Ethernet on the Zynq ZC706

18-545 Advanced Digital Design

Terence An, Eddie Nolan, Dale Zhang

December 12, 2015

Chapter 1

Introduction

This paper is a guide to start building ethernet on the Zynq ZC706 board. It was originally written as the final project for a 18-545 project which didn't complete because they struggled to build an ethernet adapter in programmable logic and have it properly communicate with the Processing System. The intention of this paper is to aid future groups in completing an ethernet adapter, as well as providing the necessary background and deterring groups from fruitless avenues. This guide will expect a very minimal understanding of Vivado because most students in 18-545 have had very limited exposure to Vivado. We will attempt to provide the pertinent references as needed.

That being said, going through Lab 2 in *Vivado Design Suite Tutorial* [6] will probably be the fastest way to understand the work flow. Also, chapter 2 and chapter 4 of *UltraFast Design Methodology Guide for the Vivado Design Suite* [5] will be superbly helpful in learning to use Vivado, especially for using Intellectual Property (IP) in Vivado. Finally, if you still want more details on using IP, you can refer to the Vivado guide on *Designing with IP* [3] and *Designing IP Subsystems Using IP Integrator* [2]. If you'd like more information on Vivado in general, refer to the *Getting Started* [4] guide and the *Designs Flows Overview* [1].

Chapter 2

Ethernet Background

In this chapter we'll provide the basics of ethernet, just enough to get you started. We start at the lowest level, and work our way up to the peripheral port on the processing system. If any of these sections are found to be lacking, you can find more information from the 802.3ab standard available on the IEEE Standard Association. Wikipedia is also your friend.

2.1 Logical Link Layer

The logical link layer may also be referred to as the physical layer. This network layer deals with how the bits are formatted into frames and how they're transmitted.

2.1.1 Ethernet Frame

T

802.3 Ethernet packet and frame structure

Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46(42)[b]–1500 octets	4 octets	12 octets
Layer 2 Ethernet frame			← 64–1518(1522) octets →						
Layer 1 Ethernet packet	← 72–1526(1530) octets →								

Figure 2.1: Ethernet Frame Format
source:https://en.wikipedia.org/wiki/Ethernet_frame

Chapter 3

Basic Approaches

Chapter 4

Ethernet PL IP Guide

Chapter 5

PetaLinux Networking

Chapter 6

Alternate Approaches

Chapter 7

More on PL

Chapter 8

Miscellaneous

8.1 Personal Statements

8.1.1 Dale Zhang

For me, this class was challenging for a multitude of reasons. First off, I hadn't touched Verilog since I took 18-240 a few years ago, and I also wasn't very comfortable with HDL. In addition, in the context of our project, I had very little knowledge on ethernet and networking, so I had to learn a lot about how our project worked as I went along.

Over the course of the semester, there are definitely a few things I wish I had done differently. Since Terence and Eddie were both far more knowledgeable about Ethernet and networking, I often took a backseat to them when the group was making decisions. However, at some points in the semester, instead of asking for their help understanding some of the concepts driving our design, I would try to do it myself, without very much success. This led to me spending far more time on some tasks than I should've. This definitely limited my effectiveness as a team member.

Another mistake we made as a team was underestimating how much work actually needed to go into this project. Towards the beginning of the semester, we didn't put in much lab time outside of class periods and mandatory lab time. It first really caught up to us around mid semester with the first status meeting, where we saw how far behind we were, and how much more time we would need to commit for the rest of the semester.

Some advice I'd have for anyone planning to pursue an FPGA Ethernet project in the future is not to spend too much time trying to do research, and to start actually working on the board as soon as possible. In addition, ethernet on FPGA is not very well documented, and much of the documentation available is incorrect or incomplete.

For the class in general, it's definitely better to spend the long hours working on your project earlier in the semester, before your other classes have started to pick up. In addition, at the beginning of the semester, try and pick a project that you can be passionate about and that you would really like to see succeed. At times, I felt very unmotivated to go in an work on the project simply because I wasn't particularly excited about our final product.

To all future students reading this, good luck with the class and have fun!

Bibliography

[1] Xilinx. *Design Flows Overview*. 2015. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug888-vivado-design-flows-overview-tutorial.pdf.

[2] Xilinx. *Designing IP Subsystems Using IP Integrator*. 2015. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug994-vivado-ip-subsystems.pdf.

[3] Xilinx. *Designing with IP*. 2015. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug896-vivado-ip.pdf.

[4] Xilinx. *Getting Started*. 2015. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug910-vivado-getting-started.pdf.

[5] Xilinx. *UltraFast Design Methodology Guide for the Vivado Design Suite*. 2015. URL: http://www.xilinx.com/support/documentation/sw_manuals/ug949-vivado-design-methodology.pdf.

[6] Xilinx. *Vivado Design Suite Tutorial*. 2015. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug888-vivado-design-flows-overview-tutorial.pdf.

Appendix A

Appendix