DSP Packing을 활용한 Edge FPGA 기반 Neural Network 가속화 연구

이창근, 임완수*
성균관대학교
2chang0906@skku.edu, *wansu.lim@skku.edu

A Study on DSP Packing-Based Neural Network Acceleration for Edge FPGAs

Lee Changgeun, Lim Wansu*
Sungkyunkwan Univ.

요 약

본 논문에서는 edge FPGA 환경에서 신경망 가속을 위한 DSP 자원 효율 향상을 목적으로 DSP packing 기반 연산 구조를 제안한다. 최근 딥러닝 기반 인공지능 모델은 행렬 곱셈 연산에 의해 연산량이 급격히 증가하고 있으며, 제한된 자원을 갖는 edge 환경에서는 DSP 활용 효율이 전체 성능을 결정하는 핵심 요소로 작용한다. 그러나 HLS 기반 FPGA 설계 환경에서는 단순한 연산 비트폭 축소만으로 DSP 사용량 감소를 기대하기 어려운 한계가 존재한다. 이를 해결하기 위해 본 연구에서는 두 개의 저정밀 곱셈 연산을 하나의 DSP 곱셈 연산으로 결합하는 DSP packing 기법을 적용하였다. 특히 인공지능 모델 양자화 흐름을 반영하여 INT4×INT8 및 INT4×INT4 곱셈을 대상으로 HLS 기반 모델링 및 합성을 수행하였다. 제안한 구조에서는 입력 값을 unsigned 영역으로 변환한 뒤 비트 단위 packing을 통해 단일 곱셈으로 두 개의 곱셈 결과를 생성하고, 비트 추출 및 보정 연산을 통해 원래의 signed 곱을 복원한다. HLS 합성 결과, DSP packing을 적용한 저정밀 구조는 baseline INT8×INT8 구조 대비 DSP 사용량을 절반으로 감소시키면서도 연산 지연을 유사한 수준으로 유지하였다. 이러한 결과는 제안한 DSP packing 기법이 자원이 한정된 edge FPGA 환경에서 양자화된 신경망 추론을 효율적으로 가속하기 위한 설계 기법이 될 수 있음을 보여준다.

Ⅰ. 서 론

최근 딥러닝 기반 인공지능 모델은 자연어 처리, 컴퓨터 비전 등 다양한 분야에서 우수한 성능을 보이며 폭넓게 활용되고 있다. 이러한 모델의 연산량은 지속적으로 증가하고 있으며, 특히 행렬 곱셈 연산은 전체 연산량의 대부분을 차지하는 핵심 연산으로 작용한다. 한편, 실시간 처리와 저전력 동작이 요구되는 edge 환경에서는 제한된 하드웨어 자원 내에서 신경망 추론을 효율적으로 수행하는 것이 중요한 과제로 대두되고 있다.[1]

FPGA는 병렬 처리 구조와 전력 효율 측면에서 신경망 가속기로 활용될 수 있는 유연한 하드웨어 플랫폼이다. 특히 다양한 연산 정밀도를 지원하고, 사용자 정의 데이터 경로를 구성할 수 있다는 점에서 edge 환경에 적합한 가속기로 주목받고 있다. 그러나 edge FPGA는 서버급 FPGA에 비해 가용 자원이 제한적이며, 대규모 행렬 곱셈 기반의 인공지능 연산을 그대로 적용하기에는 자원 부족 문제가 발생할 수 있다. 이 중에서도 곱셈 및 누산 연산이 주로 매핑되는 DSP 블록은 전체 연산 성능을 좌우하는 핵심 자원으로, DSP를 얼마나 효율적으로 활용하느냐가 edge FPGA 기반 신경망 가속 성능을 결정하는 중요한 요소로 작용한다.[2]

하지만 HLS 기반 FPGA 설계 환경에서는 단순히 연산 비트폭을 줄이더라도 DSP 사용량이 자동으로 감소하지 않는 경우가 많다. 이는 HLS 컴파일러가 각 곱셈 연산을 독립적인 DSP 연산으로 매핑하기 때문이며, 결과적으로 저정밀 연산의 장점이 충분히 하드웨어 자원 절감으로 이어지지 못한다. 특히 edge FPGA 환경에서는 이러한 한계가 DSP 자원 낭비로 직결되어, 전체 시스템 성능과 확장성을 제한하는 요인으로 작용한다.[3]

본 논문에서는 이러한 문제를 해결하기 위해, edge FPGA 환경에서 신경망 연산의 DSP 자원 효율을 향상시키기 위한 DSP packing 기반 가속 구조를 제안한다. 제안한 구조는 저정밀 연산을 단순히 적용하는 것이 아니라, DSP 내부 연산 구조를 고려하여 두 개의 저정밀 곱셈을 하나의 DSP 곱셈 연산으로 결합함으로써 DSP 사용량을 효과적으로 감소시키는 것을 목표로 한다. 이를 통해 제한된 DSP 자원을 갖는 edge FPGA 환경에서도 신경망 연산의 효율적인 가속이 가능함을 보이고자 한다.[4]

Ⅱ. 본론

본 연구에서는 edge FPGA 환경에서 신경망 연산의 DSP 자원 효율을 분석하기 위해, HLS 기반으로 타일 단위 행렬 곱셈(GEMM) 커널을 모델링하고 합성 결과를 비교하였다. 신경망 추론에서 합성곱 및 트랜스포머 기반 연산은 대부분 행렬 곱셈과 MAC 연산으로 구성되며, 특히 제한된 자원을 갖는 edge FPGA에서는 DSP 사용 효율이 성능과 집적도에 큰 영향을 미친다. 본 실험에서 사용한 커널은 고정 크기 타일 연산으로, $C_{TM\times TN} += A_{TM\times TK} \times B_{TK\times TN}$ 형태의 행렬 곱 연산을 수행하며, 실험에서는 $TM = 16$, $TN = 16$, $TK = 64$로 설정하였다. 즉, 커널은 1회 호출에 16×64 행렬과 64×16 행렬의 곱을 통해 16×16 출력 타일을 계산한다. 연산량 관점에서 보면, 출력 타일은 총 256개의 원소로 구성되며 각 원소는 64개의 곱셈과 누산을 수행한다. 따라서 커널 호출당 총 곱셈-누산(MAC) 수는 $16 \times 16 \times 64 = 16{,}384$회이다. 본 논문에서 보고하는 latency는 이러한 타일 단위 GEMM 커널을 1회 실행하는 데 소요되는 시간이며, 실제 신경망 추론에서는 이와 같은 커널이 반복적으로 호출되어 전체 행렬 곱 연산을 구성한다.

표 1. HLS 합성 결과 (다양한 자료형과 DSP packing에 따른 자원 사용량 변화)

Packing	DSP	LUT	FF	Latency
int8×int8 (baseline)	256	9351	24634	710ns
int4×int8 (DSP pack)	128	40249	16433	670ns
int4×int4 (DSP pack)	128	25385	16433	670ns

※ systolic array 기반 [16×64][64×16]의 행렬 곱을 수행하는 HLS 커널

기준 설계(baseline)는 INT8×INT8 정밀도의 곱셈을 사용하는 타일 GEMM 커널로 구현하였다. 이 커널은 TK 방향으로 누적 계산을 수행하며, 각 TK 스텝마다 16×16개의 곱셈-누산(MAC) 연산이 동시에 이루어지도록 설계되었다. 이를 위해 HLS에서는 내부 반복문을 병렬화하여 다수의 곱셈 연산이 한 사이클에 처리되도록 합성한다. 연산 자체의 개수는 동일하더라도, 이를 얼마나 많은 하드웨어 유닛으로 동시에 처리하도록 구성했는지에 따라 DSP 사용량이 결정된다.

한편, Xilinx FPGA에서 제공하는 DSP48E2 블록은 본래 중·고정밀 연산을 효율적으로 처리하기 위해 설계된 연산 유닛으로, 일반적으로 최대 18×27 비트 곱셈과 48비트 누산을 지원한다. 이러한 구조는 INT8×INT8 이상의 연산에는 적합하지만, INT4×INT4와 같은 저정밀 연산을 매핑할 경우 DSP 내부의 연산 자원의 비트폭을 충분히 활용되지 못하는 현상이 발생한다. 즉, 단일 INT4×INT4 곱셈이 하나의 DSP 블록 전체를 점유하게 되어, DSP의 비트 폭 대비 실제 사용되는 연산 비트 수가 매우 작아지는 문제가 있다. 이러한 특성으로 인해 HLS 환경에서는 연산 비트폭을 단순히 줄이더라도 컴파일러가 하나의 곱셈연산을 하나의 DSP에 맵핑하게 되고, 이는 제한된 DSP 유닛의 활용도를 감소시킨다.

이러한 문제를 해결하기 위해 본 연구에서는 INT4×INT4 연산을 대상으로 DSP packing 기법을 적용하였다. 먼저 signed INT4 값 $a, b \in [-8, 7]$ 에 대해 bias를 적용하여 unsigned 값으로 변환한 뒤 $(a_u = a + 8, b_u = b + 8)$, 두 개의 가중치를 하나의 곱셈 연산에 packing하였다. 이때 활성값은 두 lane에 복제하고, 가중치는 각각 다른 lane에 배치하여 하나의 DSP 곱셈 연산으로 두 개의 저정밀 곱셈 결과를 동시에 생성한다. 구체적으로,

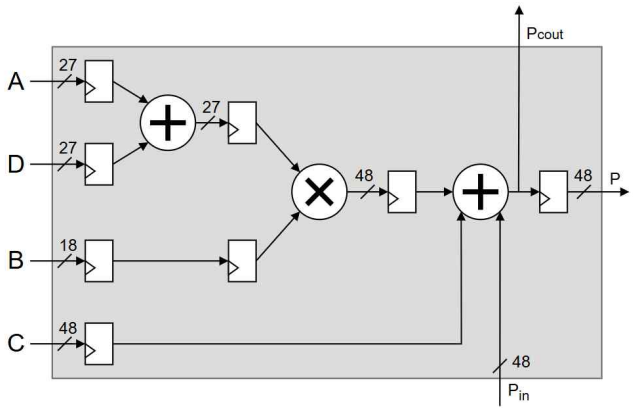
$$A_p = a_u + (a_u \ll 8),\ B_p = b_{0u} + (b_{1u} \ll 8)$$

로 packing한 후 단일 곱셈 $P = A_p \cdot B_p$를 수행하며, 각 lane의 곱셈 결과는 비트 추출을 통해 분리한다. 이후 bias로 인해 도입된 오프셋을 보정하여 원래의 signed 곱셈 결과를 복원한다. 이를 통해 하나의 DSP 곱셈 연산으로 두 개의 MAC 연산을 동시에 처리할 수 있도록 구성하였다. INT4×INT8 연산의 경우 곱셈 결과의 비트폭이 증가하므로 lane간 간섭을 방지하기 위해 더 큰 비트 쉬프트가 적용된다.

이와 같은 DSP packing 구조에서는 하나의 DSP 곱셈 연산이 두 개의 저정밀 곱셈 결과를 동시에 생성하므로, 기존 구조에서 곱셈 연산마다 하나의 DSP가 할당되던 방식과 달리 동일한 연산량을 절반의 DSP 자원으로 처리할 수 있다. 실제 HLS 합성 결과에서도 확인할 수 있듯이, 제안한 INT4×INT4 DSP packing 구조는 처리 요소 수와 연산 지연을 유지하면서 DSP 사용량을 baseline 대비 약 50% 수준으로 감소시켰다.

표에 제시된 합성 결과에서, DSP packing을 적용한 INT4×INT8 및 INT4×INT4 구조 모두에서 DSP 사용량은 baseline인 INT8×INT8 대비 절반인 128로 감소하였다. 특히 INT4×INT4 구조에서는 보정 연산의 복잡도가 상대적으로 낮아 LUT 사용량 또한 감소하는 경향을 보였다. 반면, INT4×INT8 구조에서는 DSP 절감 효과는 동일하게 유지되었으나, 추가적인 보정 연산으로 인해 LUT 사용량이 증가하는 결과를 확인할 수 있었다. 또한, DSP packing을 적용한 모든 저정밀 구조에서 연산 지연은 baseline과 유사한 수준을 유지하였다. 이는 제안한 DSP packing 기법이 DSP 자원 효율을 크게 향상시키면서도 파이프라인 처리 효율을 유지할 수 있음을 뜻한다.

그림1. Xilinx FPGA의 DSP48E2 블록 구조



Ⅲ. 결론

본 논문에서는 edge FPGA 환경에서 신경망 가속을 위한 DSP packing 기반 연산 구조를 제안하고, HLS 기반 모델링 및 합성을 통해 그 효과를 분석하였다. 실험 결과, 단순한 정밀도 축소만으로는 DSP 사용량 감소가 달성되지 않음을 확인하였으며, DSP 내부 연산 구조를 고려한 packing-aware 설계가 필수적임을 보였다. 제안한 구조는 DSP 사용량을 절반으로 감소시키면서도 연산 지연을 유지함으로써, 제한된 자원을 갖는 edge FPGA 환경에서 효율적인 신경망 가속이 가능함을 입증하였다. 향후 연구에서는 제안한 DSP packing 기법을 트랜스포머 기반 attention 연산 및 보다 복잡한 신경망 구조로 확장하여, edge AI 가속 전반에 적용하는 방향으로 연구를 확장할 예정이다.

ACKNOWLEDGMENT

이 논문은 과학기술정보통신부가 지원한 '2025년도 주문연구기업 성장사다리 구축(글로벌 기업 도약)사업'으로 지원을 받아 수행된 연구 결과입니다. [RS-2025-25459504] 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2024-00349885).

참 고 문 헌

[1] Tian, Chunwei, et al. "A survey on deep learning fundamentals." Artificial Intelligence Review 58.12 (2025): 381.

[2] Guo, Kaiyuan, et al. "[DL] A survey of FPGA-based neural network inference accelerators." ACM Transactions on Reconfigurable Technology and Systems (TRETS) 12.1 (2019)

[3] Ronak, et, al. "Mapping for maximum performance on FPGA DSP blocks." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 35.4 (2015)

[4] Sommer, Jan, et al. "Dsp-packing: Squeezing low-precision arithmetic into fpga dsp blocks." 2022 32nd International Conference on Field-Programmable Logic and Applications (FPL). IEEE, 2022.