Computation-in-Memory based on Memristive Devices

Du Nguyen, Hoang Anh

DOI
10.4233/uuid:ba02810b-e380-4c88-a4ed-d6bd2598ab2f

Publication date
2019

Document Version
Final published version

COMPUTATION-IN-MEMORY BASED ON MEMRISTIVE DEVICES

COMPUTATION-IN-MEMORY BASED ON MEMRISTIVE DEVICES

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on Friday, 13 September 2019 at 10:00 o'clock

by

Hoang Anh DU NGUYEN

Master of Science in Computer Engineering,
Delft University of Technology, the Netherlands
born in Danang, Vietnam.

This dissertation has been approved by the

promotor: Prof. Dr. Ir. S. Hamdioui
copromotor: Dr. Ir. M. Taouil

Composition of the doctoral committee:

Rector Magnificus,	chairman
Prof. Dr. Ir. S. Hamdioui	Delft University of Technology, promotor
Dr. Ir. M. Taouil	Delft University of Technology, copromotor
Independent members:	
Prof. Dr. Ir. K.L.M. Bertels	Delft University of Technology
Prof. Dr. H. Corporaal	Eindhoven University of Technology
Prof. Dr. F. Catthoor	Katholieke Universiteit Leuven, Belgium
Prof. Dr. M. B. Tahoori	Karlsruhe Institute of Technology, Germany
Dr. R. V. Joshi	T. J. Watson Research Center, IBM, USA
Reserved members:	
Dr. Ir. P. F. A. Van Mieghem	Delft University of Technology

Keywords: Computer architecture, resistive computing, Computation-in-Memory

Cover designed by: Van Sanh Le

ISBN 978-94-6384-060-6

An electronic version of this dissertation is available at
https://doi.org/10.4233/uuid:ba02810b-e380-4c88-a4ed-d6bd2598ab2f

Dedicated to

my father for the motivation to start this journey,
and Nhi An for the courage to end this journey.

CONTENTS

SUMMARY

In-memory computing is a promising computing paradigm due to its capability to alleviate the memory bottleneck. It has even higher potential when implemented using memristive devices or memristors with various beneficial characteristics such as nonvolatility, high scalability, near-zero standby power consumption, high density, and CMOS compatibility. Exploring in-memory computing architectures in the combination with memristor technology is still in its infancy phase. Therefore, it faces challenges with respect to the development of the devices, circuits, architectures, compilers and applications.

This thesis focuses on exploring and developing in-memory computing in terms of architectures (including classification, limited schemes of instruction set, micro-architecture, communication and controller, as well as automation and simulator), and circuits (including logic synthesis flow and interconnect network schemes).

In-Memory architecture classification and survey - We first investigate the state-of-the-art of in-memory computing and propose a classification to have an overview on both existing and unexplored architectures. The classification is based on three main criteria: computation location (i.e., where the results are produced), memory technology (i.e., the memory technology is used), and computation parallelism (i.e., the maximum parallelism level can be exploited). Based on the computation, four main classes are derived: Computation-inside-Memory Array (CIM-A) which produces results *inside* the memory and within the memory *array*, Computation-in-Memory Peripheral (CIM-P) which produces results *inside* the memory and within the *peripheries*, Computation-outside-Memory Near (COM-N) which produces results *outside* the memory and *near* the memory core, and Computation-outside-Memory Far (COM-F) which which produces results *outside* the memory and *far* from the memory core, respectively. Subsequently, we review and compare the four classes and existing architectures quantitatively. The proposed classification and survey show not only the architectures that were explored in details in this dissertation, but also potential architectures that can be explored in the future.

Architecture Level- We propose two architectures representing CIM-A and CIM-P class from the above classification. For CIM-A class, we first propose a concept of integrating computation and memory into one physical device, specifically memristive or memristor devices. This concept has potentials to alleviate the memory wall or memory bottleneck in particular, and the architecture and technology wall in general. We demonstrate the potentials of this concept using a health care application and mathematical application. Thereafter, we use this concept to perform a parallel addition using only the crossbar array which also stores the operands of the function. We show the preliminary result of this parallel adder in comparison with conventional architectures such as

multicore and GPU architecture. Subsequently, we implement the above parallel adder while taking into consideration its controller, communication and interconnect network schemes. Two implementations using two distinct logic designs are compared with a multicore architecture. The results show that the two implementations outperform multicore architecture at least two orders of magnitude in terms of performance, energy, and area combined metrics. In addition, the controller and communication pose relatively large overheads in this architectures. Even though many aspects of this architecture in particular and CIM-A class in general were explored, there are still many questions regarding to the integration between CMOS controller and memristor crossbars, the trade-off between isolation and parallelism, as well as the generation of complex and efficient functional units as building blocks, etc.

For CIM-P class, we first show the potentials of Computation-in-Memory (CIM) core (i.e., a memristor crossbar with capability to perform logical operations using peripheral circuitry). Thereafter, we propose different architectures that can integrate an arithmetic CIM core into different position of the memory hierarchy (i.e., computational cache, main memory or accelerator). Thereafter, we select the architecture that use CIM core as accelerator due to the current state of memristive devices and in-memory computing architecture. Subsequently, we investigate the potentials of this architecture using an analytical model. Finally, we build a simulation platform to port applications executing on the proposed architecture. With this, we verify our assumptions on the analytical model and explore potential applications for the proposed architecture. Both the analytical and simulation results show that the proposed architecture outperforms the conventional architecture at least one order of magnitude in terms of performance and energy. It is worth to notice that the architecture as well as simulation platform are in their infancy stage, and more efforts are required to fully utilize the architecture's potentials on big data applications.

Circuit Level - As it is essential to build basic blocks for in-memory computing architectures, we propose logic synthesis automation tools and interconnect networks to realize digital complex function on memristor crossbar. First we propose a generic synthesis framework to map a digital arithmetic function described in hardware description language (HDL) on memristor circuits. We demonstrate this framework using two case studies of 2-bit counter and 8-bit adder. As this framework is a preliminary result, efforts are still required to automate the framework and explore more complex functions. Thereafter, we propose different interconnect network schemes that can be used in a memristive circuit/design. Using a case study of parallel adder (based on CIM-A class architecture), we demonstrate three schemes including direct scheme using only *copy* operation, indirect scheme using CMOS circuits (i.e., controller), and hybrid scheme which combines the direct and indirect scheme. The results show that hybrid scheme provides the highest performance and lowest energy consumption, hence, should be considered to be used in CIM architecture. It is worth to emphasize that the proposed solution are roughly evaluated and more detail implementations are required to realize these solutions in certain designs/systems.

SAMENVATTING

Gegevensverwerking-in-geheugen is een veelbelovend computerparadigma vanwege de mogelijkheid om het geheugenknelpunt te verlichten. Het heeft een nog hogere potentie wanneer het wordt geïmplementeerd met behulp van geheugenresistieve elementen of geheugenweerstanden met verscheidene voordelige kenmerken zoals niet-vluchtigheid, hoge schaalbaarheid, bijna nul standby-stroomverbruik, hoge dichtheid en CMOS-compatibiliteit. Het verkennen van gegevensverwerking-in-geheugenarchitecturen in combinatie met geheugenweerstandstechnologie bevindt zich nog in de kinderschoenen. Daarom staat het voor uitdagingen met betrekking tot de ontwikkeling van de elementen, schakelingen, architecturen, compilers en applicaties.

Dit proefschrift richt zich op het verkennen en ontwikkelen van gegevensverwerking-in-geheugen op het gebied van architecturen (waaronder classificatie, beperkte schema's van instructieset, micro-architecturen, communicatie en controllers, evenals automatisering en simulators), en schakelingen (waaronder logische-synthesestappenplan en verbindingsnetwerkschema's).

Gegevensverwerking-in-geheugenarchitecturenclassificatie en –overzicht - We onderzoeken eerst de state-of-the-art van gegevensverwarking-in-geheugen en stellen een classificatie voor om een overzicht te hebben van zowel bestaande als niet eerder onderzochte architecturen. De classificatie is gebaseerd op drie hoofdcriteria: berekeningslocatie (d.w.z. waar de resultaten worden geproduceerd), geheugentechnologie (d.w.z. de geheugentechnologie die wordt gebruikt) en berekeningsparallellisme (d.w.z. het maximale parallellisme dat kan worden benut). Op basis van berekening worden vier hoofdklassen afgeleid: Gegevensverwerking-in-Geheugen Array (GiG-A) die resultaten produceert in het geheugen en in de geheugenarray, Gegevensverwerking-in-Geheugen Perifeer (GiG-P) die resultaten produceert in het geheugen en binnen de periferie, Gegevensverwerking-buiten-Geheugen Dichtbij (GbG-D) die resultaten buiten het geheugen en in de buurt van het geheugen produceert, en Gegevensverwerking-buiten-Geheugen Ver (GbG-V) die resultaten buiten het geheugen produceert en ver van het geheugen, respectievelijk. Vervolgens bekijken en vergelijken we de vier klassen en bestaande architecturen kwantitatief. De voorgedragen classificatie en overzicht tonen niet alleen de architecturen die in detail in dit proefschrift zijn onderzocht, maar ook potentiële architecturen die in de toekomst kunnen worden onderzocht.

Architectuurniveau - We stellen twee architecturen voor die de GiG-A- en GiG-P-klasses vertegenwoordigen uit de bovenstaande classificatie. Voor de GiG-A-klasse stellen we eerst een concept voor voor het integreren van zowel berekening als geheugen in één fysiek apparaat, in het bijzonder geheugenresistieve- of geheugenweerstandselementen. Dit concept heeft de potentie om de geheugenmuur of het geheugenknelpunt in

het bijzonder, en de architectuur- en technologiemuur in het algemeen, te verlichten. We demonstreren de mogelijkheden van dit concept met behulp van een toepassing voor de gezondheidszorg en een wiskundige toepassing. Daarna gebruiken we dit concept om een parallelle optelling uit te voeren met alleen de kruisschakelingsarray waarin ook de operanden van de functie worden opgeslagen. We vergelijken het voorlopige resultaat van deze parallelle met conventionele architecturen zoals multicore- en GPU-architecturen. Vervolgens implementeren we de bovengenoemde parallelle opteller, rekening houdend met de controller, communicatie en verbindingsnetwerkschema's. Twee implementaties met twee verschillende logische ontwerpen worden vergeleken met een multicore-architectuur. De resultaten laten zien dat de twee implementaties beter presteren dan een multicore-architectuur op het gebied van prestaties, energie en combinaties van prestatiepunten. Bovendien vormen de controller en communicatie relatief grote overheadkosten in deze architecturen. Hoewel veel aspecten van deze architectuur in het bijzonder voor de CIM-A-klasse ook en in het algemeen werden onderzocht, zijn er nog steeds veel vragen over de integratie tussen CMOS-controller en geheugenweerstandskruisschakelingen, de wisselwerking tussen isolatie en parallellisme, evenals het genereren van complexe en efficiënte functionele eenheden als bouwstenen, enz. Voor de GiG-P-klasse tonen we eerst de mogelijkheden van de Gegevensverwerking-in-Geheugen (GiG)-kern (d.w.z. een geheugenweerstandskruisschakeling met de mogelijkheid om logische bewerkingen uit te voeren met behulp van perifere schakelingen). Daarna stellen we verschillende architecturen voor die een rekenkundige GiG-kern kunnen integreren in verschillende posities van de geheugenhiërarchie (d.w.z. computercache, hoofdgeheugen of versneller). Daarna selecteren we de architectuur die de GiG-kern als versneller gebruikt vanwege de huidige status van geheugenresistieve elementen en GiG-architectuur. Vervolgens onderzoeken we de potentie van deze architectuur met behulp van een analytisch model. Tot slot bouwen we een simulatieplatform om applicaties te porten die op de voorgestelde architectuur worden uitgevoerd. Hiermee verifiëren we onze veronderstellingen over het analytische model en verkennen we mogelijke toepassingen voor de voorgedragen architectuur. Zowel de analytische als de simulatieresultaten tonen aan dat de voorgestelde architectuur beter presteert dan de conventionele architectuur in termen van prestaties en energie met ten minste één orde van grootte. Het is de moeite waard om op te merken dat zowel de architectuur als het simulatieplatform in de kinderschoenen staan en dat er meer werk verricht moet worden om de mogelijkheden van de architectuur voor big data-applicaties volledig te kunnen benutten.

Schakelingsniveau - Omdat het essentieel is om basisblokken te bouwen voor GiG-architecturen, stellen we automatiseringsmiddelen voor logische synthese voor en verbinden we netwerken om de digitale complexe functie op de geheugenweerstandskruisschakeling te realiseren. Eerst stellen we een generiek syntheseplan voor om een digitale rekenkundige functie in hardware description language (HDL) op geheugenweerstandsschakelingen te implementeren. We demonstreren dit stappenplan met behulp van twee casestudy's, een 2-bits teller en een 8-bits opteller. Omdat dit stappenplan een voorlopig resultaat is, zijn er nog inspanningen nodig om het stappenplan te automatiseren en complexere functies te verkennen. Daarna stellen we verschillende ver-

bindingsnetwerkschema's voor die kunnen worden gebruikt in een geheugenresistieve schakeling/ontwerp. Aan de hand van een case study van een parallelle opteller (gebaseerd op de architectuur van de GiG-A-klasse), demonstreren we drie schema's inclusief een direct schema met alleen een kopieerbewerking, een indirect schema met CMOS-schakelingen (d.w.z. een controller) en een hybride schema dat het directe en indirecte schema combineert . De resultaten laten zien dat een hybride schema de hoogste prestaties en het laagste energieverbruik oplevert en daarom moet worden beschouwd als degene om te gebruiken in de GiG-architectuur. Het is de moeite waard om te benadrukken dat de voorgestelde oplossing ruwweg wordt geëvalueerd en dat meer gedetailleerde implementaties nodig zijn om deze oplossingen in bepaalde ontwerpen/systemen te realiseren.

ACKNOWLEDGEMENTS

Growing up in an Eastern culture, I believe that fate leads me to the things that I deserve. Therefore, first, I am thankful for all the opportunities that showed me the way, all the difficulties that diverted my way, all the good that encouraged me, and all the bad that changed me. Furthermore, having a chance to meet and interact with all the people mentioned below is, to my belief, my fortunate fate.

I would like to acknowledge my supervisors Prof. dr. ir. Said Hamdioui, Dr. ir. Mottaqiallah Taouil, and Prof. dr. ir. Koen Bertels. First, I would like to thank Prof. dr. ir. Said Hamdioui, my promotor. To be honest, being your student is challenging, but rewarding. Despite your unmanageable schedule, you always insisted on having our weekly meeting, spending time on brainstorm sessions and correcting my papers thoroughly. Said, thank you for investing efforts in me, both from a research and personal point of view. Moreover, I would like to thank Dr. ir. Mottaqiallah Taouil, my daily supervisor and former office-mate. You are first of all a nice colleague that is always available to discuss existing and non-existing problems, to correct my messy code, to teach me how to write proper code and English, as well as to suffer with me all the 'Friday' paper deadlines. With all your contributions and energy, I am glad that you have been officially my supervisor. Motta, thank you for your time, availability and patience in me, both in research and non-research related matters. Lastly, I would like to acknowledge Prof. dr. ir. Koen Bertels, my former promotor. Koen, thank you for being my first promotor. I remember your encouragements both at the coffee corner and during my evaluation meetings; that gave me courage to continue this path every time it beated me down. In addition, I would like to thank Koen as the previous head of the QCE department for his inspiring talks, generous lady activities, exciting social events, and great team spirit. Different from other PhD lives elsewhere, I am lucky to have a quite open connecting working environment, to share my PhD burdens and to enjoy life "a bit" during my PhD.

I would like to thank my memristor team members, officially, Lei, Jintao, Muath, Uljana and unofficially, Imran, Razvan, Adib, Berna for all the constructive discussions and feedback. We have been gone through our first days with very little knowledge of memristors, to lively discussions on every new idea we had. It has been a long memorable journey to me. With ups and downs, we shared our troubles and joys. I cannot say that I enjoyed all of it, but I am sure the sufferings were less thanks to your presence along the way.

I would like to thank specially my Brazilian research friends Prof. Luigi Carro, Dr. Marco Antonio Zanata Alves, and Paulo Cesar Santos for many long inspiring discussions and support during my hardest time of working on the SiNUCA simulator. Your unconditional supports mean a lot to me and my work.

I would like to thank all of my co-authors Francky Catthoor, Dietmar Fey, Barbareschi Mario, Bosio Alberto, Traiola Marcello, Vatajelu Elena Ioana for all the effort invested in our research. I also would like to thank the committee members for accepting their role, reading this dissertation and providing useful feedback. A special thanks to Daniel for helping me translating the thesis summary, as well as to Lingling and Nga for being my paranymphs.

I would like to thank the supporting staffs of QCE for handling my day-to-day work smoothly. Lidwina and Joyce, thank you for taking care of the paper work, collaborating with me during colloquium organization, and inventing exciting lady activities. Erik and Eef, thank you for the always-available servers, website and work stations.

I would like to thank all my colleagues for their creativity in both scientific and social activities; they are my source of inspiration and joy. Imran, Miki, Pascal, and Guilherme, thank you for jointly organizing the colloquia with me; it is my pleasure to work with you guys. Mafalda, Joost, and Leon, thank you for various interesting social events; we had a great time, especially that time when I tried my first tequila. Anthony, Roel, Berna, Adib, Lei, Razvan, Jintao, Motta, Mohammad, Innocent, Abdullah, Muath, and Abduqader, thank you for being my great office mates; in my moody days, you have been patiently listening to my complaints and responsibly sharing all the challenges with me. Shanshan, Mahroo, Carmina, Hale, Misa, and Lingling, thank you for giving me helpful breaks from my research during the lady activities; it has been enjoyable and powerful to be with you girls. Daniel, Jintao, Troya, Guilherme, Haji, Moritz, and Muath, thank you for good talks and enjoyable lunch time; I wish you guys to have Nature papers soon! Nicoleta, Xiang, Lizhou, Yande, Baozhou, Nauman, Hamid, Cuong, George, and Joost, thank you for the "gezellig" atmosphere in our lab and social events; it is my pleasure to know and talk to you guys.

I am sure that I could not get through the five tough years without my Vietnamese Community in Delft (VCiD). I would like to thank anh Nghi, anh Chi, chi Phuong, anh Bach Duong, anh Hung, anh Thang, and anh Hieu for building a warm and strong community. I would like to thank the members of the online group of Vietnamese ladies in Delft (Hoi chi em) for sharing experiences, plants and good dishes. Vinh - Minh - Dau, chi Huong - anh Dung - be Ngoc, Nhat Anh - Nga, Phan Anh - Ninh - Bo, anh Canh - chi Kim Anh - Ben, Son - Linh - Sumo - Sumi, Dao Tung - Nhung - Mai, and anh Hieu - chi Trang - Gau, Vinh - Diem - Cherry, anh Thao - chi Thao - Nu - Na, anh Phuc - Thao Nguyen, Trang Phan - Eric, chi Lan, Thao Nguyen, Ha, Tinh - Tue, Tin - Huong, Thu, chi Tran, anh Duoc, Thanh Vo, Thien (Alex), Viet, and Vi thank you for all the gatherings and making me feel like I always have a big family to look after me here.

Along the way of gaining a degree, I always had remote supports from my angels (in Vietnamese "quý nhân") to not only overcome difficulties, but also choose a right direction and exploit the best of any situations. I would like to send a special thank to anh Cuong - chi Anh - nha Tom Xiu for many exact advice at the right moment, for your care and con-

cern when seeing anything unusual on my facebook. I would like to also send a special thank to prof. James Peckol for being my remote supervisor and mentor in persuading me to trust my guts and carry on until now. A special thank also to Nghia for always being there, encouraging me, and listening to my problems. Another special thanks to my "USA aids" Trang Le - Tho - Anthony and "Australia aids" Thien throughout the journey. Getting to know and keeping in touch with all of you for the whole time are my blessings.

Last but not least, I would like to express my deepest thank to my big and small family. Mom and Dad, thank you for raising me as an independent girl, always supporting me, and providing me the best condition to pursue this long path. My dear grandmother-in-law and parents-in-law, thank you for always understanding and supporting me through the journey that is completely strange to you. I would like to also thank my sister - Hoang Phuong for taking care of my parents while I am away from home. And a special thank to my sister-in-law - chi Hien and her family for taking care of my parents-in-law while my husband joined me in the Netherlands years ago. Certainly, anh - my best friend and long life partner, thank you for always standing by me, sharing my sorrows and joys and problems, cooking comforting food when I was "always" upset, and never losing hope in me or my PhD. I would like to thank fate, God, or any super power that brings you to my life.

Hoang Anh Du Nguyen Delft, 17 October, 2018