

# FPGA 가속기를 활용한 실시간 차선 유지 시스템 개발에 관한 연구

이채원<sup>1</sup>, 김민하<sup>1</sup>, 한지윤<sup>1</sup>, 홍수빈<sup>1</sup>, 신수경<sup>1</sup><sup>1</sup>이화여자대학교 전자전기공학부 학부생smcwlove@ewhain.net, minjang7910@ewha.ac.kr, tjpyk@ewhain.net  
hsb0531@ewhain.net, wtrgls2612@ewhain.net

## A Study on Real Time LDWS using FPGA Accelerator (Lane Detection Warning System)

Chae-won Lee<sup>1</sup>, Min-Ha Kim, Ji-Yun Han<sup>3</sup>,Su-Been Hong<sup>4</sup>, Soo-Kyung Shin<sup>5</sup><sup>1</sup>Dept. of Electronic Electrical Engineering, EWHA Womans University

### 요 약

본 연구는 FPGA 가속기를 활용하여 실시간으로 차선을 검출하고, 이를 유지하는 시스템을 개발한다. 차선 검출에는 Sobel Filter 와 Hough 변환을 이용하며 실시간을 위한 데이터 처리 속도 개선에는 FPGA 의 PL Logic 과 메모리 최적화 기법을 사용한다. 이로써 설치 용이한 부착형 방식의 LDWS 를 통해 낮은 수준의 자율 주행을 가능케 한다.

#### 1. 서론

본 연구는 FPGA 가속기를 활용하여 실시간으로 차선을 검출하고 이를 유지하는 시스템을 개발한다. 이 시스템은 카메라를 통해 실시간으로 영상을 입력 받고, Sobel Filter 와 Hough 변환을 이용해 차선을 검출한 후, 차선의 이탈 여부를 자체적으로 판단한다. FPGA 의 PL Logic 과 Memory Access 경로를 분할을 통해 데이터 처리 성능을 향상시키고 차선 이탈 시 경보를 울려 사용자가 실시간으로 위험을 감지하도록 한다.

#### 2. 기존 시스템과의 차별점

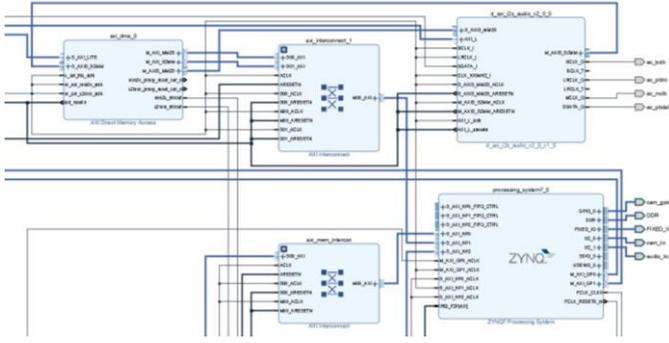
최근 출시되고 있는 차량들은 대부분 Radar 및 LiDAR 를 활용하는 LDWS 가 기본으로 장착되어 있다. 그렇지 못한 구형 차량에는 이러한 시스템들이 적용되어 있지 않다. 구형 차량들이 LDWS 를 장착하기 위해서는 대략 50 만원 이상의 고가의 장비를 정비소에서 장착하는 방법이 유일하다. 따라서 본 프로젝트는 저비용이며 간단한 장치로 구성되어 개인이 설치 가능한 부착형 LDWS 장비를 개발하는 데 차별점을 둔다.

#### 3. 객체 탐지 알고리즘

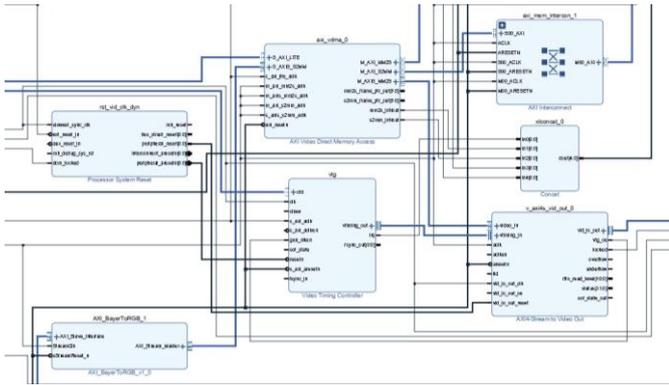
Pcam 5c 카메라 모듈로부터 실시간으로 입력 받은 영상을 Sobel Filter 와 Hough 변환을 활용하여 전처리하고, 이를 차선의 유형(흰색 점선, 흰색 실선, 황색 선)별로 탐지하여 객체 분류를 시행한다. 이로써 검출된 차선을 바탕으로, 차량이 차선을 이탈할 경우 오디오 코덱을 이용하여 경보음을 사용자에게 전달한다. Sobel Filter 를 사용하여 영상 내 물체의 edge 를 검출하고, Hough 변환을 통해 검출된 edge 에서 직선 및 곡선을 탐지하여 차선을 인식한다. 흰색과 황색 차선을 구분하기 위해 HSV 색상 마스크를 적용하여 각 차선의 색상을 분류하고, 차선의 기울기와 위치 정보를 바탕으로 차선을 추출한다. 사용자의 도로 주행 과정에서 차선 이탈이 감지되었을 때, 경보음 및 경고 문구를 차선 유형에 따라 단계를 나누어 각 단계별로 다른 경보음을 사용하여 사용자에게 알린다.

#### 4. 하드웨어 아키텍처

제기된 시스템은 FPGA Part 로 Xilinx 사의 7 series Programmable Logic(PL), ARM Cortex-A9 Processing System(PS), 이외의 경우 Zynq 7000 series 의 SoC 를 사



[그림 1] Audio Codec Block connected to DMA



[그림 2] Video Stream Block connected to DMA

용하였다. Pcam 의 영상 데이터는 실시간 고속 처리를 위해 MIPI D-PHY 와 CSI, 그리고 DVI 블록을 적용하였으며, Master 와 Slave 간 연결을 단순화시켜 데이터 전송의 복잡도를 개선하고자 AXI 통신 블록을 활용하였다.

아키텍처는 데이터 속도 개선을 위하여 첫째, [그림 1], [그림 2]와 같이 DMA(Direct Memory Access) Block 을 활용함으로써 주변 장치인 Pcam 모듈과 Audio Codec 이 CPU 로 접근하지 않고도 각각의 Slave 가 Master 인 PS 에 Interrupt 를 통해 통신할 수 있도록 구성하였으며 둘째, Memory 형태로써 Quad SPI Flash 를 사용, 4 bit 를 병렬 처리할 수 있도록 선정하였다. 또한 DMA 의 경우 실시간 영상 처리를 위한 Pcam 모듈의 영상 처리 부분을 Audio 모듈과 분리, 병목 현상을 방지할 수 있도록 설계하였다. [표 1]은 제기된 시스템의 Resource Utilization 결과이다.

[표 1] Resource Utilization for the proposed system

Xilinx Platform	Zynq 7020
LUTs	9391 (17.65%)
Flip flops	14917 (14.02%)
BRAMs	15 (10.71%)

### 5. 결과 및 성능 분석

[표 2]는 본 연구의 시스템과 기존 시스템의 성능을 비교한 결과이다. 제기된 시스템은 60 FPS 의 실시간 성능과 낮은 전력 소모를 보여준 반면, GPU 기반 시스템은 향상된 FPS 와 높은 전력 소모량을 보인다. 이외 CPU 기반의 시스템은 0.48 로 실시간 처리가 어려운 정도의 낮은 성능을 보였다.

System	FPS	Power
FPGA (Sobel + Hough)	60 FPS	Low
Legacy FPGA	30 FPS	Middle
GPU	118 FPS	High
CPU	0.48 FPS	Low

[표 2] System Performance Comparison

### 6. 결론

본 연구에서는 FPGA 가속기를 활용한 실시간 차선 탐지 시스템의 설계 및 구현 방법을 제시하였다. 구현된 시스템은 저비용이며 간단한 장치로 구성되어, 구형 차량에도 자율주행의 보조를 이룰 수 있어 효율적으로 활용될 수 있다. 향후 연구로써 도로 및 교통 상황에서의 변수를 폭 넓게 고려, 시스템 성능을 개선하고 훼손된 차선 또한 검출이 가능하도록 알고리즘을 발전시켜 정확도를 향상시키고자 한다.

### 참고문헌

- [1] Saidani, T., Ghodhbani, R., Alhomoud, A., Alshammari, A., Zayani, H. and Ben Ammar, M., "Hardware Acceleration for Object Detection using YOLOv5 Deep Learning Algorithm on Xilinx Zynq FPGA Platform," Engineering, Technology & Applied Science Research. 14, 1, 13066–13071, 2024
- [2] Babu, P., Parthasarathy, E., "Hardware acceleration for object detection using YOLOv4 algorithm on Xilinx Zynq platform," J Real-Time Image Proc 19, 931–940, 2022
- [3] Z. Li and J. Wang, "An improved algorithm for deep learning YOLO network based on Xilinx ZYNQ FPGA," 2020 International Conference on Culture-oriented Science & Technology (ICCST), Beijing, China, 2020, pp. 447-451
- [4] 윤신필, 노동건. , "OpenCL 을 이용한 FPGA 기반 신호처리 가속기," 한국통신학회 학술대회논문집, 2017

※ 본 논문은 과학기술정보통신부 대학디지털교육역량강화 사업의 지원을 통해 수행한 ICT 멘토링 프로젝트 결과물입니다.