Recent advances in Systems for Artificial Intelligence

1.Shaik Anjum, Student, Dept. Of ECE, KHIT, Guntur

2.T.N. Sai Kumar, Student, Dept. of ECE, KHIT, Guntur

3. Mr, B. Ramachandraiah, Assoc.Prof, Dept. Of ECE, KHIT, Guntur

Abstract : The trends in AI chip development can be divided into two categories of applications: server and edge. As for the server, recently, Application specific integrated circuit chips designed for learning functionalities have become mainstream, with competitively computational performance. However, the limitations of Moore's Law have started to impose themselves on this emerging type of AI chips, creating the need for new technological innovation. Meanwhile, regarding edge AI chips, research on data compression technology is advancing to lower power consumption while maintaining high performance. Further improvements have been made since 2015 in recognition accuracy in binary/ternary; moreover, there has been research on in-memory processing to configure 1 bit by combining memory and arithmetic element, where non-volatile memory can achieve higher performance and lower power consumption. With these backdrops, this paper summarizes the progress made to date in the field of AI Integrated circuit technology while also identifying the future direction of next-generation technologies

Introduction: Because of the numerous advancements made in the field of deep learning technology , various novel artificial intelligence (AI) processors have been developed and various products based on these chips have been released recently. In this study, we explain the technical content through several trends. We define an AI chip as a "chip specializing in realizing the operations of the brain" and targets deep learning chips with advanced abstraction .

As shown in Fig. 1, the evolution and emergence of AI chips can be broadly divided into two periods: one from 2013 to 2015, and the second from 2015 onward. The first period is the basic research (Basic) phase, which can be seen as the period during which the implementation method of the basic net models was explored, and the second period can be considered as the practical application research phase. This can be further divided into two phases, one in which high efficiency was pursued, and the other phase that is more diversified and Versatile than the second phase. This phase is the period in which products with low power consumption and high performance were developed, particularly targeting edge applications such as mobile/IoT applications. This paper will discuss various specific chip types.

The Basic Phase has four activity flows. The first is the pursuit of the circuit configuration of the convolutional neural network (CNN). Google's Tensor Processing Unit (TPU), whose basic configuration and basic design were considered to have been realized between 2013 and 2014. This basic chip design is dedicated to server inferences.

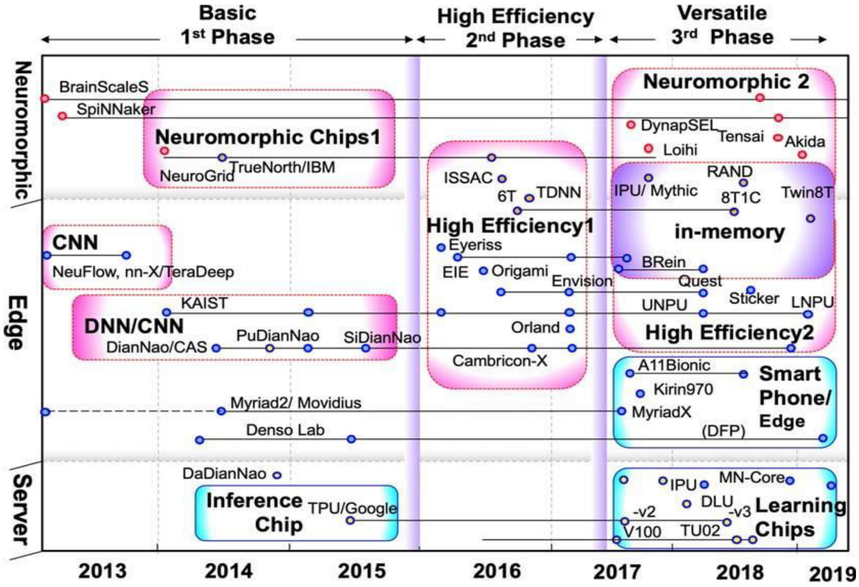
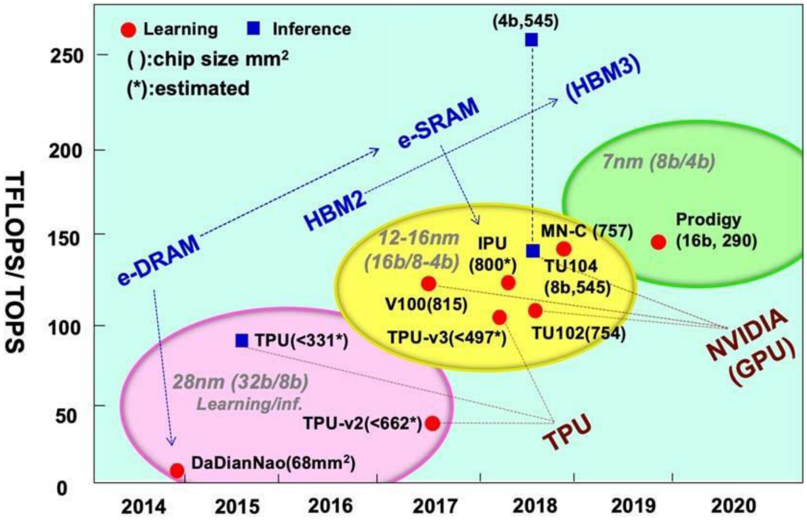


Fig. 1. AI Integrated circuit trend.

2. Trends in AI chips for servers

we first discuss the typical chip throughput [tera flops per second (TFLOPS) or tera operation per second (TOPS)] for each generation of AI chips used for full-scale inference and learning in servers. Next, the basic circuit system configuration of the AI chip will be described using Google's TPU[10)] as a motif. An overview of the two major factors used to determine performance, computation, and the data transfer memory bandwidth will be outlined.

Figure 2 describes server chip throughput since 2014. The subscript is the code name or chip name. ■ indicates chips for inference-making, and • indicates chips for learning. Although learning and inference-making cannot be easily diverted to each other, large differences are usually indistinguishable in the basic configuration. All chips are ASICs except for the NVIDIA GPUs. Typical examples include DRAM-embedded DaDianNao by CAS, TPU for inference by Google. Fig 2



CAS(DaDianNao), Graphcore(IPU), MN-C chip(PFN), Tachyum(Prodigy)

Google's TPU is an accelerator that specializes in 8 bit integer and inference processing for servers. Around 2013– 2014, Google anticipated exponential rise in demand in the future and accordingly started designing for the early introduction to data centers in 2015.

Figure 3 shows a circuit block diagram of the TPU chip. Data (e.g. image data) is transferred from the host CPU/main memory from the lower side of the chip and stored in a local buffer. Meanwhile, the weights are loaded from the DDR3 DRAM (8 GB) on the upper side and are transferred into the chip and two-dimensionally deployed (position fixed: two weights are stored on each element) in a matrix multiply unit. The data and weight transfer rates are 10 GB s^{-1} and 30 GB s^{-1}, respectively. In the unit, PEs that handle 8 bit multiply-accumulate (MAC) operations are arranged in an array (256 × 256 = 64k). A total of 256 bits of data are input from the left side of the unit, and the multiplications with the weights at the first column of the array are executed. This is a vector–vector multiplication of one of the so-called matrix (weight)-vector (input data) multiplication operations, which is usually suitable to process a fully connected layer used for multilayer perceptron (MLP). The data movements in the next step (one clock) are shown in the unit with two arrows. The data (D) are transferred in the x direction clock by clock in a horizontally systolic manner, and the multiplied results (A) are transferred in the −y direction by one grid and are summed (accumulated) in the−y direction through 256 clocks. As a result, one summedvalue is output from the lower side of the unit after 256 clocks. Similar processing is performed independently for each column, and 256 summed values are output. It should be noted that if 256 data bits of the one vector are simulta- neously input from the left side of the unit, the summation of 256 multiplied results (A) in the one column cannot be easily performed. Therefore, a slightly complicated control opera- tion is performed; when entering the unit, each element of the input vector is input with a one-clock delay each other in the −y direction by the systolic data setup circuit. Such a data flow is called systolic data flow in the vertical direction. The 256 outputs of the unit are temporarily stored in an accumulator. When the number of input data is greater than 256, the results are summed in the accumulator.

Then, after performing activation, normalization, and pooling, calculations for the layer are completed; the data are transferred to the local buffer, and they become the input for the subsequent layer. In other words, the data go through one round to the next layer. The workload of the activation function and pooling processing is as small as a few percent of the total workload. The calculation for the convolution layer can be performed by one-dimensionally arranging weights of the filters and input feature map pixels in the –y direction.

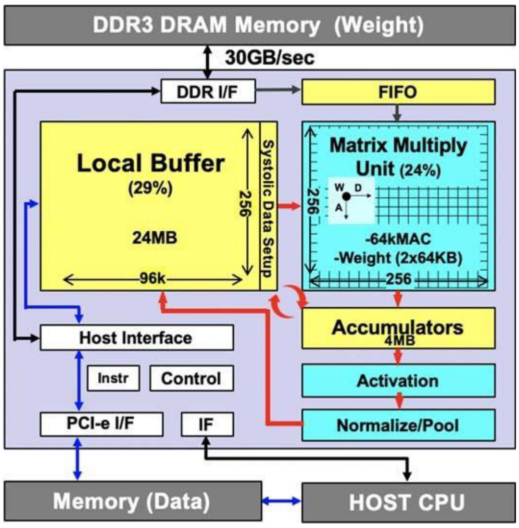


Figure 3 TPU Block diagram

3. Edge AI chips:

The field of applications of edge chips is broad, extending from high performance automotive and robotic applications to smart phones, IoT, wearables, and always-on products; thus, required performances of throughput and power consumption are also widely spread. In automobiles, the throughput of 10 TOPS and more is required, and the power consumption is approximately 10 W. Conversely, for smart phones, it is essential that the power is 1 W or lower. Improving the throughput under this power restriction is critical. Therefore, implementation of the data compression technology on the edge AI that can increase the throughput and reduce the power consumption is the key. It is noteworthy that with always-on products, the ultimate compression rate is assumed. The focus of this section is on the technology, particularly for smart phones and always- on products.

Eyeriss chip was announced by MIT/NVIDIA in February 2016 at the International Solid-State Circuits Conference (ISSCC). It targets convolutional layer-oriented models such as Network in Network/GoogleNet at that time. In this chip, DRAM is external.

The first technique is a dataflow control method that incorporates a proprietary technique while keeping in mind the reuse of both the data and filter weights in convolutional computing.

The weights (W) are transferred in the x direction, input activa- tions (I) are transferred diagonally, and the accumulated outputs (O) is transferred in the y direction. Each PE is responsible for calculating the one-dimensional convolu-tional computation (1D Conv) with the weights (one row of filter) and input activations (one row of input feature map); the weights, input activations, and intermediate outputs of the 1D Conv are stored in local register files (or SRAM) of W-LRF, I-LRF, and O-LRF, respectively, and reused with changing one of the weights or the input activations. Consequently, repeated data flow to and from the PE array can be prevented, and thus power wastage due to data transfer is eliminated. This technique is named the row stationary dataflow (RS) method. Typical dataflow methods, other than the RS method, include the weight stationary (WS) method, which is suitable for batch processing used in TPU, and the output stationary (OS) method, which is relatively suitable for sparse compression (sparsification) used in ShiDianNao. The second technique is a method of detecting the zero value of the input activations and skipping the multiplication operation. In the CONV layer, ReLU is usually used as an activation function, and thus, more than 50% of the activa- tions are zero data, and wasteful power consumption is reduced by 45%. This technology is currently used in most AI chips.

The third is a network on chip (NoC) function placed in the SRAM buffer block. Recognition ID numbers are given to each PE, and using the numbers, the NoC transfer effectively data and weights to the PEs. For example, data and/or weights multicasting can be performed efficiently. This technology controls the RS dataflow as explained above. Furthermore, the size of the logical two-dimensional PE array can be efficiently reconfigured with respect to changes in model size (node size, number of channels, number of filters, etc.).

The fourth point is that lossless compression/decompres- sion processing (run length compression), is performed in the input/output data to the external DRAM, and the compression efficiency is approximately 1/2.

4. Discussion

This section will first discuss throughput and power con- sumption, which are important performance factors for AI chips. It will finally characterize the relationship between applications and weights that represent the scale of the model, and discuss the importance of memory embedding.

Throughput and chip size shows the relationship between the peak throughput (TOPS or FLOPS/s) and the chip area. The area size was used by estimating the area of the circuit related to the operation of the neural network. In addition, only throughputs were regularized to 28 nm and 700 MHz. Therefore, it is clear that the throughput is uniquely determined in proportion to the area and is inversely proportional to the number of bit precision. Most typically, 4 bits constitutes 1 TOPS mm^{-2} (28 nm, 700 MHz). If the weight memory is embedded, the calculation portion will occupy approximately 10% of the whole chip (with DaDianNao, calculation logic = 6%, buffer = 5%); conse- quently, if the figure is shifted one digit (10 to 1) to the left, the result will become almost consistent with the rule. With EIE/DNPU/ENVISION, the speed can be increased almost proportionally by selecting a low-bit on the circuit. By contrast, with the TPU, the operation speed does not change even after changing to 4 bits. EIE can improve the performance up to 30-fold by incorporating the functions of pruning and non-zero detection function.

References

1) P. Merolla et al., Science 345, 668 (2014).
2) H. Momose and T. Asai, J. Jpn. Soc. Artificial Intell. 33, 23 (2018) [in Japanese].
3) C. Farabet, E. Culurciello, and Y. LeCun, Conference, 2011, p. 109.
4) V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, Computer Vision and Pattern Recognition Workshops (CVPRW 2014), 2014, p. 23.
5) Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, The 42nd Int. Symp. on Computer Architecture (ISCA), 2015.
6) Y. Chen, T. Krishna, J. Emer, and V. Sze, Proc. 2016 IEEE Int. Solid-State Circuits Conf. (ISSCC), Session 14.5, 2016, p. 262.
7) T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, ASPLOS '14 Proc. 19th Int. Conf. on Architectural Support for Programming Languages and Operating Systems, 2014, p. 269.
8) Y. Chen et al., Proc. 47th IEEE/ACM Int. Symp. on Microarchitecture (MICRO'14), 2014, p. 609.
9) D. Fu Liu, T. Chen, S. Liu, J. Zhou, S. Zhou, O. Temam, X. Feng, X. Zhou, and Y. Chen, ASPLOS '15 Proc. 20th Int. Conf. on Architectural Support for Programming Languages and Operating Systems, 2015, p. 369.
10) N. Jouppi et al., Proc. 44th Annual Int. Symp. on Computer Architecture (ISCA), 2017, p. 1.
11) S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. Horowitz, and W. Dally, Proc. 2016 ACM/IEEE 43rd Annual Int. Symp. on Computer Architecture (ISCA), 2016, p. 243.
12) D. Shin, J. Lee, J. Lee, and H. Yoo, Proc. 2017 IEEE Int. Solid-State Circuits Conf. (ISSCC), Session 14.2, 2017, p. 240.
13) J. Dean, NIPS 2017 Workshop, Deep Learning at Supercomputer Scale, Panel Discussion, 2017.
14) J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H. J. Yoo, Proc. 2018 IEEE Int. Solid-State Circuits Conf. (ISSCC), Session 13.3, 2018, p. 218.
15) K. Ueyoshi, K. Ando, K. Hirose, S. Takamaeda-Yamazaki, J. Kadomoto, T. Miyata, M. Hamada, T. Kuroda, and M. Motomura, Proc. 2018 IEEE Int. Solid-State Circuits Conf. (ISSCC), Session 13.2, 2018, p. 216.
16) J. Zhang, Z. Wang, and N. Verma, Proc. 2016 Symp. on VLSI Circuits Digest of Technical Papers, 2016, p. C252.
17) H. Valavi, P. Ramadge, E. Nestler, and N. Verma, Proc. 2018 Symp. on VLSI Circuits Digest of Technical Papers, 2018, p. C141, C13-5.
18) K. Ando et al., Proc. 2017 Symp. on VLSI Circuits Digest of Technical Papers, 2017, p. C24, C2-1.
19) J. Choi, S. Venkataramani, V. Srinivasan, K. Gopalakrishnan, Z. Wang, and P. Chuang, The Conf. on Systems and Machine Learning (SysML) 2019, 2019 [https://mlsys.org/Conferences/2019/doc/2019/168.pdf].
20) R. Mochida, K. Kouno, Y. Hayata, M. Nakayama, T. Ono, H. Suwa, R. Yasuhara, K. Katayama, T. Mikawa, and Y. Gohou, Proc. 2018 Symp. on VLSI Technology Digest of Technical Papers, 2018, p. T175, T16-4.
21) I. Giannopoulos, A. Sebastian, M. Le Gallo, V. P. Jonnalagadda, M. Sousa, M. N. Boon, and E. Eleftheriou, Proc. Int. Electron Device Meeting 2018, 2018, p. 629, 27.7.