Virtex-4 FX12 PowerPC and MicroBlaze Edition Kit Reference Systems

UG494 (v1.1) May 29, 2008
Xilinx is disclosing this user guide, manual, release note, and/or specification (the “Documentation”) to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© 2008 Xilinx, Inc. All rights reserved.

XILINX, the Xilinx logo, the Brand Window, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.
## Revision History

The following table shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/11/08</td>
<td>1.0</td>
<td>Initial Xilinx release.</td>
</tr>
<tr>
<td>5/29/08</td>
<td>1.1</td>
<td>Updated to EDK 10.1</td>
</tr>
</tbody>
</table>
Preface: About This Guide

Chapter 1: Hardware Platform

Chapter 2: HelloWorld Software Application

Chapter 3: LynuxWorks BlueCat Linux
Chapter 4: MontaVista Linux

Introduction .................................................................................. 31

Executing the MontaVista Linux Image ........................................ 31
- Executing the MontaVista Linux Image Using the Pre-Built Bitstream .................................................................................. 31
- Executing the MontaVista Linux Image from XPS ......................... 34

Executing MontaVista Linux Commands ..................................... 34

Building the MontaVista Linux Image ........................................ 35
- Installing and Setting up the MontaVista Linux Development Environment .......................................................... 35
- Generating the BSP ..................................................................... 35
- Rebuilding the Kernel Image ......................................................... 37

Booting the MontaVista Linux Image from Parallel Flash ............. 38
Preface

About This Guide

The Embedded Development HW/SW Kit - Virtex®-4 FX12 PowerPC® and MicroBlaze™ Processor Edition showcases various features of the Virtex-4 FX12 ML403 development board. This kit includes a reference system with a HelloWorld software application, bootable BlueCat Linux image, and a bootable MontaVista Linux image. This document describes the hardware platform, the HelloWorld software application, the BlueCat Linux image, and the MontaVista Linux image.

The reference system is available at

www.xilinx.com/support/documentation/user_guides/ml403_emb_ref_mb.zip
www.xilinx.com/support/documentation/user_guides/ml403_emb_ref_ppc.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.
• Chapter 4, “MontaVista Linux,” includes information on how to execute the provided MontaVista Linux image and how to build a similar image using the MontaVista Linux development tools.

References

References used throughout this user guide are listed below.

1. BlueCat Linux User’s Guide
2. BlueCat Linux Board Support Guide for Xilinx Spartan®-3E 1600E Boards
3. UG083 Getting Started Tutorial for ML401/ML402/ML403/ML405 Evaluation Platforms
4. XAPP963 Using and Creating Flash Files for the MicroBlaze Development Kit - Spartan-3E Edition
5. XAPP969 Getting Started with EDK and Linux 2.6
Additional Resources

To find additional documentation, see the Xilinx website at:


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:


Conventions

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

Typographical

The following typographical conventions are used in this document:

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

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning or Use</th>
<th>Example</th>
</tr>
</thead>
</table>
| Vertical ellipsis   | Repetitive material that has been omitted              | IOB #1: Name = QOUT’  
| .                   | .                                                       | .  
| .                   | .                                                       | .  
| 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:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning or Use</th>
<th>Example</th>
</tr>
</thead>
</table>
| 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](http://www.xilinx.com) for the latest speed files. |
Chapter 1

Hardware Platform

Introduction

The Virtex-4 FX Development Kit includes two reference systems that target the ML403 development board. One reference system is based on the PowerPC 405 (PPC405) processor and the other reference system is based on the MicroBlaze processor. The PPC405 processor reference system is created to run the MontaVista Linux image described in Chapter 4, “MontaVista Linux.” The MicroBlaze processor reference system is created to run the BlueCat Linux image described in Chapter 3, “LynuxWorks BlueCat Linux.”. The HelloWorld software application described in Chapter 2, “HelloWorld Software Application,” can be run on either the MicroBlaze processor reference system or the PPC405 processor reference system. Both the PowerPC and MicroBlaze processor systems are described in this chapter.

PowerPC 405 Processor Reference System

Block Diagram

The block diagram for the PowerPC 405 reference system is shown in Figure 1-1.

![PowerPC 405 System Block Diagram](image)

Figure 1-1: PowerPC 405 System Block Diagram
Address Map

The address map for the IP cores in the PowerPC 405 reference system is given in Table 1-1.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Peripheral</th>
<th>Base Address</th>
<th>High Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>xps_bram_if_cntlrl_1</td>
<td>xps_bram</td>
<td>0xFFFF8000</td>
<td>0xFFFFFFFF</td>
</tr>
<tr>
<td>PushButtons_Position</td>
<td>xps_gpio</td>
<td>0x81400000</td>
<td>0x8140FFFF</td>
</tr>
<tr>
<td>LEDs_4Bit</td>
<td>xps_gpio</td>
<td>0x81420000</td>
<td>0x8142FFFF</td>
</tr>
<tr>
<td>IIC_EEPROM</td>
<td>xps_iic</td>
<td>0x81600000</td>
<td>0x8160FFFF</td>
</tr>
<tr>
<td>xps_intc_0</td>
<td>xps_intc</td>
<td>0x81800000</td>
<td>0x8180FFFF</td>
</tr>
<tr>
<td>TriMode_MAC_GMII</td>
<td>xps_ll_temac</td>
<td>0x81C00000</td>
<td>0x81C0FFFF</td>
</tr>
<tr>
<td>SYSACE_CompactFlash</td>
<td>xps_sysace</td>
<td>0x83600000</td>
<td>0x8360FFFF</td>
</tr>
<tr>
<td>xps_timebase_wdt_1</td>
<td>xps_timebase_wdt</td>
<td>0x83A00000</td>
<td>0x83A0FFFF</td>
</tr>
<tr>
<td>xps_timer_1</td>
<td>xps_timer</td>
<td>0x83C00000</td>
<td>0x83C0FFFF</td>
</tr>
<tr>
<td>RS232_Uart</td>
<td>xps_uartlite</td>
<td>0x84000000</td>
<td>0x8400FFFF</td>
</tr>
<tr>
<td>FLASH</td>
<td>xps_mch_emc</td>
<td>0xFF000000</td>
<td>0xFF7FFFFF</td>
</tr>
<tr>
<td>DDR_SDRAM</td>
<td>mpmc</td>
<td>0x00000000</td>
<td>0x03FFFFF</td>
</tr>
<tr>
<td>DDR_SDRAM_SDMA</td>
<td>mpmc</td>
<td>0x84600000</td>
<td>0x8460FFFF</td>
</tr>
</tbody>
</table>

System Configuration

The PPC405 reference system runs off a reference clock frequency of 100 MHz from the oscillator on the board. The PLBv46 bus and the DDR memory run at 100 MHz. The PPC405 processor runs at 200 MHz.

PPC405 Processor Configuration

The PowerPC processor is configured to run at 200 MHz. The instruction cache and data cache are both enabled, with a cache size of 16KB each. The cacheable block of main memory is accessed via the PLBv46 Interface Modules (PIM) of the Multi-Port Memory Controller (MPMC).

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

XPS LL TEMAC Configuration

In the MontaVista Linux demonstration, the Ethernet MAC can run at 10 Mb/s, 100 Mb/s, or 1000 Mb/s, depending on the attached network.

XPS IIC Configuration

In the ML403 development board, a unique Ethernet MAC address is stored in an EEPROM, which is accessed from the FPGA using an IIC bus. The XPS IIC core is included in the system so that MontaVista Linux can read the stored Ethernet MAC address.
XPS MCH EMC Configuration

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

XPS UART Lite Configuration

The XPS UART Lite core is configured to use interrupts and is set to a baud rate of 9600, 8 data bits, and no parity.

MicroBlaze Processor Reference System

Block Diagram

The block diagram for the MicroBlaze reference system is shown in Figure 1-2.

Address Map

The address map for the IP cores in the MicroBlaze reference system is given in Table 1-2.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Peripheral</th>
<th>Base Address</th>
<th>High Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>dlmb_cntlr</td>
<td>lmb_bram_if_cntlr</td>
<td>0x00000000</td>
<td>0x00001FFF</td>
</tr>
<tr>
<td>ilmb_cntlr</td>
<td>lmb_bram_if_cntlr</td>
<td>0x00000000</td>
<td>0x00001FFF</td>
</tr>
<tr>
<td>debug_module</td>
<td>mdm</td>
<td>0x84400000</td>
<td>0x8440FFFF</td>
</tr>
<tr>
<td>xps_bram_if_cntlr_1</td>
<td>xps_bram</td>
<td>0x86608000</td>
<td>0x8660BFFF</td>
</tr>
<tr>
<td>Ethernet_MAC</td>
<td>xps_ethernetlite</td>
<td>0x81000000</td>
<td>0x8100FFFF</td>
</tr>
<tr>
<td>Push.Buttons_Position</td>
<td>xps_gpio</td>
<td>0x81420000</td>
<td>0x8142FFFF</td>
</tr>
</tbody>
</table>
System Configuration

The MicroBlaze reference system runs off a reference clock frequency of 100 MHz from the oscillator on the board. The PLBv46 bus, the MicroBlaze processor and the DDR run at 100 MHz in this system.

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 8KB. The cacheable block of main memory is accessed via the XCL Port Interface Modules (PIM) of the Multi-Port Memory Controller (MPMC).

The instruction cache and data cache are both enabled, with a cache size of 8KB. 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 Xilinx Parallel Flash device, which is used to store the hardware configuration bitstream and bootloader application, as well as the BlueCat Linux kernel image.

XPS UART Lite Configuration

The XPS UART Lite core is configured to use interrupts and is set to a baud rate of 9600, 8 data bits, and no parity.
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 HelloWorld software application can run on either the MicroBlaze or the PowerPC 405 reference system.

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), to download 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 Parallel 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 Virtex-4 FX 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 and downloaded 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 Virtex-4 FX12 ML403 development board.
2. Connect the serial cable between the host computer and the RS232 port on the Virtex-4 FX12 ML403 development board.
3. Apply power to the Virtex-4 FX12 ML403 development board.
4. Start a HyperTerminal (or similar) session on the host computer with the settings shown in Figure 2-1. Select the COM port corresponding to the connected serial port on the host.
computer. Set the Baud Rate to 9600, Data bits to 8 bits, Parity to None, Stop bits to 1 bit, and Flow control to None, as shown in Figure 2-1.

5. In an EDK shell, change directories to the ready_for_download directory in either the MicroBlaze or PowerPC reference system.

6. Use iMPACT to download the bitstream by using the following command:
   ```
   impact -batch ug494.cmd
   ```

7. Invoke XMD and connect to the processor by the following command:
   ```
   xmd -opt ug494.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.

```
Walking LED'S test..Observe the LED'S...

LED's test PASSED
Button and LED Test
Press West Button & see if LED 0 glows
Press South Button & see if LED 1 glows
Press East Button & see if LED 2 glows
Press North Button & see if LED 3 glows
Press Centre Button & see if all the 4 LED glows
```

```
Writing Psuedo random data at address... 0x8C020FFC
Reading Psuedo random data at address.. 0x8C020FFC
Memory Test PASSED!

Press any key to continue....

Type <Menu> for options
```

Figure 2-2: HelloWorld Output

b. For an explanation of the available tests in the application, see the section “Commands 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 either the MicroBlaze or PowerPC 405 reference system project in XPS.
3. Implement the hardware design and create the hardware bitstream by selecting Hardware → Generate Bitstream in XPS.
4. In the Applications tab, build the helloworld project by right-clicking on the project and selecting Build Project. This will create the software executable for the application.
5. Download the bitstream to the board by selecting Device Configuration → Download Bitstream in XPS.
Chapter 2: HelloWorld Software Application

6. After the bitstream has downloaded, launch the XMD by selecting Debug → Launch XMD... in XPS.

7. Download the HelloWorld application executable using the following command inside XMD:

   `dow helloworld/executable.elf`

8. To run the software application, use the run command inside XMD.
   a. After the HelloWorld software application runs, the HyperTerminal output will be as shown in Figure 2-2.
   b. For an explanation of the available commands in the application, see the section “Commands in the HelloWorld Software Application”.

**Commands 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.

*Note:* The Menu command options are case sensitive.

```
->Menu

Xilinx Virtex-4 ML403 Development kit Demo Menu
Menu                   Test DDR2 SDRAM
Led                    Test the LEDs
PBT                    Test Push Buttons
Test <addr><No bytes><data> Perform All factory tests
and <addr><No bytes> Write DDR2 mem locations with given data
Menu                   Read No of DDR2 mem locations and print data
Menu                   Display Menu Options
Menu                   Clear Screen
Menu                   Quit

Type <Menu> for options
```  

*Figure 2-3: HelloWorld Menu*

Table 2-1 lists the commands that are available in the HelloWorld application and describes each one.

**Table 2-1: Description of the HelloWorld Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mem</td>
<td>The Mem test performs a destructive 32-bit wide memory test on a 132K byte block of the DDR SDRAM memory. This test erases, writes, reads, and verifies the DDR memory in the Virtex™-4 FX12 ML403 development board. The results of the test will be displayed in the HyperTerminal.</td>
</tr>
<tr>
<td>Led</td>
<td>The Led test flashes each LED with a delay so that it is visible. Once all the LEDs are flashed, it sends the test pass message to the HyperTerminal.</td>
</tr>
<tr>
<td>PBT</td>
<td>The PBT test reads the push buttons and sends a message to the HyperTerminal on the value of the push button pressed. Hold down the push buttons before running the command.</td>
</tr>
<tr>
<td>Test</td>
<td>This performs all the factory tests mentioned above for the Virtex™-4 FX12 ML403 development board and displays the results to the HyperTerminal.</td>
</tr>
</tbody>
</table>
Booting the HelloWorld Application from Parallel Flash

This section includes steps on how to program the HelloWorld application into the Parallel Flash. These steps includes details on how to use, create, and boot Parallel Flash files for the Virtex-4 FX12 ML403 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. The target board must be configured with the project bitstream before XPS can program the SREC file into the flash memory. Select `Device Configuration → Download Bitstream` in XPS.
3. In XPS, select `Device Configuration → Program Flash Memory`.
4. In the Program Flash Memory dialog box, choose the file to program to be `/helloworld/executable.elf` under the project root directory. Change the program offset to the desired value, allowing enough room for the MCS file, which will be programmed at `0x00000000`. The provided example files use an offset of `0x00100000`. The external DDR memory is set as the Scratch Memory. A bootloader is created by clicking the `Create Flash Bootloader Application` check box in the Program Flash Memory dialog box.
5. After creating the bootloader files, it is strongly suggested to disable the bootloader from displaying its progress. This is done by commenting out the following line in the `bootloader.c` generated file:

   ```
   #define VERBOSE
   ```

### Table 2-1: Description of the HelloWorld Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mwr &lt;addr&gt;&lt;No bytes&gt;&lt;data&gt;</code></td>
<td>This test writes the given data to the DDR memory locations specified. The address range should be within the DDR base address and high address.</td>
</tr>
<tr>
<td><code>mrd &lt;addr&gt;&lt;No bytes&gt;</code></td>
<td>This test reads the number of bytes specified from the DDR memory location given. The address range should be within the DDR base address and high address.</td>
</tr>
<tr>
<td><code>Menu</code></td>
<td>This command lists the menu options for the user.</td>
</tr>
<tr>
<td><code>cls</code></td>
<td>This command clears the HyperTerminal screen.</td>
</tr>
</tbody>
</table>
6. Once the bootloader application is created, the next step is to create the MCS file for the Helloworld bootloader using the iMPACT tool. A MCS file is used to program the StrataFlash with the system bitstream and bootloader loaded in the bitstream. Refer to UG083 ML40x Getting Started Tutorial for ML401/ML402/ML403/ML405 Evaluation Platforms for Preparing PROM files & Programming the PROM for the ML403 Evaluation Platform board.

**Note:** The UG083 tutorial details the steps for generating/programming using EXO file format. The same procedure should be followed for the MCS format.

**Caution!** If the Helloworld application image is downloaded into the Platform Flash, the original Platform flash contents will be overwritten. The original Platform Flash image can be downloaded from the following location:

http://www.xilinx.com/products/boards/ml403/files/Platform_Flash_ML403.zip
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 Virtex-4 FX12 ML403 development board and the hardware platform that is described in Chapter 1, “Hardware Platform.”. The kernel and file system are downloaded into the DDR memory and run completely out of the external memory.

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

- Use a debugger, such as XMD (provided as part of the EDK tools), to download the image file directly into DDR, through the MicroBlaze Debug Module (MDM). This method is described in the section “Executing the BlueCat Linux Image”.

- 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 configuration mode switches to Platform Flash 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 Virtex-4 FX device and the BlueCat Linux kernel image must be downloaded to the DDR 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 bluecat 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 MicroBlaze system root directory, follow these steps:

1. Connect the Platform USB cable or the Parallel IV JTAG cable between the host computer and the Virtex-4 FX12 ML403 development board.

2. Connect the serial cable between the host computer and the RS232 port on the Virtex-4 FX12 ML403 development board.

3. Apply power to the Virtex-4 FX12 ML403 development board.

4. Start a HyperTerminal (or similar) session on the host computer with the settings shown in Figure 3-1. Select the COM port corresponding to the connected serial port on the host computer. Set the Baud Rate to 9600, Data bits to 8 bits, Parity to None, Stop bits to 1 bit, and Flow control to None, as shown in Figure 3-1.
5. In an EDK shell, change directories to the ready_for_download directory in the MicroBlaze reference system.

6. Use iMPACT to download the bitstream by using the following command:
   \texttt{impact -batch ug494.cmd}

7. Invoke XMD and connect to the processor by the following command:
   \texttt{xmd -opt ug494.opt}

8. Download the BlueCat Linux kernel image into DDR memory at the starting location 0x8c000000 using the following command:
   \texttt{dow -data ../bluecat/standalone.kdi 0x8c000000}

\textit{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 0x8c000000
```

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

```
Linux version 2.6.13-4 (sandara@bluecat2 -> sandara@xilinx32rel) (gcc version 4.1.2) #1 Wed Feb 13 02:39:45 EST 2008
On line 0 total pages: 16584
DMA zone: 1684 pages, LIFO batch: 7
Private zone: 0 pages, LIFO batch: 1

Built 1 modules
Kernel command line: roandisk_size=28472 hda=bsrcap hdb=bsrcap hdd=bsrcap udd=bsrcap root=191
xps_init: 100 at 0x8e000000 mapped to 0xe0000000
PID hash table entries: 512 (order=9, 5120 bytes)
pxp_timer: 100 at 0x8e000000 mapped to 0xe0000000
Console: Xilinx 9600 Baud Lite
Dentry cache hash table entries: 16384 (order=4, 66552 bytes)
Inode-cache hash table entries: 8192 (order=3, 32768 bytes)
Memory: 8668k available

Calculating delay loop... 61.84 mspsIP (ipg)=309248
Mount-Cache hash table entries: 512
NET: Registered protocol family 16
xpsinit: PG at 0x81000000 mapped to 0x40000000 device: 10.185 not using IRQ
xpsinit: PG at 0x81000000 mapped to 0x40200000 device: 10.186 using IRQ5

tty0: at 0x10000000 device: driver initialized: 16 RAW disk 28472 size 1024 blocks
Device liveness: Can not lock (EST_NOLOYEE_LOCK), giving up
eth0: Using ifconfig
eth0: No PHY detected, assuming a DHCP at address 0
xpsinit: Xilinx ENCLite #9 at 0x8f000000 mapped to 0x40400000, irq=3
EMC Flash on Xilinx board:Found 2 x16 devices at 0x60 in 32-bit bank
int80: Sharp Extended Query Table at 0x80331
Using buffer write method
console: 6001: Emacs suspend on write enabled
0: offset=0x0 size=0x4000000 blocks=32
Registering a NAND ENC Flash at 0x60000000
NET: Registered protocol family 2
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCP established hash table entries: 496 (order=3, 5276 bytes)
TCP bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCP reso registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
NET: Registered protocol family 1
IF route cache hash table entries: 1024 (order=0, 4096 bytes)
TCE established hash table entries: 496 (order=3, 5276 bytes)
TCE bind hash table entries: 496 (order=3, 26304 bytes)
TCP Hash tables configured (established 496 bind 496)
TCE reso registered
TCE bsc registered
```

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 MicroBlaze reference system project in XPS.

3. Implement the hardware design and create the hardware bitstream by selecting Hardware → Generate Bitstream in XPS.
Chapter 3: LynuxWorks BlueCat Linux

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 DDR memory at the starting location 0x8C000000 using the following command:
   
   ```
   dow -data bluecat/standalone.kdi 0x8c000000
   ```

   **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 0x8c000000
   ```

   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
-bash-3.00H ifconfig
eth0   Link encap:Ethernet HWaddr 00:00:C0:A3:E5:14
       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)
```

**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 4 172.17.1.200`, 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 remotely as 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* for Release 5.4.

The steps described in this section include using the BlueCat Linux Spartan-3E BSP. Even though this reference system is for Virtex-4 FX device, the Spartan-3E BSP contains the needed components to build a working image for the Virtex-4 FX. 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](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 Spartan-3E 1600E 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 Spartan-3E 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*.

   **Note:** When running the commands in these steps, `bsp = sp3e`.

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*. 
Chapter 3: LynuxWorks BlueCat Linux

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 xzf $BLUECAT_PREFIX/boot/edk_user_repository.tar.gz

Note: Alternatively, the bsp folder from the tar file can be placed in the project root directory. The resulting structure will be <project root directory>/bsp/linux_bc54_v1_00_a/.

Generating the BSP

With the use of the BlueCat Linux MLD, XPS 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.sp3e/developer/developer.config

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

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 prebuilt image files.
   ```
   BlueCat:$ make clean
   ```

   ```
   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 DDR memory, put in the path to the new kernel image instead of the path to the pre-built kernel image in the `bluecat` directory.

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 DDR memory and boots BlueCat Linux. These are the steps to program the BlueCat Linux image into parallel Flash.

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. A bootloader, `bootloader_bclinux`, is also provided in the reference system for bootloading the BlueCat Linux image.

**Note:** Before starting these steps, make sure that the standalone OS is chosen under Software Platform Settings. If changing the OS to standalone, make sure to set stdout and stdin in the OS and Libraries settings.

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 file to program and the addresses of the DDR and Flash memory according to the system.

   **Note:** The bootloader provided in the `bootloader_bclinux` directory assumes the image has been programmed at an offset of 0x00100000.
2. In XPS, compile a bitstream, `download.bit`, that includes the system configuration and the bootloader application. This can be the bootloader generated from the five steps detailed in XAPP0963 Using and Creating Flash Files for the MicroBlaze Development Kit - Spartan-3E Edition or the provided bootloader in `bootloader_bclinux`. Compile the bitstream by marking the bootloader application to Initialize BRAMs and then selecting Device Configuration → Update Bitstream in XPS.

3. Once the bootloader application is created, the next step is to create the MCS file for the system using the iMPACT tool. A MCS file is used to program the StrataFlash with the system bitstream and bootloader loaded in the bitstream. Refer to UG083 ML40x Getting Started Tutorial for ML401/ML402/ML403/ML405 Evaluation Platforms for Preparing PROM files & Programming the PROM for the ML403 Evaluation Platform board.

**Note:** The UG083 tutorial details the steps for generating/programming using EXO file format. The same procedure should be followed for the MCS format.

**Caution!** If the BlueCat Linux image is downloaded into the Platform Flash, the original Platform flash contents will be overwritten. The original Platform Flash image can be downloaded from the following location:

http://www.xilinx.com/products/boards/ml403/files/Platform_Flash_ML403.zip
MontaVista Linux

Introduction

The MontaVista Linux reference system demonstrates MontaVista Linux running on the PowerPC 405 processor. An example MontaVista Linux image is provided that is tailored to the ML403 development board and the PPC405 hardware platform that is described in Chapter 1, “Hardware Platform.” The MontaVista Linux image, composed of the kernel and the ramdisk file system, is downloaded into the DDR memory and run completely out of the external memory.

The methods for downloading and running the MontaVista Linux image are listed below:

1. Use a debugger, such as XMD (provided as part of the EDK tools), to download the image file directly into DDR. This method is described in the section “Executing the MontaVista Linux Image”.

2. Program Flash memory with the MontaVista Linux image. This method is described in the section “Booting the MontaVista Linux Image from Parallel Flash”. Once Flash memory is programmed, the MontaVista Linux demonstration can be run by setting the configuration mode switches to Platform Flash and either powering up the development board or depressing the PROG button on the board.

Executing the MontaVista Linux Image

To execute the MontaVista Linux reference system, program the hardware bitstream to the Virtex-4 FX12 device and download the MontaVista Linux image to the DDR 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 MontaVista Linux image is downloaded from the mvlinux directory.

Executing the MontaVista Linux Image Using the Pre-Built Bitstream

Execute the reference system using the files in the ready_for_download directory in the PPC405 system root directory by following the subsequent steps:

1. Connect the Platform USB cable or Parallel IV JTAG cable between the host computer and the ML403 development board.

2. Connect the serial cable between the host computer and the RS232 serial port on the ML403 development board.

3. Apply power to the ML403 development board.
4. Start a HyperTerminal (or similar) session on the host computer with the settings shown in Figure 4-1. Select the COM port corresponding to the connected serial port on the host computer. Set the Baud Rate to **9600**, Data bits to **8** bits, Parity to **None**, Stop bits to **1** bit, and Flow control to **None**, as shown in Figure 4-1.

![Figure 4-1: HyperTerminal Settings](image)

5. In an EDK shell, change directories to the `ready_for_download` directory in the PowerPC 405 reference system.
6. Download the bitstream using iMPACT by using the command:
   ```
   impact -batch ug494.cmd
   ```
7. Invoke XMD and connect to the processor by using the command:
   ```
   xmd -opt ug494.opt
   ```
8. Download the MontaVista Linux image into DDR memory using the command:
   ```
   dow ../mvlinux/zImage.initrd.elf
   ```
9. Start the image running and boot MontaVista Linux by using the XMD command:
   ```
   run
   ```

   **Caution!** This Linux kernel includes the SystemACE driver. If there is no CompactFlash card inserted into the ML403 board, the Linux kernel will hang during boot-up because the SystemACE driver will block while looking for a card. To allow the Linux kernel to boot completely, ensure that there is a CompactFlash card inserted into the ML403 board.

   **Note:** Upon boot-up, the MontaVista Linux kernel is set up to request an IP address from a DHCP server, if the board is connected to a network with a DHCP server running. If the board is not connected, the DHCP request will time out, which may take 3-4 minutes.
a. After MontaVista Linux boots, the HyperTerminal output will be as shown in Figure 4-2.

```plaintext
Executing the MontaVista Linux Image

R
Figure 4-2: MontaVista Linux Boot Output
```
b. Log into MontaVista Linux by using the username root.

c. For example commands to run in MontaVista Linux, see the section “Executing MontaVista Linux Commands”.

## Executing the MontaVista Linux Image from XPS

Execute the MontaVista Linux reference system using XPS by following the subsequent steps:

1. Perform steps 1-4 in the “Executing the MontaVista 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 MontaVista Linux image into DDR memory using the command:
   ```
   dow mvlinux/zImage.initrd.elf
   ```
7. Start the image running and boot MontaVista Linux by using the XMD command:
   ```
   run
   ```
   a. After MontaVista Linux boots, the HyperTerminal output will be as shown in Figure 4-2.
   b. Log into MontaVista Linux by using the username root.
   c. For example commands to run in MontaVista Linux, see the section “Executing MontaVista Linux Commands”.

## Executing MontaVista Linux Commands

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

This MontaVista 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 4-3. In the figure, the board IP address is 192.168.0.100.

If the board is connected to a network with a DHCP server, MontaVista Linux will request an IP address during boot up. If that is not the case, the IP address can be set manually. To turn on the ethernet port 0, eth0, and assign the board an IP address, the command `ifconfig eth0 up board_IP_address`, is issued.

```
eth0  Link encap:Ethernet  HWaddr 00:00:35:00:22:00
     inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
     UP BROADCAST RUNNING MTU:1500 Metric:1
     RX packets:201 errors:0 dropped:0 overruns:0 frame:0
     TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:6693 (65.1 KiB)  TX bytes:2360 (2.3 KiB)
     Interrupt:5
```

```
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)
```

**Figure 4-3: Ethernet Configuration Settings**

To ping a remote computer from the development board, use `ping -c 4 IP_address` in MontaVista Linux to ping the remote computer four times.

Use the command `telnet board_IP_address` to telnet from a networked computer to the board. All of the Linux commands can now be performed remotely as if the user were logged into the console on a HyperTerminal.

---

**Building the MontaVista Linux Image**

This section briefly describes the process for rebuilding the MontaVista Linux image that is included with this reference system.

**Installing and Setting up the MontaVista Linux Development Environment**

The process for installing and setting up the MontaVista Linux development environment is outlined in the steps below. These steps reference the XAPP969, *Getting Started with EDK and Linux 2.6* application note.

1. Follow the steps outlined in the “Installing MontaVista Linux Professional Edition 4.0.1 for ML40x” section in the XAPP969, *Getting Started with EDK and Linux 2.6* application note to install the Monta Vista Linux development toolkit.
2. Create a new directory called `linux` in the EDK project directory.
3. Follow the steps outlined in the “Setting Up the Development Environment for MontaVista Pro 4.0.1 Linux” section in the XAPP969, *Getting Started with EDK and Linux 2.6* application note to set up the MontaVista Linux development environment.
Chapter 4: MontaVista Linux

Generating the BSP

XPS can update the MontaVista Linux kernel source tree to match a specific hardware configuration. Follow these steps to generate the BSP and update the MontaVista Linux kernel source tree.

1. Open the reference system in XPS.
2. Select Software → Software Platform Settings... in XPS.
3. Select linux_2.6 in the OS field in the Software Platform Settings window as shown in Figure 4-4.

![Software Platform Settings](image)

Figure 4-4: Software Platform Settings OS Selection
4. Select the OS and Libraries option on the left of the Software Platform Settings window to set parameters necessary for generating the BSP. The setting of these parameters is listed below and shown in Figure 4-5.
   a. Click **Edit...** in the connected_periphs field. In the Add/Delete List of Parameter-Values box, the following list of peripherals is displayed:
      - RS232_Uart
      - LEDs_4Bit
      - Push.Buttons_Position
      - IIC_EEPROM
      - SysACE_CompactFlash
      - TriMode_MAC_GMII
      - xps_intc_0
   Click **OK** to accept the settings in the Add/Delete List of Parameter-Values window.
   b. The ML403 has 64 MB of DDR memory. Enter **0x04000000** to the memory size parameter to assign all the memory for Linux.
   c. Set the field uart16550 bus clock freq to **100000000**.
   d. Set the target directory field to `<edk_proj>/linux` to cause the Linux BSP to be copied directly into the Linux kernel that is being used for this system.
   e. Select **ramdisk** from the drop down menu in the rootfs type field.

5. In the Software Platform Settings window, click **OK** to accept the settings.
6. In the Applications tab in XPS, right click on Default:ppc405_0_bootloop, and enable Mark to Initialize BRAMs. Ensure that no other applications are marked for BRAM initialization.

7. Select Software → Generate Libraries and BSPs in XPS Build the Linux BSP.

Rebuilding the Kernel Image

This is the final step in creating a bootable MontaVista Linux kernel image. To recreate the image provided with this reference system, follow these steps:

1. MontaVista Linux 4.0.1 includes a prebuilt RAM disk image. To use this RAM disk image, copy it to the Linux kernel source tree. From the `<edk_proj>/linux` directory, run the command:

```bash
cp /opt/montavista/pro/devkit/ppc/405/images/ramdisk.gz arch/ppc/boot/images/ramdisk.image.gz
```

**Note:** This command assumes that MontaVista Pro 4.0.1 has been installed in the `/opt/montavista` directory.

2. To build the MontaVista Linux image, enter the following command in the host computer shell:

```bash
make zImage.initrd
```

This command produces a MontaVista Linux image that will mount the ramdisk image as the root filesystem at startup. The Linux image will be stored in `<edk proj>/linux/arch/ppc/boot/images/zImage.initrd.elf`.

3. Refer to the steps in the section “Executing the MontaVista Linux Image” to run the newly created kernel image. Put in the path to the new kernel image instead of the path to the pre-built kernel image in the `mvlinux` directory when downloading the kernel image through XMD into DDR memory.

Booting the MontaVista Linux Image from Parallel Flash

To boot the MontaVista Linux image from parallel Flash, program the Linux image and a bootloader application into Flash. The bootloader application copies the Linux image from Flash to DDR memory and boots MontaVista Linux. The steps below outline the process to program the MontaVista Linux image into parallel Flash.

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. A bootloader, `bootloader_mvlinux`, is also provided in the reference system for bootloading the MontaVista Linux image.

**Note:** Before starting these steps, make sure that the standalone OS is chosen under Software Platform Settings. If changing the OS to standalone, make sure to set stdout and stdin in the OS and Libraries settings.

1. Convert the MontaVista Linux image `zImage.initrd.elf` to `zImage.bin` format so that it can be programmed into the Flash. This can be done by using the following command in the EDK Shell window.

```bash
powerpc-eabi-objcopy -O binary zImage.initrd.elf zImage.bin
```
2. Using this zImage.bin, follow the steps as mentioned in the “Programming the BlueCat Linux Image KDI File into StrataFlash and Creating New Bootloader Files” section in the XAPP963 Using and Creating Flash Files for the MicroBlaze Development Kit - Spartan-3E Edition application note. The steps provide information on how to program the Flash with the Linux image. Modify the file to program and the addresses of the DDR and Flash memory according to the system.

**Note:** The bootloader provided in the bootloader_mvlinux directory assumes the image has been programmed at an offset of 0x00100000.

3. In XPS, compile a bitstream, download.bit, that includes the system configuration and the bootloader application. This can be the bootloader generated from the five steps detailed in the XAPP963 Using and Creating Flash Files for the MicroBlaze Development Kit - Spartan-3E Edition application note or the provided bootloader in bootloader_mvlinux. Compile the bitstream by marking the bootloader application to Initialize BRAMs and then selecting **Device Configuration → Update Bitstream** in XPS.

4. Once the bootloader application is created, create the MCS file for the MontaVista Linux system using the iMPACT tool. A MCS file is used to program the StrataFlash with the system bitstream and bootloader loaded in the bitstream. Refer to UG083 ML40x Getting Started Tutorial for ML401/ML402/ML403/ML405 Evaluation Platforms.

**Note:** The UG083 tutorial details the steps for generating/programming using EXO file format. The same procedure should be followed for the MCS format.

**Caution!** If the MontaVista Linux image is downloaded into the Platform Flash, the original Platform flash contents will be overwritten. The original Platform Flash image can be downloaded from the following location:

http://www.xilinx.com/products/boards/ml403/files/Platform_Flash_ML403.zip