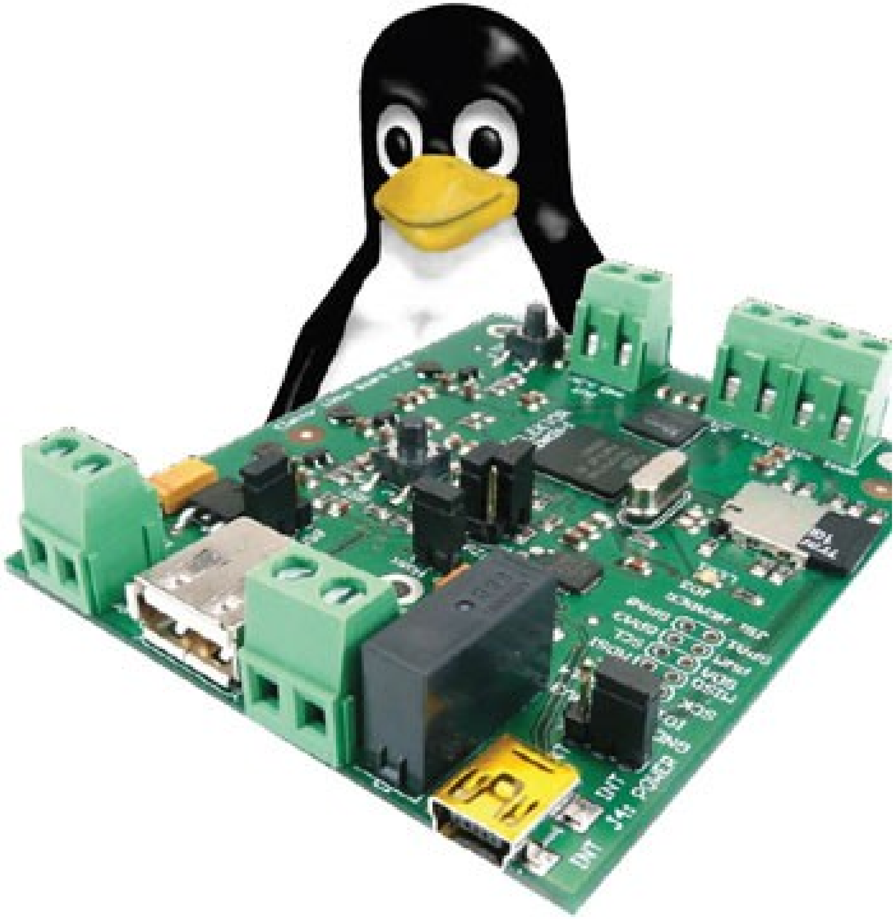
ZedBoard with Petalinux

James Bueghly
Jovana Andrejevic
Jia Fu Low

August 7, 2015

Petalinux Tools

- Xilinx embedded Linux solution
- Version 2015.2
- Develop and package projects from command line
- No need for SDK

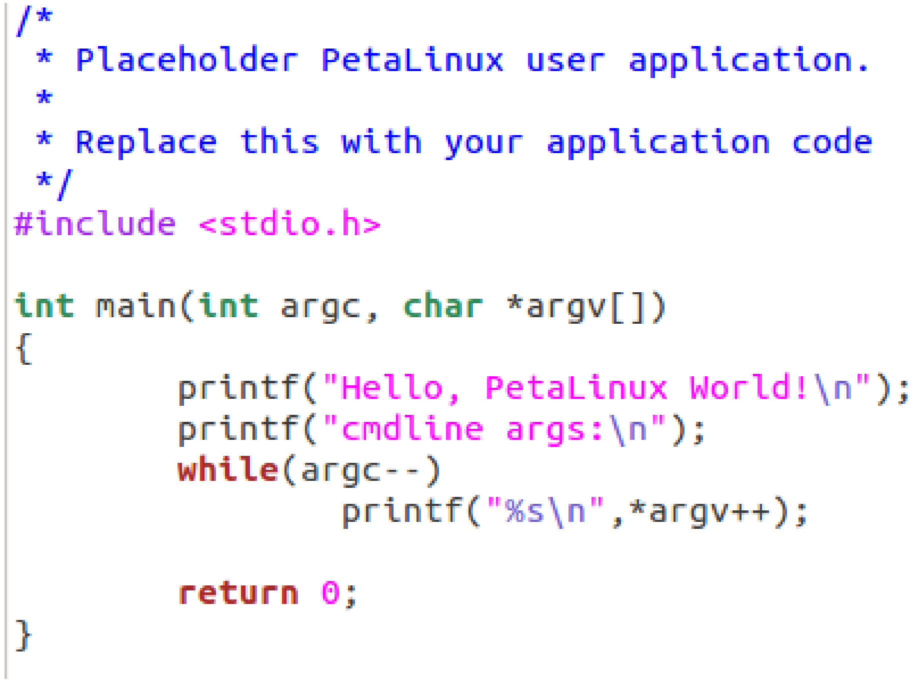


Steps to Create Project

1) Develop hardware design in Vivado; generate and export bitstream
2) Open terminal and source Petalinux tools; create new project with petalinux-create
3) Establish hardware support by directing exported Vivado hardware files to project folder
4) Develop and enable user applications
5) Build the project with petalinux-build
6) Copy BOOT.BIN and image.ub to SD card
7) Boot on ZedBoard and run user applications in terminal

Hello, Petalinux World

- Uses templated Avnet-Digilent-Zedboard board support in Petalinux
- Create Hello, Petalinux World application
- Build, boot, and run on the board



```
/*
 * Placeholder PetaLinux user application.
 *
 * Replace this with your application code
 */
#include <stdio.h>

int main(int argc, char *argv[])
{
        printf("Hello, PetaLinux World!\n");
        printf("cmdline args:\n");
        while(argc--)
                printf("%s\n",*argv++);

        return 0;
}
```

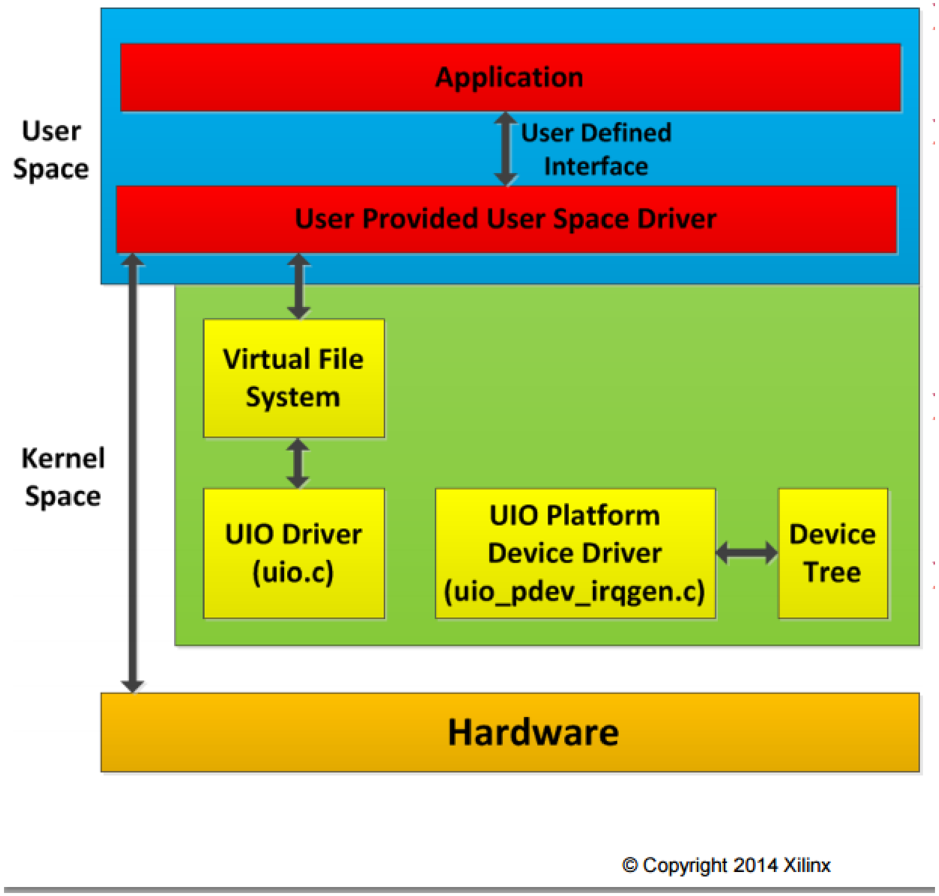
Accessing Physical Memory

- dev/mem:
  - Direct access to physical memory
  - Complete mapping of every physical memory address to a virtual memory address
  - Easiest to implement, but not flexible

- UIO:
  - Kernel driver facilitates hardware access from user space
  - User space driver maps memory using dev/uio and mmap
  - Applications written in user space can interface with hardware

Accessing BRAM

- Simple UIO driver method with uio_pdrv_genirq
- uio_pdrv_genirq: Basic built-in kernel driver
- Device tree structure created from hardware design export; just a few manual modifications needed
- Write values to BRAM from user space and read them from serial terminal



© Copyright 2014 Xilinx

Setting up Ethernet

- Requires modification of device tree file system-top.dtsi to incorporate ethernet
- Must enable dropbear server in kernel configuration
- SSH from computer
- File transfer with SCP

Next Steps

- Learn basics of kernel drivers
- Implement kernel driver and incorporate within general UIO setup
- Implement DMA memory transfer using a series of kernel space and user space drivers
  - Interrupt handling must be done in kernel space
  - Intermediate wrapper driver must interface with xilinx_xdma low level driver