Transformer Encoder 의 Hardware Accelerator 에 관한 연구

유예송 [1], 김채윤 [2], 박혜령 [3], 안채원 [4]
[1] 이화여자대학교 뇌인지과학부 학부생
[2] 이화여자대학교 전자전기공학부 학부생
[3] 이화여자대학교 전자전기공학부 학부생
[4] 이화여자대학교 전자전기공학부 학부생

welcome-to-yesong@ewhain.net, sally2172014@ewhain.net, hyeryeong0103@ewhain.net, chaewon2k2@ewhain.net,

A Study on Hardware Accelerator for Transformer Encoder

Ye-Song Yu[1], Chae-Yoon Kim[2], Hye-Ryeong Park[3], Chae-Won Ahn[4]
[1]Dept. of Brain Cognitive Science, Ewha Womans University
[2]Dept. of Electronic and Electrical Engineering, Ewha Womans University
[3]Dept. of Electronic and Electrical Engineering, Ewha Womans University
[4]Dept. of Electronic and Electrical Engineering, Ewha Womans University

요 약

데이터의 규모가 방대해지고 AI 모델의 구조적 복잡성이 증가함에 따라 AI 하드웨어 가속기의 성능이 더욱 중요해졌다. 특히 LLM 의 핵심을 이루는 Transformer 모델이 주목받고 있으나, Transformer 의 하드웨어 가속기 연구는 타 모델에 비해 상대적으로 늦게 진행되었다. 그 이유에는 최적화가 어려운 복잡한 연산과 메모리 접근패턴이 있다. Transformer 는 Self-Attention 메커니즘을 사용해 입력 시퀀스 내 모든 요소 간의 관계를 계산하는 구조로[1], 매우 많은 양의 연산과 메모리 사용을 요구한다. NLP 기술이 생활 곳곳에서 대체될 수 없는 도구로 자리 잡은 만큼 Transformer Accelerator 가 더 많이 연구, 개발될 필요가 있다.[2] 본 연구는 Verilog HDL 로 하드웨어에 최적화된 Transformer Encoder 를 구현한 후 합성/실행하여 FPGA 칩에 업로드한다. transformer 의 encoder 에 알맞은 accelerator 를 제작하여 다양한 NLP 모델의 등장과 개발을 촉진하고자 한다. 또 각 모델에 따라 특화 연산기를 제작하는 연구 파이프라인을 구축한다.

1. 서론

본 연구는 최근 대두되고 있는 자연어 처리 기술에 특화된 하드웨어 가속기를 설계한다. 기존의 GPU 는 다수의 병렬 프로세싱 유닛을 가지고 있어 딥러닝 연산에 강점을 보이나, 고정된 하드웨어 아키텍처를 가지고 있어 하드웨어 수준에서 연산 최적화를 할 수 없다. Transformer 에 최적화된 하드웨어 아키텍처를 새롭게 제안하고자 프로그래밍 가능한 하드웨어 칩인 FPGA 로 Transformer 맞춤형 AI accelerator 를 구현하였다. 특히 Transformer 의 핵심 연산인 행렬 곱셈, Scaled Dot-Product Attention 연산, 비선형 활성화 함수 그리고 FFN 연산을 하드웨어에 설계함으로써 연산 효율성과 처리 속도를 크게 향상시켰다. 연구 과정에서 simulation 을 위해 Modelsim 을, PL 개발을 위해 Vivado 를, PS 개발을 위해 Vitis 를, 또 통합 개발환경 구축을 위해 Ubuntu 를 사용하였다. Zybo Z7-20 보드를 이용하여 하드웨어 가속을 담당하는 PL(Programmable Logic) 영역뿐만 아니라 Arm Cortex 프로세서가 내장된 PS 영역까지 통합시켰다.

2. Transformer Accelerator 의 요구사항

Transformer Encoder 는 입력 sentence 를 받아 문장의 맥락정보를 담은 고차원 벡터로 변환하는 역할을 수행한다. 다른 mechanism 들과 달리 토큰들을 통합적으로 처리하여 관계를 파악한다. 특히 Multi-Attention Head Module 에서는 각각의 독립적인 Attention Head 연산을 병렬적으로 수행하여 하나의 입력 시퀀스로부터 각각 서로 다른 특징 정보들을 추출한다. 대규모 데이터에서 입력 시퀀스 내 모든 요소 간의 관계를 계산하기 위해서는 신속하고 정확하게 필요 연산을 수행하는 것이 중요하다. 행렬 곱셈과 Scaled Dot-Product Attention 연산, 비선형 활성화 함수 연산을 빠르고 정확히 처리할 수 있도록 해야 한다.

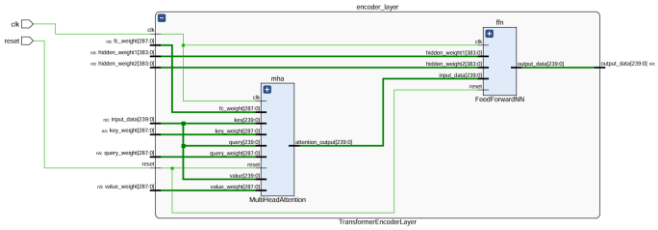
대규모 데이터를 지연 없이 주고받는 것도 중요하다. AXI4 lite 를 위한 모듈을 만든 후 Transformer Encoder 의 최상위 모듈에서 을 인스턴스화하였다. 이

때, AXI4 lite 모듈에 user logic 을 추가하여 각 slave register 가 input/output port 에 assign 되도록 하였다. AXI4 lite 를 활용하여 향후 AXI4 나 DMA 와 같은 고성능 인터페이스로 쉽게 확장할 수 있는 유연성을 확보하였다.

3. 주요 모델 설명

Top Module 과 Transformer Encoder Layer Module, MultiHead Attention Module, Attention head Module, FeedForward NN Module, Linear Transform Module, ReLU Module 등을 구현하였다. 각 모듈 간 데이터 흐름과 계산량을 최소화하기 위해 파이프라인 구조를 적용하여 병렬 연산을 극대화하였고, 필요한 연산만을 실행하여 불필요한 오버헤드를 줄였다.[3] 특히 MultiHead Attention 과 FeedForward NN 의 경우, 매트릭스 연산과 벡터화를 통해 메모리 접근을 최적화했다.

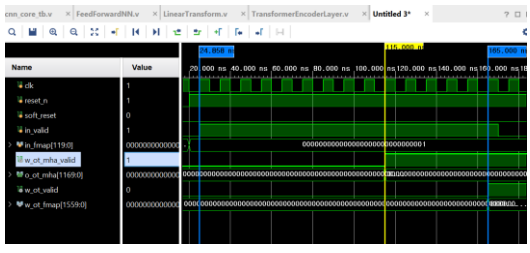
Top Module 이 11 개 이하의 토큰으로 이루어진 하나의 문장을 받아들이고 Transformer Encoder Layer Module 로 전달한다. 이후 MultiHead Attention Module 을 지나며 attention mechanism 이 실행되고 주어진 query 가 각 key 와 얼마나 관련이 있는지를 계산한다. 그 다음 FeedForward NN Module 을 지나며 모델이 복잡한 패턴이나 비선형적인 관계를 학습한다.



(그림 1) Transformer Encoder Layer Module 의 구조 일부

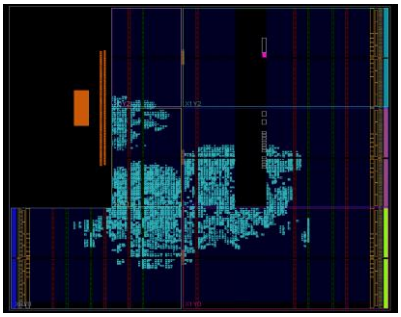
4. 시뮬레이션 및 검증

testbench 파일을 통해 시뮬레이션을 돌려 총 input_valid 신호가 들어가고 9 clock cycle 후에 multihead attention 의 output 이, 5 clock cycle 후에 feedforward NN 의 output 이 출력되어 결과적으로 transformer encoder 의 output 은 총 14 cycle 후에 출력되는 것을 알 수 있다.



(그림 2) 파형 시뮬레이션 결과

Synthesis 과정에서 생성된 netlist 를 실제 FPGA 하드웨어에 배치하고 라우팅하였다.



(그림 3) FPGA 의 내부 리소스 디자인

FPGA 의 각 논리 블록(CLB, DSP 등)의 위치가 결정된 모습이다. 라우팅 단계에서 배치된 논리 블록 간의 연결 또한 잘 설정되었다.

"본 논문은 과학기술정보통신부 대학디지털교육역량 강화사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다"

참고문헌

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., "Attention is All You Need", *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30, pp. 5998-6008, 2017.

[2] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, 2019, pp. 4171-4186.

[3] Wu, R., Guo, X., Du, J., and Li, J., *Accelerating Neural Network Inference on FPGA-Based Platforms—A Survey*, *Electronics*, vol. 10, no. 9, pp. 1025, 2021.