16-병렬 MAC 연산을 위한 FPGA 기반 오프로드 가속기 설계

조현건[1], 송현우[2], 이동재[2], 김민선[3]
[1]건국대학교 전자전기공학부 학부생
[2]동국대학교 전자전기공학부 학부생
[3]서강대학교 인공지능학과 학부생

zlak0@konkuk.ac.kr, isaacno1@dgu.ac.kr, 2020111827@dgu.ac.kr, minseon23@sogang.ac.kr

Implementation of a 16-Parallel MAC Operation Offloading Accelerator on FPGA

Hyun-Keon Cho[1], Hyun-Woo Song[2], Dong-Jae Lee[2], Min-Seon Kim[3]
[1]Dept. of Electronic and Electrical Engineering, Konkuk University
[2]Dept. Of Electronic and Electrical Engineering, Dongguk University
[3]Dept. of Artificial Intelligence, Sogang University

요 약

본 논문은 고양이 얼굴 인식 기반 스마트 급식기의 FPGA CNN 가속기를 설계했다. 저사양 에지 컴퓨팅 환경에서 CNN 모델의 실시간 추론 성능을 확보하기 위해 ZYNQ FPGA를 활용한 Offloading Architecture를 개발하였다. CNN 연산 가속을 위해 FPGA의 Programmable Logic(PL)에 16-병렬 MAC 연산기를 채택하여 연산 부하를 효과적으로 분산시켰다. 또한, Weight-Stationary Dataflow 및 Weight Double Buffering 기법을 적용함으로써 온칩 메모리 활용 효율을 극대화하고 데이터 전송으로 인한 병목 현상을 최소화하였다. 개발된 FPGA 가속기는 ResNet50 및 CatFaceNet의 추론 속도를 성공적으로 향상시켰으며, 1.82fps의 성능을 달성했다.

1. 서론

고양이 얼굴 인식 스마트 급식기는 개별 고양이 인식하고 구별해 개인화된 급식 서비스를 제공한다. 고양이 얼굴 인식을 위해 CNN 모델을 사용한다. 라즈베리 환경에서 CNN 연산을 가속하기 위해 저사양 FPGA Cora z7-07(XC7Z007S-1CLG400)를 사용해 가속기를 구성했다. 보드의 자원을 최대한 활용하기 위해 offloading architrecture의 16 병렬화된 MAC 연산 가속기를 설계했다.

2. 주요 알고리즘

- **Offloading Architecture**

ZYNQ architecture의 PL, PS 자원을 최대한 활용하기 위해 라즈베리와의 통신 및 convolution 연산을 제외한 모든 CNN 연산을 PS(CPU)에서 총괄한다. PL은 feature, kernel를 받고 MAC 연산 결과를 메모리에 저장한다. 이때 PL은 DMA(Direct Memory Access Controller)로 DRAM에 접근하며 고속 streaming을 받고, 이 데이터 입력에 즉각적으로 연산을 진행해 메모리 접근에 의한 지연을 최소화한다.

- **Weight-Stationary Dataflow**

convolution 연산은 같은 kernel을 슬라이딩하며 같은 값의 재사용률이 매우 높다. kernel 전송에 의한 지연을 해소하기 위해 PL에 전송된 kernel값을 BRAM에 저장 및 재사용해 kernel에 의한 메모리 접근을 최소화한다.**[1]**

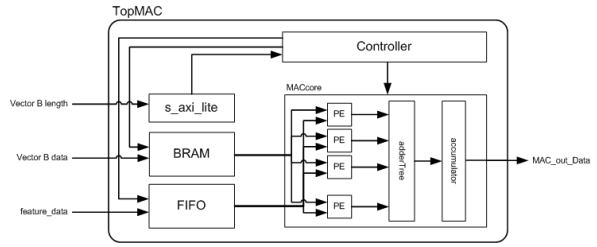
- **Weight Double Buffering**

kernel을 재사용해도 여전히 많은 메모리 접근이 요구된다. 또한 PL에서 feature, kernel 입력과 동시에 연산이 진행되어도 PS의 컨트롤과 통신에 의한 지연 또한 발생한다. 이를 최소화하기 위해 double buffering된 메모리로 다음 kernel 데이터를 미리 입력받는다.

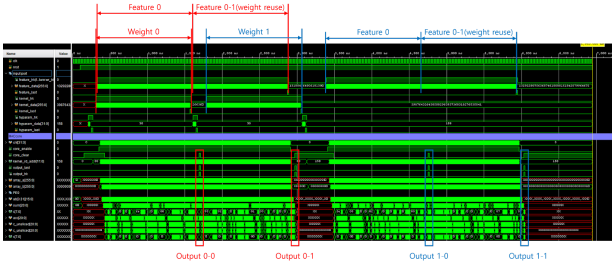
3. 디지털 회로 설계

kernel, feature 입력 전, iteration, BRAM read_head_address, BRAM write_head_address를 입력 받는다. 이때 kernel은 BRAM에 write_head_address부터 저장되며, feature은 FIFO에 저장됨과 동시에 BRAM의 read_head_address의 값부터 MACcore에서 MAC 연산을 진행한다.

MACcore은 8FP feature, kernel 값을 병렬화된 PE에서 곱하고 addtree로 합산되어 누적된다

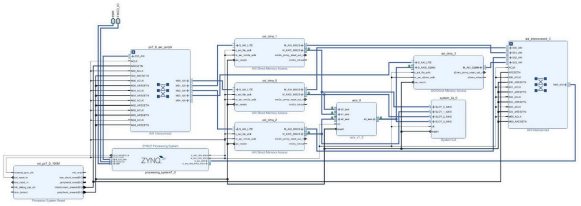


(그림 1) Block diagram.



(그림 2) Synthesis timing simulation.

DRAM과 고속 streaming을 위해 4개의 DMA가 사용되었다. 사용된 보드의 최대 병렬 수 16개의 PE로 설계되어 feature, kernel 값은 128bit, output 값은 최소 bitwidth인 32bit로 설정했다.



(그림 3) Block design.

Resource	LUT	LUTRAM	FF	BRAM
Utilization	12954	5610	10675	41.5
Utilization(%)	89.96	93.5	37.07	83

<표 1> Resource utilization(XC7Z007S-1CLG400)

4. 소프트웨어 설계

효율적인 연산을 위해 convolution 을 GEMM(General Matrix to Matrix Multiplication)으로 변환후 PL에 전달한다. convolution의 stride, padding에 따라 feature array를 재배치한다.. 그리고 weightWeight-stationary dataflow에 따라 kernel 전체를 전송 후, feature을 전송해 MAC 연산을 진행한다. 이때 PL의 가속기 내부 BRAM에 저장되어 재사용된다.

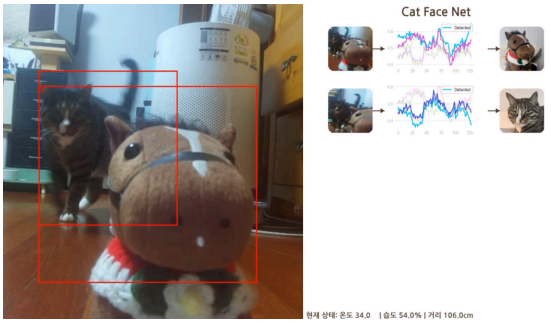
스마트 급식기와 ethernet 통신으로 카메라 이미지를 전송받아 연산 후 그 결과를 수신한다.

Weight-Stationary Dataflow

W, H: feature size / C : feature channel
S, R : kernel size / M : kernel channel
F, E : output size
DMA_read(address, length)
DMA_write(address, length)

```
DMA_read(weight, R*S*C*M)
for m in 0:M:
    DMA_read(feature, R*S*C*F*E)
    for f in 0:F:
        for e in 0:E:
            DMA_write(output + (m*E+e)*F+f, 1)
```

본 연구의 가속기는 FPGA기반 연산 가속 CNN모델을 적용한 맞춤형 반려묘 개별 인식 및 자동 급식시스템 EasyFeeder에 활용되었다. 객체 인식에는 YOLO를 활용하였다. 인식된 고양이는 TripletLoss로 학습된 MobileFaceNet 기반 CatFaceNet[1]을 통해 embedding vector을 얻고, Cosine similarity으로로 구별한다. 그림은 실제 환경에서 두 객체를 개별적으로 인식하는 모습[2]이다.



(그림 4) 스마트 급식기 시스템.

5. 결과

본 연구의 가속기로 ResNet50 추론 결과 0.15fps의 성능을 보였다. 스마트 급식기의 CatFaceNet에서 1.82fps의 성능을 보였다.

Acknowledgements

본 논문은 과학기술정보통신부 대학디지털교육역량강화사업의 지원을 통해 수행한 한이음 드림업 프로젝트 결과물입니다.

참고문헌

[1] R. Venkatesan et al., "MAGNet: A Modular Accelerator Generator for Neural Networks," 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Westminster, CO, pp. 1-8, USA, 2019

1) CatFaceNet은 kaggle의 LCW(Labeled Cats In The Wild) 데이터셋을 통해 학습된 MobileFaceNet 기반 커스텀 모델
2) 실제 고양이 두마리로 진행해야 하나, 연구자의 집에 고양이가 1마리라 인형으로 대체함