Getting Started with the MicroBlaze Development Kit - Spartan-3E 1600E Edition

UG258 (v1.0) June 29, 2006

XILINX®

Xilinx does not assume any liability arising out of the application or use of the Design; nor does Xilinx convey any license under its patents, copyrights, or any rights of others. You are responsible for obtaining any rights you may require for your use or implementation of the Design. Xilinx reserves the right to make changes, at any time, to the Design as deemed desirable in the sole discretion of Xilinx. Xilinx assumes no obligation to correct any errors contained herein or to advise you of any correction if such be made. Xilinx will not assume any liability for the accuracy or correctness of any engineering or technical support or assistance provided to you in connection with the Design.

THE DESIGN IS PROVIDED "AS IS" WITH ALL FAULTS, AND THE ENTIRE RISK AS TO ITS FUNCTION AND IMPLEMENTATION IS WITH YOU. YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE NOT RELIED ON ANY ORAL OR WRITTEN INFORMATION OR ADVICE, WHETHER GIVEN BY XILINX, OR ITS AGENTS OR EMPLOYEES. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DESIGN, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOST DATA AND LOST PROFITS, ARISING FROM OR RELATING TO YOUR USE OF THE DESIGN, EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE TOTAL CUMULATIVE LIABILITY OF XILINX IN CONNECTION WITH YOUR USE OF THE DESIGN, WHETHER IN CONTRACT OR TORT OR OTHERWISE, WILL IN NO EVENT EXCEED THE AMOUNT OF FEES PAID BY YOU TO XILINX HEREUNDER FOR USE OF THE DESIGN. YOU ACKNOWLEDGE THAT THE FEES, IF ANY, REFLECT THE ALLOCATION OF RISK SET FORTH IN THIS AGREEMENT AND THAT XILINX WOULD NOT MAKE AVAILABLE THE DESIGN TO YOU WITHOUT THESE LIMITATIONS OF LIABILITY.

The Design is not designed or intended for use in the development of on-line control equipment in hazardous environments requiring fail-safe controls, such as in the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, life support, or weapons systems ("High-Risk Applications"). Xilinx specifically disclaims any express or implied warranties of fitness for such High-Risk Applications. You represent that use of the Design in such High-Risk Applications is fully at your risk.

Revision History

The following table shows the revision history for this document.

	Version	Revision
06/29/06	1.0	Initial Xilinx release.

Table of Contents

Preface: About This Guide

Chapter 1: MicroBlaze Development Kit - Getting Started

Chapter 2: MicroBlaze Development Kit uClinux Reference System

Chapter 3: MicroBlaze Development Kit Web Server Reference System

About This Guide

The MicroBlaze™ Development Kit - Spartan™-3E 1600E Edition showcases various features of the SP3E1600E Development Platform. This kit includes reference systems and demonstrations. This document describes how to use and run them.

Guide Contents

This manual contains three chapter:

Additional Resources

To search the database of silicon and software questions and answers, or to create a technical support case in 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**

Convention	Meaning or Use	Example
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`**`}`

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-4 User Guide.*
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

Chapter 1

MicroBlaze Development Kit - Getting Started

Overview

The MicroBlaze™ Development Kit - Spartan™-3E 1600E Edition is designed to aid the user in utilizing the SP3E1600E development board more efficiently. This getting started guide provides a detailed description of what is included in the kit along with instructions on how to use the resources included in this kit. The development kit comes with a number of pre-installed demonstrations and examples. This document describes how to use and run them.

The development kit includes reference systems and demonstrations. These are included on the Development Kit Reference CD as well as pre-loaded in the Xilinx Platform Flash, SPI Flash, and StrataFlash on the development board. Please read the remainder of the document for specific details on these resources.

The best way to get started with the development kit is by inserting the Development Kit Reference CD located in the development kit box.

The reference systems included in this kit require the use of a computer installed with the following software:

- Integrated Software Environment™ (ISE) 8.1i, including Service Pack 3
- Xilinx Embedded Development Kit (EDK) 8.1i, including Service Pack 2

Note: These software configurations are *required* for the reference systems to function properly. Please ensure the proper versions prior to working with the reference systems. Detailed information about how to properly install your software is included in the Software Installation section later in this document.

For current information about the included development platform, visit the following Web page:

http://www.xilinx.com/sp3e1600e

Quick Start Information

The following section is provided to help the user get up and running with the development kit quickly. It contains basic instructions and requirements to get started with the development kit without having to read the entire manual. This is recommended for users familiar with Xilinx products. For a user that is new to Xilinx products, it is recommended that the user read the rest of this document as it contains more detailed information and instructions on the resources included in this development kit.

Step 1 - Install the ISE WebPack and Xilinx Platform Studio™ (XPS/EDK) software. Please make sure to install the necessary software as indicated below:

- ISE 8.1i WebPack, including Service Pack 3
- EDK 8.1i, including Service Pack 2

The software service packs are in the Development Kit Reference CD under the `\software_service_packs` directory. Detailed installation instructions are included later in this document.

Step 2 - Set up the development board and connect it to the host machine. This includes connecting the following:

- Board power supply
- Download cable - host to target
- Null modem serial cable - host to target
- Cross-over ethernet cable - host to target

Detailed instructions of how to accomplish these tasks are found later in this document.

Step 3 - Choose the reference system or demonstration of choice. Demonstrations are located in the Xilinx Platform Flash, SPI Flash, and StrataFlash on the development board. These demonstrations are selected by changing the configuration mode jumpers. The following demonstrations are pre-loaded on the development board:

- Hello World Demonstration - Master Serial configuration mode
- Web Server Demonstration - SPI configuration mode
- µCLinux Demonstration - BPI Up configuration mode

In addition to the demonstrations, the Development Kit Reference CD contains the source files for the reference systems. The reference systems included on the CD are:

- Web Server Reference System
- µCLinux Reference System

To use these designs, copy the contents for the desired reference system local to the host machine. Once copied over, invoke Xilinx Platform Studio (XPS) and open the project file.

Detailed instructions for these reference systems are included in the remaining chapters of this document. Please refer to these chapters for more information.

Kit Contents

The information below describes the contents of the development kit. The development kit contains all of the following items.

- Spartan-3E SP3E1600E development board
- ISE WebPack Software

- EDK/Xilinx Platform Studio Software
- Development Kit Reference CD - includes reference systems and documentation
- Universal power supply and EU and UK power adaptors
- USB download cable
- Null modem serial cable
- Cross-over ethernet cable

Development Kit Reference CD

The development kit includes a resource CD that contains useful materials for expediting the creation of embedded designs using Xilinx platforms. The necessary directory structure and files representing the reference systems are included in ZIP files, which can be copied and unzipped.

The top level directories included on the CD are:

- **images** – contains the web page images
- **css** – contains the web page style sheets
- **templates** – contains the web page style templates
- **documentation** – includes this document, the reference system user guide, as well as the SP3E1600E board schematic, BOM, and Gerber files
- **software_service_packs** – contains the required service pack installs for XPS and ISE software packages. The next section details how to install these files.
- **reference_systems** – contains the files relevant to the reference systems included with this kit. This directory includes the following directories:
  - **SP3E1600E_HelloWorld** – contains the bit file and HyperTerminal setup file for this demonstration
  - **WebServer** – contains the XPS source project for the Web server reference system
  - **uCLinux** – contains the XPS source project for the μClinux reference system

  Note: A directory called **ready_for_download** is included under the WebServer and μCLinux directories. The **ready_for_download** directory contains the compiled bitstream file that can be quickly downloaded using iMPACT. Also included in this directory is an example HyperTerminal settings file.

Software Installation

The development kit includes the required design software to work with the reference systems. A full version of EDK and ISE WebPack are included. To use the reference systems included in this kit, both software packages are required to be installed. To properly install the software packages, please insert the installation CD and follow the installation instructions. To obtain an installation code, each software package must be registered. Information on how to accomplish this is included in the install.

Once the software package has been installed, it is necessary to add the latest service packs. These have been included on the Development Kit Reference CD as well. Navigate into the `software_service_packs` directory on the CD. Each software package has a directory containing the required service pack. Nagivate into each directory and double click on the `setup.exe` file. This will launch the service pack installer. Follow the installation directions to complete the process.

Demo Description and Configuration Mode Settings

Demo Description

The demos that are available in the various PROM devices on the board are described below. All of the code for these demos are loaded into PROM at the factory. The configuration setting from the factory is for the Hello World Demo. The PROM files for these demos can be found on the following Spartan-3E Web page link:

http://www.xilinx.com/sp3e1600e

SP3E1600E_HelloWorld Demo

This is a simple demonstration that exercises a few of the board features. The demo will flash the LEDs and read the DIP and pushbutton switches. Messages will be displayed on the 2x16 LCD display and, if a serial terminal window is open, diagnostic information will appear in the terminal window. This application is loaded into the Platform Flash PROM.

Web Server Demo

This is a demonstration of a Web server running on the MicroBlaze soft processor. Refer to the Web Server Reference System Chapter for details and instructions on how to run the Web server demo. The Web server application is loaded into the SPI Serial Flash PROM.

μClinux Demo

This is a demonstration of μClinux running on the MiroBlaze™ soft processor. Refer to the μClinux Reference System Chapter for details and instructions on how to run the Web server demo. The μClinux image is loaded into the Intel StrataFlash PROM.

Address Map for SPI and BPI Flash PROMs

Table 1-1 shows the address map for each of the PROM files in the StrataFlash PROM device and Table 1-2 shows the address map for the PROM file in the SPI Serial Flash PROM device.

Table 1-1: **StrataFlash PROM Address Map**

PROM File	Start Address	End Address	Total Space
μClinux Design (Bit file)	0x00000000	0x000B62E3	746KB
Unused	0x000B62E4	0x002FFFFF	2399KB
μClinux Kernel Image	0x00300000	0x008E98A7	6199KB
Unused	0x008E98A8	0x009FFFFF	1140KB
Web Server Software Application	0x00A00000	0x00A89031	561KB
Unused	0x00A89032	0x00AFFFFF	487KB
Web Server Memory File System	0x00B00000	0x00B4DEE3	319KB
Unused	0x00B4DEE4	0x00FFFFFF	4923KB

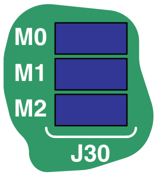
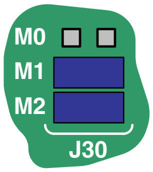
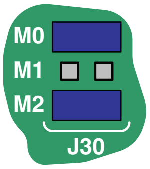
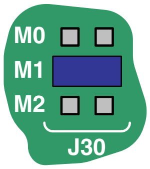
Table 1-2: **SPI Serial Flash PROM Address Map**

PROM File	Start Address	End Address	Total Space
Web Server Design (Bit file)	`0x00000000`	`0x000B62E3`	`746KB`
Unused	`0x000B62E4`	`0x007FFFFF`	`7642KB`

Configuration Mode Settings

Table 1-3 shows the jumper block settings used to control the FPGA's configuration mode. They select the configuration memory source for power-up and for depressing the PROG button.

Table 1-3: **Development Kit Demo and Configuration Mode Jumper Settings**

Demo	Configuration Mode	Configuration Image Source	Jumper Settings (J30)
Hello World Demo	Master Serial	Platform Flash PROM	
Web Server Demo	SPI	SPI Serial Flash PROM	
µClinux Demo	BPI-Up	StrataFlash PROM	
Not Applicable	JTAG	Downloaded from host via USB-JTAG port	

Executing the Demos

1. Position the SP3E1600E board so the SPARTAN-3E and XILINX logos are oriented upright.
2. Make sure the power switch, located in the upper left corner of the board, is in the *off* position.
3. Plug the power supply adapter cable into the SP3E1600E board. Plug in the power supply to AC power.
4. Assure that the configuration jumpers to Master Serial mode, as shown in Table, for the Hello World Demo.
5. Connect a null modem serial cable between your PC and the SP3E1600E board's DTE connector and open a serial terminal program.
   a. The serial configuration settings need to be set as follows:
      - Bits per second = **115200**
      - Data bits = **8**
      - Parity = **None**
      - Stop bits = **1**
      - Flow control = **None**

   b. There is a HyperTerminal script file, `SP3E1600E.ht`, provided in the `/reference_system/SP3E1600E_HelloWorld` directory on the Development Kit Reference CD. Double click on this file to open a HyperTerminal window with the appropriate settings.
6. Turn on the SP3E1600E board's main power switch. If the configuration jumpers are set to Master Serial mode, then the Hello World Demo will automatically start.
7. To restart or load another demo, set the configuration jumpers to the proper settings for the desired demo, and depress the PROG pushbutton or simply power-cycle the main power switch.
   a. The selected demo will load from the appropriate PROM and the DONE LED will light upon completion of the download.

Note: The Web Server Demo requires an ethernet connection from the board to a host computer.

Chapter 2

MicroBlaze Development Kit μClinux Reference System

Introduction

This reference system demonstrates μClinux running on the MicroBlaze™ soft processor and was designed using the Embedded Development Kit (EDK).

An example basic μClinux kernel is provided that is tailored to the Spartan™-3E 1600E Edition Development Kit board and this reference design.

The kernel and ROM file system are downloaded into the DDR memory and run completely out of the external memory.

The methods for downloading and running the μClinux kernel demonstration are:

- By powering up from the Intel StrataFlash PROM with the FPGA configuration mode pins set to BPI-up mode
- By depressing the PROG button on the board with the FPGA configuration mode pins set to BPI-up mode, thereby invoking download from the Intel StrataFlash PROM
- By using a debugger, such as XMD (provided as part of the EDK tools), and downloading the image file directly into DDR, through the MicroBlaze Debug Module

The μClinux kernel image is pre-loaded into the Intel StrataFlash on the board from the factory.

Details for obtaining the FLASH code images and the procedure for how to re-program the FLASH memory can be found on the following Spartan-3E Web page link:

http://www.xilinx.com/sp3e1600e

Hardware Specifics

This reference system targets the Spartan-3E 1600E Edition development board. The system uses the MicroBlaze processor with cache turned on for both the instruction cache (I-cache) and the data cache (D-cache). As shown in Figure 2-1, the system also includes the MCH OPB DDR memory controller, the OPB Ethernet MAC, the OPB EMC memory controller, and the OPB UART Lite IP cores. An OPB Timer and OPB Interrupt controller are also needed for the μClinux kernel.

See Table 2-1 for the address map of the system.

Block Diagram

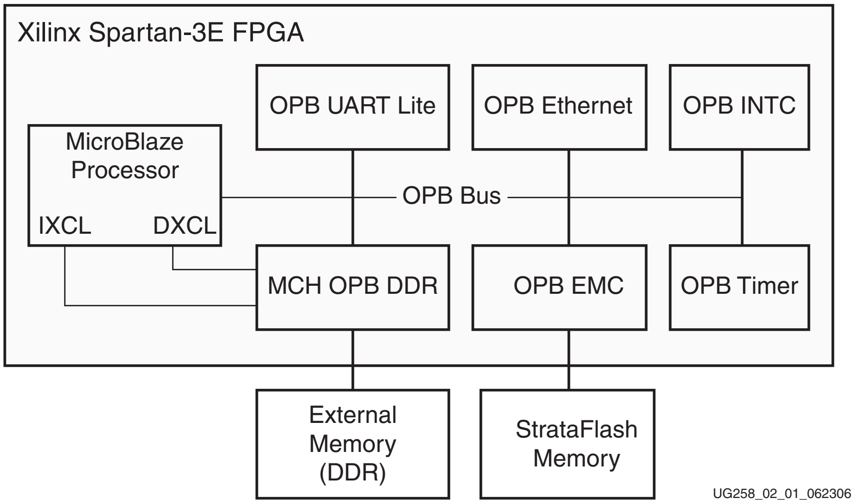


Figure 2-1: **Block Diagram**

Address Map

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

Table 2-1: **Reference System Address Map**

Instance	Peripheral	Base Address	High Address
debug_module	opb_mdm	0x41400000	0x4140FFFF
dlmb_cntlr	lmb_bram_if_cntlr	0x00000000	0x00001FFF
ilmb_cntlr	lmb_bram_if_cntlr	0x00000000	0x00001FFF
RS232_DTE	opb_uartlite	0x40600000	0x4060FFFF
FLASH_16x8	opb_emc	0x21000000	0x21FFFFFF
DDR_SDRAM_32Mx16	mch_opb_ddr	0x22000000	0x23FFFFFF
Ethernet_MAC	opb_ethernet	0x40C00000	0x40C0FFFF
opb_timer_1	opb_timer	0x41C00000	0x41C0FFFF
opb_intc_0	opb_intc	0x41200000	0x4120FFFF

System Configuration

This reference system runs off a reference clock frequency of 66.67MHz from the auxiliary oscillator on the board. The OPB BUS, Ethernet MAC, and the MicroBlaze processor run at 66.67MHz, while the DDR runs at 133MHz.

The *MicroBlaze Development Kit Spartan-3E 1600E Edition Board User Guide*, which is a part of the Development Kit Reference CD supplied with this kit, provides additional information on board specifics and UCF location constraints for the different cores used in this reference system.

MicroBlaze Processor Configuration

The MicroBlaze with MCH OPB DDR is configured with the I-cache and D-cache enabled. A cache size of 8KB is set for both the instruction cache (I-cache) and the data cache (D-cache). The cacheable block of main memory resides between `0x22000000` and `0x2200FFFF`. The MCH interfaces of the MCH OPB DDR memory controller are connected to the MicroBlaze processor and configured to use the XCL protocol.

The Instruction Cache and Data Cache sections in the *MicroBlaze Processor Reference Guide* provide details on the MicroBlaze processor caches.

OPB Ethernet Configuration

The μClinux RTOS requires that the OPB Ethernet be set to use Scatter-Gather DMA capability and that the interrupts be set to *on*. In the μClinux 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.

OPB EMC Configuration

The OPB EMC memory controller is connected to an external Intel StrataFlash PROM, which is used to store the hardware configuration bitstream and bootloader application, as well as the μClinux kernel image file.

OPB UART Lite Configuration

The OPB UART Lite core is configured to use interrupts. It is set up to use the DTE-Style serial port with the baud rate set to **115200**. This requires the use of the Null-modem serial cable provided with this Kit.

Software Applications

The software sources and executable files for simple memory and peripheral tests have been provided for use in basic functional testing.

> The source code for the memory and peripheral tests are located under the project root directory as `/TestApp_Memory/src/TestApp_Memory.c` and `/TestApp_Peripheral/src/TestApp_Peripheral.c`, respectively.
>
> The executable files for these tests are located under the project root directory as `/TestApp_Memory/executable.elf` and `/TestApp_Peripheral/executable.elf`, respectively.

Executing the Reference System

Provided with this Reference System

A μClinux-ready MicroBlaze system and μClinux kernel are provided in this reference system. This basic μClinux implementation is built using the μClinux version 2.4.32 distribution.

The μClinux kernel image is stored in the Intel StrataFlash PROM and is available in `/uClinux_images` under the project root directory.

Note: The method and details for building a μClinux kernel are beyond the scope of this document.

Executing the μClinux Reference System

To execute the μClinux reference system, the hardware bitstream and the μClinux kernel image must be downloaded into main memory (DDR) by:

1. Downloading the bitstream directly from the `/ready_for_download` directory or by generating it from the project in XPS. The bitstream is composed of the hardware configuration and a simple bootloop software application, which loops the MicroBlaze processor at a known location until another application is executed.
2. Downloading the μClinux kernel image from the directory `/uClinux_images` under the project root directory.

Generating the Bitstream

The system netlist and bitstream can be generated by:

1. Opening the μClinux project in XPS, then
2. Generating the bitstream by selecting **Hardware** → **Generate Bitstream** in XPS

Downloading the Bitstream

To configure the Spartan-3E device, the bitstream can be downloaded by using one of two methods:

1. Using the download option in XPS, then connecting the USB cable from the PC to the board and selecting the **Device Configuration** → **Download Bitstream** option
2. Using the iMPACT configuration tool and downloading the bitstream.

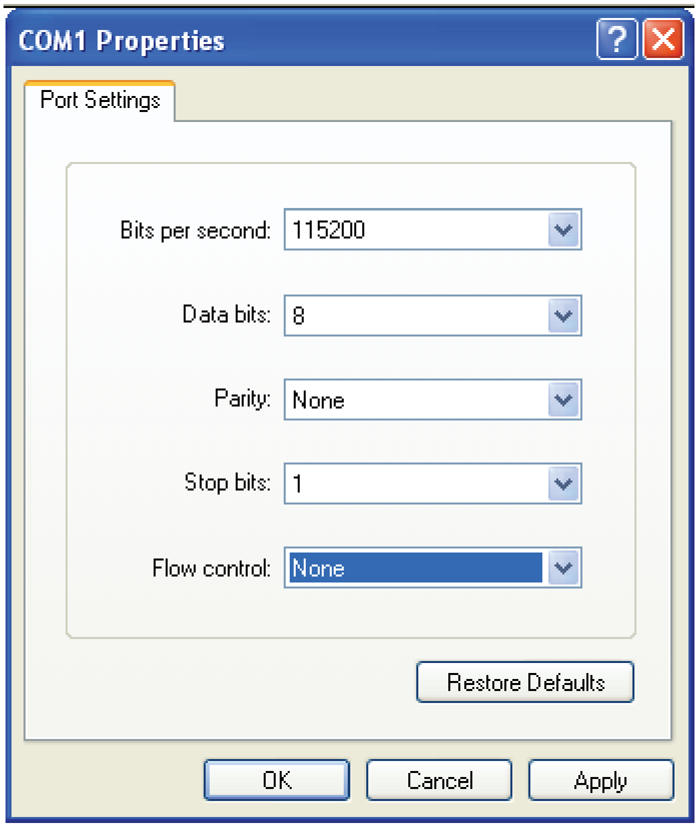
The bitstream can be found under the project root directory as `/ready_for_download/download.bit.`

Downloading the μClinux Kernel

Before the μClinux demonstration can be executed, a serial communications utility, such as HyperTerminal, must be set up. To configure the HyperTerminal and run the μClinux demonstration on MicroBlaze, the following steps are followed:

1. A serial cable from the COM port of the PC to the DTE serial port on the board is connected.

2. A HyperTerminal or similar program is set to a Baud Rate of **115200**, Data Bits to **8**, Parity to **None** and Flow Control to **None**, as shown in Figure 2-2.
3. A crossover Ethernet cable to the host PC and the board's Ethernet port is connected. The USB cable must remain connected from the PC to the board.


COM1 Properties

Port Settings

Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Figure 2-2: **HyperTerminal Settings**

Downloading the μClinux Kernel through XMD

To download and boot-up the μClinux kernel image via XMD, the following commands listed below are invoked in an XMD window. The image file can be found under the project root directory as `/uClinux_images/image.bin`.

1. In XPS, **Debug** → **Launch XMD** is selected.
2. In the XMD command window, the directory to the project's `/uClinux_images` directory is changed as follows:

   ```
   cd uClinux_images
   ```
3. The μClinux image into DDR memory at the starting location `0x22000000` is downloaded using the command:

   ```
   dow -data image.bin 0x22000000
   ```

4. To start the kernel image running, which will boot-up µClinux on the HyperTerminal, the following command is used:

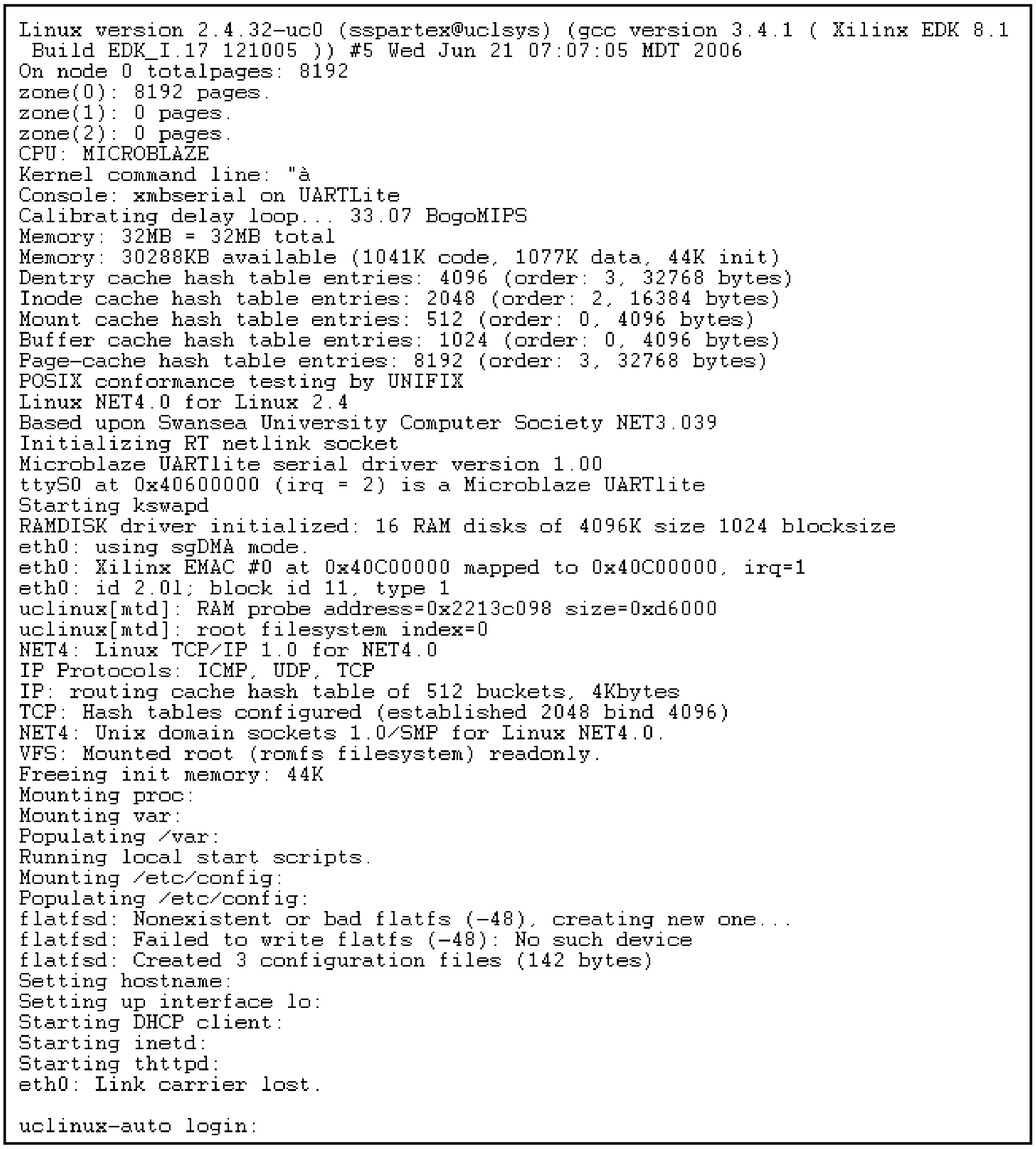
   ```
   con 0x22000000
   ```

   a. After the µClinux kernel boots up, the output on the HyperTerminal will be as shown in Figure 2-3.

5. Login into µClinux with a username of **`root`** and a passwd of **`root`**.

Note: Upon boot-up, the µClinux 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 DCHP request will time out; however this is not an error. The message to the HyperTerminal is shown below:

```
# Jan  1 00:01:03 dhcpcd[33]: timed out waiting for a valid DHCP
server response
```



```
Linux version 2.4.32-uc0 (sspartex@uclsys) (gcc version 3.4.1 ( Xilinx EDK 8.1
 Build EDK_I.17 121005 )) #5 Wed Jun 21 07:07:05 MDT 2006
On node 0 totalpages: 8192
zone(0): 8192 pages.
zone(1): 0 pages.
zone(2): 0 pages.
CPU: MICROBLAZE
Kernel command line: "à
Console: xmbserial on UARTLite
Calibrating delay loop... 33.07 BogoMIPS
Memory: 32MB = 32MB total
Memory: 30288KB available (1041K code, 1077K data, 44K init)
Dentry cache hash table entries: 4096 (order: 3, 32768 bytes)
Inode cache hash table entries: 2048 (order: 2, 16384 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Microblaze UARTlite serial driver version 1.00
ttyS0 at 0x40600000 (irq = 2) is a Microblaze UARTlite
Starting kswapd
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
eth0: using sgDMA mode.
eth0: Xilinx EMAC #0 at 0x40C00000 mapped to 0x40C00000, irq=1
eth0: id 2.01; block id 11, type 1
uclinux[mtd]: RAM probe address=0x2213c098 size=0xd6000
uclinux[mtd]: root filesystem index=0
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 4096)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
VFS: Mounted root (romfs filesystem) readonly.
Freeing init memory: 44K
Mounting proc:
Mounting var:
Populating /var:
Running local start scripts.
Mounting /etc/config:
Populating /etc/config:
flatfsd: Nonexistent or bad flatfs (-48), creating new one...
flatfsd: Failed to write flatfs (-48): No such device
flatfsd: Created 3 configuration files (142 bytes)
Setting hostname:
Setting up interface lo:
Starting DHCP client:
Starting inetd:
Starting thttpd:
eth0: Link carrier lost.

uclinux-auto login:
```

UG258_02-03_062606

Figure 2-3: **µClinux Boot-Up Screen**

Executing μClinux Commands

This build of μClinux supports many basic Linux commands. The list of commands and tools available to run can be found in the `/bin` directory.

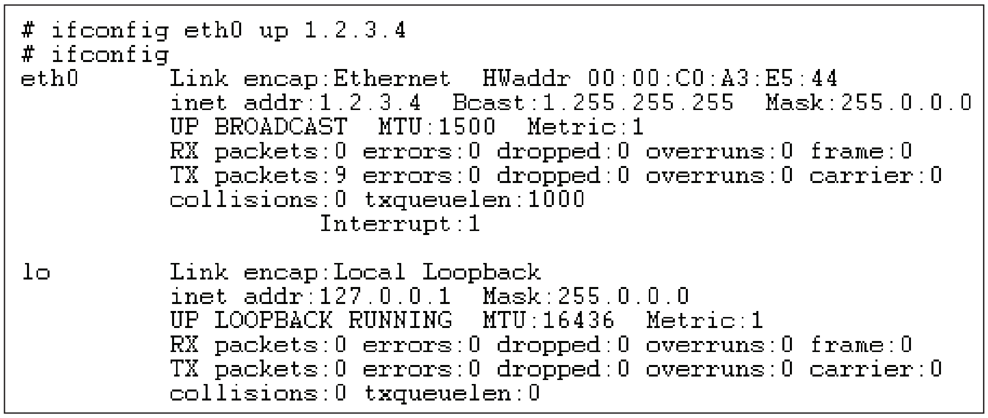
The file system is read-only, except for the `/var` directory and its sub-directories. New directories can be created under the `/var` directory using the **`mkdir`** command. Files under the `/var` directory structure can be created or edited using the **`vi`** editor. Text files can be viewed using the **`cat`** command.

This μClinux 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 turn on the ethernet port 0, `eth0`, and assign the board IP address to `1.2.3.4`, the command, **`ifconfig eth0 up 1.2.3.4`**, is issued.

To confirm the ethernet configuration settings, the command, **`ifconfig`**, is issued.

The result on a HyperTerminal of the configuration settings for the `eth0` (Ethernet) and `lo` (Local Loopback) ports are as shown in Figure 2-4.



```
# ifconfig eth0 up 1.2.3.4
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:C0:A3:E5:44
          inet addr:1.2.3.4  Bcast:1.255.255.255  Mask:255.0.0.0
          UP BROADCAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
                    Interrupt:1

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
```

UG258_02_04_062306

Figure 2-4: **Ethernet Configuration Settings**

To *ping* a remote computer at IP address `1.2.3.9` from the development board, this example command string, **`ping -c 4 1.2.3.9`**, is used to *ping* the remote computer 4 times.

To *telnet* from a networked computer to the board, issue the command, **`telnet 1.2.3.4`**.

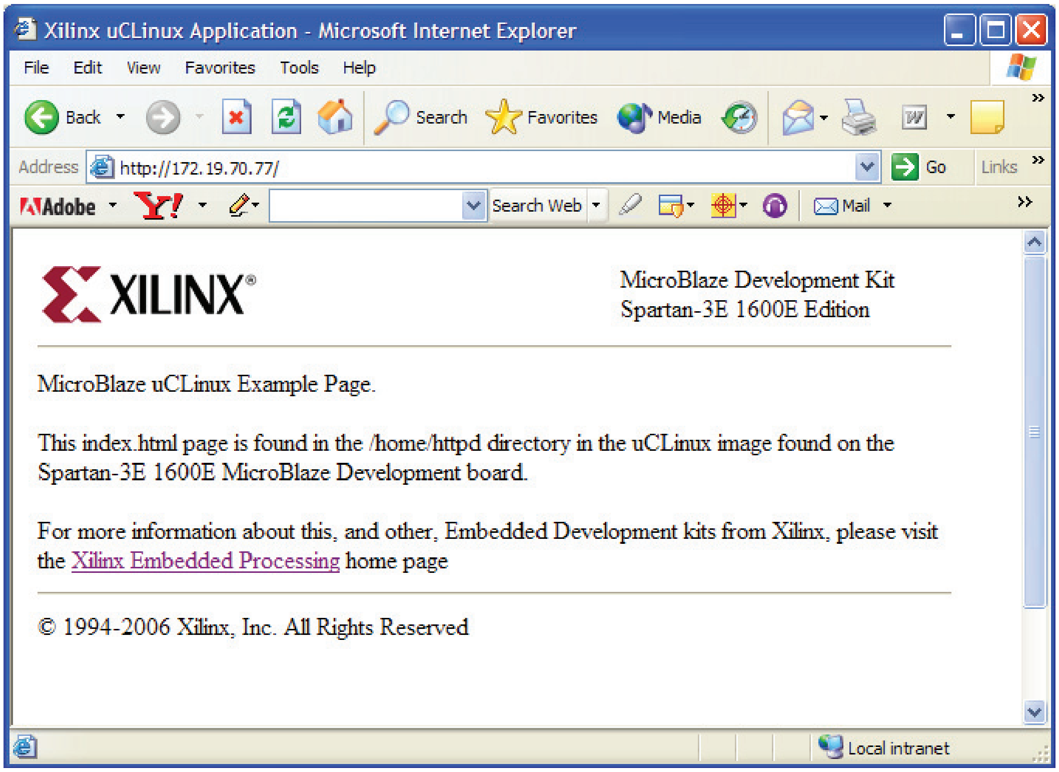
> The user will login as is done from the HyperTerminal, using the login username of **`root`** and the password of **`root`**. All of the Linux commands can now be performed remotely as if the user were logged into the console on a HyperTerminal.

To invoke `ftp` from a networked computer to the board (for transferring files), the command, **`ftp 1.2.3.4`**, is issued on a command shell window on the remote computer:

> The user will login as is done from the HyperTerminal using the login username of **`root`** and the passwd of **`root`**. The basic `ftp` commands work for transferring files.

From a Web browser on a networked or remote computer, the example index.html web page which has been integrated into the kernel image, can be opened by entering the URL **`http://1.2.3.4/`**, as shown in Figure 2-5

The location of the example index.html in the µClinux file system is `/home/httpd/index.html`


Xilinx uCLinux Application - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
Go
Links
Adobe
Search Web
Mail
Local intranet

Address: http://172.19.70.77/

XILINX®

MicroBlaze Development Kit
Spartan-3E 1600E Edition

MicroBlaze uCLinux Example Page.

This index.html page is found in the /home/httpd directory in the uCLinux image found on the Spartan-3E 1600E MicroBlaze Development board.

For more information about this, and other, Embedded Development kits from Xilinx, please visit the Xilinx Embedded Processing home page

© 1994-2006 Xilinx, Inc. All Rights Reserved

UG258_02_05_062806

Figure 2-5: **Spartan-3E 1600E Edition Sample Web Page**

Chapter 3

MicroBlaze Development Kit Web Server Reference System

Introduction

This reference system demonstrates a Web server running on the MicroBlaze™ soft processor, designed using the Embedded Development Kit (EDK). This system is designed for the Spartan™-3E 1600E development board.

To run the Web server demonstration, the Web server software application executable and memory file system (MFS) image must be downloaded into the DDR memory. The methods for running the Web server demonstration are:

- By powering up with the FPGA configuration mode pins set to SPI mode
- By depressing the PROG button on the board with the FPGA configuration mode pins set to SPI mode
- By using a debugger, such as XMD (provided as part of the EDK tools), and downloading the Web server application and MFS image files directly into DDR, through the MicroBlaze Debug Module.

The Web server application and the MFS image are pre-loaded into the parallel NOR StrataFlash on the board. The SPI serial flash is pre-loaded with a bootloader application that loads the application and MFS image into the DDR memory and executes the Web server software application.

Details for obtaining the flash memory files and the procedure for how to re-program the flash memory can be found on the following Spartan-3E Web page link:

http://www.xilinx.com/s3e1600e

Hardware Specifics

This reference system targets the Spartan-3E 1600E Edition development board. The system uses the MicroBlaze processor with a cache turned on for both the instruction cache (I-cache) and the data cache (D-cache). As shown in Figure 3-1, the system also includes the MCH OPB DDR memory controller, the OPB Ethernet MAC, the OPB EMC, the OPB GPIO, and the OPB UART Lite IP Cores. An OPB INTC is included to handle multiple interrupts. An OPB Timer is also needed for XilKernel.

See Table 3-1 for the address map of the system.

Block Diagram

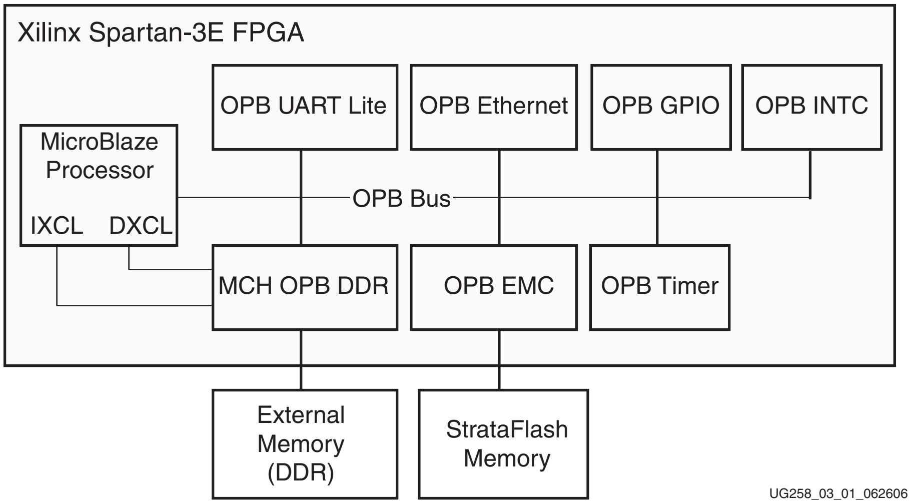


Figure 3-1: **Web Server Reference System Block Diagram**

Address Map

The address map for the IP cores in the reference system is given in Table 3-1.

Table 3-1: **Reference System Address Map**

Instance	Peripheral	Base Address	High Address
debug_module	opb_mdm	0x41400000	0x4140FFFF
dlmb_cntlr	lmb_bram_if_cntlr	0x00000000	0x00001FFF
ilmb_cntlr	lmb_bram_if_cntlr	0x00000000	0x00001FFF
RS232_DTE	opb_uartlite	0x40600000	0x4060FFFF
LEDs_8Bit	opb_gpio	0x40020000	0x4002FFFF
DIP_Switches_4Bit	opb_gpio	0x40000000	0x4000FFFF
FLASH_16Mx8	opb_emc	0x21000000	0x21FFFFFF
DDR_SDRAM_32Mx16	mch_opb_ddr	0x22000000	0x23FFFFFF
Ethernet_MAC	opb_ethernet	0x40C00000	0x40C0FFFF
opb_timer_1	opb_timer	0x41C00000	0x41C0FFFF
opb_intc_0	opb_intc	0x41200000	0x4120FFFF

System Configuration

This reference system runs off a reference clock frequency of 66.67MHz from the auxiliary oscillator on the board. The OPB BUS, Ethernet MAC, and MicroBlaze processor are run at 66.67MHz, while the DDR is runs at 133MHz.

The *MicroBlaze Development Kit Spartan-3E 1600E Edition Board User Guide*, which is located on the Development Kit Reference CD supplied in this kit, provides additional information on board specifics and UCF location constraints for the different cores used in this reference system.

MicroBlaze Processor and MCH OPB DDR Configuration

The MicroBlaze processor with the MCH OPB DDR memory controller is configured with the instruction cache and data cache enabled. A cache size of 8KB is set for both the instruction cache (I-cache) and the data cache (D-cache). The MCH interfaces of the MCH OPB DDR memory controller are connected to the MicroBlaze processor and configured to use the XCL protocol.

The Instruction Cache and Data Cache sections in the *MicroBlaze Processor Reference Guide* provide more details on the MicroBlaze processor caches.

OPB Ethernet Configuration

The OPB Ethernet is set to use no DMA functionality, as this is not required for the Web server demonstration. The OPB Ethernet is set to use interrupts. In the demonstration, the Ethernet MAC is running at 100 Mb/s. No other special settings are needed.

OPB EMC Configuration

The OPB EMC memory controller is connected to an external StrataFlash PROM, which is used to store the Web server software application and MFS image.

OPB UART Lite Configuration

The OPB UART Lite core is configured to use interrupts. It is set up to use the DTE-Style serial port with the baud rate set to **115200**. This requires the use of a Null-modem serial cable.

OPB GPIO Configuration

In the system, eight GPIO output bits are connected to the LEDs on the board and four input GPIO bits are connected to the DIP switches.

Web Server Software Application

The Web server source code is located in the project’s `/WebServer` directory. The lwIP and XilMFS libraries accessed by the Web server design are included by the Library Generator (LibGen) utility and have been added to the project via the Software Platform Settings dialog.

On this system, the Web server is running HTTP 1.1. A file system, built using the LibXil MFS library, stores the files for the Web page. The Web page source, including all HTML files and images, is located in the project’s `/WebPage` directory. The server receives requests at port 80. Every request is processed, and replies are sent by the server to the client.

Operations Performed by the Web Server

The Web server displays a Web page in which the following operations can be performed:

- **HEX Number LED Display** – When a two-digit HEX number is typed in the Web page, it is displayed as an eight-bit binary number on the LEDs on the board when submitted via the Submit button.
- **Image Hosting** – The Web server can process images in GIF and JPG formats stored in the memory file system.
- **DIP Switches** – The binary value of the DIP switches is displayed on the Web page when the Web page is loaded. This value is updated when the Web page is reloaded.
- **Host Adobe Acrobat PDF files** – The Web server can process PDF files stored in the memory file system.
- **Custom Commands** – The Web server reference design implements several commands, accessible via the .xwscmd extension. These commands can be used to view the value of the DIP switches, change the LED values, etc.

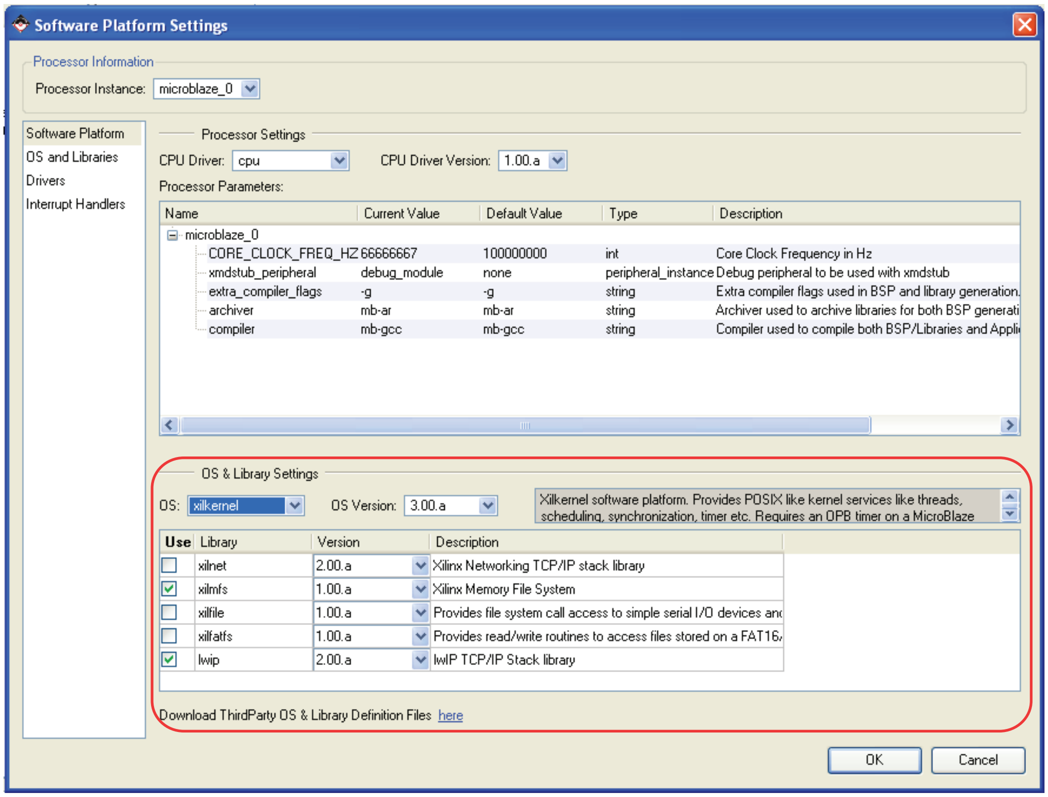
Software Platform Settings

The EDK contains software library support for networking and memory file systems. The Web server design uses the lwIP TCP/IP stack, the Xilinx MicroKernel (XMK) operating system, and the XilMFS library.

To view the library settings for the Web server design, the following steps are followed:

1. To open the Software Platform Settings dialog box, select **Software** → **Software Platform Settings** in XPS.

2. Note the OS & Library Settings table on the lower half of the Software Platform section of the Software Platform Settings dialog box. The table indicates that the lwIP and XilMFS libraries are used in the design and that XilKernel is selected. The OS and library settings are shown in Figure 3-2.



Processor Information

Processor Instance: microblaze_0

Processor Settings

CPU Driver: cpu   CPU Driver Version: 1.00.a

Processor Parameters:

Name	Current Value	Default Value	Type	Description
microblaze_0				
CORE_CLOCK_FREQ_HZ	66666667	100000000	int	Core Clock Frequency in Hz
xmdstub_peripheral	debug_module	none	peripheral_instance	Debug peripheral to be used with xmdstub
extra_compiler_flags	-g	-g	string	Extra compiler flags used in BSP and library generation.
archiver	mb-ar	mb-ar	string	Archiver used to archive libraries for both BSP generati
compiler	mb-gcc	mb-gcc	string	Compiler used to compile both BSP/Libraries and Appli

OS & Library Settings

OS: xilkernel   OS Version: 3.00.a

Xilkernel software platform. Provides POSIX like kernel services like threads, scheduling, synchronization, timer etc. Requires an OPB timer on a MicroBlaze

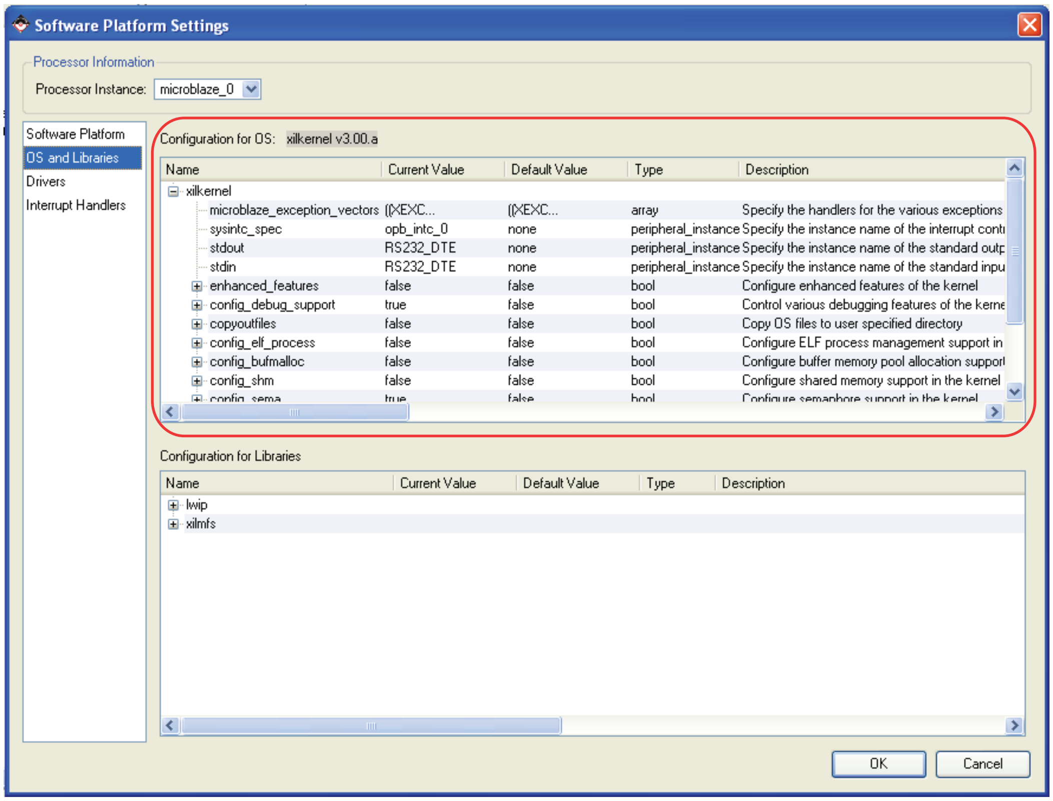
Use	Library	Version	Description
□	xilnet	2.00.a	Xilinx Networking TCP/IP stack library
☑	xilmfs	1.00.a	Xilinx Memory File System
□	xilfile	1.00.a	Provides file system call access to simple serial I/O devices anc
□	xilfatfs	1.00.a	Provides read/write routines to access files stored on a FAT16,
☑	lwip	2.00.a	lwIP TCP/IP Stack library

Download ThirdParty OS & Library Definition Files here

UG258_03_02_062306

Figure 3-2: **OS & Library Settings Table in Software Platform Settings Dialog Box**

3. Select the OS & Libraries section of the Software Platform Settings dialog box. The top half of this section is the configuration settings for XilKernel, as shown in Figure 3-3.



Processor Instance: microblaze_0

Configuration for OS: xilkernel v3.00.a

Name	Current Value	Default Value	Type	Description
xilkernel				
microblaze_exception_vectors	((XEXC...	((XEXC...	array	Specify the handlers for the various exceptions
sysintc_spec	opb_intc_0	none	peripheral_instance	Specify the instance name of the interrupt contı
stdout	RS232_DTE	none	peripheral_instance	Specify the instance name of the standard outp
stdin	RS232_DTE	none	peripheral_instance	Specify the instance name of the standard inpu
enhanced_features	false	false	bool	Configure enhanced features of the kernel
config_debug_support	true	false	bool	Control various debugging features of the kerne
copyoutfiles	false	false	bool	Copy OS files to user specified directory
config_elf_process	false	false	bool	Configure ELF process management support in
config_bufmalloc	false	false	bool	Configure buffer memory pool allocation support
config_shm	false	false	bool	Configure shared memory support in the kernel
config_sema	true	false	bool	Configure semaphore support in the kernel

Configuration for Libraries

Name	Current Value	Default Value	Type	Description
lwip				
xilmfs				

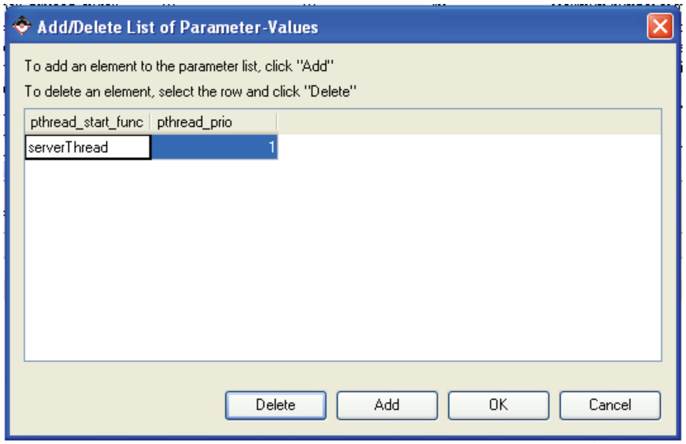
UG258_03_03_062306

Figure 3-3: **XilKernel Settings in the Software Platform Settings Dialog Box**

The following steps are used to obtain more details on some of the XilKernel settings for this reference system:

a. Under the xilkernel menu, note that the stdout and stdin attributes are set to the UART Lite instance name in the project. Also, note that the interrupt controller instance name is specified for the sysintc_spec attribute.

b. Expand the config_pthread_support menu. Click on the Current Value of the static_pthread_table attribute to bring up the Add/Delete List of Parameter-Values box. This table, shown in Figure 3-4, lists threads that are invoked by XilKernel, when the kernel is launched. Note that there is one thread in the table. This thread, serverThread, begins the Web server main tasks.



Add/Delete List of Parameter-Values

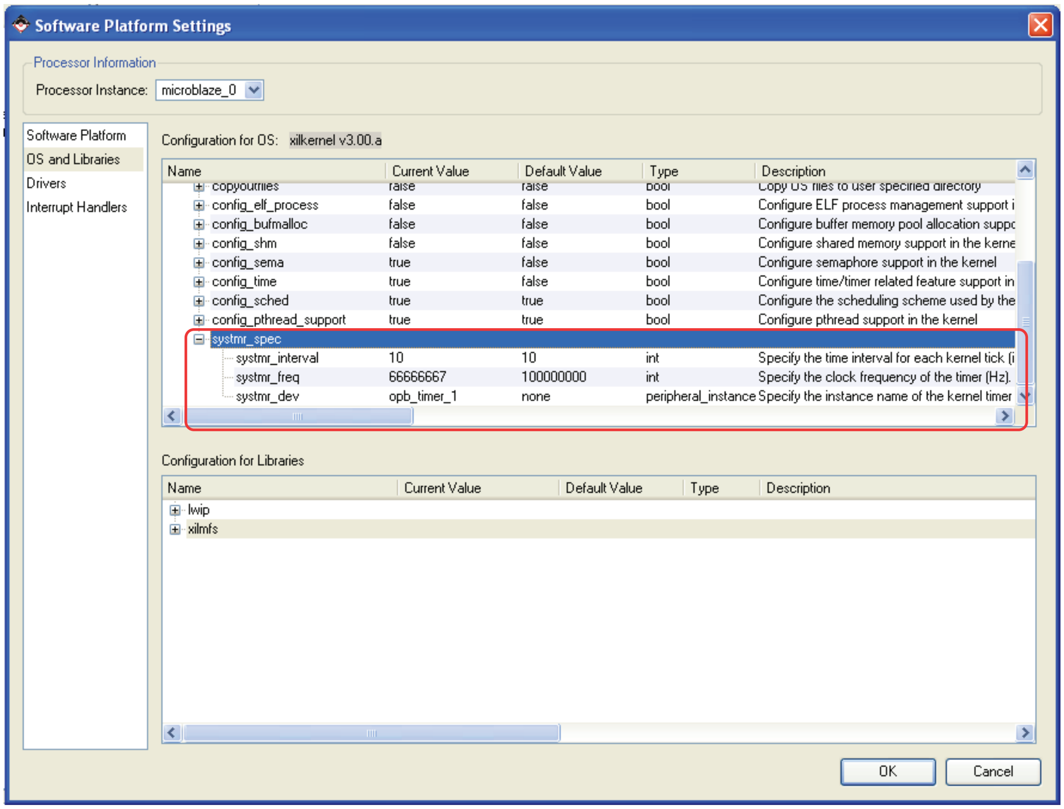
To add an element to the parameter list, click "Add"
To delete an element, select the row and click "Delete"

pthread_start_func	pthread_prio
serverThread	1

UG258_03_04_062306

Figure 3-4: **Static Pthread Table**

c. Expand the systmr_spec menu. The menu expansion is shown in Figure 3-5. Note that the systmr_dev value is specified to be the timer instance name in the project. This must be specified for MicroBlaze processor systems.



Software Platform Settings

Processor Information

Processor Instance: microblaze_0

Configuration for OS: xilkernel v3.00.a

Name	Current Value	Default Value	Type	Description
⊞ copyoutfiles	false	false	bool	Copy OS files to user specified directory
⊞ config_elf_process	false	false	bool	Configure ELF process management support i
⊞ config_bufmalloc	false	false	bool	Configure buffer memory pool allocation suppc
⊞ config_shm	false	false	bool	Configure shared memory support in the kerne
⊞ config_sema	true	false	bool	Configure semaphore support in the kernel
⊞ config_time	true	false	bool	Configure time/timer related feature support in
⊞ config_sched	true	true	bool	Configure the scheduling scheme used by the
⊞ config_pthread_support	true	true	bool	Configure pthread support in the kernel
⊟ systmr_spec				
systmr_interval	10	10	int	Specify the time interval for each kernel tick (i
systmr_freq	66666667	100000000	int	Specify the clock frequency of the timer (Hz).
systmr_dev	opb_timer_1	none	peripheral_instance	Specify the instance name of the kernel timer

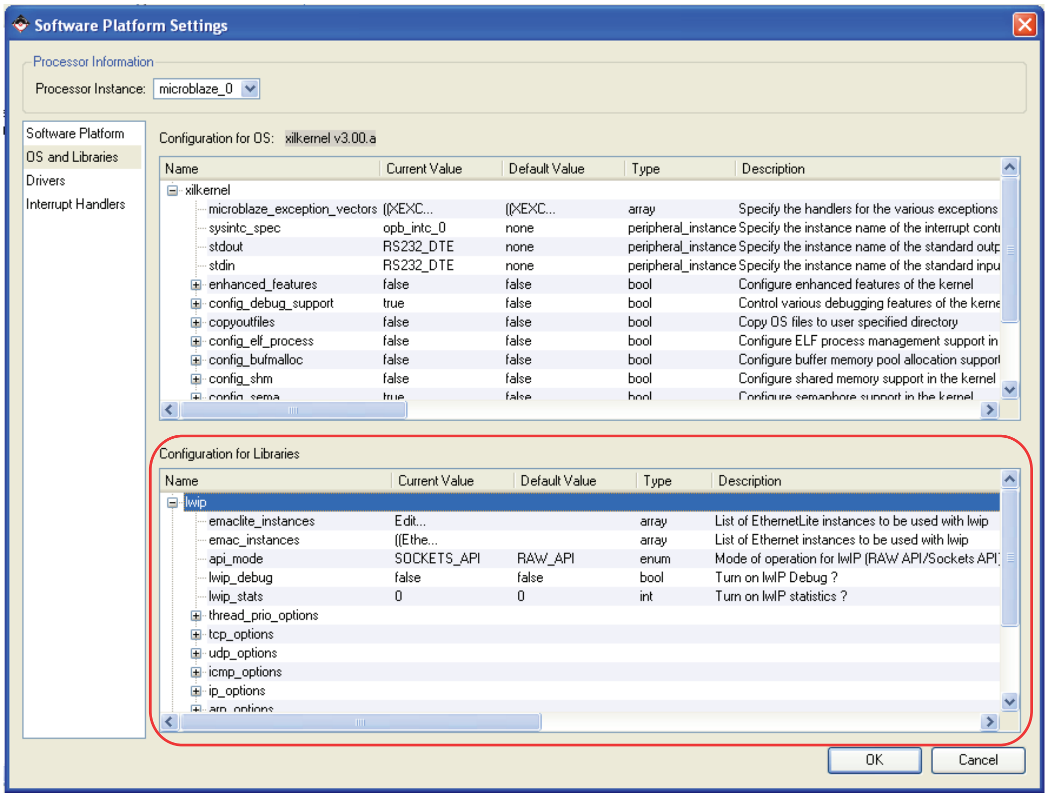
Configuration for Libraries

Name	Current Value	Default Value	Type	Description
⊞ lwip				
⊞ xilmfs				

UG258_03_05_062606

Figure 3-5: **System Timer Menu**

4. The bottom half of the OS & Libraries section is the configuration for the included libraries, lwIP and XilMFS. Expand the lwip menu to view the settings for the lwIP library. The lwIP library settings are shown in Figure 3-6.



Software Platform Settings

Processor Information

Processor Instance: microblaze_0

Configuration for OS: xilkernel v3.00.a

Name	Current Value	Default Value	Type	Description
⊟ xilkernel				
microblaze_exception_vectors	((XEXC...	((XEXC...	array	Specify the handlers for the various exceptions
sysintc_spec	opb_intc_0	none	peripheral_instance	Specify the instance name of the interrupt conti
stdout	RS232_DTE	none	peripheral_instance	Specify the instance name of the standard outp
stdin	RS232_DTE	none	peripheral_instance	Specify the instance name of the standard inpu
⊞ enhanced_features	false	false	bool	Configure enhanced features of the kernel
⊞ config_debug_support	true	false	bool	Control various debugging features of the kerne
⊞ copyoutfiles	false	false	bool	Copy OS files to user specified directory
⊞ config_elf_process	false	false	bool	Configure ELF process management support in
⊞ config_bufmalloc	false	false	bool	Configure buffer memory pool allocation support
⊞ config_shm	false	false	bool	Configure shared memory support in the kernel
⊞ config_sema	true	false	bool	Configure semaphore support in the kernel

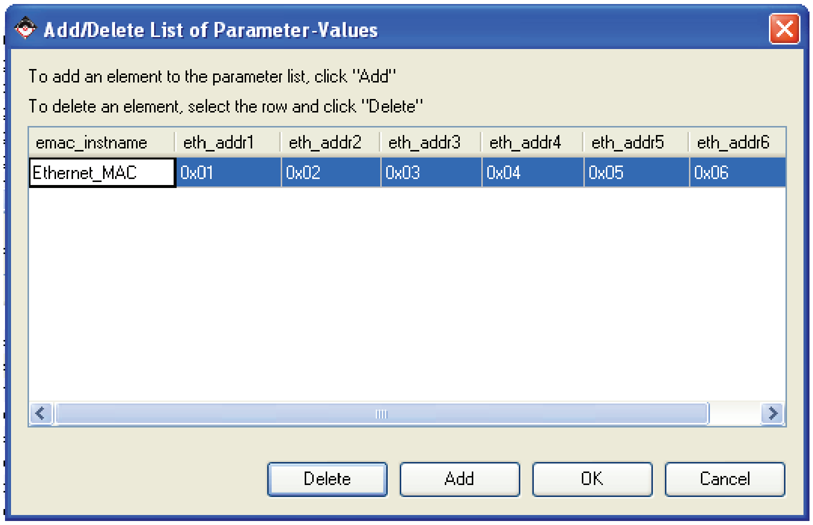
Configuration for Libraries

Name	Current Value	Default Value	Type	Description
⊟ lwip				
emaclite_instances	Edit...		array	List of EthernetLite instances to be used with lwip
emac_instances	((Ethe...		array	List of Ethernet instances to be used with lwip
api_mode	SOCKETS_API	RAW_API	enum	Mode of operation for lwIP (RAW API/Sockets API)
lwip_debug	false	false	bool	Turn on lwIP Debug ?
lwip_stats	0	0	int	Turn on lwIP statistics ?
⊞ thread_prio_options				
⊞ tcp_options				
⊞ udp_options				
⊞ icmp_options				
⊞ ip_options				
⊞ arp_options				

UG258_03_06_062306

Figure 3-6: **LwIP Settings**

Click the Current Value field for the emac_instances attribute to bring up the Add/Delete List of Parameter-Values box, shown in Figure 3-7. Note that the table has the Ethernet_MAC instance with a default MAC address of 01-02-03-04-05-06.



Add/Delete List of Parameter-Values

To add an element to the parameter list, click "Add"

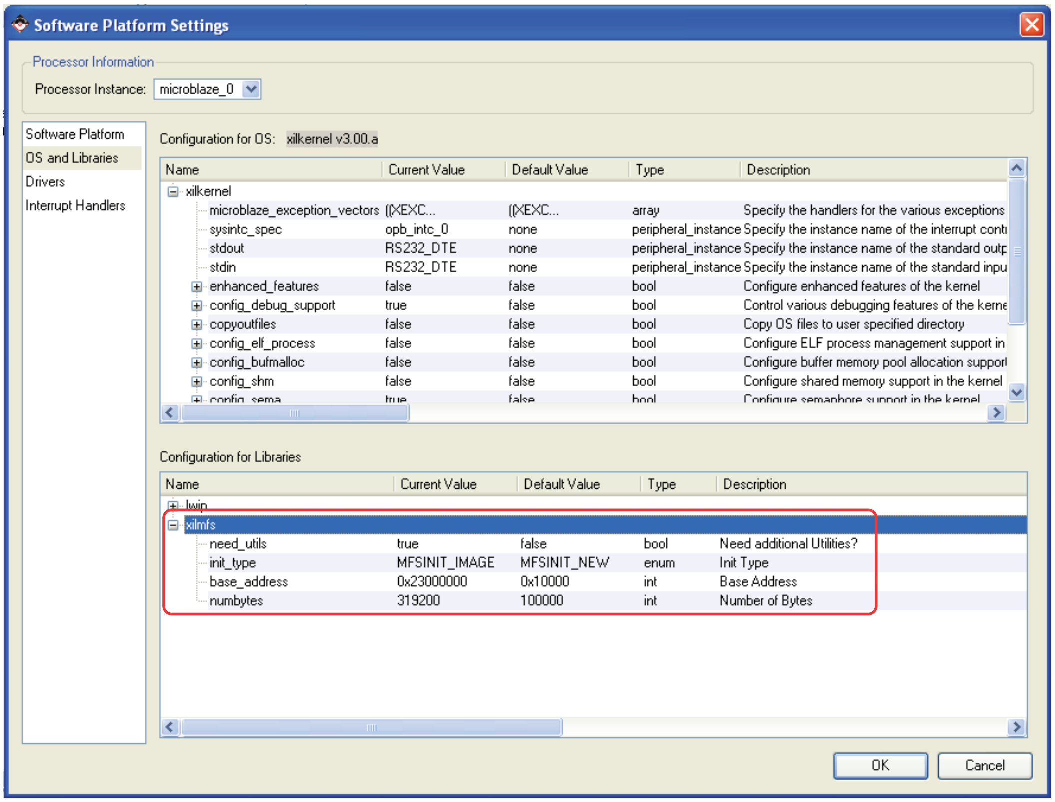
To delete an element, select the row and click "Delete"

emac_instname	eth_addr1	eth_addr2	eth_addr3	eth_addr4	eth_addr5	eth_addr6
Ethernet_MAC	0x01	0x02	0x03	0x04	0x05	0x06

UG258_03_07_062306

Figure 3-7: **Ethernet Instance**

5. To view the XilMFS configuration settings, expand the xilmfs menu. The XilMFS settings are shown in Figure 3-8. Note that the init_type field is set to MFSINIT_IMAGE. This creates a file system based on a pre-loaded memory image. The base_address field contains the starting address for the file system memory. The field is set to `0x23000000`, which is in DDR memory space. The numbytes field contains the size of the memory file system.



Processor Instance: microblaze_0

Configuration for OS: xilkernel v3.00.a

Name	Current Value	Default Value	Type	Description
xilkernel				
microblaze_exception_vectors	((XEXC...	((XEXC...	array	Specify the handlers for the various exceptions
sysintc_spec	opb_intc_0	none	peripheral_instance	Specify the instance name of the interrupt conti
stdout	RS232_DTE	none	peripheral_instance	Specify the instance name of the standard outp
stdin	RS232_DTE	none	peripheral_instance	Specify the instance name of the standard inpu
enhanced_features	false	false	bool	Configure enhanced features of the kernel
config_debug_support	true	false	bool	Control various debugging features of the kerne
copyoutfiles	false	false	bool	Copy OS files to user specified directory
config_elf_process	false	false	bool	Configure ELF process management support in
config_bufmalloc	false	false	bool	Configure buffer memory pool allocation support
config_shm	false	false	bool	Configure shared memory support in the kernel
config_sema	true	false	bool	Configure semaphore support in the kernel

Configuration for Libraries

Name	Current Value	Default Value	Type	Description
lwip				
xilmfs				
need_utils	true	false	bool	Need additional Utilities?
init_type	MFSINIT_IMAGE	MFSINIT_NEW	enum	Init Type
base_address	0x23000000	0x10000	int	Base Address
numbytes	319200	100000	int	Number of Bytes

UG258_03_08_062306

Figure 3-8: **XilMFS Settings**

Executing the Web Server Reference System

This section details the steps necessary to run the Web server demonstration. A pre-built bitstream, compiled software application, and MFS image are in the `ready_for_download` directory under the project root directory. These files can be used to run the reference system, or the files can be generated manually. The following steps include information on how to generate and download the hardware bitstream, create an MFS image, download and run the Web server software application, and set up both the Web server and Web client.

Generating the Bitstream

To generate the system netlist and bitstream, follow these steps:

1. Open the Web server project in XPS.
2. Generate the bitstream by selecting **Hardware** → **Generate Bitstream** in XPS.

Generating the MFS Image

The Web Server reference design requires that a memory file system be initialized for download into the DDR memory before the design can be executed. The MFS image must be generated before the Web Server is downloaded.

To generate the MFS image, follow these steps:

1. Open an EDK shell by selecting **Project → Launch EDK Shell...** in XPS.
2. Change the directory to the project's `/WebPage` directory with the following command:

   ```
   cd WebPage
   ```
3. Remove any preexisting MFS images in the project's `/WebServer` directory by using the following command:

   ```
   rm ../WebServer/*.mfs
   ```
4. Generate the MFS image as follows:

   ```
   mfsgen -cvbfs ../WebServer/image.mfs 600 404.html index.html
   logoV2005.gif webserverpdf.pdf
   ```

   After the command executes, the following text will appear in the EDK shell window:

   ```
   mfsgen
   Xilinx EDK 8.1.02 EDK_I.20.4
   Copyright (c) 2004 Xilinx, Inc. All rights reserved.

   404.html 710
   index.html 1838
   logoV2005.gif 1148
   webserverpdf.pdf 50979
   MFS block usage (used / free / total) = 110 / 490 / 600
   Size of memory is 319200 bytes
   Block size is 532
   mfsgen done!
   ```

   The command line options and syntax for the **`mfsgen`** command are listed in the *OS and Libraries Document Collection* document, in the EDK documentation directory.
5. Exit the EDK shell.

Compiling the Web Server Code and Downloading the Bitstream

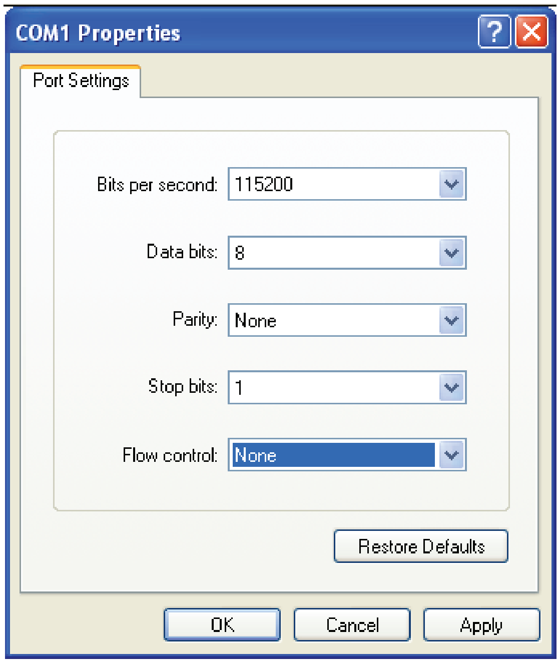
To compile the Web server software application code and configure the FPGA, follow these steps:

1. In XPS, select **Software → Build All User Applications** to compile the software application.
2. Connect the USB cable from the PC to the board.
3. Select **Device Configuration → Download Bitstream** in XPS. The bitstream is downloaded to the board by the iMPACT tool. When the FPGA device is configured, the DONE light is illuminated on the board.

Running the Web Server Demonstration

A serial communications utility, such as HyperTerminal, must be set up before the Web server demonstration is executed. To configure HyperTerminal and run the Web server demonstration on the MicroBlaze processor, follow these steps:

1. Connect a serial cable from the COM port of the PC to the DTE serial port on the board.
2. Set a HyperTerminal or similar program to a Baud Rate of **115200**, Data Bits to **8**, Parity to **None** and Flow Control to **None**. See Figure 3-9 for the HyperTerminal settings.



Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

UG258_03_09_062306

Figure 3-9: **HyperTerminal Settings**

3. Connect a crossover Ethernet cable to the host PC and the Ethernet port of the board. The USB cable must remain connected from the PC to the board.
4. In XPS, select **Debug** → **Launch XMD.**
5. In the XMD command window, change the directory to the project's `/WebServer` directory as follows:

   ```
   cd WebServer
   ```

6. Download the MFS image into DDR memory using the following command:

   ```
   dow -data image.mfs 0x23000000
   ```

7. Download the Web server executable file using the following command:

   ```
   dow executable.elf
   ```

8. Run the application as follows:

   `run`

   When the Web server application runs, it will output text as the MFS is initialized. This ouput will read as follows:

```
Xilinx Web Server Demonstration
(c) Xilinx, Inc. 2005

System Initialization in progress:

Initializing MFS...done
404.html 000002c6
index.html 0000072e
logoV2005.gif 0000047c
webserverpdf.pdf 0000c723
Initializing GPIO...done
Initializing the lwIP library...
Calling tcpip_init
Waiting tcpip_init
tcpip_init_done : Called
xemacif_pkt_process_thread created
lwIP initialization done
```

   After the initialization is complete, the Web server application will output the following text:

```
##################################################################
# Xilinx lwIP TCP/IP Demo Application (Web Server)                 #
#                                                                  #
# XILINX IS PROVIDING THIS DESIGN, CODE, OR INFORMATION "AS IS"    #
# SOLELY FOR USE IN DEVELOPING PROGRAMS AND SOLUTIONS FOR          #
# XILINX DEVICES.                                                  #
#                                                                  #
# (c) Copyright 2005 Xilinx, Inc.                                  #
# All rights reserved                                              #
#                                                                  #
##################################################################


This demonstration requires manual entry of your IP configuration
settings. Please fill in the following fields:

IP Address :
```

Configuring the Web Server and Web Client

The Web server demonstration requires manual entry of the IP settings. To configure both the server and client to run the demonstration, follow these steps:

1. In the HyperTerminal window, fill in the IP settings for the Web server. An example is to use an IP address of **1.2.3.4**, a subnet mask of **255.255.255.0**, and a gateway of **1.2.3.1**. In HyperTerminal, these settings will appear as the following text:

```
This demonstration requires manual entry of your IP configuration
settings. Please fill in the following fields:

IP Address : 1.2.3.4
Is the address 1.2.3.4 correct? (y/n)
Subnet Mask: 255.255.255.0
Is the address 255.255.255.0 correct? (y/n)
Gateway    : 1.2.3.1
Is the address 1.2.3.1 correct? (y/n)
```

Note: For the IP address, subnet mask, and gateway, a period is used to seperate the fields. Input a period to move to the next field. After each setting is input, the Web server application will ask the user to accept the setting. Type "y" to accept the setting.

After the IP configuration settings have been input, the Web server demonstration will output text to the HyperTerminal stating that Web server is accepting connections. This text will appear similar to the following:

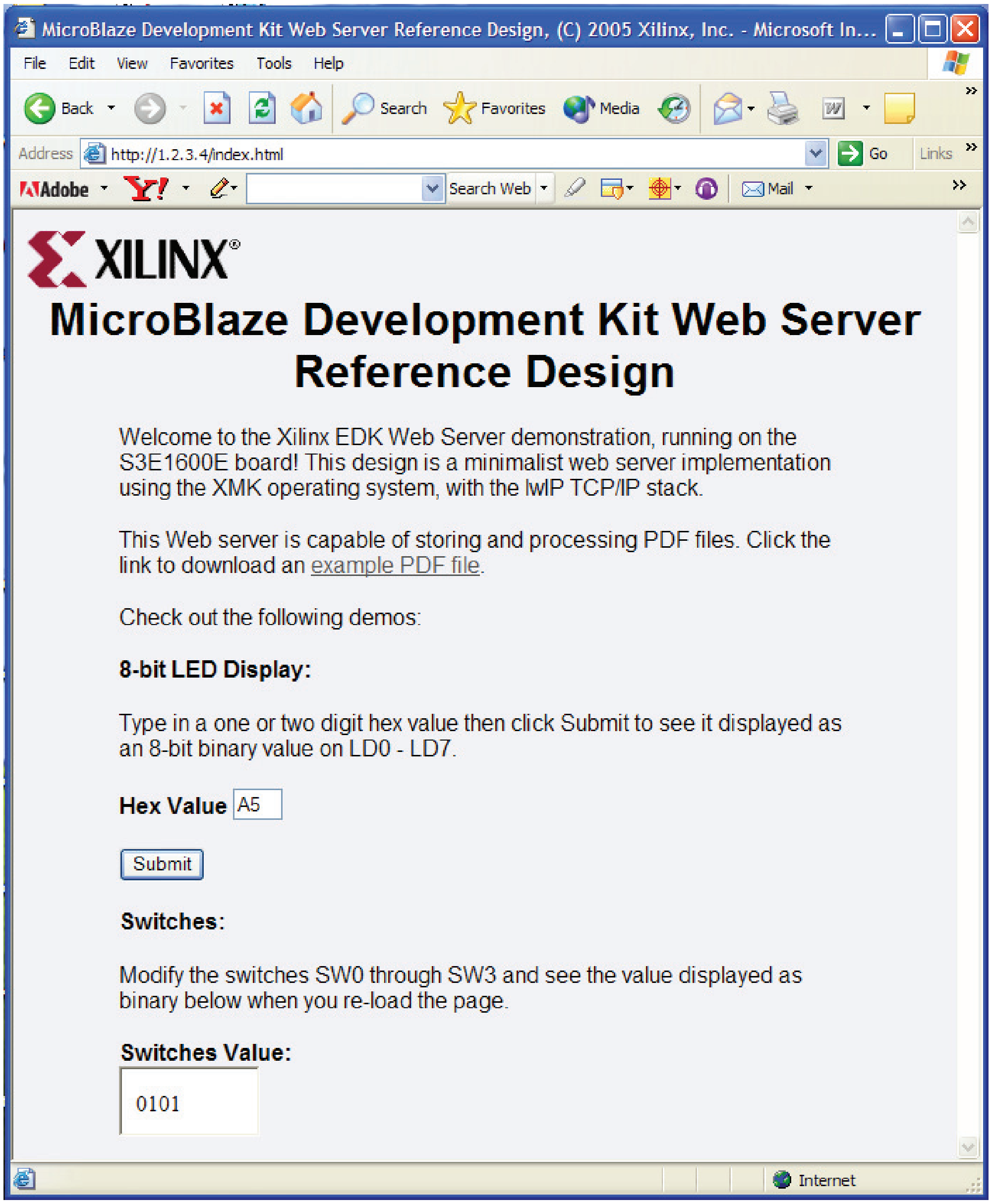
```
##################################################################
# Xilinx lwIP TCP/IP Demo Application (Web Server)                 #
#                                                                  #
# Web Server is now configured and accepting connections.          #
#                                                                  #
##################################################################

MAC Address: 01-02-03-04-05-06
IP Address : 1.2.3.4
Subnet Mask: 255.255.255.0
Gateway    : 1.2.3.1

Server initialization complete.
HTTP server is now accepting new connections on port 80
```

2. Modify the IP address of the host PC so that it is on the same subnet as the Web Server. To work with the example in step 1, the host PC can be set to have an IP address of **1.2.3.9**, a subnet mask of **255.255.255.0**, and a gateway of **1.2.3.1**.

   Note: When changing the IP settings on the PC, note the original property settings, so that the original properties can be reset after the web server demonstration is run.

3. On the host PC, open an HTML browser and point to the http://x.x.x.x URL, where **x.x.x.x** is the IP address specified for the Web Server. For the example used in step 1, use http://1.2.3.4 for the URL.

4. The Web server demo Web page should appear in the browser. The Web page is shown in Figure 3-10. Follow the instructions on the Web page to read the values of the DIP switches and write to the LEDs on the board.


MicroBlaze Development Kit Web Server Reference Design, (C) 2005 Xilinx, Inc. - Microsoft In...
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://1.2.3.4/index.html
Go
Links
Adobe
Search Web
Mail
Internet

XILINX®

MicroBlaze Development Kit Web Server Reference Design

Welcome to the Xilinx EDK Web Server demonstration, running on the S3E1600E board! This design is a minimalist web server implementation using the XMK operating system, with the lwIP TCP/IP stack.

This Web server is capable of storing and processing PDF files. Click the link to download an example PDF file.

Check out the following demos:

8-bit LED Display:

Type in a one or two digit hex value then click Submit to see it displayed as an 8-bit binary value on LD0 - LD7.

Hex Value A5

Submit

Switches:

Modify the switches SW0 through SW3 and see the value displayed as binary below when you re-load the page.

Switches Value:

0101

UG258_03_10_062306

Figure 3-10: **Web Server Reference System Web Page**