Porting Linux Kernel on FPGA based Development Boards

Rita Nagar[1], Ravi Kiran Jadi[2], Prabir Saha[3]
[1]Mody Institute of Technology & Science, (Lakshmangarh) Sikar
[2]Lecturer, Dept. of Physics, Government Degree College, Baruva
[3]Assistant Professor, Dept. of ECE, NIT Meghalaya

Abstract
Embedded systems design is a most emerging field that integrates the hardware and software application. Linux has successfully made its imprints to the embedded world as its free source code and support of various processor architecture. For porting Operating System the kernel source code customized, cross-compiled and dumped on development board. The board surrounded by number of peripherals components that can be used to create a complex system on which embedded application can be run easily. It demands mostly in industrial applications. This paper gives the general idea of porting technique and developing application for embedded system.

Keywords
Linux, Kernel, FPGA, Operating System

I. INTRODUCTION
This paper introduces various technologies for the development of embedded systems such as microcontroller, DSP processor, ASIC, ARM and now FPGA. Although Intel supports various architectures like ARM, SPARC, PowerPC but PowerPC can`t be ignored when comes to performance. All system provides the reliability but power consumption and reconfigurabilty feature makes it different. The FPGA is known for its reconfiguration ability to implement new hardware modules in it. With the technology advance partially reconfigurable FPGAs have been developed which can alter a part of their fabric during runtime[1]. It introduces the porting of Linux kernel running on ppc and application development. Porting the Linux kernel on PowerPC integrates the hardware and software, which helps to develop the new standalone SOC support. Embedded system improves the performance. Linux is ported as it provides portability and support most architectures. Linux platform is used to develop Linux kernel image, cross compiler and other software required to run the kernel image on PowerPC.

II. SYSTEM DESCRIPTION
The porting of Linux kernel on Power is based on both hardware and software design. On software side board compatible kernel image is required and Base System Package of board is used to support the hardware. The Xilinx virtex5- ML507[2] is officially supported by five operating systems: Xilinx standalone, Xilinx Xilkernel, Linux_2_6 and VxWorks_6_5,VxWorks_6_3.Xilinx 12.2 generate the bit stream to create the BSP.[3] The Xilinx provides different peripheral to develop a system such as FLASH, JTAG, 10/100 Ethernet, DVI and USB support for communication. External RAM is also there for bigger application. The Embedded Linux host system port kernel on target board of 32-bit processor. Virtex5-ML507 FPGA board Building hardware bitstream for the supported board. The Linux2.6.34.7 kernel is used in this Project which has display drivers for monitor support and all the architecture are in <linuxkernel/arch>.

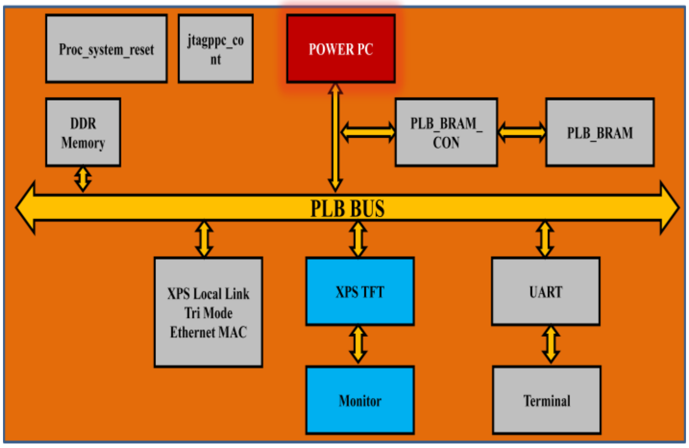

Fig1.Virtex5 development board

III. RUNNING LINUX ON XILINX FPGA
For building the Linux kernel we need just four software Cross-compiler, Linux kernel source, Root File system and Xilinx tool. First install cross-development tools, so that you can compile the kernel either by ELDK 4.2 and Croostool-ng.

3.1 Cross-compiler Toolchain

To build kernel image for PowerPC 405 or 440 in FPGA cross-compiler(CC) is needed[4].Commands for installing ELDK -

```
$ mkdir /mnt/CC
$ chmod ag+r /mnt/cc
$ mount -o loop ppc-2007-01-19.iso /mnt/CC
$mkdir /opt/ELDK
then exit from root user
$ chown <username> /opt/ELDK
$ cd /mnt/ELDK
$ ./install -d /opt/ELDK
```

In this way cross-compiler ELDK is install and ready to use.

3.2 Generate BSB (Base System Builder)

Linux BSB can be created using Xilinx tool by selecting OS platform as Linux_2_6. Set Mark as initialize brams as Default:ppc 440_0 to provide OS support to design so that it funstion as bootloader and generate Address , generate libraries and BSP ,ml507.dts will be created under implement in edk.To change the console to monitor make small modification in same dts other wise its default value will be ttyUL0.

bootargs = "console=tty0 rw root=/dev/ram";
linux,stdout-path = "/plb@0/tft@86e00000";

rw is necessary to make image read-write. Copy contents of this dts in kernel in kernel source (arch / powerpc / boot / dts /virtex440-ml507.dts). Generate netlist, Generate bitstream, update bitstream. In case of virtex4-ML410,xparameter_ml40x.h file located in build directory>ppc405_0/libsrc/linux_2_6_v1_00_c/linux/asch/ppc/platform/4xx/xparameters/xparameter_ml40x.h.At last we get download.bit bistream which we needed.

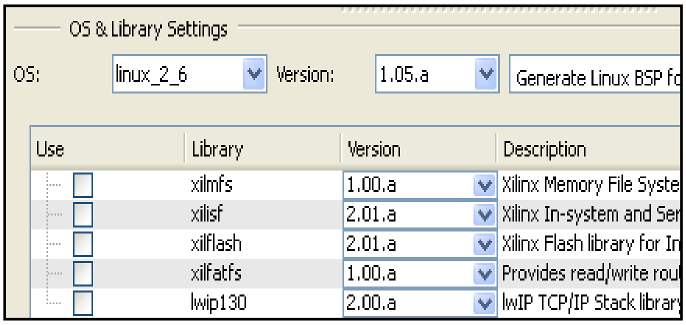

Fig2. OS & Library settings

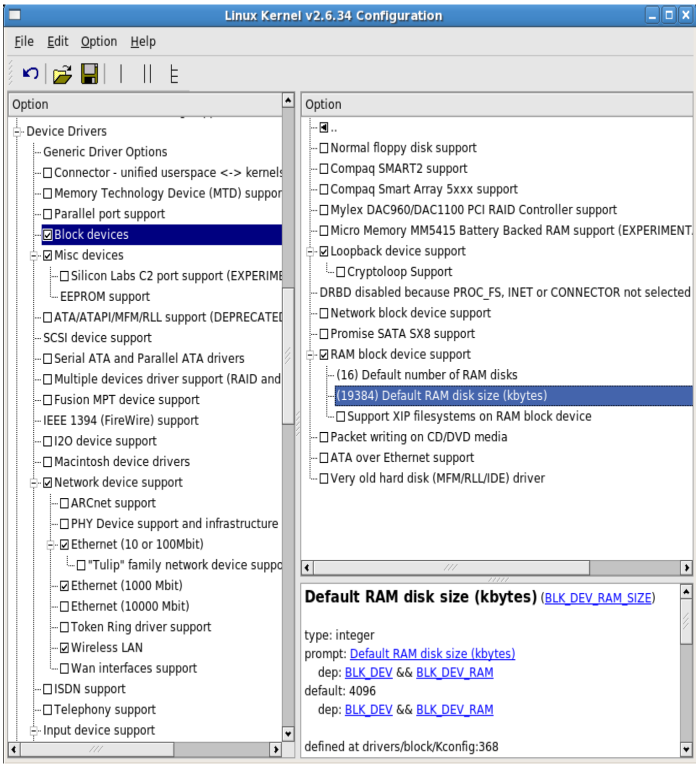

Fig2.1 .Customizing Linux

3.3 Customizing Linux kernel

The Linux kernel supports a lot of different processor architecture Linux. All development boards use the similar technique for porting but they are different for PowerPC and ARM. For getting hardware details PowerPC use linux_2.6 but in ARM in source code only [1]. Commands like make xconfig, make gconfig, make menuconfig are used to customize kernel. Navigate to kernel directory and first set the environment variable [3] and make configuration of FPGA board for PowerPC processor by –

 # export ARCH=powerpc
 # make 44x/virtex5_defconfig
Then change the file system size and other parameters –

 # make 44x/virtex5_xconfig

save the configuration and Copy file system in *ramdisk.image.gz* in the kernel directory(arch/powerpc/boot/). The default kernel images require a RAM disk to be present in the kernel source tree to act as the root file system. Finally build the kernel image by giving command on terminal.

$ make SimpleImage.initrd.virtex440-ml507

This will take 10-15 minutes depends upon the functionality we are using.A *simpleImage.initrd.virtex440-ml507.elf* will be created in kernel source directory (arch/powerpc/boot).Its size will be in MB.

IV. RUN LINUX SYSTEM

4.1 Downloading Reference Bitstream

There are two ways to download bitstream to FPGA system either we can do it from impact or from Xilinx. When bitstream download successful it show “Program Succeeded”.

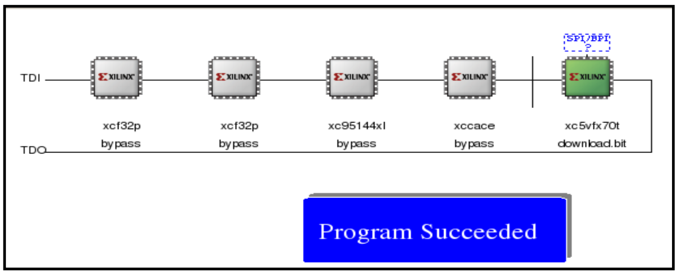

Fig3.bitsream downloaded using IMPACT

4.2 Download the kernel Image

We have given monitor support so we can see the kernel booting on monitor. For downloading kernel image from bash or EDK shell, launch the Xilinx Microprocessor Debugger (XMD) by using the following command:

[bash]$ xmd

Connect to the processor and download kernel image by using the following command:

XMD% connect ppc hw

XMD% dow simpleImage.initrd.virtex440ml507.elf

XMD% con

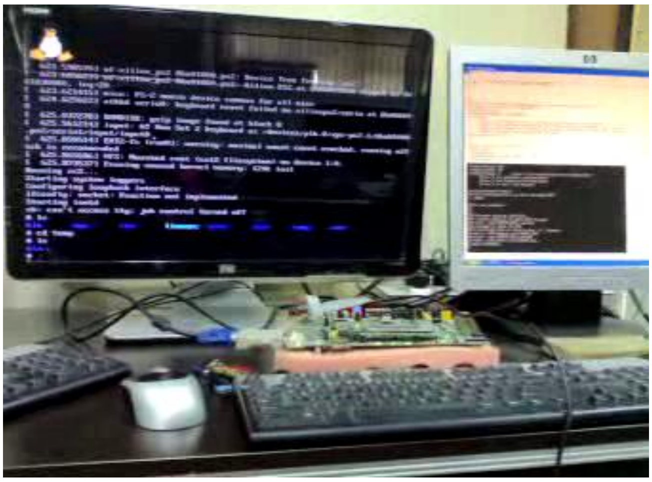

Fig 3.1 System setup

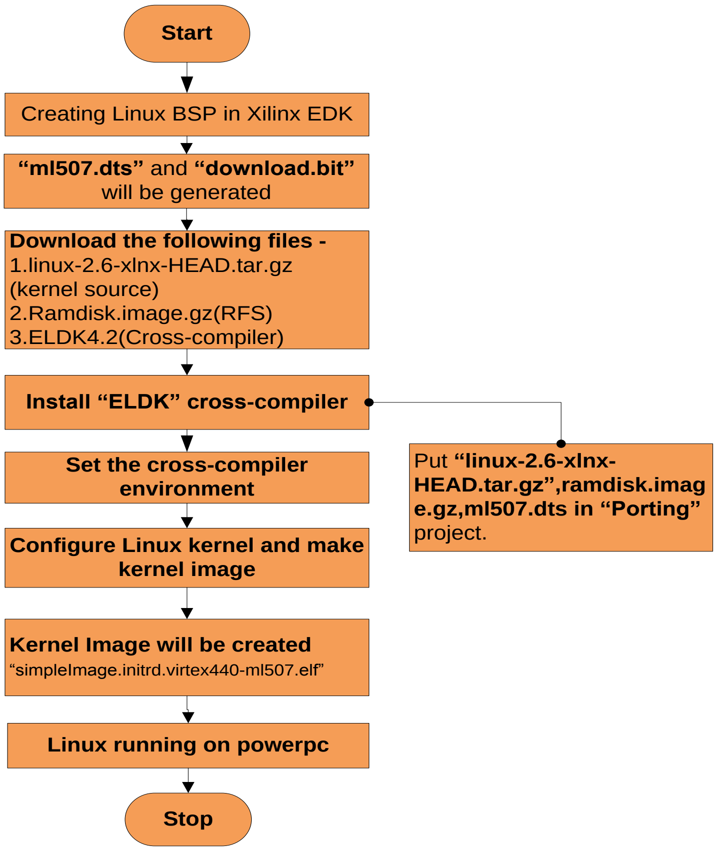

Fig 4.1 Software Development flow

V. RUN APPLICATION USING FTP

After porting we can also run application on PPC. For running application we have Ethernet connection so file can be easily sent to ramdisk by ftp after cross compiling it using ELDK. We can run application by transferring the application from host to target board using ftp. Set the board IP , connect both host and target in LAN. Open telnet service in host(windows XP) and transfer the cross-compiled file in target system.

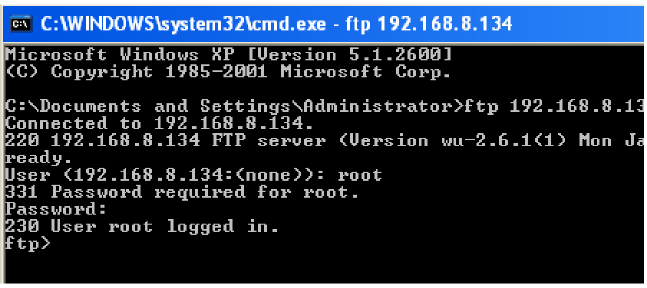

Fig4.ftp transfer of application

The application is in executable form in ASCII format before transferring convert in binary form. We are taking simple hello program.

ftp>binary

ftp>put hello

After transferring file to ramdisk(RFS) using ftp ,execute this application as in same way we do in Linux.

VI. CONCLUSIONS

This paper gives the details to customize kernel and the whole development flow of embedded system with FPGA and embedded Linux before carrying out the actual application. Further we can add and design peripheral IP cores and develop the corresponding drivers in Linux running on PowerPC.

ACKNOWLEDGEMENT

I am very thankful to Director Dr. Chandra Shekar, CEERI Pilani, and Dr. Kota Solomon Raju, for providing necessary facilities towards carrying out this work.

REFERENCES

[1] Zhou Qingguo; Yao Qi; Li Chanjuan; Hu Bin; , "Port embedded Linux to XUP Virtex-II Pro development board," *IT in Medicine & Education, 2009. ITIME '09. IEEE International Symposium on* , vol.1, no., pp.165-169, 14-16 Aug. 2009

[2] Pratyusha.gandham1 , ramesh n.v.k2, "porting the linux kernel to an arm based development board," K.L. University, A.P,India , Vol. 2, pp.1614-1618 Issue 2,Mar-Apr 2012.

[3] M. Tim. Jones , "Embedded Linux on the PowerPC," Embedded Linux Journal,Volume 2001 Issue 4, July/August 2001 Pages 36 - 39

[4]Kris Chaplin, "Getting Started using Git," 16 January 2009.

[5]KarimYaghmour, "Building Embedded Linux Systems", O'Reilly & Associztes ,May 2003.