ECEN 449 Lab Report 4

Philip Smith - 624002014 (Sec. 511)

February 17, 2020

Introduction

Similar to the previous lab this time we are tasked with synthesizing and implementing a Zynq processor. Instead of only running one program on it, however, we will compile and install the Linux kernel onto it. This is accomplished by installing a series of boot loaders that will eventually lead to a normal Linux boot procedure from an initial ramdisk. This particular computer will have DDR3 RAM, a memory management unit, SD card peripheral, and interrup controller, as well as a timer.

Procedure

We follow roughly the same procedure as in Lab 3 by creating the Zynq 7 processor, however this time we will enable several additional peripherals to be used by the kernel in addition to our multiply IP module.

In addition to burning the Zynq 7 on the FPGA we must also compile U-boot and the Linux kernel.

Results

During this lab I encountered several issues mainly because of how I was organizing my work. I use a git system to distribute my work wherever I need it. When pushing the compiled linux kernel to my laptop there were issues with files not getting transferred which I later found was due to the .gitignore inside that particular directory structure. To mitigate this I simply archived it first.

Other than that there were no issues and the lab went smoothly.

Conclusions

Questions

1. Because this memory is located internally (inside the CPU) I am led to believe that this local storage serves as a CPU cache. Such memory would reside in all modern processors within the CPU as it considerably reduces the amount of time to access repeated data.

2. This is a standard GNU/Linux system and therefore we can determine this by running the following command:

```
# ls -l / | grep "^dr-"
dr-xr-xr-x  95 root root     0 Feb  5 03:56 proc
dr-xr-xr-x  13 root root     0 Feb 17 16:56 sys
```

   All other root directories are writeable by root or an appropriate account. Changes made are not persistent and will not survive a reboot.

3. Adding a new peripheral would constitute recompiling the kernel with the new address space information added/updated.

Appendix A -

The following is the code for :