



Architectural Design of Artificial Intelligence Inference Accelerator

Haozhe Li

College of Information Science and Technology, Northwest University, Xi'an, China

hl23886@essex.ac.uk

Abstract. The paper present RaPiD, a revolutionary low-precision accelerator with a wide spectrum of supported precisions from 16-bit floating-point to 2-bit fixed-point, the first system supporting both training and inference operations at very low precision levels without sacrificing performance. On top of latest 7nm EUV technology, RaPiD demonstrates high efficiency with 3.5 TFLOPS/W at FP8 precision and 16.5 TOPS/W at INT4 precision. This makes it very suitable for AI computation that must be fast as well as powerful. Quickloop, an innovative accelerator engine utilizing reinforcement learning, significantly accelerates AI accelerator design by reducing turnaround and exploration times. Field Programmable Neural Array (FPNA) is also highlighted because it supports post-fabrication reconfigurability, making edge AI viable for it. The paper also elaborates on optimization techniques for micro-AI platforms, including Neural Architecture Search, quantization, and compression, that enable cost-effective runtime for DNN on resource-limited systems. FPNA, RaPiD, Quickloop, and micro-AI optimization schemes together constitute a valuable contribution in hardware design for AI and the future of intelligent, efficient, and resource-optimized implementations for AI applications across industries.

Keywords: AI Inference Accelerator, Rapid, Quickloop, FPNA, Low-Precision Computing

1 Introduction

The rapid evolution of artificial intelligence (AI) has given rise to the demand for specialized hardware accelerators capable of performing complex computations with minimal power consumption.

Traditional AI accelerators struggle to strike a balance between performance and power, particularly in edge computing and low-resource environments. All these challenges are met by this paper, however, presenting RaPiD, a new low-precision accelerator that is capable of supporting all precision levels from 16-bit float to 2-bit fix, all with uncompromising performance. RaPiD, manufactured in state-of-the-art 7nm EUV process, achieves unprecedented computing efficiency, making it poised to be the probable sweetheart of AI training and inference workloads [1]. Along with

RaPiD, the paper introduces Quickloop, a new reinforcement learning-based design exploration engine to enable auto-design of AI accelerators [2].

Quickloop delivers significant design turnaround time savings and improves the quality of the final product, and hence it is a useful tool for researchers and engineers [2]. The FPNA, an edge AI reconfigurable accelerator, is also discussed in the paper. FPNA's reconfigurability after manufacturing allows adaptation to various topologies of neural networks, making it suitable for novel AI models and algorithms. The paper also discusses optimization methods for micro-AI platforms, including model compression, quantization, and Neural Architecture Search (NAS), facilitating efficient implementations in resource-limited scenarios [3].

They enable efficient implementation of deep neural networks in low-resource environments. They are crucial in broadening the range of application domains of AI on more devices and industries. Collectively, these trends signify notable advancements in AI hardware design, offering more powerful and efficient solutions [4]. Furthermore, they enhance adaptability to meet the evolving demands of AI applications [5-7].

2 Artificial Intelligence Inference Accelerator

2.1 RaPiD

RaPiD, a novel low-precision accelerator, which is capable of supporting a range of precisions from 16-bit floating-point to 2-bit fixed-point. It is indeed the first such accelerator that can handle training and inference tasks at extremely low levels of precision and yet uses much less power but with no performance loss. RaPiD is built using the latest 7nm EUV (Extreme Ultraviolet Lithography) technology, thus making it achieve marvelous performance in both the performance and efficiency aspects. RaPiD, to FP8 precision, can achieve 3.5 TFLOPS/W (Tera Floating-Point Operations Per Second per Watt), and in case of employing INT4 precision, the figure is 16.5 TOPS/W (Tera Operations Per Second per Watt). These figures actually show that RaPiD is a good performer, capable of executing heavy workloads but with low power consumption, which is exactly what is needed for AI computations that demand speed and power efficiency [1].

In practical applications, RaPiD performs effectively, especially in DNN inference computations. Running with 4-bit precision, batch size merely 1, RaPiD can achieve an average rate of about 7 TOPS/W. This makes it particularly suitable for edge devices and applications where power consumption is a concern. For training, RaPiD is content with 8-bit floating-point calculation, generating a bit more than 203 TFLOPS when used in multi-core configurations. All of these indicate very well the optimal balance RaPiD obtains between computation power and power, making it a very good candidate as an AI accelerator for both inference and training. The efficiency with which it can operate at very low precision with minimal loss in terms of reduced performance signifies ultra-low-power AI systems may be more practical in the future [1].

Figure 1 illustrates how ongoing research progress has consistently reduced the precision requirements for neural network training and inference. Specifically, for edge deployment applications, inference precision has been successfully optimized to function efficiently with only 2-4 bits for both weight and activation representations [1].

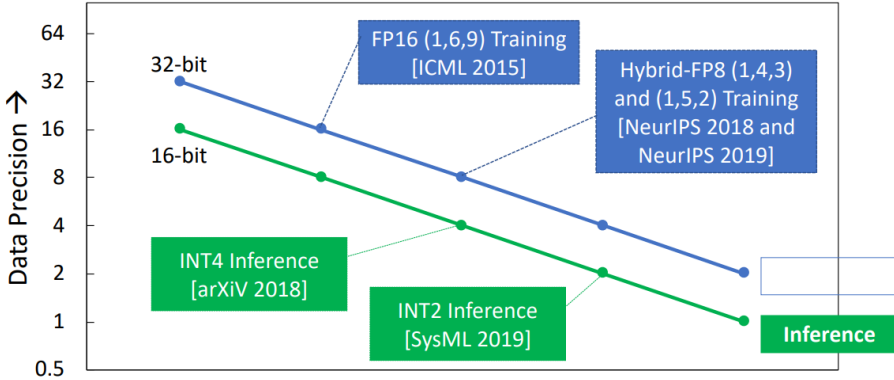


Fig. 1. Roadmap for precision scaling in training and inference [1].

2.2 Quickloop

One very striking AI accelerator design innovation is Quickloop. Quickloop is an engine specially tailored to accelerate exploration and design of AI accelerators. Quickloop uses the OpenAI Gym, a widely adopted environment for running reinforcement learning algorithms. Quickloop, using reinforcement learning, eliminates much of the design exploration, which translates to more efficient and superior development. In fact, on the Gemini DNN accelerator, Quickloop has much lower turnaround time (TAT) of over 30%. This reduction is significant as it enables testing of many configurations and parameters. It surpasses the usual practices in standard FPGA design workflows. Switching between different configurations becomes easier, facilitating the development of more efficient and optimized AI accelerators [2].

In addition to simply its inherent capacity, Quickloop incorporates a data-driven process that complements the design phase. This process capitalizes on the experience of previous design cycles, and it directs and enhances the next designs that are being created. This means that there is less effort and time to attain the performance levels needed. With design exploration driven into automation and optimization, Quickloop not only accelerates AI accelerator development but also ensures the quality and productivity of the final product. For this very reason, Quickloop can be truly said to be an essential resource for future AI hardware-developing researchers and engineers. Quickloop is weaved with OpenAI-Gym framework and is composed of Quicksteps, as shown in Figure 2 [2][4].

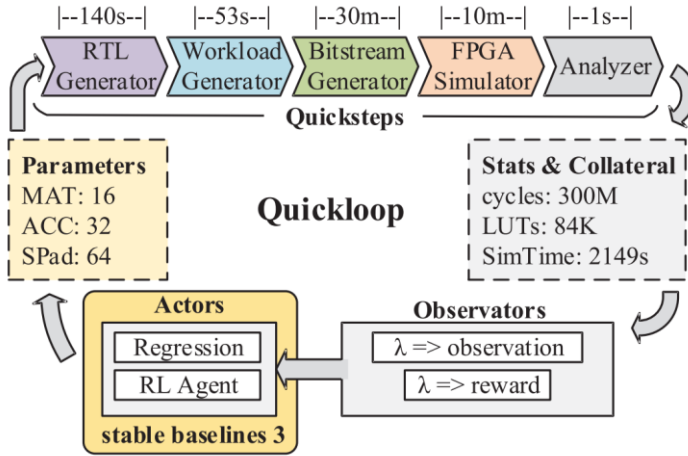


Fig. 2. A Quickloop is based on the OpenAI Gym framework and is made up of Quicksteps [2].

Quickloop's architecture is based on Gemini's systolic array design, as shown in Figure 3. In this setup, multipliers are positioned on the left, while multiplicands and bias components are placed at the top of each processing element (PE). Each PE includes a combinational multiply-accumulate (MAC) unit, as depicted in the diagram. This provides a basic overview of Gemini's architecture, with more detailed information available in the referenced materials. Gemini's RoCC interface includes a DMA controller for managing data transfers through the main processor's memory bus. For design space exploration, Gemini's configurable parameters—such as accumulator size, ScratchPad capacity, and systolic array dimensions—are exposed through the DSELayer framework. This framework generates both the Gemini RTL implementation and a parameterized header file, which is then used by the DNN mapper for optimization [2].

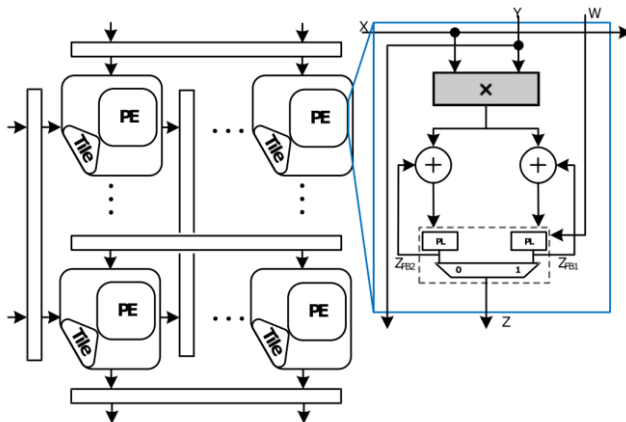


Fig. 3. Gemini systolic array Micro-architecture [2].

3 FPNA's Role in Edge AI Applications

Traditional methods for implementing Deep Neural Networks (DNNs) on-chip often lead to inflexible hardware designs, especially in Application-Specific Integrated Circuits (ASICs) [8, 9]. While these solutions offer high computational performance, they lack the flexibility to modify core neural network structures (such as CNNs and RNNs) after deployment. As a result, adapting to new network architectures or different DNN scales requires expensive hardware redesign and manufacturing, making them unsuitable for rapid deployment. Field Programmable Gate Arrays (FPGAs) offer a more flexible alternative, but their computational power is still inadequate for demanding Army edge computing applications [10]. Figure 4 compares energy efficiency across various computing platforms, emphasizing the benefits of the proposed Field Programmable Neural Array (FPNA) solution [3].

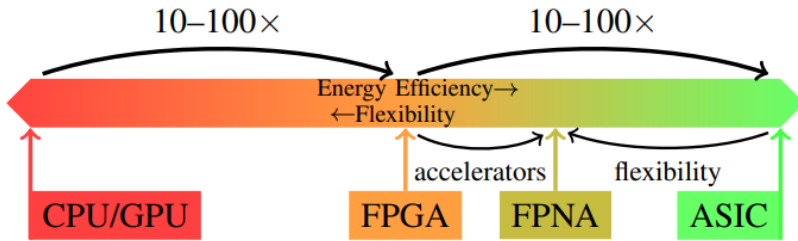


Fig. 4. Comparative computing efficiency for AI platforms [3].

In certain edge AI applications, there has been rumor of the use of something referred to as the Field Programmable Neural Array (FPNA). This refers to an AI accelerator that diverges from traditional designs by offering post-manufacturing reconfigurability and adaptability. It is a reconfigurable accelerator designed for use in resource-constrained environments. In fact, this reconfigurability, being flexible once built, makes it possible to be capable of supporting more than a single type of neural networks and structures. Therefore, this flexibility, as it seems, is extremely helpful, and it transforms easily when requirements change, which is one of the strongest points. It would also seem that this post-manufacturing reconfigurability is a large thing, as it will make the hardware upgradeable at the same time as the software. This would make sure that it would be compatible with future breakthroughs, as new models and algorithms come out [3].

When talking about FPNA, there is no doubt that it does have high computational efficiency. This is shown to be highly effective in the edge devices, where the power consumption is usually something which needs to be carefully estimated. With 4-bit accuracy, it achieves up to 545.4 GOPs/W, demonstrating its operational efficiency. Something other traditional hardware will not be easily able to copy. The efficiency is also partly due to the fact that it uses low-precision calculations, and the hardware design is included. It minimizes power consumption while maintaining competitive performance, making it suitable for resource-constrained applications. To finish it off, FPNA also seems to have implementations for various different precision modes,

such as 4-bit, 8-bit, and 16-bit. These modes are there to help serve from low-workload to high-workload of AI workloads. So here in this instance, the flexibility seems really helpful to provide optimal performance for most applications [3].

4 Others

The research also addresses the challenge of designing neural network accelerators optimized for micro-AI platforms. These platforms are meant to run in a scenario where resources are highly constrained. They really need to rely on sophisticated methods so that deep neural networks, or DNNs, can run effectively, but without consuming a lot of computer or memory space. A number of methods are used to this effect, some of which include model compression, quantization, and NAS. Model compression, in a sense, helps by reducing the size of the neural network, cutting out some of those redundant parameters. Quantization lowers the precision of the calculations, and therefore both memory and computational needs reduce. NAS, however, differs in approach. It automates the design of neural network architectures, aligning models with the constraints of the hardware in use.

Optimizing these accelerators in micro-AI platforms is one of the biggest challenges. This problem is all about trying to maintain inference accuracy but, at the same time, reduce resource consumption. Low-precision computing drastically reduces computational and memory demands while maintaining output quality. When the right amount of precision is chosen intentionally for each task, designers can create accurate AI accelerators that operate effectively within platform constraints. There are additional arguments also about the way hardware accelerators are employed, especially to speed up micro-AI inference on very limited devices. These hardware-based accelerators, they shift more computation-heavy tasks off the host processor to make the device conduct AI work in a better manner.

So overall, in the paper, the inventions that have been talked about—RaPiD, Quickloop, FPNA, and even micro-AI platform optimization techniques—are contributing a lot to the move towards developing AI accelerators which not only turn out to be more general-purpose in nature but also effective in execution. RaPiD, with its low energy consumption and adaptable precision, is highly valuable for both training and inference tasks. Quickloop, being one that can enable exploration of design to be made possible to run automatically, accelerates the manner in which AI accelerators are optimized, and FPNA, being a reconfigurable and computation-effective one, reconfigures itself into something that is perfectly suited for AI on edge applications. In addition, micro-AI optimization methods allow deep neural networks to perform more effectively even under resource-constrained environments, thereby allowing AI to be used everywhere and in every possible manner. In general, it is these technologies that allow the creation of AI hardware to continue, along with it the potential of future AI systems to be more capable, more versatile, and more efficient.

In addition to being for performance enhancement or power optimization, other implications arise. By enabling the AI accelerators to become feasible with lower

precision and to be more adaptive, these technologies are opening new uses of AI to come to the forefront, particularly where constraints of resources and power have gotten in the way previously. Edge devices such as smartphones, IoT devices, and self-driving cars benefit from the improved efficiency and flexibility of these accelerators. In this way, local AI work can be done to a greater extent without depending on cloud computing. And this not only leads to change in the form of how much quicker things will respond, but also results in privacy and security to get better since sensitive data remains within the device itself.

Post-production reconfigurability of accelerators like FPNA enables them to adapt to the rapid evolution of AI models and algorithms. As new architectures of neural networks are being developed and new applications are being discovered, such reconfigurable accelerators, they may be reused, adapting to these innovations. This prolongs the hardware's lifespan and makes it more worthwhile in the long term. Such flexibility, especially in research and development areas, is valued since fast experimentation and experimentation with new ideas is needed to move forward in such areas. The application of reinforcement learning in Quickloop's design process represents a significant shift from traditional AI accelerator development practices.

Quickloop simplifies the search for design parameters, accelerating development. It also uncovers potentially better or lower-cost designs that are unattainable through conventional methods. That can enable faster innovation in AI hardware construction, and the accelerators are further specializing, i.e., they're solving specific problems or applications and not just making them more powerful but application-specific. Optimizing micro-AI approaches requires highly specialized hardware tailored to specific configurations.

As more sectors, including medical centers, farming activities, and production plants, increasingly depend on AI, it would also seem that there is a greater need for tailored hardware created specifically to operate within the limits of each sector. Methods like model compression, quantization, and NAS are crucial for enabling AI to operate across a wide range of devices, from small sensors to industrial machinery. Globally, innovations like RaPiD, Quickloop, FPNA, and micro-AI optimization techniques are seen as groundbreaking in AI hardware design. They not only improve AI hardware efficiency but also broaden AI's applicability across diverse sectors. The types of environments and domains on which AI can operate, those are being extended. As AI evolves rapidly, these technologies mark a turning point, making hardware systems more efficient, powerful, and adaptive to meet growing global demand.

5 Conclusion

In summary, the paper provides a broad overview of emerging trends in AI hardware design, focusing on low-precision accelerator design, reconfigurable design, and optimization for resource constraints.

RaPiD, with industry-best efficiency and programmability, is the new benchmark for AI accelerators and is well-positioned for training and inference workloads.

Quickloop revolutionizes the design process for AI accelerators by employing reinforcement learning to significantly reduce construction time while improving product quality. FPNA's reconfigurability after manufacturing provides unprecedented flexibility, making it particularly suitable for edge AI applications where flexibility and efficiency are critical.

All the aforementioned developments, individually and collectively, contribute to the evolution of AI systems that are more effective, flexible, and responsive, meeting the growing global demand for AI solutions. With the progress of AI, the innovations in this paper will future-proof AI hardware as a more effective, more powerful, and more multifaceted device for the diverse needs of different industries.

References

1. Venkataramani, S., Srinivasan, V., W. Wang, S. Sen, J. Zhang, A. Agrawal, M. Kar, S. Jain, A. Mannari, H. Tran, Y. Li, E. Ogawa, K. Ishizaki, H. Inoue, M. Schaal, M. Serrano, J. Choi, X. Sun, N. Wang, C.-Y. Chen, A. Allain, J. Bonano, N. Cao, R. Casatutak, M. Cohen, B. Fleischer, M. Guillorn, H. Haynie, J. Jung, M. Kang, K.-h. Kim, S. Koswatta, S. Lee, M. Lutz, S. Mueller, J. Oh, A. Ranjan, Z. Ren, S. Rider, K. Schelm, M. Scheuermann, J. Silberman, J. Yang, V. Zalani, X. Zhang, C. Zhou, M. Ziegler, V. Shah, M. Ohara, P.-F. Lu, B. Curran, S. Shukla, L. Chang, and K. Gopalakrishnan, RaPiD: AI Accelerator for Ultra-low Precision Training and Inference, in 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), (IEEE, Piscataway, NJ, 2021), pp. 153-166
2. Inayat, K., Muslim, F. B., T. Mahmood, and J. Chung, Journal of Systems Architecture 155, 103260 (2024)
3. Gadfort, P., and Ayorinde, O. A.: FPNA: A Reconfigurable Accelerator for AI Inference at the Edge, in 2021 IEEE 34th International System-on-Chip Conference (SOCC), (IEEE, Piscataway, NJ, 2021), pp. 242-247
4. Sze, V., Y.-H. Chen, T.-J. Yang, and J. S. Emer, Proceedings of the IEEE 105, 2295-2329 (2017)
5. Mantovani, P., R. Margelli, D. Giri, and L. P. Carloni, HL5: A 32-bit RISC-v processor designed with high-level synthesis, in 2020 IEEE Custom Integrated Circuits Conference (CICC), (IEEE, Piscataway, NJ, 2020), pp. 1-8
6. Tang, X., E. Giacomini, A. Alacchi, B. Chauviere, and P.-E. Gaillardon, OpenFPGA: An open-source framework enabling rapid prototyping of customizable FPGAs, in 2019 29th International Conference on Field Programmable Logic and Applications (FPL), (IEEE, Piscataway, NJ, 2019), pp. 367-374
7. Zhang, C., P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, Optimizing FPGA-based accelerator design for deep convolutional neural networks, in Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, (ACM, New York, NY, 2015), pp. 161-170
8. Raffin, A., A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, Journal of Machine Learning Research 22, 1-8 (2021)
9. Shin, D., J. Lee, J. Lee, and H. J. Yoo, DNPU: An 8.1TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks, in 2017 IEEE International Solid-State Circuits Conference (ISSCC), (IEEE, Piscataway, NJ, 2017), pp. 240-241
10. Zhang, C., P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, Optimizing FPGA-based accelerator design for deep convolutional neural networks, in Proceedings of the 2015

ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, (ACM, New York, NY, 2015), pp. 161-170

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

