Course Overview

XUP Embedded Linux on MicroBlaze Workshop

1. Objectives

Broadly, the objective of this workshop is to equip university academics with the resources, high level skills and confidence to introduce embedded Linux on Xilinx MicroBlaze to their teaching and research programs. Through the course of the workshop attendees will progress towards the following outcomes

- Basic understanding of the Linux operating system, and Embedded Linux concepts, including
  - Linux kernel architecture
  - device driver concepts and loadable modules
  - memory management and MMU vs noMMU Linux kernels
- Ability to use the Xilinx EDK and PetaLinux to achieve a working MicroBlaze Linux kernel bringup
- Ability to build, modify and debug Embedded Linux applications on the MicroBlaze platform
- Understanding of the hardware interfacing options for adding custom hardware to MicroBlaze/Linux systems
- Ability to develop a simple custom hardware peripheral for MicroBlaze
- Ability to develop a simple Linux device driver for a custom hardware peripheral
- Ability to integrate an open source IP core from the opencores.org website
- Understanding of some of the key research directions and activities surrounding operating systems and Embedded Linux on FPGA platforms
- Ideas for student project work based on the platform
- Links to helpful resources beyond the workshop materials

2. Prerequisites

- Basic FPGA + hardware design

- Conceptual understanding of FPGAs, programmable hardware and logic design.
    - Detailed knowledge of VHDL/Verilog and ISE tools not required
- Basic system-on-chip and reconfigurable computing concepts
    - Soft-core CPUs and IP cores
    - Xilinx EDK exposure helpful but not mandatory
- Basic Linux and embedded systems
    - Previous Linux or embedded Linux exposure helpful but not required

3. Course Agenda

- **Overview and intro**
    - Schedule
    - Objectives
    - Lab environment
- **Embedded Linux demystified**
    - Development
    - Kernel architectures and device driver model
    - Embedded development issues
    - Toolchains issues
- **LAB: Prebuilt boot/demo/explore**
    - Workstation login
    - Serial console open
    - Board power on and autoboot
    - Login, explore
    - Differences and similarities between embedded and desktop Linux
- **Introduction to PetaLinux**
    - What is it?
    - Why does it exist
    - Access and installation
        - Source tree layout
        - Basic usage
    - Kernel and system configuration
    - Other distributions
- **LAB: Build toolchain**
    - Explore the Linux menu configuration system
    - Build and boot your own MicroBlaze Linux kernel
- **Application development and debug**
    - Role of user applications
    - Cross-compilation and cross-debugging
        - gdb and gdbserver
- **LAB: Application development and debug**

- Create and modify user application
- Cross-building and installing application
- Running and debugging

- **TCP/IP networking**
  - Basic network capabilities
  - Network support to improve the development process
  - Remote system upgrade and management
  - Supported networking devices for MicroBlaze/Linux
- **LAB: Network File System (NFS)**
  - Use NFS to speed the development process
  - Enable and build an embedded web server
  - Create simple web-enabled applications
- **Device drivers and loadable modules**
  - Concepts of the Linux device driver model
  - The "platform device" concept
  - Loadable kernel modules
- **LAB: Device driver development**
  - Create a loadable module
  - Implement a device driver
  - Load a kernel module on MicroBlaze
- **MMU or not to MMU**
  - Basic purpose of hardware memory management
  - Role of MMU
  - MMU options for the MicroBlaze CPU
  - Factors in choosing to enable MMU
- **LAB: MMU vs Non-MMU**
  - Explore kernel configuration settings related to MMU
  - Memory and process protection
- **Introduction to Xilinx MicroBlaze and EDK**
  - Base System Builder
  - Major hardware pieces of MicroBlaze systems
  - MicroBlaze architectural features
  - Key bus infrastructure
- **LAB: BSB and Board Bringup**
  - Go through BSB to build a system from scratch
  - Configure the MicroBlaze for Embedded Linux
  - Build and boot a Linux kernel
- **Custom hardware development and interfacing**
  - System interfacing options
  - PLB and FSL bus architectures
  - Create/Import Peripheral wizard
  - IP Core simulation
- **LAB: Use of Create/Import Peripheral wizard**
  - The Create/Import Peripheral wizard for PLB and FSL cores
  - Create a simple custom peripheral

- Add the custom peripheral and rebuild hardware

- **Device driver for custom hardware**
  - Device driver options for FSL and PLB peripheral
  - Create a simple character device driver
- **LAB: Custom device driver**
  - Create a real PLB device driver
  - Communicating with the PetaLinux generic FSL driver
- **OpenCores and WishBone**
  - Basics of WishBone bus protocol
  - Main principles for adding WishBone cores
  - How to find cores on opencores.org
- **LAB: Integrate OpenCores IP**
  - Integrate an OpenCores IP core to a MicroBlaze system
- **Going further, and review**
  - How to use what you have learned
  - Review the workshop outcomes