

PowerPC Linux

- Using Git
- Open Source Linux (OSL)
- QEMU System Model
- Archive
 - Using Git
 - Open Source Linux (OSL)
 - OSL Main Page
 - PowerPC Linux
 - PowerPC Linux On ML510
 - MicroBlaze Linux
 - MicroBlaze Linux - Little Endian
 - Kernel Debug
 - Device Drivers
 - Development Branch
 - FPU on PowerPC 440
 - Expanding A File System
 - Device Tree Generator

Site Utilities

- Home
- What is a Wiki Site?
- How to edit pages?
- How to join this site?
- Recent changes
- List all pages
- Page Tags
- Site members
- Site Manager

Add a new page

New Page

Fold

Table of Contents

Introduction

Tested Hardware and Reference Systems

- Xilinx ML507 Board Reference System
- Xilinx ML405 Board Reference System

Driver Status

Infrastructure

- Device Tree
- PowerPC GNU Tools
- Setting Up the Build Environment
- Prebuilt Ramdisk Image

The Linux Kernel For PowerPC

- Getting the Kernel From Xilinx
- Configuring The Kernel
- Building the Linux Kernel
 - Creating Custom Kernel Images With Device Trees
 - More Build Details
- Loading and Running the Kernel on the Board

Console UART Selection

Debugging Kernel Boot Problems

Introduction

The goal of this page is to help users understand how to get the PowerPC Linux kernel source code, build it and run it on a Xilinx board. This site assumes that those testing the Linux kernel on Xilinx PowerPC are knowledgeable of Embedded Linux. It is not intended to be tutorial on Linux.

Tested Hardware and Reference Systems

The primary test platforms are the ML507 (Virtex5) and the ML405 (Virtex4) from Xilinx. The kernel has been ran on other boards but is not currently being tested on them.

Xilinx ML507 Board Reference System

A device tree, virtex440-m1507.dts, is provided in the kernel tree at arch/powerpc/boot/dts, which describes a specific reference design for this board. This reference design is the easiest way to get Linux running on the ML507.

The reference design (bit stream) is included in the Virtex5 FXT Development Kit from Xilinx at the following url: <https://secure.xilinx.com/webreg/clickthrough.do?cid=111799>. Users are required to have or create a Xilinx user account, but the system is available for free. This reference design is not available on this wiki.

Note: The bit stream that is included on the ML507 by default will not work with the ML507 device tree file in the kernel tree. It will boot to some point and then hang.

Xilinx ML405 Board Reference System

A device tree, virtex405-m1405.dts, is provided in the kernel tree at arch/powerpc/boot/dts, which describes a specific reference design for this board. This reference design is the easiest way to get Linux running on the ML405. The bit stream (not a complete EDK system) can be downloaded at the following location.

[download link for ML405 PowerPC bit stream](#)

Base System Builder in the Xilinx EDK can be used to create complete systems which will run Linux on the ML405.

Driver Status

See the device driver page that is common for the PowerPC and Microblaze Linux kernel at [OSL Device Drivers](#).

Infrastructure

Device Tree

The PowerPC kernel uses device tree to describe the hardware system such that the kernel can configure itself during boot. The device tree generator can be pulled from the Xilinx GIT repository at [git://git.xilinx.com/device-tree.git](http://git.xilinx.com/device-tree.git). See the following page for more information about the process, <http://xilinx.wikidot.com/device-tree-generator>.

You may need to edit the device tree file by hand. A nice overview of the device tree file format may be found at http://devicetree.org/Device_Tree_Usage. This includes a good description of the PCI node.

PowerPC GNU Tools

See the following page, [PowerPC GNU Tools](#), for information about the GNU Tools for PowerPC to build the Linux kernel and applications.

Setting Up the Build Environment

Set environment variable CROSS_COMPILE to point to the correct cross compiler (consider the DENX ELDK). For the ELDK 4.1, the value of CROSS_COMPILE should be "ppc_4xx-".

Update the path to allow access to the binaries for the tools you are using. For the ELDK, the bin and usr/bin directories of the ELDK need to be in your path.

Prebuilt Ramdisk Image

A root file system is required for Linux to boot and the easiest way to accomplish that is to use a ramdisk. A prebuilt ramdisk is provided to help get a system up and running easily. The ramdisk image, `ramdisk.image.gz`, must be saved into the kernel directory `arch/powerpc/boot` directory. A prebuilt PowerPC ramdisk image can be downloaded at the following location.

[download link for ramdisk image for PowerPC Linux](#)

The PowerPC Linux kernel is using the `initrd` ramdisk image rather than an `initramfs` image as used on MicroBlaze. The kernel command line does require `"root=/dev/ram"` for the ramdisk to be used as the root file system.

A kernel image which includes the ramdisk image is named `simpleImage.initrd.virtex*`. The kernel build process takes the kernel image and a ramdisk image and combines them into a single kernel image that can be downloaded to memory using XMD.

The Linux Kernel For PowerPC

Getting the Kernel From Xilinx

The following command gets the kernel tree from the Xilinx Git server.

```
bash> git clone git://git.xilinx.com/linux-2.6-xlnx.git
```

Configuring The Kernel

The easiest way to get a kernel configuration that's close is to use the provided default configuration for the boards provided by Xilinx.

Configure the kernel for virtex4 on the ML405 using the following command.

Note: This assumes you are in the `/opt/SoftDev/Xilinx/linux-2.6.31` directory (or whatever version of Linux you received using git). Also assumed is the setup of the ppc tool chain, see PowerPC GNU Tools.

```
make ARCH=powerpc 40x/virtex4_defconfig
```

Configure the kernel for virtex5 on the ML507 using the following command.

```
make ARCH=powerpc 44x/virtex5_defconfig
```

Building the Linux Kernel

The following commands build the kernel and create an elf file, `simpleImage*virtex405-ml405.elf`, or `simpleImage*virtex440-ml507.elf`, in the `arch/powerpc/boot` directory. If you do not download the ramdisk image referenced above, the `"make ARCH=powerpc simpleImage.initrd.virtex..."` command will fail.

Build the ML405 kernel using the following command.

```
make ARCH=powerpc simpleImage.virtex405-ml405
make ARCH=powerpc simpleImage.initrd.virtex405-ml405
```

Build the ML507 kernel using the following command.

```
make ARCH=powerpc simpleImage.virtex440-ml507
make ARCH=powerpc simpleImage.initrd.virtex440-ml507
```

A kernel image created by the build target `simpleImage*` includes a device tree image. A kernel image created by the build target `simpleImage.initrd*` includes a device tree image and a ramdisk image. These kernel images allow a self contained kernel to be downloaded on a board using a Xilinx probe without any kind of boot loader like u-boot.

Creating Custom Kernel Images With Device Trees

Device tree files are located in the `arch/powerpc/boot/dts` directory. All Xilinx device tree files that used to build a device tree image into the kernel image must be named `virtex440-<your custom name>.dts` for the PowerPC 440 or `virtex405-<your custom name>.dts` for the PowerPC 405.

The kernel build target (in the make command line) for the respective device tree is `simpleImage.virtex440-<your custom name>` or `simpleImage.virtex405-<your custom name>`.

This is a great feature of the build system as multiple kernel images can be built and remain in the `arch/powerpc/boot` directory. Testing with multiple hardware configurations, each represented by a different device tree file, can be easily accomplished.

More Build Details

These details are not needed typically, but may be helpful to better understand the PowerPC Linux kernel build.

There is a kernel configuration option called Extra Targets and it is setup in the default kernel configuration files for the ML405 and the ML507. It is set to build target `simpleImage.virtex405-ml405` and `simpleImage.initrd.virtex405-ml405` for the ML405 and `simpleImage.virtex440-ml507.elf` and `simpleImage.initrd.virtex440-ml507.elf` for the ML507.

The purpose of this was to not break the old kernel build command line (using a `zImage`) that existed before the `simpleImage` build target was added. The following command will cause the default images noted above to be built.

```
make ARCH=powerpc zImage
```

Loading and Running the Kernel on the Board

Start XMD from the Linux kernel root directory.

On the ML405 board, use the following commands.

```
XMD% connect ppc hw -debugdevice deviceNr 3 cpunr 1
XMD% dow arch/powerpc/boot/simpleImage.virtex405-ml405.elf
XMD% run
```

On the ML507 board, use the following commands.

```
XMD% connect ppc hw -debugdevice deviceNr 5 cpunr 1
XMD% dow arch/powerpc/boot/simpleImage.virtex440-ml507.elf
XMD% run
```

There should be kernel console output on the serial port at 9600 baud.

Console UART Selection

The 16550 UART is the default UART in the kernel configuration and requires 'console=ttyS0' in the kernel command line. To use the UART Lite, the driver and the UART Lite console must be enabled in the kernel configuration and 'console=ttyUL0' must be in the kernel command line. The kernel command line is taken from the device tree file by default.

Debugging Kernel Boot Problems

See the page [Debugging Kernel Boot Problems](#) for more details.

page revision: 1, last edited: 6 Apr 2012, 17:17 GMT (9 days ago)

[Edit](#) [Tags](#) [History](#) [Files](#) [Print](#) [Site tools](#) [+ Options](#)

Xilinx, Inc.

Engineering Innovation

Xilinx delivers programmable platforms to help design engineers make their vision a reality.








Company

- [About Us](#)
- [Investor Relations](#)
- [Xilinx in the News](#)
- [Xcell Journal](#)
- [Jobs at Xilinx](#)

Support

- [Documentation](#)
- [Downloads](#)
- [Licensing](#)
- [Training](#)
- [Solution Centers](#)

Connect

-  [Contact Us](#)
-  [Connect on Facebook](#)
-  [Follow us on Twitter](#)
-  [Connect on LinkedIn](#)
-  [Watch us on YouTube](#)
-  [Join our Community](#)
-  [Subscribe to Newsletter](#)