# Task No – 08: FPGA architecture for an ADAS solution for an EV Vehicle

**Date –** 16th July 2024

**Members of team:** Nichenametla Yadhu Vamsi, Rashmi Singh, Chandan Kumar, Vishesh Chhaperwal, Kush Thakur, Ranga Sudharani, Khushi Prajapati

 An autonomous vehicle, often known as a driverless automobile, is able to operate itself and perform important tasks without the need for human intervention thanks to its ability to understand its environment.

Self-driving cars utilize a variety of sensors to sense their surroundings, including thermographic cameras, radar, lidar, sonar, GPS, odometry, and inertial measurement units. Advanced control systems evaluate sensor data to identify appropriate travel routes, as well as impediments and necessary signage.

Customers anticipate the same level of interaction with their in-car entertainment system as they do with their smartphones, tablets, and gaming systems. The traditional infotainment system is also getting new driving aid functions.

The autonomous driving and advanced driver assistance systems (ADAS) industries are evolving, leading to more complex computation and sensing requirements.  [FPGAs have a particular advantage over traditional silicon technology](https://www.allaboutcircuits.com/news/why-the-industry-is-demanding-fpgas-for-adas/), which makes them ideal for handling the continually changing demands of the autonomous driving business.

The fact that FPGAs are programmable logic devices is the most important feature of their construction. Of course, the software on a CPU can be upgraded, but the semiconductor technology on a computer cannot. FPGAs, on the other hand, maybe changed or reprogrammed to perform new purposes endlessly. Because of this, FPGAs can keep OEMs up to date on the latest trends in programmable or configurable hardware architectural systems.

### ADAS Requirements

ADAS features to be implemented include:

* Collision Avoidance
* Lane Keeping Assistance
* Adaptive Cruise Control
* Traffic Sign Recognition
* Blind Spot Detection
* Pedestrian Detection
* Parking Assistance

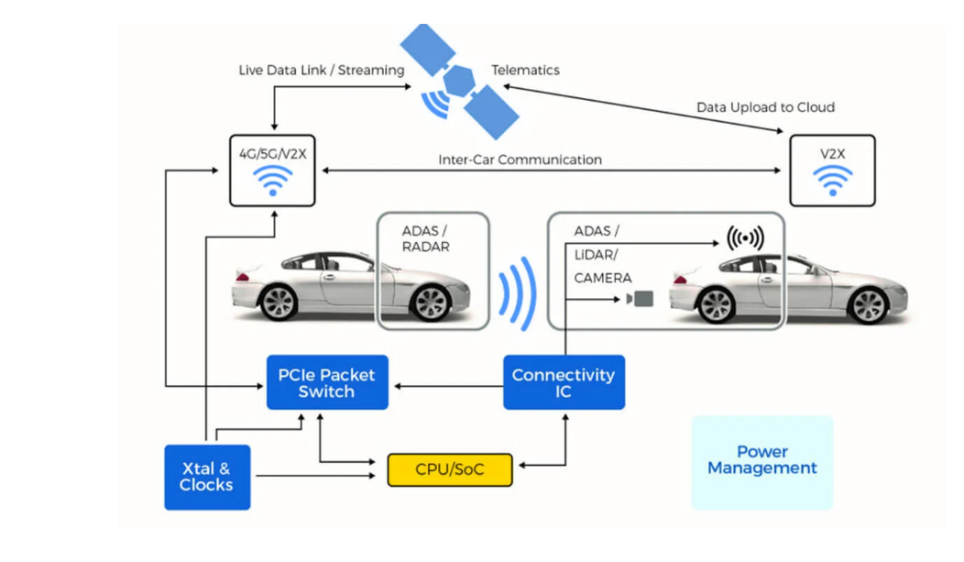
### Sensor Integration

### 1.1 Sensor Types and Data Rates

* **Cameras:** High-resolution, typically requiring bandwidths for 1080p or higher at 30-60 FPS.
* **Radar:** Medium to high frequency, providing range and velocity data.
* **Lidar:** High data rate for 3D mapping.
* **Ultrasonic Sensors:** Lower data rates, used for close-range object detection.

### 2. Data Processing Requirements

* **Image Processing:** High throughput for real-time video streams from multiple cameras.
* **Signal Processing:** Efficient algorithms for radar and lidar data.
* **Data Fusion:** Integration of sensor data to create a comprehensive environmental model.



### FPGA Advantages

* **Parallel Processing:** Ability to handle multiple data streams simultaneously.
* **Low Latency:** Essential for real-time safety applications.
* **Reconfigurability:** Flexible updates and optimizations.
* **Power Efficiency:** Critical for extending EV battery life.

### 1. FPGA Architecture Design

### 1.1 Sensor Interfaces

* **Cameras:** MIPI CSI-2 interfaces for high-speed video data.
* **Radar:** SPI or dedicated radar signal processing blocks.
* **Lidar:** High-speed serial interfaces.
* **Ultrasonic Sensors:** Standard interfaces like I2C or CAN.

### 1.2 Data Processing Blocks

* **Image Processing Pipeline:**
  + Image filtering
  + Feature extraction
  + Object detection and tracking
* **Signal Processing Blocks:**
  + FFTs for radar signal processing
  + Object detection algorithms
* **Data Fusion Engine:**
  + Combining sensor data for a unified environmental model

### 1.3 Control Logic

* **Decision-Making Algorithms:** Real-time response to sensor data.
* **Safety Mechanisms:** Redundancy and watchdog timers for reliable operation.

### 1.4 Communication Interfaces

* **High-Speed Communication:** Ethernet or PCIe for data exchange.
* **Vehicle Control Interfaces:** CAN bus for integration with vehicle control systems.

### 2. Software and Firmware

### 2.1 Driver Software

* Interfaces for sensor data acquisition.

### 2.2 Middleware

* Data processing and fusion management.

### 2.3 Application Software

* ADAS algorithms and decision-making logic.

### 2.4 Update Mechanisms

* Secure methods for FPGA configuration and software updates.

### 3. Deployment and Maintenance

### 3.1 Integration

* Seamless integration with vehicle systems.

### 3.2 Monitoring

* Real-time system monitoring and diagnostics.

### 3.3 Updates

* Over-the-air updates for improved functionality and issue resolution.

Developing an FPGA architecture for an ADAS solution in an EV involves integrating various sensors, processing large amounts of data in real-time, and ensuring the system is power-efficient, reliable, and updatable. By following the outlined steps and considering the detailed requirements, an efficient and effective ADAS system can be implemented to enhance the safety and functionality of electric vehicles.

References:

1)<https://fpgainsights.com/fpga/fpga-for-automotive-electronics-and-advanced-driver-assistance-systems-adas/>

2) <https://china.xilinx.com/publications/presentations/c_D2-01.pdf#:~:text=URL%3A%20https%3A%2F%2Fchina.xilinx.com%2Fpublications%2Fpresentations%2Fc_D2>

3) <https://www.digikey.com/en/articles/the-selection-and-use-of-fpgas-for-automotive-interfacing>