

Módulos IP para MicroBlaze

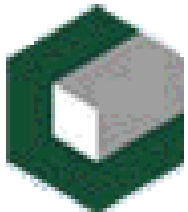
Maestría en Sistemas Digitales

Alejandro J. Cabrera Sarmiento

Dpto. de Automática y Computación

Universidad Tecnológica de La Habana “José Antonio Echeverría”
CUJAE

alex@automatica.cujae.edu.cu

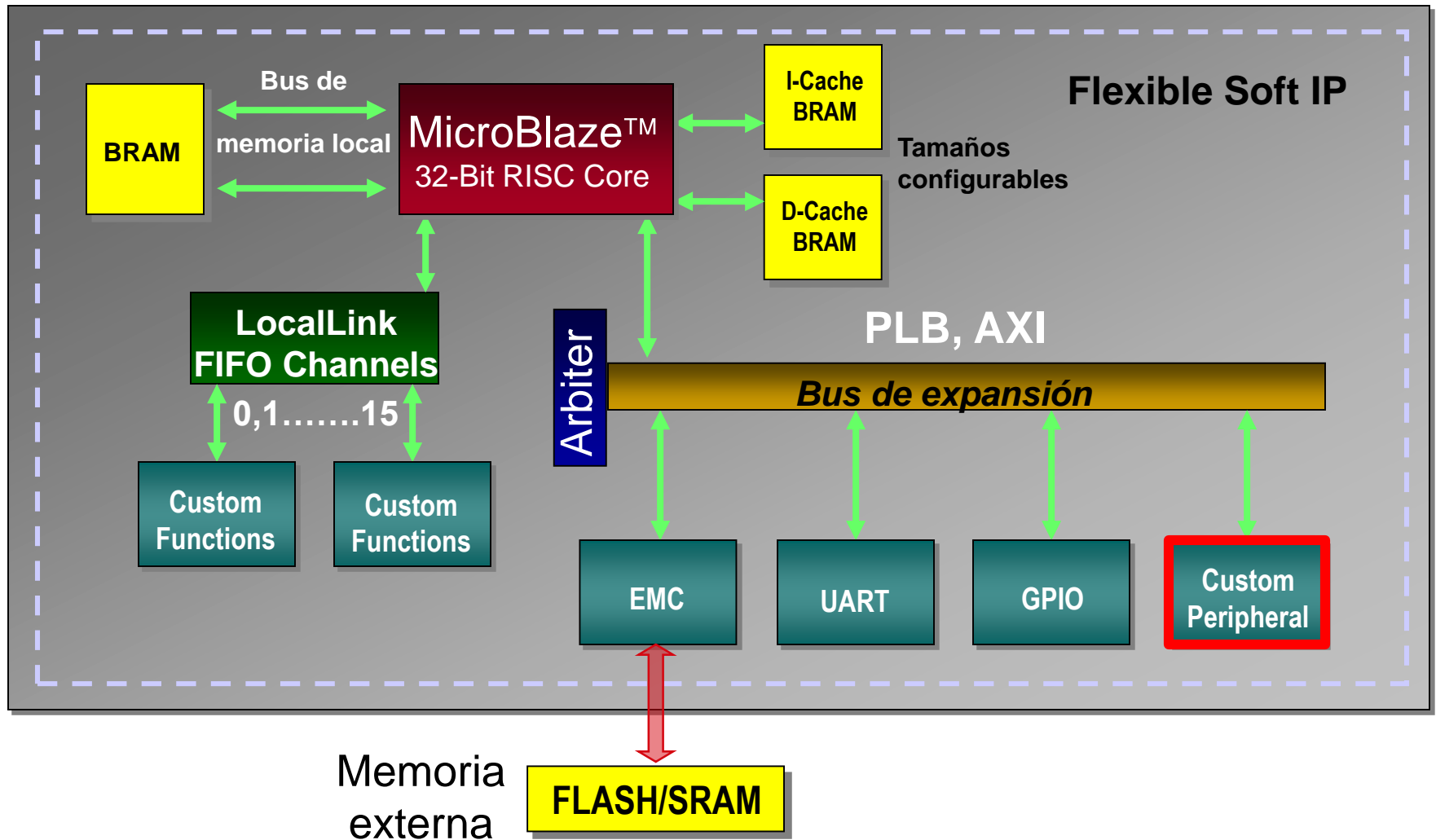


Sumario

- **Módulos IP para MicroBlaze**
- **Procesador MicroBlaze**
- **Módulos IP compatibles con el bus PLB**
 - Puerto de entrada/salida paralelo
 - Temporizador
 - Puerto serie
- **Interrupción en MicroBlaze**
 - Mecanismo de IT
 - Controlador de interrupciones



Sistema de procesamiento MicroBlaze



Procesador MicroBlaze

MicroBlaze es un procesador **RISC de 32 bits** optimizado para implementación sobre **FPGAs de Xilinx**

- Instrucciones de 32 bits con tres operandos y dos modos de direccionamiento
 - Similar a **MIPS**
- Bus de direcciones de 32 bits
- Arquitectura **Harvard**
 - Buses **separados** de instrucciones y datos
- **Dos tipos de buses:**
 - **LMB:** Local Memory Bus
 - **PLB:** Processor Local Bus (IBM)
- 32 registros de propósito general de 32 bits
- Multiplicación hardware

Procesador MicroBlaze (cont.)

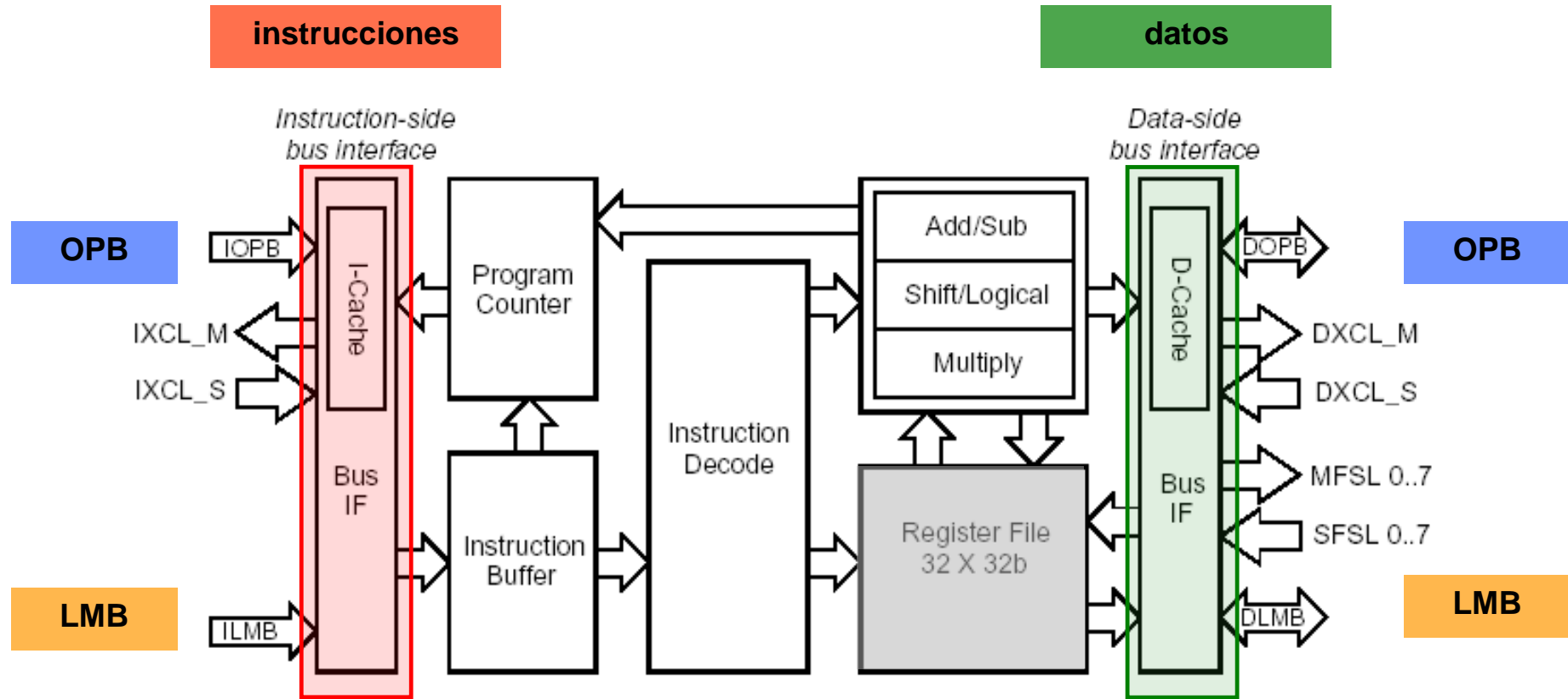
Elementos opcionales:

- Caché de instrucciones y datos
- Coprocesador aritmético
 - Unidad de punto flotante (FPU)
- Interfaces FSL
- Unidad de manejo de memoria (MMU)
 - Soporte de modo protegido

Toda la configuración de MicroBlaze se establece mediante **parámetros**



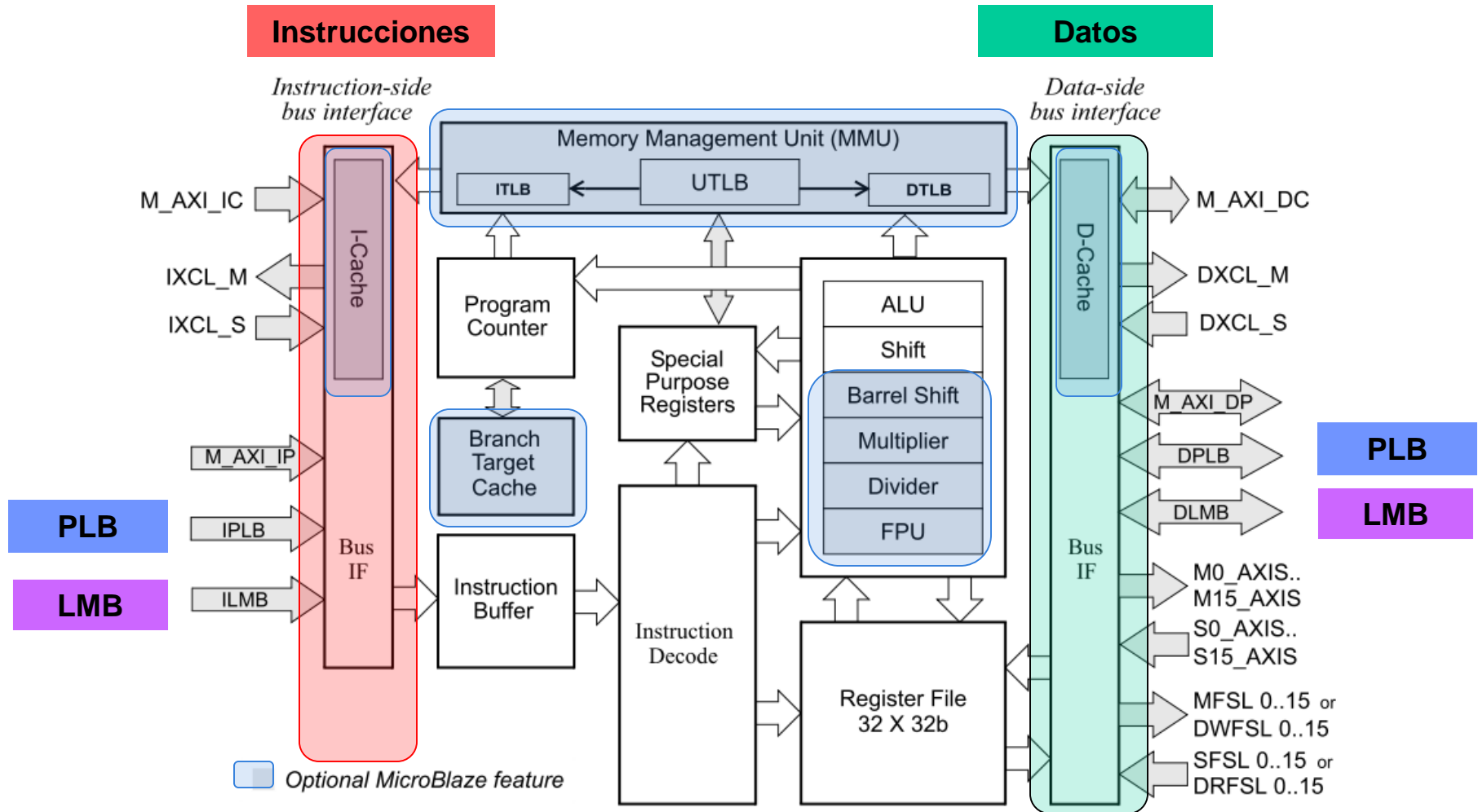
Diagrama de Bloques



Versión **3.0** de MicroBlaze, en **EDK 6.3**



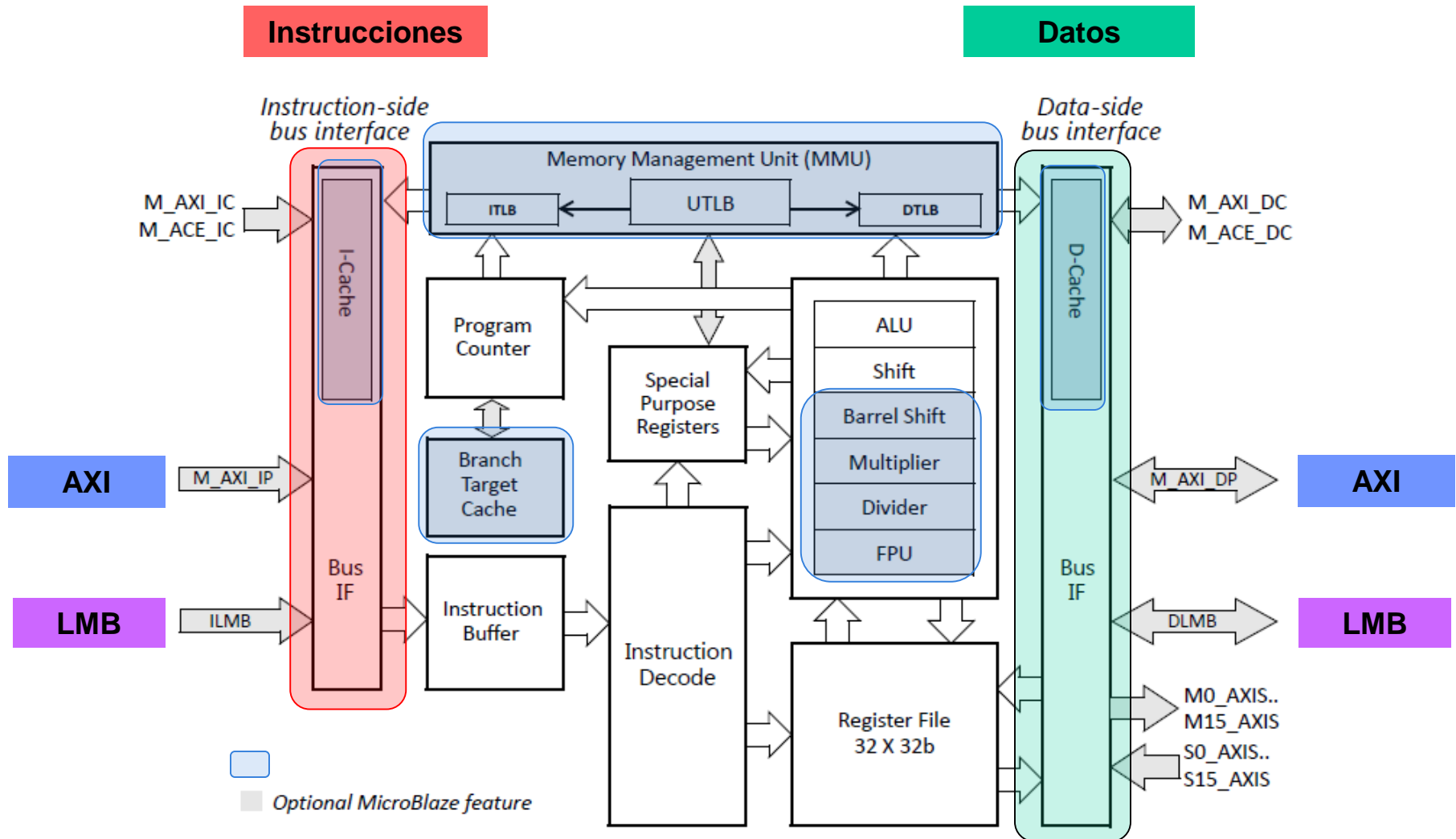
Diagrama de Bloques



Versión **8.0** de MicroBlaze, en **EDK 12.4**



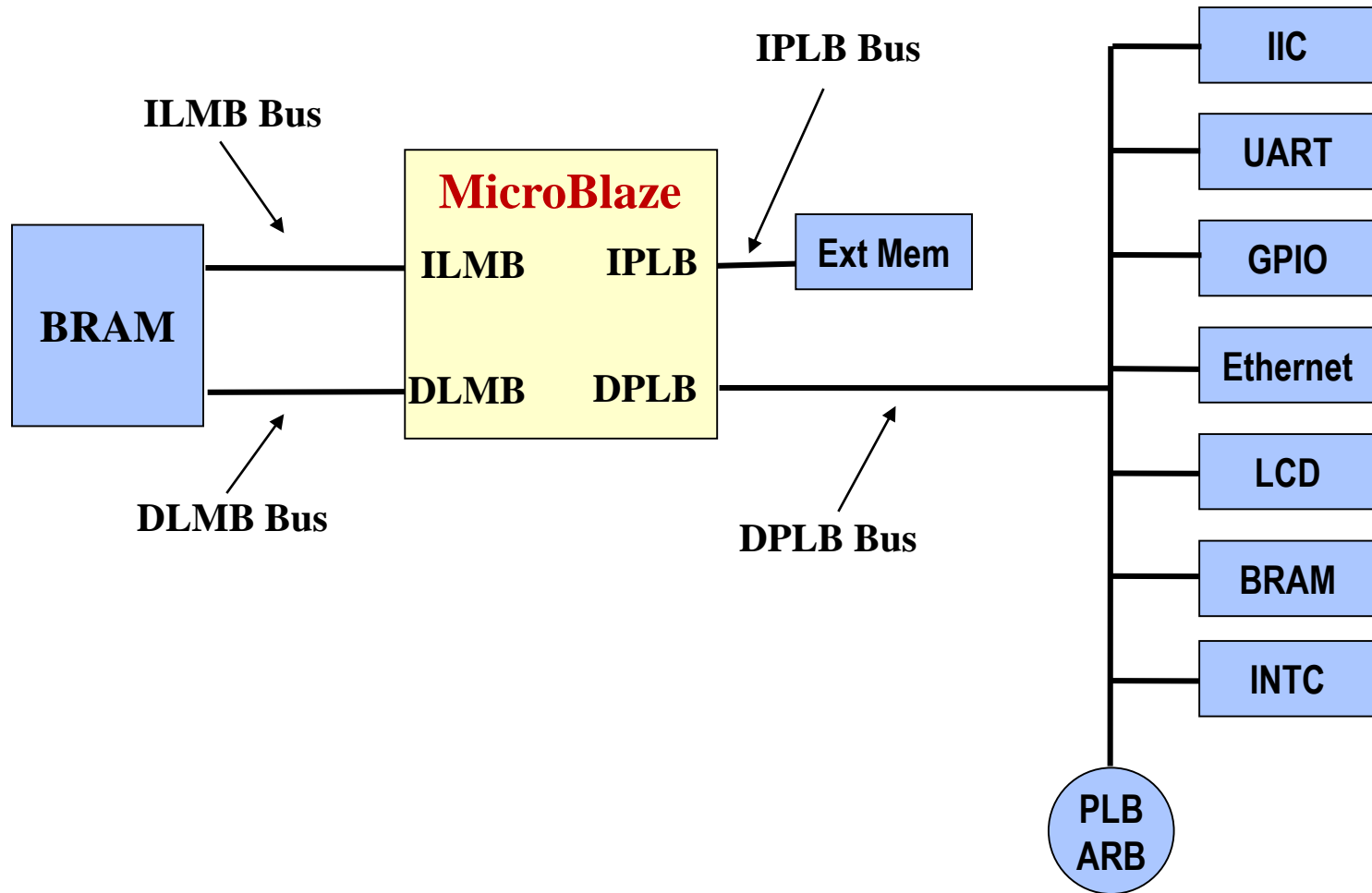
Diagrama de Bloques



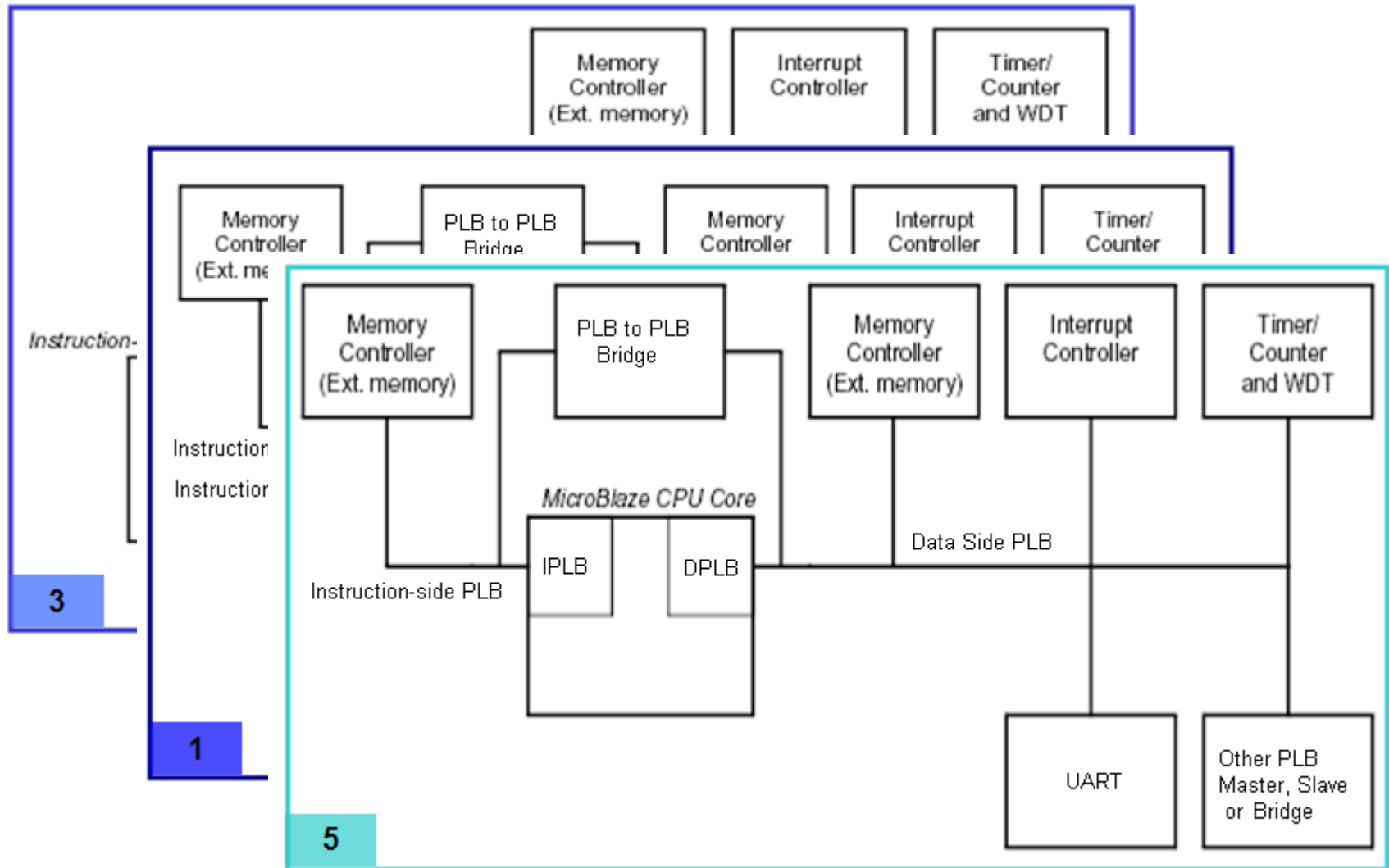
Versión **10.0** de MicroBlaze, en **Vivado 2016.4**



Buses de MicroBlaze



Configuración de buses de MicroBlaze

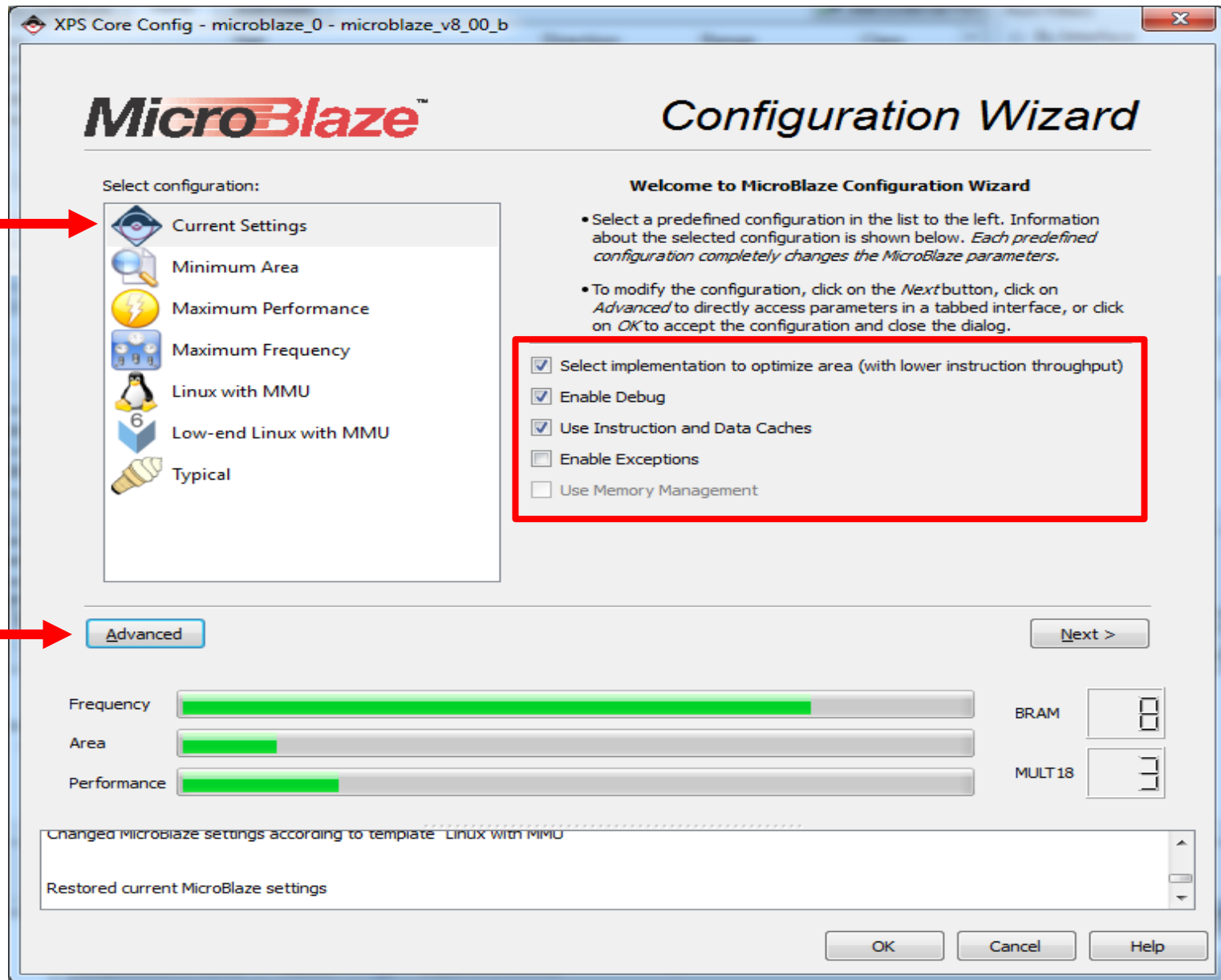


Parámetros de MicroBlaze

El core MicroBlaze puede **configurarse** mediante **parámetros** que activan, seleccionan o dimensionan determinadas características.

Parameter Name	Feature/Description	Allowable Values	Default Value	EDK Tool Assigned	VHDL Type
C_FAMILY	Target Family	aspartan3 aspartan3a aspartan3adsp aspartan3e aspartan6 qspartan6 qspartan6l spartan3 spartan3a spartan3adsp virtex6 spartan3an spartan3e spartan6 spartan6l qrvirtex4 qrvirtex5 qrvirtex4 qrvirtex4 qrvirtex6 virtex4 virtex5 virtex6l	virtex5	yes	string
C_DATA_SIZE	Data Size	32	32	NA	integer
C_INSTANCE	Instance Name	Any instance name	micro blaze	yes	string
C_D_PLB	Data side PLB interface	0, 1	0	yes	integer
C_D_AXI	Data side AXI interface	0, 1	0	yes	integer
C_D_LMB	Data side LMB interface	0, 1	1	yes	integer
C_I_PLB	Instruction side PLB interface	0, 1	0	yes	integer
C_I_AXI	Instruction side AXI interface	0, 1	0	yes	integer
C_I_LMB	Instruction side LMB interface	0, 1	1	yes	integer

Configuración de MicroBlaze



Configuración de MicroBlaze (cont.)

XPS Core Config - microblaze_0 - microblaze_v8_00_b

Instructions Optimization Exceptions **Cache** MMU Debug Interrupt and Reset PVR

Enable Instruction Cache ☒ Enable Data Cache ☒

Instruction Cache Feature Data Cache Feature

Size of the I-Cache in Bytes 2kB Size of D-Cache in Bytes 2kB

Instruction Cache Line Length 4 Data Cache Line Length 4

I-Cache Base Address 0x44000000 D-Cache Base Address 0x44000000

I-Cache High Address 0x47FFFFFF D-Cache High Address 0x47FFFFFF

Enable I-Cache Writes ☒ Enable D-Cache Writes ☒


Use Cache Links for All I-Cache Memory Accesses ☒ Use Cache Links for All D-Cache Memory Accesses ☒

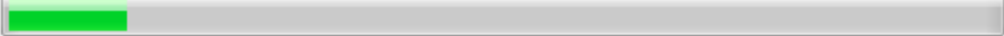
Use Distributed RAM for I-Cache Tags ☐ Use Distributed RAM for D-Cache Tags ☐

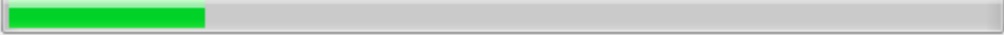
Number of I-Cache Streams 0 Enable Write-back Storage Policy ☐

Number of I-Cache Victims 0 Number of D-Cache Victims 0

Wizard

Frequency  BRAM 8

Area  MULT18 3

Performance 

Módulos IP para MicroBlaze

Incluidos en EDK

○ Periféricos estándar

- XPS GPIO
- XPS Timer/Counter
- XPS UART-Lite
- XPS Interrupt Controller

- XPS TimeBase/WDT
- XPS SPI Interface
- XPS UART (16550)
- ...

○ Debug

- MDM
- ChipScope ICON
- ChipScope PLB IBA

○ Buses

- LMB
- PLB
- FSL

○ Controladores de memoria

- LMB BRAM Controller
- XPS BRAM Controller
- MPMC (DDR Controller)
- XPS MCEM Controller (Flash, SRAM)
- XPS S ACE Controller (Compact Flash)

○ IPs de usuarios

- XPS IPIF Interface



Procedimiento para trabajar con módulos IP

- **Dominar HW del módulo IP**
 - **Características**
 - **Estructura interna**
 - **Señales**
 - **Registros asociados**
 - **Mecanismo de interrupción**
 - **Parámetros de configuración**
- **Dominar las funciones SW disponibles**
 - **Drivers de bajo / alto nivel**
 - **Parámetros**
 - **Utilización de las funciones**

Puerto de E/S paralelo *XPS GPIO*

- Número de **canales** configurables (**1 ó 2**)
- Cada **canal** puede ser configurado como **entrada solamente** ó **salida solamente**
 - **Reducción del consumo de recursos**
- Número de **bits** configurable (**1 a 32 bits**)
- Cada **bit** puede ser configurado como **entrada o salida**
 - **Registro de “dirección”**
- Posibilidad de generación de **interrupción**
 - **Transición en un bit de entrada**
- Valores de **RESET** independientes para cada bit



XPS GPIO (data sheet)



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

DS569 Apr 19, 2010

Product Specification

Introduction

This document describes the specifications for the General Purpose Input/Output (GPIO) core for the Processor Local Bus (PLB). The XPS GPIO is a 32-bit peripheral that attaches to the PLBv4.6.

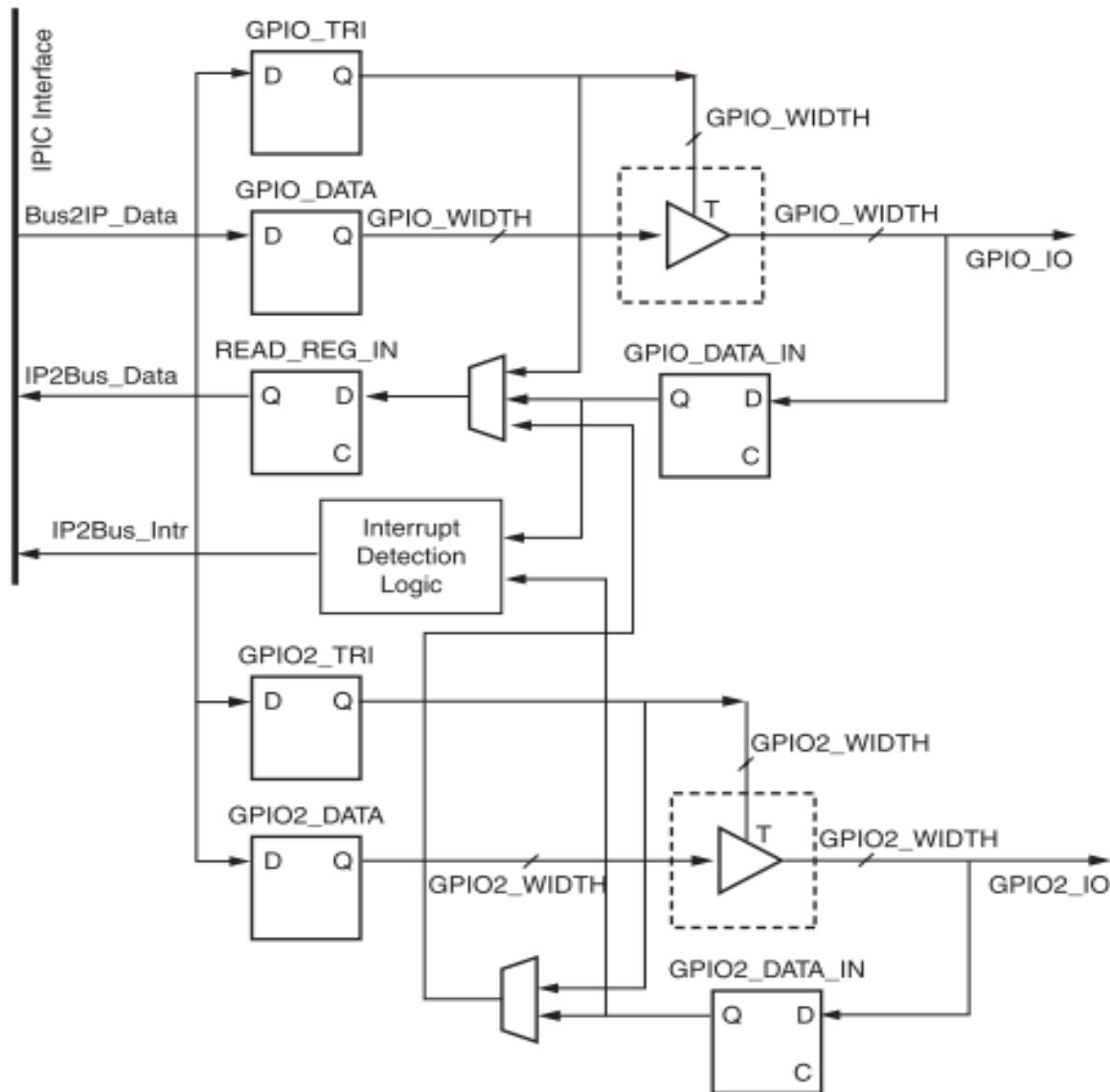
Features

- Connects as a 32-bit slave on PLB v4.6 bus of 32, 64 or 128 bits
- Configurable as single or dual GPIO channel(s)
- Number of GPIO bits configurable from 1 to 32 bits
- Each GPIO bit can be dynamically programmed as input or output
- Width of each of the channels can be individually configured
- Independent reset values for each bit of all registers
- Optional interrupt request generation

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	See EDK Supported Device Families .	
Version of Core	xps_gpio	v2.00.a
Resources Used		
	Min	Max
Slices	Refer to the Table 12 , Table 13 , Table 14 , Table 15 and Table 16	
LUTs		
FFs		
Block RAMs	N/A	



Estructura interna



GPIO_CORE Dual Channel Implementation

Registros asociados (datos)

- En cada canal:
 - Registro de **datos** (*GPIOx_DATA*)
 - Registro de “**dirección**” (*GPIOx_TRI*)
 - Para configuración dinámica de E/S
 - “1”: **Entrada**; “0”: **Salida**

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



Registros asociados (interrupción)

- Registro de **habilitación global** (**GIER**), **1 bit**
- Registro de **habilitación de cada canal** (**IP_IER**), **2 bit**
- Registro de **estado** (**IP_ISR**) de solicitud, **2 bit**
 - Es también el **registro de solicitud**
 - Un bit para cada canal
 - Se setea al existir **una transición en un pin**
 - El bit de solicitud **se limpia al escribir un “1”**

Register Name	Description	PLB Address	Access
GIER	Global Interrupt Enable Register	C_BASEADDR + 0x11C	Read/Write
IP_IER	IP Interrupt Enable Register	C_BASEADDR + 0x128	Read/Write
IP_ISR	IP Interrupt Status Register	C_BASEADDR + 0x120	Read/TOW ^[1]

Notes:

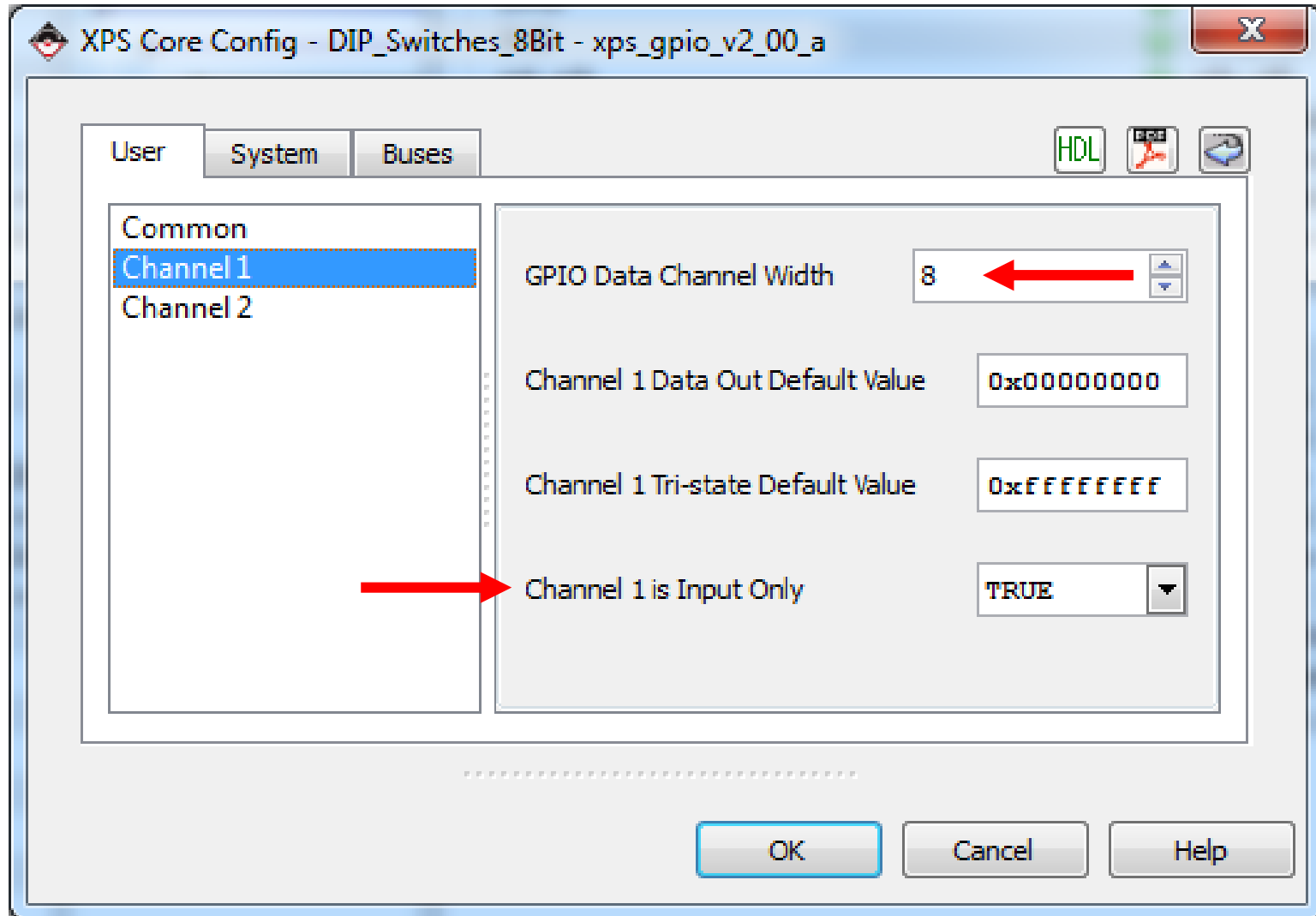
1. Toggle-On-Write (TOW) access toggles the status of the bit when a value of “1” is written to the corresponding bit

Parámetros de configuración

GPIO Parameters				
GPIO Channel1 Data Bus Width	C_GPIO_WIDTH	1-32	32	integer
GPIO Channel2 Data Bus Width	C_GPIO2_WIDTH	1-32	32	integer
XPS GPIO Interrupt	C_INTERRUPT_PRESENT	0 = Interrupt control module is not present 1 = Interrupt control module is present	0	integer
GPIO_DATA reset value	C_DOUT_DEFAULT	Any valid std_logic_vector	00000000	std_logic_vector
GPIO_TRI reset value	C_TRI_DEFAULT	Any valid std_logic_vector	FFFFFFFF	std_logic_vector
Use dual channel	C_IS_DUAL	0 = Single channel is enabled 1 = Both the channels are enabled	0	integer
GPIO2_DATA reset value	C_DOUT_DEFAULT_2	Any valid std_logic_vector	00000000	std_logic_vector
GPIO2_TRI reset value	C_TRI_DEFAULT_2	Any valid std_logic_vector	FFFFFFFF	std_logic_vector
Future Usage	C_ALL_INPUTS	0,1	0	Integer
Future Usage	C_ALL_INPUTS_2	0,1	0	Integer



Parámetros de configuración



Funciones disponibles

En driver de bajo nivel *xgpio_l.h* v.3.00a

Dependen de la versión del driver

➤ Escritura en registro:

- void *XGpio_WriteReg* (u32 **BaseAddress**, u32 **RegOffset**, u32 **Data**)

➤ Lectura en registro:

- u32 *XGpio_ReadReg* (u32 **BaseAddress**, u32 **RegOffset**)

Funciones generales de lectura/escritura en registros



Funciones disponibles

The screenshot displays the 'Bus Interfaces' window in Xilinx ISE. The 'Addresses' tab is active, showing a list of bus interfaces. The 'LED_7SEGMENT' instance is selected, and a context menu is open over it. The menu options are:

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

The 'View API Documentation' option is highlighted in blue. The 'Bus Interface Filter' panel on the right shows various filters, including 'By Connectic', 'By Bus Stand', and 'By Interface'.

Name	Bus Nam	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_if_cntlr	2.10.b
ilmb_cntlr		lmb_bram_if_cntlr	2.10.b
SRAM_256Kx32		xps_mch_emc	3.01.a
mdm_0		mdm	2.00.a
DIP_Switches_8Bit		xps_gpio	2.00.a
LED_7SEGMENT		xps_gpio	2.00.a
LEDs_8Bit			
SPLB	mb_...		
Push_Buttons_3Bit			
RS232			
clock_generator_0			
proc_sys_reset_0			

Funciones disponibles (cont.)

The screenshot displays the Xilinx Processor IP Library interface. On the left, a sidebar shows the 'Software Drivers' section with 'gpio v3_0' selected. The main content area is titled 'Xilinx Processor IP Library' and 'Software Drivers'. It features a navigation bar with links: 'Main Page', 'Data Structures', 'File List' (highlighted), and 'Globals'. Below this, the title 'Xilinx Driver gpio v3_00_a File List' is shown, followed by the text 'Here is a list of all documented files with brief descriptions:'. A table lists the files: xgpio.c, xgpio.h, xgpio_extra.c, xgpio_g.c, xgpio_i.h, xgpio_intr.c, xgpio_l.h (highlighted with a red arrow), xgpio_selftest.c, and xgpio_sinit.c. At the bottom, the copyright notice 'Copyright © 1995-2010 Xilinx, Inc. All rights reserved.' is visible.

er gpio v3_00_a: gpio v3_00_a]

bug Simulation Window Help

er gpio v3_00_a: File Index]

bug Simulation Window Help

Xilinx Processor IP Library

Software Drivers

gpio v3_0

This file contains the driver.

The Xilinx GPIO controller provides the following general features:

- Support for up to 100 pins
- Each of the drivers can be configured to support different pin counts
- Configurable support for different pin configurations

The driver provides an API to the user. Refer to the documentation for more information.

Xilinx Driver gpio v3_00_a File List

Here is a list of all documented files with brief descriptions:

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

Funciones disponibles (cont.)

The screenshot displays the Xilinx Processor ID Library software interface. The main window is titled "Xilinx Processor ID Library" and shows a list of registers and interrupt status definitions. A red box highlights the following register definitions:

```
#define XGPIO_DATA_OFFSET
#define XGPIO_TRI_OFFSET
#define XGPIO_DATA
#define XGPIO_TRI2
#define XGPIO_GIE
#define XGPIO_ISR
#define XGPIO_IER
```

Below the register definitions, the "Interrupt Status and" section shows bit definitions for the interrupt registers. A red box highlights the following interrupt status definitions:

```
#define XGPIO_IR_M
#define XGPIO_IR_C
#define XGPIO_IR_C
```

On the right side of the interface, the "Global Interrupt Enable Register bitmaps and masks" section shows bit definitions for the Global Interrupt Enable register. A red box highlights the following interrupt status definitions:

```
#define XGPIO_GIE_GINTR_ENABLE_MASK
```

The "Defines" section at the bottom right shows the following function definitions:

```
#define XGpio_WriteReg(BaseAddress, RegOffset, Data)
#define XGpio_ReadReg(BaseAddress, RegOffset)
```

Puerto serie *XPS UART Lite*

- Puerto serie **asincrónico** simple, full duplex
 - **Configurable**, no programable
- **Razón de baudios** configurable
- **5-8 bits** x caracter
- **Paridad** par, impar o **ninguna**
- **FIFOs** de Tx y Rx (**16 caracteres**)
- **Lógica de detección de errores**
 - **Paridad, sobrescritura y trama**
- **Posibilidad de generación de interrupción**
 - **Por Rx y por Tx**



XPS UART Lite (data sheet)



XPS UART Lite (v1.01a)

DS571 April 19, 2010

Product Specification

Introduction

The XPS Universal Asynchronous Receiver Transmitter (UART) Lite Interface connects to the PLB (Processor Local Bus) and provides the controller interface for asynchronous serial data transfer. This soft IP core is designed to interface with the PLBV46.

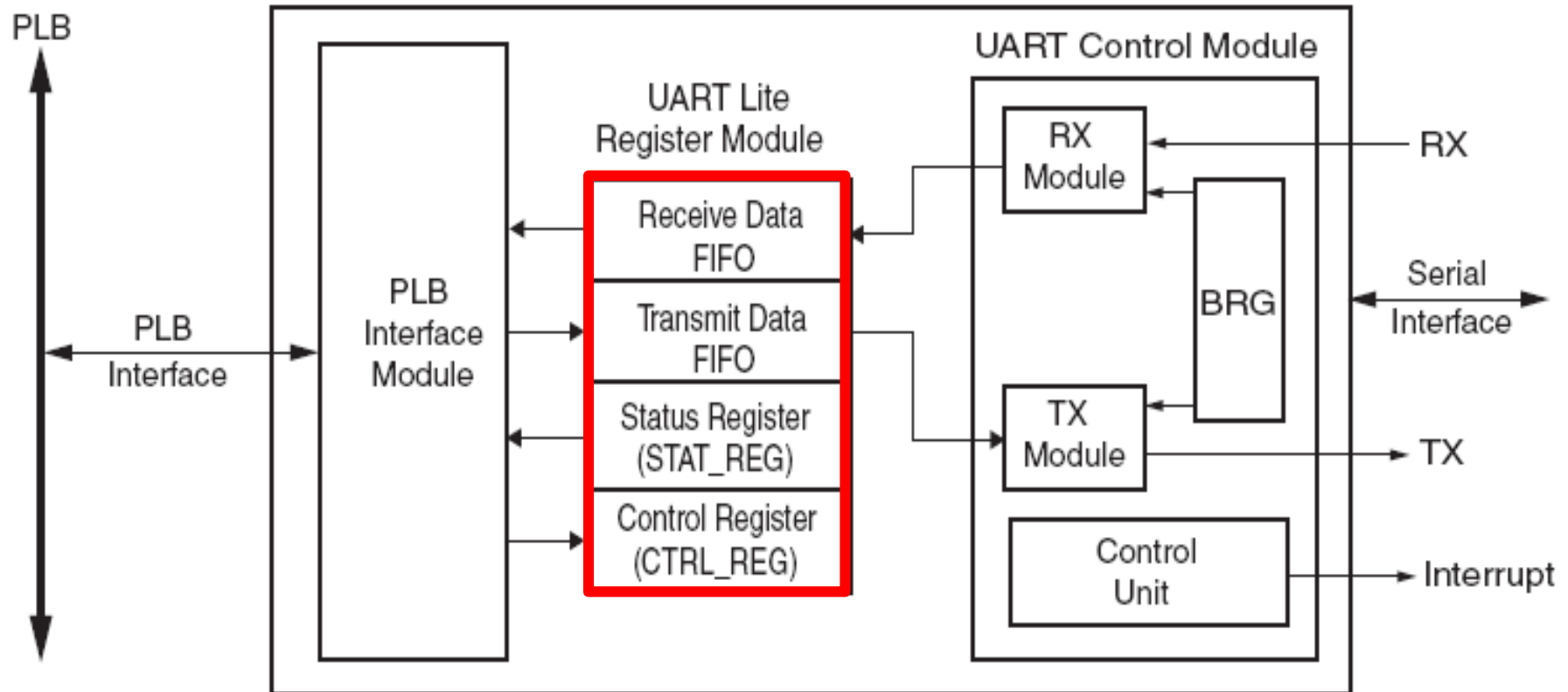
Features

- PLB interface is based on PLB v4.6 specification
- Supports 8-bit bus interfaces
- One transmit and one receive channel (full duplex)
- 16-character Transmit FIFO and 16-character Receive FIFO
- Configurable number of data bits in a character (5-8)
- Configurable parity bit (odd or even)
- Configurable baud rate

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	See EDK Supported Device Families .	
Version of Core	xps_uartlite	v1.01a
Resources Used		
	Min	Max
Slices	Refer to the Table 9 , Table 10 , Table 11 , Table 12 and Table 13	
LUTs		
FFs		
Block RAMs	N/A	



Estructura interna



Registros asociados

- **FIFO de Recepción**
- **FIFO de Transmisión**
- **Registro de Estado**
- **Registro de Control**

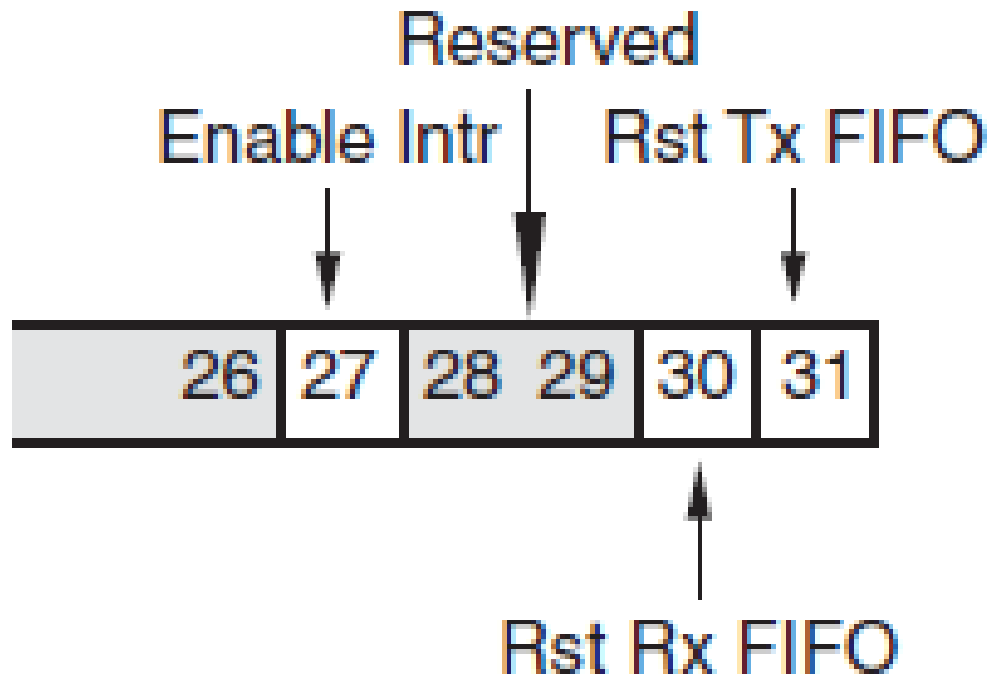
Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
C_BASEADDR + 0x0	Rx FIFO	Read ^[1]	0x0	Receive Data FIFO ^[3]
C_BASEADDR + 0x4	Tx FIFO	Write ^[2]	0x0	Transmit Data FIFO ^[3]
C_BASEADDR + 0x8	STAT_REG	Read ^[1]	0x4	UART Lite Status Register
C_BASEADDR + 0xC	CTRL_REG	Write ^[2]	0x0	UART Lite Control Register

Notes:

1. Writing of a read only register has no effect.
2. Reading of a write only register returns zero.
3. When system reset is applied both Rx FIFO and Tx FIFO are reset/cleared.

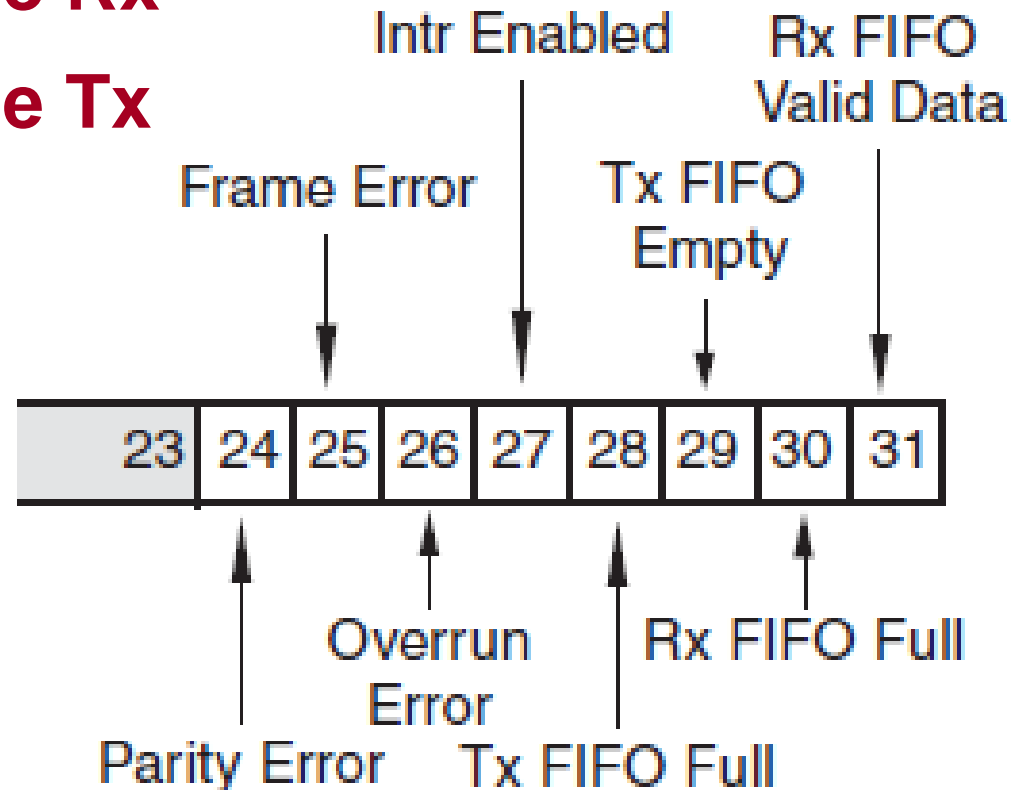
Registro de Control

- **Habilitación de Interrupción**
- **Limpiar FIFOs de Tx y de Rx**



Registro de Estado

- Estado de **habilitación de IT**
- Estado de **FIFO de Rx**
- Estado de **FIFO de Tx**
- **Errores**
 - **Paridad**
 - **Sobreescritura**
 - **Trama**



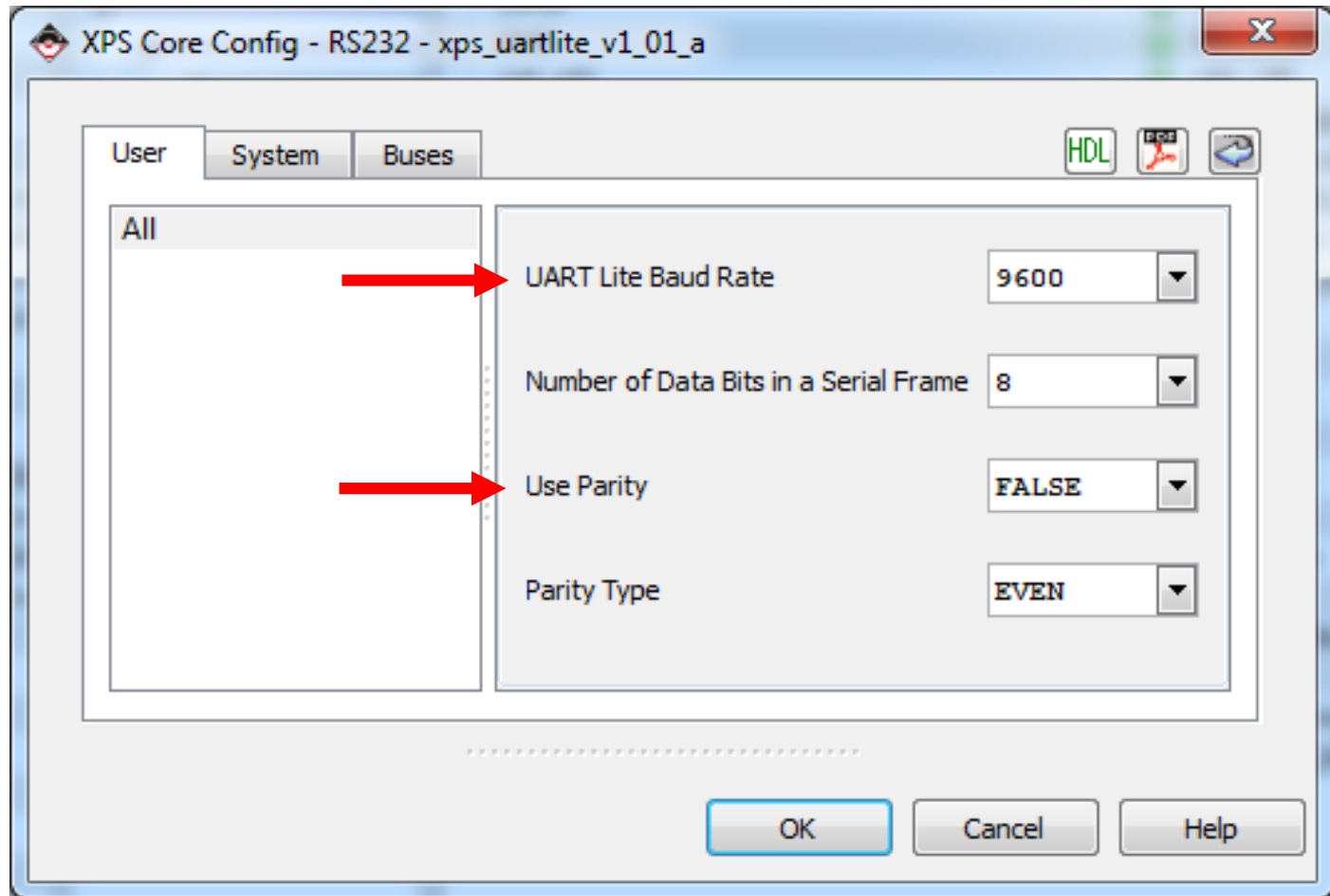
Parámetros de configuración

XPS UART Lite Design Parameters

Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameter				
Target FPGA family	C_FAMILY	See C_FAMILY parameter values .		string
System clock frequency (in Hz) driving the UART Lite peripheral	C_SPLB_CLK_FREQ_HZ	integer (ex. 100000000)	100_000_000	Integer
PLB Parameters				
PLB Base Address	C_BASEADDR	Valid Address ^[1]	None ^[3]	std_logic_vector
PLB High Address	C_HIGHADDR	Valid Address ^[2]	None ^[3]	std_logic_vector
PLB least significant address bus width	C_SPLB_AWIDTH	32	32	integer
PLB data width	C_SPLB_DWIDTH	32, 64, 128	32	integer
Baud rate of the UART Lite in bits per second	C_BAUDRATE	integer (ex. 128000)	9600 ^[5]	Integer
The number of data bits in the serial frame	C_DATA_BITS	5 - 8	8	Integer
Determines whether parity is used or not	C_USE_PARITY	0 = Do not use parity 1 = Use parity	1	Integer
If parity is used, determines whether parity is odd or even	C_ODD_PARITY	0 = Even parity 1 = Odd parity	1	Integer



Parámetros de configuración



Funciones disponibles

En driver de bajo nivel *xuartlite_l.h* v.2.00a

Dependen de la versión del driver

- Transmisión de un caracter:
 - void *XUartLite_SendByte* (u32 BaseAddress, u8 Data)
- Lectura de un carácter recibido :
 - u8 *XUartLite_RecvByte* (u32 BaseAddress)

Funciones generales de Tx/Rx
Esperan por disponibilidad



Funciones disponibles (cont.)

- **Escritura** en registro de **CONTROL**:
 - void *XUartLite_SetControlReg* (u32 **BaseAddress**, u32 **Mask**)
- **Lectura** en registro de **ESTADO**:
 - u32 *XUartLite_GetStatusReg* (u32 **BaseAddress**)
- **Información** de **ESTADO** (Boolean):
 - int *XUartLite_IsReceiveEmpty* (u32 **BaseAddress**)
 - int *XUartLite_IsTransmitFull* (u32 **BaseAddress**)



Funciones disponibles (cont.)

○ Asociadas a la INTERRUPCIÓN:

- int *XUartLite_IsIntrEnabled* (u32 **BaseAddress**)
- void *XUartLite_EnableIntr* (u32 **BaseAddress**)
- void *XUartLite_DisableIntr* (u32 **BaseAddress**)

Revisar documentación de driver
xuartlite_l.h v.2.00a



Temporizador *XPS Timer*

- Número de **timers** configurables (1 ó 2)
- Cada **timer** puede ser de hasta 32 bits (min 8)
- **Conteo ascendente o descendente**
- **Tres modos de operación:**
 - **Generación** (salida activa al desbordarse)
 - **Captura de eventos** (x activación de entrada)
 - **Modulación de ancho del pulso (PWM)**
 - Requiere de **ambos contadores**
- Posibilidad de generación de **interrupción**
 - Por **captura o desbordamiento**
- **NO** tiene funciones de **conteo** (externo)



XPS Timer (data sheet)



LogiCORE IP XPS Timer/Counter (v1.02a)

DS573 April 19, 2010

Product Specification

Introduction

This document describes the specifications for a XPS Timer/Counter core for the Processor Local Bus.

The XPS Timer/Counter is a 32-bit timer module that attaches to the PLB bus.

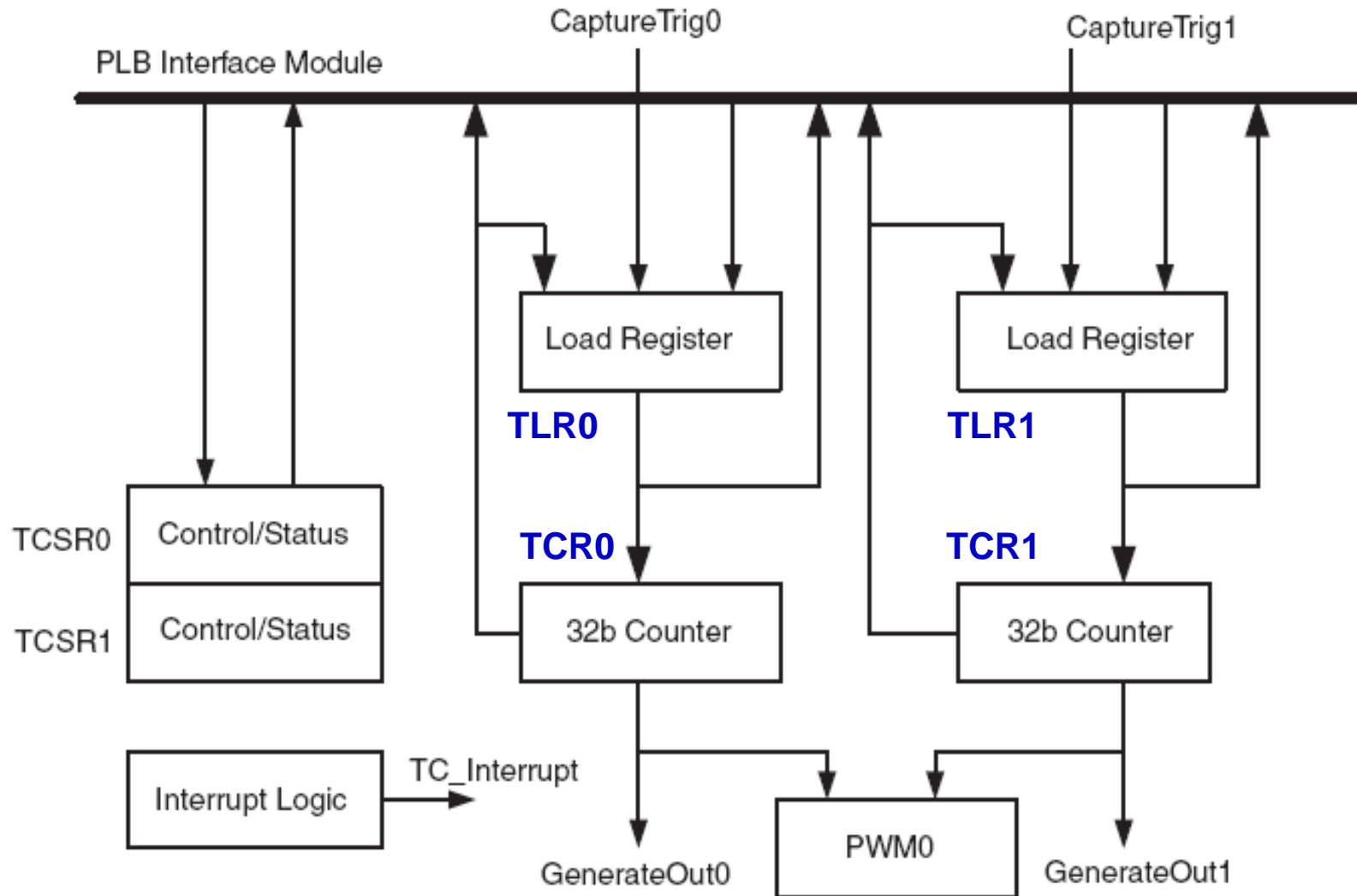
Features

- Connects as a 32-bit slave on PLB V4.6 buses of 32, 64 or 128 bits
- PLB interface with byte-enable support
- Two programmable interval timers with interrupt, event generation, and event capture capabilities
- Configurable counter width
- One Pulse Width Modulation (PWM) output
- Freeze input for halting counters during software debug

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Spartan®-6, Virtex®-6/-6CX, Spartan-3, Spartan-3A, Spartan-3E, Automotive Spartan-3/3E/3A/3A DSP, Spartan-3 ADSP, Virtex-4, QVirtex-4, QRVirtex-4, Virtex-5/5FX	
Version of Core	xps_timer	v1.02a
Resources Used		
	Min	Max
Slices	Refer to the Table 9 , Table 10 , Table 11 , Table 12 , and Table 13 .	
LUTs		
FFs		
Block RAMs	N/A	
Provided with Core		
Documentation	Product Specification	



Estructura interna



Registros asociados

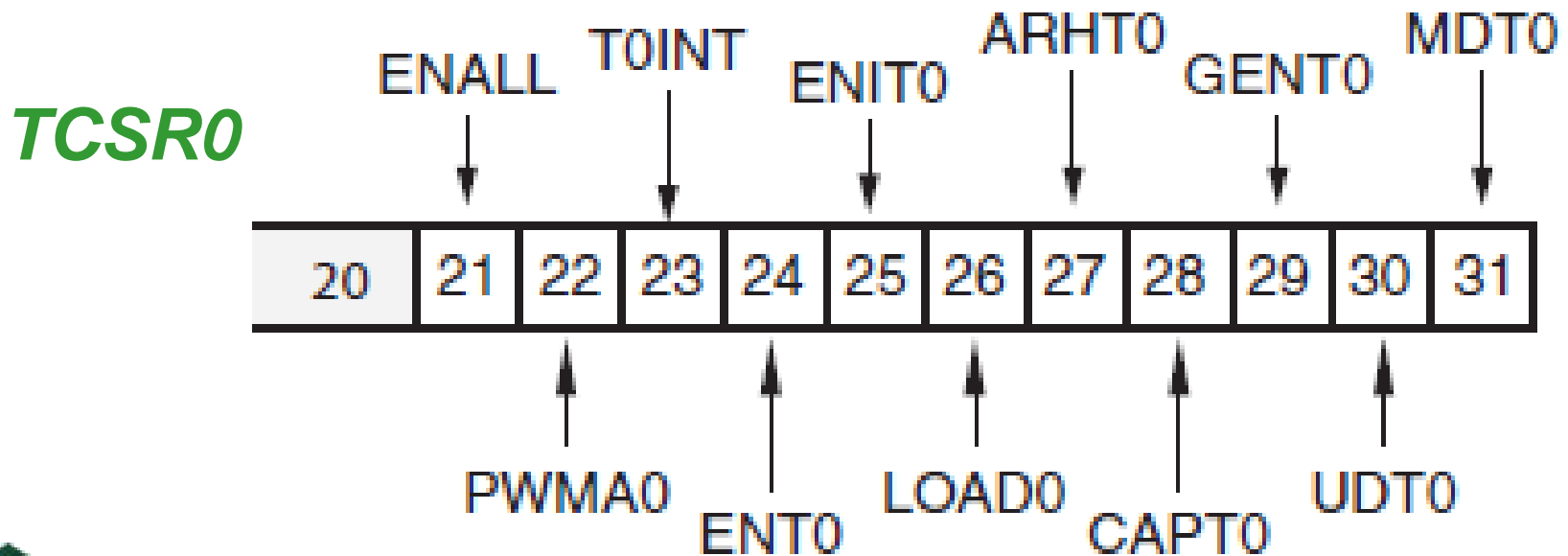
Por cada canal...

- Registro de **Conteo** (*TCRx*)
- Registro de **Carga** (*TLRx*)
- Registro de **Control y Estado** (*TCSRx*)

Register	Address (Hex)	Size	Type	Description
TCSR0	C_BASEADDR + 0x00	Word	Read/Write	Control/Status Register 0
TLR0	C_BASEADDR + 0x04	Word	Read/Write	Load Register 0
TCR0	C_BASEADDR + 0x08	Word	Read	Timer/Counter Register 0
TCSR1	C_BASEADDR + 0x10	Word	Read/Write	Control/Status Register 1
TLR1	C_BASEADDR + 0x14	Word	Read/Write	Load Register 1
TCR1	C_BASEADDR + 0x18	Word	Read	Timer/Counter Register 1

Registro de Control y Estado

- Modo del Timer
- Habilidadación de Timer(s)
- Habilidadación de Interrupción
- Sentido de conteo (asc/desc)
- Autorecarga, etc.

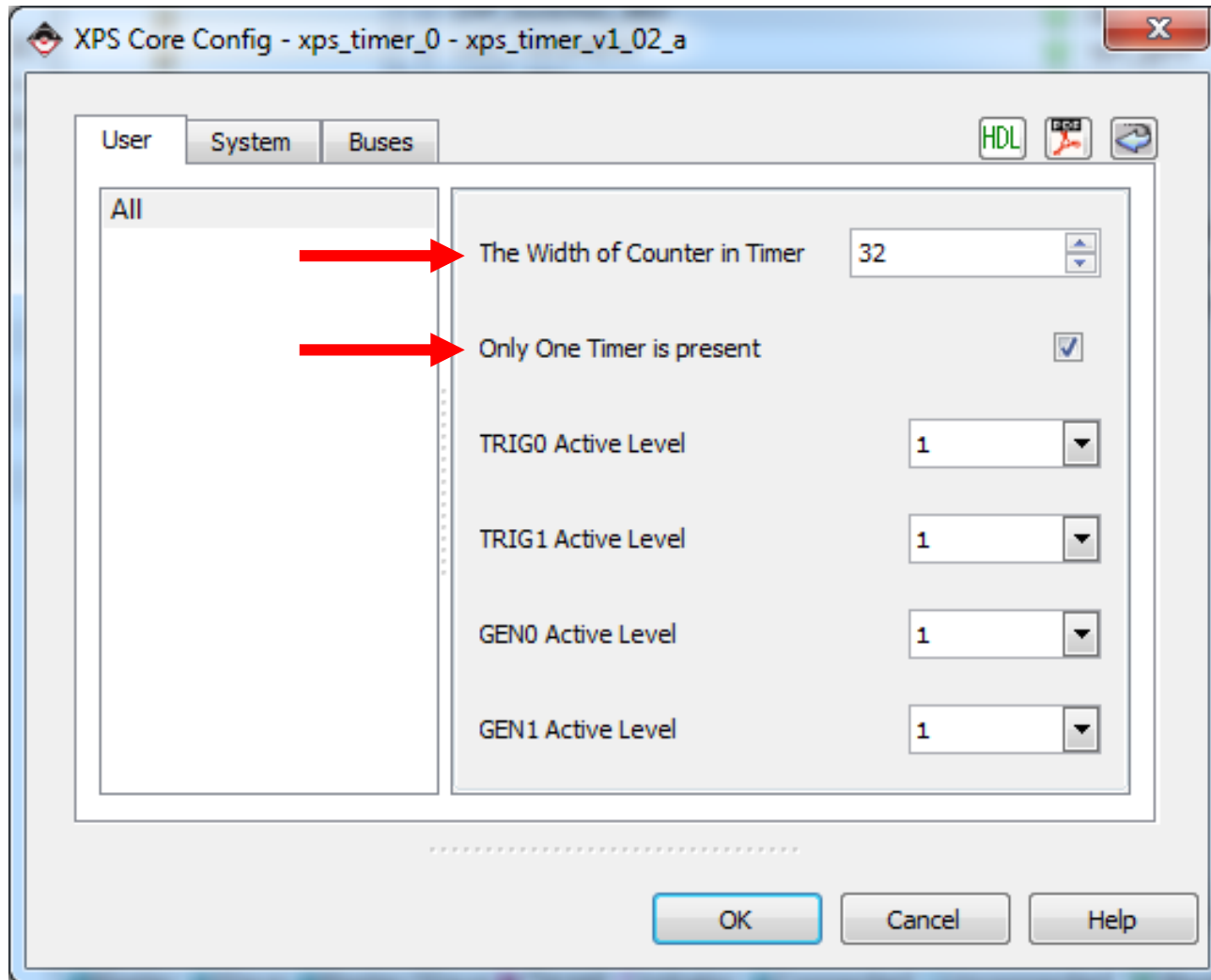


Parámetros de configuración

XPS Timer/Counter Parameter				
The width in bits of the counters in the XPS Timer/Counter	C_COUNT_WIDTH	8 - 32	32	integer
Number of Timer modules	C_ONE_TIMER_ONLY	0 = Two timers are present 1 = One timer is present (No PWM mode)	0	integer
Assertion level for CaptureTrig0	C_TRIG0_ASSERT	'0' = CaptureTrig0 input is low-true '1' = CaptureTrig0 input is high-true	1	std_logic
Assertion level for CaptureTrig1	C_TRIG1_ASSERT	'0' = CaptureTrig1 input is low-true '1' = CaptureTrig1 input is high-true	1	std_logic
Assertion level for GenerateOut0	C_GEN0_ASSERT	'0' = GenerateOut0 output is low-true '1' = GenerateOut0 output is high-true	1	std_logic
Assertion level for GenerateOut1	C_GEN1_ASSERT	'0' = GenerateOut1 output is low-true '1' = GenerateOut1 output is high-true	1	std_logic



Parámetros de configuración



Funciones disponibles

En driver de bajo nivel *xtmrctr_l.h* v.2.02a

Dependen de la versión del driver

- **Lectura de un registro:**
 - **u32** *XTmrCtr_ReadReg* (**u32** BaseAddress, **u8** TimerNumber, unsigned RegOffset)
- **Escritura en un registro:**
 - **void** *XTmrCtr_WriteReg* (**u32** BaseAddress, **u8** TimerNumber, unsigned RegOffset, **u32** ValueToWrite)

Funciones generales de lectura/escritura en registros



Funciones disponibles (cont.)

Asociadas al Registro de Carga:

- **u32** *XTmrCtr_GetLoadReg* (**u32** BaseAddress, **u8** TmrCtrNumber)
- **void** *XTmrCtr_SetLoadReg* (**u32** BaseAddress, **u8** TmrCtrNumber, **u32** RegisterValue)

Asociadas al Registro de Conteo:

- **u32** *XTmrCtr_GetTimerCounterReg* (**u32** BaseAddress, **u8** TmrCtrNumber)
- **void** *XTmrCtr_LoadTimerCounterReg* (**u32** BaseAddress, **u8** TmrCtrNumber)



Funciones disponibles (cont.)

Asociadas al Registro de Control y Estado:

- **u32** *XTmrCtr_GetControlStatusReg* (**u32** BaseAddress, **u8** TmrCtrNumber)
- **void** *XTmrCtr_SetControlStatusReg* (**u32** BaseAddress, **u8** TmrCtrNumber, **u32** RegisterValue)
- **void** *XTmrCtr_Enable* (**u32** BaseAddress, **u8** TmrCtrNumber)
- **void** *XTmrCtr_Disable* (**u32** BaseAddress, **u8** TmrCtrNumber)
- **void** *XTmrCtr_EnableIntr* (**u32** BaseAddress, **u8** TmrCtrNumber)
- **void** *XTmrCtr_DisableIntr* (**u32** BaseAddress, **u8** TmrCtrNumber)
- **int** *XTmrCtr_HasEventOccurred* (**u32** BaseAddress, **u8** TmrCtrNumber)

Máscaras del registro *TCSR*

<i>Función</i>	<i>Opciones</i>	<i>XTC_CSR_Máscara_MASK</i>
Modo de trabajo	"1": Captura	CAPTURE_MODE
Conteo up/down	"1": conteo descendente	DOWN_COUNT
Habilitar señal de salida	"1": habilita GenerateOut	EXT_GENERATE
Habilitar entrada de captura	"1": habilita CaptureTrig	EXT_CAPTURE
Autorecarga o retención	"1": Autorecarga	AUTO_RELOAD
Cargar el timer	"1": TLR ---> TCR	LOAD
Habilitar (o no) interrupción	"1": Habilita interrupción	ENABLE_INT
Habilitar el timer	"1": Habilita el timer	ENABLE_TMR
Bandera de solicitud de IT	"1": Solicitud de IT	INT_OCCURED
Modo PWM	"1": Establece modo PWM	ENABLE_PWM
Habilitación general	"1": Habilita el XPS Timer	ENABLE_ALL



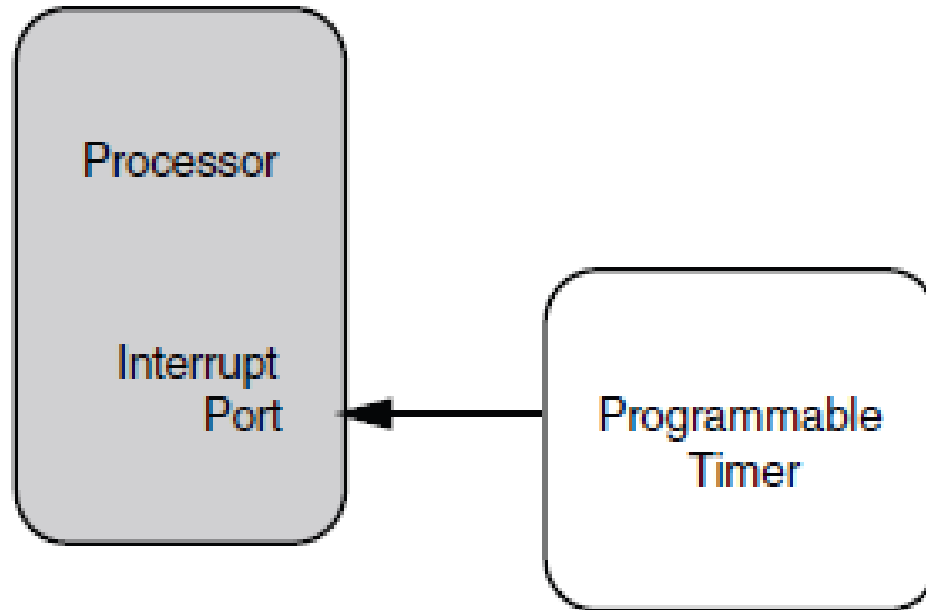
Interrupción en MicroBlaze (1)

- Una entrada (puerto) de solicitud de IT
 - Interrupt
 - Configurable x flanco o x nivel
- Enmascarable
 - *microblaze_enable_interrupts();*
 - *microblaze_disable_interrupts();*
- Puede conectarse directamente a la solicitud de IT de UN periférico o a la salida de un Controlador de IT (IntC)
- No implementa stack x HW ni tiene registro SP

Funciones
incluidas en
standalone

(1) Características adicionales a partir de v. 8.30

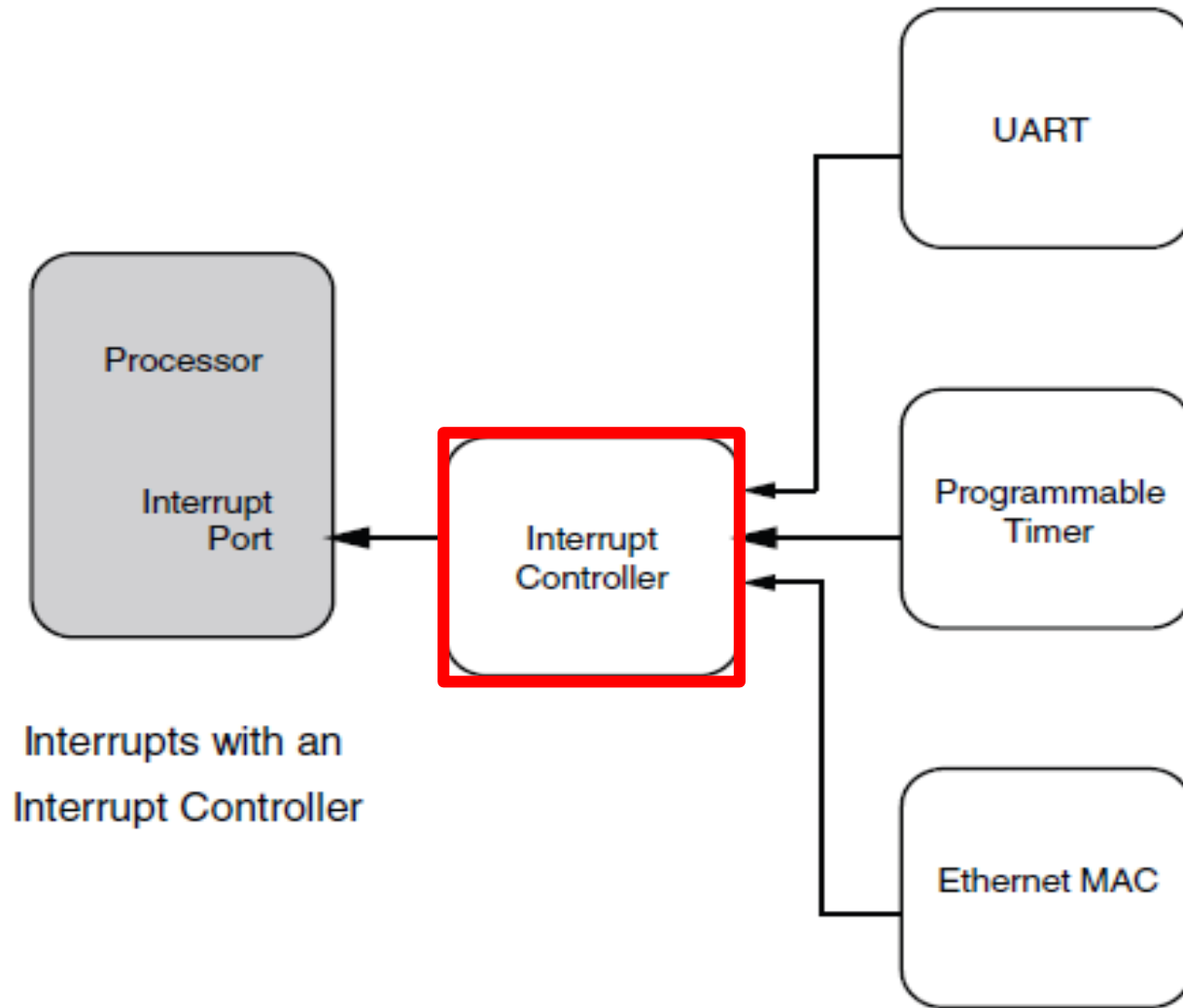
Configuraciones de IT



Interrupts without an
Interrupt Controller



Configuraciones de IT (cont.)



Mecanismo de IT en MicroBlaze

1. Finaliza la instrucción en curso
2. Salva la dirección de la siguiente instrucción en R14
3. Inhabilita la IT
4. Carga PC con la dirección 0x10
 - Primera instrucción del manipulador de IT de bajo nivel de la plataforma SW (o del SO)
 - Usualmente un salto a otra dirección
5. Se ejecuta el manipulador de IT principal de la plataforma SW (o del SO)...

Mecanismo de IT en MicroBlaze (cont.)

6. El manipulador de IT principal de la plataforma SW (o del SO)...
 - Salva todos los registros de MB en el stack de la aplicación
 - Este stack lo maneja la plataforma SW
 - Transfiere el control al siguiente nivel del manipulador de IT de la plataforma SW
7. Este nivel puede ser: la rutina de IT de la aplicación o el manipulador de IT del controlador de IT

Mecanismo de IT en MicroBlaze (cont.)

8. El manipulador de IT principal de la plataforma SW (o del SO)...
 - Consulta una tabla de vectores de IT de MB para determinar el siguiente nivel
 - La tabla de vectores de IT de MB es creada por la plataforma SW
9. Si NO existe controlador de IT, el siguiente nivel es el manipulador de IT (rutina de IT) de la aplicación

Mecanismo de IT en MicroBlaze (cont.)

10. Necesidad de colocar la dirección de esta rutina en la tabla de vectores de IT

- *void microblaze_register_handler (XInterruptHandler Handler, void *DataPtr)*

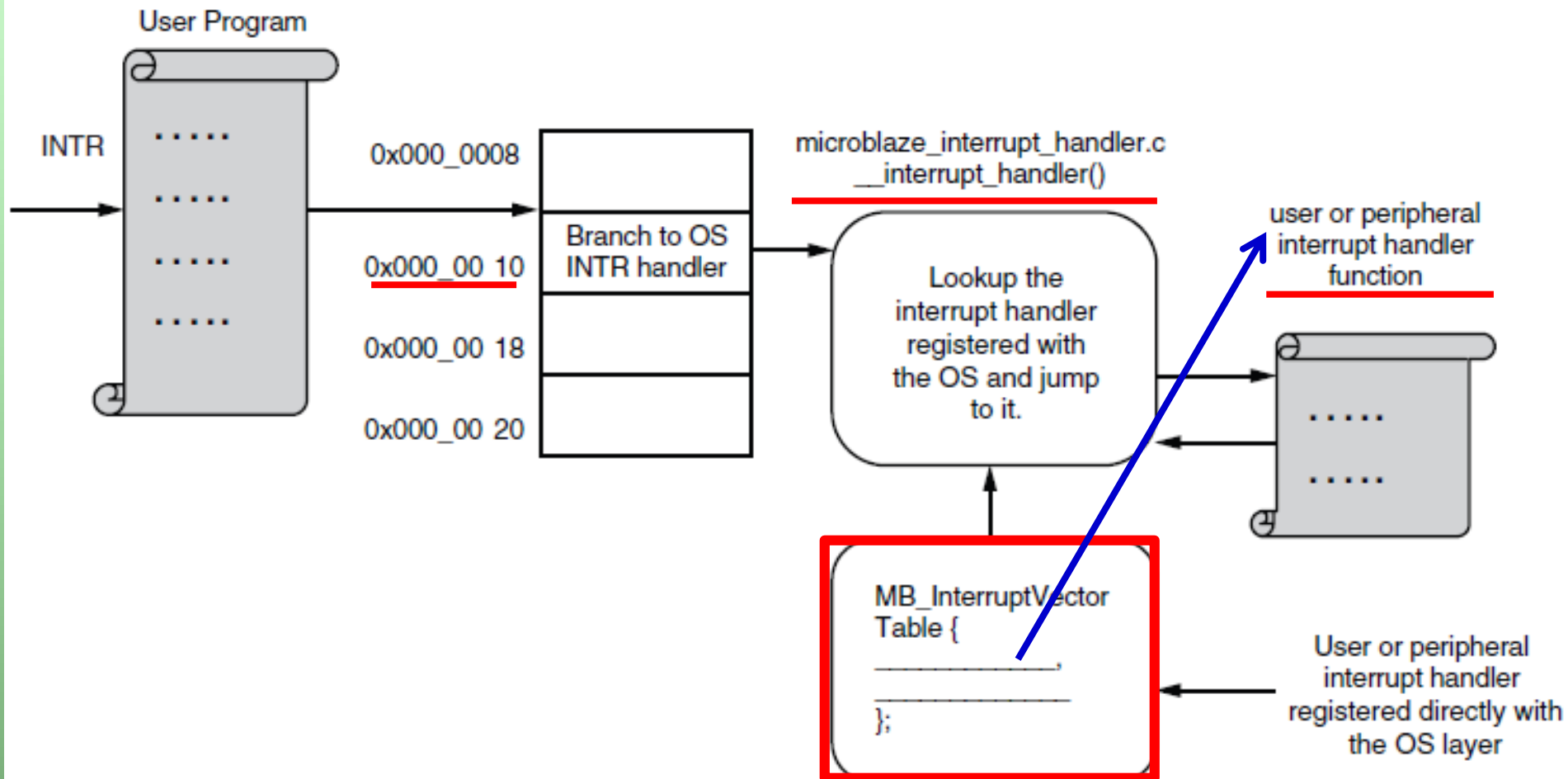
Función
incluida en
standalone

11. La rutina de IT tiene que desmarcar la solicitud de IT en el periférico

12. Al finalizar la rutina de IT, se transfiere el control al manipulador de IT principal de la plataforma SW

- Restaura los registros de MB
- Restaura PC (dirección de la siguiente instrucción)
- Habilita la IT en MB

Flujo de IT en MicroBlaze sin IntC



Mecanismo de IT en MicroBlaze (cont.)

9. Si existe controlador de IT (IntC), el siguiente nivel es el manipulador de IT del IntC
10. Encuesta las entradas de solicitud de IT activas y habilitadas en el IntC
11. Por cada IT activa, accede en orden de prioridad a la Tabla de Vectores de IT del IntC para ejecutar la rutina de IT del periférico
12. La Tabla de Vectores de IT del IntC es creada por el driver del IntC



Mecanismo de IT en MicroBlaze (cont.)

13. Necesidad de colocar la dirección de esta rutina en la tabla de vectores de IT del IntC

- *void XIntc_RegisterHandler (.....)*

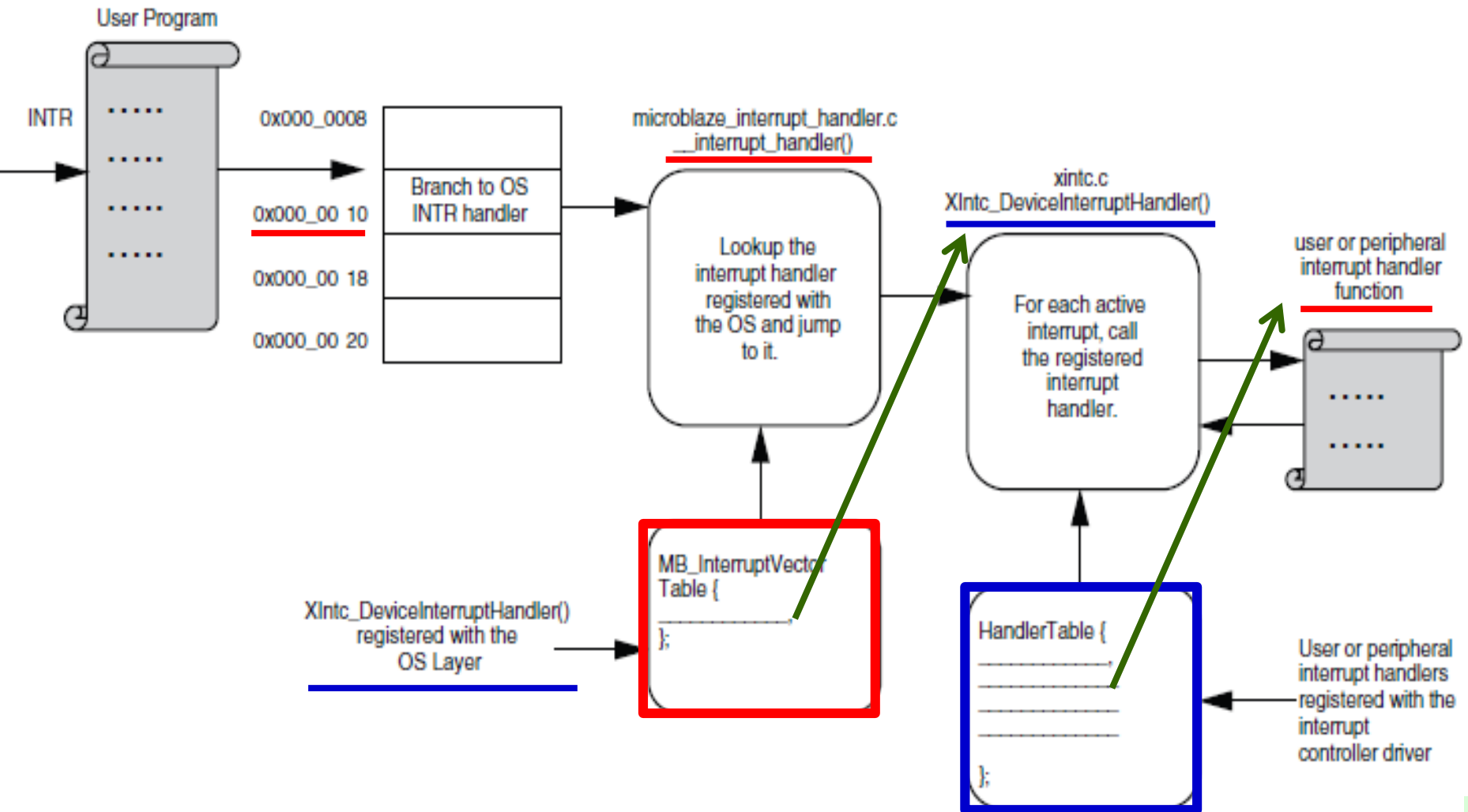
14. La rutina de IT tiene que desmarcar la solicitud de IT en el periférico

Función
incluida en
xintc_1.h

15. Al finalizar la rutina de IT, se transfiere el control al manipulador de IT del IntC, y de este al manipulador de IT principal de la plataforma SW

- Restaura los registros de MB
- Restaura PC (dirección de la siguiente instrucción)
- Habilita la IT en MB

Flujo de IT en MicroBlaze con IntC



Controlador de Interrupciones

XPS INTC

- Número de entradas configurable (1 a 32)
- Lógica de prioridades fijas (0: más prioritaria)
- Entradas de IT enmascarables
 - Registro de habilitación (máscara)
- Solicitud por frente o por nivel
 - Independiente para cada entrada
- Posibilidad de conexión en cascada



XPS INTC (data sheet)



LogiCORE IP XPS Interrupt Controller (v2.01a)

DS572 April 19, 2010

Product Specification

Introduction

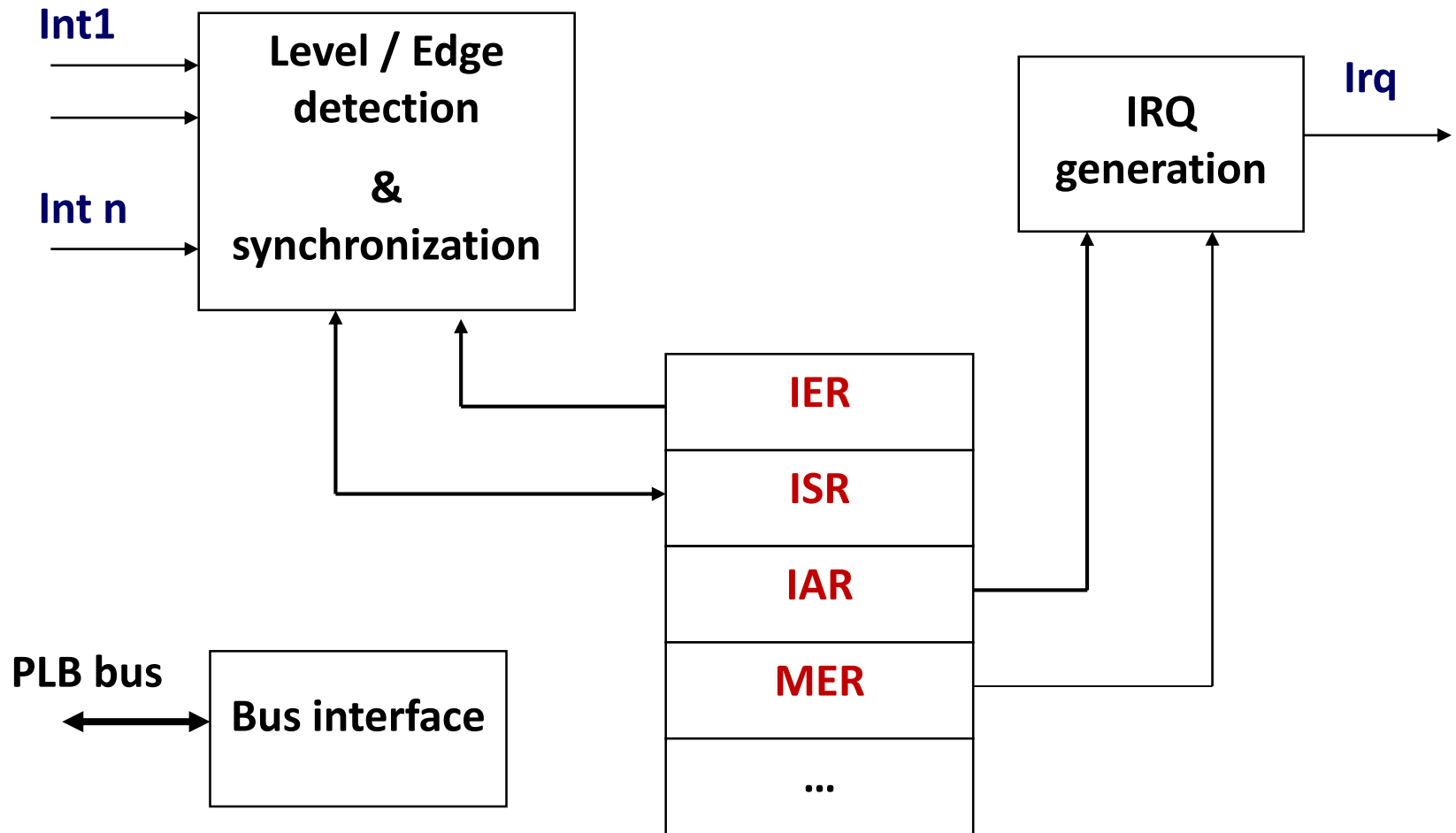
The LogiCORE IP XPS Interrupt Controller (XPS INTC) concentrates multiple interrupt inputs from peripheral devices to a single interrupt output to the system processor. The registers for checking, enabling and acknowledging interrupts are accessed through a slave interface for the Processor Local Bus (PLB V4.6). The number of interrupts and other aspects can be tailored to the target system.

Features

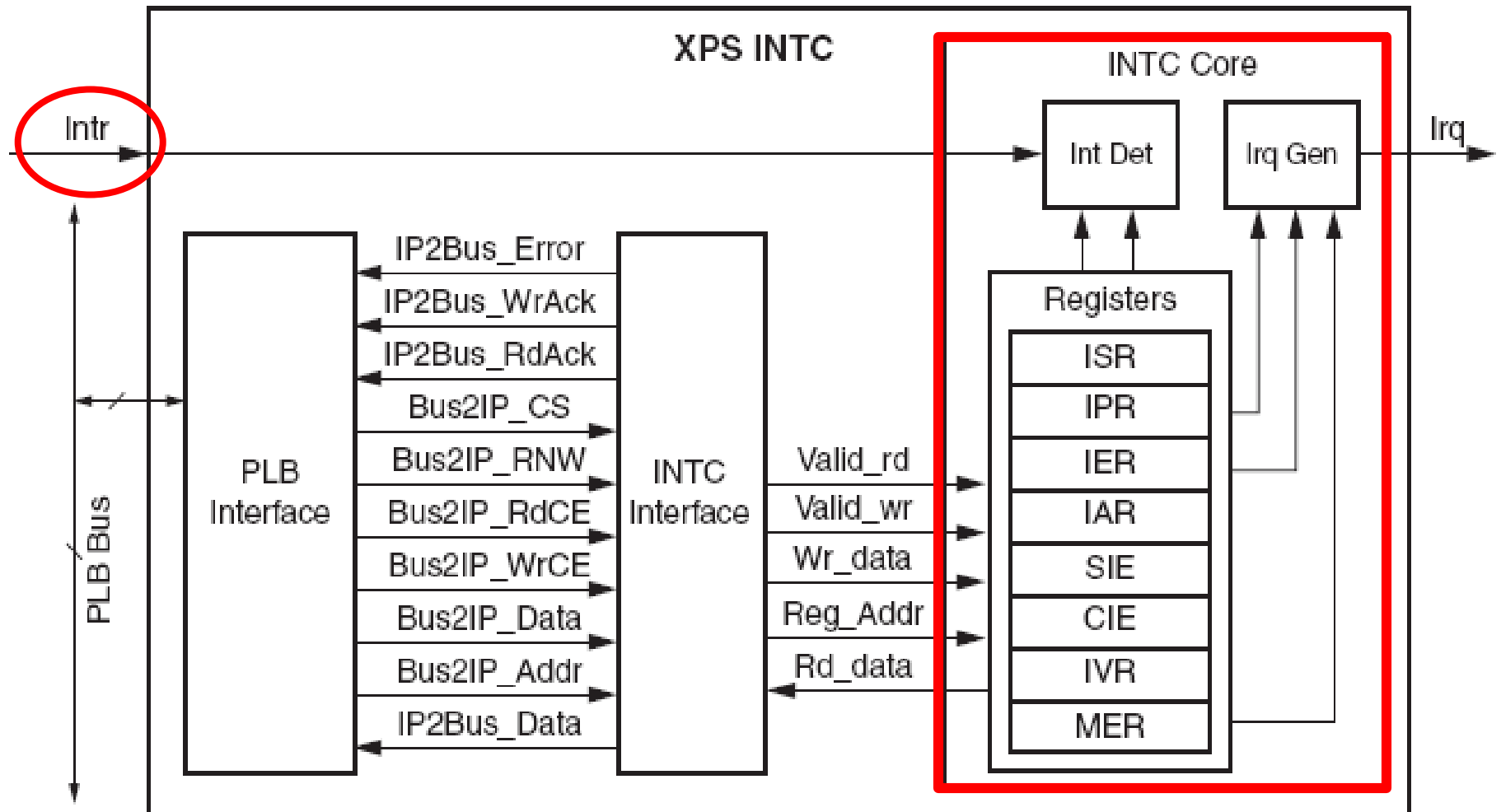
- Connects as a 32-bit slave on PLB V4.6 bus of 32, 64 and 128-bit data width
- Configurable number of (up to 32) interrupt inputs
- Single interrupt output
- Easily cascaded to provide additional interrupt inputs
- Priority between interrupt requests is determined by vector position. The least significant bit (LSB, in this case bit 0) has the highest priority
- Interrupt Enable Register for selectively disabling individual interrupt inputs
- Master Enable Register for disabling interrupt request output

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Spartan®-6, Virtex®-6/6CX, Spartan-3, Spartan-3A, Spartan-3E, Automotive Spartan-3/3E/3A/3A DSP, Spartan-3 ADSP, Virtex-4, QVirtex-4, QRVirtex-4, Virtex-5/5FX	
Version of Core	xps_intc	v2.01a
Resources Used		
	Min	Max
Slices	See Table 13 , Table 14 , Table 15 , Table 16 , Table 17 , and Table 18	
FFs		
LUTs		
Block RAMs	N/A	
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs & Application Notes	N/A	

Estructura interna simplificada



Estructura interna



Note:

Intc Interface: Design module does not exist by this name. This interface is part of top-level XPS INTC.

Parámetros de configuración

INTC Parameters				
Number of interrupt inputs	C_NUM_INTR_INPUTS	1 - C_SPLB_NATIVE_DWIDTH	2	integer
Type of interrupt for each input 1 = Edge 0 = Level	C_KIND_OF_INTR	See ⁽⁷⁾	ALL 1's	std_logic_vector
Type of each edge sensitive input 1 = Rising 0 = Falling Valid if C_KIND_OF_INTR = 1's	C_KIND_OF_EDGE	See ⁽⁷⁾	ALL 1's	std_logic_vector
Type of each level sensitive input 1 = High 0 = Low Valid if C_KIND_OF_INTR = 0's	C_KIND_OF_LVL	See ⁽⁷⁾	ALL 1's	std_logic_vector
Indicates the presence of IPR	C_HAS_IPR	0 = Not Present 1 = Present	1	integer
Indicates the presence of SIE	C_HAS_SIE	0 = Not Present 1 = Present	1	integer

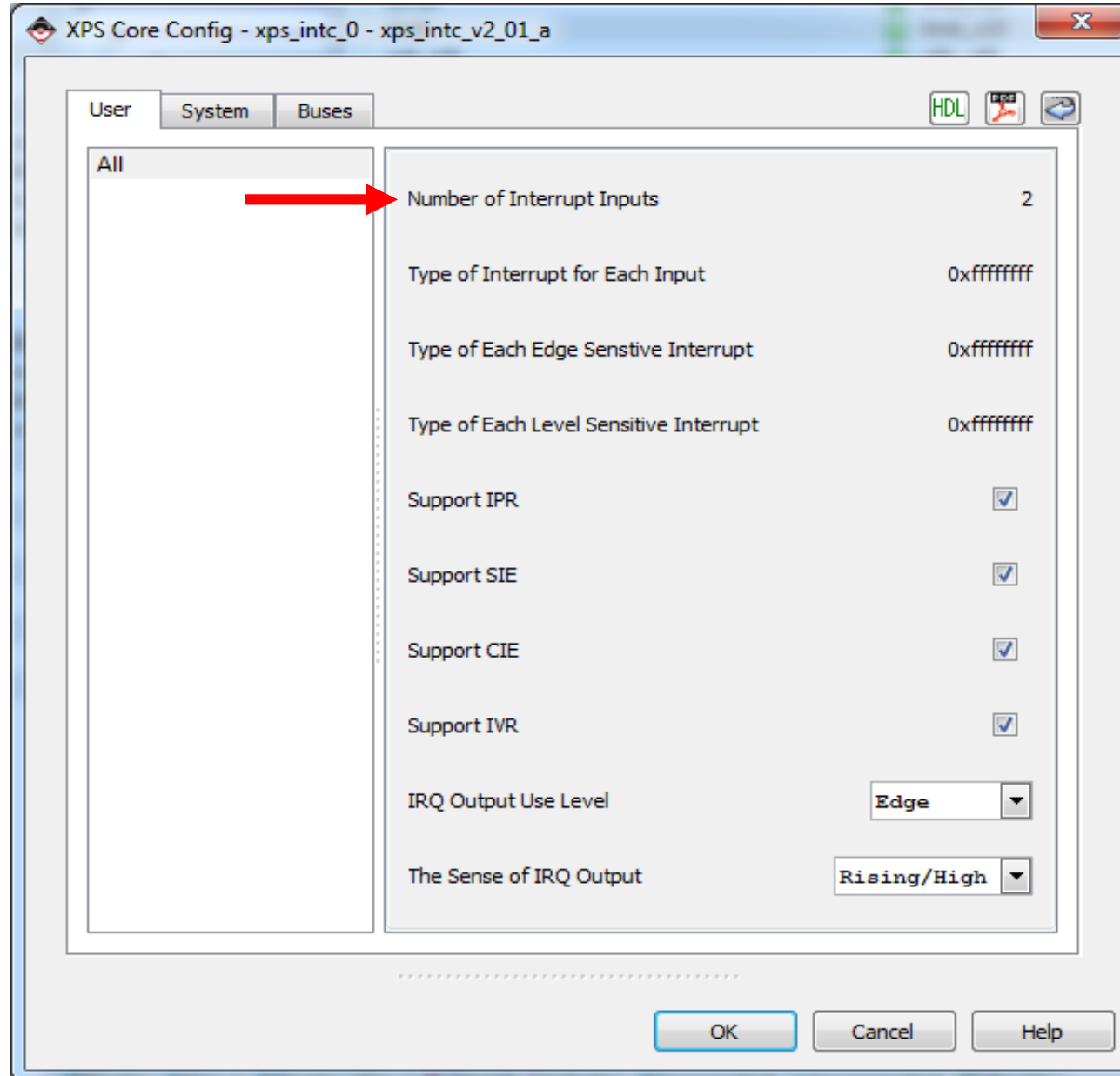


Parámetros de configuración (cont.)

Indicates the presence of CIE	C_HAS_CIE	0 = Not Present 1 = Present	1	integer
Indicates the presence of IVR	C_HAS_IVR	0 = Not Present 1 = Present	1	integer
Indicates level or edge active Irq	C_IRQ_IS_LEVEL	0 = Active Edge 1 = Active Level	1	integer
Indicates the sense of the Irq output	C_IRQ_ACTIVE	0 = Falling / Low 1 = Rising / High	1	std_logic



Parámetros de configuración



Registros asociados

- Registro de **Habilitación** (*IER*)
- Registro de **Habilitación Maestro** (*MER*), 2 bit
 - Sirve también para habilitación de IT x SW (bit **HIE**)
- Registro de **Estado** (*ISR*)
- Registro de **Reconocimiento de IT** (*IAR*)
 - Para limpiar la solicitud de IT
- Registros opcionales: *SIE, CIE, IVR, IPR*



Registros asociados

Register Name	Abbreviation	OPB Offset
Interrupt Status Register	ISR	0 (00h)
Interrupt Pending Register	IPR	4 (04h)
Interrupt Enable Register	IER	8 (08h)
Interrupt Acknowledge Register	IAR	12 (0Ch)
Set Interrupt Enable Bits	SIE	16 (10h)
Clear Interrupt Enable Bits	CIE	20 (14h)
Interrupt Vector Register	IVR	24 (18h)
Master Enable Register	MER	28 (1Ch)



Registros asociados

Register Name	Base Address + Offset (Hex)	Access Type	Abbreviation	Reset Value
Interrupt Status Register	C_BASEADDR + 0x0	Read / Write	ISR	All Zeros
Interrupt Pending Register	C_BASEADDR + 0x4	Read	IPR	All Zeros
Interrupt Enable Register	C_BASEADDR + 0x8	Read / Write	IER	All Zeros
Interrupt Acknowledge Register	C_BASEADDR + 0xC	Write	IAR	All Zeros
Set Interrupt Enable Bits	C_BASEADDR + 0x10	Write	SIE	All Zeros
Clear Interrupt Enable Bits	C_BASEADDR + 0x14	Write	CIE	All Zeros
Interrupt Vector Register	C_BASEADDR + 0x18	Read	IVR	All Ones
Master Enable Register	C_BASEADDR + 0x1C	Read / Write	MER	All Zeros



Master Enable Register (MER)

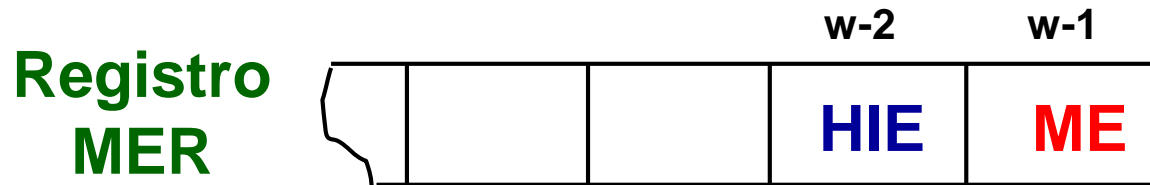
- Registro de dos bit: **ME** y **HIE**
- Habilitación general del XIntC (**ME**)

`void XIntc_MasterEnable (u32 BaseAddress)`

`void XIntc_MasterDisable (u32 BaseAddress)`

- Posibilidad de IT x SW (**HIE** = 0)
- Una vez seteado **HIE**, IT x HW (hasta **RESET**)

`void XIntc_SetIntrSvcOption (u32 BaseAddress, int Option)`



Funciones disponibles

En driver de bajo nivel *xintc_l.h* v2.02a

Dependen de la versión del driver

void XIntc_MasterEnable (**u32** BaseAddress)

void XIntc_MasterDisable (**u32** BaseAddress)

void XIntc_EnableIntr (**u32** BaseAddress, **u32**
EnableMask)

void XIntc_DisableIntr (**u32** BaseAddress, **u32**
DisableMask)

void XIntc_AckIntr (**u32** BaseAddress, **u32**
AckMask)

u32 XIntc_GetIntrStatus (**u32** BaseAddress)

Funciones disponibles (cont.)

En driver de bajo nivel *xintc_l.h* v2.02a

Dependen de la versión del driver

```
void XIntc_RegisterHandler (u32 BaseAddress,  
    int InterruptId, XInterruptHandler Handler, void  
    *CallbackRef)
```

```
void XIntc_SetIntrSvcOption (u32 BaseAddress,  
    int Option)
```

```
void XIntc_DeviceInterruptHandler (void  
    *Deviceld)
```



XIntc_DeviceInterruptHandler (..)

This function is the **primary interrupt handler** for the driver. It must be connected to the interrupt source such that is called when an interrupt of the interrupt controller is active. It will resolve which interrupts are active and enabled and call the appropriate interrupt handler. It uses the AckBeforeService flag in the configuration data to determine when to acknowledge the interrupt. Highest priority interrupts are serviced first. The driver can be configured to service only the highest priority interrupt or all pending interrupts using the {XIntc_SetOptions()} function or the {XIntc_SetIntrSrvOption()} function.

- This function assumes that an **interrupt vector table** has been previously initialized. It does not verify that entries in the table are valid before calling an interrupt handler.
- **Parameters:** *DeviceId* is the zero-based device ID defined in *xparameters.h* of the interrupting interrupt controller. It is used as a direct index into the configuration data, which contains the **vector table for the interrupt controller**. Note that even though the argument is a void pointer, the value is not a pointer but the actual device ID. The void pointer type is necessary to meet the XInterruptHandler typedef for interrupt handlers.

Secuencia de programación de IT

- **Habilitar la salida de solicitud de IT del periférico**
- **Habilitar la entrada correspondiente a dicho dispositivo en el controlador de IT**
- **Habilitar la salida de solicitud de IT del controlador de IT**
- **Habilitar la entrada de IT de MicroBlaze**

- *microblaze_enable_interrupts();*
- *microblaze_disable_interrupts();*

Incluidas
en
standalone



Enlace con la subrutina de IT

- [Las **rutinas de IT** para cada uno de los periféricos conectados al controlador de IT pueden especificarse en el **fichero MSS**]
 - *PARAMETER int_handler =*
<nombre_de_la_rutina>, int_port = Interrupt
- **LibGen** crea automáticamente una **tabla de vectores** a las diferentes rutinas de IT ordenados según las prioridades correspondientes.

Obsoleto. NO soportado a partir de v. 13

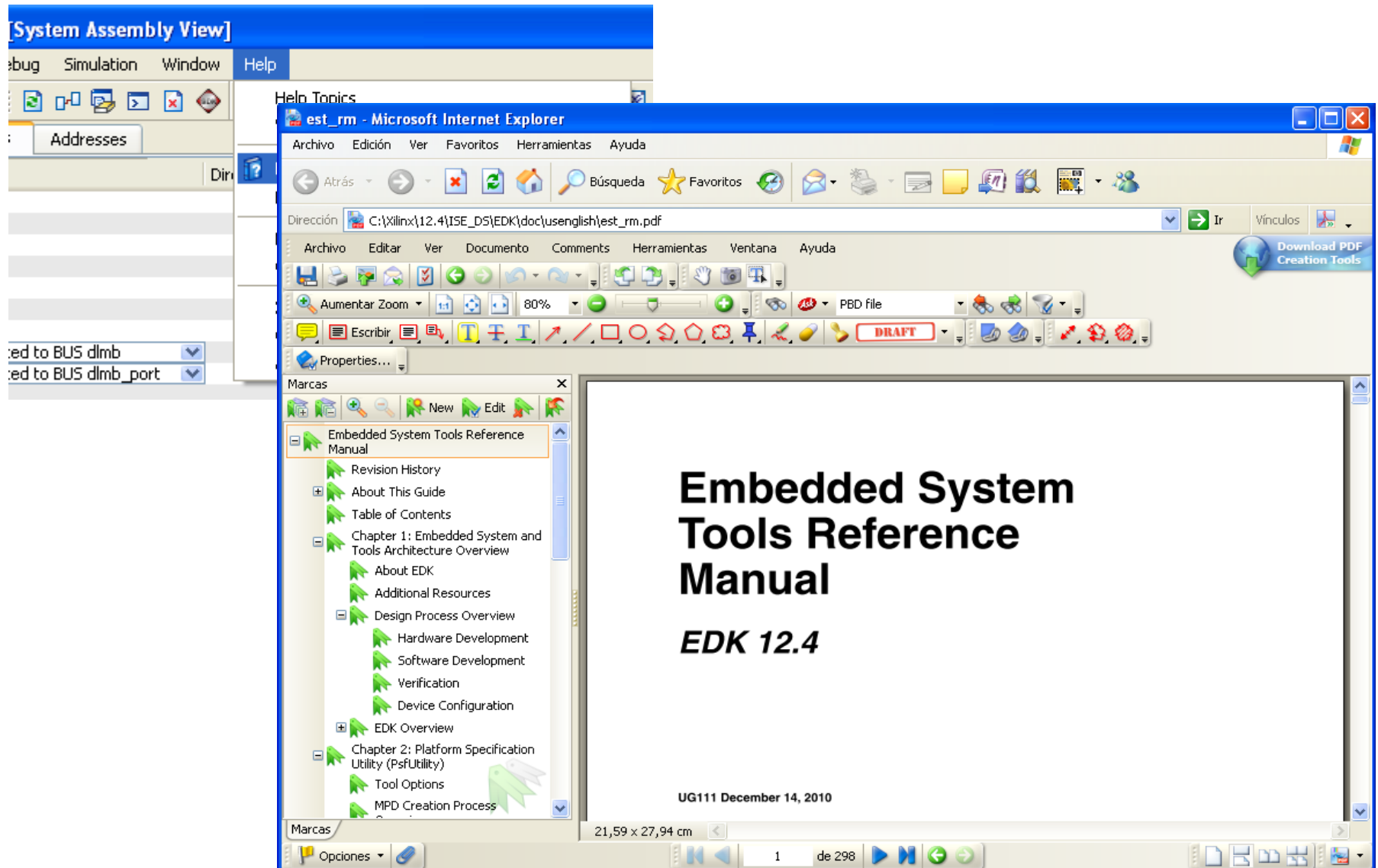


Enlace con la subrutina de IT (cont.)

- Los vectores de IT también pueden ser ubicados dinámicamente en la **tabla de vectores del controlador de IT** utilizando la función:
 - void *XIntc_RegisterHandler* (u32 **BaseAddress**, int **InterruptId**, XInterruptHandler **Handler**, void * **CallBackRef**)
 - Ejemplo: *XIntc_RegisterHandler* (**XPAR_XPS_INTC_0_BASEADDR**, **XPAR_XPS_INTC_0_TIMER_0_INTERRUPT_INTR**, (XInterruptHandler) **Timer0_interrupt_handler**, NULL);
- Revisar API



Documentación



Documentación

➤ Manuales

- *Getting Started with the Embedded Development Kit*
- *Platform Studio User Guide*
- *Embedded System Tools Ref. Manual* → *Xilinx Platform Studio*
- *MicroBlaze Processor Reference Guide*
- *Processor IP Reference Guide*

➤ Soporte Web

- MicroBlaze
 - <http://www.support.xilinx.com/microblaze>
- Módulos IP
 - http://www.support.xilinx.com/ise/embedded/edk_ip.htm
- EDK
 - <http://www.support.xilinx.com/edk>

