VERILOG MODELLING OF MODBUS TCP AT 100 MBPS

TAN ZHE JIE

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer and Microelectronic System)

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

FEBRUARY 2022

DEDICATION

This project report is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

In preparing this project report, I was in contact with many people. They have contributed towards my understanding and thoughts. I wish to express my sincere appreciation to my supervisor, Dr. Zulfakar bin Aspar, for encouragement, guidance, critics, and friendship. I am also very thankful to my presentation examiner Assoc. Prof. Ir. Dr. Muhammad Nadzir Bin Marsono and Dr. Ismahani Ismail for their advice during the presentation. Without their continued support and interest, this thesis would not have been the same as presented here. I am also indebted to Intel Technology Sdn Bhd for funding my Master study.

My fellow postgraduate student should also be recognised for their support. My sincere appreciation also extends to all my colleagues and others who have aided at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. Lastly, I am grateful to all my family member.

ABSTRACT

With the continuous development of industry automation, industrial control systems and programmable logic devices are being widely used in the manufacturing production. Machines are required to work either in connection to each other or remotely controlled at a centralized control room using Internet of Things (IoT), Supervisory Control and Data Acquisition Systems (SCADA) or other communication means. Among the many industrial networking protocols, Modbus TCP is widely adopted. Software implementation of Modbus TCP network is common in the industry. Although software does the job, it is a burden to the processor. There are also Modbus TCP hardware modules selling in the market. But dedicated hardware incurs high cost and not scalable for any feature change. Hence, this project aims to analyse and design a hardware Modbus TCP client and server communication node with the help of RTL-ASMChart and Petri Net. It will be implemented at 100Mbps Ethernet speed within the appropriate power, performance, and area. This design is coded in SystemVerilog and validation is done in Quartus ModelSim simulation. Running testbench in ModelSim and Wireshark show the design is function as expected, after it can be compiled and fit into the target Cyclone V FPGA. Timing closure and throughput expectation of 100Mbps is met in Quartus, with power consumption of around 350mW. Round trip test results showed that RTL designed TCP module has speed improvement over the software TCP method of Windows operating system.

ABSTRAK

Dengan perkembangan automasi industri yang berterusan, sistem kawalan industri dan peranti logik yang dapat diprogramkan digunakan secara meluas dalam pengeluaran pembuatan. Mesin diperlukan untuk berfungsi sama ada saling berhubungan atau dikawalkan dari jarak jauh di bilik kawalan terpusat menggunakan *Internet of Things (IoT)*, *Supervisory Control and Data Acquisition Systems (SCADA)* atau cara komunikasi lain. Antara banyak protokol rangkaian industri *Modbus TCP* banyak digunakan. Rangkaian *Modbus TCP* secara perisian adalah popular di industri. Walaupun perisian menjalankan tugas, ia menjadi beban kepada pemproses. Terdapat juga modul fizikal *Modbus TCP* yang dijual di pasaran. Tetapi harga modul fizikal agak mahal dan tidak boleh dinaik taraf untuk sebarang perubahan ciri. Oleh itu, projek ini bertujuan untuk menganalisis dan melaksanakan nod komunisasi klien dan pelayan *Modbus TCP* dengan bantuan *RTL-ASMChart* dan *Petri Net*. Ia akan dilaksanakan pada kelajuan *Ethernet* 100Mbps dalam kuasa, prestasi dan kawasan yang sesuai. Projek ini dikodkan dalam *SystemVerilog* dan pengesahan dilakukan dalam simulasi *Quartus ModelSim*. Menjalankan ujian dalam *ModelSim* dan *Wireshark* menunjukkan modul berfungsi seperti yang diharapkan, selepas ia boleh disusun dan dimuatkan ke dalam *FPGA Cyclone V* yang ditentukan. Penutupan masa dan jangkaan pemprosesan sebanyak 100Mbps dipenuhi di *Quartus*, dengan penggunaan kuasa sekitar 350mW. Keputusan ujian pergi dan balik menunjukkan bahawa modul TCP mempunyai peningkatan kelajuan berbanding dengan kaedah TCP perisian dalam *Windows operating system*.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ALM	-	Adaptive Logic Module
ALUT	-	Combinational Adaptive Look-up Table
API	-	Application Programming Interface
ARP	-	Address Resolution Protocol
ASCII	-	American Standard Code for Information Interchange
ASIC	-	Application Specific Integrated Circuit
CIDR	-	Classless Inter Domain Routing
CRC	-	Cyclic Redundancy Check
DCHP	-	Dynamic Host Configuration Protocol
DDRIO	-	Double Date Rate Input Output
DSCP	-	Differentiated Services Code Point
DUT	-	Device Under Test
FCS	-	Frame Check Sequence
FIFO	-	First In, First Out
FPGA	-	Field Programmable Gate Array
FSM	-	Finite State Machine
GMII	-	Gigabit Media-Independent Interface
GMT	-	Global Mediterranean Time
GPIO	-	General-Purpose-Input-Output
HDL	-	Hardware Description Language
HPS	-	Hard Processor System
I/O	-	Input and Output
I2C	-	Inter-Integrated Circuit protocol
ICMP	-	Internet Control Message Protocol
ID	-	Identification
IoT	-	Internet of Things
IP	-	Internet Protocol
IPG	-	Interpacket Gap
IPv4	-	Internet Protocol version 4
JTAG	-	Joint Test Action Group interface

LED	-	Light Emitting Diode
LUT	-	Lookup Table
LVDS	-	Low Voltage Differential Signalling
MAC	-	Media Access Control
MBAP	-	Modbus Application Protocol Header
Mbps	-	Mega bits per second
MDC	-	Management Interface Clock
MDIO	-	Management Data Input/Output
MLAB	-	Memory Logic Array Block
MSB	-	Most Significant Byte
OSI	-	Open System Interconnect
OSPF	-	Open Shortest Path First protocol
PC	-	Personal Computer
PCS	-	Physical Coding Sublayer
PDU	-	Protocol Data Unit
PHY	-	Physical Transceiver
PLC	-	Programmable Logic Controller
PLL	-	Phased Locked Loop
PMA	-	Physical Medium Attachment
PMD	-	Physical Medium Dependent
PPA	-	Power, Performance and Area
RAM	-	Random Access Memory
RFC	-	Request for Comment publication
RGMII	-	Reduced Gigabit Media-Independent Interface
RIP	-	Routing Information Protocol
ROM	-	Read-only Memory
RTL	-	Register Transfer Level
RTT	-	Round Trip Time
RTU	-	Remote Terminal Unit
RX	-	Receive path
SACK	-	Selective Acknowledgement
SCADA	-	Supervisory Control and Data Acquisition Systems
SCTP	-	Stream Control Transmission Protocol

Seq Ack	-	Sequence and Acknowledge Number
SFD	-	Start Frame Delimiter
SLR	-	Systematic Literature Review
SRAM	-	Static Random-Access Memory
TCP	-	Transmission Control Protocol
TCP/IP	-	A name that represents whole suite of Internet Protocol
TTL	-	Time to Live
TX	-	Transmit path
UDP	-	User Datagram Protocol
UTP	-	Unshielded Twisted Pair
VoIP	-	Voice over Internet Protocol

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Problem Background

With the continuous advancement of manufacturing automation, industrial control systems and physical devices like Programmable Logic Controllers (PLC) are widely used in factory production. They must issue real time control command to and read sensors data from the machinery, processing equipment and power distribution equipment in a factory without any hiccup for 24 hours in a day. Large scale control systems, also known as Supervisory Control and Data Acquisition (SCADA) systems are connected to a local computer network or the Internet and enable remote operation across the globe. Moreover, under the banner of Industry 4.0 more and more traditional industry players are heeding the call to attach their physical component to the digital infrastructure, connecting them to the Internet as part of Internet of Things (IoT).

Present day industrial control system has networking protocols that are pervasive, such as Profibus, Profinet, EtherCAT, CANopen and Modbus. Among many of these protocols, Modbus protocol is being widely used in many areas, for instance in intelligent building system, machinery, independent sensors, or actuators. Modbus is an open and royalty-free communication protocol for digital/analog I/O information. It is used for register data transfer between industrial control and monitoring devices such as PLC, Remote Terminal Unit (RTU), sensors, and actuators. Control devices from different manufacturers can be connected to a single industrial network for centralized monitoring using Modbus protocol. First published in 1979 by Modicon Inc and now governed by Modbus Org, Modbus was designed to work over serial lines such as TIA/EIA 232,422 or 485. Over the years it has become the common industry standards for almost half a century. Large part of the existing automation structures is using Modbus protocol, especially serial Modbus.

Serial Modbus has two variants, RTU and ASCII, which vary mainly in way of data encoding, but they feature the same data frame structure. Subsequently Modbus TCP was established to operate over standard Ethernet, TCP/IP-based networks. With that said, all three variants are still being used and served its purpose. Modbus RTU over serial lines commonly deployed for Remote Terminal Units (RTU) and PLC, while Modbus TCP is generally used between and servers with SCADA features.

1.2 Problem Statement

Software implementation of Modbus TCP network is common in the industry. Many resources such as library or firmware for the protocol are readily available and widely supported in many operating system. Although software does the job, it is a burden to the processor. The operating system needs to dedicate some threads to maintain the connection and facilitate data transfer. It is especially the case for Industrial Ethernet when the sensor is frequently transferring data to the host PC in real time. This will take up precious processing resources which otherwise can be utilized for much important tasks. Besides, processor speed may also be a limiting factor of network performance.

On the other end of spectrum, there is hardware module for Modbus TCP connection selling in the market. It may come in the form of ASIC semiconductor chip or dedicated gateway module that slots into industrial server rack. But dedicated hardware incurs high cost. Moreover, they are not flexible and scalable for any feature change if it is required in the future. The supported functionality is determined by the hardware manufacturer. Any change in feature or additional capacity often require purchasing a separate module.

Implementing Modbus TCP on a FPGA can be a good option in terms of flexibility for feature change without compromising network performance. Some portion of software TCP/IP protocol can also be hardened into hardware to speed up certain repetitive tasks. Price-wise FPGA is placed between the lower cost of software approach versus dedicated hardware which in general much more expensive.

1.3 Research Objectives

The objectives of the research are:

- To analyze and design a hardware Modbus TCP client and server communication node
- To implement the Modbus TCP module at 100Mbps Ethernet speed requirement within the appropriate power, performance, and area (PPA)
- To design, simulate and test the design using SystemVerilog hardware description language

1.4 Scope

Out of the various Industrial Ethernet protocol that is common today (PROFINET, EtherCAT, CANopen, etc..) this project is designed for Modbus protocol only, specifically Modbus TCP variant. Other protocol will not be supported in the design. In addition to that, the design will be a dedicated communication node focusing on data transfer between the client and server on a network. Additional functionality such as telemetry and data analytics are not part of this design.

This project is based on Terasic DE0-Nano-SOC Board, which carries an Intel Cyclone V FPGA. This board also equips with Ethernet networking capability, with a RJ45 connector and a Gigabit Ethernet Physical Transceiver (PHY) readily available. The PHY chip can support 10/100/1000 Mbps. Design validation is expected to be done through testbench simulation in Quartus Prime environment, and if possible, physical prototyping of the network through Ethernet cable interface to a computer.

REFERENCES

[1] M. Khalil-Hani, *Advanced Digital System Design,* Universiti Teknologi Malaysia, 2019.

[2] C. A. Petri and W. Reisig, "Petri net," Scholarpedia, 2008. [Online]. Available: http://www.scholarpedia.org/article/Petri_net. [Accessed 2008 2021].

[3] Intel*, FPGA Architecture White Paper v1.0,* 2006.

[4] Acromag Inc, *Technical Reference – Modbus TCP/IP*, 2005.

[5] Modbus Org*, Modbus Application Protocol V1.1b3,* 2012.

[6] Defense Advanced Research Projects Agency*, RFC 2131 - Dynamic Host Configuration Protocol*, 1997.

[7] Defense Advanced Research Projects Agency*, RFC793 - Transmission Control Ptotocol Specification,* 1981.

[8] Defense Advanced Research Projects Agency*, RFC 768 - User Datagram Protocol,* 1980.

[9] Defense Advanced Research Projects Agency*, RFC791 - Internet Protocol Specification,* 1981.

[10] Defense Advanced Research Projects Agency*, RFC792 - Internet Control Message Protocol Specification,* 1981.

[11] IEEE*, IEEE 802.3-2018 - IEEE Standard for Ethernet,* IEEE, 2018.

[12] Defense Advanced Research Projects Agency*, RFC826 - Address Resolution Protocol Specification*, 1982.

[13] Verification Guide, "Systemverilog Queue - Verification Guide," [Online]. Available: https://verificationguide.com/systemverilog/systemverilog-queue/. [Accessed January 2022].

[14] Wireshark, "LibpcapFileFormat," [Online]. Available: https://wiki.wireshark.org/Development/LibpcapFileFormat.

[15] J. N. Chhatrawala, N. Jasani and V. Tilva, "FPGA based data acquisition with Modbus protocol," in *International Conference on Communication and Signal Processing (ICCSP)*, 2016.

[16] B. Li, G. Chen, L. Wang and Z. Hao, "Tower Crane Remote Wireless Monitoring System Based on Modbus/Tcp Protocol," in *IEEE International Conference on Computational Science and Engineering (CSE)* , 2017.

[17] F. Yncio, R. Peña, A. Cadiboni, R. Fernández, G. Ahrtz and C. S. Tellechea, "A Modbus client for the identification of an energy recovery system for a water distribution network," in *IEEE Power, Instrumentation and Measurement Meeting (EPIM)*, 2018.

[18] G. Meza, C. d. Carpio, N. Vinces and M. Klusmann, "Control of a three-axis CNC machine using PLC S7 1200 with the Mach3 software adapted to a Modbus TCP/IP network," in *IEEE International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2018.

[19] Q. Bai, B. Jin, D. Wang, Y. Wang and X. Liu, "Compact Modbus TCP/IP protocol for data acquisition systems based on limited hardware resources," *IOPScience Journal of Instrumentation,* vol. 13, 2018.

[20] W. You and H. Ge, "Design and Implementation of Modbus Protocol for Intelligent Building Security," in *IEEE International Conference on Communication Technology (ICCT)*, 2019.

[21] S. Mangkalajan, W. Koodtalang, T. Sangsuwan and N. Pudchuen, "Virtual Process Using LabVIEW in Combination with Modbus TCP for Fieldbus Control System," in *IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2019.

[22] L. Chen and Z. Su, "Realization of Modern Tram Data Acquisition System Based on Labview and Modbus TCP," in *International Conference on Safety Produce Informatization (IICSPI)*, 2019.

[23] Y. He and X. Lv, "The Application of Modbus TCP in Universal Testing Machine," in *IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2021.

[24] Anybus by HMS Networks, "Modbus TCP connectivity solutions with Anybus," [Online]. Available:

https://www.anybus.com/technologies/industrial-ethernet/modbus-tcp. [Accessed June 2021].

[25] Microchip, "Modbus TCP for the Microchip TCP/IP Stack Application Note," [Online]. Available: https://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en566544. [Accessed June 2021].

[26] Softing Industrial , "Modbus TCP Server (Slave) for Intel Altera FPGA | Softing," [Online]. Available: https://industrial.softing.com/products/protocol-software-and-sdks/modbus-tcp-server-for-altera-fpga.html. [Accessed June 2021].

[27] Softing Industrial, "Industrial Ethernet Implementation using FPGAs | White Paper".

[28] Modbus Org, *MODBUS Messaging on TCP/IP Implementation Guide V1.0b,* 2006.

[29] hypernyan, "GitHub - hypernyan/eth_vlg," [Online]. Available: https://github.com/hypernyan/eth_vlg. [Accessed Nov 2021].

[30] Microchip, *KSZ9031RNX Gigabit Ethernet Transceiver Datasheet,* 2017.

[31] Terasic Inc., *DE0-Nano-SoC User Manual Rev D0,* 2019.

[32] Intel Community, "Windows 10 driver support for USB Blaster ? - Intel Communities," [Online]. Available: https://community.intel.com/t5/Programmable-Devices/Windows-10-driver-support-for-USB-Blaster/td-p/55323.

[33] Intel Altera, "Quartus II Handbook Volume 2: Design Implementation and Optimization," 2015.

[34] G. Sutter, M. Ruiz, S. Lopez-Buedo and G. Alonso, "FPGA-based TCP/IP Checksum Offloading Engine for 100 Gbps Networks," in *International Conference on ReConFigurable Computing and FPGAs (ReConFig),* 2018.