XAPP963 (v1.0) November 15, 2006

Using and Creating Flash Files for the MicroBlaze Development Kit - Spartan-3E Edition

Author: Casey Cain

Abstract

This application note describes the files for programming the StrataFlash memory for the MicroBlaze™ Development Kit - Spartan™-3E 1600E Edition. The development kit includes two reference systems, a Web server reference system and a μClinux reference system. These systems are pre-loaded into the Intel StrataFlash on the SP3E1600E board that is included with the development kit. The μClinux system is programmed to run in BPI Up configuration mode and the Web server system is programmed to run in BPI Down configuration mode. All files needed to run the reference systems from the StrataFlash memory are provided for the user. This application note describes how to use the provided files, as well as create new files, to run the Web server reference system and the μClinux reference system from the StrataFlash memory.

Introduction

The SP3E1600E board supports a variety of FPGA configuration options. One of the options is to program the on-board Intel StrataFlash parallel NOR Flash PROM, then configure the FPGA from the image stored in the Flash PROM using either BPI Up or BPI Down configuration modes. For BPI Up mode, the FPGA configuration image source is stored in the StrataFlash parallel Flash PROM, starting at address `0x00000000` and incrementing through address space. For BPI Down mode, the FPGA configuration image source is stored in the StrataFlash parallel Flash PROM, starting at address `0x00FFFFFF` and decrementing through address space. For the SP3E1600E board shipped with the MicroBlaze Development Kit - Spartan-3E 1600E Edition, two reference system configurations are pre-loaded into the StrataFlash PROM to show both BPI Up and BPI Down configuration modes.

Hardware and Software Requirements

The hardware and software requirements are:

- SP3E1600E development board
- USB download cable
- RS232 null modem serial cable and serial communications utility (HyperTerminal)
- Xilinx Platform Studio 8.2.02i
- Xilinx Integrated Software Environment (ISE) 8.2.03i

Using the Provided Flash Files

The files required to run the μClinux demonstration and the Web server demonstration out of the StrataFlash memory are provided at the following link:

- http://www.xilinx.com/sp3e1600e

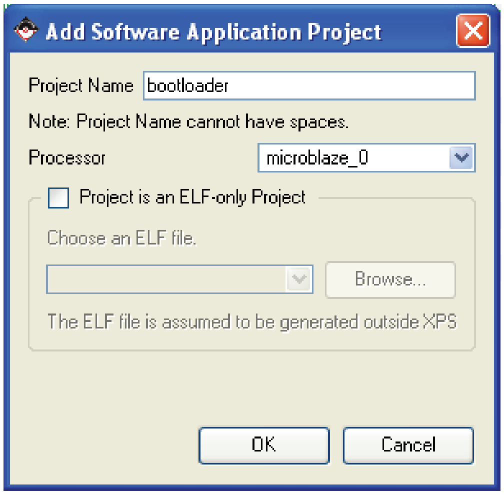
At the above Website, click on the Design Examples link to find the flash files. This section includes details on how to use the provided flash files to boot the μClinux and Web server demonstrations out of the StrataFlash.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Adding the Bootloader Files to the Project

A bootloader is a piece of code that is loaded into a flash PROM. On power-up, the bootloader code is loaded from BRAM. When the bootloader is run, it copies the software executable from the PROM into the DDR memory and starts it running. The provided bootloader files must be added as a software application project to the reference system so that a bitstream can be generated that includes the bootloader executable. To add the files as a new software application project, follow these steps:

1. Open the reference system project in XPS.
2. Select **Software** → **Add Software Application Project...** in XPS. This will open the Add Software Application Project dialog box, which is shown in Figure 1.



Add Software Application Project

Project Name: bootloader

Note: Project Name cannot have spaces.

Processor: microblaze_0

□ Project is an ELF-only Project

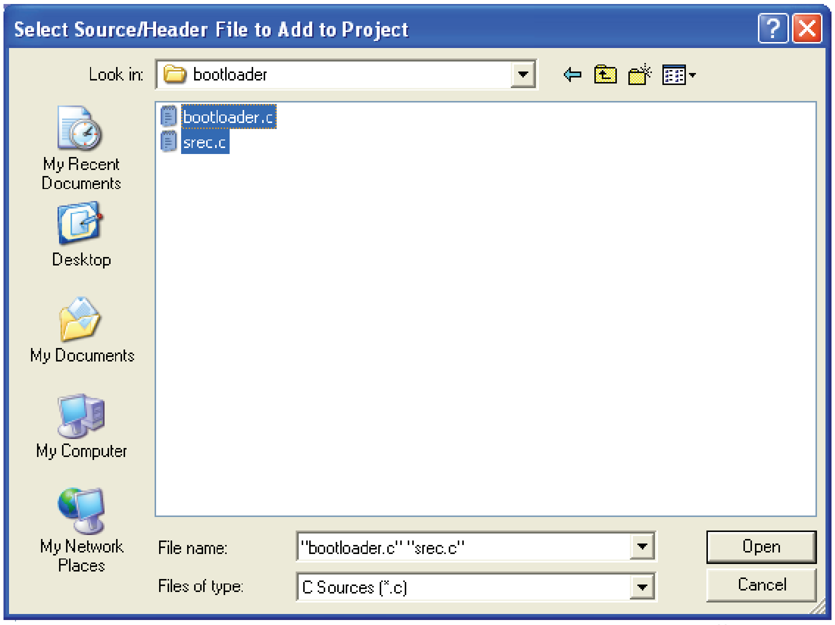
Choose an ELF file.

The ELF file is assumed to be generated outside XPS

X963_01_101006

Figure 1: **Add Software Application Project Dialog Box**

3. In the Software Application Project dialog box, enter a project name for the new bootloader software application project. Click OK.
4. In XPS, select the Applications tab. Right click Sources under the bootloader project and add the existing bootloader C files to the project, using the dialog box shown in Figure 2.



Select Source/Header File to Add to Project

Look in: bootloader

bootloader.c
srec.c

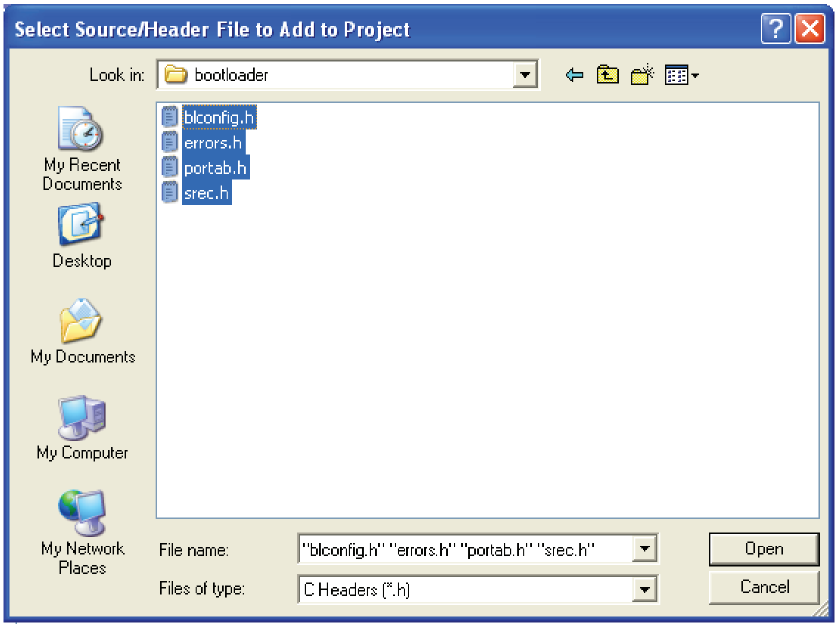
File name: "bootloader.c" "srec.c"

Files of type: C Sources (*.c)

X963_02_101006

Figure 2: **Add the Bootloader Source C Files**

5. Right click Headers under the bootloader project and add the existing bootloader C header files to the project, using the dialog box shown in Figure 3.



Look in: bootloader

blconfig.h
errors.h
portab.h
srec.h

File name: "blconfig.h" "errors.h" "portab.h" "srec.h"

Files of type: C Headers (*.h)

X963_03_101006

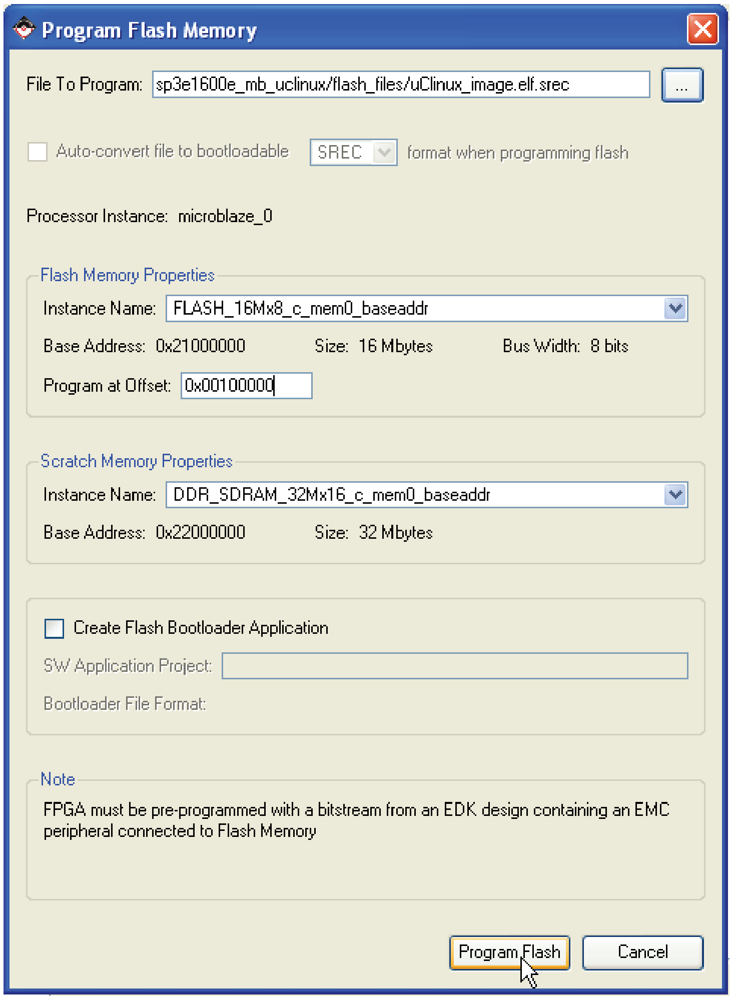
Figure 3: **Add the Bootloader C Header Files**

Loading an SREC File into StrataFlash

An S-record (SREC) file consists of ASCII character strings, specially formatted for loading data into memory. The executable ELF files for the µClinux and Web server reference systems have been converted into bootloadable SREC files. Bootloading ELF files is not typically done because the complexity of the bootloader would be increased. Converting the ELF file to an SREC file simplifies the bootloader and allows the bootloader to be created with the EDK tools. To load an existing SREC file into the StrataFlash, follow these steps:

1. Connect the USB programming cable, the serial cable, and the power supply to the SP3E1600E development board. Power on the board.
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 the provided SREC file. For the µClinux image SREC file, change the program offset to `0x00100000`. For the Web server application SREC file, change the program offset to `0x00700000`. These program offsets are the addresses where the provided bootloaders are programmed to find the SREC files. The Program Flash Memory settings are shown in Figure 4.



Program Flash Memory

File To Program: sp3e1600e_mb_uclinux/flash_files/uClinux_image.elf.srec

□ Auto-convert file to bootloadable SREC format when programming flash

Processor Instance: microblaze_0

Flash Memory Properties

Instance Name: FLASH_16Mx8_c_mem0_baseaddr

Base Address: 0x21000000 Size: 16 Mbytes Bus Width: 8 bits

Program at Offset: 0x00100000

Scratch Memory Properties

Instance Name: DDR_SDRAM_32Mx16_c_mem0_baseaddr

Base Address: 0x22000000 Size: 32 Mbytes

□ Create Flash Bootloader Application

SW Application Project:

Bootloader File Format:

Note

FPGA must be pre-programmed with a bitstream from an EDK design containing an EMC peripheral connected to Flash Memory

X963_04_101006

Figure 4: **Program Flash Memory Dialog Box for an SREC File**

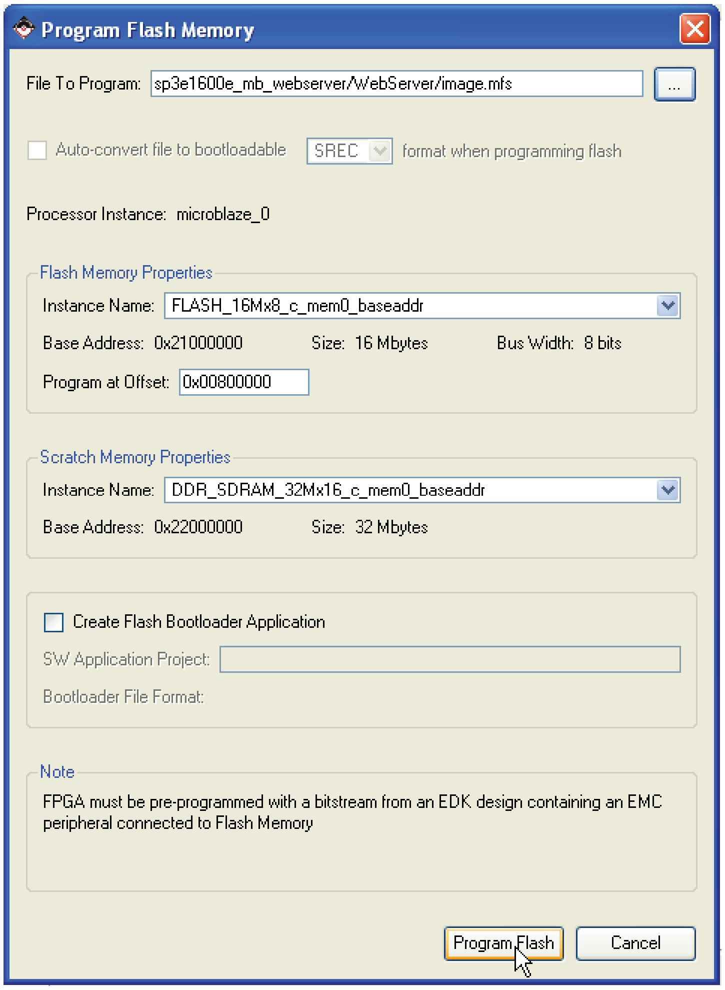
5. Click the Program Flash button.

Loading the Web Server MFS Image into Flash

This section is only relevant to the Web server demonstration, so if running the µClinux demonstration, skip to the next section, "Loading an MCS File into StrataFlash". To run the Web server demonstration properly, the MFS image must be loaded into the StrataFlash. To load the MFS image into the StrataFlash memory, follow these steps:

1. Ensure that the target board is programmed with the hardware bitstream.
2. In XPS, select **Device Configuration** → **Program Flash Memory.**

3. In the Program Flash Memory dialog box, choose the file to program to be the provided MFS image file, `webserver_image.mfs`. Change the program offset to `0x00800000`. This program offset is the address where the provided Web server bootloader is programmed to find the MFS image. The Program Flash Memory settings for the MFS image are shown in Figure 5.



Program Flash Memory

File To Program: sp3e1600e_mb_webserver/WebServer/image.mfs

□ Auto-convert file to bootloadable SREC format when programming flash

Processor Instance: microblaze_0

Flash Memory Properties

Instance Name: FLASH_16Mx8_c_mem0_baseaddr

Base Address: 0x21000000 Size: 16 Mbytes Bus Width: 8 bits

Program at Offset: 0x00800000

Scratch Memory Properties

Instance Name: DDR_SDRAM_32Mx16_c_mem0_baseaddr

Base Address: 0x22000000 Size: 32 Mbytes

□ Create Flash Bootloader Application

SW Application Project:

Bootloader File Format:

Note

FPGA must be pre-programmed with a bitstream from an EDK design containing an EMC peripheral connected to Flash Memory

X963_05_101006

Figure 5: **Program Flash Memory Dialog Box for the Web Server MFS Image**

4. Click the Program Flash button.

Loading an MCS File into StrataFlash

The Xilinx PicoBlaze™ NOR Flash Programmer is used to load an MCS file into the StrataFlash. To download the files for the PicoBlaze NOR Flash Programmer for the SP3E1600E board, go to the following link:

- http://www.xilinx.com/sp3e1600

From that Website, click on the Design Examples link to find the PicoBlaze StrataFlash Programmer files. The PicoBlaze StrataFlash Programmer PDF file documents the steps necessary to load an MCS file into the StrataFlash memory. After following those steps, the MCS file will be loaded into the StrataFlash memory and the Web server demonstration is ready to be run in BPI down mode. Proceed to the section, "Running the Demonstrations out of Flash", to run the demonstration.

Creating New Flash Files for the µClinux Demonstration

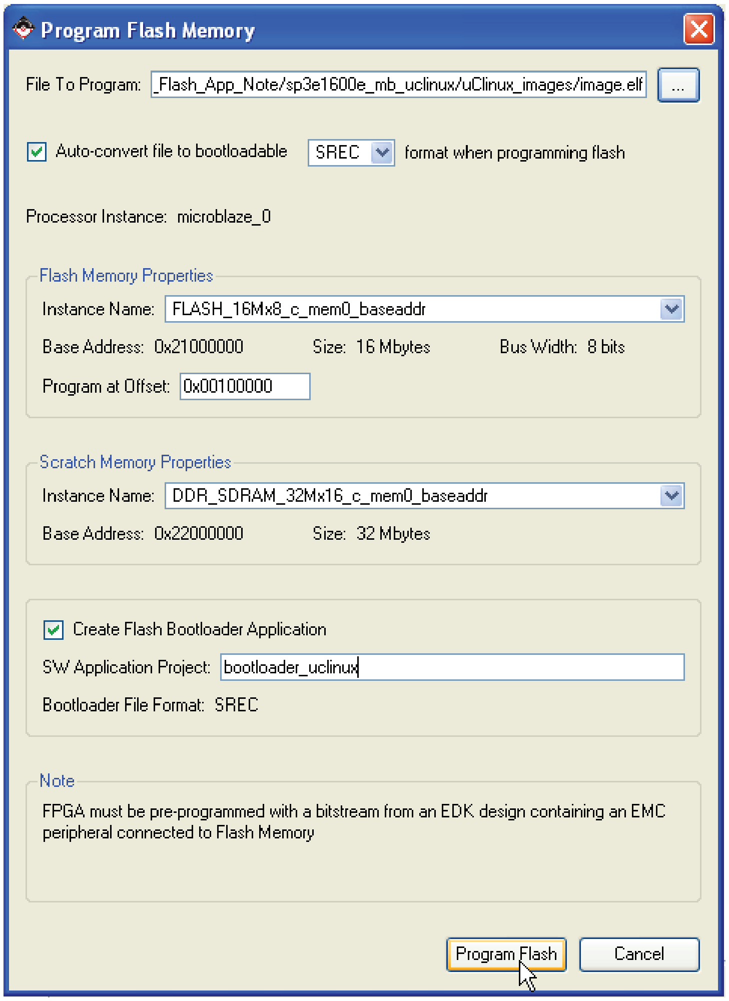
Instead of using the provided flash files, new files can be generated. The µClinux reference system is pre-loaded into the StrataFlash on the SP3E1600E board from the factory to run in BPI Up configuration mode. For more information on the µClinux reference system, refer to UG258, *Getting Started with the MicroBlaze Development Kit - Spartan-3E 1600E Edition.* This section describes how to create new flash files for BPI Up mode using the µClinux reference system.

Creating New Bootloader Files and a New µClinux Image SREC File

Instead of using the existing bootloader files and µClinux SREC file, new files can be created using the EDK tools. The bootloader files must be created because they will become part of the MCS file in the next section, "Generating an MCS File for BPI Up Configuration Mode". The µClinux SREC file will be loaded into the StrataFlash in the following steps in this section. To create new bootloader files and to load a new µClinux image SREC file into the StrataFlash, follow these steps:

1. Open the reference system project in XPS.
2. Connect the USB programming cable, the serial cable, and the power supply to the SP3E1600E development board. Power on the board.
3. 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.
4. In XPS, select **Device Configuration** → **Program Flash Memory.**

5. In the Program Flash Memory dialog box, choose the file to program to be `/uClinux_images/image.elf` under the project root directory. Check the *Auto-convert file to bootloadable SREC format when programming flash* check box. 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`. A bootloader is created by clicking the *Create Flash Bootloader Application* check box in the Program Flash Memory dialog box. The Program Flash Memory settings are shown in Figure 6.



Program Flash Memory

File To Program: _Flash_App_Note/sp3e1600e_mb_uclinux/uClinux_images/image.elf

☑ Auto-convert file to bootloadable SREC format when programming flash

Processor Instance: microblaze_0

Flash Memory Properties

Instance Name: FLASH_16Mx8_c_mem0_baseaddr

Base Address: 0x21000000 Size: 16 Mbytes Bus Width: 8 bits

Program at Offset: 0x00100000

Scratch Memory Properties

Instance Name: DDR_SDRAM_32Mx16_c_mem0_baseaddr

Base Address: 0x22000000 Size: 32 Mbytes

☑ Create Flash Bootloader Application

SW Application Project: bootloader_uclinux

Bootloader File Format: SREC

Note

FPGA must be pre-programmed with a bitstream from an EDK design containing an EMC peripheral connected to Flash Memory

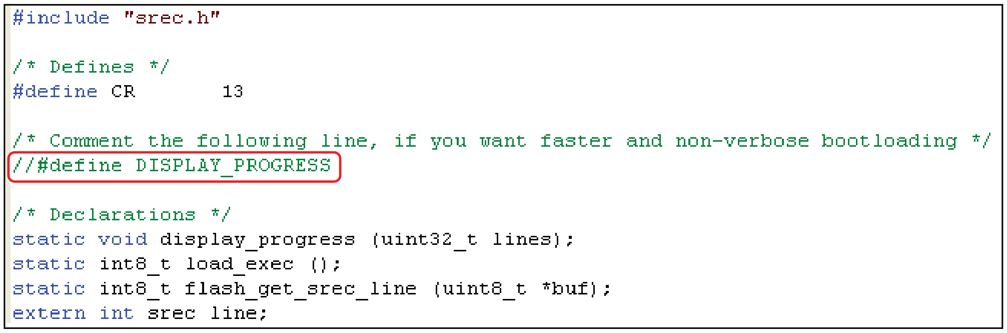
Figure 6: **Program Flash Memory Dialog Box for the μClinux ELF**

6. Click the Program Flash button. This will create a bootloader software application project and program the flash memory with the μClinux image.

7. 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 DISPLAY PROGRESS
   ```

   The line to comment out to allow non-verbose bootloading is shown in Figure 7. After modifying the C file, be sure to recompile the software application.



```
#include "srec.h"

/* Defines */
#define CR       13

/* Comment the following line, if you want faster and non-verbose bootloading */
//#define DISPLAY_PROGRESS

/* Declarations */
static void display_progress (uint32_t lines);
static int8_t load_exec ();
static int8_t flash_get_srec_line (uint8_t *buf);
extern int srec line;
```

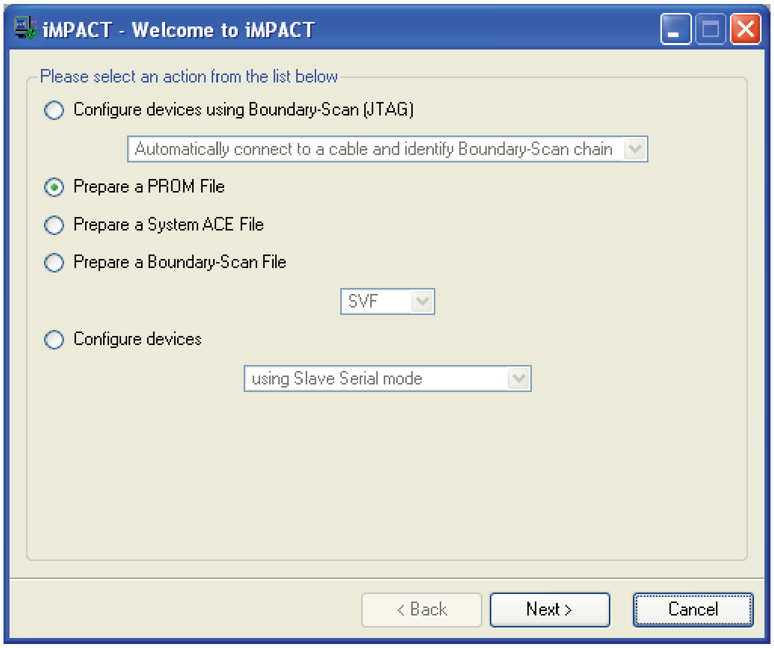
X963_07_101006

Figure 7: **Code to Select Non-Verbose Bootloading**

Generating an MCS File for BPI Up Configuration Mode

An MCS file is used to program the StrataFlash with the system bitstream and bootloader loaded in the bitstream. An MCS file for the μClinux reference system can be generated by using the iMPACT tool. For BPI Up configuration mode, the μClinux MCS file is loaded at address `0x00000000`, incrementing in address space. To generate a new MCS file for BPI up configuration mode, follow these steps:

1. In XPS, compile a bitstream, `download.bit`, that includes the system configuration and the bootloader application.
2. Launch the iMPACT GUI and create a new project.
3. In the Welcome to iMPACT dialog box, select the *Prepare a PROM file* option, as shown in Figure 8. Click the Next button.



iMPACT - Welcome to iMPACT

Please select an action from the list below

○ Configure devices using Boundary-Scan (JTAG)

Automatically connect to a cable and identify Boundary-Scan chain

● Prepare a PROM File

○ Prepare a System ACE File

○ Prepare a Boundary-Scan File

SVF

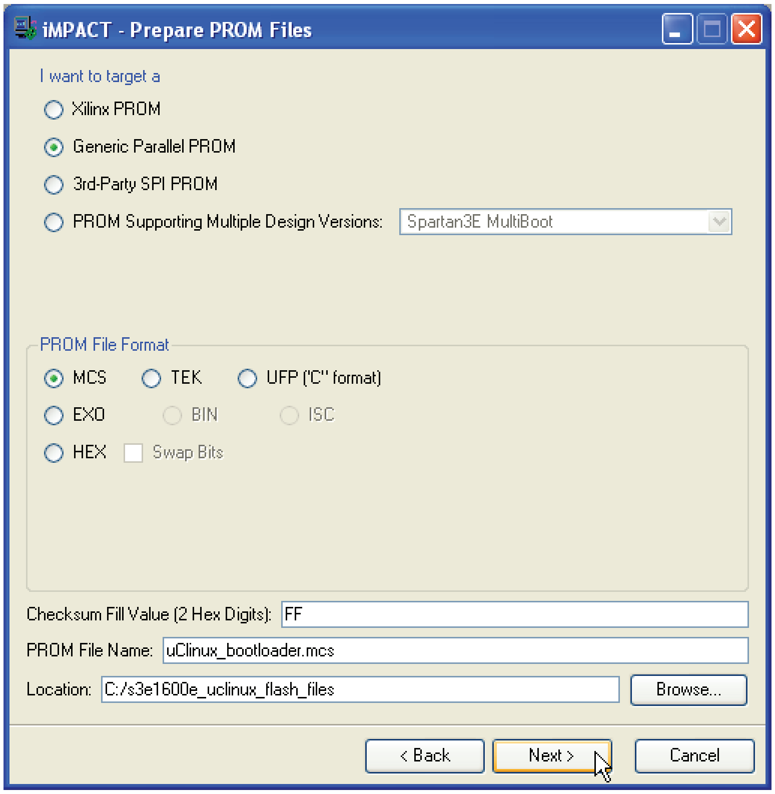
○ Configure devices

using Slave Serial mode

X963_08_101006

Figure 8: **Welcome to iMPACT Screen**

4. In the Prepare PROM Files dialog box, select the target to be a Generic Parallel PROM. Ensure that the file format is selected to be MCS. Enter an appropriate PROM file name and location. The dialog box is shown in Figure 9. Click Next.



I want to target a

○ Xilinx PROM
● Generic Parallel PROM
○ 3rd-Party SPI PROM
○ PROM Supporting Multiple Design Versions: Spartan3E MultiBoot

PROM File Format

● MCS ○ TEK ○ UFP ('C'' format)
○ EXO ○ BIN ○ ISC
○ HEX □ Swap Bits

Checksum Fill Value (2 Hex Digits): FF

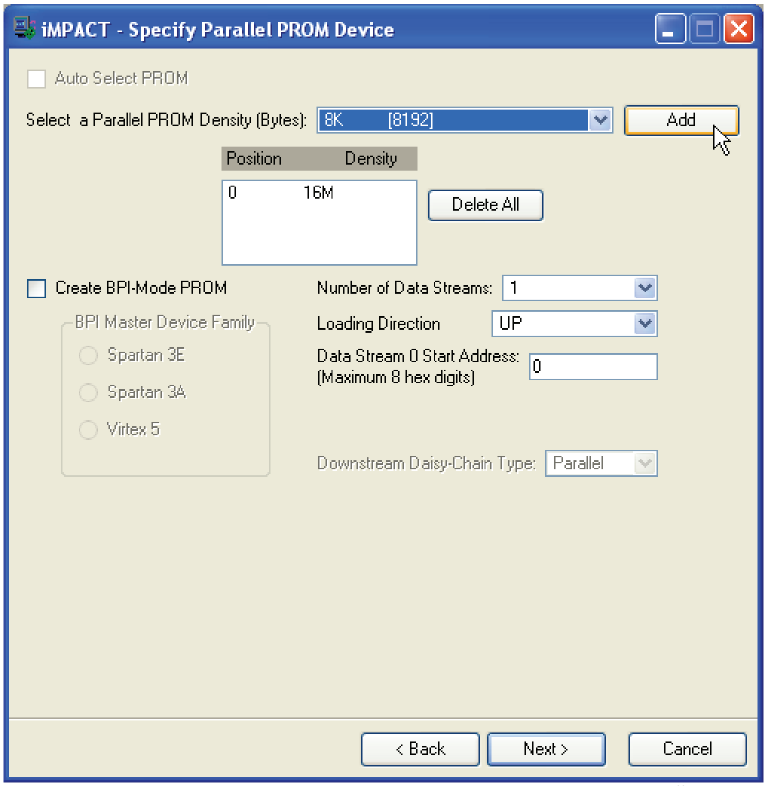
PROM File Name: uClinux_bootloader.mcs

Location: C:/s3e1600e_uclinux_flash_files

X963_09_110306

Figure 9: **Prepare PROM Files Dialog Box**

5. In the Specify Parallel PROM Device dialog box, shown in Figure 10, select a parallel PROM density of 16M and click Add. Ensure the Number of Data Streams is set to 1, the Loading Direction is set to UP, and the Data Stream 0 Start Address is set to 0. Click Next.


iMPACT - Specify Parallel PROM Device

□ Auto Select PROM

Select a Parallel PROM Density (Bytes): 8K [8192]

Position	Density
0	16M

□ Create BPI-Mode PROM

BPI Master Device Family

○ Spartan 3E

○ Spartan 3A

○ Virtex 5

Number of Data Streams: 1

Loading Direction: UP

Data Stream 0 Start Address: (Maximum 8 hex digits) 0

Downstream Daisy-Chain Type: Parallel

X963_10_110306

Figure 10: **Specify Parallel PROM Device Dialog Box**

6. In the File Generation Summary dialog box, click Finish.
7. The iMPACT screen will prompt for a device file to be added to Data Stream: 0. Select the `download.bit` file in the μClinux reference system project. When asked to add another design file to Data Stream: 0, click No. Click OK.
8. Select **Operations** → **Generate File...** in iMPACT to generate the MCS file.

Loading an MCS File into StrataFlash

The Xilinx PicoBlaze NOR Flash Programmer is used to load an MCS file into the StrataFlash. To download the files for the PicoBlaze NOR Flash Programmer for the SP3E1600E board, go to the following link:

- http://www.xilinx.com/sp3e1600

From that Website, click on the Design Examples link to find the PicoBlaze StrataFlash Programmer files. The PicoBlaze StrataFlash Programmer PDF file documents the steps necessary to load an MCS file into the StrataFlash memory. After following those steps, the MCS file will be loaded into the StrataFlash memory and the μClinux demonstration is ready to be run in BPI up mode. Proceed to the section, "Running the Demonstrations out of Flash", to run the demonstration.

Creating New Flash Files for the Web Server Demonstration

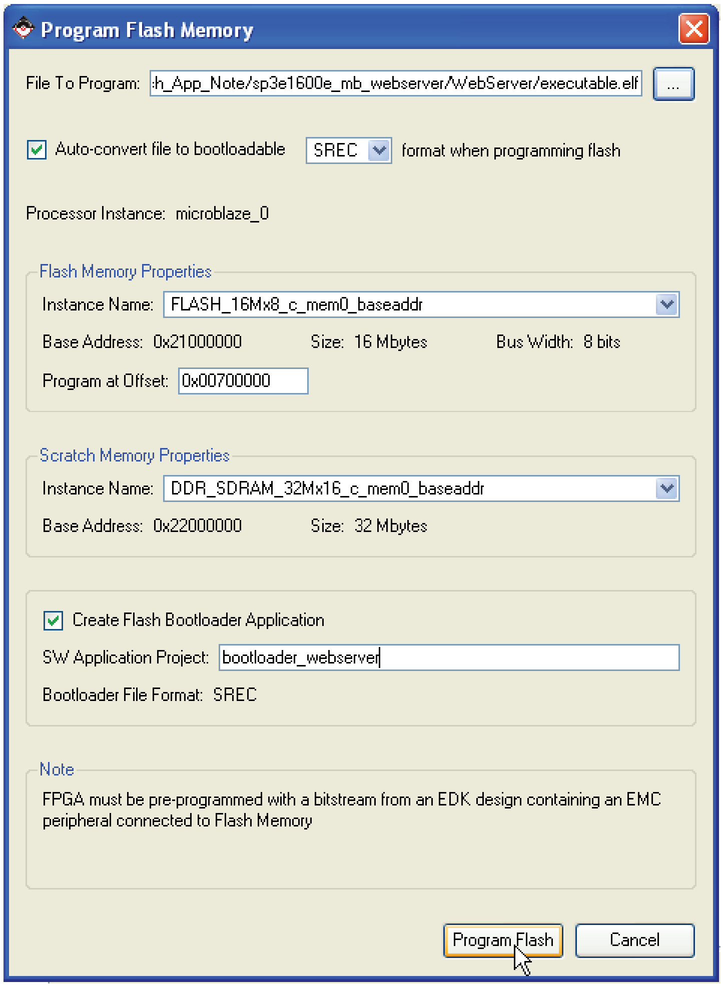
Instead of using the provided files, new files can be generated. The Web server reference system is pre-loaded into the StrataFlash on the SP3E1600E board from the factory to run in BPI Down configuration mode. For more information on the Web server reference system, refer to UG258, *Getting Started with the MicroBlaze Development Kit - Spartan-3E 1600E Edition.* This section describes how to create new flash files for BPI Down mode using the Web server reference system.

Creating New Bootloader Files and a New Web Server SREC File

Instead of using the existing bootloader files and Web server SREC file, new files can be created using the EDK tools. The bootloader files must be created because they will become part of the MCS file in the next section, "Generating an MCS File for BPI Down Configuration Mode". After the bootloader files are created in the EDK, they will have to be modified for the Web server demonstration to operate properly. To create new bootloader files and to load a new Web server SREC file into the StrataFlash, follow these steps:

1. Open the reference system project in XPS.
2. Connect the USB programming cable, the serial cable, and the power supply to the SP3E1600E development board. Power on the board.
3. 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.
4. Compile the Web server software application by selecting **Software** → **Build All User Applications** in XPS.
5. In XPS, select **Device Configuration** → **Program Flash Memory.**

6. In the Program Flash Memory dialog box, choose the file to program to be `/WebServer/executable.elf` under the project root directory. Check the *Auto-convert file to bootloadable SREC format when programming flash* check box. Change the program offset to the desired value, allowing enough room for the MCS file, which will be programmed down from `0x00FFFFFF`. The provided example files use an offset of `0x00700000`. A bootloader is created by clicking the *Create Flash Bootloader Application* check box in the Program Flash Memory dialog box. The Program Flash Memory settings are shown in Figure 11.



Program Flash Memory

File To Program: :h_App_Note/sp3e1600e_mb_webserver/WebServer/executable.elf

☑ Auto-convert file to bootloadable SREC format when programming flash

Processor Instance: microblaze_0

Flash Memory Properties

Instance Name: FLASH_16Mx8_c_mem0_baseaddr

Base Address: 0x21000000 Size: 16 Mbytes Bus Width: 8 bits

Program at Offset: 0x00700000

Scratch Memory Properties

Instance Name: DDR_SDRAM_32Mx16_c_mem0_baseaddr

Base Address: 0x22000000 Size: 32 Mbytes

☑ Create Flash Bootloader Application

SW Application Project: bootloader_webserver

Bootloader File Format: SREC

Note

FPGA must be pre-programmed with a bitstream from an EDK design containing an EMC peripheral connected to Flash Memory

X963_11_101006

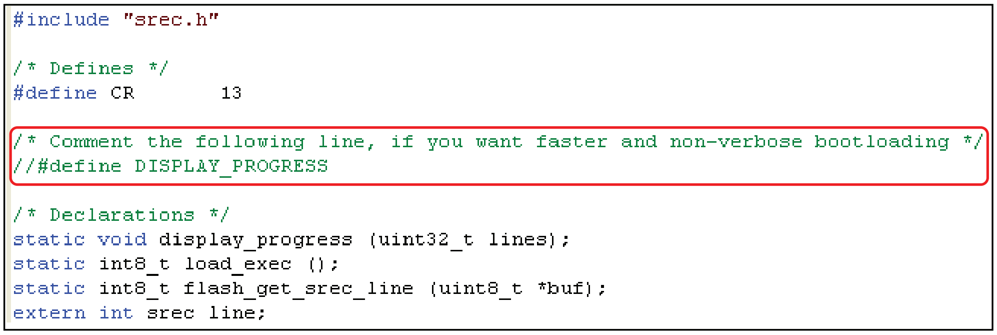
Figure 11: **Program Flash Memory Dialog Box for the Web Server ELF**

7. Click the Program Flash button. This will create a bootloader software application project and program the flash memory with the Web server software executable in SREC format.

8. 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 DISPLAY PROGRESS
   ```

   The line to comment out to allow non-verbose bootloading is shown in Figure 12. After modifying the C file, be sure to recompile the software application.



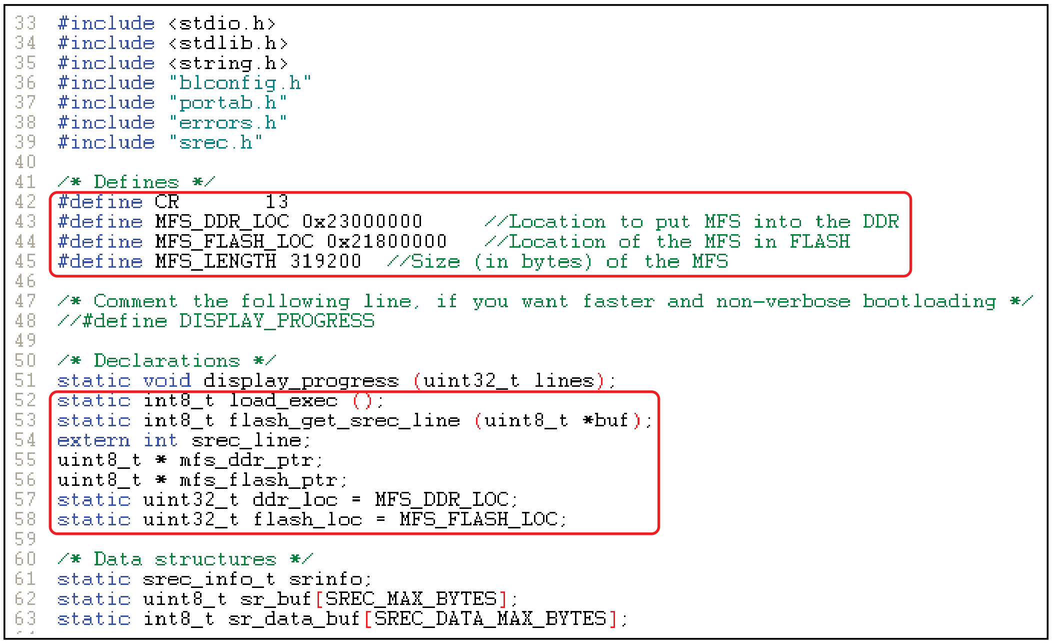
```
#include "srec.h"

/* Defines */
#define CR        13

/* Comment the following line, if you want faster and non-verbose bootloading */
//#define DISPLAY_PROGRESS

/* Declarations */
static void display_progress (uint32_t lines);
static int8_t load_exec ();
static int8_t flash_get_srec_line (uint8_t *buf);
extern int srec line;
```

X963_12_101006

Figure 12: **Code to Select Non-Verbose Bootloading**

9. The bootloader must be modified to copy the MFS image from flash into DDR for the Web server demonstration to fully operate. This is done by adding lines of code to the `bootloader.c` file that the EDK generates. To use the provided bootloader files which include the required modifications, refer to the section, "Adding the Bootloader Files to the Project". To modify the newly created bootloader, add the following pieces of code:

   a. Code is required to define the location in flash where the MFS image resides and the location in DDR to put the MFS image. This code is shown in Figure 13. The MFS_FLASH_LOC parameter should be set to the location in flash memory of where the MFS image will be placed.

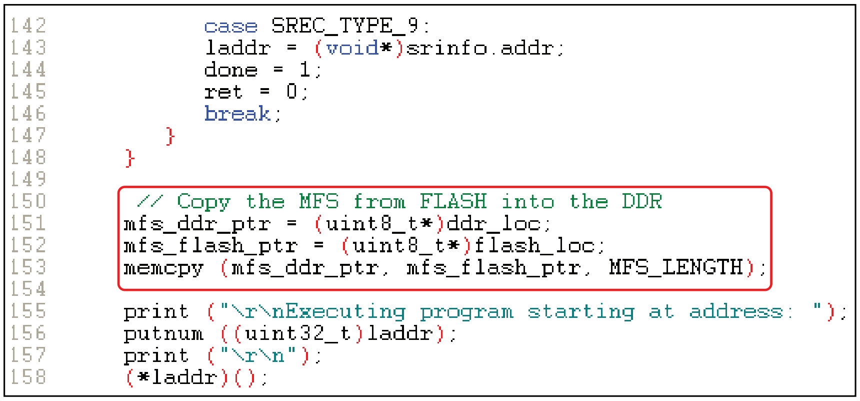


```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "blconfig.h"
#include "portab.h"
#include "errors.h"
#include "srec.h"

/* Defines */
#define CR        13
#define MFS_DDR_LOC 0x23000000     //Location to put MFS into the DDR
#define MFS_FLASH_LOC 0x21800000   //Location of the MFS in FLASH
#define MFS_LENGTH 319200  //Size (in bytes) of the MFS

/* Comment the following line, if you want faster and non-verbose bootloading */
//#define DISPLAY_PROGRESS

/* Declarations */
static void display_progress (uint32_t lines);
static int8_t load_exec ();
static int8_t flash_get_srec_line (uint8_t *buf);
extern int srec_line;
uint8_t * mfs_ddr_ptr;
uint8_t * mfs_flash_ptr;
static uint32_t ddr_loc = MFS_DDR_LOC;
static uint32_t flash_loc = MFS_FLASH_LOC;

/* Data structures */
static srec_info_t srinfo;
static uint8_t sr_buf[SREC_MAX_BYTES];
static int8_t sr_data_buf[SREC_DATA_MAX_BYTES];
```

X963_13_101006

Figure 13: **Web Server Bootloader Code Definitions and Declarations**

b. Code is also required for the bootloader to copy the MFS image from flash into DDR when the bootloader runs. This code is shown in Figure 14. After modifying the C file, be sure to recompile the software application.



```
      case SREC_TYPE_9:
      laddr = (void*)srinfo.addr;
      done = 1;
      ret = 0;
      break;
   }
}

 // Copy the MFS from FLASH into the DDR
mfs_ddr_ptr = (uint8_t*)ddr_loc;
mfs_flash_ptr = (uint8_t*)flash_loc;
memcpy (mfs_ddr_ptr, mfs_flash_ptr, MFS_LENGTH);

print ("\r\nExecuting program starting at address: ");
putnum ((uint32_t)laddr);
print ("\r\n");
(*laddr)();
```

X963_14_101006

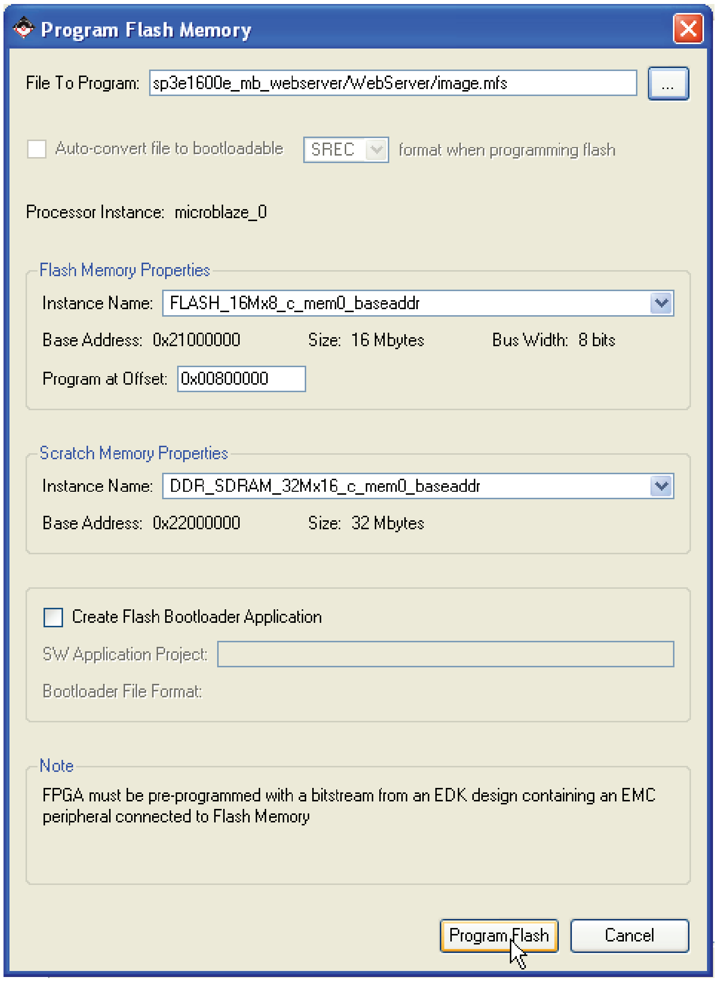
Figure 14: **Web Server Bootloader Code to Copy the MFS Image**

Loading the Web Server MFS Image into Flash

To run the Web server demonstration properly, the MFS image must be loaded into the StrataFlash. An MFS image file is included with the released Web server reference system in the `ready_for_download` and `WebServer` directories under the project root directory. This file can be used or a new MFS image can be generated. For instructions on how to create a new MFS image, refer to UG258, included with the SP3E1600E Development Kit. To load the MFS image into the StrataFlash memory, follow these steps:

1. Ensure that the target board is programmed with the hardware bitstream.
2. In XPS, select **Device Configuration** → **Program Flash Memory.**

3. In the Program Flash Memory dialog box, choose the file to program to be the MFS image file. Change the program offset to the desired value, making sure not to take up space used by the SREC file, and allowing enough room for the MCS file, which will be programmed down from `0x00FFFFFF`. The program offset must also match the address specified to the bootloader in the MFS_FLASH_LOC parameter. The provided example files use an offset of `0x00800000`. This offset is the address the provided Web server bootloader is programmed to find the MFS image. The Program Flash Memory settings for the MFS image are shown in Figure 15.


Program Flash Memory

File To Program: sp3e1600e_mb_webserver/WebServer/image.mfs

□ Auto-convert file to bootloadable SREC format when programming flash

Processor Instance: microblaze_0

Flash Memory Properties

Instance Name: FLASH_16Mx8_c_mem0_baseaddr

Base Address: 0x21000000 Size: 16 Mbytes Bus Width: 8 bits

Program at Offset: 0x00800000

Scratch Memory Properties

Instance Name: DDR_SDRAM_32Mx16_c_mem0_baseaddr

Base Address: 0x22000000 Size: 32 Mbytes

□ Create Flash Bootloader Application

SW Application Project:

Bootloader File Format:

Note

FPGA must be pre-programmed with a bitstream from an EDK design containing an EMC peripheral connected to Flash Memory

X963_15_101006

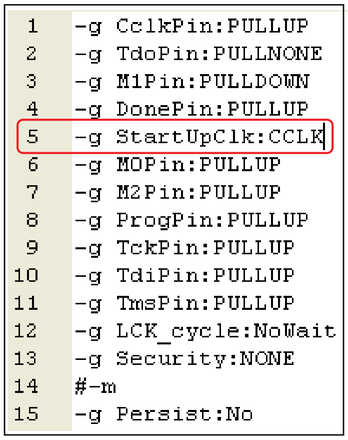
Figure 15: **Program Flash Memory Dialog Box for the Web Server MFS Image**

4. Click the Program Flash button.

Generating an MCS File for BPI Down Configuration Mode

An MCS file is used to program the StrataFlash with the system bitstream and bootloader loaded in the bitstream. An MCS file for the Web server reference system can be generated by using the PROMGen command. For BPI Down configuration mode, the Web server MCS file is loaded at address `0x00FFFFFF`, decrementing in address space. To generate a new MCS file for BPI Down configuration mode, follow these steps:

1. In the XPS project, change the bitgen options file, `/etc/bitgen.ut`, so that the startup clock is set to CCLK instead of JTAGCLK. The changed file will look like Figure 16.



```
-g CclkPin:PULLUP
-g TdoPin:PULLNONE
-g M1Pin:PULLDOWN
-g DonePin:PULLUP
-g StartUpClk:CCLK
-g M0Pin:PULLUP
-g M2Pin:PULLUP
-g ProgPin:PULLUP
-g TckPin:PULLUP
-g TdiPin:PULLUP
-g TmsPin:PULLUP
-g LCK_cycle:NoWait
-g Security:NONE
#-m
-g Persist:No
```

X963_16_101006

Figure 16: **Bitgen Options File with CCLK**

2. In XPS, compile a bitstream, `download.bit`, that includes the system configuration and the bootloader application.
3. Select **Project** → **Launch EDK Shell...** in XPS.
4. In the EDK shell, run the following command from the project root directory to generate an MCS file:

   ```
   PROMGen -p mcs -d FFFFFF implementation/download.bit -s 16384 -o
   webserver_bootloader.mcs
   ```

   In the above PROMGen command, these are the options used:

   - `-p mcs` sets the PROM format to MCS
   - `-d FFFFFF` sets the loading direction to down, starting at an address of 0x00FFFFFF
   - `implementation/download.bit` is the input bitstream file
   - `-s 16384` sets the PROM size to be 16384 kilobytes
   - `-o webserver_bootloader.mcs` sets the name of the output file

   Note: If an MCS file already exists, use the -w option to overwrite it.

Loading an MCS File into StrataFlash

The Xilinx PicoBlaze NOR Flash Programmer is used to load an MCS file into the StrataFlash. To download the files for the PicoBlaze NOR Flash Programmer for the SP3E1600E board, go to the following link:

- http://www.xilinx.com/sp3e1600

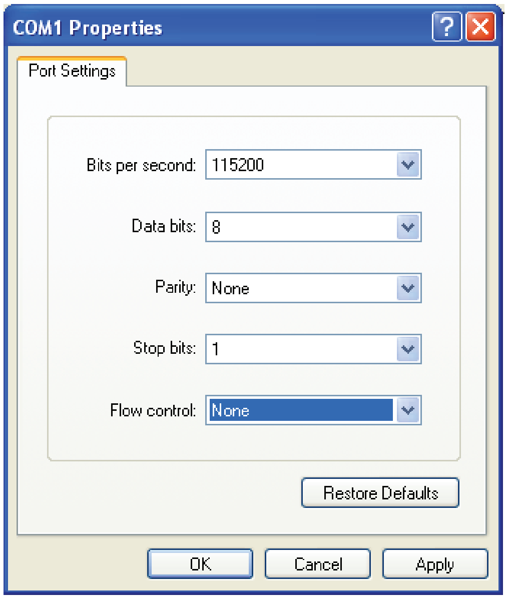
From that Website, click on the Reference Systems link to find the PicoBlaze StrataFlash Programmer files. The PicoBlaze StrataFlash Programmer PDF file documents the steps necessary to load an MCS file into the StrataFlash memory. After following those steps, the MCS file will be loaded into the StrataFlash memory and the Web server demonstration is ready to be run in BPI down mode. Proceed to the section, “Running the Demonstrations out of Flash”, to run the demonstration.

Running the Demonstrations out of Flash

Once all the files are loaded into the StrataFlash memory as detailed in the previous sections, the demonstrations can be run out of flash. The following sections detail how to run the μClinux demonstration and the Web server demonstration out of the StrataFlash.

Preparing to Run the Demonstrations out of Flash

To run the demonstrations, ensure that the SP3E1600E development board is set up properly with the serial cable and power supply connected. A serial terminal program, such as HyperTerminal must be set up to view the output of the demonstrations. The serial terminal program should be set to Baud Rate of **115200**, Data Bits to **8**, Parity to **None**, and Flow Control to **None**, as shown in Figure 17.



Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

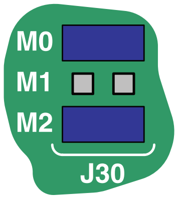
Flow control: None

X963_17_101006

Figure 17: **HyperTerminal Settings for the Demonstrations**

Running the μClinux Demonstration in BPI Up Mode

To run the files out of the StrataFlash in BPI Up mode, set the J30 configuration jumpers as shown in Figure 18. Once the configuration jumpers are set properly, power cycle the development board or press the PROG button to start the μClinux demonstration.



X963_18_101006

Figure 18: **Configuration Jumper Settings for BPI Up Mode**

Running the Web Server Demonstration in BPI Down Mode

To run the files out of the StrataFlash in BPI Down mode, set the J30 configuration jumpers as shown in Figure 19. Once the configuration jumpers are set properly, power cycle the development board or press the PROG button to start the Web server demonstration.

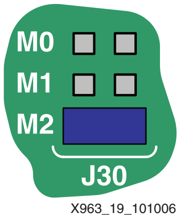


Figure 19: **Configuration Jumper Settings for BPI Down Mode**

References

UG257, *MicroBlaze Development Kit Spartan-3E 1600E Edition User Guide*

UG258, *Getting Started with the MicroBlaze Development Kit - Spartan-3E 1600E Edition*

Conclusion

The μClinux and Web server reference systems are provided with the MicroBlaze Development Kit - Spartan-3E Edition. The SP3E1600E development board provided with the kit has the μClinux demonstration pre-loaded to run in BPI Up configuration mode and the Web server demonstration to run in BPI Down configuration mode. This application note describes how to use the files provided and create new files for programming the Intel StrataFlash PROM with the μClinux and Web server demonstrations in the same configuration.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
11/15/06	1.0	Initial Xilinx release.