Enabling Gigabit IP for Embedded Systems

Nicholas Tsakiris

B. Eng. (Computer Systems) (Honours)

Flinders University of South Australia

A Thesis Submitted for the Degree of Masters by Research

Flinders University

School of Computer Science, Engineering and Mathematics

Adelaide, South Australia

2009

(Submitted 17th February 2009)

Dedicated to my parents for their support and love.

Abstract

For any practical implementation of chip design, there needs to be a hardware platform available for the purpose of prototyping and implementation of FPGA-based programs, whether they are written in VHDL or Verilog. Communication between the platform and a computer is a useful feature of many hardware solutions as it allows for the capability of regular data transmission between the two devices. Furthermore, the ability to communicate between the platform and a computer at high-speeds requires a specially constructed interface, one that can be modified by the designer at their choosing.

There are a number of commercial packages which provide a hardware platform to perform this task, however there are drawbacks to many of the available options. Some may require special hardware to connect to a computer using proprietary connectors or boards, which increases the cost and reduces the flexibility of any solution. Other options may have limited access to the internal structure of the interface, limiting the ability of the developer to modify the interface to suit their needs. There may be an extra cost to provide the code to the interface, separate from the board, which can also tax design budgets.

This dissertation provides a solution in the form of a Gigabit Ethernet connection with a custom IP/network layer written in VHDL to facilitate the connection. With an increasing number of IP-enabled devices available such as IPTV and set top boxes, the ability to link hardware using Ethernet is very useful and so the development of

a lean and capable network layer was considered a suitable focus for the project. The overall goal has been to provide an interface which is cheap, open, robust and efficient, retaining the flexibility a developer might require to modify the code to their needs.

After covering some basic background information about the project, the dissertation looks at the requirements of the board and interface, as well as the alternative interface solutions which were looked at before deciding on Gigabit Ethernet. The protocols used in Ethernet are then covered, with both an explanation of the structure of each and their relevance to the implementation. The Finite State Machines which control operation of the interface are covered in depth, with an explanation of their interconnectivity to each other and how they fit in the data-flow between the computer and the board. Error correction and reliability is discussed, as well as any remaining components critical to the operation of the interface.

Pipelining, the method of design which provides the speed required for Gigabit Ethernet, is covered along with the extra speed optimisation techniques used in the design such as RAM swinging buffers. Testing and synthesis are covered which ensure the design is as robust as possible, both in simulations and in real-world applications. The final design was implemented on a Xilinx Spartan 3 FPGA (XC3S5000-5FG900C) and capable of a maximum speed of 128.287 MHz, which is more than enough to satisfy the requirements of Gigabit Ethernet under a variety of network conditions. The interface code occupies 1,166 slices of logic on the FPGA (3% of the total amount of logic available), making it sufficiently compact to run large projects on the same chip. The core was tested on physical hardware and performed correctly at real line Gigabit speeds. Configuration of the computer along with the method of connecting to the board and transferring data is mentioned, with explanation of the code run on the computer to make this possible. Finally, the dissertation provides an example application through the use of JPEG2000 image compression/decompression.

"I Nicholas Tsakiris, certify that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text."

Candidate:

Nicholas Tsakiris

Acknowledgements

I would like to thank my supervisor, Professor Greg Knowles, for his invaluable help in getting me through my candidature. To my colleagues at the Flinders University of South Australia, School of Informatics and Engineering, I would like to say a grateful thank you. In particular, to Paul Gardner-Stephen for his help in understanding the intricacies of networking protocols, as well as Geoff Cottrell, Craig Peacock and Terry MacKenzie for their assistance. I would also like to thank the academics and staff of that school for their continual support over the past few years. Finally, I would like to thank my friends and family who have helped me during the period of my candidature.

Contents

Bibliography **128**

List of Figures

List of Tables

Chapter 1

Introduction

FPGAs (Field-Programmable Gate Arrays) are a useful tool in the electronics industry for constructing prototype designs before mass fabrication onto dedicated hardware and are often used themselves as part of the final design. They are re-programmable, flexible and extendable, with the capability to run several programs at once and at different speeds. For many designs, interfacing with a computer may be required for data I/O, programming and debugging. If the requirements of the design call for high-speed data transfer with a computer, it is preferable to find some way to accomplish this using an existing interface on the computer, to maximise portability and reduce the dependency on specially-designed hardware.

A particularly common interface on many computers is the Ethernet port, normally used for wired network connections to LANs and WANs. The commonality of this interface makes it ideal for interfacing with an FPGA prototyping board, particularly if both network adaptors involved are capable of Gigabit speeds. However, to actually receive and transmit data using Gigabit Ethernet and have that data available for other programs that reside on the FPGAs is not necessarily straightforward, particularly if one wishes to customise aspects of the interface. There are IP cores available for purchase from several vendors which can provide Gigabit Ethernet functionality for FPGAs, but these generally reside as black-boxes and due to them being distribu-

ted as encrypted netlists, do not provide the designer with anything but the inputs and outputs of the core, which makes them unsuitable for the designer who wishes to modify the interface code directly. Sometimes it is possible to obtain the source code for these black-boxes, but the extra cost of the code can add substantially to the overall cost of the IP core. For example, Alcatel provides a fully-featured Gigabit Ethernet core for Altera FPGAs,[2] however costs start at $30,000 for an encrypted netlist without code. The source code can be purchased, but for an additional cost.

The purpose of this dissertation is to cover the design of a custom IP/network layer, one which has low cost, high reliability and an open structure for easy manipulation. The primary focus was to find an efficient engineering solution to a practical problem, the problem being how to develop the layer to work with low-power devices. Efficient engineering would solve this problem and provide the ability to use low cost hardware to support Gigabit Ethernet line speeds. The design of the core makes it streamlined for typical FPGAs and does not require higher-end hardware.[3,4] The base platform for its design was a Xilinx Spartan 3 FPGA, but the core can be implemented on other FPGAs so long as the base clocking speed of 125 MHz can be obtained. It is not just FPGAs which would benefit from such a design; there are also an increasing number[5] of IP-enabled devices (eg. IPTV, set-top boxes, fridges) which would benefit from a fast and lean network layer without the bloat of extra protocols and functionality which would not be needed in these highly-specialised devices. For this to be achieved, certain features which are available with commercial solutions are not present, but the benefits of a simpler core are evident once the designer has to put the solution to use. The dissertation also covers the physical implementation of the core on real-world hardware as well as the tests performed to validate the core's accuracy and reliability.

Achieving these requirements and solving the problem of an efficient design required some compromises. ARP support was not implemented due to lack of time. TCP

support was not implemented due to the fact that the protocol is never implemented entirely in hardware but rather a software/hardware combination using an embedded CPU, which was not available with the sole Spartan 3 FPGA. The Treck TCP/IP core for Xilinx FPGAs is an example of a core which could perform as an offload engine for processing TCP packets, when run on an embedded or soft processor on an FPGA such as MicroBlaze or a PowerPC CPU.[6] However, even with an embedded CPU the size of the core would increase in size and complexity to a level that was not desirable for achieving the lean and clean architecture, which were part of the goals of the design. The lack of packet error detection/correction that is an inherent part of TCP was still provided through the use of tags. The issue of achieving full Gigabit speeds on the base hardware (the Spartan 3) was ultimate the main factor in determining how to construct the core and still satisfy the requirements of the problem.

This chapter introduces several important concepts and ideas which are needed to fully understand the issues raised in this dissertation. Section 1.1 provides a brief introduction into FPGAs, what they are and how they can be used. Section 1.2 explains what VHDL is and what its purpose is with regards to chip design. Finally, Section 1.3 provides a short introduction to Ethernet extending to Gigabit Ethernet and its purpose for this design.

1.1 FPGAs

A *field-programmable gate array* is a semiconductor device which contains logic components (also known as logic blocks) which are programmable. By selectively programming the device these logic blocks can function as basic logic gates such as AND, OR, XOR, NOT, or can be extended into more complex combinational functions such as encoders, decoders or simple mathematical functions. Modern FPGAs also contain

special logic designed to act as memory elements such as RAMs or FIFOs and depending on the type of FPGA the memory elements may be constructed from flip-flops or dedicated memory blocks on the chip. The key function of an FPGA is to provide the ability to run logic programs with the advantage that the FPGA can be re-programmed multiple times, whereas a regular integrated circuit with support for logic gates would have a fixed design, permanently selected and unable to be altered. Despite being slower than a dedicated chip with a permanent design, FPGAs have a much greater level of flexibility and coupled with the ability to easily be reprogrammed, are ideal for running prototype designs and also for performing multiple tasks with the same hardware.

FPGAs have existed since the mid 1980's when Xilinx released the XC2064, the first FPGA. Despite only supporting a size of 1,000 gates, compared to sizes 10,000 times greater in 2004, this initial form of the FPGA proved very popular.[7] The ability to program the same chip over and over again provided cost-effective design development and increased the development of chip design theory and application. FPGAs have a wide range of applications, from digital signal processors (DSPs) to cryptography and beyond.

1.2 VHDL

To program an FPGA, a design-entry language suitable for specifying how the logic blocks interconnect together to perform their tasks is used. For this we use a Hardware Description Language (HDL), which encompasses any computer language specifically designed to formally describe electronic circuits. There are two main languages for this purpose: VHDL and Verilog. VHDL (VHSIC Hardware Description Language, fully expanded as Very-High-Speed Integrated Circuit Hardware Description Language)[8] is the language used by the Gigabit Ethernet project in this dissertation. It is capable

of rendering the entire structure of the FPGA including logic, connections and ports and also allows easy simulation capability due to the construction of a testbench. Verilog[9] is another widely-used HDL, but although Verilog is somewhat simpler and easier to code, VHDL was chosen for this design for reasons of familiarity.

1.3 Ethernet

1.3.1 History

Ethernet is the most common technology used on Local Area Networks (LANs) today. Developed in the 1970s by Xerox Corporation, the experimental version of Ethernet ran at 3 Mbit/s, but the first widespread standard of Ethernet ran at a speed of 10 Mbps in 1985 and later at 100 Mbps (sometimes referred to as *Fast Ethernet*) in 1995, at which point Ethernet had become the regular network system for most computers. The first Ethernet networks, 10BASE5, used thick yellow cable with vampire taps as a medium. Later versions of Ethernet (10BASE2) used thinner coaxial cable with BNC connectors as the connection medium. Currently Ethernet has many varieties that vary both in speed and physical medium used. The most common forms used currently are 10BASE-T, 100BASE-TX and 1000BASE-T. All three utilise twisted pair cables and 8P8C modular connectors, more commonly known as RJ45 (Registered Jack 45) connectors.[10] These forms run at 10 Mbit/s, 100 Mbit/s and 1 Gbit/s speeds respectively.[11]

The RJ45 medium is made from copper cabling, which is suitable for 10BASE-T and 100BASE-TX but can sometimes cause problems with the higher 1000BASE-T form of Ethernet. Due to the significant increase in speed and bandwidth requirements, 1000BASE-T is less tolerable of imperfections in the network cabling than previous standards and electrical noise can potentially degrade a Gigabit connection

severely when used with poor-quality or inappropriately specified copper cabling. Most modern Ethernet cabling can support 1000BASE-T satisfactorily, but signal degradation becomes more of a problem the longer the cable becomes. Fibre optic variants of Ethernet are commonly seen connecting buildings or network cabinets in different parts of a building but are rarely seen connected to end systems for cost reasons. Their advantages lie in performance, electrical isolation and distance, up to tens of kilometres with some versions. Fibre cabling is therefore a lot more desirable when dealing with super-fast Ethernet connections such as 1000BASE-SX in a large environment, but is not required in most small networks due to the quality of regular copper cabling.[12]

1.3.2 Gigabit Ethernet

Gigabit Ethernet is a form of the Ethernet standard which allows for high-speed transfers up to one Gigabit per second. The standard was approved by the IEEE in 1998 and later adopted by ISO. The initial standard for Gigabit Ethernet was known as IEEE802.3z, however the most commonly implemented form of Gigabit Ethernet (IEEE 802.3ab) was ratified a year later by the IEEE and uses unshielded twisted pair cabling as opposed to fibre cabling in the initial standard. The reason for the latter standard being more useful is because it allows existing copper cabling infrastructure, used for 10/100 MBit Ethernet, to remain in place without having to be replaced by fibre optic.[13] The fibre version of Gigabit Ethernet is known as 1000BASE-SX and can transmit along a single fibre line at a distance of 500m or more with modern cabling before requiring an endpoint. The unshielded twisted pair variant, 1000BASE-T, generally has a maximum length of 100m. The medium chosen however does not affect the operation of the network layer and is up to the

requirements of the environment as to which medium to choose. The same core can be used for either.

Gigabit is the logical successor for 10/100 MBit connections found in virtually all NICs (Network Interface Card) and has stood as a standard for some time and support has become very common, with most motherboards with integrated Ethernet supporting Gigabit, as well as new cards generally supporting it as well. The increase in speed is not only due to the higher clocking speeds (125MHz as opposed to 25MHz in 100 MBit), but also double the transmission bits (8 bits instead of 4). This results in a potential 10 times increase in available bandwidth, which makes it useful for high-speed transfers to and from an FPGA. Furthermore, since Gigabit Ethernet is a common standard, it is trivial to find hardware which can support this standard at a reasonable cost without having to resort to other, more exotic forms of data transfer between an FPGA and a PC.