JLR's Chiplet Challenge

For Inter-IIT Techmeet 12.0

Team 15

December, 2023

Abstract

This report serves as a continuation of the Mid-Evaluation Report submitted for the Inter-IIT Tech Meet 12.0 Problem Statement, JLR's Chiplet Challenge. It attempts to answer Parts 1.2 and 1.3 along with 2.2 and 2.3 of the Problem Statement. Part 1 of the PS requires the depiction of our chosen automotive applications through a detailed micro-architecture diagram of the chiplet-based processor, mentioning all used interconnects. It also requires a discussion on which parts of the package could be procured off-the-shelf and which should be designed by JLR.

Part 2 of the PS can broadly be divided into three sections. Part 2.2.a requires a detailed discussion on interconnect technology and aspects like their latency and communication efficiency. Part 2.2.b delves into ensuring overall secure and reliable communication. Part 2.3 finally requires a comprehensive analysis of various thermal management approaches to devise a cooling solution that will efficiently suck heat out of the package. The following is an extensive compilation of our research, analysis, and proposed solution.

Contents

1. Building up to the Micro-architecture, Choosing an In-Vehicle Network Architecture

The Mid-Evaluation Report attempted to answer Parts 1.1 and 2.1 of the Problem Statement. For Part 1.1, we outlined key domains present in an automobile that govern its safety, comfort, and efficient operation and then narrowed down the scope to domains where the use of chiplets would add the most value, the chosen domains being the **Advanced Driver Assistance System (ADAS) and Infotainment System.**

Approaching the task of drawing the microarchitecture diagrams for our chiplet-based processors first requires a clear understanding of the **computational requirements** of the **specific functions** that the application performs. Since the chiplet-based processor would be driving the application specific Electronic Control Unit (ECU), it would also be necessary to have an understanding of the kind of **sensor inputs** that would be entering the ECU module. This would consequently require knowledge about the **communication protocols** used by various sensors and the exact **path** traversed by their raw data through the in-vehicle network to reach the relevant ECU.

The optimum choice for the in-vehicle network architecture (network of sensors/ECUs/actua- tors) and the connecting protocols to be used heavily depends upon the application requirements in terms of latency, data bandwidth, and safety criticality.

1.1. Domain v/s Zonal Architectures

Most vehicles today are built using what is called a **domain-based ECU architecture**. This categorizes ECUs into domains based on specific functions, for example, the powertrain, regardless of their physical location in the vehicle. Each domain-specific network of ECUs has a **domain controller module** that it communicates with. The controller module is also used for communication between ECUs of the same network along with any domain-based processing that is required. It is also responsible for communicating with other domain networks governed by independent domain controllers.

However, with the constantly growing number of features, each of these domain-specific networks of ECUs is now large enough to stretch across the

entire vehicle, overlapping with each other and **complicating the vehicle harnessing**.

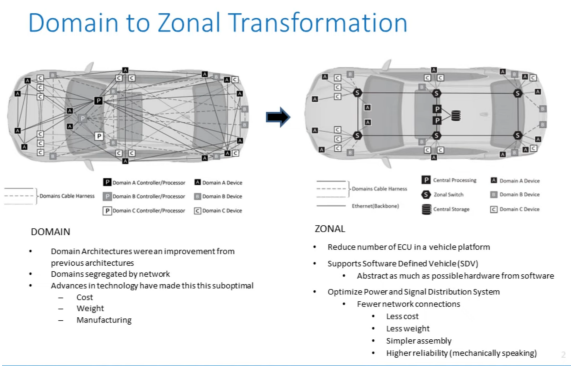


Figure 1: Domain v/s Zonal Architecture [63]

A **zonal architecture**, in contrast to a domain architecture, groups many - if not all - domain functions based on their **geographical location**, or zone inside the car. This also includes zonal power distribution and load control. All zonally grouped ECUs are connected to a ***zonal module*** which would then further directly communicate with the ***central vehicle compute*** which will handle a major portion of all zonal data routing, compute and processing.

A zonal module thus behaves as a network data bridge between the vehicle's computing system and local edge nodes like smart sensors and ECUs. To reduce cabling in the vehicle, a zonal module will also distribute power to different edge nodes (by implementing semiconductor smart-fuse capabilities), handle low-level computing, and drive local loads like motors and lighting.

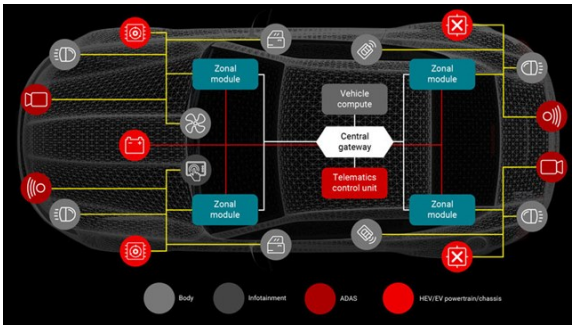


Figure 2: Zonal Architecture (power distribution) [2]

The architecture significantly reduces the vehicle's harnessing complexity, with fewer wires, connectors, and ECUs, thus reducing the overall cost and weight of the vehicle.

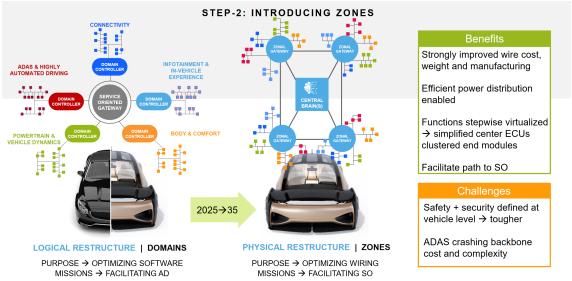


Figure 3: Domain to Zonal

Adopting a zonal architecture also aligns with or directly corresponds to trends being currently observed in the automotive industry like:

- Centralized Vehicle Architectures

Implementing a zonal architecture requires moving functionality away from individual ECUs in the domain architecture towards a **central** "brain-like" entity acting as a powerful compute unit, that takes care of all required data processing from individual zones.

- Software-Defined Vehicles

By abstracting hardware formerly present in individual ECUs, in the form of software, almost all major functions now become software-defined, giving OEMs much more control to add more functionality or value to the vehicle [63], from high-level software maintenance with over-the-air (OTA) updates; to firmware-over-the-air (FOTA) updates and always-on cloud connections to enable new functions and to improve features such as autonomous driving [2]. It also opens up new opportunities for **service-oriented** business models, for example: offering post-sale feature upgrades like improved battery range or new driving assist features or even an option for remote diagnostics or predictive maintenance alerts [8].

Instead of adding new functions through newer ECUs, those functions can now directly be downloaded from the cloud inside a flexible HW with over-the-air (OTA) updates. This allows for continuous integration and continuous deployment (CI/CD). [8]

1.2. Gateways, Switches and the use of Automotive Ethernet

Referring to **Figure 3** (Domain v/s Zonal Architecture), the domain-based architecture has separate networks for Infotainment, Powertrain, Body, and Climate, each network utilizing its own communication protocol optimum for its required data bandwidth. A **central gateway** is tasked with bridging data between all the different networks to facilitate communication between ECUs of two

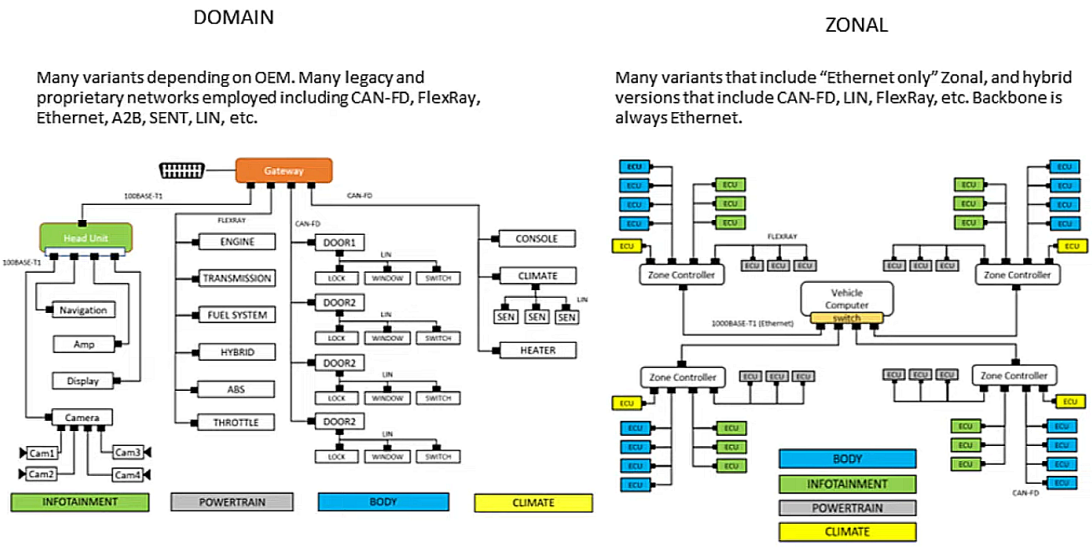


Figure 4: Domain v/s Zonal Architecture [63]

different domains. Gateways are built on sophisticated and costly software requiring high interoperability between a diverse range of legacy communication protocols used in automotive networks like CAN-FD, LIN, FlexRay, Ethernet, A2B, etc. for data type inter-conversion. This process also introduces undesirable latency in communication which could hamper the proper functioning of latency-critical applications [63].

With the use of gateways being undesirable, zonal architectures and the automotive industry as a whole are gradually moving towards the utilization of standardized methods of addressing and routing data traffic, namely **Automotive Ethernet**.
The **Ethernet Protocol Standard** is extremely adept at efficient data traffic management, capable of utilizing hardware-based devices like switches to drastically reduce the dependence on software for routing and translation of messages.

Automotive Ethernet provides a scalable solution to high-speed, low-latency communication requirements, capable of handling the escalating data bandwidth requirements of modern in-vehicle systems. This is also why all **backbone communication**, i.e. communication between all zonal modules and the central compute utilize Automotive Ethernet for fast and reliable communication. For context into the usefulness of Automotive Ethernet, the newly defined Single Pair Ethernet (SPE) supports speeds from 10 Mbps to 10 Gbps, defined through IEEE 802.3cg (10 Mbps), IEEE 802.3bw (100 Mbps), IEEE 802.bu (1 Gbps) and IEEE 802.3ch (10 Gbps). All of these new Ethernet technologies work over a single-pair cable and can communicate at distances as far as 15 meters, which is long enough to cover the longest link in a vehicle. Ethernet can also enable the time synchronization of sensor data using IEEE 802.1AS timestamping to achieve low latency [20].

While Ethernet is capable of extremely high-bandwidth, these speeds are not necessary in all scenarios. For example, communicating with the door control module or the heating, ventilation, and air-conditioning system does not require a 100-Mbps data rate. A 10-Mbps Ethernet PHY or alternative network protocol like Controller Area Network (CAN) is better for lower-speed and less bandwidth-intensive use cases, while it is better to reserve higher speeds for sending aggregated camera and autonomous-driving sensor data from the zonal modules to the central computing system [20].

1.3. Communication in the Zonal Network

When it comes to zonal module design, they can have a combination of three different functionalities [10]:

- Zonal I/O Aggregator:
  A simple gateway or switch-based zonal module, strictly in place for aggregation of sensor/ECU I/O. It has no compute capabilities.
- Zonal Controller:
  Capable of acting as an aggregator along with basic computing capabilities on par with that of a Micro Controller Unit. It satisfies minor edge-processing requirements for performing simple threshold comparisons.

- Zonal Processor:
  Capable of acting as an aggregator and equipped with a high level of compute capability for running involved programs or algorithms.

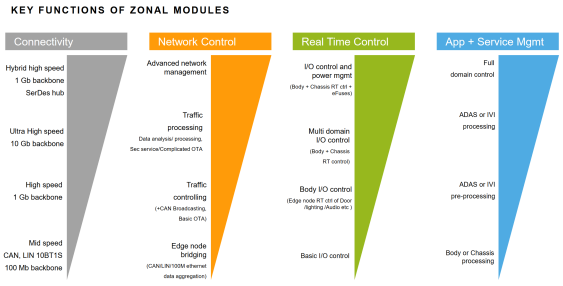


Figure 5: Zonal Modules - Key Functionality [10]

In the proposed zonal architecture, all backbone communication is ensured through high-bandwidth Automotive Ethernet, the central compute utilizes a purely switch-based routing, all edge-nodes use bandwidth requirement-specific proprietary or legacy communication protocols like CAN-FD, Ethernet, LIN, FlexRay, A2B etc.

Edge-To-Edge Commnuication:
(between 2 CAN-based ECUs)
First requires translation of CAN signal from the transmitting ECU to Ethernet inside the Zonal Gateway. Then transmission through the central compute to the receiving zonal module. Finally translation of Ethernet to CAN in the receiving Zonal Gateway and then transmission to the target ECU.

Edge-To-Center Commnuication:
(from a CAN-based ECU to the central compute)
First requires translation of CAN signal from the transmitting ECU to Ethernet inside the Zonal Gateway. Then transmission to the central compute.

Within-Zone Commnuication:
(between 2 CAN-based ECUs)
Zone controller acts as CAN gateway.

Refer to the given figure

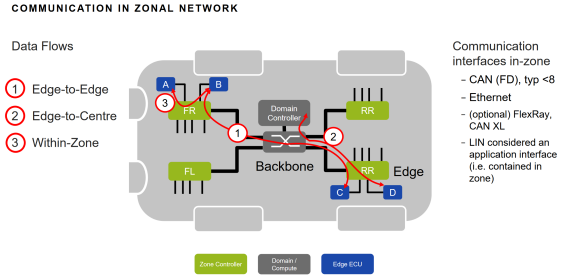


Figure 6: Zonal Network Communication [10]

All translation of legacy protocols to Ethernet requires the **IEEE 1722 Standard** which is a part of the **IEEE TSN (Time Sensitive Network) Standards**. The IEEE TSN Standards ensure reliable communication and include standards for Transport (1722), Time Synchronisation (gPTP or Generic Precision Time Protocol), Resource Management, Deterministic Latency (Enhancements for scheduled traffic), Redundancy (Frame Replication), and Security [10].

1.4. Transitioning To Zonal Architectures

It is extremely difficult to migrate from a domain architecture to a fully zonal architecture in just a single step for any OEM. It is more common for an OEM to go through an intermediate step of a hybrid architecture, between domain and zonal, a partial zonal architecture, with a few select domains separate from the main zonal architecture given their high bandwidth requirements or safety-criticality like the ADAS.

The final fully zonal architecture aims to be completely Ethernet-based, where all sensor/ECU inputs are compatible with the Ethernet protocol for easy packaging and transmission. The zonal controller modules would be switch-based instead of gateway-based, drastically reducing communication latency and dependence on software for interconversion between protocols [63].

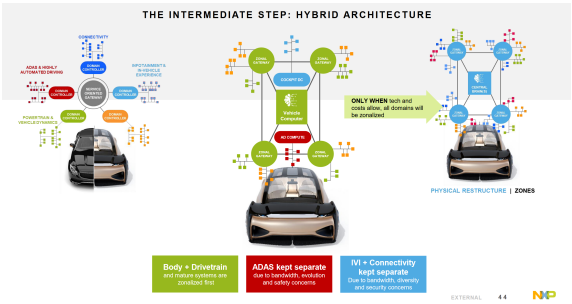


Figure 7: Partial-Zonal (Hybrid)

1.5. *Proposed Architectures for the ADAS and Infotainment System*

The ADAS is a safety-critical application that requires high bandwidth and low-latency communication to process real-time data from multiple Cameras, LIDAR, and RADAR sensor modules to make split-second decisions. With these ADAS sensor modules not having reached full Ethernet standards and the current level of zonal architecture utilizing zonal controller modules that still require the use of gateways to handle legacy communication protocols, it is difficult to ensure seamless and low-latency data transfer through the whole network.

As the level of ADAS improves and we approach full vehicle autonomy, it becomes increasingly essential to deliver raw sensor data with no compression to our processing package to ensure that feature extraction occurs with the highest accuracy possible [20]. To be able to deliver raw data at high speeds, we have chosen to directly connect our ADAS sensor modules with the ADAS ECU that houses our chiplet-based processor package via either high-bandwidth Ethernet or GMSL SerDes protocols ensuring low interference high-speed transmission.

Since applications other than the ADAS like the Infotainment System do not necessitate super low-latency transmission and communication, minor latency overheads observed due to communication through gateway-based zonal controller modules will not affect the overall functioning of the system or lead to a hazardous situation for the occupants of the vehicle.

2. Obstacle Detection

Obstacle detection is the process of finding and detecting barriers in the path of a moving object. We have chosen to focus on this application in the ADAS system as it is fundamental to safety systems in automotive technology. It also forms the backbone of **collision avoidance** and **adaptive cruise control systems**. Obstacle detection systems must distinguish between obstacles and non-obstacles, as well as identify the type of obstacle. It is also an important part of many applications, such as surveillance, self-driving cars, or robotics. Current automobiles use various sensor technologies like radar, lidar, cameras, and ultrasonic sensors are used for implementing obstacle detection algorithms. The detection of obstacles is done with the proposed algorithm called **"YOLO"**(You only look once). This non-maximum suppression algorithm plays an essential role in object detection and tracking. The process of object detection can be framed as a regression problem instead of a classification task by spatially separating bounding boxes and associating probabilities to each of the detected images using a single convolutional neural network (CNN). Before we delve deeper into the functional flow diagram of our implementation of obstacle detection let us have a look at the few common building blocks in neural networks.

2.1. *CNN*

Deep Convolutional Neural Network (CNN) based models are the frequently used type of models that used to fulfill the task of Object detection. They work on powerful **GPU's** that consume a lot of **power**. We need CNN models with low complexity which can run on embedded processors for practical distribution of object detectors on mobile devices. As an important feed-forward neural network in the field of deep learning, convolutional neural network (CNN) has been widely used in image classification, face recognition, natural language processing and document analysis in recent years.

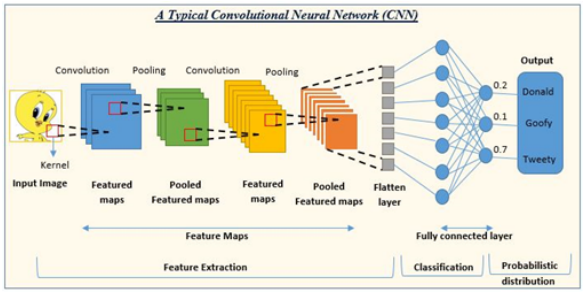


Figure 8: Generic CNN Architecture [52]

2.1.1. *The Convolution operation*

Mathematically, convolution is the summation of the element-wise product of 2 matrices. The image obtained from the Image signal processing block (ISP) is a 3-D matrix with the first two dimensions of the 3-D matrix representing the spatial dimensions of the image, typically denoted as height (H) and width (W), These dimensions define the size of the image in terms of pixels. The values in the third dimension typically represent the intensity of the RGB colour channels.

Each layer of the image is convolved with a 2-D **Kernel** which essentially leads to 3-D convolution between the image and the feature filter. In image processing, a **kernel**, **convolution matrix** or a **feature filter** is essentially a small 2-D matrix of weights. It is used for blurring, sharpening, embossing, edge detection and more.

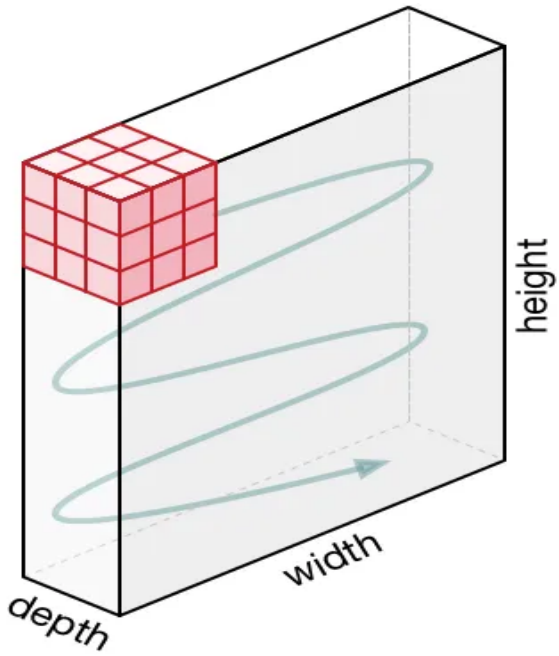


Figure 9: 3-D Convolution [64]

This feature filter then traverses over the entire image performing repeated convolutions until the entire image is covered. The results of these **convoloution operations** give us an idea of how well our **feature matches** a particular section in the image.

2.1.2. ***Feature Maps***

A feature map is the output of a convolutional layer representing specific features in the input image or feature map. During the forward pass of a CNN, the input image is convolved with one or more filters to produce multiple feature maps. The movement of the filter across the image is characterised by a parameter called **Stride**, which refers to the number of pixels by which the filter moves across the input image in each step. It essentially determines how much the filter jumps during the convolution operation. Each **feature map** corresponds to a specific filter and represents the **response** of that filter to the input image. Each element in the feature map represents the activation of a specific neuron in the network, and its value represents the degree to which the corresponding feature is present in the input image. We **"normalise"** the weights in the kernel before feature extraction. Normalization is defined as the division of each element in the kernel by the sum of all kernel elements, such that the sum of the elements of a normalized kernel is unity. This is done to ensure the information contained in the image is not lost upon convolution.

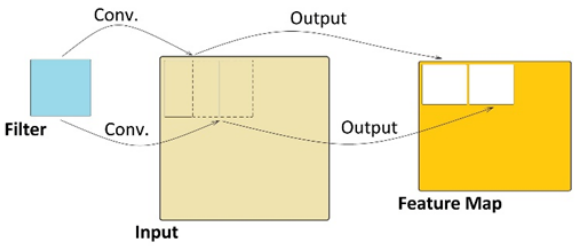


Figure 10: Feature Map Extraction [70]

2.1.3. ***Pooling of Feature Maps***

Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters we have to monitor and the amount of computation performed as we progress along the neural network. The pooling layer summarises the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarised features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

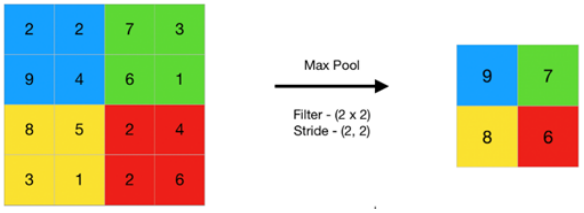


Figure 11: Max Pooling [13]

2.1.4. ***Flatten Operation***

The intuition behind the flattening layer is to convert data after pooling into 1-dimensional array for feeding the next layer. The output of this convolutional layer is flattened into a single long feature vector which is connected to the final classification model, called fully connected layer. In CNN models there are many more than three convolutional kernels,i.e. 16 kernels or even 64 kernels in a convolutional layer. The flattening layer is a crucial component of convolutional neural networks as they connect CNN's to ANN's, allowing the neural network models to learn complex patterns and make predictions. While the flatten layer performs a simple operation of converting multi-dimensional arrays to a one-dimensional array, it is a fundamental tool in image-processing neural

network models.

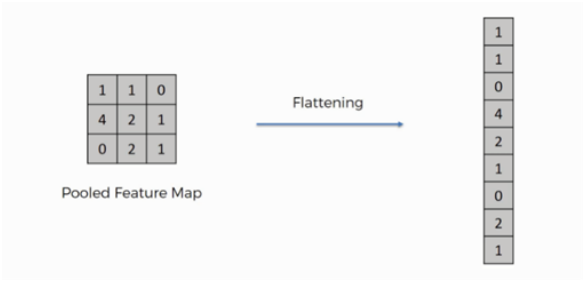


Figure 12: Flattening of Convolved Layer [14]

2.1.5. Feature classification

After passing through several layers of a convolutional neural network finally the 1-dimensional array obtained after flattening is given to the Artificial Neural Networks (ANN), which are multilayer fully-connected neural networks as shown in figure 1. The **fundamental** unit of information processing in an ANN is called a **neuron**. Every neuron in a particular layer of a fully connected ANN is connected with every other neuron in the next layer through a set of weights, there may also exist hidden layers between the intial and final layer. The 1-dimensional **feature vector** after flattening is passed on the first layer of the ANN, following this every neuron gets kind of a **vote** in deciding what kind of feature is contained in the image, the strength of these votes is determined by the **weights** in neural network. Finally, after all the votes are polled together, the last layer of the ANN gives us a probabilistic distribution of our image belonging to one of the many possible classes.

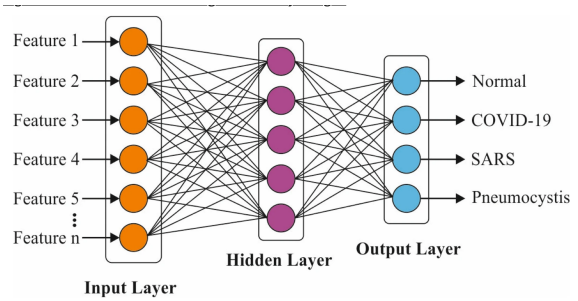


Figure 13: An Example of an ANN classification [1]

2.2. The YOLO Algorithm [9]

You only look once **(YOLO)** is a state-of-the-art, real-time object detection algorithm that employs CNN as its backbone. It was introduced in 2015 by **Joseph Redmon**, **Santosh Divvala**, **Ross Girshick**, and **Ali Farhadi** in their famous research paper titled "You Only Look Once: Unified, Real-Time Object Detection". YOLO is a **one-stage** object detection algorithm, meaning it performs object detection and classification in a single pass through the entire neural network. The algorithm shows the **class** of the objects present in the image and **bounding boxes** will be used to show where they are in the pictures.

Why YOLO?? YOLO offers numerous advantages over contemporary object detection algorithms like R-CNN, SSD retina Net etc, it's trademark features being:

1. **Speed:** YOLO is known for its high speed, outperforming many other algorithms in real-time performance, especially in video processing applications which are predominant in automobiles. It's generally faster than R-CNN and RetinaNet but may be slightly slower than SSD.
2. **Simple Architecture:** YOLO offers a unified and simpler architecture compared to multi-stage detectors like R-CNN, thus making it more straightforward to implement and understand.
3. **Accuracy:** While YOLO is fast, it might sacrifice a bit of accuracy compared to two-stage detectors like Faster R-CNN due to its single-shot nature. However, YOLO strikes a good balance between speed and accuracy.

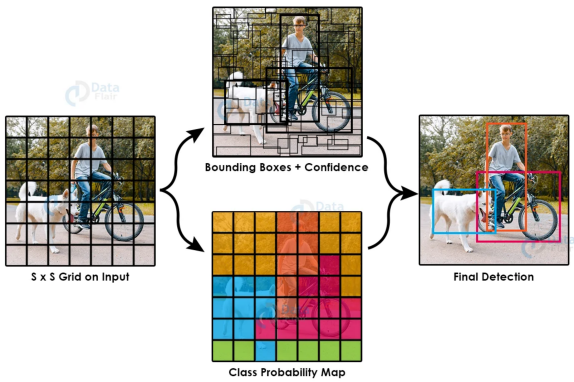


Figure 14: YOLO detecting multiple classes [72]

On January 10th, 2023, the latest version of YOLO which is YOLO8 launched claiming advancements in structure and architectural changes with better results.

2.3. Optimizing Obstacle Detection architecture with Hardware Accelerators [73]

2.3.1. Introduction

Later versions of the YOLO algorithm, especially starting from **YOLO V4** and beyond, tend

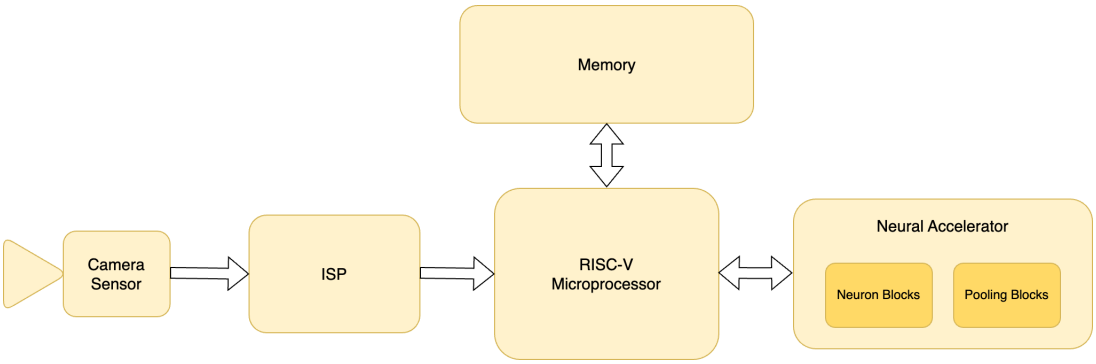


Figure 15: RISC V hardware accelerator designed for the YOLO V2 obstacle detection system

to require a substantial amount of computation and are also architecturally complex compared to earlier versions. We have chosen to implement a particular version of YOLO, specifically the **YOLO V2-tiny**(algorithm) through our functional block diagram as shown in figure(insert fig no).
We will also explore the necessity of hardware accelerators and the benefits of using chiplets in our obstacle detection system. We shall start our discussion from our camera sensor and continue with the flow of our functional block diagram.

2.3.2. Decoding the CMOS Image Sensor

The principal sensor feeding information to the ECU in the case of obstacle detection is the camera module which consists of numerous image sensors. An image sensor is a device that converts an **optical image** into **electronic** signal. These days CMOS image sensors are used mostly in digital cameras, smart phones, camera modules and imaging devices. It contains a **bayer** colour **filter mosaic array** and underlying active pixel sensor **photodiodes**.It also contains an ADC, analog signal processing units, clock and timing control. All these ensure that the information contained in the image in the form of pixels is seamlessly converted to digital data for further processing.

A CMOS image sensor grid consists of multiple image sensors arranged together, this sensor alignment ensures that each pixel sensor captures light of a certain colour only, either Red, Blue or green. The **Bayer pattern** refers to the arrangement of colour filters on the sensor's pixels. The Bayer filter is a **Mosaic** pattern composed of red, green and blue colour filters that cover individual pixels in a specific pattern. The Bayer filter ensures that the image from the sensor is **essentially 2-D** i.e each pixel contains only a single colour(either R,G or B). This process is called **Mosaicing** and the image obtained from the sensor is called a **"Mosaiced Image"** or a **"bayer image"**. The bayer image is essentially application of a bayer filter to an R-G-B image. There are different kinds of bayer patterns like BGGR, RGBG, GRBG,or RGGB each depending on the alignment of pixels on the CMOS Sensor grid.

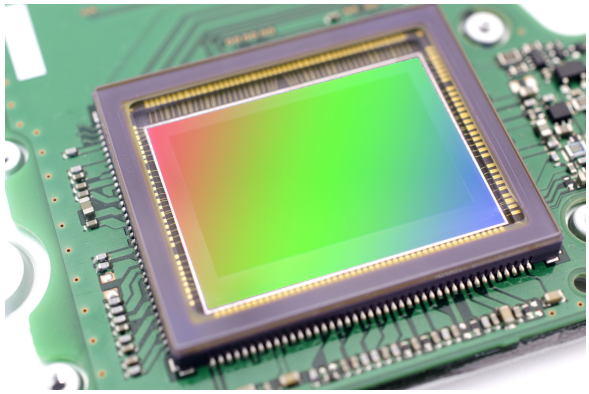


Figure 16: CMOS image sensor grid [45]

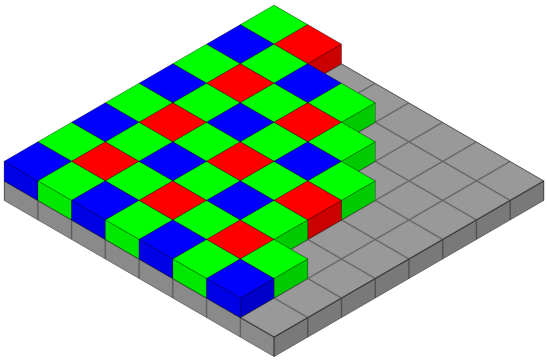


Figure 17: Bayer Mosaic Pattern [7]

2.3.3. Image signal processing(ISP)

The bayer image is transferred to the ISP module using the MI-PI CSI2 communication protocol which is a widely adopted, high-speed communication protocol for transmission of still and

video images from image sensors to application processors. The ISP module performs the crucial task of **Demosaicing**, which is **reconstructing** the **missing** colour **information** from the neighbouring pixels and creating a complete, full-colour image. We can think of it like filling in the blanks in a colouring book. The ISP module analyzes the patterns and relationships between the coloured pixels, intelligently estimating the missing pixel values. The most commonly used demosaicing method is **Bilinear interpolation** which computes missing color values by taking weighted averages of neighboring pixel values. This demosaicing method is simple and computationally efficient but can result in reduced image quality, especially in portions of the image with high contrast. There are other methods like **bicubic interpolation** or **edge based filtering** however in our implementation of obstacle detection we stick to using bilinear interpolation.

The ISP block also handles other functions like

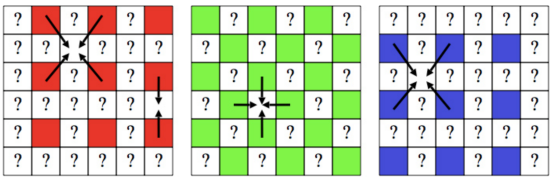


Figure 18: Bilinear Demosaicing [17]

Denoising: Certain Standard filtering algorithms used on images like Gaussian filtering and median filtering are applied and other custom filter algorithms can be designed as per need. Our implementation uses the built-in filters on MATLAB Simulink.

Advanced Features: Some ISPs offer additional features like white balance correction, correction for lens imperfections like lens shading, distortion etc.. and HDR processing.

The Main Compute : We have chosen our MPU to be based on **RISC V architecture**, primarily because it is known for it's good performance, **flexibility** and ease of customisation **Open-source.** The architecture of various training models defines the order in which the different operations like convolution and pooling take place across the various layers of a neural network. On average the Darknet 19 model contains around 1000 feature filters YOLO v2 is traditionally trained on different architectures such as VGG-16 and GoogleNet. However, we have proposed an architecture based on Darknet-19 model.

The reason for choosing the Darknet architecture is its lower processing requirement in comparison to other architectures. For an Image size 224x224, the

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Figure 19: Convolutional layers in the darknet19 model [49]

darknet19 model requires a compute of only 5.58 FLOPS vs the 30.69 FLOPS required for VGG-16 and the 8.52 FLOPS required for GoogleNet.

Neural accelerator : We have implemented a simplified Neural accelerator block on hardware in Verilog, this hardware consists of a collection of 4 Neurons and 4 pooling blocks. We have already seen what a pooling block in neural networks, now let's look at what exactly is the function of neurons here

The Neurons act as tiny decision-makers, processing information and passing it on. ReLU (Rectified Linear Unit) activation functions play a crucial role in this process, acting as gatekeepers that determine whether a neuron "triggers" and transmits information further.

Each neuron receives a weighted sum of its inputs from all the other neurons in the previous layer. This sum acts as an input signal to the ReLU function. The ReLU function as a threshold gate. It has a "cutoff" point, typically set at zero. It's behaviour is similar to a ramp function for positive inputs.

If the input signal is greater than zero, the ReLU function "activates" the neuron, passing the signal forward with no change. The neuron becomes transparent. Conversely, if the input signal is less than or equal to zero, the ReLU function "deactivates" the neuron, stopping the signal in its

tracks. The gate remains closed, blocking information flow.

Decision Making Flow: The compute which is our RISC V based processor calls the convolution function,Object detection is a very practical application and YOLO has too big network to be implemented on embedded CPUs, and that's why we want to design an accelerator for it. Whenever convolution is called the control of data flow shifts to the hardware accelerator, there the convolution of the image is sped up as it is purely a hardware accelerator thereby making huge savings on latency. After convolution the results are stored in the memory module which is updated after each convolution, similarly the status of memory is also updated after each pooling operation.
This is where chiplets come into play, Chiplets offer **modularity** and **upgradability,** Now everytime the version of YOLO changes we would only need to change the values of **weights** and **Bias** stored in he memory,so OEM's would no longer need to **change** their **hardware** everytime an update pops up. And even if the darkNet model changes the manufactures would just have to replace the **accelerator chiplet** keeping the rest of the package intact, This is of course a **huge cost benefit** from the manufacturer's perspective. Additionally we can also use **Shakti**,India's **indigenous processor**. The H-class Shakti processor is a **64-bit processor** aimed at highly parallel enterprise, HPC and analytics applications. The cores can be a combination of C or I class, single-thread performance driving the core choice. The H-class has up to **128 cores** with multiple accelerators per core, these many cores can surely **help** in parallel processing for **image processing applications**.

2.3.4. Matlab Simulation Results

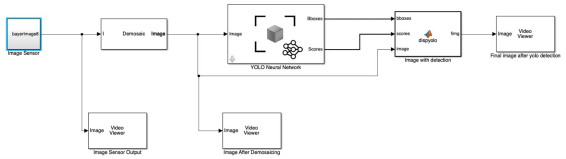


Figure 20: Simulink Flow Diagram

We set up a simulation environment in MATLAB Simulink where we simulated the entire image processing pipeline mosaiced camera sensor image as the input which is demosaiced and processed through the YOLO Neural Network, giving out images with bounding boxes and labels around the detected objects.

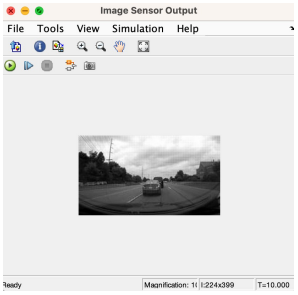


Figure 21: Image Sensor Output

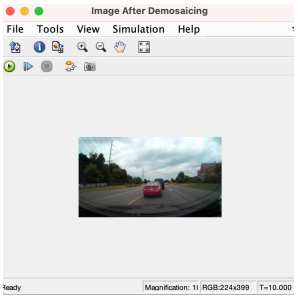


Figure 22: Demosaiced Image

3. Throughput Analysis

In the proposed architectures, throughput optimization can be done by taking care of two parameters, node size and topology of chiplets. Node size refers to the nm technology used for manufacturing chiplets. Topology is the physical placement of chiplets on the interposer with optimum interconnections between all combinations of communicating pairs, for example, compute-to-compute, compute-to-memory, compute-to-I/O, etc.

3.1. Node size

Decrease in nm technology has scope for decrease in critical path delay of the circuit. If an SoC has 3 modules, manufactured with a particular technology, there is a high increase in cost or decrease in nm scale for the whole SoC. But in chiplet based design, scaling down the transistor size of chiplet having highest latency will give considerable increase in throughput but lesser increase in cost.

We tried to test this thought on our simulation's RISC module. We changed the nm scale from 90 to 65 nm using Synopsys' Design Compiler. According to the timing report generated, you can see that critical path delay decreases from 43.46 ns to 9.16 ns (62.5ns is the clock period).

Chiplets offer a significant advantage by allowing for targeted scaling down of transistor size, resulting in a substantial decrease in critical path delay and increased throughput, all while minimizing the associated cost impact.

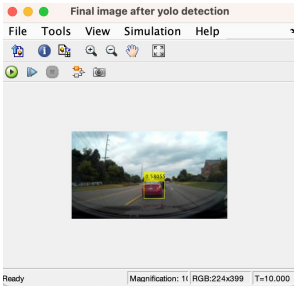


Figure 23: Output image after Object Detection

```
                                                          0.00       9.16 r
data arrival time                                                     9.16

clock CLOCK (rise edge)                                  62.50      62.50
clock network delay (ideal)                               0.00      62.50
clock uncertainty                                         0.00      62.50
msrv32_csr_file_0/MC/minstret_out_reg[63]/CK (DFQM2RA)
                                                          0.00      62.50 r
library setup time                                       -0.01      62.49
data required time                                                  62.49
--------------------------------------------------------------------------
data required time                                                  62.49
data arrival time                                                   -9.16
--------------------------------------------------------------------------
slack (MET)                                                         53.33
```

Figure 24: Synopsis output for 65nm process

3.2. ***Topology***

Since the D2D links limit the Inter-Chiplet Interconnect (ICI) data width, we want to operate them at the highest frequency possible to maximize their throughput. To run such links at high frequencies without introducing unacceptable bit error rates, we must limit their length to a minimum. The length of D2D links is minimized if we only connect adjacent chiplets. However, with such restricted connections, the shape and arrangement of chiplets has a significant impact on the performance of the ICI.

In a regular mesh arrangement, each non-border chiplet is connected to 4 other chiplets. In HexaMesh[24], the arrangement is optimized by arranging chiplets in a circle around the central chiplet in a honeycomb pattern where each non-border chiplet is connected to 6 other chiplets. This topology reduces the network diameter by 42% and increases the bisection bandwidth 130%. HexaMesh uses uniform and rectangular chiplets, which ensures that employing the HexaMesh arrangement does not increase the complexity of designing or manufacturing a chip.

We've used RapidChiplet[48], A toolchain for rapid design space exploration of chiplet architectures, to simulate Mesh and HexaMesh topologies for chiplets used in application 1's architecture. HexaMesh topology results in lower latencies and higher C2C throughput with peak fraction of theoretical maximum throughput of 1.0.

```
data arrival time                                                     43.46

clock CLOCK (rise edge)                                  62.50      62.50
clock network delay (ideal)                               0.00      62.50
clock uncertainty                                         0.00      62.50
msrv32_csr_file_0/MC/minstret_out_reg[63]/CK (QDFFHLX1)
                                                          0.00      62.50 r
library setup time                                       -0.63      61.87
data required time                                                  61.87
--------------------------------------------------------------------------
data required time                                                  61.87
data arrival time                                                  -43.46
--------------------------------------------------------------------------
slack (MET)                                                         18.40
```

Figure 25: Synopsis output for 90nm process

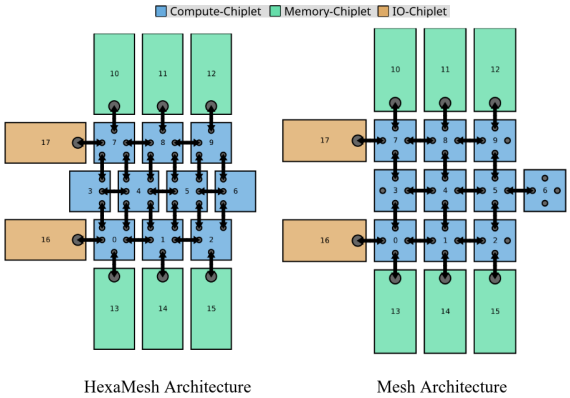


Figure 26: HexaMesh vs Mesh Topology

4. Microarchitectures

4.1. ***Role of CPU***

OS does General Compute Tile scheduling to determine which process will own which General Compute Tile for execution while another process is on hold (in queue of processes). General Compute Tiles themselves work as a task scheduler and assign workloads to domain specific accelerators (DSAs) based on the scheduling policy and by monitoring their runtime status. General Compute Tile allocates memory for use in the computation and then initiates transfer of input data into the caches of DSAs and finally launches a computation kernel (code to be run by DSA).

The workloads in an ADAS are of 2 types, sensor data specific workload and sensor fusion. Sensor fusion uses multiple sensor processed data (lidar, radar and camera) to form a single model or image of the environment around a vehicle.

4.2. ***Need for domain-specific accelerators***

Execution of all tasks comprising video, image data and algorithms based on neural networks will take a lot of time per program in General Compute Tile. Hence domain specific accelerators are used in an SoC. A domain-specific accelerator (DSA) is a specialised hardware component or subsystem designed to accelerate certain types of computations or workloads within a specific domain or application. Image and video processing is taken care

Topology	HexaMesh	Mesh
Avg. C2C Latency	58.73	71.13
Avg. C2M Latency	85.6	98.0
Avg. C2I Latency	88.7	104.2
Avg. M2I Latency	118.67	129.0
Avg. C2C Throughput	0.239	0.132
Peak Fraction of Theoritical Max	1.0	0.75

Table 1: Rapidchiplet Simulation Results (C: Compute, M: Memory, I:I/O)

of by Graphics Processing Tiles and neural data processing by Neural Acceleration Tiles. Graphics Processing Tiles have 1000s of cores and multiple threads to parallelise data processing by manifolds for higher throughput. Same goes with an Neural Acceleration Tile for neural processing.

Other than these, there is a scope of adding more custom IPs for algorithm specific accelerations, like additional DSP chiplet with FFTs and MAC units.

4.3. ***ADAS chiplet based microarchitecture***

So before we go on to design the microarchitecture diagram for ADAS applications, we need to analyse the flow of particular processes going on. ADAS applications use an array of different sensors to visualise the environment and make decisions likewise. In general, camera, radar, IR/lidar and GPS are used. For simplicity, we used camera and lidar sensors for our microarchitecture diagram.

4.3.1. ***Data Flow***

First we analyse the flow of the camera sensor. Camera sensor transmits raw data to CSI2 I/O in ADAS chiplet based processor.

Parallel data from the CSI2 transmitter (four lanes) is encrypted by a lightweight encryptor (required before cable transmission) and serialised by a SERDES IP and sent to CSI2 receiver (CSI2 D-PHY) via a GMSL cable. Before sending data into CSI2 PHY (in host's I/O Tile), it is deserialized (SERDES), decrypted and encrypted for C2C transmission using In-line Cryptography.

Most LiDAR sensors use ethernet as a communication interface. LiDAR raw data is transmitted through the ethernet bus using the TCP/IP ethernet protocol. This data is decrypted using LWC before it enters the host's I/O tile in the ethernet Transceiver. From here the data is encrypted for C2C transmission to the Compute tile through In-line cryptography.

Camera sensor outputs a mosaiced raw image which is demosaiced and filtered before proceeding with further processes. This is achieved by using an ISP (Image Signal Processing) Chiplet. ISP chiplet contains blocks for hardware acceleration required in preprocessing of images such as demosaicing, noise filters, lens correction and gradient correction.

In contrast LiDAR preprocessing is completely executed in software, Because research has proven that there is a strong influence of rain, snow, fog, etc and other environmental parameters on the LiDAR readings,so making a preprocessing block in Hardware is not exactly feasible as we would have to change the hardware frequently. So Using a LiDAR processing software that is already compatible with most if not all of the available LiDARs allows you to use the most appropriate sensor(s) for your application and environment. Next step is the execution of the required algorithms on camera and lidar inputs which are further used in sensor fusion. Various compute tiles are required for processing these algorithms such as General Compute Tiles, Graphics Processing Tiles and Neural Acceleration Tiles which collectively make a compute unit. We require an HBM (High Bandwidth Memory) to store the operating system and programs for required algorithms.

4.3.2. ***Architecture***

For given data flow mentioned above here we propose microarchitecture of chiplet based processor (host) for ADAS applications. An I/O tile (chiplet) is an interface between host and external units. It also takes care of encryption and decryption. There is a separate chiplet for ISP. Lidar data preprocessing is taken care of by compute tile. Compute unit has a collection of 4 General Compute Tiles, 2 Graphics Processing Tiles and 4 Neural Acceleration Tiles, each as a chiplet, connected via suitable interconnects and arranged in an efficient topology (Hexamesh). Each General Compute Tile has direct access to 2 Neural Acceleration Tiles and a Graphics Processing Tile. This together applies required algorithms on the data coming from sensors. Each General Compute Tile has 4 cores and L1 cache pairs, L2 cache. This L2 cache is shared by all cores and also by all other General Compute Tiles via CCIX interconnect technology. All these chiplets were based on 2.5D packaging whereas HBMs on either sides of the compute unit are based on 3D packaging.

The interconnects between each pair of chiplets are enlisted in the table given in figure. The selection is based on parameters like bandwidth and

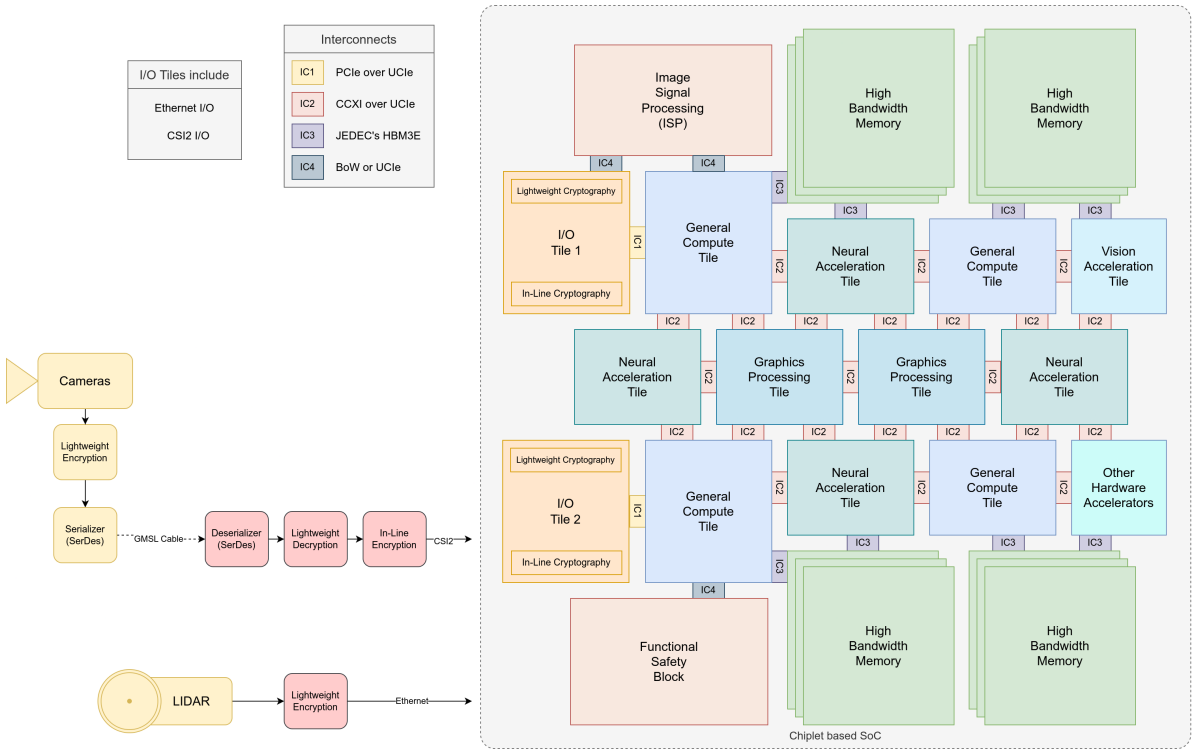


Figure 27: ADAS Microarchitecture Diagram

CSI2 (Camera Serial Interface 2): It is a high speed and bidirectional protocol for image and video data transmission from camera to host (ADAS chiplet based processor).
GMSL (Gigabit Multimedia Serial Link): It is a serial link technology that is used for video distribution in cars.
SERDES (Serializer-Deserializer): It is a functional block that Serializes and Deserializes digital data used in high-speed chip-to-chip communication.

throughput which will be discussed in the later sections.

4.4. Infotainment chiplet based microarchitecture

So before we go on to design the microarchitecture diagram for Infotainment applications, we need to analyse the flow of particular processes going on. Infotainment applications focus on creating an AI assisted cockpit for the driver, they also span a wide range of applications all the way from In cabin entertainment or multimedia to smart cabin and vehicle monitoring

Out of the numerous applications we have chosen to focus on the following

1. Audio processing for NLP(Natural language processing) applications.
2. Stitching images for surround view on principal display.
3. In-cabin cameras for driver or occupant monitoring systems.
4. GPS communication to and from the vehicle.
5. Communication with USB peripherals.

4.4.1. Data Flow

- For application 1: The processed audio signal from the sensor is transmitted through the ethernet AVB(Audio Video bridge) which functions as the PHY layer, the data is then decrypted by using LWC before entering the host I/O tile, Here ethernet transceiver packages the data and then it is encrypted for C2C transmission to the compute tile throught In-line cryptography.

- For application 2: Sensor fusion for all the images from various cameras is performed in the ADAS compute tile, and this data is transmitted using an ethernet bus to the hosts I/O tile using the TCP/IP ethernet protocol. This data is then decrypted by using LWC before entering the hosts I/O tile, Here an ethernet transceiver packages the data and then it is encrypted for C2C transmission to the compute tile through In-line cryptography. From the compute tile the data is exchanged to the I/O tile through Ethernet

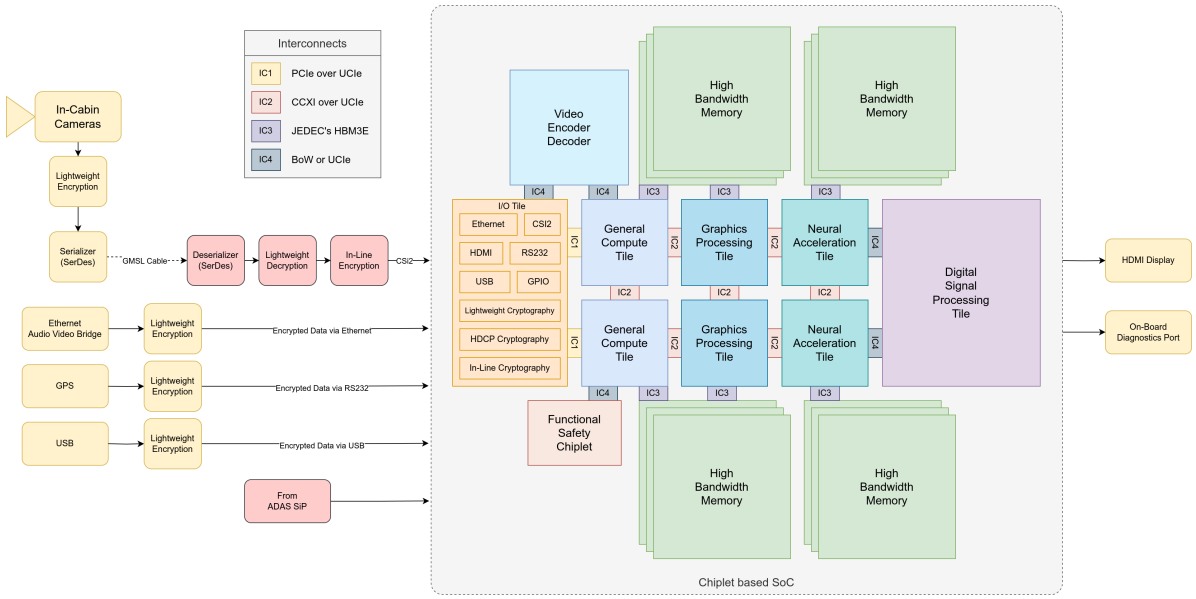


Figure 28: Infotainment Microarchitecture Diagram

using the same mechanism discussed above. This ethernet data is decrypted using LWC and then this signals the HDMI transmitter(Tx) within the I/O tile, the data is then encrypted by using HDCP and it transmitted via the HDMI PHY to a serializer, from here the serialized data is carried out of the I/O tile using GMSL and is deserialized near the display, finally the data is decrypted by the HDMI receiver(Rx) present in the display and the image/video is displayed onto the screen.

- For application 3: Parallel data from the camera which functions as our CSI2 transmitter (four lanes) is encrypted by a lightweight encryptor (required before cable transmission) and serialized by a SERDES IP and sent to CSI2 receiver (CSI2 D-PHY) via a GMSL cable. Before sending data into CSI2 PHY (in host's I/O Tile), it is deserialized (SERDES), decrypted and encrypted for C2C transmission to the DSP block of the compute tile using In-line Cryptography.

- For application 4: GPS signals from a satellite receiver are wirelessly received through the RS-232 GPS transceiver, from here the data is encrypted for C2C transmission to the compute tile, through In-line cryptography. Decision-making based on these signals will be done by the compute tile.

- For application 5: Serialised data from an external USB is received at the peripherals of the I/O tile from here it is decrypted using LWC and sent to the USB IP block, from here the data is encrypted for C2C transmission to the compute tile, through In-line cryptography.

4.4.2. ***Architecture***

For given data flow mentioned above here we propose microarchitecture of chiplet based processor (host) for Infotainment applications.The An I/O tile (chiplet) is an interface between host and external units. It also takes care of encryption and decryption. Compute unit has a collection of 2 General Compute Tiles, 2 Graphics Processing Tiles and 2 Neural Acceleration Tiles, each as a chiplet, connected via suitable interconnects and arranged in an efficient topology (Hexamesh). Each General Compute Tile has direct access to 2 Neural Acceleration Tiles and a Graphics Processing Tile. This together applies required algorithms on the data coming from sensors. Each General Compute Tile has 4 cores and L1 cache pairs, L2 cache. This L2 cache is shared by all cores and also by all other General Compute Tiles via CCIX interconnect technology. All these chiplets were based on 2.5D packaging whereas HBMs on either sides of the compute unit are based on 3D packaging. This microarchitecture diagram specifically focuses on the Main ECU of the infotainment systems, However it must be noted that the infotainment system also contains numerous modules for connectivity like Wi-Fi, bluetooth etc, however they are placed outside our main package and hence are not shown in our micro-architecture diagram,Moreover there is little scope of using chiplets in these modules.

The interconnects between each pair of chiplets are enlisted in the table given in figure. The selection is based on parameters like bandwidth and throughput which will be discussed in the later sections.

4.5. Justification for chosen interconnects

- UCIe through PCIe protocol is chosen for the interconnection to IO blocks because it is the most widely supported standard for peripherals and interoperability is crucial for the ability to get i/o chiplets off the shelf.
- CCXI over UCIe is preferred for compute and accelerator tiles because CCXI allows for a symmetric coherent interface which has a lot of potential in terms of scalability which can be useful in automotive applications.
- JEDEC's HBM3E interface offers high-capacity high-bandwidth on-chip memory with properly developed IPs for easier development.
- For more niche applications, BoW or UCIe with custom protocols are used to ensure optimal operation on a per-chiplet basis.

4.6. Off-the-shelf

4.6.1. Of-the-shelf IPs

In the above proposed microarchitecture diagrams, various I/O IPs were used which can be taken off the shelf. The compute unit stated in the above diagrams consisting General Compute Tiles, Graphics Processing Tiles and Neural Acceleration Tiles can be taken off the shelf and it allows the designer to customize number of chiplets used in compute unit according to need and extent of computation. Memory units can also be take off the shelf according to the need. There are bunch of off the shelf IPs available for security also which will be discussed in further sections.

4.6.2. Custom IPs

Blocks like ISP should be tweaked according to the used camera sensors, image filtering required and external lighting conditions. Same can be extended to blocks for hardware acceleration of particular algorithms. For example, if there is extensive use of computer vision algorithms we may use custom accelerator blocks.

4.6.3. Of-the-shelf but customizable IPs

Synopsis provides off the shelf IP chiplets such as ethernet chiplet, memory expander chiplet and AI accelerator chiplet. These come with flexibility to support multiple package types and are customizable according to specific target applications.

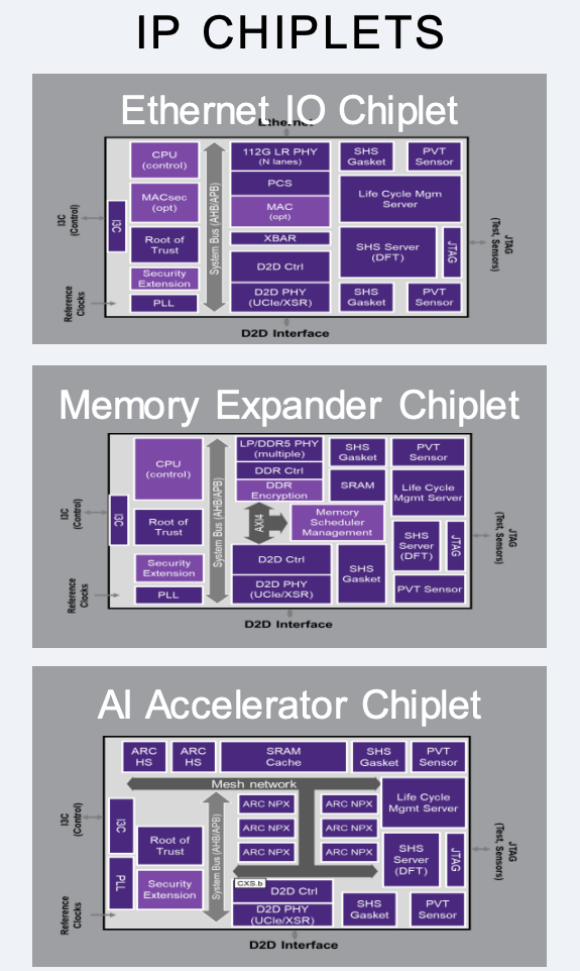


Figure 29: Synopsis off the shelf IP chiplets [5]

These IP chiplets also come with inbuilt security features so that use of these chiplets do not compromise the overal security of the package.

With these of the shelf IP chiplets in hand, the manufacturer can put focus upon designing custom chiplets for hardware acceleration.

5. Interconnects

5.1. Summary of Available Technologies

While major industry players like Intel, AMD, and TSMC offer a number of proprietary die-to-die (D2D) interconnects, open standards must be adopted to ensure interoperability between chiplets made by various firms. A more detailed comparison of proprietary and open interconnect technologies was provided as part of our MidEval report but a summary of the most notable interconnect technology standards is listed below.

- **UCIe:** UCIe is a multi-layer multi-protocol communication technology with well defined physical layer (PHY), D2D adapter layer and

PHY	HBM3	OpenHBI	BoW (Advanced)	UCIe (Advanced)	XSR (Serial)	BoW (Laminate)	UCIe (Standard)
Packaging	2.5D	2.5D	2.5D	2.5D	2D	2D	2D
Bandwidth (Tbps/mm)	1.50	2.30	5.12	5.00	1.75	1.03	1.8
Latency (ns)	4	4	2	2	15	2	2
D2D Reach (mm)	2	2	5	2	50	25	25

Figure 30: Comparision of Inter-Chiplet PHY Technologies [18]

protocol layer. PCIe and CXL protocols are natively mapped with the ability to stream any other protocol. The latest revision, **UCIe 1.1**, introduces enhancements like runtime health monitoring & field repairability and improve reliability with automotive applications in mind. It also introduces the support to multiplex multiple protocols which will allow for a greater degree of interoperability between chiplets from different vendors. UCIe currently supports both standard (2D, organic laminate) and advanced (2.5D, Silicon interposer, bridge, etc) packaging with support for 3D packaging already in motion.

- **BoW:** Bunch of Wires (BoW) defines a physical interface (PHY) specification. It centers on defining the electrical interface and does not provide definitions for higher-level protocols. It primarily concentrates on 2D packaging but also supports 2.5D.
- **OpenHBI:** OCP's Open High Bandwidth Interface (OpenHBI) is an optimized chiplet interconnect which is compatible with DWORD channel and interoperable with HBM3 electricals. OpenHBI supports 2.5D packaging and OpenHBI-L (Laminate) supports 2D packaging.
- **HBM:** JEDEC's High Bandwidth Memory (HBM) is a on-chip three-dimensional DRAM technology with stacks of multiple DRAM dies, which are interconnected by TSVs (Through-Silicon Vias), and microbumps. It is tightly coupled to the host compute die with a distributed interface. HBM uses a wide-interface architecture to achieve high-speed, low-power operation. The latest revision, **HBM3E** can operate on a bandwidth of over 1.2 TB/s compared to the last version, HBM3, with a bandwidth of 819 Gbps/s.
- **USR, XSR SerDes:** Ultra Short Reach (USR) and eXtra Short Reach (XSR) SerDes is among the only open standards for serial chiplet interconnets. Serial interconnects provide a proven, scalable and flexible interface at lower manufacturing and packaging cost but with some drawbacks in latency and power consumption as compared to parallel interconnects.

Protocols

Besides the physical interface, the choice of a suitable protocol is critical to ensure seamless coordination and syncronization among the interconnected chiplets. It dictates how data is shared, cached and updated, influencing factors such as coherency and symmetry. Moreover, the protocol establishes a standardized framework, enabling interoperability among chiplets from different manufacturers. A summary of the standard protocols for a variety of applications is listed below.

- **PCIe:** Peripheral Component Interconnect Express (PCIe) is a standardized interface for that enables vendors to mix and match various peripheral devices for functions such as connectivity, i/o expansion, graphics, memory, and storage. The latest revision, **PCIe 6.0** doubles the bandwidth and introduces Integrity and Data Encryption (IDE) which greatly improves reliability and security, making it more suitable for automotive applications.
- **CXL:** Compute Express Link (CXL) targets intensive workloads for CPUs and purpose-built accelerators where efficient, coherent memory access between a Host and Device is required. It is an asymmetric and coherent interface designed for i/o, cache and memory interconnection through CXL.io, CXL.cache and CXL.mem protocols respectively.
- **CCIX:** Cache Coherent Interconnect for Accelerators (CCIX) enables two or more devices to share data in a cache coherent manner. It is different from CXL in that it is symmetric and inherently more complicated but it allows for a greatly scalable architecture of CPUs and accelerators, which makes development of SoC lineup with varying capabilities very convenient.
- **AMBA CHI:** Arm's Advanced Microarchitecture Bus Architecture (AMBA) Coherent Hub Interface (CHI) architecture provides the performance and scale required for systems with a very large network of processors, accelerators, and memory. More fundamentally, CHI is high speed, credited, and packetized, which makes it ideal for chiplets as well.

5.2. *Factors to consider while choosing an interconnect technology*

- **Performance:** High throughput and low latency interconnects come at the expense of shorter reach, higher manufacturing cost and packaging limitations. Thought must be given to whether the application needs as much performance and other factors must be kept in mind.
- **Packaging:** The choice and performance of the interconnect is limited by how the chiplets are packaged together. Technologies like UCIe and BoW support both 2D and 2.5D packaging while XSR only supports 2D and HBM only supports 2.5D. 2.5D interconnects typically provide greater efficiency, increased bandwidth and reduced latency, but this comes with the trade-off of limited reach and higher manufacturing cost.
- **Protocol:** Support for the protocols suitable for the specific chiplets being interconnected is essential for seamless communication.
- **Market Viability:** Wide market support for the interconnect is crucial for interoperability among chiplets from different vendors and for streamlined development. It must be made sure that the technologies chosen are currently viable and will continue to be a few years down the line.
- **Application:**
  - **Scaling:** Connecting clusters of CPUs or accelerators require high performance interconnects with suitable protocols to be able to handle the topology in a scalable manner.
  - **Aggregating:** Factors like interoperability and market support are crucial while aggregating disparate functions like Co-Packaged Optics or Application Specific Integrated Circuits (ASICs).
  - **Disaggregating:** Disaggregating components like processing and I/O in SoCs to allow optimal process node per component and reusability needs interconnects with wide support for different process nodes and packaging technologies.
  - **Splitting:** Splitting large compute or switch dies to improve yield and workaround reticle limit benefits from an interconnect controllable at lower levels with high performance and support for custom protocols while factors like interoperability do not matter as much.

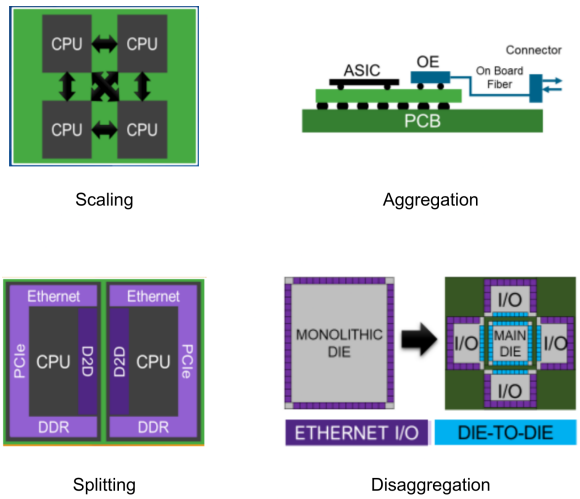


Figure 31: Example Chiplet Applications [5]

5.3. *Optimal interconnect choices for automotive applications*

Looking at the currently available standards, we can immediately rule out a few options. There's little benefit from using Intel's AIB technology as UCIe is the evolution of AIB and all efforts from Intel and other companies are directed towards UCIe instead. OCP's OpenHBI standard is also dwindling in support and UCIe offers better metrics at same data rate with several automotive enchancements for greater reliability.

The optimal interconnect technologies and their applications are:

- **UCIe 1.1:** UCIe is very strongly supported, has outstanding metrics and focuses on automotive applications in its recent revision and there are plans for further enhancements in the future. UCIe 1.1 is suitable for interconnecting compute, I/O and other digital chiplets that require high performance and interoperability with the suitable protocols.
- **BoW:** BoW is well received in the market and provides lower level control which is beneficial for interconnecting analog components like ADCs, DACs or Analog Front-Ends (AFEs) to digital components. It supports both 2D and 2.5D packaging so cost sensitive solutions can be created using BoW and UCIe together.
- **HBM3E:** HBM3E is widely adopted and is a great option for very high bandwidth ap-

plications like connecting memory to digital chiplets. It only supports 2.5D packaging and may not be viable for cost sensitive applications and previously mentioned interconnects can come in use.

- **XRS SerDes:** XSR is a popular standard and a viable option when serial interconnects are necessary. Serial interconnects offers ease of use and a much longer reach than other technologies. It can be used when interconnecting Optical or Networking components.

6. Innovation

Here we are proposing a unique interconnect, i.e. photonic interconnect, that will be very useful in high-speed data transmission between GPU chiplets and memory. Since GPU does a lot of processing, high-speed interconnect will give an edge to increase the throughput of ADAS and make critical decisions.

6.1. Photonic Interconnects (GPU chiplet to memory communication)

Since optical signals can travel at the speed of light in a waveguide, one-hop data communication is feasible, irrespective of the location of the source and destination nodes in chip-based architectures. This requires the development of interconnection networks that utilize silicon photonic devices and waveguides to modulate and transmit data between processing units(GPU chiplets) and memory in a chiplet-based system. Wavelength division multiplexing along with an optical tunable splitter is used for the transmission of data between chiplet and memory.

- **WDM-based Silicon Photonic Links:** Distinct wavelengths of light are utilized to transmit different signals, with each signal occupying a unique wavelength. Research has demonstrated that up to 80 wavelengths can be multiplexed in a single waveguide, with each operating at a data rate of 25 Gbps. In comparison to electrical networks, the power consumption of transmitters and receivers in a photonic link is significantly lower. As transmitters and receivers account for the majority of the total power consumption in a photonic link, high energy efficiency can still be achieved, even as communication distances increase.

- **Optical Tunable Splitter:** At a specific wavelength, the tunable optical splitter operates in the transient region between the on- and off-resonance state. It is based on an active microring resonator i.e. essentially a PIN diode structure with n-type and p-type doping inside and outside, respectively. By applying a voltage to control the carrier concentration in the ring waveguide, the refractive index can be actively controlled. When no bias voltage is applied, the splitter is in the off-resonance state, and all power passes through the through port. Conversely, when a certain bias voltage is applied to the PIN diode, the splitter allows α fraction of the power to pass the drop port in the on-resonance state, as shown in Fig. 2(b), while the remaining $1 - \alpha$ fraction of the power passes through the through port along the waveguide. The bias voltage can vary from 0V to 5V, resulting in a corresponding split ratio of 0.4 to 1.8. For this purpose, a fast digital-to-analog converter (DAC) module generates any voltage between 0V and the modulation voltage required for the tunable splitter. Note that commercially available 4- to 6-bit DACs that switch at 2-5 GHz already meet these requirements. To achieve a split ratio exceeding the range of 0.4 to 1.8, multiple optical tunable splitters must be cascaded.

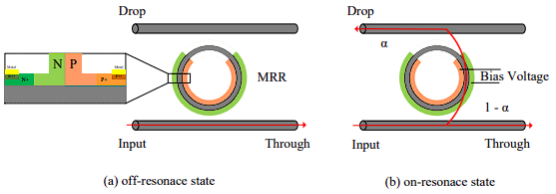


Figure 32: The tunable splitter in different states [74]

6.2. Photonic Interposer and GPU Chiplets

The interconnection between chiplet and memory uses 2.5 D integrated technology. For more understanding of the communication of a general system, an example with 4 GPU chiplets and a memory (GLC) is shown. Communication between 1 chiplet and memory pair is shown, and connection in a system can be understood likewise.

The circuits of transmitters, receivers, CU, and SMs (including the PLC) on the chiplet 2 are integrated on a separate silicon chiplet die and connected to the photonic interposer via micro-bumps. The CU is used to control the state of the corresponding modulator or filter at the time of communication, thus controlling SM to send or receive data. The micro-bump has two main roles, one is

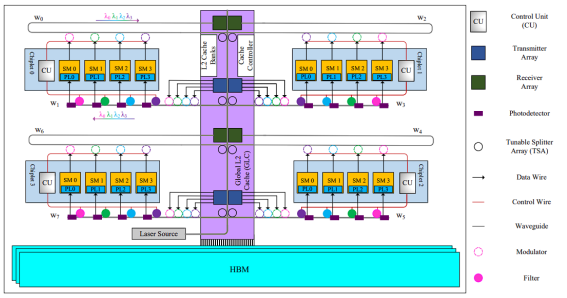


Figure 33: Four chiplets and 4 SMs per chiplet architecture [74]

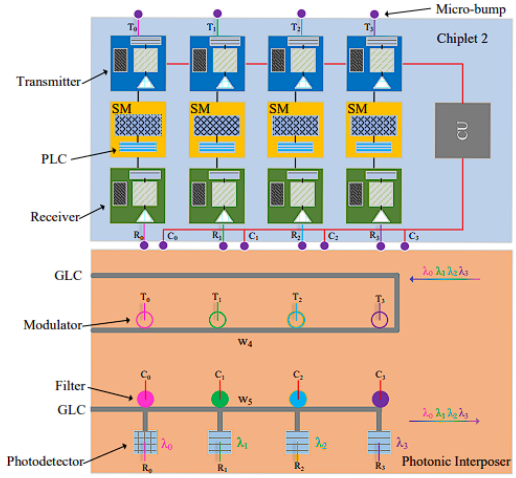


Figure 34: Physical layer layout of photonic network [74]

for the GPU chiplets to transmit CU control signals and thermal tuning signals to the photonic interposer, and the second role is to realize data communication between the photonic interposer and the GPU chiplets. Optical devices such as photodetectors, filters and modulators are placed in the photonic interposer. Each filter is connected to the same waveguide w_5 from GLC, and the corresponding wavelength is detected and transmitted to the photodetector through a separate local waveguide. Each modulator is connected to waveguide w_4 and is responsible for modulating the corresponding wavelength to transmit data from the PLC back to the GLC. The modulators and filters are vertically coupled to the waveguides.

The modulated wavelengths, namely w_0, w_1, w_2, and w_3, are selected using vertically coupled filters and then transmitted through local waveguides to the corresponding PDs in waveguide w_5. The photocurrent signals detected by the PDs are then transmitted to the receiver via microbumps. After O/E conversion into electrical signals, the data is forwarded to the PLC of each SM on chiplet 2.

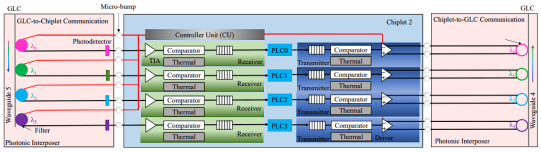


Figure 35: GPU chiplet and photonic interposer in chiplet 2 [74]

7. Safety and Security

Within this section, our primary focus revolves around addressing the intricacies of 2.2.b, offering a detailed exploration of diverse attack vectors and comprehensive strategies for mitigation. We go beyond the surface, providing a nuanced report that spans preventive measures against IP overuse and the seamless implementation of dynamic safety protocols.

7.1. Importance of security in automotive industry

The automotive industry is undergoing a significant transformation. Cars are becoming more sophisticated and valuable with increased connectivity and capabilities to provide a better user experience. They are also collecting and transmitting more and more sensitive data and thus are becoming very attractive targets for attacks. Cybercrime in the automotive industry is growing rapidly. According to the AV-TEST Institute, the number of malicious programs targeting automobiles has increased to roughly 1.1 billion at the end of 2020, from approx. 65 million in 2011. Upstream Security reported in a 2019 cyber hack security study that there was a 94% year-over-year growth in automotive hacks since 2016. [6]

As vehicles have come to rely heavily on software and an increasingly complex software supply chain, the cyber threat landscape continues to evolve, and security and safety standards are more critical than ever.

7.2. ISO 26262 and ASILs

ISO 26262 mandates a functional safety development process (from specification all the way through production release) that automotive OEMs and suppliers must follow and document (for compliance) to have their devices qualified to run inside commercial (passenger) vehicles. It outlines a risk classification system (Automotive Safety Integrity Levels, or ASILs) and aims to reduce possible hazards caused by the malfunctioning behavior of electrical and electronic (E/E) systems. [69]

Adopting ISO 26262 helps ensure that the safety of car components is considered from the beginning of the development process. It provides a comprehensive framework for managing safety throughout the entire lifecycle of an automotive component, from initial risk assessment to final decommissioning. By following ISO 26262, automotive manufacturers can ensure that their suppliers are meeting safety standards, preventing costly issues from arising during the production process.

ASIL refers to Automotive Safety Integrity Level. It is a risk classification system defined by the ISO 26262 standard for the functional safety of road vehicles. [66] The standard defines functional safety as "the absence of unreasonable risk due to hazards caused by malfunctioning behavior of electrical or electronic systems." ASILs establish safety requirements—based on the probability and acceptability of harm—for automotive components to be compliant with ISO 26262.

There are four ASILs identified by ISO 26262 - ASIL A, B, C, and D. ASIL A represents the lowest degree and ASIL D represents the highest degree of automotive hazard. Systems like airbags, antilock brakes, and power steering require an ASIL-D grade—the highest rigor applied to safety assurance—because the risks associated with their failure are the highest. On the other end of the safety spectrum, components like rear lights require only an ASIL-A grade. Headlights and brake lights generally would be ASIL-B while cruise control would generally be ASIL-C.

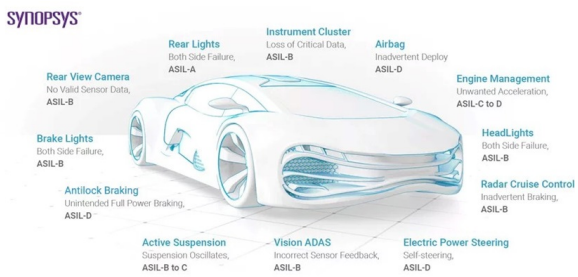


Figure 36: ASILs as per ISO 26262 [66]

ISO 26262 is a goal-based standard that's all about "preventing harm". Despite their challenges, ASIL classifications are intended to "prevent harm" and help us achieve the highest safety rating possible for myriad automotive components across a long and often disjointed supply chain. Key benefits include Establishing safety requirements to mitigate risks, managing and tracking safety requirements, and ensuring that standardized safety procedures have been followed in the final product.

Due to the expanded number of attacks on connected cars in ADAS/HAD and V2X/Infotainment technologies such as Bluetooth/BLE, WiFi, cellular including 5G, GPS, USB, and in-car networks such as CAN, MIPI, and automotive Ethernet, a holistic cybersecurity engineering is required. Cybersecurity impacts every level of the automotive supply chain starting with semiconductor SoCs.[67] and SRC roadmap

7.3. Possible vulnerabilities

1. **Hardware Trojans**: Hardware Trojan threats can originate from either chiplet OCMs that want to compromise other chiplets at the post-integration phase or rogue foundries for tainting the reputations of Challenge Promising Technology chiplet OCMs. Therefore, hardware Trojans can be inserted by altering (a) the design-in-design houses or (b) masks in untrusted foundries. To detect Trojans in chiplets, statistical methodology can be used to derive a set of test patterns and new verification methodologies (pre- and post-silicon) for the excitation of rare conditions at internal signals, as these are typically chosen as Trojan triggers for stealthy behaviors. It is highly improbable for trojan Application-Specific Integrated Circuits (ASICs) to be introduced into a car's communication system once the car has been manufactured.
2. **IP Piracy**: Some chiplet original component manufacturers (OCMs) design the hardware and rely on the offshore foundries to fabricate the silicon, which introduces the risks of IP piracy since a rogue foundry can extract high-level design information from the physical layouts for future IP infringement. Such attacks can be prevented by having a reliant and robust supply chain.
3. **Network attacks**: Data sent between various automotive parts or vehicles and other systems is manipulated in a network attack. Such data tampering may result in the interception of private information or the modification of control systems. Given the high likelihood of such attacks, it underscores the urgency for the implementation of safe and secure communication protocols.
4. **Attacks on the Cloud**: Modern cars are becoming increasingly connected to the cloud for services like internet upgrades and data storage. Attacks directed at the cloud have the potential to corrupt updates, reveal private car data, and interrupt vital services.
5. **IoT attacks**: Hackers may exploit holes in connectivity between automobiles and external devices when cars join the Internet of Things (IoT). Unauthorized parties can get

access to and alter car functionalities by breaching the IoT ecosystem.

6. **Reverse engineering**: One of the most prevalent IP piracy techniques is reverse-engineering de-packaged ICs to extract the physical layout, which can be applied to individual chiplets and SiP systems.
7. **Side-channel attacks**: The security of cryptography devices applied in cars often relies on strict secrecy on the design of the embedded systems, and this type of design is vulnerable to leakages from insiders Most modern cryptographic devices are implemented using semiconductor logic gates, CMOS logic (Complementary Metal-Oxide-Semiconductor). Unfortunately CMOS logic has a data-dependent power consumption enlarging the risk of side-channel analysis.
8. **Counterfeit chiplets**: Because of their potential defects, insufficient performance, and poor reliability, counterfeit chiplets (like overproduced chiplets) from rogue foundries or remarked ones (e.g., claiming higher grade or performance) likely induce unexpected failures if they are integrated into final SiP products.
9. **Sensor manipulation**: The manipulation of LiDAR sensors poses a concerning potentiality. Picture a scenario where a bad actor deceives a LiDAR sensor into detecting a non-existent vehicle obstruction. This misleading information could then be transmitted to the electronic control unit (ECU), prompting the vehicle to make unnecessary swerving maneuvers.

These attacks are categorized as follows: [16]

1. **Spoofing**: masquerading as a legitimate user, process, or system element
2. **Tampering**: modification/editing of legitimate information
3. **Repudiation**: denying or disowning a certain action executed in the system
4. **Information disclosure**: data breach or unauthorized access to protected information
5. **Denial of Service**: disruption of service for legitimate users
6. **Elevation of privilege**: gaining higher privilege access to a system element by a user with restricted authority.

7.4. Mitigation methods

1. **Encryption**: Using reliable encryption methods to protect data sent between internal and external systems, networks, and vehicle components. locking the circuitry such that the original functionality is available only when the correct key is applied.
2. **Watermarking**: The countermeasures can be passive watermarking and active design obfuscation. Watermarking embeds a unique identifier (e.g., a sequence of bits stored in dedicated registers) into the chiplet to enable IP authenticity verification as proof of ownership.
3. **Firewalls**: Applying firewalls to a vehicle's network allows for the monitoring and managing of data flow while thwarting malicious invasions and unauthorized communications.
4. **Intrusion Detection and Prevention Systems (IDPS)**: These systems keep track of the networks and parts of vehicles in real-time, spotting and blocking any suspicious activity or unauthorized access attempts.
5. **Secure Boot and Firmware Verification**: Ensure that only legitimate and authorized software can run on vehicle control units to prevent unauthorized alterations.
6. **Access Control**: Restricting access to vehicle systems through robust authentication and authorization processes to stop unauthorized users from taking over.
7. **Over-the-Air (OTA) Updates Security**: Using safe techniques to update car software while assuring its validity and preventing possible tampering.
8. **Security Audits and Penetration Testing**: Regularly evaluating automotive systems for vulnerabilities via penetration testing and security audits, spotting weak spots before nefarious attackers take advantage of them.
9. **Physical Unclonable Functions (PUFs)**: PUF primitives or other security primitives can generate responses by leveraging device process variations, thus uniquely fingerprinting every silicon die, serving as an indication against counterfeit chiplets.
10. **Masking**: Data masking substitutes original values in a data set with randomized data using various data shuffling and manipulation techniques. The obfuscated data maintains the unique characteristics of the original data so that it yields the same results as the original data set.

Protecting Against Security Breaches: From a hardware perspective, preventing unintended activities comes down to ensuring that silicon chips, as well as the high-speed interfaces transferring the data to ECUs, are protected from exploitation. It's important not to limit the information

provided to prevent user errors due to information overflow. At best, incidents are often flagged internally to be investigated at a future date if the log is ever retrieved, normally due to an accident. Usually, such system errors or failures trigger a recall that requires a trip to the dealership and, for the carmaker, a potentially expensive fix. A better approach would be to build in the security to prevent a malicious act from causing the error in the first place. Look at the chips supporting each sensor, as well as the high-speed buses that transfer sensor data to the respective ECU. Deploy redundancy where necessary, so that if one area goes down, this won't affect another critical area.

*7.5. **Functional safety***

The rising complexity of functional safety and security features of cars necessitates heightened control and monitoring. A prevalent trend involves transitioning from rudimentary hardware controls to an adaptable solution featuring an embedded CPU and software-managed "safety island." This section delves into the principles surrounding a standard safety island implementation, offering illustrative examples to showcase its practical applications.

As functional safety is something that is used by almost everyone in the automotive industry, there is no need for a company to manufacture its own safety island. Following are some of the processors that we found that will effectively ensure functional safety while complying with ISO 26262 standards:

1. Synopsys ARC EM22FS Safety Processor [56]
2. Cast Inc. EMSA5-FS [22]
3. Texas Instruments Jacinto 7 Processors [34]

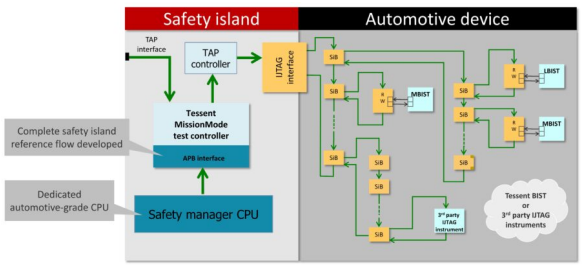


Figure 37: Typical Safety Island Architecture [35]

Functional safety in cars is a comprehensive strategy aimed at guaranteeing the dependable performance of essential safety functions, even when facing potential faults. This approach involves meticulous hazard analysis, rigorous risk assessment, and the implementation of safety requirements. Notably, it encompasses the integration of redundancy and continuous monitoring mechanisms to ensure the reliability of safety-critical systems. Beyond legal compliance and meeting regulatory standards, functional safety plays a pivotal role in instilling consumer confidence and competitiveness within an automotive industry increasingly characterized by advanced technologies, such as ADAS and autonomous features." As stated by Texas Instruments in their datasheet. Though dedicated functional safety islands are not required by the law, it is necessary in today's automotive industry, especially in densely populated countries like India to ensure the safety of not only the driver and passengers but also the pedestrians and anyone in the vicinity of the vehicle. It also takes us one step further toward level 5 autonomous driving.

Advantages of using dedicated functional safety modules:

1. Fundamental diagnostics, which covers test circuitry for memory, clocks, power, core, and interconnect.
2. Hardware isolation capabilities like separate voltage/ power/reset, firewalls, memory management units (MMUs), and microprocessors (MPUs), which simplify Freedom From Interference (FFI) in systems that support mixed-criticality operations (Eg: ASIL-B and ASIL-D).
3. Application-specific hardware diagnostics such as freeze-frame detection.

*7.6. **Die-to-die connection security in chiplets***

Integrating multiple chiplets into a heterogeneous package opens the door for security breaches and potential risks associated with malicious modifications or attacks on the individual chiplets during design, assembly, or testing. Moreover, since chiplets are typically designed and manufactured by different vendors, there exists a potential risk wherein a malicious actor could exploit vulnerabilities within one of the vendors' systems. Such compromise could then be leveraged to jeopardize the entire chiplet-based system.

Some of this depends on the complexity of the supply chain for chiplets. Companies like Intel, AMD, and Marvell develop their own chiplets, which minimizes supply chain threats. But companies assembling and integrating commercially developed chiplets will have a much tougher time. [12]

Presently, there is a necessity for coordinated efforts among those involved in developing chiplets since certain countermeasures require seamless collaboration between the chiplets and the accompanying software. Moreover, the adoption of chiplets introduces an expanded attack surface, giving rise to two distinct categories of issues.

7.6.1. *Vulnerabilities in die-to-die connections*

The first category pertains to **Hardware Trojans**. In heterogeneous systems where chiplets originate from different manufacturers, there is a conceivable threat wherein some companies might engage in reverse engineering of the chips within the chiplet, replacing them with malicious counterparts. While this may sound like a scenario from science fiction, it is a tangible concern, particularly in the context of IoT and similar applications where achieving such compromise is potentially more feasible than in larger systems.

The second challenge is associated with the broader attack surface, primarily due to the more **traceable connections between chiplets**. This heightened traceability opens the door to potential man-in-the-middle attacks.

The interconnect functions akin to a highway, featuring multiple lanes extending in various directions. In instances where two traffic flows intersect, a priority arbitration policy comes into play, determining which traffic flow receives precedence. Requests deemed more 'important' take priority, such as those originating from programs integral to the computer's essential operations.

On-chip interconnects enable communication between the processor cores, and the vulnerability lies in situations when programs run concurrently on multiple cores. **Delays** can occur when multiple cores use the same interconnect to transmit data across the chip, and malicious agents can use these delays for **side-channel attacks** where encrypted information can be stolen from elsewhere in the system.

In a scenario where an individual or organization has successfully disaggregated their System-on-Chip (SoC), assuming control over both ends and possessing a thorough understanding of their supply chain, the risk of a malicious chiplet infiltrating the supply chain appears to be quite low. In this situation, a package has been created, incorporating both their die and certain third-party die that necessitate communication between them. The challenge arises in establishing a standardized protocol for communication where one of the chiplets takes the lead, designating itself as the manager and the other as the subordinate. This involves a crucial step of authenticating the chiplet's legitimacy. Presently, there is no standardized approach for this process, and while the UCIe specification acknowledges this as a future consideration, a universally accepted standard has yet to be established. [38]

7.7. *Encryption*

The main purpose of encryption is to protect the confidentiality and integrity of data, ensuring that only authorized parties can access and understand the information. Various encryption algorithms have been developed over the past years, offering high security, low latency, low power utilization, etc. Based on the type of data, the protocols of transmission, the level of security, and the hardware requirements, we propose a few security protocols that can be implemented in our system.

7.7.1. *Inline Encryption*

For communication between two chips, we will use line-by-line/inline encryption. Inline encryption is a technique used to encrypt data as it is transmitted or stored, with the encryption and decryption processes being seamlessly integrated into the data path. The term "inline" refers to the fact that encryption and decryption operations occur in real-time and are embedded within the data flow, without requiring separate steps.

Off-the-shelf IPs to consider:

1. Synopsys SD/eMMC Host Controller IP [61]
2. Rambus IME IP 340 [28]

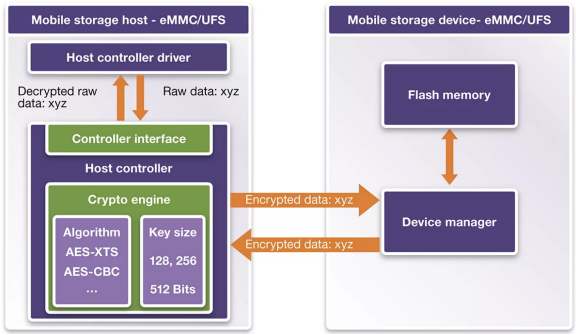


Figure 38: Mobile Storage Host Controller IP with built-in cryptographic engine [21]

Why Inline Encryption?

1. It provides **real-time protection** for data as it is processed, transmitted, or stored. This ensures that sensitive information is immediately secured, reducing the window of vulnerability.
2. It reduces the risk of **unauthorized access** and data exposure. So, even if the attacker gains access to the storage or intercepts data during transmission, the encrypted data remains unreadable without the appropriate decryption keys.
3. Inline encryption systems include mechanisms for **secure key management**, making it easier for organizations to handle encryption keys in a centralized and secure manner.
4. It can be implemented within hardware components, such as storage devices or network

equipment, which **optimizes resource utilization**. This often results in minimal impact on system performance compared to post-processing encryption methods.
5. It operates in the background without requiring additional user intervention or specialized handling, which makes it **easy to implement** without disrupting existing workflows.
6. Inline encryption solutions are **scalable** and can accommodate growing data volumes and processing demands.

7.7.2. Lightweight Encryption

For communication between different boards, lightweight encryption algorithms can be used. Lightweight encryption refers to cryptographic algorithms and protocols that are specifically designed to provide security with minimal computational and resource requirements. These lightweight cryptographic solutions are well-suited for constrained environments, such as embedded systems, devices, and applications where there are limitations on processing power, memory, and energy consumption. The algorithms are designed to execute quickly, minimizing the time required for encryption and decryption operations, and also require less computational overhead.

Off-the-shelf IPs:

1. Rambus ASCON-IP-41 [47]
2. Sparkle Suit by KISA [55]

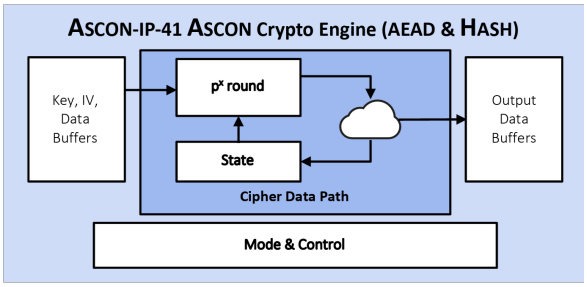


Figure 39: ASCON-IP-41 Architecture [47]

Why Lightweight Encryption?

1. Lightweight cryptography is designed to operate efficiently in **resource-constrained environments**, requiring minimal processing power, memory, and energy.
2. These algorithms are optimized for speed providing **low latency**, which makes them suitable for applications with strict performance requirements.
3. These algorithms and hardware are designed to **minimize energy consumption**, which makes them suitable for battery-powered devices.
4. Lightweight cryptography can contribute to **cost-effective** hardware implementations as they require less powerful and expensive hardware components.
5. Many lightweight encryption algorithms have undergone **standardization** processes, ensuring interoperability and compatibility across different hardware platforms.

7.7.3. HDCP

High-bandwidth Digital Content Protection (HDCP) is a digital content protection system that safeguards audio and video when it's being transmitted down a particular connection. It is supported by DisplayPort, HDMI, and the legacy DVI. HDCP encrypts the signal coming from a digital source device, like a Blu-ray player, and then performs a digital "handshake" with a compatible display or another device, where the two devices share secret codes to see if they are both legitimate and support HDCP. If that handshake is accepted, then the signal can be decrypted and displayed. [68]

Off-the-shelf IPs:

1. DisplayPort Intel FPGA IP [19]
2. Synopsys Embedded Security Modules for HDCP 2.3 on HDMI IP [57]

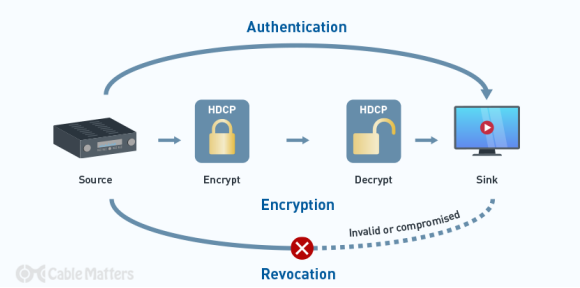


Figure 40: Three main protocols of HDCP [68]

Why HDCP?

1. HDCP helps **prevent unauthorized duplication** or copying of high-definition content.
2. It employs encryption to **secure** the communication between source devices and display devices. This encryption helps ensure that the content remains secure during transmission.
3. HDCP allows for the secure transmission of high-definition content **without degradation in quality**.
4. HDCP is widely supported across a range of consumer electronic devices, including TVs, monitors, projectors, and audio-video receivers. This ensures that users can enjoy high-quality content across various devices without **compatibility** issues.

7.7.4. ***Ethernet Security***

These security IPs can be used to transmit secure data through ethernet cables between LiDAR and Camera modules, and the main processing board. Data security between Ethernet-connected devices is expanding due to multiple factors, such as the exponential growth of data containing sensitive and private information, new laws and regulations, and the technological advances in markets such as cloud computing, mobile/5G, and automotive, to support faster, more scalable and ultimately more efficient networking architectures. [58]

Off-the-shelf IPs:

1. Synopsys MACsec Security Modules for Ethernet [59]
2. Cisco IEEE 802.1AE (MACsec) IPs [29]

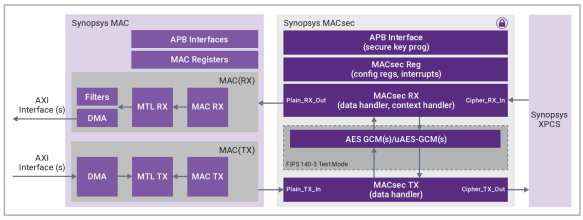


Figure 41: Synopsys Ethernet Security Solution block diagram [59]

Why Ethernet Security?

1. Encryption and secure communication protocols help **prevent unauthorized access** to data as it travels over the Ethernet network.
2. Ethernet security helps in maintaining the **integrity** of the data. By using techniques such as checksums and hashing, it becomes possible to detect and prevent data tampering during transmission.
3. Security measures can help **prevent or mitigate** the impact of Denial-of-Service attacks, which aim to disrupt the normal functioning of a network by overwhelming it with traffic.
4. By ensuring the security of sensitive data, organizations can **reduce the risk of legal consequences**.

7.7.5. ***Security of PCIe interfaces***

PCI Express (PCIe) is a high-speed interface standard used for connecting various hardware components within a computer system. While PCIe itself doesn't have inherent security features, securing PCIe involves implementing measures to protect the integrity, confidentiality, and availability of data and resources transmitted over PCIe interfaces.

Off-the-shelf IPs to consider:

1. Synopsys PCI Express (PCIe) IP Solutions[60]

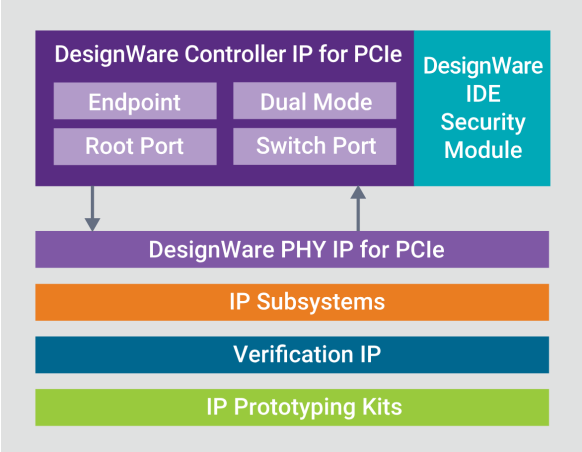


Figure 42: Solutions for PCIe security offered by Synopsys [60]

Why securing PCI Express?

1. PCIe interfaces can be susceptible to **side-channel attacks**, where an attacker gains information by analyzing the physical characteristics of the communication. Security measures help mitigate the risk of such attacks.
2. Security measures help protect against **physical attacks**, such as tampering or the insertion of malicious devices into PCIe slots.
3. PCIe supports multiple devices connected to the same bus. Security measures help **isolate and protect the communication channels** between these devices, preventing one compromised device from affecting others.
4. Security measures, such as digital signatures and secure boot processes, help ensure the **authenticity** of PCIe devices and components, preventing the use of counterfeit or compromised hardware.

7.7.6. ***USB Security***

USB authentication and encryption are implemented at the system level. Authentication requires a combination of trusted domains for executing secure software, secure storage for keys and certificates, and security accelerators for user-friendly response time. With authentication, the host can authenticate a trusted peripheral such as a keyboard, mass storage device, power supply, allowing pipe setup, providing certain power levels and voltages, enabling security options, and more.

Off-the-shelf IPs to consider:

1. Synopsys USB IP Solutions [62]

Why securing USB?

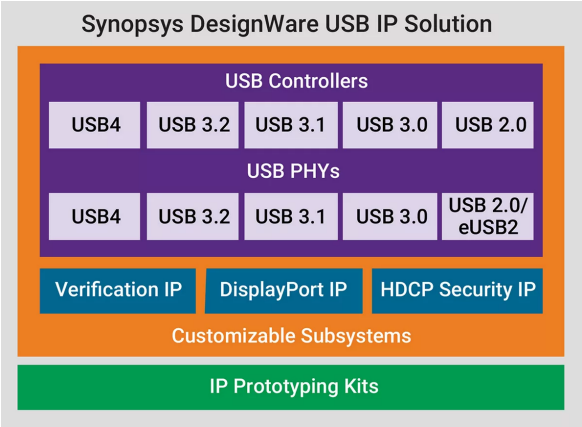


Figure 43: Various IP Solutions provided by Synopsys [62]

1. With the increasing number of data breaches and cyber threats, securing USB devices helps **prevent data breaches** and unauthorized disclosure of sensitive information.
2. USB security measures help mitigate the risk of **insider threats**, where individuals with authorized access might misuse USB drives to steal or compromise sensitive data.
3. USB security ensures the **confidentiality** and privacy of the information stored on portable devices
4. Implementing security measures, such as disabling autorun features and scanning USB drives for malware, helps **protect systems** from potential threats.

7.8. Watermarking IPs

The flexibility of reusable IPs expedites the creation of SoC products and brings a lot of potential profits to the IP providers. Systems-on-chips (SoCs) combine dozens of IP cores licensed from different vendors. The reuse of IP cores comes with a risk of IP piracy and overuse. Problems could be in various forms, such as claiming someone else's IP as your own or reselling it, not giving an IP designer credit where it is due, and open-source IPs being used for commercial purposes. Watermarking, the process of marking an asset with a known structure, has been proposed to detect IP theft and overuse. [4]

Watermarking in hardware IPs is the mechanism of embedding a signature (or a unique code) into an IP core without altering the original functionality of the design. The ownership of the IP can be later verified when the watermark is extracted. The IP watermarking steps are illustrated in the given figure

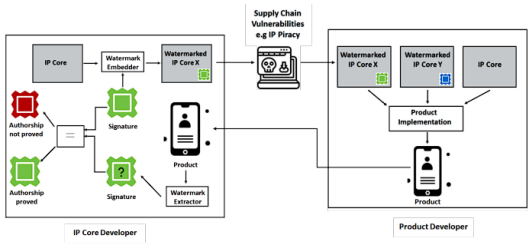


Figure 44: IP Watermarking Embedding and Verification [4]

7.8.1. Existing watermarking techniques [4]

1. **Side Channel-based Watermarking:** Side-channel analysis can utilize the physical information leaked from a cryptographic device and is frequently used to retrieve secret keys and their security issues. The side-channel-based watermark is that instead of leaking out secret information, the side-channel is engineered to contain a watermarking signal. The main idea is to use a side-channel, e.g., power consumption, to embed a watermark into an IP core. Then the verifier uses that side-channel information to extract the watermark and prove ownership.
2. **Constraint-based Watermarking:** Several NP-hard optimization problems are used in every phase of the IC design process (i.e., system synthesis, behavioral synthesis, logic synthesis, and physical synthesis). NP-hard problems are solved using EDA tools available at that stage. A generic optimizer is used to solve constraint satisfaction problems (CSP). As a result, the watermarked design can be derived from the algorithmic constraints added to such a solution.

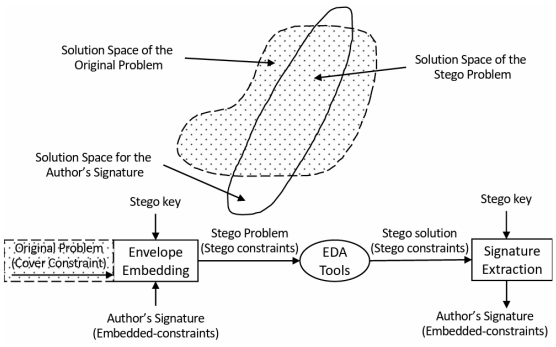


Figure 45: Basic idea of constraint-based watermarking [4]

3. **Digital Signal Processing Watermarking:** Watermarking using digital signal processing (DSP) is introduced at the algorith-

mic level of the design flow. The primary goal of the techniques is to enable designers to make slight modifications to the decibel (dB) requirements of filters without compromising their operation.

4. **FSM-based watermarking:** The FSM-based watermark is embedded at the behavioral level by introducing additional FSM states or transitions which does not interfere with the normal chip functionality.

*7.8.2. **Technique feasible for our application***

This is a more likely scenario where the IP owner has access to the IC but lacks access to the IP. We believe this is a more practical case because the attacker, who steals the IP or overuses it, should not be expected to allow the IP owner easy access to the IP. In general, this is a scenario where the IP is pirated by an attacker or a rogue SoC integrator without any contract with the IP owner. As a result, only the inputs and outputs of the IC, incorporating the test structure, are available to prove ownership of the IP. Hence, attention and innovative research are required to design watermarks that prove ownership outside of the contract. Keeping this in mind, side-channel-based watermark signature extraction is feasible. However, the circuitry of the watermark must be resilient against removal and reverse engineering attacks. RTL, gate-level, or layout-level abstraction is possible.

*7.9. **Protection against external attacks on Vehicular Sensors***

Today's modern vehicles contain anywhere from sixty to one hundred sensors and exhibit the characteristics of Cyber-Physical Systems (CPS). There is a high degree of coupling, cohesiveness, and interactions among the vehicle's CPS components (e.g., sensors, devices, systems, systems-of-systems) across sensing, communication, and control layers. Cyber-attacks in the sensing or communication layers can compromise the security of the control layer. In this section, we discuss the attacks on two major sensors in our system, LiDAR and Camera, and some countermeasures. [15]

*7.9.1. **LiDAR***

Attacks on LiDAR:

1. **Replay Attack**: Attackers can receive and record signals sent by the LiDAR. At a later point in time, the attackers can conduct a replay attack by sending the recorded signals back to the LiDAR in order to cause the LiDAR to map non-existent objects
2. **Relay Attack**: Replay attacks can be extended in order to carry out a relay attack, which can disrupt the LiDAR's ability to accurately gauge the distances of nearby objects. During a relay attack, attackers receive LiDAR signals and transmit those signals to a receiver in a different location. The second receiver can then send the signals back to the LiDAR leading to an incorrect map location of nearby objects.
3. **Blinding Attack**: Blinding attacks are carried out by injecting a light source of the same wavelength as the LiDAR's pulses. The external light source will cause the LiDAR to experience saturation, which would effectively deny its services to the vehicle. LiDARs transmit infrared light pulses, so the attacker's light source must have to be infrared light. As infrared light is not detectable by the human eye, blinding attacks can occur without the vehicle's occupants' knowledge.
4. **Spoofing Attack**: Spoofing attacks cause LiDARs to detect non-existent objects. One can spoof a LiDAR system and cause it to undercalculate or sometimes even overcalculate distances.
5. **Jamming Attack**: This type of Attack directly emits light back at the scanner unit on the vehicle that uses the same frequency band as the laser.
6. **Denial-of-Service Attack**: Attackers can conduct Denial-of-Service (DoS) attacks on LiDARS by injecting an enormous number of fake objects created using jamming or spoofing. If the number of injected objects is larger than the maximum number of objects that a LiDAR can track, then the system becomes unstable.

Countermeasures for attacks on LiDAR systems: [46]

1. **Redundancy**: It is possible to use different types of wavelengths for Lidar vision. Although some wavelengths have drawbacks in terms of range, combining multiple wavelength Lidar makes it harder for the attacker to both signals at the same time. The costs for the required hardware will exceed the budget for the attacker model considered for this work. Another way of adding redundancy would be to use Vehicle-to-Vehicle (V2V) communication. If an attacker mounts a front/side/rear or roadside attack, the chances are it will only affect a single vehicle. If other AVs share their measurements, the attacked AV could compare the measurements with what other vehicles observe. It is assumed that a

comparison with other vehicles requires the vehicle to detect tampering, ask other vehicles to share their data, validate the data (e.g. an attacker may intentionally share incorrect data), compare it, and decide on an action. This process may cost too much time and utilize an expensive link between vehicles.

2. **Random Probing**: Another option that takes less effort to implement, is to intentionally skip certain pulses. When a pulse is skipped, it introduces an effect that is similar to varying the scan speed. If the Lidar skips a pulse, it can still listen for incoming pulses. If it notices a response, this may indicate an attacker is tampering. It depends on the application whether this is acceptable or not. However, at a scan frequency of 50 Hz, missing a few pulses will probably not have much effect on the resolution, especially at close range.
3. **Probe multiple times**: This countermeasure is only effective against random jamming. If an attacker is not in sync with the pulse signal generated by the Lidar, counterfeit pulses will appear at random intervals in the attack window. For instance, if the Lidar measures three times at the same position and measures three different distances (e.g. 40 m, 10 m, 150 m), this measurement is likely to be invalid.
4. **Increase accuracy and precision of transceiver**: A denial-of-service attack can be mounted on the Lidar by jamming or spoofing a large number of objects. Such an attack can be prevented by increasing the maximum number of objects a sensor module can detect.

7.9.2. Camera

Attacks on Camera:

1. **Blinding Attack**: Blinding attacks disable the functionality of the vehicle's camera sensors. A strong laser beam focused on the camera leads to higher tonal values, and the attacker exploits this phenomenon to conceal the camera feed, causing complete blindness to the vehicular sensory inputs. This may lead to vehicle distortion or even emergency braking.
2. **Auto-Control Attack**: Auto-control attacks also target camera sensors. Attackers continually direct bursts of light at the camera in an attempt to manipulate the auto controls so that the image cannot stabilize. This type of attack can generally only be implemented in what terms as a front/rear/side attack.

Countermeasures for attacks on Camera modules: [46]

1. **Redundancy**: By using multiple cameras it is harder for an attacker to blind all of the cameras at the same time. By introducing multiple cameras that perceive the same image (or at least overlap), the attacker has to put more effort into the attack to blind both cameras at the same time. This does require more space to fit the cameras and a pair of cameras need to be carefully calibrated so the overlapping image is not misaligned. The software should blend the separate images together. As long as the cameras have a static position with respect to each other, the parameters for blending the images together have to be set up only once
2. **Optics and materials**: Integrating a removable near-infrared-cut filter, a technique that is available to security cameras, can filter near-infrared light on request. The filter can be applied by switching an electromagnet. During daytime, the filter is applied to yield a better image. During night time the filter is removed to make use of infrared light for night vision. When the filter is applied, it will also block infrared light sources, hence this countermeasure is only effective during the daytime. To improve this countermeasure, the filter could also be applied when the camera decides it is needed, for instance when it is jammed (see next countermeasure), or when the auto controls cannot be optimized for the bright lighting conditions anymore.
3. **Spectral analysis**: A prism will decompose an incoming light beam into several beams per color because the refraction index is wavelength-dependent. By positioning an image sensor such as a Charge-Coupled Device (CCD) or Complementary Metal Oxide Semiconductor (CMOS) to receive this beam, the wavelengths and intensity can be measured. The light sources used in this work have a characteristic wavelength.

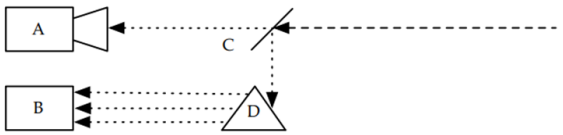


Figure 46: Spectral Analysis [15]

For instance, the red laser has a steep peak at 650 nm. Since environmental light has an influence on the amount of light needed to

blind a camera, it is assumed that it needs at least the same intensity to be visible to the camera. Therefore, if the light spectrum is observed over time, it can be possible to detect whether an attacker is pointing a light source into a camera. To perceive a camera image and do a spectral analysis at the same time, the incoming light should be split with a beam splitter. A beam splitter is an optical device that splits an incoming beam. One part will pass through, while the other part will reflect off. Both beams will have a lower intensity, which may be undesired for the camera system.

4. **Image channel separation**: The red channel is most sensitive to near-infrared light, so if that channel is jammed, it can use the other channel to filter the near-infrared light. The same approach works by extending it to a multi-band channel camera. This countermeasure can be implemented in software efficiently and thus requires no extra space in a camera. An attacker may decide to use several light sources to rapidly change color. This renders the above countermeasure less effective, since all channels may be overexposed.

8. Thermal Management

The progress of integrated circuit technology makes the integration of electronic chips continue to increase, i.e. the number of devices per unit area continues to grow. The challenges that arise due to the complex and interconnected nature of 2.5D and 3D-ICs, in which multiple dies are stacked on top of each other and connected using TSVs and microbumps include but are not limited to **heat transfer**, **electromigration**, **stress and strain**, and **thermal expansion**.[26]

Of these, heat transfer to the ambient is probably the main problem raised by 3D-IC design. The working reliability of electronic chips is closely related to the operating temperature and the maximum operating temperature of electronic chips, i.e. **the peak temperature, is usually between 85 °C and 120 °C** [31], [3] . It has also been verified that **the operational reliability of electronic chips decreases by 50% for every 10°C increase** in operating temperature [51]. Mithal et al.[39] found that for **every 1 °C decrease in the temperature of electronic chips, the failure rate of the electronic chips would decrease by about 4%**. Electronic chips may weather the immediate heat storm, but the cost is swift and silent. Their lifespan shrivels, demanding costly replacements and elaborate cooling systems. High temperatures don't just threaten immediate failure; they silently steal the chips' future, burdening both performance and budgets. Thermal management isn't a luxury; it's a lifeline for both chip health and financial sustainability. At present, the solutions mainly include natural cooling, forced convection, frequency conversion technology, and energy storage technology.[43],[50] [26]

Due to the high density of transistors and other components, most of the heat is trapped in the system, which contributes to the increased temperature. This phenomenon is called **self-heating**. **Joule heating** resulting from long connections is another major problem area that contributes to overall temperature increase. These heat sources must be monitored and analyzed when designing 2.5D and 3D-ICs to ensure reliable performance.

In all, there are mainly 2.5-D and 3-D packaging technologies for chiplet heterogeneous integration.

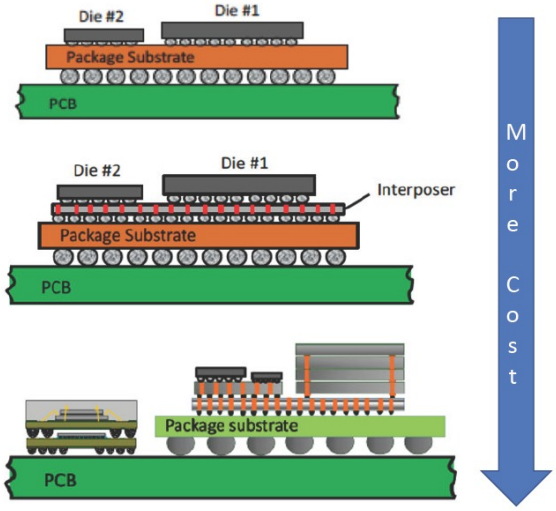


Figure 47: The increasing cost of thermal simulations while transitioning from 2D to 2.5D to 3D[53]

Compared with the 3-D packaging technology, the 2.5-D packaging avoids the thermal issue caused by the high heat flux of 3-D stacked dies. The 2.5-D packaging uses an interposer with TSVs and redistribution layers (RDLs). Multiple chips are stacked on the interposer side-by-side via microbumps (μ -bumps) to connect to each other or the substrate. However, the 2.5-D packaging technology also has some challenges in the aspect of thermal design, such as **thermal interface material (TIM) selection**, **bonding approach**, and **thermal crosstalk** between chiplets. Especially for chiplets with higher thermal design power (TDP), the thermal design of the packaging is challenging.

Proper thermal modeling and analysis are crucial to the successful thermal design of the 2.5-D packaging.[2]

8.1. *Chiplet-based Packaging With a 2.5-D Interposer*

8.1.1. *Traditional Method (Heat Flux below $10W/cm^2$)*

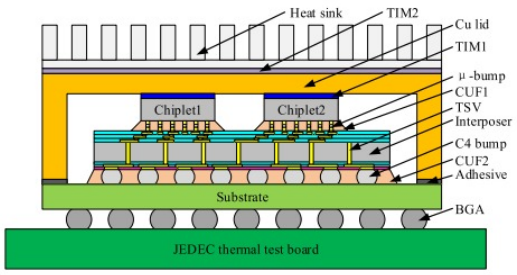


Figure 48: Schematic of chiplet-based packaging with a 2.5D interposer[75]

Fig. 1 shows the schematic of a chiplet-based packaging with a 2.5-D TSV interposer. Two chiplets are mounted on the interposer side-by-side through μ -bumps. A capillary underfill1 (CUF1) is filled in between the chiplets and the interposer. The interposer is bonded on an organic substrate via the controlled collapse chip connection (C4) bumps. A CUF2 is filled in between the interposer and the substrate. The substrate is assembled on a JEDEC thermal test board through BGA solder joints. A Cu lid is attached to the top surfaces of the chiplets and periphery of the interposer through the Thermal Interposer Material(TIM1) and adhesive, respectively. To further improve the heat dissipation capability from the top side, a heat sink is attached to the top surface of the lid through a layer of TIM2.[2] This is the traditional method of cooling, with the heat path flowing through the semiconductor dies itself and through the lid in a flip chip ball grid array (FCBGA) package[75]. The heat sink can also use active cooling and be coupled with a fan to produce a **Heat Sink Fan(HSF) Assembly**.

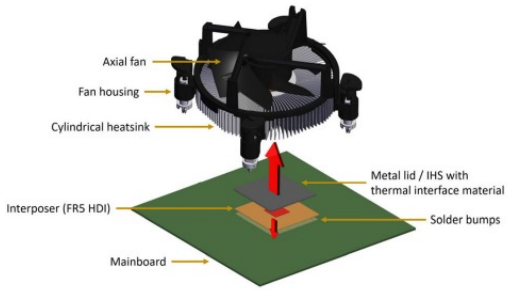


Figure 49: A typical FCPGA structure with the red arrows denoting the primary and secondary paths of heat transfer[11]

$$\frac{\partial \rho}{\partial x} + \nabla \cdot (\rho u) = 0 \tag{1}$$

The momentum equation is expressed as follows:

$$\frac{\partial (\rho u)}{\partial t} + \rho (u \cdot \nabla u) = -\nabla p + \nabla \cdot \tau + \rho g \tag{2}$$

where ρ is the density, t is the time, u is the velocity, τ is the stress tensor, p is the static pressure, g is the body acceleration, and ρg is the gravitational body force.

The energy equation for a liquid region can be written as

$$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot (\rho h u) = \nabla \cdot [(k + k_t) \nabla T] + S_h \tag{3}$$

where h is the enthalpy, k is the molecular conductivity, k_t is the conductivity due to turbulent transport, and S_h is the volumetric heat sources, while for solid regions, a simple conduction equation is solved. The equation includes the heat flux due to conduction and volumetric heat sources within the solid

$$\frac{\partial (\rho u)}{\partial t} = \nabla \cdot (k \nabla T) + S_h \tag{4}$$

where k is the conductivity, T is the temperature, and S_h is the volumetric heat source .

I. THERMAL MODELING THEORETICAL BACKGROUND

1. Navier–Stokes Equations

When conducting thermal modeling and simulations via CFD softwares like ANSYS Icepak, Comsol, Flotherm etc., the mass conservation, momentum, and energy conservation equations are solved.[27] The corresponding equations are:

The mass conservation equation can be written as follows:

II. THERMAL MODELLING

When conducting the thermal modeling of the chiplet-based 2.5-D packaging via ANSYS Icepak, the assumptions and simplifications were made by our team as follows:

- The control volume where the simulation occurs is assumed to be filled with static air.
- Gravity acts in the downward direction, i.e. into the thermal test board.

- The TIM layers and the microbumps are approximated to cuboidal packages.
- Radiation becomes particularly important as the miniaturization of electronic components and the increase in power densities have made thermal management a critical aspect of microelectronic design and is not ignored.
- The cooling fan used is ADDA 8381HB AT1, which has a rated power of 2.88 W.

Object	Material	Thermal conductivity
Chiplet	GaAs	46 W/mK
Interposer	Silicon	148 W/mK
TIM	SiO_2	1.5 W/mK
Lid	Copper	400 W/mK
Substrate	Epoxy	0.3 W/mK

III. OBSERVATION AND RESULT

The simulation results were in concordance with the expected values. The Heat Sink Fan Assembly (HSF) is capable of cooling power outputs of magnitude close to 90W. Further modifications, in the TIM layer and other innovations (as discussed in the following sections), can bring about better results.

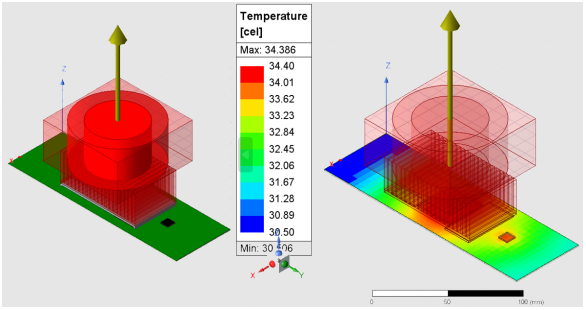


Figure 50: Observation of the simulation performed by our team in ANSYS with 20 rectangular fins at a height of 28mm

For the initial power outputs of the 3 chiplets (0.33, 1.5, 7.5, 12 W, respectively) the peak temperature obtained was 33.689 C and is depicted in Figure 50. For such a small magnitude of power output, the simulation can be conducted without the blower/fan.

The fin shapes too were varied, keeping the heights constant. The shapes taken into consideration were plate, cylindrical, columnar, square and staggered(pin) fins respectively. The heat dissipation is directly proportional to the surface area and as expected the results depicted in fig. show that the staggered fins provide the best cooling performance. Similarly, a study of the fin heights was

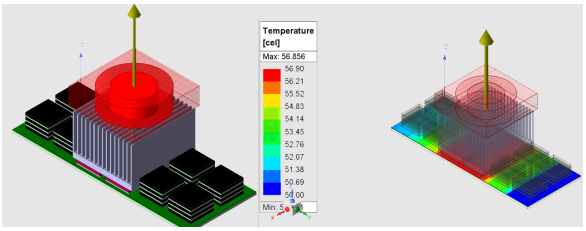


Figure 51: Observation of the simulation performed in ANSYS of GPU by our team

conducted and the findings are tabulated in Figure 51, with the taller fins providing better heat dissipation.

The cooling system suggested, i.e. the traditional cooling with a fan/blower is appropriate for powers of 50-60 W, is an economically viable approach and can be further improved with appropriate modifications.

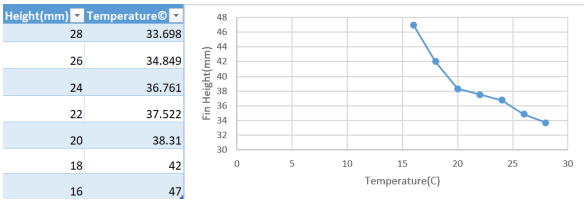


Figure 52: Comparison of fin height with temperature for a rectangular fin

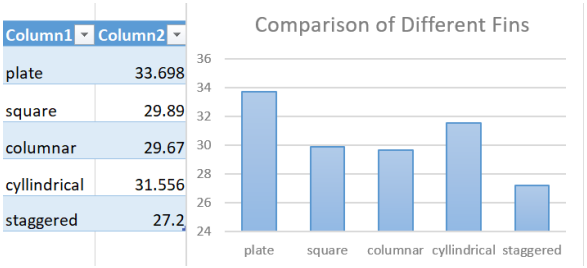


Figure 53: Comparison of fin shapes with temperature

8.2. Obstacles faced while packing in 2.5D

Obstacles faced :

The core principle of chiplet technology revolves around the ideal that there should not be a monopoly in the market and that a heterogenous integration of chiplets must be possible and ideal. This leads to the procurement and integration of various chiplets of varying geometrical specification and size.

- The primary problem faced in thermal management is the varying height of the dies placed on the common interposer. In this case, the TIM layers with different thicknesses should

be applied to realize thermal contact between the die and the common metal lid. The TIM layer has the worst heat conductance in the traditional heat flow path and hence the thinnest possible layer would have to be chosen. [11]

- However, another problem also arises. If microchannels are fabricated in flip-chip devices, the fluid supply is difficult to manage. Either a double-walled lid is applied, or inlets/outlets are opened through the metal lid traditional heatsink, and fan structures cannot be used anymore. [11]

- Unlike their 2D counterparts, in 3D IC chip stack, individual chip faces within a 3D stack are not readily accessible for conventional heat sink mounting, necessitating alternative approaches. Mounting heat sinks on the ends of the stack, while seemingly feasible, presents its own set of complications. The heat generated by the interior chips must traverse the entire stack length to reach the heat sink, significantly increasing the thermal resistance. This extended heat flow path impedes efficient heat dissipation and can lead to overheating, potentially compromising chip performance and reliability. [32]

- Furthermore, the presence of multiple heat sources within the stacked chips exacerbates the thermal challenge. The interaction of these heat sources can create localized areas with concentrated heat, referred to as "hot spots." These hot spots can exceed the allowable junction temperatures of the individual chips, leading to thermal runaway, performance degradation, and ultimately, chip failure.[32]

- Introducing interlayer cooling with microchannels and introducing fins in the coolant flow paths extend the thermal dissipation capability of a 3D stack; however this is often accompanied with taller microchannels that lead to longer lengths of through-silicon-vias (TSVs). Placement of TSVs, microchannels, walls and fins present conflicting design requirements. [32]

8.3. Thermal Interface Material (TIM)

As mentioned above, minimizing the resistance at the interfaces in the packaging of microelectronics is a primary objective to improve heat transfer. The presence of roughness and waviness at the interface of the solids creates a non-conformational type of contact interface which leads to a thermal contact resistance (TCR). This is caused due to the surface irregularities with voids and holes which are then occupied by the surrounding medium, i.e. air, which in turn has a very low thermal conductivity (0.028 W/mK).

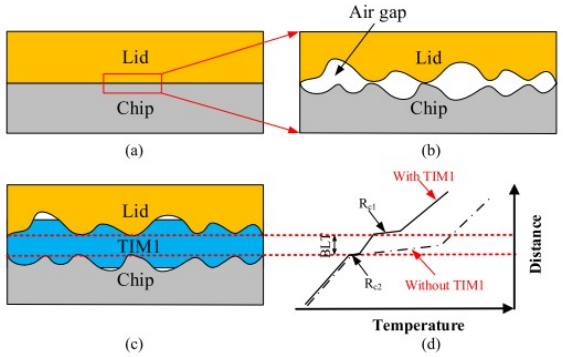


Figure 54: Schematic of thermal interfacial resistance between the chip and the lid. (a) Direct contact of lid and chip. (b) Zoomed-in view of the contact interface. (c) Indirect contact of lid and chip with TIM1. (d) Effect of TIM on the temperature change of the chip and lid.[75]

One way to decrease this contact resistance is by increasing the surface area of contact. This can be brought about by the application of pressure. Narayana and Narayan [41] in their study on TCR between similar and dissimilar materials and Devananda and Prabhu [42] for Cu–Cu interfaces both concluded the same independently. However, it is hardly an ideal solution as we are dealing with the domain of microelectronics and stringent load constraints exist.

An effective method to reduce the TCR is by incorporating a Thermal Interface Material (TIM)

with a high thermal conductivity. The factors affecting the total TCR then would be thermal conductivity, bond line thickness, clamping pressure, surface roughness, compressive modulus, and flatness of the surface. [32]

8.4. First Modification (Heat flux between 10 and 15 W/cm^2)

One category of TIMs that is not mentioned above includes the Ceramics group of materials. If the system under development is facing difficulty in thermal aspects, then to improve heat dissipation high thermal conductivity insulators can be used instead of the thermally resistive interlayer dielectrics. Isotropic, high thermal conductivity ceramic insulators such as AlN are superior for heat removal for intermediate layers of the 3D stack, while strongly anisotropic materials such as hBN are effective at reducing peak temperatures due to

Table 2: A review and compilation of the various kinds of TIM used currently in industry along with their advantages and disadvantages

Sl no.	**Thermal interface materials (TIM)**	**Typical examples of commercially available TIMs**	**Advantages**	**Disadvantages**
1	Thermal grease/paste: polymer (silicone-based, PEG-based), sodium silicate-based thermal greases	Artic silver, ShinEtsu G751, ShinEtsu G765, ShinEtsu G750, BerquistTIC-7500, G641,DC 340,P12,Eupec, Unial Rhodorsil 340, etc	Low cost, easy to apply, conformable, wide operating temperature range	Low thermal conductivity, pump-out, drying out, messy
2	LMA	Eutectic(Bi–Pb–Sn–In–Cd), Indium–Bismuth–Tin alloy (Ga–In), (Ga, In–Bi–Sn), (17Sn26In57Bi), (17Sn51In32Bi), (51 In, 32.5Bi, 16.5Sn), (100 Ga), (75.5Ga, 24.5In), ($Ga_{62.5}In_{21.5}Sn_{16}$), (GaIn10), ($GaIn_{20.5}Sn_{13.5}$)	High thermal conductivity, low melting point	Toxic low mechanical strength, expensive
3	Metallic foils	Copper foil, aluminum foil, lead foil, Gold foil, Tinfoil, Indium foil	High thermal conductivity, reusable	High cost, difficult to apply, limited conformability
4	PCM	ThemaxHF-60110-BT, Chromerics T725 Bergquist HiFlow, PowerStrate 60(AG), Orcus inc. FSF 52	High thermal conductivity during phase change, long life	High cost, limited availability, temperature dependence
5	Gels	Vinyl terminated silicone oil	High thermal conductivity (some), conformable	Low thermal conductivity (some), can leak, messy

Table 3: A comparison of the thermal conductivities of SiO2, AlN and hBN

Sl.No	Material	Thermal Conductivity (W/mK)
1	Silicon Dioxide (SiO2)	1.4 [16]
2	Aluminum Nitride (AlN)	250 [17]
3	Hexagonal Boron Nitride (hBN)	400 [18]

localized hot spots and also for reducing temperatures in applications that require thermal decoupling between dies.[33]

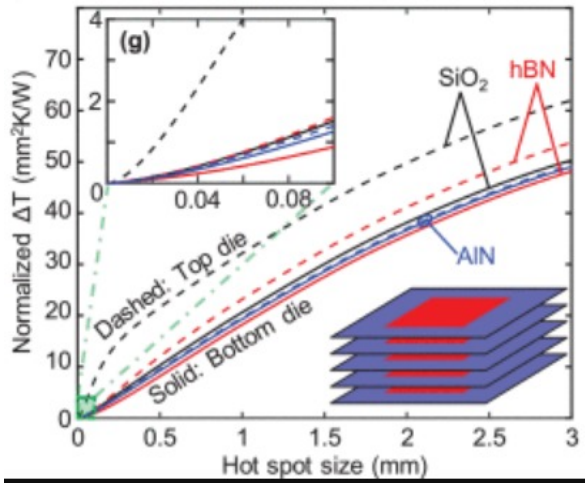


Figure 55: Normalized peak δ T in the top (dashed) and bottom dies (solid lines) as a function of hot spot size, for a single column of hot spots comparing using different TIM materials like AlN, SiO2 and hBN[33]

This modification alone can reduce hotspot temperatures by about 20 percent. hBN is often preferred for its higher thermal conductivity and softer, more machinable nature while AlN is used when mechanical strength is crucial, despite its lower thermal conductivity. [11]

HOPC (High-performance organic polymer composite) shows potential as a next-generation thermal interface material (TIM) due to its high thermal conductivity, low thermal resistance, and good mechanical properties. HOPC/Cu NPs are composite materials consisting of highly conductive copper nanoparticles embedded within a host matrix of thermally and electrically insulating polymers or ceramics. This unique combination leverages the exceptional thermal conductivity of Cu NPs while mitigating electrical conductivity risks, making them ideal for electronics applications. They boast superior heat transfer (10-50+ W/mK) compared to traditional TIMs, thanks to Cu NPs' high conductivity and low interfacial resistance. Their conformability and electrical insulation make them ideal for efficient heat dissipation in demanding HPC applications. However, agglomeration of NPs can significantly reduce their thermal performance, requiring advanced dispersion techniques. Additionally, current synthesis methods can be costly, hindering large-scale implementation. Finally, ensuring long-term stability under demanding HPC conditions remains an ongoing challenge.

The thermal management landscape is undergoing a paradigm shift with the emergence of carbon nanotubes (CNTs) as next-generation thermal interface materials (TIMs). Their exceptional thermal conductivity, exceeding 2000 W/mK, eclipses traditional materials like solder and thermal grease by orders of magnitude, offering unparalleled heat dissipation capabilities. This translates to significant advancements in thermal management across various industries, from high-performance electronics and LEDs to spacecraft and automotive applications.

However, the widespread adoption of CNT-based TIMs is currently hindered by several key challenges. The primary obstacle remains cost, as the intricate synthesis processes involved in CNT production necessitate further optimization for economic viability. Additionally, achieving uniform dispersion and optimal alignment of CNTs within the TIM matrix presents a significant technical challenge, as their inherent intermolecular forces can lead to aggregation and hinder their thermal performance. Furthermore, ensuring efficient interfacial contact between CNTs and the heat source/sink requires careful consideration of surface properties and interface engineering techniques.

8.5. Second Modification (Heat flux between 15 and 25 W/cm^2)

Wu et al. [71] propose the application of PCM inside the chip package instead of the conventional thermal interface material (TIM) layers. In the proposed model, the space between the chips and the underside of the lid is compartmentalized and where the lid side acts as a condenser and the other side acts as a heat source cum evaporator. A suitable liquid is filled in, to a particular extent whose phase change properties allow the transfer of heat by successive evaporation from the bottom and condensation on the bottom of the lid in a cyclic manner.

Compared to traditional package sample using TIM, this modification reduces the thermal resistance from ∼0.46 to ∼0.1°C/W under a 3-W power input, for a thickness of 1.6 mm (about 0.06 in), and with appreciable space for improvement.

8.6. Third Modification (power range at which it operates)

Instead of the traditional HSF assembly, Hoang et al. [25] designed a 3-D-printed cold plate characterized with water coolant. The printed metal fin structures were strong enough to undergo pressure from the fluid flow even at high flow rates and small fin structures. It was observed that for the coolant inlet temperature 25°C and aluminum cold plate, the junction temperature was kept below 63.2°C at an input power of 350 W and the

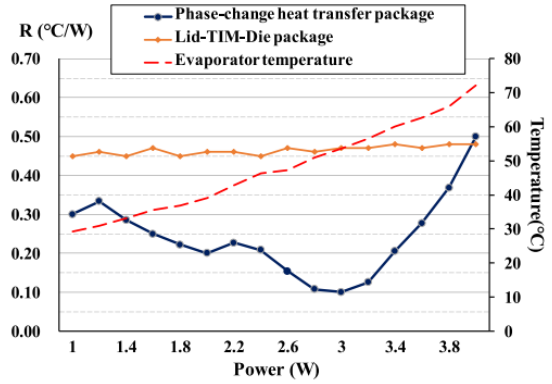


Figure 56: Thermal resistance of the junction case of two package samples changing with temperature. The dashed line is the evaporator temperature of the phase-change heat-transfer sample under a steady state.[71]

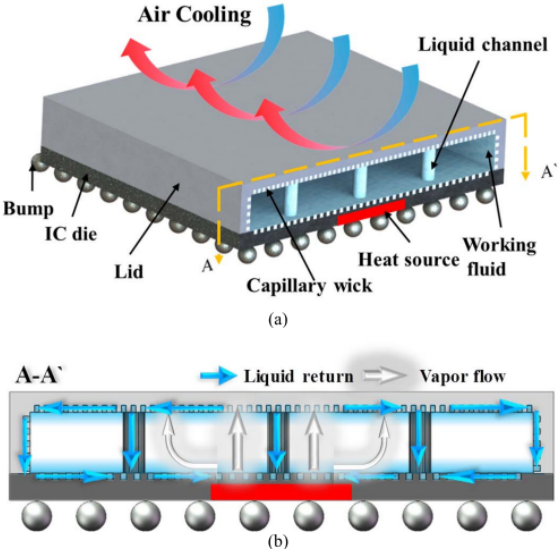


Figure 57: a) Schematic and structure diagram of the packaging method. (b) Flow of vapor and liquid in the chamber[71]

pressure drop did not exceed 23 Kpa at a flow rate of 0.75l/min. [11] We have only been considering the primary heat flow path where the heat flows from the chiplets at the bottom through the TIM layers to the lid, heat sink and then to the ambient. However, there is also a secondary heat flow path that transfers a significant amount of the heat towards the package substrate and the mainboard, which in turn, can act as a heat-emitting structure (a passive cooling structure). Matsumoto et al. [36] devised a cooling mechanism, assuming a substrate that consists of organic dielectric materials and copper in a 3-D package for high-power applications. One possible way is to enhance the role or create a secondary heat flow path by applying microchannel-based cooling.

8.7. Fourth Modification (Typically 30 W/cm^2 and above)

Microfluidic interposers, with embedded microchannels for direct liquid cooling, offer a promising solution for heat dissipation. To maximize the effectiveness of this approach, the placement of microchannels within the interposer is critical.

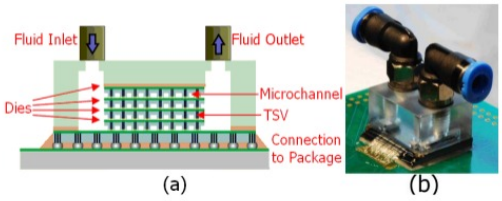


Figure 58: (a)3D stacked IC with interlayer microchannel cooling,(b) A 3D IC test vehicle,with lateral wire-bonds for electrical IO, fabricated with fluid manifold mounted on a printed circuit board.

Locating microchannels close to the heat source, typically the chiplet junction, minimizes the thermal resistance between the junction and the circulating coolant. This minimizes the reliance on secondary heat flow paths, such as conduction through the interposer to the motherboard and subsequent natural convection to the ambient.

Natural convection, with its low heat transfer coefficient (5-6 W/m2*K), results in significant thermal resistance and elevated chip temperatures. By creating an additional, low-resistance pathway through the microchannels, the overall thermal resistance is significantly decreased. The effective-

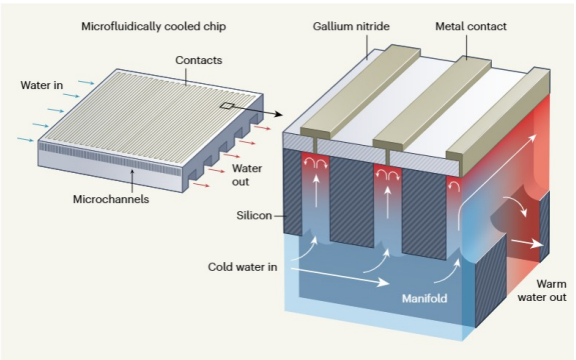


Figure 59: An integral cooling system for microchips using microchannels. Van Erp et Al[65]

ness of this microfluidic cooling approach is directly linked to the mass or volumetric flow rate of the coolant. Higher flow rates enhance heat

transfer efficiency and further reduce, leading to improved thermal performance and lower chiplet temperatures.

Keiji Matsumoto et. Al[25]], the fabrication of microchannels and microstructures as heat sinks was discussed and analyzed in detail. These microstructures were fabricated directly in the backside of the active silicon semiconductor dies by applying CMOS-compatible wet chemical etching process steps [37]. The proposed microchannel cooling system directly integrates the microchannels into the substrate, enabling direct contact between the heat source and the cooling fluid. This eliminates the need for numerous TIM layers and minimizes the heat conduction path length, resulting in improved heat transfer performance. Furthermore, the absence of alkali metal based TIMs mitigates potential reliability issues associated with threshold voltage instability.

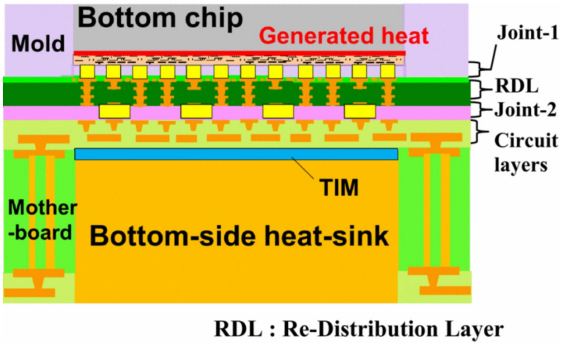


Figure 60: A schematic view of additionally using the secondary heat pathway (through the substrate)[25]

The lengths of these channels are also longer than in the case of in-chip realization as the length of a typical interposer is longer than the chiplet lengths, resulting in a higher pressure drop and increased hydrodynamic resistance. This justifies the need for a wider, deeper and larger number of microchannels. These microchannels are required to be parallel to ensure a uniform heat transfer coefficient throughout the passage of the coolant. [11]

The analytical procedure for the representation of such a model involves the modelling of the solid part(interposer) as an RC- circuit. The conducting modules of the solid are modelled as resistors and the heat holding modules as capacitors. Near the inlets, the local Nusselt number and, hence, the heat transfer coefficient are significantly higher[11]. The heat transfer coefficient can be calculated as follows:

$$h = \frac{k_f \cdot Nu}{D_H} \tag{5}$$

is the thermal conductivity of the fluid, and DH is the hydraulic diameter [29]. The DH hydraulic diameter can be determined for a square-based column as follows:

$$D_H = \frac{2 \cdot a \cdot b}{(a+b)} \tag{6}$$

and the Nusselt Number

$$N_u = \frac{0.065 \cdot (D_H/L) \cdot Re \cdot Pr}{1 + 0.04 \cdot 1 + 0.04 \cdot [(D_H/L) \cdot Re \cdot P)] \cdot \frac{2}{3}} \tag{7}$$

Further, the Energy Conservation equation is simplified, rearranged and solved to obtain the resistances.

$$R_{th\_uch} = \frac{1}{\frac{d}{dt}(c_p \cdot \left(1 - e^{-hA\frac{d(c_p)}{dt}}\right))} \tag{8}$$

The parallel microchannels are treated as parallel channels of resistors and the equivalent resistance is calculated accordingly.

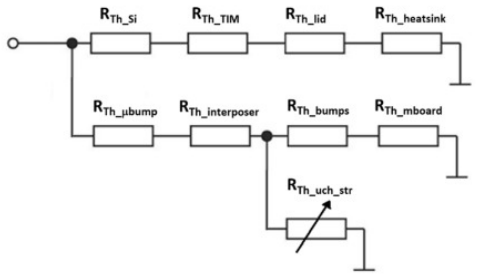


Figure 61: Lumped element model of the heat map[11]

In [37], a novel methodology was presented to determine the Rth partial thermal resistance and cross-verify the analytically calculated and simulated values.This novel methodology is based on Joint Electron Device Engineering Council (JEDEC) JESD51 1-14 standard test method [40], an analytical model was built, and a hydrodynamic–thermal modeling tool was developed in ANSI C. The simulation results according to the research paper are as follows: The coolant used was deionized water and the fan used was a Sanyo Denki 40 × 40 × 15 mm nonlinear axial fan.

The relationship between flow rate and temperature is non-linear; a drop below a critical flow rate

JUNCTION TEMPERATURES OVER THE CHIPLETS AT THE GIVEN FLOW RATE FOR THE FIRST SCENARIO (OTL—OVER-THE-LIMIT: 300 °C+)

Volumetric flow rate [CCM]	CPU junction temperature min – max [°C]	GPU junction temperature min – max [°C]	Memory junction temperature min – max [°C]
0	OTL	OTL	OTL
10	107 – 150	56.6 – 123	42.5 – 64.2
100	49.2 – 60.1	33.5 – 46.2	24.6 – 30.6

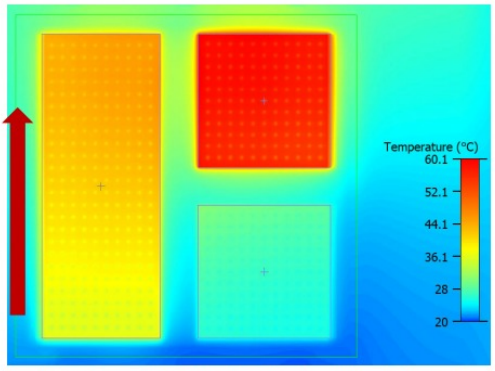


Fig. 5. Temperature distribution along the dies in the first scenario. (Red arrow indicates the direction of the fluid flow.)

Figure 62: Temperature distribution along the dies in the first scenario. (Red arrow indicates the direction of the fluid flow.[11])

results in a disproportionate increase in temperature.

The innovative chip-level microscale cooling solution, originally designed for backside integration, can also shine in integrated structures by residing within the silicon interposer layer. This alternative placement proves surprisingly effective, especially for chiplets crafted with diverse manufacturing techniques and generating significant heat. This underlines the versatility and potential of the novel cooling approach presented in this research.

The obstacles faced include but are not limited to:

- Complexity: Microfluidic channels require precise microfabrication techniques like photolithography and etching, which can be expensive and time-consuming.
- Leakage and clogging: Tiny channels are susceptible to leaks and blockages due to microparticles or chemical reactions within the coolant. Maintaining channel integrity is crucial.
- Flow control: Achieving uniform and controlled flow across complex microchannel networks can be challenging, potentially leading to uneven heat dissipation.
- Material compatibility: The coolant and microchannel materials must be compatible to avoid chemical reactions, corrosion, or degradation.
- Pressure drop: High flow rates in microchannels can lead to significant pressure drops, requiring powerful pumps and potentially affecting system efficiency

*8.8. **Chiplet-based Packaging with a 3-D Interposer***

Due to their inherent thermal limitations arising from stacked configurations and limited surface areas, 3D chiplet architectures demand novel cooling approaches. A diverse array of methods has emerged, each offering distinct performance characteristics and implementation considerations. Here's an overview of some prominent methods:

*8.8.1. **Microfluidic cooling***

The cooling solution previously explored deeply in the case of 2.5D cooling can be extrapolated into 3D cooling too. The same mathematical model can be employed with a considerable amount of rigorous calculations, to simulate the required thermal conditions. The benefits of this approach are undeniable. Microfluidic cooling boasts exceptional heat transfer, minimizing thermal resistance and enabling unparalleled cooling efficiency. Targeted cooling is another feather in its cap, as channels can be strategically placed near specific hot spots, delivering precise heat dissipation where it's needed most. Additionally, its inherent scalability makes it adaptable to diverse chip layouts and varying heat generation profiles. However, like any formidable warrior, microfluidic cooling demands its dues. The intricate fabrication process, a testament to its technological prowess, also translates to a hefty price tag. Costs can soar above $100 per chip, making it a significant investment. Leakage risks, ever present in such micro-engineered systems, necessitate meticulous material selection and rigorous maintenance protocols. Finally, space constraints within the chiplet package can pose challenges for integrating complex designs.[54]

*8.8.2. **Jet Impingement cooling***

This technique, wielding high-velocity jets of liquid or gas, directly attacks concentrated heat zones with laser-like precision. Unlike other cooling methods that might struggle with uneven heat distribution, jet impingement excels in its targeted approach, offering distinct advantages.

- One of its greatest strengths lies in its precise heat dissipation capabilities. This targeted approach effectively addresses uneven heat distributions, leaving other areas untouched

and significantly outperforming methods that might leave thermal hotspots smoldering.

- Furthermore, jet impingement boasts a compact design, requiring minimal space within the chiplet package. This compact footprint makes it a valuable asset for small-form-factor devices where space is a precious commodity. Unlike sprawling cooling systems that can consume valuable real estate, jet impingement offers a space-saving solution without compromising on performance.

- Cost-effectiveness is another compelling advantage of jet impingement cooling.

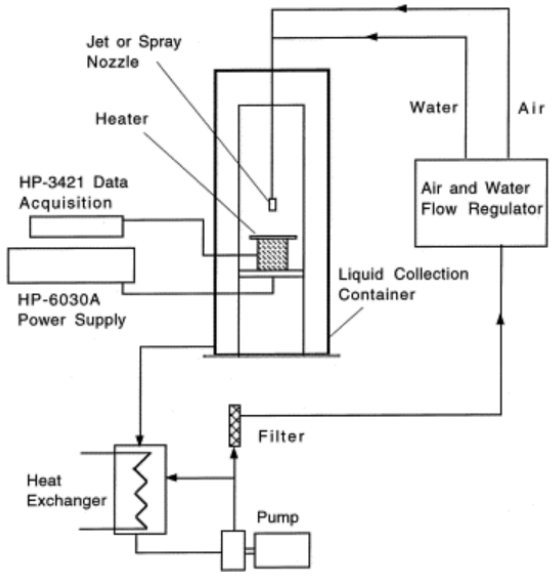


Figure 63: Schematic of jet/spray heat transfer experimental apparatus[44]

Compared to advanced techniques like microfluidics, jet impingement presents a more budget-conscious option. With fabrication costs estimated around $50 per chip, it offers a viable solution for applications where cost constraints are a factor.[30]

However, like any potent weapon, jet impingement demands careful consideration of its limitations. One such hurdle is the significant noise pollution generated by the high-velocity jets. which can be a major obstacle for noise-sensitive environments. Additional soundproofing or mitigation measures might be necessary to address this challenge.

Another potential drawback is the risk of surface erosion. Uncontrolled fluid forces can act like a microscopic sandblaster, gradually eroding the chiplet surface over time. Careful design and optimization of jet parameters are crucial to ensure long-term chip health and prevent surface damage.

Finally, it's important to note that while jet impingement excels at localized cooling, it may not be sufficient for applications with high overall heat generation. [30] The high transport capabilities of spray cooling is demonstrated in the experiments of Hodgson et al. [44], who showed that the spray cooling heat transfer (at the stagnation point of a cylinder) was as much as 30 times that of the single-component air flow. The results of Graham and Ramadhyani reveal surface heat fluxes as high as 0.6 MW/m2 with surface temperatures maintained below 70°C and 80°C with methanol and water sprays, respectively. [54], [23]

8.9. ***Spray Cooling***

Unlike the forceful jets of jet impingement or the complex networks of microfluidics, spray cooling utilizes a fine mist of liquid to achieve thermal dissipation, offering distinct advantages[44]:

- Uniform and Gentle Heat Transfer: The mist directly contacts the chiplet surface, facilitating efficient heat absorption. Subsequent evaporation releases the absorbed heat, resulting in a uniform and gentle cooling effect, minimizing the risk of concentrated hot spots.

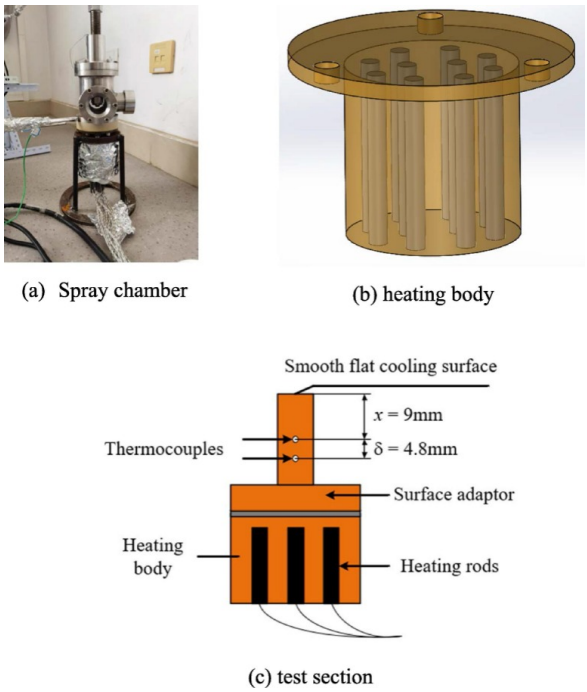
- Scalability and Adaptability: The inherent flexibility of the spray approach allows for adaptation to diverse chiplet layouts and heat generation profiles. Nozzles can be strategically positioned to target specific areas, and the mist's flow rate can be adjusted to suit varying heat loads.

- Low-Noise Operation: Compared to the noise generated by other techniques, spray cooling operates with minimal acoustic emissions. This quiet operation makes it ideal for applications requiring silence, such as scientific equipment or noise-sensitive environments.

- Cost-Effectiveness: While not the most affordable option, spray cooling remains a cost-effective solution compared to advanced techniques like microfluidics. Fabrication costs typically fall around $30 per chip, making it a viable option for a broad range of applications.[54]

However, like any technical solution, spray cooling presents certain limitations that require careful consideration:

- Lower Heat Transfer Capacity: Spray cooling's heat transfer efficiency is generally lower than other techniques. While effective for moderate heat loads, it may not be sufficient



(a) Spray chamber (b) heating body (c) test section

Figure 64: Spray chamber, heating body and test section)[44]

for applications generating significant overall heat.

- Water Consumption: The continuous misting process necessitates a constant supply of coolant, potentially leading to higher water consumption compared to other methods. This factor is crucial in situations where water scarcity or environmental considerations are paramount.
- Condensation Risk: Under specific conditions, the evaporative process can lead to condensation on the chiplet surface, posing potential risks for corrosion or electrical short circuits. Careful design and material selection are crucial to mitigate this risk.

References

[1] *A Peer-to-Peer Smart Food Delivery Platform Based on Smart Contract*, https://www.mdpi.com/2079-9292/11/12/1806.

[2] P. Aberl, S. Haas, and A. Vemuri, *How a Zone Architecture Paves the Way to a Fully Software -Defined Vehicle.*

[3] E. M. Alawadhi and C. H. Amon, "Pcm thermal control unit for portable electronic devices: experimental and numerical studies", IEEE Transactions on components and packaging technologies **26**, 116–125 (2003).

[4] N. N. Anandakumar, M. S. Rahman, M. M. M. Rahman, R. Kibria, U. Das, F. Farahmandi, F. Rahman, and M. M. Tehranipoor, "Rethinking watermark: providing proof of ip ownership in modern socs", Cryptology ePrint Archive (2022).

[5] *Annual Update on Interfaces 20230125$_D$−103$_P$osner*, https://chipletsummit.com/proceedings/.

[6] *Automotive Cybersecurity Starts with Chips IP*, https://www.synopsys.com/designware-ip/technical-bulletin/automotive-cybersecurity-starts-with-chips.html.

[7] *Bayer filter*, https://en.wikipedia.org/wiki/Bayer_filter.

[8] M. van den Beld, *How Zonal E/E Architectures with Ethernet Are Enabling Software-Defined Vehicles*, https://www.nxp.com/company/blog/how-zonal-e-e-architectures-with-ethernet-are-enabling-software-defined-vehicles:BL-HOW-ZONAL-EE-ARCHITECTURES.

[9] F. Bi and J. Yang, "Target detection system design and fpga implementation based on yolo v2 algorithm", in 2019 3rd international conference on imaging, signal processing and communication (icispc) (IEEE, 2019), pp. 10–14.

[10] A. Birnie, *TRANSITION TO ZONAL ARCHITECTURES: CHALLENGES AND NXP SOLUTIONS*, June 2021.

[11] G. Bognár, G. Takács, and P. G. Szabó, "A novel approach for cooling chiplets in heterogeneously integrated 2.5 d packages applying microchannel heatsink embedded in the interposer", IEEE Transactions on Components, Packaging and Manufacturing Technology (2023).

[12] *Chiplet Security Risks Underestimated*, https://semiengineering.com/chiplet-security-risks-underestimated/.

[13] *CNN — Introduction to Pooling Layer*, https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/.

[14] *Convolutional Neural Networks (CNN): Step 3 - Flattening*, https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening.

[15] *Cybersecurity Attacks in Vehicular Sensors*, https://www.osti.gov/servlets/purl/1763654.

[16] *Cybersecurity for Electric Vehicle Charging Infrastructure*, https://www.osti.gov/biblio/1877784.

[17] *Demosaicing*, https://slazebni.cs.illinois.edu/spring19/assignment0.html.

[18] *Developing High-performance Chiplet Systems at Low TCO and High Sustainability*, https://chipletsummit.com/proceeding_files/a0q5f0-00001WuE0/20230125_B-103_Farjad.PDF.

[19] *DisplayPort Intel® FPGA IP User Guide*, https://www.intel.com/content/www/us/en/docs/programmable/683273/22-1-20-0-1/about-this-ip.html.

[20] M. Eaker, *Zone architecture, Ethernet drive vehicle of the future*, https://e2e.ti.com/blogs_/b/behind_the_wheel/posts/zone-architecture-ethernet-drive-vehicle-of-the-future.

[21] *eMMC and UFS Inline Encryption in Android-Based Mobile Applications*, https://www.synopsys.com/designware-ip/technical-bulletin/emmc-and-ufs-inline-encryption-2017q4.html.

[22] *EMSA5-FS 32-bit Embedded RISC-V Functional Safety Processor*, https://www.cast-inc.com/sites/default/files/pdfs/2023-11/cast_emsa5-fs.pdf.

[23] K. Graham and S. Ramadhyani, "Experimental and theoretical studies of mist jet impingement cooling", (1996).

[24] *HexaMesh: Scaling to Hundreds of Chiplets with an Optimized Chiplet Arrangement*, https://arxiv.org/pdf/2211.13989.pdf.

[25] C. H. Hoang, A. Azizi, N. Fallahtafti, S. Rangarajan, V. Radmard, C. Arvin, K. Sikka, S. Schiffres, and B. Sammakia, "Design and thermal analysis of a 3-d printed impingement pin fin cold plate for heterogeneous integration application", IEEE Transactions on Components, Packaging and Manufacturing Technology **12**, 1091–1099 (2022).

[26] W. Hua, L. Zhang, and X. Zhang, "Research on passive cooling of electronic chips based on pcm: a review", Journal of Molecular Liquids **340**, 117183 (2021).

[27] *Icepak User's guide*.

[28] *IME-IP-340 Inline Memory Encryption Engine*, https://www.rambus.com/security/inline-memory-encryption/ime-ip-340/.

[29] *Innovations in Ethernet Encryption (802.1AE - MACsec) for Securing High Speed (1-100GE) WAN Deployments*, https://www.cisco.com/c/dam/en/us/td/docs/solutions/Enterprise/Security/MACsec/WP-High-Speed-WAN-Encrypt-MACsec.pdf.

[30] S. Jones-Jackson, R. Rodriguez, and A. Emadi, "Jet impingement cooling in power electronics for electrified automotive transportation: current status and future trends", IEEE Transactions on Power Electronics **36**, 10420–10435 (2021).

[31] R. Kandasamy, X.-Q. Wang, and A. S. Mujumdar, "Transient cooling of electronics using phase change material (pcm)-based heat sinks", Applied thermal engineering **28**, 1047–1057 (2008).

[32] S. G. Kandlikar and A. Ganguly, "Fundamentals of Heat Dissipation in 3D IC Packaging and Thermal-Aware Design", in *3D Microelectronic Packaging*, Vol. 64, edited by Y. Li and D. Goyal, Springer Series in Advanced Microelectronics (Springer, Singapore, 2021), 10.1007/978-981-15-7090-2_13.

[33] Ç. Köroğlu and E. Pop, "High thermal conductivity insulators for thermal management in 3d integrated circuits", IEEE Electron Device Letters **44**, 496–499 (2023).

[34] *Leverage Jacinto™ 7 Processors Functional Safety Features for Automotive Designs*, https://www.ti.com/lit/fs/spry336a/spry336a.pdf?ts=1702233626752.

[35] *Manage automotive test, safety, and security with a safety island*, https://blogs.sw.siemens.com/tessent/2021/06/28/manage-automotive-test-safety-and-security-with-a-safety-island/.

[36] K. Matsumoto, H. Mori, and Y. Orii, "Thermal performance evaluation of dual-side cooling for a three-dimensional (3d) chip stack: additional cooling from the laminate (substrate) side", in 2016 international conference on electronics packaging (icep) (IEEE, 2016), pp. 163–168.

[37] K. Matsumoto, H. Mori, and Y. Orii, "Thermal performance evaluation of dual-side cooling for a three-dimensional (3d) chip stack: additional cooling from the laminate (substrate) side", in 2016 international conference on electronics packaging (icep) (2016), pp. 163–168, 10.1109/ICEP.2016.7486804.

[38] *MIT Researchers Warn of Interconnect Security Vulnerabilities, Propose Mitigation Strategies*, https://www.enterpriseai.news/2022/08/16/mit-researchers-warn-of-interconnect-security-vulnerabilities/.

[39] P. Mithal, "Design of experiment based evaluation of the thermal performance of a flipchip electronic assembly", in Asme international mechanical engineering congress and exposition, Vol. 15533 (American Society of Mechanical Engineers, 1996), pp. 109–115.

[40] G. L. Morini, "Single-phase convective heat transfer in microchannels: a review of experimental results", International journal of thermal sciences **43**, 631–651 (2004).

[41] S. R. Narayana and P. K. Narayan, "Effect of load and interface materials on thermal contact resistance between similar and dissimilar materials", Applied Mechanics and Materials **592**, 1493–1497 (2014).

[42] K. NarayanPrabhu et al., "The effect of load and addition of mwcnts on silicone based tims on thermal contact heat transfer across cu/cu interface", Materials Research Express **6**, 1165h9 (2019).

[43] L. Navarro, A. De Gracia, D. Niall, A. Castell, M. Browne, S. J. McCormack, P. Griffiths, and L. F. Cabeza, "Thermal energy storage in building integrated thermal systems: a review. part 2. integration as passive system", Renewable energy **85**, 1334–1356 (2016).

[44] K. Oliphant, B. Webb, and M. McQuay, "An experimental comparison of liquid jet array and spray impingement cooling in the non-boiling regime", Experimental Thermal and Fluid Science **18**, 1–10 (1998).

[45] *PCB Layout for CMOS sensors*, https://www.cb-distribution.es/news-es/cmos_routing/.

[46] *Practical Cyber-Attacks on Autonomous Vehicles*, https://essay.utwente.nl/66766/1/Stottelaar_BA_EE.pdf.

[47] *Rambus ASCON-IP-41 Product Brief*, https://go.rambus.com/ascon-ip-41-product-brief.

[48] *RapidChiplet: A Toolchain for Rapid Design Space Exploration of Chiplet Architectures*, https://arxiv.org/pdf/2311.06081.pdf.

[49] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger", in Proceedings of the ieee conference on computer vision and pattern recognition (2017), pp. 7263–7271.

[50] H. Rong, H. Zhang, S. Xiao, C. Li, and C. Hu, "Optimizing energy consumption for data centers", Renewable and Sustainable Energy Reviews **58**, 674–691 (2016).

[51] C. S. Sharma, S. Zimmermann, M. K. Tiwari, B. Michel, and D. Poulikakos, "Optimal thermal operation of liquid-cooled electronic chips", International journal of heat and mass transfer **55**, 1957–1969 (2012).

[52] H. Sharma, D. Padha, and N. Bashir, "D-kap: a deep learning-based kashmiri apple plant disease prediction framework", in 2022 seventh international conference on parallel, distributed and grid computing (pdgc) (IEEE, 2022), pp. 576–581.

[53] Sheldon, Lloyd, Popelar, Suh, Ghaffarian, Alles, and Lee, "Nepp etw 2018: 2.5/3d packaging", in Nepp etw (2018).

[54] *Sintech*, http://www.sintechheater.com/.

[55] *Sparkle Suit by KISA*, https://sparkle-lwc.github.io/.

[56] *Synopsys ARC EM22FS Safety Processor*, https://www.synopsys.com/dw/doc.php/ds/cc/arc_em_22fs_ds.pdf.

[57] *Synopsys Embedded Security Modules for HDCP 2.3 on HDMI IP*, https://www.synopsys.com/dw/ipdir.php?ds=security-hdcp-esm.

[58] *Synopsys MACsec Security Modules for Ethernet*, https://www.synopsys.com/dw/ipdir.php?ds=security-ethernet-macsec.

[59] *Synopsys MACsec Security Modules for Ethernet Datasheet*, https://www.synopsys.com/dw/doc.php/ds/s/macsec-ethernet-security-module.pdf.

[60] *Synopsys PCI Express (PCIe) IP Solutions*, https://www.synopsys.com/designware-ip/interface-ip/pci-express.html.

[61] *Synopsys SD/eMMC Host Controller IP*, https://www.synopsys.com/dw/ipdir.php?ds=dwc_sd_emmc_host_controller.

[62] *Synopsys USB IP Solutions*, https://www.synopsys.com/designware-ip/interface-ip/usb.html.

[63] I. C. Systems, *The Shift Towards Zonal Network Architecture (Intrepid Tech Day '23)*, https://www.youtube.com/watch?v=vpmHJo8tUgE&t=238s.

[64] *Understanding 1D, 2D, and 3D convolutional layers in deep neural networks*, https://www.sefidian.com/2019/02/24/understanding-1d-2d-and-3d-convolutional-layers-in-deep-neural-networks/.

[65] R. Van Erp, R. Soleimanzadeh, L. Nela, G. Kampitsis, and E. Matioli, "Co-designing electronics with microfluidics for more sustainable cooling", Nature **585**, 211–216 (2020).

[66] *What is ASIL?*, https://www.synopsys.com/automotive/what-is-asil.html.

[67] *What Is Automotive Cybersecurity? Top 12 Examples*, https://www.upgrad.com/blog/automotive-cybersecurity/.

[68] *What is HDCP: The Complete Guide*, https://www.cablematters.com/Blog/HDMI/what-is-hdcp-the-complete-guide.

[69] *What is ISO 26262?*, https://www.synopsys.com/automotive/what-is-iso-26262.html.

[70] *What Is the Purpose of a Feature Map in a Convolutional Neural Network*, https://www.baeldung.com/cs/cnn-feature-map.

[71] J. Wu, Y. Kong, S. Zhu, Y. Xu, J. Miao, R. Liu, S. Yun, Y. Ye, and B. Jiao, "Design and test of a low junction-to-case thermal resistance packaging method", IEEE Transactions on Components, Packaging and Manufacturing Technology **11**, 2130–2139 (2021).

[72] *Yolo Object Detection – Machine Learning Project*, https://projectgurukul.org/yolo-object-detection-project/.

[73] G. Zhang, K. Zhao, B. Wu, Y. Sun, L. Sun, and F. Liang, "A risc-v based hardware accelerator designed for yolo object detection system", in 2019 ieee international conference of intelligent applied systems on engineering (iciase) (IEEE, 2019), pp. 9–11.

[74] H. Zhang, Y. Chen, Z. Huang, H. Zhang, and F. Dai, "Seechip: a scalable and energy-efficient chiplet-based gpu architecture using photonic links", in Proceedings of the 52nd international conference on parallel processing (2023), pp. 566–575.

[75] M. Zhou, L. Li, F. Hou, G. He, and J. Fan, "Thermal modeling of a chiplet-based packaging with a 2.5-d through-silicon via interposer", IEEE Transactions on Components, Packaging and Manufacturing Technology **12**, 956–963 (2022).