XAPP925 (v1.3) June 1, 2007

Reference System: Using the OPB EPC with the Cypress CY7C67300 USB Controller

Author: Sundararajan Ananthakrishnan

Summary

The application note demonstrates the use of the On-Chip Peripheral Bus (OPB) External Peripheral Controller (EPC) to support the Cypress CY7C67300 USB controller in a PowerPC™ 405 processor based reference system.

This application note describes the:

- OPB EPC parameter settings to control the Cypress CY7C67300 USB controller
- Interrupt control logic to handle the interrupt signal of the Cypress CY7C67300 USB controller
- USB HPI keyboard standalone software application to test the Cypress CY7C67300 USB controller
- Porting of the Linux Driver for the Cypress CY7C67300 USB controller in the OPB EPC reference system

The OPB EPC reference system is targeted for the Xilinx Virtex™-4 ML403 Evaluation Platform.

Included Systems

The reference system for the Xilinx Virtex-4 ML403 Evaluation Platform is included with this application note. The reference system is available at:

www.xilinx.com/bvdocs/appnotes/xapp925.zip

Introduction

The OPB EPC is used to control peripherals that are connected externally to Xilinx FPGAs. The most commonly used external devices are LAN controllers, USB controllers, and IEEE 1394 (FireWire) controllers.

The OPB EPC is an OPB slave only device. It does not support any DMA operations from the external devices. The OPB EPC supports both multiplexed and non-multiplexed address and data buses where the data bus width can be 8, 16, or 32 bits wide.

The standalone software application provided with this reference system is executed from the cacheable region of the external DDR memory.

The Cypress CY7C67300 USB controller is interfaced to the OPB EPC through the Host Peripheral Interface (HPI). Detailed information of the HPI interface is found in Chapter 5, *HPI Transport Module*, of the OTG-Host BIOS User's Manual which is available on the Cypress website.

Hardware and Software Requirements

The hardware and software requirements are:

- Xilinx Virtex-4 ML403 Evaluation Platform
- Xilinx Platform USB cable or Parallel IV programming cable
- RS232 serial cable and serial communication utility (HyperTerminal)
- Xilinx Platform Studio 9.1.01i
- Xilinx Integrated Software Environment (ISE™) 9.1.03i

Reference System Specifics

The reference system has the PowerPC 405 processor with the caches enabled to use the instruction cache (I-cache) and the data cache (D-cache) from the external PLB DDR memory. The PLB2OPB Bridge is used to connect the OPB EPC to the PowerPC 405 processor. The OPB UART Lite core with interrupts, the OPB Interrupt Controller (OPB INTC) and OPB GPIO cores are also used in the reference system.

The reference system is shown Figure 1 and the address map of the reference system is shown in Table 1.

Block Diagram

The IP cores in the reference system are shown in Figure 1.

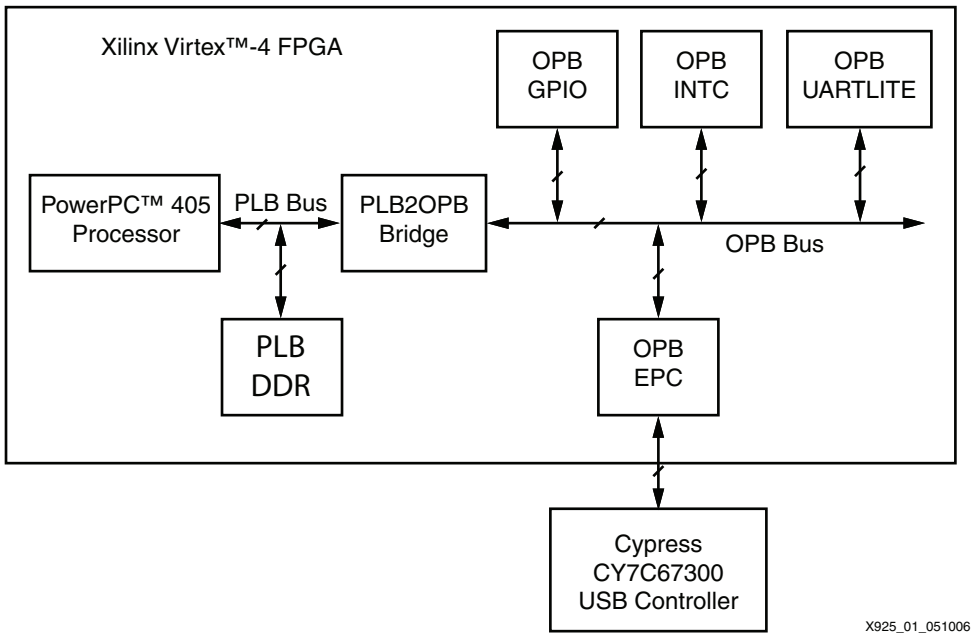

Figure 1: **Reference System Block Diagram**

Address Map

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

Table 1: **Reference System Address Map**

Peripheral	Instance	Base Address	High Address
plb_ddr	plb_ddr_0	0x00000000	0x03FFFFFF
opb_epc	opb_epc_0	0xA5000000	0xA500FFFF
opb_gpio	opb_gpio_0	0x90000000	0x900001FF
opb_uart16550	opb_uart16550_0	0xA0000000	0xA0001FFF
opb_intc	opb_intc_0	0xD1000FC0	0xD1000FDF

System Configuration

This Xilinx Virtex-4 ML403 Evaluation platform based reference system has the PowerPC™ 405 processor with the caches enabled. The PLB2OPB bridge connects the OPB EPC to the PowerPC 405 processor in the reference system. The PLB DDR SDRAM is used as the external memory and is configured to allow cacheline transactions from the PowerPC 405 processor in the reference system.

The reset signal of the Cypress CY7C67300 USB controller is connected to the OPB GPIO core. The reset signal is controlled by the 13th bit of the *ML40x Control Register 2* of the Xilinx Virtex-4 ML403 Evaluation platform. Setting this bit to 1 resets the CY7C67300 USB controller and setting this bit to 0 enables the normal operation of the CY7C67300 USB controller.

Note: For further details about the ML40x control registers, see the *ML40x EDK Processor Reference Design UserGuide*, (ug082.pdf).

The OPB INTC core handles the interrupt signal from the Cypress CY7C67300 USB controller.

The following section describes the various OPB EPC configuration settings and the interrupt control logic that handles the interrupt signal of the Cypress CY7C67300 USB controller.

Setting the OPB EPC Common Parameters

The OPB EPC can support up to a maximum of four peripherals. In the reference system, the Cypress CY7C67300 USB controller is the only external peripheral connected to the OPB EPC, therefore, the parameter *Number of External Peripherals,* is set to 1.

The address bus of the Cypress CY7C67300 USB controller connected to the OPB EPC system is four bits wide, therefore, the parameter *Maximum Address Bus width of all External peripheral*, is set in the OPB EPC to 4.

Because the Cypress CY7C67300 USB controller has a 16-bit data width, the parameter *Maximum Data Bus width of all External peripheral*, is set to 16. The Cypress CY7C67300 USB controller is the only external peripheral that is connected in this reference system, therefore the parameter, *Maximum Data Bus width and Address/Data-Multiplexed Address Bus width of all External Peripheral*, is also set to 16.

The OPB EPC does not support burst transactions from or to the external devices, therefore, parameter, *Enable Burst Support in External Peripherals*, is not enabled.

Figure 2 shows the parameter settings for the OPB EPC when the **Common** tab is selected.

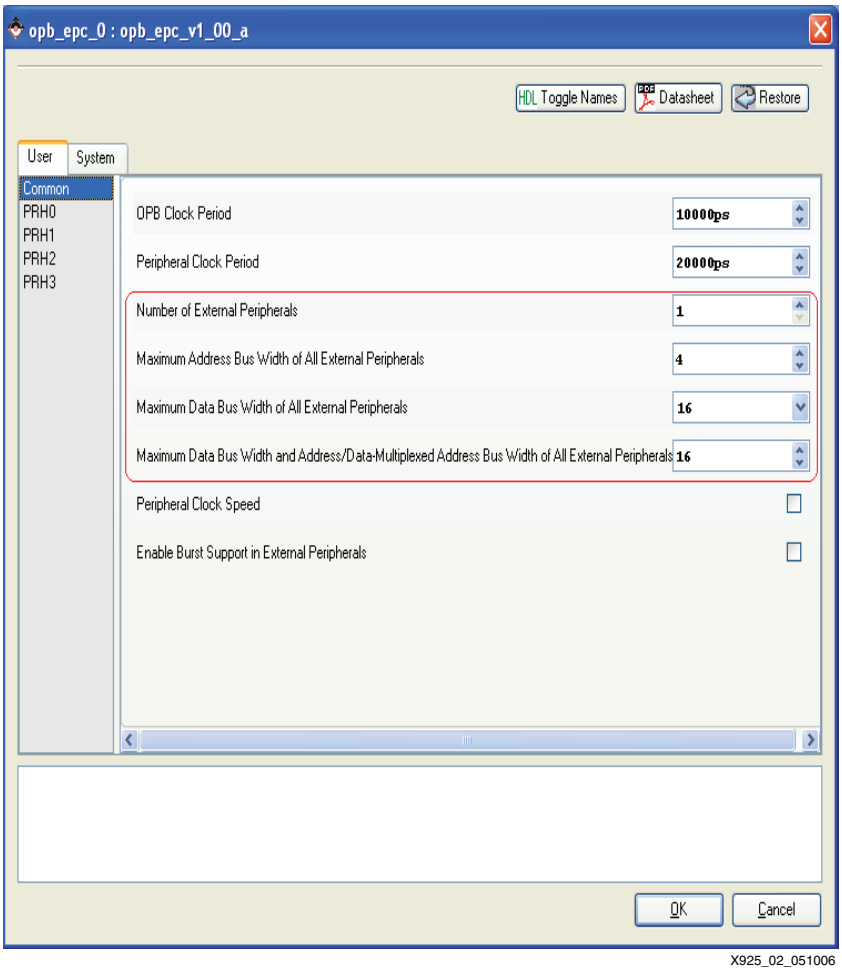

Figure 2: **Setting the OPB EPC Common Parameters**

Setting the OPB EPC Peripheral 0 (PRH0) Parameters

Because the Cypress CY7C67300 USB controller is the only external peripheral connected to the OPB EPC, the parameter settings are done only for the Peripheral 0 (PRH0).

The OPB EPC does not support access to the external FIFO within the Cypress CY7C67300 USB controller, therefore, the parameter, *Support FIFO Access in External Peripheral*, is not enabled, and the parameter, *FIFO Offset from Base Address*, is set to 0.

The Cypress CY7C67300 USB controller has the 4-bit address bus width and a 16-bit data bus width, respectively. Therefore, the parameter, *Address Bus width of External peripheral*, is set to 4 and the parameter, *Data Bus width of External peripheral*, is set to 16.

The data bus width of the external peripheral is 16-bits wide and the OPB bus is 32-bits wide. The parameter, *Support Multiple Cycle Access in peripheral to Match an OPB Cycle*, is enabled, thereby enabling the OPB EPC to run multiple cycles on the peripheral interface for a single OPB read/write cycle.

The OPB EPC can operate in both synchronous and asynchronous modes. In the reference system, the OPB EPC is configured to asynchronous mode of operation by not enabling the parameter, *Peripheral Access Mode*.

The OPB EPC core supports both the multiplexed and non-multiplexed address and data buses, where the data bus width can be 8, 16, or 32 bits. The Cypress CY7C67300 USB

controller has a separate address and data bus, and the parameter, *Peripheral Bus type*, is not enabled.

The timing parameters for the OPB EPC are set for the asynchronous mode of operation. The timing values are set by referring to the datasheet of the Cypress CY7C67300 USB controller.

Figure 3 shows the OPB EPC peripheral interface parameter settings for PRH0.

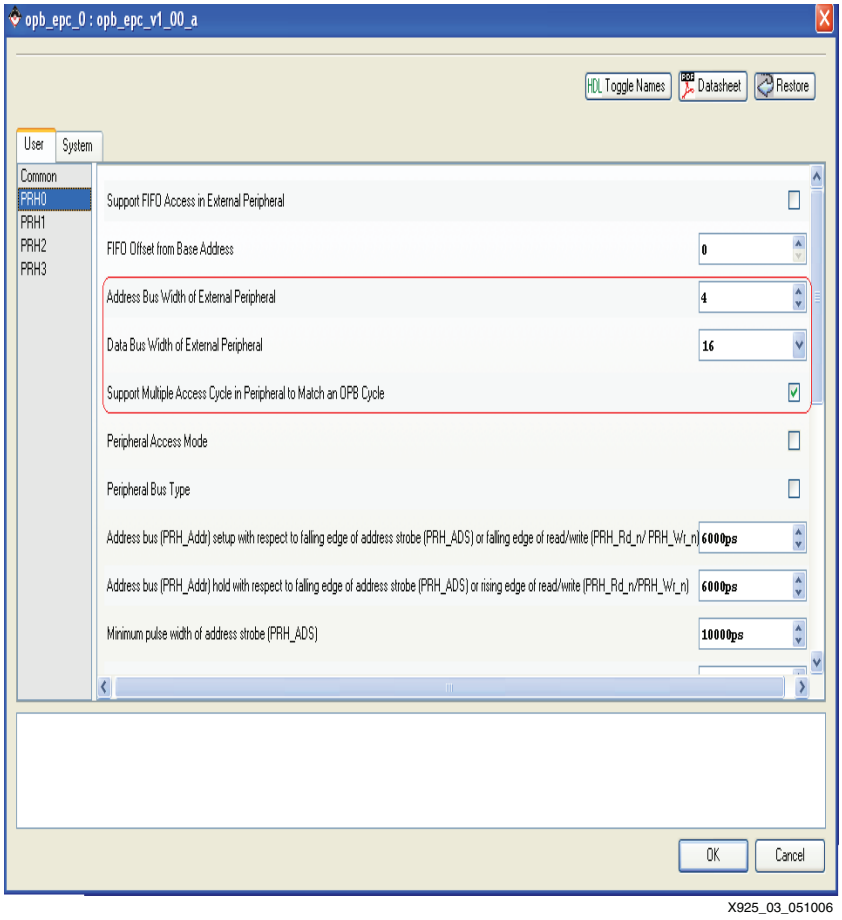

Figure 3: **Setting the OPB EPC Peripheral (PRH0) Parameters**

Interrupt Control Logic

The Cypress CY7C67300 USB controller generates an interrupt depending on several conditions. The OPB EPC does not support routing of the interrupt through the OPB EPC core, therefore, the interrupt signal is directly routed to the processor through the OPB INTC core in the system. The application software running on the PowerPC™ 405 processor will manage the interrupt handling.

Obtain the list of interrupts by expanding the opb_intc_0 tree node and clicking on the last port under the **Net** for **Intr**. This will bring up the **Interrupt Connection Dialog** box.

Add the **usb_hpi_int** interrupt input signal to Connected Interrupts of the OPB INTC as shown in Figure 4.

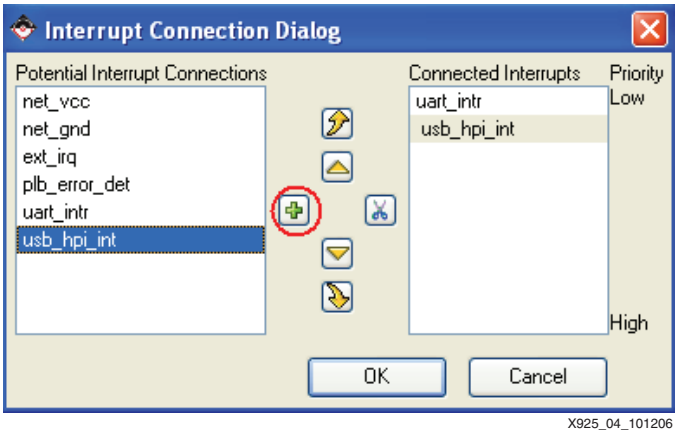

Figure 4: **Cypress USB Controller Interrupt Signal Connection**

The Software Application

This section describes the software application that is used for testing the Cypress CY7C67300 USB controller. The software application is found under the project root directory, `usb_hpi_test/src/usb_hpi_test.c`.

The software application is executed from the PLB DDR memory and should be marked to not load into BRAM. The linker script has all options set to the external memory.

The Cypress CY7C67300 USB controller is configured in the HPI mode. In this mode, the Cypress CY7C67300 USB controller must be initialized with the boot up sequence. Once the boot up sequence is over, the Cypress CY7C67300 USB controller is ready to operate in the normal HPI mode with the external host controller (OPB EPC).

The Cypress CY7C67300 USB controller is configured as a host and a USB keyboard (without a built-in hub) is connected to the USB host port on the Xilinx Virtex-4 ML403 Evaluation Platform. All the key presses on the USB keyboard are communicated using the mailbox registers through the OPB EPC to the PowerPC™ 405 processor on the Xilinx FPGA. The software then reads the data from mailbox registers into the local buffer and compares it with the internal set of upper and lower keyboard data. All the key presses are displayed on the UART HyperTerminal. The software application continuously polls the HPI registers to detect another key press event.

Executing the Reference System

There are two EDK project files provided with this reference system. The *system.xmp* is the EDK project file for the stand-alone application and *system_linux.xmp* is the EDK project file for the Linux application.

This section outlines the stand-alone application that is a part of the system.xmp project.

To execute the reference system, the bitstream needs to be generated and the software application compiled. The bitstream and the compiled software application for this system are available in ready_for_download directory under the project root directory. A HyperTerminal or similar program needs to be configured to use the COM port and the RS232 connector of the ML403 board needs to be connected to the COM port.

Set the HyperTerminal to Baud Rate of **9600**, Data Bits to **8**, Parity to **None,** and Flow Control to **None** as shown in Figure 5.

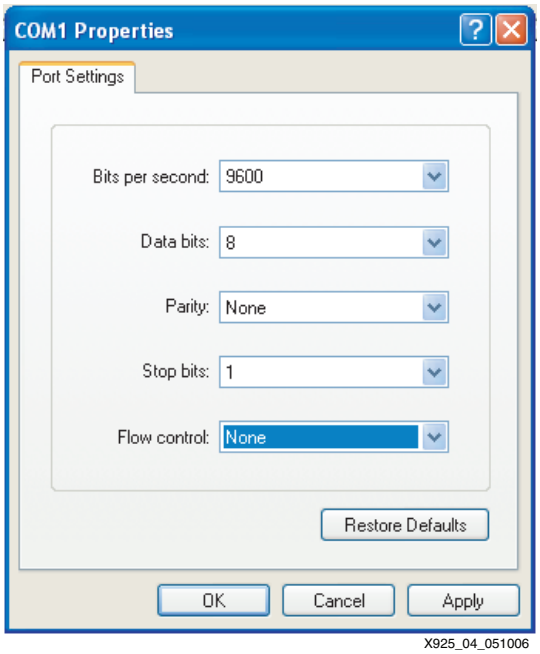

Figure 5: **HyperTerminal Settings**

Executing the Reference System using the Pre-Built Bitstream and the Compiled Software Applications

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

1. Change directories to the `ready_for_download` directory.
2. Use iMPACT to download the bitstream by using the following:

   ```
   impact -batch xapp925.cmd
   ```

3. Invoke XMD and connect to the PowerPC 405 processor by the following command:

   ```
   xmd -opt xapp925.opt
   ```

4. Download the executables by the following command:

   ```
   dow executable.elf
   ```

Executing the Reference System from EDK

To execute the system using EDK, follow these steps:

1. Open `system.xmp` inside EDK.
2. Use **Hardware→Generate Bitstream** to generate a bitstream for the system.
3. Use **Software→Build All User Applications** to build the software applications.
4. Download the bitstream to the board with **Device Configuration→Download Bitstream**.
5. Launch XMD with **Debug→Launch XMD...**
6. Download the executables by the following command:

   ```
   dow executable.elf
   ```

Running the Software Applications

To run the either of software applications, use the ***run*** command inside XMD. The status of the software application is displayed in the HyperTerminal data screen.

As the software application executes, the following messages can be seen in the HyperTerminal window:

```
ML40x USB Keyboard Demo
Loading executable file for USB controller chip.

Please connect a USB Keyboard to the ML40x.
Key presses will appear on the UART terminal.
Note: USB Keyboard must not have a built-in hub.
```

Once this message appears, then the USB Keyboard input appears in the HyperTerminal.

Linux USB Driver for the Cypress CY7C67300 USB Controller

This section outlines the steps to be taken when using the Linux USB Driver for the Cypress CY7C67300 USB controller in the Xilinx PowerPC™ 405 reference systems with the OPB EPC. The driver has been tested in host mode in MontaVista Linux 3.1 on a Xilinx Virtex-4 ML403 Evaluation Platform.

This driver has been ported to the Xilinx PowerPC™ 405 reference system from the existing driver provided by Cypress for a StrongArm platform. The Linux driver for a StrongArm platform is available as a part of the CY3663 EZ-OTG/EZ-Host development kit which is available on the Cypress website.

Note: The pre-compiled Linux image for this reference system is available under the Project root directory, `linux/zImage.initrd.elf`. This image can be downloaded directly to the reference system.

Integrating the Cypress USB driver into MontaVista Linux

1. Open the system_linux project in EDK. The system_linux project contains all the settings required for generating the Linux BSP for the reference system. This project directory will hereby be referred as <edk_proj>.
2. Clean the SW libraries with **Software→Clean Libraries.**
3. Generate the Linux BSP with **Software→Generate Libraries and BSPs.**

   The resulting Linux BSP is located in **<edk_proj/pc405_0/libsrc/linux_mvl31_v1_01_b/linux.**
4. Exit EDK.
5. Create a copy of the MontaVista Linux kernel for the ML300 board and place it in a directory of your choice. This directory will hereby be referred as <linux_dist>.

   The Linux kernel and the tools to build the Linux kernel are available from the MontaVista website.

   For further information about using MontaVista Linux with EDK, see *OS and Libraries Document Collection (oslibs_rm.pdf)*, which is available as a part of the EDK installation.
6. Apply the ML300 LSP patches which are available on the MontaVista website.
7. Patch the Linux kernel for use with the ML40x board:

   ```
   $ cd <edk_proj>/linux
   $ ./patch_linux <linux_dist>
   ```

   Completing step 7 copies the .config file from the linux directory to the Linux kernel, patches the Linux kernel with necessary changes for the ML40x board, and copies the Linux BSP into the Linux kernel. The .config file is specific to this OPB EPC reference system.

8. Patch the Linux USB Driver:

   Copy the USB driver patch (<edk_proj>/linux/cypressusbdriver.patch) to the MontaVista Linux kernel. Apply the patch by running the following command in the MontaVista Linux kernel:

   ```
   $ cd <linux_dist>
   $ patch -Np1 < cypressusbdriver.patch
   ```

9. Change the Linux Cypress Driver source code to match the reference system created.

   Changes to be made in the file <linux_dist>/drivers/usb/cy7c67300/cy7c67200_300_hcd.c for the board specific values are:

   - base_addr should be the base address defined for the OPB EPC controller. The base address is 0xA5000000 for this reference system.
   - data_reg_addr is the address of the next register and for the current HW connections, the address is base_addr + 0x4. The value is 0xA5000004 for this reference system
   - irq is the IRQ number and it should be equal to (31 – the vector ID) of the USB interrupt. This is 31 for this reference system.

     Changes to be made in the file <linux_dist>/drivers/usb/cy7c67300/cy7c67200_300_lcd.c are:

   Modify the address offset of the HPI registers based on the interface connections of the OPB EPC to the Cypress CY7C67300 USB controller.

   The address offset for the HPI registers is 0x4 in this reference system, therefore, the register definitions are multiplied by 4. This is accomplished by left shifting the following defines by 2.

   ```
   #define HPI_STAT_ADDR    (HPI_STAT_PORT << 2)
   #define HPI_MBX_ADDR     (HPI_MBX_PORT  << 2)
   #define HPI_DATA_ADDR    (HPI_DATA_PORT << 2)
   #define HPI_ADDR_ADDR    (HPI_ADDR_PORT << 2)
   ```

10. Comment the line in the file <linux_dist>/arch/ppc/boot/simple/embed_config.c:

    - Change the following line around line number 749:

      ```
      #error I2C needed for obtaining the Ethernet MAC address
      ```

      to

      ```
      /* #error I2C needed for obtaining the Ethernet MAC address */
      ```

    This change is made because the system is expecting the Ethernet MAC address to be read from the IIC device, but the IIC device is not a part of the reference system.

11. Copy the RAM Disk from the MontaVista installation to the kernel:

    ```
    $ cp /<path to the MontaVista Linux
    Installation>/opt/montavista/pro/devkit/ppc/405/images/
    ramdisk.gz /<linux_dist>/arch/ppc/boot/images/ramdisk.image.gz
    ```

12. Build the Linux Kernel:

    ```
    $ make oldconfig dep zImage.initrd
    ```

13. Download the Linux image <linux_dist>/arch/ppc/boot/images/zImage.initrd.elf and run the Linux image.
14. Login as root. There is no password.

Testing the USB driver

This driver has been tested in host mode with a USB Flash drive (USB 1.1 compatible) in MontaVista Linux 3.1 on Xilinx Virtex-4 ML403 Evaluation Platform and has NOT been tested for the Peripheral or Device mode.

- Connect the USB Flash Drive to the Xilinx Virtex-4 ML403 Evaluation Platform.
- Run the following commands in the HyperTerninal window to mount the device for a VFAT file system:

```
# mkdir /mnt/usbcf
# mount -t vfat /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/usbcf
```

- Once the device has been mounted the standard file access operations commands such as, `'ls', 'rm', 'cp'`, etc., can run on the device (/mnt/usbcf).
- After using the device. unmount the device needs by typing the following command in the HyperTerminal window:

```
# umount /mnt/usbcf
```

Conclusion

This application note describes how to set up the OPB EPC in a PowerPC 405 processor system. The reference system is built for the Xilinx Virtex-4 ML403 Evaluation Platform. The system includes a software application which is used for testing the Cypress CY7C67300 USB controller. The application note also illustrates on how to use the Linux Driver for the Cypress CY7C67300 USB controller in the OPB EPC reference system.

References

The current versions of the following documents are referred:

- DS325, *OPB External Peripheral Controller Product Specification*
- UG082, *ML40x EDK Processor Reference Design User Guide*
- *Cypress CY7C67300 EZ-Host Programmable Embedded USB Host/peripheral Controller Datasheet*
- *OTG-Host BIOS User's Manual*
- *OS and Libraries Document Collection*

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
5/23/06	1.0	Initial Xilinx release.
10/20/06	1.1	Updated for EDK 8.2.01i.
2/14/07	1.2	Updated for EDK 8.2.02i; corrected .../xapp925.zip Hyperlink on pg1.
6/1/07	1.3	Updated for EDK 9.1.01i.