무인이동체의 기능 동작 속도를 향상시키는 가변 구조를 갖는 가속기

김민지[1], 송원재[2], *이성주[1]

[1]정보통신학과 및 지능형 융합드론학과, 세종대학교

[2]전자정보통신공학과, 세종대학교

minji@itsoc.sejong.ac.kr, wonjae@itsoc.sejong.ac.kr, *seongjoo@sejong.ac.kr

Accelerator with variable structure that improves the functional operating speed of unmanned vehicles

Minji Kim[1], Wonjae Song[2], Seongjoo Lee[1]*

[1]Dept. of Information and Comm. Eng. and Covergence Engineering for Intelligent Drone, Sejong Univ.

[2]Dept. of Electrical Eng., Sejong Univ.

요 약

현재 자율주행 이동체인 차량 및 드론이 발전함에 따라서 이와 관련된 연구도 활발하게 진행되고 있으며, 이에 따른 가속기 연구 또한 활발하게 진행되고 있다. 현재 다양하게 연구되고 있는 분야는 CPU와 GPU 그리고 다양한 종류의 FPGA가 설치 되어진 무인 이동체이다. 본 논문은 CPU와 GPU의 성능보다 뛰어난 DPU인 Alveo를 연구하였다. 계산량이 많고 복잡한 algorithm의 기능은 이런 기능을 동작했을 때의 속도가 무엇보다 중요하다. 그래서 CPU와 GPU 그리고 Alveo의 속도를 비교하여무인 이동체의 성능을 개선시키려고 했으며 이에 어울리는 가속기인 Alveo를 제안하였다.

Ⅰ. 서 론

4차 산업 혁명으로 자율주행 이동체에 대한 관심이 나날이 증가하고 있는 가운데 무인 이동체의 기능이 동작하는 속도를 향상시키는 연구가 진행되고 있다. 무인 이동체의 기능이 작동되는 속도를 향상시키고자 CPU와 GPU 그리고 다양한 종류의 FPGA를 설치할 때도 있다[1][2].또한 현재 무인 이동체에 자율주행 등 복잡한 연산을 하는 기능들을 위해서도 성능 좋은 CPU나 GPU 또는 다양한 종류의 FPGA가 무인 이동체에 설치되고 있다[3][4]. 또한 한 가속기에 정착하여 연산 실행률을 향상시키는 연구가 진행되고 있으며, 가속기의 단점을 해결하고자 algorithm을 개선하는 연구 또한 다양하게 진행되고 있다.

본 논문에서는 가속기들의 종류 중 하나인 CPU와 GPU의 성능 비교에 더불어 DPU의 성능을 함께 비교해 어떤 가속기가 무인 이동체에 설치될 때 기능 동작 속도를 가장 향상시킬 수 있는지 알아볼 것이다. 모든 종류의 GPGPU가 많지만 말 그대로 범용이기 때문에 성능에 한계가 있었다. Xilnx의 DPU인 Alveo는 실리콘에 고정된 단위 프로세서로 구성된 GPU와 달리 단위 프로세서 수준에서 하드웨어를 재구성하기 때문에 단위 프로세서당 전력 효율 측면에서 장점이 있는 가속기이다. 본 논문에서는 간단한 계산, vector addition 방법으로 Alveo의 계산 속도를 기존 CPU 및 GPU와 얼마나 다른지 비교한다. 구현은 Intel의 CPU와 NVIDIA의 GPU 그리고 Xilinx의 DPU인 Alveo를 이용해 구현을 진행할 것이다.

Ⅱ. 본론

본 논문에서는 GPGPU의 종류인 CPU, GPU 그리고 DPU의 계산 시간을 측정하면서 어떤 GPGPU가 가장 계산 속도가 뛰어난지 비교했다.

CPU, GPU 그리고 DPU인 Alveo의 계산 시간을 측정하기 위해 똑같은 프로그램을 이용했는데, 간단한 algorithm 중에서 vector addition을 사용했다. 서로 다른 배열을 더할 때 같은 index에 존재하는 값을 더해 연산하는 방식이다. 식으로 표현하면 다음과 같다.

$$C[n] = A[n] + B[n], (n = 0, 1, 2, \ldots, m) \cdots (1)$$

먼저 GPU의 계산 속도를 측정하기 위해서 Linux에서 Cuda 프로그램을 통해 계산 시간을 알아내었다. Cuda를 이용하기 위해서 .cu 파일을 생성해 코딩을 하고 test라는 파일을 생성했다. test라는 파일을 GPU 프로파일러를 이용해 실행하면 다음 그림에서 볼 수 있듯이, vector addition에서 사용된 모든 커널 및 메모리 복사본의 요약을 빠르게 확인할 수 있다.

그림 1과 같이 GPU의 vector addition 계산 시간은 약 180us이다.

다음으로 CPU와 DPU인 Alveo의 계산 속도를 측정하기 위해서 Linux에서 Vitis 프로그램을 통해 계산 속도를 알아내었다. Vitis 프로그램을 통해서 .cpp 파일에다가 코딩을 하고 build를 통해서 .xo 파일을 만들고 .xo

*교신저자 : 이성주

파일을 이용해서 Hardware 컴파일을 하여서 .xclbin 파일을 만든다. 그리고 Linux에서 .cpp 파일을 이용해 .exe 파일을 만든다. Alveo에 .xclbin을 다운로드 하고 .exe 파일을 이용해 .xclbin 파일을 실행한다. 이때 Linux의 debug 모드를 이용한다. debug 모드에서 실행을 시키면 아래의 그림과 같이 CPU와 DPU인 Alveo의 계산 시간이 다음과 같이 나온다.

그림 2와 같이 CPU의 vector addition 계산 시간은 약 23000us이고, Alveo의 계산 시간은 50us이다.

본 논문에서는 4,194,304개의 값을 가지고 각 vector들을 덧셈하여 CPU, GPU 그리고 Alveo의 계산 시간 차이를 비교하였다. 그리고 이를 통해서 Alveo의 연산 속도가 CPU보다는 약 460배 정도 빠르고 GPU보다는 약 3배 정도 빠른 것을 확인할 수 있었다. 그리고 이는 연산량이 많을수록 속도 차이가 커지는 것을 알 수 있었다. 밑의 그림은 16,777,216개의 값을 가지고 각 vector들을 덧셈하였을 때의 CPU, GPU 그리고 Alveo의 계산 시간이다.

그림 3, 4와 같이 CPU는 약 56327us이고, GPU는 약 721us이다. 마지막으로 Alveo는 약 40us이다. Alveo의 연산 속도가 CPU보다 약 1400배 정도 빠르고 GPU보다 약 18배 정도 빠른 것을 확인할 수 있다. 계산량이 커질수록 CPU, GPU와 Alveo의 계산 속도 차이가 커지는 것을 알 수 있다.

Ⅲ. 결론

본 논문에서는 CPU, GPU와 Alveo의 연산 속도 차이로 인한 Alveo의 성능이 뛰어남을 보여주었다. Alveo는 CPU를 뛰어넘은 가속기인 GPU도 우월한 성능을 보여주었고 이는 복잡한 알고리즘이나 계산량이 많을수록 더 잘 보여주는 것을 알 수 있었다. 따라서 Alveo는 무인 이동체 기능의 속도를 빠르게 향상시키는데 기여할 수 있을 것으로 기대되어 진다.

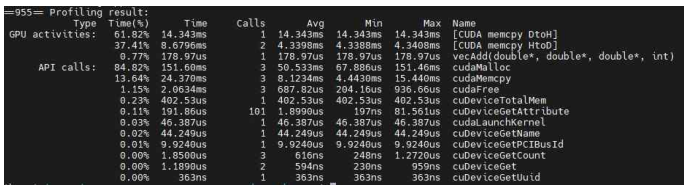

그림 1. 4,194,304개 값의 vector addition GPU 계산 시간

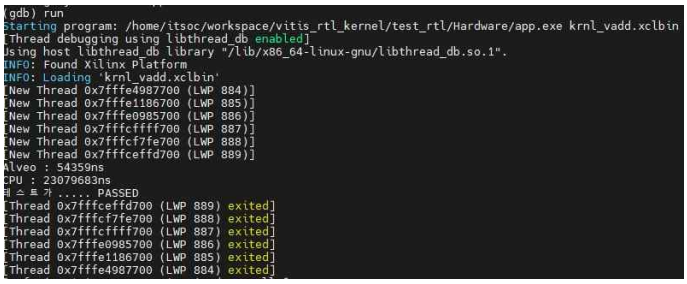

그림 2. 4,194,304개 값의 vector addition CPU와 Alveo 계산 시간

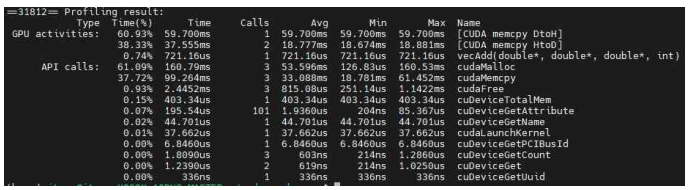

그림 3. 16,777,216개 값의 vector addition GPU 계산 시간

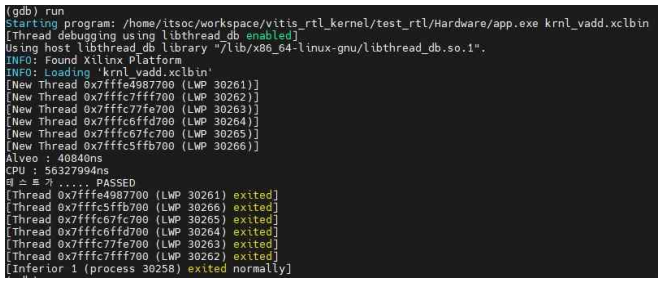

그림 4.. 16,777,216개 값의 vector addition CPU와 Alveo 계산 시간

표 1. CPU, GPU, Alveo의 계산시간과 속도 비교

	CPU	GPU	Alveo
계산시간	56327us	721us	40us
속도	1	80	1400

ACKNOWLEDGMENT

본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단 (No.2020R1A2C1007546) 및 정부(교육부)의 재원으로 한국연구재단의 이공분야 대학중점연구소지원사업 (No. 2020R1A6A1A03038540)의 지원을 받아 수행되었으며, 검증은 위한 EDA관련 툴은 IDEC의 지원을 받았음.

참 고 문 헌

[1] A. Fotouhi, M. Ding and M. Hassan, "Understanding autonomous drone maneuverability for Internet of Things applications," 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017, pp. 1-6, doi: 10.1109/WoWMoM.2017.7974336.

[2] R. Hossain, S. Magierowski and G. G. Messier, "GPU Enhanced Path Finding for an Unmanned Aerial Vehicle," 2014 IEEE International Parallel & Distributed Processing Symposium Workshops, 2014, pp. 1285-1293, doi: 10.1109/IPDPSW.2014.144.

[3] P. A. Rad, D. Hofmann, S. A. Pertuz Mendez and D. Goehringer, "Optimized Deep Learning Object Recognition for Drones using Embedded GPU," 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2021, pp. 1-7, doi: 10.1109/ETFA45728.2021.9613590.

[4] B. B. Kövari and E. Ebeid, "MPDrone: FPGA-based Platform for Intelligent Real-time Autonomous Drone Operations," 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2021, pp. 71-76, doi: 10.1109/SSRR53300.2021.9597857.