POLITECNICO DI TORINO

Dipartimento di Elettronica e Telecomunicazioni (DET)

Master of Science in Electronic Engineering

Politecnico di Torino

Master Degree Thesis

Design of a data aggregation circuit for Autonomous Driving LiDAR sensors

Supervisors:

Prof. Luciano Lavagno

Ing. Carlo Oppedisano

Ing. Sebastiano Cicciarella

Candidate:

Claudia Rapisardi

October 2021

Contents

List of Figures

List of Tables

Acronyms

AD: Autonomous Driving

ADAS: Advanced Driver Assistance Systems

AMCW: Amplitude Modulated Continuous Wave

AP: Application Processor

ASIC: Application Specific Integrated Circuit

AV: Autonomous Vehicle

CSI-2: Camera Serial Interface 2

EBR: Embedded Block RAM

ECC: Error Correction Code

ECU: Electronic Control Unit

EoT: End of Transmission

EVB: evaluation board

FMCW: Frequency Modulated Continuous Wave

FOV: Field Of View

FPGA: Field Programmable Gate Array

GPIO: General-Purpose Input/Output

I^2C: Inter-Integrated Circuit

IP: Intellectual Property

LiDAR: Light Detection and Ranging

LLP: Low Level Protocol

LVDS: Low-Voltage Differential Signaling

MEMS: Micro Electro-Mechanical Systems

MIPI: Mobile Industry Processor Interface

NVCM: Non-Volatile Configuration Memory

PF: Packet Footer

PH: Packet Header

SAE: Society of Automotive Engineers

SerDes: Serializer/Deserializer

SoT: Start of Transmission

ToF: time-of-flight

VC: Virtual Channel

VCSEL: Vertical-Cavity Surface-Emitting Laser

WC: Word Count

Abstract

The automotive industry is facing an unprecedented era of change, revolutionizing the entire human driving experience, making it safer and more comfortable. The growth of Advanced Driver Assistance Systems (ADAS) is enabled by the data fusion collected from several types of sensors (camera, radar, ultrasonic). The gap between today's ADAS and its higher levels can be filled with the introduction of LiDAR sensors.

Light Detection and Ranging technology creates a 3D rich point cloud of the surrounding environment by using laser beams and time-of-flight distance measurement, allowing accurate real time objects detection, even at long distances and poor light or weather conditions when cameras lose their reliability. To cover the field of view required for ADAS functionalities from Level 3 and up, a LiDAR platform needs at least three optoelectronic modules (each equipped with a laser transmitter and a laser receiver) sending MIPI CSI-2 raw data streams to the CPU, which must process them to create a rich point cloud, decisive element for the autonomous driving strategy. However, Automotive CPUs usually only have up to two CSI-2 ports. MIPI Camera Serial Interface 2, with its high bandwidth D-PHY physical layer, is widely adopted in the automotive industry as interface for cameras, radar and LiDAR sensors, supporting a wide range of applications, raw data format, resolutions and frame rates.

Starting from these premises, the application deepened in this Thesis work provides a cutting-edge solution for a high-performance LiDAR system developed by Marelli Automotive Lighting: the evaluation, the design and the implementation of a Proof

Of Concept that aggregates data coming from two optoelectronic modules, to provide a unique output data stream to the CPU. Specifically, the main goal of this Thesis work is the development of a data aggregation circuit based on FPGA, exploiting one of the Long Packet fields named Virtual Channel, whose aim is to provide separate channels for different data flows interleaved in the same output data stream. The solution analyzed refers to an FPGA of the CrossLink Automotive family offered by Lattice Semiconductor. It supports a wide variety of protocols, pays special attention to energy efficiency and makes available all the features necessary for design purposes, as the entire blocks that implement MIPI protocol. This application requires flexibility, verification, and ability to iterate quickly: features offered by the reference software environment Lattice Diamond. It provides a complete set of tools for design entry, implementation, synthesis, analysis, programming and simulation activities.

At the end of the implementation the project tested on the evaluation board provides positive results. The input MIPI CSI-2 raw data collected from the two modules are correctly merged in a single MIPI CSI-2 output data stream to be sent to the Automotive System on Chip, as demonstrated by the MIPI protocol decoding performed by the oscilloscope.

Overall, this FPGA-based implementation proves to be extremely suitable and convenient for the project purposes. Indeed, comparing this technique with other hardware solutions previously investigated (ASIC, SerDes), many advantages in terms of flexibility, power consumption, costs, number of components involved (resulting in a reduced PCB area) make it an optimal choice for any possible future development.

CHAPTER 1

ADVANCED DRIVER ASSISTANCE SYSTEMS (ADAS)

1.1 The general scenario

The automotive industry is facing an unprecedented era of change. While acceleration, top speed, horsepower and design were the most important criteria for buying a car in the past, electronics innovations are the defining criteria of today (and tomorrow). Futurists have long dreamed of vehicles driving themselves, but the reality of fully autonomous vehicles remained out of reach until recently, when new technologies suddenly turned the fantasies of the past into present-day realities. Fully automated cars and trucks that drive us (instead of us driving them) will become a reality soon. The general public is aware that several large corporations and major automakers are developing self-driving technology. Less well known, however, is the extensive effort into assisted driving technologies and the semiconductor innovations enabling them. These technologies are rapidly changing car design, providing an evolution in automotive control that has put semi-autonomous vehicles on the roads now, and fully autonomous options coming in the next years.

Semi and fully autonomous automotive control, based on advanced electronic sensing

and processing, delivers real benefits in fuel savings, mobility and convenience, travel time and the efficient use of roadways. Most important, however, are new forms of control that work actively to promote safety, not only for drivers and passengers, but also other vulnerable road users as well. Vehicle control on the one hand represents a remarkable opportunity to enhance road safety, and on the other hand a thriving market for those offering enabling electronic technology. Leading-edge semiconductor solutions will help accelerate the introduction of these new capabilities, providing greater safety while sharing in this significant market.

Active safety depends on, among other things, Advanced Driver Assistance Systems (ADAS), a set of electronics-based technologies that are designed to aid in safe vehicle operation. ADAS innovations help prevent accidents by keeping cars at safe distances from each other, alerting drivers to dangerous conditions, protecting those in the car and on the street from bad driving habits, and performing other safety-related operations. If self-driving cars promise to free drivers so that they can use their time more effectively during long trips, ADAS features will help minimize collision repairs, prevent injuries and save lives. Actually, a large number of today's new motor vehicles have technology that helps drivers avoid drifting into adjacent lanes or making unsafe lane changes, or that warns drivers of other vehicles behind them when they are backing up, or that brakes automatically if a vehicle ahead of them stops or slows suddenly.

These and other safety technologies use a combination of hardware and software to help vehicles identify certain risks so they can warn the driver to act to avoid a crash. The continuing evolution of automotive technology aims to deliver even greater safety benefits and, one day, deliver automated driving systems that can handle the whole task of driving. ADAS represent an impressive evolution in vehicle sensing, intelligence and control that will ultimately lead to self-driving cars.

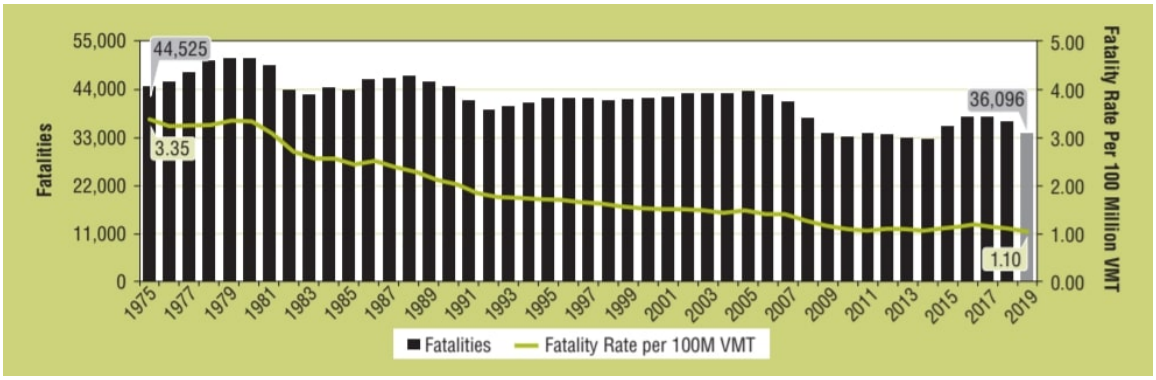
It is certain that, in long term, our experience of driving will be entirely revolutionized.

1.2 ADAS - concepts and challenges

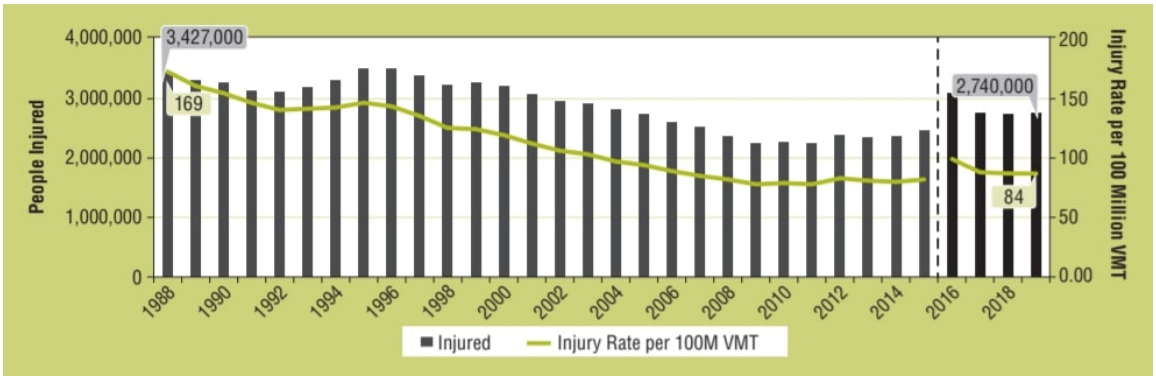
Every year, automobile manufactures invest millions of dollars in funding development of cutting-edge technologies to keep drivers safe and accident free while operating their vehicles. These technologies are widely known in the industry as Advanced Driver Assistance Systems (ADAS): they are a major driver of innovation, one of the fastest growing segments in the automotive domain, and for these reasons they are becoming more and more present in nowadays car. There is not a single definition of the acronym, one of the more representative ones can be [1]: "*An ADAS is a vehicle control system that uses environment sensors (e.g. radar, laser, vision) to improve driving comfort and traffic safety by assisting the driver in recognizing and reacting to potentially dangerous traffic situations. Since an ADAS can even autonomously intervene, an ADAS-equipped vehicle is popularly referred to as an 'intelligent vehicle'*". The basic definition of ADAS is technology that helps drivers when they are driving or parking. In a broader sense, ADAS enhance car and road safety by minimizing human error. In other words, ADAS technologies have the potential to improve the driving experience by making it safer and more comfortable. They work by alerting or assisting drivers to prevent or mitigate crashes. Purely by way of example, if the vehicle detects an object such as another vehicle or a cyclist in a location where the driver may not be able to see them, features such as Blind Spot Warning or rear backup warning will alert the driver. Likewise, if the system determines that the vehicle is drifting out of its lane, it could activate Lane Departure Warning to alert the driver. When these kinds of detection are coupled with a technology that takes actions beyond a simple warning, ADAS become an active safety system, meaning that the vehicle actively controls braking or steering.

There are multiple benefits of implementing ADAS: not only does it helps to improve

the safety of the driver and the passengers, but also helps to enhance the overall user experience. First of all, the safety benefits of automated vehicles are paramount. Traffic accidents are high on the list of causes of injury and death for the world population.



(a) Fatalities and fatality rate



(b) People injured and injury rate

Figure 1.1: Motor vehicle crashes

In 2019, the National Highway Traffic Safety Administration (NHTSA) estimated that 36096 people died in motor vehicle crashes on U.S. roads, as shown in figure 1.1a; this represents a 2.0% decrease from 36835 fatalities in 2018 (or 739 fewer fatalities). The number of people injured on roadways increased to 2.74 million, rising from 2.71 million in 2018 (an increase of 1.1%), as shown in figure 1.1b. This figure contains data from the National Automotive Sampling System (NASS) General Estimates System (GES) for the years 1988 to 2015. The estimates from CRSS 2016-2019

and NASS GES 1988-2015 are not comparable as they are based on different sample designs. Moreover, the estimated number of police-reported crashes increased from 6.74 million in 2018 to 6.76 million in 2019 (a 0.3% increase) [2].

Since traffic accidents are overwhelmingly caused by human error (as much as 90%), assisting drivers so that they can control their vehicles more safely is an obvious point of attack for reducing deaths and damages [3]. Although it is important to remember multiple factors contribute to all crashes, the largest portion of driver error issues involve the driver failing to recognize hazards, including distraction. Many of the most promising ADAS technologies are designed to identify and react to potential hazards faster than a human driver. ADAS can have the potential to remove human error from the crash equation. That is why the availability of ADAS technologies in new cars is growing rapidly. A recent survey by the National Safety Council [4] concluded that in 2013 ADAS technologies were present in less than 5% of new passenger vehicles. By 2018, availability in new cars varied from 24% for lane keeping assist to 42% for automatic emergency braking, as shown in figure 1.2.

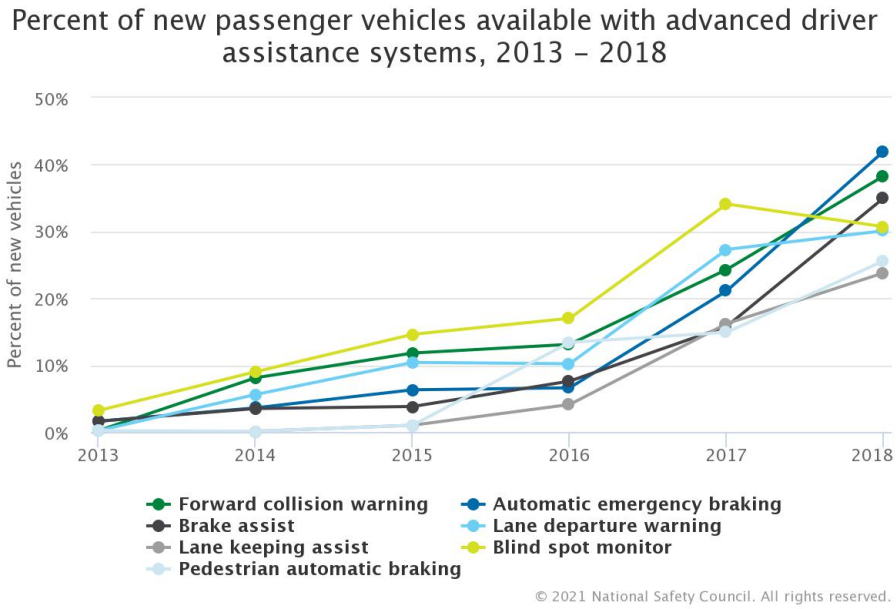


Figure 1.2: Percent of new passenger vehicles available with ADAS

To start forecasting the potential impact ADAS technologies may have on traffic

safety, a recent NHTSA study [4] identified the crashes occurring today that could be prevented or mitigated if all vehicles where equipped with ADAS technologies. The estimates also assume that ADAS are fully effective in preventing or mitigating the crashes they are designed to impact. It is calculated that collectively the five ADAS technology groups could impact 3.59 million total crashes per year, or about 62% of all crashes. Forward collision prevention accounts for 1.7 million crashes, while lane keeping assist impacts another 1.12 million crashes, as shown in figure 1.3.

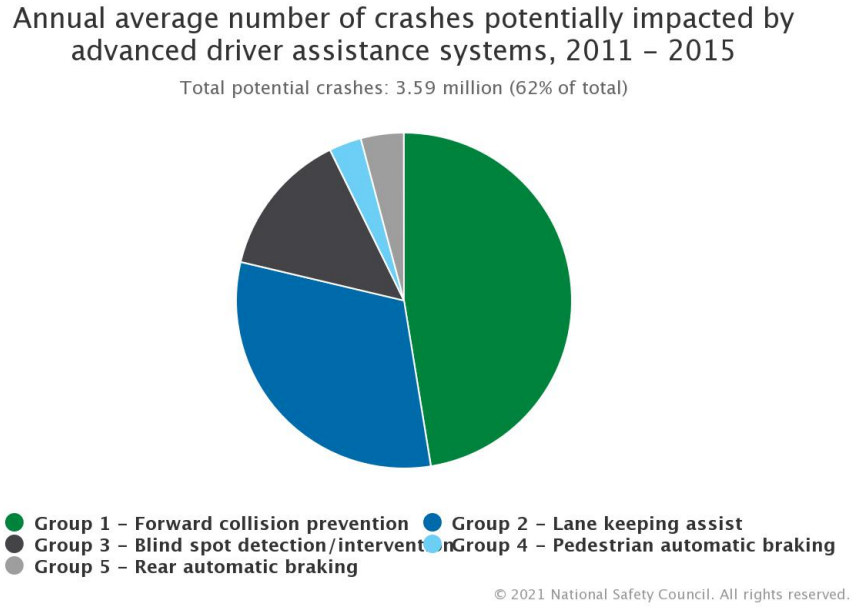


Figure 1.3: Annual average number of crashes potentially impacted by ADAS

ADAS technologies have the potential to prevent 20841 deaths per year, or about 62% of total traffic deaths. Lane keeping assist accounts for 14844 of this savings, while pedestrian automatic braking accounts for another 4106 lives saved, as shown in figure 1.4.

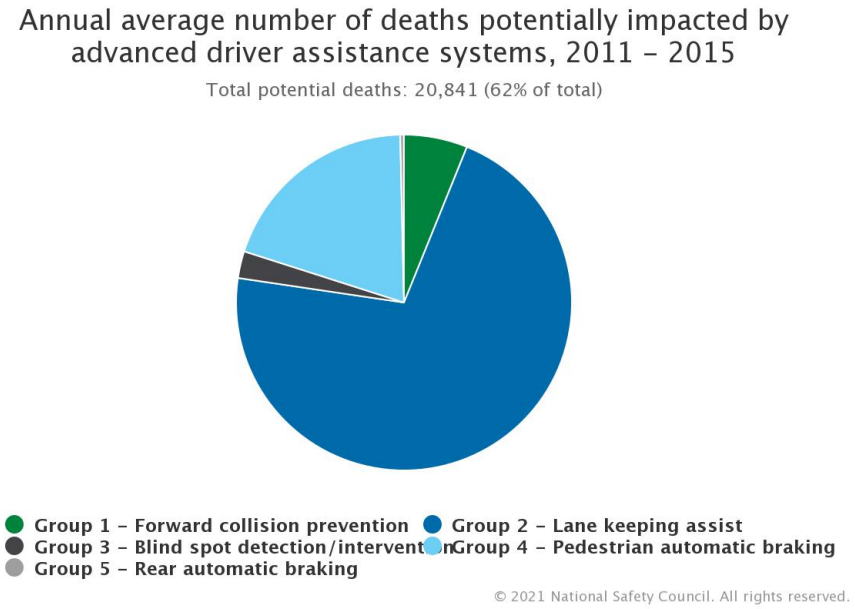


Figure 1.4: Annual average number of deaths potentially impacted by ADAS

ADAS technologies can also potentially prevent or mitigate 1.69 million injuries (about 60% of total traffic injuries). The majority of the injury reduction is achieved by forward collision prevention and lane keeping assist, as shown in figure 1.5.

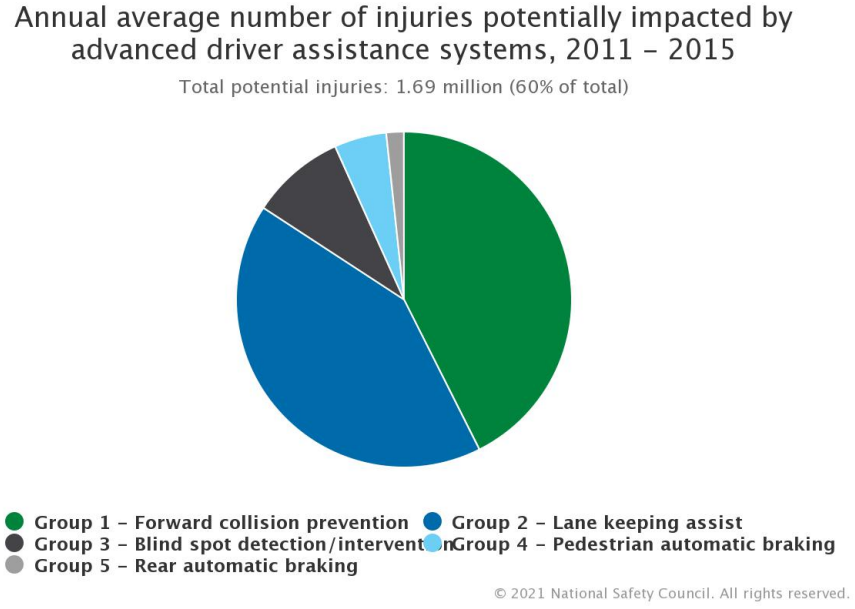


Figure 1.5: Annual average number of injuries potentially impacted by ADAS

As a direct consequence of all these considerations automated vehicles can deliver additional economic and societal benefits, eliminating all costs and expenses related to accidents. Another invaluable benefit ADAS offers is that streets filled with automated vehicles can help regulate traffic flow and reduce traffic congestion. In this way, time and money spent and dedicated to driving can be saved or heavily reduced. Advanced Driver Assistance Systems are electronic systems embedded in the vehicle in order to assist the driver, detect nearby obstacles or driver errors, and respond accordingly. Sensors, processors, actuators, mapping systems and various software systems come together to make all this possible. Let us take a look at the main components that power these technologies.

Sensors: even if drivers are the primary decision makers behind the wheel, with ADAS some of the decisions can be taken by the systems.

Sensors are used to ensure that adequate safety measures are taken, based on a specific situation. They perceive the world around them, and then either provide information to the driver or take automatic action based on what it perceives. Sensors can detect what other vehicles and pedestrians are doing on the road relative to its own position as well as detect driver distraction or inattention. However, no one sensor cannot guarantee safety on its own. Hence, multiple and complimentary sensors and the corresponding fusion of data are used as a part of ADAS, which together provide information and redundancy to enhance safety, overcome the drawbacks of individual sensor solutions and improve the driver's performance.

The standard sensor suite comprises cameras, radar, ultrasonic and LiDAR technologies.

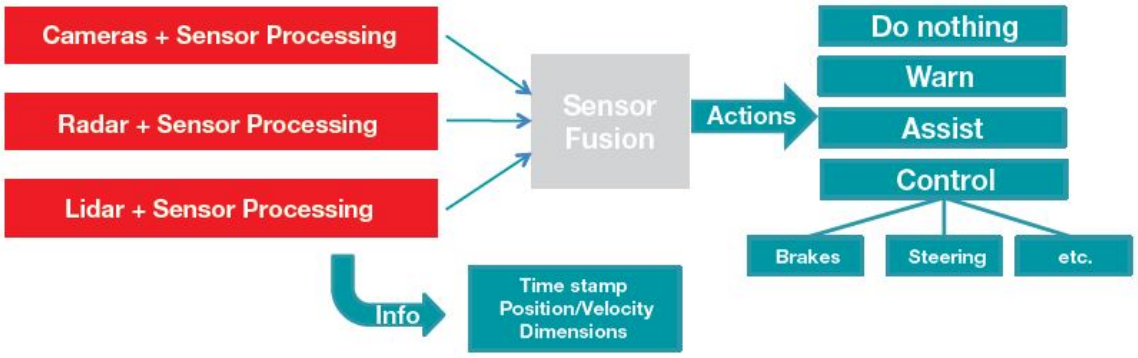


Figure 1.6: Sensor fusion technology

Processors: more sensors and more cameras with higher resolution all translate into requirements for high bandwidth communications and high performance processing. In ADAS applications, processors are employed for everything, from building a real time 3D spatial model of a car's surroundings to calculating proximity and threat levels based on the environment. However, due to the length of qualification processes in the automotive industry, the adoption of advanced manufacturing technologies is almost six years slower than the rate at which average smartphone processors are adopted [5]. Nevertheless, the systems have to recognize items (such as stop signs, pedestrians or other vehicles), classify them, and decide what to do. Each of these time-critical tasks serves to narrow the data stream while increasing the algorithm's complexity. As a result, the system requires heterogeneous processing, ranging from dedicated video signal processing hardware for the raw data input, through programmable signal processing for object scanning and recognition, to a high performance microprocessor for decision-making that affects vehicle operation. Hence, solutions and processors that offer advanced deep learning and networking capabilities are needed to solve the design challenges in Advanced Driver Assistance Systems and automotive gateway applications.

In this context, using more energy-efficient hardware than conventional general-purpose central processing units is absolutely important, which is why emerging ADAS hardware must rely on graphics processing units, digital signal proces-

sors and image signal processors customized to reduce power consumption for ADAS applications. Moreover, as the embedded systems for ADAS operate in real time, they have strict timing constraints, which establish a latency minimization requirement.

All these specifications limit the range of processors to use, making this component a core element of ADAS systems.

Actuators: the electrification of the actuation systems in vehicles has been a major facilitator of ADAS. This allows the various ADAS systems to interact easily with other electrical components of the vehicle. Specifically, processors collect and analyze data from vehicle sensors and then the ADAS system makes the resulting decision feasible by the actuators.

Actuator systems support everything, from electric power steering to acceleration and autonomous braking.

Mapping systems: the geographical and infrastructure information is collected, stored and updated via sensors to govern the vehicle's exact location. When referring to the higher levels of autonomous driving, this information is maintained and communicated to the control system even if the GPS coverage fails. Vehicles are about to become a lot more communicative: with other road users, with the infrastructure they pass on their journeys, with cloud-based services, and even with the energy grid; this communication should enable new capabilities. The auto industry, and the regulators that enable it, have come up with a series of acronyms to denote the various ways in which connected cars will communicate with other entities.

- V2V stands for Vehicle-to-Vehicle Communication. Cars, vans, trucks, and even motorbikes communicate directly with each other to share information about road conditions and hazards, and to collaborate on managing traffic.

- V2I stands for Vehicle-to-Infrastructure Communication and refers to techniques for connecting cars with road-management systems, such as traffic lights and speed signs.
- V2N stands for Vehicle-to-Network Communication and enables access to in-vehicle service providers and infotainment streams.
- V2P stands for Vehicle-to-Pedestrian Communication. When a pedestrian is near a crossing, his smartphone can communicate to nearby vehicles and infrastructure his presence and his will to cross the road.
- V2G stands for Vehicle-to-Grid Communication and implies a future in which hybrid and electric vehicles that are on charge become part of an intelligent energy distribution grid, helping to smooth out peaks and troughs in power demand by sinking energy as needed.

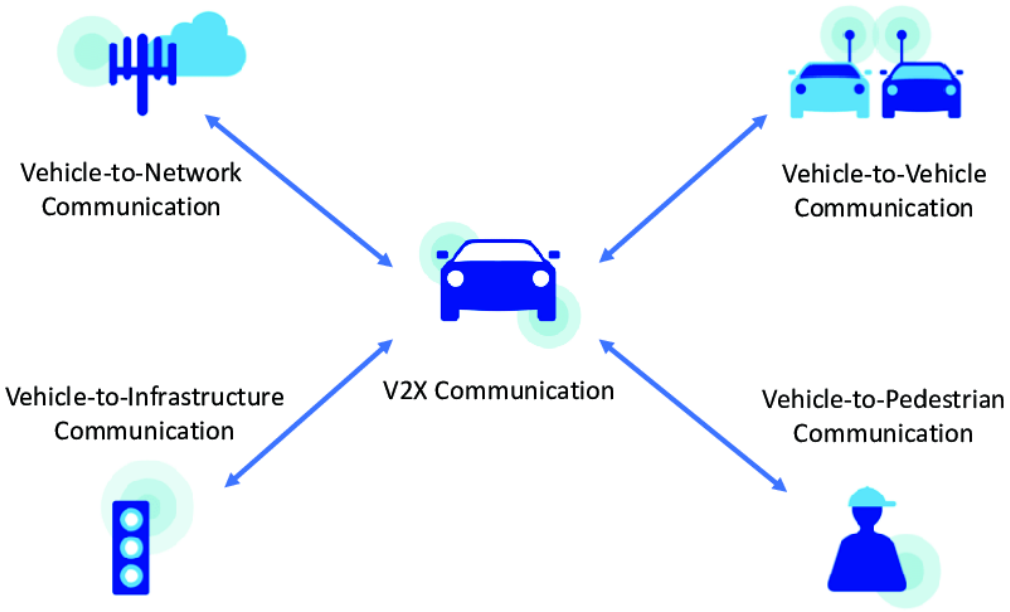


Figure 1.7: V2X Communication

These kinds of communication have given raise to the concept of V2X (Vehicle-to-Everything): a vehicular communication system that supports the transfer of information from a vehicle to the parts of the road transportation system

that may affect the vehicle. The main purpose of V2X technology is to improve road safety, energy savings, and traffic efficiency on the roads. In a V2X communication system, the information travels from the vehicle sensors and other sources through high bandwidth and high reliability links, improving the driver's awareness of potential dangers and reducing the severity of injuries, road accident fatalities, and collision with other vehicles.

Software: today, with changing customer expectations, almost every automotive company relies as much on software as it does on the actual vehicle.

ADAS systems are becoming increasingly effective and are seeing greater adoption of technologies such as cloud, mobility, deep learning and artificial intelligence.

Continuous improvement and effective cooperation of all these components allow ADAS to revolutionize the way the world drives.

The concept and the development of Advanced Driver Assistance Systems have been around longer than most people realize. Ever since the first automobiles rolled onto the road, manufacturers have been introducing technology to ensure they avoid crashing into each other. But it was not until the mid-1990s that innovation really intelligently assisted cars and drivers.

Back in 1992, Mitsubishi unveiled a very basic camera operated tracking system that could track lane markings on the road. If the driver drifted across those road markings, an alarm would sound to warn the driver. This was available on the Mitsubishi Debonair, and it was the world's first Lane Departure Warning (LDW) system. Toyota had the next major breakthrough in lane departure technology in 2004. They added a system to the Crown Majesta model that would monitor the road conditions and actually assist the driver. This was achieved by sending commands to the power steering system to subtly encourage the driver to make a steering correction. This is when the terms lane keeping assist, lane assistance, and lane assist began to be used.

A vocal minority rejected the idea of losing some of their autonomy, but over time people seemed to get used to the technology. It also helped that many subsequent systems could be turned off completely if the driver wished to do so.

Figure 1.8: Lane Departure Warning system

In the same years, Mitsubishi became the first OEM to offer an Adaptive Cruise Control (ACC) system after equipping its 1995 Diamante sedan with a Preview Distance Control system, which introduced LiDAR in the front bumper and a miniature camera in the rear-view mirror. It was able to sense when the distance to the vehicle ahead was closing and would automatically ease off the accelerator or make the transmission downshift to slow the car. Its limitation, however, was that it could not operate the brakes, so when the speed difference with the vehicle in front was too great, it had to resort to alerting the driver with audible and visual warnings. With no braking intervention, an operational limit of 108 km/h and poor performance in the rain, Mitsubishi decided to keep the system solely for the Japanese market, where it suited the road conditions and generally clement weather.

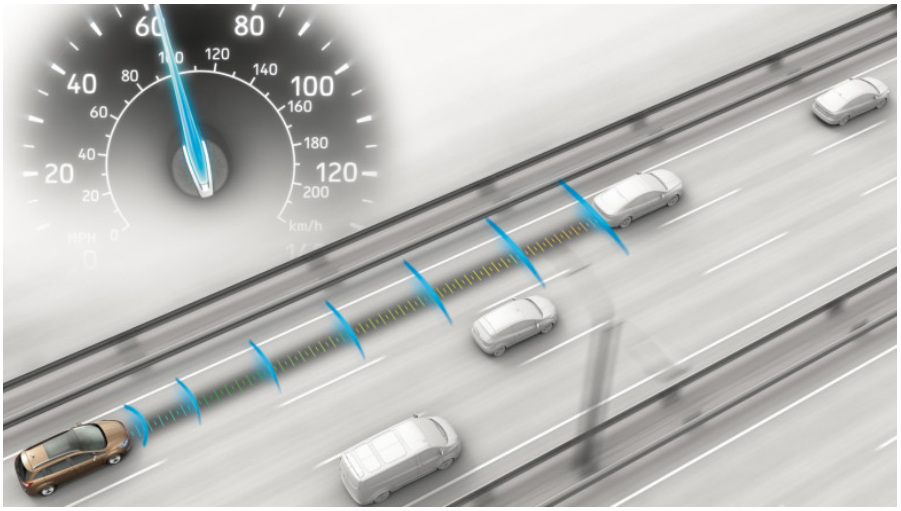


Figure 1.9: Adaptive Cruise Control system

Another crucial driver assist technology was the Blind Spot Warning (BSW), a system of protection developed by Volvo. This system was first introduced on the redesigned 2007 Volvo S80 sedan. It used sensors to monitor the side of the vehicle for vehicles approaching blind spots. In many systems, a visual alert appeared on or near the side-view mirrors if a vehicle was detected.

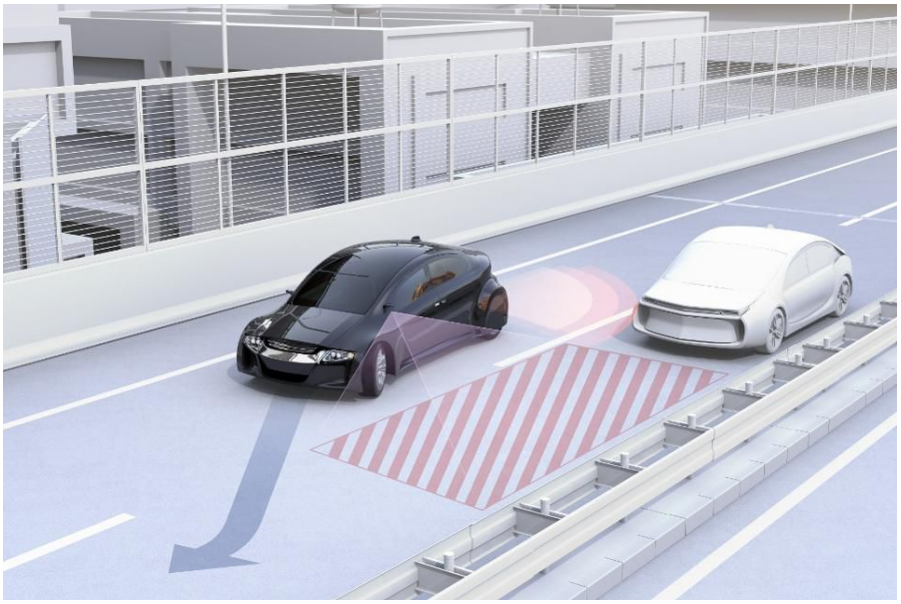


Figure 1.10: Blind Spot Warning system

Current research indicates systems like ACC and LDW work harmoniously to create a safer driving environment via technical automation. Still, there is a school

of thought that drivers may develop a false sense of security due to over-reliance on technology. Despite all, it is important to remember that current cars do not drive themselves and, as nearly every owner's manual states, "*the operator is responsible for safe vehicle operation*".

1.3 Six levels of technology

In the automotive industry, the concept of "automatic assistance" is often misunderstood, resulting in disasters and accidents caused not only by distracted but above all uninformed drivers.

This has occurred with Tesla and other luxury car brands with the promotion, for commercial purposes, of ADAS benefits and features beyond their actual capabilities. Basically, the seller might say to the customer "just press this button and the car almost drives itself". After purchasing the car, the new owner engages the ADAS system and starts playing a game on his phone. This lack of understanding of ADAS limitations result in accidents with some fatalities.

Moreover, it is of fundamental importance to stress the difference between ADAS and AD: it is the extent to which a driver is involved in driving the car. The idea behind ADAS is to enhance functions that help drivers avoid accidents as much as possible while driving and help them reach their destination comfortably. On the other hand, Autonomous Driving is the idea that the vehicle can reach the destination without the human involvement.

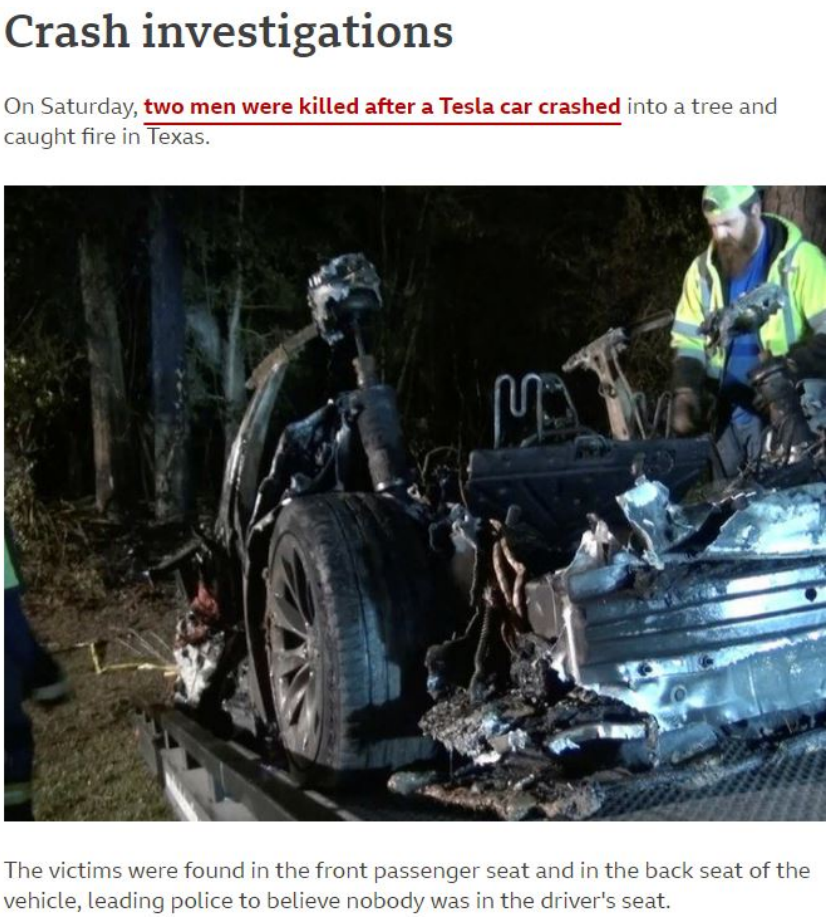


Figure 1.11: Tesla car crash [6]

Because OEMs, software companies and the aftermarket are all developing autonomous cars and the components that supports them, a common language is necessary to describe the technology to avoid confusion.

In this context, different international standards organizations (SAE, ISO, NHTSA, etc.) are involved in defining a set of autonomous driving levels, functional safety levels and other requirements and characteristics for ADAS and AD systems, providing a common terminology. Lack of full standardization might make the system have difficultly being understandable by the driver.

Specifically, SAE International (standing for Society of Automotive Engineers) is a global association committed to advancing mobility knowledge and solutions for the benefit of humanity. By engaging nearly 200000 engineers, technical experts and volunteers, they connect and educate mobility professionals to enable safe, clean, and

accessible mobility solutions. SAE is the leader in connecting and educating engineers while promoting, developing and advancing aerospace, commercial vehicle and automotive engineering.

Figure 1.12: SAE International 2020 Annual Report

SAE International recently unveiled a new visual chart that is designed to clarify and simplify its J3016 Levels of Driving Automation standard for consumers. It serves as the industry's most-cited reference for automated-vehicle capabilities. The J3016 standard defines taxonomy with supporting terms and definitions for six levels of driving automation in the context of motor vehicles and their operations on roadways. These levels range from SAE Level 0 (no automation), where a fully engaged driver is required at all times, to SAE Level 5 (full vehicle autonomy), where an automated vehicle operates independently, without a human driver. The update is the latest iteration of the J3016 graphic first deployed in 2016. As the industry gets closer to producing AVs in volume, the SAE J3016 Technical Standards Committee saw the need to more clearly explain the features in each of the six driving levels, and how they relate to consumers' increased safety and convenience. This latest update to SAE's Levels of Driving Automation refines the previous version with the addi-

tion of several new terms, substantial refinement and clarification of misinterpreted concepts, and restructuring of certain definitions into more logical groupings. The latest J3016 graphic is a "living document". It will continue to evolve gradually as the industry and the technical standard J3016 itself evolves [7].

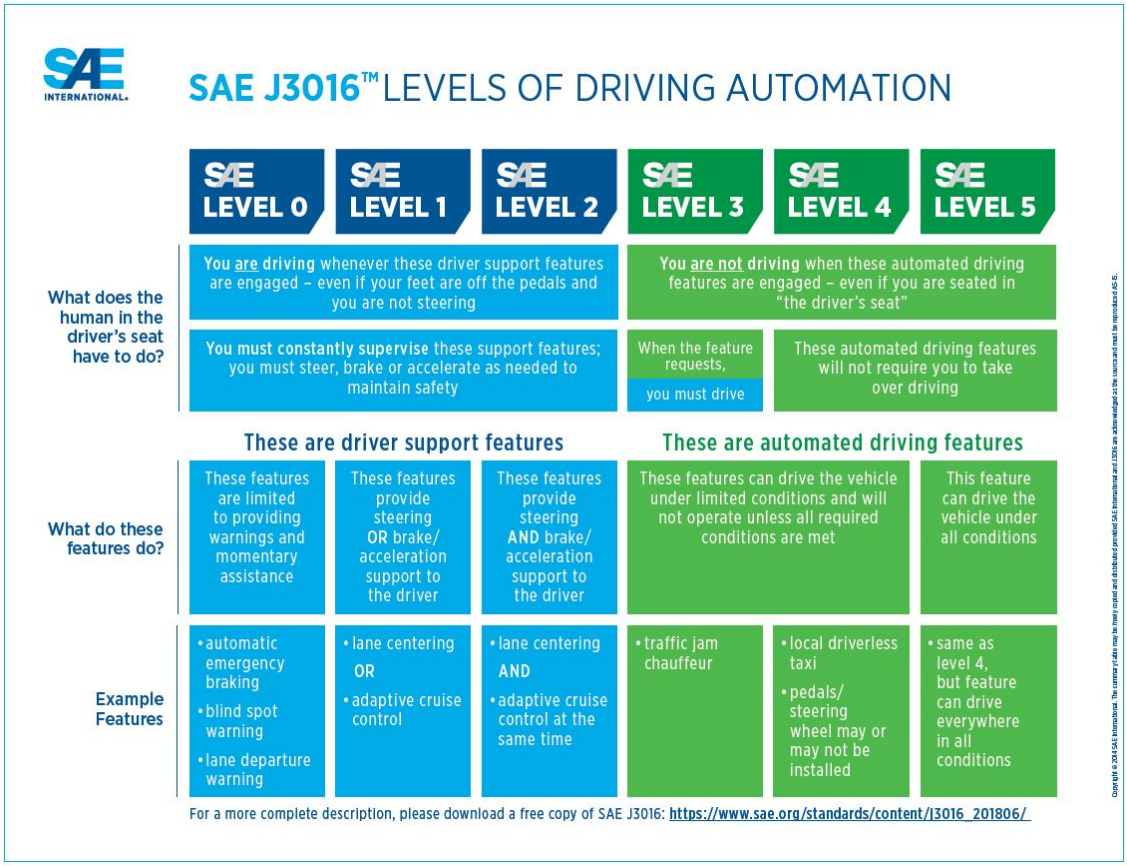


Figure 1.13: J3016 automated-driving graphic update

These levels, along with additional supporting terms and definitions, can be used to describe the full range of driving automation features equipped on motor vehicles in a functionally consistent and coherent manner.
Here is what those levels generally mean.

LEVEL 0, NO DRIVING AUTOMATION: as the name suggests, Level 0 relies completely on the driver to perform all longitudinal and lateral tasks, such as steering, braking, accelerating or slowing down. The driver is in complete

control of (and responsible for) the act of driving.

While not featuring any form of automation, the system will give some warnings. These could be, for example, lane departure or forward collision warnings. As they only inform the driver through alerts and notifications, they still fall under Level 0.

LEVEL 1, DRIVER ASSISTANCE: while the driver cannot renounce control of the car, Level 1 systems assist with some driving tasks. The vehicle only controls or intervenes to control the speed or steering of the vehicle, but not both at the same time.

An example of such an ADAS function is Adaptive Cruise Control, where the car will keep a set speed and safe distance between the car ahead by automatically applying the brake when traffic slows and resuming its original speed when traffic clears.

Another use case is Lane Keep Assist, which brings the car back into the middle of the lane in case the vehicle veers off slightly without activating the turn signal.

LEVEL 2, PARTIAL DRIVING AUTOMATION: moving up to Level 2, the driving task is shared between the vehicle and the driver. The vehicle usually takes over the two primary driving functions of lateral and longitudinal control. This can be achieved, for example, by combining Adaptive Cruise Control with Lane Keeping. In this case, the driver is allowed to temporarily take their hands of the wheel. However, the driver still needs to have constant situational awareness and monitor the surrounding environment.

Some of the most notable examples of carmakers using Level 2 automation are GM's Super Cruise, the Mercedes-Benz Drive Pilot, the Tesla Autopilot, Volvo's Pilot Assist and the Nissan ProPilot Assist 2.0.

LEVEL 3, CONDITIONAL DRIVING AUTOMATION: the jump from Level 2 to Level 3 is substantial from a technological perspective, but subtle from a

human point of view.

The system is able to relieve the driver of the need to have continuous control of the longitudinal and lateral movement of the vehicle. The car can accelerate past a slow-moving vehicle, monitoring its surroundings, changing lanes, and controlling the steering, throttle, and braking. All the driver has to do is keep paying attention and be ready to take back control when the vehicle calls for it. This level allows you to take your hands off the wheel and eyes off the road (as long as you remain alert). The result is a relaxed driver on certain occasions, like when driving in traffic jams.

One of the most relevant aspects of a Level 3 system is that it is able to recognize its limits and when the conditions of the external environment exceed its possibilities. In these conditions, the system is deactivated by giving a warning to the driver who must regain control. Moreover, if the system is disabled and there are no external conditions to be active, then it does not turn on. This happens, for example, if the roadway gets too narrow, if the vehicle speed exceed certain limits, if the road signs are not clearly visible or when road works are present.

However, the complexity of predicting how and when the driver manages to regain control of the car has prompted some carmakers to abandon the idea of Level 3.

LEVEL 4, HIGH DRIVING AUTOMATION: the key difference between Level 3 and Level 4 automation is that Level 4 vehicles can intervene if things go wrong or there is a system failure. In this sense, the interaction between human and machine lowers as the vehicle's capability increases. Steering, braking, accelerating and monitoring the environment are taken out of the driver's hands, as well as changing lanes, turning and signaling. The vehicle can handle highly complex driving situations, such as the sudden appearance of construction sites, without any driver intervention. At the moment, this is allowed for specific,

predefined circumstances, such as on controlled access highways. For the driver, this means he can safely relax and even read a book, while the car responsibly and safely drives on the highway and, possibly, even on city roads.

However, a human still has the option to manually override. Moreover, the car can still prompt the driver to take back control, but if it receives no response, the car is able to bring itself to a safe stop.

An example of Level 4 autonomy is the Waymo test car.

LEVEL 5, FULL DRIVING AUTOMATION: it requires zero human attention. There's no need for a steering wheel, no need for brakes and no need for pedals. The autonomous vehicle controls all driving tasks under all conditions, including the monitoring of environment and identification of complex driving conditions like busy pedestrian crossings. This also means that the vehicle can perform a combination of several tasks simultaneously, whether adaptive cruise control, traffic sign recognition, lane departure warning, emergency braking, pedestrian detection, collision avoidance, cross traffic alert, surround view, park assist, rear collision warning or park assistance.

At Level 5, passengers would be able to safely work, eat or even to take a nap while the car takes up entire driving functions. This has important implications, as every person in an autonomous car becomes a passenger, relieved from the stress of driving and with free time on their hands.

Fully autonomous cars are undergoing testing in different parts of the world, but none are yet available to the general public mainly due to the unavailability of the necessary technologies.

The promise of driverless cars is getting ever closer to being a reality, but for most us, our first trip in an autonomous vehicle is likely to be in a robotaxi. The combined challenge of cost, regulation and geographic scale make it unlikely that autonomous cars will go into mass production just yet. What is far more likely is that organisations will use robotaxi as a stepping stone to the

mass production of autonomous passenger cars.

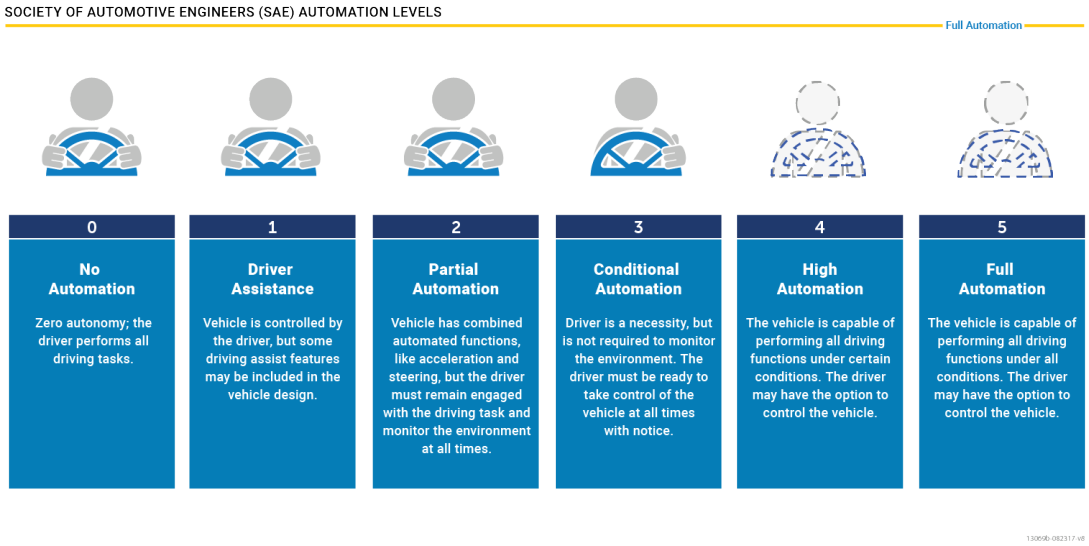


Figure 1.14: Levels of Driving Automation

Overall, the world of autonomous driving does not consist of only one single dimension. By allowing technology into the driver seat, the automotive industry is making a bid to reduce accidents on the road, increase driver comfort and powertrain efficiency.

Through their ambitious timelines for the deployment of high levels of automated driving, OEMs are driving the rapid deployment of autonomous vehicles and the growth of sensor types that support the technology. Specifically, SAE Levels 3, 4 and 5 will not be achievable without employing a variety of different sensors. These sensors help each other in case of failures and allow a redundancy that autonomous driving (and not just those) need. To make autonomous vehicles successful, passengers need to trust the different sensors and technologies used; the key to this trust lies in the merging of inputs from various types of sensors to increasingly improve accuracy, redundancy and safety.

ABI Research, a market-foresight advisory company providing strategic guidance on

the most compelling forward-looking technologies, forecasts 8 million consumer vehicles shipping in 2025 will feature SAE Level 3 and 4 technologies, where drivers will still be necessary but are able to completely shift safety-critical functions to the vehicle under certain conditions, and SAE Level 5 technology, where no driver will be required at all. This, in turn, will increase the application of Light Detection and Ranging (LiDAR) sensors that sustain these technologies. As many as 36 million LiDAR units are expected to ship in 2025, corresponding to a market value of 7.2 billion dollars [8]. "*The gap between today's ADAS and higher level autonomous vehicles will be filled with the addition of LiDAR, which will help to provide reliable obstacle detection and Simultaneous Localization and Mapping (SLAM)*", says Shiv Patel, Research Analyst at ABI Research.

For conditional and high-level automation applications within the consumer market, i.e. SAE Level 3 and Level 4, solid state LiDAR solutions from companies such as XenomatiX and LeddarTech have emerged as the LiDAR form factor that will not only help enable robust sensing on autonomous vehicles but also, more importantly, satisfy stringent pricing requirements set by OEMs.

In fully autonomous applications, i.e. SAE Level 5, where the aim is to eliminate the driver completely, much more expensive, traditional mechanical LiDAR solutions, with their higher resolution for robust sensing, remain the go-to option.

Finally, although the performance of solid state LiDAR continues to improve, mechanical LiDAR as part of a broader suite of other sensor types is currently seen as the only short-term option to enable full automation as soon as possible.

CHAPTER 2

LIGHT DETECTION AND RANGING (LiDAR)

2.1 LiDAR - an overview

The growth of Advanced Driver Assistance Systems and Autonomous Driving solutions is the catalyst for the adoption of several types of sensors being incorporated into vehicles. Radars, cameras, and ultrasonic sensors have become the industry standard, answering the call for advancements in road safety.

Automotive companies are coming up with innovative technologies in Advanced Driver Assistance Systems, using new and affordable sensors. In fact, for inclusive vehicle safety solutions, ADAS systems cannot be dependent on just vision and radar-based systems; they require more efficient systems capable of providing highly accurate data for improved driver assistance.

More recently, Light Detection and Ranging (LiDAR) technology has been added to the list and is actively being deployed in vehicles on the production line. Light Detection and Ranging is a sensing method that enables autonomous vehicles to "see" the surrounding world, creating a virtual model of the environment to facilitate decision-making and navigation. A LiDAR sensor creates a 3D map of the surrounding envi-

ronment by using laser beams and time-of-flight (ToF) distance measurements. ToF, which is LiDAR's working principle, provides distance information by measuring the travel time of emitted light. Reflected light signals are measured and processed by the vehicle to detect, identify, and decide how to interact with or avoid objects.

LiDAR systems were born in the 1960s, just after the advent of the laser. This technology was originally developed by NASA and the U.S. military to track lunar and satellite distances. During the Apollo 15 mission in 1971, astronauts mapped the surface of the moon, giving the public the first glimpse of what LiDAR could do. Before LiDAR was even considered for automotive and self-driving use, one of the popular use cases of LiDAR was archaeology. It provided a ton of value for mapping large-scale swaths of land, and both archaeology and agriculture benefited tremendously from it. It was not until the 2000s when LiDAR was first utilized on cars, where it was made famous by Stanley (and later, Junior) in the 2005 DARPA Autonomous Vehicle Grand Challenge.

Figure 2.1: Stanley: the robot that won the DARPA Grand Challenge in 2005

Earlier iterations of self-driving car LiDARs were electromechanical and mounted

on bases on the top of the vehicle that rotated mechanically to emit laser light in 360 degrees; the laser and collector continuously rotated to scan the area around it. These moving parts had to be precise to obtain measurements suitable for autonomous navigation. In addition, moving parts meant that the sensor would be less resilient to vibrations. Driving in rough terrain, for example, could negatively impact measurements.

So, whereas vision and radar-based sensors are capable of providing a high level of automation to vehicles, the conception of a fully automated self-driving vehicle is impossible without LiDAR-based sensors. Indeed, for SAE Levels 3, 4 and 5, automotive companies have to rely on all the three types of ADAS sensors, i.e. vision, radar and LiDAR-based sensors. All these sensor modules complement each other to provide a safe and comfortable automated driving experience. In addition to the technologies complementing each other, it is also important to have sufficient overlap in order to increase redundancy and improve safety. Sensor fusion is the concept of using multiple sensor technologies to generate an accurate and reliable map of the environment around a vehicle. Although vision-based systems assist in high visibility conditions, helping by providing parking assistance, recognizing traffic signs, identifying road markings and more, radar-based systems perform in low visibility conditions, covering a relatively longer range. When it comes to sensing the vehicle's surroundings with a 360 degrees field of view (FOV), LiDAR-based systems are highly accurate in object detection and recognition of 3D shapes, even for longer distances. LiDAR system's 3D mapping capability also helps in differentiating between cars, pedestrians, trees, people, or other objects, while also calculating and sharing details of their velocity in real time.

Advanced Driver Assistance Systems based on LiDAR sensors are the most innovative and efficient technologies for autonomous vehicles. Along with vision and radar-based systems, LiDAR systems provide high accuracy, precision in object detection and recognition in ADAS. The combination of amazing navigation, predictability and

high resolution object tracking has meant that LiDAR is the key sensor in self-driving cars today; it is highly effective in creating a safe and comfortable automated driving experience.

In the late years, LiDAR has progressed from a useful measurement technique suitable for studies of aerial mapping, towards a kind of new Holy Grail in electronic and optomechanical engineering. The fuel of all this activity is the lack of an adequate solution in all aspects for LiDAR imaging systems for automobile either because of performance, lack of components, industrialization or cost issues. LiDAR imaging systems for automotive require a combination of long-range, high spatial resolution, real time performance and tolerance to solar background in the daytime, which has pushed the technology to its limits. Different specifications with different working principles have appeared for several possible usage cases, including short and long-range, or narrow and wide fields of view. As mentioned above, rotating LiDAR systems were the first to achieve the required performances, using a rotating wheel configuration at high speed and multiple stacked detectors. However, automotive applications required additional performance, like the capability to industrialize the sensor to achieve reliability and ease of manufacturing in order to get a final low cost unit; or even to have a small, nicely packaged sensor fitting in small volumes of the car. There is a continuing need to obtain extreme miniaturization and/or longer-range in complex vehicular surround sensing applications, and this at a reasonable cost and in a compact, semiconductor-integrated form factor.

2.2 Measurement principles

The most well known measurement principle employed for LiDAR imaging is the time-of-flight (ToF), which is used to determine the distance of objects from a sensor. LiDAR is an active, non-contact range-finding technology in which an optical

signal is projected onto an object, the target, and the reflected or backscattered signal is detected and processed to determine the distance, allowing the creation of a 3D rich point cloud of a part of the environment. Hence, the distance to the target is measured based on the round-trip delay of light waves that travel to the target. This may be achieved by modulating the intensity, phase, and/or frequency of the transmitted signal, and measuring the time required for that modulation pattern to appear back at the receiver.

The emitted beam is in the non-visible spectrum with wavelengths in a range from 860 nm to 1550 nm; its typical value is 940 nm.

In the most straightforward case, a short light pulse is emitted towards the target, and the arrival time of the pulse echo at the detector sets the distance. This method enables to reach long distances while maintaining average power below the eye-safety limit.

A second approach is the so called Amplitude Modulated Continuous Wave (AMCW): the phase of the emitted and backscattered detected waves are compared enabling to measure distance. In this case, the reflected signal coming from distant objects arriving at the receiver is not as strong as in the pulsed case, which makes the amplitude to remain below the eye-safe limit at all times. However, the digitization of the back-reflected intensity level becomes difficult at long distances.

Finally, a third approach is defined by Frequency Modulated Continuous Wave (FMCW) techniques, enabled by direct modulation and demodulation of signals in the frequency domain, allowing detection by a coherent superposition of the emitted and detected wave. FMCW presents two outstanding benefits ahead of the other techniques: it achieves resolutions in range measurement well below those of the other approaches, although its main benefit is to obtain velocimetry measurements using the Doppler effect.

The three techniques mentioned are briefly discussed in the following subsection.

2.2.1 Pulsed approach

Pulsed time-of-flight are powerful techniques for accurate and affordable distance measurements.

All time-of-flight sensors measure distances using the time that photons take to travel between two points: from the sensor's emitter to a target and then back to the sensor's receiver. Direct and indirect ToF both offer specific advantages in specific contexts. Both can simultaneously measure intensity and distance for each pixel in a scene.

Let us focus on the direct pulsed time-of-flight.

The distance to the target is determined by multiplying the speed of light by the time a light pulse takes to travel the distance to the target. Since the speed of light is a given constant within the same optical medium, the distance to the object is directly proportional to the traveled time. The measured time is obviously representative of twice the distance to the object, as light travels to the target forth and back, and, therefore, must be halved to give the actual range value to the target:

$$d = \frac{c}{2} ToF \tag{2.1}$$

where d is the distance to the target, c is the speed of light in free space (c = $3{\times}10^8$ m/s) and ToF is the time it takes for the pulse of energy to travel from its emitter to the observed object and then back to the receiver.

Figure 2.2 shows a simplified scheme of a typical implementation.

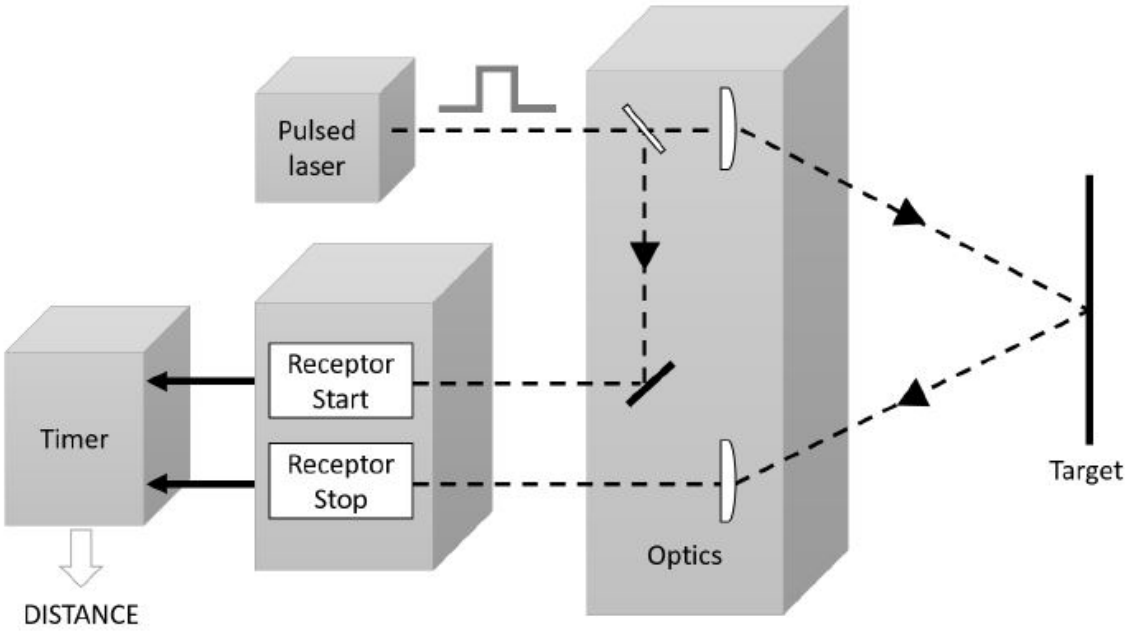


Figure 2.2: Direct pulsed ToF measurement principle

The achievable resolution in range is directly proportional to the resolution in time counting available. As a consequence, the resolution in depth measurement is dependent on the resolution in the time counting electronics. A typical resolution value of the time interval measurement can be assumed to be in the 0.1 ns range, resulting in a resolution in depth of 1.5 cm. Such values may be considered as the current reference, limited by jitter and noise in the time counting electronics.

The pulsed principle directly measures the round trip time between light pulse emission and the return of the pulse echo resulting from its backscattering from a target object. Thus, pulses need to be as short as possible (usually a few nanoseconds) with fast rise and fall times and large optical power. However, once a light pulse is emitted and reflected onto an object, only a fraction of the optical energy may be received back at the detector. Assuming the target is an optical diffuser (which is the most usual situation), this energy is further divided among multiple scattering directions. Thus, pulsed methods need very sensitive detectors working at high frequencies to detect the weak pulses received.

The advantages of the pulsed approach include a simple measurement principle based

on direct measurement of time-of-flight and the limited influence of background illumination due to the use of high energy pulses.

However, it is limited by the signal-to-noise ratio (SNR) of the measurement, where intense light pulses are required while eye-safety limits need be kept. Moreover, very sensitive detectors are necessary, which may be expensive depending on the detection range.

The pulsed approach is, despite these limitations, the one most frequently selected at present in the different alternatives presented by manufacturers of LiDAR imaging systems for autonomous vehicles, due to its simplicity and its capability to function properly in many environments.

Now, let us focus on the indirect pulsed time-of-flight.

A particular type of indirect ToF, called range gating, can be used to determine the distance travelled by a light pulse that has been transmitted and subsequently reflected by a target object. This method is particularly effective because by combining it, an at least partially simultaneous spot pattern projection and a low power semiconductor light source, a substantially miniaturized, full solid state and energy-efficient long-range distance detection method can be implemented. Range gated imagers integrate the detected power of the reflection of the emitted pulse for the duration of the pulse. The amount of temporal overlap between the pulse emission window and the arrival of the reflected pulse depends on the return time of the light pulse, thus, on the distance travelled by the pulse.

In particular, such a system [9] can be schematically represented as follows.

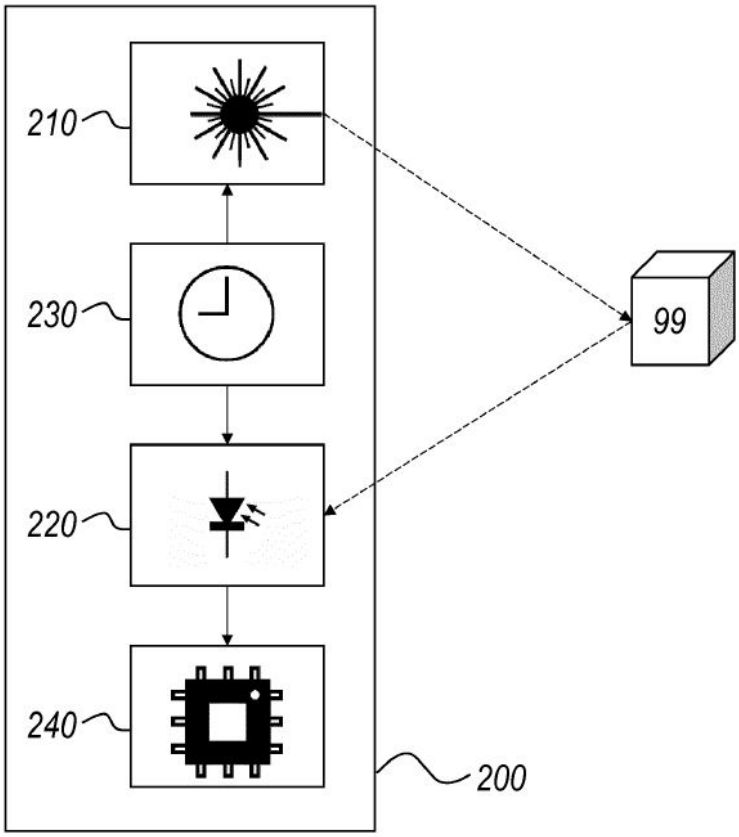


Figure 2.3: The measurement system

The system comprises a solid state light source (210 in figure 2.3) for projecting a pattern of a sequence of spots, which may be repeated periodically, onto the object (99 in figure 2.3). A detector (220 in figure 2.3) is arranged near the light source and configured to detect light reflected by the object.

The light beam bouncing off the object is illustrated as an arrow in dashed lines, travelling from the light source to the object and back to the detector. It should be noted that this representation is strictly schematic, and not intended to be indicative of any actual relative distances or angles.

A synchronization (230 in figure 2.3), which may include a conventional clock circuit or oscillator, is configured to operate the solid state light source so as to project the pattern of spots onto the object during a first predetermined time window and to operate the detector so as to detect a first amount of light representing the reflected light by the object at substantially the same time. It further operates the detector to

detect a second amount of light representing the reflected light by the object during the subsequent second predetermined time window. These two time windows are preferably of the same duration, to facilitate noise and ambient light cancellation by subtracting one of the detected amounts from the other one.

Appropriate processing (240 in figure 2.3) are configured to calculate the distance to the object as a function of the first and second amount of reflected light.

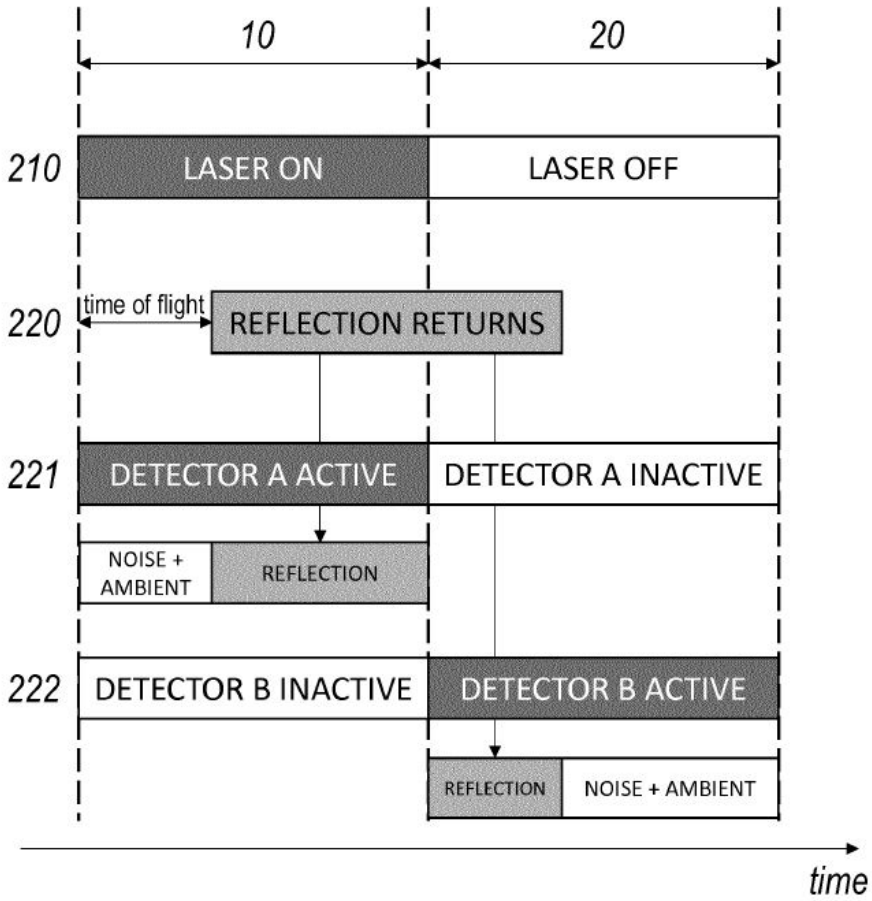


Figure 2.4: Timing diagram for light projection and detection

In order to better understand how this system works let us analyze figure 2.4, the timing diagram for light projection and detection.

During the first time window (10 in figure 2.4), the solid state light source is in its ON state, emitting the pattern of light spots into the scenery. During the second time window (20 in figure 2.4), the solid state light source is in its OFF state.

The arrival of the reflected light at the detector is delayed relative to the start of the projection by an amount of time that is proportional to the distance travelled (few ns/m in free space). Due to this delay, only a part of the reflected light will be detected at the first well of the detector, which is only activated during the first time window. Thus, the charge accumulated in this first well during its period of activation (the first time window) consists of a part representing only the noise and the ambient light impinging on the pixel prior to the arrival of the reflected pulse, and a part representing the noise, the ambient light and the leading edge of the reflected pulse.

The latter part of the reflected pulse will be detected at the second well of the detector, which is only activated during the second time window (which follows the first one). Thus, the charge accumulated in this second well during its period of activation consists of a part representing the noise, the ambient light and the travelling edge of the reflected pulse, and a part representing only the noise and the ambient light impinging on the pixel after the arrival of the reflected pulse.

The greater the distance between the reflecting object and the system, the smaller the proportion of the pulse that will be detected in the first well and the larger the proportion of the pulse that will be detected in the second well.

It is important to highlight that the term "well" designates a storage provided in the semiconductor substrate, e.g. a capacitor, that stores electrical charges generated by the conversion of photons impinging on the pixel. An advantage of charge accumulation at well level is that read-out noise is minimized, leading to a better signal-to-noise ratio.

To conclude, the distance from the target object is calculated as follows:

$$d = c\frac{t_{ON}}{2}\frac{Q_B}{Q_A + Q_B} \tag{2.2}$$

where c is, again, the speed of light in free space, t_{ON} is the duration of the pulse

(LASER ON in figure 2.4) and Q_A and Q_B are the charges stored in the two storage wells.

2.2.2 Amplitude Modulated Continuous Wave (AMCW) approach

The AMCW approach consists of using the intensity modulation of a continuous light wave instead of laser pulses mentioned before.

This method exploits the phase shift induced in an intensity-modulated periodic signal in its round-trip to the target, in order to obtain the range value [10]. The optical power is modulated with a constant frequency f_M, so the emitted beam is a sinusoidal or square wave of frequency f_M. After reflection from the target, a detector collects the received light signal. The distance value d is deduced from the phase shift $\Delta\Phi$ occurring between the reflected and the emitted wave, according to the following formula:

$$\Delta\Phi = k_M d_{tot} = \frac{2\pi f_M}{c} 2d \implies d = \frac{c}{2}\frac{\Delta\Phi}{2\pi f_M} \tag{2.3}$$

where d and c are, again, the distance to the target and the speed of light in free space; k_M is the wavenumber associated to the modulation frequency, d_{tot} is the total distance travelled and f_M is the modulation frequency of the amplitude of the signal.

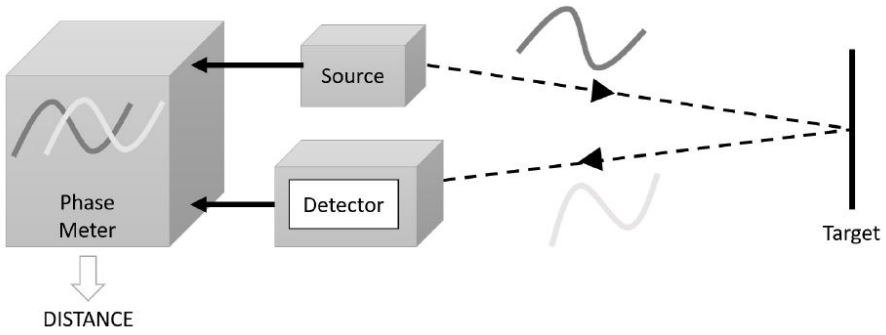


Figure 2.5: Phase measurement principle used in AMCW

Figure 2.5 shows the operation scheme of a conventional AMCW sensor.

There are a number of techniques that may be used to demodulate the received signal and to extract the phase information from it. For example, phase measurement may be obtained via signal processing techniques using mixers and low-pass filters.

In the AMCW approach, the resolution is determined by both the frequency f_M and the resolution of the phase meter fixed by the electronics. Specifically, increasing f_M, the resolution increases too. However, larger f_M frequencies bring on shorter unambiguous range measurements, meaning the phase value of the return signal at different range values starts to repeat itself after a 2π phase displacement. Thus, a significant trade-off appears between the maximum non-ambiguous range and the resolution of the measurement.

AMCW cameras have been commercialized since the 90s and are usually implemented as parallel arrays of emitters and detectors. Furthermore, AMCW modulation is usually implemented on LEDs rather than lasers, which further limits the available power and thus the SNR of the signal. This SNR limitation restricts their applications outdoors, although they show excellent performance in indoor environments, specially for large objects. They have been used inside vehicles in different applications, like passenger or driver detection and vehicle interfacing.

2.2.3 Frequency Modulated Continuous Wave (FMCW) approach

In the case of the FMCW approach, the source is normally a laser diode that enables coherent detection. The signal is sent to the target, and the reflected signal that arrives at the receiver, after a traveled time ToF, is mixed with a reference signal built from the emitter output [10].

For a static target, the delay between the collected light and the reference causes a constant frequency difference f_r, also called beat frequency. Letting the instantaneous

frequency vary under a linear law, f_r results directly proportional to ToF, hence, proportional to the target range too.

$$f_r = slope \cdot \Delta\tau = \frac{B}{T} ToF = \frac{B}{T}\frac{2d}{c} \implies d = f_r \frac{cT}{2B} \tag{2.4}$$

where B is the bandwidth of the frequency sweep, T denotes the period of the ramp, and $\Delta\tau$ equals the total travelled time ToF.

Figure 2.6 depicts all these fundamental parameters.

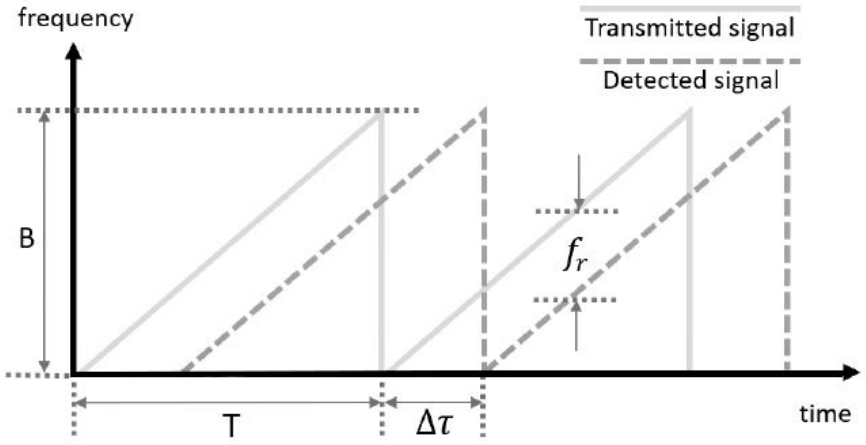


Figure 2.6: Frequency modulation and detection in FMCW

In practice, the frequency difference between outgoing and incoming components is translated into a periodic phase difference, which causes an alternating constructive and destructive interference pattern at the frequency f_r. By using FFT to transform the beat signal from time domain to frequency domain, the peak of beat frequency is easily translated into distance.

Usually, a triangular frequency modulation is used rather than a ramp, as shown in figure 2.7.

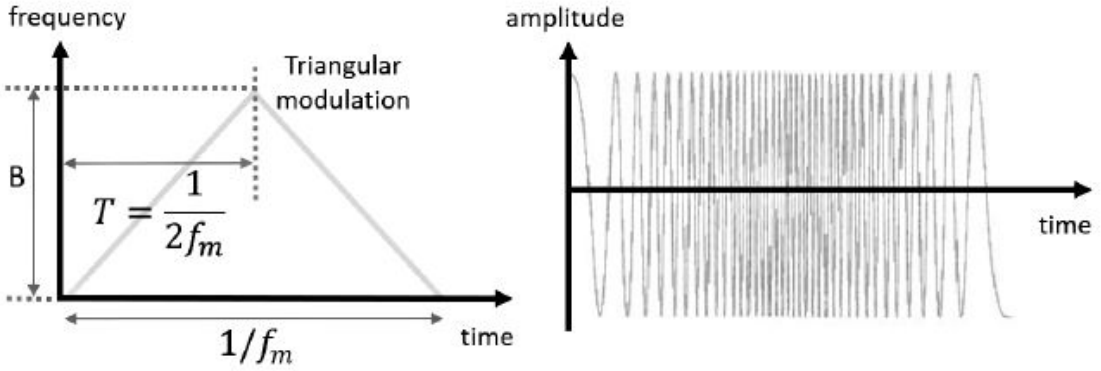


Figure 2.7: Triangular frequency modulation FMCW

The modulation frequency in this case is denoted as f_m. Hence, the rate of frequency change can be expressed as $2f_mB$, while the resulting beat frequency is given by:

$$f_r = \frac{4df_mB}{c} \tag{2.5}$$

This type of detection has the very relevant advantage of measuring not only the range but also the relative velocity of the target and its sign, using the same signal. If the target moves, the beat frequency will be related not only to d but also to the velocity of the target v_r relative to the sensor. The velocity contribution is taken into account by the Doppler frequency f_d, which will affect the sweep of the beat frequency up or down (figure 2.8), according to the following relations:

$$f^+ = f_r + f_d \quad \text{and} \quad f^- = f_r - f_d \tag{2.6}$$

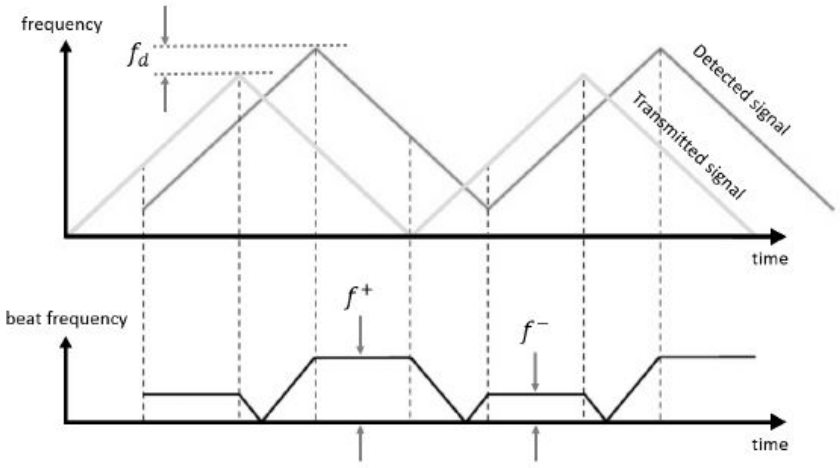


Figure 2.8: Triangular frequency modulation FMCW for a moving target

In this case, the range can be calculated as follows:

$$d = \frac{cT}{4B}(f^+ + f^-) \tag{2.7}$$

while the relative velocity and its direction can also be calculated through the Doppler effect:

$$v_r = \frac{\lambda}{2} f_d = \frac{\lambda}{4}(f^+ + f^-) \tag{2.8}$$

Overall, the FMCW empowers improvements in resolution of range measurements between one and two orders of magnitude when compared to the other methods, and the use of FFT signal processing allows to measure speed of the target simultaneously. Despite these great advantages, this coherent system, to be reliable, must be absolutely stable in its working conditions; aspects like temperature drift or linearity of electronics become extremely important, particularly for applications that demand robustness and need units performing stably for several years.
It is easy to guess that the main benefit in autonomous vehicle applications is its capability to sense simultaneously the speed value and its direction, together with range.

2.3 Illumination strategies

Once the three main measurement strategies used in LiDAR imaging systems have been presented, it is worth noting all of them have been presented as pointwise measurements. However, LiDAR images of interest are always 3D rich point clouds, which achieve accurate representations of more or less large fields of view around the object of interest.

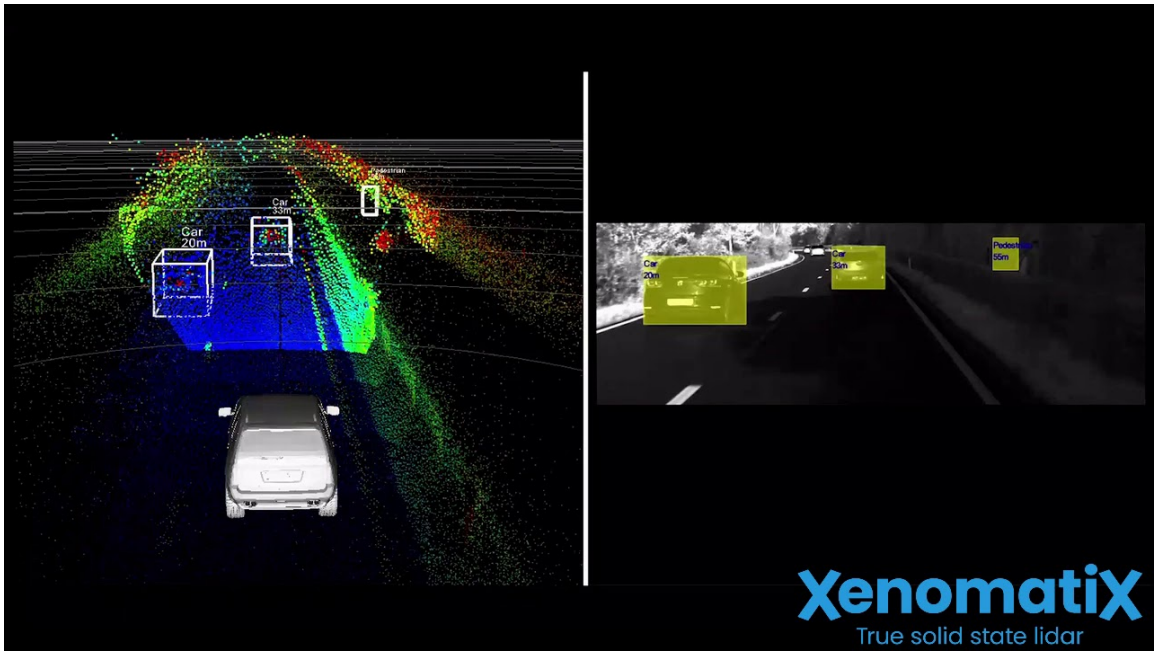


Figure 2.9: Rich point cloud example by XenomatiX

A number of strategies have been proposed in order to build LiDAR images out of the repetition of point measurements, but they can essentially be grouped into three different families: scanning components of different types, detector arrays, and mixed approaches.

Scanning systems are used to sweep a broad number of angular positions of the field of view of interest using some beam steering components, while detector arrays exploit the capabilities of electronic integration of detectors to create an array of receiving elements, each one capturing illumination from separate angular sections of the scene

to deliver a ToF value for each individual detector.
Some of the strategies have also been successfully combined with each other, depending on the measurement approach or requirements.

2.3.1 Scanners

Currently, in the automotive LiDAR market, most of the proposed commercial systems rely on scanners of different types.
In the most general approach, the scanner element is used to re-position the laser spot on the target by modifying the angular direction of the outgoing beam, in order to generate a point cloud of the scene [10]. In autonomous vehicles, three main categories may be found: mechanical scanners, which use rotating mirrors and galvanometric or piezoelectric positioning of mirrors and prisms to perform the scanning, Micro Electro-Mechanical Systems (MEMS) scanners, which use micro mirrors actuated using electromagnetic or piezoelectric actuators to scan the field of view and Optical Phased Arrays (OPAs), which perform pointing of the beam based on a multi-beam interference principle from an array of optical antennas.
It is worth noting that other approaches have been proposed based on alternative working principles, such as liquid crystal waveguides, electrowetting, groups of micro lens arrays and even holographic diffraction gratings.

Mechanical scanners

LiDAR illumination systems based on mechanical scanners use high-grade optics and some kind of rotating or galvanometric assembly, usually with mirrors or prisms attached to mechanical actuators, to cover a wide field of view. These systems consist of units with sources and detectors that jointly rotate around a single axis.

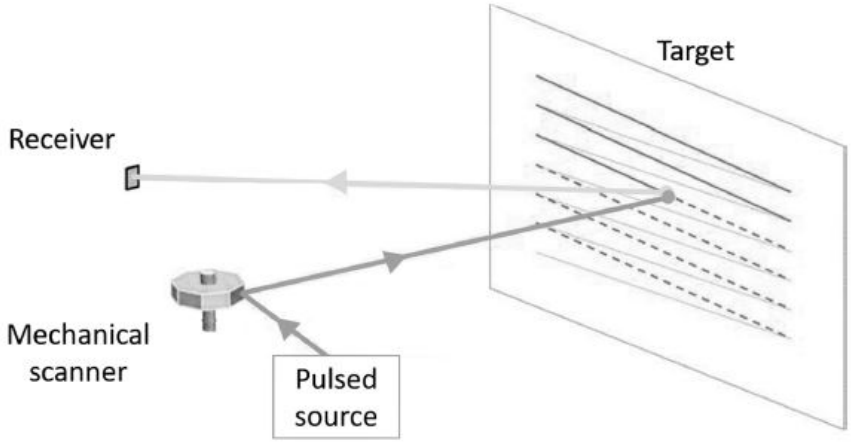


Figure 2.10: Mechanical scanning system scheme

This may be done by sequentially pointing the beam across the target, as depicted in figure 2.10, by rotating the optical configuration around a mechanical axis, where a number of detectors may be placed in parallel along the spinning axis. In the best case, 360 degrees FOVs of the sensor may be achieved, covering the whole surroundings of the vehicle.

This is the most popular scanning solution for many commercial LiDAR sensors, as it provides straight and parallel scan lines with a uniform scanning speed over a vast FOV, or angularly equispaced concentric data lines. They can achieve large spatial resolution in the direction of turn (usually horizontal) although they become limited in the orthogonal direction (usually vertical) where the density of the point cloud is limited by the number of available sources and detectors measuring in parallel.

Despite the current prevalence of this type of scanners due to their simple and efficient arrangement and high efficiency in long-range applications, there are numerous disadvantages: the question of reliability and maintenance of the mechanisms, the mass and inertia of the scanning unit which limits the scanning speed, the lack of flexibility of the scanning patterns and the issue of being misalignment-prone under shock and vibration, beyond being power-hungry, hardly scalable, bulky and expensive. However, the advantages presented make

these systems the sensors of choice for autonomous vehicle research and development, such as algorithms training or robotaxis. Although several improvements are being introduced, there is a quite general agreement that mechanically scanning LiDAR needs to move towards a solid state version.

Micro Electro-Mechanical Systems (MEMS) scanners

MEMS-based LiDAR scanners enable control of laser beam position using tiny mirrors with only a few millimeters in diameter, whose tilt angle varies when applying a stimulus, so that the angular direction of the incident beam is modified and the light beam is directed to a specific point in the scene. Various actuation technologies are developed including electrostatic, magnetic, thermal and piezoelectric. Depending on the applications and the required performance (scanning angle, scanning speed, power dissipation or packaging compatibility), one or another technology is chosen. The most common stimulus in LiDAR applications based on MEMS scanners is the voltage: the mirrors are steered by drive voltages generated from a digital representation of the scan pattern stored in a memory.

Thus, MEMS scanners substitute macroscopic mechanical scanning hardware with an electromechanical equivalent reduced in size. A reduced FOV is obtained compared to the previously described rotary scanners because they have no rotating mechanical components. However, using multiple channels and fusing their data allow us to create FOVs and point cloud densities able to compare with or improve mechanical LiDAR scanners.

MEMS scanners typically have resonance frequencies well above those of the vehicle, enhancing maintenance and robustness aspects, even if in the automotive industry there are many doubts regarding the reliability and durability of this kind of technology (e.g. it is difficult to understand if the mirror is damaged or if it is breaking; it is not possible to receive any feedback of this type).

MEMS scanning mirrors are categorized into two classes according to their op-

erating mechanical mode: resonant and non-resonant. On one hand, the former provide a large scan angle at a high frequency and a relatively simple control design. However, the scan trajectory is sinusoidal, i.e. the scan speed is not uniform. Moreover, their design needs to strike a balance where the combination of scan angle, resonance frequency and mirror size is combined for the desired resolution, while still keeping the mirror optically flat to avoid additional image distortions which may affect the accuracy of the scan pattern. In addition, laser power handling at the surface of the mirror is also an issue that needs be carefully taken into account to avoid mirror damage.

On the other hand, the latter (also called quasi-static MEMS mirrors) provide a large degree of freedom in the trajectory design. Although a rather complex controller is required to keep the scan quality, desirable scanning trajectories with constant scan speed at large scan ranges can be generated by an appropriate controller design. Unfortunately, one key spec, such as the scanning angle, is quite limited in this family compared to resonant MEMS mirrors.

The spot projector can be implemented either with a single mirror with two oscillation axes or using two separate, orthogonal mirrors oscillating each along one axis. Single-axis scanners are simpler to design and fabricate, and are also more robust to vibration and shock; however, dual-axis scanners provide important optical and packaging advantages, essentially related to the simplicity of the optical arrangement and accuracy required in the relative alignment of the two single-axis mirrors. One crucial difficulty with dual-axis scanners is the crosstalk between the two axes.

The most common system architecture is raster scanning, where a low frequency, linear vertical scan (quasi-static) is paired with an orthogonal high frequency, resonant horizontal scan.

Due to its promising advantages (in particular being lightweight, compact and with low power consumption) MEMS-based scanners for LiDAR have received

increasing interest for their use in automotive applications. These systems have shown in parallel the feasibility of the technology in different scenarios, such as space applications and robotics.

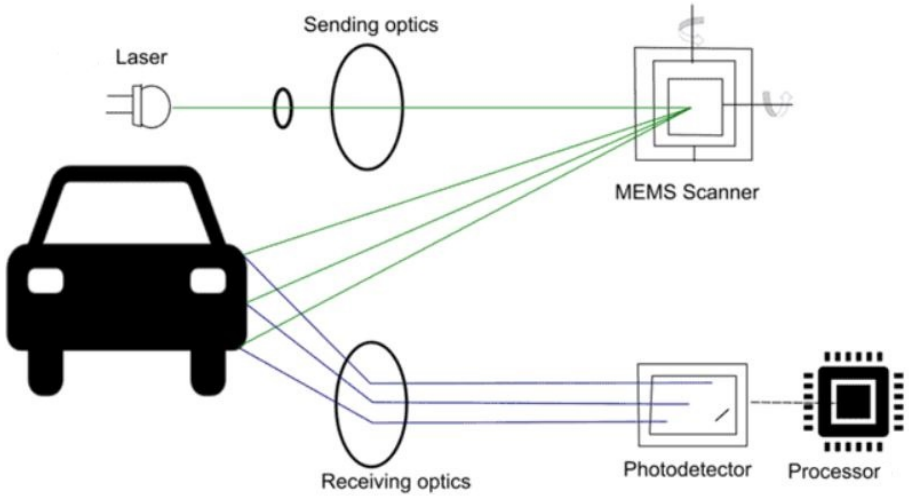


Figure 2.11: MEMS scanning system scheme

Optical Phased Arrays (OPAs)

An Optical Phased Array (OPA) is a novel type of solid state device that allows to steer the beam using a multiplicity of micro-structured waveguides. The OPA principle is similar to phased array radar. By aligning the phases of several coherent emitters, the emitted light interferes constructively in the far-field at certain angles enabling to steer the beam. In an OPA device, an optical phase modulator controls the speed of light passing through the device. Regulating the speed of light enables control of the optical wave-front shape [11], as shown in figure 2.12.

For instance, the top beam is not delayed, while the middle and bottom beams are delayed by increasing amounts. This phenomenon effectively allows the deflection of a light beam, steering it in different directions.

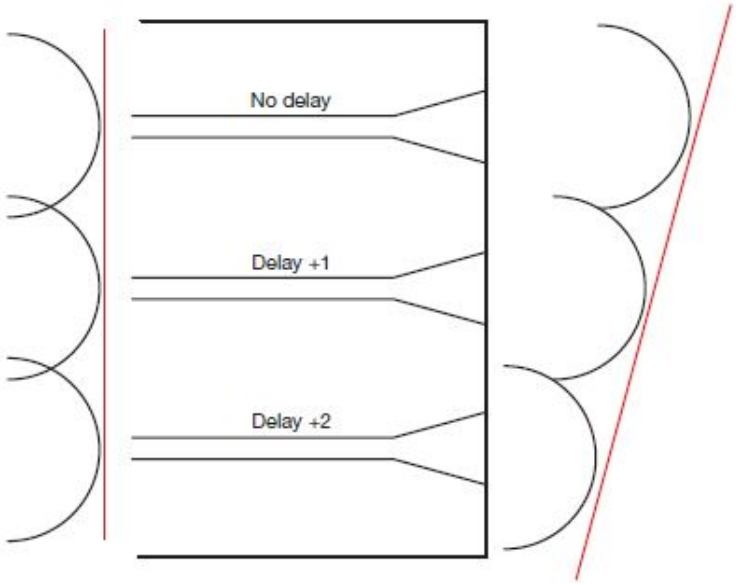


Figure 2.12: OPA working principle

OPAs can achieve very stable, rapid, and precise beam steering. Since there are no mechanical moving parts at all, they are robust and insensitive to external constraints such as acceleration, allowing extremely high scanning speeds over large angles. Moreover, they are highly compact and can be stored in a single chip. OPAs have gained interest in recent years as an alternative to traditional mechanical beam steering or MEMS-based techniques because they completely lack inertia, which limits the ability to reach a large steering range at high speed.

However, the insertion loss of the laser power is still a drawback, as it is their current ability to handle the large power densities required for long-range LiDAR imaging.

As a developing technology with high potential, the interests on OPA for automotive LiDAR is growing in academia and industry, even though it is still under test for long-range LiDAR. However, OPAs are operative in some commercially available units targeting shorter and mid ranges.

2.3.2 Detector arrays

Due to the lack of popularity in the automotive market of scanning LiDAR approaches based on moving elements, alternative imaging methods have been proposed to overcome their limitations beyond MEMS scanners and OPAs.

These scannerless techniques typically combine specialized illumination strategies with arrays of receivers: transmitting optical elements illuminate a whole scene while a linear array (or matrix) of detectors receives the signals of separate angular subsections in parallel, allowing to obtain range data of the target in a single-shot (figure 2.13) making it easy to manage real-time applications.

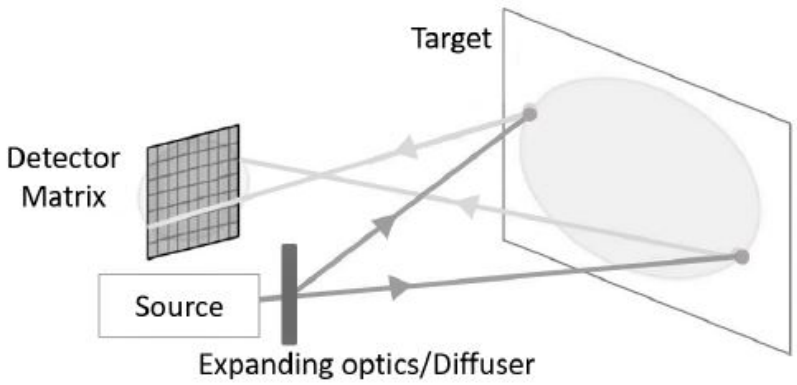


Figure 2.13: Detector arrays working principle

The illumination may be pulsed (flash or multi-beam) or continuous (AMCW or FMCW LiDARs). With the exception of FMCW LiDARs, where coherent detection enables longer ranges, flash imagers or imagers based on AMCW principle (ToF cameras) are limited to medium/short ranges. In flash LiDARs the emitted light pulse is dispersed in all directions, significantly reducing the SNR, while in ToF cameras the phase ambiguity effect limits the measured ranges to a few meters. Multi-beam LiDARs, instead, offer longer ranges and lower power consumption than flash ones. A brief description of the basic working principles is provided next.

Flash

One very successful architecture for LiDAR imaging systems in autonomous vehicles is flash LiDAR, which has progressed to a point where it is very close to commercial deployment in short and medium-range systems.

The principle of solid state flash LiDAR is to illuminate the FOV with a single flash from a laser source, then detect the backscattered light by an array of photodetectors. So, flash LiDAR operation is very similar to that of a standard digital camera using an optical flash.

In flash LiDAR, a single large-area laser pulse illuminates the target environment in front of it and appropriate expanding optics are employed to broaden the beam to cover the scene of interest. The backscattered light is collected by the receiver which is divided among multiple detectors, as shown in figure 2.13. Each detector in the array is individually triggered by the arrival of a pulse return, and measures both its intensity and the range, using the conventional time-of-flight principle. Hence, both the optical power imaged onto a 2D array of detectors and the 3D point cloud are directly obtained with a single laser beam on the target.

Since the light intensity from the transmitter is dispersed with a relatively large angle to cover the full scene, and such a value is limited by eye-safety considerations, the measurement distance is dependent on sensing configurations, including aspects like emitted power, sensor and detector type and sensitivity. It can vary from tenths of meters to medium distances, although at present in automotive they are typically used in the 20 m to 60 m range, also and above all for reasons related to eye-safety regulations. The divergence of the illuminating area and the backscattering at the target significantly reduce the amount of optical power available, so very high peak illumination power and very sensitive detectors are required in comparison to scanners. This has caused the present flash setups to keep being concentrated in sensing at medium or short-range

applications in autonomous vehicles, where they take advantage of their lack of moving elements, and they have acceptable costs in mass production due to the simplicity of the setup.

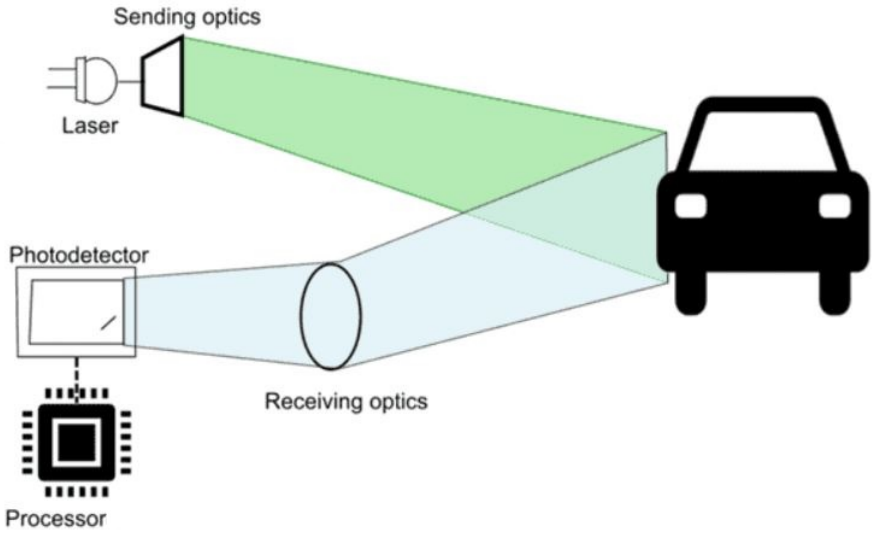


Figure 2.14: Flash working principle

The presence of retro-reflectors in the real-world environment is a significant problem to these cameras. In roads and highways, for instance, retro-reflectors are commonly used in traffic signs and license plates. In practice, retro-reflectors overflow the SPAD (Single Photon Avalanche Diodes) detector with photons, saturating it, and blinding the entire sensor, rendering it useless. Some schemes based on interference have been proposed to avoid such problems. Issues related to mutual interference of adjacent LiDARs, where one LiDAR detects the illumination pattern of the other, are also expected to be a hard problem to solve in flash imagers.

On the positive side, since flash LiDARs capture the entire scene in a single image, the data capture rate can be very fast if compared to mechanical laser scanning; in addition, the method is very resilient to vibration effects and movements, which otherwise could distort the image. Other advantages include the elimination of scanning optics and moving elements and potential for creating

a miniaturized system.

This has resulted in the existence of systems based on flash LiDARs effectively being commercialized at present in the automotive market.

Multi-beam

Completely different from the competitive LiDAR landscape, multi-beam technique offers a no-scanning LiDAR, detecting the whole scene in one flash but without the constraints of shorter range or high power. Multi-beam is an innovative, simple and high performant concept, with ranges beyond 200 m and reasonable and manageable power consumption. The high resolution point clouds need no post-treatment for time-space correction like scanning LiDARs, allow for a much higher frame rate and correct much more easily for one laser fall-out.

Remember that flash LiDARs are largely used for applications that require maximum 20 to 60 meters range; multi-beam approach solves this limitation concentrating the emitted photons only in the spot that would be measured, ensuring high resolution and reaching long ranges required at highway speeds. The only limit to reach farther distances is dictated by eye-safety regulations. For applications that do not require a laser Class 1 classification, multi-beam can be effective at 500 m and more.

In this context, VCSEL arrays become a good candidate source for such systems. VCSELs are a relatively recent type of semiconductor lasers with short pulse widths and small beam divergence. In a short time they have gained the reputation as the perfect choice for solid state LiDAR. Specifically, VCSELs and VCSEL arrays have recently become the near infrared illuminator sources of choice for 3D detection systems in mobile devices. Interest in the use of VCSELs and associated arrays for automotive applications has grown due to the versatility and economy associated with VCSEL devices. Since VCSELs are grown, processed and tested while still in the wafer form, there is significant

economy of scale resulting from the ability to conduct parallel device processing, while set up times and labor content are minimized. Additionally, they benefit from scalability of manufacturing and adaptability of integration into electronic packaging.

Vertical-Cavity Surface-Emitting Lasers (VCSELs) are so called surface emitters in which the light is emitted perpendicular to the chip's surface to allow the beam to be easily collimated by means of etched micro-lenses. The light oscillates perpendicular to the semiconductor layers and escapes through the top or bottom of the device. The extremely short rise times enable fast pulse sequences in the low nanosecond range and below.

Purely by way of example, figure 2.15 shows a VCSEL illuminator module providing high pulsed-optical power with narrow beam divergence in a compact, surface-mountable module.

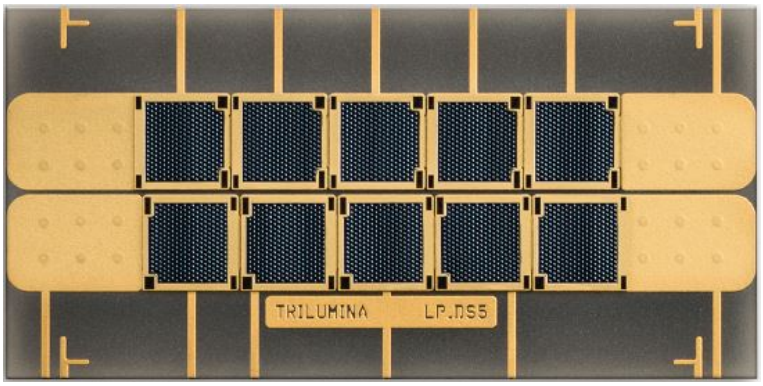
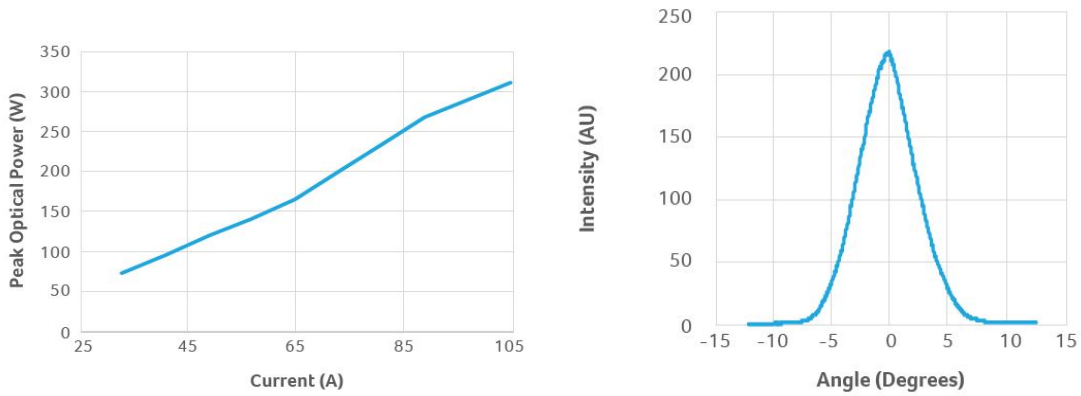


Figure 2.15: VCSEL illuminator module

This module has two channels that can be independently driven, each with five connected VCSEL arrays. Every VCSEL array contains hundreds laser elements connected in parallel. Let us assume to apply 100 A of pulsed current simultaneously to both channels, this package can produce illumination with a far-field divergence angle of less than 15 degrees at an optical power of 300 W [12].



(a) Typical peak power as a function of current per channel

(b) Typical beam divergence at 300 W

Figure 2.16: Typical electro-optical characteristics

The beams of the VCSEL elements combine incoherently in the far field, creating a powerful, circular combined beam with low speckle.

Micro-lenses, monolithically integrated directly into substrate, maintain low beam divergence without costly external optics.

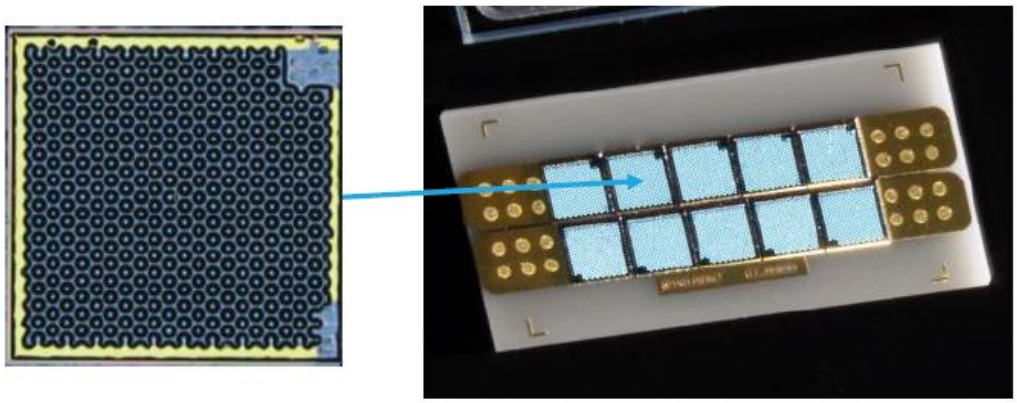


Figure 2.17: VCSEL chips with etched micro-lenses

In this type of module, the VCSEL arrays are flip-chip bonded to a ceramic sub-mount that has metal vias connecting to the back side, for surface mounting the module directly on a board.

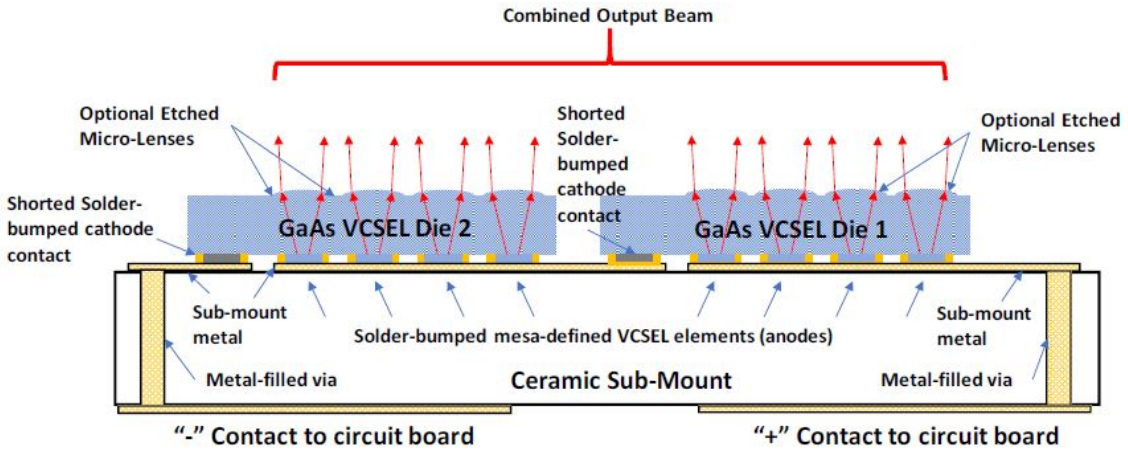


Figure 2.18: Cross-section of series-connected VCSELs

This packaging architecture eliminates wire bonds, drastically reducing parasitic losses and optimizing transmission of pulsed current to the laser arrays. In addition, the flip-chip configuration and the sub-mount's high thermal conductivity enhance heat dissipation.

Each individual VCSEL element is considered a point source while the entire array acts as an extended source, thus enhancing eye safety. 1500 VCSELs distributed across a broad area allow a higher maximum permissible eye exposure limit than would an equivalent point source.

All these characteristics make VCSELs better suited to a wide range of applications than conventional edge-emitting diode lasers and LEDs.

Leading market research institutes forecast rapid growth for the global VCSEL market until 2030.

In this context, it is worth recalling that some successful proposals of LiDAR imagers have mixed the two imaging modalities presented above, that is, they have combined some scanning approach together with some multiple detector arrangement. In conclusion, a summary of the key points described is provided in table 2.1.

	Mechanical scanners	**MEMS scanners**	**OPAs**	**Flash**	**Multi-beam**
Working principle	Galvos, rotating mirrors or prisms	MEMS micromirrors	Phased array of antennas	Pulsed illumination	Pulsed illumination
Main advantage	360 deg FOV in horizontal	Compact and lightweight	Full solid state	Fast frame rate	High resolution
Main disadvantage	Moving elements, bulky	Laser power management	Lab-only for long-range	Limited range, blindable	Short laser pulses management

Table 2.1: Summary of illumination strategies

2.4 An example of LEVEL 3 technology: Traffic Jam Chauffeur

Tailbacks and slowdowns caused by heavy traffic are one of the most common problems in cities all over the world. Slow moving and traffic require drivers' full and complete attention. In this context, the slightest distraction can cause a collision, albeit often at low speed.

These situations become much less problematic on board cars equipped with Traffic Jam Chauffeur technology, which can achieve Level 3 automated driving in congested traffic at speeds of up to 60 km/h, freeing drivers from traffic problems. With traffic jam pilot engaged, drivers no longer need to continuously monitor the vehicle and the road. However, they must remain alert and capable of taking over the task of driving when the system prompts them to do so. When the driver activates the functionality, the system takes over the driving task in slow-moving traffic jam managing starting, accelerating, steering and braking. Thanks to its extensive sensor sets, the Traffic Jam Chauffeur is also able to handle demanding situations, like vehicles cutting across the lane. In this way, the vehicle follows the lane and adjusts speed considering various factors such as keeping a safe distance to the vehicle in front or following the

speed limit; if a preceding slower vehicle is detected the car overtakes automatically as soon as it is safely possible.

When the optional is activated, the drivers can take their foot off the accelerator and their hands off the wheel. They are no longer required to constantly monitor the vehicle and are able to devote their attention to other activities. They can turn their attention from the traffic and the car's steering to do things like answer their email, write text messages, tend their appointment calendar, read the news, or plan for their vacation.

As soon as the speed rises above 60 km/h or the line of vehicles breaks up, the traffic jam pilot informs the driver that they need to take charge of driving once again. Moreover, in this highly automated driving mode, a camera checks whether the driver is ready to retake control of the vehicle: it analyzes various criteria, including the position and movement of the head as well as monitoring the eyes. If the driver ignores notification and subsequent warnings, the vehicle will brake continuously until it comes to a stop within its lane.

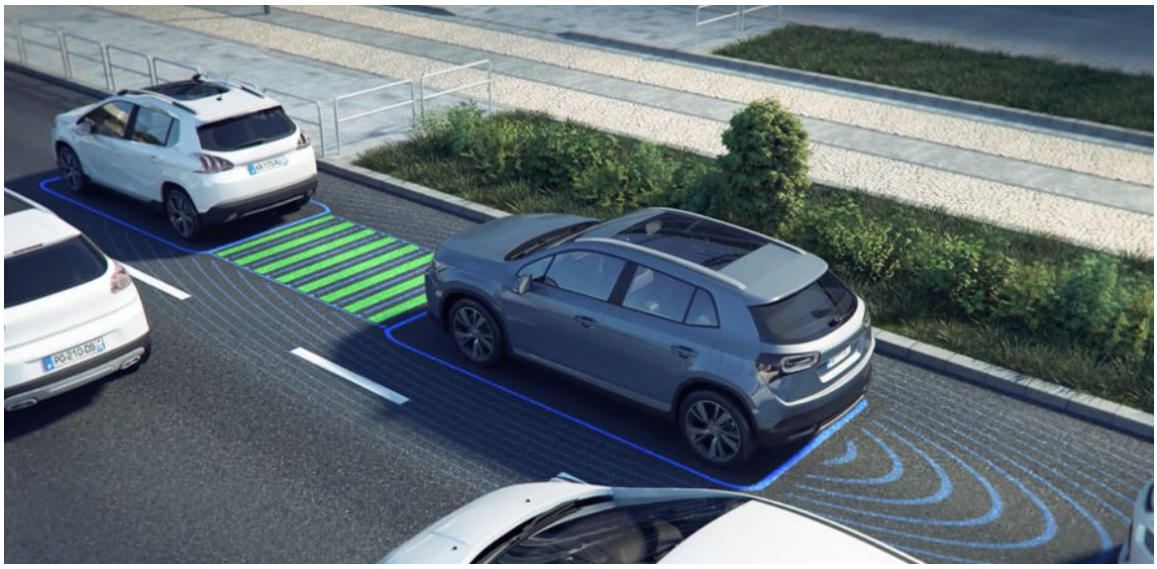


Figure 2.19: Traffic Jam Chauffeur

This technology helps to reduce the number of accidents linked to human errors. By entirely delegating vehicle control, drivers take advantage of their time to do

something else while remaining in a position to take back control. This new innovation brings drivers peace of mind and comfort and prevents the fatigue arising from monotonous driving.
Technically speaking, the basic requirement for use of the Traffic Jam Chauffeur is the highly detailed collection of data about the environment surrounding the car. A complete set of sensor can include:

- ultrasonic sensors on the vehicle's front, sides and rear
- 360 degrees cameras on the vehicle's front, rear and exterior mirrors
- mid and long-range radar sensors for vehicle's corners and front, respectively
- LiDAR sensors on the vehicle's front
- driver observation camera on the top of the instrument panel

Performing more complex ADAS functions requires not only input from more cameras and other sensors such as ultrasound, LiDAR and radar, but also the fusion of data from those different sensor elements. Fusion also enables overcoming the drawbacks of individual sensor solutions and can provide some level of redundancy. The more tasks that driver assistance systems assume along the way to autonomous driving, the greater the number of sensors needed in the vehicle.

2.5 System architecture: from vehicle to LiDAR platform

Sensor systems are becoming increasingly widespread on cars right now, especially with Level 3 autonomous cars right around the corner. The demand for sensors is skyrocketing and will continue to do so. With this demand comes the need for more

sophisticated processors and central sensor fusion units to interpret the vast amount of sensor data now being collected.

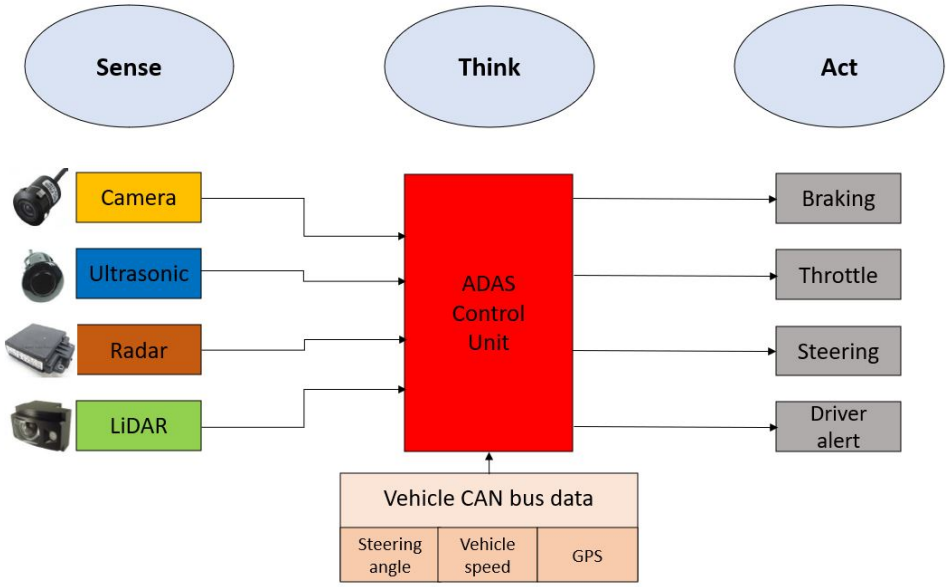


Figure 2.20: Sense-think-act paradigm

Figure 2.20 is a simplified diagram explaining what automated vehicles need to be capable of. In principle, it needs to have the same skills as a human driver, only better. Firstly, it has to be able to perceive and interpret its surroundings and its interior ("Sense").

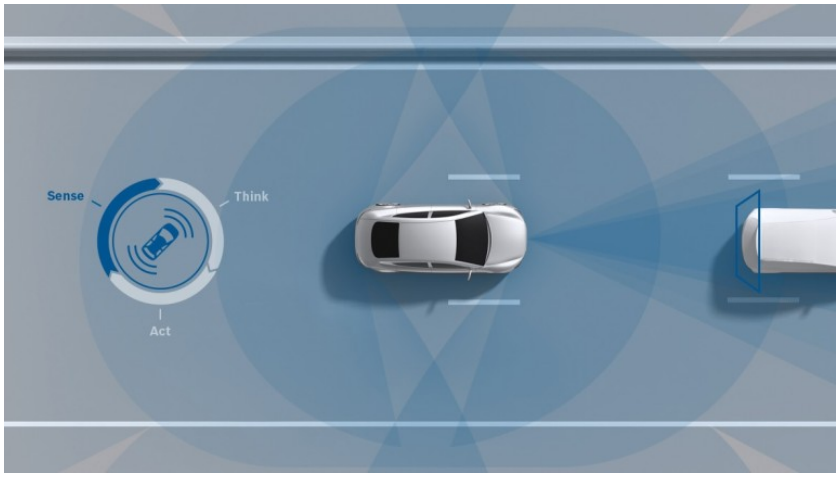


Figure 2.21: Sense

Camera, ultrasonic, radar and LiDAR sensors are the eyes and ears of an automated car and supply all the information required by the vehicle to recognize its entire surroundings. Additional sensors are even aligned inward into the passenger compartment, enabling the system to make autonomous decisions as to whether the driver is capable of taking control of the vehicle again if necessary.
Secondly, the vehicle needs to combine, merge and process all the information and data received in order to forecast and derive a suitable driving strategy ("Think").

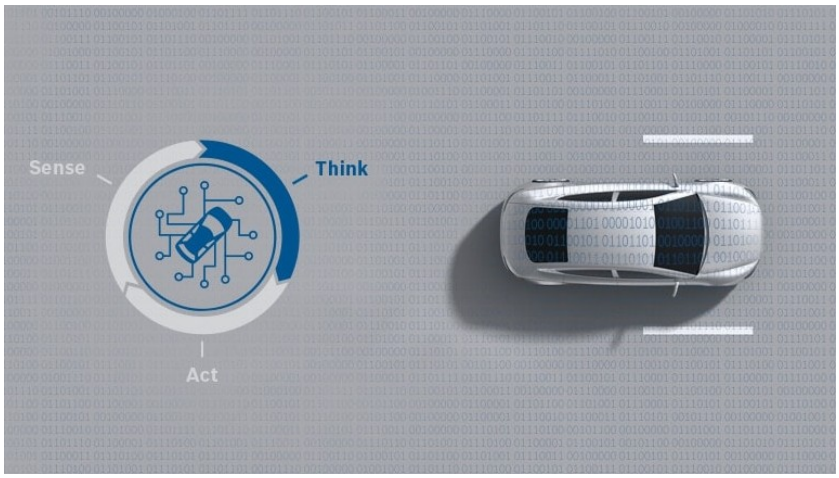


Figure 2.22: Think

This task is performed by the ADAS Control Unit (also known as perception unit), a dedicated module responsible for all the ADAS functions, thanks to the use of software and artificial intelligence algorithms (machine learning and deep learning) which exploit the information collected from sensors (it also has raw data processing capabilities) as well as large amount of data acquired from other connected systems. For instance, it is possible to identify whether an object perceived by the sensors is a stop sign, vehicle, pedestrian, or cyclist. It is furthermore feasible to determine whether the object is moving and, if so, in which direction and at what speed. Based on the interpretation models, it can be also possible to derive the likely future behavior of these static or dynamic objects. Moreover, sensor redundancy and the consequent data fusion increase the measuring range and improve the reliability and

accuracy of the measurements. In fact, let us suppose to have two sensors, camera and radar: under certain conditions, such as poor night visibility or fog, one of the two sensors (camera) loses its reliability and consequently the concept of redundancy fades. This is the reason why LiDAR is necessary.

Automated vehicles need to decide in real time which driving strategy is the best in order to solve the current traffic situation and reach their destinations; top priority in all decisions is given to the safety of all driving maneuvers. The ADAS Control Unit, with its impressive computing power and extensive memory, masters all the automated driving functions. Meeting high security and safety requirements, it collects and merges several technologies for a very precise environment model and calculates highly complex functional algorithms for a safe and dynamic vehicle behavior, even at high speeds. Traditionally, ECUs for individual ADAS applications were distributed around the vehicle depending on their specific function; but these decentralized system architectures are not sufficient for complex driver assistance systems required for higher levels of automated driving. Thus, these modern systems require high computing performing and combine multiple functions into one integrated controller, or ADAS ECU. Typically, this ECU has a small size and a moderate power consumption. Obviously, the perception unit also requires data and information such as the vehicle's speed, positioning and its "manual" steering angle (for example, when the driver assistance systems are temporarily disabled or when the driver takes direct control of the vehicle). This type of information is typically shared through CAN bus, while data collected by camera and LiDAR sensors are managed via LVDS (Low-Voltage Differential Signaling) and Automotive Ethernet interfaces respectively.

Thirdly, the vehicle needs to use its powertrain, steering and braking power to move its wheels in such a way that the planned driving strategy is put into practice ("Act").

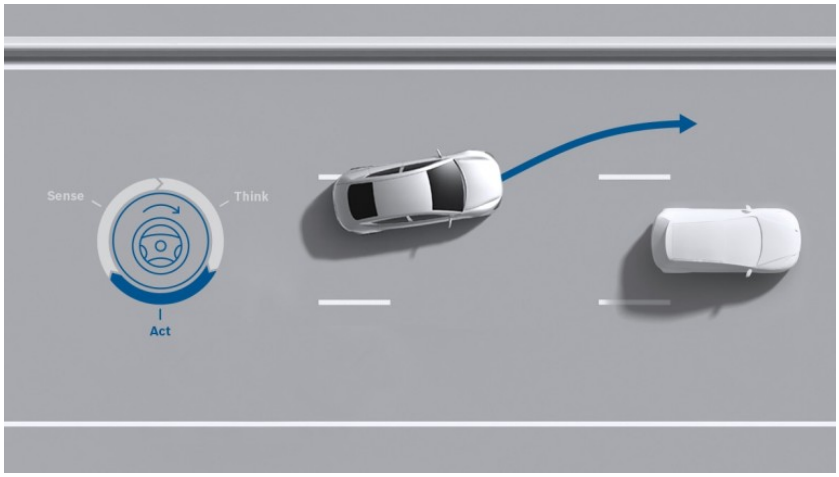
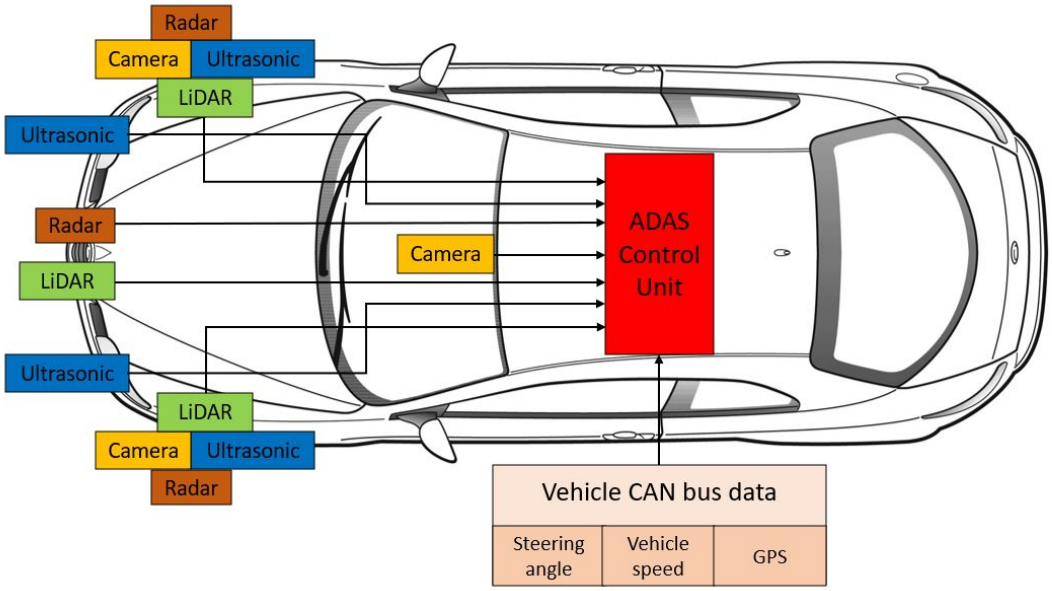
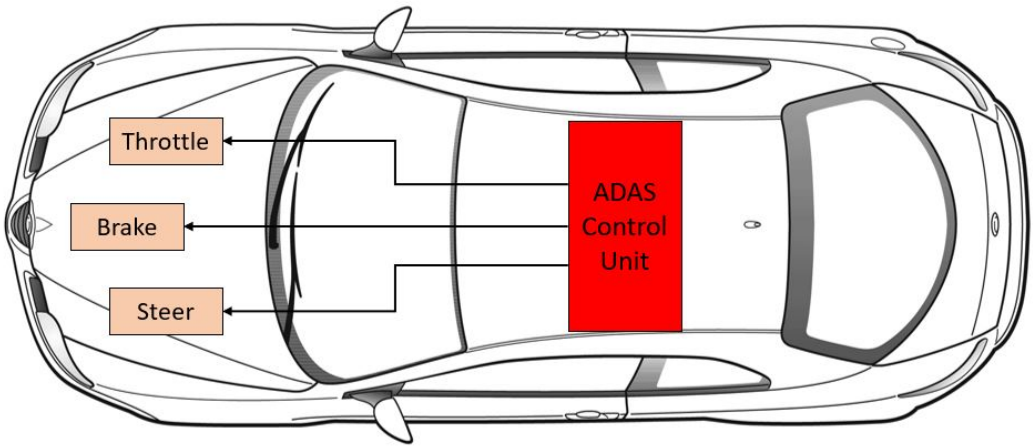


Figure 2.23: Act

The exact driving maneuver are imposed and powered by the ADAS Control Unit through CAN bus. The driving strategy specifies how the automated vehicle has to act on the road and calculates all the required parameters for its behavior. The vehicle must autonomously decide where and when to accelerate, brake and steer, all on the basis of information which can suddenly change. Because the vehicle knows the current road conditions, when there is an icy stretch, for example, it can adapt its driving strategy accordingly and reduce speed for safely crossing the roadway. As it constantly perceives and evaluates its own position and its surroundings, it is also capable of bypassing obstacles or changing lanes as required; and because its software's artificial intelligence learns the characteristic behavior of objects, it can drive proactively and react in good time in critical situations, for example, by activating the brake system when a pedestrian is about to cross the road ahead.



(a) Sense and think



(b) Act

Figure 2.24: Example of system architecture at vehicle level

This process of sensing, thinking and acting, which takes place during the car's entire journey, portrays and outlines the system architecture at vehicle level that is shown in figure 2.24.

Going down to a lower level of detail, it is possible to investigate the structure of the LiDAR platform, characterizing the internal architecture of this sensor which is shown in figure 2.25.

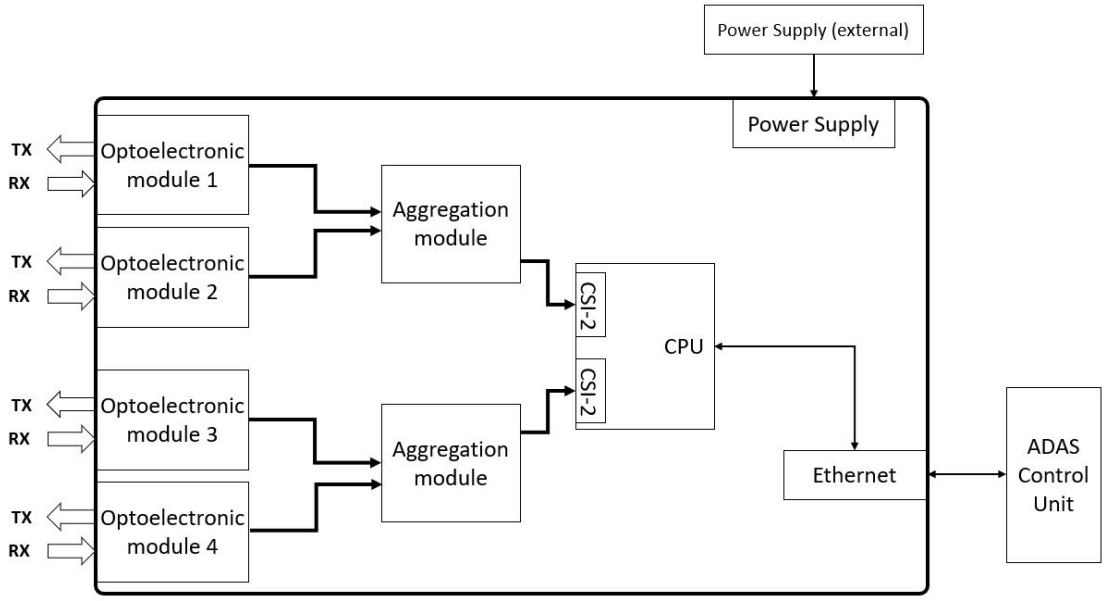


Figure 2.25: Example of LiDAR sensor architecture

Typically, considering the worst case scenario, four solid state optoelectronic modules are necessary (each equipped with a laser transmitter and a laser receiver) in order to obtain and cover the required field of view. Generally, these modules send MIPI CSI-2 data streams to the CPU (it could be a System on a Chip), which must process them to create a rich point cloud with the highest possible resolution. It is important to point out that CPUs must be Automotive Grade, that is why their availability on the market is considerably reduced. Often, however, commercial CPUs only have two CSI-2 ports; hence, the absolute need to aggregate data coming from the different optoelectronic modules to provide a unique output data stream to the CPU (at most two). This goal can be achieved with many different technical hard-

ware solutions (ASIC, Serializer/Deserializer, FPGA).

Moreover, there is a power supply block that takes the power from the vehicle and provides it to all blocks within the LiDAR platform.

The entire communication with the ADAS Control Unit usually takes place via Automotive Ethernet interface.

CHAPTER 3

DESIGN OF A DATA AGGREGATION MODULE FOR LiDAR SENSORS

3.1 MIPI specifications for automotive industry

Nowadays the auto industry is being transformed by several global trends, including a growing embrace of electric vehicles, increasing vehicle automation, tighter safety and fuel economy standards.

In this context, it is more and more common to hear new cars with advanced electronics referred to as “smartphones on wheels” or “mobile data centers.”

Consumers can realize the use of mobile technologies in automotive in ”visible” features such as high resolution front cluster displays connected to back-up rear cameras, infotainment displays with GPS navigation, Bluetooth, Wi-Fi and cellular connectivity.

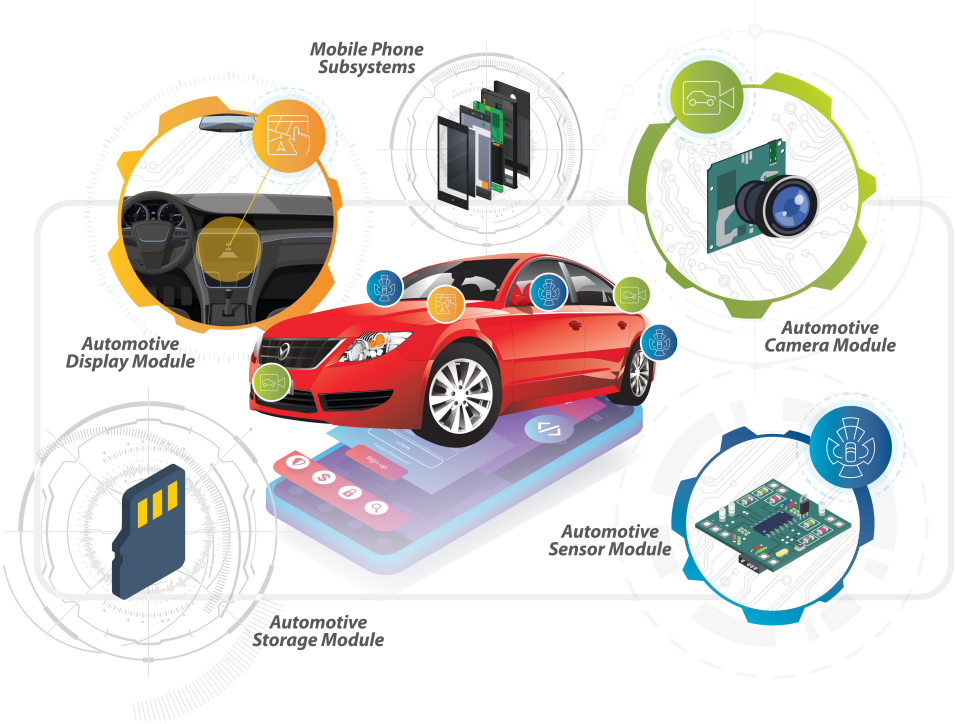


Figure 3.1: Mobile technologies in automotive

Vehicles are becoming smarter, more connected and automated and, as they progress along higher SAE levels of driving automation, they will be enabled by increasingly sophisticated sensor electronics and processing, brought together by high speed interconnects. Ultrasonic sensors, optical cameras, radio-based radars and light-based LiDAR sensors provide a large amount of data at extremely high rates, delivering them to Electronic Control Units by high speed interfaces.

It can be clearly observed from figure 3.2, that annual revenue for all ADAS technologies is predicted to reach more than 65 billion dollars by 2026 [13].

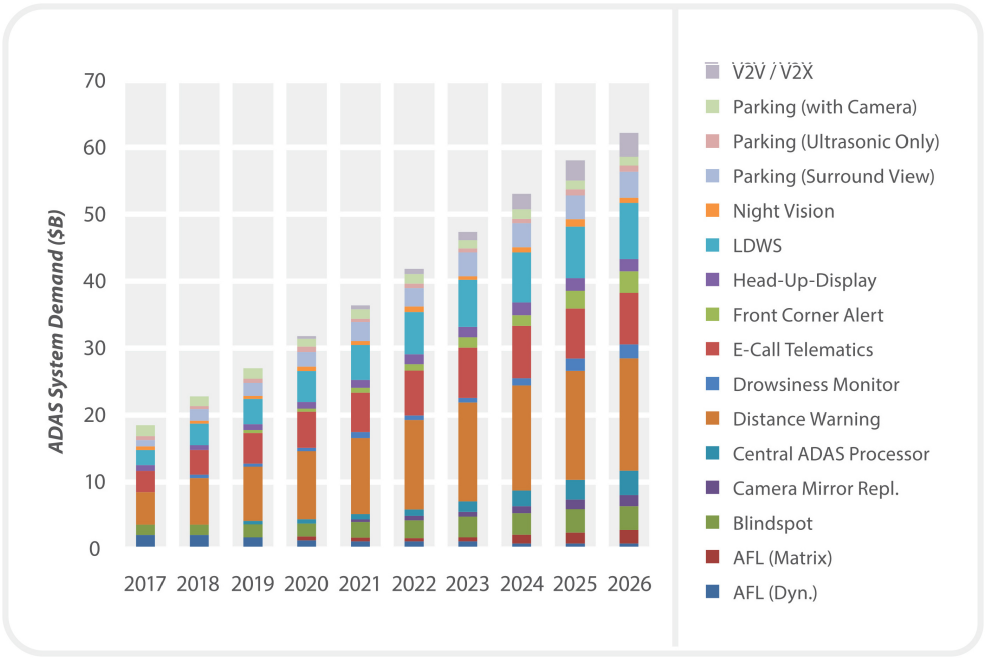


Figure 3.2: Automotive ADAS growth forecast

In this rapidly evolving landscape, while the existing broadly adopted automotive interfaces, such as LIN and CAN, see continued use for lower speed application (mainly control applications), auto manufacturers and suppliers have no clear standardized solutions for high speed interfaces between cameras, LiDARs and ECUs, and for the most part they have had to rely on proprietary solutions. Although these solutions employ good technology, the presence of multiple and fragmented competing solutions creates confusion in the market, and the lack of a single standard limits economies of scale.

In 2015, MIPI Alliance, founded in 2003 by ARM, Nokia, Samsung, STMicroelectronics and Texas Instruments, identified the need for a unified in-vehicle connectivity specifications that would meet the automotive industry's need for high speed and bandwidth, low latency, functional safety, low power consumption, low electromagnetic interference (EMI) and small form factor. MIPI implementations reduce design complexity and cost, simplify integration and accelerate time-to-market. MIPI Alliance specifications have supported companies consolidate their integration approaches while creating their own high level designs that differentiate their products.

Today, the impact of MIPI Alliance is well established: all major chip vendors use MIPI Alliance specifications; the organization now numbers more than 300 members. MIPI Alliance continues to evolve its technology roadmap to deal with mobile influenced markets and help drive growth in these exciting, new ecosystems.
Specifically, part of the MIPI interface specifications are being reused from the mobile ecosystem into automotive industry.
Since its introduction, MIPI Camera Serial Interface 2 (MIPI CSI-2) has been widely adopted in automotive as an interface for cameras, and increasingly for radar and LiDAR sensors. With support from the high bandwidth D-PHY physical layer, CSI-2 supports a wide range of applications, resolutions, frame rates and color depths, with flexible-pin-count PHY configurations. CSI-2, with its raw data format capability, ensures fine image capture even when lighting changes suddenly and dramatically, such as when a vehicle exits a weakly lit tunnel into bright sunlight.
MIPI Alliance has also developed a physical layer (PHY) for short-reach connections to CSI-2-based cameras and LiDARs: MIPI D-PHY. It is based on two-wire differential signaling, standardized and optimized for low power use with the CSI-2 protocol. Its configuration manages up to four data Lanes and one clock-forwarding Lane on a 10-wire interface port, with high performance, high noise immunity and jitter tolerance, and low-latency transitions between high-speed and low-power modes.

3.1.1 MIPI CSI-2 protocol

It is well known that incompatible, proprietary interfaces prevent devices from different manufacturers from working together. This raises system costs and reduce its reliability; moreover, the lack of a clear industry standard slows innovations and inhibits new product market entry.
Camera Serial Interface 2 (MIPI CSI-2) provides the mobile industry a standard, robust, scalable, low power, high speed, cost-effective interface between a peripheral

device (e.g. LiDAR or camera) and a host processor. The CSI-2 specification defines a standard data transmission interface between transmitter and receiver: it is a uni-directional, differential, serial interface with data and clock signals [14]. The most popular physical layer of this interface is the MIPI Alliance Specification for D-PHY.

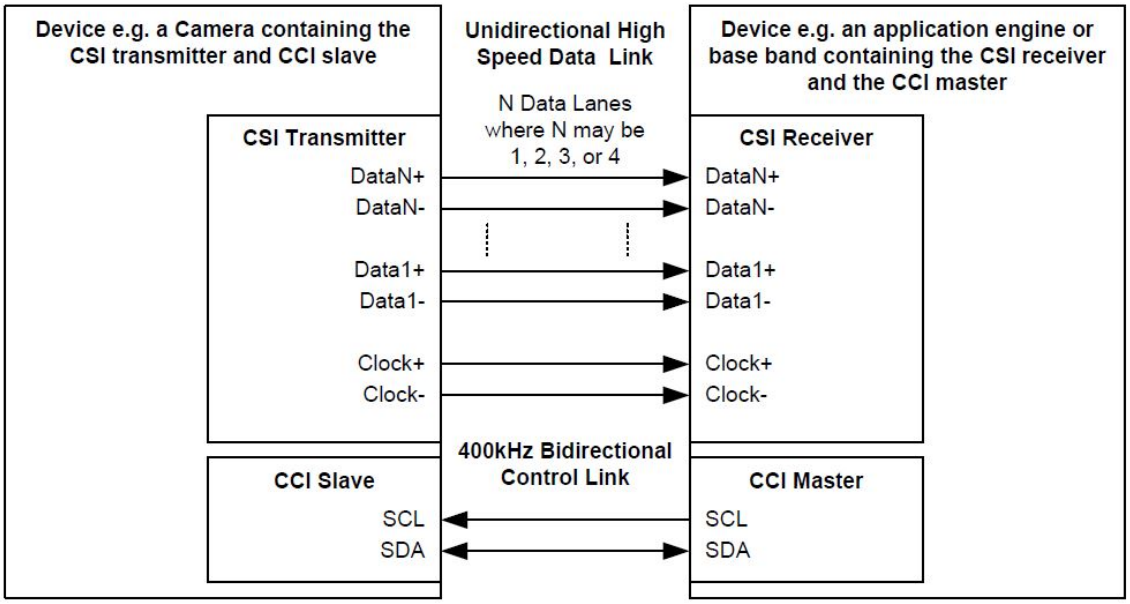


Figure 3.3: CSI-2 and CCI transmitter and receiver interfaces

Figure 3.3 shows the connections between CSI-2 transmitter and receiver described above; in addition, it illustrates the control interface (referred as CCI): a two-wire, bi-directional serial interface compatible with I2C standard for controlling the D-PHY transmitter. CCI shall support 400 kHz operation and 7-bit slave addressing. A CSI-2 receiver shall be configured as a master and a CSI-2 transmitter shall be configured as a slave on the CCI bus. Note that the terms master and slave, when referring to CCI, should not be confused with similar terminology used for D-PHY's operation; they are not related. In fact, a CSI transmitter shall be configured as a master and a CSI receiver as a slave on CSI-2 bus.

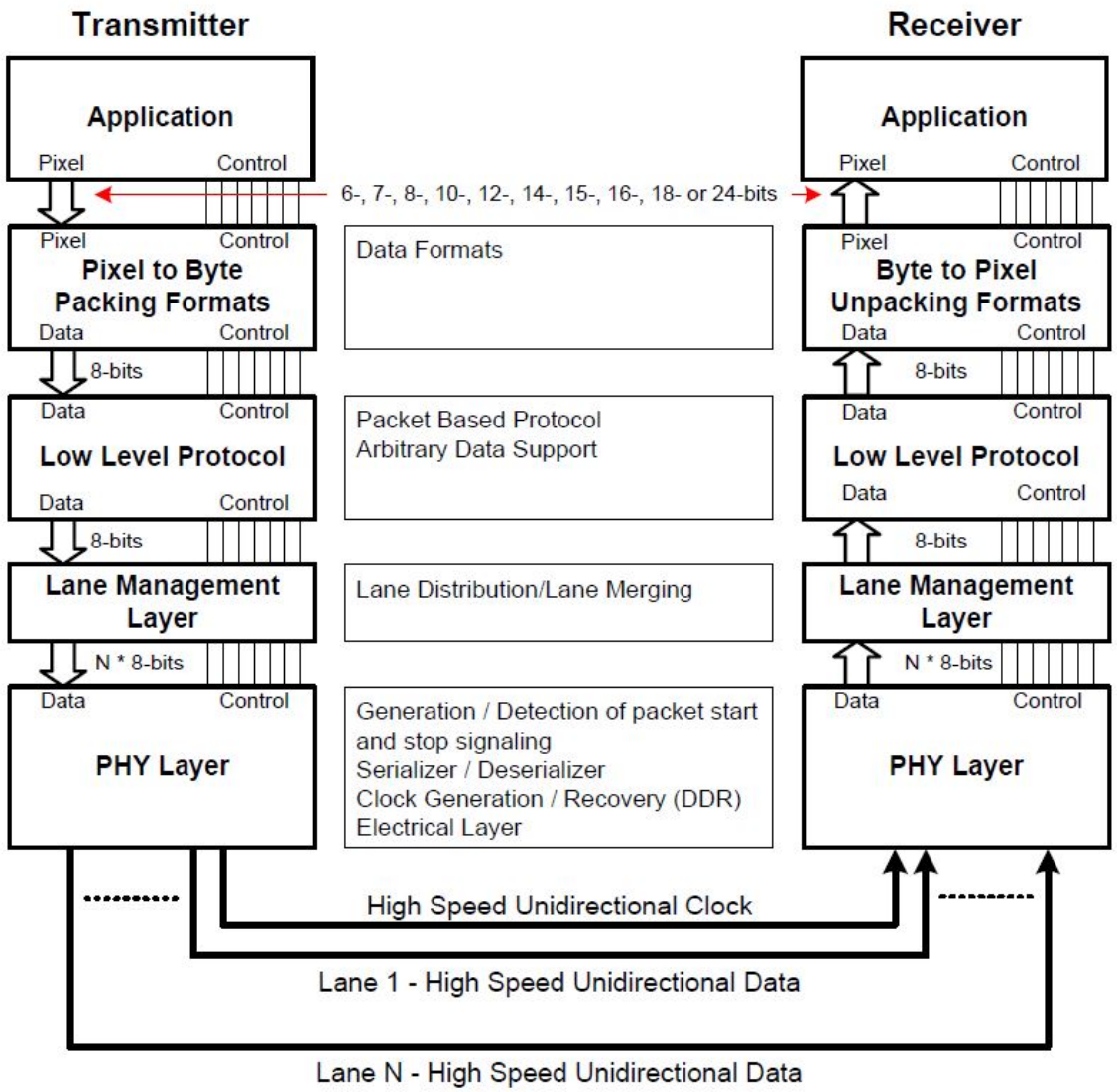


Figure 3.4: CSI-2 layer definitions

Figure 3.4 illustrates the CSI-2 layer structure, which is explained in the following subsections.

PHY Layer

The physical layer for CSI-2 specifies the characteristics of the transmission medium (electrical conductors), the input/output circuitry, the clocking mechanism, the timing relationship between clock and data Lanes, the Start of Transmission (SoT) and the End of Transmission (EoT) events.

It is composed of between one and four unidirectional data Lanes and one clock

Lane. All CSI-2 transmitters (unidirectional masters) and receivers (unidirectional slaves) shall support continuous clock behavior: the clock Lane remains in high-speed mode generating active clock signals between the transmission of data packets. CSI-2 optionally may support non-continuous clock behavior: the clock Lane enters in low-power state between the transmission of data packets.
For more detailed information on D-PHY, please refer to Section 3.1.4.

Lane Management Layer

First of all, it is important to highlight that CSI-2 is a Lane-scalable specification: applications requiring more bandwidth than that provided by one data Lane can expand the data path to two, three, or four Lanes wide. In this regard, between the PHY and higher functional layers is this layer (Lane Distribution/Lane Merging) that handles multi-Lane configurations. Specifically, the transmitting side distributes the data stream as a sequence of packet bytes across N Lanes. On the receiving side, incoming bytes are collected from N Lanes and merged together into a recombined data stream that restores the original stream sequence. Thus, the Lane distributor takes a transmission of N-byte length (arbitrary), buffers it up, and then sends groups of N bytes in parallel across N Lanes. Before sending data, all Lanes perform the SoT sequence in parallel to indicate to their corresponding receiving units that the first byte of a packet is beginning. At the end of the transmission, there may be "extra" bytes since the total byte count may not be an integer multiple of the number of Lanes, N. The Lane distributor, as it buffers up the final set of less-than-N bytes in parallel for sending to N data Lanes, de-asserts its "valid data" signal into all Lanes for which there is no further data. Although multiple Lanes all start simultaneously with parallel "start packet" codes, they may complete the transaction at different times, sending "end packet" codes one cycle (byte) apart.

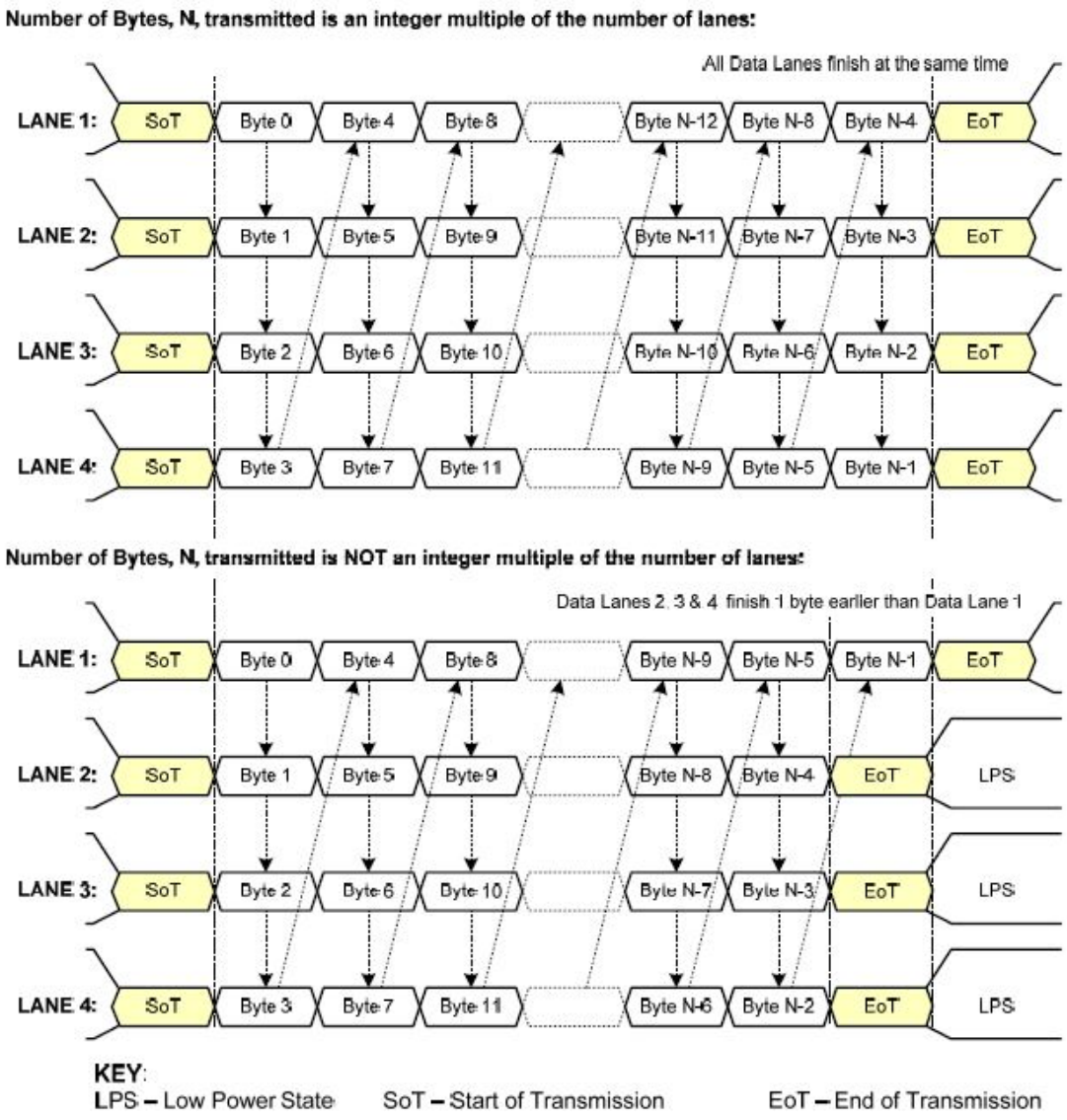


Figure 3.5: Multi-Lane configuration: four Lanes example

The N PHYs on the receiving side collect bytes in parallel, and feed them into the Lane Merging Layer. This reconstructs the original sequence of transmitted bytes.

Low Level Protocol (LLP)

LLP is a byte orientated, packet based protocol that supports the transport of serial arbitrary data (Payload) between SoT and EoT events. Two packet structures are defined for LLP communication: Long Packets and Short Packets. For each packet structure, exit from the low-power state followed by the Start of Transmission sequence indicates the start of the packet; the End of Transmission sequence followed

by the low-power state indicates the end of the packet.

Purely by way of example and for simplicity, all explanations proposed here are single Lane configurations.

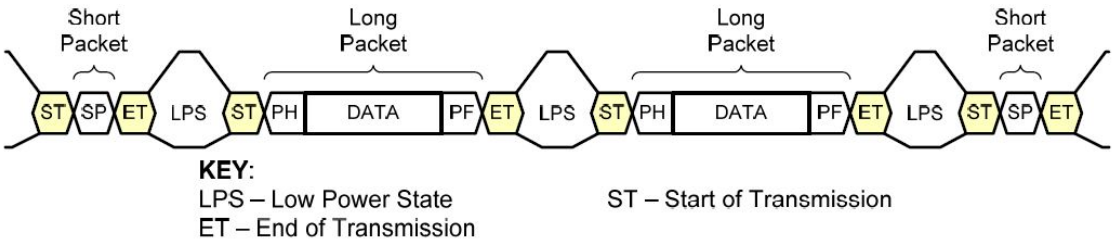


Figure 3.6: Low Level Protocol (LPP) overview

Long Packet format

Figure 3.7 illustrates the structure of the LLP Long Packet.

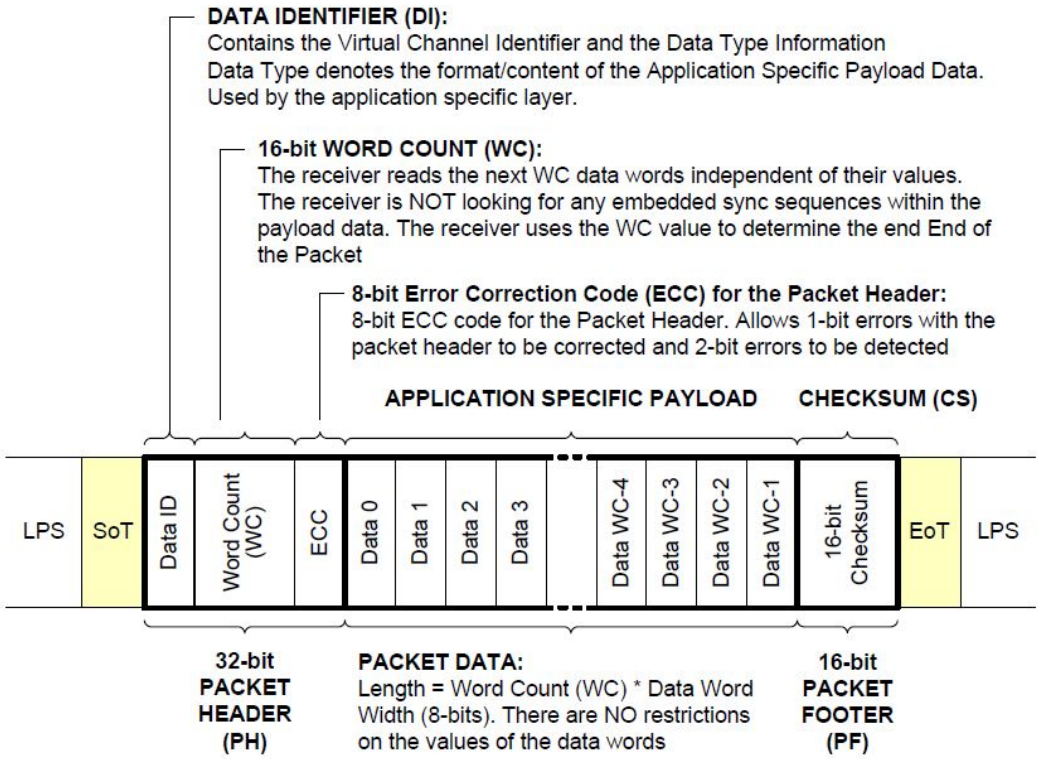


Figure 3.7: Long Packet structure

It consists of three elements: a 32-bit Packet Header (PH), an application specific

data Payload with a variable number of 8-bit data words and a 16-bit Packet Footer (PF).

The Packet Header is further composed of three elements: an 8-bit Data Identifier (Data ID), a 16-bit Word Count field (WC) and an 8-bit Error Correction Code (ECC). The Packet Footer has just one element, a 16-bit Checksum.

- **Data Identifier (Data ID)**:

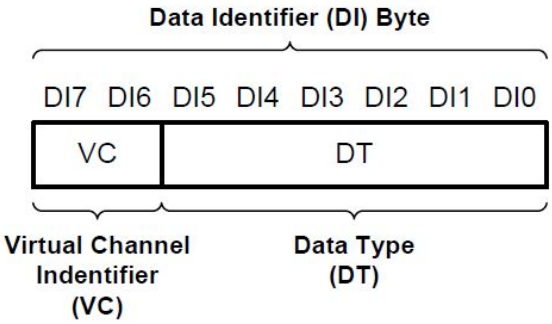


Figure 3.8: Data ID field

Data ID field is one byte defining the Virtual Channel (VC) and the Data Type (DT) values. The former is contained in the two MSBs, the latter is included in the remaining six LSBs.

Virtual Channel purpose is to provide separate channels for different data flows that are interleaved in the same data stream. CSI-2 supports up to four data streams; so, valid channel identifiers are 0 to 3.

The Data Type value specifies the format of the Payload data.

As shown in table 3.1, there are eight different Data Type classes: the first two classes denote Short Packet Data Types; the remaining six classes denote Long Packet Data Types. Within each class there are up to eight different Data Type definitions.

- **Word Count (WC)**: this 16-bit field indicates the number of 8-bit data words in the Payload between the end of the Packet Header and the start of the Packet Footer.

Data Type	Description
0x00 to 0x07	Synchronization Short Packet Data Types
0x08 to 0x0F	Generic Short Packet Data Types
0x10 to 0x17	Generic Long Packet Data Types
0x18 to 0x1F	YUV Data
0x20 to 0x27	RGB Data
0x28 to 0x2F	RAW Data
0x30 to 0x37	User Defined Byte-based Data
0x38 to 0x3F	Reserved

Table 3.1: Data Type classes

- **Error Correction Code (ECC)**: it allows single-bit errors in the Data ID and the Word Count to be corrected and two-bit errors to be detected.

At the end of the Packet Header, the receiver reads the Word Count * 8-bit data of the Payload.

- **Checksum**: the 16-bit Checksum sequence is the only field of the Packet Footer.

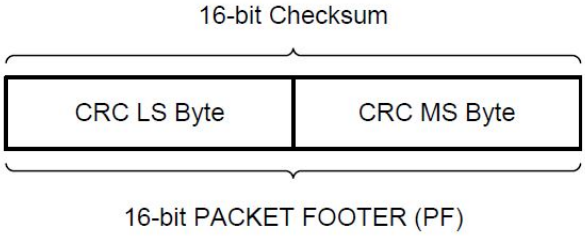


Figure 3.9: Checksum field

Once the receiver has read the data Payload, it reads the Checksum in the Packet Footer.

The Checksum is calculated as 16-bit CRC (Cyclic Redundancy Check) over each data packet in order to detect possible errors in transmission.

The checksum is sent over CSI-2 bus to the receiver to verify that no errors have occurred in the transmission.

When the Word Count is zero, then the Checksum calculation results in 0xFFFF.

After the EoT sequence the receiver begins looking for the next SoT sequence.

Short Packet format

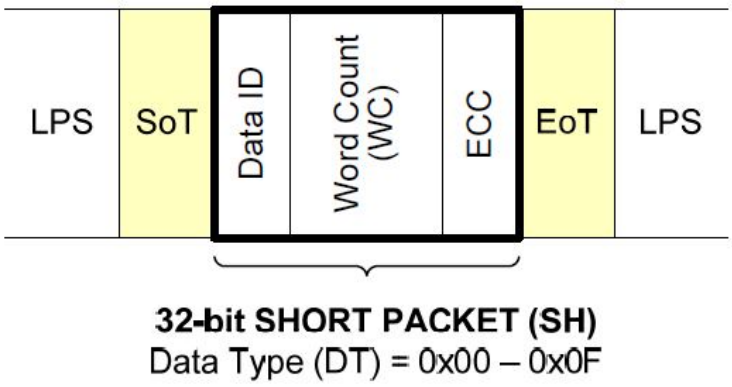


Figure 3.10: Short Packet structure

Figure 3.10 illustrates the structure of the LLP Short Packet. It consists of Packet Header only (PF is not present) and provides frame start, frame end, line start and line end information. A Short Packet shall be identified by Data Types between 0x00 and 0x0F, as described in table 3.1.

The Word Count field in the Packet Header shall be replaced by a Short Packet Data Field. For Frame Synchronization Data Types the Short Packet Data Field shall be the frame number. For Line Synchronization Data Types the Short Packet Data Field shall be the line number. Table 3.2 specifies Frame and Line Synchronization Data Types.

Data Type	**Description**
0x00	Frame Start Code
0x01	Frame End Code
0x02	Line Start Code (optional)
0x03	Line End Code (optional)
0x04 to 0x07	Reserved

Table 3.2: Synchronization Short Packet Data Type codes

For Generic Short Packet Data Types the content of the Short Packet Data Field

shall be user defined.

In general, each byte shall be transmitted least significant bit first.

Payload data may be transmitted in any byte order restricted only by data format requirements. Multi-byte elements such as Word Count, Checksum and the Short Packet 16-bit Data Field shall be transmitted least significant byte first.

Between Low Level Protocol packets there must always be a transition into and out of the low-power state, as shown in figure 3.11.

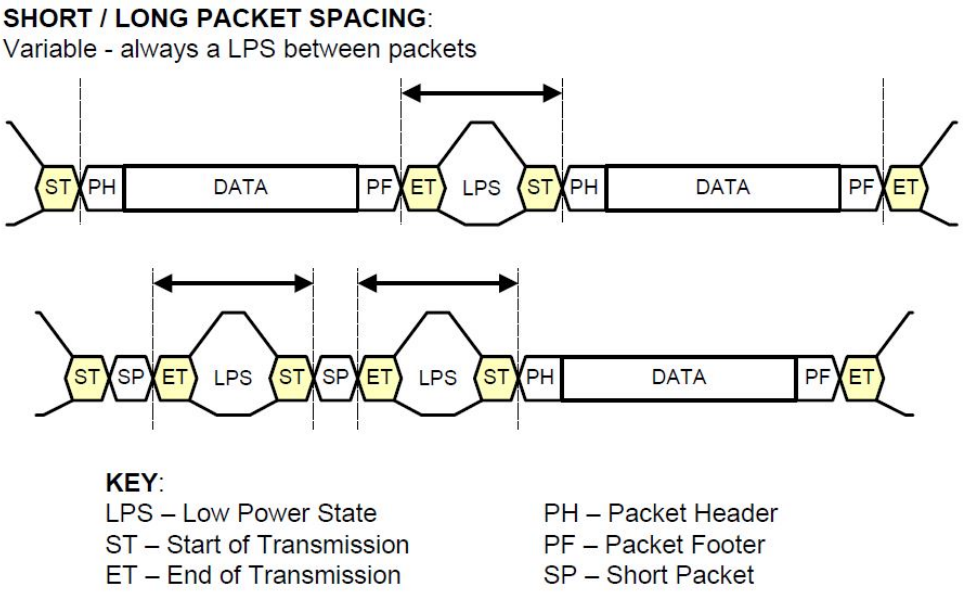


Figure 3.11: Packet spacing

The period between the Packet Footer of one Long Packet and the Packet Header of the next Long Packet is called Line Blanking Period (it is indicated by the two-headed horizontal arrow at the top of figure 3.11).

Each image frame shall begin with a frame start packet containing the Frame Start Code. It shall be followed by one or more Long Packets containing image data and zero or more Short Packets containing synchronization codes. Each image frame shall end with a frame end packet containing the Frame End Code (table 3.2). The period between the Frame End packet in frame N and the Frame Start packet in frame N+1 is called Frame Blanking Period.

Pixel/Byte Packing/Unpacking Layer

The CSI-2 supports image applications with varying pixel depth from six to twenty four bits per pixels. In the transmitter this layer packs pixels from the Application Layer into bytes before sending the data to the Low Level Protocol layer. In the receiver this layer unpacks bytes from the Low Level Protocol layer into pixels before sending the data to the Application Layer, that describes higher-level encoding and interpretation of data contained in the stream.

3.1.2 Data format: RGB888

RGB888 data format is defined by 0x24 Data Type code.
Data transmission is performed by transmitting a BGR byte sequence. This sequence is illustrated below.

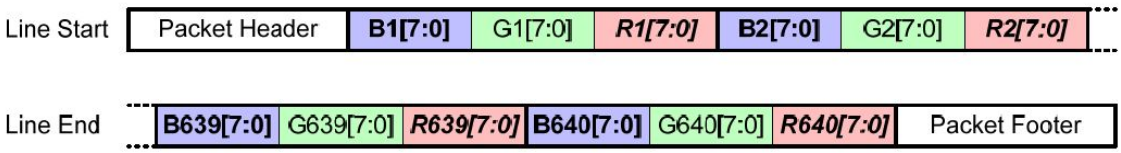


Figure 3.12: 640 pixels sequence example

Packet size constraints are specified in the following table.

Pixels	Bytes	Bits
1	3	24

Table 3.3: RGB888 packet data size constraints

Bit order in transmission follows the general CSI-2 rule, LSB first. The pixel to byte mapping is illustrated in figure 3.13.

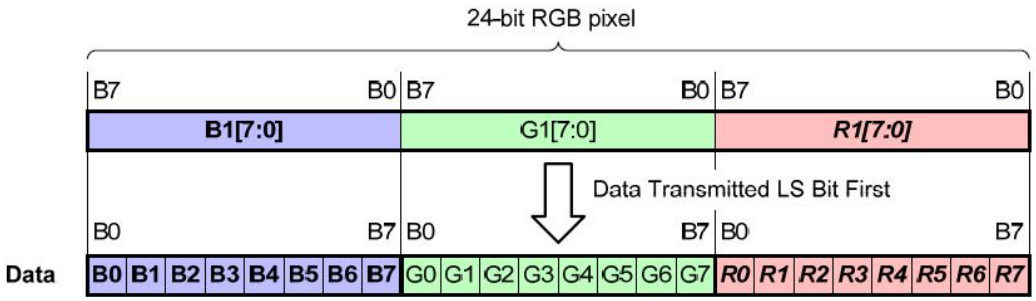


Figure 3.13: RGB888 bitwise transmission on CSI-2 bus

An example of RGB888 frame format is illustrated below.

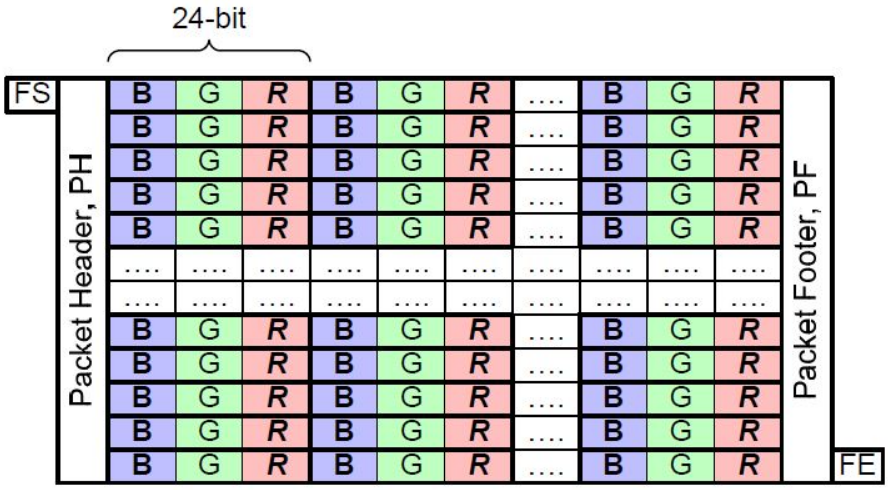


Figure 3.14: RGB888 frame format

RGB888 is a typical data format for cameras; LiDAR does not usually use this format because it does not transmit color information. The brief description of this data format is only useful for preliminary tests purpose, not shown in the discussion.

3.1.3 Data format: RAW14

Raw images data are named so because they are not yet processed and therefore consist only of a variation in the intensity of light.

LiDARs typically handle transmission of 14-bit raw data called RAW14, which is

identified by 0x2D Data Type code.

Packet size constraints are specified in the following table.

Pixels	Bytes	Bits
4	7	56

Table 3.4: RAW14 packet data size constraints

For every four pixels, seven bytes of data is generated.

An example sequence is illustrated below.

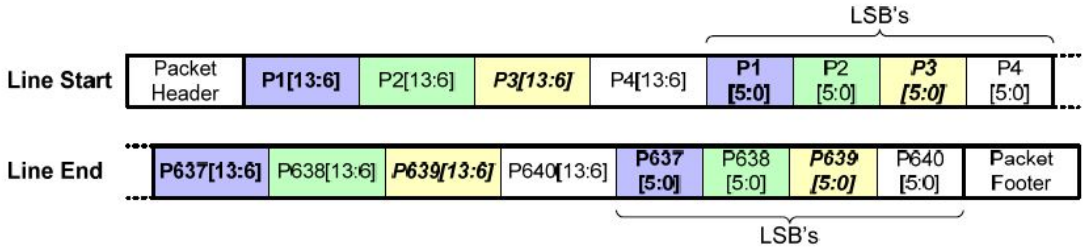


Figure 3.15: 640 pixels sequence example

The LSBs for P1, P2, P3 and P4 are distributed in three bytes as shown in figure 3.15 and figure 3.16; the same is true for the LSBs for P637, P638, P639 and P640. The bit order during transmission follows the general CSI-2 rule, i.e. LSB first.

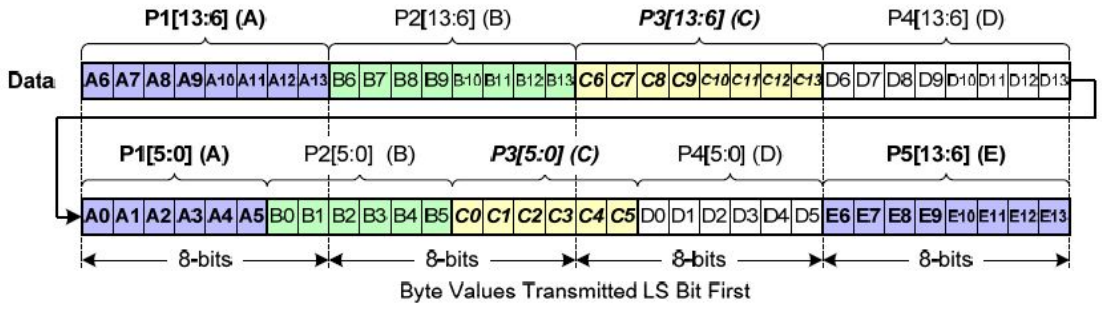


Figure 3.16: RAW14 bitwise transmission on CSI-2 bus

An example of RAW14 frame format is illustrated below.

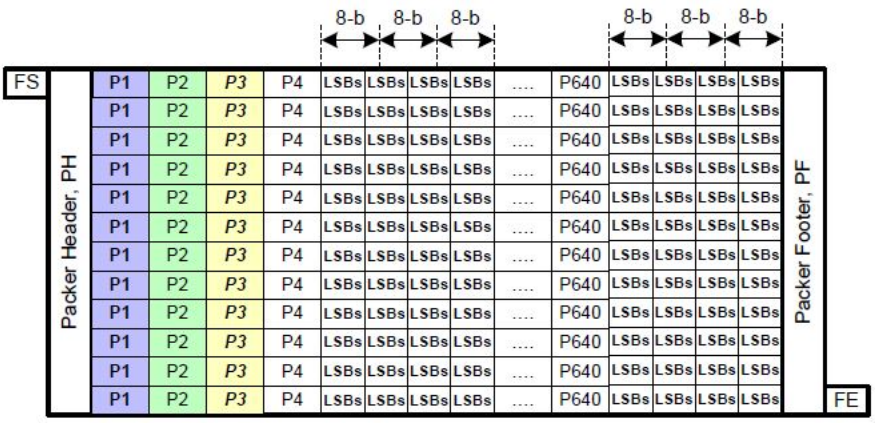


Figure 3.17: RAW14 frame format

3.1.4 D-PHY operation

D-PHY describes a source synchronous, high speed, low power, low cost PHY solution [15]. It enables significant extension of the interface bandwidth with very low power consumption for advanced applications. Implementing this specification reduces time-to-market and design cost by standardizing the interface between products from different manufacturers.

The D-PHY provides a synchronous connection between master and slave. A practical PHY configuration consists of a DDR (Double Data Rate) clock signal and one or more data signals. The clock signal is always unidirectional, originating at the master and terminating at the slave (forward direction); the data signals can either be unidirectional or bi-directional, depending on the selected options. Specifically, the PHY uses two wires (Lines) per data Lane plus two wires for the clock Lane. This gives four wires for the minimum PHY configuration. For a fixed clock frequency, the available data capacity of a PHY configuration can be increased by using more data Lanes (multiple data Lanes configuration). If N is the number of data Lanes, this means that this configuration requires 2 * (N+1) interconnect wires.

The actual maximum achievable bit rate in high-speed mode is determined by the performance of transmitter, receiver and interconnect implementations. This specification is primarily intended to define a solution for a bit rate range of 80 to 1500

Mbps per Lane without deskew calibration and up to 2500 Mbps if it supports deskew capability. The maximum data rate in low-power mode is 10 Mbps. Unlike many of the existing interfaces, D-PHY is unique because it can switch between high-speed and low-power mode in real time depending on the need to transfer large amounts of data or to conserve power to extend battery life.

In high-speed mode (for fast-data traffic) each Lane is terminated on both sides. It is driven by a differential signal (Dp and Dn Lines), charactezized by a low swing (e.g. 200 mV). In low-power mode (for control purposes) signals have a large swing (e.g. 1.2 V) and all wires are operated single-ended and non-terminated. For both high-speed and low-power transmission, data taken from a receiver and provided to a transmitter on any Lane shall be an integer number of bytes; there is no maximum number of bytes implied by the PHY. For serial transmission, data shall be serialized in the transmitting PHY and deserialized in the receiving one.

Hence, in order to meet both low-power and high-speed requirements, the D-PHY provides a low-power transmitter (LP-TX), a high-speed transmitter (HP-TX) and a serializer for the transmission of specific data; while on the receiving side, it provides a low-power receiver (LP-RX), a high-speed receiver (HS-RX), a deserializer and a low-power conflict detector (LP-CD) to receive those specific MIPI D-PHY signals.

During normal operation either a HS-TX or a LP-TX drives a Lane, differentially or single-ended respectively. This results in two possible high-speed Lane states and four possible low-power Lane states.

State code	Line voltage levels		High-speed	Low-power
	Dp-Line	**Dn-Line**	**Burst mode**	**Control mode**
HS-0	HS Low	HS High	Differential-0	N/A
HS-1	HS High	HS Low	Differential-1	N/A
LP-00	LP Low	LP Low	N/A	Bridge
LP-01	LP Low	LP High	N/A	HS-Rqst
LP-10	LP High	LP Low	N/A	LP-Rqst
LP-11	LP High	LP High	N/A	Stop

Table 3.5: Lane states description

During normal operation a data Lane will be either in Control or high-speed mode. For data Lanes and for clock Lanes the Stop state serves as general standby state.

High-speed (burst) mode

High-speed data transmission occurs in bursts; it starts from, and ends with, a Stop state (LP-11).

To aid receiver synchronization, data bursts shall be extended on the transmitter side with a leader and a trailer sequence that shall be eliminated on the receiver side. These leader and trailer sequences can therefore only be observed on the transmission lines.

During a HS data burst the clock Lane shall be in high-speed mode, providing a Double Data Rate (DDR) clock to the slave side: data shall be sampled on both the rising and falling edges of the clock signal. Data is launched in a quadrature relationship to the DDR clock such that the clock signal edge is used directly by the receiver to sample the received data.

It is important to point out that during HS receiver operation, termination impedance Z_{ID} is required between Dp and Dn pins. Z_{ID} shall be disabled when the module is not in HS receive mode. The nominal value of reference characteristic impedance level is 100 Ω in differential mode, while it is 50 Ω in single-ended per line.

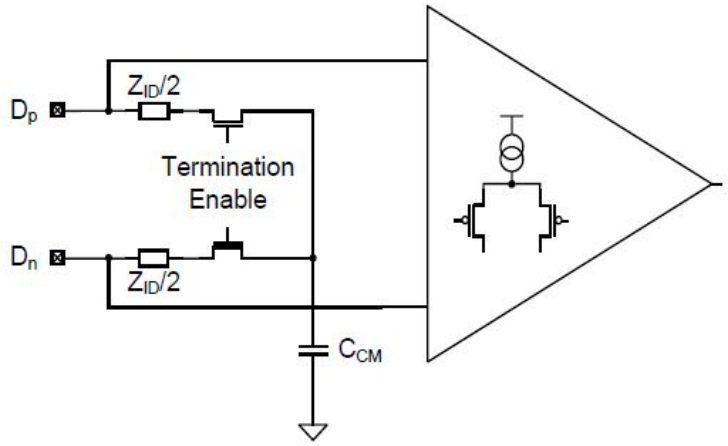


Figure 3.18: HS receiver implementation example

Consequently, to obtain a matched line, the PCB layout must be optimized so that there is the same characteristic impedance at the transmitter side.

Start of Transmission (SoT)

A data Lane leaves the Stop state and prepares for high-speed mode by means of a Start of Transmission (SoT) procedure. Table 3.6 describes the sequence of events on transmitter and receiver side which is also shown graphically on the left side of figure 3.19.

TX	**RX**
Drives Stop state (LP-11)	Observes Stop state
Drives HS-Rqst state (LP-01) for time T_{LPX}	Observes transition from LP-11 to LP-01 on the Lines
Drives Bridge state (LP-00) for time $T_{HS-PREPARE}$	Observes transition from LP-01 to LP-00 on the Lines, enables Line termination after time $T_{D-TERM-EN}$
Enables high-speed and disable low-power simultaneously	
Drives HS-0 for a time $T_{HS-ZERO}$	Enables HS-RX and waits for timer $T_{HS-SETTLE}$ to expire in order to neglect transition effects
	Starts looking for leader sequence
Inserts the HS Sync-sequence '00011101'	
	Synchronizes upon recognition of leader sequence '011101'
Continues to transmit high-speed payload data	
	Receives payload data

Table 3.6: Start of Transmission sequence

High-speed data transmission

Figure 3.19 shows the sequence of events during the transmission of a data burst. Transmission can be started and ended independently for any Lane. However, for most applications the Lanes will start synchronously but may end at different times due to an unequal amount of transmitted bytes per Lane.

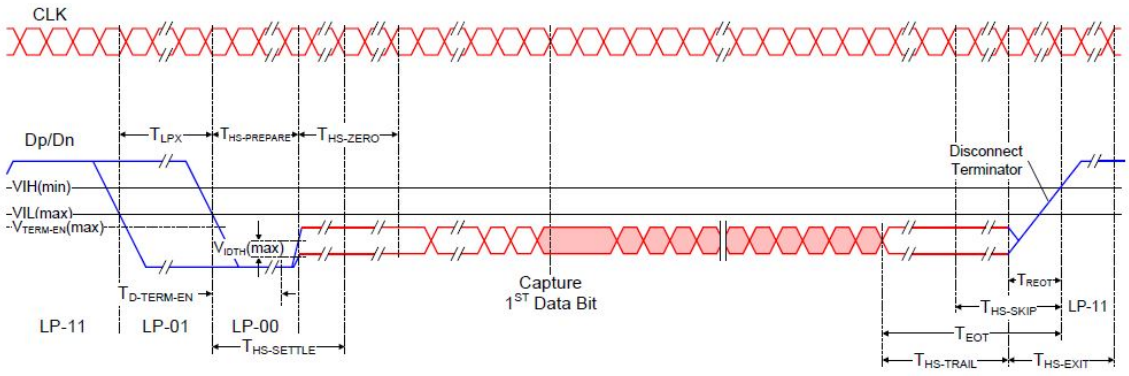


Figure 3.19: High-speed data transmission in burst

End of Transmission (EoT)

At the end of a data burst, a data Lane leaves high-speed transmission mode and enters the Stop state by means of an End of Transmission (EoT) procedure. Table 3.7 describes the sequence of events on transmitter and receiver side.

TX	**RX**
Completes transmission of payload data	Receives payload data
Toggles differential state immediately after last payload data bit and keeps that state for a time $T_{HS-TRAIL}$	
Disables the HS-TX, enables the LP-TX and drives Stop state (LP-11) for a time $T_{HS-EXIT}$	Detects the Lines leaving LP-00 state and entering Stop state (LP-11) and disables termination
	Neglect bits of last period $T_{HS-SKIP}$ to hide transition effects
	Detect last transition in valid data, determine last valid data byte and skip trailer sequence

Table 3.7: End of Transmission sequence

The above sequence is shown graphically on the right side of figure 3.19.

3.2 Technical solutions for data aggregation

MIPI Alliance is continuously developing the world's most comprehensive set of interface specifications for mobile-influenced products and automotive solutions. The majority of image sensors and application processors (AP) in the consumer market use the Mobile Industry Processor Interface Camera Serial Interface 2 as a video signal interface. In recent times, it has also made significant progress in improving Advanced Driver Assistance Systems and other applications for the automotive world specializing on automotive specifications in accordance with the strict requirements of OEMs, Tier 1 suppliers, SoC designers and other automotive vendors, such as

reliability, functional safety and electromagnetic interference (EMI) reduction.
Trends like the spread of displays, cameras and LiDAR sensors generate a growing demand for high performance interfaces in the automotive industry. The only technology that can manage the explosion of video, audio, and communications with simple implementation and affordable costs are high speed serial links. High bandwidth, performance, and reliability of these serial links will continue to be key requirements as the automotive industry advances toward fully autonomous cars.
Consequently, the case study proposed in this Master Degree Thesis explores and deepens this kind of themes, focusing on a project developed by Marelli Automotive Lighting, one of the biggest automotive Tier 1 suppliers worldwide.

Figure 3.20: Marelli Automotive Lighting logo

Marelli Automotive Lighting designs, develops and integrates a wide and complete range of top notch solutions for external automotive lighting systems, as well as innovative sensors to support ADAS/AD features including LiDAR sensor technology, showing an advanced know-how in all the key technologies that enable evolution in these areas. Innovation, performance and quality in development, production and delivery of automotive systems are met by combining extraordinary design and innovative technology.
Specifically, the application evaluated for this Thesis work fits into this context, responds to the needs listed above and tries to propose a cutting-edge solution to obtain a high performance LiDAR system.
Therefore, focusing back on the LiDAR platform architecture (figure 2.25), it should

be stressed once again that it is true that most automotive CPUs support CSI-2 interfaces to receive image data from LiDAR sensors, but it is even more true that many of those on the market are equipped with only one or two CSI-2 ports; instead, the number of sensors on board is expected to escalate quickly and, in the worst case scenario, three to four optoelectronic modules are required to achieve the desired LiDAR field of view.

Starting from these premises, it is definitely necessary to evaluate and design a hardware module that aggregates data from optoelectronic modules and provides a unique output data stream to the CPU. This hardware block must be subsequently integrated into the LiDAR Mainboard, designed and produced by the company.

Different possible technical solutions can be evaluated in this regard, analysing pros and cons of each system.

First of all, an ASIC chip can fit this application. ASIC stands for Application Specific Integrated Circuit and, as the name implies, it is developed for a specific application: it is designed for one sole purpose and it functions the same its whole operating life (in this case, to get raw data aggregation). Functionalities and the digital circuitry are permanently drawn into silicon and can not be changed anymore. An ASIC is able to operate at high frequency (because the circuit is optimized for its specific function), it can include complete analog circuitry on the same die and its power consumption can be minutely controlled and optimized, but its lack of flexibility and scalability does not make it a suitable component for prototyping or for applications requiring frequent upgrades. Moreover, ASICs can be very hard, expensive and time-consuming to design, so they are not always the best choice.

Another very relevant solution, used to implement the first prototype of the project, is based on Serializer/Deserializer technology (SerDes). A Serializer circuit consists of functional blocks that convert parallel input data into a serial stream (one bit at a time) that is then transmitted to a receiver on a high speed connection, such as LVDS (Low-Voltage Differential Signaling); thus, at the receiver side the serial stream

is converted back into a parallel one by the Deserializer circuit. Both Serializer and Deserializer are functional blocks on transmitting and receiving chips and are Parallel In Serial Out (PISO) and the Serial In Parallel Out (SIPO) respectively. SerDes has emerged as the primary solution where there is a need for fast data movement and limited I/O, allowing designers to speed up data communication without having to increase the number of pins. In fact, the SerDes solution is often implemented for transmissions between different boards or different systems placed at long distances (e.g. one board containing optoelectronic module and Serializer and another board containing Deserializer and System on Chip). More particularly, in order to aggregate two CSI-2 data streams coming from two optoelectronic modules, two Serializers and one Deserializer are necessary. The former (one for each LiDAR optoelectronic module) converts MIPI signals into LVDS and directs these signals to the Deserializer. The latter aggregates the signals coming from the two LiDAR modules and reconverts the result from LVDS to MIPI. The output signal is then sent and processed within the System on Chip. Obviously, the use of these three components implies a greater area occupation on the PCB.

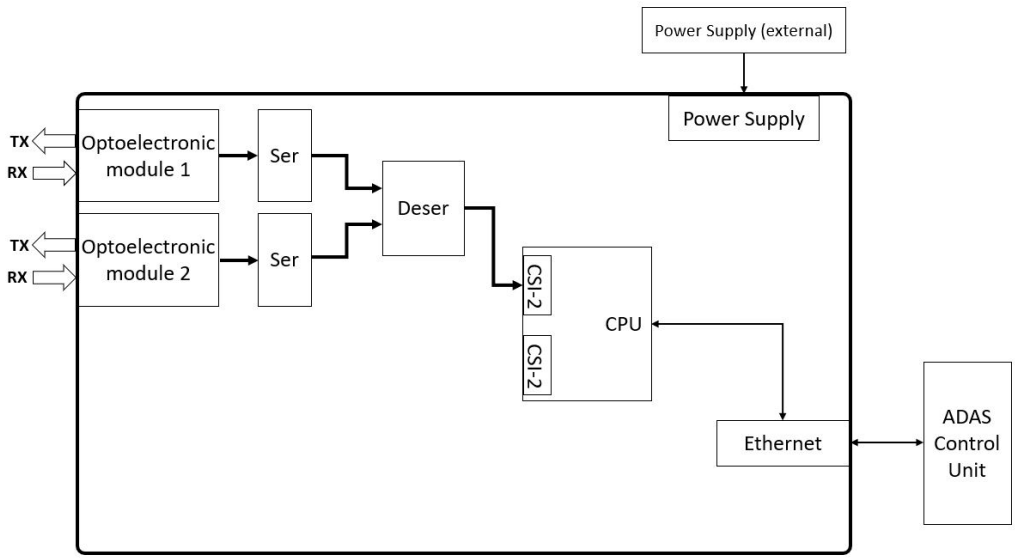


Figure 3.21: Example of data aggregation based on SerDes-solution

The architecture in figure 3.21 is just an example showing the aggregation oper-

ation for only two optoelectronic modules.

Lastly, with this Thesis work a third alternative solution has been studied and deepened in order to combine optimal performance and flexibility: an FPGA would replace the entire SerDes module.

FPGA stands for Field Programmable Gate Array. The first FPGAs were designed in the mid-1980s, and they have been used as a key component of many different types of tech products ever since. The concept behind the original FPGAs was to create a more flexible alternative to ASICs. They are inherently flexible chips that, as their names suggests, can be programmed and re-programmed in the "field", that is, after the chip is built. This "updateability" is an incredibly useful capability because it allows companies to add new features (or fix bugs in existing functions) to devices which include FPGAs.

Housed within the confines of an FPGA are series of logic blocks and high speed interconnects that can be used to control the inner workings of the chip. Conceptually, it is not much different than getting a big set of Lego blocks that can be rearranged and reconfigured exactly as desired.

Technically speaking, an FPGA is made up of thousands of Configurable Logic Blocks (CLBs) embedded in an ocean of programmable interconnects. Each CLB is primarily made of Look-Up Tables (LUTs), Multiplexers and Flip-Flops used to implement complex logic functions.

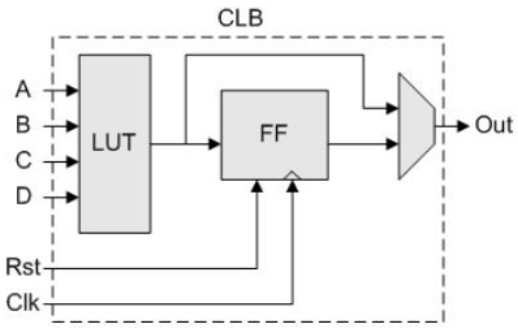


Figure 3.22: CLB inner structure

Apart from CLBs and routing interconnects, many FPGAs also contain dedicated RAM blocks, DSP blocks, External Memory Controllers, PLLs, Clock Managers, Multi-Gigabit Transceivers etc.

Typically, FPGA design relies on Hardware Description Languages (HDLs), i.e. Verilog or VHDL, which describe all the modules (hardware blocks) needed within the design, specifying inputs, outputs, and how they are produced. The two major purposes of HDLs are logic simulation and synthesis. During simulation, inputs are applied to a module, and the outputs are checked to verify that the module operates correctly. Logic simulation is essential to test a system before it is built. During synthesis, the code description of a module is transformed into logic gates: logic synthesis translate HDL code into a netlist describing the hardware (e.g. the logic gates and the wires connecting them). The logic synthesizer might perform optimizations to reduce the amount of hardware required. The synthesis tool generates the configuration file for programming the FPGA.

FPGA providers develop their own in-house toolchain or offer a customised version of other suppliers tools for development, simulation and design synthesis.

Overall, the optimal design solutions are often provided by FPGAs. They have evolved rapidly over the years and are now able to meet numerous design requirements in terms of flexibility, processing speed and power consumption, making them useful for a wide range of applications.

So, to sum up, a comparison table is provided among the various technical solutions presented above.

	ASIC	SerDes	FPGA
Time-to-market	high	low	low
PCB area	low	high	low
Power consumption	low	high	low
Number of components	low	high	low
Flexibility	low	medium	high
Design cost	high	low	low
Application cost/Volume	low	medium	medium

Table 3.8: Summary of technical solutions

As can be seen from the table 3.8, the FPGA-based solution is one of the most suitable and convenient technique for the purpose of the project; for all the reasons listed above, it has been selected for the LiDAR Mainboard prototype. The ultimate goal is the development of a data aggregation circuit for autonomous driving LiDAR sensors that collects multiple MIPI CSI-2 raw data from two optoelectronic modules and provides a single CSI-2 output to the System on Chip (dual MIPI CSI-2 to single MIPI CSI-2 aggregation).

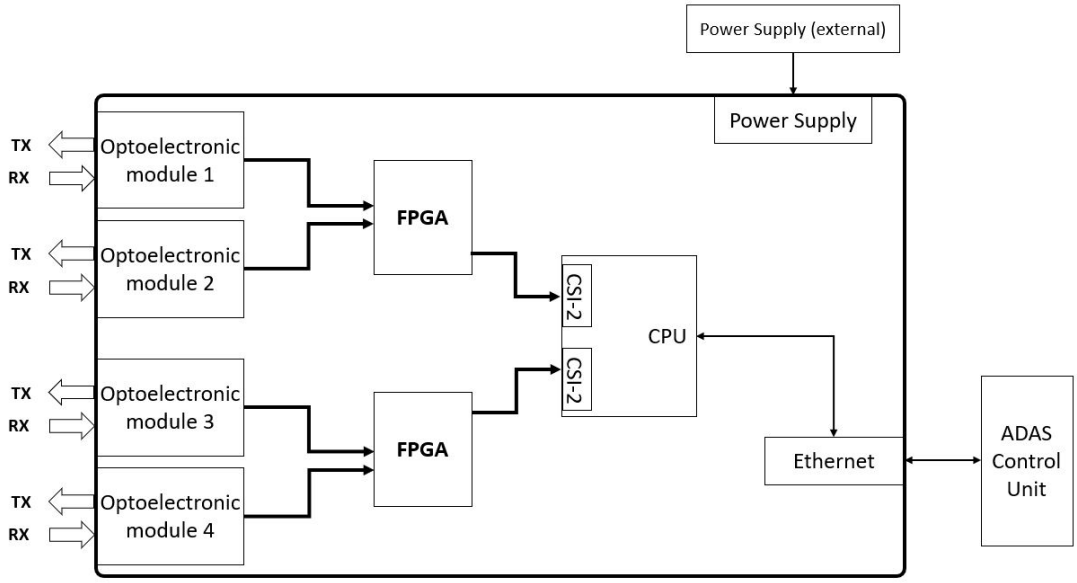


Figure 3.23: Example of data aggregation based on FPGA-solution

3.3 CrossLink Automotive Family from Lattice Semiconductor

Let us focus now on the aim of the project: the design and implementation of a hardware module able to aggregate two distinct MIPI CSI-2 data flows coming from two LiDAR optoelectronic modules. This block must provide a single output MIPI CSI-2 data stream on four data Lanes and one clock Lane to the CPU (System on Chip).

The goal of this Thesis work has been the evaluation of a technical solution based on FPGA.

The starting point was a feasibility study, an in-depth analysis of several FPGA families from different manufacturers and suppliers, in order to choose the best available solution according to project needs.

Obviously the application could have been implemented on any kind of FPGA, thanks to the high flexibility that this hardware solution inherently offers. However, on many of these hardware families the development of the entire block that implements the MIPI protocol used in order to offer the data aggregation functionality would have been required. All solutions of this type have been discarded because linked to an unacceptable feedback in terms of time and resources involved. Furthermore, the implemented protocol would require a certified approval by the MIPI Alliance.

Actually, several families of FPGA already integrate transmitting and receiving D-PHY blocks. Since the use of the CSI-2 protocol is the core of the project, the attention has been paid to this peculiarity, narrowing the choice to the only type of hardware that have such integrated feature.

In particular, the solution analyzed refers to an FPGA of the CrossLink Automotive family offered by Lattice Semiconductor, which makes available all the features necessary for design purposes.

Lattice Semiconductor FPGAs pay special attention to energy-efficient devices in the current fast-growing automotive industry and also offers a suite of design and verification tools. The CrossLink family embraces specialized FPGAs which, in addition to programmable logic and remarkable I/O capabilities, integrate hardware specifications widely used in industrial and automotive applications, including the physical layer for high speed data communication MIPI D-PHY, the core Camera Serial Interface 2 and Display Serial Interface 2, all in very compact package.
CrossLink Automotive from Lattice Semiconductor, a programmable video bridging device supporting a variety of protocols and interfaces for mobile image sensors and displays, is based on Lattice mobile FPGA 40 nm technology and combines the extreme flexibility of an FPGA with the low power, low cost and small footprint of an ASIC [16]. The device is capable of supporting high resolution, high bandwidth contents.
In general, this device offers the possibility to develop a wide variety of applications, but the one suitable for the purpose of the project is the dual MIPI CSI-2 to single MIPI CSI-2 data aggregation.

3.3.1 Architecture overview

CrossLink Automotive provides three key building blocks:

- two banks of flexible programmable I/O supporting a variety of standards, including MIPI RX D-PHY (but not TX), LVDS, LVCMOS, etc.
- a programmable logic core providing LUTs, memory, and system resources
- up to two embedded 4 data Lanes (and 1 clock Lane) Hard MIPI D-PHY TX/RX blocks, able to transmit and receive

In addition to these blocks, CrossLink Automotive also provides key system resources including a Power Management Unit (PMU), flexible configuration interface,

additional CMOS GPIOs and user I^2C/SPI configuration blocks.

A simplified version of the device block diagram is shown in figure 3.24.

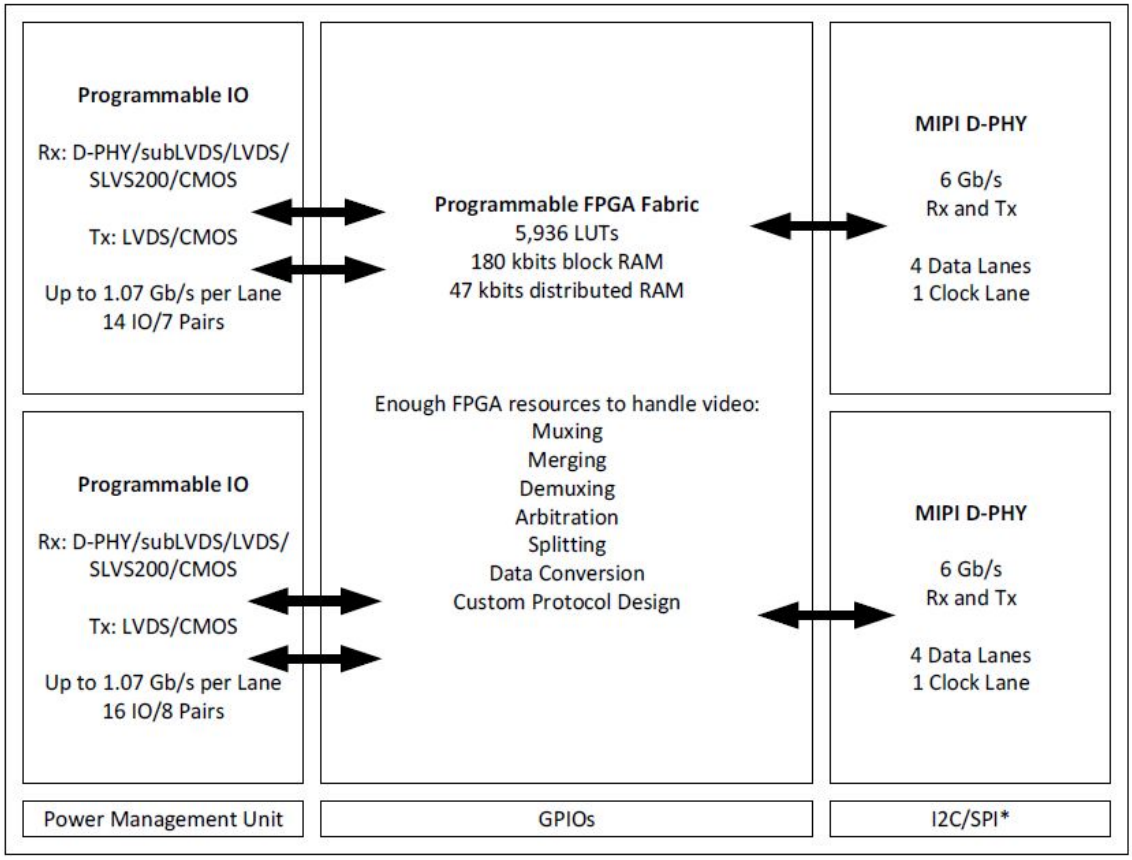


Figure 3.24: CrossLink Automotive simplified block diagram

The two programmable I/O and the the dedicated CMOS GPIOs blocks are also called banks 1, 2 and 0 respectively. All of them have independent configurable voltage levels based on V_{CCIO} supply. Bank 0 GPIOs does not include differential signaling capabilities and only supports Single Data Rate (SDR) interfaces, while bank 1 and bank 2 support both SDR and Double Data Rate (DDR) interfaces.

sysI/O buffers are distributed across three banks: CrossLink Automotive supports single-ended buffers on all three banks, while differential I/O is supported on bank 1 and bank 2. The sysI/O buffers allow a wide variety of standards to interface to a range of systems, including LVDS, LVCMOS and MIPI.

The two Hard MIPI D-PHY quads can be configured for both CSI-2 and DSI ap-

plications, supporting both high-speed (HS) and low-power (LP) modes. Lattice Semiconductor provides a set of pre-engineered IP (Intellectual Property) modules; by using these configurable soft core IPs as standardized blocks, designers are free to focus on unique aspects of their design (increasing the productivity).

The programmable FPGA fabric cosists of 5936 four input Look-Up Tables (LUT4) arranged alongside dedicated registers in Programmable Functional Units (PFU). These PFU blocks are the building blocks for logic, arithmetic, RAM and ROM functions. Moreover, considering a device simplified scheme, there are rows of Embedded Block RAM (EBR) between rows of PFUs, with I^2C, Hard MIPI D-PHY blocks, one NVCM, programmable I/O banks and PMU arranged on the top and bottom of the device as shown in figure 3.25.

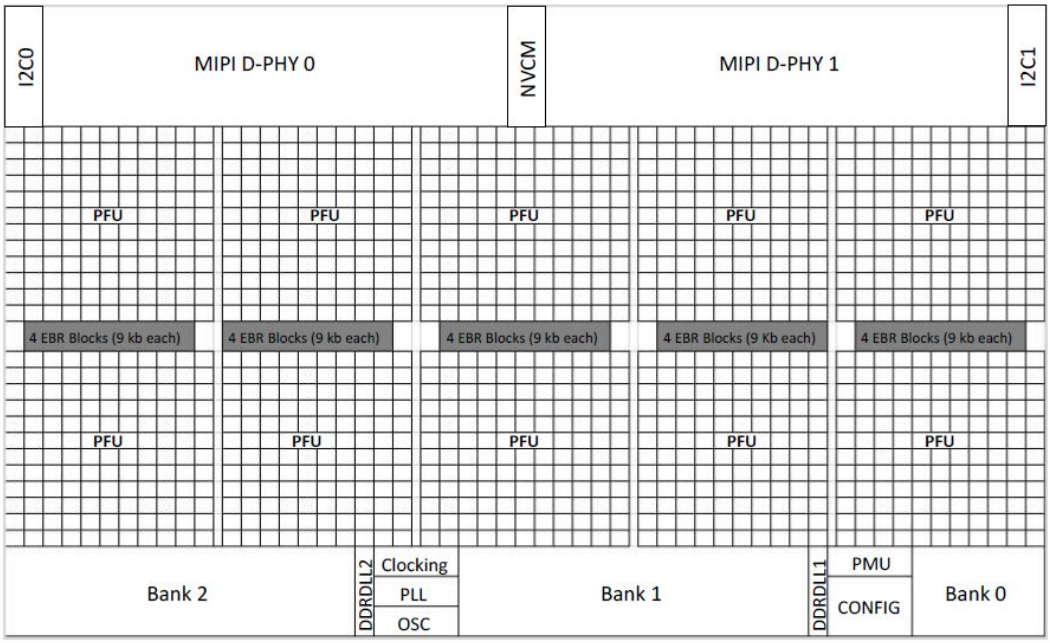


Figure 3.25: More detailed CrossLink Automotive block diagram

CrossLink Automotive is a SRAM-based programmable logic device that includes an internal Non-Volatile Configuration Memory (NVCM), as well as flexible SPI and I^2C configuration modes [17].

The device provides four different modes for loading the configuration data into the

SRAM memory:

- Self-Download (NVCM) mode, retrieving bitstream from internal NVCM
- Master SPI mode, retrieving bitstream from an external SPI Flash
- Slave SPI mode – System microprocessor writes bitstream to CrossLink Automotive through SPI port
- Slave I^2C mode – System microprocessor writes bitstream to CrossLink Automotive through I^2C port

In this regard, CrossLink Automotive provides a set of sysCONFIG I/O pins to interact with the FPGA for programming, configuration and access to resources within it. Or else, they can be reconfigured to act as general purpose I/O.
In order to be correctly operational, the FPGA goes through a sequence of states. Before applying power supply, CRESETB pin must be kept low. As external power ramps up, a Power On Reset (POR) circuit inside the FPGA becomes active, ensuring the external I/O pins are in high-impedance state and monitoring supply voltages levels: in this way the initialization process starts. After CrossLink drives CDONE pin low, it enters the memory initialization phase: this state purpose is to clear all of the SRAM memory inside the FPGA. CrossLink remains in this state until the CRESETB pin is deasserted (high).
When CRESETB pin toggles from low to high, the device enters the Master Configuration Mode: it first attempts to boot from the NVCM; if the device fails, it attempts to fetch configuration data from an external SPI Flash (using the MSPI mode).
Holding CRESETB low postpones the Master auto booting event and allows the slave configuration ports SSPI/SI^2C (Slave SPI or Slave I^2C) to detect a Slave Active condition: an external SPI or I^2C Master sends the Activation Key to the FPGA while CRESETB is held asserted (low), to prevent the device from entering Master Configuration Mode. Slave Active condition means that the slave port is addressed,

the Activation Key is sent and matches the pre-defined key code.

Wake-up is the transition from configuration mode to User Mode and it starts when the device has correctly received all of its configuration data; asserting the CDONE pin high, the FPGA enters User Mode.

In User Mode the device begins performing the logic operations in the design. While in User Mode, CRESETB can be toggled from high to low at any time to re-enter Configuration Mode. The current User Mode configuration of the CrossLink device remains in operation until it is actively cleared or power is lost.

The sequence of states described above is illustrated in the configuration flow of figure 3.26.

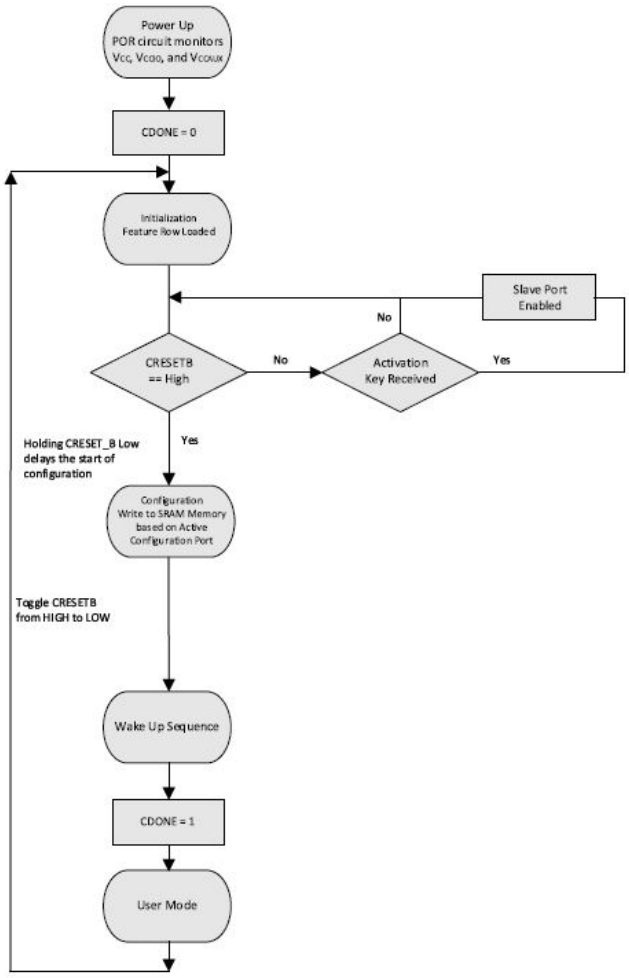


Figure 3.26: Configuration flow

Overall, the use of an FPGA that supports multiple I/O standards, coupled with some MIPI soft IP cores, provides a fast, cost-effective and risk-free upgrade path.

3.3.2 The Evaluation Board

In order to achieve rapid development and performance evaluation of the dual MIPI CSI-2 to single MIPI CSI-2 data aggregation application, the evaluation board CrossLink LIF-MD6000 Master Link Board has been employed [18]. Obviously, the board's key component is the CrossLink FPGA in 81 csfBGA package, with 4.5 x 4.5 mm^2 footprint and 0.5 mm pitch (part number LIF-MD6000-6MG81I).

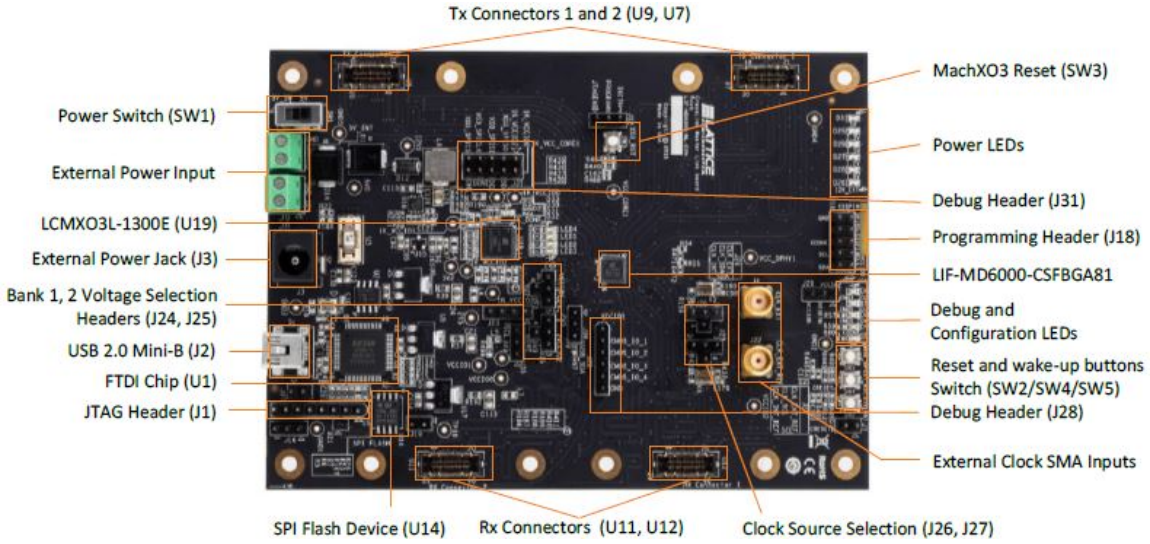


Figure 3.27: Master Link and its key components top view

Figure 3.28 depicts the evaluation board schematic showing the main components mounted on it.

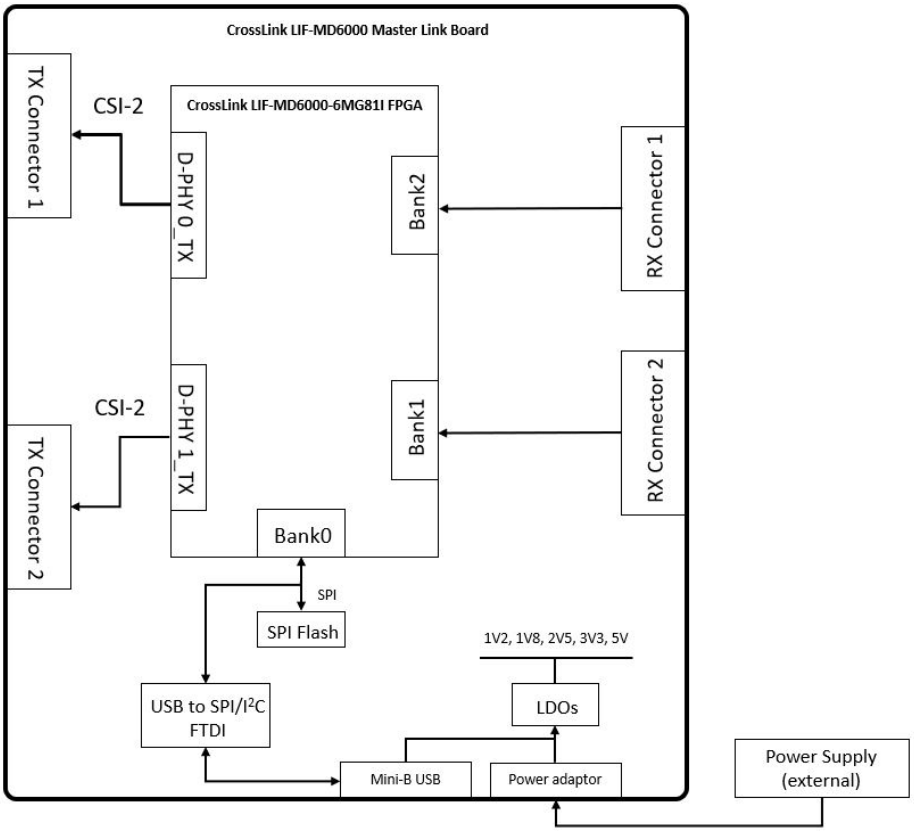


Figure 3.28: LIF-MD6000 Master Link Board schematic (main components)

A simple switch is used to connect/disconnect power to/from board.

The power supply is provided by an external adaptor or Mini-B USB: the former provides 12 V power source, the latter 5 V and it is also used for device programming. There are five voltage regulators on the board used to supply the 5 V, 3.3 V, 2.5 V, 1.8 V, and 1.2 V rails. The input to these regulators is from either the external 12 V adaptor or the Mini-B USB connector, as shown in figure 3.29.

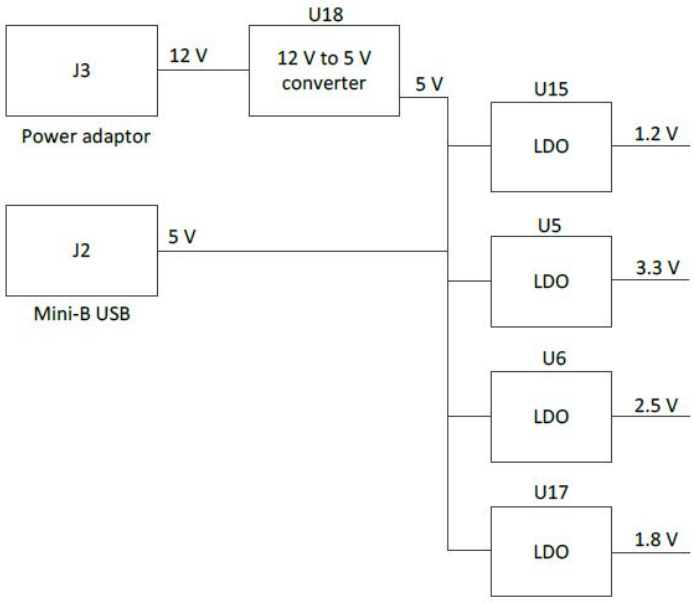


Figure 3.29: Power supply block and voltage regulators

The Mini-B USB connector is also used for programming the board, as mentioned before. It is connected to the FTDI chip which provides interfaces for SPI and I^2C to program CrossLink and/or the SPI Flash memory, already integrated in the board.

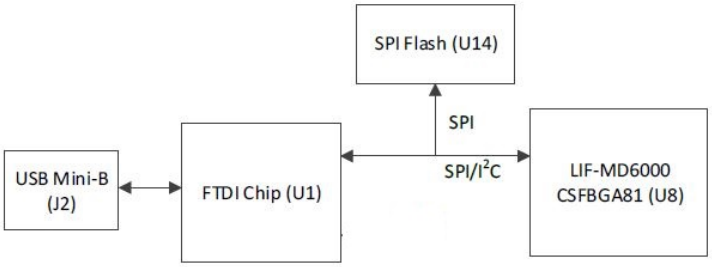


Figure 3.30: Programming block

The presence of LEDs and test points allows verification of the correctness of supply voltages, current flow, configuration and application status.
It is clear that many more components and functions than those exploited are available on the board (due to CrossLink flexibility); only those functional to the specific

application have been deepened and used.

3.4 MIPI CSI-2 Virtual Channel aggregation: project design

In this project design CrossLink is used to interface to multiple MIPI CSI-2 optoelectronic LiDAR sensors and aggregate raw data collected to a unique CSI-2 output data stream to be sent to an automotive System on Chip.

The aggregation uses the Virtual Channel merge method, meaning that it is performed arbitrating data packets based on VC [19].

The main purpose of the activity, which allows the verification of the simplest version of this functionality, is the combination of raw data coming from only two optoelectronic modules in a single output data stream.

Although the final Marelli's prototype requires the use of two identical optoelectronic LiDAR modules, the setup of this Thesis work replaces one of the two modules with a development device due to the availability of only one module from the supplier; this component is capable of generating a MIPI CSI-2 pattern (like the other LiDAR sensor) with similar data type and at a slightly different frequency. It must be clear that when both modules are available, a new design will not be necessary, but it will be enough to modify and customize the project already created.

Figure 3.31 illustrates a simplified version of the block level diagram of the current MIPI CSI-2 Virtual Channel aggregation design; all requirements and specifications are explained below.

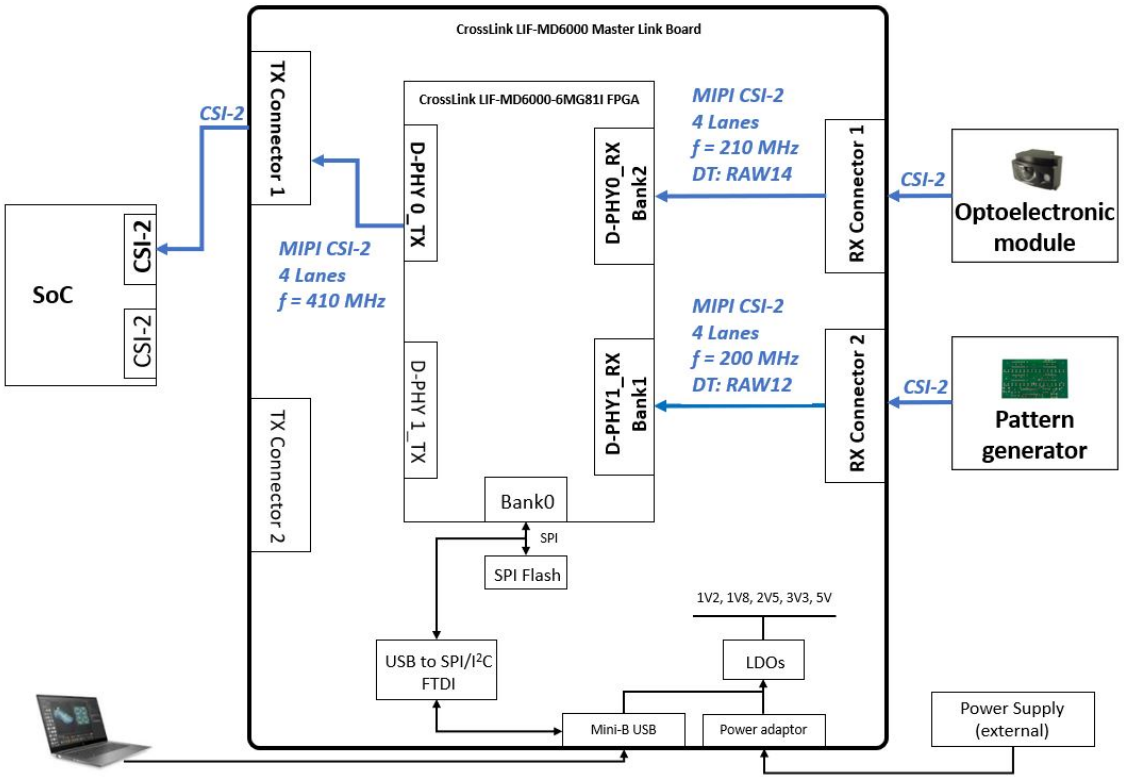


Figure 3.31: Simplified block diagram

In general, applications like this require flexibility, verification, and the ability to iterate quickly. In this respect, during the evaluation and implementation of the functionality, Lattice Diamond has been the reference software tool [20]. This FPGA design and verification software environment allows large complex designs to be efficiently implemented and optimized using CrossLink Automotive, facilitating and accelerating development of Lattice FPGA-based applications. Diamond contains a complete leading-edge set of tools for design entry, implementation, synthesis, analysis, on-chip hardware debug, programming and simulation. Thus, this software is a robust and complete environment from entering the design to programming the device.

Design entry

The latest version 3.12 of Lattice Diamond has been used.

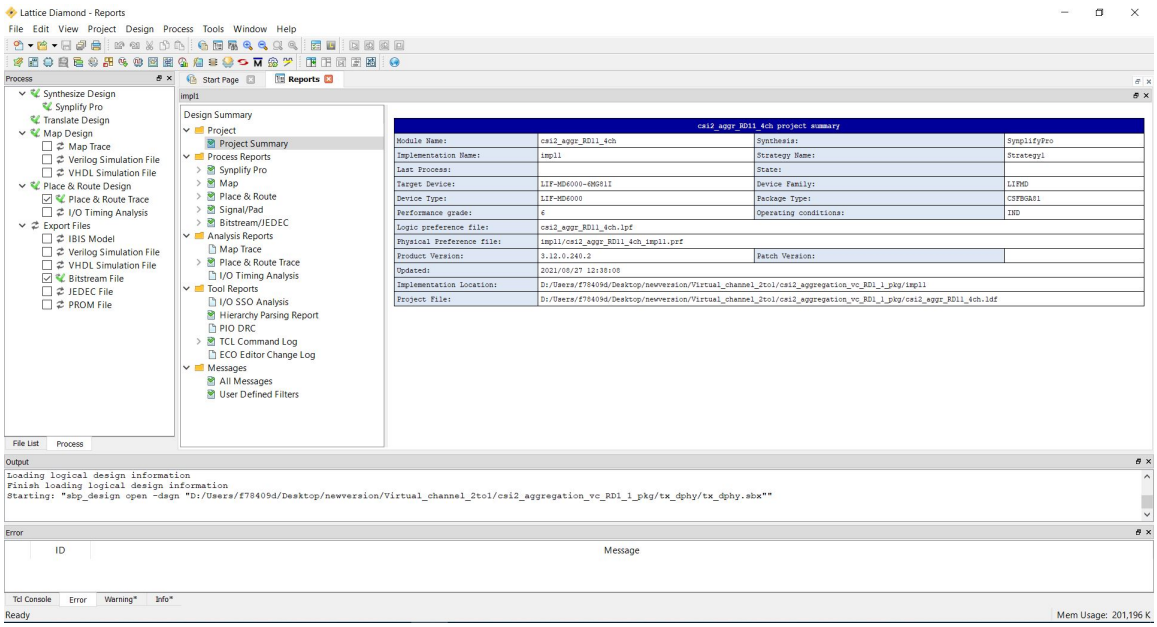


Figure 3.32: Lattice Diamond homepage

The homepage of Diamond offers:

- *File List* and *Process* views for controlling implementation
- *Start Page* and *Reports* views offering quick links to opening projects, software updates, on-line help and design reports
- *Output* window for warnings, errors, and scripting control
- Menus, icons, and controls for all integrated tool views

Lattice Diamond also includes an intuitive HDL text editor; in this specific case, Verilog has been adopted as Hardware Description Language even because a part of the top-level source code, which describes and implements the hardware blocks associated with the protocol, has been partially developed and provided by Lattice, in order to allow the user to focus exclusively on analyzing, modifying and customizing the code according to project needs.

Coming back to the block diagram of figure 3.31, one of the two Hard MIPI D-PHYs provided by CrossLink has been used as MIPI TX. Instead, the RX modules have been implemented by soft macros using programmable I/O blocks

available in banks 1 and 2 (MIPI D-PHY Soft IPs). Here, the two modules generating two MIPI CSI-2 data streams must be connected.

In order to create, configure and generate the hardware blocks, the integrated tool *Clarity Designer* has been used: it is the interface to the Lattice catalog that contains a great number of functional modules and IPs to be smoothly embedded and customized within the project.

The starting point has been the creation of the receiving and transmitting blocks.

Through the *Catalog* tab of the *Clarity Designer* tool and by accessing the *Lattice IP Server*, the respective IPs have been downloaded and subsequently installed: Versions 1.4 and 1.3 have been chosen for CSI-2 D-PHY receivers and transmitter respectively, since they are the most updated versions to support business requirements. Figure 3.33 shows the configuration window for the CSI-2 D-PHY receiver IP in the *Clarity Designer* environment: this channel has been renamed *rx_ch0*. The following parameters have been set:

- *RX Interface*: CSI-2
- *Number of RX Lanes*: 4
- *RX Gear*: 8
- *RX DPHY IP*: Soft DPHY
- *RX Line Rate*: 420 (Mbps)
- *DPHY Clock Mode*: Continuous

The other minor settings, useless for this specific application, are disabled (as default).

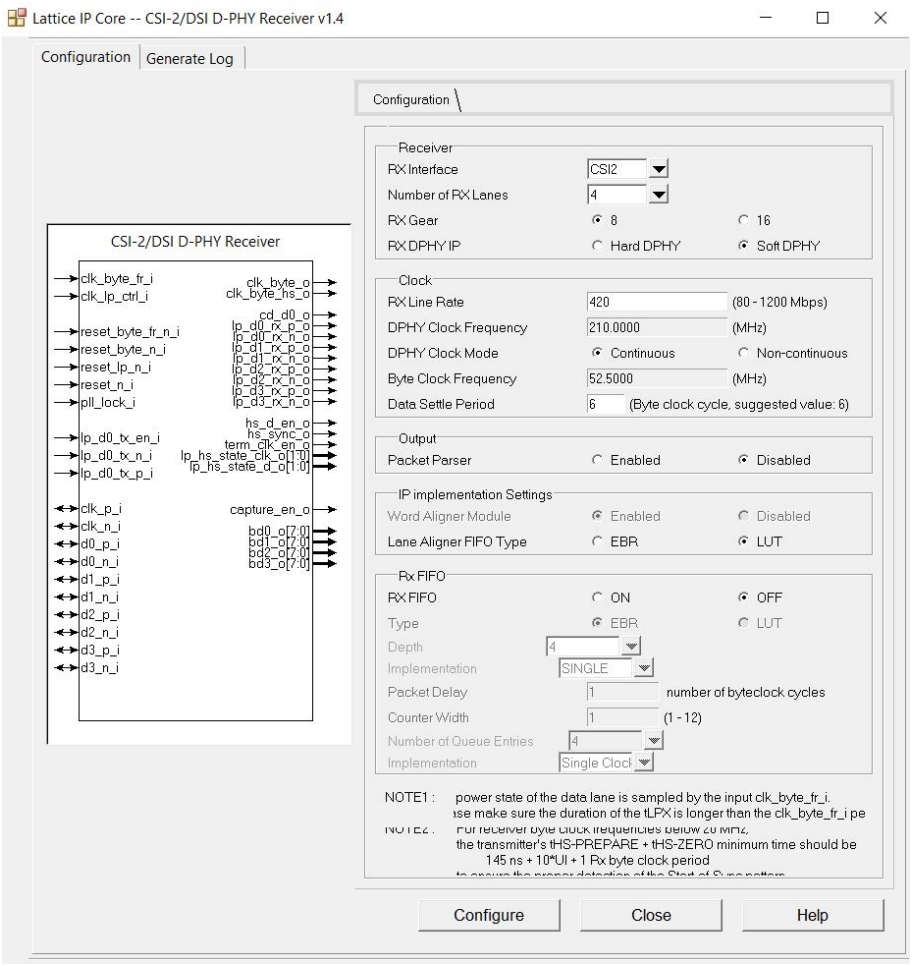


Figure 3.33: *rx_ch0* D-PHY soft IP configuration in *Clarity Designer*

The configuration fields for the first receiver channel have been set to allow the use of the current available hardware release of LiDAR optoelectronic sensor working at an operating frequency of 210 MHz (420 Mbps per Lane), in compliance with the maximum frequency of 1.07 Gbps per Lane of the programmable I/O blocks. This module uses four MIPI CSI-2 data Lanes with RAW14 data type. The *Byte Clock Frequency* is automatically calculated by the tool starting from the *DPHY Clock Frequency* via a frequency divider, setting the proper value of *RX Gear*. It is used to generate, from *rx_ch0*, the reference transmitter IP clock. The *DPHY Clock Mode* is continuous, it always works in high-speed (never in low-power); in order to test a first version of the functionality, this mode significantly simplifies clock management. By the way, it is the most

used setting when managing MIPI communication. Clearly, both the receiving channels have been implemented as Soft D-PHY on CrossLink banks 1 and 2 because, unlike the two Hard MIPI D-PHY blocks, they cannot be used for transmitting channels.

Purely by way of example, the top-level source code snippet related to the first receiver block instantiation is shown below.

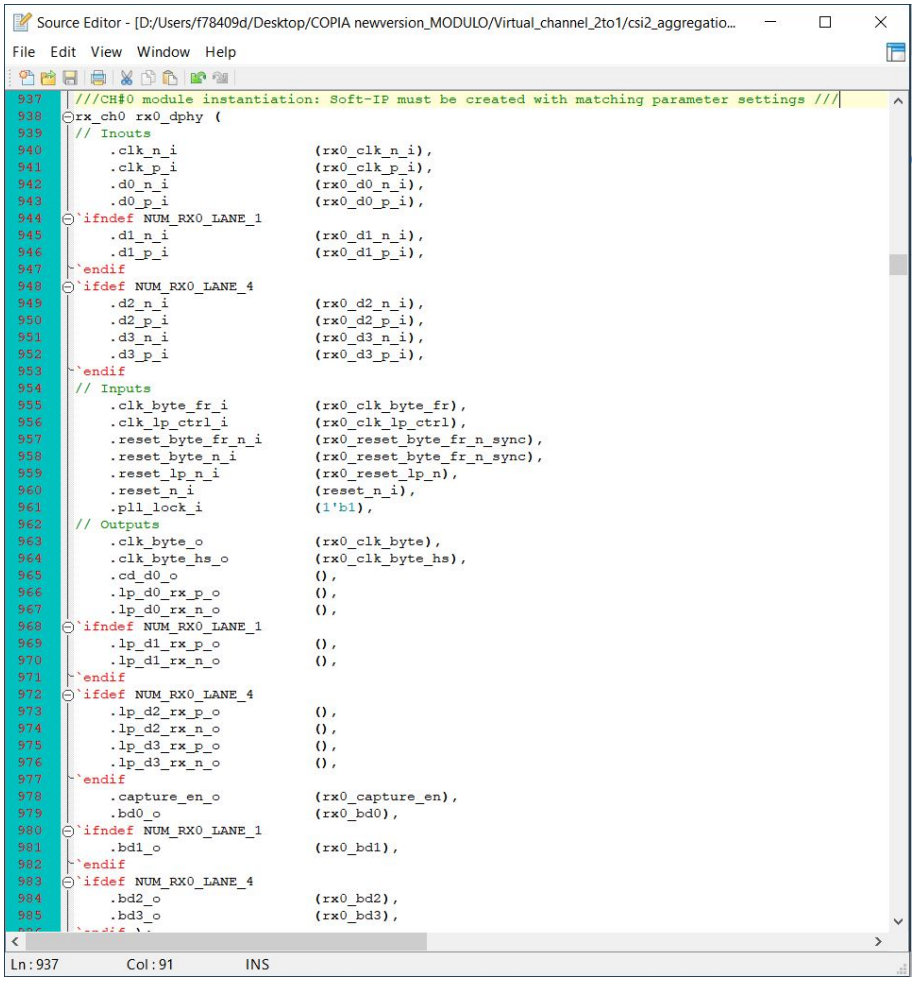


Figure 3.34: *rx_ch0* instantiation in the top-level source code

A second receiver channel has been created following the same procedure (separate IPs creation is required for each channel even through their configurations are the same): this channel has been renamed *rx_ch1*. The development device connected to this second channel generates a pattern on four MIPI CSI-2 data

Lanes exactly like the first one; however, it works at a frequency of 200 MHz managing RAW12 data. All other parameters listed have been set as for the first optoelectronic module and are shown in figure 3.35.

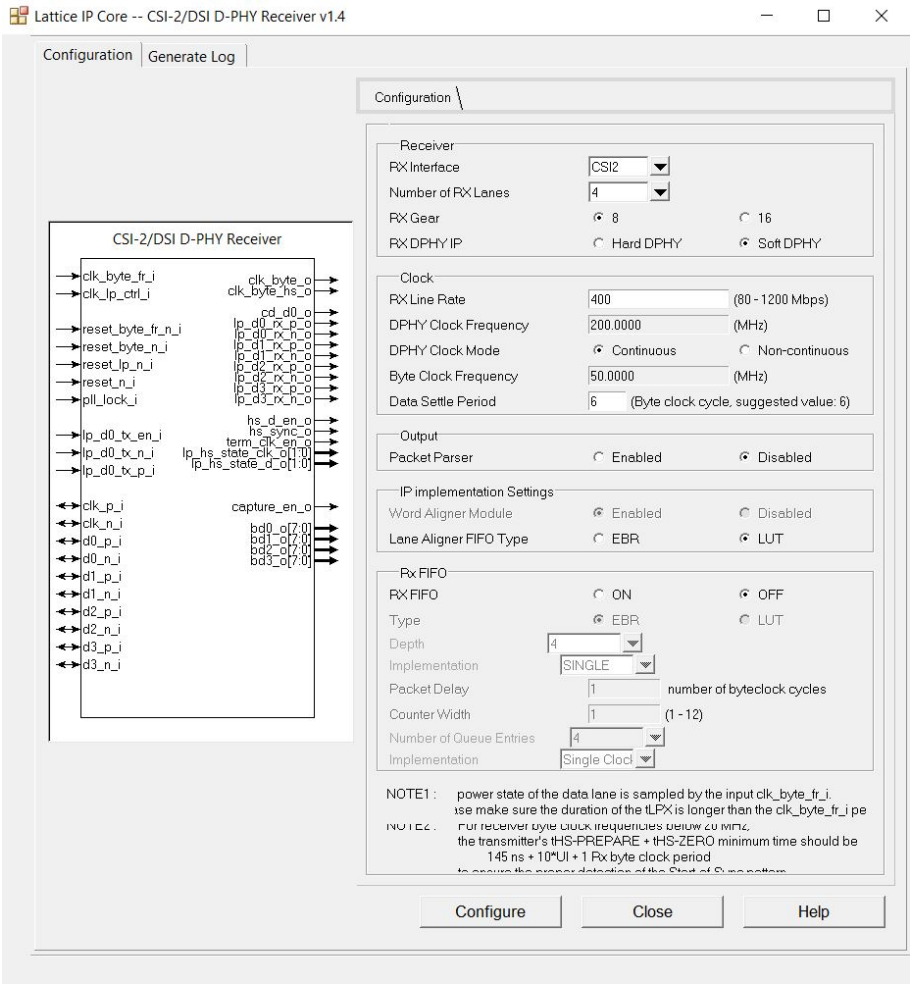


Figure 3.35: *rx_ch1* D-PHY soft IP configuration in *Clarity Designer*

Up to this point, a sub-design file, renamed *rx_dphy*, has been created; it includes the two receiver blocks: *rx_ch0* and *rx_ch1*. This file can be accessed and edited from the *File List* tab, allowing in a simple way modifications to design parameters. In this way it will be possible to easily replace the pattern generator with the second module LiDAR sensor.

Figure 3.36 shows the configuration window for the CSI-2 Hard MIPI D-PHY transmitter IP in the *Clarity Designer* environment: this channel has been re-

named *tx_ch*. The following parameters have been set:

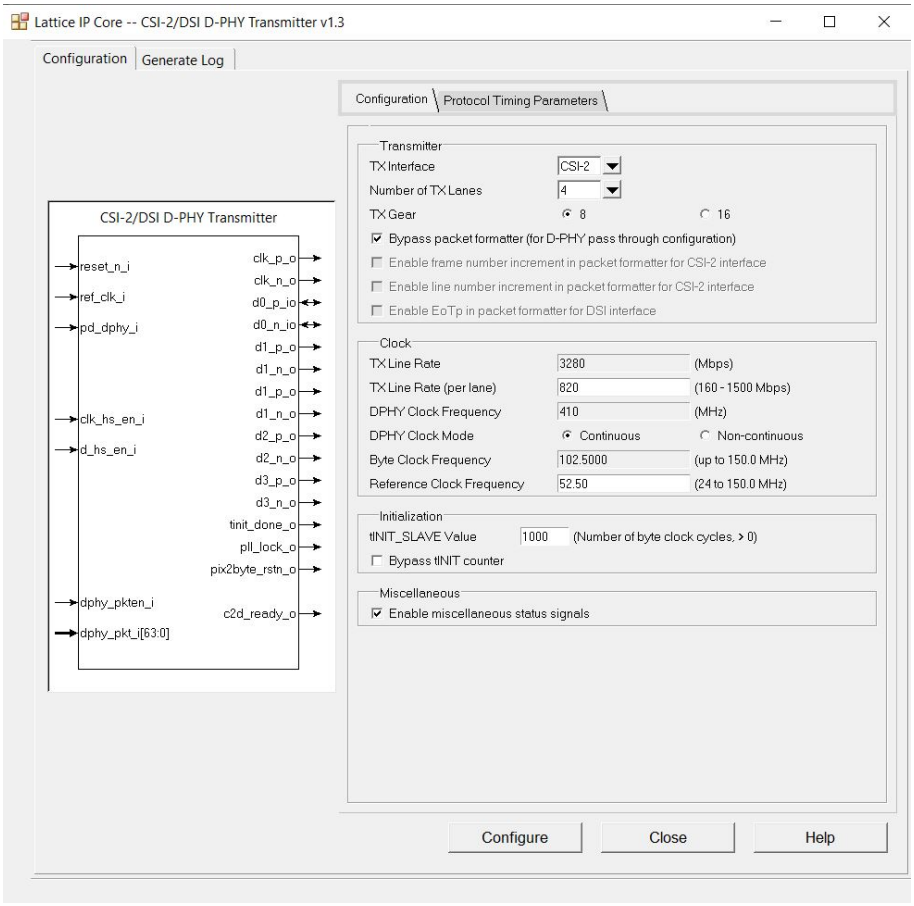


Figure 3.36: *tx_ch* Hard D-PHY IP configuration in *Clarity Designer*

TX Line Rate (per lane) is one of the key factors of this design. In fact, the transmitter bandwidth must be fast enough to cover all of receiver channels data transmissions. Calculating the minimum required TX Lane bandwidth is derived from the following equation:

$$TX\ Line\ Rate\ (per\ lane) = \frac{1}{m}\sum_{k=0}^{n-1} number\_of\_Lanes\_RX[k] * Line\_Rate\_RX[k] \tag{3.1}$$

where m is the number of TX Lanes (4 in this case) and n is the number of RX channels (2 in this case).

Then, $TX\ Line\ Rate\ (per\ lane) = (4 * 400 + 4 * 420)/4 = 820\ Gbps$ (410 MHz).

So, the total *TX Line Rate* is 3280 Mbps (minimum value) and represents the total transmitter channel bandwidth, in compliance with the maximum frequency of 6 Gbps of the Hard MIPI D-PHY blocks. The minimum frequency value has been set to optimize electromagnetic compatibility and power consumption. The *Byte Clock Frequency* is automatically calculated by the tool starting from the *TX Line Rate (per lane)* via a frequency divider, setting the proper value of *TX Gear*. The *DPHY Clock Mode* is continuous, it always works in high-speed (never in low-power). In addition, this reduces the overhead of TX data transmission. The *Reference Clock Frequency* is set according to the one associated to *rx_ch0*, as explained before.

At this point, another sub-design file, renamed *tx_dphy*, has been generated; it includes the transmitter block *tx_ch*. This channel also uses four MIPI CSI-2 data Lanes and provides Virtual Channel data aggregation coming from the two receiver blocks.

All other minor unexplained parameters shown in *Clarity Designer* windows are set by default according to FPGA specifications.

The instance names of receiver and transmitter channels blocks created in *Clarity Designer* must match the ones in the top-level design file. Purely by way of example, the top-level source code snippet related to the transmitter block instantiation is shown below.

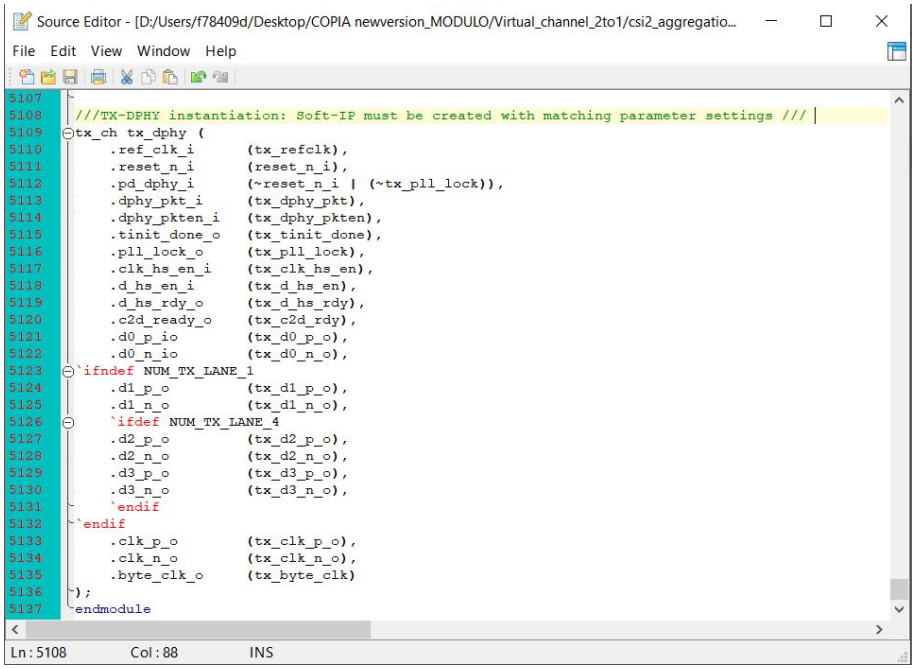


Figure 3.37: *tx_ch* instantiation in the top-level source code

To sum up, the top-level design file consists of all the modules listed below:

- *rx_dphy*
- *lane_align*
- *csi2_parser*
- *rx_buffer*
- *tdm_ctrl*
- *tx_dphy_if*
- *tx_dphy*

Two Lane aligners blocks have been added to the design (one for each receiver channel): this module takes the byte data coming out from the receiver, checks the D-PHY sync sequence (HS Sync-sequence in table 3.6) and then aligns the byte data timing among Lanes.

Then, one CSI-2 parser module has been instantiated for each RX channel to handle CSI-2 protocol and Virtual Channel overwrite. Upon receiving byte

data from receivers or Lane aligners, *csi2_parser* detects Short and Long Packet Headers; next, it replaces default VC values with the ones specified in synthesis directives file (explained later on) and calculates ECC value based on this new VC. As a result, VC and ECC are replaced with new values and sent to the next module (*rx_buffer*).

FIFO buffers, one for each receiver channel, store the high-speed transmission data to be read out, therefore, they must be suitably sized. Since the optoelectronic module form factor is a sensitive information, only that of the pattern generator is reported below. The development device data sheet specifies that the frame resolution is 3840 x 1024 pixel (h x v). Considering that RAW12 data type implies that each pixel is represented by 12 bits, the total number of bits contained in the Long Packet Payload (representing a frame line) can be derived as follows:

$$\#bit = h * pixel\_depth = 3840\,pixel * 12\,bit = 46080\,bit \tag{3.2}$$

In addition to Payload bits, 32 bits of Packet Header and 16 bits of Packet Footer should be considered (as reported in the MIPI CSI-2 specification [14]). Hence, the RX buffers depth size has been set to the maximum available value (*2048*) because data size of one HS transmission must not exceed FIFO size; this means that, in this configuration, each buffer can store 2048 x 32 bit of data. This value affects necessary Embedded Block RAM used in the device. Number of EBR per RX channel needed is calculated as follows:

$$\#EBR = (BUFFER\_DEPTH/512) * 2 = 8 \tag{3.3}$$

By specification, total number of EBR must not exceed 20, which are those physically present in the device. So, along with ready flag signal assertion, the

buffer notifies *tdm_ctrl* that high-speed data is ready to be acquired from this channel.

tdm_ctrl is a single module that monitors the ready flag of the RX channels and takes the HS data transmission of the selected channel. When multiple RX channels data are available, signified by the assertion of the ready flags, the channel selection is based on a Round Robin algorithm (arrival order scheduling).

Finally, data are sent to the last module for transmission: *tx_dphy_if*. This module reallocates byte data coming from *tdm_ctrl* and sends them to *tx_dphy*. *tx_dphy_if* makes both clock and data Lanes from LP to HS mode in order to begin high-speed data transmission.

Synthesis

Within the project, a *synthesis_directives.v* file has been included in the *File List* window. It is used for both design compilation by Diamond and synthesis run by Synopsys Synplify Pro for Lattice integrated tool.

Source Editor - [D:/Users/f78409d/Desktop/TESI newversion_MODULO/Virtual_channel_2to1/csi2_aggregation_...

File Edit View Window Help

```
1  ///// Synthesis Directives for MIPI CSI2 Aggregation /////
2
3  //`define EXT_REF_CLK   // External reference clock not used
4
5  ///// Directives for RX /////
6
7  `define NUM_RX_CH_2     // Number of RX channels: 2, 3, 4, or 5
8
9  `define NUM_RX0_LANE_4  // Number of Lanes RX0: 1, 2, or 4
10 `define NUM_RX1_LANE_4  // Number of Lanes RX1: 1, 2, or 4
11
12 //`define RX0_DPHY_HARD // not used (Hard DPHY used as TX)
13 //`define RX1_DPHY_HARD // not used (Hard DPHY used as TX)
14
15 `define RX0_GEAR_8  // 8 or 16; 16 is not allowed in case of 4 lane
16 `define RX1_GEAR_8  // 8 or 16; 16 is not allowed in case of 4 lane
17
18 `define RX0_CLK_MODE_HS_ONLY    // DPHY Clock Mode: continuous, always in high-speed
19 //`define RX0_CLK_MODE_HS_LP    // Non-continuous clock (high-speed and low-power) not used
20 `define RX1_CLK_MODE_HS_ONLY    // DPHY Clock Mode: continuous, always in high-speed
21 //`define RX1_CLK_MODE_HS_LP    // Non-continuous clock (high-speed and low-power) not used
22
23 //`define WORD_ALIGN    // effective only for RX Hard DPHY (not used)
24
25 `define RX0_LANE_ALIGN
26 `define RX1_LANE_ALIGN
27
28 `define RX0_NEW_VC 4'd0 // RX0 Virtual Channel assignment
29 `define RX1_NEW_VC 4'd1 // RX1 Virtual Channel assignment
30
31 `define RX0_FRAME_COUNT // Frame Counter value in FS/FE Short Packet
32 `define RX1_FRAME_COUNT // Frame Counter value in FS/FE Short Packet
33
34 `define RX0_FRAME_COUNT_MAX 16'd2   // maximum frame count value; 2 - 65535
35 `define RX1_FRAME_COUNT_MAX 16'd3   // maximum frame count value; 2 - 65535
36
37 `define RX0_BUFFER_DEPTH_2048   // 512, 1024, or 2048
38 `define RX1_BUFFER_DEPTH_2048   // 512, 1024, or 2048
39
40 ///// Directives for TX /////
41
42 `define NUM_TX_LANE_4   // Number of Lanes TX: 1, 2, or 4
43 `define TX_GEAR_8   // 8 or 16
44
45 `define TX_CLK_MODE_HS_ONLY // DPHY Clock Mode: continuous, always in high-speed
46 //`define TX_CLK_MODE_HS_LP // Non-continuous clock (high-speed and low-power) not used
```

Figure 3.38: *synthesis_directives.v* file

Beyond parameters and blocks already described before (number of receiver channels, number of Lanes, clock mode and gear for receivers and transmitter, Lane aligners, buffer depth), new ones have been defined. First of all, a different unique Virtual Channel identification has been assigned to each receiver: VC 0 (*4'd0*) for RX0 and VC 1 (*4'd1*) for RX1. Thus, incoming Virtual Channel values on RX CSI-2 data are overwritten by these new VC values. Next, the Frame Counter has been enabled and the maximum value that this field (present in the Short Packet) can take has been defined; this value indicates which frame is being processed.

All the other parameters, defined in the file as comments, are variations that could be enabled in case of changes of the current configuration. For example, the third line of synthesis directives would define an external reference clock used to generate the TX reference clock. Typically, this solution is not convenient because it would increase the pin count and the Bill Of Material. As previously specified, for this application the RX0 Byte Clock Frequency is directly suitable to generate the TX IP Reference Clock Frequency, without the use of any external clock source. An external reference clock is necessary if the continuous receiver byte clocks are not appropriate to generate the desired clock for TX D-PHY. The top-level source code snippet of figure 3.39 shows the adopted solution.

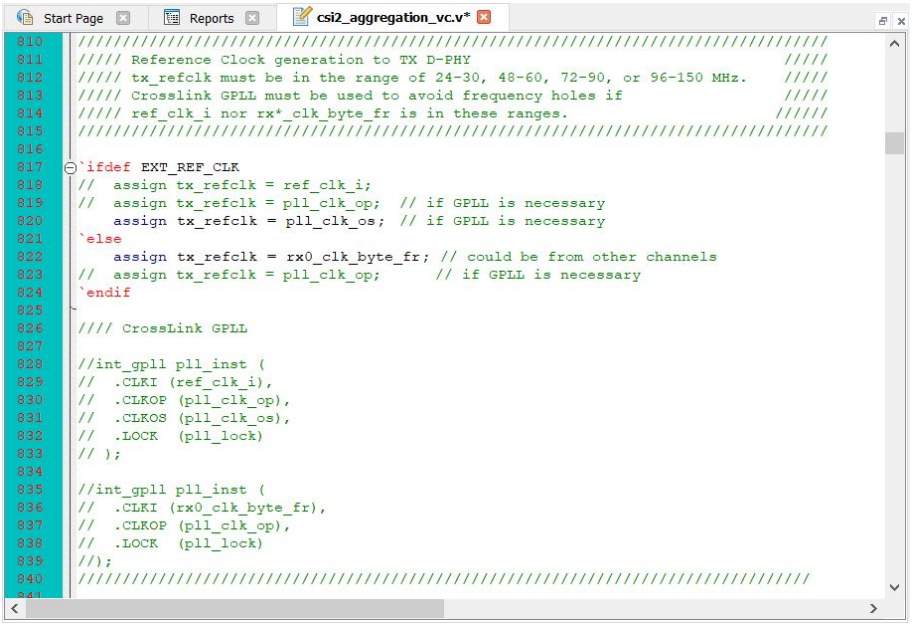


Figure 3.39: Top-level source code snippet for clock generation

Clearly, all these synthesis settings shall correspond to those set in the IP blocks and in the top-level source code.

As a result, the project has been compiled to verify the correctness or presence of errors in the source code.

Diamond includes the industry-leading synthesis solution, Synopsys Synplify

Pro for Lattice. After design has been successfully compiled and synthesized, the tool automatically produces an RTL view: a high-level graphic representation of the design for analysis and cross-probing with source code.

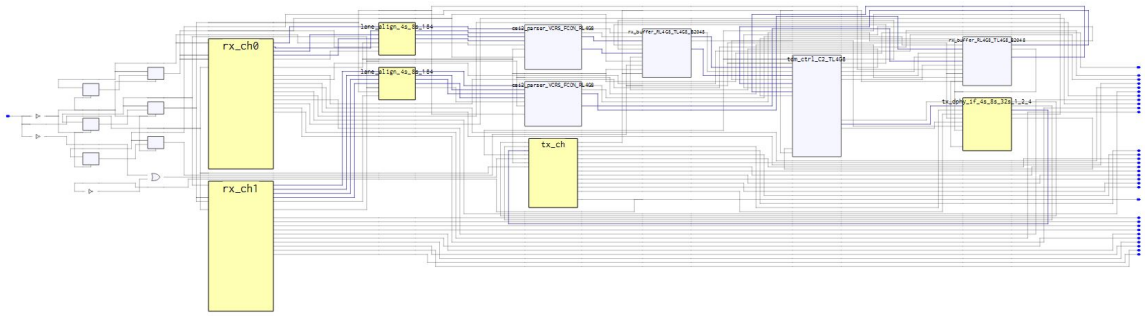


Figure 3.40: RTL view

Enlarging the schematic or selecting an object, instances, ports and nets can be analyzed to understand if the code matches the project design. For example, figure 3.41 (obtained by enlarging the left part of the previous one), shows the reset logic implementation of the device, which reaches all the blocks present in the design; the *reset_n_i* input port, is a physical pin which must be subsequently mapped.

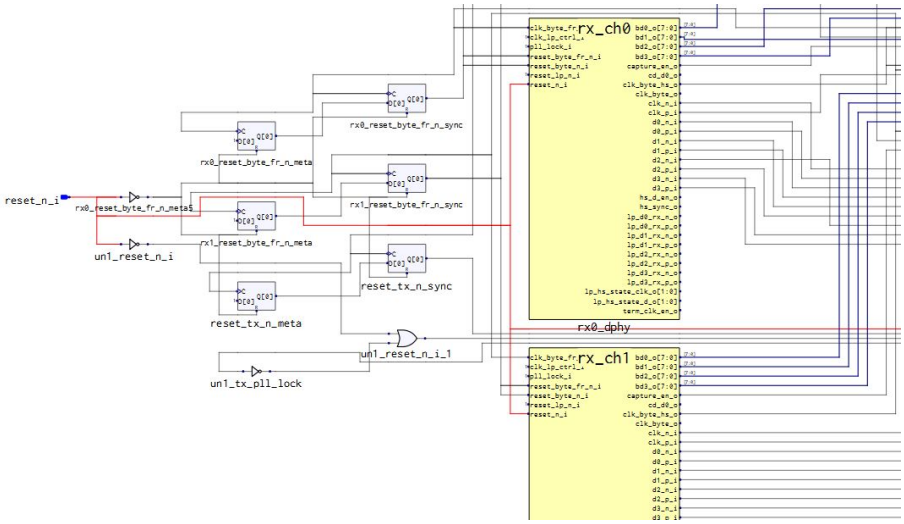


Figure 3.41: Reset logic in RTL view

Implementation

A key component of Lattice Diamond is the *Spreadsheet View*. As soon as the target device has been specified, it provides an interactive tabular format for viewing and assigning design constraints such as port and pin assignments, global preferences, timing preferences and more, to optimize placement and routing.

The *Port Assignment* sheet provides the design signals list and offers the possibility to view or edit pin location and other attributes by entering them directly on the spreadsheet. Ports are grouped by direction (input, output or bidirectional).

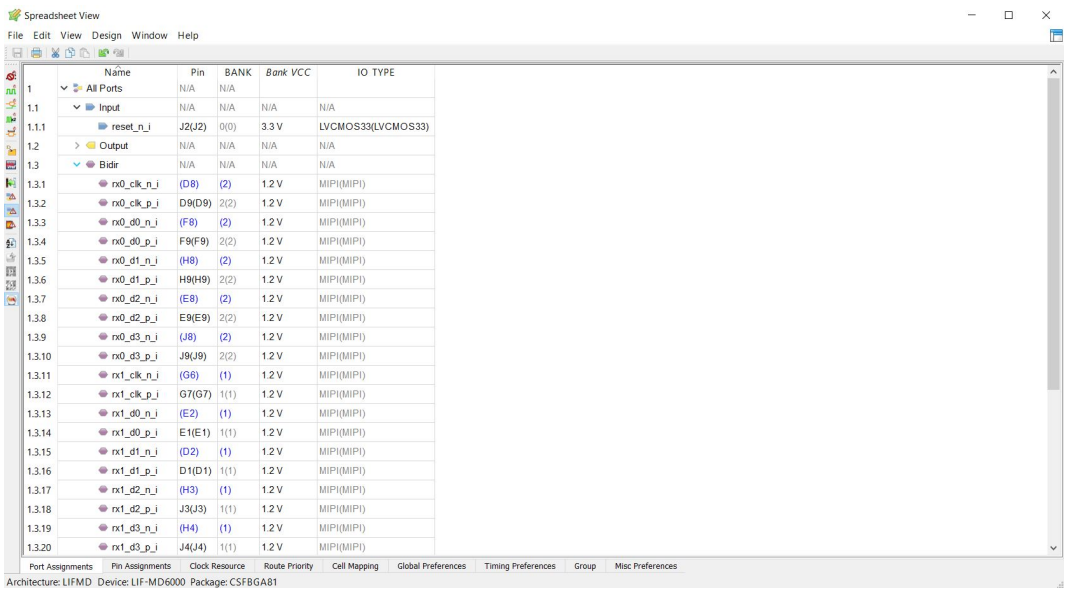
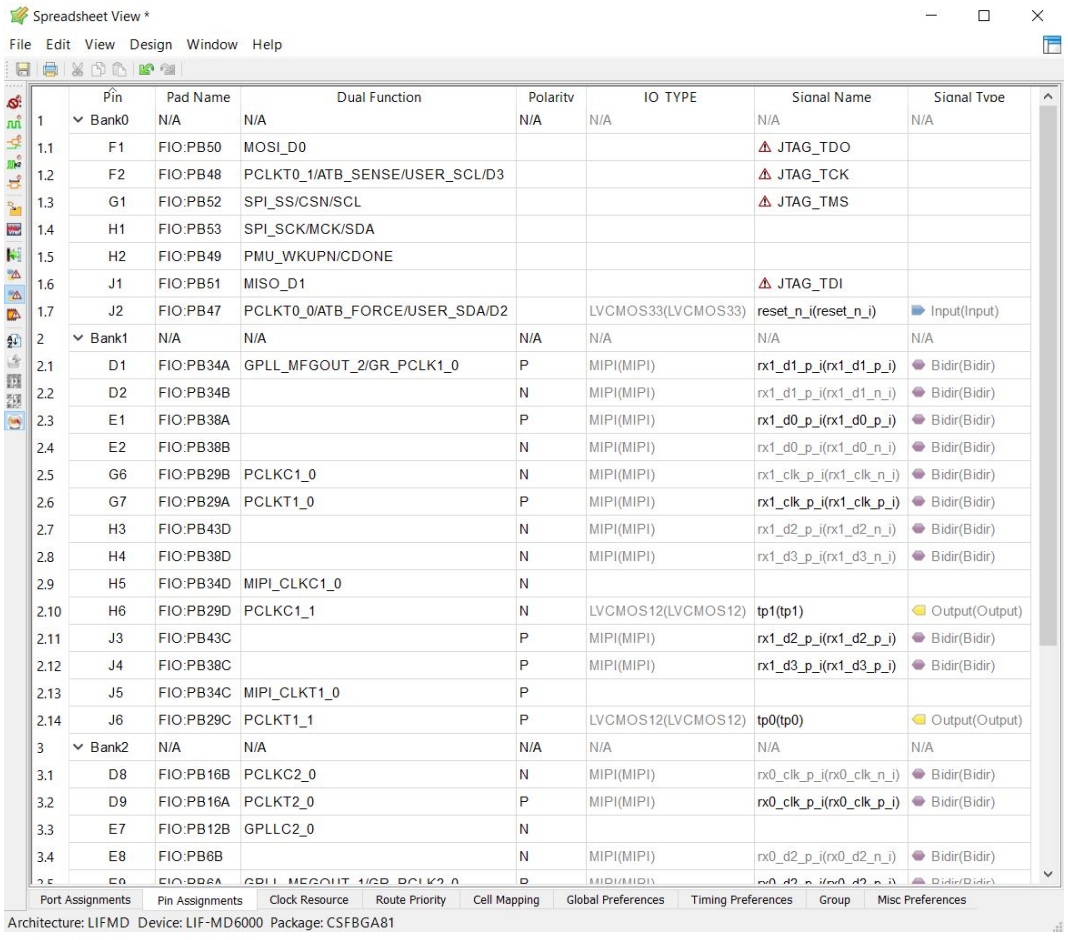


Figure 3.42: *Port Assignment* sheet overview

First of all, programmable I/O banks have been assigned to both the receiver channels: in the *BANK* column, all the MIPI RX0 differential signals (four data Lanes plus one clock Lane) have been assigned to bank 2 and all those related to RX1 to bank 1. In the *Pin* column, all the pins associated to the receivers ports have been assigned according to the EVB hardware architecture. Obviously, the transmitter signals are not set as they have been implemented

in the Hard MIPI D-PHY block and, therefore, mapped to specific fixed pins of the device. Afterwards, the voltages related to the three banks have been set: 3.3 V for bank 0 (V_{CCIO0}) and 1.2 V for banks 1 and 2 (V_{CCIO1} and V_{CCIO2}), in compliance with MIPI protocol. The last column assigns the protocol type to each I/O port.

The *Pin Assignment* sheet provides the device pin list and allows to shows or edit the signals assignments that have been made. Pins are grouped by bank number. In this window, the differential pin pairs are displayed but only the signals with positive polarity are assigned; the complementary ones are assigned accordingly automatically. Anyway, the sheet provides all the information about the signals associated with the device pins.



	Pin	Pad Name	Dual Function	Polarity	IO TYPE	Signal Name	Signal Type
1	Bank0	N/A	N/A	N/A	N/A	N/A	N/A
1.1	F1	FIO:PB50	MOSI_D0			JTAG_TDO	
1.2	F2	FIO:PB48	PCLKT0_1/ATB_SENSE/USER_SCL/D3			JTAG_TCK	
1.3	G1	FIO:PB52	SPI_SS/CSN/SCL			JTAG_TMS	
1.4	H1	FIO:PB53	SPI_SCK/MCK/SDA				
1.5	H2	FIO:PB49	PMU_WKUPN/CDONE				
1.6	J1	FIO:PB51	MISO_D1			JTAG_TDI	
1.7	J2	FIO:PB47	PCLKT0_0/ATB_FORCE/USER_SDA/D2		LVCMOS33(LVCMOS33)	reset_n_i(reset_n_i)	Input(Input)
2	Bank1	N/A	N/A	N/A	N/A	N/A	N/A
2.1	D1	FIO:PB34A	GPLL_MFGOUT_2/GR_PCLK1_0	P	MIPI(MIPI)	rx1_d1_p_i(rx1_d1_p_i)	Bidir(Bidir)
2.2	D2	FIO:PB34B		N	MIPI(MIPI)	rx1_d1_p_i(rx1_d1_n_i)	Bidir(Bidir)
2.3	E1	FIO:PB38A		P	MIPI(MIPI)	rx1_d0_p_i(rx1_d0_p_i)	Bidir(Bidir)
2.4	E2	FIO:PB38B		N	MIPI(MIPI)	rx1_d0_p_i(rx1_d0_n_i)	Bidir(Bidir)
2.5	G6	FIO:PB29B	PCLKC1_0	N	MIPI(MIPI)	rx1_clk_p_i(rx1_clk_n_i)	Bidir(Bidir)
2.6	G7	FIO:PB29A	PCLKT1_0	P	MIPI(MIPI)	rx1_clk_p_i(rx1_clk_p_i)	Bidir(Bidir)
2.7	H3	FIO:PB43D		N	MIPI(MIPI)	rx1_d2_p_i(rx1_d2_n_i)	Bidir(Bidir)
2.8	H4	FIO:PB38D		N	MIPI(MIPI)	rx1_d3_p_i(rx1_d3_n_i)	Bidir(Bidir)
2.9	H5	FIO:PB34D	MIPI_CLKC1_0	N			
2.10	H6	FIO:PB29D	PCLKC1_1	N	LVCMOS12(LVCMOS12)	tp1(tp1)	Output(Output)
2.11	J3	FIO:PB43C		P	MIPI(MIPI)	rx1_d2_p_i(rx1_d2_p_i)	Bidir(Bidir)
2.12	J4	FIO:PB38C		P	MIPI(MIPI)	rx1_d3_p_i(rx1_d3_p_i)	Bidir(Bidir)
2.13	J5	FIO:PB34C	MIPI_CLKT1_0	P			
2.14	J6	FIO:PB29C	PCLKT1_1	P	LVCMOS12(LVCMOS12)	tp0(tp0)	Output(Output)
3	Bank2	N/A	N/A	N/A	N/A	N/A	N/A
3.1	D8	FIO:PB16B	PCLKC2_0	N	MIPI(MIPI)	rx0_clk_p_i(rx0_clk_n_i)	Bidir(Bidir)
3.2	D9	FIO:PB16A	PCLKT2_0	P	MIPI(MIPI)	rx0_clk_p_i(rx0_clk_p_i)	Bidir(Bidir)
3.3	E7	FIO:PB12B	GPLLC2_0	N			
3.4	E8	FIO:PB6B		N	MIPI(MIPI)	rx0_d2_p_i(rx0_d2_n_i)	Bidir(Bidir)

Figure 3.43: *Pin Assignment* sheet overview

At this point, the design has been mapped, placed and routed through the *Process* window. This step maps the design to the target FPGA converting the logical components into placeable ones.

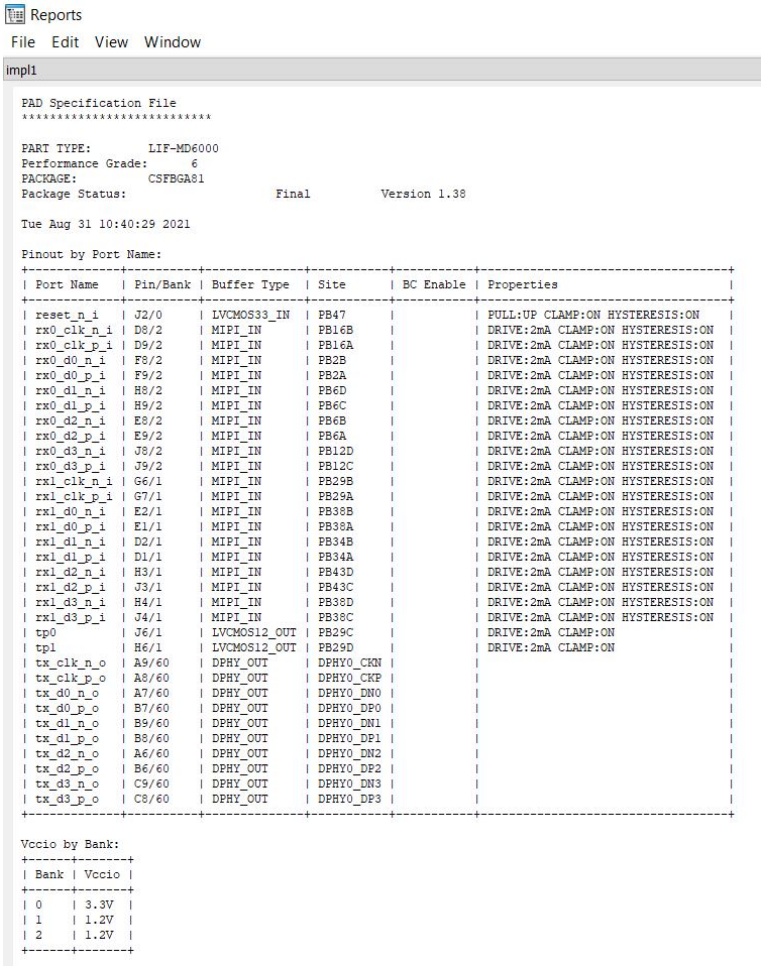
Through the *File List*, the one related to LPF constraint can be accessed. It is the source file storing only user-defined logical preferences.

Figure 3.44: The Logical Preference File (.lpf)

It may be noted that even in this file there are no references to the pins location of the transmitter block (as it is implemented on the Hard MIPI D-PHY); it has been only indicated which of the two Hard banks to use to allocate the transmitter block (*MIPIDPHY0*).

Mapping, placing and routing processes depend on the .lpf file. Thus, if the design was mapped but this file was modified afterwards, mapping and any downstream processes must be rerun.

At the end of these processes a report file is produced, allowing the verification of the correctness of the pins assignment and other settings.



Reports

File Edit View Window

impl1

```
PAD Specification File
**********************************

PART TYPE:        LIF-MD6000
Performance Grade:      6
PACKAGE:          CSFBGA81
Package Status:                   Final          Version 1.38

Tue Aug 31 10:40:29 2021

Pinout by Port Name:
+-------------+---------+--------------+-----------+-----------+-----------------------------------+
| Port Name   | Pin/Bank | Buffer Type  | Site      | BC Enable | Properties                        |
+-------------+---------+--------------+-----------+-----------+-----------------------------------+
| reset_n_i   | J2/0     | LVCMOS33_IN  | PB47      |           | PULL:UP CLAMP:ON HYSTERESIS:ON    |
| rx0_clk_n_i | D8/2     | MIPI_IN      | PB16B     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_clk_p_i | D9/2     | MIPI_IN      | PB16A     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_d0_n_i  | F8/2     | MIPI_IN      | PB2B      |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_d0_p_i  | F9/2     | MIPI_IN      | PB2A      |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_d1_n_i  | H8/2     | MIPI_IN      | PB6D      |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_d1_p_i  | H9/2     | MIPI_IN      | PB6C      |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_d2_n_i  | E8/2     | MIPI_IN      | PB6B      |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_d2_p_i  | E9/2     | MIPI_IN      | PB6A      |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_d3_n_i  | J8/2     | MIPI_IN      | PB12D     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx0_d3_p_i  | J9/2     | MIPI_IN      | PB12C     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_clk_n_i | G6/1     | MIPI_IN      | PB29B     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_clk_p_i | G7/1     | MIPI_IN      | PB29A     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_d0_n_i  | E2/1     | MIPI_IN      | PB38B     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_d0_p_i  | E1/1     | MIPI_IN      | PB38A     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_d1_n_i  | D2/1     | MIPI_IN      | PB34B     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_d1_p_i  | D1/1     | MIPI_IN      | PB34A     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_d2_n_i  | H3/1     | MIPI_IN      | PB43D     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_d2_p_i  | J3/1     | MIPI_IN      | PB43C     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_d3_n_i  | H4/1     | MIPI_IN      | PB38D     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| rx1_d3_p_i  | J4/1     | MIPI_IN      | PB38C     |           | DRIVE:2mA CLAMP:ON HYSTERESIS:ON  |
| tp0         | J6/1     | LVCMOS12_OUT | PB29C     |           | DRIVE:2mA CLAMP:ON                |
| tp1         | H6/1     | LVCMOS12_OUT | PB29D     |           | DRIVE:2mA CLAMP:ON                |
| tx_clk_n_o  | A9/60    | DPHY_OUT     | DPHY0_CKN |           |                                   |
| tx_clk_p_o  | A8/60    | DPHY_OUT     | DPHY0_CKP |           |                                   |
| tx_d0_n_o   | A7/60    | DPHY_OUT     | DPHY0_DN0 |           |                                   |
| tx_d0_p_o   | B7/60    | DPHY_OUT     | DPHY0_DP0 |           |                                   |
| tx_d1_n_o   | B9/60    | DPHY_OUT     | DPHY0_DN1 |           |                                   |
| tx_d1_p_o   | B8/60    | DPHY_OUT     | DPHY0_DP1 |           |                                   |
| tx_d2_n_o   | A6/60    | DPHY_OUT     | DPHY0_DN2 |           |                                   |
| tx_d2_p_o   | B6/60    | DPHY_OUT     | DPHY0_DP2 |           |                                   |
| tx_d3_n_o   | C9/60    | DPHY_OUT     | DPHY0_DN3 |           |                                   |
| tx_d3_p_o   | C8/60    | DPHY_OUT     | DPHY0_DP3 |           |                                   |
+-------------+---------+--------------+-----------+-----------+-----------------------------------+

Vccio by Bank:
+------+-------+
| Bank | Vccio |
+------+-------+
| 0    | 3.3V  |
| 1    | 1.2V  |
| 2    | 1.2V  |
+------+-------+
```

Figure 3.45: Report file snippet

Programming

Finally, in the *Process* window, by simply clicking on the *Export Files* item, the programming file to be loaded into the FPGA has been generated. In particular, the *Bitstream File* selected generates the configuration file which contains all the design's configuration information that defines the internal logic and interconnections of the FPGA, as well as device specific information.

After the design has been placed and routed and the bitstream file generated, Diamond's fully integrated *Programmer* tool has been used to program the target device.

Programmer detects the cable type used, scans the device chain, creates the programming .xcf file, and downloads the data file into the device.

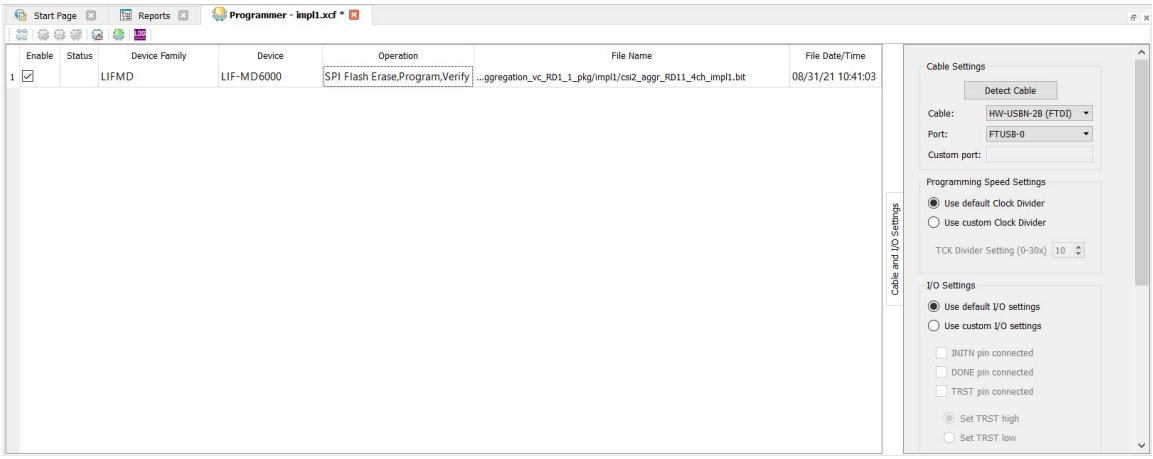


Figure 3.46: *Programmer* window overview

The *Programmer* view displays the device to be programmed for the current project (resulting from a scan action) and the selected bitstream file previously generated. In this specific case, the FPGA has been programmed using the Master SPI Configuration Mode. The bitstream file has been programmed into an external SPI Flash using a download cable. When the external Flash contains the configuration data, it is sufficient to turn the EVB off and on again, so that the FPGA can retrieve the configuration from the SPI Flash. Programming mode preferences are set in the *Operation* field window, as shown in figure 3.47.

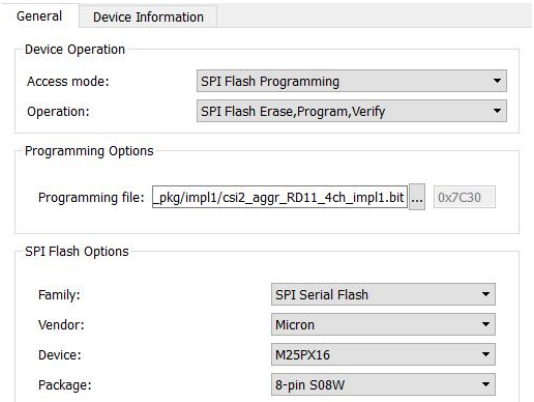


Figure 3.47: *Operation* window for programming mode

In this window the programming mode, the operations to be performed at each programming attempt, the bitstream file and specific information related to the external Flash mounted on the EVB have been set.

The presence of a set of LEDs on indicates that the device is programmed.

3.4.1 Measurement setup and hardware tests

Once the FPGA has been configured and programmed, the correct operation of the data aggregation circuit has been tested and verified directly on the CrossLink LIF-MD6000 Master Link Board. This has been made possible not only by the presence of an external SPI Flash memory on the evaluation board (which allows a quick FPGA re-programming) but above all by the wide availability of equipment in the company laboratory.

The measurement setup shown in figure 3.48 has been prepared for the experimental hardware tests of the dual MIPI CSI-2 to single MIPI CSI-2 data aggregation.

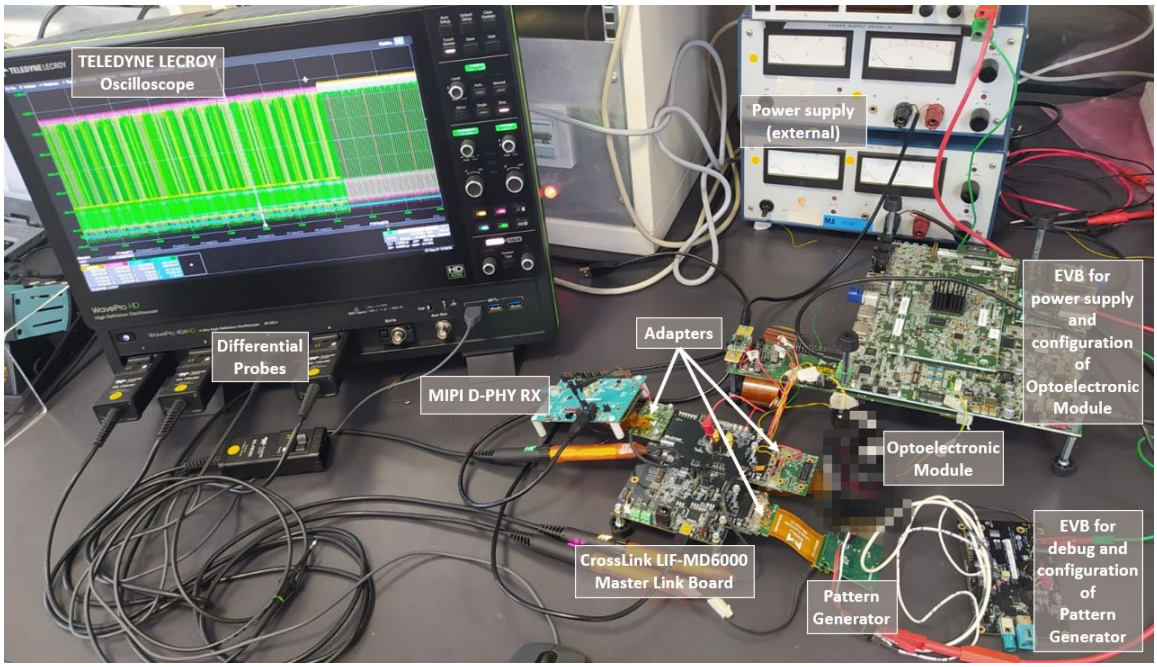


Figure 3.48: Measurement setup

First of all, the LiDAR optoelectronic module and the pattern generator have been connected, by means of adapters and rigid-flex PCBs, to RX Connector 1 and RX Connector 2 of the EVB, respectively.
Similarly, an evaluation board acting as MIPI CSI-2 receiver has been connected to TX Connector 1. In this regard, the System on Chip required by Marelli's final prototype (to process the aggregated data flow) has not been connected due to its temporary unavailability and not having to handle data processing in any way; thus, the only purpose of this EVB is to terminate the circuit allowing the correct visualization of the MIPI CSI-2 output aggregated data stream on the oscilloscope.
Note, that tests carried out to validate the functionality exploit a data pattern of the optoelectronic module and not its actual projected and reflected laser beams, forbidden for prototypes due to eye-safety issues. For this reason, both the optoelectronic module and the pattern generator need two additional evaluation boards, whose aims are the devices activation and the two MIPI CSI-2 input patterns generation. Specifically, these steps are carried out through proprietary software tools that allow the configuration of internal registers in which data format and working frequencies are specified. With regard to the development device, the selected data format and frequency value are the closest to the actual LiDAR module ones (210 MHz and RAW14 are not available in the setting options).
Just by way of explanation, a more detailed image presenting the CrossLink LIF-MD6000 Master Link Board and the receiving and transmitting modules is provided below.

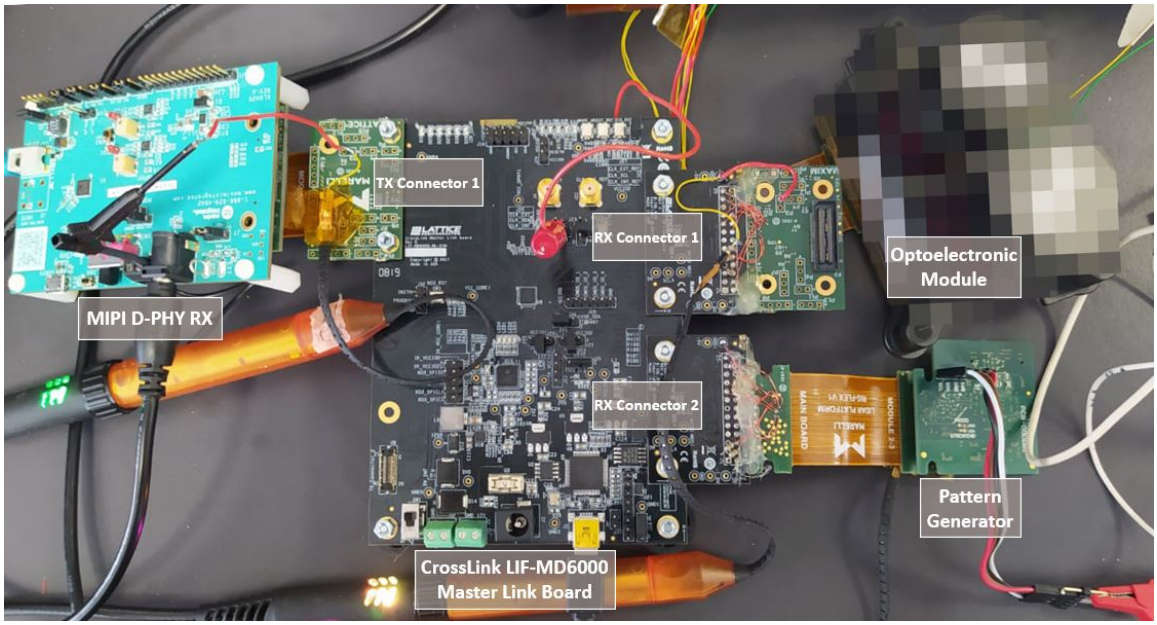


Figure 3.49: Detailed measurement setup

In order to monitor not only the incoming MIPI CSI-2 data stream but especially the outgoing one, thus verifying the correctness of the subsequent data aggregation, a Teledyne LeCroy high definition oscilloscope, with 4 GHz bandwidth, 12 bit of vertical resolution and 20 GS/s sample rate, has been employed to provide accuracy, details and precision. As can be clearly seen, the technical specifications of this oscilloscope largely cover the bandwidth of the application to be tested: remember that the fastest signal to check, i.e. the output, has a clock frequency of 410 MHz. In addition, its powerful processing system combined with the integrated *MIPI CSI-2 (D-PHY)* protocol packet allow to take advantage of the serial decoding options to check whether the captured waveforms are protocol compliant. Four high-bandwidth differential probes have been further employed to capture MIPI signals, soldering them to the traces vias available on the adapters, based on measurements of interest to be made.

Obviously, in order to validate that the design works properly, the four data Lanes and the clock Lane have been all acquired and decoded on both the receiving and the transmitting channels. However, having four channels on the oscilloscope, tests have been performed on five different setups to verify the behavior of each Lane on

the three channels, but, only the acquisitions related to Lane 0 of each channel are shown below, allowing to simultaneously monitor the behavior of both the receivers and the transmitter, therefore, the two distinct data streams in input and the output that aggregates the previous ones.

Single-ended measurements have been carried out in correspondence of each channel; probes have been soldered between the positive Line of Lane 0 and ground to display transitions from high-speed to low-power mode, and vice versa.

Figure 3.50 shows a zoom of the frames captured on the three channels.

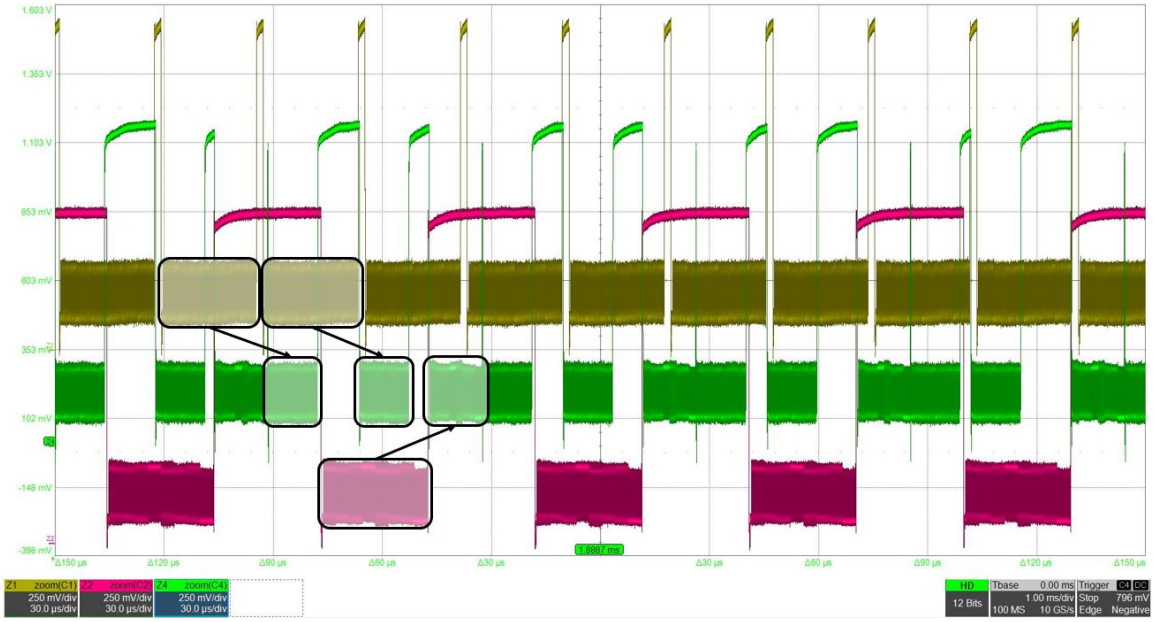


Figure 3.50: Data streams acquisition

Three distinct MIPI CSI-2 data streams can be recognized. They are arranged in a sequence of "parts" of Long Packets transmitted at high-speed and characterized by low swing (about 200 mV), spaced by short duration low-power states with a large swing (about 1.2 V). It is important to specify that "parts" of the Long Packet are depicted ("parts" of frame line) because only Lane 0 has been captured, not all four (data stream is distributed as a sequence of bytes across four Lanes as shown in figure 3.5). Moreover, Short Packets are not visible in this zoomed image as they are only expected for the beginning and end of the frame.

In particular, the yellow waveform (top) represents the RAW14 data stream sent by the 210 MHz optoelectronic module, the magenta one (bottom) the RAW12 data flow received from the development device working at 200 MHz. Finally, the bright green track (center) displays the 410 MHz output data stream on the transmitting channel, resulting from the Virtual Channel data aggregation.

From this image, FIFO buffer operation can be clearly understood: each Long Packet coming from the receiver channels is fully buffered and then transmitted out as soon as its loading is complete, as indicated by the arrows in the figure. Of course, the data written into the buffer first comes out of it first.

Overall, when the System on Chip is connected, it will receive the aggregated data and will be able to distinguish, recompose and process the lines of the two frames acquired, referring to the Virtual Channel contained in the Packet Header of each Long and Short Packet; according to this project settings, the VC identifier 0 indicates the frame coming from the optoelectronic module, the VC identifier 1 the one received by the development device.

At this point, the three data streams have been entirely decoded by the oscilloscope, thanks to the presence of an integrated decoding packet that ensures compliance of selected waveforms with the MIPI CSI-2 protocol.

Figure 3.51 illustrates the raw data on Lanes 0 decoded following the order in which they enter and exit the buffer: incoming bytes are correctly aggregated in output (as shown by the black boxes in figure).

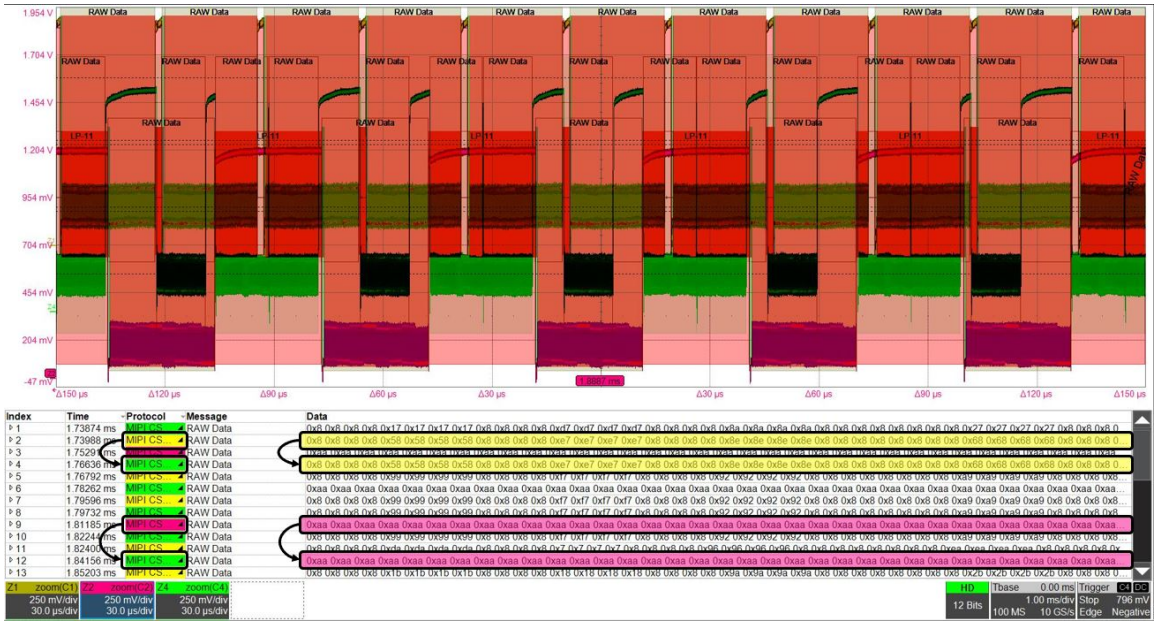


Figure 3.51: Data streams decoding

Unfortunately, the information related to Long Packet fields cannot be extracted when setting the decoding on three different channels at once.
In order to retrieve this information, the decoding setting has been modified and fixed to detect only the aggregated data output on the transmitting channel.

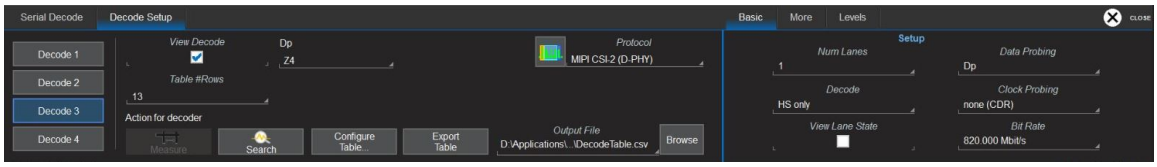


Figure 3.52: Decode setup

The decoding of the current MIPI CSI-2 Virtual Channel data aggregation is displayed below.

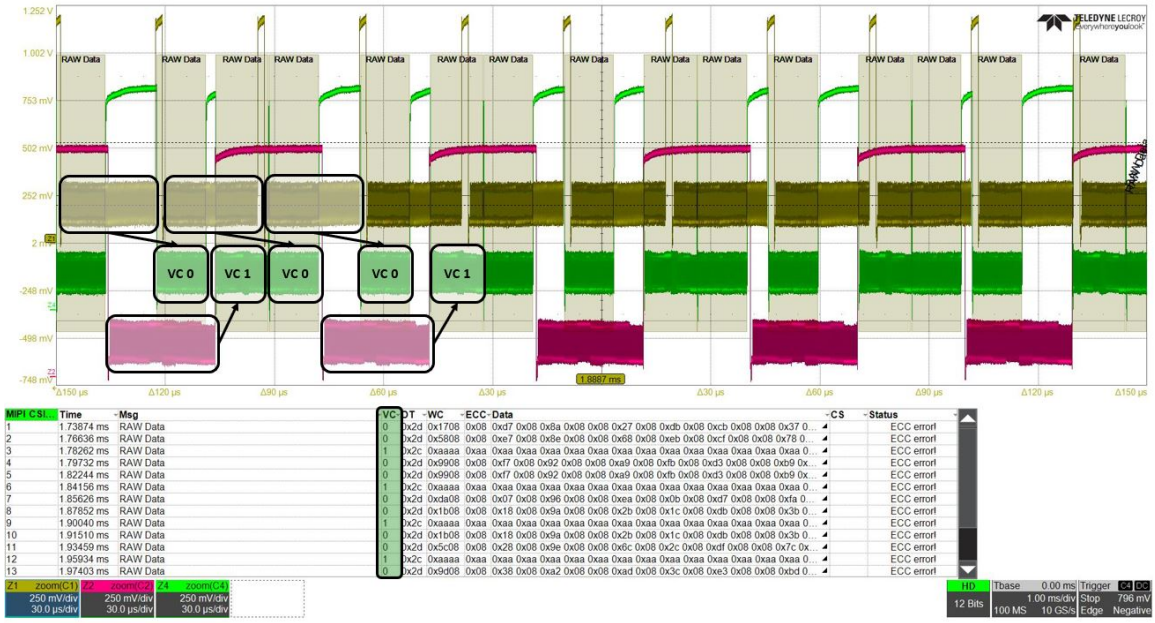


Figure 3.53: VC data aggregation decoding

As mentioned before, with the decoding carried out on the transmitting channel only (set as in figure 3.52), the value of the Virtual Channel (VC), associated with the Long Packet decoded, can be read. In this way, the SoC will be able to understand which frame the line corresponding to the Long Packet belongs to, and therefore to reconstruct the two distinct images.

MIPI CSI-2 (D-PHY) decoding also detects Data Type codes (DT) of the Payload of each Packet. Indeed, the hexadecimal code *0x2d* corresponds to RAW14 of the first receiving channel (VC 0), instead, *0x2c* to RAW12 of the second one (VC 1).

The first thing anyone notices is the *ECC error!* in *Status* column: remember that the Error Correction Code identifies errors in the Packet Header. In this specific case, the error is related to the Word Count (WC) which, instead of indicating the number of 8-bit data words in the Payload, currently assumes an incorrect value (it is already a part of the Payload) due to the decoding setting fixed on Lane 0 only for a data stream on four Lanes. Indeed, in a four-Lane configuration, the two bytes WC are distributed on Lane 1 and Lane 2.

By capturing and decoding the signals on the four Lanes of the transmitting channel,

it has been verified that the error no longer appears. In addition, the value shown in the WC column, displaying the number of bytes contained in the data Payload, is correct if compared with the frame resolution provided by module manufacturer (sensitive information).

For the sake of completeness, having already verified the correctness of Virtual Channel data aggregation, single-ended measurements have been carried out at all four Lanes of the transmitter channel *tx_ch*, which sends the aggregated MIPI CSI-2 data stream to the SoC.

At this point, a Short Packet, corresponding to a frame start, has been analyzed.

Figure 3.54: Short Packet structure

Figure 3.54 illustrates the Start of Transmission (SoT) procedure, previously described in table 3.6: the four Lanes leave the Stop state and prepare for high-speed mode. After a time $T_{HS-ZERO}$ when all positive Lines' Lanes are at '0' (HS-0), the HS Sync-sequence '00011101' can be easily recognized. At the synchronization end, the Short Packet indicating the start of frame is shown.

Within the Packet Header, the different fields of the current Short Packet can be analyzed.

- Data ID byte on Lane 0 (yellow) contains VC and DT. The former is 0, as data come from the LiDAR module (idenfied by VC 0), the latter 0x00 represent the Frame Start Code established by MIPI protocol (table 3.2)
- First byte of WC, replaced by Short Packet Data Field, on Lane 1 (magenta)

contains the first byte 0x48 indicating the frame counter

- Second byte of WC, replaced by Short Packet Data Field, on Lane 2 (blue) contains the second byte 0x22 indicating the frame counter
- ECC byte on Lane 3 (green) which does not report any errors, as confirmed by *Status* column

0x2248 Short Packet Data Field hexadecimal code indicates that frame number 8776 has been captured. Obviously, Payload and CS fields are not present as the Short Packet does not contain data Payload and Packet Footer.
Overall, the hardware tests carried out can be considered the worst case scenario, as two different devices have been employed. Indeed, the LiDAR optoelectronic module and the pattern generator manage different data types and different frame rates, they work at slightly different frequencies in a completely asynchronously (it is not possible to synchronize the beginning of the two frames). Nevertheless, the tests on the evaluation board and the MIPI CSI-2 decoding of the oscilloscope prove the correctness of the FPGA-based data aggregation circuit in accordance with the protocol.

3.4.2 Power consumption analysis

Power Calculator tool, included with Diamond software, has been used in order to estimate the power consumption of the design solution. Parameters such as voltage, temperature, resource utilization, activity and frequency have been exploited to calculate the device power dissipation.
After the design information is added, *Power Calculator* provides accurate power consumption analysis for the design.
The tool reports both dynamic and static portion of the power dissipation. The former is the power consumed by the used resources while they are switching, hence, it is directly proportional to the frequency at which the resource is running and the

number of resource units used. Instead, the latter is the power consumed by the used and unused resources.

Anyway, Diamond creates a .pcf file, added in the *File List* window, storing power analysis results exploiting information extracted from the design project.

This tool provides two modes for reporting power consumption: estimation mode, which can be used before completing the design, and calculation mode, after placement and routing.

The second alternative has been selected to calculate the power consumption in terms of actual model estimate, based on project files and device information. This preference has been set in the yellow tab in the upper right corner in figure 3.55.

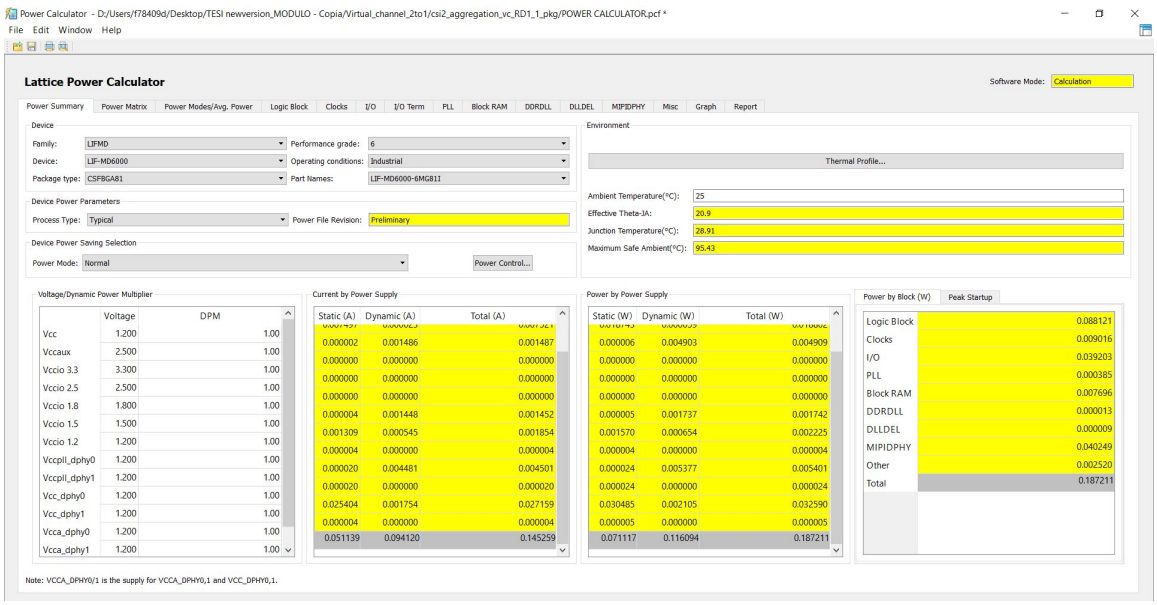


Figure 3.55: *Power Summary*

Power Summary shows the targeted device, operating conditions, voltages, and other useful information. Additional pages are available from the tabs arranged at the top of the window.

Logic Block page in figure 3.56 shows the list of resources and logic used in the design. Dynamic power calculation requires both their frequency values, manually entered according to *Clarity Designer* settings, and the activity factor percentage, set to 60%,

value that can reflect a real use case.

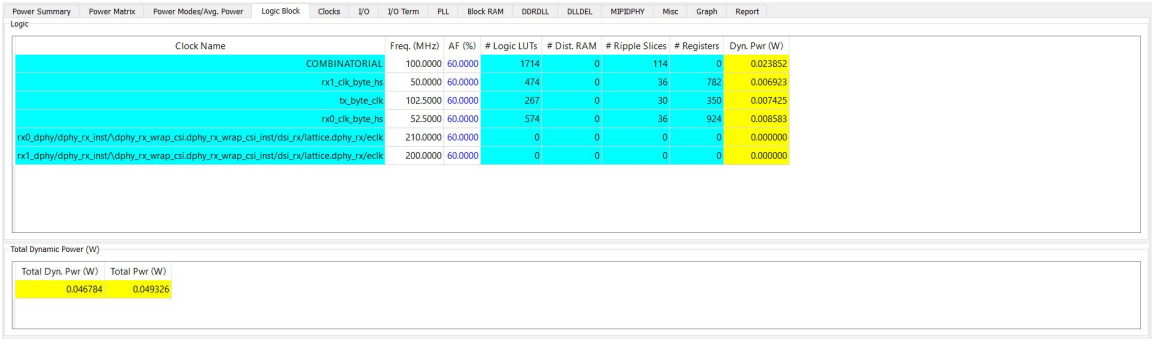


Figure 3.56: *Logic Block* tab

Through *MIPIDPHY* tab, its channel ID, transmitting mode and data rate have been set according to Logical Preference File (.lpf) and *Clarity Designer* tab respectively, as shown in figure 3.57.

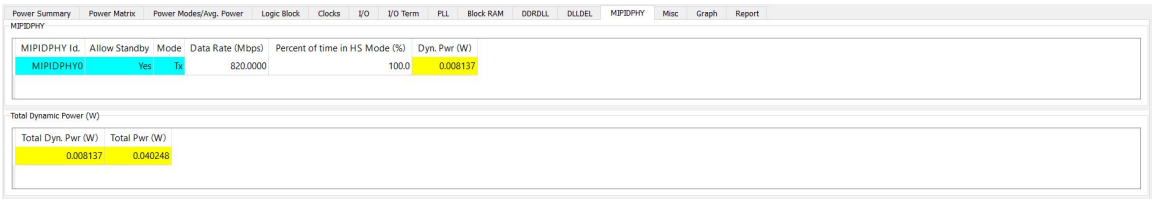


Figure 3.57: *MIPIDPHY* tab

Updated estimates of power consumption are then displayed based on these changes. Hence, after *Power Calculator* fetches project information and the parameters described above are properly set, the tool automatically calculates and displays all the power consumption details, already shown in figure 3.55.
In addition, three groups of graphs, plotted in figure 3.58, have been realized to evaluate how power consumption is affected by supply voltage, clock frequency and ambient temperature; each group displays a typical and worst (to reflect the maximum amount of current consumed by the circuit) case plot.

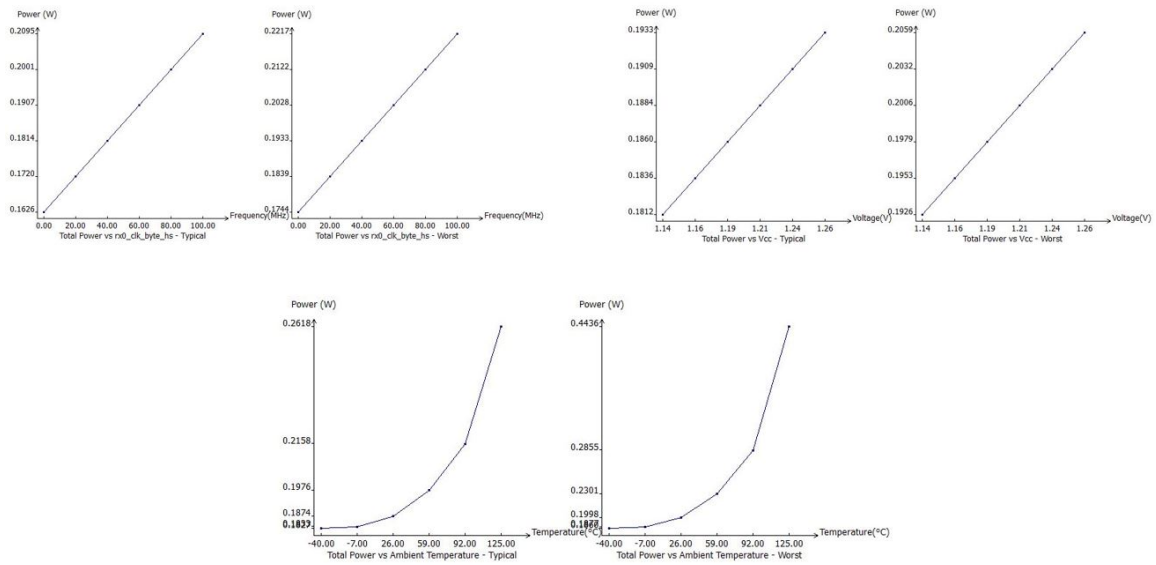


Figure 3.58: Power dissipation trends

Power consumption varies almost linearly as a function of supply voltage and frequency, but has an exponential behavior with respect to temperature.

Overall, it should be remembered that the FPGA mounted on the evaluation board is classified as an industrial and not automotive device; for this reason, the total power consumption of 187.21 mW at an ambient temperature of 25 °C represents only a preliminary estimate. In general, an operating temperature range between -40 °C and +125 °C is considered for automotive applications; however, at +125 °C the maximum junction temperature would be exceeded, thus, only +85 °C can be set as admitted upper limit. The total power dissipation values at these lower and upper limits are 182.72 mW and 210.95 mW respectively. These values are always significantly lower (about 25%) than those obtained with the SerDes solution of the first prototype.

CHAPTER 4

CONCLUSIONS

The automotive industry is witnessing a technological revolution and Advanced Driver Assistance Systems are one of the fastest growing segments in this area, having the potential to enhance the way people drive and thereby drastically reduce vehicle-related accidents. High levels of automated driving, specifically SAE Levels 3, 4 and 5, can only be achieved if a variety of different sensors are employed together in order to increasingly improve accuracy, redundancy and safety. Radars, cameras, and ultrasonic sensors have become the industry standard, but more recently Light Detection and Ranging (LiDAR) technology has been added to this list, allowing accurate real time objects detection, even at long distances and poor light or weather conditions, when cameras lose their reliability. This is the reason why LiDAR is turning essential. As a result, vehicles are becoming smarter and more automated and, as they progress along higher SAE levels of driving automation, they will be enabled by increasingly sophisticated sensor electronics and processing, brought together by high speed interconnects. In this scenario, MIPI Camera Serial Interface 2 has been widely adopted as an interface for cameras, radar and LiDAR sensors in order to meet the automotive industry's needs for high performance, speed and bandwidth, low latency and power consumption, small form factor, reduced design complexity and cost, simplified integration and accelerated time-to-market. Thus, MIPI CSI-2, with its high bandwidth

D-PHY physical layer, provides the automotive industry a standard, scalable, high speed, low power, serial interface for data transmission between a peripheral device (e.g. LiDAR sensor) and a host processor (e.g. CPU or System on Chip).

Referring to a generic LiDAR platform, which needs at least three solid state optoelectronic modules for the required field of view, and recalling that an automotive grade CPU typically has at most two CSI-2 ports, the large demand to manage more data streams than the SoC ports is becoming absolutely necessary. Starting from these premises, an hardware solution has been evaluated to meet this need and aggregate two distinct MIPI CSI-2 raw data flows from two optoelectronic modules, providing a single CSI-2 output to the automotive System on Chip.

This goal can be achieved with many different technical solutions, previously described and compared in table 3.8. In this regard, although the first prototype of the project developed by Marelli Automotive Lighting was implemented with Serializer/Deserializer technology, theoretically, the most effective solution to investigate, implement and test would be the one based on FPGA, in order to combine optimal performance and flexibility.

A careful feasibility study resulted in the choice of an FPGA of the CrossLink family offered by Lattice Semiconductor, which already made available MIPI D-PHY receiver and transmitter blocks, while the software environment Lattice Diamond provided a complete set of tools for design entry, implementation, synthesis, analysis and programming activities.

At the end of the implementation, the project tested on the evaluation board provided positive results. The input MIPI CSI-2 raw data collected from the two modules have been correctly merged, according to Virtual Channel value, into a single MIPI CSI-2 output data stream sent to the Automotive System on Chip, as demonstrated by the waveforms captured by the oscilloscope (figure 3.53), equipped with the embedded MIPI protocol analyzer. In addition, an accurate power consumption analysis based on the actual design showed a significant reduction in power dissipation of about 25%

of the solution based on SerDes technology of the first prototype. More particularly, in order to aggregate two CSI-2 data streams, two Serializers (one for each LiDAR module) and one Deserializer are necessary; in case of FPGA-based solution, a single component is able to complete the data aggregation operation, meaning less area occupation on the PCB. Even an ASIC can work at high frequency while its power consumption can be meticulously controlled (since the circuit is optimized for its specific function), but its lack of flexibility and its slow time-to-market do not make it a suitable component for prototyping or for applications that require frequent updates. Overall, this FPGA-based implementation proves to be extremely suitable and convenient for the project purposes. Indeed, comparing this technique with other hardware solutions previously investigated, many advantages in terms of flexibility, performance, power consumption, time-to-market, costs and number of components involved make it an optimal choice for any possible future development and mass production.

Bibliography

[1] O. Gietelink, J. Ploeg, B. De Schutter, M. Verhaegen (July 2006), *Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations*, Vehicle System Dynamics.

[2] National Highway Traffic Safety Administration (December 2020), *Overview of Motor Vehicle Crashes in 2019*.

[3] Texas Instruments (2017), *Making cars safer through technology innovation*.

[4] `https://injuryfacts.nsc.org/motor-vehicle/occupant-protection/advanced-driver-assistance-systems/data-details/`

[5] M. Phadke (January 2019), *Improving Driver Safety using ADAS – Components and Features*.
URL: `https://www.einfochips.com/`

[6] C. Godwin, BBC News, San Francisco (23 April 2021), *Making cars safer through technology innovation*.

[7] J. Shuttleworth, SAE International (2019), *SAE Standards News: J3016 automated-driving graphic update*.

[8] O. Bay, ABIresearch, New York (18 April 2018), *ABI Research Forecasts 8 Million Vehicles to Ship with SAE Level 3, 4 and 5 Autonomous Technology in 2025*.

[9] XenomatiX (12 April 2018), *SYSTEM AND METHOD FOR DETERMINING A DISTANCE TO AN OBJECT*.

International Pubblication Number: WO 2018/065429 AI
International Application Number: PTC/EP2017/075096

[10] S. Royo, M. Ballesta-Garcia, Universitat Politècnica de Catalunya (30 September 2019), *An Overview of Lidar Imaging Systems for Autonomous Vehicles.*

[11] Texas Instruments (May 2020), *An Introduction to Automotive LIDAR.*

[12] TriLumina Corporation (December 2018), *315 SMT VCSEL Illuminator Module Datasheet.*

[13] MIPI Alliance (October 2019), *Driving the Wires of Automotive*, MIPI specifications in automotive and the A-PHY solution.

[14] MIPI Alliance (April 2019), *MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2).*

[15] MIPI Alliance (August 2014), *Specification for D-PHY.*

[16] Lattice Semiconductor (December 2020), *CrossLink Automotive Family Data Sheet*, FPGA-DS-02013-1.7.

[17] Lattice Semiconductor (March 2021), *CrossLink Programming and Configuration Usage Guide*, FPGA-TN-02014-1.4.

[18] Lattice Semiconductor (January 2020), *CrossLink LIF-MD6000 Master Link Board - Revision D*, FPGA-EB-02027 Version 1.0.

[19] Lattice Semiconductor (September 2019), *MIPI CSI-2 Virtual Channel Aggregation*, FPGA-RD-02051-1.2.

[20] Lattice Semiconductor (October 2020), *Lattice Diamond 3.12 User Guide.*