특집논문 (Special Paper)
방송공학회논문지 제30권 제2호, 2025년 3월 (JBE Vol.30, No.2, March 2025)
https://doi.org/10.5909/JBE.2025.30.2.170
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

실시간 3D 페이셜 생성을 위한 모듈 단위의 FPGA 가속기 설계

윤 승 환[a], 오 수 민[a], 이 학 범[a], 김 정 우[a], 이 혜 린[a], 서 영 호[a]‡

Modular FPGA Accelerator Design for Real-Time 3D Facial Generation

Seung-Hwan Yoon[a], Su-Min Oh[a], Hak-Bum Lee[a], Jung-Woo Kim[a], Hye-Rin Lee[a], and Young-Ho Seo[a]‡

요 약

본 연구는 음성 기반의 실시간 3D 페이셜 생성 시스템을 FPGA를 활용하여 설계하였다. 기존 연구는 모든 과정을 최적화하는 방식으로 효율성을 추구했으나, 수정이나 확장이 어려운 단점이 있었다. 이에 본 연구는 각 연산을 단계별로 독립적으로 구현하고 연결하는 방식으로, 실시간 동작을 유지하면서도 유연한 시스템을 구현했다. FPGA는 병렬 처리 성능이 뛰어나 3D 페이셜 생성 작업에 적합하며, 시스템을 모듈 단위로 분할하여 확장성과 유연성을 극대화했다. 이 접근은 개발 시간과 비용 절감은 물론, 향후 성능 향상과 기능 확장에 유리한 기반을 제공한다. 본 연구는 효율적인 실시간 데이터 처리와 대규모 연산 처리의 중요성을 강조하고, 다양한 응용 가능성을 제시한다.

Abstract

We present a system for real-time 3D facial generation based on voice input, utilizing FPGA technology. While existing research typically focuses on optimizing the entire process for efficiency, such designs often lack flexibility and are difficult to modify or expand. In contrast, we adopt a modular approach, implementing each operation independently and linking them together. This ensures real-time performance while maintaining system flexibility. FPGA is ideal for this task due to its high parallel processing capabilities, which are crucial for large-scale data processing and real-time responses in 3D facial generation. By designing the system in modular units, we maximize scalability and flexibility. This approach not only reduces development time and cost but also lays a foundation for future performance improvements and feature expansions. We emphasize the importance of efficient real-time data processing and large-scale computation handling, offering new possibilities for various applications.

Keyword : FPGA, SoC, 3D Facial Generation, Deep Learning, Real Time

a) 광운대학교 전자재료공학과(Department of Electronic Materials Engineering, Kwangwoon University)
‡ Corresponding Author : 서영호(Young-Ho Seo)
E-mail: yhseo@kw.ac.kr
Tel: +82-2-940-8362
ORCID: https://orcid.org/0000-0003-1046-395X

※ 이 논문의 연구 결과 중 일부는 2024년 한국방송·미디어공학회 2024년 추계학술대회에서 발표한 바 있음.
※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음. (IITP-2024-RS-2022-00156225)
※ 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2024년도 문화기술 연구개발사업으로 수행되었음. (과제명 : 정교한 한국어 발화가 가능한 3D 애니메이팅 AI를 이용한 웹툰 IP 활용 플랫폼의 개발, 과제번호 : RS-2024-00397183, 기여율: 100%)
· Manuscript March 3, 2025; Revised March 11, 2025; Accepted March 12, 2025.

Ⅰ. 서 론

음성 기반의 실시간 3D 페이셜 생성을 위한 시스템은 매우 복잡한 과정을 필요로 한다. 사용자의 음성을 녹음하고 이를 인코딩한 뒤, 해당 데이터를 전송하여 연산을 진행한다. 연산된 결과는 다시 PC로 전송되어 렌더링 과정이 이루어진다. 이러한 복잡한 연산과 데이터 흐름을 실시간으로 처리하기 위해서는 고성능의 하드웨어와 효율적인 시스템 설계가 필수적이다.

기존 연구들은 시스템의 모든 과정을 최적화하여 진행하는 방식이 주로 사용되었다[1-4]. 이러한 설계 방식은 높은 효율을 가지는 가속기를 만들 수 있지만, 수정이나 기능 확장이 어려운 단점이 있다. 본 연구에서는 이러한 문제를 해결하고자 모든 과정을 최적화하지 않고, 각 연산을 단계별로 독립적으로 구현하고, 이를 연결하는 방식으로 시스템을 설계하였다. 이를 통해 실시간 동작이 가능한 시스템을 구현하면서도, 기능 확장이나 수정이 용이한 유연한 시스템 구조를 확보할 수 있었다. 이러한 접근 방식은 개발 시간과 비용을 크게 절감하는 동시에, 향후 성능 향상 및 기능 확장에 유리한 기반을 마련하는 데 중요한 역할을 한다.

본 연구에서는 이러한 연산을 FPGA(Field Programmable Gate Array)에서 수행하도록 설계하였다. FPGA는 높은 병렬 처리 성능과 효율적인 연산 처리 능력을 제공하여, 실시간 연산에 적합한 환경을 구축할 수 있다. 특히, 반복적이고 병렬적인 연산에 강점을 지닌 FPGA는 대규모 데이터 처리와 실시간 응답을 요구하는 3D 페이셜 생성 작업에 이상적인 솔루션을 제공한다. 또한, 전체 과정을 모듈 단위로 분할하여 설계함으로써 각 모듈은 독립적으로 개발되고, 서로 호환 가능한 형태로 구성되어 시스템의 확장성을 극대화한다. 이러한 접근 방식은 시스템의 유연성을 확보하고, 향후 기능 확장이나 성능 향상을 위한 기반을 마련하는 데 중요한 역할을 한다.

본 연구에서 제시하는 FPGA 기반의 실시간 3D 페이셜 생성 시스템은 고성능 하드웨어를 통한 효율적인 연산과 데이터 흐름 처리를 가능하게 하며, 향후 다양한 분야에 응용 가능한 확장성을 제공한다. 이 연구는 실시간 데이터 처리 및 대규모 연산 처리의 필요성을 충족시키는 효율적인 시스템 설계의 중요성을 강조하고, 향후 다양한 응용 가능성에 대한 새로운 가능성을 제시한다.

Ⅱ. FPGA PS-Side Configuration

본 장에서는 PS 영역만을 활용하여 딥러닝 연산과 통신 기능을 구현하는 전체 개발 과정을 단계별로 설명한다. PS 영역의 유연성을 최대한 활용하여 초기 시스템 검증을 수행하고, 이후 PL 영역으로 확장을 위한 기반을 마련하는 것이 주요 목표이다. 본 장에서는 FPGA 메모리 및 통신 인터페이스 정의, 하드웨어 플랫폼 제작, 어플리케이션 개발, 그리고 Petalinux 환경 구성을 다룬다.

1. FPGA 하드웨어 플랫폼 제작

본 연구에서는 PS 영역 내의 메모리 자원으로 SD Card와 DDR을 활용한다. SD Card는 운영체제, 어플리케이션, 딥러닝 가중치 등 주요 데이터를 저장하는 주 기억장치로

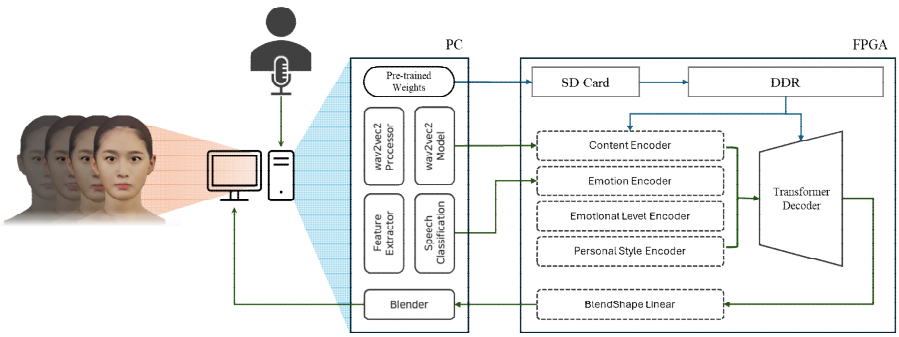

그림 1. 전체 시스템도
Fig. 1. Overall System Diagram

사용된다. 반면, DDR은 PS의 연산 수행에 필요한 데이터를 신속하게 처리하기 위한 임시 저장 공간으로 활용된다. PC에서 인코딩된 오디오 데이터는 Ethernet 포트를 통해 FPGA의 GEM(Gigabit Ethernet MAC)으로 전송된다. GEM으로 수신된 데이터는 LPD(Low-Power Domain) Main Switch로 전달된 후, IOP(Inbound) 경로를 통해 SD Card Controller로 이동하여 SD Card에 저장된다. 이후 저장된 데이터는 SD Card Controller를 통해 IOP(Outbound) 경로로 전송되고, LPD Main Switch를 거쳐 CCI(Cache Coherent Interconnect)를 통해 DRAM Controller로 전달되어 DRAM에 저장된다. DRAM에 저장된 데이터는 CCI를 통해 APU로 이동하며, 이곳에서 딥러닝 연산이 수행된다. 연산 결과는 다시 DRAM에 저장되고, LPD Main Switch와 IOP(Outbound) 경로를 거쳐 GEM으로 전달된다. 최종적으로, 결과 데이터는 Ethernet을 통해 PC로 전송된다. 이러한 체계적인 데이터 경로는 시스템의 효율적 운영과 안정성을 보장하며, 향후 PL 영역으로의 모듈 단위 확장을 위한 견고한 기반을 제공한다.

2. 어플리케이션 개발

PS 영역에서 동작하는 어플리케이션은 C++ 기반으로 개발되며, 딥러닝 연산을 위한 모델은 PS 상에서 최적화된 환경에 맞춰 구현된다. PC에서 SD Card로 데이터를 전송할 때는 SSH 프로토콜을 활용하며, 연산 결과를 PC로 전송할 때는 소켓 프로그래밍 기법을 적용하여 TCP 기반의 안정적인 실시간 데이터 전송을 보장한다. 본 시스템에서 서버는 3D 모델 렌더링을 담당하며, 클라이언트는 FPGA에서 추론된 BlendShape 계수를 실시간으로 서버에 전달한다[6]. 서버는 Blender 3D 렌더링 엔진을 활용하여 52개의 BlendShape 계수에 대한 3D 얼굴 애니메이션을 처리하고, 클라이언트는 TCP 프로토콜을 사용하여 서버와 데이터를 교환한다[7]. 서버 소켓은 60fps의 프레임 속도를 유지하면서 비동기적으로 데이터를 처리하며, 클라이언트는 지속적으로 정보를 서버에 전달하여, 수신된 데이터를 타임코드에 맞춰 3D 모델을 렌더링하는 방식으로 실시간 3D 얼굴 애니메이션을 생성한다. 이와 같이 최적화된 네트워크 통신 구조는 실시간 렌더

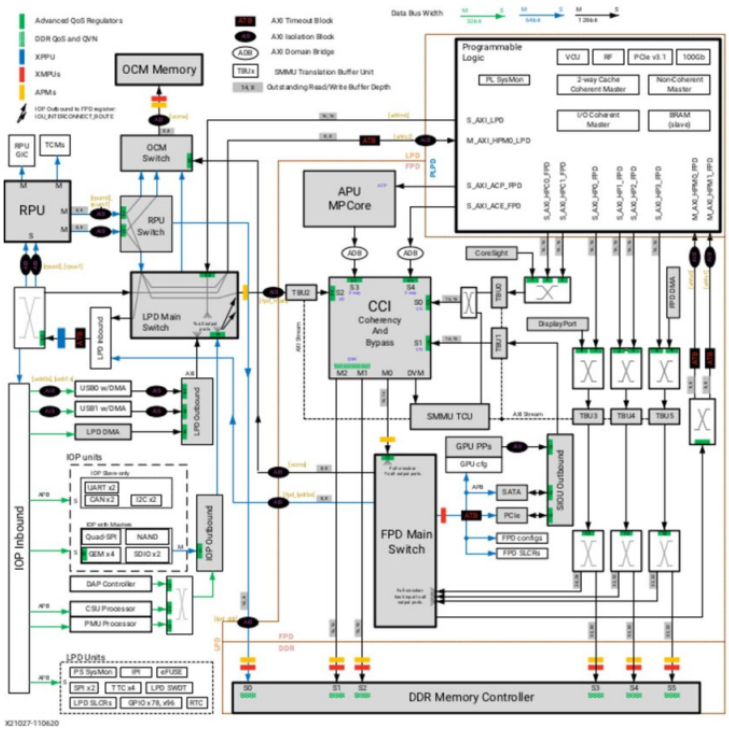

그림 2. Zynq UltraScale+ Processing System의 블록도[5]
Fig. 2. Block Diagram of Zynq UltraScale+ Processing System[5]

링의 요구 사항을 충족시킴과 동시에, 전체 시스템의 높은 효율성과 낮은 지연을 유지하도록 설계되었다.

3. Petalinux 환경 구성

PS 영역에서 운영되는 소프트웨어 환경을 구축하기 위해 Petalinux[8] 기반의 운영체제를 구성하였다. Xilinx SDK를 활용하여 Petalinux 프로젝트를 생성하고, FPGA 보드에 최적화된 커널 및 파일 시스템을 구성하였다. FPGA 메모리 및 통신 인터페이스와 연동되는 커널 드라이버와 사용자 어플리케이션을 개발하여 딥러닝 연산 및 데이터 전송 기능의 초기 검증을 수행하였다.

Ⅲ. FPGA PL-Side Design

3장에서는 PL 영역 설계를 다룬다. 본 연구에서는 전체 딥러닝 모델 중 Transformer[9][10]를 PL 영역에 구현하였으며, 이를 위해 PS-PL 인터페이스 및 하드웨어-소프트웨어 파티셔닝을 수행하였다. 또한, AXI 전송 속도를 고려하여 행렬곱 연산을 적절히 분할하여 효율적인 데이터 처리가 가능하도록 설계하였다.

1. PS-PL 인터페이스

PS와 PL이 비동기적으로 데이터를 송수신하기 위해 Dual Port BRAM을 활용한 인터페이스를 구성하였다. 이 인터페이스는 데이터 영역과, 해당 데이터를 현재 PS 또는 PL 중 어느 쪽이 사용 중인지를 나타내는 헤더 영역으로 구성된다. 초기에는 헤더 영역이 모두 0(PS)으로 설정된다. PS가 데이터 영역에 데이터를 입력한 후 헤더 영역의 값을 1(PL)로 전환하면, PL은 이를 감지하고 데이터 영역을 읽어 연산을 시작한다. 연산이 완료되면, PL은 결과 데이터를 데이터 영역에 저장하고 헤더 영역의 값을 다시 0(PS)으로 전환한다. PS는 헤더 영역의 값이 0임을 확인한 후 데이터 영역을 읽어 후속 처리를 진행한다.

이 방법은 PS와 PL이 독립적으로 동작하면서 데이터를 주고받을 수 있도록 구성되어 있어, 연산이 블로킹되지 않

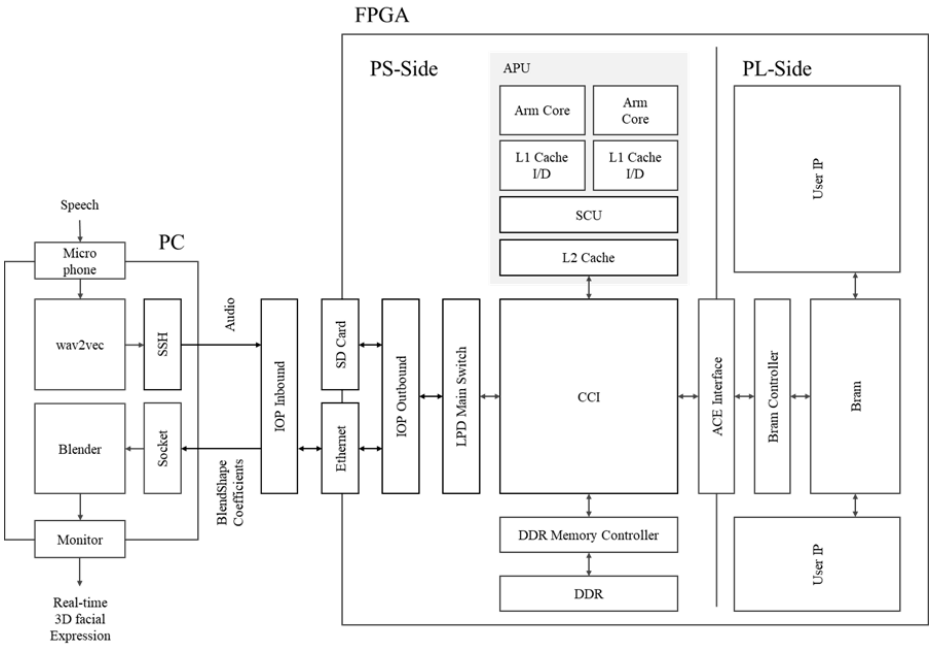

그림 3. 구현된 하드웨어 플랫폼의 블록도
Fig. 3. Block Diagram of Implemented Hardware Platform

고 비동기적으로 수행될 수 있다. 헤더 영역을 활용한 데이터 교환 방식은 복잡한 동기화 메커니즘이 필요하지 않아 제어 로직이 단순하며, Dual Port BRAM을 통해 PS와 PL이 동시에 메모리에 접근할 수 있어 오버헤드가 적다. 또한, 인터페이스 구조가 간단하여 향후 다른 연산 모듈을 추가하거나 데이터 흐름을 확장하는 것이 용이하다. 마지막으로, 헤더 플래그를 통해 현재 데이터가 PS와 PL 중 어느 영역에서 사용 중인지 쉽게 파악할 수 있어 디버깅 및 검증이 용이하다.

2. 하드웨어-소프트웨어 파티셔닝

하드웨어-소프트웨어 파티셔닝은 시스템 성능과 자원 활용을 최적화하기 위해 특정 연산을 하드웨어(PL)와 소프트웨어(PS)로 분할하는 과정이다. 본 연구에서는 Transformer 모델의 연산 중 계산량이 크고 병렬 처리가 효과적인 부분을 PL에서 수행하고, 상대적으로 제어가 중요한 연산은 PS에서 담당하도록 설계하였다[11][12].

PL 영역에서는 높은 연산량을 요구하는 행렬곱 연산을 병렬화하여 구현하였다. 행렬곱은 다수의 곱셈-덧셈 연산으로 구성되며, FPGA의 병렬 구조를 활용하면 CPU나 GPU보다 더 높은 처리 성능을 얻을 수 있다. 이를 위해 Systolic Array 구조를 기반으로 Processing Element(PE) 배열을 구성하여 다중 데이터 스트림을 동시에 처리할 수 있도록 하였다. 또한, AXI 인터페이스의 전송 속도를 고려하여 행렬곱 연산을 적절한 크기로 분할하고, 데이터 흐름을 최적화함으로써 메모리 접근 병목 현상을 최소화하였다.

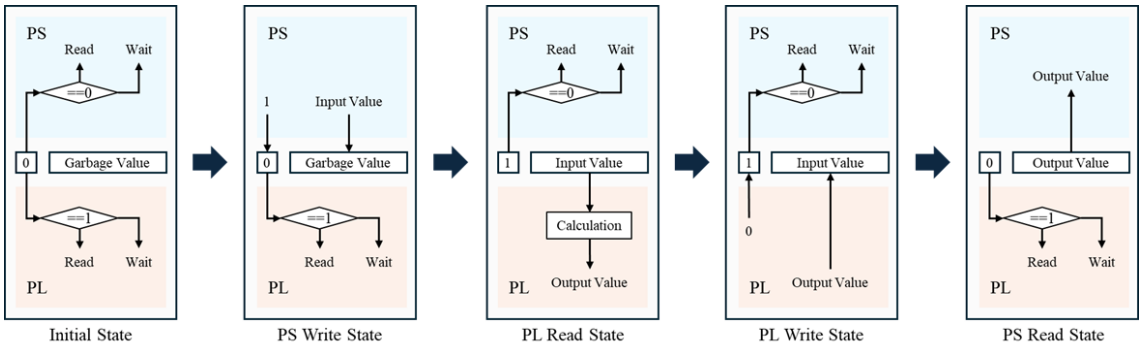

그림 4. Dual-Port BRAM을 사용한 헤더 기반의 PS-PL 인터페이스의 동작 과정
Fig. 4. Operation process of the PS-PL interface based on header using Dual-Port BRAM

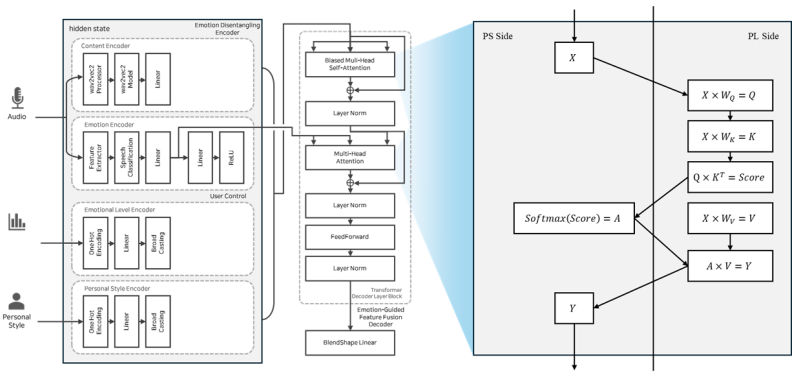

그림 5. Transformer 연산의 하드웨어-소프트웨어 파티셔닝 예시
Fig. 5. Hardware-Software Partitioning Example of Transformer Operations

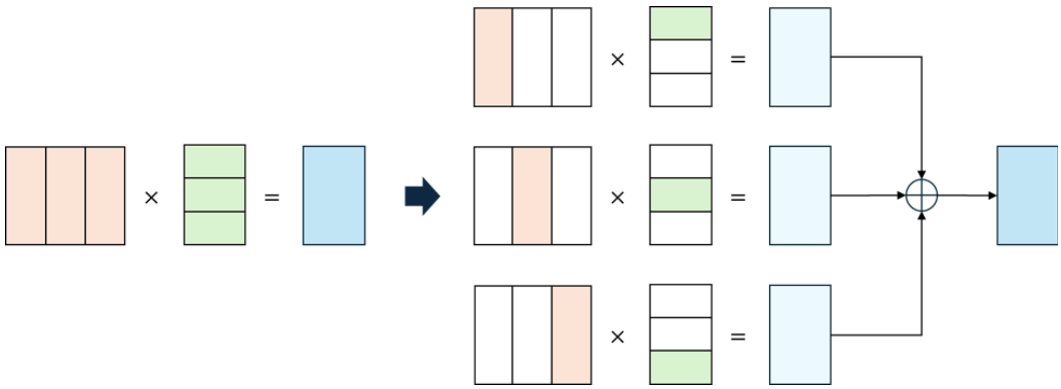

그림 6. 행렬곱의 분할 수행
Fig. 6. Matrix Multiplication Partitioning

이를 통해 높은 연산 성능을 유지하면서도 효율적인 데이터 전송이 가능하도록 설계하였다.

반면, Softmax 연산은 PS에서 수행하도록 결정하였다. Softmax는 지수 함수와 나눗셈 연산이 포함되어 있어 부동소수점 연산이 필요하며, FPGA에서 직접 구현할 경우 자원 소모가 크고 복잡한 연산 과정이 요구된다. 따라서 PS에서 Softmax 연산을 처리하고, 결과를 다시 PL로 전달하는 방식으로 시스템을 구성하였다.

이러한 하드웨어-소프트웨어 파티셔닝 전략은 성능과 자원 활용의 균형을 맞추고, 모듈 단위의 구현을 통해서 시스템의 확장성과 유지보수성을 향상시킨다.

Ⅳ. 연구 결과

본 연구에서는 초기 PS 기반 모델에서 한 단계씩 진행할 때마다 End-to-End 방식으로 검증을 수행하였으며, 검증 방법으로는 렌더링 결과의 PSNR(Peak Signal-to-Noise Ratio)을 측정하였다. 이는 데이터 입출력, 연산, 스트리밍까지 포함한 전체 과정을 정량적으로 평가하기 위함이다.

검증 대상인 3D Facial Expression Model로는 EmoTalk[10]을 사용하였다. EmoTalk은 음성 신호를 입력으로 받아 감정을 반영한 얼굴 표정을 생성하는 3D 얼굴 애니메이션 모델이다. 본 연구에서는 EmoTalk에서 정의한 8가지 감정을 기반으로 감정의 강도, 발화 속도, 문장 내용, 그리고 발화자의 특성을 조합하여 총 320개의 샘플을 추출하였다. 각 샘플에 대해 원본 모델과 FPGA에서 스트리밍된 3D 얼굴 데이터를 비교하였으며, 이를 위해 두 결과를 각각 Blendshape 계수 형태로 출력한 후, 3D 메쉬를 생성하고 정면에서 2D 이미지로 변환하여 PSNR 값을 계산하였다. 이를 통해 데이터 입력부터 최종 렌더링까지의 전 과정을 포함하는 End-to-End 검증을 수행하였으며, FPGA 기반 연산이 원본 모델과의 동일성을 검증하였다.

검증 결과, 30FPS에서 평균 PSNR이 90dB 이상으로 측정되었다. 이는 FPGA 기반 연산을 통해 생성된 3D 얼굴이 원본 모델과 동일하다는 것을 시사한다. 또한, Emotalk이 30FPS 영상을 목표로 하는 만큼, FPGA 기반 가속기의 정량 목표가 30FPS라는 점을 고려할 때, 개발된 FPGA 기반 아키텍처가 높은 연산 정확도를 요구하는 3D 얼굴 애니메이션 시스템에서도 효과적으로 활용될 수 있음을 확인하였다.

표 1. 렌더링 결과에 대한 PSNR 측정 결과
Table 1. PSNR Measurement Results of Rendering Output

Emotion	PSNR(dB) ↑
Neutral	93.781
Calm	92.766
Happiness	91.854
Sadness	92.575
Anger	92.002
Fear	92.622
Disgust	91.753
Surprised	92.938

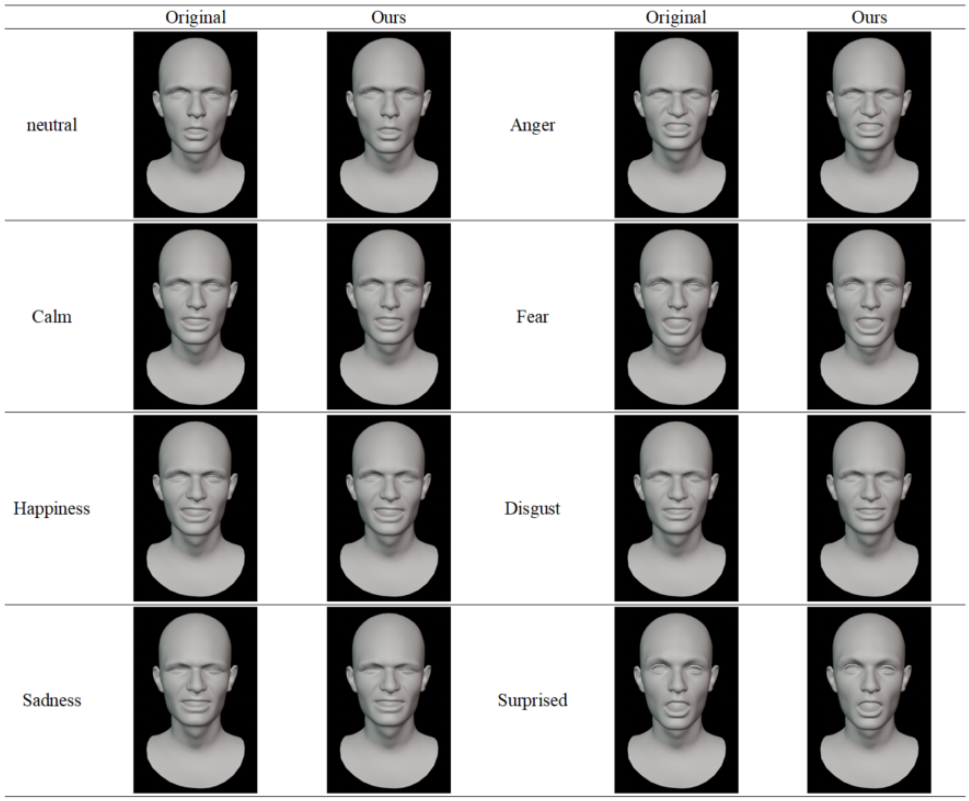

그림 7. Emotalk의 렌더링 결과 (HDTF)
Fig. 7. Rendering Results of Emotalk Output (HDTF)

그림 8. 텍스처를 적용한 실시간 렌더링 결과
Fig. 8. Real-Time Rendering Results with Texture

V. 결 론

본 연구에서는 실시간 3D 페이셜 생성을 위한 시스템을 FPGA(Field Programmable Gate Array)를 활용하여 설계하였다. 이 시스템은 음성을 녹음하고 인코딩한 후, 이를 전송하여 연산을 진행하며, 연산된 결과를 PC로 전송해 렌더링하는 복잡한 과정을 실시간으로 처리할 수 있다. FPGA의 병렬 처리 성능과 효율적인 연산 처리 능력은 이러한 복잡한 작업을 실시간으로 처리하는 데 중요한 역할을 한다.

이 연구에서 제시하는 접근법은 기존의 방식과는 달리, 모든 과정을 최적화하지 않고 단계별로 독립적으로 개발한 후 연결하는 방식으로 실시간 3D 페이셜 생성 시스템을 구현할 수 있음을 보여준다. 기존 연구들은 시스템의 모든 과정을 최적화하여 높은 효율을 가지는 가속기를 만드는 데

초점을 맞추지만, 이러한 설계 방식은 수정이나 기능 확장이 어려운 문제를 동반한다. 반면 본 연구에서는 각 모듈을 독립적으로 개발하여 연결하는 방식으로 설계함으로써, 시스템의 유연성을 극대화하고, 기능 확장과 수정이 용이하게 하였다. 이 접근 방식은 시스템의 확장성을 높이고, 초기 모델 검증과 빠른 프로토타이핑을 가능하게 하여, 개발 시간과 비용을 크게 단축시킬 수 있다. 특히, 복잡한 실시간 연산을 요구하는 시스템에서 이러한 단계별 접근법은 실시간 동작을 위한 효율적인 하드웨어 가속기를 구축하는 데 중요한 기여를 한다.

참 고 문 헌 (References)

[1] Ullah, Salim, et al. “High-performance accurate and approximate multipliers for FPGA-based hardware accelerators.” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 41.2 (2021): 211-224.
doi: https://doi.org/10.1109/TCAD.2021.3056337

[2] Mani, V. R. S., A. Saravanaselvan, and N. J. M. J. Arumugam. “Performance comparison of CNN, QNN and BNN deep neural networks for real-time object detection using ZYNQ FPGA node.” Microelectronics Journal 119 (2022): 105319.
doi: https://doi.org/10.1016/j.mejo.2021.105319

[3] C. H. Yu, H. E. Kim, S. Shin, K. Bong, H. Kim, Y. Boo, ... & J. Oh, “2.4 ATOMUS: A 5nm 32TFLOPS/128TOPS ML System-on-Chip for Latency Critical Applications,” Proceedings of the 2024 IEEE International Solid-State Circuits Conference (ISSCC), vol. 67, pp. 42-44, Feb. 2024. IEEE.
doi: https://doi.org/10.1109/ISSCC49657.2024.10454509

[4] H. Chen, J. Zhang, Y. Du, S. Xiang, Z. Yue, N. Zhang, ... & Z. Zhang, “A comprehensive evaluation of FPGA-based spatial acceleration of LLMs,” Proceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 185-185, Apr. 2024.
doi: https://doi.org/10.1145/3656177

[5] Xilinx, Zynq UltraScale+ Device Technical Reference Manual, UG1085, v2.4, Dec. 2023. Retrieved from https://docs.amd.com/viewer/book-attachment/dqE2tE0k~iMhpEDoQwXIKg/11~gTSJf1Bn~faUQSW8eVw (accessed Dec. 21, 2023).

[6] Lewis, John P., et al. “Practice and theory of blendshape facial models.” Eurographics (State of the Art Reports) 1.8 (2014): 2.
doi: https://doi.org/10.2312/egst.20141042

[7] Xue, Ming, and Changjun Zhu. “The socket programming and software design for communication based on client/server.” 2009 Pacific-Asia Conference on Circuits, Communications and Systems. IEEE, 2009.
doi: https://doi.org/10.1109/PACCS.2009.89

[8] Xilinx, PetaLinux Tools Documentation: Reference Guide, UG1144, v2024.1, June 2024. Retrieved from https://docs.amd.com/viewer/book-attachment/MVyApcmU3R9Mm97zSMBTWg/A1uhF~YnkvK0u6G775Tu_Q (accessed June 21, 2024).

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, ... & I. Polosukhin, “Attention is all you need,” Proceeding of Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.
doi: https://doi.org/10.48550/arXiv.1706.03762

[10] Z. Peng, H. Wu, Z. Song, H. Xu, X. Zhu, J. He, H. Liu, and Z. Fan, “EmoTalk: Speech-Driven Emotional Disentanglement for 3D Face Animation,” Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Vol. 34, pp. 997-998, Dec. 2023.
doi: https://doi.org/10.48550/arXiv.2303.11089

[11] Joulin, Armand, et al. “Efficient softmax approximation for GPUs.” International conference on machine learning. PMLR, 2017.
doi: https://doi.org/10.48550/arXiv.1609.04309

[12] López-Vallejo, Marisa, and Juan Carlos López. “On the hardware-software partitioning problem: System modeling and partitioning techniques.” ACM Transactions on Design Automation of Electronic Systems (TODAES) 8.3 (2003): 269-297.
doi: https://doi.org/10.1145/785411.785412

저 자 소 개

윤 승 환

- 2025년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2025년 3월 ~ 현재 : 광운대학교 전자재료공학과 일반대학원(석사과정)
- ORCID : https://orcid.org/0009-0004-2066-6376
- 주관심분야 : 하드웨어 설계, 딥러닝, 영상 코덱

저 자 소 개

오 수 민

- 2025년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2025년 3월 ~ 현재 : 광운대학교 전자재료공학과 일반대학원(석사과정)
- ORCID : https://orcid.org/0009-0008-9583-5399
- 주관심분야 : 하드웨어 설계, 딥러닝, 영상 코덱

이 학 범

- 2024년 2월 : 광운대학교 전자재료공학과학과 졸업(공학사)
- 2024년 3월 ~ 현재 : 광운대학교 전자재료공학과 일반대학원(석사과정)
- ORCID : https://orcid.org/0000-0003-0721-4944
- 주관심분야 : 다중 카메라 Calibration, 3D 휴먼 모션 재구성, NPU설계

김 정 우

- 2024년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2024년 3월 ~ 현재 : 광운대학교 전자재료공학과 일반대학원(석사과정)
- ORCID : https://orcid.org/0009-0003-0913-0709
- 주관심분야 : 하드웨어 설계, 딥러닝, LLM

이 혜 린

- 2022년 2월 ~ 현재 : 광운대학교 전자융합공학과 재학
- ORCID : https://orcid.org/0009-0008-3657-1565
- 주관심분야 : 하드웨어 설계, 딥러닝, 영상 코덱

서 영 호

- 1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2001년 2월 : 광운대학교 일반대학원 졸업(공학석사)
- 2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)
- 2005년 9월 ~ 2008년 2월 : 한성대학교 조교수
- 2008년 3월 ~ 현재 : 광운대학교 전자재료공학과 교수
- ORCID : https://orcid.org/0000-0003-1046-395X
- 주관심분야 : 실감미디어, 2D/3D 영상 신호처리, SoC 설계, 디지털 홀로그램