Implementation of Custom Peripheral Interfaces on Linux Ported Embedded Processor in FPGA

Sowparnika A K, Sreedevi K, Swapna P P
M. Tech Student, VLSI & Embedded Systems, Department of Electronicss, Govt. Model Engineering College, Thrikkakkara
sowparnika.mec@gmail.com

Abstract— Linux has made steady progress in the embedded area as it is open source and supports various kinds of processor architectures. The Virtex-5 XC5VLX50T development board provides an advanced hardware platform that consists of a high performance FPGA. MicroBlaze soft core processor present in FPGA along with interfacing peripherals can be used to create a complex system to meet various embedded applications. This paper mainly discuss about porting Linux operating system to an Xilinx Virtex5 FPGA XC5VLX50T by configuring MicroBlaze soft processor inside it and its peripherals and implement various peripheral interfaces for communication between embedded systems. Hardware environment for standard interfaces namely UART and Ethernet is created using Xilinx ISE 13.1 tool and the embedded processor in Virtex5 FPGA is configured to have Linux with Memory Management Unit support for processor applications.

Keywords— Linux, Kernel, MicroBlaze, FPGA, Linux porting, Xilinx, Ethernet, UART

INTRODUCTION

The Field Programmable Gate Array (FPGA) is a general-purpose device which is filled with digital logic building blocks. As FPGAs become more powerful in terms of available reconfigurable hardware resources, they are increasingly used as accelerator devices because of their inherent ability to process more data in parallel. A common trend is to connect FPGA based board to a general purpose processor usually a general purpose computer, for offloading computationally intensive part or functions of an application to FPGA. The FPGA is characterized with its reconfiguration ability to implement new hardware modules in it. A processor built from dedicated silicon is referred to as a hard processor. Another category of processor is soft processor. A soft processor is built using the FPGAs general-purpose logic. The soft processor is typically described in a Hardware Description Language (HDL) or netlist.

This paper introduces steps to port Linux kernel on the soft core processor and implement the peripheral interfaces. The soft processor intended to be used is MicroBlaze configured on Xilinx Virtex5 FPGA XC5VLX50T evaluation module ML505. Porting the Linux kernel on MicroBlaze integrates the hardware and software. As embedded system improves the performance and the Linux provides portability and flexibility to support most processor architectures, a combination of both can make a considerable difference. Linux platform is used to build cross compiler as well as Linux kernel image required to run Linux operating system on MicroBlaze processor.

SYSTEM DESCRIPTION

The MicroBlaze embedded processor soft core is a reduced instruction set computer optimized for implementation in Xilinx Field Programmable Gate Arrays. Porting of Linux kernel on MicroBlaze 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. Five operating systems are available with Xilinx virtex5 ML505 board: Xilinx standalone, Xilinx Xilkernel, Linux_2_6 and VxWorks_6_5, VxWorks_6_3. Xilinx ISE 13.1 is used to create the hardware environment of the system. The linux-2.6-xlnx kernel used in this project supports MicroBlaze processor architecture. The Linux kernel is configured and cross-compiled using the Cross compiler (microblaze-unknownlinux-gnu-gcc-) and finally the kernel image is transferred to the target development board. The Virtex5 development board is shown in Figure1. MicroBlaze soft processor is configured on this XC5VLX50T FPGA present in the development board.

International Journal of Engineering Research and General Science Volume 4, Issue 3, May-June, 2016
ISSN 2091-2730

Figure 1: Virtex5 Development Board

BUILDING HARDWARE ENVIRONMENT

The Block Diagram of the MicroBlaze soft processor configured with peripheral interfaces is shown in the Figure 2. The peripherals used are UART, Ethernet, 8 bit LED and an xps_timer interface. While configuring UART interface, RS232_Uart interface type is selected as xps_uartlite with Buad rate 9600 bits per second. Data bit width is selected as 8 and Interrupt option is selected. Parity is selected as none. Ethernet Interface is configured to have Hard_Ethernet_MAC. Here also interrupt option is selected.

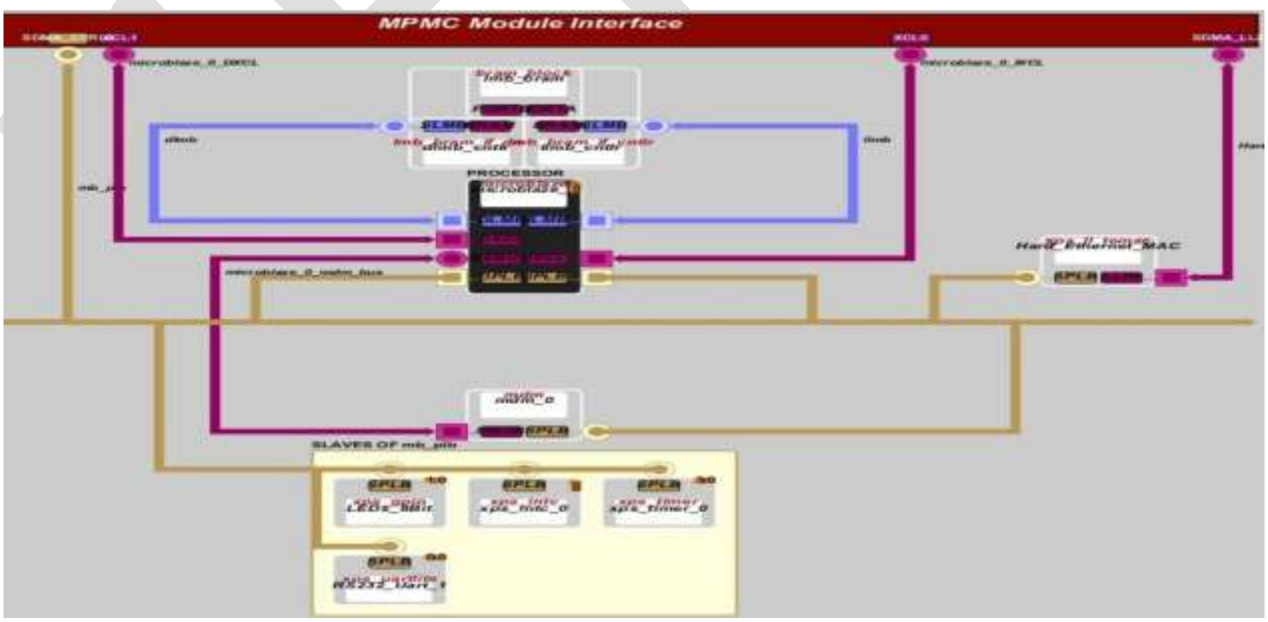

Figure 2: Block Diagram

Configure MicroBlaze processor to have Linux with MMU option as in Figure 3. Other essential options like Enable Floating Point Unit, Enable Barrel Shifter, Enable Integer Multiplier, Enable Integer Divider etc are enabled. Memory Management option is selected as VIRTUAL. A ucf file for the system is also added.

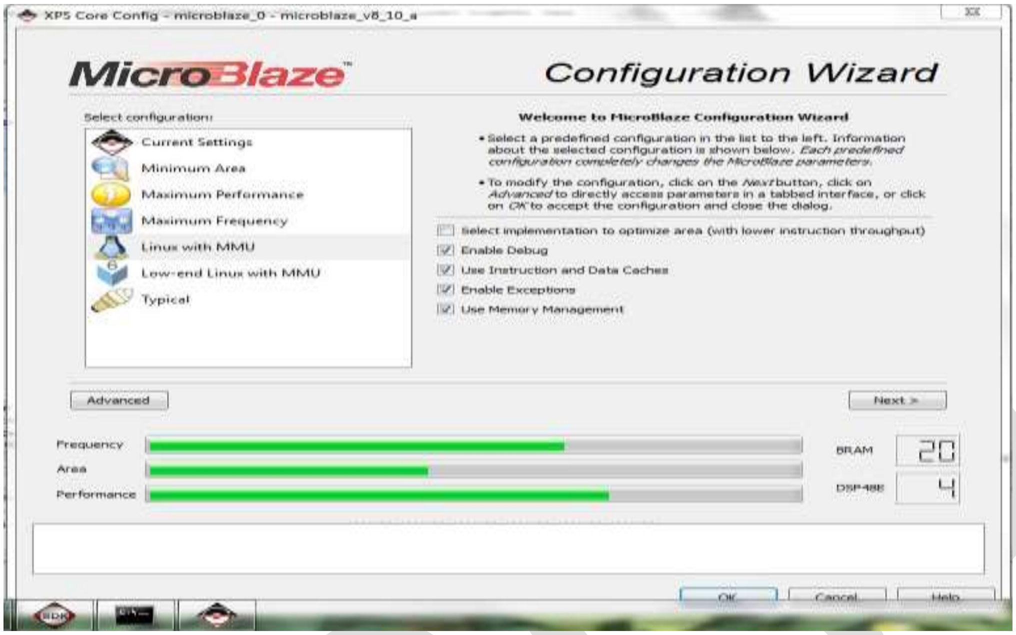

Figure 3: MicroBlaze Configuration Wizard

BUILDING BSP

Linux BSB is created using Xilinx tool by selecting OS platform as device-tree_v0_00_x using Xilinx SDK tool. Bootargs value is given as console=ttyUL0 and console device is selected as RS232_Uart_1. Xilinx.dts and system.mhs files are created using Xilinx EDK tool. These two files are required in configuring the Linux kernel

CONFIGURING AND BUILDING LINUX KERNEL

Cross-compiler, Linux kernel source code and Root File system are required for building the Linux kernel image. The tool chain used in the entire kernel building process is microblaze-unknown-linux-gnu-. For getting hardware details, Xilinx.dts file formed under implementation folder in EDK is copied to *linux-2.6-xlnx/arch/microblaze/boot/dts* folder.

$ cp source_dir_of_Xilinx_dts_file/Xilinx.dts linux-2.6- xlnx/arch/microblaze/boot/dts/ Xilinx.dts

Configuring Linux Kernel

The Linux kernel supports a large number of processor architectures. The Linux Kernel is configured using the tool chain for MicroBlaze architecture. The parameters used in the configuration steps are taken from the system.mhs file formed under implementation in EDK. The parameters include, USE _BARREL_ SHIFT, USE_FPU, the kernel base address, etc.

For customizing the kernel according to our requirement, commands like make xconfig, make defconfig, make menuconfig are used. The default kernel images require a RAM disk to be present in the kernel source tree to act as the root file system. Finally the kernel is configured according to the values in system.mhs file using the following commands on terminal.

$make ARCH=microblaze CROSS_COMPILE= Source_dir_of_toolchain/microblaze-unknownlinux-gnu-gcc- mmu_defconfig

$make ARCH=microblaze CROSS_COMPILE= Source_dir_of_toolchain/microblaze-unknownlinux-gnu-gcc- menuconfig

Compiling Linux Kernel

To cross-compile the Linux kernel the following command is used.

$make ARCH=microblaze CROSS_COMPILE= Source_dir_of_toolchain/microblaze-unknownlinux-gnu-gcc- simpleImage.xilinx

This will take 15-20 minutes depending upon the functionalities selected. The Linux Kernel Image named as simpleImage.xilinx is formed in kernel source directory ie, at arch/microblaze/boot. Its file size will be in MB.

DOWNLOADING LINUX KERNEL IMAGE

To download bitstream to FPGA system, the virtex5 evaluation board is connected to PC using JTAG interface. The kernel image is downloaded through this debugging interface tool. For downloading kernel image from EDK shell, Xilinx Microprocessor Debugger (XMD) is used with the necessary commands as shown.

$xmd

XMD% connect mb mdm

XMD% dow simpleImage.xilinx

XMD% run

The Linux booted on MicroBlaze processor with filesystem listed can be viewed using Teraterm window. It is shown in the Figure 4. The root filesystem created for Linux is listed with its different directories.

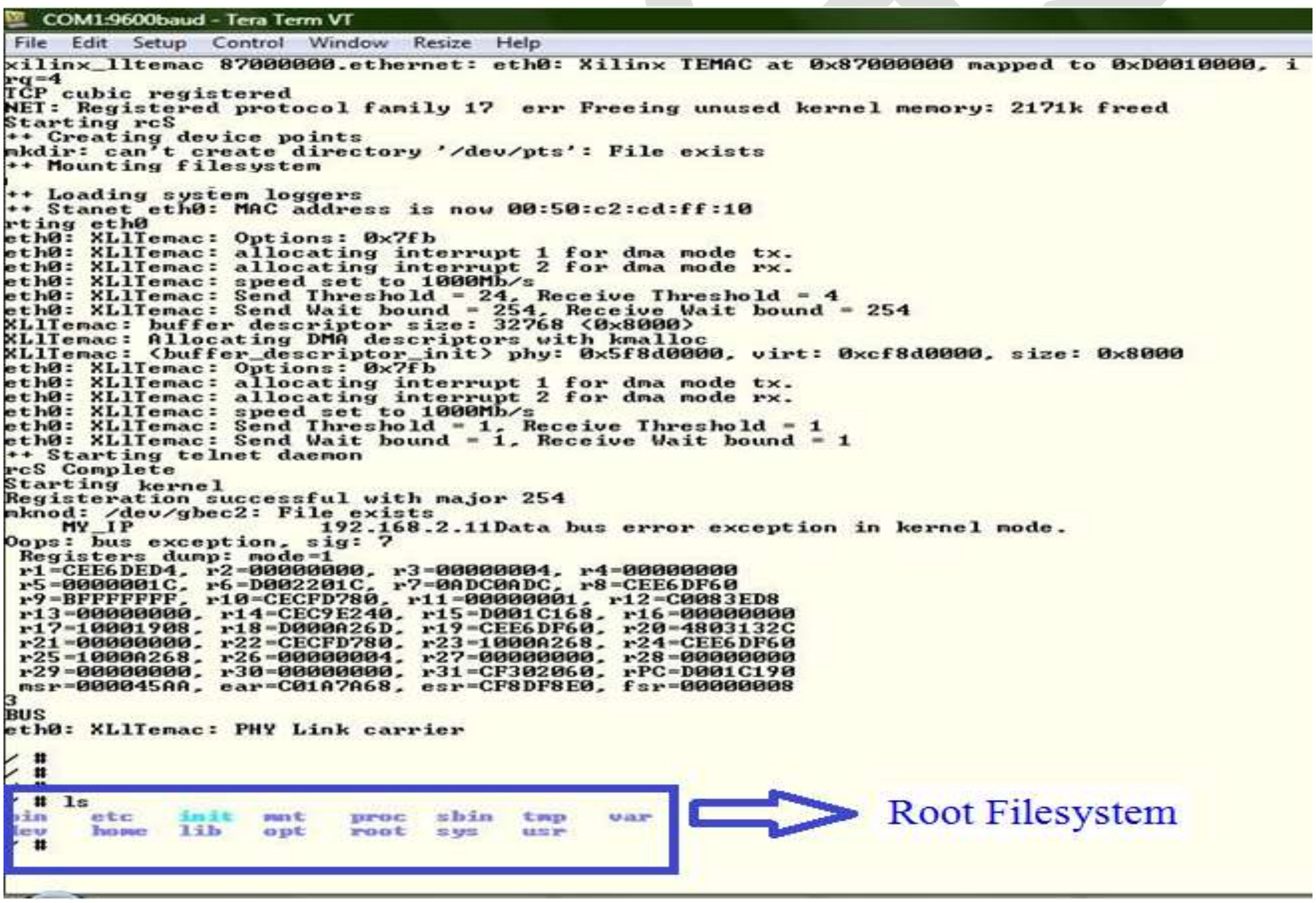

Figure 4: Booted Linux Kernel Image

PERIPHERAL INTERFACE IMPLEMENTATION

UART

After porting Linux on MicroBlaze processor on Virtex5 evaluation platform, for implementing and testing the UART interface, connect the Virtex5 board to PC using serial cable. Open Teraterm pro tool and establish a new serial port connection between development board and t he PC. Teraterm is configured to have a serial connection with baud rate of 9600 bits per second and data bit width as 8 bits. As the UART interface is up and working, booting of Linux Image is viewed through the Teraterm window. Verification of serial interface is successfully carried out through the process.

ETHERNET

After porting Linux image, connect the development board and the PC through Ethernet cable. The PC is assigned an IP address in the same network as the target development Board. The connection between these devices is tested using ping command as given in the Figure 5.

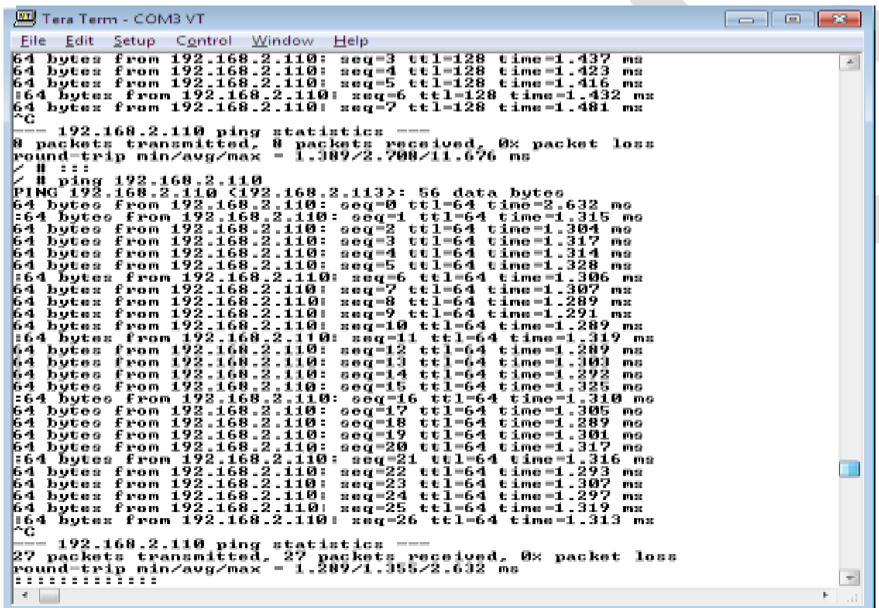

Figure 5: Ethernet Connection

ACKNOWLEDGMENT

The authors thank Shri. S. KedarnathShenoy, Scientist ‘G’ Director, NPOL for granting permission to carry out this project in a successful manner. Authors also thank Mr. Suresh M., Scientist ‘G’, NPOL, Mrs. Jayamma T M, Scientist ‘F’ and Mr. Anilkumar K, Scientist ‘D’, NPOL, for providing an opportunity with valuable guidance, help, and insightful comments.

CONCLUSION

Embedded Linux operating system is configured and compiled for MicroBlaze soft core processor architecture and ported to Xilinx Virtex5 XC5VLX50T FPGA on evaluation board ML505. UART and Ethernet interfaces are implemented on FPGA for communication between various embedded systems.

REFERENCES:

[1] Rita Nagar, Ravi Kiran Jadi, and Prabir Saha, “Porting Linux on FPGA basedDevelopment Boards,” International Conference on Computing, Communication and Sensor Network(CCSN) 2012.

[2] ZHOU Qingguo, YAO Qi, LI Chanjuan, Hu Bin,“Port Embedded Linux to XUP Virtex-II Pro Development Board” Distributed & Embedded System Lab (DSLab), Lanzhou University, China International Conference on Computing, Communication and Sensor Network 2012.

[3] Rajendra Prasad.M, S.Ramasubba Reddy, V. Sridhar,“ Framework To Port Linux Kernel On PowerPC Based Embedded System Used For Telecom Application-IPBTS”,International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.4, October 2011.

[4] Karim Yagmour,Jhon Masters, Building Embedded Linux Systems, 2nd ed., O’Reilly Media, Inc.1005 Gravenstein heighway, North, Sebastopol CA 95472.

[5] MicroBlaze Processor Reference Guide, UG081 (v13.3).

[6] ML505 Evaluation Platform User Guide, UG347 (v3.0.1) July 21, 2008.

[7] Michel Johnston and Nigel Dick, “Portinng Linux to the MPC6280ADS”, Freescale Semiconductor Application Note, Document Number : AN2579, Rev.1, 11/2006.

[8] Prathyusha Gandham, Ramesh N.V.K, “Porting the Linux Kernel to an ARM based Development Board”, International Journal of Engineering Research and Applications(IJERA) ISSN: 2248-9622 Vol. 2, Issue 2, Mar-Apr 2012, pp.1614-1618.

[9] Andres M. Leiva-Cochachin, Fredy Chalco- Mendoza,“Embedding Linux with Ability to Analyze Network Traffic on a Development Board based on FPGA”, International Journal of Computer Applications (0975 8887) Volume 77 No.17, September 2013.

[10] K. Eswar Kumar, M. Kamaraju, Ashok Kumar Yadav, “Porting and BSP Customization of Linux on ARM Platform”, International Journal of Computer Applications (0975 8887), Volume 80 No 15, October 2013.

[11] Archana Mahatme, V.P.Mahatme, “Soft Core An Approch to Embedded System Design”, International Journal on Recent and Innovation Treands in Computing and CommunicationISSN: 2321-8169 Volume:3 Issue: 2045049.

[12] Mahendra Swain, Abhishek Kumar Srivastava,“Custom Linux Kernel for Raspberry Pi Using Ubuntu 12.04 Host”, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735. PP 13-18