

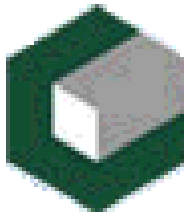
Desarrollo de aplicaciones

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

- Desarrollo de aplicaciones en sistemas empotrados
- Herramientas de desarrollo GNU
 - **Compilador y Linker**
 - **Utilidades**
- Gestión de proyectos de software en **XPS**
- Codificación de aplicaciones
 - **Drivers de dispositivos**
 - **Bibliotecas**
 - **Sistema Operativo**
- Carga y ejecución de programas



Desarrollo de Aplicaciones para PC

- Las herramientas de desarrollo y la aplicación *se ejecutan en el mismo procesador* (código nativo)
- El *sistema operativo* proporciona mecanismos para cargar el programa en memoria (*loader*), compartir la CPU con otros procesos (*scheduler*) y acceder a los dispositivos de E/S (*drivers*)



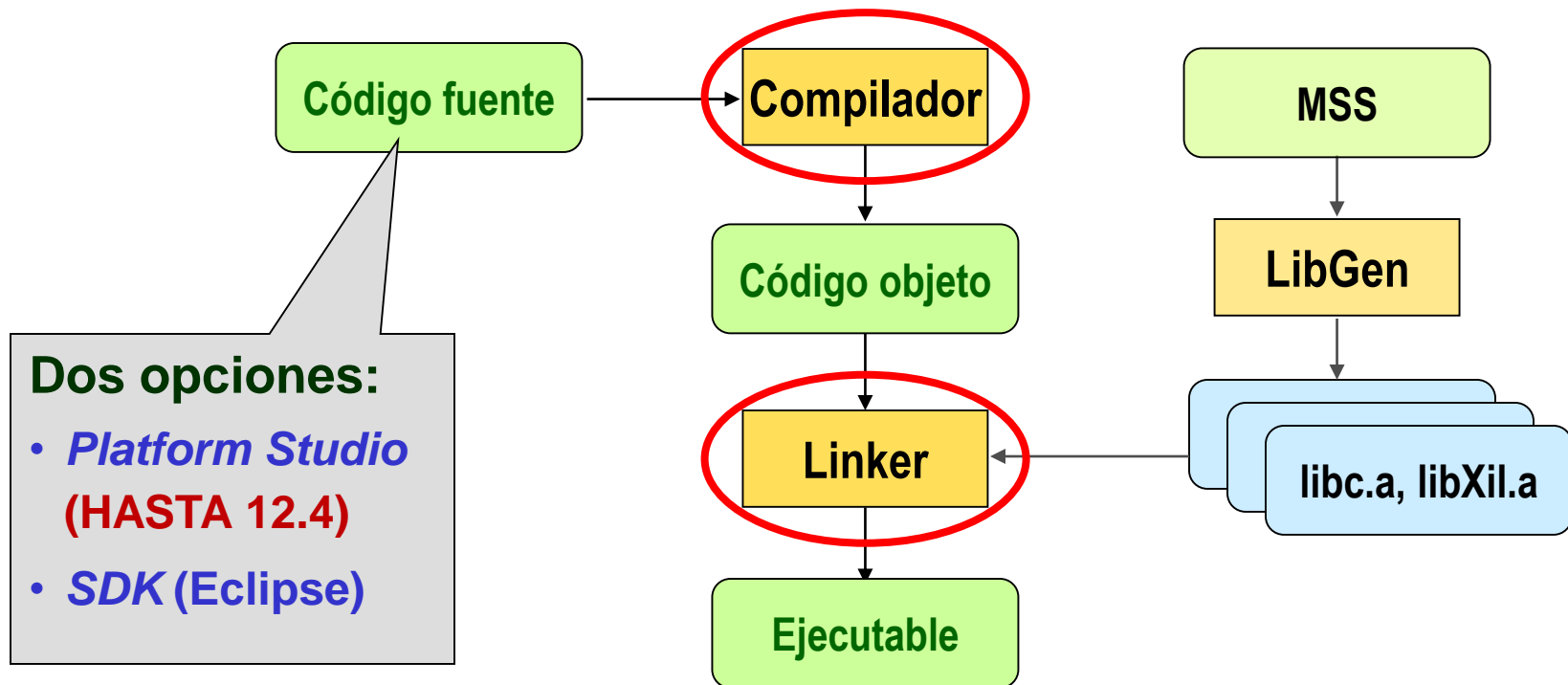
Desarrollo de Aplicaciones para un Sistema Empotrado

- Las herramientas de desarrollo *se ejecutan en un procesador diferente* al procesador de destino de la aplicación (**compilación cruzada**).
- El programa ejecutable que será *descargado* en el sistema empuetrado debe contener el *código de la aplicación*, junto con **código de arranque** y las rutinas de **sistema operativo**, funciones de **bibliotecas** y **drivers** de dispositivos utilizados por la aplicación.

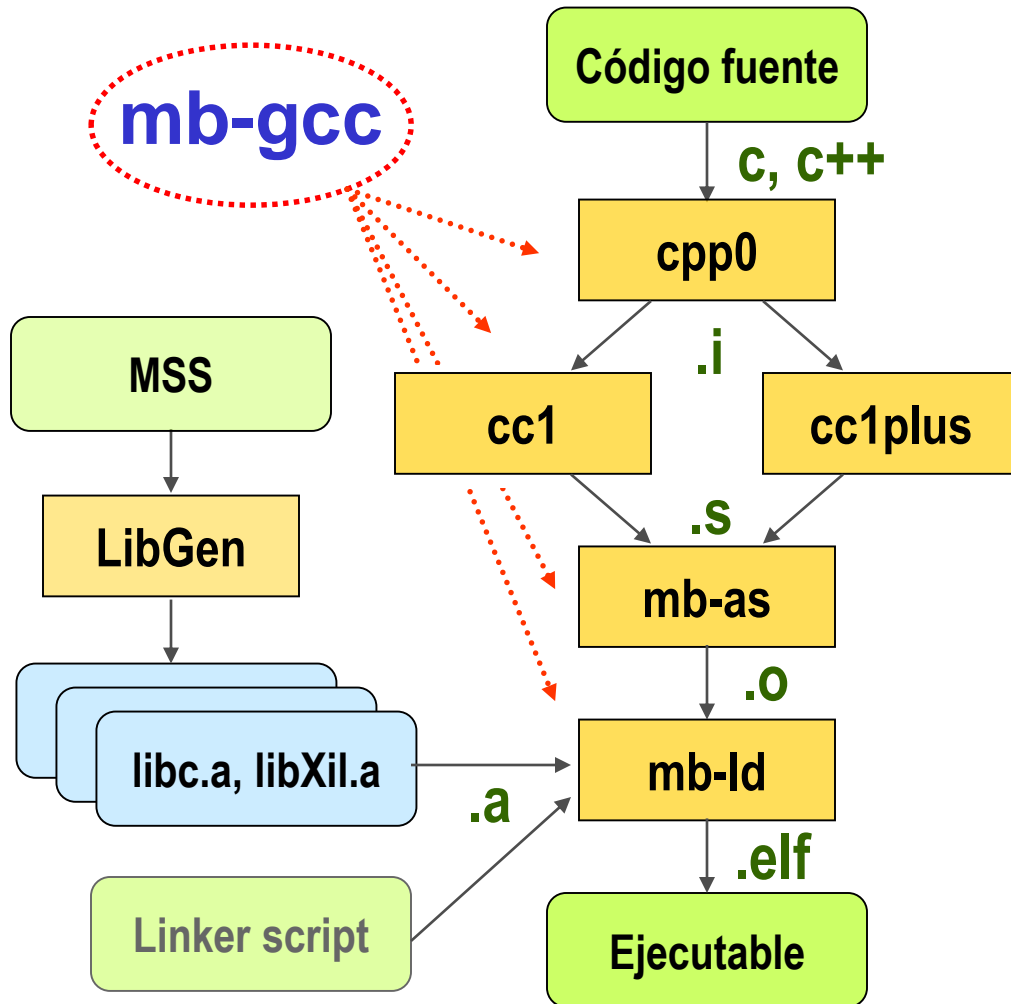


Desarrollo de aplicaciones en EDK

- **Entrada:** código fuente en C, C++ o ensamblador
- **Herramientas:** Entorno de desarrollo GNU



Herramientas de desarrollo GNU



Preprocesador

Expande ficheros fuente incluyendo ficheros de cabecera

Compilador

Verifica sintaxis
Genera código ensamblador

Ensamblador (**mb-as**)

Convierte a lenguaje máquina (código objeto)

Linker (**mb-ld**)

Combina módulos binarios.
Determina el mapa de memoria del ejecutable



Herramientas de desarrollo GNU *(cont.)*

UTILIDADES

❑ **mb-ar** (*ar*)

- Crea, modifica y extrae módulos de bibliotecas

❑ **mb-objdump** (*objdump*)

- Muestra información de ficheros objetos y ejecutables

❑ **mb-size** (*size*)

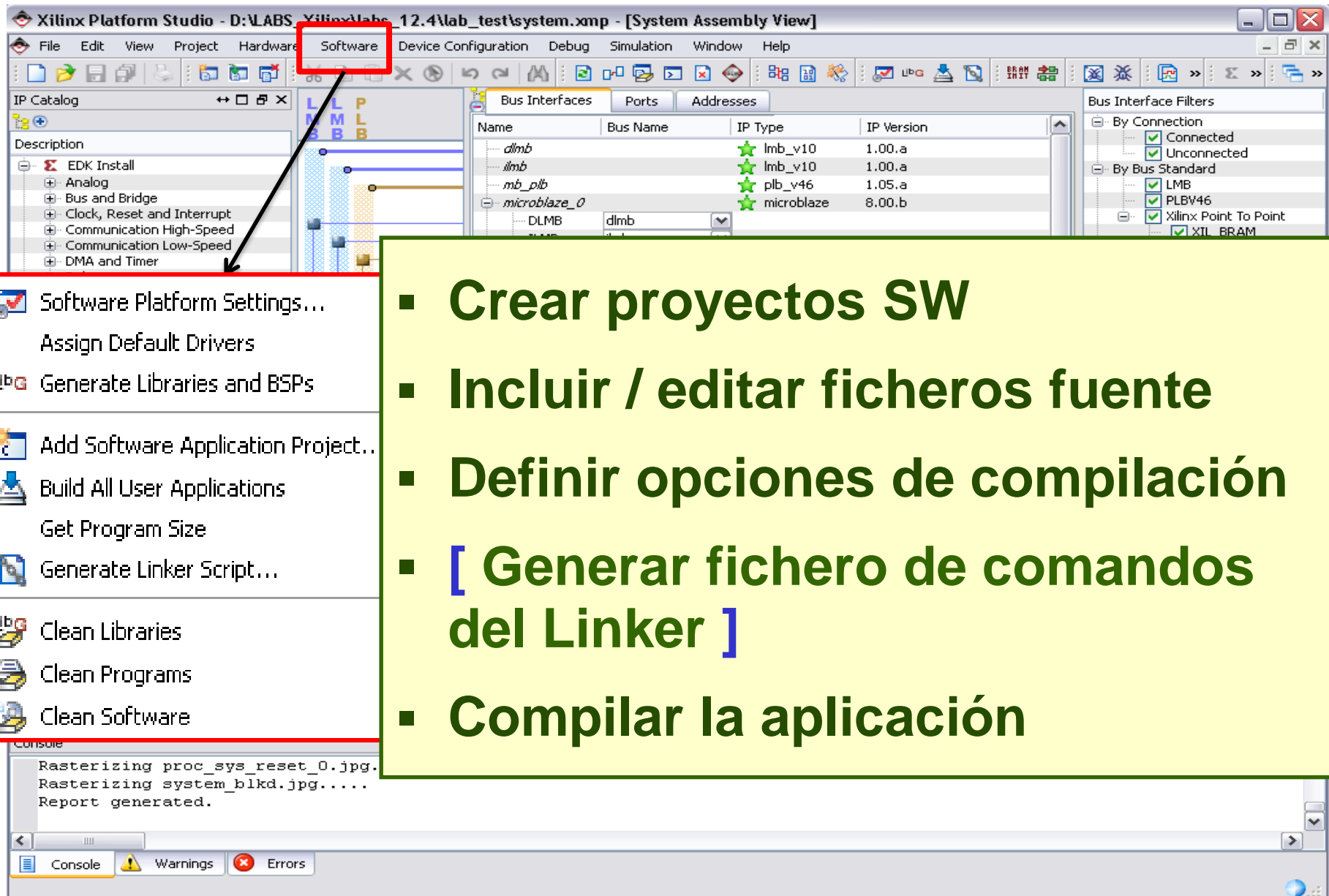
- Lista el tamaño de las secciones de un fichero objeto

❑ **mb-strings** (*strings*)

- Imprime las cadenas de caracteres de un fichero



Gestión de Proyectos de Software en XPS



The screenshot shows the Xilinx Platform Studio (XPS) interface. The 'Software' menu is highlighted with a red box, and an arrow points to it from the 'Software' tab in the top menu bar. The 'Software' menu is open, displaying a list of tasks. A red box highlights the first four items: 'Software Platform Settings...', 'Assign Default Drivers', 'Generate Libraries and BSPs', and 'Add Software Application Project...'. The 'Console' window at the bottom shows the output of a rasterization process.

Software Menu Items:

- Software Platform Settings...
- Assign Default Drivers
- Generate Libraries and BSPs
- Add Software Application Project...
- Build All User Applications
- Get Program Size
- Generate Linker Script...
- Clean Libraries
- Clean Programs
- Clean Software

Console Output:

```
Rasterizing proc_sys_reset_0.jpg.  
Rasterizing system_blkd.jpg.....  
Report generated.
```

Bus Interfaces Table:

Name	Bus Name	IP Type	IP Version
dmb		lmb_v10	1.00.a
ilmb		lmb_v10	1.00.a
mb_plb		plb_v46	1.05.a
microblaze_0		microblaze	8.00.b
DLMB	dmb		

Bus Interface Filters:

- By Connection
 - ☒ Connected
 - ☒ Unconnected
- By Bus Standard
 - ☒ LMB
 - ☒ PLBV46
 - ☒ Xilinx Point To Point
 - ☒ XIL BRAM

- Crear proyectos SW
- Incluir / editar ficheros fuente
- Definir opciones de compilación
- [Generar fichero de comandos del Linker]
- Compilar la aplicación

Gestión de Aplicaciones (*Applications tab*)

The screenshot displays the 'Applications' window in Xilinx ISE DS v13. The 'Software Projects' tree on the left lists several projects, including 'TestApp_Memory_microblaze_0' and 'TestApp_Peripheral_microblaze_0'. Red arrows point from specific items in the tree to callout boxes: from 'Add Software Application Project...' to a box with the same text; from 'Project: TestApp_Memory_microblaze_0' to a box containing 'Set Compiler Options...', 'Mark to Initialize BRAMs' (checked), 'Build Project', 'Clean Project', and 'Delete Project...'; and from the 'Sources' folder under 'TestApp_Peripheral_microblaze_0' to a box containing 'Add Existing Files...' and 'Add New File...'. At the bottom, the 'Applications' tab is highlighted in the software's interface.

Add Software Application Project...

Set Compiler Options...

- ✓ Mark to Initialize BRAMs
- Build Project
- Clean Project
- Delete Project...

Add Existing Files...

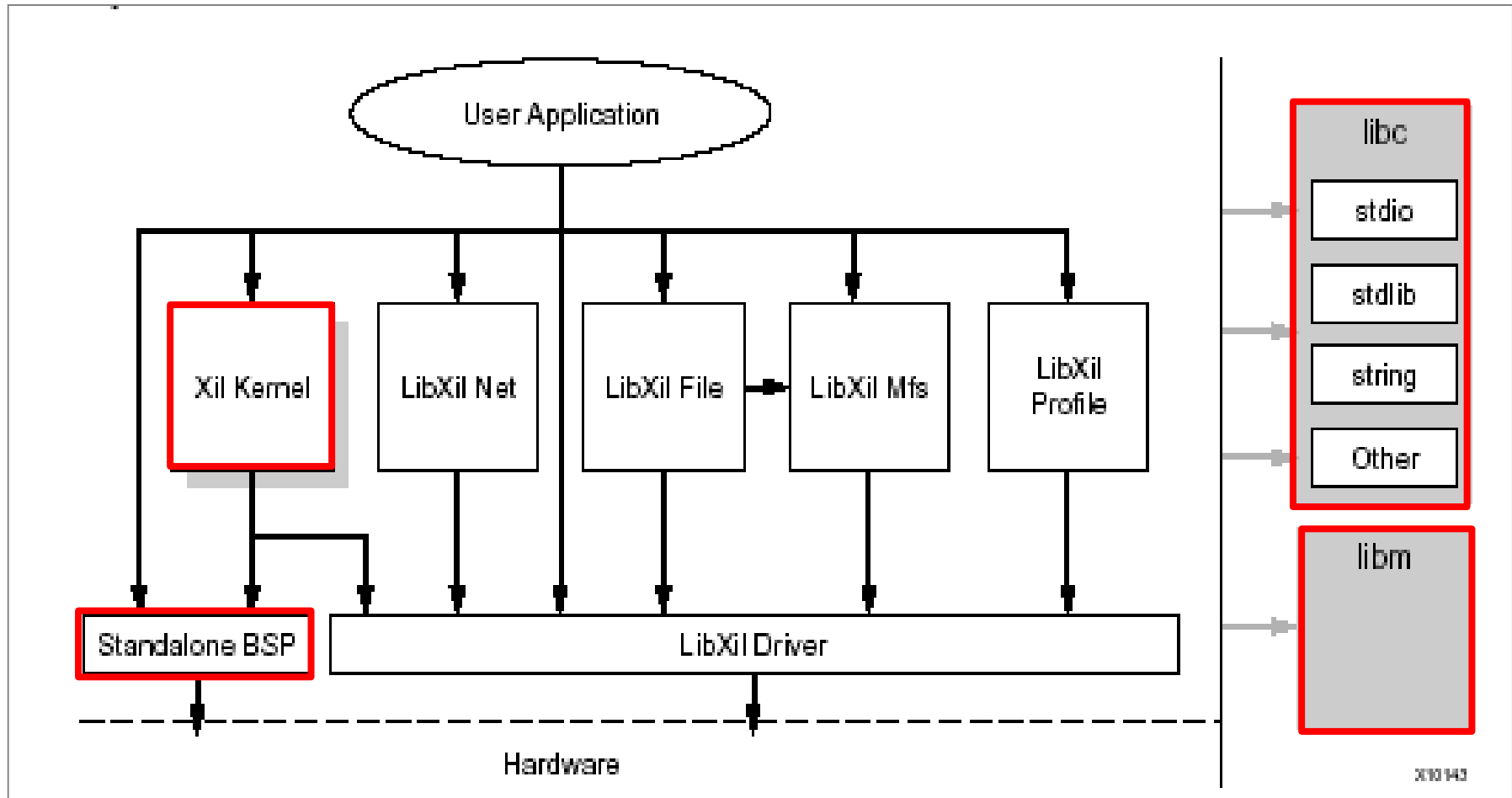
Add New File...

- Se elimina a partir de ISE-DS v13
- Reemplazado por SDK (Eclipse)

Codificación de Aplicaciones

- **EDK proporciona una serie de paquetes de software que facilitan la codificación de las aplicaciones de usuario:**
 - **Drivers de periféricos**
Incluyen estructuras de datos, macrodefiniciones y funciones que facilitan el uso de los distintos módulos IP
 - **Bibliotecas**
 - Estándar: C (**libc**), matemática (**libm**)
 - Bibliotecas **libXil**: Net, File, MFS, Profile
 - **Sistemas operativos**
 - Standalone
 - XilKernel

Codificación de Aplicaciones (cont.)



Drivers de Periféricos

- Un driver es una pieza de software que facilita la utilización del hardware del sistema
- Xilinx proporciona un conjunto de drivers con las siguientes características:
 - Máxima portabilidad
Los drivers se proporcionan como **código fuente en ANSI C**
 - Configurabilidad de la FPGA
Permite la utilización de múltiples dispositivos sin replicación de código.
 - Contempla casos simples y complejos
 - **Nivel 0:** drivers de bajo nivel con **baja funcionalidad** (poca memoria)
 - **Nivel 1:** drivers de alto nivel con **altas prestaciones** (mucha memoria)

Drivers de Periféricos (cont.)

ARQUITECTURA EN CAPAS

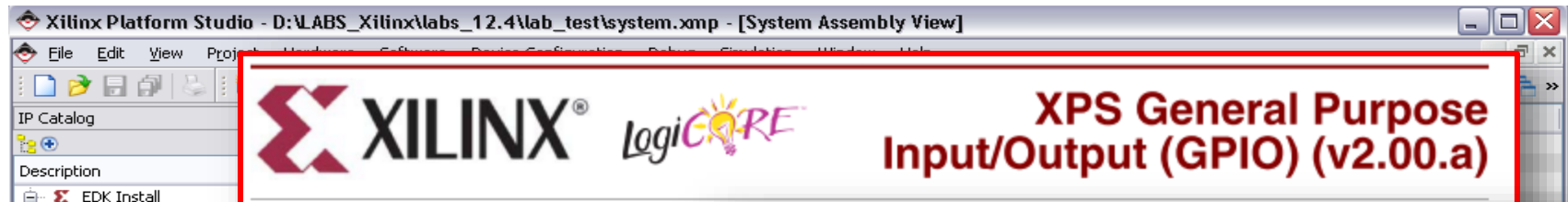


- Permite parámetros de configuración
- Soporta múltiples instancias
- E/S por consultas o interrupciones
- Interfaces de transferencia con buffers
- **Alto consumo de memoria**

- Soporta sólo funciones primarias
- Poco o nulo chequeo de errores
- No permite parámetros de configuración
- Entrada/Salida mediante consultas
- **Bajo consumo de memoria**



Información Hardware (*data sheet*)



XPS General Purpose Input/Output (GPIO) (v2.00.a)

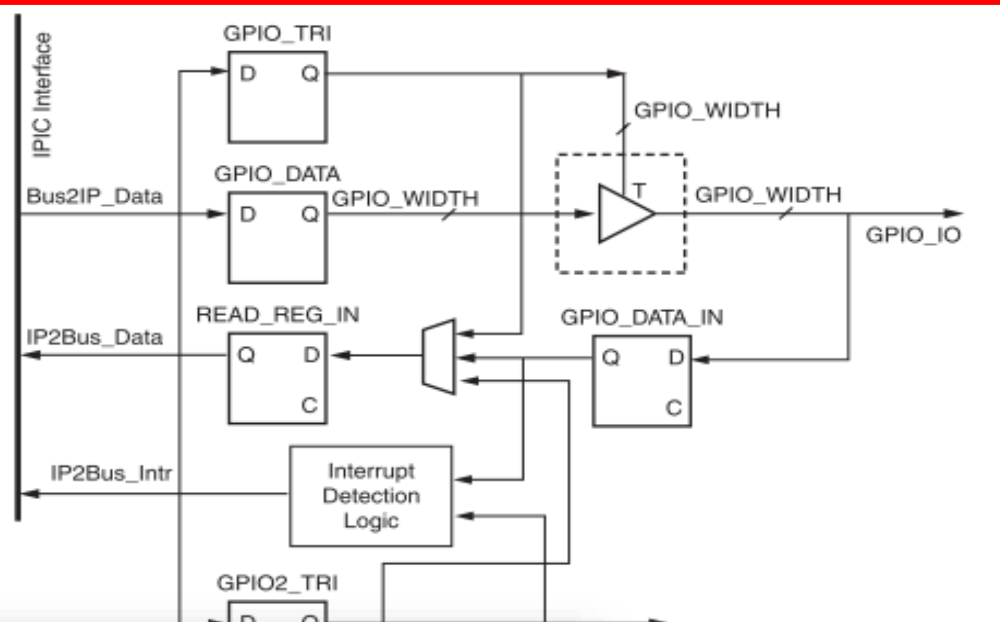
DS569 Apr 19, 2010

Introduction

This document describes the specification of the General Purpose Input/Output (GPIO) peripheral that attaches to the PLBv4.6.

Features

- Connects as a 32-bit slave on PLB v4.6 or 128 bits
- Configurable as single or dual GPIO channels
- Number of GPIO bits configurable from 1 to 32
- Each GPIO bit can be dynamically programmed as input or output



Configure IP ...

View MPD

View IP Modification

View PDF Datasheet

Driver: generic_v

Delete Instance

Make This IP Local

XPS GPIO Registers

Register Name	Description	PLB Address	Access
GPIO_DATA	Channel 1 XPS GPIO Data Register	C_BASEADDR + 0x00	Read/Write
GPIO_TRI	Channel 1 XPS GPIO 3-state Register	C_BASEADDR + 0x04	Read/Write
GPIO2_DATA	Channel 2 XPS GPIO Data register	C_BASEADDR + 0x08	Read/Write
GPIO2_TRI	Channel 2 XPS GPIO 3-state Register	C_BASEADDR + 0x0C	Read/Write

Información Software (API doc)

The screenshot shows the Xilinx Platform Studio interface. The main window displays the 'Xilinx Processor IP Library' for 'gpio v3_00_a'. A red box highlights the left sidebar and the main content area. The sidebar contains a list of IP categories and a context menu for the selected IP. The main content area shows the software drivers for the selected IP, including a description and a list of features.

Xilinx Processor IP Library

Software Drivers

[Main Page](#) | [Data Structures](#) | [File List](#) | [Globals](#)

gpio v3_00_a

This file contains the software API definition of the Xilinx General Purpose I/O (XGpio) device driver.

The Xilinx GPIO controller is a soft IP core designed for Xilinx FPGAs and contains the following general features:

- Support for up to 32 I/O discretes for each channel (64 bits total).
- Each of the discretes can be configured for input or output.
- Configurable support for dual channels and interrupt generation.

The driver provides interrupt management functions. Implementation of interrupt handlers is left to the user. Refer to the provided interrupt example in the examples directory for details.

This driver is intended to be RTOS and processor independent. Any needs for dynamic memory management, threads or thread mutual exclusion, virtual memory, or cache control must be satisfied by the layer above this driver.

Initialization & Configuration

Context Menu (Left Sidebar):

- Configure IP ...
- View MPD
- View IP Modifications (Char...
- View PDF Datasheet
- Browse HDL Sources...
- Driver: gpio_v3_00_a
- Delete Instance
- Make This IP Local

Bottom Bar:

- Console
- Warnings
- Errors

XPS GPIO (*low-level driver*)

Xilinx Processor

Software Drivers

Xilinx Driver

Here is a list of a

- xgpio.c
- xgpio.h
- xgpio_extra.c
- xgpio_g.c
- xgpio_i.h
- xgpio_intr.c
- xgpio_l.h**
- xgpio_selftest.c
- xgpio_sinit.c

Copyright © 1995

Registers

Register 0

- #define
- #define
- #define
- #define
- #define
- #define
- #define

Interrupts

Bit definition

- #define
- #define
- #define

Global Interrupts

Bit definition

- #define

Defines

- #define
- #define

Define Documentation

Xilinx Driver gpio v3_00_a: xgpio_l.h File Reference

I/O direction reg for 1st channel

#define XGpio_WriteReg(BaseAddress, RegOffset, Data)

Write a value to a GPIO register. A 32 bit write is performed. If the GPIO core is implemented in a smaller width, only the least significant data is written.

Parameters:

- BaseAddress* is the base address of the GPIO device.
- RegOffset* is the register offset from the base to write to.
- Data* is the data written to the register.

Returns:

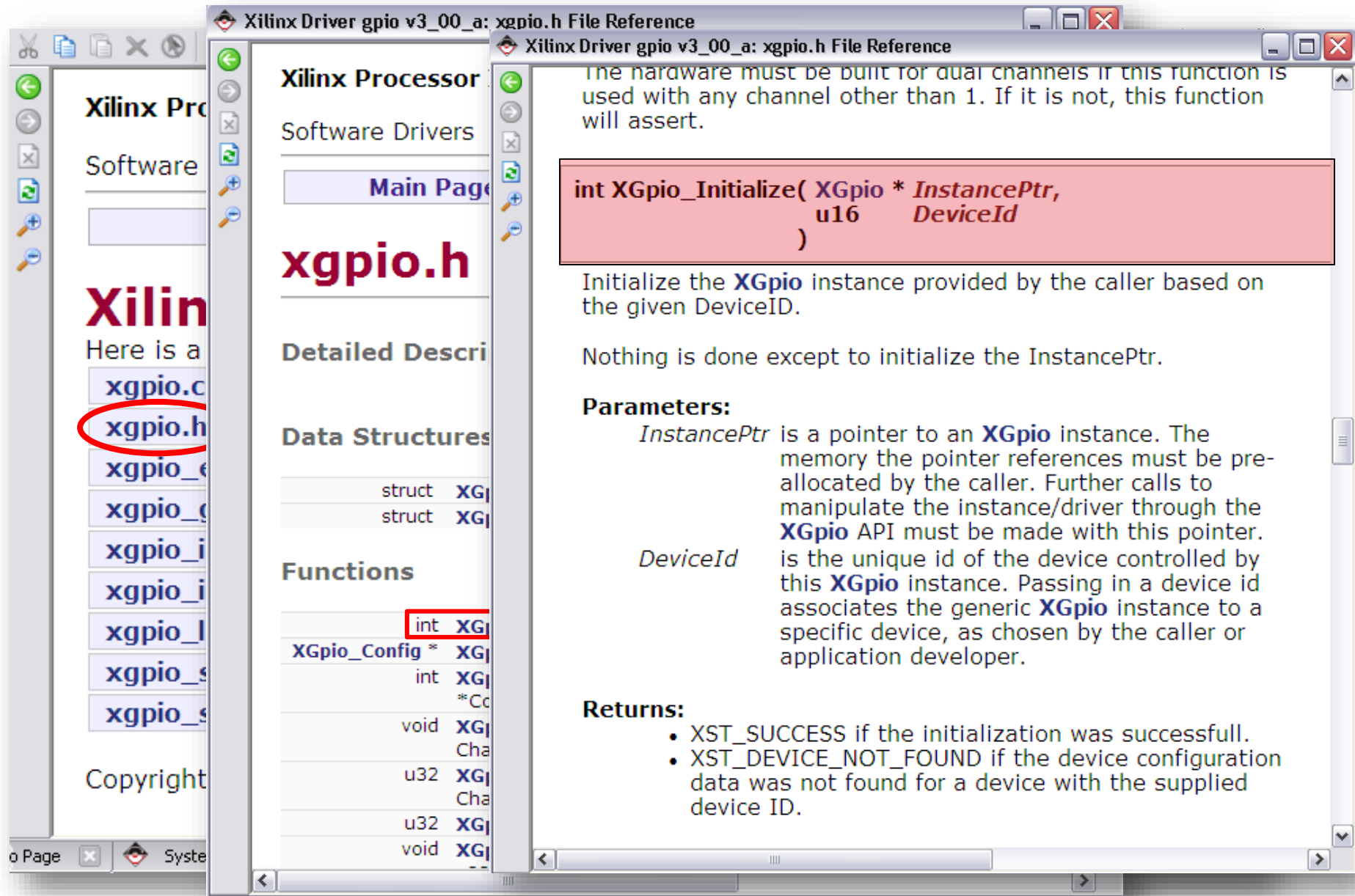
None.

Note:

C-style signature: void **XGpio_WriteReg(u32 BaseAddress, u32 RegOffset, u32 Data)**

Copyright © 1995-2010 Xilinx, Inc. All rights reserved.

XPS GPIO (*high-level driver*)



Xilinx Processor Software Drivers

Main Page

xgpio.h

Detailed Description

Data Structures

struct	XGpio_Config
struct	XGpio_Instance

Functions

int	XGpio_Initialize(XGpio * InstancePtr, u16 DeviceId)
XGpio_Config *	XGpio_ConfigGet(XGpio_InstancePtr)
int	XGpio_DeviceIdGet(XGpio_InstancePtr)
void	XGpio_DeviceIdSet(XGpio_InstancePtr, u16 DeviceId)
u32	XGpio_Read(XGpio_InstancePtr, u32 ChannelId)
u32	XGpio_Write(XGpio_InstancePtr, u32 ChannelId, u32 Data)
void	XGpio_WriteConfig(XGpio_InstancePtr, XGpio_Config * Config)

Xilinx Driver gpio v3_00_a: xgpio.h File Reference

The hardware must be built for dual channels if this function is used with any channel other than 1. If it is not, this function will assert.

```
int XGpio_Initialize( XGpio * InstancePtr,
                    u16 DeviceId
                    )
```

Initialize the **XGpio** instance provided by the caller based on the given DeviceID.

Nothing is done except to initialize the InstancePtr.

Parameters:

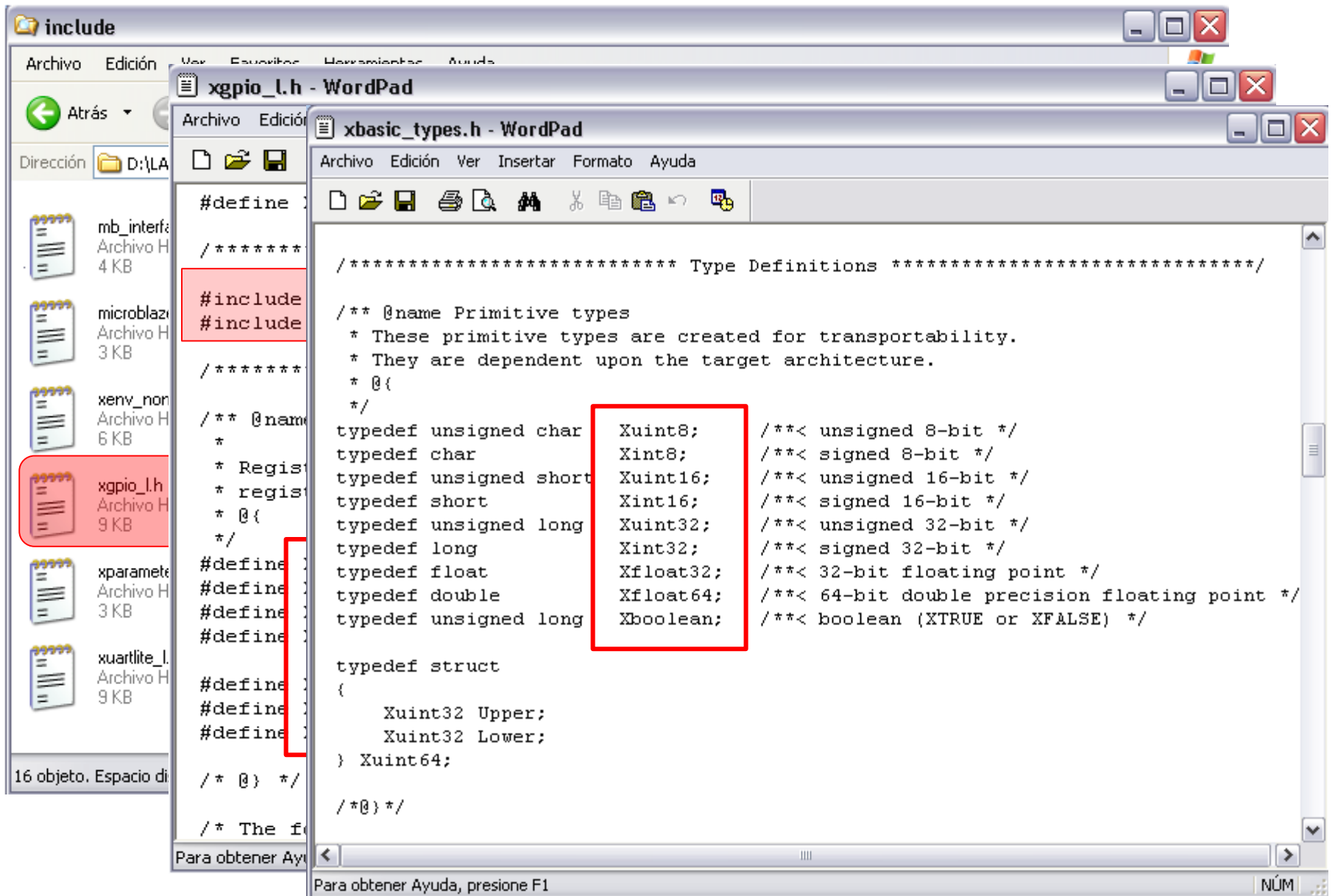
InstancePtr is a pointer to an **XGpio** instance. The memory the pointer references must be pre-allocated by the caller. Further calls to manipulate the instance/driver through the **XGpio** API must be made with this pointer.

DeviceId is the unique id of the device controlled by this **XGpio** instance. Passing in a device id associates the generic **XGpio** instance to a specific device, as chosen by the caller or application developer.

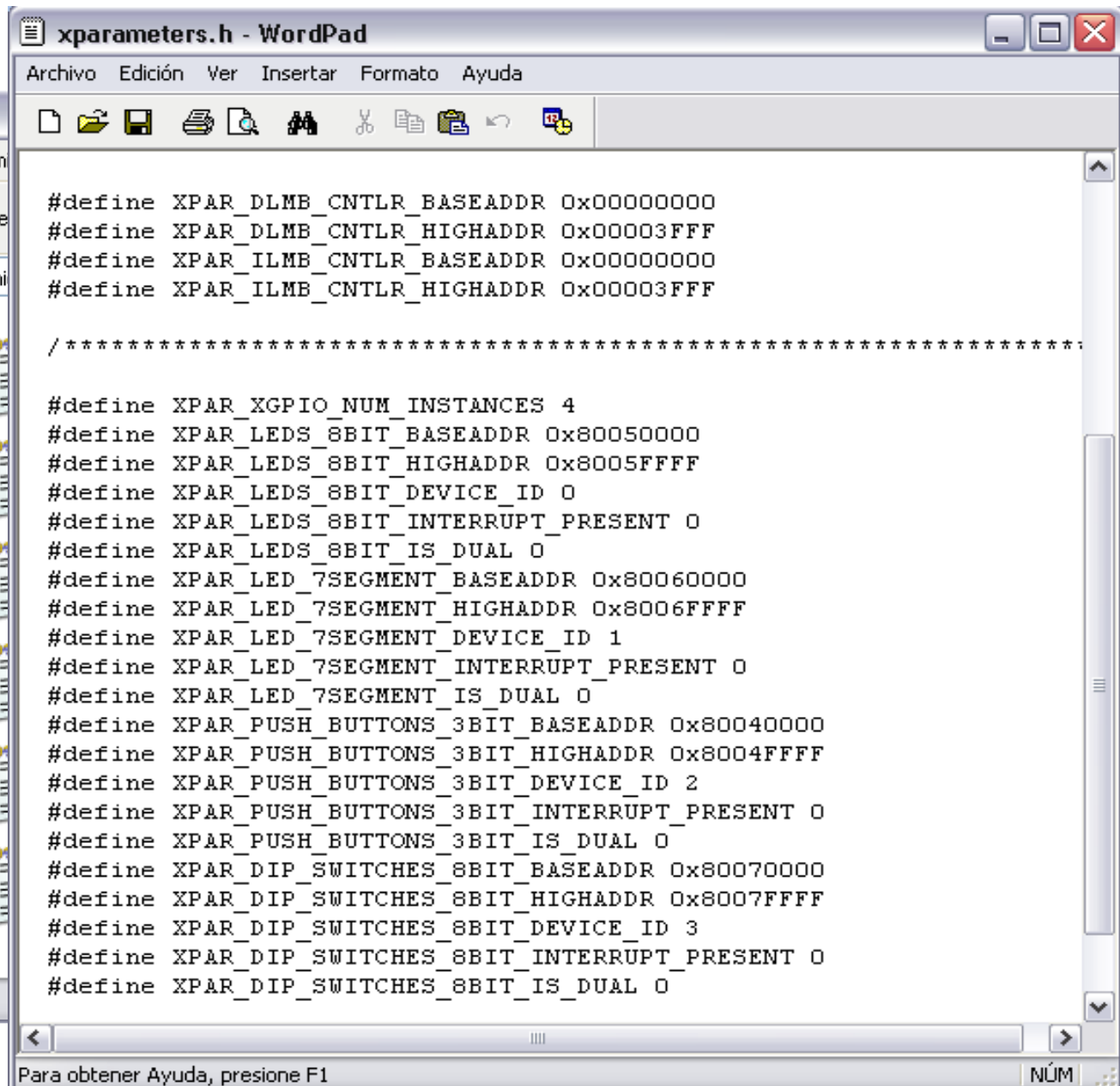
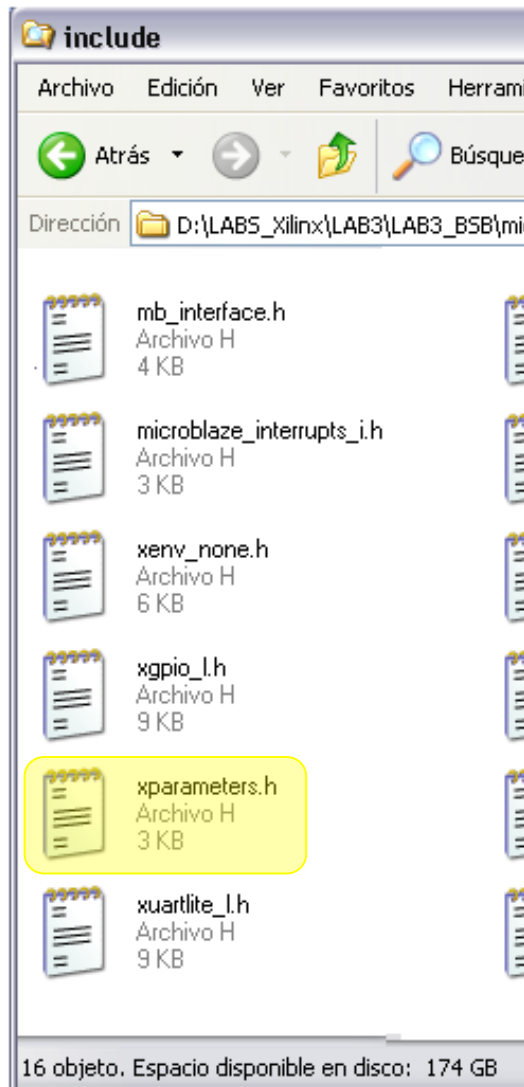
Returns:

- XST_SUCCESS if the initialization was successful.
- XST_DEVICE_NOT_FOUND if the device configuration data was not found for a device with the supplied device ID.

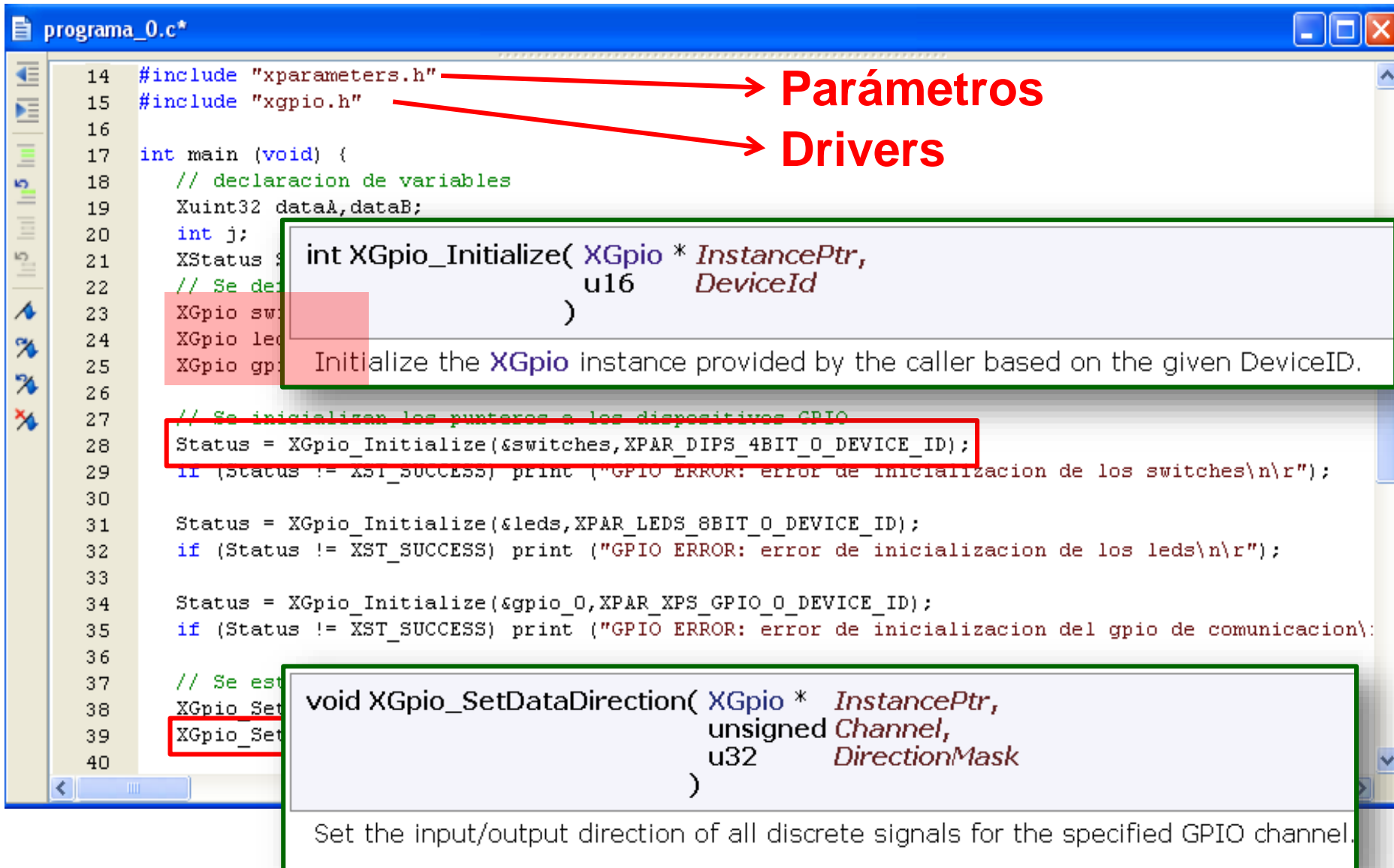
Directorio *microblaze_0/include*



xparameter.h



Programa fuente



The image shows a screenshot of a C program source code editor. The code is for a program named `programa_0.c`. It includes headers `xparameters.h` and `xgpio.h`. The `main` function declares variables `dataA` and `dataB` of type `Xuint32` and an integer `j`. It then calls `XGpio_Initialize` three times to initialize GPIO instances for switches, LEDs, and a communication GPIO. Finally, it calls `XGpio_SetDataDirection` to set the data direction for the communication GPIO. Two callout boxes provide details about the `XGpio_Initialize` and `XGpio_SetDataDirection` functions.

```
14 #include "xparameters.h"
15 #include "xgpio.h"
16
17 int main (void) {
18     // declaracion de variables
19     Xuint32 dataA, dataB;
20     int j;
21     XStatus Status;
22     // Se declaran los punteros a los dispositivos GPIO
23     XGpio switches;
24     XGpio leds;
25     XGpio gpio_0;
26
27     // Se inicializan los punteros a los dispositivos GPIO
28     Status = XGpio_Initialize(&switches, XPAR_DIPS_4BIT_0_DEVICE_ID);
29     if (Status != XST_SUCCESS) print ("GPIO ERROR: error de inicializacion de los switches\n\r");
30
31     Status = XGpio_Initialize(&leds, XPAR_LEDS_8BIT_0_DEVICE_ID);
32     if (Status != XST_SUCCESS) print ("GPIO ERROR: error de inicializacion de los leds\n\r");
33
34     Status = XGpio_Initialize(&gpio_0, XPAR_XPS_GPIO_0_DEVICE_ID);
35     if (Status != XST_SUCCESS) print ("GPIO ERROR: error de inicializacion del gpio de comunicacion\n\r");
36
37     // Se establecen los pines de salida
38     XGpio_SetDataDirection(&gpio_0, 0, 0x00000000);
39     XGpio_SetDataDirection(&gpio_0, 0, 0x00000000);
40 }
```

Parámetros Drivers

int XGpio_Initialize(XGpio * InstancePtr, u16 DeviceId)
Initialize the XGpio instance provided by the caller based on the given DeviceID.

void XGpio_SetDataDirection(XGpio * InstancePtr, unsigned Channel, u32 DirectionMask)
Set the input/output direction of all discrete signals for the specified GPIO channel.

Programa fuente (cont.)

```
programa_0.c*
38  XGpio_SetDataDirection(&leds, 1, 0x00000000);
39  XGpio_SetDataDirection(&gpio_0, 1, 0x00000000);
40
41  // Bucle del programa principal
42  print("***** Inicio del programa\r\n");
43
44  while(1){
45
46      // Se lee el valor de los switches
47      dataA=XGpio_DiscreteRead(&switches,1);
48      // Se pasa el valor de los switches al gpio de comunicacion
49      XGpio_DiscreteWrite(&gpio_0,dataA);
50
51      xil_printf("***** dataA= %d\r\n",dataA);
52
53      // Se lee el dato que proviene del gpio de comunicacion
54      dataB=XGpio_DiscreteRead(&gpio_0,1);
55
56      xil_printf("***** dataB= %d\r\n",dataB);
57
58      // Se envia a los leds el dato recibido
59      XGpio_DiscreteWrite(&leds, 1,dataB);
60      for (j=1; j<10000; j++)
61      {
62      }
63
64 }
```

**u32 XGpio_DiscreteRead(XGpio * InstancePtr,
unsigned Channel
)**

Read state of discretes for the specified GPIO channel.

**void XGpio_DiscreteWrite(XGpio * InstancePtr,
unsigned Channel,
u32 Data
)**

Write to discretes register for the specified GPIO channel.



Bibliotecas *libXil*

❑ Xilinx Micro-Kernel (*XMK*) incluye las siguientes bibliotecas

- **LibXil File**

Proporciona **acceso a ficheros y dispositivos** a través de interfaces como **open, close, read, write, ...**

- **LibXil MFS**

Permite manejar un **sistema de ficheros simple basado en memoria** con funciones como **mfs_init_fs, mfs_create_dir, mfs_rename_file, ...**

- **LibXil Net**

Proporciona **soporte de redes** mediante TCP/IP con funciones como **xilsock_init, xilsock_recvfrom, xilsock_sendto, ...**

- **LibXil Profile**

Proporciona un mecanismo para generar **perfiles de ejecución de programas** mediante histogramas y grafos de llamada compatibles con GNU ***gprof***



Sistemas Operativos

- ❑ **Standalone (*Standalone Board Support Package – SBP*)**
 - Nivel inferior de módulos software para acceder a las funciones específicas del procesador
 - **Interrupciones, excepciones, caché, etc.**
- ❑ **XilKernel**
 - Kernel pequeño, robusto, modular y altamente configurable
 - Procesos y scheduling, sincronización, memoria, colas de mensajes
- ❑ **Sistemas operativos de otras fuentes**
 - **uClinux**: Univ. Queensland (Australia)
 - **Nucleus RTOS**: Accelerated Technology
 - **Petalinux**: Petalogix (Xilinx)

SO Standalone

Lo que apenas se ve...

“**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**.”



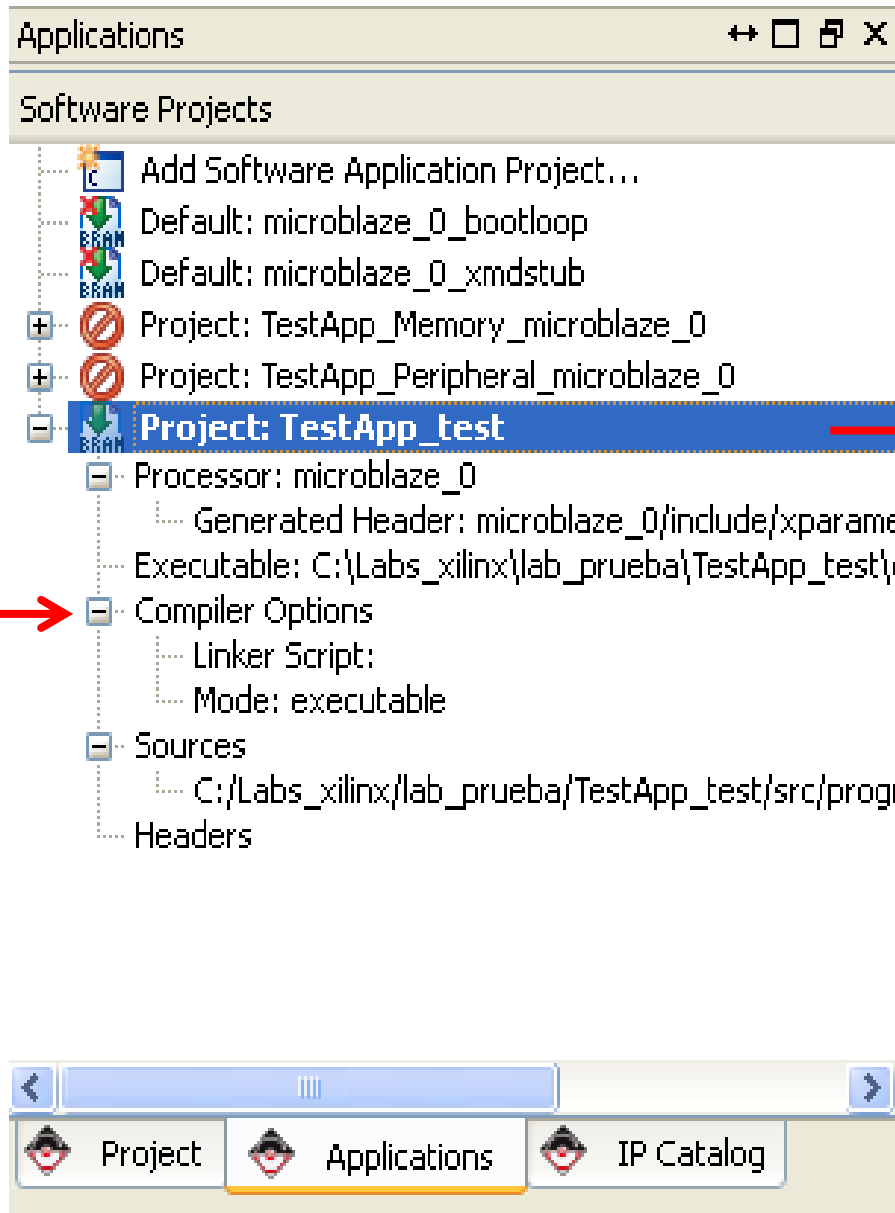
SO Xilkernel

Lo que apenas se ve...

“**Xilkernel** is a **simple and lightweight kernel** that provides **POSIX style services** such as **scheduling, threads, synchronization, message passing** and **timers**. The kernel **requires a programmable timer** that is either built-in or attached to the processor as a peripheral.”



Opciones de Compilación



Set Compiler Options...

✓ Mark to Initialize BRAMs

Build Project

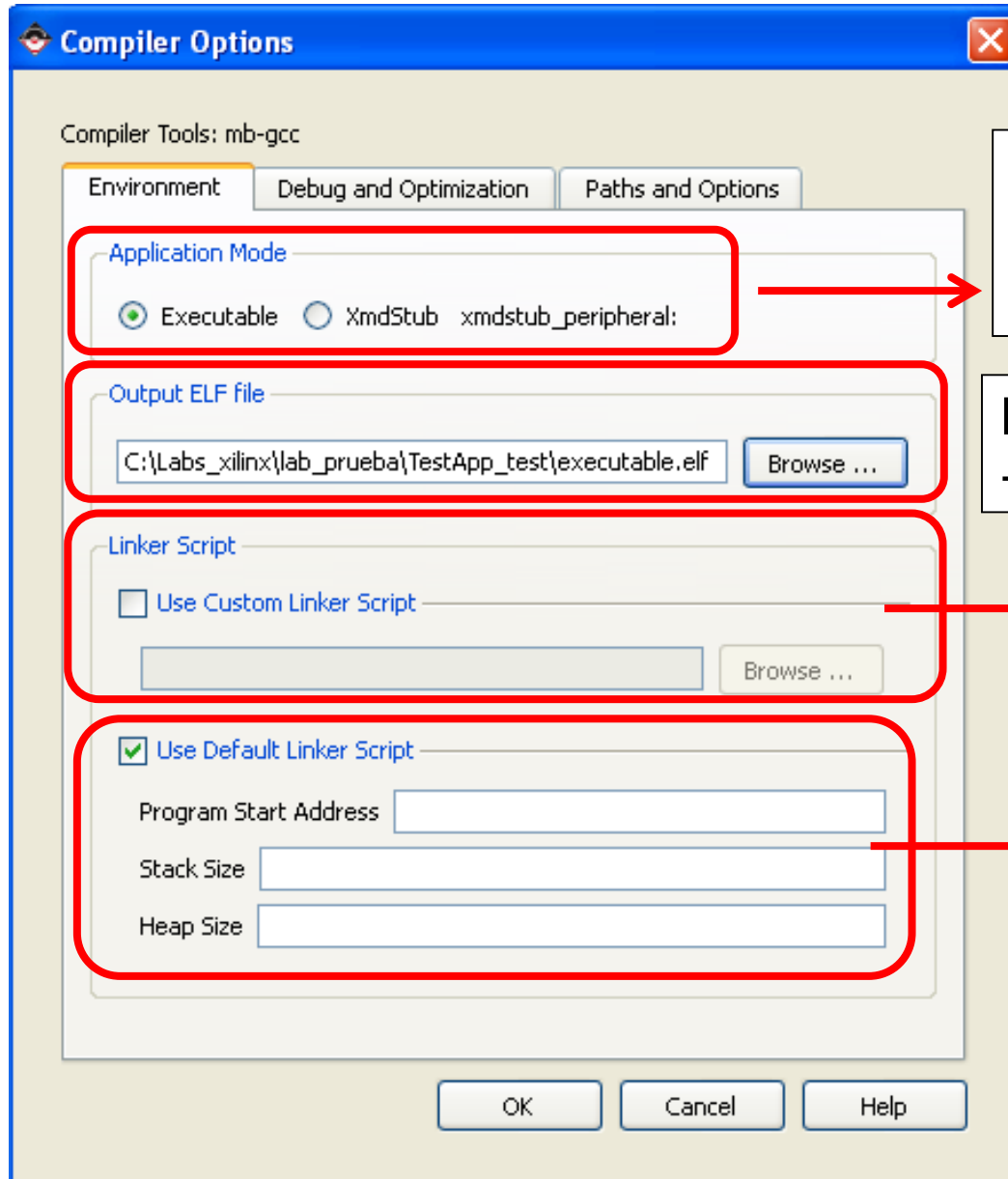
Clean Project

Delete Project...

Make Project Inactive

Generate Linker Script...

Opciones de compilación: *Entorno*



Modo de operación

- Ejecutable
- XmdStub (controlado por *XMD*)

Nombre del fichero de salida

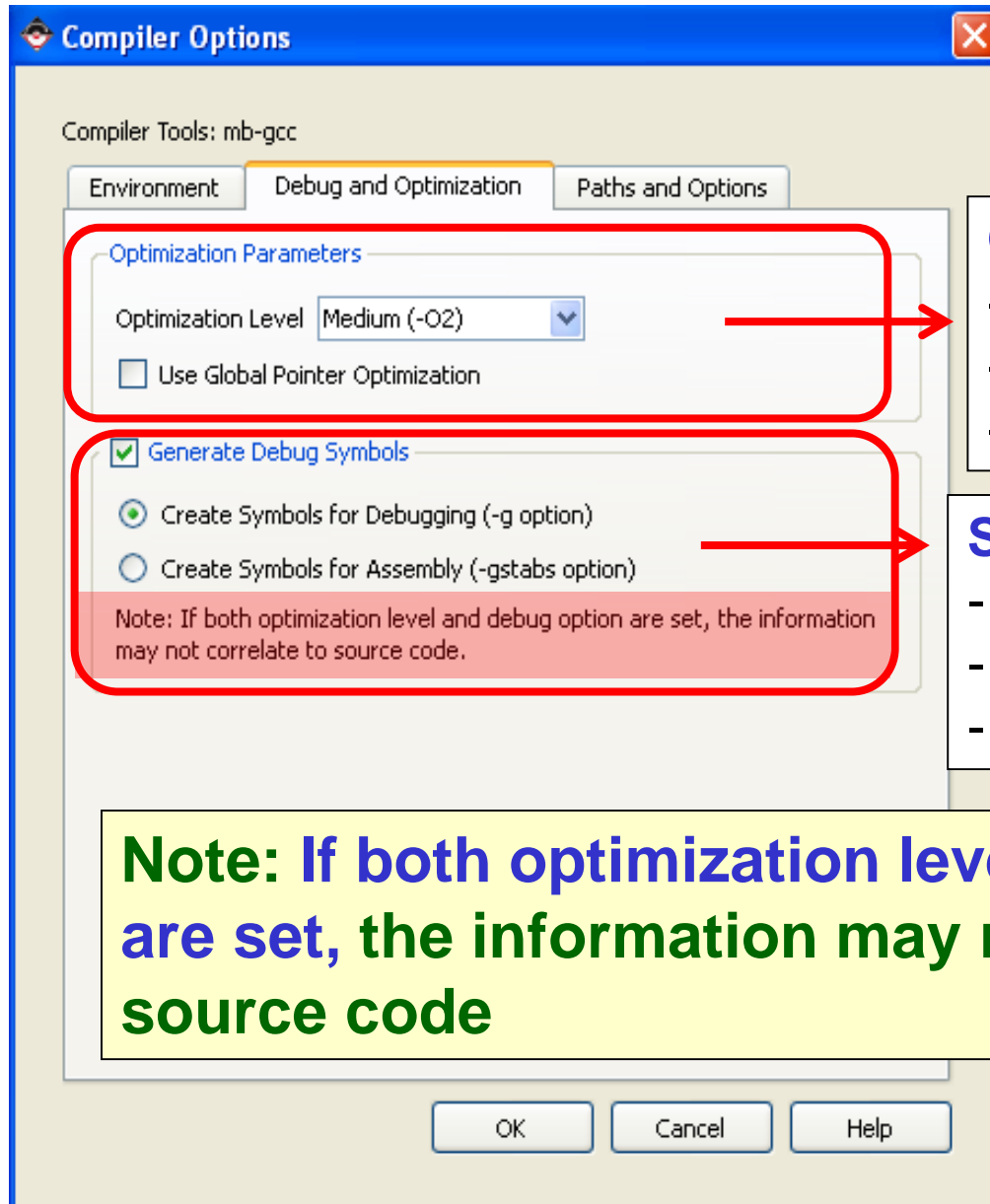
- NO TIENE que ser **executable.elf**

Fichero Linker Script

Memoria

- Dirección de comienzo
- Tamaño del Stack
- Tamaño del Heap

Compilación: *Depuración y Optimización*



Optimización

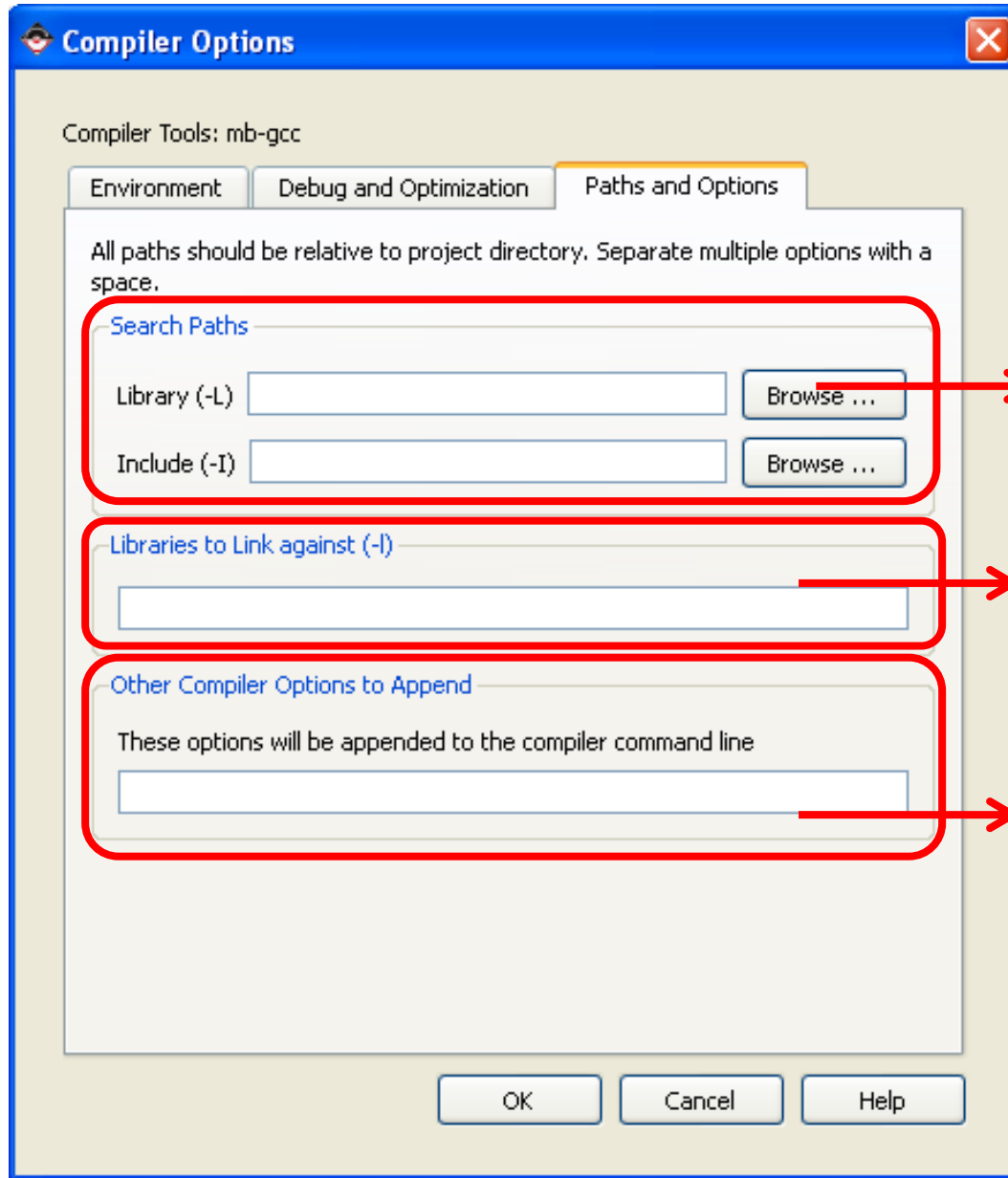
- Niveles: 0 – 3, Size
- Punteros globales
- Activa perfiles de ejecución

Símbolos para Depuración

- No genera símbolos
- Crea símbolos para depuración
- Crea símbolos para ensamblador

Note: If both optimization level and debug option are set, the information may not correlate to source code

Compilación: *Directorios y Opciones*



Camino de búsqueda

- Bibliotecas
- Ficheros de include

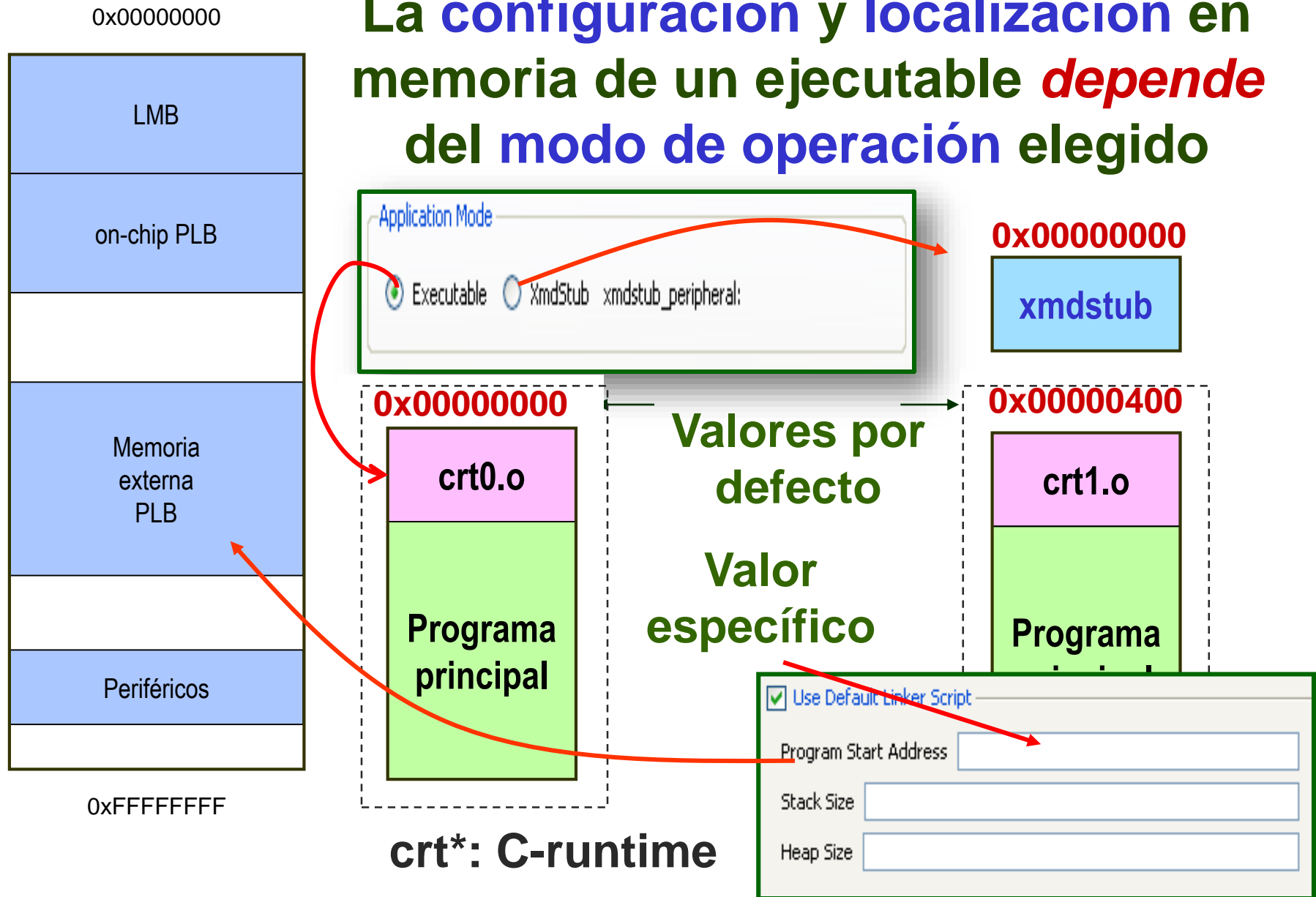
Opciones del Linker

- Bibliotecas
- Fichero de comandos

Parámetros adicionales para el compilador

Espacio de direcciones de memoria

La configuración y localización en memoria de un ejecutable *depende* del modo de operación elegido



Secciones de ficheros ejecutables

- Un fichero ejecutable consta de **diferentes secciones...**
- ...que pueden ser **asociadas a direcciones de memoria específicas**
- Para ello es necesario utilizar un **fichero de comandos** para el *Linker*
- ...que define el layout y la **dirección de comienzo de cada sección**

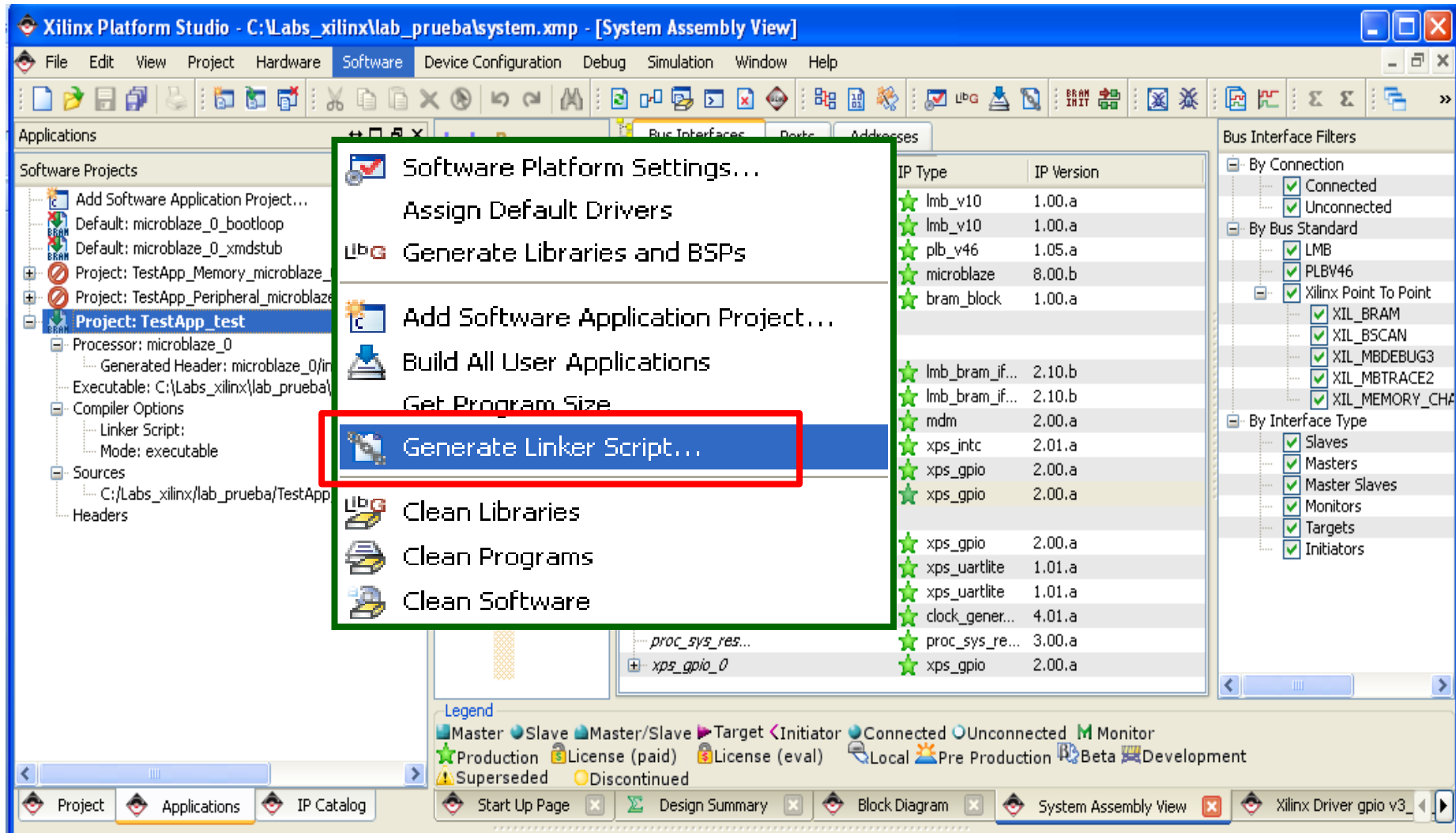
.text	Texto (código)
.rodata	Sólo lectura
.sdata2	Sólo lectura pequeños
.data	Lectura/escritura
.sdata	Lectura/escritura pequeños
.sbss	Pequeños no inicializados
.bss	No inicializados

Secciones de ficheros ejecutables

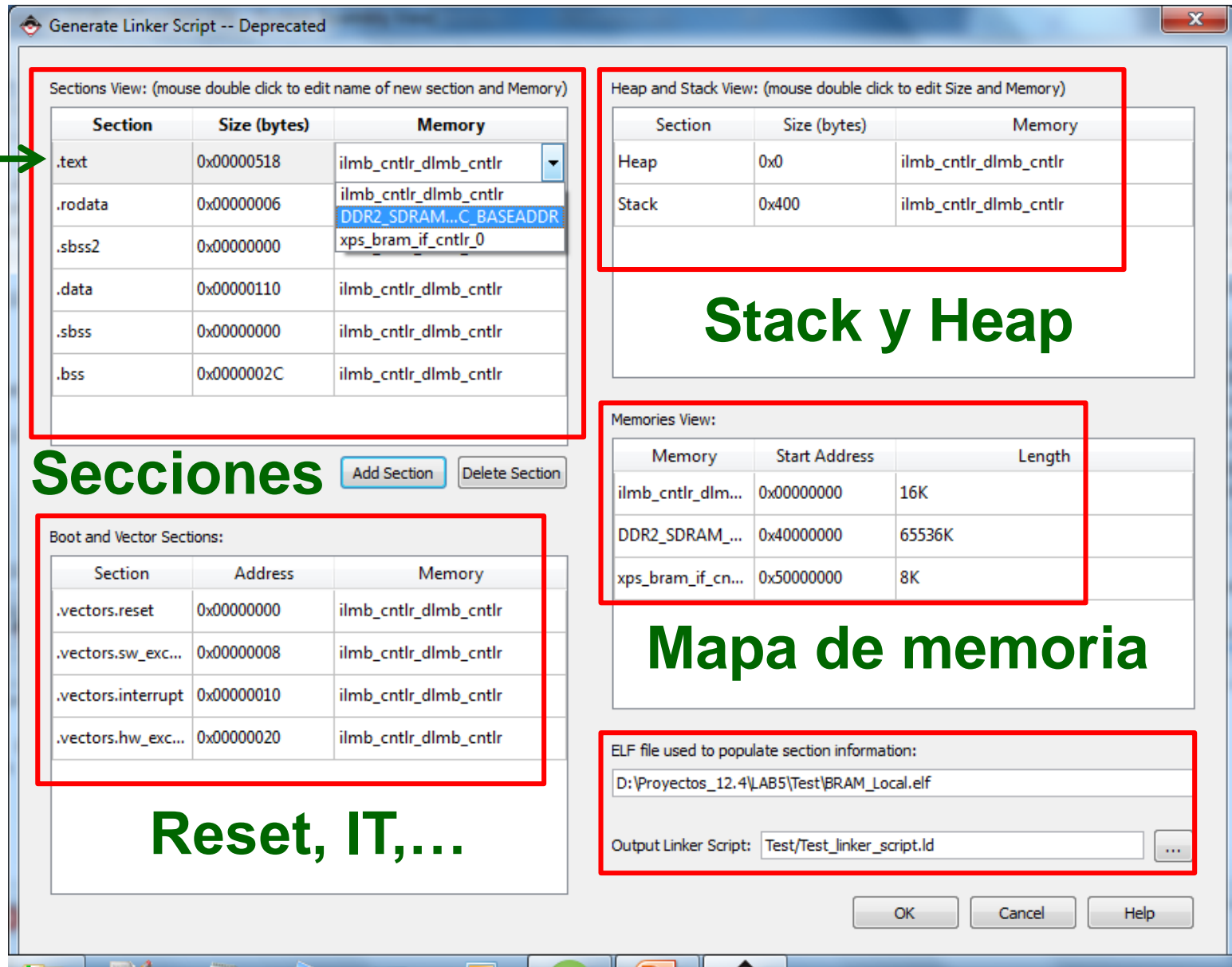
```
int ram_data[10] = {0,1,2,3,4,5,6,7,8,9};    /* DATA */  
const int rom_data[10] = {9,8,7,6,5,4,3,2,1}; /* RODATA */  
int l; /* BSS */  
main() {  
...  
l = l + 10; /* TEXT */  
...  
}
```



Fichero de comandos del *Linker*



Fichero de comandos del *Linker*



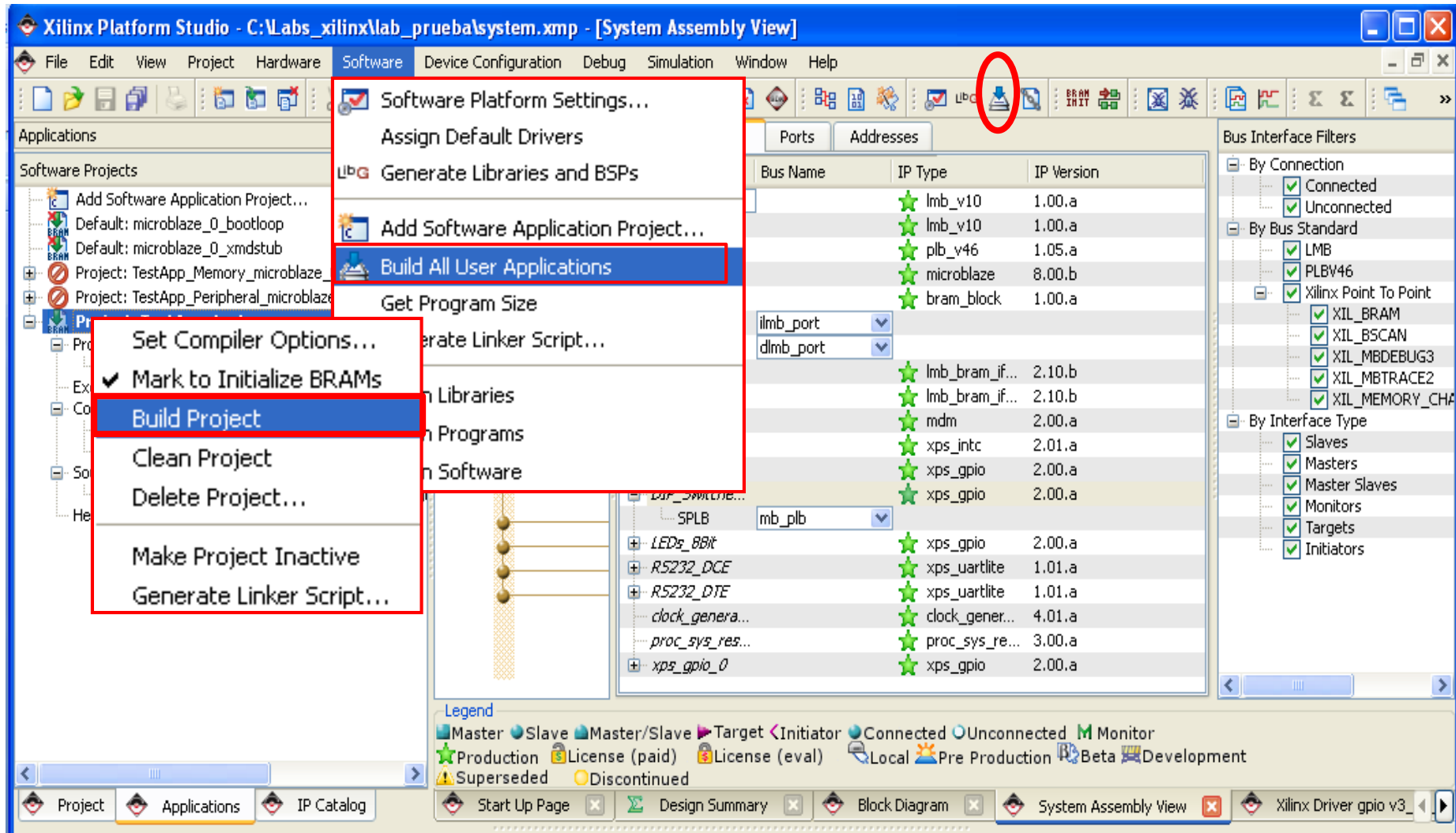
Fichero de comandos del *Linker*

```
7
8  /* Copyright (c) 2009 Xilinx, Inc.  All rights reserved.          */
9  /* Description : MicroBlaze Linker Script                          */
10 /******
11
12  _STACK_SIZE = DEFINED(_STACK_SIZE) ? _STACK_SIZE : 0x400;
13  _HEAP_SIZE = DEFINED(_HEAP_SIZE) ? _HEAP_SIZE : 0x200;
14
15  /* Define Memories in the system */
16  MEMORY
17  {
18      ilmb_cntlr_dlmb_cntlr : ORIGIN = 0x00000050, LENGTH = 0x00003FB0
19      DDR2_SDRAM_C_MPMC_BASEADDR : ORIGIN = 0x44000000, LENGTH = 0x04000000
20  }
21  /* Specify the default entry point to the program */
22
23  ENTRY(_start)
24
25  /* Define the sections, and where they are mapped in memory */
26  SECTIONS
27  {
28      .vectors.reset 0x00000000 : {
29          *(.vectors.reset)
30      }
31      .vectors.sw_exception 0x00000008 : {
32          *(.vectors.sw_exception)
33      }
```

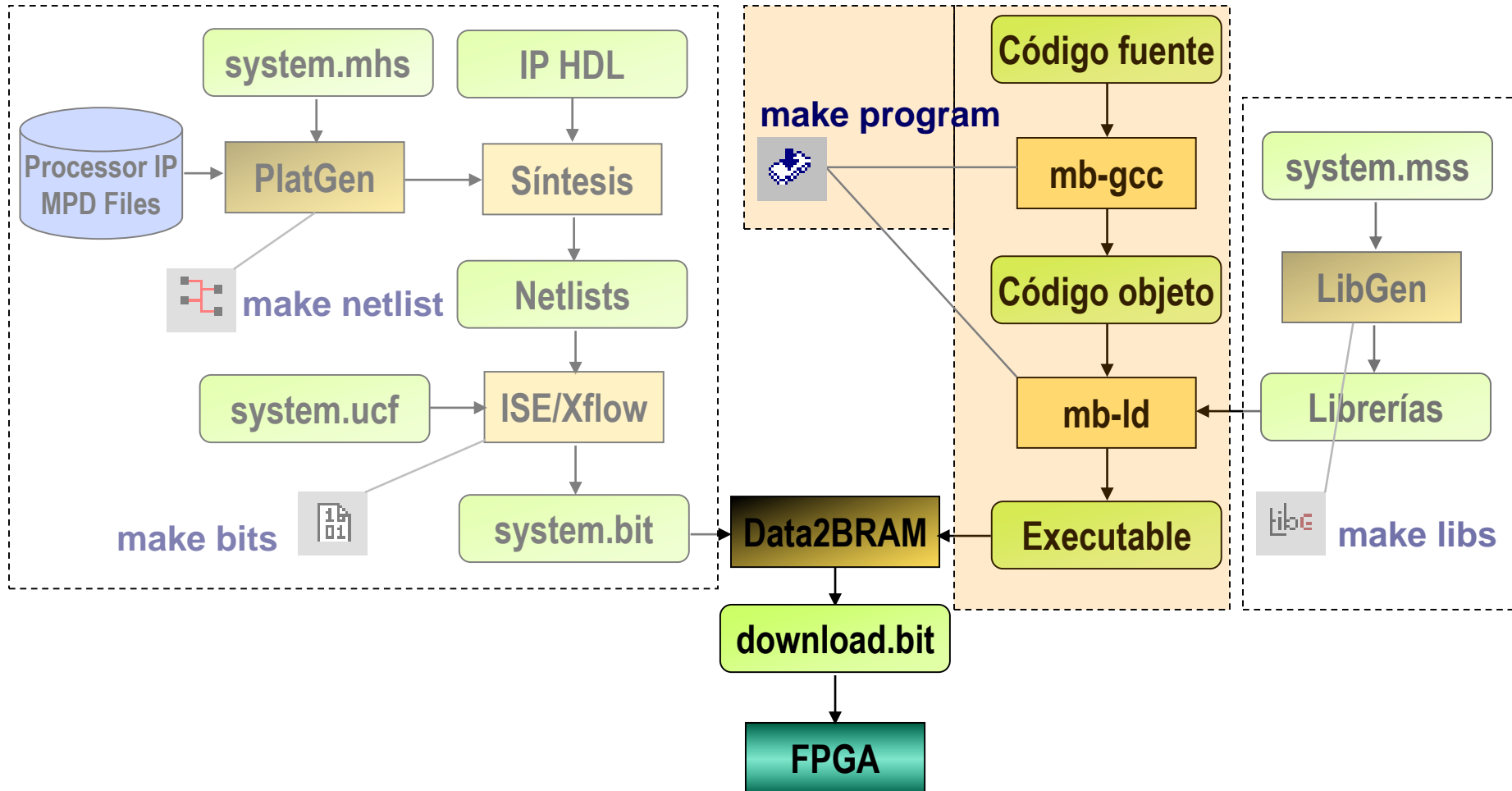
Fichero de comandos del *Linker*

```
35  .vectors.interrupt 0x00000010 : {  
36      *(.vectors.interrupt)  
37  }  
38  .vectors.hw_exception 0x00000020 : {  
39      *(.vectors.hw_exception)  
40  }  
41  .text : {  
42      *(.text)  
43      *(.text.*)  
44      *(.gnu.linkonce.t.*)  
45  } > ilmb_cntlr_dlmb_cntlr  
46  
47  .init : {  
48      KEEP (*(.init))  
49  } > ilmb_cntlr_dlmb_cntlr  
50  
51  .fini : {  
52      KEEP (*(.fini))  
53  } > ilmb_cntlr_dlmb_cntlr  
54  
55  .rodata : {  
56      __rodata_start = .;  
57      *(.rodata)  
58      *(.rodata.*)  
59      *(.gnu.linkonce.r.*)  
60      __rodata_end = .;  
61  } > ilmb_cntlr_dlmb_cntlr  
62
```

Compilación de aplicaciones



Compilación de aplicaciones



Carga de programas en memoria

The screenshot shows the Xilinx Platform Studio interface with the 'Device Configuration' menu open. The menu options are: Update Bitstream, Download Bitstream, Program Flash Memory, Set Compiler Options..., Mark to Initialize BRAMs (highlighted with a checkmark), Build Project, Clean Project, and Delete Project... A red box highlights the 'Device Configuration' menu, and another red box highlights the 'Mark to Initialize BRAMs' option. A callout box points to the 'BRAM INIT' icon in the toolbar, stating: 'Update bitstream with software program information'. Another callout box states: 'Ejecuta **data2BRAM**: system.bit [+ *.elf] = download.bit'. A third callout box states: 'Cuando la aplicación reside en **memoria BRAM** es posible inicializar la memoria al descargar el fichero de configuración de la FPGA'. A fourth callout box states: 'Cuando se emplea **memoria externa** es preciso utilizar el **debugger**'. The background shows the 'Software Projects' tree with 'Project: TestApp_test' selected, and the 'Bus Interfaces' tab active.

Xilinx Platform Studio - C:\Labs_xilinxlab_pruebasystem.xmp - [System Assembly View]

File Edit View Project Hardware **Software** Device Configuration Debug Simulation Window Help

Applications

Software Projects

- Add Software Application Project...
- Default: microblaze_0_bootloop
- Default: microblaze_0_xmdstub
- Project: TestApp_Memory_microblaze_0
- Project: TestApp_Peripheral_microblaze_0
- Project: TestApp_test**
 - Processor: microblaze_0
 - Generated Header: microblaze_0/include
 - Executable: C:\Labs_xilinx\lab_prueba\TestApp_test
 - Compiler Options
 - Linker Script:
 - Mode: executable
 - Sources
 - C:\Labs_xilinx\lab_prueba\TestApp_test\
 - Headers

Update Bitstream

Download Bitstream

Program Flash Memory

Set Compiler Options...

☒ Mark to Initialize BRAMs

Build Project

Clean Project

Delete Project...

Update bitstream with software program information

Ejecuta **data2BRAM**:
system.bit [+ *.elf] =
download.bit

Quando la aplicación reside en **memoria BRAM** es posible inicializar la memoria al descargar el fichero de configuración de la FPGA

Quando se emplea **memoria externa** es preciso utilizar el **debugger**

Configuración de la FPGA

Xilinx Platform Studio - C:\Labs_xilinx\lab_prueba\system.xmp - [System Assembly View]

File Edit View Project Hardware Software **Device Configuration** Debug Simulation Window Help

Applications

Software Projects

- Add Software Application Project...
- Default: microblaze_0_bootloop
- Default: microblaze_0_xmdstub
- Project: TestApp_Memory_microblaze_0
- Project: TestApp_Peripheral_microblaze_0
- Project: TestApp_test**
 - Processor: microblaze_0
 - Generated Header: microblaze_0/include/xparam.h
 - Executable: C:\Labs_xilinx\lab_prueba\TestApp_test\...
 - Compiler Options
 - Linker Script:
 - Mode: executable
 - Sources**
 - C:/Labs_xilinx/lab_prueba/TestApp_test/src/prog...
 - Headers

Update Bitstream

Download Bitstream

Program Flash Memory

Sin título [Configuration Mode] - iMPACT

File Edit View Mode Operations Output Help

Boundary-Scan **Slave Serial** **SelectMAP** **Desktop Configuration**

TDI — [XILINX] — [XILINX] — TDO

xc3s400 download.bit

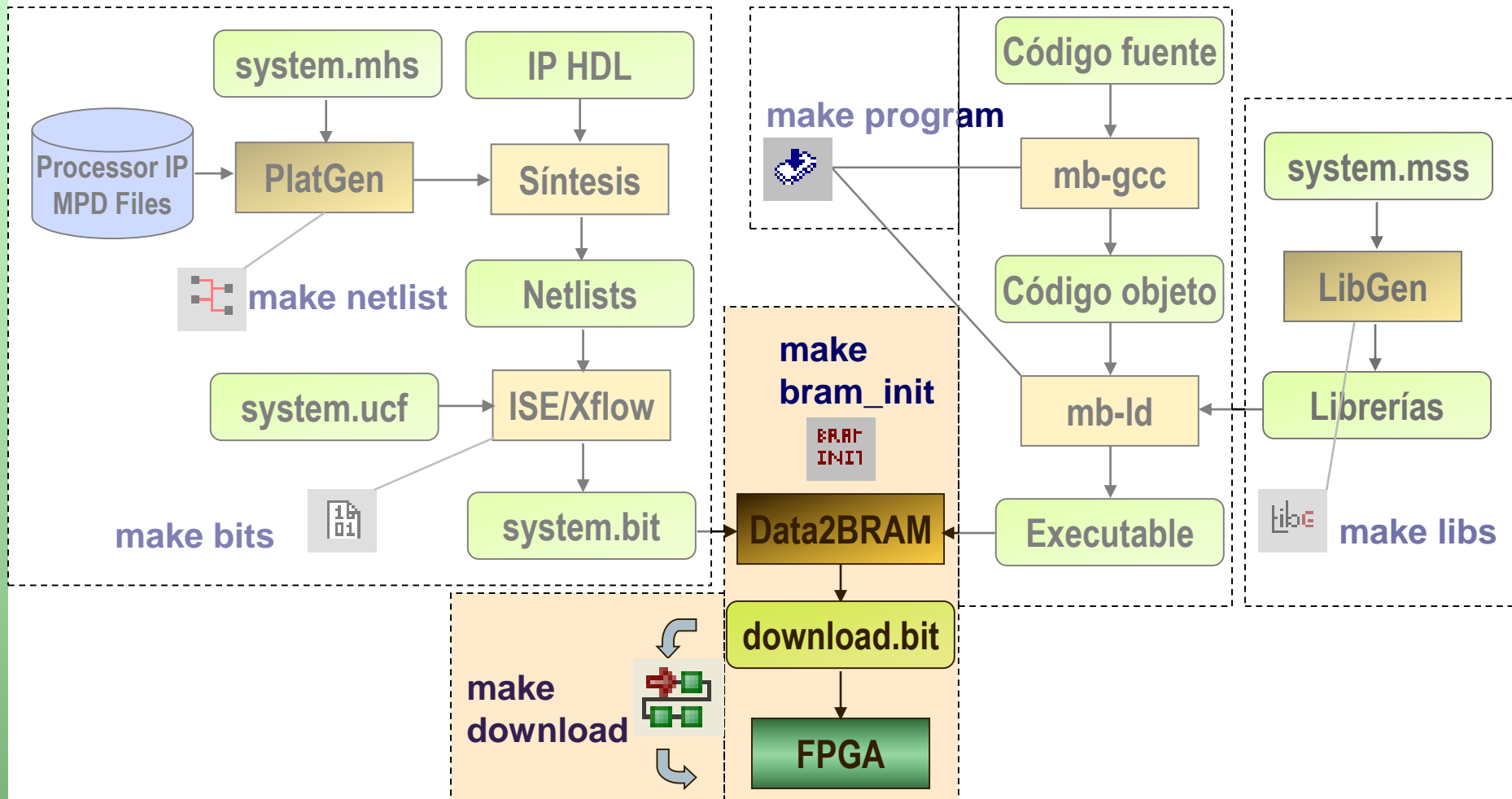
xc402s -- File ? --

PROGRESS_END - End Operation.
Elapsed time = 8 sec.

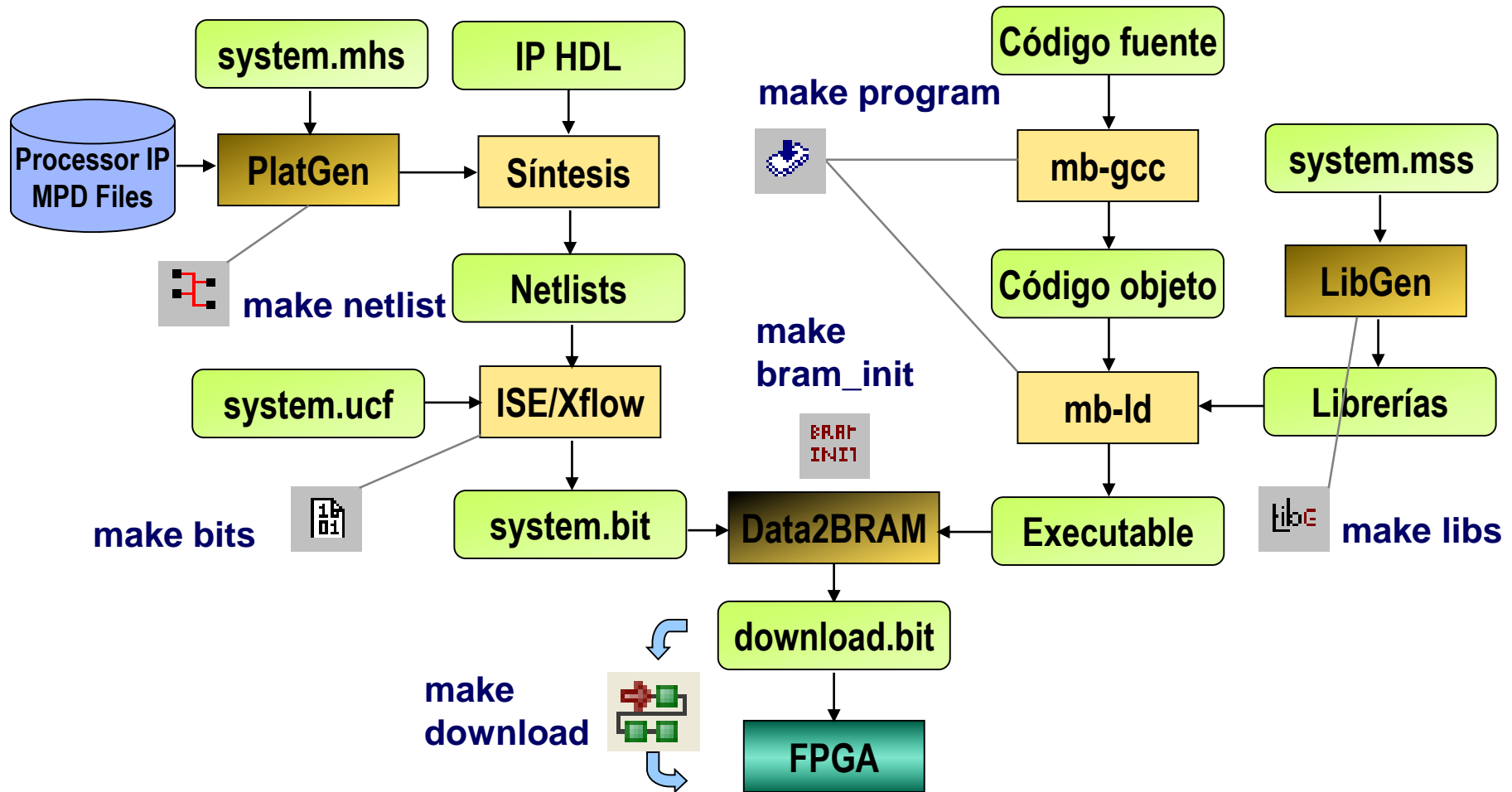
For Help, press F1

Configuration Mode Boundary-Scan

Configuración, descarga y ejecución



Configuración, descarga y ejecución



Documentación

➤ Manuales

- *Embedded System Tools Ref. Manual → GNU Compiler Tools*
- *Platform Studio User Guide → Address Management*
- *Embedded System Tools Guide → Microprocessor Software Specifications*
- *Embedded System Tools Guide → Microprocessor Driver Definition*
- *Embedded System Tools Guide → Microprocessor Library Definition*
- *EDK OS and Libraries Reference Manual*
- *Processor IP Reference Guide*
- *Xilinx Drivers*

➤ Soporte Web

- EDK
 - <http://www.support.xilinx.com/edk>

