InterIIT Internal Hackathon - JLR Problem Statement (12.0)

Shankha Deep Bhattacharyya & Arthreya Krishnakumar

October 2024

Contents

1 Introduction

The architecture of Modern Day Electric vehicles is vastly different from one 20 years ago. From the most basic EVs running with bare minimum electronics, to self driving cars, the technology has progressed tremendously in the past few decades. A modern day EV has hundreds of electronic Control units (ECUs) interconnected via kilometers of wires, all in increasingly smaller and smaller spaces. This is accompanied by an rapid growth in performance and exponential growth of other features in our cars. Cars have gone from an object of utility to a symbol of luxury. The latest and greatest features of today make them vastly different beasts even compared to cars from yesteryear. These changes have been accompanied by several new challenges, that require newer solutions to keep up with the pace of innovation.

2 Evolution of E/E Architecture in EVs

The Electrical/Electronic (E/E) architecture in EVs form the base of a modern EV. It has undergone significant changes as the demands for performance, flexibility, and scalability have increased. Over the past decade, automakers have transitioned from simpler, more fragmented designs to advanced, centralized architectures that enable modern features like over-the-air (OTA) updates, real-time data processing, and autonomous driving capabilities. Such changes also necessitate changes in the overall system architecture.

2.1 Distributed Architecture

In the early stages of EV design, manufacturers relied on a distributed or federated architecture, where each function had its own dedicated electronic control unit (ECU).

Key Features:

- **Dedicated ECUs for each function:** Each vehicle function (powertrain, braking, battery management, climate control, infotainment) was managed by a separate ECU.
- **Communication** These ECUs communicated via CAN (Controller Area Network), a widely used automotive communication protocol for low-latency control.

Challenges:

- **Wiring Complexity:** As more features were added to EVs, the number of ECUs increased, leading to a very complex wiring harness. This made assembly more difficult and raised manufacturing costs.
- **Scalability:** Adding new features meant installing additional ECUs, which was costly and inefficient. The system lacked flexibility for future updates.
- **Weight and Cost:** The large number of ECUs and the wiring required added extra weight to the vehicle, negatively impacting efficiency. Moreover, the cost of materials and assembly increased with the rising complexity of the system.

2.2 Domain Centralized Architecture

As EV technology advanced, manufacturers began to recognize that having too many independent ECUs was impractical. This led to the modular or domain-based E/E architecture, where similar functions were grouped together under domain controllers.

Key Features:

- **Domain Controllers:** ECUs were grouped into domains based on function (e.g., body, powertrain, infotainment), each with its own controller. This reduced the total number of ECUs by integrating multiple functions into one domain. Grouping similar functions under domain controllers reduced the complexity of the wiring harness and the number of individual ECUs.
- **Gateway Modules:** A gateway was introduced to manage communication between domains, facilitating more efficient data exchange between different parts of the vehicle.
- **Newer Communication Protocols** The system still primarily relied on CAN, though higher data rate protocols like LIN and FlexRay began to appear for specific use cases.

- **Better Integration:** Functions like regenerative braking, which involves coordination between the braking system and the powertrain, could now be better integrated within a domain.

Challenges:

- **Fragmented Communication:** Despite the reduction in ECUs, domains were still largely siloed, meaning that the data flow between domains was not seamless. Information had to be passed through gateways, which introduced latency and complexity.
- **Limited Flexibility:** While an improvement over the distributed architecture, modular architectures still required physical changes to hardware to add new features or update existing ones, which limited their scalability.

2.3 Centralized E/E Architecture

As the complexity of EV functions continues to increase, particularly with the rise of advanced driver-assistance systems (ADAS) and autonomous driving features, the industry is moving toward centralized E/E architectures. This phase marks a significant departure from the domain-based system, consolidating many functions into a smaller number of high-performance domain controllers.

Key Features:

- **Powerful Domain Controllers:** Domain controllers become more powerful and capable of handling multiple functions, such as battery management, powertrain control, and infotainment, within a single controller.
- **Ethernet-Based Communication:** To handle the growing data rates and lower latency needs, high-speed Ethernet has begun to replace traditional CAN as the communication backbone. Ethernet allows much faster data transmission and is more suitable for advanced functions like ADAS.
- **Software Flexibility:** One of the key innovations is the introduction of over-the-air (OTA) updates, which allow manufacturers to update vehicle software remotely. This provides the ability to fix bugs, improve performance, or introduce new features without physical changes to the hardware.
- **Scalability:** OTA updates made it much easier to add new features or improve existing ones through software changes, providing a more scalable architecture.

Challenges:

- **High Interconnect Complexity:** The interconnects between sensors, actuators, and centralized controllers have become more complex, particularly in systems requiring real-time responses.
- **Thermal Management:** As more computing power is consolidated into a few controllers, managing heat generation within these components becomes more challenging.

2.4 Service-Oriented Architecture (SOA): Zonal and Software-Defined Vehicles

The latest and most advanced development in E/E architecture is the shift toward service-oriented architecture (SOA), which places even greater emphasis on software-defined vehicles. In this model, most vehicle functions are managed by zonal controllers and centralized compute platforms, with a strong focus on software as the primary driver of functionality.

Key Features:

- **Zonal Controllers:** Instead of organizing the architecture around functions, zonal controllers manage all electrical components within a specific physical region of the vehicle (e.g., front left, rear right). This further reduces the number of ECUs and wiring.
- **Centralized Compute Platforms:** A few powerful central computers manage critical functions like ADAS, battery management, and infotainment. These platforms integrate sensor data (cameras, LiDAR, radar) and process it in real-time, enabling advanced functions like autonomous driving.
- **Ethernet Backbone:** High-speed Ethernet is now the standard for communication between zonal controllers and the central compute platforms. This ensures the low latency and high bandwidth needed for processing large data streams.

- **Service-Oriented Software:** In an SOA, vehicle functions are treated as software services. This means they can be updated, modified, or added without affecting other parts of the system. Software modules are isolated and abstracted, making it easier to roll out new features or updates over time.

Challenges:

- **Software Complexity:** With more reliance on software, the complexity of managing vehicle functions increases. Ensuring that critical functions like braking or steering operate reliably under all conditions requires significant software engineering expertise.
- **Cybersecurity:** As vehicles become more connected, particularly with cloud-based services and OTA updates, they become more vulnerable to cyberattacks. Robust cybersecurity measures are necessary to protect vehicle systems and data.

3 Role of Chiplets

As the E/E architecture of EVs involve, so do their computational needs. Newer functions such as autonomous driving and complex infotainment systems require huge amounts of computational power. And this requirement increases year-over-year. OEMs have to package more and more powerful hardware into the limited space available inside the car every year to keep up with their latest features.

The enormous scalability offered by a chiplet-based computer architecture provides the perfect solution for this growing problem.

3.1 What is a Chiplet-Based Architecture?

Chiplet-based architectures represent a significant shift in the design and manufacturing of processors and integrated circuits. Instead of creating a single monolithic chip, chiplets are smaller, modular units that are interconnected to function as a cohesive system on a chip (SoC). This modular design approach allows chip manufacturers to mix and match different components to build complex processors, leading to several advantages in performance, cost, and flexibility.

Key Components of Chiplet-Based Architectures

1. **Chiplets**
   - **Definition:** Chiplets are smaller, semi-independent functional units that perform specific tasks within a larger system. They are typically specialized for particular functions like memory, processing, graphics, or interconnectivity.
   - **Heterogeneous Integration:** Different types of chiplets can be combined in one package, allowing for a heterogeneous architecture, where components are optimized for their specific function (e.g., combining CPU, GPU, and memory chiplets in a single package).
2. **Interconnect Technologies**
   - **Definition:** Chiplets need to communicate effectively, so high-bandwidth interconnects like Advanced Interface Bus (AIB), UCIe (Universal Chiplet Interconnect Express), or Infinity Fabric (from AMD) are used to enable efficient communication between chiplets. These interconnects help achieve performance close to that of monolithic chips while allowing the flexibility of modularity.
   - **3D Packaging:** Chiplet architectures can also utilize 3D stacking, where multiple layers of chiplets are stacked on top of each other, significantly improving performance and reducing space requirements by shortening interconnect distances.
3. **Manufacturing Benefits**
   - **Yields:** In traditional monolithic chip designs, a single defect can render an entire chip unusable. By using chiplets, manufacturers can improve manufacturing yields, as smaller chiplets are easier to produce with fewer defects, lowering production costs.
   - **Customization:** Chiplet architectures allow manufacturers to reuse certain chiplets across multiple products, providing customization while reducing the need for developing entirely new chips for every product variation. For example, the same CPU chiplet can be paired with different memory or GPU chiplets for varying product tiers.

Advantages of Chiplet-Based Architectures

1. **Scalability and Modularity**
    - **Scalability:** Chiplets can be used to very easily scale up systems. Traditional methods usually involve the ground up design of new components in order to increase their capacity. In a chiplet based architecture, the capacity or throughput can be increased by simply increasing the number of chiplets integrated using the interconnect. It is no longer required to design the entire system right from the beginning.
    - **Modular Design:** Manufacturers can design systems by combining different chiplets tailored for specific applications. This provides tremendous flexibility, enabling scalability based on performance needs. High-performance systems can be built by integrating more powerful chiplets, while cost-effective systems can use simpler or fewer chiplets.
    - **Customization:** Chiplets allow for greater product differentiation without redesigning an entire chip. A single design can support a wide range of devices, from high-performance computing to mobile platforms.
2. **Reduced Costs**
    - **Economies of Scale:** Since chiplets are smaller and simpler to manufacture, they are often less expensive than creating large monolithic chips. Additionally, manufacturers can reuse the same chiplet across different products, spreading development costs over a broader range of applications.
    - **Improved Yield:** Smaller chips tend to have fewer defects, so the yield (the number of usable chips from a wafer) is higher. This reduces waste and overall manufacturing costs.
3. **Faster Innovation Cycles**
    - With chiplet-based designs, manufacturers can update specific components (e.g., upgrading only the CPU chiplet for better performance) without redesigning the entire chip. This modular upgrade approach leads to shorter design and production cycles, allowing for quicker market responsiveness.
4. **Power Efficiency**
    - Chiplet architectures can optimize power consumption by allocating specialized tasks to more energy-efficient chiplets. For example, a power-efficient chiplet could handle routine functions, while more power-hungry chiplets are only used when high performance is required.
5. **Design Flexibility for Different Technologies**
    - Chiplet architectures allow the integration of different technologies (e.g., silicon, photonics, or memory technologies) in one package. This means that specialized technologies can be combined into one design, enhancing performance and enabling new types of computing capabilities.

4 Proposed Applications

As evident from the above discussion, the application of chiplets would have a significant positive impact on the centralized and service-oriented architectures of EVs. However, such technology has not yet been implemented by the industry. Therefore, we propose **three places of application** of a chiplet-based architecture in an EV where it would provide the best results. These are:

1. Advanced Driver Assistance Systems (ADAS)
2. Integrated Powertrain and Body Control Unit
3. Infotaiment & Communications Unit

4.1 Advanced Driver Assistance Systems (ADAS)

An Advanced Driver Assistance System (ADAS) is a collection of technologies designed to enhance vehicle safety and improve driving experience by assisting the driver in various functions. ADAS uses sensors, cameras, radar, and LiDAR to monitor the vehicle's surroundings and provide real-time data to onboard computers. These computers, depending on the level of automation, make the vehicle perform certain actions. According to the Society of Automotive Engineers (SAE), the levels of automation in autonomous driving systems are:

1. **Level 0: No Automation**: Driver has full control; no automated systems.
2. **Level 1: Driver Assistance**: Driver controls most functions; one system assists with steering or speed.
3. **Level 2: Partial Automation**: Vehicle controls steering and speed; driver must stay engaged and monitor the environment.
4. **Level 3: Conditional Automation**: Vehicle handles all tasks in certain conditions; driver must be ready to intervene.
5. **Level 4: High Automation**: Vehicle operates autonomously in specific conditions; no driver intervention required within those conditions.
6. **Level 5: Full Automation**: Vehicle is fully autonomous in all conditions; no driver involvement needed.

The maximum level of automation currently achieved is level 4 automation. Hence, our design shall implement that level of automation.

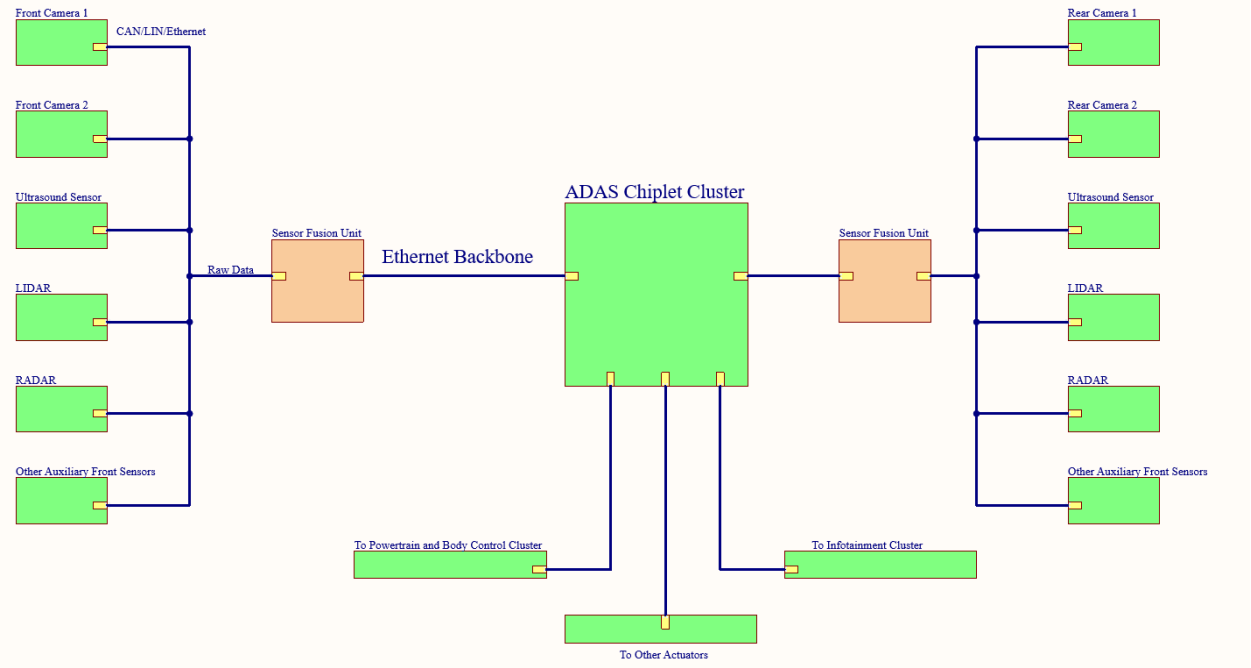

Figure 1: Overall System Architecture of ADAS

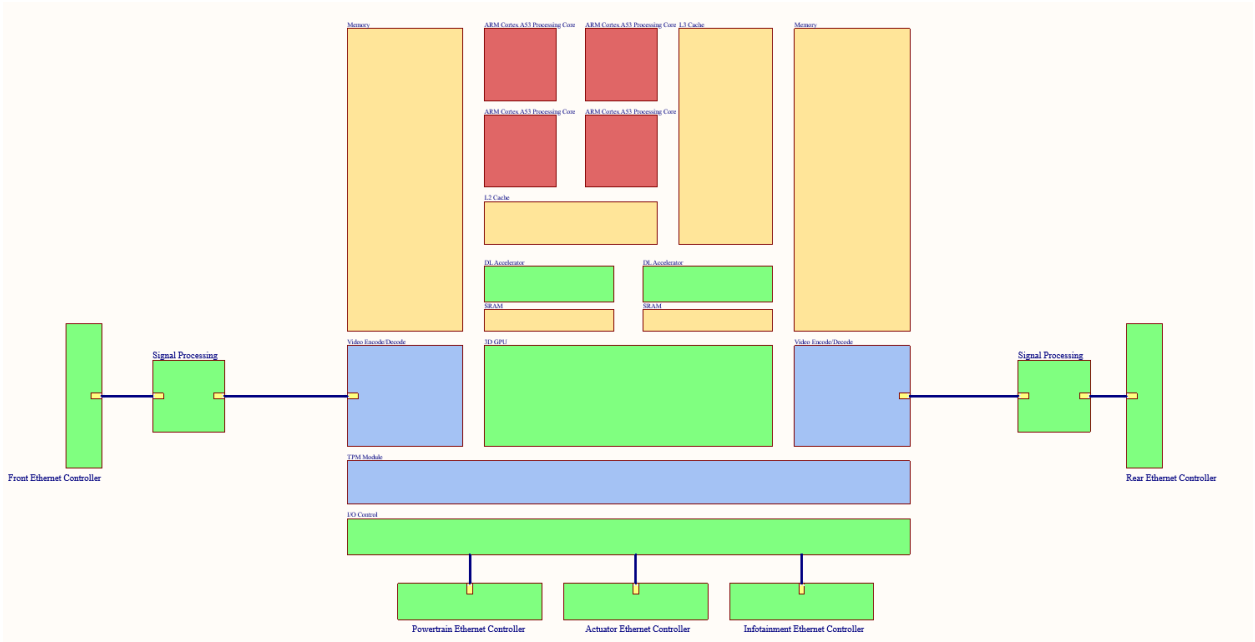

Figure 2: ADAS Chiplet Cluster Internal Structure

4.2 Powertrain and Body Control Unit

This cluster controls all major EV powertrain and body control functions. It integrates several individual ECUs found in a traditional EV. The compute resources are pooled together into a powerful compute cluster, and each ECU function is run as different State Machines via Virtualization. The processing cluster is powerful enough to handle the computation of all the ECUs that are integrated into this. By integrating the ECUs, the need for having multiple interfacing units to communicate between them is eliminated, and so is the latency.

The computations done by the individual ECUs are now offloaded to the chiplet-based processing cluster shown in the diagram. Hence, these ECUs are now essentially software defined, rather than a hardware unit. The outputs generated from these computations from each (now software-defined) system is stored in the shared memory unit. Any other system that needs the outputs of the computation or the data collected by the sensors, can access it from the shared memory.
The functions performed by the Powertrain and Body Control Unit include:

- Determination of PWM signals for the motor inverter, based on user input from throttle pedal.
- Estimation of Battery SoC. SoH and temperature from the data provided by the slave BMS.
- Control of charging of the battery pack.
- Control of thermal management system pumps, ensuring proper cooling wherever required.
- Ensures stability of vehicle in dynamic driving conditions, and alerts the driver to any potential hazards.
- Provides inputs to the required servomotors for power steering assistance.
- Performs ABS control computations.
- Monitors the data obtained from the TPMS to ensure the safety of the vehicle.
- Controls the HVAC system of the car.
- Controls body functions such as automatic windshield wipers, etc.

In order to streamline data collection and transmission, we perform data-level sensor fusion for sensors that are physically close together, combining their raw data. This is then sent to the Powertrain and Body Control Unit via a Ethernet. The Controller receives the data, and the appropriate sensor data gets stored in the shared memory, from where all systems that require the data can access it. A more detailed explanation of the sensor fusion technique has been given later under the ADS subsystem section.

For sensors that transmit non-critical data, a separate interface module has been employed. The memory module shown is shared by all the subsystems of the controller, ensuring any data collected by one is available for utilisation by the other. For the transmission of data, the same Ethernet buses that were used by the sensors are used to transmit the required control signals to the appropriate actuators.

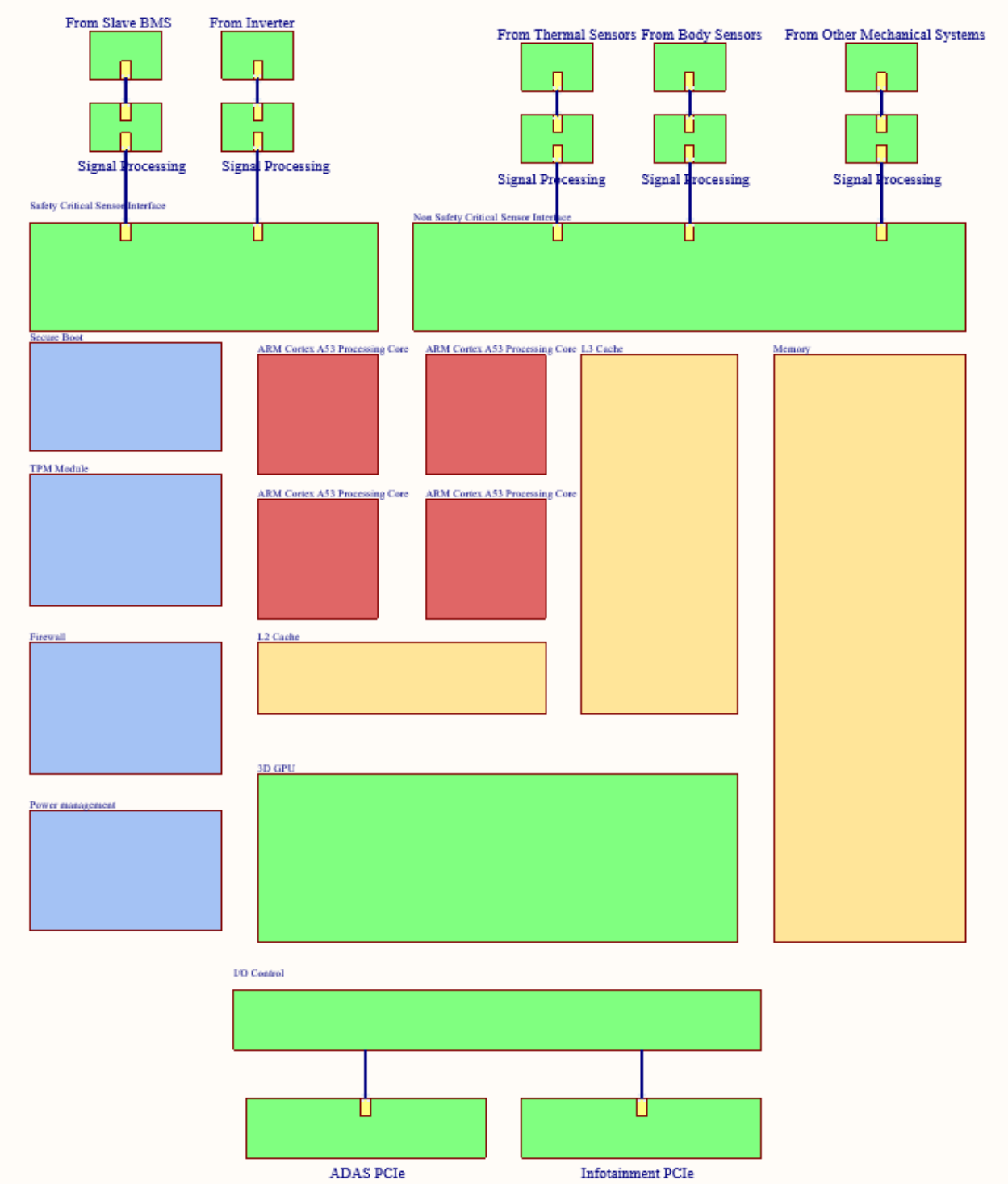

Figure 3: Powertrain and Body Control Unit

4.3 Infotainment and Communications Unit

This subsystem integrates the infotainment ECU and all other ECUs involved with data logging and telemetry. It's main purpose is to perform all Infotainment functions along with all V2X functions required by the car. It interacts with the user interface and conveys all the information that needs to be passed on to the driver and other passengers. It integrates various connectivity functions, such as WiFi, Bluetooth, etc, for

entertainment systems and also supports smart connection technologies such as IoT integration and V2X communication.

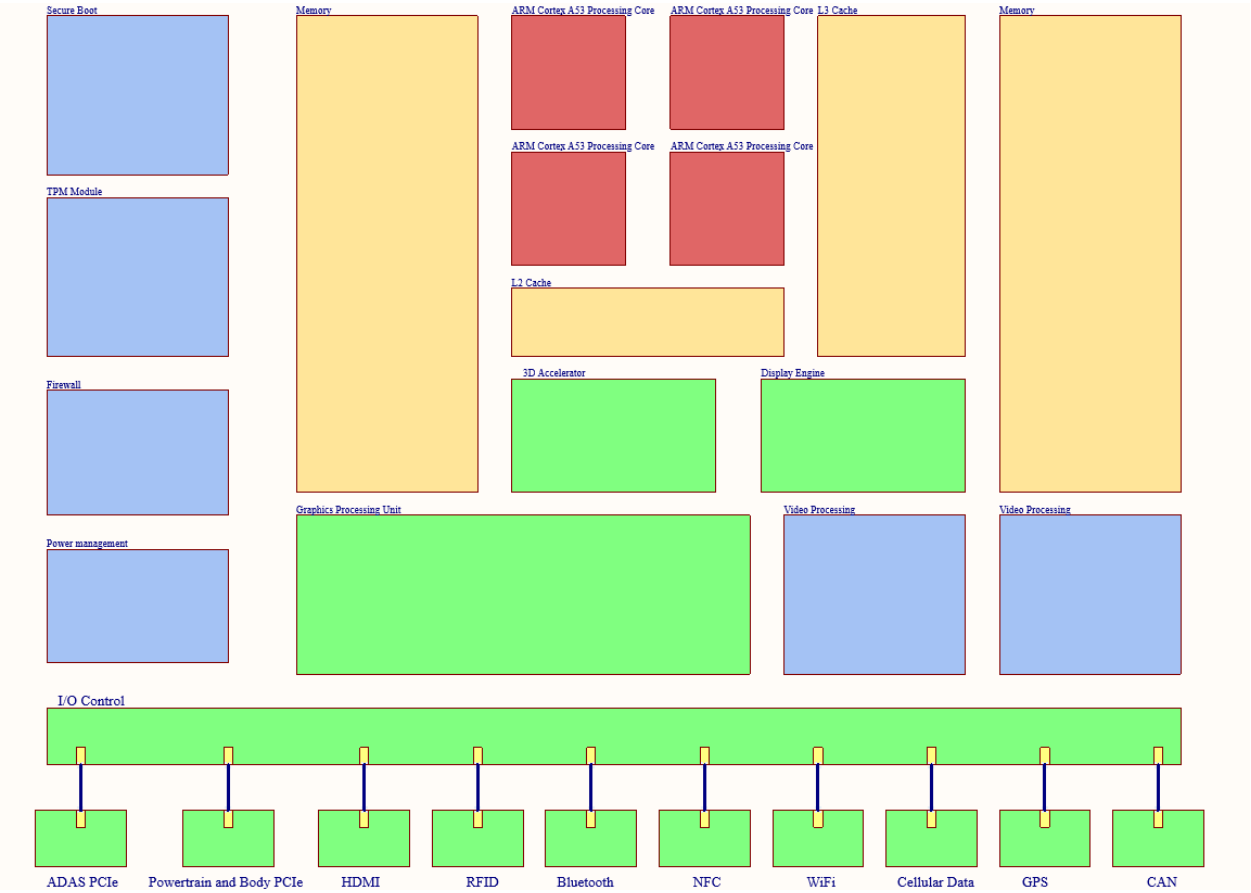

Figure 4: Infotainment and Communications Unit

Several wired interfaces such as HDMI, USB, and Ethernet connect with various infotainment instruments such as touchscreen displays. It is used to convey all critical data to the driver along with other features such as playing the radio and multimedia content, either from the internet or from a USB media device, etc.

Other services such as smart fleet & traffic management, Over-the-Air Software updates, Health monitoring of the Car, and IoT integration services require V2X (Vehicle to Everything) communication. This is made possible by the various wireless modules present such as WiFI, Bluetooth, Cellular Data etc

Vehicle-to-Everything (V2X) connectivity is a vital component of Intelligent Transportation Systems (ITS), fostering communication between vehicles, infrastructure, and various elements within the transportation ecosystem. This comprehensive framework encompasses several types of communication. Vehicle-to-Vehicle (V2V) communication enables direct information exchange between vehicles, enhancing road safety through collision avoidance and cooperative adaptive cruise control. Vehicle-to-Infrastructure (V2I) or Vehicle-to-Roadside (V2R) communication involves interaction with elements like traffic lights and road signs, optimizing traffic signals and providing real-time information on road conditions. Vehicle-to-Cloud (V2C) communication extends connectivity to the cloud, supporting services like over-the-air updates and remote diagnostics. Vehicle-to-Infrastructure/Internet of Things (V2IoT) communication broadens connectivity to include interactions with IoT devices for smart city applications.

These communication types leverage various wireless technologies. Dedicated Short-Range Communication (DSRC)/Vehicular Ad Hoc Networks (VANETs) ensures low-latency, secure communication for V2V and V2I scenarios. Cellular networks, such as LTE and 5G, offer broader coverage and bandwidth for V2C communication and cloud-based services. Zigbee is employed for low-speed communication in V2C, ADAS, and Forward Collision Alert Systems. Ultrawideband (UWB) supports high-speed data transfer for accident avoidance and precise vehicle positioning. Wi-Fi, WiMAX, Radio Frequency Identification (RFID), and Bluetooth are utilized for specific communication needs within the V2X ecosystem.

In essence, V2X connectivity enhances safety, efficiency, and intelligence in transportation by facilitating seamless communication through various wireless technologies, paving the way for a more connected and automated future in transportation systems. The V2X communication in the car is enabled by the various wireless modules such as WiFi, Bluetooth, RFID, etc, as shown in the figure.

To ensure secure data transfer between the car and the cloud, we have connected all the incoming and outgoing signals between the Processing Cluster and the peripheral interfaces through a Hardware Firewall. It monitors the incoming and outgoing network traffic and allows only those information packets that are parts of established and legitimate connections. They also employ other techniques, such as Network Address Translation (NAT) and Virtual Private Networks (VPN), making it more difficult for other unwanted entities to gain access to and tampering with sensitive information.

The Infotainment and Connectivity Cluster often times needs to share data and instructions with the Integrated VCU. Data is shared via the shared memory module accessible to all 3 subsystems. For passing instructions, we use 2 interrupt controllers, one to and one from the Integrated VCU subsystem. This allows exchange of instructions as required by the situation.

5 Communication Technologies in Semiconductor Industries

The communication segment within automotive semiconductors is crucial for enabling various functions, from basic sensor integration to advanced driver assistance systems (ADAS). Leading protocols include I2C (Inter-Integrated Circuit) and SPI (Serial Peripheral Interface), commonly used for short-distance communication between microcontrollers and peripheral devices. CAN (Controller Area Network) facilitates robust vehicle networking, supporting real-time data transmission among different electronic control units (ECUs). LIN (Local Interconnect Network) serves lower-bandwidth applications, such as interior lighting and seat control, while FlexRay offers higher bandwidth and deterministic data transmission for safety-critical applications. Ethernet, increasingly adopted in automotive systems, provides high-speed communication for data-intensive applications, enabling the integration of advanced features like infotainment and vehicle-to-everything (V2X) communication. UART (Universal Asynchronous Receiver-Transmitter) and serial protocols are essential for simpler point-to-point communications.

As the automotive landscape evolves, technologies like Vehicle Ethernet are gaining traction, promising faster data rates and improved interoperability among devices. The shift towards Software-Defined Vehicles (SDV) emphasizes the need for flexible communication architectures that can accommodate over-the-air (OTA) updates and support advanced functionalities. Looking ahead, the development of 5G communication and V2X protocols will further enhance vehicle connectivity, enabling seamless communication between vehicles and infrastructure, ultimately leading to safer and more efficient transportation systems.

5.1 Scenerio for the Communication at Semiconductor Level over past decade

1. **I2C (Inter-Integrated Circuit)**
   Need and Implementation: I2C is widely used for connecting low-speed peripherals to microcontrollers, such as sensors and EEPROMs.
   Details: I2C supports a bandwidth of up to 3.4 Mbps in High-Speed mode and has low latency that varies with bus speed. It can accommodate multiple devices, allowing up to 127 on a single two-wire bus (SDA, SCL). While it offers moderate robustness, it can be susceptible to noise over long distances. The protocol includes an ACK/NACK bit for error checking and is generally reliable for short-distance communication.

2. **SPI (Serial Peripheral Interface)**
   Need and Implementation: SPI is employed for high-speed communication between microcontrollers and peripherals, making it ideal for applications requiring fast data transfer.
   Details: With a bandwidth of up to 50 Mbps or more, SPI features low latency and typically operates in a point-to-point configuration with one master and one or more slaves. It utilizes a four-wire interface (MOSI, MISO, SCK, SS) and is highly robust due to its lower susceptibility to noise. However, it lacks built-in error checking, which may necessitate additional mechanisms. Overall, SPI is very reliable for short-distance applications.

3. **CAN (Controller Area Network)**
   Need and Implementation: CAN is a robust vehicle bus standard designed to facilitate communication among electronic control units (ECUs) without the need for a host computer.
   Details: The bandwidth of CAN reaches 1 Mbps for Classical CAN and up to 8 Mbps for CAN FD. It

supports low to moderate latency and can accommodate up to 1,024 nodes in a multi-master configuration. Known for its high robustness against noise, CAN includes various error-checking mechanisms such as bit monitoring, CRC, and acknowledgment. This makes it highly reliable for safety-critical applications.

4. **LIN (Local Interconnect Network)**
   Need and Implementation: LIN is a low-cost, low-speed communication protocol used in automotive applications for connecting simple devices.
   Details: LIN supports a bandwidth of up to 20 Kbps with moderate latency and can connect up to 16 nodes in a single-master configuration with multiple slaves. Its robustness is moderate, making it suitable for non-critical applications. Basic error detection mechanisms are employed to ensure reliability in less critical scenarios.

5. **FlexRay**
   Need and Implementation: FlexRay is utilized for high-speed and fault-tolerant communication in safety-critical applications such as Advanced Driver Assistance Systems (ADAS).
   Details: FlexRay offers a bandwidth of up to 10 Mbps with deterministic latency that guarantees response times. It can support up to 64 nodes through dual-channel redundancy, enhancing its robustness significantly for automotive environments. Advanced error detection and correction mechanisms contribute to its extreme reliability in safety-critical systems.

6. **Ethernet**
   Need and Implementation: Ethernet is increasingly used in automotive systems for high-speed data transmission, supporting advanced applications like infotainment and vehicle-to-everything (V2X) communication.
   Details: It provides bandwidth ranging from 100 Mbps to over 10 Gbps with low latency and highly scalable channel capacity. Ethernet can be configured in various ways (switched or point-to-point) and adheres to high standards defined for automotive environments. Error checking is performed using CRC checks, making it very reliable, especially in high-bandwidth applications.

7. **UART (Universal Asynchronous Receiver-Transmitter)**
   Need and Implementation: UART is commonly used for asynchronous serial communication between microcontrollers and peripherals.
   Details: Typically supporting bandwidths of up to 1 Mbps, UART operates with low latency in a point-to-point configuration using single-ended signal lines. Its robustness is moderate due to distance limitations; however, it employs parity bits and framing checks for error checking. Overall, UART remains reliable for simple communication tasks.

CAN (Controller Area Network) is currently the dominant protocol in automotive systems due to its robustness and suitability for moderate data transmission needs. However, with the increasing demand for higher data rates and lower latency, especially in advanced applications like ADAS and infotainment, there is a shift toward Ethernet and FlexRay. These newer technologies offer higher bandwidth and deterministic communication, making them better suited for the growing complexity of modern vehicles.

5.2 Communication technologies utilised in Chiplet

1. **Interlaken**
   Interlaken is a high-speed chip-to-chip interface optimized for efficient data transfer between chiplets. It supports multiple lanes and high data rates, making it suitable for modern applications that require rapid communication between components. The protocol is designed to minimize latency while maximizing bandwidth, achieving speeds of up to 2.6 Tbps. Interlaken employs a scalable architecture that allows for configurable bandwidth and supports up to 2048 logical channels and 48 lanes, facilitating its use in diverse applications from edge computing to cloud environments.

   - **Efficiency:** Interlaken enhances efficiency through mechanisms such as reduced pin count by utilizing serial communication rather than parallel interfaces, significantly lowering the number of physical connections required on a printed circuit board (PCB). This not only reduces manufacturing costs but also minimizes PCB design complexity. Additionally, the protocol is optimized for energy-efficient data transfer, reducing power consumption per bit by maximizing payload efficiency.

- **Reliability:** Reliability is paramount in chip-to-chip communications, and Interlaken addresses this through advanced error detection mechanisms such as Reed-Solomon Forward Error Correction (FEC) and flow control on individual data channels. These features ensure data integrity and allow for error recovery without application-layer intervention. The protocol also utilizes a 64b/67b encoding scheme that reduces running disparity and enhances the reliability of data transmission over long distances.
- **Scalability:** Scalability is a fundamental characteristic of Interlaken. The architecture allows designers to configure the number of lanes and their respective speeds according to application requirements, enabling reuse across different designs. Furthermore, Interlaken can efficiently manage communications among multiple chiplets within a single package or across separate dies.

2. **Coherent Interconnects**
Coherent interconnects such as CCIX (Cache Coherent Interconnect for Accelerators) and CXL (Compute Express Link) are vital for systems requiring memory coherence among chiplets. These technologies enable shared memory access across different chiplets, facilitating efficient data sharing and synchronization.

   - **Efficiency:** Coherent interconnects optimize memory access patterns by allowing multiple chiplets to operate on a consistent view of memory. This reduces overhead associated with data duplication and improves overall system throughput.
   - **Reliability:** By ensuring memory coherence, these interconnects enhance reliability in multi-chiplet systems by preventing data inconsistencies that can arise from concurrent accesses by different processing units.
   - **Scalability:** Coherent interconnects are designed to scale with the increasing complexity of multi-chiplet architectures. They support a growing number of chiplets while maintaining performance levels necessary for high-demand applications such as AI and machine learning.

3. **Die-to-Die Interfaces**
Technologies like UCIe (Universal Chiplet Interconnect Express) focus on enabling communication between various chiplets regardless of their manufacturing processes. This enhances interoperability among different chiplets from various vendors.

   - **Efficiency:** Die-to-die interfaces reduce the need for extensive routing on PCBs by allowing direct connections between chiplets, leading to shorter signal paths and lower latency in communications.
   - **Reliability:** These interfaces are designed with robust signaling technologies that ensure reliable data transmission across varying manufacturing processes and environmental conditions.
   - **Scalability:** Die-to-die interfaces facilitate the integration of diverse functionalities within a single package, allowing designers to scale systems up or down based on specific application requirements without compromising performance.

5.3 Overview of Key IP Blocks in Chiplet Communication

1. **Serializer/Deserializer (SerDes)**
Function: Converts parallel data to serial form for transmission, and vice versa, ensuring high-speed communication.
Details:

   - **Speed:** Supports multi-gigabit data rates, such as 10 Gbps to 112 Gbps.
   - **Use Case:** Employed in high-speed interfaces like PCIe, Ethernet, and chiplet data links.
   - **Key Features:** Low power consumption, efficient serialization, and support for encoding schemes like 8b/10b.

2. **Protocol Controllers**
Function: Manages communication protocols like PCIe, CXL, and CCIX to handle high-speed data transfer and ensure compliance with standards.
Details:

   - **PCIe Controller:** Enables high-speed data transfer with support for up to 64 GT/s in PCIe 5.0 or newer.
   - **CXL Controller:** Provides low-latency memory sharing and interconnects for data center workloads.

- **CCIX Controller:** Ensures cache coherence between heterogeneous processors (e.g., CPU and FPGA/GPU).
- **Key Features:** High-speed data transfer, error detection, flow control, and power management.

3. **Error Correction Codes (ECC)**
   Function: Ensures data integrity by detecting and correcting errors during data transmission or storage.
   Details:
   - **Common Algorithms:** Includes Hamming Code, Reed-Solomon, and LDPC.
   - **Use Case:** Used in memory systems, networking, and data transmission over error-prone links.
   - **Key Features:** Single or multi-bit error correction, low latency, and improved data reliability.
4. **Switch/Router IPs**
   Function: Facilitates data routing between chiplets, providing scalable and efficient communication across modules.
   Details:
   - **Network-on-Chip (NoC):** Implements packet-based data transmission between IP blocks or chiplets.
   - **Use Case:** Used in multi-chip modules (MCM) and SoCs for managing large-scale inter-chip communication.
   - **Key Features:** Low-latency routing, support for Quality of Service (QoS), and fault tolerance.
5. **Clock and Data Recovery (CDR)**
   Function: Recovers the clock signal from the incoming data stream, ensuring synchronized communication.
   Details:
   - **Use Case:** Employed in high-speed serial data links such as PCIe and SerDes interfaces.
   - **Key Features:** High-speed operation with low jitter, phase-locked loops (PLLs), and synchronization at multi-gigabit data rates.
6. **Memory Controllers**
   Function: Manages memory access, coordinating read/write operations to ensure efficient sharing of memory across chiplets.
   Details:
   - **DRAM Controller:** Handles external DRAM access with optimized latency and power management.
   - **SRAM/Cache Controllers:** Manages on-chip cache or SRAM for low-latency data access.
   - **Key Features:** Memory coherence protocols (like MESI), ECC for error correction, and memory arbitration to maximize bandwidth efficiency.

6 Optical Interconnects for Die-to-Die Communication in Automotive Applications

1. Introduction to Optical Interconnects

Optical interconnects utilize light signals, typically in the infrared spectrum, to transmit data between components. In the context of die-to-die communication for chiplets, optical interconnects can replace electrical traces with waveguides, silicon photonics, and optical fibers, offering higher data transmission rates and enhanced energy efficiency.

Advantages of Optical Interconnects:

- **High Bandwidth:** Optical links can transmit data at speeds up to the terabit-per-second (Tbps) range, significantly surpassing the capacity of electrical interconnects.
- **Low Latency:** The speed of light enables optical interconnects to minimize propagation delays.
- **Low Power Consumption:** Optical signals transmit more data per unit of power, which is essential for automotive systems constrained by thermal limits.

- **Immunity to Electromagnetic Interference (EMI):** In automotive environments, high EMI levels necessitate robust communication systems. Optical interconnects are inherently immune to such interference.

2. Relevance to Automotive Applications

Automotive systems, particularly for autonomous driving and advanced driver-assistance systems (ADAS), require high data rates, real-time processing, and reliable communication between various components.

Key Challenges in Automotive Systems:

- **Operating Conditions:** Automotive systems face wide temperature ranges, vibrations, and harsh environmental conditions.
- **Latency Requirements:** Real-time systems like ADAS require low-latency communication between chiplets.
- **Scalability:** Optical interconnects provide scalable bandwidth to accommodate the increasing complexity of automotive electronics.

6.1 Performance Data for Optical Interconnects

- Latency Optical interconnects can reduce latency to sub-nanosecond levels. For instance, the propagation delay for optical signals in a waveguide is approximately 5 ps/mm, compared to 50–100 ps/mm for electrical signals. Studies show a 65% reduction in latency for die-to-die communication across 1 cm when using optical interconnects over copper.
- Bandwidth and Communication Efficiency Optical interconnects support terabit-per-second (Tbps) data rates. Wavelength Division Multiplexing (WDM) allows multiple data channels to transmit over a single waveguide, enabling communication rates exceeding 1 Tbps. Experimental results show that a silicon photonic interconnect with four wavelengths achieved 400 Gbps total bandwidth, with 100 Gbps per wavelength Advanced modulation formats like Quadrature Amplitude Modulation (QAM) can further increase spectral efficiency.
- Energy Efficiency

  Optical interconnects consume less power than electrical interconnects, with energy efficiency often below 1 pJ/bit, compared to 5–10 pJ/bit for electrical systems. Laser-based optical interconnects consume less power over longer distances, avoiding the need for repeaters or amplifiers. Optical interconnects in short-distance communication (1–5 mm) can achieve energy per bit as low as 0.2 pJ/bit.
- Robustness and Reliability

  Optical interconnects are resistant to EMI, which is crucial in automotive environments where electrical noise is prevalent. Silicon photonics has shown resilience across temperature ranges from -40°C to +125°C, suitable for automotive conditions. Error correction schemes, such as Forward Error Correction (FEC), can ensure reliable data transmission with bit-error rates as low as 10^{-15}.

4. Challenges and Solutions for Optical Interconnects in Automotive Applications

a. Thermal Management

Effective thermal management is essential to maintain wavelength stability and laser efficiency. Integrated micro-cooling systems and thermal isolation techniques are being developed to ensure stable operation in high-temperature environments.

b. Packaging and Integration

Integrating optical interconnects into chiplet architectures requires advanced packaging techniques like 2.5D interposer technology or 3D stacking. Efficient light coupling between optical and electronic components can be achieved through vertical cavity surface-emitting lasers (VCSELs) or grating couplers.

c. Cost

Although optical interconnect technology is currently more expensive than electrical interconnects, advances in silicon photonics manufacturing are expected to reduce costs significantly over time. The increasing demand for advanced automotive systems, especially autonomous vehicles and electric vehicles (EVs), will drive further cost reductions.

Optical interconnects offer a promising solution for die-to-die communication in automotive systems, providing high bandwidth, low latency, energy efficiency, and immunity to EMI. Optical interconnects are poised to meet the growing demands for high-speed, secure, and efficient communication in automotive systems, especially as autonomous driving and electric vehicle technologies evolve.