

Design of a Scalable Network Interface to Support Enhanced TCP and UDP Processing for High Speed Networks

A thesis submitted for the degree of
Doctor of Philosophy
by
Mohamed Elbeshti



Murdoch University
2014

Declaration

To the best of my knowledge, this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for award of any other degree in any other university.

Signature:

Date: July 07, 2014

Copyright by
Mohamed Elbeshti
2014

Dedicated to my Parents, Amira, Ahmed, Aseal, Yousef and Omar
for their support...

Abstract

Communication networks have advanced rapidly in providing additional services, with improvements made to their bandwidth and the integration of advanced technology. As the speed of networks exceeds 10 Gbps, the time frame for completing the processing of TCP and UDP packets has become extremely short. The design and implementation of high performance Network Interfaces (NIs) that can support offload protocol functions for current and next-generation networks is challenging. In this thesis two software approaches are presented to enhance protocol processing of TCP and UDP in the network interface. A novel software Large Receive Offload (LRO) approach for enhancing the receiving side has been proposed. The LRO works by aggregating the incoming TCP and UDP packets into larger packets inside the NI's buffer. The receiving side software has been improved to support out-of-order packets. The second proposed software solution is applied on the Large Send Offload (LSO). The proposed LSO function processing is implemented by segmenting TCP and UDP messages that are larger than the Maximum Transmission Unit to the Maximum Segment Size. New packet headers are generated for each new outgoing packet.

A scalable programmable NI based 32-bit RISC core is presented that can support 100 Gbps network speeds. Acceleration of the processing time frame required at the NI has been implemented to prevent hazards (such as Data Hazard and Control Hazard) during the execution of the LRO and the LSO functions. An R2000/3000 RISC has been used in order to test the LRO and LSO functions and to discover the instruction set that is most suitable. Following this the VHDL NI was implemented with three pipeline RISC cores, a simple DMA controller and Content Addressable Memory. An evaluation of the desired RISC clock rate that is required to process TCP and UDP streams at 100 Gbps was conducted. It was determined that a RISC core running at 752 MHz with a DMA clock of 3753 MHz was able to process packets 512 bytes or larger fast enough to support 100 Gbps network speeds.

Publications

Journal Articles:

- M. Elbeshti, M. Dixon, and T. Koziniec Large Sending Offload: Design and Implementation for High-speed Communications Rate up to 100 Gbps, *International Journal of New Computer Architectures and their Applications (IJNCAA)*. Vol 3, No. 2. 2013.

Conference Papers:

- M. Elbeshti, M. Dixon, and T. Koziniec, Design and Simulating Specialized Embedded Cores for UDP Network Interface Processing: *24th International Conference on Modeling and Simulation (MS 2013)* Canada, 2013.
- M. Elbeshti, M. Dixon, and T. Koziniec, Design consideration for efficient network interface supporting the Large Receive Offload with embedded RISC: *36th International Conference on Telecommunications and Signal processing (TSP)*, Italy, 2013.
- M. Elbeshti, M. Dixon, and T. Koziniec, Design and Simulating a Network Interface-based RISC Cores: *19th Asia-Pacific Conference on Communications*, Bali, 2013.
- M. Elbeshti, M. Dixon, and T. Koziniec, Design a Scalable Ethernet Network Interface Supporting the Large Receive Offload for 100 Gbps: *12th International Symposium on Communications and Information Technologies (ISCIT)*, Australia, 2012.
- M. Elbeshti, M. Dixon, and T. Koziniec, RISC core supporting the Large Sending Offload in 100 Gbps: *12th International Symposium on Communications and Information Technologies (ISCIT)*, Australia, 2012.
- M. Elbeshti, M. Dixon, and T. Koziniec, An Evaluation of the TCP and UDP Processing Requirements Network Interface Design at 100 Gbps: *13th International Symposium on Advances of High Performance Computing and Networking (HPCN)*, Canada, 2011.

- M. Elbeshti, M. Dixon, and T. Koziniec, TCP and UDP Processing Requirements for Network Interface Design at 100 Gbps: *3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Budapest, 2011.
- M. Elbeshti, M. Dixon, and T. Koziniec, Facing the Challenges of Designing a Network Interfaces for 100 Gbps: *2nd International Conference on Research Challenges in Computer Science*, China, 2010.

Posters and Demonstration:

- Mohamed Elbeshti, Designed and Implemented a Programmable Network Interface for High-speed Communications: *Hot Chips: A Symposium on High Performance Chips*, 2013.

Table of Contents

Abstract	iv
Publications	v
Table of Contents	vii
List of Figures	xiii
List of Tables	xvi
Acknowledgements	xvii
List of Abbreviation	xviii
Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Protocol Processing Overview	2
1.3 Towards 100 Gbps Packet Processing	4
1.3.1 Protocol Processing Considerations	5
1.3.2 Network Interface Design Approaches	5
1.4 Programmable-based Network Interface Design	6
1.4.1 Processing Rate	7
1.4.2 Core Structure.....	8
1.5 Thesis Contributions	8
1.6 Research Approach	10
1.7 Organization of the Thesis	11
Chapter 2 Overview of the Protocol Processing.....	15
Chapter 2	15
2.1 Introduction	15
2.2 Overview of the Protocol Processing at a Server.....	16
2.2.1 Host CPU Time Required for Protocol Processing	17
2.2.2 Protocol Processing Time Used by the Network Interface.....	18
2.2.3 Packet Processing Time.....	20
2.2.3.1 Maximum Number of Packets in one Second.....	20

2.2.3.2	Maximum throughput of a workstation is dependent on packet size	22
2.2.4	Reducing Protocol Overhead and Enhancing Server Performance	25
2.2.5	Techniques Used for Protocol Processing in the Host Area.....	26
2.2.6	Extending the Ethernet Frame Size	28
2.2.7	Techniques Implemented Inside the Network Interface.....	28
2.3	An Analysis of the Existing Protocol Processing Solutions for Gigabit	31
2.4	Aims of the Research	33
2.5	Research Contributions	34
2.6	Offloading Processing and Related Work.....	35
2.6.1	Large Receive Offload in the Virtual Driver.....	35
2.6.2	Receive Side Coalescing	36
2.6.3	Large Receive Offload Concerns in High-Speed Networks.....	36
2.6.4	Large Segment Offload (LSO) Enhancements.....	37
2.7	Programmable Packet Processor Design Methodology	39
2.7.1	Network Interface Hardware-based.....	40
2.7.2	Network Interface Programmable-based.....	41
2.8	Programmable Approaches for Packet Processing	42
2.8.1	Network Interface Using a Single Processor.....	43
2.8.2	Using a General-Purpose Processor Supported by a DMA for Transferring Packets to or from a Network	44
2.8.3	Using a Dual Core Engine supported with DMA for UDP Protocol.....	45
2.8.4	Multiprocessing Cores for Packet Processing	46
2.8.5	Target Core Observation	48
2.9	Conclusion	49
 Chapter 3 Large Receive Offload Methodology.....		50
3.1	Introduction	50
3.1	Related Implementations of Large Receive Offload Processing	51
3.1.1	Virtual Large Receive Offload	51
3.1.2	LRO performance within the host area.....	54
3.1.3	Receive Side Coalescing	55
3.2	Enhancing the Large Receive Offload Processing.....	56
3.2.1	Lost Large Packet Treatment inside the Network Interface	58
3.2.2	Large Packet Processing.....	60
3.3	Primary Design and Structure for the Receiving Side	61

3.4	Receiving Side Processing	62
3.4.1	TCP Processing Methodology	64
3.4.1.1	Linked-list Structure Format	67
3.4.1.2	Out-of-order Processing	70
3.4.2	UDP/IP Processing Methodology.....	72
3.5	Verification of the LRO Processing.....	72
3.5.1	Modelling SPIM Simulator Architecture	74
3.5.2	Simulation Processing Analysis	77
3.5.3	Instruction Cycles	79
3.5.4	Instruction Type.....	81
3.5.5	Total Registers.....	82
3.5.6	RISC Clock Rate	83
3.6	Network Interface Design Considerations for 100 Gbps	85
3.6.1	DMA for Data Movements.....	86
3.6.2	Local Bus Width.....	87
3.6.3	Pipeline Stages.....	87
3.6.4	Lookup Memory	88
3.6.5	Overlapped Processing	88
3.7	Conclusion	89
	Chapter 4 Large Send Offload Methodology.....	91
4.1	Introduction	91
4.2	Sending Side Block Diagram	91
4.3	Protocol Processing Methodology	94
4.3.1	UDP Processing.....	96
4.3.2	TCP Processing	99
4.4	SPIM Simulator for LSO	100
4.5	Simulation Results	103
4.5.1	Instruction Types	105
4.5.2	RISC Clock Rate	106
4.6	Design Consideration for 100 Gbps at the Sending Side.....	108
4.6.1	Enhancing Packet Processing	109
4.7	Conclusion	110

Chapter 5 A Scalable Network Interface Architecture for 100 Gbps...112

5.1	Introduction	112
5.2	Network Interface Model	113
5.2.1	Network Interface Buffering	115
5.2.2	Data Transfer	117
5.2.2.1	DMA for Data Transfer	117
5.2.2.2	Bus Width.....	120
5.2.3	Content Addressable Memory	120
5.2.4	The CAM Implementation inside the proposed NI	122
5.3	The Network Interface FIFOs	127
5.4	The Interface Buffers	131
5.4.1	Memory Management.....	131
5.4.2	The Receiving and Sending Buffer	134
5.4.3	Receiver and Transmission Line Buffers	134
5.5	Conclusion	137

Chapter 6 Developing the RISC Core for TCP/IP and UDP/IP Processing138

6.1	Introduction	138
6.2	RISC Pipeline.....	138
6.3	Instructions Set Representation.....	142
6.3.1	Arithmetic and Logic Operation Instructions	143
6.3.2	Branch Instructions.....	144
6.3.3	Memory Access Instructions	146
6.4	Pipeline Hazard	148
6.5	RISC Registers	153
6.6	Components required for RISC cores	155
6.7	Packet Data Path	155
6.7.1	The PCI Interface.....	159
6.7.1.1	PCI Interface at the Packet Processing Unit.....	160
6.7.1.2	Reading data from the Receiving buffer	162
6.7.1.3	Reading data from Sending side.....	163
6.7.2	Interrupt Moderation Window Size.....	164
6.8	Conclusion	166

Chapter 7 LRO and LSO Processing Analysis inside the PPU	168
7.1 Introduction.....	168
7.2 Enhancement to Improve Packet Processing	168
7.3 Processing Analysis	171
7.3.1 Large Receive Offload Analysis through Full-System Simulation.....	173
7.3.1.1 TCP processing cycles	174
7.3.1.2 UDP processing cycles.....	175
7.3.2 Large Send Offload Analysis through Full-System Simulation.....	189
7.3.2.1 TCP processing cycles	189
7.3.2.2 UDP processing cycles.....	190
7.4 The Payload length Path.....	199
7.5 Conclusion	203
Chapter 8 VHDL Simulation Results	205
8.1 Introduction.....	205
8.2 Packet Processing Enhancements for High-Speed Networks.....	205
8.3 RISC Clock Rate for 100 Gbps.....	210
8.4 Results.....	211
8.5 The DMA and RISC Clock Rate for 100 Gbps.....	217
8.6 Conclusion	218
Chapter 9 Conclusion and Future Work	219
9.1 Summary of Contributions.....	219
9.2 Future Work	222
References	223
Appendix A Data Collection	231
A.1 Collection of real TCP and UDP streams from multiple tests.....	231
A.2 Tests Methodology.....	232
A.2.1 Receive side Flows	233

A.2.1.1 Commands:	234
A.2.2 Send Side Flows	234
A.2.2.1 Commands:	234
Appendix B Schematic Diagrams.....	235

List of Figures

Figure 2.1: Ethernet frame format.....	19
Figure 2.2: Theoretical maximum throughput for 10 Gbps	25
Figure 2.3: Workstation architecture.....	39
Figure 3.1a: Receive side data flow when Large Receive Offload is not implemented .	52
Figure 3.1b: Receive side data flow when Large Receive Offload is implemented	52
Figure 3.2: Extract part of the LRO code shows packets that do not match the LRO requirements in a separate buffer	53
Figure 3.3: Offloading the LRO approach to the Network Interface.....	57
Figure 3.4: The Ethernet network interface structure	62
Figure 3.5: Segment Message of a TCP stream	65
Figure 3.6: Processing flow of TCP of LRO.....	66
Figure 3.7: Linked-list data structure	68
Figure 3.8: Lookup Memory structure	71
Figure 3.9: Illustrates inter-packet processing	73
Figure 3.10: Receiving block diagram	74
Figure 3.11: Packet Processing Unit based SPIM simulator	75
Figure 3.12: Programmed I/O approach for data movement	77
Figure 3.13: A TCP/IP and UDP/IP Hexadecimal format	78
Figure 3.14: Total percentage of data movements of LRO.....	81
Figure 3.15: Floating Point registers during the processing of the proposed LRO	83
Figure 3.16: RISC clock rate for packet header processing.....	84
Figure 3.17: MIPS required for the Receiving side using Programmed I/O.....	85
Figure 3.18: DMA approach for data movement inside the Network Interface	87
Figure 3.19: Overlapped processing at the receiving side	89
Figure 4.1: Sending side Model	93
Figure 4.2: Four pointers are used with the new approach for segmenting packets	96
Figure 4.3: Processing flow of TCP and UDP of LSO	97
Figure 4.4: Procedure of sending a UDP user data application	98
Figure 4.5: Procedure of sending a TCP user data application	99
Figure 4.6: Sending block diagram	100
Figure 4.7: SPIM simulator block diagram.....	101
Figure 4.8: Communication between the host and the NI	102
Figure 4.9: Processing flow.....	103
Figure 4.10: Total percentage of data movements of LSO processing	105
Figure 4.11: RISC clock rate for packet header processing.....	107
Figure 4.12: Amount of MIPS required for sending side using Programmed I/O.....	108
Figure 4.13: Pipeline processing at the sending side	110
Figure 5.1: Network interface block diagram	114
Figure 5.2: DMA structure	118

Figure 5.3: DMA cycles to transfer data from the source to the destination	118
Figure 5.4: DMA channel	119
Figure 5.5: CAM structure when the Linked-list	121
Figure 5.6: CAM based implementation of the look-up-table	123
Figure 5.7: CAM-based search engine block diagram.....	124
Figure 5.8: Cycles required during read operations	125
Figure 5.9: The two FIFOs are used to send data from the receiver RISC processor ..	129
Figure 5.10: Sends the TCP active connections to the receiving side through FIFO 3	130
Figure 5.11: The FIFO carries the information needed for segmenting a message	130
Figure 5.12: Circulation Buffer architecture	132
Figure 5.13a: Tracking the size of the RB	133
Figure 5.13b: Signal sent when 200 pages of the RB are occupied	133
Figure 5.14: Receiving Buffer Interface architecture	135
Figure 5.15: Sending buffer interface architecture	136
Figure 6.1a: Normal Structure of RISC instruction pipeline	139
Figure 6.1b: Structure of RISC instruction pipeline	139
Figure 6.2: Block diagram of the Fetch, Decode, Execute and Write/Back	141
Figure 6.3a: Arithmetic/Logic instruction formation.....	143
Figure 6.3b: Arithmetic/Logic immediate instruction formation	143
Figure 6.4: Arithmetic instructions	144
Figure 6.5: Branch instruction format.....	145
Figure 6.6: Branch instruction example	145
Figure 6.7: Load/Store instruction format.....	146
Figure 6.8: Load and store instructions with memory address	147
Figure 6.9: LCAM instruction format.....	147
Figure 6.10: Load a memory address from CAM.....	148
Figure 6.11a: Before scheduling the branch-delay slot.....	149
Figure 6.12: Before scheduling procedure	151
Figure 6.13: Delay slot technique for Data hazard	151
Figure 6.14: Forward mechanism used in the simulator	153
Figure 6.15: Latching the output of the Arithmetic Logic Unit	153
Figure 6.16: RISC register file	154
Figure 6.17: The topology of the test environment.....	156
Figure 6.18: Tested model for sending and receiving packets	157
Figure 6.19: Transferring packets from the Receiving Buffer to the Host Memory.....	160
Figure 6.20: Timing diagram captured from the simulation for burst transfer	161
Figure 7.1: Tested model for sending and receiving packets at the PPUnt	170
Figure 7.2: Large Receive Offload Processing cycles characteristics	172
Figure 7.3: Timing diagram for TCP BOM packet Instructions	177
Figure 7.4: Total number of instructions for TCP BOM packet without idle cycles....	178
Figure 7.5: Total number of instructions for TCP COM packet without idle cycles....	179
Figure 7.6: Total number of instructions for TCP EOM packet without idle cycles....	180
Figure 7.7: Total number of instructions for TCP SSM packet without idle cycles.....	181

Figure 7.8 : Total number of instructions for the TCP out-of-order packet when the sub linked-list is equal to “0” in the CAM	182
Figure 7.9: Total instructions for TCP out-of-order packet when the sub linked-list is not equal to “0” in the CAM.....	183
Figure 7.10: Total number of instructions to process the UDP BOM packet	184
Figure 7.11: Total number of instructions to process the UDP COM packet	185
Figure 7.12: Total number of instructions to process the UDP EOM packet	186
Figure 7.13: Total number of instructions to process the UDP SSM packet	187
Figure 7.14: Total instructions for the UDP out-of-order packet when the sub linked-list is not equal to “0” in the CAM.....	188
Figure 7.15: Total number of instructions to process the TCP BOM packet	191
Figure 7.16: Total number of instructions to process the TCP COM packet	192
Figure 7.17: Total number of instructions to process the TCP EOM packet	193
Figure 7.18: Total number of instructions to process the TCP SSM packet	194
Figure 7.19: Total number of instructions for the UDP BOM packets	195
Figure 7.20: Total number of instructions for the UDP COM packets	196
Figure 7.21: Total number of instructions for the UDP EOM packets	197
Figure 7.22: Total number of instructions for the UDP SSM packets	198
Figure 8.1: Total RISC idle cycles when the DMA clock is double RISC clock rate ..	206
Figure 8.2: Desired DMA clock rate for LRO and LSO.....	207
Figure 8.3: RISC clock rate for LSO and LRO for UDP/IP when the DMA is 3759 Mhz for receiving-side and 2115MHz for sending-side	211
Figure A.1: Network topology	231
Figure A.2: A snapshot of the Hexadecimal file from WirShark	235
Figure B.1: VHDL based Packet Processing Unit architecture	237
Figure B.2: Structure of RISC instruction VHDL based pipeline	238
Figure B.3: DMA schematic diagram	239
Figure B 4: RISC register file schematic diagram	240
Figure B.5: CAM schematic diagram	241
Figure B.6: Receiver Buffer Interface (RBI) schematic diagram	242
Figure B.7: Minimize Data Hazard by latching the output of the ALU by forwarding hardware (U12) to be read within next instruction (forward mechanism).	243
Figure B.8: VHDL block diagram for DMA entity	244
Figure B9:VHDL block diagram for Register entity	245
Figure B.10: VHDL block diagram for PipeLine entity	246
Figure B.11:VHDL block diagram for CAM entity	247

List of Tables

Table 2.1: The frame rate per second applied to different line speed rates	22
Table 2.2: Minimum and Maximum-sized Ethernet frames	23
Table 2.3: Some of Network Processor cores used as network processor	43
Table 3.1: A comparison between the virtual LRO processing and offloaded LRO	58
Table 3.2: Number of cycles needed to complete out-of-order TCP or UDP packet	80
Table 3.3: Instruction types that are used with LRO processing	82
Table 4.1: Number of cycles within the SPIM simulator.....	104
Table 4.2: Instruction types used with LRO processing	106
Table 5.1: Input and output signals of the CAM	125
Table 6.1: RISC Instructions.....	142
Table 6.2: Number of occurrence for Conditional Branch instructions	150
Table 6.3: Number of occurrence of the Read after Write (R/W) hazard for UDP	152
Table 6.4: Number of occurrence of the Read after Write (R/W) hazard for TCP	152
Table 6.5: Number of Register files size for LRO and LSO.....	155
Table 6.6: Shows the components needed for the LSO and the LRO functions.....	155
Table 6.7: The Interrupt Moderation sizes and absolute time	165
Table 7.1: The number of RISC instructions required to process the LRO when the DMA is double RISC's clock.....	200
Table 7.2: RISC cycles while performing the Large Send Offload, when the DMA clock rate is double the RISC's clock rate.....	202
Table 8.1: Total number of RISC instructions to complete the processing of the LRO for TCP and UDP when the DMA clock is 3759 MHz.....	208
Table 8.2: Total number of RISC instructions to complete the processing of the LSO for TCP and UDP when the DMA clock is 2115 MHz	209
Table 8.3: Packet processing at the receiving side when the RISC clock is 752 MHz and the DMA is 3759 MHz.....	213
Table 8.4: LSO packet processing time when the RISC clock is 752 MHz and the DMA is 3759	214
Table 8.5: LRO packet processing time when the RISC clock is 1449 MHz and the DMA is 3759	215
Table 8.6: LSO packet processing time when the RISC clock is 1449 MHz and the DMA is 3759	216
Table 8.7: The RISC and DMA clock rate supporting LRO and LSO for TCP and UDP at 100 Gbps	217

Acknowledgements

All praises are due to Almighty God “Allah”, Who provided me with the strength and willingness to undertake this work and the opportunity to contribute a drop in the sea of knowledge.

I am most grateful to my supervisors, Mike Dixon and Terry Koziniec. Their open doors and persistent encouragement was invaluable for the completion of this research. Also appreciated the time that Patrick, Lynette spent with me

Finally, these acknowledgements would not complete without appreciating the unwavering support of my family including my father, my wife and children, and the memory of my mother.

List of Abbreviation

ACK	ACK - Acknowledges received data
BOM	Beginning of Message
CAM	Content Addressable Memory
COM	Continuation of Messages
DMAC	Direct Memory Access Controller
EOM	End of Message
FIN	FIN - (Final) Cleanly terminates a connection
HI	Host Interface
HNIC	Host-NI Level of Communication Buffer
LI	Line Interface
LRO	Large Receive Offload
LSO	Large Send Offload
MAC	Media Access Control
RB	Receiving Buffer
RBI	Receiving Buffer Interface
REP	Receiving Embedded Processor
RISC	Reduce Instruct Set Computer
SB	Sending Buffer
SBI	Receiving Buffer interface
SEP	Sending Embedded Processor
SN	Sequence Number
SSM	Single Segment Message
SYN	SYN - (Synchronize) Initiates a connection
VHDL	Very High-Speed Description Language