반도체공학회 논문지 (TSE: Transactions on Semiconductor Engineering), Vol. 2, No. 1, Jan., 2024
https://doi.org/10.22895/tse.2024.0001 ISSN (On-line) 3022-2028

깊이별 분리 합성곱을 위한 다중 스레드 오버랩 시스톨릭 어레이

Multithreaded and Overlapped Systolic Array for Depthwise Separable Convolution

윤종호 [1], 이승규 [2], 강석형 [1,+]
(Jongho Yoon [1], Seunggyu Lee [1], and Seokhyeong Kang [1,+])

요 약

깊이별 분리 합성곱 (Depthwise Separable Convolution)을 처리할 때, processing element (PE)의 저활용성은 시스톨릭 어레이 (SA)의 한계점 중 하나이다. 본 연구에서는 깊이별 합성곱의 처리량을 극대화하기 위한 새로운 SA 아키텍처를 제안한다. 더불어, 제안된 SA 는 깊이별 합성곱 계산 중에 유휴 PE 에서 후속 점별 합성곱 (pointwise convolution)을 수행하여 활용도를 증가시킨다. 모든 깊이별 합성곱 연산 후에는 모든 PE 를 활용하여 나머지 점별 합성곱 연산의 속도를 향상시킨다. 결과적으로, 제안된 128×128 SA 는 MobileNetV3 연산 시, 기본 SA 및 RiSA 와 비교하여 속도가 4.05 배, 1.75 배 향상되고, 에너지 소비량을 각각 66.7 %, 25.4 % 감소한다.

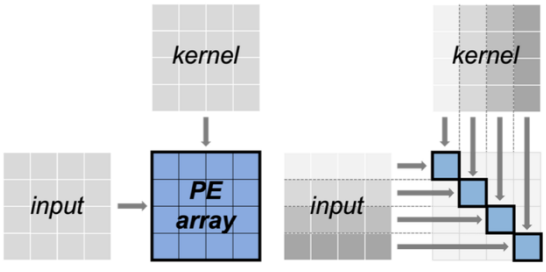
ABSTRACT

When processing depthwise separable convolution, low utilization of processing elements (PEs) is one of the challenges of systolic array (SA). In this study, we propose a new SA architecture to maximize throughput in depthwise convolution. Moreover, the proposed SA performs subsequent pointwise convolution on the idle PEs during depthwise convolution computation to increase the utilization. After the computation, we utilize unused PEs to boost the remaining pointwise convolution. Consequently, the proposed 128x128 SA achieves a 4.05x and 1.75x speed improvement and reduces the energy consumption by 66.7 % and 25.4 %, respectively, compared to the basic SA and RiSA in MobileNetV3.

KEY WORDS
시스톨릭 어레이; 하드웨어 가속기; 깊이별 분리 합성곱;

I. 서 론

Convolutional neural network (CNN)은 다양한 비전 작업에 흔히 사용되는 딥러닝 모델이다. 일반적인 합성곱 (convolution) 연산은 많은 연산량을 필요로 하기 때문에 이를 빠르게 처리하기 위한 다양한 CNN 가속기가 제안되었다. CNN 하드웨어 가속기 중 하나인 시스톨릭 어레이 (SA)는 여러 processing element (PE)가 타일 형태로 구성된 구조이다. SA는 데이터의 재사용률이 높은 특정 유형의 연산, 예를 들어 행렬 곱셈과 같은 작업을



(a) Traditional convolution (b) Depthwise convolution

가속하는데 활용된다. 전통적인 합성곱 연산도 데이터 재사용률이 높으며 데이터 매핑 기법을 통해 행렬 곱셈으로 변환할 수

[1] Department of Electrical Engineering, Pohang University of Science and Technology [+]Corresponding author: Seokhyeong Kang, shkang@postech.ac.kr
[2] Hyosung Ventures
(Received Dec. 15, 2023, Accepted Jan. 04, 2024)

있다 [1]. 따라서 SA는 Google TPU [2] 및 NVIDIA 텐서 코어 [3]와 같은 여러 하드웨어 가속기에 적용된다.

리소스가 제한된 환경에서 CNN 모델의 성능을 유지하기 위해 경량화 및 가속화 연구가 활발히 진행 중이다. 특히, MobileNet [4]–[6] 및 EfficientNet [7]과 같은 모델은 깊이별 분리 합성곱 (depth-wise separable convolution) 을 도입해 작은 모델 크기와 적은 연산량으로도 높은 모델 성능을 달성할 수 있었다. 이후 깊이별 분리 합성곱은 ConvNet [8] 및 CoAtNet [9]과 같은 대규모 모델에도 적용되었다. 그러나 깊이별 분리 합성곱의 한 부분인 깊이별 합성곱은 데이터 재사용률이 낮아 SA에서 가속화하기 어려운 문제가 있다. 일반적인 $\boldsymbol{N \times N}$ SA에서 깊이별 합성곱은 오직 $\boldsymbol{1/N}$ 만 활용할 수 있다 [12].

이러한 낮은 PE 활용률을 개선하기 위한 여러 방법이 제안되었다. PE 어레이를 하위 어레이로 분할하여 개별적으로 활용하는 방법도 있지만 [10], [11], 이러한 방법은 여러 개의 독립적인 작업을 병렬로 처리하기 때문에 클라우드 기반 가속기 등 대규모 SA에서만 그 효과적이다. 또한 깊이별 분리 합성곱의 성능 개선은 어레이의 크기나 하위 어레이의 수에 따라 달라질 수 있다. SA에서 깊이별 분리 합성곱을 가속화하기 위해 데이터 흐름을 재구성하는 방법도 도입되었지만 [12], [13], 이 방법은 팬아웃이 높거나 여전히 깊이별 분리 합성곱 계산 시 대부분의 PE가 유휴 상태로 남아 있다는 단점이 있다. 또한 중복된 곱셈으로 인해 PE의 처리량 (throughput)이 최대화 되지 않는다. 따라서 이러한 문제들을 해결하고 깊이별 분리 합성곱에서 처리량과 PE 활용도를 높이기 위한 새로운 SA가 필요하다.

본 논문에서는 불필요한 곱셈을 제거하기 위해 새로운 PE 구조를 제안한다. 1D PE 체인 구조에 수직 데이터 라인을 추가하여 각 PE가 적절한 데이터 라인을 선택하여 불필요한 곱셈을 제거하고 깊이별 분리 합성곱의 처리량을 극대화한다. 또한, 깊이별 분리 합성곱을 가속화하면서도 표준 합성곱 연산 속도에 영향을 주지 않도록 유연한 데이터 라인을 통해 유휴 PE에 다른 연산을 할당할 수 있다. 깊이별 분리 합성곱 연산이 완료되면 사용하지 않는 PE를 재할당하여 처리량과 PE 활용도를 높이는 방법을 제시한다.

본 논문의 구성은 다음과 같다. 제 II 장에서는 다양한 CNN 가속기 연구들에 대해 살펴보고, 새로운 SA 구조와 PE 재할당 방법을 제안한다. 제 III 장에서는 연산 가속, 합성 면적 및 에너지 소모 등의 결과를 제시하고 제 IV 장에서 논문을 마무리한다.

그림 1. SA에서 합성곱 별 PE 활용률

II. 본 론

1. 배경 및 관련 연구

(1) 시스톨릭 어레이

SA는 여러 개의 PE로 구성된 하드웨어 가속기로, 주로 일반 행렬 곱셈 (GEMM)을 위한 용도로 사용된다. 이 SA는 2차원 어레이 형태로 배치되며, 데이터는 배열의 가장자리 PE에 입력되거나 출력된다. 내부 PE는 인접한 PE로부터 데이터를 전달받는다. 데이터가 PE로 입력되면 동일한 행 또는 열의 모든 PE를 통과할 때까지 재사용된다. 각 PE는 입력, 가중치 및 부분합을 입력으로 받으며, 일반적으로 두 가지가 다음 인접 PE로 전달되고, 나머지는 PE 내에서 반복적으로 사용된다. 따라서 SA는 데이터 를 높은 재사용 비율로 활용하며, 외부 메모리 액세스 횟수를 줄여 높은 계산 속도를 달성할 수 있다. 합성곱 연산도 convolution lowering과 같은 데이터 매핑 방법을 사용하여 GEMM으로 변환할 수 있으며, 그림 1(a)와 같이 SA에서 계산이 가능하다.

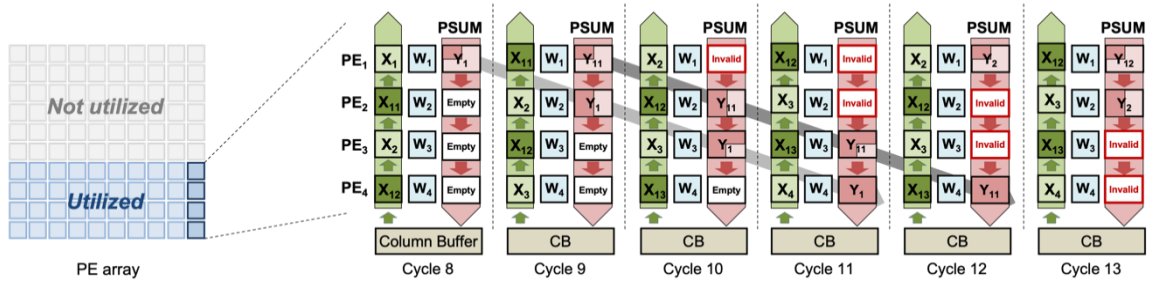


그림 2. 두 개의 입력 데이터 스트림이 있는 1D PE 체인의 깊이별 합성곱 메커니즘

(2) 깊이 별 분리 합성곱

깊이 별 분리 합성곱은 경량 CNN 모델에서 우수한 성능을 보이며, 이 연산은 채널 별로 합성곱을 수행하는 깊이별 합성곱(depthwise convolution)과 그 결과를 하나의 채널로 합치는 점별 합성곱(pointwise convolution)으로 구성된다. 깊이별 합성곱은 각 채널에 대해 개별적으로 수행되기 때문에 그림 1(b)와 같이 SA에서 PE의 활용률이 낮다. 반면 점별 합성곱은 입력 채널을 단일 채널로 압축하는 1×1 커널을 사용한다. 따라서 점별 합성곱은 데이터 재사용률이 높으며, SA 구조에서 PE 활용률을 높일 수 있다.

(3) 관련 연구

낮은 PE 활용률 문제를 해결하기 위해 PE 어레이를 여러 하위 PE 어레이로 분할하는 방법이 고려되었다. [10]은 PE 어레이를 여러 하위 PE 어레이로 분할하고 각 하위 PE 어레이에 서로 다른 작업을 할당하는 방식을 소개한다. [11]은 데이터 흐름 미러링(dataflow mirroring)을 도입하여 PE 어레이를 세밀하게 직사각형 모양의 하위 PE 어레이로 분할하는 방법을 제안한다.

그러나 이러한 방법들은 여전히 대각선 PE가 필요한 깊이별 합성곱과 같은 작업에서 PE 활용도가 낮을 수 있다. 또한 유휴 PE를 효과적으로 활용하기 어렵다. 이러한 문제를 해결하기 위해 SA 구조 기반으로 깊이별 합성곱을 가속화하는 다양한 방법이 제안되었다. [13]은 broadcasting line을 사용하여 깊이별 합성곱을 가속화하는 방법을 제시했다. 그러나 이 구조는 팬아웃이 많아서 설계 비용이 높아 PE 어레이의 크기를 확장하기 어렵다. [15]는 heterogenous SA를 도입하여 여러 데이터 흐름을 처리할 수 있지만, 작은 SA (8개 혹은 16개의 PE 사용)에서만 높은 PE 활용도를 달성할 수 있다. 또한 복잡한 구조의 전용 (dedicated) 하드웨어는 설계 비용이 저렴한 SA의 장점을 활용하지 못한다.

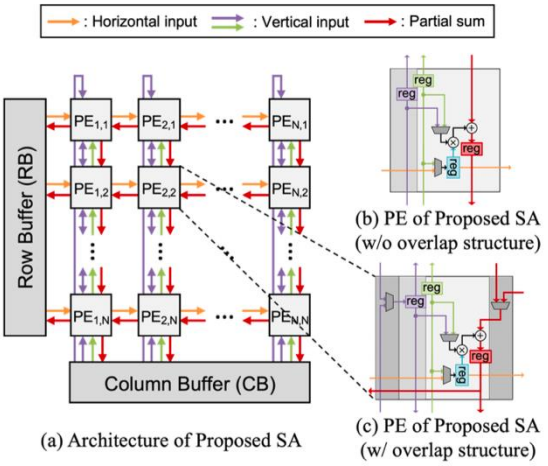


그림 3. 제안하는 SA 구조

이러한 구조적인 문제를 해결하기 위해 [12]는 PE 어레이를 독립적인 1D PE 체인 집합으로 처리하는 RiSA를 제안했다 (그림 2). 이 방법은 $K \times K$ 커널을 column−major order로 펼치고 PE 체인에 미리 로드하며, 입력 데이터를 해당 순서에 맞게 입력하는 방식을 사용한다. 입력 및 부분합 데이터가 각각 위아래로 이동함에 따라 입력 데이터는 두 사이클마다 공급되어야 하므로 입력 데이터 라인에 빈 슬롯이 발생한다. 따라서 각 입력 데이터 라인은 두 개의 입력 데이터 스트림으로 구성되어 두 행의 출력 데이터가 생성된다. 이 방식은 깊이별 분리 합성곱의 PE 활용률과 처리량을 증가시켰지만 여전히 PE 어레이를 완전히 활용하지 못한다. 1D PE 체인에서 각 입력 데이터는 평면화 된 $K \times K$ 크기의 커널을 $K \times K$ 번 곱한다. 그러나 각 입력 데이터는 K 개의 출력 데이터를 계산하기 위해 사용된다. 즉, 유효한 곱셈은 K 개 뿐이며 나머지 $(K-1) \times K$ 개의 곱셈은 유효하지 않은 부분합을 생성한다. 또한, RiSA는 1D PE 체인의 개수 S에 대해 $K^2 \times S$ 모양의 PE 어레이만 사용한다. 따라서 나머지 $(S-K^2) \times S$ 모양의 PE 어레이는 깊이별 합성곱에서 사용되지 않는다.

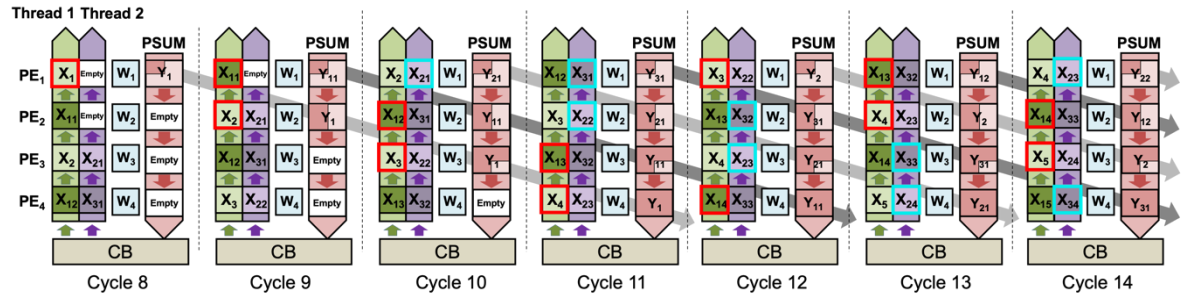


그림 4. 제안하는 다중 스레드 1D PE 체인의 깊이별 합성곱 메커니즘

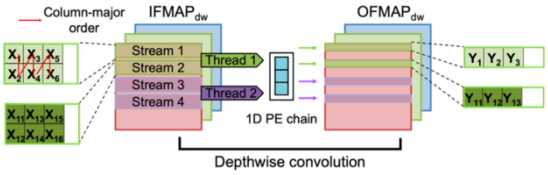


그림 5. 1D PE 체인의 입력 데이터 흐름

2. 제안 방법

본 논문에서는 곱셈 연산 과정의 유효하지 않는 부분합을 제거하기 위해 다중 스레드 입력 데이터 스트림을 지원하는 새로운 1D PE 체인을 제안한다. 이로써 다중 스레드 구조는 깊이별 합성곱 연산에서 처리량과 PE 활용률을 향상시킬 수 있다. 또한, 점별 합성곱 연산에서 유휴 PE를 활용하기 위한 오버랩 구조와 PE 재할당 방법도 제안한다. 그림 3은 제안하는 SA 및 PE 구조를 나타낸다.

(1) 1D PE 체인의 다중 스레드 구조

기존 1D PE 체인 구조 [12]에서는 곱셈 연산을 수행할 때 유효하지 않은 부분합이 발생하는 문제가 있다. 이 문제를 해결하기 위해 각 스레드에 대해 두 개의 입력 데이터 스트림을 사용하는 다중 스레드 1D PE 체인 구조를 제안한다. 각 데이터 라인은 $K \times K$ 크기의 커널에서 K번의 유효한 곱셈을 수행한다. 따라서 각 1D PE 체인에서 곱셈을 완전히 활용하기 위해 $K-1$개의 추가 데이터 라인을 도입한다. 제안하는 SA는 추가 데이터 라인을 지원하기 위해 각 PE에 $K-1$ 개의 추가 레지스터를 필요로 한다 (그림 3(b)). 그림 4는 2×2 커널과 두 개의 스레드를 사용하는 경우 1D PE 체인 구조에서 깊이별 합성곱 계산을 위한 입력 및 부분합 데이터의 이동을 보여준다. $K \times K$ 크기의 커널의 값은 $K \times K$ 사이클 동안 column-major order로 펼쳐져 1D PE 체인에 미리 로드된다. 이후 그림 5와 같이 $2 \times K$ 개의 입력 데이터 스트림이 K 개의 스레드를 통해 공급된다. 각 입력 스트림은 높이 K 의 입력 특징맵 ($IFMAP_{dw}$)으로 구성된 데이터이다. 입력 데이터가 PE 체인의 맨 위에 도달하면 부분합 데이터의 계산이 시작된다 (그림 4의 Cycle 8). 각 PE는 K 사이클마다 다른 데이터 스트림을 선택하고 유효한 부분합을 계산한다.

(2) 유휴 PE 활용을 위한 오버랩 구조

일반적인 SA에서는 대각선 PE만 깊이별 합성곱에 사용되며 대부분의 PE는 사용되지 않는다. 또한 나머지 유휴 PE의 모양이 직사각형이 아니기 때문에 PE 어레이를 분할하여 사용할 수 없다. 그러나 1D PE 체인 구조에서는 유휴 PE의 모양이 직사각형이기 때문에 PE 어레이를 분할하여 사용할 수 있다. 기존 1D PE 체인은 유효하지 않은 부분합을 생성하는데, 이로 인해 매 사이클마다 깊이별 합성곱의 출력 피처맵 ($OFMAP_{dw}$) 을 생성할 수 없어서 이어지는 점별 합성곱을 연속적으로 계산할 수 없다. 하지만 제안하는 1D PE 체인 구조는 처리량을 개선하여 점별 합성곱을 연속적으로 계산할 수 있다. 이로써 유휴 PE를 활용하고 처리량을 감소시키지 않으면서 점별 합성곱을 계산하기 위해 두 개의 데이터 라인을 추가한 오버랩 구조를 제안한다.

그림 6은 오버랩 구조에서 두 종류의 합성곱을 동시에 계산하는 메커니즘을 보여준다. 제안하는 SA에서 $S \times S$ 크기의 PE 어레이는 두 개의 PE 어레이로 분할된다. 파란 PE 어레이는 깊이별 합성곱에 사용되는 $K^2 \times S$ 크기의 PE 어레이이고, 녹색 PE 어레이는 점별 합성곱에 사용되는 $(S-K^2) \times S$ 크기의 PE 어레이이다. 제안하는 SA는 각 1D PE 체인에서 $K \times K$ 크기의 커널을 미리 저장하고, $2K$ 개의 입력 데이터 스트림이 K 스레드를 통해 PE 체인에 입력된다. 첫번째 $OFMAP_{dw}$ 값이 생성되면 배치 정규화 및 활성화 함수를 거쳐 점별 합성곱의 입력 피처맵 ($IFMAP_{pw}$) 값이 된다. 이 값은 추가 데이터 라인을 통해 녹색 PE 어레이로

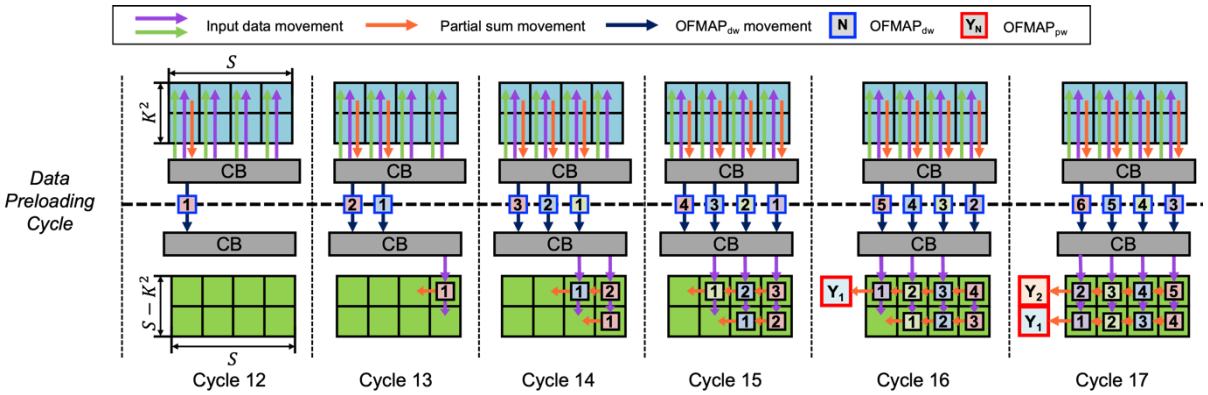


그림 6. 오버랩 구조에서 깊이별 합성곱 및 점별 합성곱을 동시에 수행하는 메커니즘

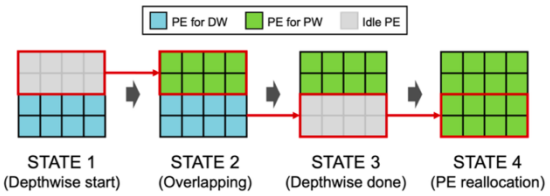


그림 7. 제안하는 PE 재할당 방법

이동하여 점별 합성곱이 이루어진다. 1D PE 체인의 다중 스레드 구조는 매 사이클마다 $OFMAP_{dw}$ 값을 생성하므로, 제안된 오버랩 구조에서 PE 어레이의 처리량을 최대화할 수 있다.

녹색 PE 어레이에서 M 채널로 구성된 점별 합성곱을 계산하기 위해 M 개의 커널을 $S-K^2$ 으로 나누어 타일링 한다. 일반적으로 M 은 $S-K^2$ 보다 크기 때문에 점별 합성곱 계산 사이클이 깊이별 합성곱 계산 사이클보다 길다. 따라서 오버랩 구조는 점별 합성곱 연산이 깊이별 합성곱 연산을 숨기는 효과가 있어 전제 깊이별 분리 합성곱의 처리량과 PE 활용도를 높인다.

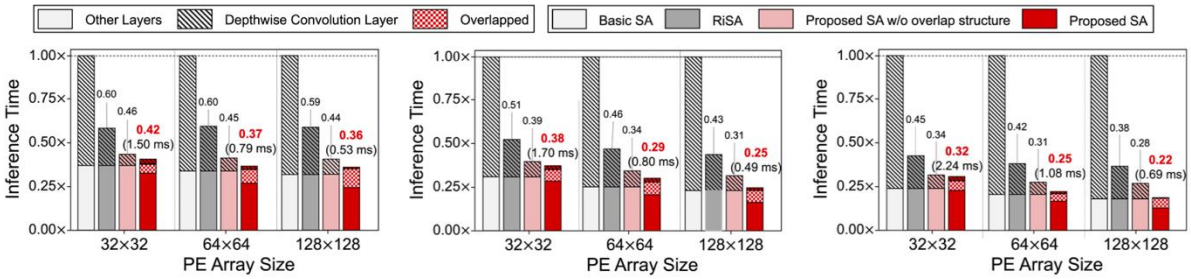
(3) PE 활용률 향상 위한 PE 재할당 방법

제안하는 오버랩구조는 두 종류의 합성곱을 동시에 계산할 때 높은 PE 활용률을 달성하지만 대부분의 연산 사이클은 점별 합성곱이 차지한다. 이때 점별 합성곱의 연산 사이클 수는 PE 어레이 크기에 의해 결정된다. 그러나 그림 6의 녹색 PE 어레이는 PE 어레이 전체가 아닌 일부분이며 ($(S-K^2)\times S$ 의 크기) 이로 인한 사이클 오버헤드가 발생한다 [16]. 이 오버헤드가 오버랩 구조로 인한 연산 사이클 감소량보다 크다면 오히려 성능이 저하될 수 있다. 따라서 이러한 문제를 해결하기 위해 PE 재할당 방법을 제안한다.

그림 7은 PE 재할당 방법의 과정을 보여준다. 깊이별 합성곱 계산 후 파란 PE 어레이는 새로운 유휴 PE 어레이로 변경된다 (그림 7의 STATE 3). 이 새로운 유휴 PE 어레이는 남은 깊이별 합성곱 연산을 위해 재할당 된다 (그림 7의 STATE 4). 복잡한 제어를 피하기 위해 타일링한 연산이 완료될 때까지 새로운 PE 어레이로의 재할당은 수행되지 않는다. 이러한 PE 재할당 방법은 입력 데이터의 방향만 변경하면 되기 때문에 동일한 PE 구조를 유지하면서 처리량과 PE 활용률을 더욱 향상시킬 수 있다.

III. 결과 분석

본 논문에서는 기본 SA, RiSA [12], 그리고 제안하는 SA 를 RTL 로 구현했다. 주로 깊이별 분리 합성곱에서 3×3 커널을 사용하므로 제안하는 SA 는 트리플 스레드 구조로 구현했다. 오버랩 구조에서 깊이별 합성곱과 점별 함성곱 사이에 배치 정규화 및 활성화 함수가 필요하다. 그러나 배치 정규화는 합성곱 레이어와 병합할 수 있고, 활성화 함수의 복잡도는 $O(1)$ 로 무시할 수 있다. 따라서 추론 시간을 측정 시 이러한 부분들에 대한 연산 시간은 무시했다. 추론 시간은 MobileNetV2 [5], MobileNetV3 [6], 그리고 EfficientNet [7]에 대해 RTL 시뮬레이션을 기반으로 측정했다. 또한 SA 의 대부분의 면적과 에너지 소모는 PE 어레이에서 발생하므로 PE 어레이 값 만을 비교한다 [12]. RTL 로 설계된 회로는 $28nm$ PDK 를 이용하여 300MHz 클럭 디자인으로 합성되었고, 총 에너지 소비량은 0.1 의 활동 계수 (activity factor)를 가정하여 계산했다.



(a) MobileNetV2 [5] (b) MobileNetV3 [6] (c) EfficientNet [7]

그림 8. 기본 SA, RiSA 그리고 제안하는 SA에 대한 여러 CNN 모델의 추론 시간 비교

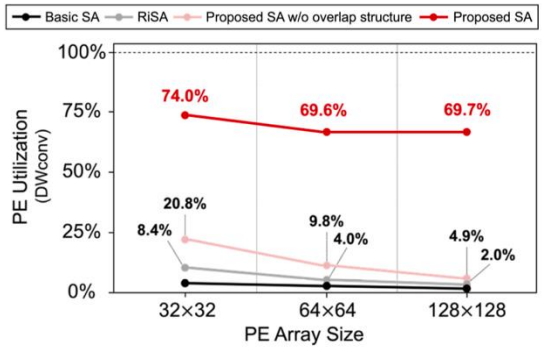


그림 9. 깊이별 합성곱의 PE 활용률

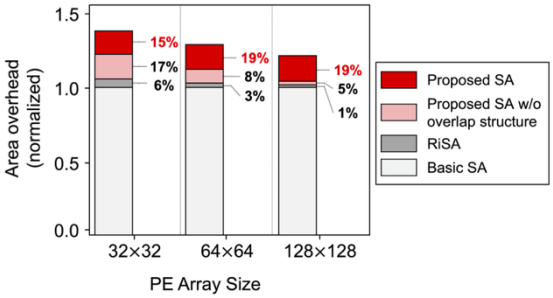


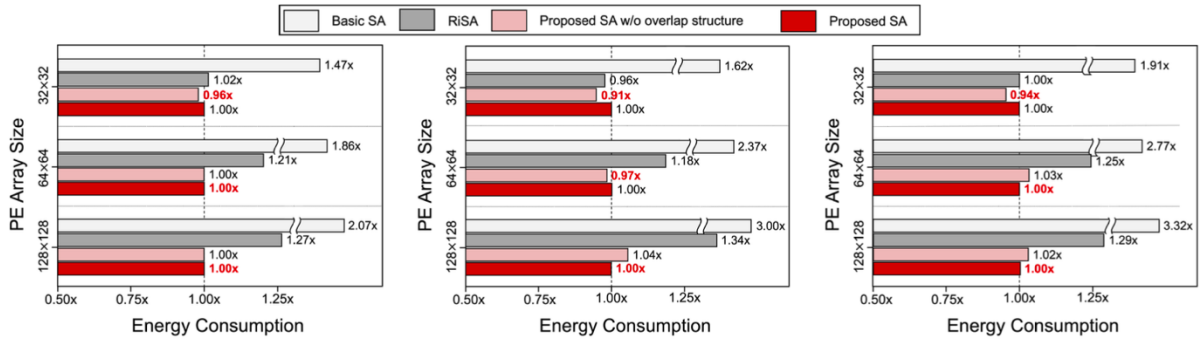
그림 10. 각 SA 디자인 별 면적 오버헤드

1. PE 활용률과 추론 시간 비교

그림 9 에서 보여지는 바와 같이, 각 SA 에 따른 깊이별 합성곱 연산의 PE 활용률을 비교한다. RiSA 와 제안하는 SA 는 모두 1D PE 체인 구조를 기반으로 하므로 동일한 수의 PE 가 사용된다. 그러나 제안하는 SA 는 다중 스레드 구조를 사용하여 PE 활용률을 높이므로, 제안하는 SA 는 평균적으로 RiSA 대비 2.5 배 더 높은 PE 활용률을 달성했다. 이때 PE 활용률은 PE 의 부분합 연산이 유효한지 여부를 고려한 것이다. 또한 그림 8 에서는 여러 CNN 모델에 대한 추론 시간을 비교한 결과를 나타낸다. 실험에 사용된 CNN 모델들은 3×3 크기의 커널 외에도 큰 커널을 사용했지만, 제안하는 다중 스레드 구조는 커널의 크기에 관계없이 깊이별 분리 합성곱 연산에서 유효한 부분합의 비율을 높여 처리량을 높인다. 64×64 크기의 SA를 사용하는 경우, 깊이별 합성곱 연산에 대해 기본 SA 및 RiSA 대비 MobilNetV3 에서 8.5 배 및 2.4 배, EffcientNet 에서 9.7 배 및 2.4 배의 속도 향상을 달성했다.

2. 면적 및 에너지 소비량 비교

그림 10 은 세 가지 크기의 SA 설계의 면적 오버헤드를 나타내고, 이 값들은 기본 SA 의 면적으로 정규화 한 값이다. 그림 11 은 각 SA 의 에너지 소비량을 나타낸다. 제안하는 SA 가 다양한 크기에 대해서 모두 가장 적은 에너지를 소모하는 것을 알 수 있다. 제안하는 SA 는 각 PE 에 추가적인 두 개의 레지스터와 다중 스레드 및 오버랩 구조에 대한 로직이 추가되어 면적 및 에너지 소모량이 증가했다. 그러나 PE 어레이 크기가 증가함에 따라 다중 스레드 구조의 오버헤드는 32×32 SA 의 23 %에서 128×128 SA 의 6 %로 감소했다. 반면 오버랩 구조는 SA 의 크기와 상관없이 무시할 수 없는 오버헤드가 발생하여 추론 속도 향상과 에너지 소모 절감이 보장될 때 적용할 수 있다. 예를 들어, 32×32 크기의 SA 에서 오버랩 구조로 인한 추론 속도 향상 효과가 미미해 다중 스레드 구조만 채택한 경우, 제안한 SA 는 MobileNetV3 의 경우 기본 SA 및 RiSA 와 비교하여 에너지 소비를 각각 43.9 % 및 6.3 %, Efficient-Net 의 경우 51.8 % 및 6.0 % 감소시켰다. 반면, 오버랩 구조는 큰 SA 에서 추론 시간을 크게 감소시켰으며 이로 인해 에너지



(a) MobileNetV2 [5] (b) MobileNetV3 [6] (c) EfficientNet [7]

그림 11. 기본 SA, RiSA 그리고 제안하는 SA에 대한 여러 CNN 모델의 에너지 소모량 비교

소비가 낮아졌다. 결과적으로 128 × 128 크기의 SA 에서 MobileNetV3 의 경우 에너지 소비가 기본 SA 및 RiSA 대비 각각 66.7 % 및 25.4 %, EfficientNet 의 경우 70.9 % 및 22.5 % 감소했다.

이때 여러 SA 크기에 대해 더 적합한 구조가 무엇인지 분석하기 위해 표 1 에서 기본 SA 를 기준으로 SA 디자인 별 면적과 추론 속도 사이의 비율인 면적 효율성을 비교한 결과를 보여준다. 32×32 SA 에서 다중 스레드 구조만 적용한 경우에 기본 SA 및 RiSA 와 비교하여 효율성이 각각 2.39 배 및 1.13 배 향상되었으며, 128×128 SA 에서 오버랩 구조를 적용한 경우에 기본 SA 및 RiSA 와 비교하여 효율성이 각각 3.59 배 및 1.37 배 향상되었다.

IV. 결 론

본 논문에서는 깊이별 분리 합성곱에서 처리량과 PE 활용률을 높이기 위해 다중 스레드 및 오버랩 구조를 적용한 SA를 제안한다. 다중 스레드 구조를 활용하여 깊이별 합성곱의 유효하지 않은 부분합을 제거하고 처리량을 향상시켰다. 또한 깊이별 합성곱 계산 중 발생하는 유휴 PE를 오버랩 구조를 통해 활용해 동시에 점별 합성곱을 계산한다. 깊이별 합성곱이 완료된 후에 발생하는 유휴 PE는 재할당 방법을 통해 남은 점별 합성곱을 계산하는데 활용한다. 결과적으로 깊이별 분리 합성곱에서 PE 활용률을 평균적으로 70 %까지 높이면서 처리량 역시 높일 수 있었다.

표 1. SA 디자인 및 어레이 크기 별 면적 효율성 (* 오버랩 구조 없는 디자인)

		EfficientNet
32	RiSA	2.10
	Ours*	**2.39**
	Ours	2.29
64	RiSA	2.31
	Ours*	2.93
	Ours	**3.12**
128	RiSA	2.62
	Ours*	3.40
	Ours	**3.59**

Acknowledgement

The EDA Tool was supported by the IC Design Education Center.

참고 문헌

[1] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran et al. “cuDNN: Efficient Primitives for Deep Learning”, *arXiv preprint arXiv:1410.0759*, 2014.

[2] N.P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal et al., “In- Datacenter Performance Analysis of a Tensor Processing Unit”, *Int. Symp. on Computer Architecture (ISCA)*, 2017, pp. 1-12.

[3] S. Markidis, S. W. D. Chien, E. Laure, I. B. Peng, J. S. Vetter, “NVIDIA Tensor Core Programmability, Performance & Precision”, *Int. Symp. on Parallel and Distributed Processing Symp. Workshops (IPDPSW)*, 2018, pp. 522-531.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang et al., “Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, *arXiv preprint arXiv:1704.04861*, 2017.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginow, L. C. Chen, “Mo- bileNetV2: Inverted Residuals and Linear

Bottlenecks", *Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510-4520.
[6] A. Howear, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, "Searching for MobileNetv3", *Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 1314-1324.
[7] M. Tan, Q. Le, "Efficientnet: Rethinking Model Scaling for Convolu- tional Neural Networks", *Proc. Machine Learning Research (PMLR)*, 2019, pp. 6105-6114.
[8] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, "A ConvNet for the 2020s", *arXiv preprint arXiv:2201.03545*, 2022.
[9] Z. Dai, H. Liu, QV. Le, M. Tan, A. Howear, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, "CoAtNet: Marrying Convolution and Attention for All Data Sizes", *Advances in Neural Information Processing Systems 34*, 2021, pp. 3965-3977.
[10] S. Ghodrati, B. H. Ahn, J. Kim, S. Kinzer, B. R. Yatham et al., "Planaria: Dynamic Architecture Fission for Spatial Multi-Tenant Acceleration of Deep Neural Networks", *Int. Symp. on Microarchitecture (MICRO)*, 2020, pp. 681-697.
[11] J. Lee, J. Choi, J. Kim, J. Lee, Y. Kim, "Dataflow Mirroring: Archi- tectural Support for Highly Efficient Fine-Grained Spatial Multitasking on Systolic Array NPUs", *Design Automation Conf. (DAC)*, 2021, pp. 247-252.
[12] H. Cho, "RiSA: A Reinforced Systolic Array for Depthwise Convolu- tions and Embedded Tensor Reshaping", *Trans. Embedded Computing Systems (TECS) 20.5s*, 2021, pp. 1-20.
[13] R. Xu, S. Ma, Y. Wang, Y. Guo, "CMSA: Configurable Multi-directional Systolic Array for Convolutional Neural Networks", *International Con- ference on Computer Design (ICCD)*, 2020, pp. 494-497.
[14] L. Bai, Y. Zhao and X. Huang, "A CNN Accelerator on FPGA Using Depthwise Separable Convolution", *IEEE Trans. Circuits and Syst. II, Exp. Briefs*, vol. 65, no. 10, pp. 1415-1419, Oct. 2018.
[15] R. Xu, S. Ma, Y. Wang, Y. Guo, "HeSA: Heterogeneous Systolic Array Architecture for Compact CNNs Hardware Accelerators", *Design, Automation & Test in Europe Conf. & Exhibit. (DATE)*, 2021, pp. 657- 662.
[16] H. T. Kung, B. McDanel, S. Q. Zhang, "Adaptive Tiling: Apply Fixed- size Systolic Arrays to Sparse Convolutional Neural Networks", *Int. Conf. on Pattern Recognition (ICPR)*, 2018, pp. 1006-1011.

윤 종 호 (Jongho Yoon)

2021년 2월 : 포항공과대학교 전자전기공학과 졸업
2021년 3월~현재 : 포항공과대학교 전자전기공학과 통합과정

<관심분야> ML-EDA, 하드웨어 가속기 설계

이 승 규 (Seunggyu Lee)

2020년 8월 : 포항공과대학교 전자전기공학과 졸업
2022년 8월 : 포항공과대학교 전자전기공학과 석사

<관심분야> 하드웨어 가속기 설계

강 석 형 (Seokhyeong Kang), 정회원

1999년 2월 : 포항공과대학교 전자전기공학과 졸업
2001년 2월 : 포항공과대학교 전자전기공학과 석사
2013년 8월 : UC San Diego 전기컴퓨터공학과 박사
2014년 8월~2018년 3월 : UNIST 전기전자공학과 조교수
2018년 3월~현재 : 포항공과대학교 전가전기공학과 부교수

<관심분야> 저전력 디자인 최적화, VLSI CAD, SoC 설계