Concatenation을 이용한 DSP의 4-bit 병렬 Multiply-Accumulation

문성빈, 강도경, *이성주

세종대학교 전자정보통신공학과, *정보통신공학과 및 지능형드론 융합전공

anstjdqls55@itsoc.sejong.ac.kr, dokyeong@itsoc.sejong.ac.kr, *seongjoo@sejong.ac.kr

A 4-bit Parallel Multiply-Accumulation On a Single DSP Using Concatenation

Moon Sung Bin, Kang Do Kyeong, *Lee Seong Joo

Dept. of Electrical Eng., Sejong Univ.

*Dept. of Information and Comm. Eng. and Convergence Engineering for Intelligent Drone, *Sejong Univ.

요 약

최근 Deep Learning에 관한 연구가 활발하면서 FPGA가 많은 주목을 받고 있다. FPGA가 Deep Learning 연산 속도를 향상시키기에 적절하기 때문이다. 그러나 FPGA의 Hardware 자원량은 제한적이다. 그러므로 FPGA의 자원들을 효과적으로 이용할 수 있는 방안들이 필요하다. 본 논문에서는 FPGA 내 DSP의 활용도를 높이기 위해 DSP에서의 4-bit parallel Multiply-Accumulate 연산기법을 제안한다. 제안하는 기법은 기존의 Multiple Multiplication과 달리 DSP에 입력을 인가하기 전 두 data를 결합(Concatenation)시키고, DSP의 pre-adder를 accumulator로 사용한다. 두 data를 결합시킬 때 적절한 guard bit들을 사용한다면 연산 후 필요한 추가 작업(overhead circuit)들을 줄일 수 있다. 본 논문에서는 4-bit signed data에 대해 implementation을 수행하여 검증하였고, 그 결과 전체 LUT 및 Flip-Flops 중 약 0.10%, 0.15%.만을 사용하였다.

Ⅰ. 서 론

다양한 응용분야에 적용되는 Deep Neural Networks (DNN)들은 많은 연산량을 요구한다. 특히, DNN 응용 중 하나인 Convolutional Neural Networks (CNN)은 Multiply-Accumulate (MAC) 연산을 알고리즘의 핵심 연산으로 사용하며 복잡한 동작들을 수행한다 [1], [2].

다음의 배경 속에서 FPGA는 Deep Learning 알고리즘의 실행 속도를 향상시키는 방법 중 하나로 나타났다 [3], [4], [5]. 이기종(Heterogenous) 특성을 가진 FPGA는 Deep Learning 알고리즘에 최적화된 연산을 수행할 수 있기 때문이다.

그러나 FPGA 내 Hardware 자원들은 한정적이므로 이러한 자원들을 효과적으로 이용할 수 있는 방법이 필요하다. 기존에는 DSP의 pre-adder를 이용해 Multiple Multiplication을 수행하는 방식으로 자원의 활용도를 높였다 [6], [7]. 두 data를 곱하기 전에 overlap 되지 않도록 하나의 data를 left shift 후 서로 더한다면 곱셈 연산을 병렬적으로 수행할 수 있다. 이 때, 두 data를 더하는 과정에서 하위 data의 sign extension에 의해 상위 data의 bit가 변하게 된다. 따라서, 연산 후 이를 보정하기 위한 추가 작업(overhead circuit)이 필요하다 [8].

본 논문에서는 결합(Concatenation)을 이용해 FPGA의 DSP에서 4-bit MAC 연산을 병렬적으로 수행하는 방법을 제안한다. 결합을 통해 두 data가 서로에게 영향을 끼치지 않도록 두 data 사이에 적절한 guard bit를 추가한다면 연산 후 보정 작업을 줄일 수 있다. 또한, pre-adder를 accumulator로 사용하여 MAC 연산의 단계를 줄일 수 있다.

본론에서 1. DSP의 parallel MAC 연산 방식을 소개하고, 이를 수행하기 위한 2. Bit 결합 구성을 제시한다. 그리고 3. 실험 결과를 분석하겠다.

Ⅱ. 본 론

1. DSP에서의 MAC 연산 병렬화

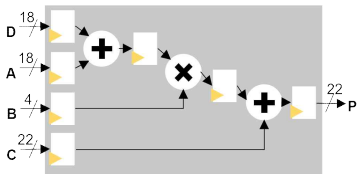

그림1. DSP의 기본 구조

그림 1의 입력 A, D는 각각 두 data의 결합이고, 입력 B는 weight와 같다. 입력 C는 이전 P의 값이다. DSP 입력에 따른 연산 결과 P는 다음과 같다.

$$P_N = (A_N + D_N)B_N + C_N, (C_N = P_{N-1}) \quad (1)$$
$$= (A_N B_N + D_N B_N) + P_{N-1} \quad (2)$$
$$= (A_N B_N + D_N B_N) + (A_{N-1} B_{N-1} + D_{N-1} B_{N-1} + C_{N-1}) \quad (3)$$

식 (3)에서 입력 C를 통해 DSP를 계속 지나면서 결과 값들이 누적되는 것을 알 수 있다. 특히, 다음 식을 보면 pre-adder를 통해 B가 곱해지기 전 A와 D를 더함으로써 parallel Multiply-Accumulation 연산을 수행할 수 있다. (우항의 '(,)'은 두 data의 결합을 의미한다.)

$$(A_N + D_N)B_N = [(d1_N, d2_N) + (d3_N, d4_N)]w_N \quad (1)$$
$$= (d1_N + d3_N, d2_N + d4_N)w_N \quad (2)$$
$$= (d1_N w_N + d3_N w_N, d2_N w_N + d4_N w_N) \quad (3)$$

* 교신저자: 이성주

2. 결합을 활용한 Bit 구성

많은 응용분야에서 FPGA의 성능을 향상시키기 위해 양자화를 통한 Approximate precision을 사용하는 경우가 많다 [9], [10]. 본 논문에서는 fixed point 형식의 4-bit signed data를 이용한다.

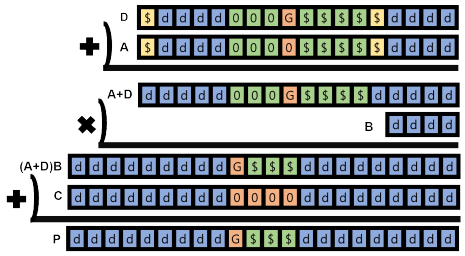

그림2. 결합 bit 구성에 따른 연산 과정

그림 2의 A, D는 DSP의 입력으로, 인가하기 전 각각 두 개의 4-bit signed data를 결합한 것이다. d는 data bit를, $는 sign extension bit를, G는 guard bit를 의미한다. 노란색 sign extension bit는 A, D 덧셈 시 overflow를 대비하기 위함이다.

초록색 sign extension bit와 0는 (A+D)와 B를 곱하기 위함이다. 곱셈 과정에서 shift가 발생하는데 상위 data의 하위 bit가 0이 아니거나, 하위 data의 상위 bit가 sign extension이 아니라면 원래 값과 달라진다.

주황색의 guard bit는 (A+D)B와 C 덧셈 시 carry로 인한 하위 data의 상위 data 오염을 막기 위함이다. (A+D)의 하위 data가 양수이고 B가 음수일 때, (A+D)B의 상, 하위 두 data 사이가 모두 1이 될 수도 있다. 이 경우 C와 덧셈 시 Carry가 발생하면 결과 값이 바뀌게 된다. 이를 방지하기 위해 A, D 두 하위 data가 음수인 경우를 제외하고 guard bit는 1로 지정한다. 그리고 결과 값 P의 상, 하위 두 data 사이는 다시 0으로 지정하여 carry로 인한 오염을 예방한다.

마지막으로 B가 음수인 경우 (A+D)가 B의 MSB와 곱해질 때 상, 하위 data에 대해 각각 2's complement가 필요하다. 하위 data에 대해서는 자동으로 반영되지만 상위 data는 bit 반전만 나타난다. 따라서, B가 음수일 때 상위 data에 '+1'을 해주기 위해 결과 값 P에 2^{14}를 더해줘야 한다.

3. 실험 결과 및 분석

실험에서는 Vivado 2021.2버전과 FPGA 'xc7z020clg484-1'를 사용했다. 1개의 DSP48E1을 가지고 Implementation을 수행한 결과 55개의 LUT, 161개의 Flip-Flops을 사용했다. 이는 각각 총 LUT 및 F/Fs의 0.1%, 0.15%을 사용한 것이다. 다음은 같은 환경에서 기존의 Multiple Multiply 방식과 비교한 표이다.

	LUT		F/F	
Multiple Multiplication with Accumulator	60개	0.11%	179개	0.17%
parallel MAC using Concatenation	55개	0.1%	161개	0.15%

표1. LUT 및 F/F Utiltization 정보

자원에 여분이 있다면 여러 개의 DSP를 이용해 parallel MAC 연산을 구현할 수 있다. 이 경우에는 MAC 연산의 더 단계를 줄일 수 있겠지만 추가적인 adder tree가 필요할 것이다.

Ⅲ. 결 론

본 논문은 DSP를 이용한 4-bit Multiply-Accumulate in parallel 방식을 소개했다. 이전에는 pre-adder를 가지고 Multiple Multiplication 하는 방식으로 DSP 활용도를 높였다. 이 때, pre-adder를 이용해 두 data를 더하는 과정에서 하위 data의 sign extension에 의한 상위 data의 bit가 변하게 된다. 따라서, 연산 후 이를 보정하기 위한 추가 작업이 필요하다.

본 논문에서 제안한 방식은 DSP에 입력을 인가하기 전 직접 두 data를 결합시키고, pre-adder를 누적 연산기로 사용했다. 결합하는 과정에서 guard bit를 직접 추가하여 data를 제어했다. 실험 결과 전체 중 0.1%의 LUT, 0.15%의 F/Fs만을 사용하여 자원 사용량을 줄였음을 알 수 있었다.

다만, 연산 후 추가 작업을 줄이기 위해 필요한 guard bit가 많다 보니 4-bit signed data에 대해서만 적용 가능하다. 그리고 DSP의 입력 단자 B에 인가되는 data는 여전히 한 개이다. 그러므로 guard bit를 줄이고, 입력 단자 B에 data들을 결합시켜 인가하는 연구가 필요하다.

ACKNOWLEDGMENT

본 연구는 산업통상자원부의 시장선도를 위한 한국 주도형 K-Sensor 기술개발(R&D)사업(과제번호 1415181734) 및 과학기술정보통신부 재원으로 한국연구재단(No. 2020R1A2C1007546)의 지원을 받아 수행된 연구이며, 검증을 위한 EDA관련 툴은 IDEC의 지원을 받았음.

참 고 문 헌

[1] Kim, Ji-Won, et al. "다양한 딥러닝 알고리즘과 활용." Communications of the Korean Institute of Information Scientists and Engineers 33.8 (2015): 25-31.

[2] Lee, Yong-Hwan, and Youngseop Kim. "Comparison of CNN and YOLO for Object Detection." Journal of the semiconductor & display technology 19.1 (2020): 85-92.

[3] Shawahna, Ahmad, Sadiq M. Sait, and Aiman El-Maleh. "FPGA-based accelerators of deep learning networks for learning and classification: A review." ieee Access 7 (2018): 7823-7859.

[4] Wang, Teng, et al. "A survey of FPGA based deep learning accelerators: Challenges and opportunities." arXiv preprint arXiv:1901.04988 (2018).

[5] Blaiech, Ahmed Ghazi, et al. "A survey and taxonomy of FPGA-based deep learning accelerators." Journal of Systems Architecture 98 (2019): 331-345.

[6] Huang, Zhangqin, Shuo Zhang, and Weidong Wang. "An efficient method of parallel multiplication on a single DSP slice for embedded FPGAs." IEEE Access 7 (2019): 100993-101008.

[7] Kalali, Ercan, and Rene Van Leuken. "Near-Precise Parameter Approximation for Multiple Multiplications on A Single DSP Block." IEEE Transactions on Computers (2021).

[8] Véstias, Mário, et al. "Parallel dot-products for deep learning on FPGA." 2017 27th International Conference on Field Programmable Logic and Applications (FPL). IEEE, 2017.

[9] Sambhav R Jain, Albert Gural, Michael Wu, and Chris Dick. 2019. "Trained Quantization Thresholds For Accurate And Efficient Fixed-Point Inference Of Deep Neural Networks." arXiv Preprint arXiv:1903.08066

[10] T. Han et al., "Convolutional neural network with INT4 optimization on Xilinx devices," Xilinx White Paper, WP521, Jun. 2020.