VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Network switching fabric for Gigabit links

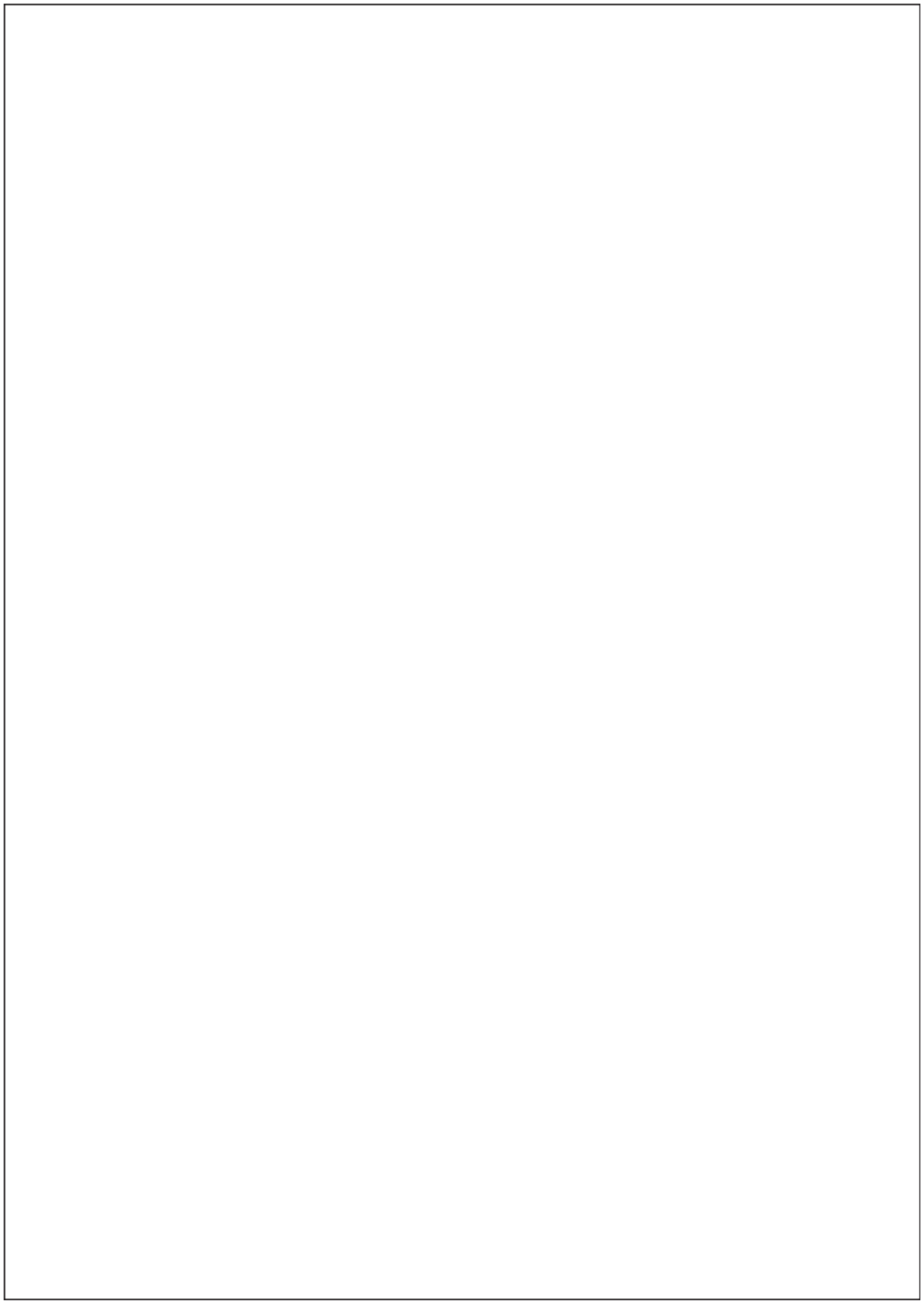
Mayur Panchal

Supervisors: Bryan Ng, Robin Dykstra

Submitted in partial fulfilment of the requirements for
Bachelor of Engineering with Honours in Electronic and
Computer Systems Engineering.

Summary of Comments on October Update.pdf

This page contains no comments



This page contains no comments

Contents

This page contains no comments

This page contains no comments

Figures

This page contains no comments

This page contains no comments

Chapter 1

Introduction

1 Fast internet connections are becoming more and more desired as high bandwidth media consumption and internet based services grow in popularity. Speed of an internet connection can be separated into two distinct metrics, latency and bandwidth. Latency is the time taken for information to travel from one place to another and the delay introduced by the device itself, while bandwidth is the amount of information that can flow reliably from end to end. When a user is accessing a website, such as Facebook, their requests traverse multiple network devices (physical hardware) to reach that ~~websites~~ 2 rvers. Throughout this travel, each network device introduces some latency, as there is processing time required to send information to the right destination. Although this delay is very small, over the course of multiple hops it can become significant.

1.1 Problem

There has been research dedicated to reducing latency. Some ways to reduce this latency include making smart network routers [1] and low latency wireless systems [2]. This research is reaching the point where network traffic can have latency as little as 1ms [3] or below. To advance further, new tools need to have the capability of measuring this latency.

3 xisting hardware solutions to measuring latency are capable of measuring less than 1 ms. These require some additional training in the software suite provided with the hardware [4], or existing systems to extend capabilites to. This drives the cost of using the device up, in terms of both time and money as learning the software suite takes time too.

Current software solutions can measure latency in ethernet network traffic down to a resolution of milliseconds and sometimes microseconds [5]. There are no software solutions that can measure latency in the order of nanoseconds reliably [6]. Measuring the latency in nanoseconds is needed as network switches have very low latency and the time between ingress and egress can range from microseconds to nanoseconds. If it could be measured accurately, then analysis could be performed on network traffic through network switches to research new ways to reduce latency.

4 xisting methods to measure latency within microseconds and nanoseconds require monolithic devices which need setting up and installing device drivers, proprietary soft-

Page: 7

Number: 1 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:32:29 a.m.
Case is that fast internet connections are good, and latency introduced by network devices is a/the *key* factor in this.

But nothing about measurement is mentioned. As this is what you're doing you should be mentioning it in some form in your introductory sections.

"Fast connections are becoming desirable ... by measuring we can ... "

Number: 2 Author: iddy Subject: Replacement Text Date: 5/10/2017 10:14:32 a.m.
website's

Number: 3 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:20:56 a.m.
needs to be tidied up.

main point: cost of learning a hardware/software suite

Number: 4 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:22:28 a.m.

ware, or even a replacement of existing hardware systems. Once setup, these devices often require data conversion of results to be compatible with data analysis methods and evaluation programs. Occasionally these programs are embedded within in manufacturers software, but the user is limited to what the manufacturer has chosen as relevant outputs for the user. These can be useful most of the time, but occassionally the user would need to export the data to another data analysis tool they are familiar with for futher processing. This inflexibility is prominent in many high end network performance analysis tools, causing delays in retrieving test results for evaluation.

A reduction in network latency would be a huge boon to future technologies. For example, less latency in video streaming can enable doctors from remote locations to perform surgery with minimal lag between actions and responses [7]. The stock market is another example of improvement with this technology, as less latency ensures that stock market trade deals can be completed faster.

> "We are running through the United States with dynamite and rock saws, so that an algorithm can run three microseconds faster." — *Kevin Slavin, How Algorithms shape our world [8]*

Many improvements made at a nanosecond scale ~~can~~ add up to microseconds or even milliseconds of latency reduction. Improvements cannot be done without knowing how much this can improve by without the tools to measure it.

A software based approach has the limitation that the timing functions must be processed by a Central Processing Unit (CPU) [9]. This means the counters for timing functions could be offset from the true value. Higher resolution latency measurements can be achieved through a hardware implementation of a packet timer. This circumvents the limitations of a software based approach.

1.2 Solution

Latency would be best measured by considering the raw electronic signals from the ethernet port on the device. To achieve this, microcontrollers, discrete logic gates and a Field Programmable Gate Array could be implemented. A microcontroller could be used but then the same issues arise, that a CPU clocking through instructions would deviate the true timing counter value. Another approach would be to use discrete logic gates to measure the timing accurately, but this will become complicated very rapidly, and propagation delays through discrete devices would reduce the accuracy. Hence the preferred choice is to implement the packet latency measurement unit on a Field Programmable Gate Array (FPGA). More information about FPGAs is in the Background section of the report. This way complex digital designs can be simplified down to a simple set of blocks, and there is no overhead that a processor would introduce. The drawback to using an FPGA is that the learning curve to developing on the platform is quite steep, and also communication with existing systems is difficult and is usually aided by the use of a microcontroller (such as an ARM or x86 based co-processor)

This report will focus on the work done so far in implementing a FPGA based packet

Page: 8

Number: 1 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:27:45 a.m.
There are some good points in here. It seems more aligned with the product you are creating too.

- Explains the current resolution -- micro-nano seconds.
- Limitation on output formats for hardware

Number: 2 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:34:14 a.m.
The idea of measuring is very important to your work. This should be mentioned earlier.

Number: 3 Author: iddy Subject: Replacement Text Date: 5/10/2017 10:28:38 a.m.
can

Number: 4 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:36:04 a.m.
is there something you can cite to support this logic?

latency timing device. This project is limited to ~~ethernet~~ [1] based networks as they are most widely used [10].

Page: 9

Number: 1 Author: iddy Subject: Replacement Text Date: 5/10/2017 10:38:08 a.m.
Ethernet (proper noun, uppercase E)

This page contains no comments

Chapter 2

Background and Related Work

2.1 General Networking

Information which is sent across a network is generally divided into smaller pieces. When each piece traverses a network, extra information is repeatedly added and removed until it has arrived at the destination. This extra information is split into multiple layers and each layer has a different function. The Open Systems Interconnection (OSI) model (Figure: 2.1) is a widely implemented standard that defines these different layers of extra information and their purpose.

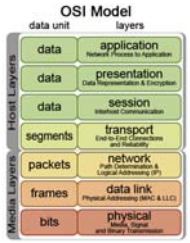


Figure 2.1: OSI Model [11]

Network switches and network routers interpret the destination based off information found in Layer 2 or Layer 3 from the OSI model, hence I must adhere to the protocol standards found at these layers. Layers 4 and above provide other benefits to the payload such as reliability and [1] is not a concern for this project.

Page: 11

Number: 1 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:45:46 a.m.
why are they not a concern? explain this

2.1.1 Layer 1

In Ethernet systems, Layer 1 refers to the physical medium that information is being sent over. Gigabit ethernet based systems use a copper wire as the physical medium, with electrical voltages representing the information. For this project, the latency is measured at this layer.

2.1.2 Layer 2

Layer 2 is responsible for transferring data between adjacent network devices [12]. This is important as information will be flowing to and from adjacent nodes, through a switch or router. To ensure that information flows through the switch or router, this layer must be taken into consideration to ensure successful transmission and reception of information. This layer is important to the project as the measuring device will be timing the latency between adjacent network devices and through network switches, both of which rely on Layer 2 information.[1]

2.1.3 Layer 3

Information from one hop to another is determined by the Layer 3 protocol. This ensures that information is correctly routed to the destination from the source. This layer is important as the information stored at this layer ensures the correct path is taken from sender to receiver. This is different to Layer 2, as this can incorporate non-adjacent nodes as well.

2.2 Latency Definitions

For this project, latency in a network will be defined in the following ways. This project will be focusing on One Way Trip time, because it is a subset of the other latencies.

2.2.1 Connection Time

Connection time describes the time that two devices take to enable information flow between them. In some cases, there are many synchronisation and authentication steps that need to take place before any information can flow. This latency is defined as the initialization of the first command, to the processing of the final command.[2]

2.2.2 [3]Return Trip Time

In some cases, the return trip time can be used to measure [4]the one-way latency of two devices. One device sends a request for an echo and when the echo is recieved, the time taken from transmition to reception is twice the latency between devices.

Page: 12

Number: 1 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:44:52 a.m.
in what way? this could be clearer

Number: 2 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:51:44 a.m.
It isn't clear what you mean by command. Is this the opening of a UDP socket? Completion of a TCP handshake? ARP and similar lower layers do not carry a concept of connection, does this apply to them?

Number: 3 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:52:15 a.m.
Do you mean round-trip time (RTT)?

Number: 4 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:52:58 a.m.
... the average* one way ...

2.2.3 One way Trip Time

This is similar to Return Trip Time but instead of requiring an echo back to the sender, the receiver is on the same device. Hence the latency of the connection is measured as the time taken for the information to flow from one port on the device to another.

2.3 Current Solutions

There are solutions present which can measure latency to a high degree (¡ 1 ms) but have some drawbacks in different areas. These can be split into two different catagories, hardware and software, referring to the process of which they measure latency.

2.3.1 Software

Ping

Ping is a Linux utility that can be used to estimate the latency to a given network server. This is for large distance measurements, as the accuracy of this ranges from seconds to milliseconds. It has also been tested that this utility is unreliable for performance extensive testing as momentary glitches can appear in networks, causing random and unpredictable results [5]. It is very useful at measuring very large latencies (¿1 milliseconds) and displaying them in a concise format for analysis by the user. This does not meet the need of having the ability to measure time in the nanosecond range, but a useful takeaway is to make sure that values are presented in an easy to analyse format (printing to screen, or to a file).

Data plane Development Kit (DPDK)

DPDK is a software implementation of rapid packet processing. This software utilises low level software drivers to interact directly with the hardware. Doing so requires specific hardware on the computer which needs to be compatible with the software itself. Reducing the packet processing time reduces the time offset created by the CPU, but not fully removes it. Examples in the source code have shown that the timing value is dependent on CPU frequencies [13] and Dynamic Frequency scaling [14] of modern CPUs can cause a change in this frequency at any time. An advantage of this solution is that it can produce a high-resolution time value (in the order of nanoseconds) and is easily accessible by a user in software for processing and analysing. A lesson learnt from this approach is that a CPU based system for measuring high resolution timers will not be precise or accurate.

PF_RING

Another software solution is PF_RING by ntop™. It is a rapid packet processing library that does not require the need for specific hardware. PF_RING allows for more efficient packet capturing and filtering by utilising more cores and threads on a CPU [15]. This library is

Page: 13

Number: 1 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:55:18 a.m.
is this something you have defined for the purposes of your work or a regularly used term? or are you trying to describe a regular term but define how it will/can be measured?

Number: 2 Author: iddy Subject: Comment on Text Date: 5/10/2017 10:54:19 a.m.
... one interface* port ...

Number: 3 Author: iddy Subject: Replacement Text Date: 5/10/2017 11:26:10 a.m.
by

more focused on throughput of packets processed rather than latency. This is less accurate than the DPDK but increases the flexibility of the platform it can implemented on. As with all software based approaches, this does not meet the requirements of being able to measure time in nanoseconds reliably. An insight from this approach is to flexibility of platform is good for users but will not be a requirement for this project.

2.3.2 Hardware

Data Acquisition and Generation (DAG) 10x2-S

The DAG 10x2-S by Endace is a hardware based packet capturing solution with high precision nanosecond scale precision [16]. This is a physical device which connects to a x86 based computer and communicates via Peripheral Component Interconnect Express (PCIe). The DAG 10x2-S is recommended for capturing network packets over gigabit ethernet links, as the other models cost more and have other unnecessary features [4]. It is an expansion card for a computer which extends the capabilities of the computer to accurately capture and timestamp packets to a high degree. This device has the capabilities to timestamp packets with a resolution of 4ns [16]. This meets all the requirements of the project but is very costly ($2500 USD [17]) and methods for obtaining a device can only be done through Endace themselves.

Field Programmable Gate Array (FPGA)

A FPGA is an array of configurable logic gates. The number of gates are in the order of thousands to millions. This many configurable gates allows for complex logical structures and digital circuits to be implemented in hardware, while consuming little physical space. This differs from a CPU, where the logical gate circuitry is fixed, and manipulation of electrical Input/output must be done by clocking through an instruction set. Without the need for an instruction set, FPGAs allow for high speed time critical applications to be implemented without the overhead of needing to clock through CPU instructions. This is an important characteristic as the electrical signals measured from the network interfaces are time critical.

NetFPGA-SUME Virtex-7 FPGA Development Board

The NetFPGA-SUME is a FPGA development board for high density and high-performance networking design. This incorporates a FPGA to process packets rapidly in hardware, much faster than any software. The Virtex-7 FPGA onboard is a recently released FPGA which can process up to 13.1 gigabits per second worth of information through its transceivers while the scope of this project is limited to 1 gigabit per second ethernet. This is also a very costly development board, costing $4999 [18] for academic customers. Due to the limited scope of this project and the cost involved with purchasing this development board, this is not a suitable platform for developing latency measuring device.

This page contains no comments

Xilinx Zynq FPGA

Xilinx is a manufacturer of FPGAs, and a family of products they produce is the Zynq-7000 range [19]. These FPGAs integrate a dual-core ARM Cortex-A9 MPCore with FPGA gate fabric enabling high performance applications to run on the FPGA, while embedded programs can run on the ARM cores. This project incorporates the FPGA section to manage the timing functions and the ARM section to run the application for storing the timing information.

2.4 Related Work

2.4.1 Moongen

MoonGen is a flexible high-speed network packet generator. The goal was to saturate 10 Gb Ethernet links using a flexible hardware platform. A key issue defined in the paper was that Existing software solutions lack performance or flexibility when precision is desired. Existing hardware solutions are expensive with inflexible software accompanying the platform. A key takeaway is that existing platforms are inflexible and lack precision.

2.4.2 Flexible High Performance Traffic Generation on Commodity MultiCore Platforms

A similar problem is solved with this paper. The goal was to saturate a 10 Gb Ethernet link using software methods. Another mention of inflexible hardware solutions was present in this paper, alluding to the fact that existing solutions that are based on hardware are lacking in flexiblity. This may seem to say that a hardware based solution may not be the correct approach to measuring latency, but this shows the novelty in a solution that is flexible and hardware based.

2.4.3 A Distributed Instrument for Performance Analysis of Real-Time Ethernet Networks

2.5 Hardware Implementation

2.5.1 Differential Signaling

Differential Signalling is a method of electrical communication using opposing electrical voltages on two wires. This is advantageous in transmission of electrical signals in electrically noisy environments as no reference point is used to infer information. If a reference point is used in an electrically noisy environment, the noise can couple to the reference point, causing errors in the information transferred. Differential signalling removes this dependency on a reference point by transferring information through the difference between the two transmission lines. By twisting the transmission lines together, external electrical

This page contains no comments

signals are coupled to both lines, and do not affect the difference between the two lines. This way information is transferred over a large distance reliably and at high speed. In Gigabit Ethernet, information is transferred through 4 pairs of twisted differential paired cables.

2.5.2 Pulse Amplitude Modulation

Pulse Amplitude modulation is a form of modulating a signal with information encoded in discrete levels of pulses. Different versions of PAM have varying levels of discrete steps. In Gigabit Ethernet, PAM-5 is used with 5 different voltage levels each corresponding to different set of predefined binary codes interpreted by the decoder.

2.5.3 IEEE 802.3ab

IEEE 802.3ab is the defining standard for gigabit ethernet communication across a copper link. It defines the use of 4 pairs of copper and the maximum length (100 m) the protocol is designed for. Ethernet Physical Transceiver (Ethernet PHY) An Ethernet PHY is an electronic component used to convert signalling from a link layer device (Such as a MAC) to the physical layer (differential signalling with PAM). Common interfaces to communicate from a MAC to a PHY include GMII, RGMII, XGMII.

2.5.4 Gigabit Media Independent Interface (GMII)

(GMII IMAGE HERE)

GMII is a protocol used in this project to communicate between the Ethernet PHY and the MAC which is implemented on the FPGA. The ethernet PHY decodes the data sent from the GMII interface, and produces the correct signalling required for the receiving Etherenet PHY. The process of clocking out data requires 8 data line connections, and a few extra lines for timing/scheduling.

2.5.5 Reduced GMII (RGMII)

(RGMII Signals HERE)

RGMII is an implemenetation of GMII, but a key difference is a reduction in the number of transmit lines. This is possible through the use of clocking all signals on a bus consisting of Dual-Data-Rate (DDR) lines. This is done through clocking the data in and out on both the rsising edge and the falling edge of the clock. This helps reduce the number of physical wires used to interconnect components on the PCB, making more space for routing other complex busses (Such as DDR3 for the RAM).

This page contains no comments

2.5.6 Advanced Microcontroller Bus Architecture

2.5.7 Advanced eXtensible Interface

2.5.8 Serial Peripheral Interface

This page contains no comments

This page contains no comments

Chapter 3

Design

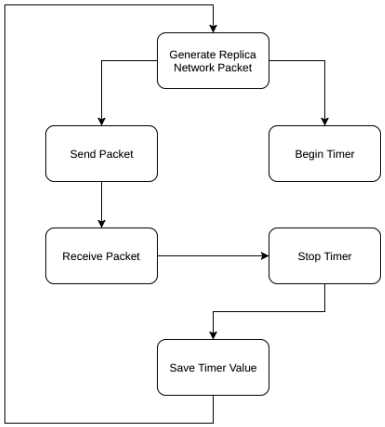


Figure 3.1: Design of Measurement Sequence

This page contains no comments

This page contains no comments

Chapter 4

Implementation

Initially a system block must generate a network packet that can be used to measure the latency. On a x86 based computer, this is done through the CPU, and an element called a MAC. The MAC is used to convert logic link control signals into physical hardware signals for the Ethernet Phy to interpret. An Ethernet Phy is required as the 802.3ab standard defines the electrical connection from one node to another to consist of 4 twisted pairs and the differential electrical signals required for this cable is not possible in a FPGA, or in a CPU, hence this is offloaded to an external IC. Connections consist of commonly found Category 5e and 6 cabling, which adheres to the tolerance for how many twists are performed per meter cable (Important for signal integrity). The connection between the MAC and the Ethernet PHY is performed through a hardware protocol standard called Media Independent Interface (MII) and then converted to the differential signals across the Category 5e/6 cable. The gigabit version of ethernet uses an extended version of MII called Reduced Gigabit Media Independent Interface (RGMII).

This page contains no comments

4.1 Block Design (Concept)

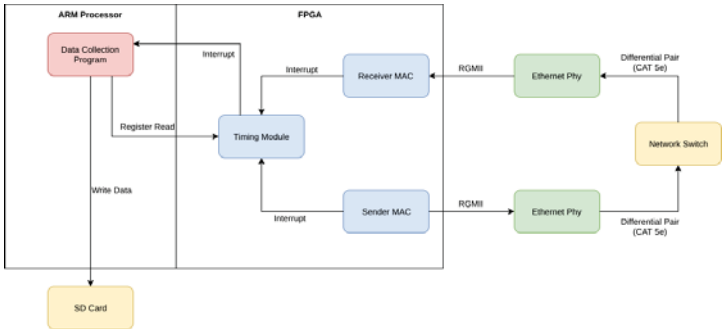


Figure 4.1: Generic Block Design Concept

Figure above shows the conceptual structure of the design. Blue represent FPGA logic blocks, Red represents Bare-Metal programs run on the ARM Processor, Green represents external ICs and Yellow represent other hardware external to the FPGA development board.

The Sender MAC, the FPGA produces a repeated signal of a network packets and an interrupt to a timing module to begin counting. The number of packets per second (PPS) is controlled by hardware switches found on the development board itself. The Ethernet Phy converts these RGMII signals into differential signals what are used in communicating over CAT 5e/6 cable. This is then switched to the other Ethernet Phy, and the receiving end detects the incoming packet. The Receiver MAC detects the network packet, and produces an interrupt for the timing module to stop counting.

The Timing Module captures the events produced by the MACs and once a value is stored in the register, it produces an interrupt for the ARM Processor to begin reading the register.

A program running on the ARM Processor intercepts the interrupt, and an interrupt service routine begins. This retrieves the value in the register and saves it to the SD card. Every time the interrupt is triggered, the value is appended to the end of a file stored on the SD Card.

Page: 22

Number: 1 Author: iddy Subject: Comment on Text Date: 5/10/2017 11:35:49 a.m.
implies this should be in the design section

4.2 Block Design (Implementation)

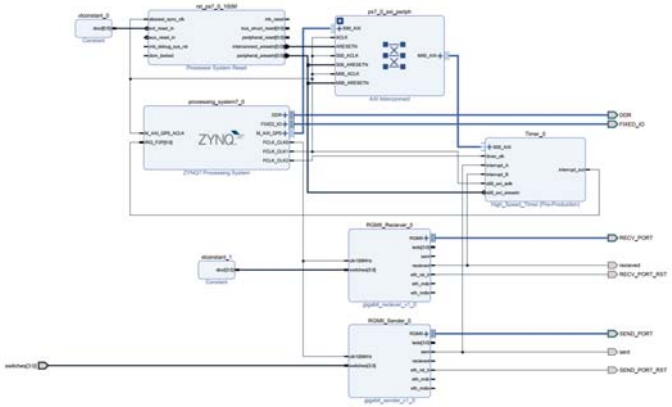


Figure 4.2: FPGA Block Design Implementation

Figure 4.2 above shows the overall diagram of the FPGA logic blocks. This is the view through the IP integrator in Vivado 2017.2. Thicker lines in the diagram represent busses, while the thinner black lines represent single wires. Connections on the left and right of the diagram represent Inputs and Outputs to the FPGA respectively.

The extra blocks in the upper area of the diagram represent constructs needed for AXI peripherals. This allows the PS to interact and access registers found in the PL of the FPGA. AXI manages the control signalling for buffering and clocking data in and out of the PS.

4.3 Reduced Gigabit Media Independent Interface (RGMII)

To provide a better understanding of RGMII, Gigabit Media Independent Interface (GMII) must be understood.

This page contains no comments

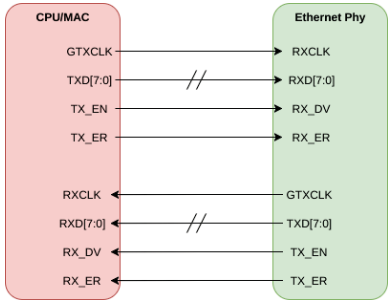


Figure 4.3: GMII Physical Wiring

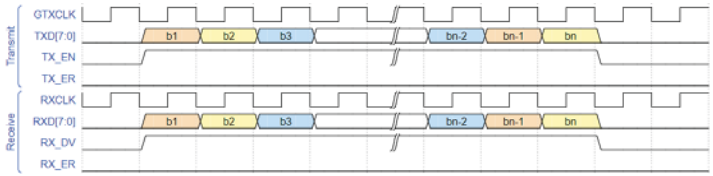


Figure 4.4: GMII Signal Timings

GMII is a full duplex interface between the MAC and an Ethernet Phy. As shown in Figure 2, the physical wiring consists of 22 wires between the two devices. The signalling shown in Figure 3 gives an example of how data is transferred to and from the Phy. Bytes transmitted to the Ethernet Phy are then converted and sent across the Cat5e/6 cable. RGMII is a Dual Data Rate Extension to GMII, by which the data is clocked on both the rising and falling edge of the clock. This means that one byte can be transmitted over four wires instead of 8, while keeping the same clock frequency. This is advantageous as there is commonly limited space on a PCB for wiring interconnects together, especially when there are more than 400 pins on a device (The XC7Z020-CLG484 found on the Zedboard, has 484 package pins).

This page contains no comments

Chapter 5

Evaluation

5.1 Accuracy Testing

To test the accuracy of the device, a loopback test was performed with different length of cables. This was setup so that there was no switch between two ethernet measurement ports, only an ethernet cable. The test is to ensure that the device is measuring the latency correctly, and able to detect the latency present in a specific length of cable. Latency measured from the device can be compared with the theoretical value of the latency which can be calculated using the following formula:

$$Latency = \frac{L}{cV_f}$$

Where L is the length of the cable, c is the speed of light (299,792,458 m/s) and V_f is velocity factor of ethernet cables (0.65).

A very short cable (4 cm) was used to measure the processing time, a constant added to every measurement. This processing time is the result of both Ethernet Phys converting between the two protocols (Differential Pairs and RGMII). Performing measurements with the short cable produced a mean value of 440 ns where 0.2095 ns of the measurement is attributed to the delay present in the cable.

This processing time is then removed from subsequent tests, making the latency values produced by the device purely the latency present in the cable. The goal from the tests is to ensure that the device is accurate to within 4 ns, which is the smallest interval of time that can be measured.

Tests were done with cables spanning 2 m, 3 m and 25 m long. These provided enough of a delay to measure with the device, and enough data to characterise the accuracy of what cables are used. The measured value was plotted with the theoretical values of what delay would be present in the cable. To obtain the delay measurement, the test was run for 20 seconds on the device, with a packet transmission frequency of 500 packets per second. The mean value of all packet latency measurements for that cable were then plotted versus the length of the cable.

This page contains no comments

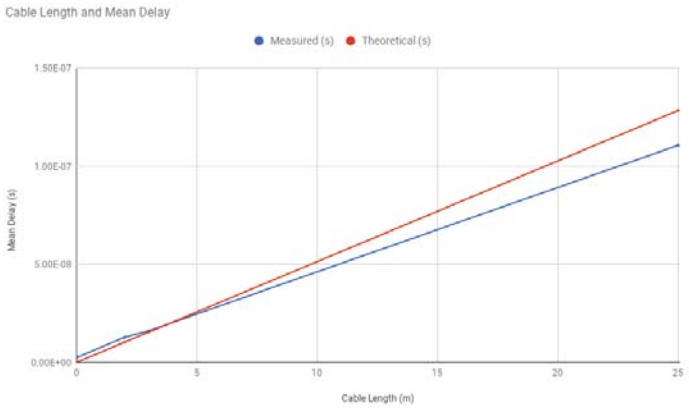


Figure 5.1: Results from Accuracy Measurements

As shown in 5.1, the results from the device show a linear trend in latency due to length of the cable. The gradient difference of the theoretical line can be attributed to the fluctuation in velocity factor through the different cables.

The difference between the theoretical and measured value of the 25 m cable was 14 ns. This is not within the accuracy margin which was required, hence 2 m and 3 m cables are used for the reliablity tests as the difference was below 2 ns.

5.2 Reliability Testing

To test reliability of the device, repeated tests were performed to show that results would stay consistent from test to test. As the goal of the project was to measure the latency in network swtiches, these tests were to measure the latency through a network switch and provide repeatable results. These results did not neccessarily have to be accurate, as the accuracy was determined in the test in accuracy testing, hence these results are only to measure the reliability.

Results from the tests were compared to an existing DPDK software based latency measurement method. This was run on two separate machines which had clocks synchronoised using PTP protocol and as a result, the timings varied significantly. This comparision was to show that the software based methods have flexibility that hardware methods are missing while also performing unreliably. The DPDK based system was compared to the FPGA based device by producing a packet on one machine, then sending the packet through the network switch to arrive at the other machine, and the time delta is returned from both ma-

This page contains no comments

chines. The FPGA device also performed the same task with the same cables, but the two connections arrived at the same machine.

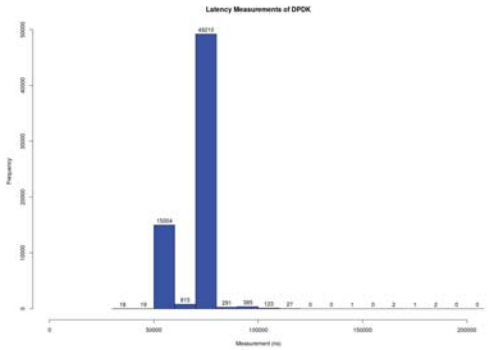


Figure 5.2: Results from DPDK Reliability Measurements

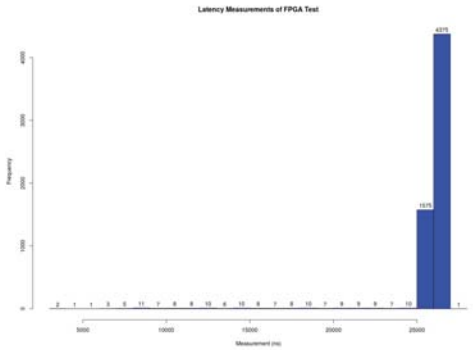


Figure 5.3: Results from FPGA device Reliability Measurements

As seen in the results, the FPGA device produced a value lower than the mean value. This is very interesting, as it is an unexpected result. Lower values produced by the device could be attributed to the device misinterpreting the reception of a previous test as the end of the current test. Regardless of this result, the device produced over 10 times lower standard deviation compared with the results from the DPDK test setup. This can be concluded as a successful test.

This page contains no comments

This page contains no comments

Bibliography

[1] "Smarter routers reduce latency in the internet," 2017. Available at https://phys.org/news/2017-02-smarter-routers-latency-internet.html.

[2] "Network latency and download speeds set to improve with 5g," 2016. Available at https://insights.samsung.com/2016/09/08/network-latency-and-download-speeds-set-to-improve-with-5g/.

[3] "5g's moment is now, four years before it's scheduled to arrive," 2016. Available at http://www.pcworld.com/article/3037452/mobile/this-is-5gs-moment-four-years-early.html.

[4] "Introducing endace dag." Available at https://www.endace.com/introducing-dag.pdf.

[5] InetDaemon, "What ping is not," 2012. Available at http://www.inetdaemon.com/tutorials/troubleshooting/tools/ping/ping_is_not.shtml.

[6] A. Thankachan, "High performance time measurement in linux," 2010. Available at https://aufather.wordpress.com/2010/09/08/high-performance-time-measuremen-in-linux/.

[7] R. Smith, "Research in lag time set to determine the future of telesurgery," 2015. Available at https://techcrunch.com/2015/06/14/lag-time-research-set-to-determine-the-future-of-telesurgery/.

[8] K. Slavin, "How algorithms shape our world," 2011. Available at https://www.ted.com/talks/kevin_slavin_how_algorithms_shape_our_world.

[9] "Game timing and multicore processors." Available at https://msdn.microsoft.com/en-us/library/windows/desktop/ee417693(v=vs.85).aspx.

[10] "Why is ethernet the most popular form of wired lan technology today?," 2014. Available at https://www.quora.com/Why-is-Ethernet-the-most-popular-form-of-wired-LAN-technology-today.

[11] "Osi model." Available at https://upload.wikimedia.org/wikipedia/commons/2/2b/Osi-model.png.

[12] IEEE, "Standard for local and metropolitan area networks: Overview and architecture," 2014. Available at http://standards.ieee.org/about/get/802/802.html.

[13] "Timer example c code," 2014. Available at http://dpdk.org/browse/dpdk/tree/examples/timer/main.c.

This page contains no comments

[14] "Intel turbo boost technology 2.0," 2016. Available at https://www.intel.com/content/www/us/en/architecture-and-technology/turbo-boost/turbo-boost-technology.html.

[15] "Pf_ring." Available at http://wiki.aanval.com/wiki/PF_RING.

[16] "Endace dag cards." Available at https://www.endace.com/endace-dag-high-speed-packet-capture-cards.html.

[17] "New low pricing on dag 10x2-s." Available at https://blog.endace.com/2016/07/08/new-low-pricing-dag-10x2/.

[18] "Netfpga-sume virtex-7 fpga development board." Available at http://store.digilentinc.com/netfpga-sume-virtex-7-fpga-development-board.

[19] Xilinx, "Zynq-7000 all programmable soc data sheet: Overview," 2017. Available at https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf.

This page contains no comments