ECEN 449 - Lab Report

Lab Number: 4

Lab Title: Linux boot-up on ZYBO Z7-10 board via SD Card

Section Number: 511

Student's Name: Kris Gavvala

Student's UIN: 929000158

Date: 3/26/2025

TA: Gautham Nemani

Introduction:

In this lab we booted linux onto the ZYBO Z7-10 board from our own microblaze processor system block design. We customized the block design with our multiply peripheral that we designed in the previous lab.

Procedure:

First, we created the block design for out processor system. We used the ZYNQ7 Processing System IP and enabled SD 0, UART 1 and TTC 0 peripherals. We then added the multiply IP peripheral that we created previously and ran the block automation to connect all parts of the system. We exported this design through a .xsa file to use for use in petalinux.

After installing petalinux in the ubuntu workstation, we created a project using the 'zynq' template and configured it using the hardware .xsa file imported from vivado.

After building the petalinux project, the zynq boot image was generated and copied to an SD card along with the image.ub and boot.scr files.

To boot linux on the Zybo z7-10, we plugged the SD card into the board and pressed the reset button to start booting linux. We then ran picocom from the ubuntu workstation terminal to see output from the linux system on the zybo board.

Results:

Linux was successfully booted onto the zybo board and its output was observed through the terminal.

Conclusion:

This lab was a good tutorial on how to use a processor system design in microblaze to boot linux on the zybo-z7-10 (or any fpga). Through this lab we learned how to use petalinux to generate files for booting linux on an fpga. This will be useful because we can directly run software on the processor system on zybo board.

Post-lab Deliverables:

Submit a lab report with the following items:

2. [8 points.] Correct format including an Introduction, Procedure, Results, and Conclusion. Be sure to summarize the process required to build the hardware and compile the Linux kernel. Warning: Missing information will result in missing points.

3. [2 points.] The output of the terminal (picocom) showing the Linux boot.

```
Run /init as init process
INIT: version 3.04 booting
Starting udev
Starting version 251.8+
ext3: Unknown parameter 'umask'
ext2: Unknown parameter 'umask'
ext4: Unknown parameter 'umask'
FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
hwclock: can't open '/dev/misc/rtc': No such file or directory
Fri Mar  9 12:34:56 UTC 2018
hwclock: can't open '/dev/misc/rtc': No such file or directory
random: crng init done
Configuring packages on first boot....
 (This may take several minutes. Please do not power off the machine.)
Running postinst /etc/rpm-postinsts/100-sysvinit-inittab...
update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge (continuing)
 Removing any system startup links for run-postinsts ...
  /etc/rcS.d/S99run-postinsts
INIT: Entering runlevel: 5
Configuring network interfaces... Cannot find device "eth0"
Starting OpenBSD Secure Shell server: sshd
  generating ssh RSA host key...
  generating ssh ECDSA host key...
  generating ssh ED25519 host key...
done.
Starting rpcbind daemon...done.
starting statd: done
hwclock: can't open '/dev/misc/rtc': No such file or directory
Starting internet superserver: inetd.
NFS daemon support not enabled in kernel
Starting syslogd/klogd: done
Starting tcf-agent: OK

PetaLinux 2023.1+release-S05010539 linux_boot ttyPS0

linux_boot login: petalinux
You are required to change your password immediately (administrator enforced).
New password:
Retype new password:
linux_boot:~$ sudo su

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

Password:
linux_boot:/home/petalinux#
FATAL: read zero bytes from port
term_exitfunc: reset failed for dev UNKNOWN: Input/output error
kgavvala19@zach-333-em4-06:/$
```

4. [4 points.] Answers to the following questions:

(a) Compared to lab 3, the lab 4 microprocessor system shown in Figure 1 has 512 MB of SDRAM. However, our system still includes a small amount of local memory. What is the function of the local memory? Does this ‘local memory” exist on a standard motherboard? If so, where?

ANSWER: The local memory can be used for the boot loader, interrupt handling, fast data transfers between peripherals and main memory, and data caching. On a standard motherboard, this local memory is probably equivalent to the CPU cache and resides close to the CPU for fast data transfer tasks.

(b) After your Linux system boots, navigate through the various directories. Determine which of these directories are writable. (Note that the man page for ’ls’ may be helpful). Test the permissions by typing ‘touch ’ in each of the directories. If the file, , is created, that directory is writable. Suppose you are able to create a file in one of these directories. What happens to this file when you restart the ZYBO Z7-10 board? Why?

ANSWER: The file gets deleted upon restart because the root filesystem is stored in RAM, which is temporary memory.

(c) If you were to add another peripheral to your system after compiling the kernel, which of the above steps would you have to repeat? Why?

ANSWER: You would have to reconfigure and rebuild the petalinux project because the device tree needs to be updated. The device tree describes the hardware configuration to the Linux kernel.