TASK 8

Group 6

Abhinay Samala, Ankush Choudhary, Chhavi, David, Disha Shakyawal, Pasupulati Madhulatha, Parikshit Ramchandra Sahu, Supriya

Introduction

Advanced Driver Assistance Systems (ADAS) are crucial for enhancing vehicle safety and driver convenience. With the rise of electric vehicles (EVs), integrating ADAS into EVs poses unique challenges and opportunities, particularly concerning power efficiency and real-time processing capabilities. Field-Programmable Gate Arrays (FPGAs) offer a flexible and powerful platform for implementing ADAS functionalities due to their parallel processing capabilities, configurability, and low power consumption.

Objectives

1. Develop a high-performance, low-latency FPGA architecture for ADAS in EVs.
2. Ensure the architecture supports multiple ADAS functions, including object detection, lane departure warning, and adaptive cruise control.
3. Optimize power efficiency to align with EV constraints.

Requirements

1. **Real-Time Processing**: The FPGA must handle high-speed data from multiple sensors (cameras, LiDAR, radar) with minimal latency.
2. **Parallelism**: Utilize the FPGA's parallel processing capabilities to manage multiple ADAS functions simultaneously.
3. **Flexibility**: The architecture should be easily reconfigurable to adapt to different ADAS algorithms and updates.
4. **Power Efficiency**: Minimize power consumption to extend the EV's range.
5. **Integration**: Seamlessly integrate with the vehicle's existing electronic control units (ECUs) and communication protocols (CAN, Ethernet).

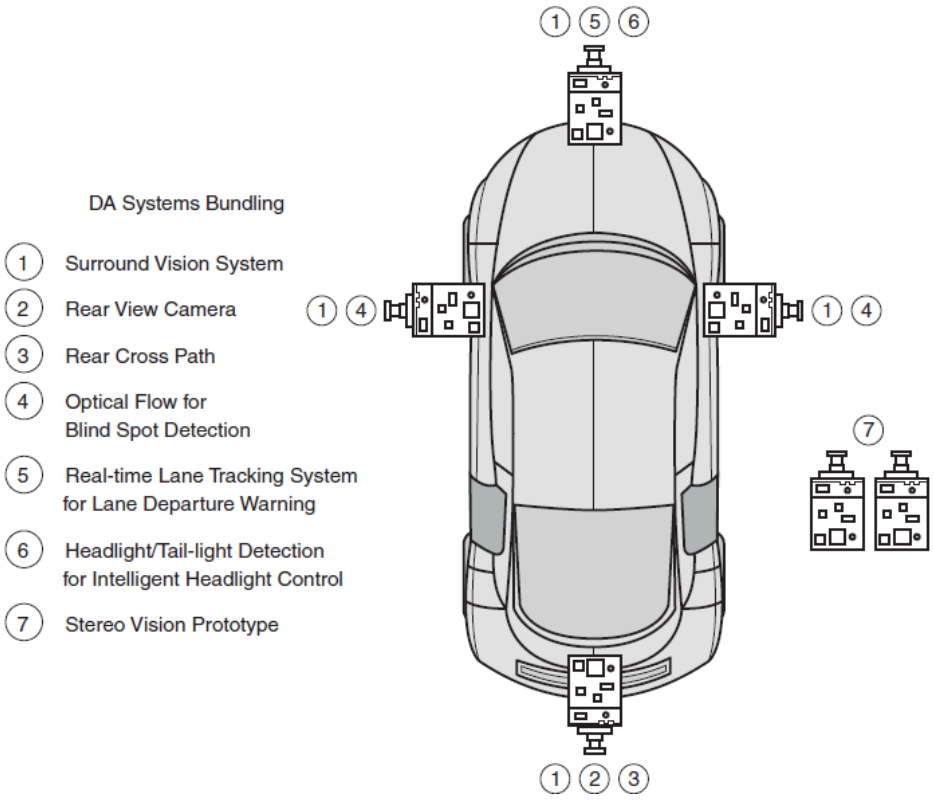

Fig 1 : Bundling Multiple Automotive Features

FPGA Architecture Design

1. Sensor Interface Module

- **Function**: Collect and preprocess data from various sensors.
- **Components**: Multiple high-speed I/O ports, A/D converters, data buffers.
- **Key Considerations**: Ensure low-latency and high-throughput data transfer.

2. Data Processing Module

- **Function**: Execute ADAS algorithms such as object detection, lane detection, and adaptive cruise control.
- **Components**:
    - **Custom Processing Units (CPUs)**: Implement specific ADAS functions using hardware description languages (HDL) like VHDL or Verilog.
    - **Digital Signal Processing (DSP) Blocks**: Accelerate complex mathematical computations required for image and signal processing.
- **Key Considerations**: Optimize for parallel processing to handle multiple functions concurrently.

3. Memory Management Module

- **Function**: Efficiently manage data storage and retrieval.
- **Components**:
    - **On-chip Memory**: Fast access memory for real-time data.
    - **External Memory Interfaces**: Interfaces for off-chip memory (DDR, SRAM) for large data storage.
- **Key Considerations**: Balance speed and capacity to ensure smooth data flow.

4. Communication Interface Module

- **Function**: Facilitate data exchange between the FPGA and the vehicle's ECUs.
- **Components**: CAN, Ethernet, and other automotive communication protocols.
- **Key Considerations**: Ensure compatibility and low-latency communication.

5. Power Management Module

- **Function**: Optimize power usage of the FPGA and its components.
- **Components**: Power gating, dynamic voltage, and frequency scaling (DVFS).
- **Key Considerations**: Minimize power consumption without compromising performance.

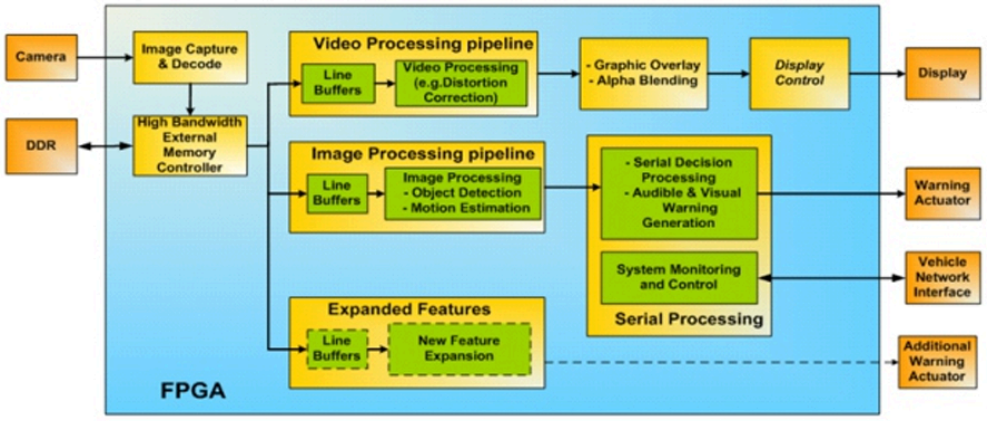

Fig 2 : Camera based ADAS using FPGA

Development Process

1. Requirement Analysis

- Collaborate with automotive engineers and ADAS experts to define specific requirements and constraints.

2. System Design

- Create a high-level architecture diagram.
- Define interfaces and data flow between modules.

3. Module Development

- Develop and test individual modules in parallel.
- Use simulation tools to verify functionality and performance.

4. Integration and Testing

- Integrate modules and test the complete system.
- Perform hardware-in-the-loop (HIL) testing to ensure real-world performance.

5. Optimization

- Profile power consumption and optimize where necessary.
- Fine-tune algorithms for improved accuracy and efficiency.

6. Deployment

- Deploy the FPGA in a prototype vehicle for field testing.
- Gather data and feedback for further refinement.

Conclusion

Developing an FPGA architecture for ADAS in EVs requires a careful balance of performance, flexibility, and power efficiency. By leveraging the parallel processing capabilities of FPGAs and following a structured development process, it is possible to create a robust ADAS solution that enhances the safety and convenience of electric vehicles.

References

- https://www.slideshare.net/slideshow/advanced-driver-assistance-system-using-fpga/38689054
- https://docs.amd.com/v/u/en-US/wp399_Auto_DA_Systems