MULTI-CORE 16-BIT CPUs FOR PLC PROCESSOR

GAN CHI QIAN

UNIVERSITI TEKNOLOGI MALAYSIA

MULTI-CORE 16-BIT CPUs FOR PLC PROCESSOR

GAN CHI QIAN

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

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

JULY 2022

DEDICATION

This thesis 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 thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Professor Dr. Zulfakar bin Aspar, for encouragement, guidance, critics and friendship. Without their continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Universiti Teknologi Malaysia (UTM) for funding my Master study. Librarians at UTM, Cardiff University of Wales and the National University of Singapore also deserve special thanks for their assistance in supplying the relevant literatures.

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

ABSTRACT

PLCs (Programmable Logic Controllers) are in great demand across a wide range of industries. A PLC can be used to model a controlled processing plant using a Ladder Logic Diagram (LLD). The PLC will read all the sensors, process the logic network of input and output, and provides the corresponding output signals to all actuators. LLD is widely being used to model most of the PLCs on the market today since it is user-friendly and simple to grasp by users from different levels of background. The PLC in this research will be sped up by employing a Ladder Rung Processor (LRP) architecture in a Field Programmable Gate Array (FPGA). However, LRP is only good for binary inputs. For more than a single bit input or data processing, a general purpose CPU is needed to save resources. The trend toward concurrent processing, resulting in multicore CPUs, will make it easier for a PLC processor to complete numerous tasks at the same time, enhancing performance under the demands of powerful applications and programmes. As an example of controlling a 5 axes robotic arm, a single core CPU of PLC can only control a maximum of 2 axes at the same time. Hence, a combination of LRP and multicore CPU approach can increase the throughput of a PLC processor to do concurrent processing for automation, control and robotic applications. Several architectures of multicore CPU will be investigated before determining the best solution for a PLC processor. Therefore, several performances of multicore CPU for a PLC processor will be evaluated. A RISC based CPU will be used in this research due to its simplicity while maintaining the possibility for future expansion. A 16 bits RISC CPU will be the baseline reference in this research. Several modifications will be done in terms of memory allocation and task scheduling for each of the cores will be tackled to make sure they are fully utilized in order to boost the performance to maximum. Verilog HDL language will be the preference language in the designation of multicore PLC processor. The cyclic scans frequency of PLC will be the performance benchmark that will be compared with the existing PLC in the market. To verify the architecture of multicore PLC processor, simulation on a PC and interfaced with a PLC on an FPGA will be implemented. A multicore approach in designing a complete PLC processor will increase the overall performance compared to a single core PLC processor by handling multiple processes in a manufacturing line.

ABSTRAK

PLC (Pengawal Logik yang Dapat Diprogramkan) sangat diminati di berbagai sektor. PLC adalah diprogramkan untuk memproses data input dalam Ladder Logic Diagram (LLD) dan untuk memberikan isyarat output kepada penggerak. LLD digunakan untuk memodelkan hampir semua PLC di pasaran hari ini kerana ia mesra pengguna dan mudah dipahami. PLC dalam penyelidikan ini akan dipercepat dengan menggunakan Array Gate Programmable Field (FGPA) di Ladder Rung Senibina pemproses (LRP). Walau bagaimanapun, LRP hanya baik untuk input binari. Untuk lebih daripada satu input bit atau pemprosesan data, CPU tujuan umum diperlukan untuk menjimatkan sumber. Kerana trend untuk melakukan serentak pemprosesan, multicores CPU akan membolehkan pemproses PLC melaksanakan pelbagai tugas pada masa yang sama dengan lebih mudah, meningkatkan prestasi semasa melakukan pelbagai tugas atau di bawah tuntutan aplikasi dan program yang kuat. Sebagai contoh pengendalian lengan robot 5 paksi, CPU teras tunggal PLC hanya dapat mengawal maksimum 2 paksi pada masa yang sama. Oleh itu, gabungan pendekatan LRP dan CPU multicore dapat meningkatkan throughput PLC pemproses untuk melakukan pemprosesan serentak untuk aplikasi automasi, kawalan dan robot. Beberapa seni bina CPU multicore akan disiasat sebelum menentukan penyelesaian terbaik untuk pemproses PLC. Oleh itu, beberapa persembahan CPU multicore untuk pemproses PLC akan dinilai. CPU berasaskan RISC akan digunakan dalam ini penyelidikan kerana kesederhanaannya sambil mengekalkan kemungkinan pengembangan masa depan. CPU RISC 16 bit akan menjadi rujukan asas dalam penyelidikan ini dan beberapa pengubahsuaian akan dilakukan dari segi peruntukan memori dan penjadualan tugas untuk setiap inti akan ditangani untuk memastikannya digunakan sepenuhnya untuk meningkatkan prestasi hingga maksimum. Bahasa Verilog HDL akan menjadi bahasa pilihan dalam sebutan multicore Pemproses PLC. Kekerapan imbasan siklik PLC akan menjadi penanda aras prestasi yang akan dibandingkan dengan PLC yang ada di pasaran. Untuk mengesahkan seni bina pemproses PLC multicore, simulasi pada PC dan dihubungkan dengan PLC pada FPGA akan dilaksanakan. Pendekatan multicore dalam merancang PLC lengkap pemproses akan meningkatkan prestasi keseluruhan berbanding dengan pemproses PLC teras tunggal dengan mengendalikan beberapa proses dalam barisan pembuatan.

TABLE OF CONTENTS

TITLE			PAGE
DECLARATION			**iii**
DEDICATION			**iv**
ACKNOWLEDGEMENT			**v**
ABSTRACT			**vi**
ABSTRAK			**vii**
TABLE OF CONTENTS			**viii**
LIST OF TABLES			**xi**
LIST OF FIGURES			**xii**
LIST OF ABBREVIATIONS			**xv**
LIST OF SYMBOLS			**xvi**
LIST OF APPENDICES			**xvii**
CHAPTER 1	**INTRODUCTION**		**1**
1.1	Problem Statement		1
1.2	Research Goal		2
	1.2.1 Research Objectives		2
1.3	Scope		2
1.4	Summary		2
CHAPTER 2	**LITERATURE REVIEW**		**5**
2.1	Introduction		5
	2.1.1 State-of-the-Arts		5
		2.1.1.1 A Comparison and Performance Evaluation of FPGA Soft-cores for Embedded Multi-core Systems	5
		2.1.1.2 Exploring Embedded Symmetric Multiprocessing with Various On-Chip Architectures	9
		2.1.1.3 FPGA-Based Multi-Core MIPS Processor Design	11

2.1.1.4 IEC 61131-3 Compliant PLC Structure Based on FPGA Multi-core Solution 13

2.1.1.5 The Effect of Multi-core Communication Architecture on System Performance 16

2.1.1.6 Towards Energy-Efficient Heterogeneous Multicore Architectures for Edge Computing 18

2.1.1.7 Performance Analysis of a Multicore based LEON3 integrating a RTOS 20

2.1.1.8 An Architecture of the RISC Processor for Programmable Controllers 21

2.1.1.9 LLD Processing Speed 25

2.1.1.10 A Translation Method of Ladder Diagram on PLC with Application to a Manufacturing Process 26

2.1.1.11 A BASIC Language Compiler for PLC Applications 29

2.1.1.12 Modeling of a Ladder Logic Processor for High Performance Programmable Logic Controller 32

2.1.1.13 IC Design for Ladder Rung Processor 37

2.2 Research Gap 40

1 41

Mariem Makni; Mouna Baklouti; Smail Niar; Mohamed Wassim Jmal; Mohamed Abid 41

A Comparison and Performance Evaluation of FPGA Soft-cores for Embedded Multi-core Systems 41

2 41

Gorker Alp Malazgirt; Bora Kiyan; Deniz Candas; Kamil Erdayandi; Arda Yurdakul 41

Exploring Embedded Symmetric Multiprocessing with Various On-Chip Architectures 41

2.3 Summary of Existing Architecture 45

CHAPTER 3 RESEARCH METHODOLOGY 47

3.1 Introduction 47

3.1.1 Project Planning 48

3.1.2 Design Methodology 49

3.1.2.1 Verilog compilation and simulation 49

3.1.2.2 Block diagram of the design architecture 49

3.1.2.3 AVR Architecture 50

3.1.2.4 Instruction set for AVR Processor 52

3.1.2.5 Ladder Logic Diagram (LLD) 56

3.1.2.6 Shared Memory 57

3.1.2.7 Multicore Design Architecture 60

3.1.2.8 Example Application for multicore Design approach**Error! Bookmark not defined.**

3.2 Gantt Chart 63

CHAPTER 4 RESULTS AND DISCUSSION 64

4.1 Compilation of Verilog design for multicore architecture 64

4.2 Waveform simulation on multicore AVR processors 65

4.2.1 Multicore AVR processors request access to shared memory 65

4.2.2 PLC processors request access to shared memory 67

CHAPTER 5 CONCLUSIONS AND FUTURE WORK 69

5.1 Research Outcomes 69

5.2 Future Works 69

REFERENCES 69

LIST OF TABLES

TABLE NO.	TITLE	PAGE
Table 1	Research gaps among all the review papers.	40

LIST OF FIGURES

FIGURE NO. **TITLE** **PAGE**

Figure 1: Microblaze MPPSoC design. 7

Figure 2: LEON3 MPPSoC design [16]. 8

Figure 3: Block scheme of the shared coarse-grained reconfigurable accelerator. **Error! Bookmark not defined.**

Figure 4: High-level view of the heterogeneous platform [21].**Error! Bookmark not defined.**

Figure 5: Example topology which is introduced different SMP architectures [23]. 10

Figure 6: MIPS processor basic pipeline [28]. 12

Figure 7: R-type, I-type & J-type instruction format. 12

Figure 8: Architecture of Programmable Logic Controller. 14

Figure 9: Structure of Logic Unit. 15

Figure 10: Proposed memory layout of a multicore microcontroller with two processor cores. **Error! Bookmark not defined.**

Figure 11: Federated architecture and IMA architecture.**Error! Bookmark not defined.**

Figure 12: Cache locking framework overview. **Error! Bookmark not defined.**

Figure 13: Fast Simplex Link (FSL) Bus Block Diagram [46]. 17

Figure 14: Eight-nodes in star topology with connections. 18

Figure 15: Multi-threaded management approach. (a) Memory organization. (b) Task creation. (c) Task execution. 20

Figure 16: Parallel architecture scheme. **Error! Bookmark not defined.**

Figure 17: Internal architecture of Android platform. **Error! Bookmark not defined.**

Figure 18: The proposed system architecture. **Error! Bookmark not defined.**

Figure 19: Program execution in the CPU using the dual-purpose scheduler's blocking and non-blocking I/O.**Error! Bookmark not defined.**

Figure 20: The operating principle of Round Robin and Fixed Priority algorithms. 21

Figure 21: Functional Block Diagram of PLCRISC. 23

Figure 22: Instruction set formats of PLCRISC. 24

Figure 23: Frequency of ladder instructions. 25

Figure 24: Block diagram of the proposed method. 27

Figure 25: LD for automotive process. 28

Figure 26: LLD for a simple safety circuit. 30

Figure 27: LLD equivalent to IL. 30

Figure 28: Converting BASIC language into mnemonic code. 31

Figure 29: Architecture of proposed ladder logic processor [62]. 33

Figure 30: A 4X4 Ladder Rung Block. 34

Figure 31: I/O diagram of Ladder Step Unit. 34

Figure 32: Logic diagram of Ladder Step Unit. 35

Figure 33: Functional table of LSU. 35

Figure 34: Example LLD mapped to LRB. 36

Figure 35: LRC Configuration table for LLD in figure 34. 36

Figure 36: Multiple Barrel Shifters feeding data to LRB. 37

Figure 37: Working flow of the system. 39

Figure 38: Pipelined RISC Architecture. **Error! Bookmark not defined.**

Figure 39: Instruction Format of RISC. **Error! Bookmark not defined.**

Figure 40 Block diagram of the proposed design architecture. 49

Figure 41: Accumulator Operation Instruction. 50

Figure 42: Data Transfer Instruction. 51

Figure 43: Branch Instruction. 52

Figure 44: Subroutine Instruction. 53

Figure 45: Other Instruction. 54

Figure 46: 16-bits program counter. 55

Figure 47 Program Memory Map. 55

Figure 48: RAM Configuration. **Error! Bookmark not defined.**

Figure 49: 16-bits Data Pointer. **Error! Bookmark not defined.**

Figure 50 16-bits Accumulator. **Error! Bookmark not defined.**

Figure 51: 16-bits Accumulator. **Error! Bookmark not defined.**

Figure 52: Basic terms in LLD. 56

Figure 53: Logic relationship and its Boolean equation. 57

Figure 54: Round Robin algorithm to control shared memory. 57

LIST OF ABBREVIATIONS

ANN	-	Artificial Neural Network
GA	-	Genetic Algorithm
PSO	-	Particle Swarm Optimization
MTS	-	Mahalanobis Taguchi System
MD	-	Mahalanobis Distance
TM	-	Taguchi Method
UTM	-	Universiti Teknologi Malaysia
XML	-	Extensible Markup Language
ANN	-	Artificial Neural Network
GA	-	Genetic Algorithm
PSO	-	Particle Swarm Optimization

LIST OF SYMBOLS

δ	-	Minimal error
D, d	-	Diameter
F	-	Force
v	-	Velocity
p	-	Pressure
I	-	Moment of Inersia
r	-	Radius
Re	-	Reynold Number

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

A single core PLC processor has limited capacity to accomplish numerous tasks at the same time. As an example of controlling of 5 axes robotic arm, a single core CPU of PLC can only control a maximum of 2 axes at the same time before a significant performance degradation is noticeable. However, an LRP is processing Ladder Logic Diagram (LLD) in parallel. Hence, an LRP is very fast, but it is very high resources consumption while has limited data processing. Hence, a combination of LRP and multicore CPU approach can increase the throughput of a PLC processor to do concurrent processing for automation, control and robotic applications.

Increasing the number of PLCs in a processing plant will lead to increase of data communication overhead, manpower cost and hardware cost. Increasing of data communication overhead between PLCs, may increase the chances of data loss, delay or even may indirectly affect the performance of PLC. More PLCs means more programs to be modelled and hence more manpower is needed to focus on specific PLCs. In some cases, different PLC model may be used which also increase design complexity. Hardware cost increase also means more money is needed to buy more PLCs, more space, power, wiring installation and related hardware implementation.

Compared to others control system, such as relay system, digital/analog control system and computer control system, PLC showed up to be a better choice due to its reliability and robustness. Hence, a higher performance PLC architecture is needed to speed up multiple executions process.

1.2 Research Goal

1.2.1 Research Objectives

The objectives of the research are:

(a) To propose a multi-core 16-bit CPUs for PLC Processor

(b) To design memory sharing mechanism for data sharing between multicores and PLC processor.

1.3 Scope

- A RISC-based CPU will be employed for Multi-core 16-bit PLC processor
- Focus on SISD (Single Instruction Single Data) architecture since PLC data is small
- Design of multicore PLC processor will be coded in Verilog HDL.
- Intel Quartus II design and simulation tools will be used.
- Number of multi-core CPU is limited to two units.
- Several simple PLC applications on Ladder Logic Diagram will be built to test the design.

1.4 Summary

This report consists of 5 chapters, Chapter 1 discussed the introduction, objectives and scopes of the project. Chapter 2 discussed about the state of the arts.

Chapter 3 discussed about the design methodology. Chapter 4 discussed the results and chapter 5 showed the conclusion and future work for the project.

REFERENCES

[11] Meilander, W. C., Baker, J. W. and Jin, M. 2003. Importance of SIMD Computation Reconsidered. In International Parallel and Distributed Processing Symposium, 2003.

[12] M. Baklouti, Ph. Marquet, M. Abid and JL. Dekeyser. A design and an implementation of a parallel based SIMD architecture for SoC on FPGA. In Proc. DASIP, 2008.

[13] P. Yiannacouras , J. G. Steffan and J. Rose, ”Exploration and customization of FPGA-based soft processors”, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 26, no. 2, pp. 266-277, 2007.

[14] Xilinx. Microblaze processor reference guide embedded development kitedk 7.1i. 2012.

[15] LEON3 Processor Users Manual XT Edition, Gaisler Research, October, 2008.

[16] M. Makni, M. Baklouti, S. Niar, M. W. Jmal and M. Abid, "A comparison and performance evaluation of FPGA soft-cores for embedded multi-core systems," 2016 11th International Design & Test Symposium (IDT), 2016, pp. 154-159, doi: 10.1109/IDT.2016.7843032.

[17] World Health Organization. Cardiovascular diseases. http://www.who.int/topics/cardiovascular diseases/en, 2016.

[18] M. Bingfeng et al. Exploiting loop-level parallelism on coarse-grained reconfigurable architectures using modulo scheduling. In Proc. DATE, pages 296–301, 2003.

[19] G. Ansaloni et al. EGRA: A coarse grained reconfigurable architectural template. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 19(6):1062–1074, 2011.

[20] R. Braojos et al. Hardware/software approach for code synchronization in low-power multi-core sensor nodes. In Proc. DATE, pages 1–6. IEEE, 2014.

[21] L. Duch, S. Basu, R. Braojos, D. Atienza, G. Ansaloni and L. Pozzi, "A multi-core reconfigurable architecture for ultra-low power bio-signal analysis," 2016 IEEE Biomedical Circuits and Systems Conference (BioCAS), 2016, pp. 416-419, doi: 10.1109/BioCAS.2016.7833820.

[22] “Intel ark product information,” http://ark.intel.com/, accessed: 2015-05-30.

[23] G. A. Malazgirt, B. Kiyan, D. Candas, K. Erdayandi and A. Yurdakul, "Exploring Embedded Symmetric Multiprocessing with Various On-Chip Architectures," 2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing, 2015, pp. 1-8, doi: 10.1109/EUC.2015.19.

[24] L. Wang, J. Tao, G. V. Laszewski, and H. Marten, "Multicores in Cloud Computing: Research Challenges for Applications," Journal of computers, Vol. 5, No. 6, pp. 958-964, 2010.

[25] A. Ashok, and V. Ravi, "ASIC Design of MIPS Based RISC Processor for High Performance", Conference Paper, pp.263-269, 2017.

[26] S. B. Ritpurkar, M. N. Thakare, and G. D. Korde, "Synthesis and Simulation of a 32Bit MIPS RISC Processor using VHDL", IEEE International Conference on Advances in Engineering & Technology Research, 2014.

[27] P.S. Kelgaonkar and S. Kodgire, "Design of 32 Bit MIPS RISC Processor Based on Soc", International Journal of Latest Trends in Engineering and Technology, Vol.6, No. 3, pp.446-450, 2016.

[28] A.S. Radhamani, and E. Baburaj," Network Traffic Monitoring and Control for Multi core processors in cloud computing applications", International Journal of Computer Information Systems and Industrial Management Applications, Vol., 5 pp. 557-563, 2013.

[29] S. S. Omran, and A. J. Ibada, "FPGA Implementation of MIPS RISC Processor for Educational Purposes". Journal of Babylon University/Pure and Applied Sciences, Vol.24, No.7, PP. 5471 - 5415 2016.

[30] Al-sudany, Sarah & Al-Araji, Ahmed & Mohammed Saeed, Bassam. (2021). FPGA-Based Multi-Core MIPS Processor Design. 21. 16-35. 10.33103/uot.ijccce.21.2.2.

31] A. Milik. High Level Synthesis – Reconfigurable Hardware Implementation of Programmable Logic Controller, PDeS'06, Brno, Czech Republic, 14-16 Feb. 2006, pp. 138-143. 2006.

32] J. Mocha and D. Kania, Hardware Implementation of a Control Program in FPGA Structures, Electrical Review, vol. 88, no. 12/2012, 95-100 (in polish).

[33] M. Chmiel, J. Mocha, E. Hrynkiewicz, and A. Milik, Central Processing Units for PLC implementation in Virtex-4 FPGA, Proc. of the 18th IFAC World Congress, August 28-September 2001, Milano, Italy.

[34] E. Hrynkiewicz and M. Chmiel, Programmable Logic Controller - Basic Structure and Idea of Programming, Electrical Review, R.88 nr 11b/2012, pp. 98-101.

[35] E. Hrynkiewicz and M. Chmiel, About Programmable Logic Controller - step by step, Electrical Review, R.88 nr 9a/2012, pp. 303-307, 2012.

[36] T. Klopot, P. Laszczyk, K. Stebel, and J. Czeczot, Flexible function block implementation of the balance-based adaptive controller as the potential alternative for PID-based industrial applications. Transactions of the Institute of Measurement and Control, Volume: 36. Issue: 8. Pages: 1098-1113, 2014.

[37] P. Chodorowski The core of the PLC CPU built using FPGA. Unpublished Master's thesis. Gliwice, 2015.

[38] Digilent, Nexys4 DDRTMFPGA Board Reference Manual. Digilent Inc. - Digital Design Engineer's Source. Available online at http://digilentinc.com/Data/Products/NEXYS4DDR/Nexys4- DDR_rm.pdf.

[39] Xilinx Inc. 7 Series FPGAs Overview. Available online at http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf, 2013.

[40] Siemens AG, S7-200 Programmable Controller System Manual. Germany, Nurnberg, 2008.

[41] Siemens AG, S7-300 Instruction List. Germany, Nurnberg, 2010.

[42] P. Chodorowski and M. Chmiel, "IEC 61131-3 compliant PLC structure based on FPGA multi-core solution," 2016 International Conference on Signals and Electronic Systems (ICSES), 2016, pp. 237-242, doi: 10.1109/ICSES.2016.7593858.

[43] Philipp Jungklass and Mladen Berekovic. "Intercore-Kommunikation fr Multicore-Mikrocontroller".In: Tagungsband Embedded Software Engineering Kongress 2018. 2018.

[44] P. Jungklass and M. Berekovic, "MemOpt: Automated Memory Distribution for Multicore Microcontrollers with Hard Real-Time Requirements," 2019 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC), 2019, pp. 1-7, doi: 10.1109/NORCHIP.2019.8906914.

[45] J. -B. Lefoul, A. T. Aurora Dugo, F. Magalhaes, G. Nicolescu, D. Assal and N. Ulysse, "Simulator-Based Framework towards Improved Cache Predictability for Multi-Core Avionic Systems," 2020 Spring Simulation Conference (SpringSim), 2020, pp. 1-12, doi: 10.22360/SpringSim.2020.HPC.002.

[46] Xilinx: XAPP529: Connecting Customized IP to the MicroBlaze Soft Processor Core Using the Fast Simplex Link (FSL) Channel. 2004

[48] A. Gamatie, G. Devic, G. Sassatelli, S. Bernabovi, P. Naudin and M. Chapman, "Towards Energy-Efficient Heterogeneous Multicore Architectures for Edge Computing," in IEEE Access, vol. 7, pp. 49474-49491, 2019, doi: 10.1109/ACCESS.2019.2910932.

[49] T. Xu, P. Cockshott and S. Oehler, "Acceleration of Stereo-Matching on Multi-core CPU and GPU," 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS), 2014, pp. 108-115, doi: 10.1109/HPCC.2014.22.

[50] The Multi2Sim Simulation Framework. http://www.multi2sim.org

[51] Pricopi, Mihai & Mitra, Tulika. (2014). Task Scheduling on Adaptive Multi-Core. Computers, IEEE Transactions on. 63. 2590-2603. 10.1109/TC.2013.115.

[52] S. Lee et al., “Characterizing Memory Write References for Efficient Management of PCM and DRAM Memory,” IEEE MASCOTS, 2011.

[53] X. Wu and A. Reddy, “SCMFS: a file system for storage class memory,” IEEE Conference on Supercomputing (SC), 2011.

[54] E. Lee, Y. Kim and H. Bahn, "QoS Management of Real-Time Applications in NVRAM-Based Multi-Core Smartphones," 2014 International Conference on Information Science & Applications (ICISA), 2014, pp. 1-4, doi: 10.1109/ICISA.2014.6847452.

[55] Y.Taylon, “Performance and Memory Space Optimizations for Embedded Systems”. Electrical Engineering and Computer Science. 2011.

[56] A. Kchaou, W. EL Hadj Youssef and R. Tourki, "Performance Analysis of a Multicore Based LEON3 Integrating a RTOS," 2018 15th International Multi-Conference on Systems, Signals & Devices (SSD), 2018, pp. 858-864, doi: 10.1109/SSD.2018.8570377.

[57] I. Warnock, Programmable controllers - operation and application, Prentice Hall, 1988.

[58] J. W. Davidson, J. R. Rabung, D. B. Whalley, "Relating Staticand Dynamic Machine Code Measurements," IEEE Trans. Computers, vol. 41, no. 4, APRIL 1992.

[59] Kyeong-Hoon Koo, Gab Seon Rho and Wook Hyun Kwon, "An architecture of the RISC processor for programmable controllers," Proceedings of IECON'94

- 20th Annual Conference of IEEE Industrial Electronics, 1994, pp. 1179-1183 vol.2, doi: 10.1109/IECON.1994.397959.

[60] Kim, Hyung Seok & Kwon, Wook-Hyun & Chang, Naehyuck. (1999). Translation method for ladder diagram with application to a manufacturing process. Proceedings - IEEE International Conference on Robotics and Automation. 1. 793 - 798 vol.1. 10.1109/ROBOT.1999.770071.

[61] Zulfakar Aspar and Nurul Huda Abd Rahman, " A BASIC Language Compiler for PLC Applications," 6th International Conference on Electrical, Control & Computer Engineering 2021 (InECCE 2021), Kuantan, Malaysia, August 2021.

[62] Z Aspar and M Khalil-Hanim, “Modeling of a ladder logic processor for high-performance programmable logic controller”, Third Asia International Conference on Modelling & Simulation, Indonesia, 2009, pp. 572-577.

[63] Chandran, V. & Sulthana, M. & M G, Sumithra & Suriya, M.. (2019). Design of a 16-Bit Harvard Structure RISC Processor in Cadence 45nm Technology. 173-178. 10.1109/ICACCS.2019.8728479.

[64] Yap Kok Soon, "A DATA PROCESSOR FOR A PROGRAMMABLE LOGIC CONTROLLER PROCESSOR"