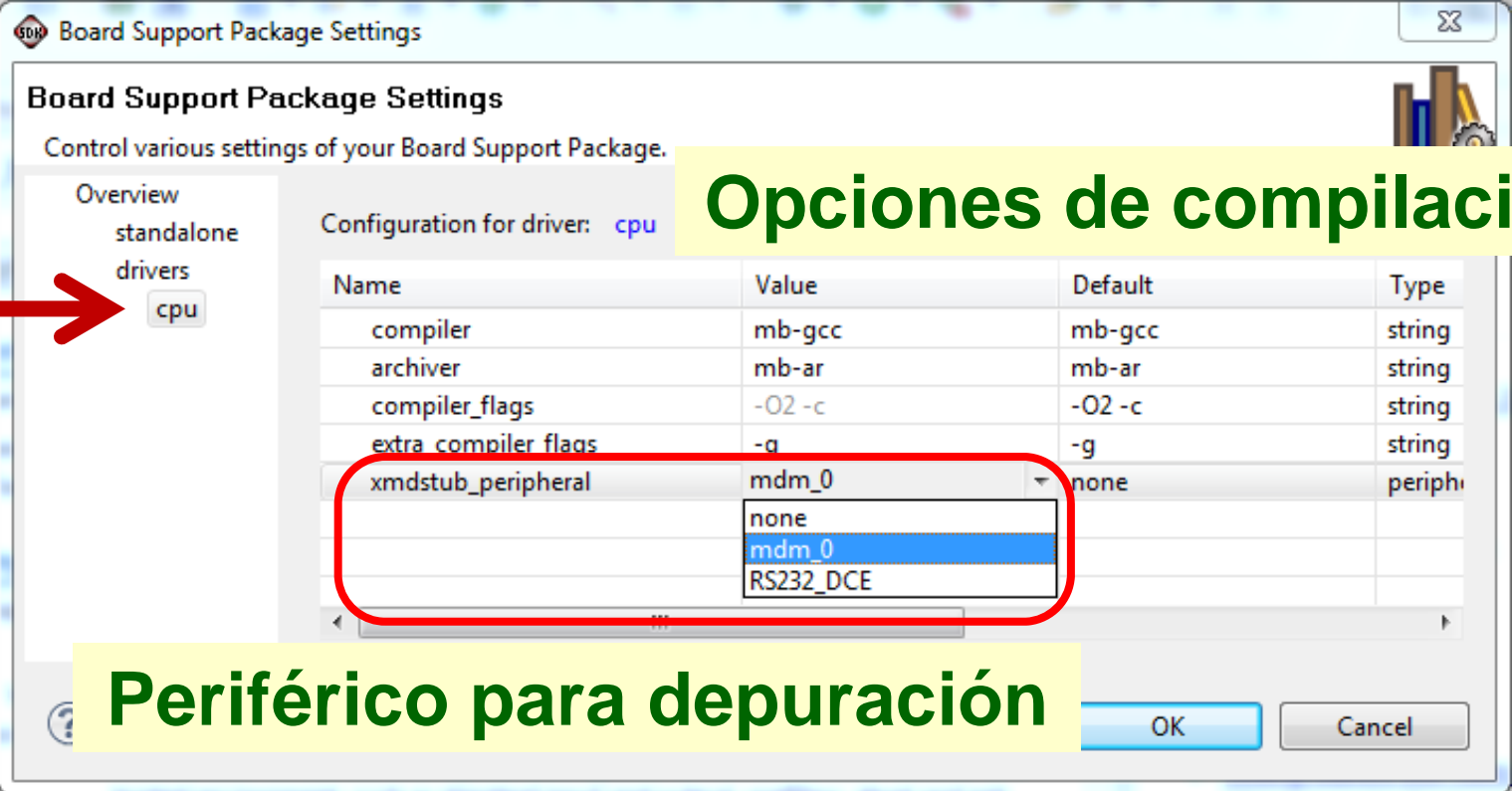
OK Cancel



3b. Configurar BSP (cont.)



3b. Configurar BSP (cont.)



Opciones de compilación

Name	Value	Default	Type
compiler	mb-gcc	mb-gcc	string
archiver	mb-ar	mb-ar	string
compiler_flags	-O2 -c	-O2 -c	string
extra_compiler_flags	-g	-g	string
xmdstub_peripheral	mdm_0	none	periph


Periférico para depuración

The dropdown menu for 'xmdstub_peripheral' shows the following options: none, mdm_0 (selected), and RS232_DCE.

Cada vez que finaliza la configuración SW (BSP), se ejecuta **LibGen** automáticamente



3c. Ejecución de LibGen



```
Problems Tasks Console X Properties Terminal 1
C-Build [LAB2_bsp_0]
make -k all
libgen -hw ../hw_platform_0/system.xml\
\
-pe microblaze_0 \
-log libgen.log \
system.mss

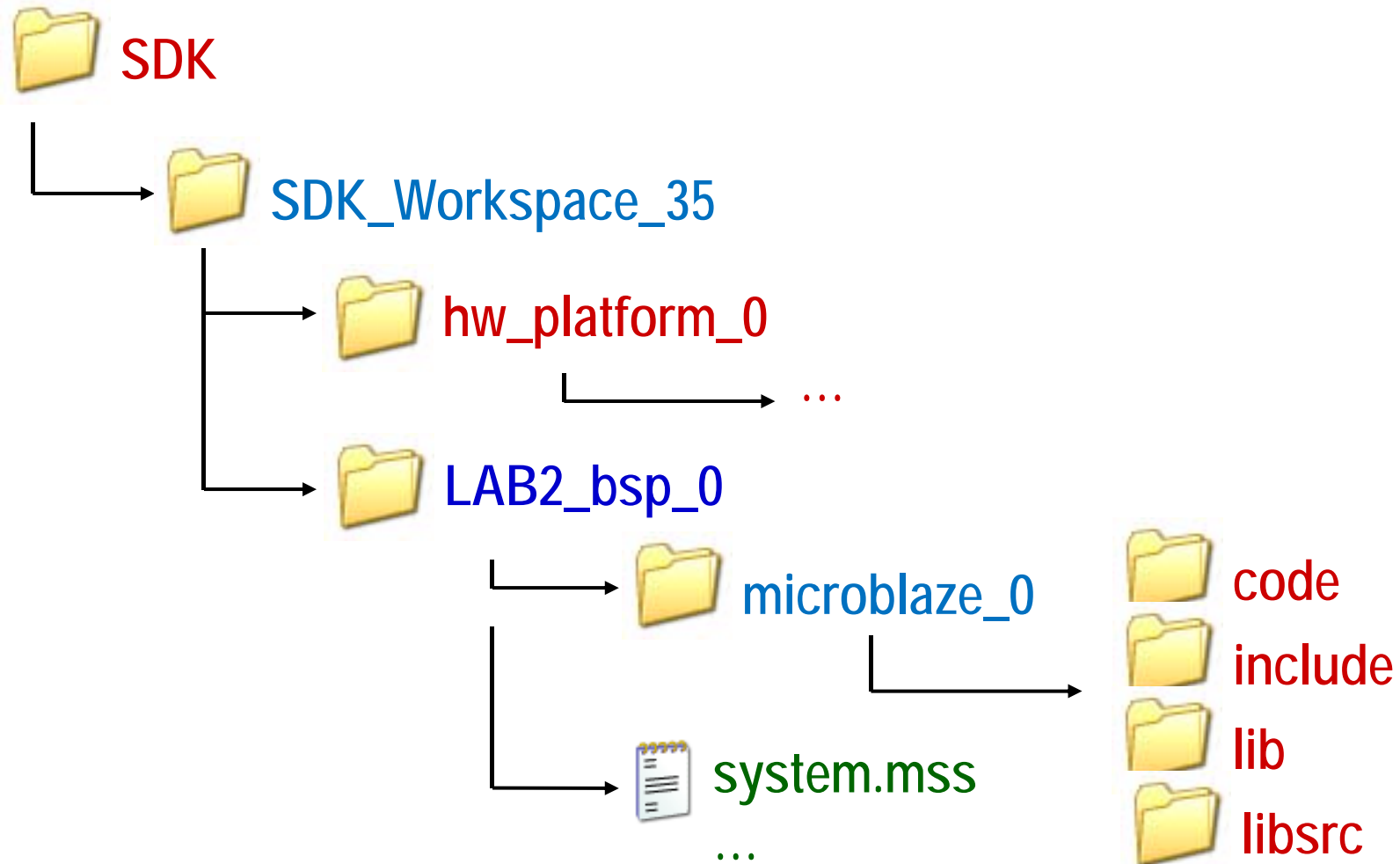
libgen
Xilinx EDK 12.4 Build EDK_MS4.81d
Copyright (c) 1995-2010 Xilinx, Inc. All rights reserved.

Command Line: libgen -hw ../hw_platform_0/system.xml -pe microblaze_0 -log
libgen.log system.mss

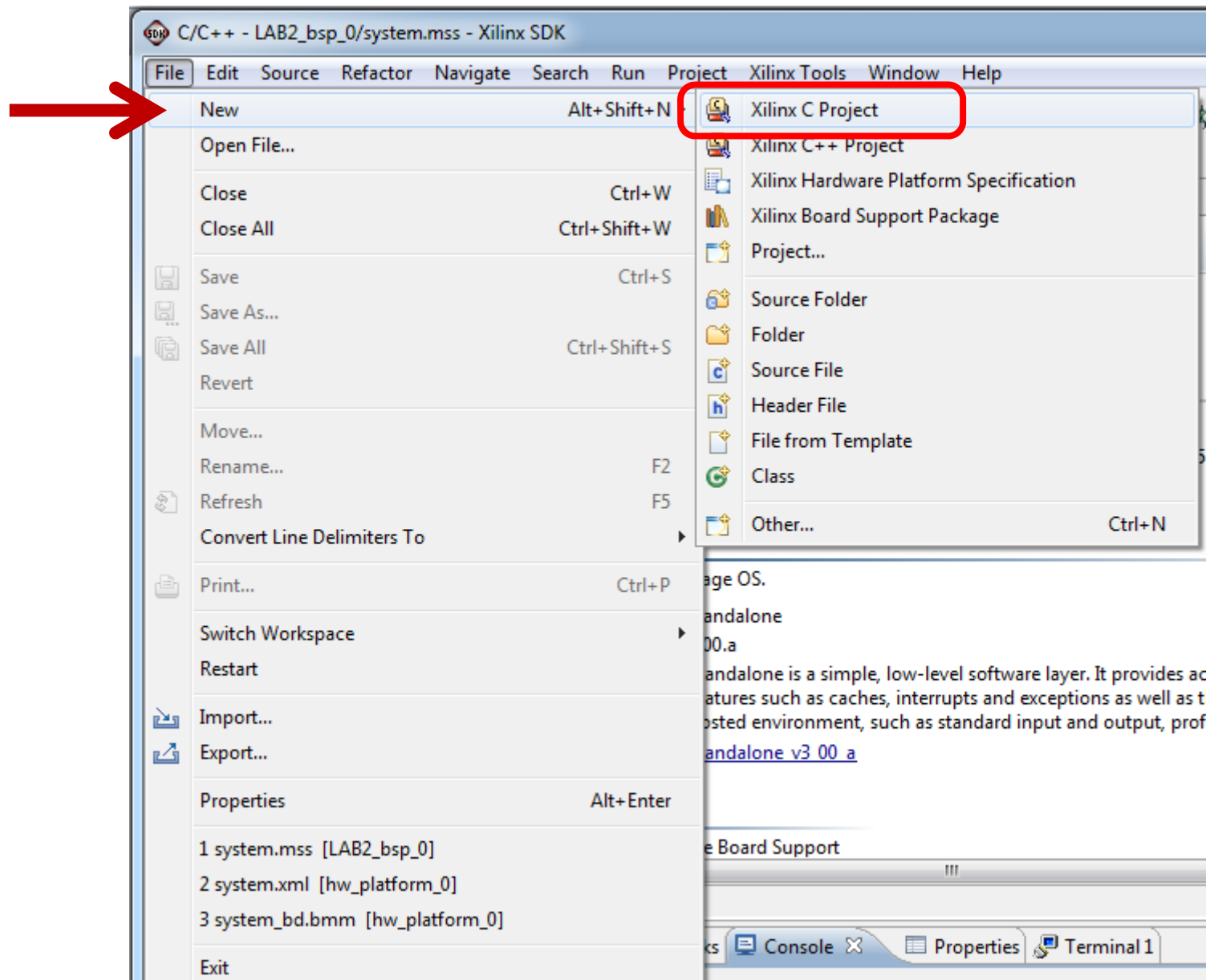
Staging source files.
Running DRCs.
Running generate.
Running post_generate.
Running include - 'make -s include "COMPILER=mb-gcc" "ARCHIVER=mb-ar"
"COMPILER_FLAGS=-mno-xl-soft-mul -mxl-barrel-shift -mxl-pattern-compare
-mcpu=v8.00.b -O2 -c" "EXTRA_COMPILER_FLAGS=-g"'.

Running libs - 'make -s libs "COMPILER=mb-gcc" "ARCHIVER=mb-ar"
"COMPILER_FLAGS=-mno-xl-soft-mul -mxl-barrel-shift -mxl-pattern-compare
-mcpu=v8.00.b -O2 -c" "EXTRA_COMPILER_FLAGS=-g"'.
Compiling common
Compiling lldma
Compiling standalone
Compiling gpio
Compiling mpmc
Compiling uartlite
Compiling cpu
Running execs_generate.
Finished building libraries
```

Estructura del BSP



4a. Crear proyecto SW



4a. Crear proyecto SW

New Project

New Xilinx C Project

Create a managed make application project. Choose from one of the sample applications.

Project name: LAB2_0

☒ Use default location

Location: D:\Proyectos_12.4\LAB2_S3A\SDK\SDK_Workspace_35\LAB2_0 [Browse...]

Hardware Platform: hw_platform_0

Processor: microblaze_0

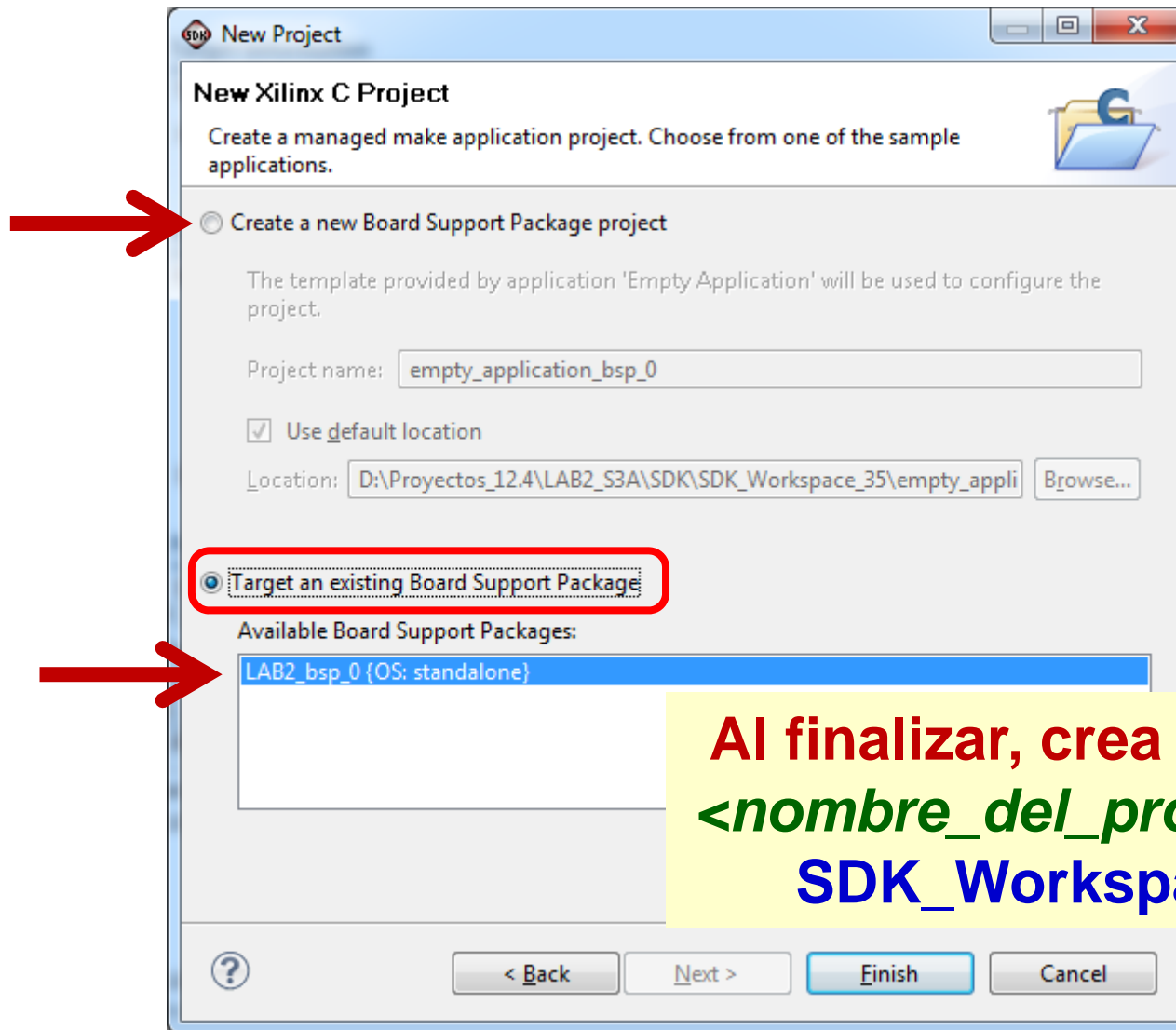
	Description
Dhrystone	
Empty Application	A blank C project.
Hello World	
lwIP Echo Server	
Memory Tests	
Peripheral Tests	
SREC Bootloader	
Xilkernel POSIX Threads Demo	

Puede seleccionarse entre aplicaciones de ejemplo

< Back Next > Finish Cancel



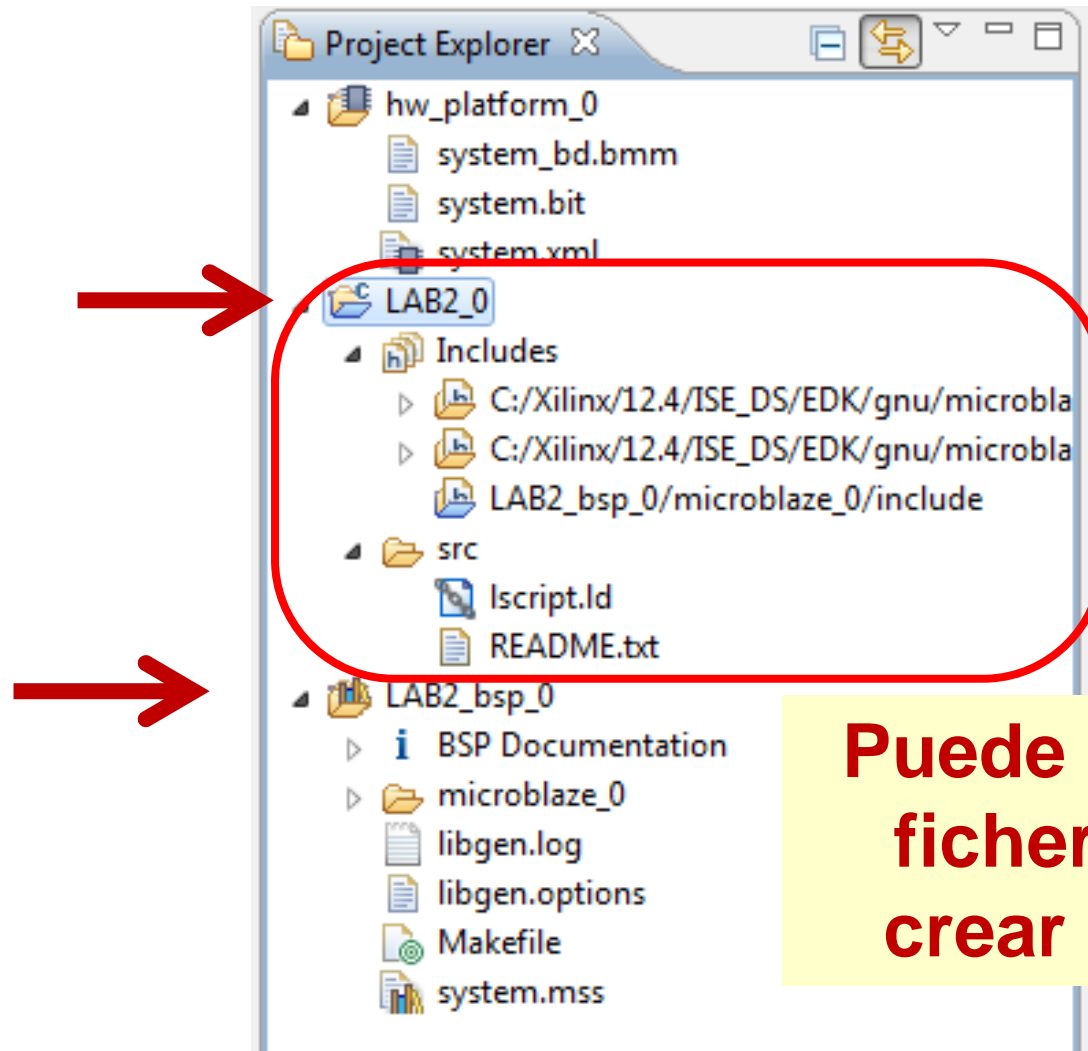
4b. Asociar con BSP



Al finalizar, crea la carpeta
<nombre_del_proyecto> en
SDK_Workspace_35



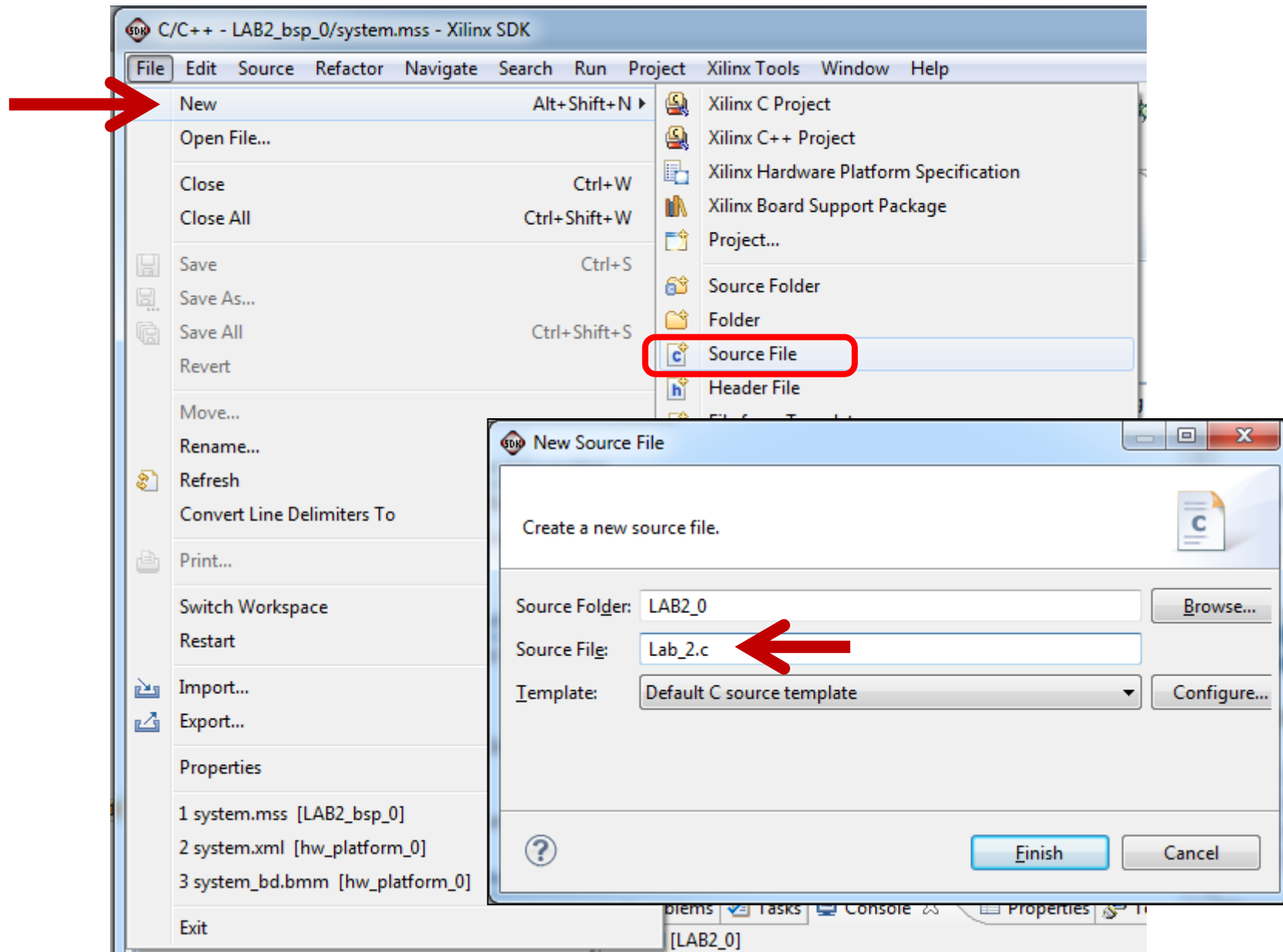
Proyecto SW



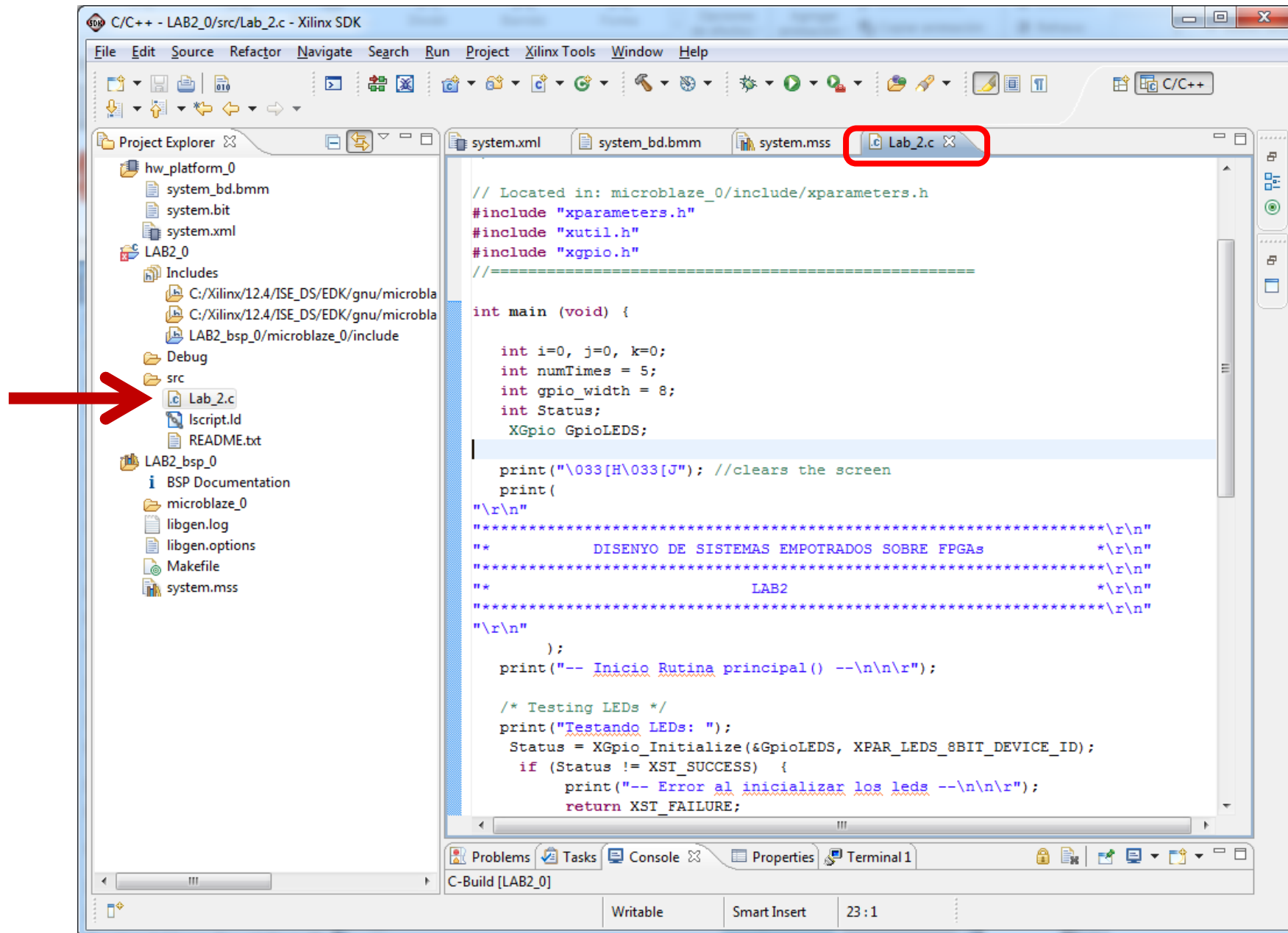
Puede añadirse un fichero fuente o crear uno nuevo



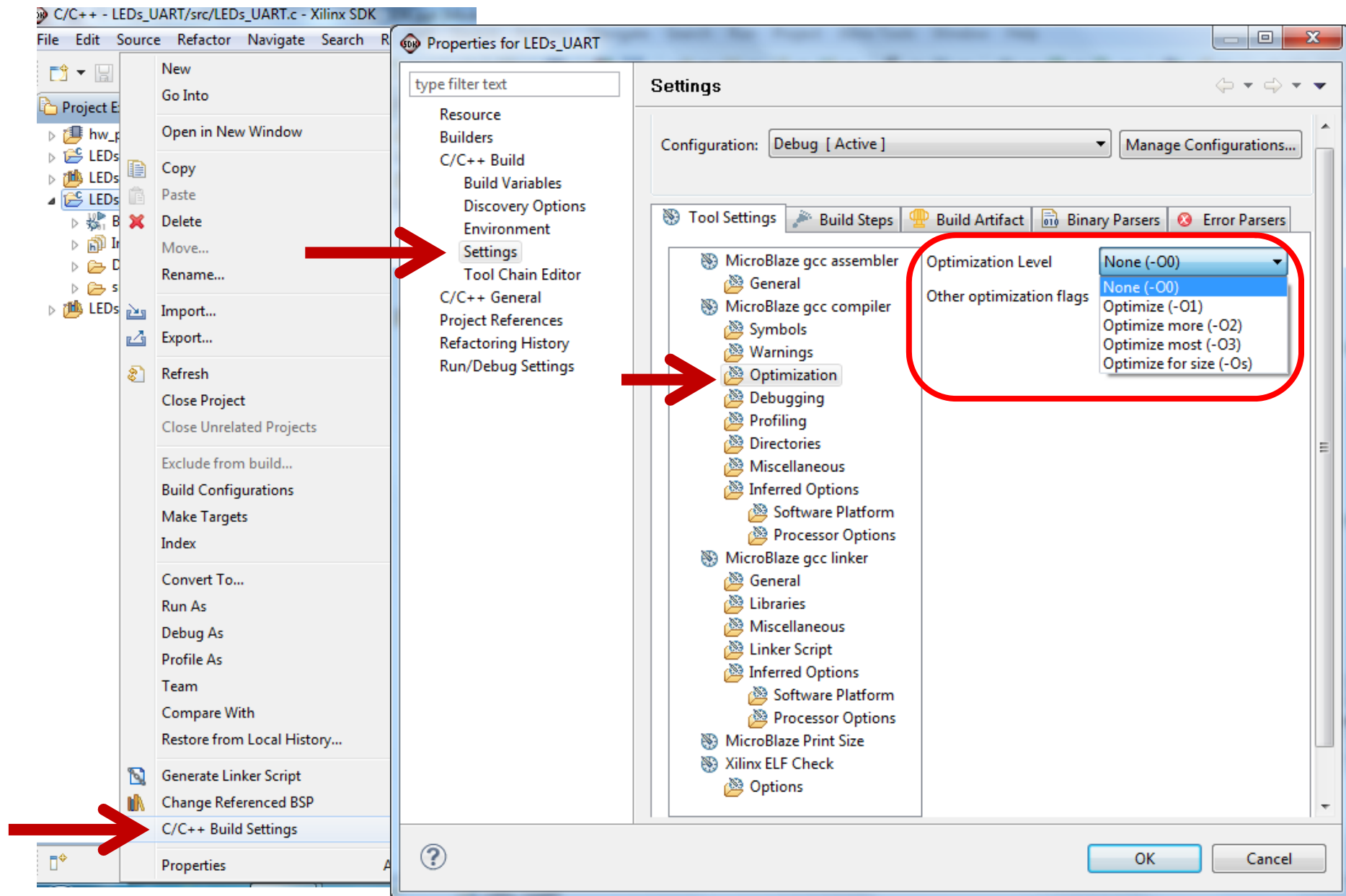
4c. Crear código fuente



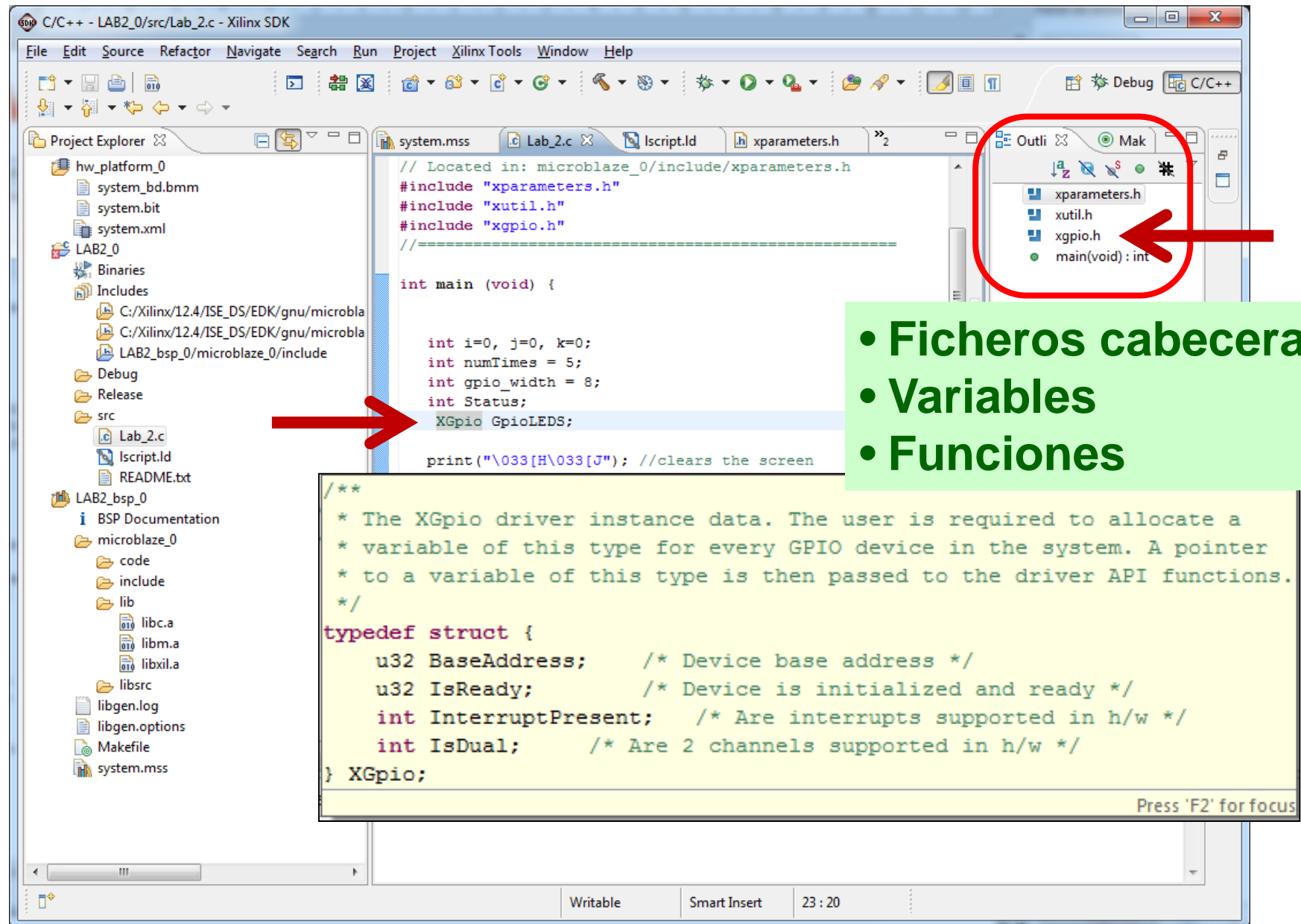
4c. Crear código fuente (cont.)



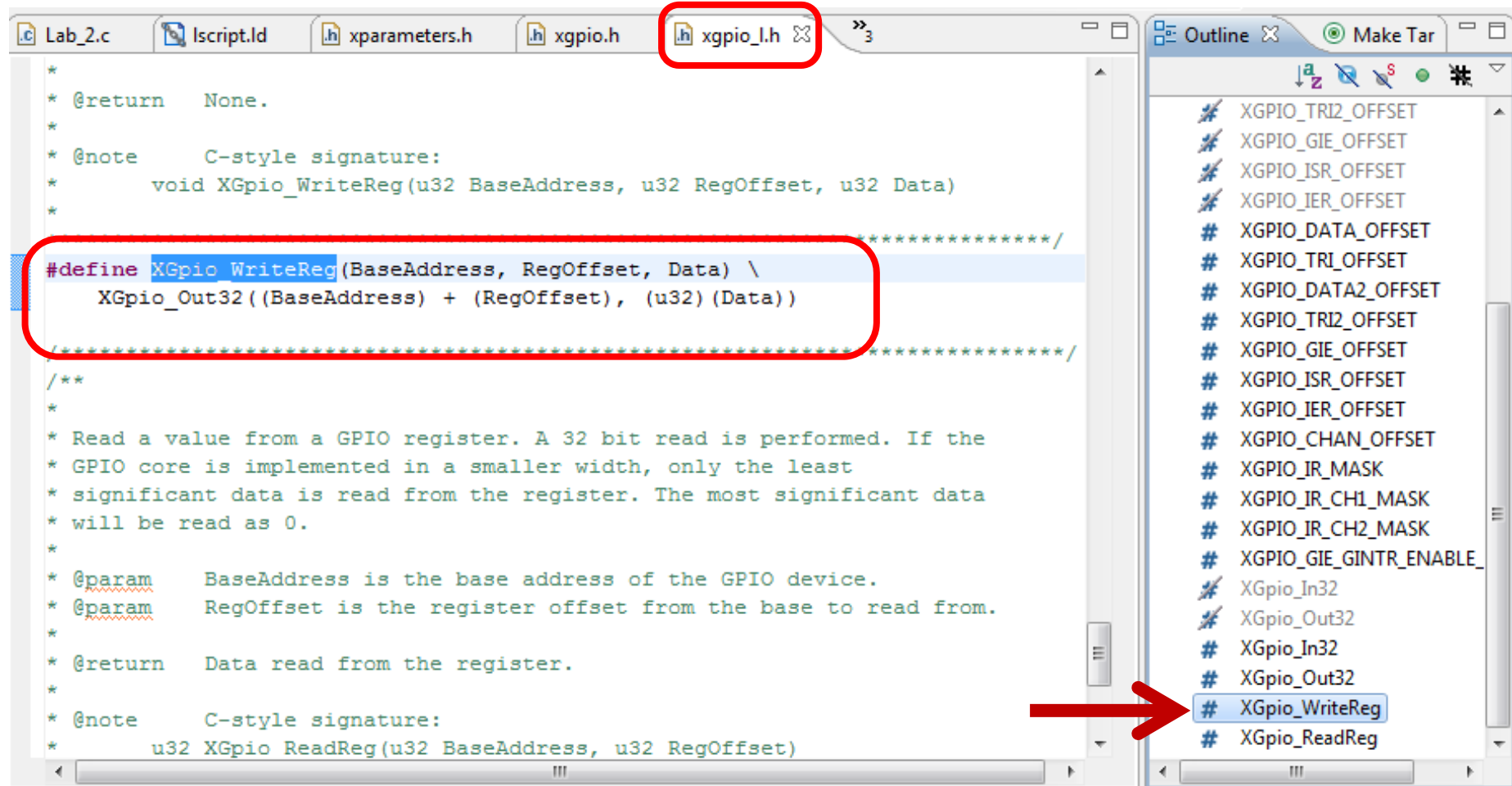
4d. Optimización de la aplicación



Navegador de código Eclipse



Navegador de código Eclipse



5. [Configurar LinkerScript]

Archivo *.ld a modificar

Output Script:
D:\Proyectos_12.4\LAB2_S3A\SDK\SDK_Workspace_35\LAB2_0\src Browse

Permite establecer el mapeo en memoria de las diferentes partes del programa

Memory	Base Address	Size
ilmb_cntlr_dlmb_cntlr	0x00000000	16 KB
DDR2_SDRAM_MPMC_BASEADDR	0x44000000	64 MB

Code Section Assignments

Section	Assigned Memory
.text	ilmb_cntlr_dlmb_cntlr

Add Section Remove Section

Data Section Assignments

Section	Assigned Memory
.rodata	ilmb_cntlr_dlmb_cntlr
.sdata2	ilmb_cntlr_dlmb_cntlr
.sbss2	ilmb_cntlr_dlmb_cntlr
.data	ilmb_cntlr_dlmb_cntlr
.sdata	ilmb_cntlr_dlmb_cntlr
.sbss	ilmb_cntlr_dlmb_cntlr
.bss	ilmb_cntlr_dlmb_cntlr

Fixed Section Assignments

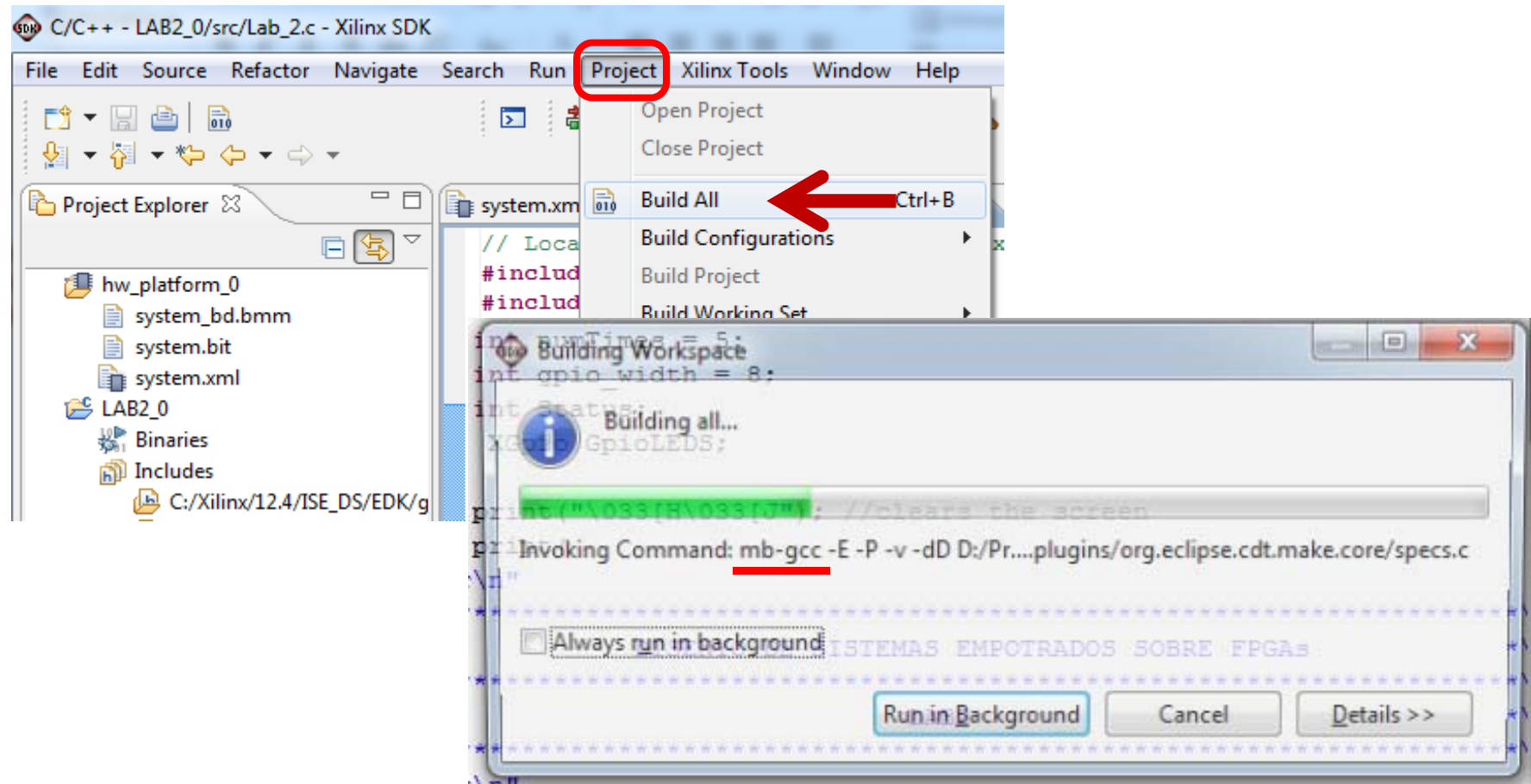
Section	Assigned Memory	Address
.vectors.reset	ilmb_cntlr_dlmb_cntlr	0x00000000
.vectors.sw_exc...	ilmb_cntlr_dlmb_cntlr	0x00000008
.vectors.interrupt	ilmb_cntlr_dlmb_cntlr	0x00000010
.vectors.hw_ex...	ilmb_cntlr_dlmb_cntlr	0x00000020

Heap and Stack Section Assignments

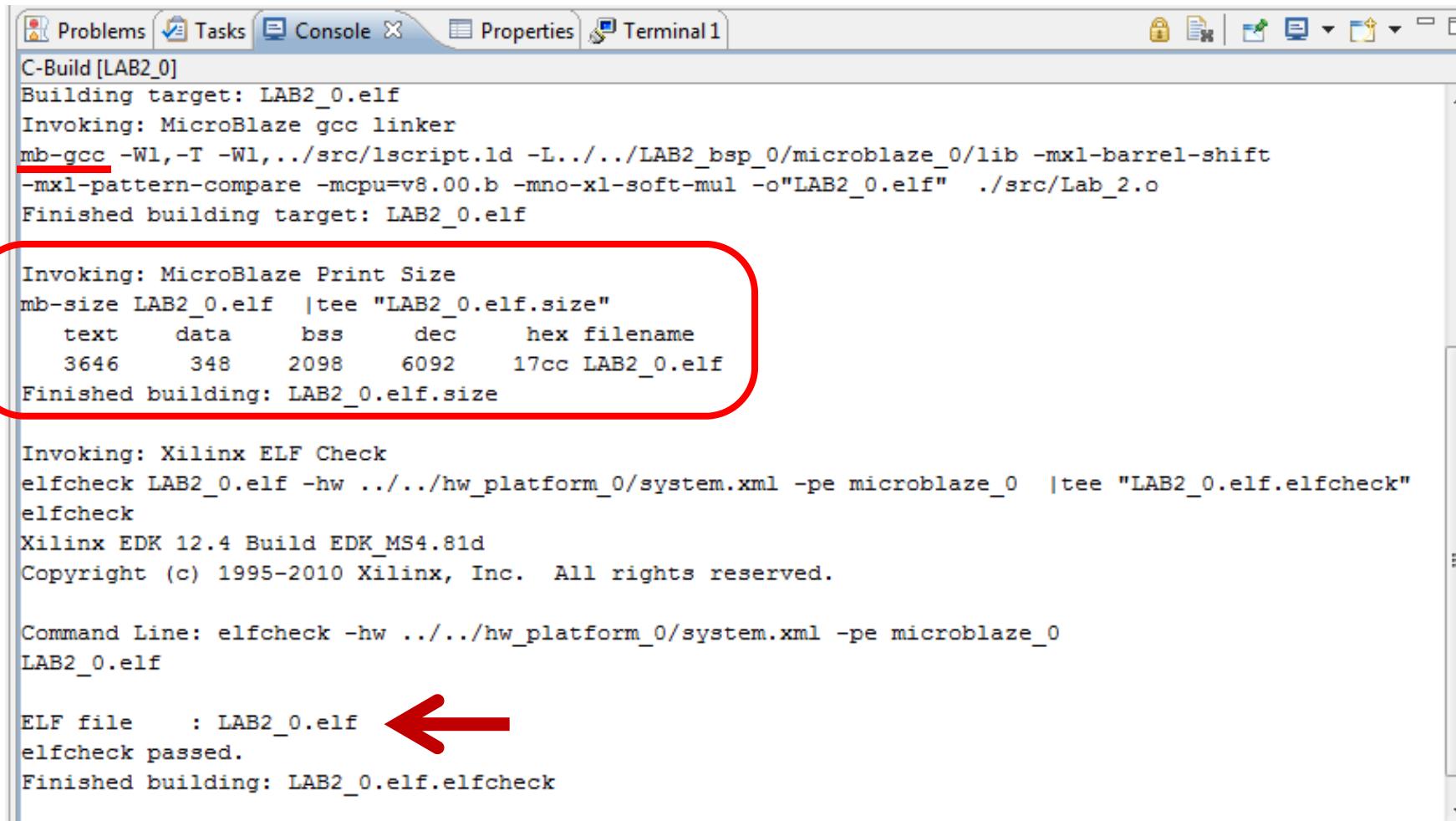
Section	Assigned Memory	Assigned Size
Heap	ilmb_cntlr_dlmb_cntlr	1 KB
Stack	ilmb_cntlr_dlmb_cntlr	1 KB

Generate Cancel

6a. Implementar aplicación



6b. Reporte de compilación



```
C-Build [LAB2_0]
Building target: LAB2_0.elf
Invoking: MicroBlaze gcc linker
mb-gcc -Wl,-T -Wl,../src/lscript.ld -L../LAB2_bsp_0/microblaze_0/lib -mxl-barrel-shift
-mxl-pattern-compare -mcpu=v8.00.b -mno-xl-soft-mul -o"LAB2_0.elf" ./src/Lab_2.o
Finished building target: LAB2_0.elf

Invoking: MicroBlaze Print Size
mb-size LAB2_0.elf |tee "LAB2_0.elf.size"
  text    data    bss     dec     hex filename
  3646     348    2098    6092    17cc LAB2_0.elf
Finished building: LAB2_0.elf.size

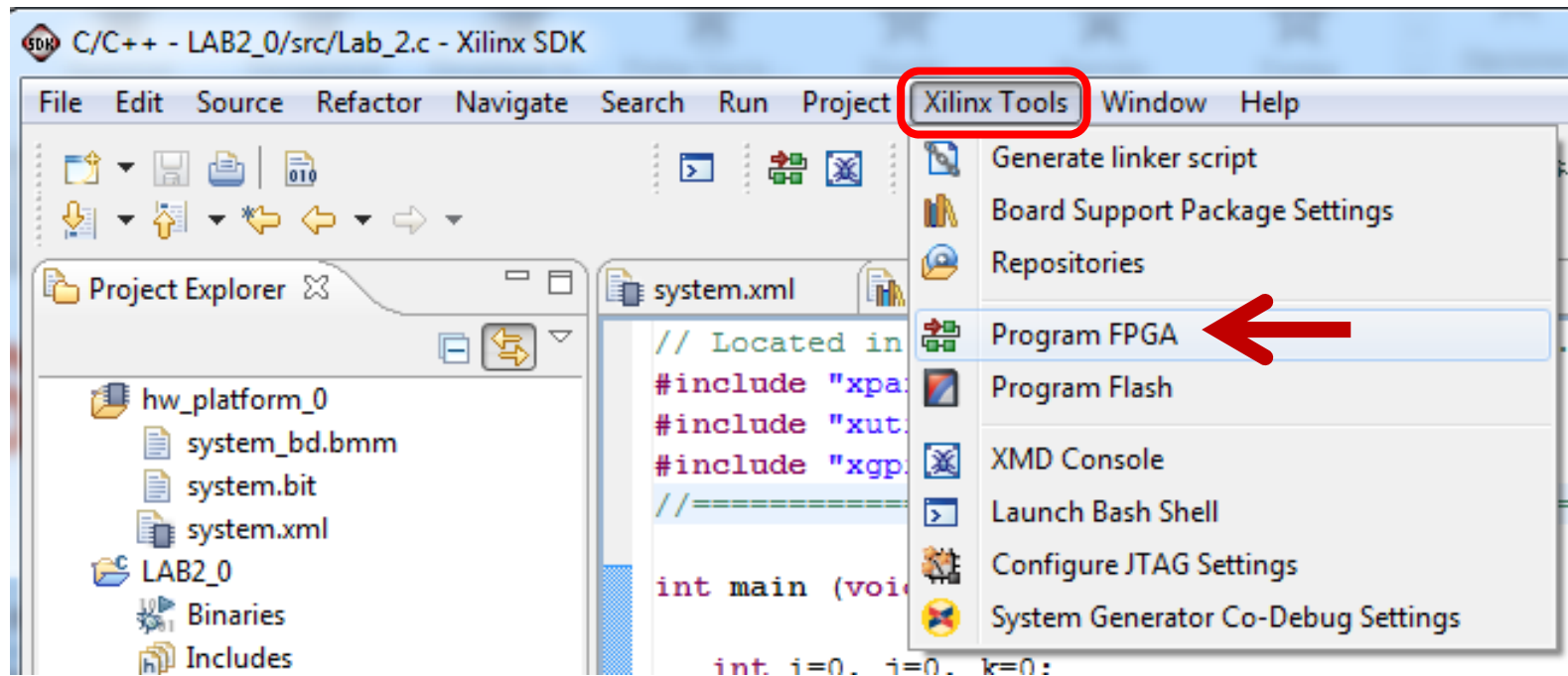
Invoking: Xilinx ELF Check
elfcheck LAB2_0.elf -hw ../hw_platform_0/system.xml -pe microblaze_0 |tee "LAB2_0.elf.elfcheck"
elfcheck
Xilinx EDK 12.4 Build EDK_MS4.81d
Copyright (c) 1995-2010 Xilinx, Inc. All rights reserved.

Command Line: elfcheck -hw ../hw_platform_0/system.xml -pe microblaze_0
LAB2_0.elf

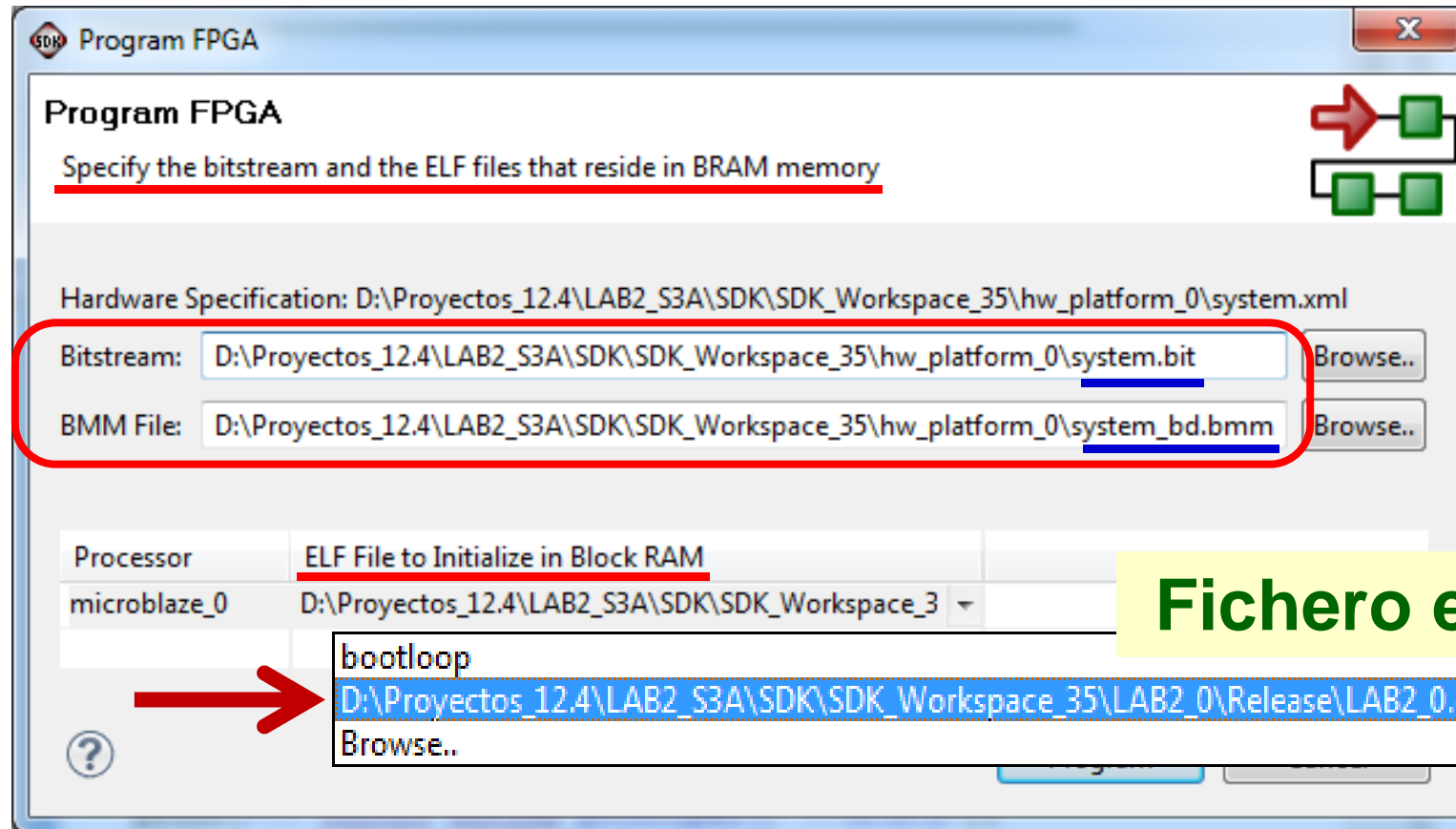
ELF file      : LAB2_0.elf
elfcheck passed.
Finished building: LAB2_0.elf.elfcheck
```



7a. Programar FPGA



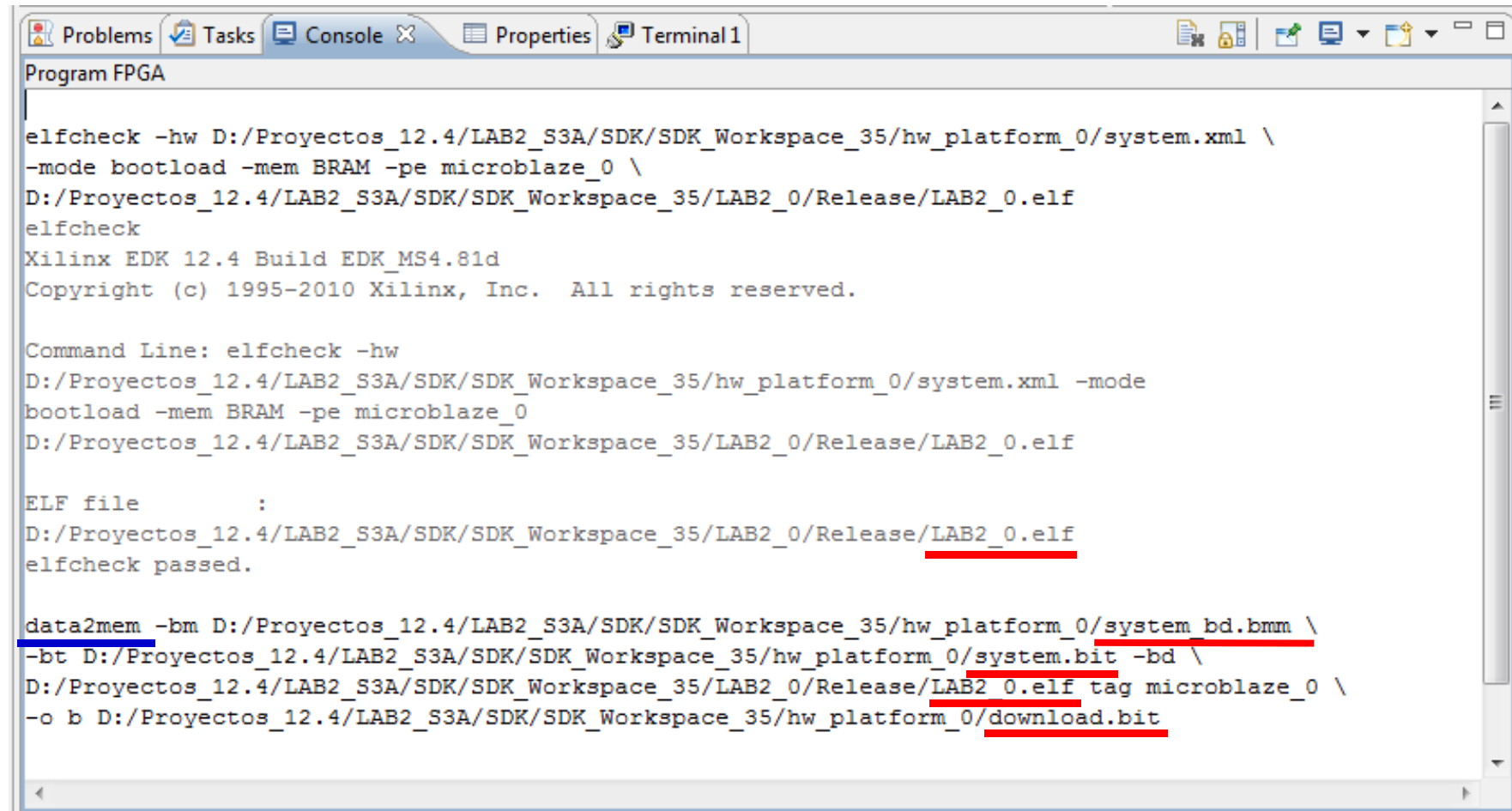
7b. Generar download.bit



Se inicializa con **bootlop** cuando la aplicación residirá en memoria externa (o se descargará con XMD)



7b. Generar download.bit (cont.)



The screenshot shows a terminal window titled "Program FPGA" with tabs for Problems, Tasks, Console, Properties, and Terminal1. The terminal displays the output of the 'elfcheck' command, which verifies the ELF file 'LAB2_0.elf'. The output indicates that the ELF file is valid and that the 'elfcheck' command passed. Below this, the 'data2mem' command is shown, which generates the 'download.bit' file from the ELF file and a BMM file. The command line is: `data2mem -bm D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system bd.bmm \ -bt D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system.bit -bd \ D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf tag microblaze_0 \ -o b D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/download.bit`. The output of the command is not visible in the screenshot.

```
Program FPGA

elfcheck -hw D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system.xml \
-mode bootload -mem BRAM -pe microblaze_0 \
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf
elfcheck
Xilinx EDK 12.4 Build EDK_MS4.81d
Copyright (c) 1995-2010 Xilinx, Inc. All rights reserved.

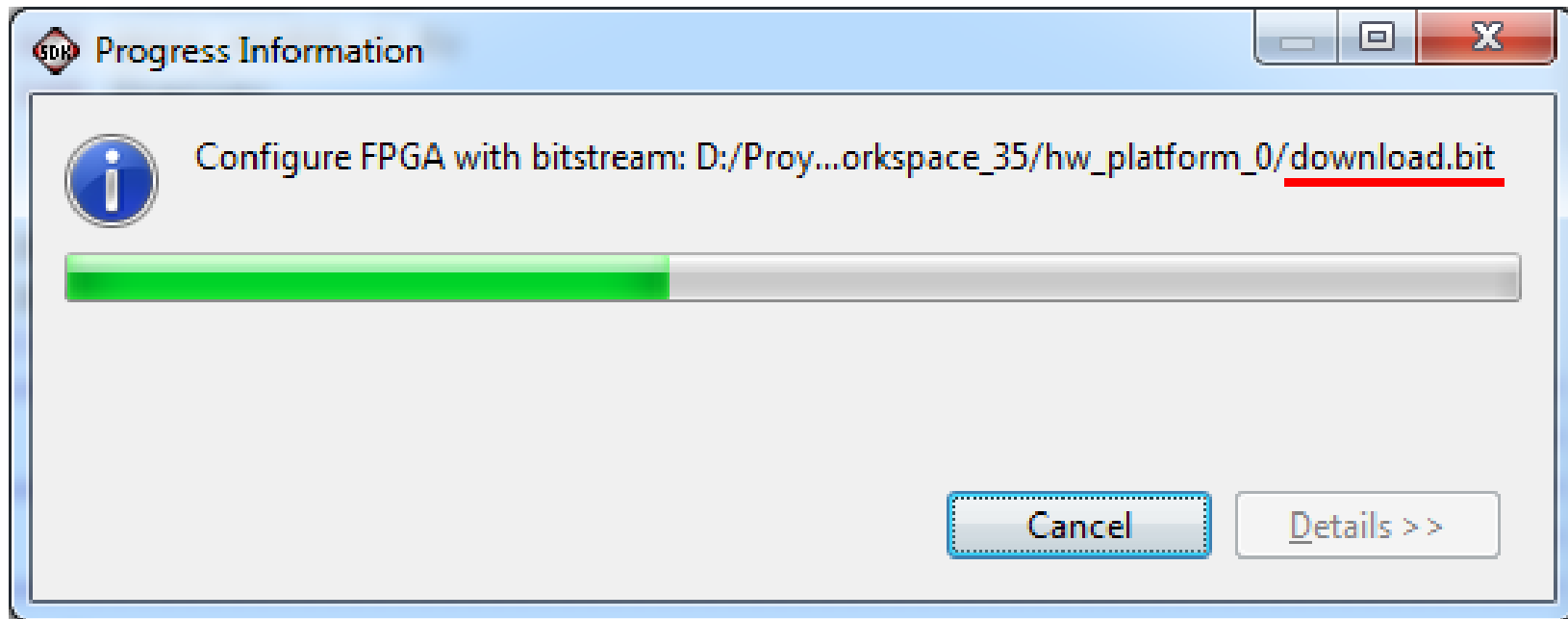
Command Line: elfcheck -hw
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system.xml -mode
bootload -mem BRAM -pe microblaze_0
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf

ELF file      :
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf
elfcheck passed.

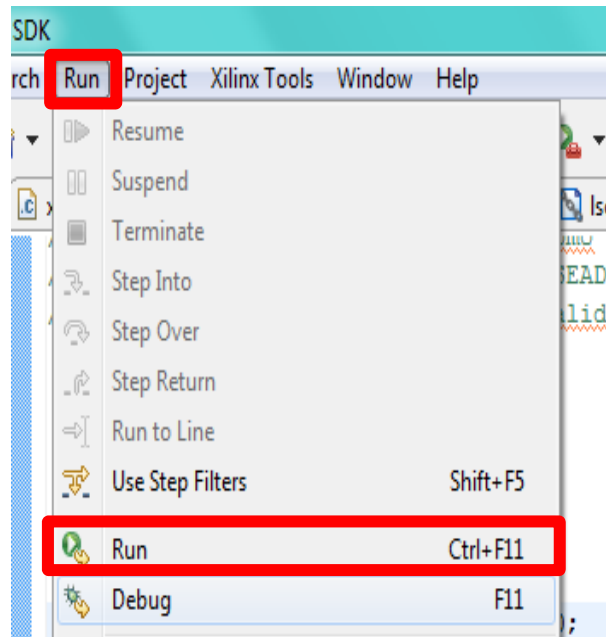
data2mem -bm D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system bd.bmm \
-bt D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system.bit -bd \
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf tag microblaze_0 \
-o b D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/download.bit
```



7c. Descargar bitstream al FPGA

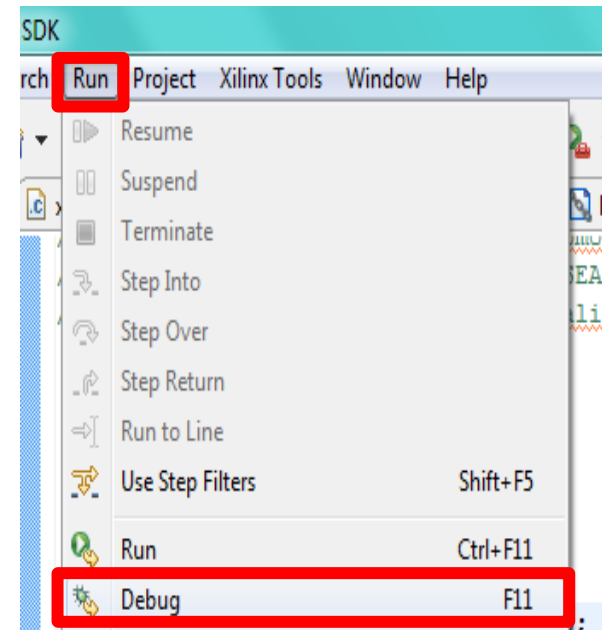


8a. Depuración sobre el FPGA



Run

Permite correr la aplicación y detenerla desde SDK

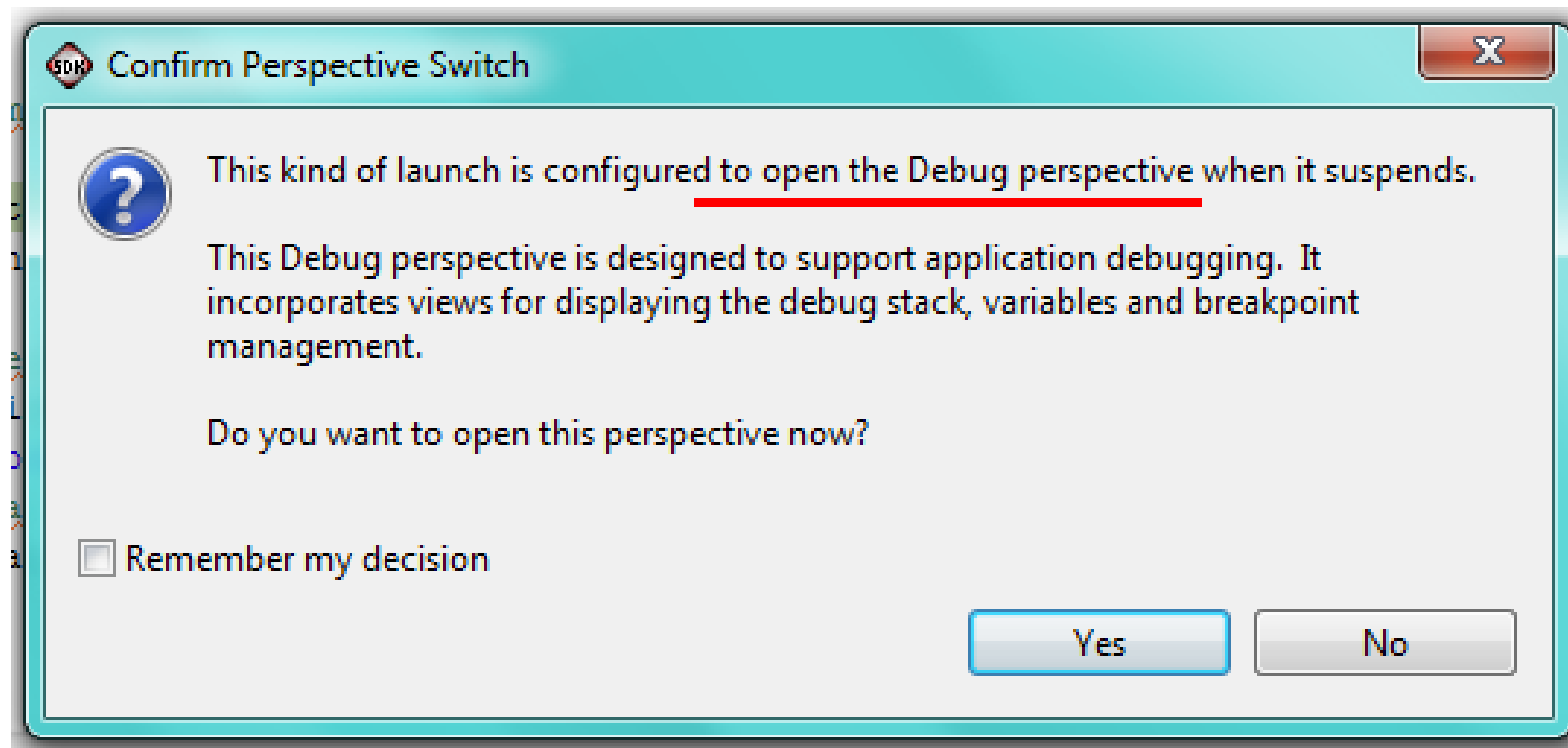


Debug

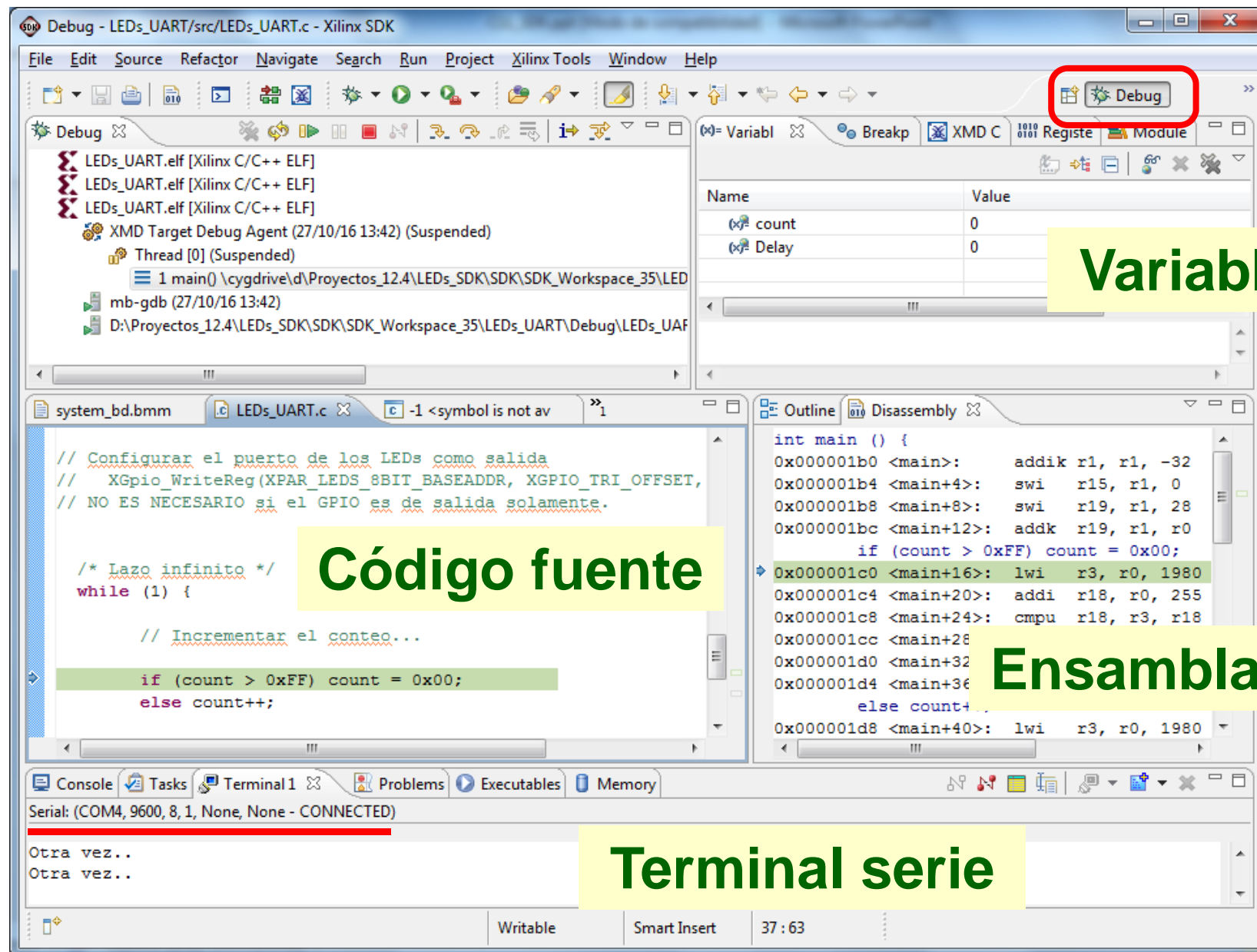
Permite depurar la aplicación desde SDK, el cual traduce cualquier acción de la interfaz de usuario a comandos GDB, y procesa las respuestas del GDB para mostrarlos en pantalla



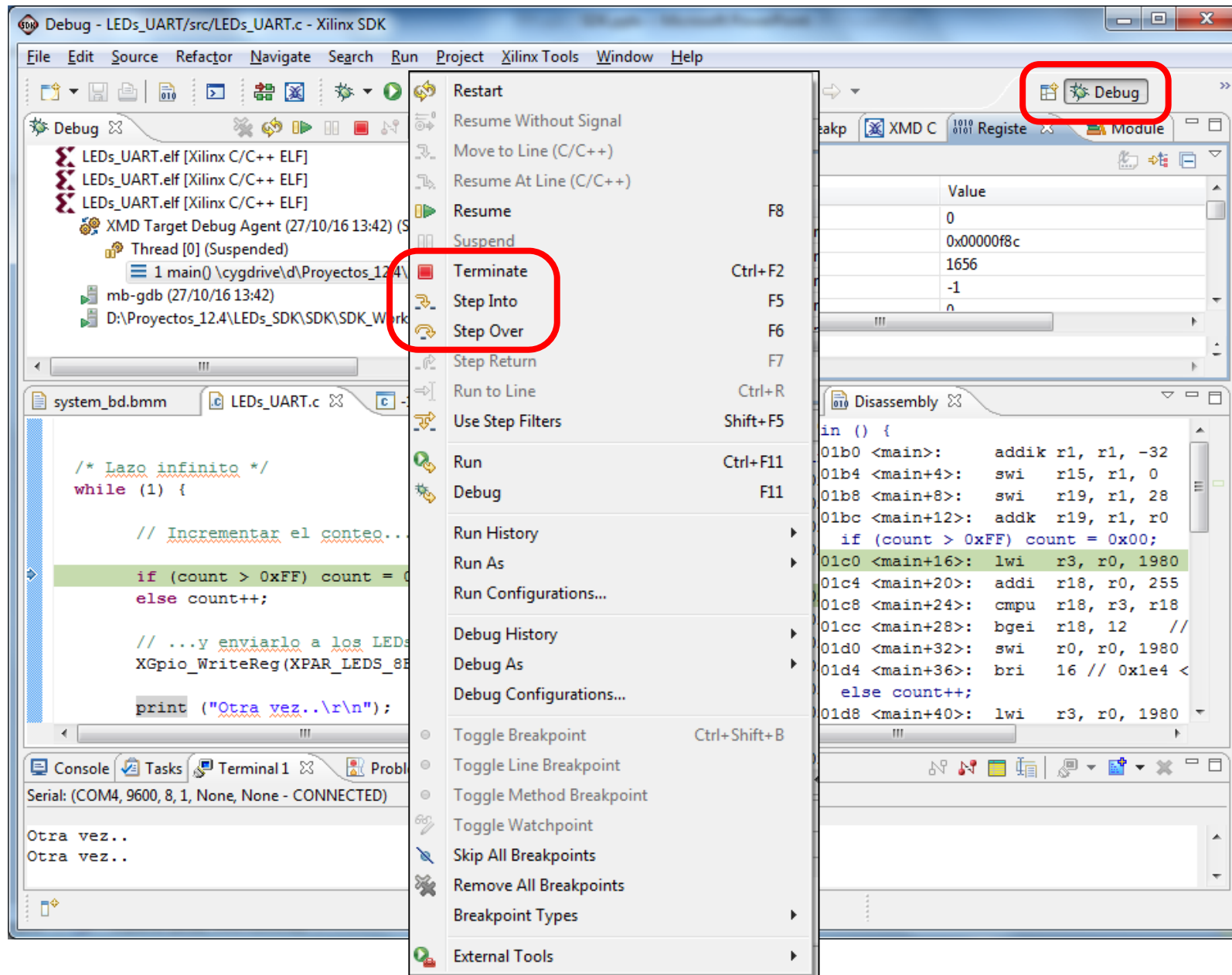
8a. Depuración sobre el FPGA



8b. Perspectiva de depuración



8c. Comandos de depuración



Documentación

- Manuales

- *Getting started with Xilinx SDK*
- *SDK Cheat Sheet Tutorials*

- Soporte Web

- SDK
 - <http://www.support.xilinx.com/sdk>

