Spartan-3A DSP 3SD1800A MicroBlaze Processor Edition Kit Reference Systems

UG486 (v1.0) February 12, 2008

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
2/12/08	1.0	Initial Xilinx release.

Table of Contents

Preface: About This Guide

Guide Contents 7
References 7
Additional Resources 8
Conventions 8
Typographical 8
Online Document 9

Chapter 1: Hardware Platform

Introduction 11
Block Diagram 11
Address Map 12
System Configuration 13
MicroBlaze Processor Configuration 13
XPS EthernetLite Configuration 13
XPS MCH EMC Configuration 13
XPS SPI Configuration 13
XPS UART Lite Configuration 13
Flash IO Multiplexer 13

Chapter 2: HelloWorld Software Application

Introduction 15
Executing the HelloWorld Software Application 15
Executing the HelloWorld Application Using the Pre-Built Bitstream 15
Executing the HelloWorld Software Application from XPS 17
Tests in the HelloWorld Software Application 18
Booting the HelloWorld Application from Serial Flash 18

Chapter 3: LynuxWorks BlueCat Linux

Introduction 19
Executing the BlueCat Linux Image 19
Executing the BlueCat Linux Image Using the Pre-Built Bitstream 19
Executing the BlueCat Linux Image from XPS 21
Executing BlueCat Linux Commands 22
Building the BlueCat Linux Kernel Image 23
Installing the BlueCat Linux Distribution 24
Getting the MLD File Set 24
Generating the BSP 25
Rebuilding the Kernel Image 26
Booting the BlueCat Linux Image from Parallel Flash 27

About This Guide

The Embedded Development HW/SW Kit - Spartan-3A DSP S3D1800A MicroBlaze Processor Edition showcases various features of the Spartan-3A DSP 1800A development board. This kit includes a reference system with a HelloWorld software application and a bootable BlueCat Linux image. This document describes the hardware platform, the HelloWorld software application, and the BlueCat Linux image.

The reference sytem is available at

www.xilinx.com/support/documentation/user_guides/ug486.zip

Guide Contents

This manual contains the following chapters:

- Chapter 1, "Hardware Platform," provides an overview of the IP cores in the reference system. This chapter includes the reference system block diagram and address map.
- Chapter 2, "HelloWorld Software Application," describes the board tests in the application, how to execute the application, and how to boot the application from SPI Flash.
- Chapter 3, "LynuxWorks BlueCat Linux," includes information on how to execute the provided BlueCat Linux image and how to build a similar image using the BlueCat Linux development tools.

References

References used throughout this user guide are listed below.

1. *BlueCat Linux Users Guide*
2. *BlueCat Linux Board Support Guide for Xilinx Spartan-3E 1600E Boards*
3. UG485 *Getting Started with the Spartan-3A DSP 1800A Starter Platform User Guide*
4. XAPP963 *Using and Creating Flash Files for the MicroBlaze Development Kit - Spartan-3E Edition*
5. XAPP1053 *Flash Memory Bootloading Using SPI with Spartan-3A DSP 1800A Starter Platform*

Additional Resources

To find additional documentation, see the Xilinx website at:

http://www.xilinx.com/literature.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

http://www.xilinx.com/support.

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*
Helvetica bold	Commands that you select from a menu	**File → Open**
	Keyboard shortcuts	**Ctrl+C**
Italic font	Variables in a syntax statement for which you must supply values	**`ngdbuild`** *design_name*
	References to other manuals	See the *Development System Reference Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **`bus[7:0]`**, they are required.	**`ngdbuild`** [*`option_name`*] *`design_name`*
Braces { }	A list of items from which you must choose one or more	**`lowpwr =`**{**`on`**|**`off`**}
Vertical bar |	Separates items in a list of choices	**`lowpwr =`**{**`on`**|**`off`**}

Convention	Meaning or Use	Example

Vertical ellipsis
.
.
.
	Repetitive material that has been omitted	`IOB #1: Name = QOUT'`
		`IOB #2: Name = CLKIN'`
		.
		.
		.

Horizontal ellipsis ...	Repetitive material that has been omitted	**`allow block`** *block_name loc1 loc2 ... locn;*

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Red text	Cross-reference link to a location in another document	See Figure 2-5 in the *Virtex-II Platform FPGA User Guide.*
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

Chapter 1

Hardware Platform

Introduction

This reference system targets the Spartan-3A DSP 1800A development board. The system is created to run the HelloWorld software application described in Chapter 2, "HelloWorld Software Application." and the BlueCat Linux image described in Chapter 3, "LynuxWorks BlueCat Linux.". The system uses the MicroBlaze processor with cache turned on for both the instruction cache and the data cache. As shown in Figure 1-1, the system also includes the MPMC memory controller, the XPS EthernetLite, the XPS MCH EMC memory controller, and the XPS UART Lite IP cores. An XPS Timer and XPS Interrupt Controller are also needed for the BlueCat linux kernel.

See Table 1-1 for the address map of the system.

Block Diagram

The block diagram for the system is shown in Figure 1-1.

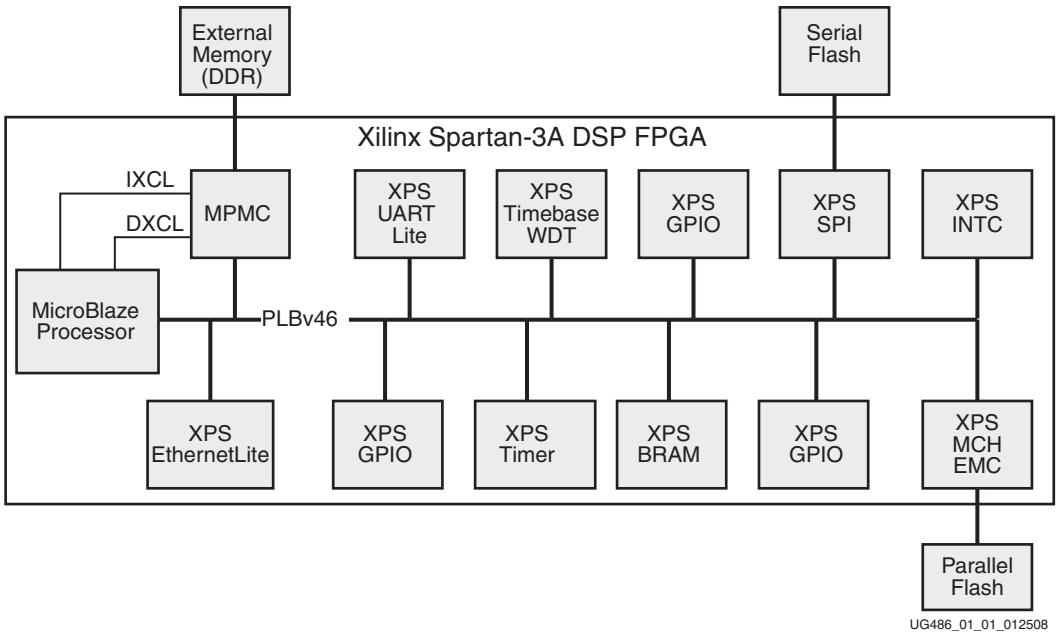

Figure 1-1: **Block Diagram**

Address Map

The address map for the IP cores in the reference system is given in Table 1-1.

Table 1-1: **Reference System Address Map**

Instance	Peripheral	Base Address	High Address
dlmb_cntlr	lmb_bram_if_cntlr	0x00000000	0x00001FFF
ilmb_cntlr	lmb_bram_if_cntlr	0x00000000	0x00001FFF
debug_module	mdm	0x84400000	0x8440FFFF
xps_bram_if_cntlr_1	xps_bram	0x41A08000	0x41A0FFFF
Ethernet_MAC	xps_ethernetlite	0x81000000	0x8100FFFF
Push_Buttons	xps_gpio	0x81400000	0x8140FFFF
LEDs_8Bit	xps_gpio	0x81420000	0x8142FFFF
DIP_Switches_8Bit	xps_gpio	0x81440000	0x8144FFFF
xps_intc_0	xps_intc	0x81800000	0x8180FFFF
SPI_FLASH	xps_spi	0x83400000	0x8340FFFF
xps_timebase_wdt_1	xps_timebase_wdt	0x41300000	0x4130FFFF
xps_timer_1	xps_timer	0x83C00000	0x83C0FFFF
RS232_Uart_1	xps_uartlite	0x84000000	0x8400FFFF
FLASH	xps_mch_emc	0x87000000	0x87FFFFFF
DDR2_SDRAM	mpmc	0x88000000	0x8FFFFFFF

System Configuration

This system runs off a reference clock frequency of 125 MHz from the oscillator on the board. The PLBv46 bus and MicroBlaze processor run at 62.5 Mhz, while the DDR2 memory runs at 125 MHz.

MicroBlaze Processor Configuration

The MicroBlaze processor is configured with the Memory Management Unit (MMU) enabled. The MMU is enabled by setting the MicroBlaze parameter C_USE_MMU to 3. This parameter implements the MMU in Virtual mode. In Virtual mode, the MMU controls effective-address to physical-address mapping and supports memory protection. Virtual mode provides greater control over memory protection. Protection and relocation enable system software to support multitasking. This capability gives the appearance of simultaneous or near-simultaneous execution of multiple programs.

The instruction cache and data cache are both enabled, with a cache size of 4KB. The cacheable block of main memory is accessed via the XCL Port Interface Modules (PIM) of the Multi-Port Memory Controller (MPMC).

More information about the MMU, the instruction cache, and the data cache, can be found in the *MicroBlaze Processor Reference Guide*.

XPS EthernetLite Configuration

The BlueCat Linux RTOS requires that the XPS EthernetLite has the interrupts set to on. In the BlueCat Linux demonstration, the Ethernet MAC can run at 10 Mb/s or 100 Mb/s, depending on the attached network. No other special settings are needed.

XPS MCH EMC Configuration

The XPS MCH EMC memory controller is connected to an external Intel J3 Parallel Flash device, which is used to store the hardware configuration bitstream and bootloader application, as well as the BlueCat Linux kernel image.

XPS SPI Configuration

The XPS SPI core is connected to an external Intel S33 Serial Flash device, which is used to store the hardware bitstream with the HelloWorld software application in BRAM.

XPS UART Lite Configuration

The XPS UART Lite core is configured to use interrupts and is set to a baud rate of 115200, 8 data bits, and no parity.

Flash IO Multiplexer

In the Spartan-3A DSP 1800A development board, the SPI Flash (XPS SPI) MISO pin and the MSB of the Parallel Flash (XPS MCH EMC) data pin are multiplexed on the board. The Flash IO Multiplexer is a custom pcore that is used to select between the Parallel Flash data (DQ) signals and the SPI Flash data (MISO) signals based on Memory Chip Enable (Active Low) of the Parallel Flash. If Memory Chip Enable is Low, then the Parallel Flash data signals are sent to the external connection, else the SPI data signals are sent to the external data pin connections.

Chapter 2

HelloWorld Software Application

Introduction

The HelloWorld software application is a simple application that exercises a few of the board features. When the application is run, it will first flash the LEDs and read the DIP and push button switches. Then, the user can select from a list of menu options, including options to allow the user to select a target memory and read/write an address with necessary data.

The methods for downloading and running the HelloWorld software application are listed below:

- One method is to use a debugger, such as XMD (provided as part of the EDK tools), and downloading the executable file directly into BRAM, through the MicroBlaze Debug Module (MDM). This method is described in the section "Executing the HelloWorld Software Application".
- Another method is to program Flash memory with the HelloWorld software application. This method is described in the section "Booting the HelloWorld Application from Serial Flash". Once Flash memory is programmed, the HelloWorld software application can be run by setting the FPGA configuration mode pins to SPI mode and either powering up the development board or depressing the PROG button on the board.

Executing the HelloWorld Software Application

To execute the HelloWorld software application, the hardware bitstream must be programmed to the Spartan-3A DSP device and the HelloWorld software application loaded into BRAM. Programming the bitstream can be done by either downloading the pre-built bitstream from the `ready_for_download` directory or generating and downloading it from XPS. Similarly, the HelloWorld executable can be downloaded from the `ready_for_download` directory or built through XPS.

Executing the HelloWorld Application Using the Pre-Built Bitstream

To execute the application using the files inside the `ready_for_download` directory in the project root directory, follow these steps:

1. Connect the Platform USB cable or the Parallel IV JTAG cable between the host computer and the Spartan-3A DSP 1800A Starter Board.
2. Connect the serial cable between the host computer and the RS232 port on the Spartan-3A DSP 1800A Starter Board.
3. Apply power to the Spartan-3A DSP 1800A Starter Board.

4. Start a HyperTerminal (or similar) session on the host computer with the settings shown in Figure. Select the COM port corresponding to the connected serial port on the host computer. Set the Baud Rate to **115200**, Data bits to **8** bits, Parity to **None**, Stop bits to **1** bit, and Flow control to **None**, as shown in Figure 2-1.

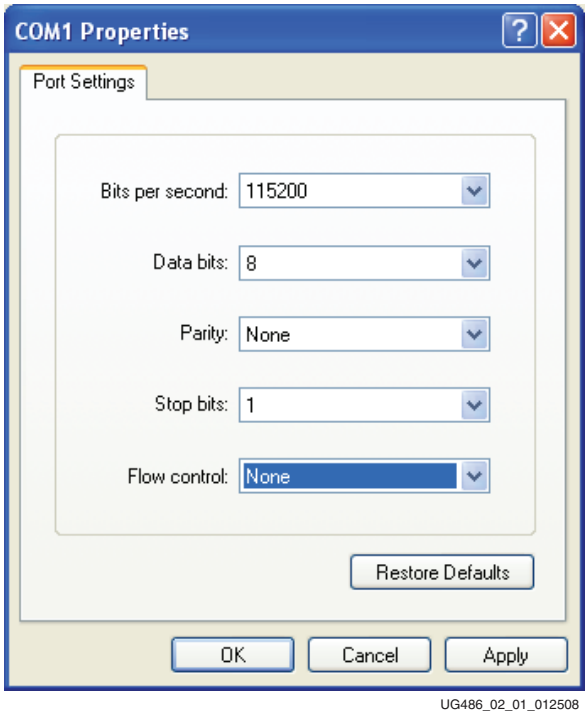

Figure 2-1: **HyperTerminal Settings**

5. In an EDK shell, change directories to the `ready_for_download` directory.
6. Use iMPACT to download the bitstream by using the following command:

   ```
   impact -batch ug486.cmd
   ```

7. Invoke XMD and connect to the processor by the following command:

   ```
   xmd -opt ug486.opt
   ```

8. Download the HelloWorld software application into BRAM using the following command:

   ```
   dow helloworld_executable.elf
   ```

9. To start the HelloWorld software application running, use the following XMD command:

   run

   a. After the HelloWorld software application runs, the HyperTerminal output will be as shown in Figure 2-2.

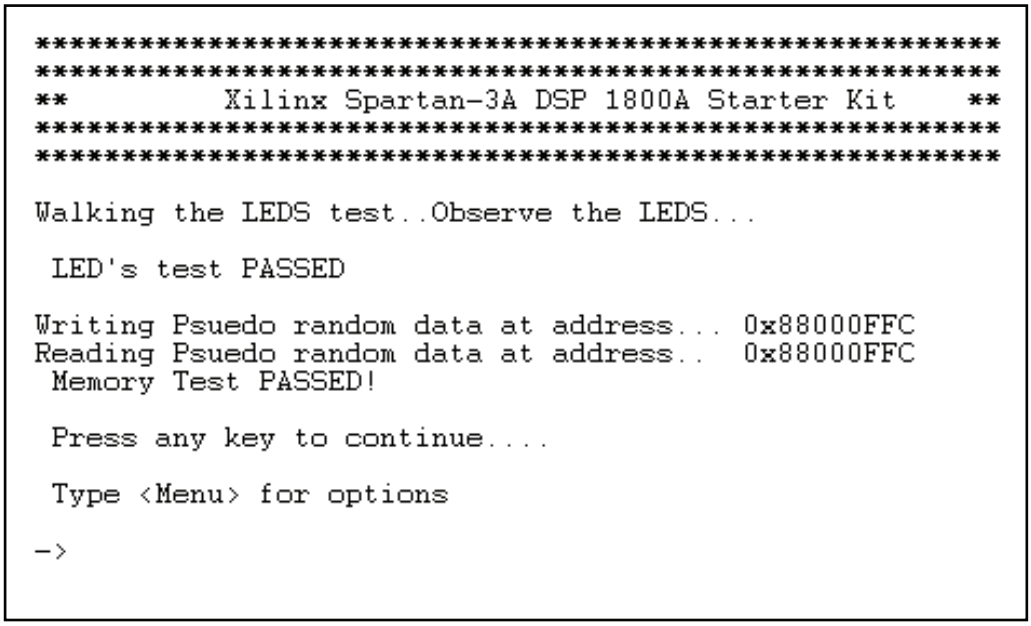

Figure 2-2: **HelloWorld Output**

   b. For an explanation of the tests in the application, see the section "Tests in the HelloWorld Software Application".

Executing the HelloWorld Software Application from XPS

To execute the reference system using XPS, follow these steps:

1. Perform steps 1-4 in the "Executing the HelloWorld Application Using the Pre-Built Bitstream" section.
2. Open the reference system project in XPS.
3. In the Applications tab, select the **helloworld** project for BRAM initialization by right-clicking on the project and selecting **Mark to Initialize BRAMs**. Ensure that no other applications are marked for BRAM initialization.
4. Implement the hardware design and create the hardware bitstream by selecting **Hardware** → **Generate Bitstream** in XPS.
5. Download the bitstream to the board by selecting **Device Configuration** → **Download Bitstream** in XPS. After the bitstream has downloaded, the HelloWorld application will execute from BRAM.
   a. After the HelloWorld software application runs, the HyperTerminal output will be as shown in Figure 2-2.

b. For an explanation of the tests in the application, see the section "Tests in the HelloWorld Software Application".

Tests in the HelloWorld Software Application

After the HelloWorld application is executed, type **`Menu`** into the terminal console to bring up the HelloWorld menu of tests, which is shown in Figure 2-3.

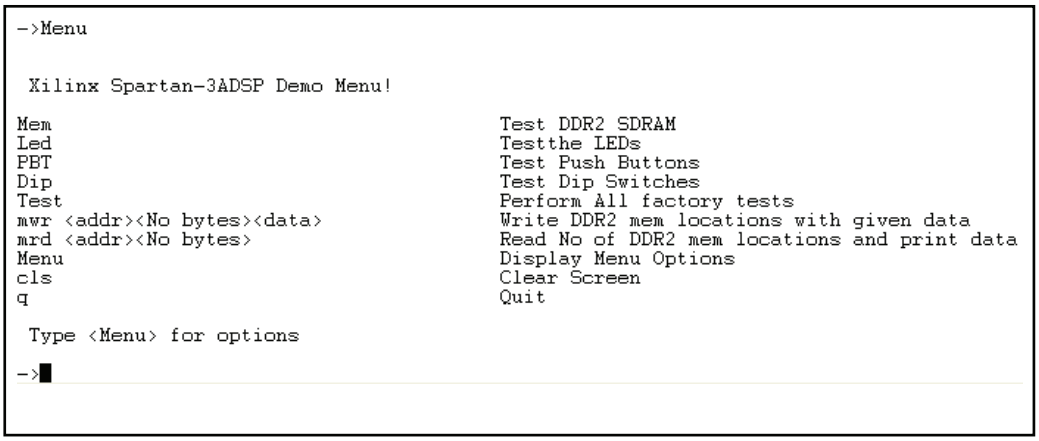

Figure 2-3: **HelloWorld Menu**

Booting the HelloWorld Application from Serial Flash

This section includes steps on how to program the HelloWorld application into the SPI Flash. These steps refer to XAPP1053, Flash Memory Bootloading Using SPI with Spartan-3A DSP 1800A Starter Platform, which includes details on how to use, create, and boot SPI Flash files for the MicroBlaze Spartan-3A DSP 1800A Edition Development Kit.

Flash files that have already been generated and are ready to use can be found in the `<project root directory>/ready_for_download/Flash_files/` directory.

1. Open the reference system project in XPS.
2. In the Applications tab, select the **helloworld** project for BRAM initialization by right-clicking on the project and selecting **Mark to Initialize BRAMs**. Ensure that no other applications are marked for BRAM initialization.
3. Select **Device Configuration** → **Update Bitstream**. This will generate the MicroBlaze hardware platform, build the Board Support Package (BSP), compile the project, and initialize the bitstream with the HelloWorld application.
4. Follow the steps outlined in the "Configure the Serial Flash" section in XAPP1053.

   Note: Make sure to edit the `burn_intel_s33.bat` file to point to the download.bit generated in this reference system project.

Chapter 3

LynuxWorks BlueCat Linux

Introduction

The BlueCat Linux reference system demonstrates BlueCat Linux running on the MicroBlaze soft processor with the MMU enabled. An example BlueCat Linux image is provided that is tailored to the Spartan-3A DSP 1800A Edition Devlopment Kit board and the hardware platform that is described in Chapter 1, "Hardware Platform.". The kernel and file system are downloaded into the DDR2 memory and run completely out of the externel memory.

The methods for downloading and running the BlueCat Linux kernel demonstration are listed below.

- One method is to use a debugger, such as XMD (provided as part of the EDK tools), and download the image file directly into DDR2, through the MicroBlaze Debug Module (MDM). This method is described in the section "Executing the BlueCat Linux Image".
- Another method is to program Flash memory with the BlueCat Linux image. This method is described in the section "Booting the BlueCat Linux Image from Parallel Flash". Once Flash memory is programmed, the BlueCat Linux demonstration can be run by setting the FPGA configuration mode pins to BPI mode and either powering up the development board or depressing the PROG button on the board.

Executing the BlueCat Linux Image

To execute the BlueCat Linux reference system, the hardware bitstream must be programmed to the Spartan-3A DSP device and the BlueCat Linux kernel image must be downloaded to the DDR2 memory. Programming the bitstream can be done by either downloading the pre-built bitstream from the `ready_for_download` directory or generating and downloading it from XPS. The BlueCat Linux kernel image is downloaded from the `bclinux_images` directory.

Executing the BlueCat Linux Image Using the Pre-Built Bitstream

To execute the reference system using the files inside the `ready_for_download` directory in the project root directory, follow these steps:

1. Connect the Platform USB cable or the Parallel IV JTAG cable between the host computer and the Spartan-3A DSP 1800A Starter Board.
2. Connect the serial cable between the host computer and the RS232 port on the Spartan-3A DSP 1800A Starter Board.
3. Apply power to the Spartan-3A DSP 1800A Starter Board.

4. Start a HyperTerminal (or similar) session on the host computer with the settings shown in Figure. Select the COM port corresponding to the connected serial port on the host computer. Set the Baud Rate to **115200**, Data bits to **8** bits, Parity to **None**, Stop bits to **1** bit, and Flow control to **None**, as shown in Figure 3-1.

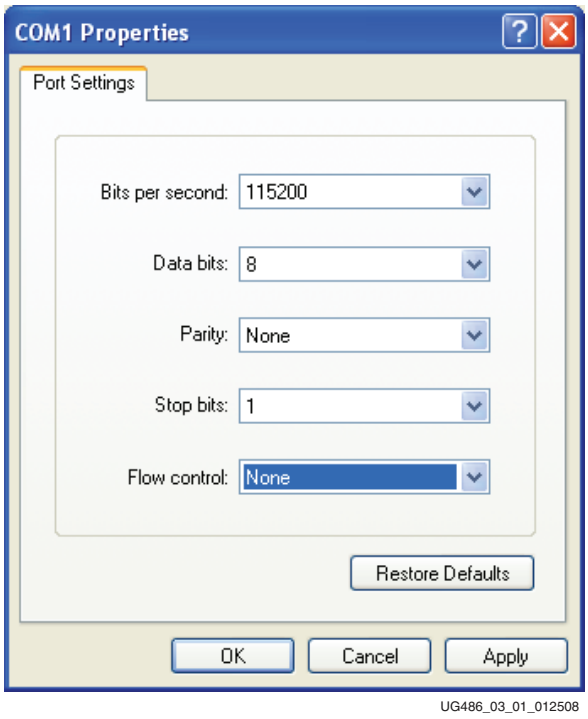

Figure 3-1: **HyperTerminal Settings**

5. In an EDK shell, change directories to the `ready_for_download` directory.
6. Use iMPACT to download the bitstream by using the following command:

   ```
   impact -batch ug486.cmd
   ```

7. Invoke XMD and connect to the processor by the following command:

   ```
   xmd -opt ug486.opt
   ```

8. Download the BlueCat Linux kernel image into DDR2 memory at the starting location `0x88000000` using the following command:

   ```
   dow -data ../bclinux_images/bclinux.kdi 0x88000000
   ```

   Note: This step may take several minutes to download the BlueCat Linux image into memory.

9. To start the kernel image running and boot BlueCat Linux, use the following XMD command:

   ```
   con 0x88000000
   ```

a. After BlueCat Linux boots, the HyperTerminal output will be as shown in Figure 3-2.

```
Linux version 2.6.13.4 (caseyc@linrock6) (gcc version 4.1.1) #1 Wed Jan 23 19:13:12 EST 2008
On node 0 totalpages: 32768
  DMA zone: 32768 pages, LIFO batch:15
  Normal zone: 0 pages, LIFO batch:1
  HighMem zone: 0 pages, LIFO batch:1
Built 1 zonelists
Kernel command line: ramdisk_size=28472 hda=bswap hdb=bswap hdc=bswap hdd=bswap root=101
xps_intc_1.00.a INTC at 0x81800000 mapped to 0xFDFFF000
PID hash table entries: 1024 (order: 10, 16384 bytes)
xps_timer_1.00.a TIMER at 0x83C00000 mapped to 0xFDFFE000
Console: Xilinx OPB UART Lite
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 124032k available
Calibrating delay loop... 30.82 BogoMIPS (lpj=154112)
Mount-cache hash table entries: 512
NET: Registered protocol family 16
xgpio00 #0 at 0x81420000 mapped to 0xC8000000 device: 10,185 not using IRQ

xgpio01 #1 at 0x81400000 mapped to 0xC8020000 device: 10,186 using IRQ#5

xgpio02 #2 at 0x81440000 mapped to 0xC8040000 device: 10,187 using IRQ#4

ttyS0 at MMIO 0x84000000 (irq = 3) is a Xilinx OPB UART Lite
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered
RAMDISK driver initialized: 16 RAM disks of 28472K size 1024 blocksize
eth0: using fifo mode.
eth0: No PHY detected.  Assuming a PHY at address 0.
eth0: Xilinx EMACLite #0 at 0x81000000 mapped to 0xC8060000, irq=2
EMC Flash on Xilinx board: Found 1 x16 devices at 0x0 in 8-bit bank
 Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
cfi_cmdset_0001: Erase suspend on write enabled
0: offset=0x0,size=0x20000,blocks=128
Registering a 16MB EMC Flash at 0x87000000
NET: Registered protocol family 2
IP route cache hash table entries: 2048 (order: 1, 8192 bytes)
TCP established hash table entries: 8192 (order: 4, 65536 bytes)
TCP bind hash table entries: 8192 (order: 3, 32768 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
TCP reno registered
TCP bic registered
NET: Registered protocol family 1
RAMDISK: Compressed image found at block 2229636
Freeing BlueCat RFS memory: 4191k freed
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 73k freed

INIT: version 2.85 booting


INIT: Entering runlevel: 1


myhostname login:
```

UG486_03_02_012508

Figure 3-2: **BlueCat Linux Boot Output**

b. Log into BlueCat Linux by using the username **`root`**.

c. For example commands to run in BlueCat Linux, see the section "Executing BlueCat Linux Commands".

Executing the BlueCat Linux Image from XPS

To execute the reference system using XPS, follow these steps:

1. Perform steps 1-4 in the "Executing the BlueCat Linux Image Using the Pre-Built Bitstream" section.
2. Open the reference system project in XPS.

3. Implement the hardware design and create the hardware bitstream by selecting **Hardware** → **Generate Bitstream** in XPS.
4. Download the bitstream to the board by selecting **Device Configuration** → **Download Bitstream** in XPS.
5. Select **Debug** → **Launch XMD...** to launch an XMD command window.
6. In XMD, download the BlueCat Linux kernel image into DDR2 memory at the starting location `0x88000000` using the following command:

   ```
   dow -data bclinux_images/bclinux.kdi 0x88000000
   ```

   Note: This step may take several minutes to download the BlueCat Linux image into memory.
7. To start the kernel image running and boot BlueCat Linux, use the following XMD command:

   ```
   con 0x88000000
   ```

   a. After BlueCat Linux boots, the HyperTerminal output will be as shown in Figure 3-2.
   b. Log into BlueCat Linux by using the username **`root`**.
   c. For example commands to run in BlueCat Linux, see the section "Executing BlueCat Linux Commands".

Executing BlueCat Linux Commands

This build of BlueCat Linux supports many basic Linux commands. The list of commands and tools available to be run are found under the `/bin` directory.

This BlueCat Linux kernel was built with networking support enabled, therefore it supports several network utilities when connected to a live network or connected directly to a remote computer.

To view the ethernet configuration settings, use the command ifconfig. Example results of using this command for the eth0 (Ethernet) and lo (Local Loopback) ports are shown in Figure 3-3. In the figure, the board IP address is 172.17.1.218. The board IP address can be changed by issuing the command **`ifconfig eth0`** *IP_address*.

```
-bash-3.00# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:C0:A3:E5:44
          inet addr:172.17.1.218  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:188 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:285384 (278.6 KiB)  TX bytes:0 (0.0 b)
          Interrupt:2

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

UG486_03_03_012508

Figure 3-3: **Ethernet Configuration Settings**

To ping a remote computer at IP address 172.17.1.200 from the development board, this example command string, **`ping -c 172.17.1.200,`** is used to ping the remote computer 4 times.

To telnet from a networked computer to the board, issue the command **`telnet`** *board_IP_address*. All of the Linux commands can now be performed remotelyas if the user was logged into the console on a HyperTerminal.

Building the BlueCat Linux Kernel Image

This section briefly describes the process for rebuilding the kernel image that is included with this reference system. To rebuild the kernel, the BlueCat Linux distribution must be obtained from LynuxWorks. For more information on the LynuxWorks BlueCat Linux distribution, see the BlueCat Linux User's Guide.

The steps described in this section include using the BlueCat Linux Spartan-3E BSP. Even though this reference system is for Spartan-3A, the Spartan-3E BSP contains the needed components to build a working image for the Spartan-3A. For more information on the Spartan-3E BSP, see the BlueCat Linux Board Support Guide for Xilinx Spartan-3E 1600E Boards.

Both the *BlueCat Linux User's Guide* and the *BlueCat Linux Board Support Guide for Xilinx Spartan-3E 1600E Boards* can be obtained from LynuxWorks at:
http://www.lynuxworks.com/support/bluecat/docs.php3

These steps are specifically written for BlueCat Linux Release 5.4.2. These steps assume the kernel is being built on a host system running Red Hat Enterprise Linux 4.0. All of the Linux commands must be run using a bash shell.

Installing the BlueCat Linux Distribution

These steps describe how to install the BlueCat Linux 5.4.2 distribution with the sp3e BSP. For more information on the directory structures of the LynuxWorks BlueCat Linux distribution and the installation procedures, see the *BlueCat Linux User's Guide* reference above.

1. To install the BlueCat Linux core components on the host machine, follow the steps outlined in the "Installing the Default Configuration" section in the Introduction and Installation chapter of the *BlueCat Linux User's Guide*.
2. To install the SP3E BSP on the host machine, follow the steps outlined in the "Installing Target Board Support" section in the Introduction and Installation chapter of the *BlueCat Linux User's Guide*.
3. After the SP3E BSP is installed, support for it must be activated in the bash shell. To activate the SP3E BSP, follow the steps in the "Activating Support for a Target Board" section in the Introduction and Installation chapter of the *BlueCat Linux User's Guide*.

Getting the MLD File Set

The MLD file set is included with the BlueCat Linux distribution. It is necessary to get the MLD set so that the BlueCat Linux kernel source tree can be updated by the EDK tools. The MLD file set is located in the `$BLUECAT_PREFIX/boot` directory and is contained in the `edk_user_repository.tar.gz` tar file. This tar file should be unpacked on the same directory level as the EDK installation, as shown in the following commands:

```
BlueCat:$ cd EDK_installation_directory
BlueCat:$ cd ..
BlueCat:$ tar xfz $BLUECAT_PREFIX/boot/edk_user_repository.tar.gz
```

Generating the BSP

With the use of the BlueCat Linux MLD, EDK can update the BlueCat Linux kernel source tree to match a specific hardware configuration. Follow these steps to generate the BSP and update the BlueCat Linux kernel source tree.

1. Open the reference system in XPS.
2. Select **Software** → **Software Platform Settings...** under XPS.
3. In the Software Platform Settings window, select **linux_bc54** in the OS field, as shown in Figure 3-4.

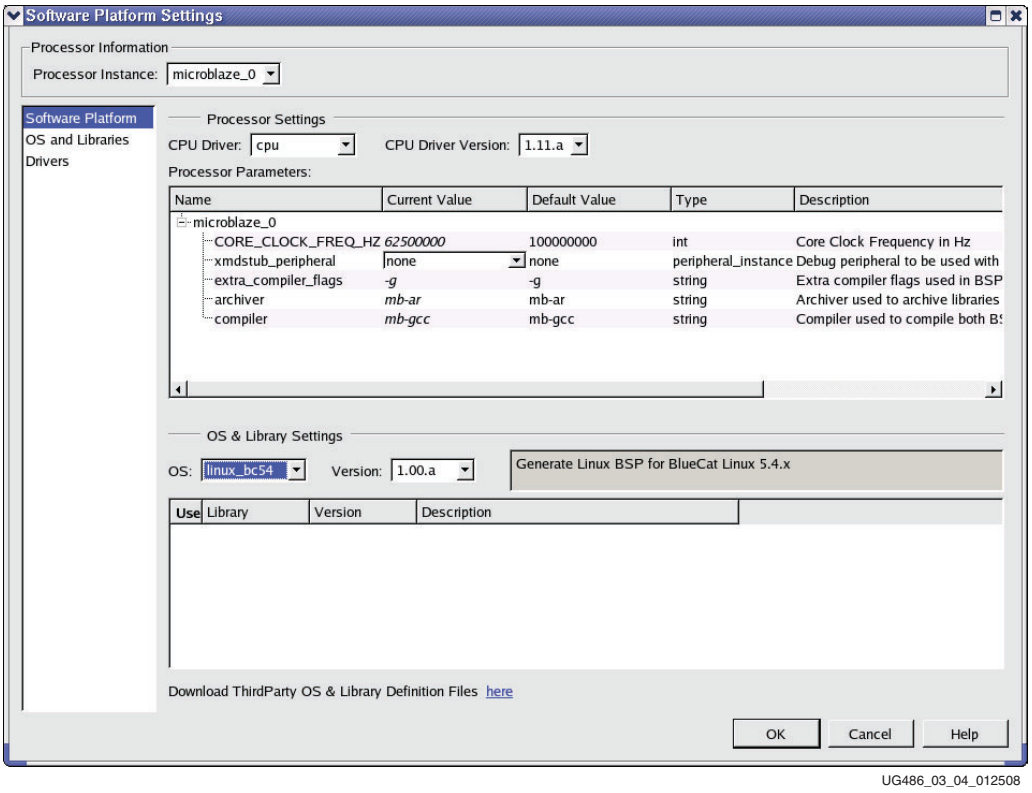

Figure 3-4: **Select BlueCat Linux for the OS**

4. Select the OS and Libraries option on the left of the Software Platform Settings window. Fill in the fields as follows:

   BLUECAT_PREFIX:

   <BlueCat Linux installation point>/usr/src/linux.sp3e

   KERNEL_CONFIG:

   <BlueCat Linux installation point>/demo/developer.sp3e/developer.config

An example showing these fields is in Figure 3-5.

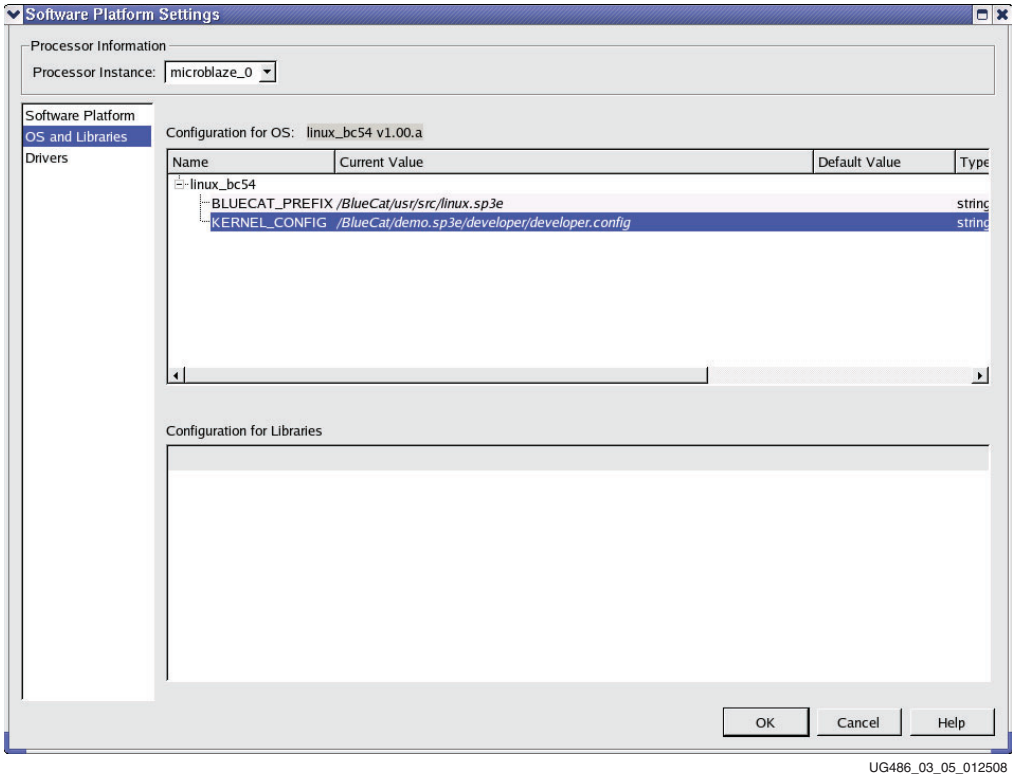

Figure 3-5: **Set the BlueCat Linux Paths**

5. Click **OK** to save the changes and close the Software Platform Settings window.
6. In XPS, select **Software** → **Generate Libraries and BSPs**. This will update the BlueCat Linux kernel source tree.

Rebuilding the Kernel Image

This is the final step to create a bootable BlueCat Linux kernel image. To recreate the image provided with this reference system, follow these steps:

1. To force all kernel components to rebuild, clean the kernel tree using the following commands:

```
BlueCat:$ cd $BLUECAT_PREFIX/usr/src/linux.sp3e
BlueCat:$ make mrproper
```

2. Navigate to the developer demo directory.

```
BlueCat:$ cd $BLUECAT_PREFIX/demo.sp3e/developer
```

3. Clean any prebuild image files.

```
BlueCat:$ make clean
```

4. Build the kernel, root filesystem, and bootable image file.

```
BlueCat:$ make all
```

Note: For this demonstration, when options come up for Journalling Flash File System support, type in N to exclude those options.

This command produces a .kdi file which is the BlueCat Linux image and is composed of a compressed kernel image and a compressed RAM disk root file system. The image will be stored in `$BLUECAT_PREFIX/demo.sp3e/developer/developer.kdi`.

5. To run the newly created kernel image, refer to the steps in the section "Executing the BlueCat Linux Image". When downloading the kernel image through XMD into DDR2 memory, put in the path to the new kernel image instead of the path to the pre-built kernel image.

Booting the BlueCat Linux Image from Parallel Flash

To boot the BlueCat Linux image from parallel Flash, the Linux image and a bootloader application must be programmed into Flash. The bootloader application copies the Linux image from Flash to DDR2 memory and boots BlueCat Linux. These are the steps to program the BlueCat Linux image into parallel Flash.

1. Follow the steps as mentioned in the section "Programming the BlueCat Linux Image KDI File into StrataFlash and Creating New Bootloader Files" in XAPP963. These steps will provide information on how to program the Flash with the Linux image. Modify the addresses of the DDR2 and Flash memory according to the system.
2. In XPS, compile a bitstream, `download.bit`, that includes the system configuration and the bootloader application. This is done by marking the bootloader application to Initialize BRAM and then selecting **Device Configuration** → **Update Bitstream** in XPS.
3. In the FLASH_BURN directory, edit the make_bpi_up.bat file to point to this new bitstream, then use the batch file to create a binary version of the bootloader bitstream. With the XPS Flash programmer, program the binary bootloader bitstream to offset `0x00000000` of the Flash memory.
4. Set the jumpers on the Spartan-3A DSP 1800A board to BPI mode (M0 and M2 closed). This will boot the FPGA in BPI mode.