INTEGRATED CIRCUIT DESIGN OF A MULTIPLIER-ACCUMULATOR FOR CONVOLUTIONAL NEURAL NETWORK BASED ON CARRY-SAVE ADDER ARCHITECTURE

LEE MEI XIANG

UNIVERSITI TEKNOLOGI MALAYSIA

INTEGRATED CIRCUIT DESIGN OF A MULTIPLIER-ACCUMULATOR FOR CONVOLUTIONAL NEURAL NETWORK BASED ON CARRY-SAVE ADDER ARCHITECTURE

LEE MEI XIANG

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Philosophy

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

DECEMBER 2021

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to my supervisor Dr. Ab Al-Hadi Ab Rahman for giving me continuous guidance and opportunity in this research. His vision and vast knowledge in this field really motivates and inspires me. He had taught me the correct and organized way to run the simulation and analysis using Synopsis Tools. I am really grateful for all the invaluable effort and patience in guiding me throughout this whole research.

Next, I would like to say thanks to my family for the continuous love and support for me in completing this research. Their encouragement and strength is what's keeping me going everyday.

I am really grateful to Silterra and CEDEC by providing and arranging the Synopsys Tools training and the middle term clinic for me. It gets me familiarized more in utilizing the software correctly. The kick-start training enables me to complete my pre-layout and post-layout designs easily.

I would also like to take this opportunity to thank the creators of the utmthesis LaTeXProject for making my thesis writing and compiling process effortless. It is so convenient and straightforward to utilise without getting so worked up in aligning and formatting issues in my thesis. As a result, I can solely focus all my time and energy on the content of the thesis.

I want to give my thanks to my supervisor, Prof. Madya Muhammad Mun'im Bin Ahmad Zabidi for his great teaching and support in my research from the beginning until the end. His advice and knowledge greatly led me to the correct direction in accomplishing this research.

Finally, I thank all the people who have directly or indirectly supported me throughout this research and making it a success in the end.

ABSTRACT

Convolutional Neural Networks (CNNs) are hierarchical biologically-inspired models that may be taught to perform a variety of detection, identification, and segmentation tasks. The key processing requirements in the CNNs process are the multiplier-accumulator (MAC) operations in the convolution layer. A perfect fusion of various multipliers and adders would yield an ideal MAC for CNNs' convolution layer. Besides this, different kernel mask sizes are required throughout the convolution layer in CNNs, depending on demand. In general, each size of kernel mask requires a unique MAC architectural configuration, thus lengthening the time spent in research and development. To overcome this problem, a flexible MAC design has been developed that allows users to choose between different sizes depending on the requirements of the CNNs. This method, known as Selective Kernel Size, may activate kernel sizes from 1×1 to 7×7. This thesis also proposes a new MAC architecture called Mulitplier-Accumulator with Carry-Save-Adder (MAC_{CSA}) to improve the MAC performance by efficiently computing the sum of three or more bits of input. Each proposed design is synthesized to Silterra 180 nm technology, and the time, power consumption, and cell area are all compared. Selective Kernel Size architecture demonstrates a substantial gain in terms of cell area and power from 56% to 80% when compared to mixing different sizes of MAC design in specific CNNs network, with some degradation in time delay. When comparing the classical addition structure to the proposed MAC with the CSA structure, results show that while the MAC with CSA is only slightly faster than the classical MAC, the power and cell area are improved by 4% to 10%.

ABSTRAK

Rangkaian Neural Konvolusi (CNN) ialah model hierarki yang diilhamkan secara biologi yang boleh diajar untuk melaksanakan pelbagai tugas pengesanan, pengenalan, dan pembahagian. Keperluan pemprosesan utama dalam proses CNN ialah operasi darab-tambah (MAC) dalam lapisan konvolusi. Gabungan sempurna pelbagai pendarab dan penambah akan menghasilkan MAC yang ideal untuk lapisan konvolusi CNN. Selain itu, topeng kernel saiz berbeza diperlukan di seluruh lapisan konvolusi dalam CNN, bergantung kepada permintaan. Secara amnya, setiap saiz topeng kernel memerlukan konfigurasi seni bina MAC yang unik, dengan itu memanjangkan masa yang diperlukan dalam penyelidikan dan pembangunan. Untuk mengatasi masalah ini, reka bentuk MAC yang fleksibel telah dibangunkan yang membolehkan pengguna memilih antara saiz yang berbeza bergantung kepada keperluan CNN. Kaedah ini, dikenali sebagai Saiz Kernel Selektif, boleh mengaktifkan saiz kernel dari 1×1 hingga 7×7. Tesis ini juga mencadangkan seni bina MAC baharu yang dipanggil Mulitplier-Accumulator dengan Carry-Save-Adder (MAC_{CSA}) untuk meningkatkan prestasi MAC dengan menggabungkan tiga atau lebih bit input untuk operasi penambahan. Setiap reka bentuk yang dicadangkan disintesis kepada teknologi Silterra 180nm, dan masa, penggunaan kuasa, dan keluasan sel semuanya dibandingkan. Seni bina Saiz Kernel Selektif menunjukkan peningkatan yang besar dari segi keluasan dan kuasa sel daripada 56% hingga 80% jika dibandingkan dengan reka bentuk MAC yang berbeza dalam rangkaian CNN tertentu, dengan sedikit kemerosotan dari segi masa. Apabila membandingkan struktur penambahan klasik kepada MAC yang dicadangkan dengan struktur CSA, keputusan menunjukkan bahawa walaupun MAC dengan CSA adalah lebih pantas sedikit daripada MAC klasik, kuasa dan kawasan sel adalah lebih baik sebanyak 4% hingga 10%.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ASIC	–	Application Specific Integrated Circuit
ASM	–	Algorithm State Machine
AM	–	Array Multiplier
BKA	–	Brent-Kung Adder
BM	–	Booth Multiplier
C_i	–	Carry Signal
C_{in}	–	Carry in
C_{out}	–	Carry out
CAVM	–	Constant Array Vector Multiplication
CLA	–	Carry Look Ahead Adder
CSA	–	Carry Save Adder
CSDC	–	Computation Sharing Differential Coefficient
CNNs	–	Convolutional Neural Networks
CPG	–	Carry Propagation Generator
$control_m$	–	control value from CU to control the high for the Kernel matrix size
$control_r$	–	control value from CU to control the length for the Kernel matrix size
CU	–	Control Unit
DSP	–	Digital Signal Processing
DU	–	Data Unit
EDA	–	Electronic Design Automation
FA	–	Full Adder
FCO	–	Fundamental Carry Operator

FWBM	–	fixed-width Booth multiplier
FIR	–	Finite Impulse Response
FPGA	–	Field Programmable Gate Arrays
G	–	Generator
G_i	–	Generation Signal
HA	–	Half Adder
IC	–	Integrated Circuit
ICC	–	Integrated Circuit Compiler
KSA	–	Kogge-Stone Adder
LSB	–	Least Significant Bit
LUT	–	Lookup Table
MAC	–	Multiplier-Accumulator
MAC-WT-CLA	–	Multiplier-Accumulator normal structure with Wallace-Tree Multiplier combine and Carry Look Ahead Adder
MAC-WTK-KSA	–	Multiplier-Accumulator normal structure with Wallace-Tree Multiplier combine and Kogge-Stone Adder
MAC-WTC-CLA	–	Multiplier-Accumulator normal structure with WT-CLA combine and Carry Look Ahead Adder
MAC-WTC-KSA	–	Multiplier-Accumulator normal structure with WT-CLA combine and Kogge-Stone Adder
MAC-WTK-CLA	–	Multiplier-Accumulator normal structure with WT-KSA combine and Carry Look Ahead Adder
MAC-WTK-KSA	–	Multiplier-Accumulator normal structure with WT-KSA combine and Kogge-Stone Adder
MAC_{CSA}	–	Multiplier-Accumulator with Carry Save Adder structure after multiplier product
MAC_{CSA}-WT-CLA	–	MAC_{CSA} structure with Wallace-Tree Multiplier combine and Carry Look Ahead Adder

MAC_{CSA}-WT-KSA– MAC_{CSA} structure with Wallace-Tree Multiplier combine and Kogge-Stone Adder

MAC_{CSA}-WTC-CLA– MAC_{CSA} structure with WT-CLA combine and Carry Look Ahead Adder

MAC_{CSA}-WTC-KSA– MAC_{CSA} structure with WT-CLA combine and Kogge-Stone Adder

MAC_{CSA}-WTK-CLA– MAC_{CSA} structure with WT-KSA combine and Carry Look Ahead Adder

MAC_{CSA}-WTK-KSA– MAC_{CSA} structure with WT-KSA combine and Kogge-Stone Adder

MUX – Multiplexer

OGA – Orthogonal Genetic Algorithm

P – Propagator

P_i – Propagation Signal

PG – Propagator and Generator

PG-4 – 4-bit of PG block

POT – Powers of Two

PPA – performance, power and area

PPC – Various Carry Generator

PPC-4 – 4-bit of Various Carry Generator

PPG – Partial Product Generator

PPRT – Partial Product Reduction Tree

QCA – Quantum-dot Cellular Automata

RCA – Ripple-Carry Adder

RMACA – Reconfigurable MAC Systolic Array Architecture

RTL – Register Transfer Level

$size_m$ – input value to control the high for the Kernel matrix size

$size_r$ – input value to control the length for the Kernel matrix size

Sm	–	carry value of size$_m$ after one clock cycle
Sr	–	carry value of size$_r$ after one clock cycle
STA	–	Static Timing Analysis
Sum$_i$	–	Sum Signal
TEC	–	Truncation Error Compensation
VLSI	–	Very-Large-Scale Integration
VHDL	–	Very High Speed Integrated Circuit Hardware Description Language
WT	–	Wallace-Tree Multiplier
WT-CLA	–	Wallace-Tree Multiplier with Carry Look Ahead Adder at Carry Propagation Generator
WT-KSA	–	Wallace-Tree Multiplier with Kogge-Stone Adder at Carry Propagation Generator

LIST OF SYMBOLS

$\oplus$	–	xor gate description
$\leq$	–	less and equal description
$\geq$	–	less and equal description
$\cdot$	–	and gate description
$\mid$	–	or gate description
$+$	–	symbol of adding process
$\times$	–	symbol of multiply process
$\%$	–	symbol of percentage
μ	–	symbol of unit prefix in the metric system denoting a factor of 10^{-6}
$\checkmark$	–	check-mark for simulation results compare to theoretical

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

Convolution Neural Networks (CNNs) is the fundamental technology that is utilised in image processing task such as image classification. In order to manage image classification task, a computer or system must be able to search for low-level characteristics in the kernel mask, such as edges and curves. The CNNs system can output the input image of a probability class to provide it the best description based on the mapping of the original input image with kernel mask. All of these processes will eventually add up to additional learning concepts spread across a network of convolution layers in CNNs.

When traversing through the convolution layer in CNN, a bundle of Multiple Accumulation (MAC) process blocks of varying sizes appear. A new MAC architecture for CNNs application will be introduced in this work. MAC is basic component used in CNNs to perform the multiply-accumulation process, which includes multiple registers as a Lookup Table (LUT) to store the feature maps and kernels for image classification class. For example, with 3×3 kernel mask used, the values of first 3 columns are multiplied by the first value of the kernel mask used respectively and accumulated for a value. The operation will be repeated for the remaining values in the kernel mask table. The output result is primarily about the MAC between the pixels of the input image with kernel size, which is significant in the convolution layer process. Because it is the primary function of convolution layers, the type of MAC utilised has a direct impact on the performance of the CNNs process.

In this study, the MAC is developed with a control unit (CU) block that allows it to traverse any Kernel Size used in different CNNs networks without changing the MAC architecture or arrangement. As a result, this MAC design increases its flexibility and covers practically all significant kernel mask sizes to suit the requirements of every sort of CNNs network. The flexibility of the MAC architecture designed also contributes

significantly to the innovation of CNN networks. The Selective Kernel Size algorithm, which is employed in MAC, will be explained in further detail in Chapter 3.

MAC is defined in two architectures throughout this work. This study's MAC_{CSA} architecture will present one with classical addition and another MAC architecture. Before the accumulator, MAC_{CSA} will replace the adding component of MAC. A normal adder can only add two groups of numbers at a time, whereas CSA can add up to three groups of numbers at once. This new design architecture is described in detail in this thesis.

To identify an effective MAC, simulation of many types of multiplier and adder is required to select the best architecture for MAC design. Dynamic Power, Leakage Power, Cell Area, and Time Delay are all important parameters to consider. For the Multiplier, it will be compared between WT, BM, and the AM benchmark. WT will have three distinct designs due to the employment of different Adders in its Carry Propagation Generator (CPG). KSA, CLA, and the default adder in EDA Tool are among the adders utilised. On the other hand, adder will be compared in the simulation between KSA, CLA, and BKA by using RCA as benchmark.

1.1 Problem Statement

The MAC is a fundamental unit block in all Digital Signal Processing (DSP) systems and plays critical functions in CNNs. Because it is a lower level component in a system, the time delay and power consumption will have a direct impact on the entire system. As a result, an efficient MAC may increase the overall performance of CNNs.

Each convolution layer in CNNs requires a different size kernel mask. According to earlier research, each size of kernel mask necessitates a separate MAC architectural configuration [1]. This lengthens the trial-and-error process of determining the best MAC size and layout of those MAC architectures for the simulation process. With the Selective Kernel Size technique, the user can quickly change the size of the MAC by entering 0 or 1 into the control unit. In other words, just one MAC

design is required for the entire convolution Layer in CNNs. Selective Kernel Size has a function in the flexibility of arrangement between each MAC when going through the invention of CNNs model.

CSA is a binary adder that can easily compute the sum of three or more bits of data. Typically, MAC for CNNs must include at least three numbers of data. In this thesis, CSA may sum three 16-bit numbers concurrently. This will undoubtedly boost MAC performance.

1.2 Objective

1. To study and compare the speed, area and power of multiplier and adder and select for MAC fusion design used in CNNs for Application Specific Integrated Circuit (ASIC) implementation.
2. To propose and implement ASIC Selective Kernel Size MAC to match the different requirement of each type of CNNs networks.
3. To propose and implement MAC_{CSA} architecture and compare with classical addition MAC architecture.

1.3 Scope of Study

1. The type of adder used in MAC:
   (a) CLA
   (b) KSA
2. The type of multiplier used in MAC:
   (a) WT with EDA Tool default adder used in CPG (WT)
   (b) WT with CLA used in CPG (WT-CLA for Multiplier, WTC for MAC)
   (c) WT with KSA used in CPG (WT-KSA for Multiplier, WTK for MAC)

3. This thesis discusses the kernel mask size of a convolution layer ranging from 1×1 matrix to 7×7 matrix.

4. The developed MAC design with 8-bit input is used in each convolution layer of CNNs using the Selective Kernel Size technique.

1.4 Contribution

The contribution of this research is as follows:-

1. Investigate various adder and multiplier architectural designs in synthesis and analysis. They are combined to discover the ideal MAC architecture design in terms of performance, power and area (PPA).

2. Redesign the classical MAC architecture so that it can be selective in the kernel matrix size utilised by the convolution layer of various types of CNNs. Only one MAC design is necessary because it can cover the majority of kernel matrix sizes utilised by the convolution layer of various types of CNNs.

3. Redesign the MAC architecture that is used in the convolution layer. The traditional MAC adds two groups of numbers before adding the preceding result. In this paper, we replace the adder with CSA, which can add up to three groups of numbers at once before accumulating with the preceding result.

1.5 Organization in Thesis

The thesis is organized as follows:-

1. The history and operation of CNNs will be covered in Chapter 2. The kernel matrix size utilised for each well-known type of CNNs will be listed. As the study's outcome is a new architecture design of MAC, the major context will be MAC used for convolution layer. Finally, a full examination of the multiplier and adder utilised in MAC architecture.

2. Chapter 3 discusses each algorithm employed, from research through design to algorithm verification. The discussion starts with how MAC performs in the Selective Kernel Size architecture and progresses to the later stages of adding multiplier and adder. The proposed working techniques for this thesis, which include all algorithmic function blocks, will be presented in depth. The classic addition MAC in Selective Kernel Size Architecture will be illustrated first, followed by the MAC_{CSA} method of selected adder and multiplier. Each method available for producing algorithm design, verify the functionality of proposed design, and simulation result of each design will also be presented.

3. The simulation results of all designs are discussed in Chapter 4 according to their level. This includes the results for the selected 16-bit adder and 8-bit multiplier, followed by the MAC architecture results for the proposed Selective Kernel Size MAC and the new MAC-CSA architecture. For each explored designs, results are presented in terms of performance, power, and area.

4. Chapter 5 provides a summary of this thesis. This includes the summary of the work that has been done, as well as the contributions. The chapter also gives a conclusion on the overall perspective of the work, and with the final section on several possible future directions for the work.

REFERENCES

1. Agarwal, S., Verma, S. and Agrawal, D. P. *Machine Intelligence and Signal Processing: Proceedings of International Conference, MISP*. Springer Nature. 2020.

2. Khan, S., Rahmani, H., Shah, S. A. A. and Bennamoun, M. *A Guide to Convolutional Neural Networks for Computer Vision*. Morgan and Claypool Publishers. 2018.

3. Shen, J., Qiao, Y., Huang, Y., Wen, M. and Zhang, C. *Towards a Multi-Array Architecture for Accelerating Large-Scale Matrix Multiplication on FPGAs*. in arxiv:1803.03790v1. Dec 2018.

4. Krohn, J., Beyleveld, G. and Bassens, A. *Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence*. Addison-Wesley Professional. 2019.

5. Simonyan, K. and Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. in arxiv preprint. 2014.

6. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. *Going Deeper with Convolutions*. IEEE conference on computer vision and pattern recognition (CVPR). 2015.

7. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. *Rethinking the Inception Architecture for Computer Vision*. IEEE conference on computer vision and pattern recognition (CVPR). 2016.

8. Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learnings*. Proceedings of the thirty-first AAAI conference on artificial intelligence. 2016.

9. He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. 770–778.

10. Cholle, F. *Deep Learning with Depthwise Separable Convolutions*. IEEE conference on computer vision and pattern recognition (CVPR). 2017.

11. Xie, S., Girshick, R., Dollár, P., Tu, Z. and He, K. *Aggregated Residual Transformations for Deep Neural Networks.* IEEE conference on computer vision and pattern recognition (CVPR). 2017.

12. Aksoy, L., Flores, P. and Monteiro, J. A tutorial on multiplierless design of FIR filters: algorithms and architectures. *Circuits, Systems, and Signal Processing*, 2014. 33(6): 1689–1719.

13. Ahmad, S. U. and Antoniou, A. *Cascade-Form Multiplierless FIR Filter Design Using Orthogonal Genetic Algorithm.* IEEE international symposium on signal processing and information technology. 2006.

14. Takahashi, Y. and Yokoyama, M. *New cost-effective VLSI implementation of multiplierless FIR filter using common subexpression elimination.* 2005 IEEE international symposium on circuits and systems, kobe. 2005.

15. Lin, T.-J., Yang, T.-H. and Jen, C.-W. *Coefficient optimization for area-effective multiplier-less FIR filters.* International conference on multimedia and expo. icme '03. proceedings (cat. no.03th8698), baltimore, md, usa. 2003.

16. Wang, Y. and Roy, K. *CSDC: A New Complexity Reduction Technique for Multiplierless Implementation of Digital FIR Filters.* IEEE transaction on circuit and systems, vol. 52, no. 9. September 2005.

17. Ye, W. B. and Yu, Y. J. *Bit-Level Multiplierless FIR Filter Optimization Incorporating Sparse Filter Technique.* IEEE transaction on circuit and systems. 2014.

18. Flores, L. A. P. and Monteiro, J. *ECHO: A Novel Method for the Multiplierless Design of Constant Array Vector Multiplication.* IEEE international symposium on circuit and systems ISCAS. 2014.

19. Ahmed, H. O., Ghoneima, M. and Dessouky, M. Concurrent MAC unit design using VHDL for deep learning networks on FPGA. *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE).* IEEE. 2018. 31–36.

20. Dettmers, T. 8-bit approximations for parallelism in deep learning. *arXiv preprint arXiv:1511.04561*, 2015.

21. Ahmed, H. O., Ghoneima, M. and Dessouky, M. Concurrent MAC unit design using VHDL for deep learning networks on FPGA. *IEEE*, 2018.

22. S. Y. Lin, C. K. T., K. H. Lin and Tseng, P. H. *Reconfigurable MAC Systolic Array Architecture Design for Three-Dimensional Convolution Neural Network.* 2020 IEEE international conference on consumer electronics - taiwan (ICCE-taiwan), taoyuan, taiwan, 2020, pp. 1-2, doi: 10.1109/icce-taiwan49838.2020.9258069. 2020.

23. E. Kwon, Y. K. and Kang, S. *Outlier-aware Time-multiplexing MAC for Higher Energy-Efficiency on CNNs.* 2019 international soc design conference (ISOCC), jeju, korea (south), 2019, pp. 119-120, doi: 10.1109/isocc47750.2019.9027750. 2019.

24. S. Lee, D. N., D. Kim and Lee, J. *Double MAC on a DSP: Boosting the Performance of Convolutional Neural Networks on FPGAs.* IEEE transactions on computer-aided design of integrated circuits and systems, vol. 38, no. 5, pp. 888-897, may 2019, doi: 10.1109/tcad.2018.2824280. 2020.

25. Tang, S. N. and Han, Y. S. *A High-Accuracy Hardware-Efficient Multiply–Accumulate (MAC) Unit Based on Dual-Mode Truncation Error Compensation for CNNs.* IEEE access, vol. 8, pp. 214716-214731, 2020, doi: 10.1109/access.2020.3040366. 2020.

26. Mavuri, P. and Velanr, B. Design and Performance Analysis of a High Speed MAC Using Different Multipliers. *2015 Fifth International Conference on Advances in Computing and Communications (ICACC)*, 2015.

27. Kelly, L. and Hai Hiung, L. Performance comparison review of 32-bit multiplier designs. *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)*, 2012.

28. Honarmand, N., M.R.Javaheri, Sedaghati-Mokhtari, N. and Afzali-Kusha, A. *Power Efficient Sequential Multiplication Using Pre-computation, ISCAS.* IEEE international symposium on circuits and systems. 2006.

29. Pachara, V. R., Cyril Prasanna, R. and Ravi, S. VLSI Design and Analysis of Multipliers for Low Power. *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2009.

30. Biswarup, M. and Aniruddha, G. Counter Based Low Power, Low Latency Wallace Tree Multiplier Using GDI Technique for On-chip Digital Filter Applications. *2019 Devices for Integrated Circuit (DevIC)*, 2019.

31. RAO, P. V. S. *Computer System Architecture*. PHI Learning Pvt. Ltd. 2008.

32. Booth., A. D. A signed binary multiplication technique. *The Quarterly Journal of Mechanics and Applied Mathematics, Volume IV, Pt. 2*, 1951.

33. Suriya, T. U. and Rani, A. A. Low power analysis of MAC using modified booth algorithm. *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE. 2013. 1–5.

34. Yen-Jen, C., Yu-Cheng, C., Shao-Chi, L. and Chun-Huo, H. A Low Power Radix-4 Booth Multiplier With Pre-Encoded Mechanism. *IEEE Access (Volume: 8)*, 2020.

35. Zabidi, M. M. A., Kamisian, I. and Ismail, I. *The Art of Digital Design*. Universiti Teknologi Malaysia. 2020.

36. S.Xing and W.W.H.Yu. *FPGA Adders: Performance Evaluation and Optimal Design*. IEEE design and test of computers, vol. 15, no. 1, pp. 24-29. January 1998.

37. Ibrahim, M., Mohammed Mosab, A., Ahmad, B. and Qasem Abu, A.-H. Cost analysis study of variable parallel prefix adders using altera cyclone IV FPGA kit. *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2017.

38. Kogge, P. M. and Stone, H. S. *A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations*. IEEE trans. on computers. August 1973.

39. Merkel, C. and Brenner, D. *8-bit Parallel Prefix Adder Using Brent Kung Tree BIST*. Eecc730. November 2008.

40. Vibhuti Dave, E. O. and Saniie, J. *Performance Evaluation of Flagged Prefix Adders for Constant Addition*. IEEE international conference. Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology. 2006.

41. Abidin, A. Z., Al Junid, S. A. M., Sharif, K. K. M., Othman, Z. and Haron, M. A. 4-bit Brent Kung parallel prefix adder simulation study using Silvaco EDA tools. *International Journal of Simulation, Systems, Science and Technology*, 2009.

42. Kai, C. C., Isaak, S. and Yusof, Y. *16-bit Fault Tolerant Sparse Kogge Stone Adder using 0.18nm CMOS Technology*. 2020 IEEE International Conference on Semiconductor Electronics (ICSE). 2020.

43. Weinberger, A. and Smith, J. A logic for high-speed addition. *National Bureau of Standards Circular*, 1958. 591: 3–12.

44. Brent, R. P. and Kung, H. T. *A regular layout for parallel adder*. IEEE trans. comput. 1982.

45. Sasamal, T. N., Singh, A. K. and Mohan, A. *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*. Springer Nature. 2019.

46. Earle, J. G. *Latched Carry-Save Adder*. IBM technical disclosure bulletin, 7 (10): 909–910. March 1965.

47. Sarvarbek, E. and Jun-Cheol, J. *Carry save adder and carry look ahead adder using inverter chain based coplanar QCA full adder for low energy dissipation*. Ph.D. Thesis. Department of Computer Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea. 2018.

48. Vikas K. Saini, S. A. and Chauhan, T. *Implementation, Test Pattern Generation, and Comparative Analysis of Different Adder Circuits*. Hindawi publishing corporation vlsi design volume 2016, article id 1260879. 2016.

49. Sarvarbek Erniyazov, J.-C. J. *Carry save adder and carry look ahead adder using inverter chain based coplanar QCA full adder for low energy dissipation*. Microelectronic engineering 211 (2019) 37–43. 2019.

50. Altera. *My First FPGA Tutorial*. Altera coorporation. July, 2008.

51. Diksha Moolchandani, S. R. S., Anshul Kumar. Accelerating CNN Inference on ASICs: A Survey. *Journal of Systems Architecture*, 4 September 2020.

52. Andrew Boutros, V. B., Sadegh Yazdanshenas. You cannot improve what you do not measure: FPGA vs. ASIC efficiency gaps for convolutional neural network inference. *ACM Trans. Reconfig. Technol. Syst. 11 (3) (2018) 20*, 2018.

53. Ian Kuon, J. R. *Measuring the gap between FPGAs and ASICs.* IEEE trans. comput.-aided des. integr. circuits syst. 26 (2) (2007) 203–215. 2007.

54. Ainy Haziyah Awab, A. A.-H. A. R. HEVC 2D-DCT Architectures Comparison for FPGA and ASIC Implementations. *TELKOMNIKA, Vol.17, No.4, August 2019, pp.791 79x*, 2019.

55. Falih S.M. Alkhafaji, M. I., Wan Z.W. Hasan and Sulaiman, N. *Robotic controller: ASIC versus FPGA-a review.* J. comput. theor. nanosci. 15 (2018) 1–25. 2018.

56. Ahmad, R. DIGITAL IC DESIGN FOR FRONT-END and SIGN-OFF USING SYNOPSYS. *Research Office Collaborative MicroElectronic Design Excellence Centre (CEDEC)*, 2014.

57. Ahmad. *R. DIGITAL IC DESIGN FOR FRONT-END and SIGNOFF USING SYNOPSYS.* Research Office Collaborative MicroElectronic Design Excellence Centre (CEDEC). 2014.

LIST OF PUBLICATIONS

1. Xiang, L.M., Mun'im Ahmad Zabidi, M., Awab, A.H. and Ab Rahman, A.A.H., 2018, July. VLSI Implmentation of a Fast Kogge-Stone Parallel-Prefix Adder. In Journal of Physics: Conference Series (Vol. 1049, No. 1, p. 012077). IOP Publishing.