Diseño de un sistema básico

utilizando MicroBlaze

(Parte 1)

Introducción

En este documento se muestra como crear un sistema básico con el procesador Microblaze utilizando la herramienta Xilinx Platform Studio (XPS). Este sistema se implementa en una placa de desarrollo Nexys 2.

Objetivos

- Crear un proyecto en XPS utilizando el Base System Builder (BSB)
- Crear un diseño simple, utilizando IP disponible en el Embedded Development Kit (EDK)

Procedimiento

El propósito de este documento es servir de apoyo para implementar el hardware y software de un sistema básico basado en un procesador softcore de 32 bits (MicroBlaze).

Se utilizará la herramienta BSB (Base System Builder) del entorno XPS para crear un sistema básico con los siguientes elementos e IP (Figura 1.1):

- MicroBlaze (versión 7.30.b)
- PLB_MDM
- LMB BRAM controlador para BRAM
- BRAM
- UART para comunicación serie
- GPIO para comandar LEDs



Figura 1.1: Sistema básico

Esta guía cuenta con 3 etapas:

- Crear un proyecto usando la herramienta Base System Builder
- Analizar el proyecto creado
- Probar el proyecto en hardware

En cada etapa hay instrucciones generales (indicadas por el símbolo (\bullet)). Estas instrucciones generales dan una idea global de la tarea a realizar. A continuación se indican instrucciones paso a paso más detalladas con figuras ilustrativas. De acuerdo al conocimiento y experiencia, pueden saltarse las instrucciones paso a paso e ir directamente a la siguiente instrucción general.

Crear el proyecto utilizando Base System Builder

Paso 1

- Iniciar el entorno Xilinx Platform Studio (XPS) y crear un nuevo proyecto. Usar la herramienta Base System Builder para generar un sistema básico con MicroBlaze y una aplicacion que verifica la memoria. Implementar el sistema en la placa de desarrollos Nexys2.
 - 1.1. Iniciar XPS seleccionando Inicio ► Todos los Programas ► Xilinx Design Tools ► ISE Design Suite 14.7 ► EDK ► Xilinx Platform Studio.
 - 1.2. Seleccionar Create New Project Using Base System Builder para iniciar el wizard. Se puede llegar al mismo diálogo por medio de File ► New BSB Project

& Create New XPS Project Using BSB Wizard
New Project Project File Browse
Select an Interconnect Type
AXI System
AXI is an interface standard recently adopted by Xilinx as the standard interface used for all current and future versions of Xilinx IP and tool flows. Details on AXI can be found in the AXI Reference Guide on xilinx.com.
O PLB System
PLB is the legacy bus standard used by Xilinx that supports current FPGA families, including Spartan6 and Virtex6. PLB IP will not support newer FPGA families, so is not recommend for new designs that may migrate to future FPGA families. Details on PLB can be found in the PLBv46 Interface Simplifications document on xilinx.com.
Select Existing .bsb Settings File(saved from previous session)
Browse
Set Project Peripheral Repository Search Path
Browse
Help OK Cancel

Figura 1.2: Creación de un nuevo proyecto utilizando Base System Builder

- 1.3. Con el botón **Browse** seleccionar la ubicación del nuevo proyecto, luego presionar **Open** para abrir el directorio. Luego presionar el botón **Save** (Figura 1.3) para guardar el archivo system.xmp.
- 1.4. Seleccionar PLB System.
- 1.5. Obtener el **Board Support Package** para la placa Nexys2 y guardarlo localmente. Con el botón **Browse** de **Set Project Peripheral Repository Search Path** seleccionar el subdirectorio Lib. Finalmente presionar **OK**.

🍪 Create New 🕽	KPS Project Using BSB Wizard	×								
New Project <u>P</u> roject File	D:\/MyS\Laboratorios\Lab1\system.xmp	Browse								
Select an Int	terconnect Type									
AXI AXI futu xilir	AXI System AXI is an interface standard recently adopted by Xilinx as the standard interface used for all current and future versions of Xilinx IP and tool flows. Details on AXI can be found in the AXI Reference Guide on xilinx.com.									
PLB PLB Virt mig doc	System is the legacy bus standard used by Xilinx that supports current FPGA families, including ex6. PLB IP will not support newer FPGA families, so is not recommend for new designs rate to future FPGA families. Details on PLB can be found in the PLBv46 Interface Simpl ument on xilinx.com.	Spartan6 and that may ifications								
Select Existin	ng .bsb Settings File(saved from previous session)									
		Browse								
Set Project F	Peripheral Repository Search Path									
D: MyS We	xys2_BSB_Support_v_3_0\lib	Browse								
Help	OK	Cancel								

Figura 1.3: Establecimiento del directorio del proyecto

- 1.6. En el Nuevo diálogo Welcome to Base System Builder seleccionar la opción I would like to create a new design y presionar Next.
- 1.7. En el diálogo Board Selection, especificar la siguiente configuración (Figura 1.4) y presionar Next.
 - Board Vendor: Digilent
 - Board Name: Nexys 2-500 Board
 - Board Revision (Verify on board): C

🍪 Base System Bui	ilder						? ×
Welcom	e Bo	ard Sys	tem	Processor	Peripheral	Cache	Summary
Board Selection							
Select a target deve	opment board.						
Board							
I would like to cr	eate a system for t	the following developm	nent board				
Board Vendor	Digilent						•
Board Name	Nexys 2-500 Boar	ď					-
Board Revision	С						•
I would like to cr	eate a system for a	a custom board					
-Board Information -							
Architecture		Device		Package		Speed Grade	
spartan3e	-	XC3S500E	+	FG320	Ŧ	-4	Ŧ
Use Stepping							_
Reset Polarity Activ	e High						_

Figura 1.4: Selección de la placa de desarrollo

1.8. En el diálogo **System Configuration**, dejar la opción por defecto **Single-Processor System** (Figura 1.5) y presionar **Next**.



Figura 1.5: Configuración del sistema

- 1.9. En el dialogo **Processor Configuration** (Figura 1.6), dejar la configuración por defecto (ver más abajo) y presionar **Next**.
 - Reference Clock Frequency: 50 MHz (Este es el oscilador externo de la placa de desarrollos; se utilizará para generar la temporización del procesador y los buses).
 - Processor type: MicroBlaze
 - System Clock Frequency–bus Clock Frequency: 50 MHz
 - Local Memory: 8 KB
 - Debug Interface: On-Chip H/W debug module

🍪 Base System Builder						? x
Welcome	Board	System	Processor	Peripheral	Cache	Summary
Processor Configuration Configure the processor(s).	1					
Reference Clock Frequency	50.00					→ MHz
Processor Type	MicroBlaze					•
System Clock Frequency	50.00					▼ MHz
Local Memory	8 KB					•
Debug Interface	On-Chip HW Debug	Module				-
Enable Floating Point	Unit					



Seleccionar y configurar a LEDs_8Bit y RS232_PORT como dispositivos externos, a los controladores dImb y ilmb como controladores internos. Generar las aplicaciones de testeo de memoria y testeo de periféricos mediante el script de enlazado (linker script).

1.10. En el diálogo Peripheral Configuration, seleccionar RS232_PORT y LEDs_8Bit como periféricos externos, según se muestra a continuación, quitar los periféricos "Push_Buttons", "SSG_Decoder", "Digilent_USB" y "Switches" de Processor 1 (MicroBlaze) Peripherals. Los controladores dlmb y ilmb no se pueden quitar porque son de uso interno del procesador. En este punto se podría utilizar Add para agregar periféricos. En la siguiente práctica se verá otro método para agregarlos al sistema.

😵 Base System Builder						? X
Welcome	Board	System	Processor	Peripheral	Cache	Summar
Peripheral Configuration To add a peripheral, drag it from	the "Available Pe	ripherals" to the pro	cessor peripheral list. To	change a core parameter,	click on the peripher	al.
Available Peripherals Peripheral Names			Processor 1 (MicroBla	aze) Peripherals	(Select All
IO Devices Push_Buttons_3Bit Switches_8Bit Ssg_Decoder_0 Digilent_Usb_Epp mem_bus_mux_0 Internal Peripherals Internal Peripherals	IO Devices Push_Buttons_3Bit Ssg_Decoder_0 Digilent_Usb_Epp mem_bus_mux_0 Internal Peripherals		Core INTEL_FLASH Core: xps_mch LEDs_8Bit Core: xps_gpic Micron_RAM Core: xps_mch	n_emc	Parameter	
xps_bram_ff_cntlr xps_timebase_wdt xps_timer		Add >	RS232_PORT RS232_PORT Parity Use Interrupt dImb_cntlr Core: Imb_bra iImb_cntlr Core: Imb_bra	m_if_cntlr m_if_cntlr	xps_uartlite None	▼

Figura 1.7: Configuración de periféricos

- 1.11. Presionar Next para pasar al diálogo Cache Configuration, no seleccionar nada y presionar Next.
- 1.12. Verificar la configuración en el diálogo **Summary** (Figura 1.8) y presionar **Finish**.

🍪 Base System Builder						? ×
Welcome	Board	System	Processor	Peripheral	Cache	Summar
Summary Below is the summary of the	system you are cr	eating.				
System Summary						
Core Name	Instance Name	Base Address	High Address			
 Processor 1 xps_mch_emc xps_gpio xps_mch_emc xps_uartlite Imb_bram_if_cntlr Imb_bram_if_cntlr 	microblaze_0 INTEL_FLASH LEDs_8Bit Micron_RAM RS232_PORT dImb_cntlr ilmb_cntlr	0x82000000 0x81400000 0x83000000 0x0000000 0x0000000 0x0000000	0x82FFFFF 0x8140FFFF 0x83FFFFFF 0x8400FFFF 0x00001FFF 0x00001FFF			
ile Location - Overall - D:\MyS\Laboratori - D:\MyS\Laboratori - D:\MyS\Laboratori - D:\MyS\Laboratori - D:\MyS\Laboratori - D:\MyS\Laboratori	ios\Lab1\system. ios\Lab1\system. ios\Lab1\data\sy: ios\Lab1\etc\fast ios\Lab1\etc\dov ios\Lab1\etc\bitg	xmp mhs stem.ucf _runtime.opt vnload.cmd yen.ut				
✓ Save Base System Builder D: \MyS\Laboratorios\Lab 1	r (.bsb) Settings Fil \system.bsb	e				
More Info				< <u>B</u> ack	Einish	Cancel

Figura 1.8: Diálogo System Summary

1.13. Aparece una vista del sistema (Figura 1.9), mostrando los periféricos y buses que se utilizan y su conexionado.

LLP	Bus Interfaces	Ports	Addresses		
M M L B B B	Name	Bus Name	IP T	ype	IP Version
8880	dlmb		*	lmb_v10	2.00.b
	ilmb		*	lmb_v10	2.00.b
🕺 💥 🐹 o	mb_plb		*	plb_v46	1.05.a
	microblaze_0		*	microblaze	8.50.c
	🗄 lmb_bram		*	bram_block	1.00.a
	🕀 dlmb_cntlr		*	Imb_bram_i	3.10.c
	ilmb_cntlr ⊡		*	Imb_bram_i	3.10.c
······································	INTEL_FLASH		*	xps_mch_e	3.01.a
<u> </u>	Micron_RAM		*	xps_mch_e	3.01.a
<u> </u>	⊕ mdm_0		*	mdm	2.10.a
<u> </u>	🖶 LEDs_8Bit		*	xps_gpio	2.00.a
<u> </u>	RS232_PORT		*	xps_uartlite	1.02.a
	clock_gener		*	clock_gene	4.03.a
	mem_bus		*	mem_bus	1.00.a
	proc_sys_re		*	proc_sys_re	3.00.a

Figura 1.9: Vista del sistema

Análisis del harware

Paso 2

Seleccionar la pestaña correspondiente a "Graphical Design View" y analizar los componentes del sistema y sus interconexiones. En la pestaña "System Assembly View" analizar el detalle de las conexiones del sistema.

2.1. Seleccionar la pestaña "**Graphical Design View**". Observar los diferentes componentes del sistema y sus interconexiones (se puede utilizar la tecla Ctrl y la rueda del mouse para realizar zoom sobre la vista).



Figura 1.10: Vista de diagrama en bloques del proyecto generado

2.2. En la vista **System Assembly** presionar el botón + y observar el detalle de las conexiones del bus del sistema.



Figura 1.11: Conexiones del bus

2.3. Seleccionar la pestaña **Ports** y expandir la vista. Aquí es donde se pueden realizar conexiones internas y externas de periféricos. Si no puede encontrar la columna **Net** haga click en el botón derecho del Mouse sobre **Connected Port** para ver las opciones disponibles y elegir **Net**.

Diseño de un sistema básico utilizando MicroBlaze

	Bus Interfaces	Ports	Addresses											S	•
Na	ame		Connected P	ort		Net			Direction	Range	Class		Frequency(Hz)	Reset Polar	ity
ė	External Ports		1												
	fpga_0_LEDs	_8Bit	LEDs_8Bit::[gr	pio_0]::GPIO_IO_	D 🥖	fpga_0_LEDs	_8Bit_GPIO_IO	-	0 -	[0:7]	NONE	T]		
	fpga_0_RS23	2_POR	RS232_PORT:	:[uart_0]::RX	/	fpga_0_RS23	2_PORT_RX_pin	-	I		NONE	-]		
	fpga_0_RS23	2_POR	RS232_PORT:	:[uart_0]::TX	/	fpga_0_RS23	2_PORT_TX_pin	-	0		NONE	-			
	fpga_0_clk_1	_sys_cl	clock_genera	tor_0::CLKIN	/	CLK_S		-	I		CLK	-	50000000		
	fpga_0_mem	_bus	mem_bus_m	ux_0::[mem_bus_	.m. 🥖	fpga_0_mem	_bus_mux_0_D	-	IO 💌	[0:15]	NONE				
	fpga_0_mem	_bus	mem_bus_m	ux_0::[mem_bus_	m. 🦊	fpga_0_mem	_bus_mux_0_FL	-	0		NONE	-]		
	fpga_0_mem	_bus	mem_bus_m	ux_0::[mem_bus_	.m. 🥖	fpga_0_mem	_bus_mux_0_FL	-	0]	NONE	T]		
	fpga_0_mem	_bus	mem_bus_m	ux_0::[mem_bus_	.m. 🥖	fpga_0_mem	_bus_mux_0_M	-	0	[0:22]	NONE	-]		
	fpga_0_mem	_bus	mem_bus_m	ux_0::[mem_bus_	.m. 🥖	fpga_0_mem	_bus_mux_0_M	-	0		NONE	-]		
	fpga_0_mem	_bus	mem_bus_m	ux_0::[mem_bus_	.m. 🥖	fpga_0_mem	_bus_mux_0_M	-	0		NONE	-]		
	fpga_0_mem	_bus	mem_bus_m	ux_0::[mem_bus_	.m. 🥖	fpga_0_mem	n_bus_mux_0_RA	-	0	[0:1]	NONE	-]		
	fpga_0_mem	_bus	mem_bus_m	ux_0::[mem_bus_	.m. 🥖	fpga_0_mem	n_bus_mux_0_RA	-	0		NONE	-]		
	fpga_0_rst_1_	_sys_rs	clock_genera proc_sys_rese	tor_0::RST et_0::Ext_Reset_In	1	sys_rst_s		•	I		RST	-		1	-
Ė	- dlmb														
	LMB_Clk		clock_genera	tor_0::CLKOUT0	1	clk_50_00001	ИHz	-	I		CLK				
	SYS_Rst		proc_sys_rese	t_0::Bus_Struct_F	les et	sys_bus_rese	t	-	I		RST				
Ė	- ilmb														
	- LMB_Clk		clock_genera	tor_0::CLKOUT0	1	clk_50_00001	ИHz	-	I		CLK				
	SYS_Rst		proc_sys_rese	t_0::Bus_Struct_F	les 📌	sys_bus_rese	t	-	I		RST				
Ė	mb_plb														
	PLB_Clk		clock_genera	tor_0::CLKOUT0	1	clk_50_00001	ИHz	-	I		CLK				
	SYS_Rst		proc_sys_rese	t_0::Bus_Struct_F	lese	sys_bus_rese	t	-	I		RST				
	Bus_Error_De	t			1	No Connect	ion	-	0		INTERRU	т			
l 📥	microblaze 0														

Figura 1.12: Puertos

2.4. Seleccionar la pestaña **Addresses**. Aquí es donde se puede asignar o cambiar el rango de direcciones de los componentes del sistema.

•	Bus Interfaces	Ports	Addresses							
Instance			Base	Name	Base Address	High Address	Size	Bus Interface(s)	Bus Name	Lock
ė	🖨 microblaze_0's Address Map									
	dlmb_cntlr		C_B4	ASEADDR	0x00000000	0x00001FFF	8K .	SLMB	dlmb	
	- ilmb_cntlr		C_BA	ASEADDR	0x00000000	0x00001FFF	8K -	SLMB	ilmb	
	LEDs_8Bit		C_B4	ASEADDR	0x81400000	0x8140FFFF	64K	SPLB	mb_plb	
	INTEL_FLAS	Н	C_M	EM0_BASE	0x82000000	0x82FFFFFF	16M	SPLB	mb_plb	
	- Micron_RAM	N	C_M	EM0_BASE	0x83000000	0x83FFFFFF	16M .	SPLB	mb_plb	
	RS232_PORT	Г	C_BA	ASEADDR	0x84000000	0x8400FFFF	64K	SPLB	mb_plb	
	mdm_0		C_BA	ASEADDR	0x84400000	0x8440FFFF	64K	SPLB	mb_plb	

Figura 1.13: Vista de la asignación de direcciones

Completar el diseño

Paso 3

- Ejecutar Platgen para generar la netlist y ejecutar las herramientas de implementación para generar el bitstream.
 - 3.1. Ejecutar la herramienta PlatGen seleccionando **Hardware** ► **Generate Netlist** o presionando el botón en la barra de herramientas.

Nota: Se generarán los directorios implementation, hdl y synthesis. Los perifericos generados y la netlist del sistema junto system.bmm serán colocados en el directorio implementation.

3.2. Generar el bitstream seleccionando Hardware ► Generate BitStream o seleccionando el botón barra de herramientas.

Nota: Este punto puede demandar más de 10 minutos, dependiendo de la máquina utilizada.

3.3. Seleccionar la pestaña **Design Summary**. Allí hay un resumen de la información de implementación del diseño, reportes y mensajes. Hay información específica respecto a la FPGA usada y mensajes de las distintas herramientas de software utilizadas. Los paneles del lado izquierdo permiten controlar la información que se muestra en el panel derecho.

Generar en SDK la aplicación para verificar la memoria

Paso 4

ا 🔶

Iniciar SDK desde XPS, generar el proyecto de prueba de la memoria con default software platform project.

- 4.1. Iniciar SDK seleccionando Project ► Export Hardware Design to SDK ... desde XPS.
- 4.2. Seleccionar el botón Export & Launch SDK con la configuración por defecto (la opción Include bitstream and BMM file marcada).
- 4.3. En la ventana Select a workspace navegar hasta ... \lab1\SDK\SDK_Export y presionar OK.
- 4.4. En SDK seleccionar File ► New ► Other... y en el siguiente diálogo, dentro de la carpeta Xilinx, seleccionar Application Project y luego presionar Next.
- 4.5. En el diálogo Application Project completar el campo Project name. Presionar Next.
- 4.6. En el diálogo **Templates** seleccionar **Memory Tests** y presionar **Finish**. En la figura 1.14 se puede observar el resultado de la compilación.



Figura 1.14: Resultado de la compilación

Verificación en hardware

Paso 5

Conectar y encender el kit. Establecer una conexión serie utilizando algún Terminal. Configurar la FPGA y verificar el correcto funcionamiento.

- 5.1. Conectar y encender el kit Nexys 2.
- 5.2. Dentro del SDK seleccionar la pestaña **Terminal 1** (si no es visible seleccionar **Window** ► **Show view** ► **Terminal**).

Nota: puede ser utilizado cualquier programa que permita una conexión serie.

- 5.3. Presionar el botón "Connect" para la conexión, seleccionar el puerto COM apropiado y configurar los diferentes parámetros de la siguiente manera:
 - Connection Type: Serial
 - Baud Rate: 9600

- Data bits: 8
- Stop bits: 1
- Parity: None
- Flow control: None

🖹 Problems 🖉 Tasks 🗐 Console 💷 Properties 🔎 Terminal 🗙 🛛 💀 🛅 🌆 🐰 🖉 👻 🔡 🕶	× - D
No Connection Selected	
	~
Botón Connect	
	-

Figura 1.15: Ubicación del botón Connect

5.4. Seleccionar en el SDK Xilinx Tools ► Program FPGA.

5.5. Seleccionar con el botón drop-down el archivo memory_test_0.elf

🐵 Program F	FPGA			X				
Program FPGA Image: Specify the bitstream and the ELF files that reside in BRAM memory								
- Hardware C	Configu	ration						
Hardware S	pecific	ation: D:\MyS\Laboratorios\Lab1\SDK\SDK_Export\Lab1_h	w_platform\	system.xml				
Bitstream:	syster	n.bit	Search	Browse				
BMM File:	syster	n_bd.bmm	Search	Browse				
-Software Co	onfigur	ation						
Processor		ELF File to Initialize in Block RAM						
microblaze	e_0	D:\MyS\Laboratorios\Lab1\SDK\SDK_Export\memon						
?		bootloop D:\MyS\Laboratorios\Lab1\SDK\SDK_Export\memory_tes Browse	st_0\Debuq\i	memory_test_0.elf				

Figura 1.16: Selección de la aplicación

5.6. Presionar **Program**.

El archivo **memory_test_0.elf** y el **system.bit** se combinan en otro archivo denominado **download.bit** que es el que efectivamente configura la FPGA.

Nota: en el caso de aparecer un mensaje de error seleccionar Xilinx Tools ► Configure JTAG Settings y en el campo Type del JTAG Cable seleccionar Digilent USB Cable. Volver a repetir el paso anterior. Si el error persiste se deberá utilizar la herramienta Digilent Adept para transferir el archivo de configuración download.bit.

- 5.7. En la herramienta **Digilent Adept** seleccionar el archivo **download.bit**, que se encuentra en el directorio **Implementation** del proyecto, y configurar la FPGA (este paso se realizará en caso de persistencia del error en el paso anterior)
- 5.8. Una vez configurada la FPGA debería verse en la terminal el siguiente mensaje:



Figura 1.18. Mensaje observado en la terminal

Conclusión

La herramienta **Base System Builder** se usa en XPS para generar un sistema basado en MicroBlaze junto con una aplicación de software para prueba. Se crean distintos archivos utilizados por las diferentes herramientas de software necesarias para implementar el sistema en FPGA. Se utilizan distintas vistas para acceder a la configuración del sistema. Una vez configurado éste, se crean las netlists necesarias para la implementación. Una vez implementado el sistema, se agrega al bitstream la información necesaria para inicializar la memoria de código del procesador con la aplicación de software.