고정소수점 기반의 트랜스포머 인코더 가속기 하드웨어 구현

박상기, 김찬훈, 노수민, 김정현, 정서호, 정기석*

한양대학교

{skpark1101, kch1103, smrho, hanagod2015, iona97, kchung}@hanyang.ac.kr

Hardware Implementation of Fixed-point based Transformer Encoder Accelerator

Sangki Park, Chan-Hoon Kim, Soo-Min Rho, Jeong-Hyun Kim, Seo-Ho Chung, Ki-Seok Chung*

Hanyang University, Seoul, Korea

요 약

본 논문은 트랜스포머 모델에서 연산 시간의 상당 부분을 차지하는 self-attention 연산의 복잡도를 해결하기 위해, 16-bit 고정소수점으로 동작하는 트랜스포머 인코더 하드웨어 가속기를 제안한다. 제안하는 가속기는 GEMM 연산을 위한 Processing element array와 softmax 함수 처리를 위한 ExCORDIC 모듈을 통해 self-attention을 가속한다. ExCOREDIC 모듈은 CORDIC 알고리즘을 기반으로 자연 지수 함수를 처리하며, 기존 CORDIC 알고리즘의 입력값 제한 문제를 해결하도록 구현되었다. 제안하는 가속기는 Verilog HDL을 이용하여 RTL 설계 구현되어 ZCU111 FPGA 장치에서 동작을 검증하였으며, PyTorch 시뮬레이션 및 HW-SW 교차검증을 진행하였다. 제안하는 가속기는 기존의 FPGA 가속기에 비해 2.64배의 전력 호율성을 보여주며, BERT-Large 모델에서 GLUE task 벤치마크 비교 결과, 기존 32-bit 부동소수점(FP32) 모델과 비교하여 정확도 감소가 최대 1% 이내로 매우 정확한 연산 결과를 보여준다.

Ⅰ. 서 론

트랜스포머 (Transformer) [1] 모델은 문장 생성, 기계 번역 등 자연어처리 분야 뿐만 아니라 이미지 분류 등 Vision 분야에서도 매우 뛰어난 성능을 보여주고 있다. 트랜스포머 모델의 핵심은 self-attention으로, 입력값의 각 token 간 상관관계를 행렬곱셈 (GEMM)과 softmax 함수로 계산한다. Self-attention은 성능이 좋지만 연산복잡도가 커서 실행 속도 저하의 주요 원인 중 하나이며, 특히 입력값의 길이가 커질수록 softmax 연산의 상대적인 연산 비중이 증가한다. [3, 5]

이를 해결하기 위해 본 논문에서는 트랜스포머 인코더 모듈 하드웨어 가속기인 TransENC를 제안한다. 해당 가속기에서는 softmax 연산의 하드웨어 복잡도를 줄이기 위해, 고정소수점으로 양자화되어 동작하는 CORDIC [2] 알고리즘을 확장한 ExCORDIC을 제안하였다. 또한 행렬 연산을 위한 GEMM 가속기 모듈 또한 16-bit 고정소수점으로 동작하도록 설계하였다. 제안하는 가속기는 Xilinx ZCU111 FPGA 장치에 구현하였으며 PyTorch 시뮬레이션 환경에서 FP32 모델과 비교하여 정확도 감소가 1% 이내였다.

Ⅱ. 본론

트랜스포머의 인코더는 크게 linear transformation, multi-head attention, feed-forward network의 세 가지 단계로 나뉜다. [1] Token 임베딩과 linear transformation 단계는 host에서 미리 계산되며, TransENC는 생성된 Query, Key, Value 행렬을 받아 이후 연산을 진행한다.

그림 1.은 제안하는 가속기의 전체 도식도를 보여준다. TransENC는 16-bit 고정소수점으로 동작하며, 단일 head 내의 하나의 입력 시퀀스씩 계산한다. 두 개의 systolic array는 각각 attention score $QK^T/\sqrt{(d_k)}$ 와

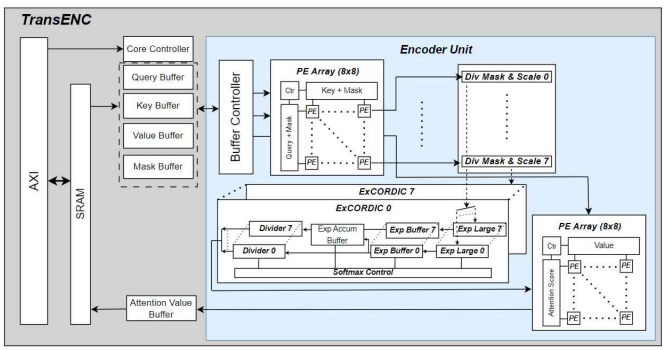

그림 1 TransENC 도식도

$Attention(Q,K,V)$ 을 수행하며, ExCORDIC 모듈은 attention 확률값인 $softmax(QK^T/\sqrt{d_k})$를 연산한다.

Systolic array는 병렬연산의 효율성을 위해 8x8 processing element (PE) array 구조로 되어있다. Q, K 행렬은 각각 지정된 버퍼에 저장이 되며, 제어장치는 이를 열 단위로 첫 번째 PE array에 전달하므로 K 행렬을 따로 전치할 필요가 없다. 각 행렬은 array 크기에 맞춰 8x8 단위로 쪼개져 연산이 진행된다. 각 PE는 16-bit 곱셈 및 누적 (MAC) 연산을 수행하여 계산된 attention score를 ExCORDIC 모듈로 전달한다. 두 번째 PE array는 attention 확률값을 전달받아 최종 attention 값을 계산하게 된다.

기존 연구들이 제안한 인코더 설계에서는 고정소수점으로 양자화되어 있더라도 softmax와 같은 비선형 함수들을 거치기 전에 다시 역양자화를 거쳐 부동소수점으로 변환하여 진행하게 된다 [4, 5, 6]. 이 과정에서 추가적인 연산 시간이 필요하며, 하드웨어가 이를 지원하기 위해서는 부동소수점을 지원하는 연산기를 따로 만들어야 하는 단점이 있다. 추가 변환을

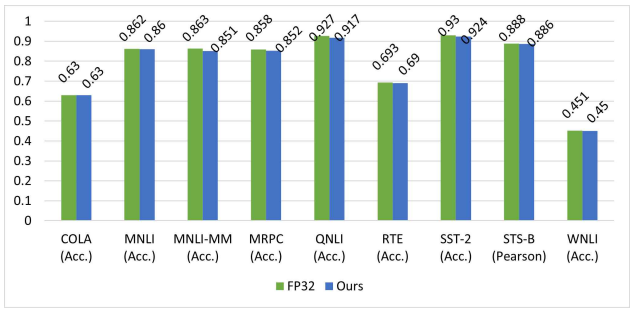

그림 2. BERT-Large 모델에서 GLUE task 결과 비교

피하기위해 LUT를 사용하는 경우, 높은 정밀도의 값을 얻기 위해 큰 메모리를 사용하거나, 메모리를 적게 쓰기 위해 정확도를 희생하는 문제가 존재한다. 본 논문에서는 이러한 문제점을 해결하기 위해 ExCORDIC 모듈을 통해 softmax 연산을 수행한다. 기존의 CORDIC 알고리즘은 입력값이 [-1, +1]로 제한되어 있기 때문에 신경망에서는 사용하기가 어려운 문제가 있다. 하지만 ExCORDIC에서는 입력값을 $ln2$로 나누어 몫 q과 나머지 r를 사용하여 문제를 해결했으며 식은 아래와 같다 [7].

$$q = floor(x/\ln 2) \tag{1}$$

$$r = x - q \times \ln 2 \tag{2}$$

나머지 r은 CORDIC rotation을 거쳐 $\exp(r)$을 계산한다. 몫은 범위 인수로 사용하여 최종 결과값은 $\exp(x)$ 는 다음과 같다.

$$\exp(x) = 2^q \times \exp(r) \tag{3}$$

해당 과정을 통해 오차범위 내 사용 가능한 입력 값의 범위를 [-20, +20]까지 확장하였다. ExCORDIC 모듈은 총 8개로, 부분 행렬의 각 열 단위로 계산하여 accumulator에 저장한다. 이후 행렬의 모든 연산이 완료되면 두 번째 systolic array로 attention 확률값을 전달한다. ExCORDIC 모듈은 총 8개가 설치되어, 8개의 열씩 저장되는 부분 행렬의 각 열을 병렬계산할 수 있도록 하였다.

Ⅲ. 실험 결과

본 논문의 TransENC는 Verilog HDL 언어를 사용하여 RTL 설계되었으며, Xilinx사의 ZCU111 FPGA 장치에 구현되었다. 하드웨어 기능 및 FPGA 검증을 위해 Vivado 2019.2와 Vitis 2019.2 툴을 사용하였다. 하드웨어 구현의 기능 비교 검증을 위한 소프트웨어 구현은 Pytorch 프레임워크와 함께 HuggingFace에서 제공하는 API를 수정하여 구현하였다. BERT-tiny와 BERT-Large 모델을 거대 언어 모델 벤치마크 중 하나인 GLUE의 9가지 태스크로 각각 학습시켰으며, 입력 시퀀스 최대 길이는 128개로 사용하였다. 소프트웨어 벤치마크는 PyTorch 환경에서 ExCORDIC를 구현한 API를 사용해 각 task의 테스트 셋을 추론하는 시뮬레이션 방식과, PYNQ 환경 [9]을 구현하여 PyTorch에서 가속기 IP의 bitstream을 직접 불러와 사용하는 방식 두 가지로 진행하였으며, 두 방식에서 정확도 차이는 없었다.

표 1. 은 트랜스포머 구조에서 싱글 헤드로 구성된 유닛 가운데 서로 다른 FPGA에서 구현된 자원 활용을 나타낸다. 본 논문에서의 결과를 통해 [8]의 전력 소비량 19.3W 대비 7.3W 의 전력만을 사용해 2.64배의 전력 소모 효율을 보여주며 동일한 기능을 위해 더 적은 자원을 사용했음을 보여준다.

그림 2.는 GLUE task 테스트셋에서 FP32 모델과 TransENC 간의

표 1. 단일 Head 모듈에서의 자원비교

	DSP	BRAM	LUT	Freq.	Power
[8]	1044	260	132 K	300 MHz	19.3 W
TransENC	128	16	126 K	215 MHz	7.3 W

PyTorch 상 정확도 시뮬레이션을 보여준다. TransENC는 FP32 모델과 비교하여 가장 큰 정확도 하락이 MNLI-MM task에서 1.2%이고 대부분의 결과에서 1%이하의 차이를 보였다. COLA task와 같이 정확도 하락이 없는 경우도 있었다. BERT-tiny의 경우 MNLI-MM task에서 4% 정도로 하락폭이 있었으나, COLA task에서는 오히려 3.1% 정확도가 오르는 결과 또한 얻을 수 있었다.

Ⅳ. 결론

본 논문에서는 고정소수점 트랜스포머Ⅳ 인코더 가속기인 TransENC를 제안하였다. 8x8 PE array 형태로 부분 행렬을 계산하는 방식으로 병렬성과 하드웨어 활용성을 높였으며, 비선형 함수인 softmax를 ExCORDIC 알고리즘으로 구현하여 기존 CORDIC의 입력 값 제한문제를 해결하고 고정소수점으로도 높은 정확도를 얻었다. 또한 End-to-end 시뮬레이션을 통해 BERT-Large 모델 GLUE task에서 정확도 하락 최대 1%를 달성하여 제안한 가속기가 언어모델에 적합함을 보였다.

ACKNOWLEDGMENT

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2020-0-01304, 모바일 자가 학습 가능 재귀 뉴럴 네트워크 프로세서 기술 개발).

참 고 문 헌

[1] Ashish Vaswani, et al., “Attention is All You Need,” Thirty-first Conference on Neural Information Processing Systems, 2017.
[2] Volder, Jack E., “The CORDIC trigonometric computing technique,” IRE Transactions on electronic computers,, vol. 3, pp. 330−334, 1959.
[3] Jacob R. Stevens, et al., “Softermax: Hardware/Software Co-Design of an Efficient Softmax for Transformers,” 58th ACM/IEEE Design Automation Conference, 2021.
[4] Qu, Zheng, et al., “Dota: detect and omit weak attentions for scalable transformer acceleration,” Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 2022.
[5] Ham, Tae Jun, et al., “ELSA: Hardware-software co-design for efficient, lightweight self-attention mechanism in neural networks,” ACM/IEEE 48th Annual International Symposium on Computer Architecture, 2021.
[6] Tiwari, et al., “Hardware implementation of neural network with Sigmoidal activation functions using CORDIC.” Microprocessors and Microsystems 39.6 (2015): 373-381.
[7] Rekha, et al., “FPGA implementation of exponential function using cordic IP core for extended input range,” IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology, 2018.
[8] Ye, Wenhua, et al., “Accelerating attention mechanism on FPGAs based on efficient reconfigurable systolic array,” ACM Transactions on Embedded Computing Systems, 2022.
[9] PYNQ, http://www.pynq.io/