REMOTE DYNAMICALLY RECONFIGURABLE NETWORK PROCESSING MIDDLEBOX

TAN TZE HON

UNIVERSITI TEKNOLOGI MALAYSIA

REMOTE DYNAMICALLY RECONFIGURABLE NETWORK PROCESSING MIDDLEBOX

TAN TZE HON

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

FEBRUARY 2015

Specially dedicated to my beloved family, lectures and friends
for their support and encouragement throughout my education.

ACKNOWLEDGEMENT

First of all, I would like to express my deepest and sincerest appreciation to my supervisor, Dr. Nadzir Marsono and co-supervisor Dr. Ooi Chia Yee. Their constant guidance and motivation lead to this fruitful research. Other than research skill and technical skill, they teach me the lessons of life, which will be very helpful for my future development. Above all, I would never realized and awakened my potential without their enlightenment and inspiration.

I would also like to thank everyone who contributed directly and indirectly to this research. I sincerely thank my fellow postgraduate students and other lecturers who help and accompany me during my study in UTM.

Lastly, I would like to thank Dr. Nadzir Marsono again for everything since the very beginning of my study in UTM. Thank you!

ABSTRACT

Remote dynamically reconfigurable platforms use dynamic reconfiguration to provide solutions for applications to cope with changes in both functional and performance requirements. Most existing remote dynamically reconfigurable platforms are inefficient in handling dynamic reconfiguration process. This is due to the use of general-purpose processor in their designs or having limited partial bitstream transmission throughput that results in long device down-time. This thesis presents an architecture of remote dynamically reconfigurable middlebox on NetFPGA development board. The developed platform relies on a customized reconfiguration controller and Internal Configuration Access Port to achieve dynamic reconfiguration. In addition, this platform uses 1Gbps Ethernet link for partial bitstreams transmission to achieve remote update. In order to offer maximum flexibility for network processing, this work includes an architecture that allows remote updates on packet-forwarding as well. This allows packet-forwarding algorithm and its implementation to be optimized or customized after deployment. A case study on network protection using this platform is included in this thesis to verify application functionality updates. All hardware designs are verified using ModelSim simulation and tested experimentally using the NetFPGA development board. The developed remote dynamically reconfigurable platform is stand-alone and can achieve remote functional update without the need of a host computer. Based on experimental results, the proposed platform achieves 350Mbps reconfiguration throughput, which is significant for mass remote update as device downtime for update is reduced. The developed platform is suitable to be used as network processing middlebox.

ABSTRAK

Platform keboleh-tatarajahan semula dinamik secara jarak jauh menggunakan fitur tatarajah semula dinamik untuk menyediakan penyelesaian kepada aplikasi dalam menangani perubahan keperluan fungsian dan prestasi. Kebanyakan platform keboleh-tatarajahan semula dinamik secara jarak jauh sedia ada adalah tidak efisien dalam pengendalian proses pentatarajahan semula. Hal ini disebabkan penggunaan pemproses tujuan am di dalam reka bentuk atau mempunyai kadar celus yang sangat terhad dalam penghantaran aliran bit separa yang boleh mengakibatkan masa henti peranti yang panjang. Tesis ini membentangkan seni bina *middlebox* keboleh-tatarajahan semula dinamik secara jarak jauh dengan menggunakan papan pembangunan NetFPGA. Platform yang telah dibangunkan bergantung kepada pengawal pentatarajahan semula tersuai dan port capaian tatarajah semula dalaman untuk mencapai pentatarajahan semula secara dinamik. Di samping itu, platform ini menggunakan pautan Ethernet selaju 1Gbps dalam penghantaran aliran bit separa untuk melaksanakan kemas kini secara jarak jauh. Dalam usaha untuk menawarkan kelenturan maksimum untuk pemprosesan rangkaian, kerja ini juga merangkumi satu seni bina yang membolehkan kemas kini jarak jauh pada algoritma ajuan paket. Hal ini membolehkan algoritma ajuan paket dan implementasinya dioptimumkan atau disesuaikan selepas kerah tugas. Satu kajian kes dalam perlindungan rangkaian dengan menggunakan platform ini terkandung dalam tesis ini untuk menentusahkan kemas kini fungsian aplikasi. Semua reka bentuk perkakasan telah ditentusahkan dengan menggunakan simulasi ModelSim dan diuji secara eksperimen dengan menggunakan papan pembangunan NetFPGA. Platform keboleh-tatarajahan semula dinamik secara jarak jauh yang telah dibangunkan boleh berfungsi secara kendiri dan dapat mencapai kemas kini fungsian secara jarak jauh tanpa memerlukan komputer hos. Berdasarkan keputusan eksperimen, platform yang dicadangkan dapat mencapai kadar celus pentatarajahan semula sebanyak 350Mbps, yakni penting kepada kemas kini jarak jauh secara besar-besaran kerana masa henti peranti semasa kemas kini telah disingkatkan. Platform yang telah dibangunkan sesuai diguna sebagai peranti perantaraan dalam pemprosesan rangkaian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AMBA	-	Advanced Microcontroller Bus Architecture
ASIC	-	Application-Specific Integrated Circuit
AXI	-	Advanced eXtensible Interface
BRAM	-	Block Random-Access Memory
CAM	-	Content-Addressable Memory
CLB	-	Configurable Logic Block
CPLD	-	Complex Programmable Logic Device
CPU	-	Central Processing Unit
DDR SDRAM	-	Double Data Rate Synchronous Dynamic Random-Access Memory
DMA	-	Direct Memory Access
EDK	-	Embedded Development Kit
EPROM	-	Erasable Programmable Read Only Memory
EEPROM	-	Electrically Erasable Programmable Read Only Memory
FIFO	-	First In, First Out
FPGA	-	Field-Programmable Gate Array
GPP	-	General Purpose Processor
HDL	-	Hardware Description Language
ICAP	-	Internal Configuration Access Port
ICON	-	Integrated CONtroller
IDS	-	Intrusion Detection System
ILA	-	Integrated Logic Analyzer
I/O	-	Input/Output
IP	-	Internet Protocol
IPS	-	Intrusion Prevention System
ISE	-	Integrated Software Environment
JTAG	-	Joint Test Action Group
LUT	-	LookUp Table
MAC	-	Media Access Control
NIC	-	Network Interface Controller

NIDS	-	Network Intrusion Detection System
OPB	-	On-chip Peripheral Bus
PC	-	Personal Computer
PLB	-	Processor Local Bus
PRM	-	Partial Reconfigurable Module
PRR	-	Partial Reconfigurable Region
RFC	-	Request for Comments
SDK	-	Software Development Kit
SDRAM	-	Synchronous Dynamic Random-Access Memory
SFP	-	Small Form-factor Pluggable
SoC	-	Systems-on-Chip
SoPC	-	System-on-Programmable-Chip
SRAM	-	Static Random-Access Memory
TCP	-	Transmission Control Protocol
UART	-	Universal Asynchronous Receiver/Transmitter
UDP	-	User Datagram Protocol
VHDL	-	Very High Speed Integrated Circuit Hardware Description Language
XPS	-	Xilinx Platform Studio

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Reconfigurable Computing as a Paradigm Shift in Computing

The inefficiency [1, 2] of the von Neumann machine paradigm [3] leads to integration of both von Neumann central processing unit (instruction driven) and non-von Neumann accelerators (data driven) implementation [2]. This is because instruction stream execution of von Neumann machine requires a lot of memory cycles. Currently, the focus of implementation has shifted towards replacing hardwired accelerators with reconfigurable devices. This approach offers better flexibility [4] and contributes to the emergence of reconfigurable computing techniques and methodologies.

Systems-on-Chip (SoC) becomes a common implementation solution in electronic systems but it lacks the flexibility to cope with the rapid changes of functionality requirement. A reconfigurable device with large number of logic elements becomes a good alternative to SoC by offering flexibility at the cost of performance. This results in the emergence of System-on-Programmable-Chip (SoPC) [5], which implement a system-on-chip consisting intellectual property (IP) cores, general-purpose processor (GPP) and custom hardware in a single reconfigurable device. To apply changes and updates to the custom hardware in the reconfigurable device, a new partial bitstream is loaded to the reconfigurable device at run time. This leads to the requirement of a good framework for remote dynamically reconfigurable platform.

1.2 Dynamic Reconfiguration in Network Processing Unit

Telecommunication bandwidth is expected to grow at a rate three times higher than computation processing capability [6]. This trend clearly shows that high performance requirement, especially throughput requirement is critical in network processing units to cope with the growing demands for bandwidth [7]. Application-specific integrated circuit (ASIC) design is a possible implementation solution as it enables low-level parallelism and processing of packets in deep pipeline [8]. Unfortunately, ASIC designs are inflexible and do not allow functional updates. Thus, ASICs usually have limited lifetime in the market due to their incapability to adapt to changes. ASIC implementation also requires longer time to be designed, fabricated and tested, compared to other implementation alternatives.

The flexibility to do functional update is an important requirement in network processing [9] as it would allow network applications to remain updated from time to time and prolongs its lifetime in the market. Flexibility requirements come when some of the execution requirements are not known during the design time or may change over time in unforeseen ways. For instance, firewall implemented in ASIC could only protect the networks from known threats during its design time [8]. Network processing units implemented in software can have functional updates but the execution is very low in performance, especially the throughput. For example, throughput of software-based router in packet processing is slower than hardware-based router in several orders of magnitude [7]. This limitation is very significant, as network application requires throughput to cope with the growing demands for bandwidth [7]. In short, both performance and flexibility are important for network application devices.

Field-programmable gate array (FPGA) is a good option to implement network application devices because FPGA implementation can offer desirable balance between flexibility and performance. FPGA has both performance advantages of ASIC solution and flexibility advantage of software solution [7]. For instance, firewall implemented in reconfigurable device is able to achieve significant improvements in performance and security [10]. The system flexibility comes from the reconfigurability feature in the FPGA devices and is used to update the system when new security threats is found [10]. With the performance and flexibility offered by reconfigurable device, NetFPGA [11] has emerged as a network application prototyping platform that utilizes FPGA devices.

Most network application especially middleboxes are required to operate in high throughput and are distributed. Additionally, middleboxes are required to remain active and operate continuously so that their connectivity with end nodes is maintained and it can be updated remotely. Thus, the cost to apply updates to such system is very high. This problem can be solved by utilizing dynamic reconfiguration feature found in reconfigurable devices. However, utilization of dynamic reconfiguration feature is not straightforward and requires proper methodology in the design process. Therefore, a good framework that is able to efficiently perform dynamic reconfiguration in reconfigurable devices is required in network applications.

1.3 Problem Statement

Extant works have shown that reconfigurable devices is a good solution for implementation that requires both performance and flexibility [7–9, 12, 13]. There are many works that have been proposed using reconfigurable device to achieve performance advantage. However, works on exploiting the flexibility in reconfigurable device are still limited. The reason lies on the fact that dynamic reconfiguration is of recent interest and is only supported by limited family of reconfigurable devices. Additionally, proper methodology in the design process is required to utilize dynamic reconfiguration.

Remote dynamic reconfiguration is capable to cope with rapid functional changes for system implemented in reconfigurable devices. To achieve this, partial bitstream should be loaded into these reconfigurable devices in the most generic and efficient way. However, some recent works [14–20] achieved partial reconfiguration by using General Purpose Processor embedded in the design. This requires additional logic resources and longer time to perform partial reconfiguration. Additionally, some of the works in [15, 19–24] used shared bus structure, which may restrict other components from using it during reconfiguration process [25].

NetFPGA development board is a network application FPGA development board. Previous works [10, 26] show that the NetFPGA development board has the potential to be developed into a remote dynamically reconfigurable platform. This is because NetFPGA uses reconfigurable device that support dynamic reconfiguration feature and also provide well-established communication framework. Recently, [26] implemented a network application that uses the partial reconfiguration in NetFPGA development board. However, JTAG interface is used to load the partial bitstream into the FPGA device, which requires longer reconfiguration time [15]. Zhang et al. [10] have designed a remote dynamically reconfigurable security system using NetFPGA development board. However, the remote dyanmic reconfiguration rely on host PC for bitstream transmission and translation, which is inefficient.

1.4 Research Objectives

Based on the background studies and existing issues, the aim of this thesis is to design and implement a remote dynamically reconfigurable platform. The main objectives of this research work are:

1. To design and implement a remote dynamically reconfigurable platform using NetFPGA development board. The developed platform does not rely on GPP or host computer to handle the dynamic reconfiguration process. Instead, the reconfiguration controller has been implemented using existing logic resources in the reconfigurable device itself. The application implemented using the developed platform should be able to be updated remotely through the Ethernet connection.
2. To design and implement a remote dynamically reconfigurable middlebox for network protection scheme. The packet-forwarding algorithm in the developed middlebox should be able to be updated remotely through the Ethernet connection. The network protection application implemented using the developed middlebox should be able to be updated remotely through the UDP/IP connection.

1.5 Scope of Work

Based on the research objectives and available resources, the scope of this research are as follow:

1. The design of remote dynamically reconfigurable platform excludes authentication mechanism. Authentication mechanism is not in the scope of work

because it is an optional feature and this feature can be extended to the developed platform when the application requires it.

2. The partial bitstream is not encrypted for dynamic reconfiguration. Partial bitstream encryption is not in the focus of this work as it is an optional feature as this feature can be included afterward depending on application requirements.
3. The case studies of network protection are targeted for stateless Network Intrusion Prevention System (NIPS) and port based firewall. However, other applications can still be implemented using the developed platform as the developed platform is functionally extensible.
4. The developed platform supports packets size up to 2048 Bytes, which is larger than the maximum transmission unit of Ethernet V2. However, the packets size can be increased by adjusting the depth of FIFO used.
5. The size of each bistream packets are limited to 1016 Bytes. Even so, the size of the bitstream packet can be increased by adjusting the depth of FIFO.

1.6 Research Contributions

This thesis contributes to two research contributions. The first contribution is the architecture for remote dynamically reconfigurable platform on NetFPGA 10G development boards. The proposed architecture supports remote update on the packet-forwarding mechanism, which allows high degree of customization and optimization after system deployment. With the proposed architectures, most existing network applications from NetFPGA repository [27] can be updated remotely when integrated into the implemented platform.

The second contribution is the architecture of a customized reconfiguration controller using available logic resources in the FPGA device. The design effort results in higher efficiency in the dynamic reconfiguration process and utilization in logic resources. Additionally, the dynamic reconfiguration process can be handled internally by the implemented reconfiguration controller, therefore reduces external component dependency. Combination of both contributions result in a better design and implementation alternative for remote dynamically reconfigurable platform, which is resource-efficient and processing efficient.

1.7 Thesis Organization

The rest of this thesis is organized based on the following structure.

Chapter 2 covers literature review of this research, which are related theoretical background and related works. Discussion on literatures mainly focus on dynamic reconfiguration and network applications.

Chapter 3 describes methodology to achieve the research objectives. This includes explanation on the architecture components, implementation flow, development environment and verification techniques.

Chapter 4 presents details on design and implementation of the proposed platform using NetFPGA development board. This chapter also includes evaluation of the implemented platform for verification and benchmark purposes.

Chapter 5 provides a case study of network protection application using the implemented remote dynamically reconfigurable platform.

Chapter 6 summarizes this thesis, stating contributions and limitations of this research and provides suggestions for future research.

REFERENCES

1. Cardoso, J. and Hübner, M. *Reconfigurable Computing: From FPGAs to Hardware/Software Codesign*. New York: Springer. 2011.
2. Hartenstein, R. The von Neumann Syndrome. *Stamatis Vassiliadis Memorial Symposium*. Delft, Netherlands. 2007. 1–7.
3. Von Neumann, J. First Draft of a Report on the EDVAC. *Annals of the History of Computing, IEEE*, 1993. 15(4): 27–75.
4. Becker, J. and Hartenstein, R. Configware and morphware going mainstream. *Journal of Systems Architecture*, 2003. 49(4): 127–142.
5. Hamblen, J. O. and Hall, T. S. Using System-on-a-Programmable-Chip Technology to Design Embedded Systems. *International Journal of Computer Applications*, 2006. 13(3): 1–11.
6. Kocovic, P. Four laws for today and tomorrow. *Journal of Applied Research and Technology*, 2008. 6(3): 133–146.
7. Lockwood, J. W., Naufel, N., Turner, J. S. and Taylor, D. E. Reprogrammable network packet processing on the field programmable port extender (FPX). *Proceedings of the 2001 ACM/SIGDA ninth international symposium on Field programmable gate arrays*. Monterey, CA, USA. 2001. 87–93.
8. Lockwood, J. W., Neely, C., Zuver, C., Moscola, J., Dharmapurikar, S. and Lim, D. An extensible, system-on-programmable-chip, content-aware Internet firewall. In: *Field Programmable Logic and Application*. Springer Berlin Heidelberg. 859–868. 2003.
9. Lockwood, J. W. An Open Platform for Development of Network Processing Modules in Reprogrammable Hardware. *IEC DesignCon'01*. Santa Clara, CA, USA. 2001. 9–19.
10. Zhang, K., Ding, X., Xiong, K., Yu, B. and Dai, S. RSS: A Reconfigurable Security System Designed on NetFPGA and Virtex5-LX110T. *1st European NetFPGA Developers Workshop*. Cambridge, England, UK. 2010.
11. Naous, J., Gibb, G., Bolouki, S. and McKeown, N. NetFPGA: Reusable router

architecture for experimental research. *Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*. Seattle, WA, USA. 2008. 1–7.

12. Lockwood, J. W., Moscola, J., Reddick, D., Kulig, M. and Brooks, T. Application of hardware accelerated extensible network nodes for internet worm and virus protection. In: *Active Networks*. Springer Berlin Heidelberg. 44–57. 2004.

13. Bobda, C. *Introduction to Reconfigurable Computing: Architectures, algorithms and applications*. Netherlands: Springer. 2007.

14. Castillo, J., Huerta, P., Lopez, V. and Martinez, J. I. A secure self-reconfiguring architecture based on open-source hardware. *International Conference on Reconfigurable Computing and FPGAs, 2005. ReConFig 2005.* Puebla City, Mexico. 2005. 7–13.

15. Krifa, M. N., Ouni, B. and Mtibaa, A. Exploring the self reconfiguration of FPGA: Design flow, architecture and performance. *International Journal on Computer Science and Engineering*, 2011. 3(4): 1713–1720.

16. Blodget, B., James-Roxby, P., Keller, E., McMillan, S. and Sundararajan, P. A self-reconfiguring platform. In: *Field Programmable Logic and Application*. Springer Berlin Heidelberg. 565–574. 2003.

17. Ismaili, Z. E. A. A. and Moussa, A. Self-Partial and Dynamic Reconfiguration Implementation for AES using FPGA. *International Journal of Computer Science Issues (IJCSI)*, 2009. 1(2): 33–40.

18. Paulsson, K., Hübner, M. and Becker, J. Dynamic power optimization by exploiting self-reconfiguration in Xilinx Spartan 3-based systems. *Microprocessors and Microsystems*, 2009. 33(1): 46–52.

19. Lagger, A., Upegui, A., Sanchez, E. and Gonzalez, I. Self-reconfigurable pervasive platform for cryptographic application. *International Conference on Field Programmable Logic and Applications, 2006. FPL'06.* Madrid, Spain. 2006. 1–4.

20. Williams, J. A. and Bergmann, N. W. Embedded Linux as a platform for dynamically self-reconfiguring systems-on-chip. *Ersa'04: the 2004 International Conference On Engineering of Reconfigurable Systems and Algorithms*. Nevada, USA. 2004. 163–169.

21. Cuoccio, A., Grassi, P. R., Rana, V., Santambrogio, M. D. and Sciuto, D. A generation flow for self-reconfiguration controllers customization. *4th IEEE International Symposium on Electronic Design, Test and Applications, 2008.*

DELTA 2008. Hong Kong, China. 2008. 279–284.

22. Claus, C., Muller, F. H., Zeppenfeld, J. and Stechele, W. A new framework to accelerate Virtex-II Pro dynamic partial self-reconfiguration. *IEEE International Parallel and Distributed Processing Symposium, 2007. IPDPS 2007*. Long Beach, CA, USA. 2007. 1–7.

23. Bomel, P., Crenne, J., Ye, L., Diguet, J.-P. and Gogniat, G. Ultra-fast downloading of partial bitstreams through ethernet. In: *Architecture of Computing Systems–ARCS 2009*. Springer Berlin Heidelberg. 72–83. 2009.

24. Bomel, P., Gogniat, G. and Diguet, J.-P. A networked, lightweight and partially reconfigurable platform. In: *Reconfigurable Computing: Architectures, Tools and Applications*. Springer Berlin Heidelberg. 318–323. 2008.

25. Hoffman, J. C. and Pattichis, M. S. A high-speed dynamic partial reconfiguration controller using direct memory access through a multiport memory controller and overclocking with active feedback. *International Journal of Reconfigurable Computing*, 2011. 2011: 1–10.

26. Yin, D., Unnikrishnan, D., Liao, Y., Gao, L. and Tessier, R. Customizing virtual networks with partial FPGA reconfiguration. *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*. New Delhi, India. 2010. 57–64.

27. NetFPGA GitHub Organization. Published online, 2014. URL `https://github.com/NetFPGA`.

28. Schallenberg, A. *Dynamic partial self-reconfiguration: Quick modeling, simulation, and synthesis*. Germany: Suedwestdeutscher Verlag fuer Hochschulschriften. 2010.

29. Fons Lluis, F. *Embedded Electronic Systems Driven by Run-time Reconfigurable Hardware*. Ph.d. dissertation. Universitat Rovira i Virgili. 2012.

30. Partial Reconfiguration in the ISE Design Suite. Published online, 2014. URL `http://www.xilinx.com/tools/partial-reconfiguration.htm`.

31. Morford, C. J. *BitMat-Bitstream Manipulation Tool for Xilinx FPGAs*. Master dissertation. Virginia Polytechnic Institute and State University. 2005.

32. NetFPGA. Published online, 2014. URL `http://netfpga.org/`.

33. Kuwatly, I., Sraj, M., Al Masri, Z. and Artail, H. A dynamic honeypot design for intrusion detection. *IEEE/ACS International Conference on Pervasive*

Services, 2004. ICPS 2004. Beirut, Lebanon. 2004. 95–104.

34. Park, K. and Kim, H. *Remote FPGA reconfiguration using MicroBlaze or PowerPC processors*. Xilinx, 2005. Application Note: XAPP441 (v1. 1) ed.

35. Mesquita, D., Moraes, F., Palma, J., Möller, L. and Calazans, N. Remote and Partial Reconfiguration of FPGAs: Tools and Trends. *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*. Nice, France. 2003. 177–185.

36. Muhlbach, S. and Koch, A. A dynamically reconfigured network platform for high-speed malware collection. *2010 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*. Quintana Roo, Mexico. 2010. 79–84.

37. Muhlbach, S. and Koch, A. A dynamically reconfigured multi-FPGA network platform for high-speed malware collection. *International Journal of Reconfigurable Computing*, 2012. 2012(4): 1–14.

38. Sato, T. and Fukase, M.-a. Reconfigurable hardware implementation of host-based IDS. *The 9th Asia-Pacific Conference on Communications, 2003. APCC 2003.* Penang, Malaysia. 2003, vol. 2. 849–853.

39. Song, H., Sproull, T., Attig, M. and Lockwood, J. Snort offloader: A reconfigurable hardware NIDS filter. *International Conference on Field Programmable Logic and Applications, 2005.* Tampere, Finland. 2005. 493–498.

40. Li, S., Torresen, J. and Soraasen, O. Exploiting reconfigurable hardware for network security. *11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2003. FCCM 2003.* Napa, CA, USA. 2003. 292–293.

41. Tummala, A. K. and Patel, P. Distributed IDS using reconfigurable hardware. *IEEE International Parallel and Distributed Processing Symposium, 2007. IPDPS 2007.* Long Beach, CA, USA. 2007. 1–6.

42. Pontarelli, S., Greco, C., Nobile, E., Teofili, S. and Bianchi, G. Exploiting dynamic reconfiguration for FPGA based network intrusion detection systems. *2010 International Conference on Field Programmable Logic and Applications (FPL)*. Milan, Italy. 2010. 10–14.

43. Antichi, G., Shahbaz, M., Giordano, S. and Moore, A. From 1G to 10G: code reuse in action. *First Workshop on High Performance and Programmable Networking 2013*. New York City, NY, USA. 2013. 31–37.

44. Wireshark. Published online, 2014. URL `http://www.wireshark.org/`.

45. ModelSim PE Student Edition. Published online, 2014. URL `http://www.mentor.com/company/higher_ed/modelsim-student-edition`.

46. Visual Studio. Published online, 2014. URL `http://msdn.microsoft.com/en-us/vstudio/aa718325%28v=vs.110%29.aspx`.

47. WinPcap. Published online, 2014. URL `https://www.winpcap.org/`.

48. ChipScope Pro Debugging Overview. Published online, 2014. URL `http://www.xilinx.com/itp/xilinx10/isehelp/ise_c_process_analyze_design_using_chipscope.htm`.

49. Platform Studio and the Embedded Development Kit (EDK). Published online, 2014. URL `http://www.xilinx.com/tools/platform.htm`.

50. Hubner, M., Gohringer, D., Noguera, J. and Becker, J. Fast dynamic and partial reconfiguration data path with low hardware overhead on Xilinx FPGAs. *2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*. Atlanta, GA, USA. 2010. 1–8.

51. Braun, L., Paulsson, K., Kromer, H., Hubner, M. and Becker, J. Data path driven waveform-like reconfiguration. *International Conference on Field Programmable Logic and Applications, 2008. FPL 2008.* Heidelberg, Germany. 2008. 607–610.

52. Canto, E., Lopez, M., Fons, F. *et al.* Self reconfiguration of embedded systems mapped on Spartan-3. *4th International Workshop on Reconfigurable Communication Centric SoCs (ReCoSoC 2008)*. Barcelona, Spain. 2008. 117–123.

53. Gonzalez, I., Aguayo, E. and Lopez-Buedo, S. Self-reconfigurable embedded systems on low-cost FPGAs. *Micro, IEEE*, 2007. 27(4): 49–57.

54. Zaidi, I., Nabina, A., Canagarajah, C. N. and Nunez-Yanez, J. Evaluating dynamic partial reconfiguration in the integer pipeline of a FPGA-based opensource processor. *International Conference on Field Programmable Logic and Applications, 2008. FPL 2008.* Heidelberg, Germany. 2008. 547–550.

55. Tan, T. H., Ooi, C. Y., Hau, Y. W., Shaikh-Husin, N. and Marsono, M. Remote dynamically reconfigurable platform using NetFPGA. *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. Melbourne, Australia. 2014. 1239–1242.

56. RFC 1700. Published online, 2014. URL `http://www.ietf.org/rfc/rfc1700.txt`.

57. Kamat, R. K., Gaikwad, P. K. and Shinde, S. A. Implementation of FPGA based firewall using behavioral synthesis. *International Journal of Computer Science and Network Security*, 2010. 1(6): 199–203.

58. Ajami, R. and Dinh, A. Design a hardware network firewall on FPGA. *2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Niagara Falls, ON, Canada. 2011. 674–678.

59. McEwan, A. A. and Saul, J. A high speed reconfigurable firewall based on parameterizable FPGA-based content addressable memories. *The Journal of Supercomputing*, 2001. 19(1): 93–103.

60. Gan, C. G. *FPGA based CAM architecture string matching for network intrusion detection*. Master dissertation. Universiti Teknologi Malaysia. 2012.

61. Attig, M., Dharmapurikar, S. and Lockwood, J. Implementation results of bloom filters for string matching. *12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2004. FCCM 2004.* Napa Valley, CA, USA. 2004. 322–323.

62. Dharmapurikar, S. and Lockwood, J. W. Fast and scalable pattern matching for network intrusion detection systems. *IEEE Journal on Selected Areas in Communications*, 2006. 24(10): 1781–1792.

63. Hieu, T. T., Thinh, T. N. and Tomiyama, S. ENREM: An efficient NFA-based regular expression matching engine on reconfigurable hardware for NIDS. *Journal of Systems Architecture*, 2013. 59(4): 202–212.

64. Sidhu, R. and Prasanna, V. K. Fast regular expression matching using FPGAs. *The 9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2001. FCCM'01.* Rohnert Park, CA, USA. 2001. 227–238.

65. Tian, X., Sun, Q., Huang, X. and Ma, Y. A dynamic online traffic classification methodology based on data stream mining. *2009 WRI World Congress on Computer Science and Information Engineering*. Los Angeles, CA, USA. 2009, vol. 1. 298–302.