

Convolutional Neural Network 기반의 워터마킹 프로세서의 설계

*이재은 서영호 김동욱

광운대학교

*jelee@kw.ac.kr

Design of watermarking processor based on convolutional neural network

*Lee, Jae-Eun Seo, Young-Ho Kim, Dong-Wook

Kwangwoon University

요약

본 논문에서는 촬영과 동시에 유통되는 생방송 영상의 실시간 지적재산권 보호를 위한 Convolutional Neural Network를 기반으로 하는 워터마킹 프로세서의 구조를 제안한다. 제안하는 워터마킹 프로세서는 전처리 네트워크와 삽입 네트워크를 최적화하여 ASIC 칩으로 제작한다. 이는 영상을 입력으로 하는 딥 러닝 분야에서 많이 사용되는 CNN을 기반으로 하기 때문에 일반적인 딥 러닝 가속기 설계로 간주된다.

1. 서론

최근 영상 콘텐츠 수요의 증가로 영상 콘텐츠들의 지적재산권 문제가 발생하고 있다. 올바른 영상 콘텐츠의 활성화를 위해서는 지적재산권 보호가 매우 중요한데, 그 방법으로 디지털 워터마킹 방법이 가장 활발하게 연구가 되고 있다[1]. 또한, 많은 컴퓨터 비전 분야에서 딥 러닝을 이용하여 최고의 성능을 도출해내고 있다. 하지만, 딥 러닝은 연산량과 메모리 사용량이 매우 커 실시간 구현을 위해서는 하드웨어 가속기가 필수적이다. 특히, 영상을 촬영과 동시에 유통하는 생방송 영상의 지적재산권을 보호할 수 있는 워터마킹 삽입 방법이 요구된다.

따라서, 본 논문에서는 Convolutional Neural Network (CNN) 기반의 가속기를 ASIC 칩으로 제작하였다. 이는 영상을 대상으로 하는 딥 러닝 분야에서 많이 사용되는 CNN을 기반으로 하기 때문에 일반적인 딥 러닝 기반의 하드웨어 가속기 설계로 간주된다.

2. 제안하는 디지털 워터마킹 네트워크의 하드웨어 구조

Table 1에 ASIC으로 설계한 디지털 워터마킹 네트워크의 구조를 자세히 나타낸다. 일반적인 CNN의 구조를 가지며 컨볼루션 연산, 배치 정규화 연산, 그리고 활성화 함수로 구성되어 있다. 각 층(layer)에 따라 배치 정규화 사용 유무와 활성화 함수 종류가 변화한다.

제안하는 디지털 워터마킹 프로세서의 전체적인 구조는 Figure 1에 나타내고 있다. 이 구조는 동작적인 역할에 따라 크게 데이터 패스부와 제어부로 구분된다. 데이터 패스부는 다시 컨볼루션 블록, 중간 블록, 그리고 후반 블록으로 나누어지며 제어부는 메모리 제어부와 메인 제어부로 구성된다. 컨볼루션 블록에서 필터 레지스터 (Filter Register)나 입력 특징 맵 레지스터 (Input Feature map, IFmap Register)를 입력으로 받아 곱셈과 누적 덧셈 연산을 수행한다. 그 결과를 부분 합 레지스

터(Partial sum, Psum Register)에 넘겨준다. 부분 합 레지스터는 이를 SRAM 메모리에 저장하여 다음 동작에 사용을 하거나 후반 블록으로 넘기게 된다. 후반 블록에서는 입력에 알파 레지스터 (Alpha Register)를 더한 후 활성화 함수를 수행하여 출력 레지스터 (Output Register)에 저장한다. 이 모든 동작은 메인 제어부의 제어에 따라 수행된다. 이러한 연산을 수행하기 전에 메모리 제어부를 중심으로 외부 메모리에서 들어오는 입력 데이터를 메모리 버퍼 (Memory buffer)로 받아 입력 인터페이스 (Input interface)를 통해 필터 레지스터 또는 입력 특징맵 레지스터 또는 알파 레지스터에 배치해준다.

Table 1. Network structure to be implemented

Network	Kernel size	number of kernels	stride	Batch Normalization	Activation function
Pre-processing Network	3×3	64	1	X	X
WM Embedding Network	3×3	64	1	○	ReLU
	3×3	64	1	○	ReLU
	3×3	64	1	○	ReLU
	3×3	1	1	X	tanh

3. ASIC flow

Figure 2에 나타내는 ASIC 설계 절차를 간략히 나타냈다. 제안한 H/W는 Verilog를 이용하여 RTL(Register Transfer Level) 수준으로 설계 하였다. Verilog 설계는 Xilinx의 Vivado로 진행하였고 Synopsys 사의 Design Compiler로 논리 합성(logic synthesis)을, IC Compiler로 floorplanning, place&routing을 진행하였다. VCS로 논

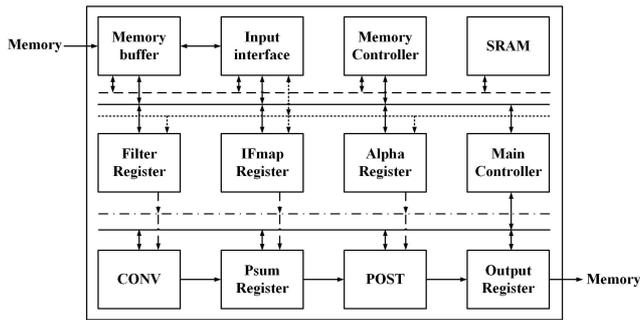


Figure 1. Global hardware architecture

리 합성 전의 동작 검증(functional verification)과 합성 후와 레이아웃 설계 후의 simulation(pre layout simulation, post layout simulation)을 수행하였다. 또한, Formality로 합성 전과 후, 레이아웃 설계 전과 후의 동작을 비교하였으며(equivalence check), Primetime으로 합성 후와 레이아웃 설계 후에 STA(static timing analysis)를 수행하여 동작을 검증하였다.

4. 실험 결과

Figure 3에 Floorplanning과 P&R까지 완료한 레이아웃 결과를 나타낸다. SRAM은 공정사의 라이브러리를 사용하여 Memory Compiler로 생성 후 배치하였다. 공정사에서 제공하는 라이브러보다 요구되는 메모리의 크기가 컸기 때문에 2개로 나눠서 진행하였다. 제작한 칩의 크기는 4,000×4,000mm이고 코어는 1,074,147.3750mm²의 면적을 가진다. 이중 메모리의 면적은 777603.84mm²을 스탠다드 셀의 면적은 296543.535mm²를 가진다.

5. 결론

본 ASIC 설계는 고속화로 호스트 영상이 유통되는 경우에 바로 적용시킬 수 있을 것이라 생각된다. 또한, CNN을 ASIC 칩으로 구현한 것으로 딥 러닝의 하드웨어 가속기로서 추후 연구에 도움이 될 것이라 사료된다.

감사의 글

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2019R1F1A1054552)

참고문헌

[1] I. J. Cox, et al., "Digital watermarking and steganography," Morgan Kaufmann Publisher, 2008.

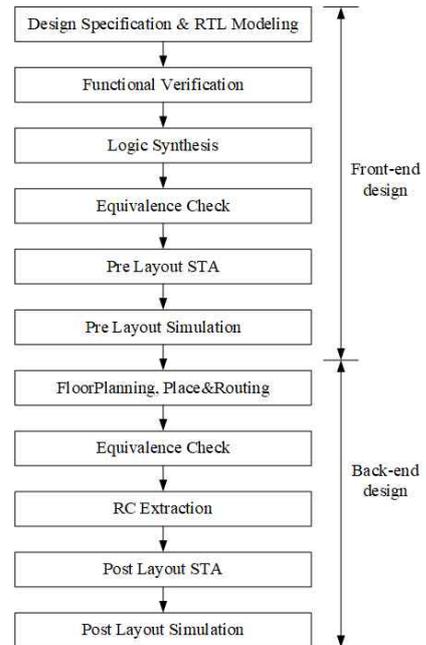


Figure 2. ASIC flow

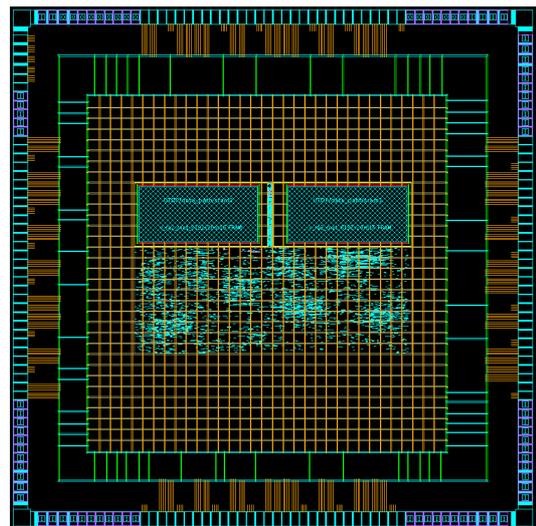


Figure 3. P&R result