Memory wall 을 극복하기 위한 PIM 가속 기술에 대한 조망

정헌희 [1], 백윤흥 [1]
[1] 서울대학교 전기 정보공학부, 반도체 공동연구소
honeyjung@snu.ac.kr, ypaek@snu.ac.kr

A Survey on PIM Acceleration Technology to Overcome Memory Wall Problem

Heon-Hui Jung[1], Yun-Heung Paek[1]
[1]Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center (ISRC), Seoul National University

요 약

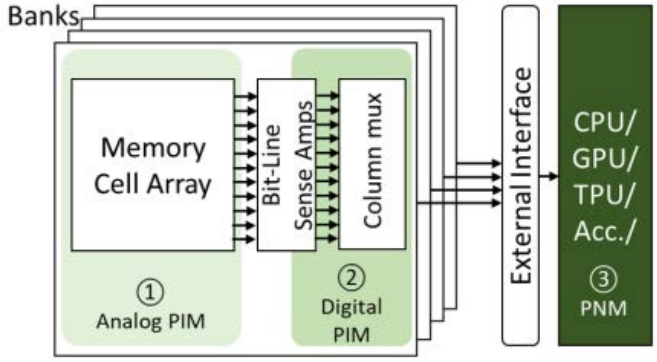
활용도가 높아지고 있는 최근의 딥러닝 애플리케이션 등을 사용하기 위해서 기존의 CPU 구조로는 한계가 있어 GPU, TPU 등의 하드웨어로 가속하려는 노력이 있어왔다. 하지만 물리적인 제약으로 인해 메모리 대역폭에 한계가 있으며, 이를 뛰어넘기 위해 메모리 안에서 직접 연산을 수행하는 Processing-in-Memory 기술이 떠오르고 있다. 본 논문은 PIM 기술을 사용할 때의 불이익을 감수하면서 장점을 최대한 활용하는 방법들에 관해서 서술하였다.

1. 서론

최근 딥러닝 기술과 블록체인 기술과 같은 여러가지 애플리케이션의 등장으로 인해 컴퓨팅에 대한 수요가 매우 증가하고 있다. 이런 기술들을 사용자에게 원활하게 공급하기 위해 GPU, TPU 등의 하드웨어를 활용한 해법들이 등장하였으며, 병렬 처리와 구조적인 변형 등을 통해 실행시간을 줄이고 throughput 을 늘리려는 시도들이 있어왔다. 하지만 특히 DNN 과 같은 어플리케이션의 경우 input, weight, output 정보들을 읽고 쓰는 과정에서 메모리 접근이 빈번하게 발생하게 된다. 하지만 세가지 모두 상당한 크기를 가지고 있으며 ResNet50 과 같은 큰 네트워크의 경우[1] 이들을 모두 on-chip 메모리에 올리기에는 한계가 있다. 따라서 연산하는 중간에 메모리에 접근하는 과정이 꼭 필요하게 되며, 이 과정에서 상당한 딜레이와 inter-chip communication 으로 인한 에너지 소비가 발생하게 된다. 또한 물리적인 문제로 인해 대역폭 또한 한정되게 되므로, 여러 연산들이 같은 메모리를 사용할 경우 딜레이는 더 심해질 수밖에 없다. 따라서 최대한 이 과정을 피하기 위해 여러 시도가 있어왔으며, 한번 메모리에서 불러온 값을 최대한 재활용해서 연산에 활용하는 data-reuse 를 극대화하는 방식의 디자인이 설계되고 있다. nn-X 의 경우 weight 값을 연산시에 불러오는 것을 막기 위해 연산 전에 미리 weight 정보를 불러와 두는 작업을 진행한다.[2] Eyeriss 의 경우 Row Stationary 기법을 활용해서 이전의 가속기에 비해 1.4~2.5 배 더 에너지 효율적인 연산을 진행할 수 있는 기법을 개발하였다.[3] 위의 data-reuse 를 활용한 설계를 활용하면 메모리 접근을 줄이게 되면서 에너지 소모와 딜레이를 줄일 수 있으나, 폰 노이만 구조로 인해 발생하는 메모리 접근 시간, 에너지 소모라는 근본적인 문제는 해결할 수 없다. 딥러닝 외에도 메모리 접근이 많이 필요한 memory-bound 프로그램의 경우에도 이런 문제가 심하다. [4]의 경우, 블록체인 연산에서 일어나는 많은 양의 memory 접근을 on-chip memory 를 활용하여 대역폭을 효율적으로 활용하려는 시도를 하였다.

하지만 Processing in Memory 기술을 활용한다면, 이보다 성능을 개선할 수 있으며, 메모리 접근으로 인한 근본적인 문제를 해결할 수 있다. 이는 고전적인 폰 노이만 구조와는 다르게 프로세서가 아닌 메모리에서 연산을 진행하는 구조로 메모리 대역폭에 제한을 받지 않으며, inter-chip communication 이 큰 폭으로 줄어들기에 이에 사용되는 에너지와 시간을 아낄 수 있다. 본 논문에서는 여러 방식의 PIM 기술 적용 방법에 대해 알아보고, 이를 통해 어떤 분야에 적용할 수 있는지에 관해 알아보고자 한다.

2. Processing-in-Memory 기술

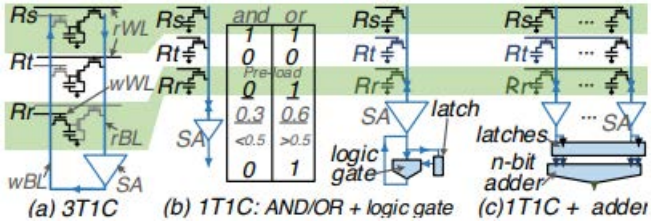


(그림 1) Processing in/near Memory variants[5]

PIM 의 종류에는 여러가지가 있으며 연산이 어디에서 어떻게 이루어 지는지에 따라 크게 analog PIM 과 digital PIM, PNM 으로 나눌 수 있다. 또한 기존의 DRAM 구조에서 벗어나 Memristor 를 사용한 ReRAM 에서 Memristor 의 특성을 활용하여 연산을 진행하는 방식도 존재한다.

2-1 Analog PIM

Analog PIM 과 Digital PIM 은 데이터가 DRAM 구조 내의 Bit-Line Sense Amplifier(BLSA) 를 지나기 전에 연산이 진행되는지, 이를 지낸 후에 연산이 진행되는지에 따라 나눌 수 있다. BLSA 는 DRAM 내부의 커패시터 내에 저장된 전압이 시간이 지남에 따라 커패시터가 방전되면서 생기는 전압 변동을 보정해 주는 역할을 한다. Analog PIM 은 BLSA 이전에 연산을 진행하는 PIM 을 말한다. 하지만 이는 아날로그식 연산법을 사용하기에 scalability 가 좋지 않고, 노이즈에 취약하며, 이를 보정해주는 기법이 필요하다.[5]



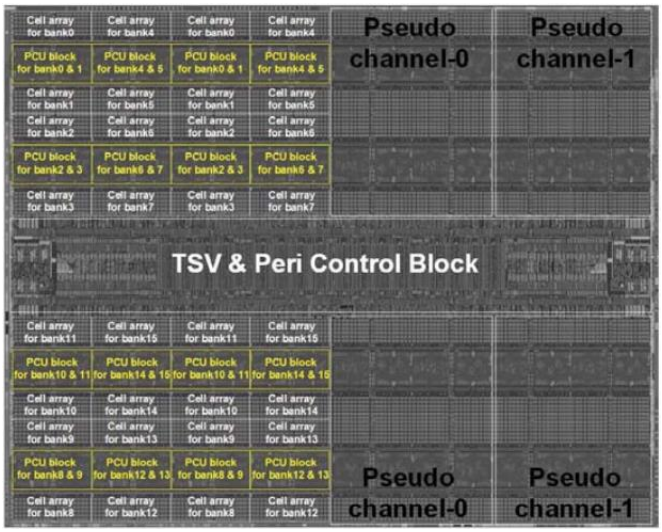
(그림 2) Sense Amplifier 를 활용한 논리 연산[6]

binary weight 를 가진 CNN 을 대상으로 GPU 에 비해 8.8 배 성능향상을 기록한 DRISA[6]의 경우 sense amplifier 의 특성을 활용하여 논리연산이 가능하도록 하였다. 여러 cell 들을 같이 sense amplifier 에 연결되도록 한 후, 이때의 전압을 활용하여 특정 전압 이상일 경우, 1, 이하일 경우 0 이 결과로 나오도록 하였다. 이를 통해 논리연산 AND, OR, NOT 를 모두 구현할 수 있으며, 앞에서 말한 노이즈에 대비해 미리 마진을 확보해 놓아 오차의 발생을 최소화 시키려 하였다. 여러 column 에서 연산을 동시에 실행할 수 있으며 큰 대역폭을 활용할 수 있다.

2-2 Digital PIM

Digital PIM 은 BLSA 를 거친 후에 연산이 되는 PIM 을 말한다. BLSA 를 거친 후이기 때문에 Analog PIM 과 같이 정확도에 대한 고려를 하지 않아도 된다는 장점이 있다. Analog PIM 과 같이 큰 메모리 대역폭을 사용할 수 있다. 하지만 DRAM 공정을 사용하여 논리 회로를 만드는 부분에서 상당한 불이익을 감수하게 된다. DRAM 공정에서 만들어지는 트랜지스터는 커패시터에서의 leakage current 를 최소화하는 것이 중요하기에 threshold voltage 가 높도록 설계된다. 하지만 이런 공정을 사용하여 논리 회로를 만들게 되면 만들어진 논리회로의 성능이 제약되게 되며 논리회로 공정을 사용하였을 때에 비해 매우 큰 면적을 사용하게 되는 등의 불이익이 존재한다.[7] 따라서 이런 불이익을 감수하면서 PIM 을 활용하려면 PIM 의 장점인 높은 메모리 대역폭과 병렬성을 활용하는 것이 중요하다.

Newton[5]의 경우 area overhead 를 줄이기 위해 코어를 새로 만들지 않고 각 DRAM bank 별로 FP16 곱셈기와 덧셈기와 같은 연산에 필요한 최소한의 논리회로만 배치하도록 하였다. 각 bank 별로 matrix-vector 곱셈이 가능하기에 병렬적으로 매우 많은 연산을 동시에 진행하면서 DNN 연산에 필요한 matrix-vector 연산을 가속할 수 있다. 또한 기존의 DRAM command 에 PIM 명령을 위한 command 를 더 추가한 형태이기에 기존 시스템에 적용하기 비교적 용이하다고 할 수 있다.



(그림 4) programmable computing unit 의 칩 내부 배치[8]

삼성에서 개발한 HBM2 기반 programmable computing unit[8]의 경우 Newton 처럼 computing core 를 포기하는 대신, bank 일부를 만들지 않고 그 자리에 computing unit 을 추가하여 연산을 할 수 있도록 하였다. 사용할

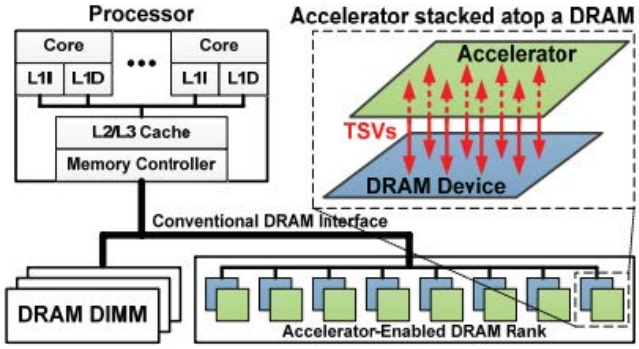
수 있는 메모리 용량이 줄었기에 큰 불이익이라고 보일 수도 있으나, 이를 역으로 활용하여 더 이상 필요 없어진 address 핀 하나를 활용해서 PIM 명령들을 전달할 수 있도록 하였다. 따라서 기존의 DRAM 칩과 동일한 핀 배치를 가지고 있기에 적용하기 매우 용이하고 reconfigurable 하다는 장점이 있다. 또한 시뮬레이션만 진행한 다른 연구들과 다르게 실제 칩을 제작하여 실현 가능함을 보였다는 것에 의의가 있다.

2-3 ReRAM 활용 PIM

ReRAM 의 경우, 커패시터와 트랜지스터로 이루어진 기존의 DRAM 과는 달리 memristor 라는 소자로 구성된 메모리상에서 연산을 진행한다.

Memristor 소자의 특성을 활용해서 정보가 저장된 memristor 를 다양한 방법으로 연결하여 AND, OR, NOT 등의 연산이 이루어지게 할 수 있다.[9] 하지만 아직 ReRAM 소자가 제대로 상용화되지 않았기에 현실적이지 않다는 문제가 있다.

2-4 Processing Near Memory(PNM)



(그림 5)PNM 의 예시[10]

상기한 DRAM 공정에서 논리회로를 만드려고 할 때의 문제를 해결할 수 있다. DRAM 내부에 논리회로를 만들지 않는 대신 DRAM 칩과 매우 가까운 곳에 가속기나 computing core 를 배치하는 형태이다.

(그림 5)의 경우처럼 DRAM 다이 바로 밑에 가속기 다이를 배치해서 연결하여 latency 를 최소화 하기도 한다.

3.결론

본 논문에서는 memory wall 을 극복하고, 메모리 접근으로 인해 발생하는 latency 와 에너지 소모 등을 최소화하며 제약되지 않은 메모리 대역폭을 활용하여 병렬성을 극대화할 수 있는 PIM 기술의 적용 방법과 구조에 관해서 알아보았다. 아직 PIM 기술이 상용화되지 않고 연구 단계이며 DRAM 공정의 한계 등 기술적 문제가 있지만, 높은 병렬성과 에너지 효율성을 활용한다면 기존의 시스템보다 성능을 끌어올릴 수 있을 것으로 기대된다.

4.Acknowledgemet

이 논문은 2022 년도 BK21 FOUR 정보기술 미래인재 교육연구단에 의하여 지원되었음.

참고문헌

[1] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, “Deep Residual Learning for Image Recognition”, In arXiv preprint arXiv:1512.03385, 2015

[2] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan and Y. Xie, "DRISA: A DRAM-based Reconfigurable In-Situ Accelerator," *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO),USA*, 2017, pp. 288-301.

[3] Y. -H. Chen, T. Krishna, J. S. Emer and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127-138, Jan. 2017

[4] Kun Wu, Guohao Dai, Xing Hu, Shuangchen Li, Xinfeng Xie, Yu Wang, and Yuan Xie. Memory-Bound Proof-of-Work Acceleration for Blockchain Applications. In Proceedings of the 56th Annual Design Automation Conference 2019 (DAC '19). Association for Computing Machinery, USA, Article 177, 1–6.

[5] M. He *et al*., "Newton: A DRAM-maker’s Accelerator-in-Memory (AiM) Architecture for Machine Learning," *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020, pp. 372-385

[6] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan and Y. Xie, "DRISA: A DRAM-based Reconfigurable In-Situ Accelerator," *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*,USA, 2017, pp. 288-301

[7] Yong-Bin Kim and T. Chen, "Assessing merged DRAM/logic technology," 1996 IEEE International Symposium on Circuits and Systems (ISCAS), USA, 1996, pp. 133-136 vol.4

[8] Y. -C. Kwon *et al*., "25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications," *2021 IEEE International Solid- State Circuits Conference (ISSCC), USA*, 2021, pp. 350-352

[9] S. Gupta, M. Imani and T. Rosing, "FELIX: Fast and Energy-Efficient Logic in Memory," 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD),Morocco , 2018, pp. 1-7

[10] A. Farmahini-Farahani, J. H. Ahn, K. Morrow and N. S. Kim, "NDA: Near-DRAM acceleration architecture leveraging commodity DRAM devices and standard memory modules," 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA),USA , 2015, pp. 283-295