Simple Hardware Design

Targeting MicroBlaze™ on Spartan™-3E Starter Kit

Lab 1: Simple Hardware Design Lab

Introduction

This lab guides you through the process of using Xilinx Platform Studio (XPS) to create a simple processor system targeting the Spartan-3E Starter Kit

Objectives

After completing this lab, you will be able to:

- Create an XPS project by using the Base System Builder (BSB)
- Create a simple hardware design by using Xilinx IP cores available in the Embedded Development Kit

Procedure

The purpose of the lab exercises is to walk you through a complete hardware and software processor system design. Each lab will build upon the previous lab. The following diagram represents the completed design (**Figure 1-1**).

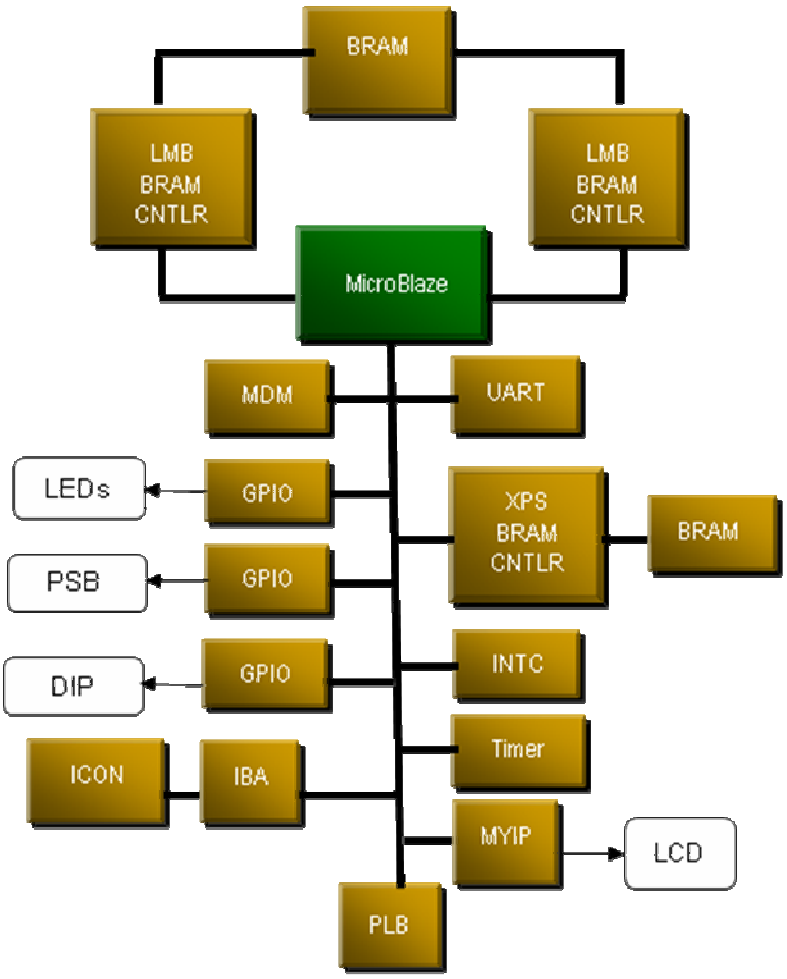

Figure 1-1. Completed Design

In this lab, you will use the BSB of the XPS system to create a processor system consisting of the following processor IP (**Figure 1-2**):

- MicroBlaze (version 7.0)
- PLB_MDM
- LMB BRAM controllers for BRAM
- BRAM
- UART for serial communication
- GPIO for LEDs

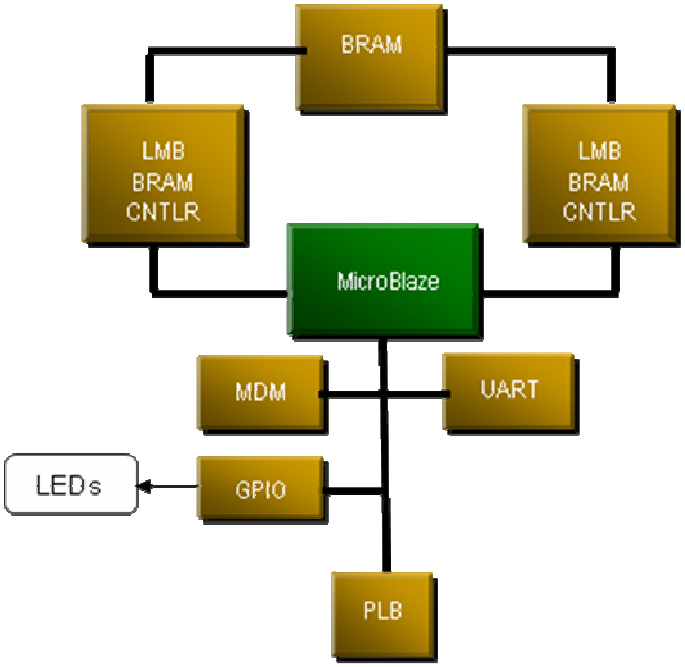

Figure 1-2. Processor IP

This lab comprises three primary steps: You will create a project using the Base System Builder, analyze the created project, and generate the processor and hardware IP netlists.

For each procedure within a primary step, there are general instructions (indicated by the symbol). These general instructions only provide a broad outline for performing the procedure. Below these general instructions, you will find accompanying step-by-step directions and illustrated figures that provide more detail for performing the procedure. If you feel confident about completing a procedure, you can skip the step-by-step directions and move on to the next general instruction.

Creating the Project Using the Base System Builder Step 1

Launch Xilinx Platform Studio (XPS) and create a new project. Use Base System Builder to generate a MicroBlaze system and memory test application targeting the Spartan-3E starter kit. All hardware options specified here are written into the MHS file.

❶ Open XPS by selecting **Start → Programs → Xilinx Platform Studio 9.2i → Xilinx Platform Studio** (**Figure 1-3**)

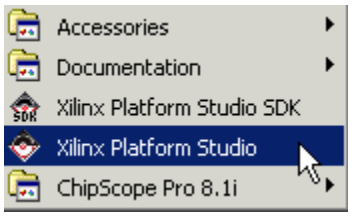

Figure 1-3. Select Xilinx Platform Studio

Do not select **Xilinx Platform Studio SDK**; this opens the Software Development Kit IDE.

❷ The following dialog box appears; click **OK** to create a project using the Base System Builder. If you clicked cancel, you can select **File** → **New Project** and the same dialog box will appear. Select **Base System Builder** option and click **OK** to start the wizard (**Figure 1-4**)

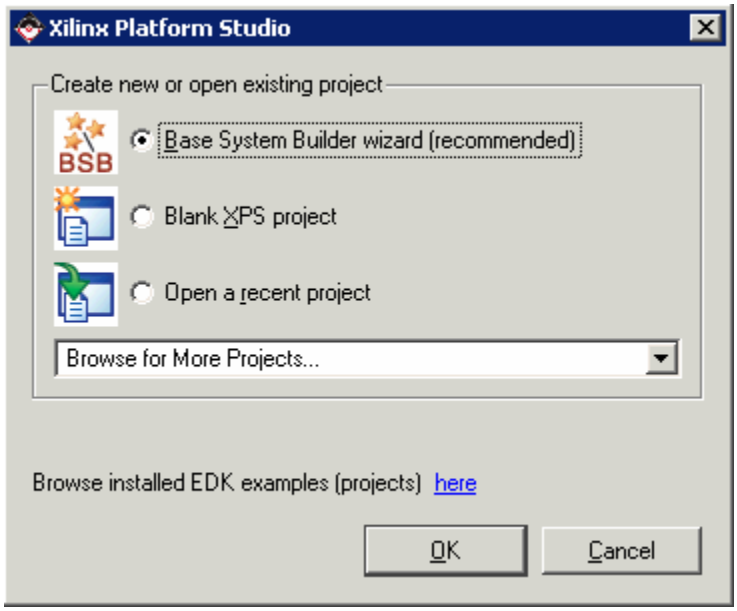

Figure 1-4. New Project Creation Using Base System Builder

This opens the **Create New XPS Project using BSB Wizard** dialog box (**Figure 1-5**).

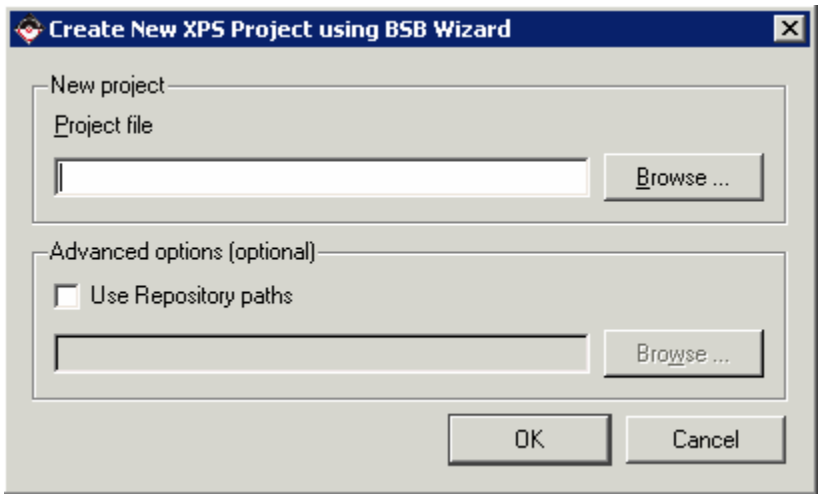

Figure 1-5. Create New XPS Project Using BSB Wizard Dialog Box

❸ Browse to **c:\xup\embedded\labs** directory, click on create a new folder icon, name it as *lab1*, select it, and click **Open** button followed by click **Save** (**Figure 1-6**) [The following figure does not match what you say in this step]

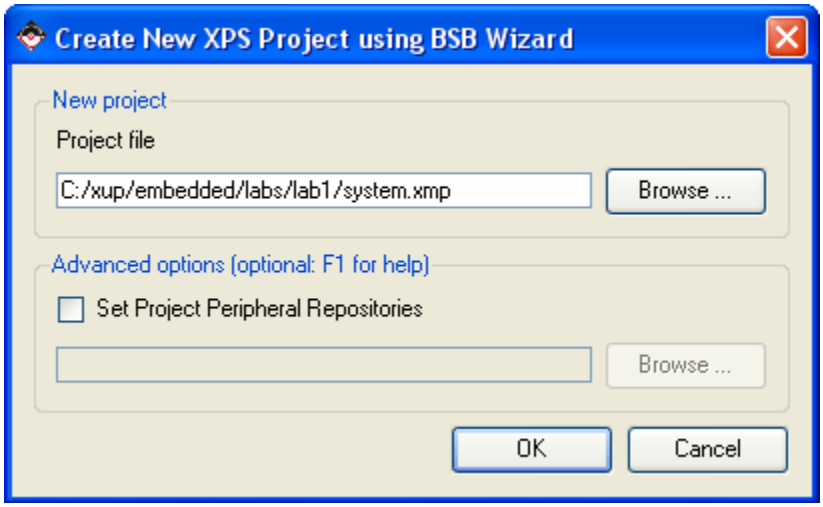

Figure 1-6. Assigning Project Directory

❹ Click **OK** to display the Welcome to Base System Builder dialog box and Select the **I would like to create a new design option**

❺ Click **Next** to display the Select Board dialog box (**Figure 1-7**). Specify the settings to match the following:

- Board Vendor: **Xilinx**
- Board Name: **Spartan™-3E Starter Board**
- Board Revision (Verify on board): **D**

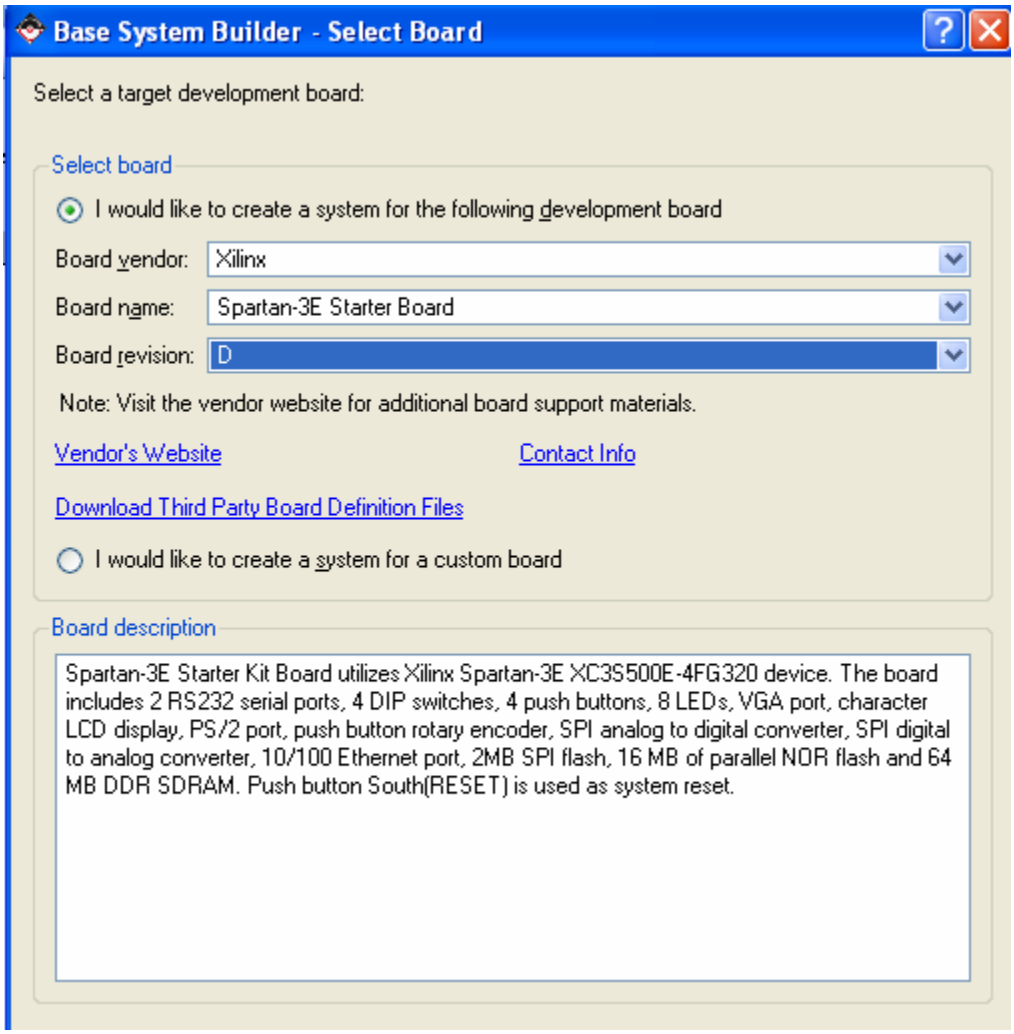

Figure 1-7. Select Board Dialog Box

❻ Click **Next** to display the Select Processor dialog box (**Figure 1-8**)

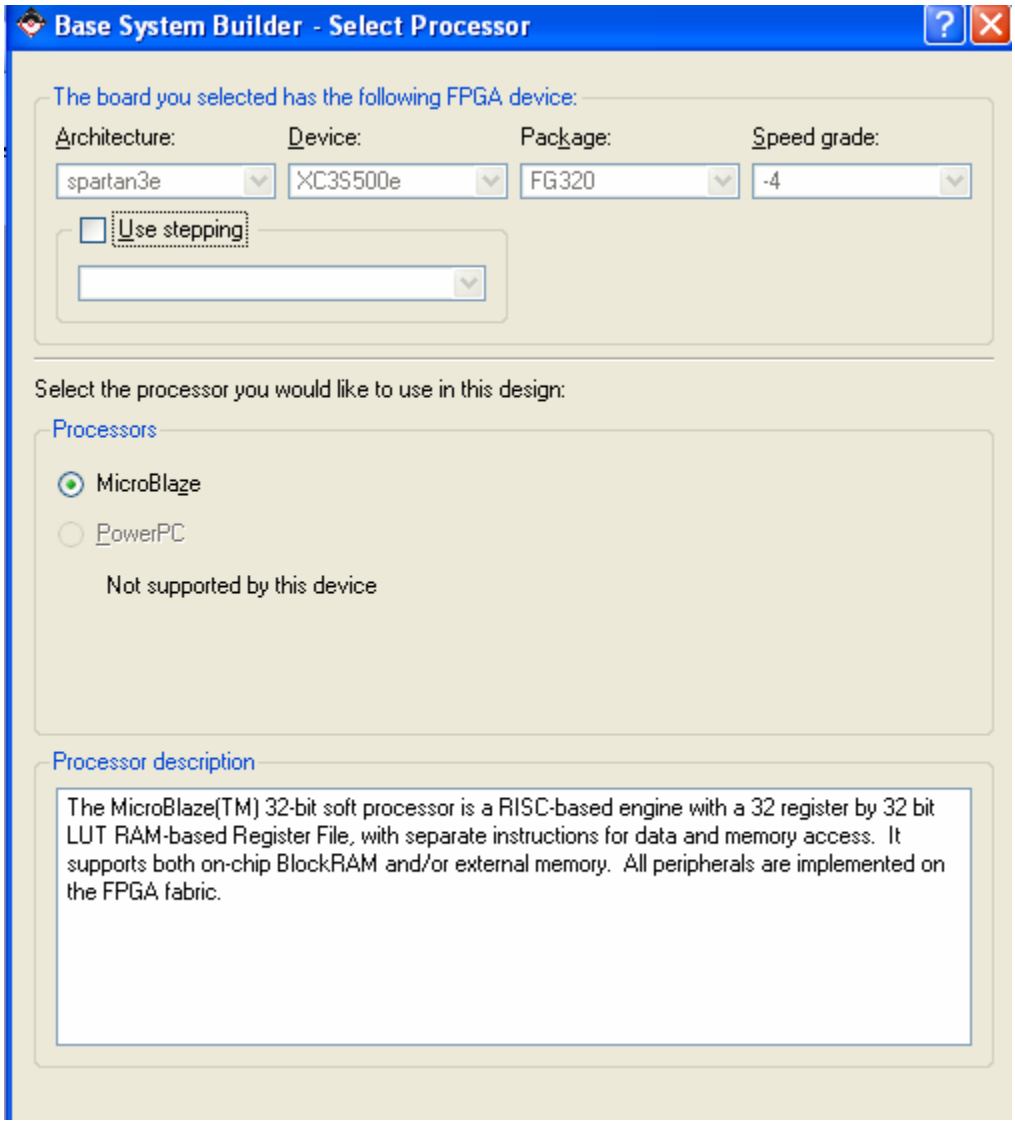

Figure 1-8. Select Processor Dialog Box

❼ Click **Next** to display the Configure Processor dialog box (**Figure 1-9**). Verify the settings to match the following:

- Reference Clock Frequency: **50 MHz**
  - This is the external clock source on the board you are using. This clock will be used to generate the processor and bus clocks. The values allowed may depend on the FPGA or board you are using because certain on-chip resources (DCMs) may be required to perform clock division or multiplication.
- Processor –bus Clock Frequency: **50 MHz**
- Debug Interface: **On-Chip H/W debug module**
- Local Data and Instruction Memory – **8 KB**

- Cache Setup: **Enable - unchecked**

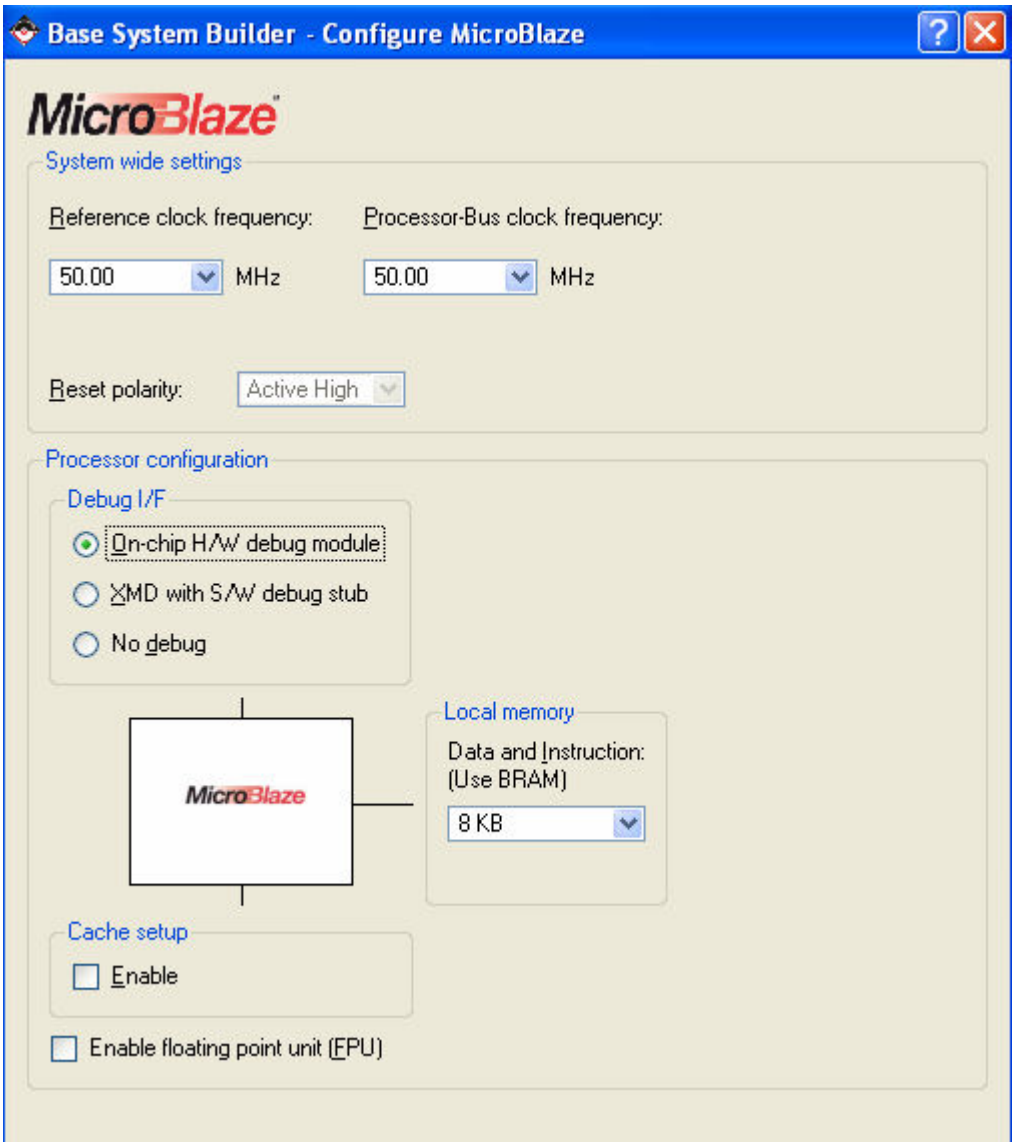

Figure 1-9. Configure Processor Dialog Box

Select LEDs_8Bit (OPB_GPIO, no interrupt), and RS232_DCE (115200 baud rate, no parity, no interrupt) as the only external devices. Generate the memory test sample application and linker script.

❶ Click **Next** to display the Configure IO Interfaces dialog box. Make sure that the **RS232_DCE** and **LEDs_8Bit** peripherals are checked and rest of the peripherals unchecked

- RS232_DCE: XPS UARTLITE, 115200 baud rate, 8 Data bits, no interrupt, no parity (**Figure 1-10**)
- LEDs_8Bit: XPS GPIO. No interrupt (**Figure 1-11**)

Note that the number of peripherals that appear on each window will depend on the resolution of your monitor.

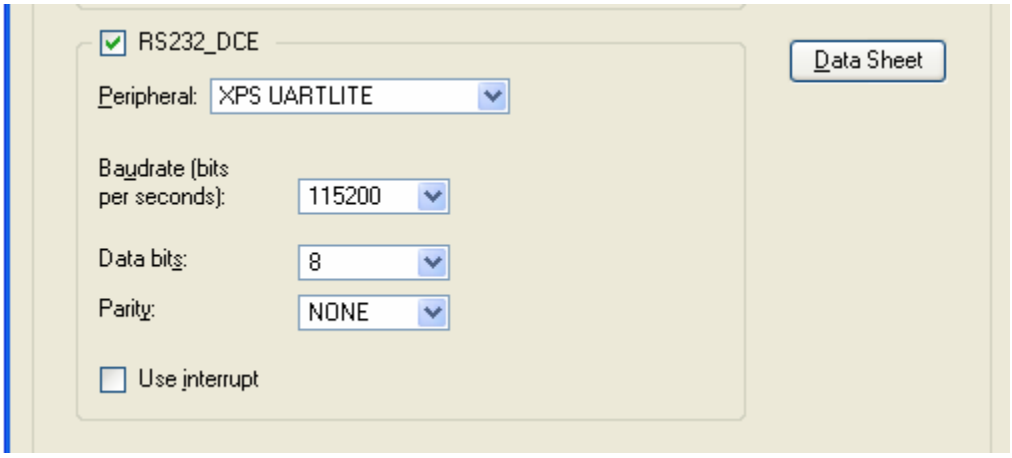

Figure 1-10. Configure RS-232 DCE

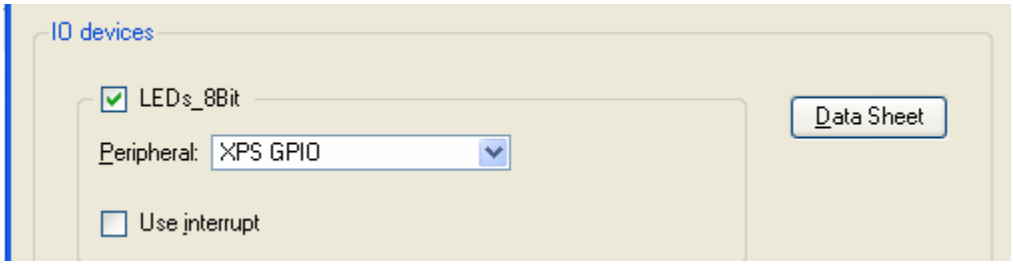

Figure 1-11. Configure XPS GPIO

Click **Next** to display the Configure Additional IO Interfaces dialog box (**Figure 1-12**) and uncheck rest of the devices present

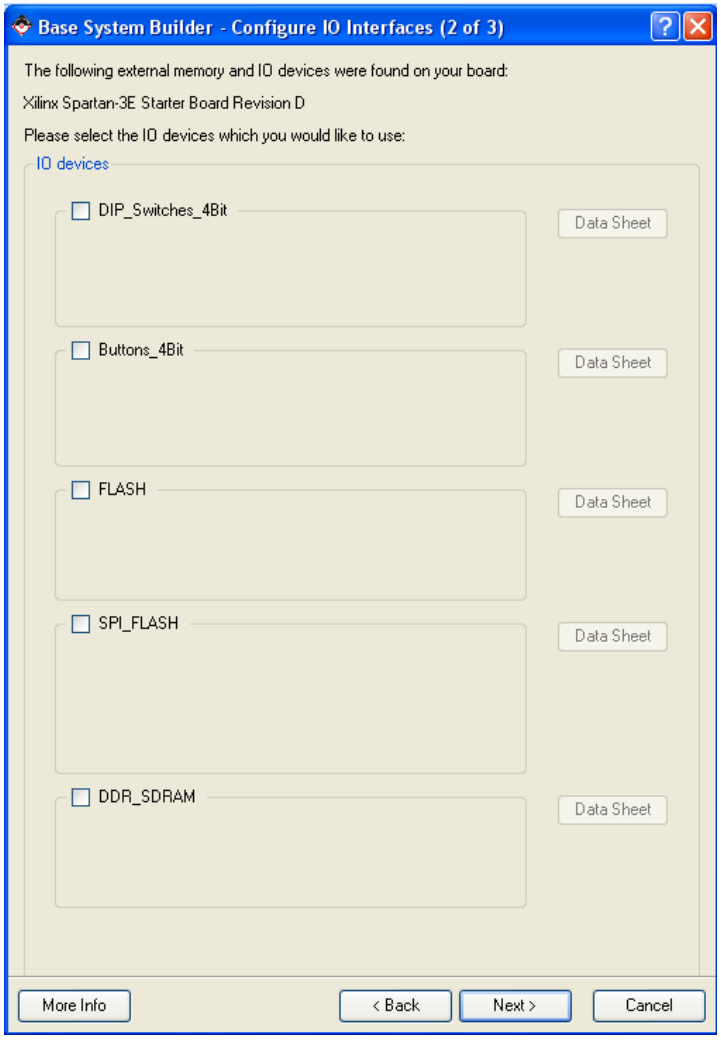

Figure 1-12. Configure Additional IO Interfaces Dialog Box

❸ Click **Next** until the Add Internal Peripherals dialog box is displayed, making sure that none of the other devices are selected (**Figure 1-13**)

At this point you could click **Add Peripheral** to add additional internal peripherals, but you will see an alternative method in the next lab for adding internal peripherals to an existing project.

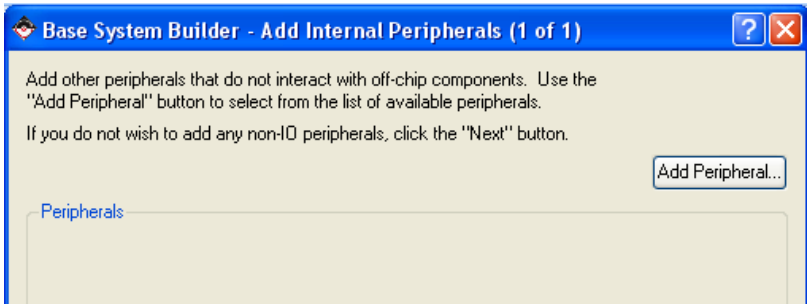

Figure 1-13. Add Internal Peripherals Dialog Box

❹ Click **Next** to display the Software Setup dialog box. Leave **Memory test** sample application selected and unselect **Peripheral selfTest** (**Figure 1-14**)

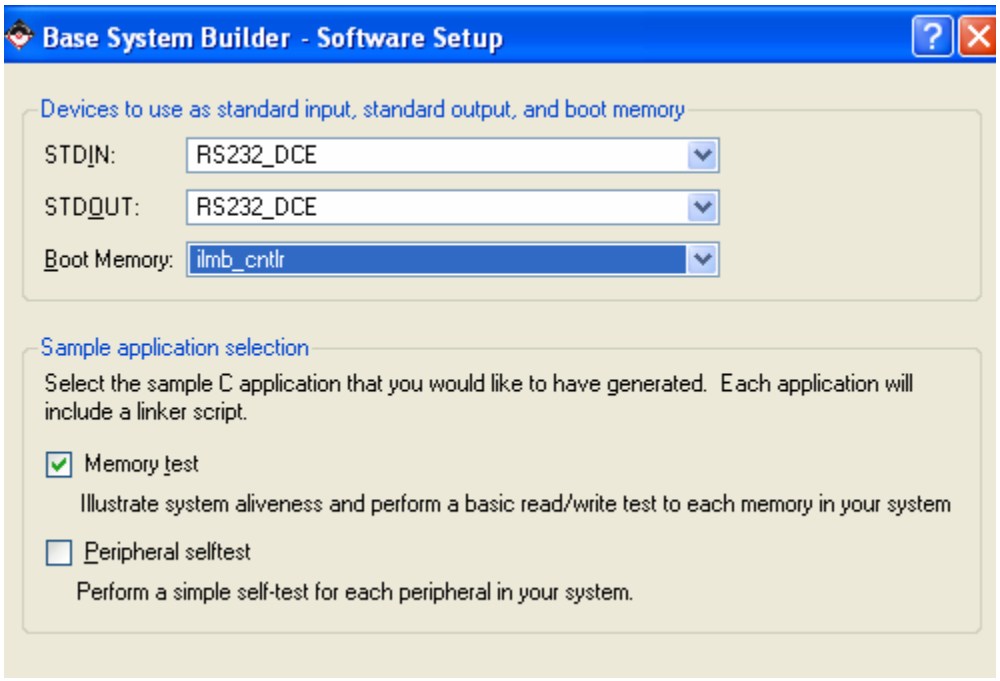

Figure 1-14. Software Setup Dialog Box

❺ Click **Next** to display the Configure Memory Test Application dialog box (**Figure 1-15**)

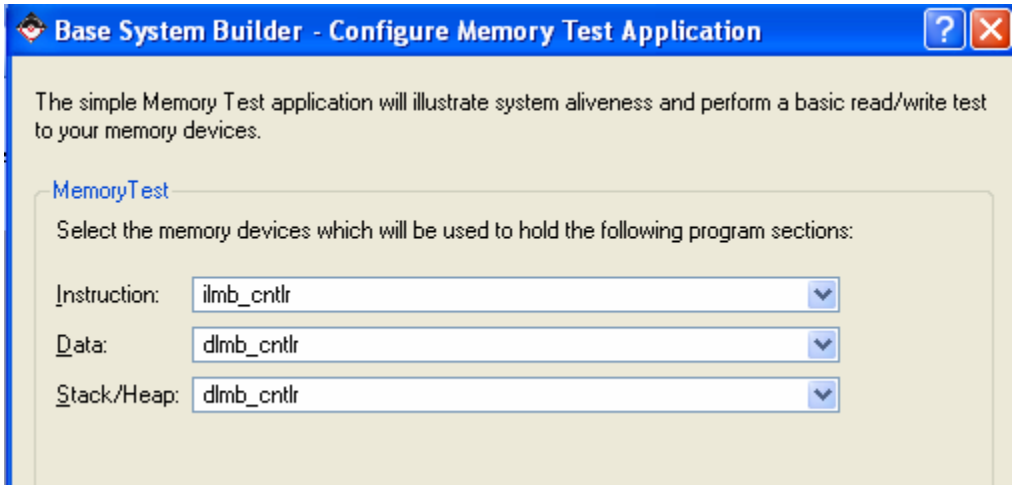

Figure 1-15. Configure Memory Test Application

❻ Click **Next** to display the System Created dialog box which summarizes the system being created (**Figure 1-16**)

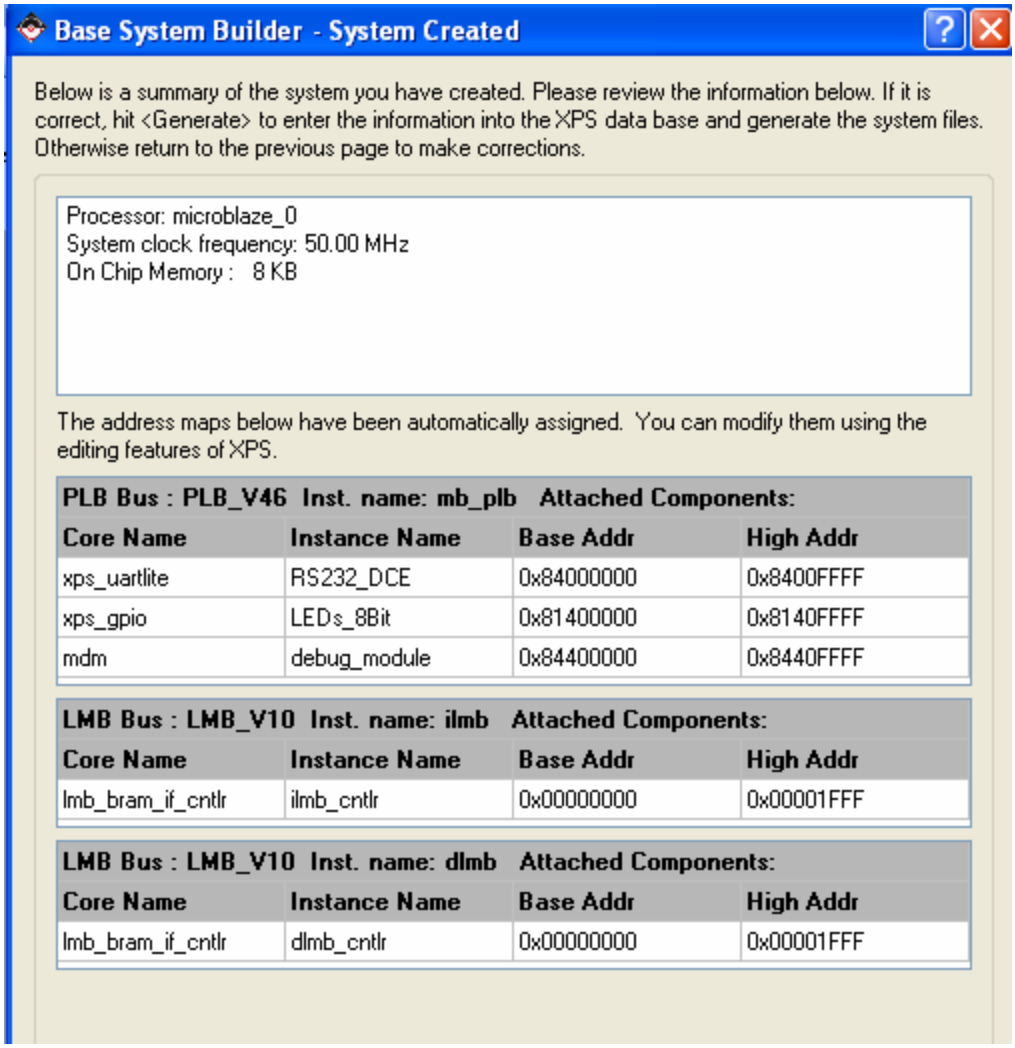

Figure 1-16. System Created Dialog Box

❼ Click **Generate**

A congratulations dialog box appears, indicating the files that BSB has created. Click **Finish**

❽ The Next Step dialog box appears, ensure **Start Using Platform Studio** is checked and click **OK**

A Software Agreement dialog may appear if this is the first time the software is run

❾ A System Assembly View1 will be displayed (**Figure 1-17**) showing peripherals and busses in the system, and the system connectivity.

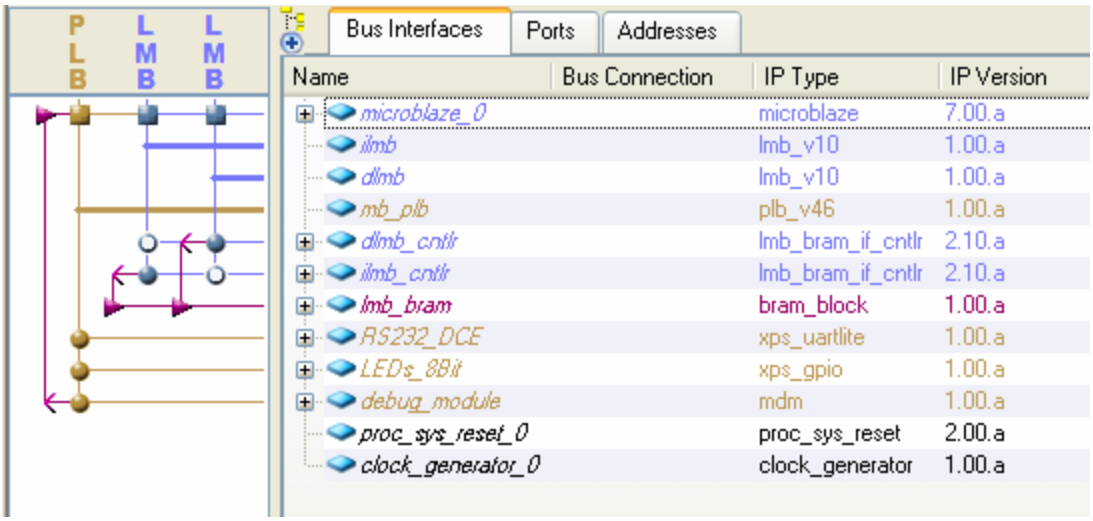

Figure 1-17. System Created Dialog Box

Analyzing the Project Step 2

Generate a block diagram of the system and study the system components and interconnections. Look in the System Assembly View and analyze the bus and port connections.

❶ Click on the **Block Diagram tab** to open a block diagram view (**Figure 1-18**) and observe the various components that are used in the design

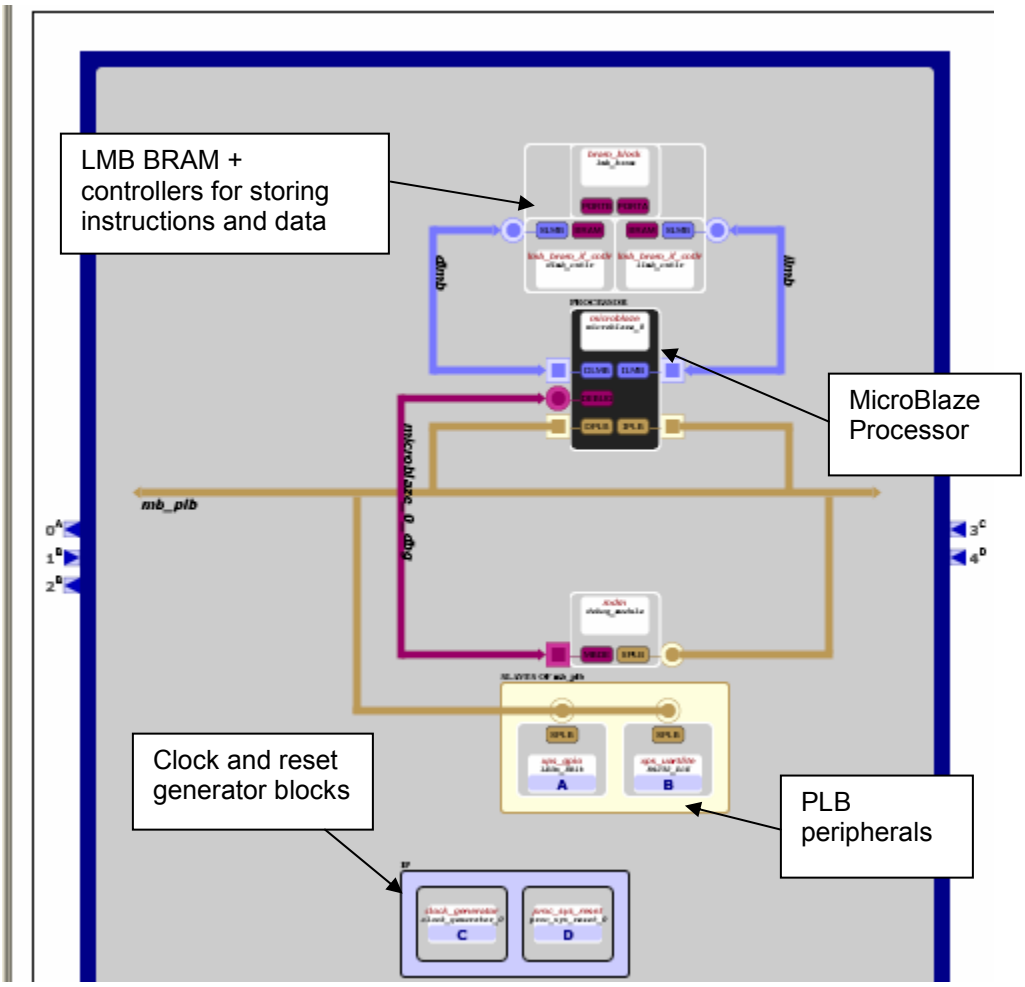

Figure 1-18. Block Diagram View of the Generated Project

You can zoom in and out and use the scroll bars to navigate around the block diagram. You will see the MicroBlaze™ processor, LMB controller and PLB bus connected to the MicroBlaze processor.

❷ Scroll down in the block diagram and see the legends as well as the I/O ports

❸ In the System Assembly View click on plus button and observe the expanded (detailed) bus connection view of the system (**Figure 1-19**)

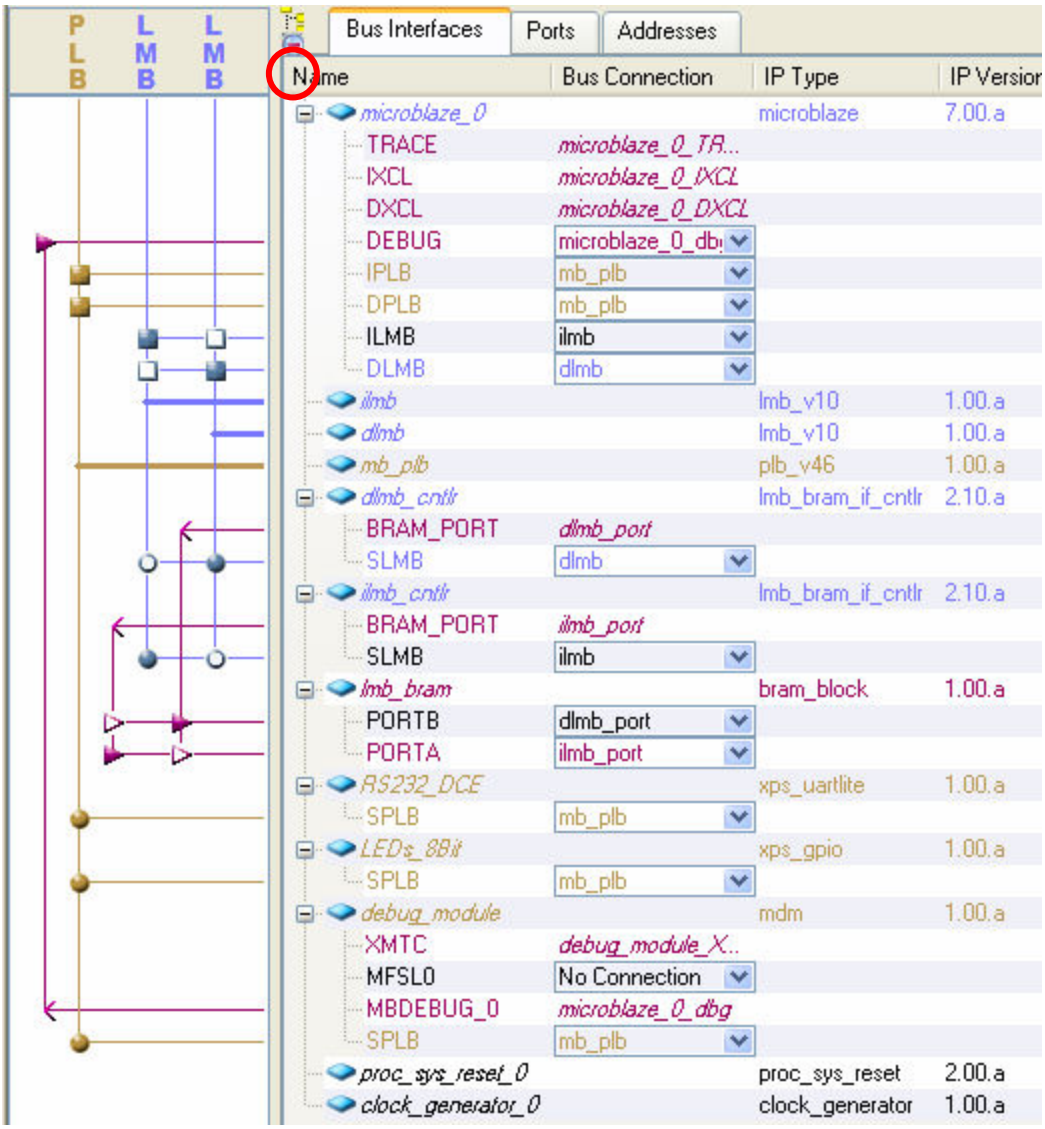

Figure 1-19. Detailed Bus Connections

1. List the bus connection to the following peripherals:

   debug_module: _____
   dlmb_cntlr: _____
   RS232_DCE: _____

❹ Click on the Ports filter and have an expanded view similar to **Figure 1-20.** This is where you can make internal and external net connections.

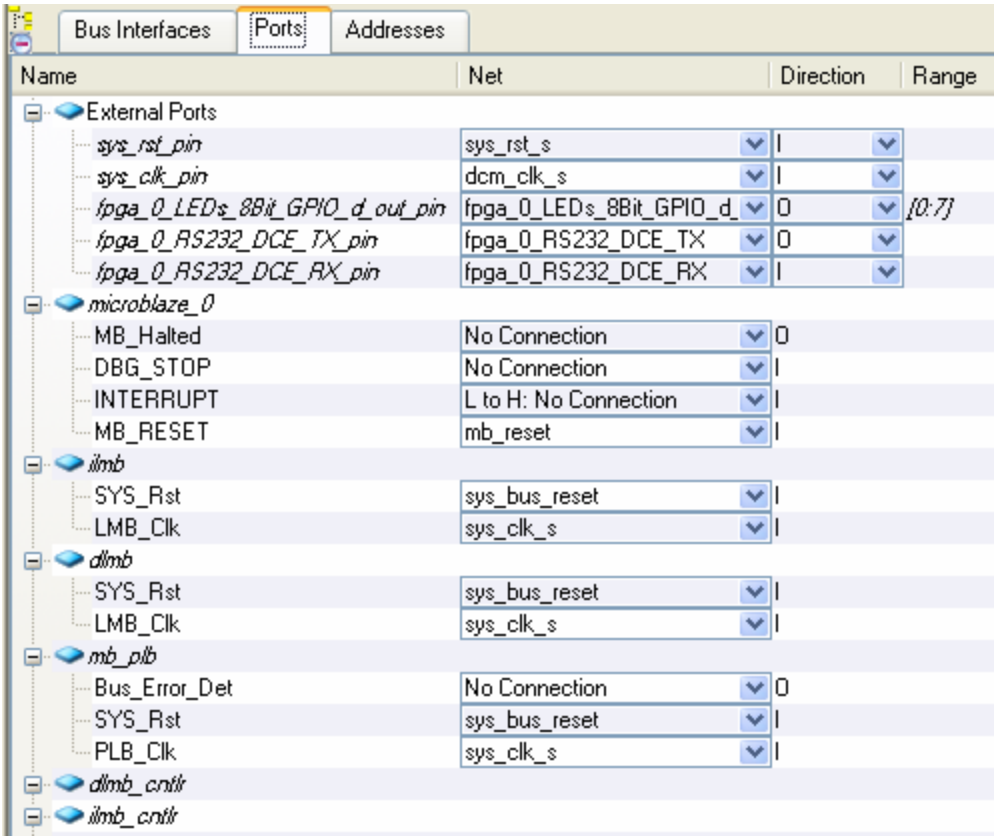

Figure 1-20. Ports Filter

2. List the nets which are connected to the following ports:

RS232_DCE – RX: _____
RS232_DCE – TX: _____
LEDs_8Bit - GPIO: _____

❺ Click on the **Addresses** tab and have an expanded view similar to **Figure 1-21.** This is where you can assign base/high addresses to the peripherals in the system.

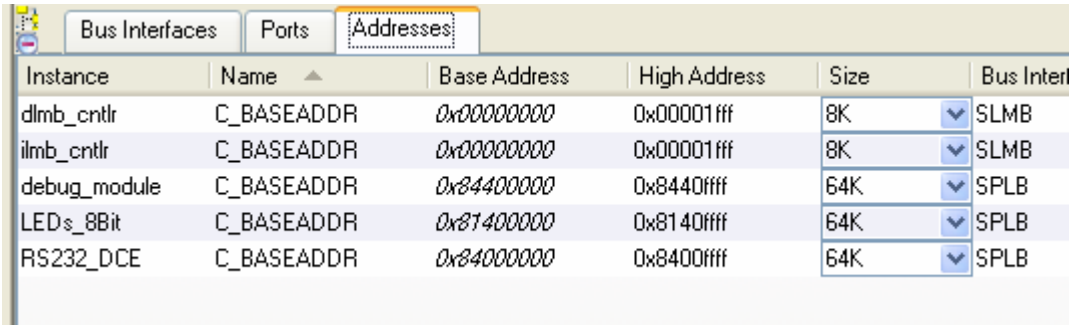

Figure 1-21. Assign Base/High Addresses

3. Select Addresses filter and list the address for the following instances:

RS232_DCE – Base address: _____
RS232_DCE – High address: _____
LEDs_8Bit – Base address: _____
LEDs_8Bit – High address: _____
dlmb_cntlr – Base address: _____
dlmb_cntlr – High address: _____
ilmb_cntlr – Base address: _____
ilmb_cntlr – High address: _____

Generating the System Netlists — Step 3

Run PlatGen to generate the system netlists (NGC) from the MHS file.

1. In XPS, select **Hardware → Generate Netlist** or click  in the toolbar
2. Observe the netlist generation in the console window as the generation progresses
3. Open Windows Explorer by selecting **Start → Programs → Accessories → Windows Explorer**
4. Browse to the ***Lab1*** project directory

   Several directories containing VHDL wrappers and implementation netlists have been created.

   4. List the directories that were created.

Verify in Hardware — Step 4

Generate bitstream and download to the board. Prior to download, the instruction memory (FPGA Block RAM) will be updated in the bitstream with the executable generated using the GNU compiler.

1. Connect and power up the Spartan-3E starter kit
2. Open a hyperterminal session (**Figure 1-22**)

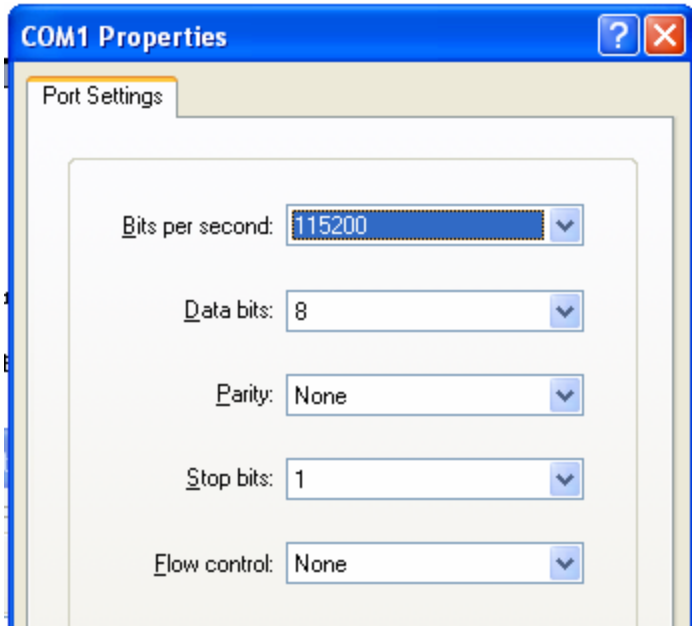

Figure 1-22. HyperTerminal Settings

❺ Select **Device Configuration → Download Bitstream** in XPS.

You should see the following output on hyperterminal

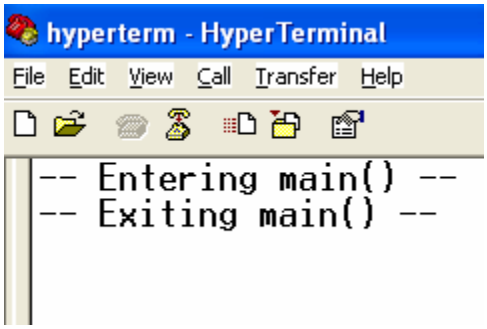

Figure 1-23. HyperTerminal Output

Conclusion

The Base System Builder can be used in XPS to quickly generate a MicroBlaze system and software test application. Several files—including an MHS file representing the processor system are created. A System Assembly View, representing the hardware system, provides hardware system parameters information. After the system has been defined, the netlist of the processor system can be created. You can verify hardware operation by downloading a bitstream (configured with test application) to the FPGA.

A Answers

1. List the bus connection to the following peripherals:

debug_module:	mb_plb
dlmb_cntrl:	dlmb
RS232_DCE:	mb_plb

2. List the nets which are connected to the following ports:

RS232_DCE – RX:	fpga_0_RS232_DCE_RX
RS232_DCE – TX:	fpga_0_RS232_DCE_TX
LEDs_8Bit - GPIO:	fpga_0_LEDs_8Bit_GPIO_d_out

3. Select Addresses filter and list the address for the following instances:

RS232_DCE – Base address:	0x84000000
RS232_DCE – High address:	0x8400ffff
LEDs_8Bit – Base address:	0x81400000
LEDs_8Bit – High address:	0x8140ffff
dlmb_cntlr – Base address:	0x00000000
dlmb_cntlr – High address:	0x00001fff
ilmb_cntlr – Base address:	0x00000000
ilmb_cntlr – High address:	0x00001fff

4. List the directories that were created.
   - __xps
   - blkdiagram
   - data
   - etc
   - hdl
   - implementation
   - pcores
   - microblaze_0
   - synthesis
   - TestApp_Memory

A Completed MHS File

```
# ##############################################################################
# Created by Base System Builder Wizard for Xilinx EDK 9.2.01 Build EDK_Jm_SP1.2
# Mon Dec 10 14:31:42 2007
# Target Board:  Xilinx Spartan-3E Starter Board Rev D
# Family:        spartan3e
# Device:        XC3S500e
# Package:       FG320
# Speed Grade:   -4
# Processor: microblaze_0
# System clock frequency: 50.00 MHz
# On Chip Memory :   8 KB
# ##############################################################################
 PARAMETER VERSION = 2.1.0


 PORT fpga_0_RS232_DCE_RX_pin = fpga_0_RS232_DCE_RX, DIR = I
 PORT fpga_0_RS232_DCE_TX_pin = fpga_0_RS232_DCE_TX, DIR = O
 PORT fpga_0_LEDs_8Bit_GPIO_d_out_pin = fpga_0_LEDs_8Bit_GPIO_d_out, DIR = O, VEC = [0:7]
 PORT sys_clk_pin = dcm_clk_s, DIR = I, SIGIS = CLK, CLK_FREQ = 50000000
 PORT sys_rst_pin = sys_rst_s, DIR = I, RST_POLARITY = 1, SIGIS = RST


BEGIN microblaze
 PARAMETER HW_VER = 7.00.a
 PARAMETER INSTANCE = microblaze_0
 PARAMETER C_INTERCONNECT = 1
 PARAMETER C_DEBUG_ENABLED = 1
 PARAMETER C_AREA_OPTIMIZED = 1
 BUS_INTERFACE DLMB = dlmb
 BUS_INTERFACE ILMB = ilmb
 BUS_INTERFACE DPLB = mb_plb
 BUS_INTERFACE IPLB = mb_plb
 BUS_INTERFACE DEBUG = microblaze_0_dbg
 PORT MB_RESET = mb_reset
END

BEGIN plb_v46
 PARAMETER INSTANCE = mb_plb
 PARAMETER HW_VER = 1.00.a
 PORT PLB_Clk = sys_clk_s
 PORT SYS_Rst = sys_bus_reset
END

BEGIN lmb_v10
 PARAMETER INSTANCE = ilmb
 PARAMETER HW_VER = 1.00.a
 PORT LMB_Clk = sys_clk_s
 PORT SYS_Rst = sys_bus_reset
END

BEGIN lmb_v10
 PARAMETER INSTANCE = dlmb
```

```
 PARAMETER HW_VER = 1.00.a
 PORT LMB_Clk = sys_clk_s
 PORT SYS_Rst = sys_bus_reset
END

BEGIN lmb_bram_if_cntlr
 PARAMETER INSTANCE = dlmb_cntlr
 PARAMETER HW_VER = 2.10.a
 PARAMETER C_BASEADDR = 0x00000000
 PARAMETER C_HIGHADDR = 0x00001fff
 BUS_INTERFACE SLMB = dlmb
 BUS_INTERFACE BRAM_PORT = dlmb_port
END

BEGIN lmb_bram_if_cntlr
 PARAMETER INSTANCE = ilmb_cntlr
 PARAMETER HW_VER = 2.10.a
 PARAMETER C_BASEADDR = 0x00000000
 PARAMETER C_HIGHADDR = 0x00001fff
 BUS_INTERFACE SLMB = ilmb
 BUS_INTERFACE BRAM_PORT = ilmb_port
END

BEGIN bram_block
 PARAMETER INSTANCE = lmb_bram
 PARAMETER HW_VER = 1.00.a
 BUS_INTERFACE PORTA = ilmb_port
 BUS_INTERFACE PORTB = dlmb_port
END

BEGIN xps_uartlite
 PARAMETER INSTANCE = RS232_DCE
 PARAMETER HW_VER = 1.00.a
 PARAMETER C_BAUDRATE = 115200
 PARAMETER C_DATA_BITS = 8
 PARAMETER C_ODD_PARITY = 0
 PARAMETER C_USE_PARITY = 0
 PARAMETER C_SPLB_CLK_FREQ_HZ = 50000000
 PARAMETER C_BASEADDR = 0x84000000
 PARAMETER C_HIGHADDR = 0x8400ffff
 BUS_INTERFACE SPLB = mb_plb
 PORT RX = fpga_0_RS232_DCE_RX
 PORT TX = fpga_0_RS232_DCE_TX
END

BEGIN xps_gpio
 PARAMETER INSTANCE = LEDs_8Bit
 PARAMETER HW_VER = 1.00.a
 PARAMETER C_GPIO_WIDTH = 8
 PARAMETER C_IS_DUAL = 0
 PARAMETER C_IS_BIDIR = 0
 PARAMETER C_ALL_INPUTS = 0
 PARAMETER C_BASEADDR = 0x81400000
 PARAMETER C_HIGHADDR = 0x8140ffff
 BUS_INTERFACE SPLB = mb_plb
 PORT GPIO_d_out = fpga_0_LEDs_8Bit_GPIO_d_out
```

```
END

BEGIN clock_generator
 PARAMETER INSTANCE = clock_generator_0
 PARAMETER HW_VER = 1.00.a
 PARAMETER C_EXT_RESET_HIGH = 1
 PARAMETER C_CLKIN_FREQ = 50000000
 PARAMETER C_CLKOUT0_FREQ = 50000000
 PARAMETER C_CLKOUT0_PHASE = 0
 PARAMETER C_CLKOUT0_GROUP = NONE
 PORT CLKOUT0 = sys_clk_s
 PORT CLKIN = dcm_clk_s
 PORT LOCKED = Dcm_all_locked
 PORT RST = net_gnd
END

BEGIN mdm
 PARAMETER INSTANCE = debug_module
 PARAMETER HW_VER = 1.00.a
 PARAMETER C_MB_DBG_PORTS = 1
 PARAMETER C_USE_UART = 1
 PARAMETER C_UART_WIDTH = 8
 PARAMETER C_BASEADDR = 0x84400000
 PARAMETER C_HIGHADDR = 0x8440ffff
 BUS_INTERFACE SPLB = mb_plb
 BUS_INTERFACE MBDEBUG_0 = microblaze_0_dbg
 PORT Debug_SYS_Rst = Debug_SYS_Rst
END

BEGIN proc_sys_reset
 PARAMETER INSTANCE = proc_sys_reset_0
 PARAMETER HW_VER = 2.00.a
 PARAMETER C_EXT_RESET_HIGH = 1
 PORT Slowest_sync_clk = sys_clk_s
 PORT Dcm_locked = Dcm_all_locked
 PORT Ext_Reset_In = sys_rst_s
 PORT MB_Reset = mb_reset
 PORT Bus_Struct_Reset = sys_bus_reset
 PORT MB_Debug_Sys_Rst = Debug_SYS_Rst
END
```