

JLR Chiplet Challenge

End Evaluation Report

Team 22



Inter IIT Tech Meet 12.0

Contents

1	Introduction	5
1.1	Motivation for using chiplets	5
1.2	Brief Overview of the Report	6
2	Application of Chiplets	6
2.1	ADAS Sensor Fusion System	6
2.1.1	Designed Chiplets For ADAS Sensor Fusion	8
2.1.2	High-Level Architecture	25
2.1.3	Partitioning and Floor-Planning	25
2.2	Connected Infotainment Subsystem(eCockpit)	27
2.3	Other Applications	34
3	Chiplet Integration	36
3.1	UCIE	37
3.1.1	UCIE Background	38
3.1.2	Packaging	39
3.1.3	Connectivity of Two Dies By UCIE	40
3.1.4	Specifications	42
3.1.5	UCIE IP's	42
3.2	Interposer	44
3.2.1	Innovation (Carbon Nanotube)	46
3.3	Innovations in package of interconnects	48
3.3.1	Hybrid of 2.5D and 3D Packaging	48
3.4	Safety and Reliability	49
3.4.1	Proposed Solution	51
3.4.2	Design of Security System:	53
4	Thermal Management	56
4.1	Interposer	56
4.1.1	Microchannels	56
4.2	Si-Diamond composite heat sink	59
5	Simulation	60
5.1	Simulation introduction	60
5.2	Gem5 and Heterogarnet	61
5.2.1	Garnet	61
5.2.2	Heterogarnet	62
5.3	Garnet Synthetic Traffic simulation	63
5.4	Simulation Parameters	63
5.5	Results	64
5.6	Alternative Simulation attempted	65
5.7	Future Simulation work	65

6 Future prospects	65
6.1 Vehicle to Everything(V2X) Chiplet	65
6.2 Moving Towards Complete Autonomy	66
6.2.1 RADAR Pre-processor	67
6.2.2 LIDAR Pre-processor	68
7 Conclusion	69
8 References	69

List of Figures

1	Example Of ADAS Vision Subsystem [5]	7
2	Conceptual diagram of ADAS Sensor Fusion chip 5	8
3	Block Diagram of Compute Cluster	11
4	Hercules™ 4 TMS570LS Safety Island	12
5	Vision Subsystem Chiplet	13
6	Controls and Calibration	17
7	EVE Details	17
8	DRM CTC	19
9	Chiplet Architecture	22
10	Memory Subsystem Chiplet	23
11	3D DRAM Stacking	24
12	Architecture of ADAS chiplet based system	25
13	Interconnection between ADAS Chiplets	27
14	Infotainment	28
15	Infotainment Compute Cluster	29
16	Display Architecture	31
17	Spectra 280 ISP architecture	33
18	High-level architecture diagram for the S32K1 family [62]	35
19	Physical layer components	38
20	Layers of UCIE	39
21	Advanced Package	39
22	Advanced Package Specifications	40
23	Internal structure of Interconnect	40
24	Connectivity of two dies	41
25	Overall Connections in Chiplet System	41
26	UCIE Key Performance Targets	42
27	Synopsis UCIE IP	43
28	Cadence UCIE IP	44
29	Interposer	45
30	Active Interposer Structure	45
31	Energy vs Technology Node	47
32	Latency vs Technology Node	48
33	Increasing interconnect density, power efficiency and scalability achieved with 2D, 2.5D and 3D packaging	48
34	Packaging Tradeoffs	49
35	Intel's Ponte Vecchio high-performance GPU for high-performance computing applications utilizes both EMIB 2.5D interconnect and Foveros 3D interconnect	49
36	TransMon	52
37	Design of Security System	54
38	Chiplet Architecture	55
39	Interposer Design	56
40	Temperature Distribution Along Dies ,Source:[50]	58
41	Pressure Drop across the Microchannels ,Source:[50]	59

42	Manufacturing-proces[52]	60
43	Advantage of Silicon Diamond composite(CMC) heatsink over Non Composite Microchannel Heatsink(NCMC)[52]	60
44	gem5 simulator workflow	61
45	Model of NoC simulated	64
46	V2X Block Diagram[54]	65
47	Complete Autonomy bloack diagram	67
48	S32R2x Block Diagram	68

List of Tables

1	Features of C6678	14
2	Features	16
3	Display Subsystem Hardware Specifications	18
4	Terminologies	18
5	CXL Specifications	21
6	Comparison Of In Memory Computation Memory Products	24
7	Adreno 660 specifications	34
8	2.5D Implementation Results for Non-Secure Versus Secure Designs, Chiplets in GlobalFoundaries 65nm, Interposer in Synopsys SAED 90nm	55
9	Simulation Parameters	64

Abstract- In the rapidly evolving landscape of automotive technology, the adoption of chiplets has emerged as a revolutionary approach to enhance flexibility, scalability, and efficiency in subsystem design. ADAS, infotainment system and body domain controller are suggested as the subsystems that can utilize the chiplet technology. In this report we give the complete structure and components of the System on Chiplets This includes micro-architecture of the Chiplet and specifications of each block, floorplanning, interconnect details as well as interposer structure and components. We have also performed simulation of UCIe interconnect using gem5 simulation software. We have also proposed security features as well as thermal management features to tackle the limitations of chiplet.

1 Introduction

With the unprecedented growth of High-Performance Compute (HPC) and Autonomous Driving (AD) seen in recent times, the traditional chip design strategy is falling short and encountering a fundamental manufacturing limit. Advanced Driving Assistance Systems, or ADAS systems, are becoming increasingly ubiquitous in cars. McKinsey estimates that by 2030, over 50% of the cars will ship with at least ADAS level 2, and some 2% (or about 1.6 million cars per year) will ship with ADAS level 4[1]. The number and types of sensors in a car are increasing exponentially, including Lidar, Radar, and ultrasonic sensors. These sensors collect data, and this data is combined and processed to give information about speed, surroundings, weather, and terrain conditions.

The automotive domain applications require highly optimized and energy-efficient functions: generic ones such as cores, GPUs, embedded FPGAs, dense and fast memories, and also more specialized ones, such as machine learning and neuro-accelerators to efficiently implement the greedy computing demand of Big Data and artificial intelligence (AI) applications.

Due to the slowdown of advanced CMOS technologies (7 nm and below), with yield issues, design, and mask costs, the innovation and differentiation through a single die solution are not viable anymore. Mixing heterogeneous technologies is the only viable option.

1.1 Motivation for using chiplets

- Due to increasing issues in advanced CMOS technologies (7 nm and below), achieving high yield on large dies at acceptable costs is not possible anymore. By dividing a system into various sub-modules, called chiplets, it is possible to yield larger systems at an acceptable cost, thanks to Known Good Die (KGD) sorting[2].
- By an elegant divide-and-conquer partitioning scheme, chiplets allow the building of modular systems from various building blocks and circuits, focusing more on functional aspects than on technology constraints.
- For chiplets, the right technology is selected to implement the right function: advanced CMOS for computing, DRAM for memory cubes like high bandwidth memory (HBM),

non-volatile memory (NVM) technology for data processing within AI accelerators, mature technology for analog functions (IOs, clocking, power management, and so on).

1.2 Brief Overview of the Report

We have given an extensive report on how chiplets can revolutionize the automotive industry, which is supported by data in published literature and data sheets of companies. The report is divided into V Sections. Section I introduces the innovations and benefits that chiplets can bring to the automotive industry. Section II gives a comprehensive explanation of three major applications of chiplets in automotive subsystems. These applications are ADAS Subsystem, Infotainment System and finally ECUs and Domain Controller. We have given detailed information, including block diagrams of each subsystem, specs and features of each subunit, references of industrial IPs that can be bought directly off the shelf and IPs that need to be custom built by JLR for each application. We have also explained the floorplanning of each chiplet and mentioned the interconnects and interposer that need to be implemented. Section III gives information about the interconnects and interposer that will be used to connect the chiplets. This which will include interconnects bandwidth, latency, and communication efficiency. We have briefed about the circuitry that can be implemented in the active interposer. We have also suggested security features that need to be implemented in the interconnects for safety and security of the host introduced certain innovations that are being researched to improve the functioning of interconnects. Section IV addresses the issue of thermal management required to cool down each system of chiplets. In Section V we have given results of simulations that we have performed using gem5. In Section VI, we have suggested future prospects that can be adopted by JLR to move towards full autonomy. Finally, in Section VI, we conclude our report by suggesting a few innovations that are being researched across the globe that can increase the efficiency in terms of cost, power, and operation of semiconductor chips.

2 Application of Chiplets

2.1 ADAS Sensor Fusion System

According to the Ministry of Road Transport and Highways (MoRTH), a total of 4,61,312 road accidents occurred in 2022, which claimed 1,68,491 lives, while 4,43,366 people were injured. This sums up to a harrowing of 19 lives being taken every hour last year. [3]

ADAS (Advanced Driver Assistance Systems) applications such as Pedestrian detection/avoidance, Lane departure warning/correction, Traffic sign recognition, Automatic emergency braking, Blind spot detection to name a few, can help avoid these tragedies.

These applications are part of technologies such as Front Camera, Park Assist (Surround View/Rear Camera) and Fusion. Cars with some of these ADAS functionalities are currently available in the market and several OEMs (Audi, BMW, Toyota and Nissan) have already announced advanced ADAS functionalities and driver-less car programs, the future of ADAS technology.

ADAS systems use a range of sensors (RGB, Radar, Ultrasonic, see Figure 1) to capture information about the surroundings and then process this information to implement the ADAS functionalities. Vision sensor-based (e.g RGB sensor) processing also known as vision analytics processing forms an integral part of ADAS systems and provide the “eyes” to these systems. Advanced Application Processors (AP) and embedded Software (SW) is required to build these ADAS systems and functionalities. [4]

The main sensors employed in a completely equipped ADAS system in a medium sized vehicle and the features that they enable are:

1. Cameras (x4):
 - (a) Front view camera: Lane departure warning, Traffic sign recognition
 - (b) Surround view cameras (x2)
 - (c) Rear view camera: Park assistance
2. Radars (x8)
 - (a) Long range radar (x1 at the front): Adaptive cruise control
 - (b) Medium range radar (x1 at the front): Emergency braking, Collision avoidance, Pedestrian detection
 - (c) Short range radar (x6): Cross traffic alert, Rear collision warning, Blind spot detection
3. Ultrasound sensors: Park assistance
4. LIDAR: (Optional) Used in cars with autonomous drive capabilities.

A typical representation of ADAS features and sensors used in a medium sized automobile is given below :

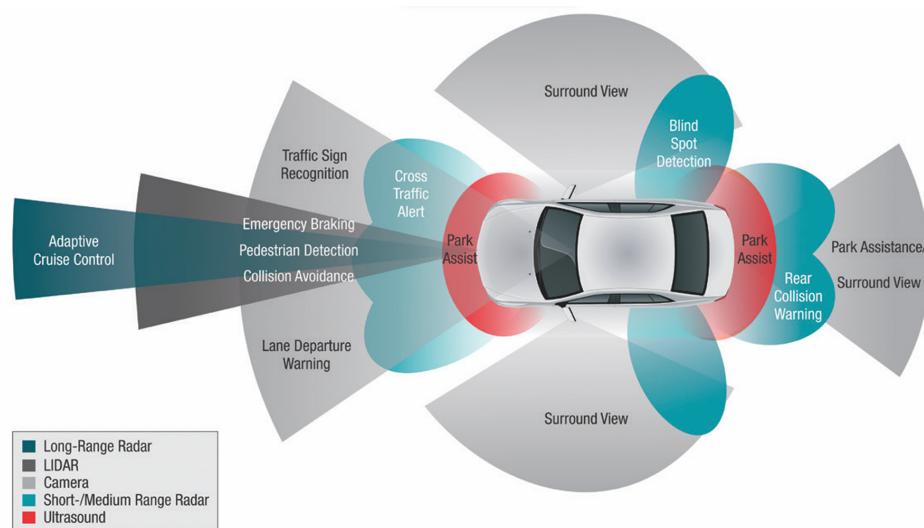


Figure 1: Example Of ADAS Vision Subsystem [5]

Advanced driver assistance systems (ADAS) today are still treated as separate systems with different systems for cameras, radars, ultrasound etc, independent from each other. Each system has its own purpose and either displays information or performs an activity (such as a chime) without consideration for any other ADAS systems. Depending on the type of sensor technology (radar, camera, ultrasound, light detection or range), this allows certain functionality, but does not make the best use of the systems.

To build fully autonomous cars, it will be necessary to combine the information and data from different sensors, exploiting their individual advantages and making up for the weaknesses each individual system always has. This is called sensor fusion. Instead of multiple, completely independent systems, the various ADAS systems feed their information into a central sensor fusion unit that can combine all of the information to provide better situational awareness.

This sensor fusion unit is quite complex thus it is beneficial to implement it using chiplets than an SoC as it offers many advantages such as higher yield, modularity, customization, power efficiency, scalability, and fault tolerance, contributing to the development of more robust and adaptable automotive sensor systems.

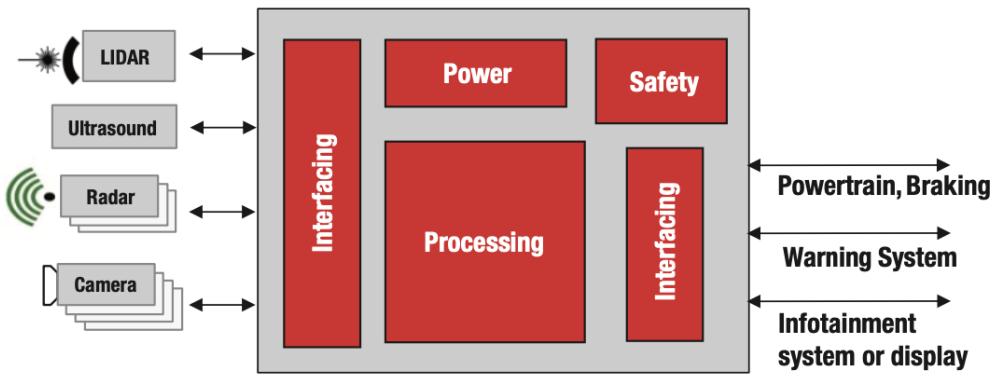


Figure 2: Conceptual diagram of ADAS Sensor Fusion chip 5

2.1.1 Designed Chiplets For ADAS Sensor Fusion

A. Compute Cluster Chiplet

We are proposing a heterogeneous compute cluster consisting of

- (a) Two core pairs i.e. four cores of **Arm Cortex-A78AE** paired with
- (b) Four cores of **Arm Cortex-A65AE** (that act as co-processors).

The proposed compute cluster chiplet is to be fabricated on **7 nm** node. This is because digital logic circuitry benefits from fabrication on a lower technology node.

Right-sized compute is the mantra of the day. Put simply, no one micro-architecture satisfies the application needs of these market segments. As an example, highly compute intensive ADAS features such as Pedestrian detection, Traffic sign recognition, emergency braking, automatic parking etc needs the system to sense data, perceive

obstacles and decide on the right path vector before engaging the vehicular controls. Just the middle two tasks, require an enormous variety of algorithmic execution.

Thus a heterogeneous compute cluster is required with the Cortex A78AE focused on applications where high performance is needed, the Cortex A65AE is focused on high-throughput applications.

Thus different core clusters can be assigned different workloads with the Cortex A65AE working on sensor data processing tasks which require high throughput. And the Cortex A78AE cores work on perception and decision making workloads which are algorithmic intensive. Here the cores would require higher levels of functional safety, and thus the A78AE cores would be operating in lock-step mode i.e. both cores of A78AE in a core pair will be executing the same code, and the result is validated only if the outputs of both cores is identical.

However, this is not fixed and the compute platform provides flexibility in terms of the configuration of the Split-Lock layout of the hardware; it's something that would be determined on a firmware level, and vendors would be able to reconfigure with a software update if they so wished. [6]

The details of the proposed compute cluster are as follows:

(a) **ARM Cortex-A78AE Processor [7]**

Specifications:

- Architecture : Armv8.2-A (Harvard)
- Microarchitecture features:
 - Pipeline: Out of Order
 - Superscaler: Yes
 - Floating Point unit: Included with INT8 dot product and IEEE FP16
 - Optional Cryptography Unit
 - Max number of CPUs in cluster: 4
 - Physical Addressing: 48 bit
- Memory system and interface features:
 - L1 I-Cache/D-Cache: 64KB (32 /64 KB) (In core)
 - L2 cache - 512KB (256 KB / 512 KB) (In core)
 - L3 Cache - 4MB (512 KB / 4MB) (Unified for 4-cores)
 - Max Clock rate - 3GHz
 - Can be implemented in 7nm or 5nm nodes.

Performance Metrics:

- Greater than 312.5 K DMIPS
- Less than 15 W (when used as a 16 core processor)

The Cortex-A78AE core cannot be instantiated as a single core. The Cortex-A78AE core must be used in a core pair configuration with a maximum of two core pairs in each cluster for a total of four cores. The DynamIQ Shared Unit

AE (DSU-AE) provides a boot-time option for the cluster to execute in either Split-mode, Lock-mode, or Mixed-mode.

- Split-mode, where the cores in each core pair operate independently of each other.
- Lock-mode, where one of the cores in a core pair functions as a redundant copy of the primary function core.
- Hybrid-mode, where the cores operate independently, as in Split-mode, while the DSU-AE operates in lock-step, as in Lock-mode.

Functional Safety features of microarchitecture:

- **Dual Core Lock-Step (DCLS)** : The Cortex-A76AE is capable of running in Dual Core Lock-Step (DCLS) when it is running in Lock-mode, and hence is able to contribute towards a system's ASIL D hardware diagnostic coverage requirements.
- **Reliability, Availability and Serviceability (RAS) features** : As part of the Armv8.2 architecture extension, Cortex-A78AE includes standardized error reporting across the core and the DSU, error injection as a means of testing fault management and data poisoning as a way of deferring error aborts till point of execution.
- **Memory protection:** Cache protection ensures that the Cortex®-A78AE core is protected against errors that result in a RAM bitcell holding the incorrect value. It supports SED (Single Error Detect) , SECDED (Single Error Correct, Double Error Detect) and interleaved parity.

(b) **Arm Cortex-A65AE (Co-Processor) [8]**

- Architecture : Armv8-A (Harvard)
- Microarchitecture features:
 - Pipeline: Out of Order
 - Superscaler: Yes
 - Floating Point unit
 - Optional Cryptography Unit
 - Max number of CPUs in cluster: 8
 - Physical Addressing: 48 bit
 - Dual Core Lock-Step
- Memory system and interface features:
 - L1 I-Cache/D-Cache: 16KB to 64KB
 - L2 cache - 64KB to 256KB
 - L3 Cache - Optional, 512KB to 4MB

(c) **Corelink-GIC-600AE [9]:** GIC-600 AE is a Generic Interrupt controller that handles interrupts from peripherals to cores and between cores

(d) **Corelink MMU-600AE [10]:** The MMU-600AE is a functional safety version of the System-level Memory Management Unit (SMMU) MMU-600 that translates

an input virtual address to an output physical address. This translation is based on address mapping and memory attribute information that is available from configuration tables and translation tables that are stored in memory.

- (e) **Core Sight ELA-600 [11]:** The ELA-600 Embedded Logic Analyzer is a component for debugging hardware-related issues. Debug signals are connected from the IP being debugged to the ELA-600, which compares the signals with a target value and drives actions.

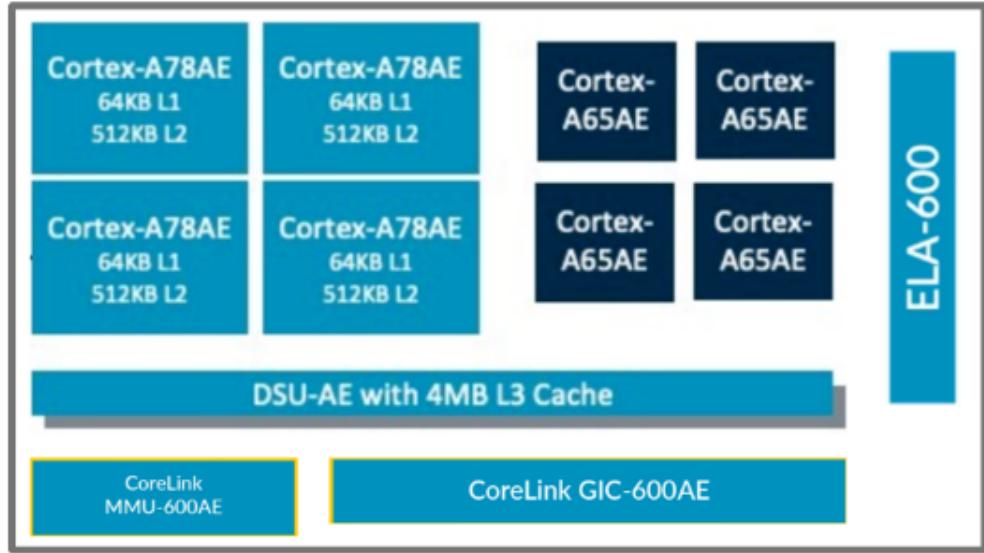


Figure 3: Block Diagram of Compute Cluster

B. Safety Island Chiplet (MCU Domain) [12]

Automotive systems are subject to strict safety standards, such as ISO 26262. These standards require the implementation of safety mechanisms to ensure that critical functions operate correctly, even in the presence of faults or failures.

The safety island MCU is designed to handle and isolate safety critical functions. It provides the following features:

- Provide fault tolerance i.e. in case of an error in a non-critical component of the chiplet system, the safety island can continue to operate independently, maintaining the essential safety functions
- Diagnostic capabilities i.e. detecting faults in the rest of the chiplet system.

Safety island in ADAS may be used to handle safety critical functions such as signalling external ECUs that control emergency braking, steering, powertrain etc. Such functions cannot be entrusted on the main compute cluster as it is already burdened with heavy processing. Thus we need a safety island.

To do this, the safety island first must have maximum freedom from interference from the rest of the system, so it needs dedicated compute, memory and I/O resources,

its own clock nets and power grid, and to the greatest extent possible it needs low complexity so that internal (to the island) failure modes can be well-understood and well-mitigated.

Overall it must continue to work even if the rest of the functionality on the chiplet based system falls apart.

- Thus the system is proposed to use **Hercules™ 4 TMS570LS Safety MCU from TI** which is present on a separate chiplet. The specifications of the MCU are as follows:
 - ARM Cortex-R4F core floating-point support
 - Up to 180 MHz
 - Lockstep safety features built-in simplify SIL-3/ASIL D applications
 - Up to 3-MB Flash/256-KB RAM with ECC
 - Memory protection units in CPU and DMA
 - Multiple communication peripherals: Ethernet, FlexRay, CAN, LIN, SPI
 - Flexible timer module with up to 44 channels
 - 12-bit analog/digital converter
 - External memory interface
- The safety island IP is available on **65 nm** technology node.

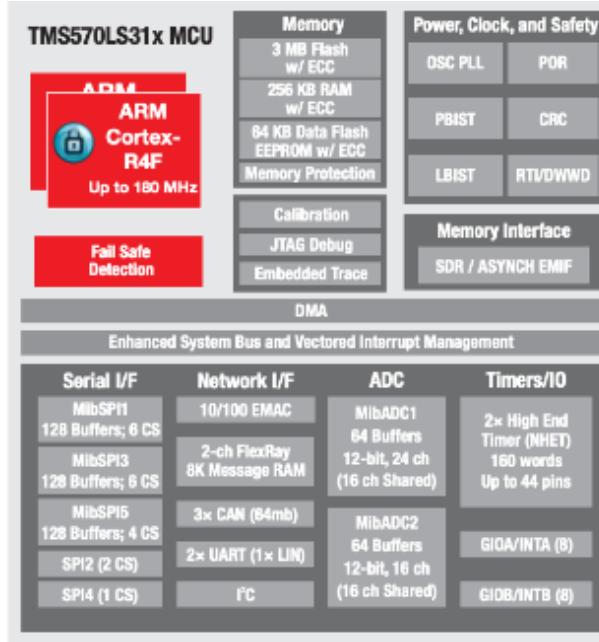


Figure 4: Hercules™ 4 TMS570LS Safety Island

C Vision Subsystem Chiplet

The Vision Subsystem block in short should acts as a Deep Learning accelerator that handles the data coming from the Application cores. The Vision Sub-system is one complete chiplet we are suggesting JLR to design on its own. There are mainly three reason to this:

Motivation from Tesla FSD chip :

- Tesla realised that they actually needed something on the order of half a million to a few million chips per year. The expense of developing an ASIC instead of using off the shelf hardware(NVIDIA GPUs) seemed very profitable for them. The neural network accelerators(NPUs) on the Tesla FSD chip are a fully custom design made by the Tesla hardware team. Thus They are also the largest component on the chip and is the most important piece of logic.[13]
- Another important aspect to note is that while most of the logic on the chip makes use of industry-proven IP blocks in order to reduce risk and accelerate the development cycle. Paying a third party to make hardware, In such a case it would mean JLR spending their own money to advance the learning curve of your competitors.
- Not only does this bring about profit in production lines, but also in enhanced efficiency and processing. They could significantly boost performance and optimize power consumption with their own ASIC.

Thus by considering the above factors, the suggestion put forward is for JLR to adopt to a chiplet based design for the whole Vision Subsystem module as given in Figure 5.

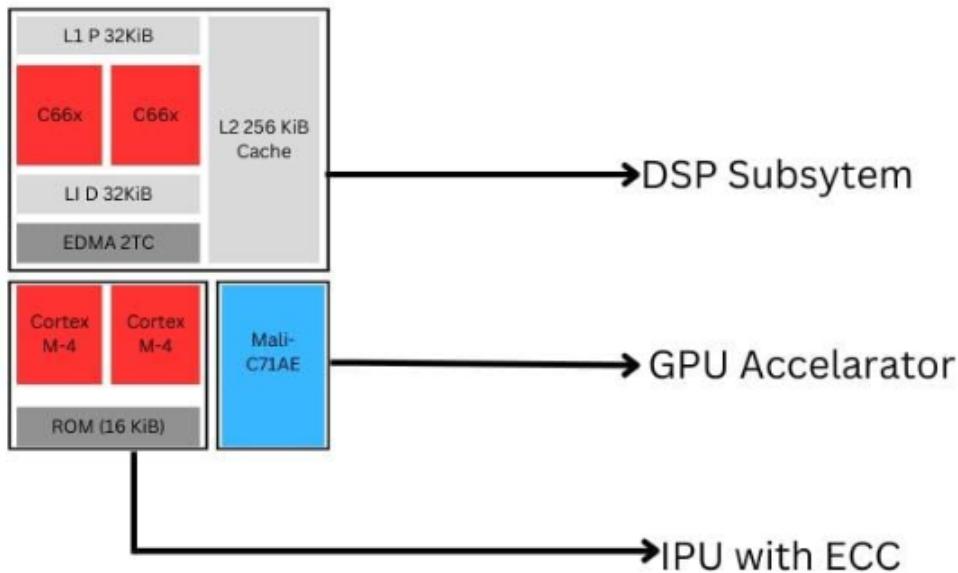


Figure 5: Vision Subsystem Chiplet

The Vision Subsystem chiplet is mainly composed of **DSP Subsystem, IPU with ECC functionality and GPU accelerator**. An alternative for GPU, ie. NPU with EVE block have also been proposed, which will be found below. JLR can choose between these two accordingly.

- DSP SUBSYSTEM

- (a) C66x DSP Core :

Basically they're high performance DSP cores designed by Texas Instruments, as a part of their TSM320C66x DSP family. The proposed DSP subsystem is using TSM320C6678 DSP. TI's C6678 DSP, is the industry's highest performing multicore DSP in production today featuring eight 1.25-GHz DSP cores and delivering 160 single-precision GFLOPS and 60 double-precision GFLOPS in just 10W. [14]. They are comprised of eight TMS320C66x™ DSP Core of **7 nm** technology node Key Features:

- Greater than 500-GFLOPS PCIe cards available today – half-length, single-width, 50W
- Standard programming model and support for OpenMP
- Free multicore software development kit and scientific programming examples
- Optimized math and imaging libraries
- Low-cost evaluation modules available for faster development

The table below lists some important attributes of the C6678 DSP.

Key Attributes	C6678
Single-/Double-precision FLOPS	160/60
Cores	8
Processor speed (GHz)	1.25
L2 memory (MB)	8
L3 memory (GB) ECC	Up to 8
Memory BW (GB/s)	12.8
Power consumption (W)	10

Table 1: Features of C6678

- (b) EDMA 2TC :

EDMA, Enhanced Direct Memory Access with 2 Transfer Controllers, basically is a subsystem that provides direct memory access functionality, allowing data to be transferred between DSP peripherals and memory without the direct involvement of the main processor. This offloads the work for the DSP. The “2TC” indicates the presence of 2 Transfer controllers within the EDMA subsystem, enhancing the parallelism of data movement in the system. These transfer controllers manage and coordinate data transfer between different peripherals and memory locations.

- IPU with ECC

IPU (Image Processing Unit) with ECC (Error Correction Code) includes 2 ARM Cortex-M4 cores, with 16KiB ROM. This part is meant to perform Vision Pre-processing Acceleration (VPAC) as well as Depth and Motion Perception Acceleration (DMPAC). The VPAC includes common vision primitive functions, performing pixel data processing tasks, such as:

- Color processing and enhancement
- Noise filtering
- Wide dynamic range (WDR) processing
- Lens distortion correction
- Pixel remap for de-warping
- On-the-fly scale generation
- On-the-fly pyramid generation.

The VPAC offloads these common tasks from the main SoC processors (ARM, DSP, etc.), so these CPUs can be utilized for differentiated high-level algorithms.

The Depth and Motion Perception Accelerator (DMPAC) is a power efficient hardware accelerator that computes dense stereo depth maps (depth) and dense optical flow vectors (motion) from camera inputs.

(a) ARM Cortex-M4 cores :

The 32-bit Arm® Cortex®-M4 processor core is the first core of the Cortex-M line up to feature dedicated Digital Signal Processing (DSP) IP blocks specifically within the IPUs, including an optional Floating-Point Unit (FPU). It addresses digital signal control applications that require efficient, easy-to-use control and signal processing capabilities and is known for its efficiency in DSP tasks.

Just like the Cortex-M3 core, the Cortex-M4 core achieves **1.25 DMIPS/MHz** and **3.42 CoreMark/MHz** thread performance. [15]

Key features of Arm® Cortex®-M4 core [15]

- Armv7E-M architecture
- Bus interface 3x AMBA AHB-lite interface (Harvard bus architecture)
AMBA ATB interface for CoreSight debug components
- Thumb/Thumb-2 subset instruction support
- 3-stage pipeline
- DSP extensions: single-cycle 16/32-bit MAC, Single cycle dual 16-bit MAC, 8/16-bit SIMD arithmetic, Hardware Divide (2-12 Cycles)
- Optional single precision Floating Point Unit (FPU), IEEE 754-compliant
- Optional 8 MPU regions with sub-regions and background region
- Integrated bit-field processing instructions and bus-level bit banding
- Non-maskable interrupt and 1 to 240 physical interrupts with 8 to 256 priority levels

- Wake-up interrupt controller
- Integrated WFI and WFE Instructions and Sleep-On-Exit capability, Sleep and Deep Sleep Signals, Optional Retention Mode with Arm Power Management Kit.
- Optional JTAG and Serial Wire Debug ports. Up to 8 breakpoints and 4 watchpoints
- Optional Instruction Trace (ETM), Data Trace (DWT), and Instrumentation Trace (ITM)

(b) 16 KiB ROM :

Presence of ROM indicates that Cortex M4 cores have a dedicated space for storing the firmware, likely to be used for storing the initial boot code.

- Mali-C71AE

Mali-C71AE is the highest-performance Image Signal Processor (ISP) from Arm. It is built on a **16 nm** FinFET process node. Mali-C71AE delivers key visual information for the smart automotive and industrial markets [16]. Mali-C71AE provides multi camera support with up to four real-time cameras, or up to 16 virtual cameras. This multi camera support offers a wide range of data output formats, and provides the flexibility to support both human and machine vision applications. Applications for the industrial market include production line monitoring and quality control. Mali-C71AE is also suited for the emerging smart automotive market. Examples include delivering key visual information to the driver for clear and convenient viewing and providing image data to machine vision systems for driver assistance, like lane keeping and collision avoidance. Mali-C71AE is the first product in the Mali family of ISP with built-in features for functional safety applications.

Features:

Camera support	4 dedicated video inputs of max resolution 4096×2560
Sensor type support	RCGB, RCCC, RCCB, RCCG, RGBIr
Channel support	Memory-to-memory processing mode for up to 16 channels
Processing performance	Up to 1.2 Giga pixels/second throughput
Safety features	Mali-C71AE supports vision systems that need to achieve ISO 26262 ASIL B diagnostic requirements in various automotive applications. Examples include <ul style="list-style-type: none"> - Advanced driver assistance systems (ADAS) - Mirror replacement - Night vision improvement

Table 2: Features

Controls and Calibration:

	Key Features	Components
Arm ISP Reference Platform	<ul style="list-style-type: none"> Speed up initial evaluation of the system, image quality, and software performance with the Arm ISP reference platform Pre-built and pre-tuned with the Mali-C71AE ISP Accelerate development time with: <ul style="list-style-type: none"> A full introduction of the Mali-C71AE An image quality evaluation before silicon is available Information on how to use the ISP tools that are provided 	<ul style="list-style-type: none"> The Arm ISP reference platform is delivered with one fully-tuned image sensor supported by FPGA logic and software components Works out-of-the-box with minimum user interaction Provides extended capabilities for monitoring ISP state by using the ISP tools <p>Arm works with many sensor vendors. If you would prefer a different image sensor with your reference platform, contact your Arm partner manager or one of our technical experts.</p>

Figure 6: Controls and Calibration

Note that this GPU will be used for all parallel processing within the ADAS Chiplet.

Alternatively within the Vision Subsystem, there is an additional option of an NPU based Embedded Vision Engine similar to the TI's Vision Acceleration Pac. The motivation behind this was the 8x throughput with EVE based parallel processor rather than giving it fully to a processor core.

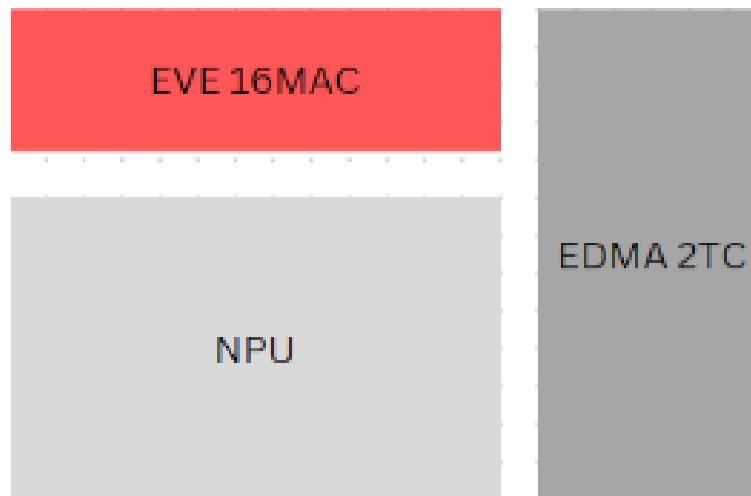


Figure 7: EVE Details

EVE - Embedded Vision Engine with 16 Multiply Accumulate Units is specifically designed for parallel processing, which can be beneficial than GPUs for application specific tasks like repetitive complex Computer Vision algorithms, Image segmentation, lane detection, object detection, etc. The proposed approach is a hybrid one leveraging the strengths of both EVE and a dedicated Neural Processing Unit (NPU) which would offload the EVE and could perform Deep Learning tasks of Neural Network based Object Detection and Segmentation, while EVE could be used for other Vision tasks that can benefit from parallel processing.

D. Display Subsystem Chiplet

The Display Subsystem (DSS) is a hardware block responsible for fetching pixel data from memory and sending it to a display peripheral like an LCD panel or an DisplayPort monitor. DSS hardware is generally divided into 2 parts :

- **Display Controller (DISPC)** is one, which handles fetching the pixel data, doing color conversions, composition, and other pixel manipulation
- **Peripherals**, which encode the raw pixel data to standard display signals, like MIPI DPI or DP.

In addition to the ASAS's DSS proposed (mainly the instrument cluster), Infotainment system has its own connected DSS or eCockpit which will be found on the later part of this report.

The DSS proposed is TI's **DSS7 of the AM65x SoC family**. Below is listed some of its hardware and supported features and driver entries [17].

Hardware Features :

DSS version	Outputs	Pipes	Video ports
DSS7	DPI, DP, DSI	2× VIDL, 2× VID	4

Table 3: Display Subsystem Hardware Specifications

(a) Driver Architecture

The driver for DSS IP is [tidss], it is a Direct Rendering Manager (DRM) driver, located in the directory drivers/gpu/drm/tidss/ in the kernel tree. tidss does not implement any 3D GPU features, only the Kernel Mode Setting (KMS) features, used to display pixel data on a display. In addition to tidss, there are a number of bridge and panel drivers located in drivers/gpu/drm/bridge/ and drivers/gpu/-drm/panel/ which provide support for various panels and bridges (both external and internal to SoC). The mapping of DRM entities to DSS hardware is roughly as follows:

DRM term	HW term
plane	DSS pipeline
crtc	DSS videoport
encoder	Internal and external bridges
connector	Connector or a panel

Table 4: Terminologies

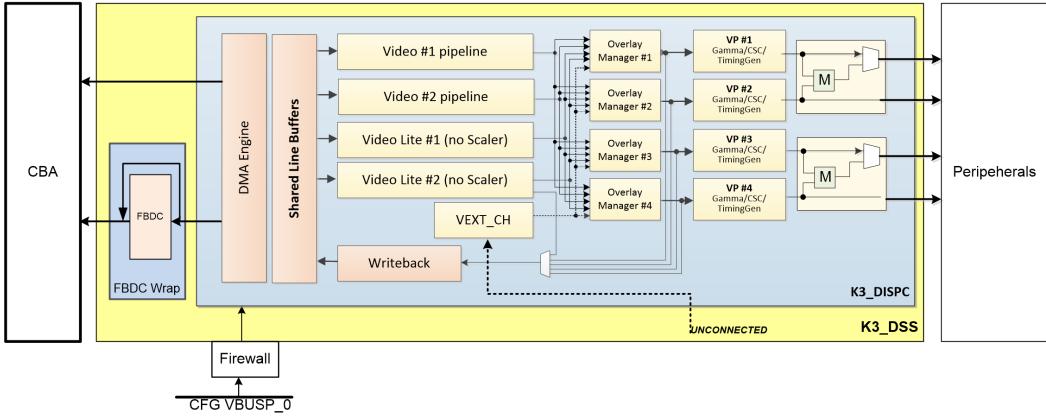


Figure 8: DRM CTC

The above image gives an overview of the DSS hardware. The arrows show how pipelines are connected to overlay managers, which are further connected to video-ports, which finally create an encoded pixel stream for display on the LCD or monitor.

- (b) Display Controller (DISPC) DISPC is the block which is responsible for fetching pixel data from the memory through DMA pipelines, and then create a pixel stream for the peripheral. The pixel stream comprises of a composition of one or more image layers which we finally want to present on the display. DISPC can be split into 3 major sub-blocks:

- Pipelines

Pipelines (or DMA channels) consist of the HW block which performs DMA to fetch image pixels (of different color formats) from RAM. Besides performing DMA, pipelines perform other functions like replication, ARGB expansion, scaling, color conversion, VC1 range mapping on the input pixels before it's passed on to the overlay manager. An overlay manager receives pixel data from one or more such pipelines, and performs the task of composing them and passing it on to the video-port. There are two types of pipelines: VIDL and VID. The difference between the two is that VID pipelines support scaling, and VIDL does not. The number of pipelines within DSS varies with the DSS IP version used in the SoC.

- Overlay Managers (Compositors)

Overlay managers are the blocks which take pixel data from one or more pipelines, layer them to form a composition, and create a pixel stream for the video-ports to consume. The compositor part takes pixel data from multiple pipelines, composing them on the basis of their position with respect to the complete overlay manager size. Tasks like alpha blending, color-keying and z-order are also performed by the compositor in the overlay manager.

- Video Ports (Timing generators)

Video ports take a pixel stream from an overlay manager, and encode it into

a standard video signal which is understood by the LCD panel/monitor or an internal peripheral (like eDP). These video standards are specified by MIPI or general video/display bodies. The timing generator part of the video port is responsible for providing the pixel stream generated by the compositor above according to the timings desired by the peripheral. The timing generator is a state machine which provides RGB data along with control signals like pixel clock, hsync, vsync and data enable. This timing info is used by the panel / peripheral to display the composited frame on the screen.

E. General Connectivity And Security Chiplet

Advanced Driver Assistance System involves a variety of sensors and communication technologies to enhance vehicle safety and improve the driving experience. For this , it needs several connectivity protocols to communicate between the chiplets and sensors. Here are some of them:

- (a) MIPI (Mobile Industry Processor Interface), CSI-2 and D-PHY:
 - High speed interface[18]
 - RAW-16 and RAW-20 color depth, increase virtual channels from 4 to 32
 - Data rate equals 16Gbps with a roadmap to 48Gbps on the downlink and an uplink rate of 200Mbps; latency is low (6 us) and reach is 15 meters.[19]

- (b) Ethernet:

Usefulness -

- Higher data throughput is required for ADAS like rear view or surround view camera systems
- Low latency is required for ADAS like for adaptive cruise control etc.
- Ethernet standards like Audio Video-Bridging, Time Sensitive Networks (TSN) enable new applications

Specifications [20]-

- Data rates are 2.5,5,10 Gb/s
- The back channel can be used to transport an I2C bus at 400 kb/s.
- it can control GPIO lines at rates of up to 1 Mb/s.

- (c) CXL:

CXL is a dynamic multi-protocol technology designed to support accelerators and memory devices. CXL provides a rich set of protocols that include I/O semantics similar to PCIe (i.e., CXL.io), caching protocol semantics (i.e., CXL.cache), and memory access semantics (i.e., CXL.mem) over a discrete or on-package link.

Specifications [21]-

Speed	Full duplex 1.x, 2.0(32GT/s) : 3.938 GB/s($\times 1$) 63.015 GB/s($\times 16$)
	3.0(64GT/s) : 7.563 GB/s($\times 1$) 121.0 GB/s($\times 16$)
Style	Serial

Table 5: CXL Specifications

(d) CAN:

Controller Area Network is a multi-commander serial bus, allowing multiple nodes to independently read and write to the bus. Each message frame includes an identifier determining priority. In case of simultaneous transmissions, the node with the highest priority gains bus control. CAN ensures reliability in harsh conditions, enabling Electronic Control Units (ECUs) to communicate via a single pair of wires.

Specifications –

CAN Communications speeds up to 10Mbps[22].

(e) PCIE:

PCIe is a communication standard for bidirectional high-speed serial buses that meet high-bandwidth, ultra low latency performance requirements. Support the high-bandwidth and low-latency systems handling the exponential increase of sensor data and user information that requires real-time processing.

Specifications –

Currently, the Bandwidth of PCIe 4.0 specification reaching 16 GT/s, PCIe 5.0 specification having 32 GT/s and PCIe 6.0 specification have 64 GT/s.[23]

(f) MOST:

The MOST (Media Oriented Systems Transport) is a high-speed multimedia network technology serial bus-based daisy-chain topology or ring topology and synchronous data communication protocol to transport the audio, video, voice and data signals via plastic optical fiber (POF) (MOST25, MOST150) or electrical conductor (MOST50, MOST150) physical layers.[24]

Specifications[25]:

- For MOST25, Bandwidth is 23 Megabaud
- It is divided into 60 different channels
- For MOST50 bandwidth get doubled

(g) FlexRay:

FlexRay is a communication bus designed to ensure high data rates, fault tolerance, operating on a time cycle, split into static and dynamic segments for event-triggered and time-triggered communications. It is developed by FlexRay Consortium.[26]

Specifications –

FlexRay supports data rates up to 10 Mbit/s, explicitly supports both star and bus physical topologies, and can have two independent data channels for fault-tolerance (communication can continue with reduced bandwidth if one channel is inoperative). The bus operates on a time cycle, divided into two parts: the static segment and the dynamic segment. The static segment is preallocated into slices for individual communication types, providing stronger determinism than its predecessor CAN. The dynamic segment operates more like CAN, with nodes taking control of the bus as available, allowing event-triggered behavior.

The security part of the chiplet is explained in Section 3.4 extensively.

Architecture of Security and connectivity Chiplet

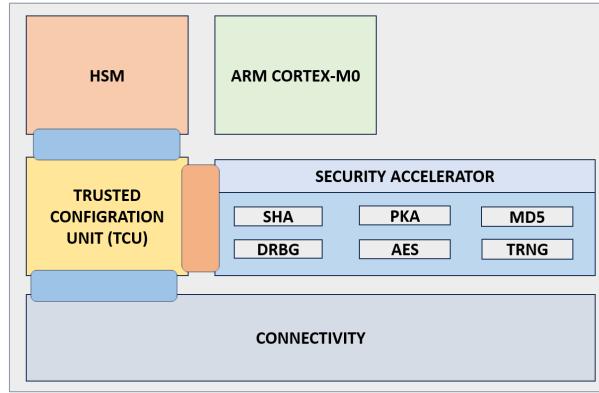


Figure 9: Chiplet Architecture

F) Memory Subsystem Chiplet

In the automotive industry, memory subsystem chiplets are vital components, significantly boosting overall system performance and reliability. These chiplets seamlessly integrate diverse memory technologies, including high-speed DDR4/DDR5 RAM and non-volatile memory, addressing the stringent requirements of modern automotive applications.

Following are the main components of the subsystem:

- (a) MMCSD
- (b) GPMC - General Purpose Memory Controller
- (c) Four External Memory Interface (EMIF) module with ECC [27]

Specifications :

Supports LPDDR4 memory types

Supports speeds up to 4266 MT/s

Up to 4x32-b bus with inline ECC up to 68 GB/s

- (d) 512KB on-chip SRAM in MAIN domain, protected by ECC
- (e) Up to 8MB of on-chip L3 RAM with ECC and coherency

Specifications :

ECC error protection

Shared coherent cache
Supports internal DMA engine

- (f) Flash Memory Interfaces
– Embedded MultiMediaCard Interface (eMMCTM 5.1)

Specifications:

Auto refresh, manual refresh, enhanced health status
Interface with additional automotive features
Capacities up to 256GB in small form-factor BGA
AEC-Q100 temperature grade
– One Secure Digital(R) 3.0 / Secure Digital Input Output 3.0 interfaces (SD3.0/SDIO 3.0)
– Universal Flash Storage (UFS 2.1) interface with two lanes

Architecture of Memory Subsystem

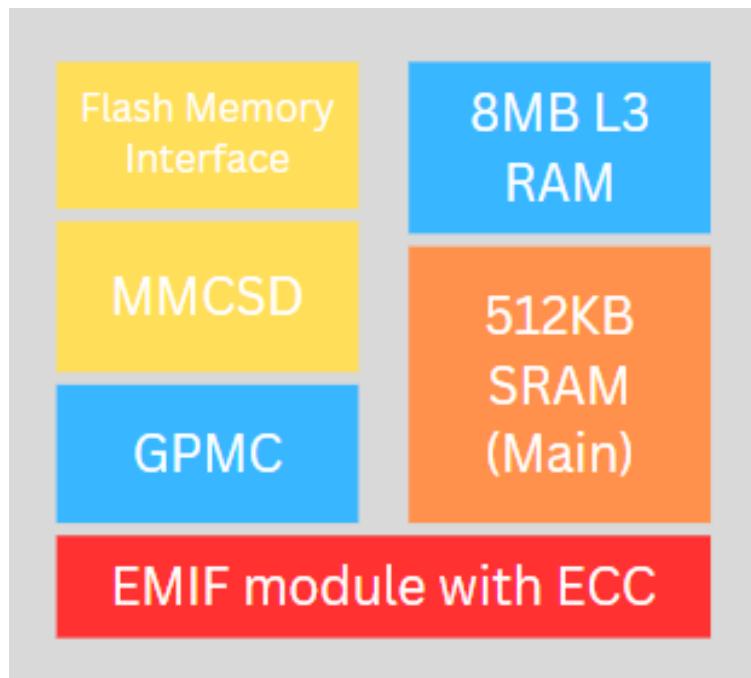


Figure 10: Memory Subsystem Chiplet

G) 3D Stacked DRAM Chiplet :

Advanced Driver Awareness Systems (ADAS) and self-driving vehicle systems demand powerful processors that require the memory capacity and bandwidth that is only possible with DRAM. However, this stacked DRAM supports **In Memory Computing** which can be used in AI application over automotive computing.

AliDAM announced the successful development of a DRAM-based 3D bonded stacked PIM chip. The chip can achieve a throughput efficiency of 184.11 QPS/W and has an on-chip memory density of 64 Mb/mm² and a bandwidth density of 2.4 GB/s/mm² [28].

TSV-based 3D stacking technology is proven to enable very-high-bandwidth and low-latency memory processor interconnects [28], and the resulting system is well suited for high throughput applications.

Example of the block diagram of 3D stacked DRAM on the logic chip is shown.

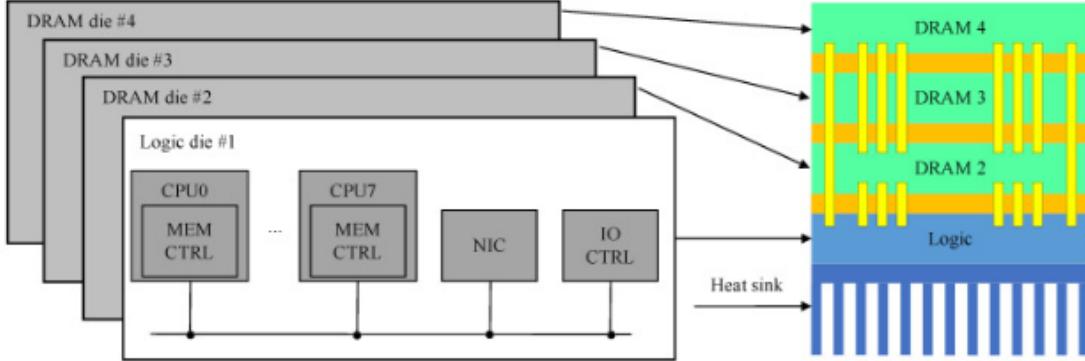


Figure 11: 3D DRAM Stacking

JLR can use the same 3D stacked DRAM for their automotive applications , where computations are at high demand such as GPU's. Also, these memories given supports IMC i.e. In Memory Computation.

The detailed comparison of IMC operation on different packaging styles is given in Table 6.

	TSMC	Intel	TSMC	CEA-Leti	Intel	Intel
Product Name	CoWoS	EMIB	InFO	INTACT	Foveros	Co-EMIB
Integrated type	2.5D	2.5D	3D	3D	3D	3D
Interposer type	Passive	Passive	-	Active	Active	Active
Interconnect pitch (μm)	40	55	-	20	36	36
PIM Application	NVIDIA GP100	Agilex FPGA	Apple A10 processor	-	Lakefield processor	Ponte Vecchio GPU
Bandwidth	717 GB/s	896 GB/s	-	527 GB/s	-	2 Tb/s
Power	235 W	-	-	~ 30 W	7 W	600 W
Frequency (GHz)	1.4	1.5	-	1.15	~ 1	1.37
Latency	-	~ 60 ps	-	0.6 ns/mm	-	-
Yield	High	High	High	High	High	High
Reusability	High	High	High	High	High	High

Table 6: Comparison Of In Memory Computation Memory Products

2.1.2 High-Level Architecture

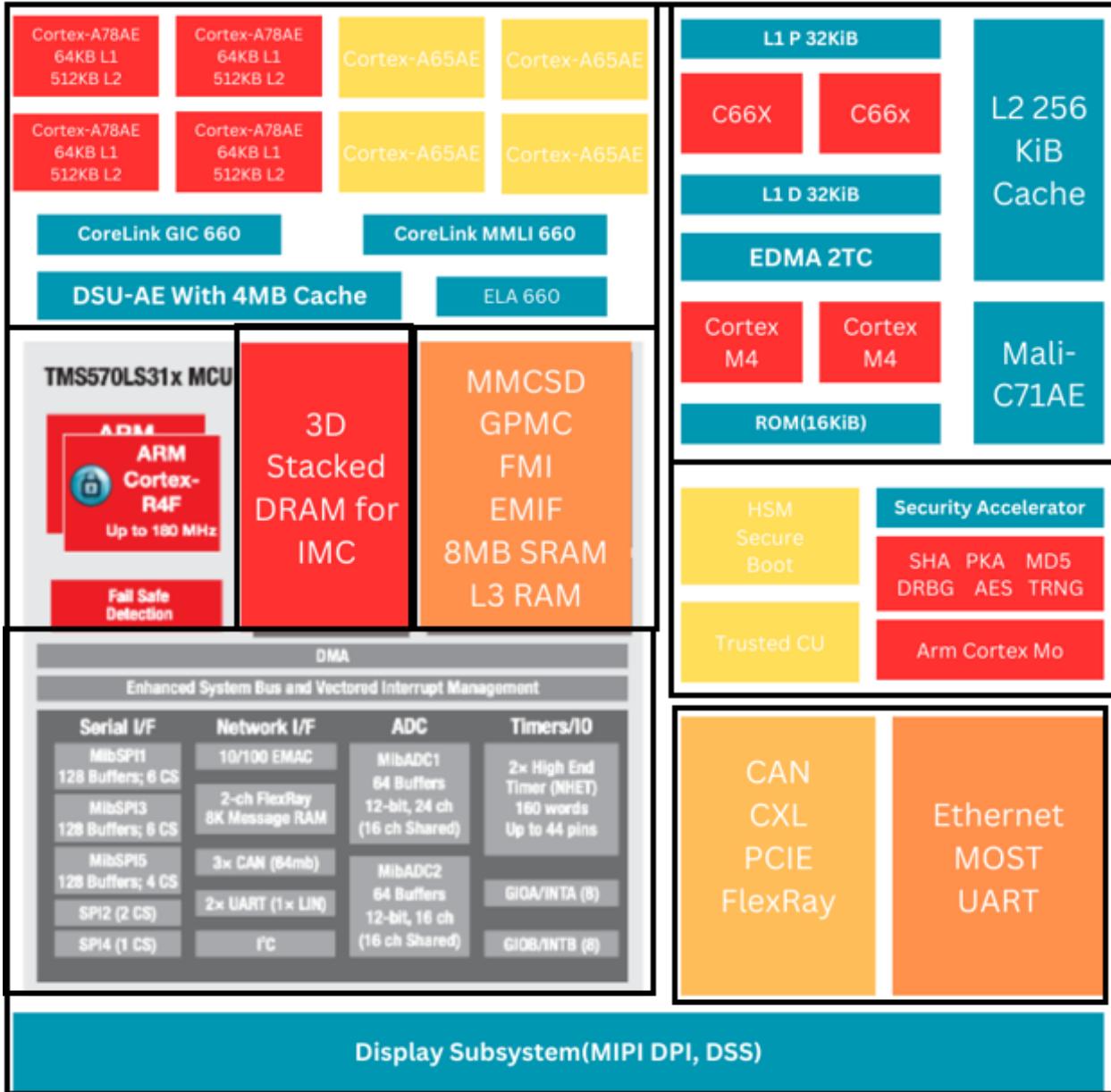


Figure 12: Architecture of ADAS chiplet based system

2.1.3 Partitioning and Floor-Planning

The entire ADAS SoC has been partitioned into 5 chiplets- Core compute cluster, Vision Subsystem, Display Subsystem, Safety Island, Memory and Connection Interface and Security.

The reasons for partitioning the chiplets in the above mentioned parts are:

- **Technology node:** The core compute cluster is the chiplet where most of the computations take place in ADAS. It is required that this chiplet is manufactured on the

latest technology node compared to other blocks to reduce latency, enhance performance and power efficiency. However, it is not cost effective to make all the chiplets on the latest node.

- **Scalability:** The vision subsystem contains the IPs regarding GPU, accelerators, DSP. These IPs keep on updating frequently compared to other subsystems because of new algorithms and architectures. Hence these IPs are combined on a single chiplet so that JLR can just replace this chiplet at every advancement step without tampering other chiplets.
- **3D stacking-** Memory is the only set of chiplets that is 3D stacked because memory is the component of an SoC that takes maximum space. To reduce this space utilization memory is 3D stacked.
- **Security at the doorway:** The external communication interface and security IPs are placed in the same chiplet so that the data collected from the sensors as well as other systems is first passed through security check and then to other subsystems of ADAS. This chiplet is placed at centre of the SoC to minimize the latency offered by security protocols and to minimize the length of UCIE.

Floorplanning

Floorplanning refers to the strategic arrangement and organization of individual chiplets on the interposer and circuit systems in the interposer. It involves determining the physical placement of chiplets and their interconnections to optimize overall system performance, power efficiency, thermal management, and manufacturability. There have been various methods suggested for floorplanning by researchers [29],[30]

Factors considered while floorplanning-

- Optimizing Interconnections: Efficient floorplanning aims to minimize the distance between chiplets to enhance communication bandwidth and reduce signal delays.
- Managing Power and Thermal Considerations: Chiplet placement impacts power consumption and heat dissipation. Floorplanning seeks to distribute heat evenly and manage power efficiently.
- Routing Complexity: Minimizing the complexity of interconnect routing simplifies the overall design and reduces signal propagation delays.
- Manufacturability and Yield: Considerations for manufacturing processes and yield are crucial. Floorplanning can contribute to improved manufacturability and yield by optimizing chiplet placement.

There are various softwares available for floorplanning such as Cadence Innovus [31], Synopsys IC Compiler [32] etc. But these EDA tools are used for SoCs. There have been proposed methods for using these tools for floorplanning [29][30].

In the proposed report thermal management is considered an important factor for floorplanning as explained in Section 4.1. The floorplanning of circuitry in the active interposer is explained in detail in Section 3.2 Figure shows the interconnection of the chiplets.

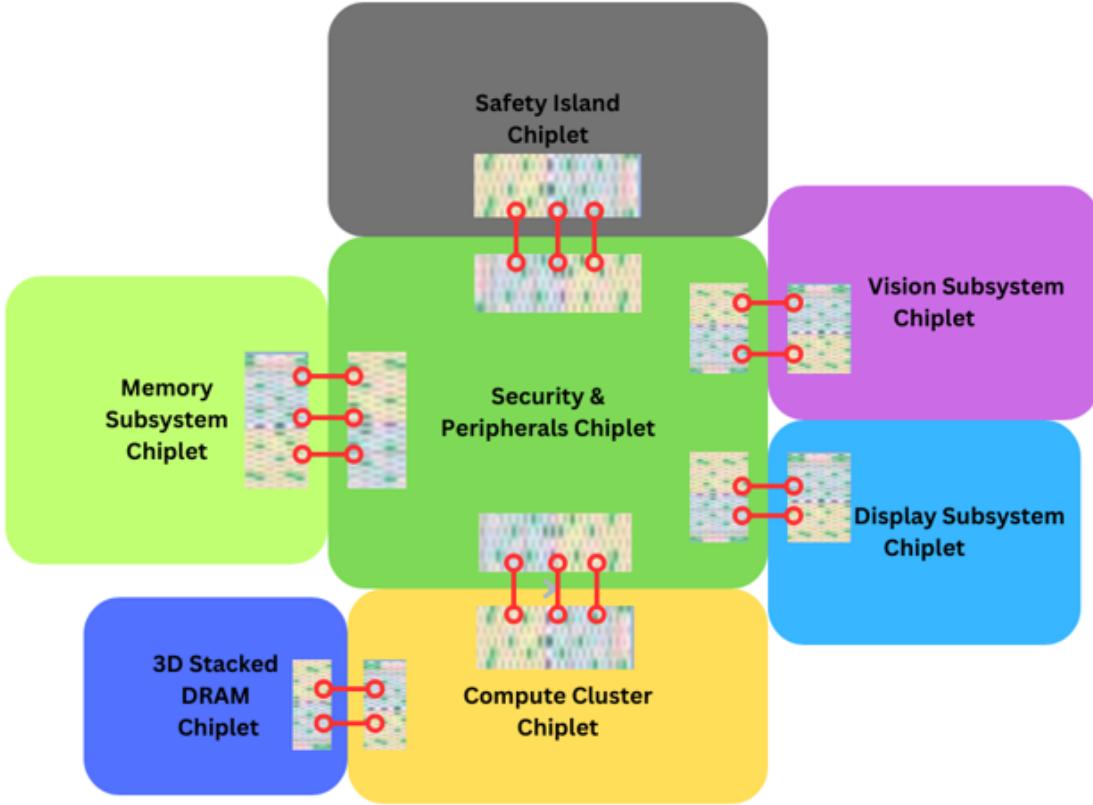


Figure 13: Interconnection between ADAS Chiplets

2.2 Connected Infotainment Subsystem(eCockpit)

Acknowledging the fact that Land Rover Pivi Pro is the brand's exclusive integrated infotainment system featuring everything from entertainment, instrument clusters and even safety features, and the fact that European motoring organization AUTOBEST honored Pivi Pro connected infotainment system as the best connected technology in the industry, awarding it with its SMARTBEST 2020 Award, the proposal is that JLR goes with a Chiplet approach for the same PIVI PRO Infotainment SoC [33].

Although PIVI PRO Multimedia SOC can be integrated for In-Vehicle Infotainment and Instrument cluster and provide various features like : “self learning algorithms” - for map updation, “Smart Voice Guidance”, “Android Auto”, “Apple CarPlay” and “Bluetooth pairing”. The reason why JLR should incorporate a chiplet based technology for PIVI PRO multimedia SoC is :

- As technology advances, newer chiplets with improved computing capabilities on higher technology nodes can be seamlessly integrated into the existing system, enhancing overall performance and responsiveness thus allow for more scalable and modular upgrades.
- Reconfigurable 3D Display: Reconfigurable 3D Display combines both 3D visualization and the ability to dynamically adjust or reconfigure its layout and content. It can also

be upgraded very rapidly. So chiplets dedicated to graphics processing and display control can enhance the visual experience on the infotainment system.

- The modular nature of chiplets will allow for easier upgrades and repairs. Also chiplet-based architecture facilitates customization and can contribute to overall power savings, since power to each tile can be controlled in an efficient manner.

Since the IP for PIVI PRO isn't available freely, based on all the available data, features and references from other automotive infotainment SOCs, a rough sketch of JLR's heterogeneous Infotainment SOC is proposed mentioning where JLR could incorporate chiplet technology for various Subsystems - Compute Clustor(ie. Specifically at the Heterogeneous Compute Cluster), Reconfigurable 3D Display comprising Graphic Units, accelerators, video codec, etc. and rest all parts would be another chiplet.

Below is a general block diagram of how the heterogeneous MultiMedia Infotainment SoC would look like [34] :

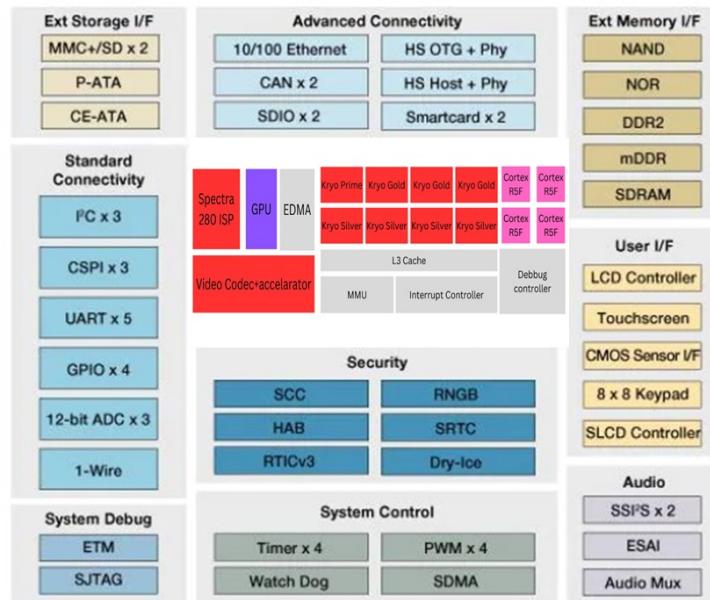


Figure 14: Infotainment

This SoC covers all aspects of a Multimedia Infotainment SoC from Sound System, Connected Radio, Digital Cluster, eCockpit. It uses SnapDragon X20 LTE modem [35] , Qualcomm aqstic Audio [36],

1. Heterogenous Compute Cluster

From the data available, it was understood that JLR uses custom Snapdragon cores in their processor. JLR could incorporate heterogeneous compute clusters based on chiplet technology here. The reason, as technology advances, newer chiplets with improved computing capabilities can be seamlessly integrated into the existing system,

enhancing overall performance and responsiveness. The proposed one is a heterogeneous compute cluster of the Kryo cores alongside an ARM Cortex R5F (Co-Processor). The Kryo cores are TSMC N4 processed on 5 nm technology node in 2023.[37]

(a) Microarchitecture

In the proposed microarchitecture arrangements, like in most Snapdragon SoCs, the Kryo cores are in a big.LITTLE (4+4 or 2+6) configuration. Accommodating Qualcomm's and JLR's flagship, the arrangement proposed is 1 Kryo Prime core + 3 Kryo Gold cores + 4 Kryo Silver cores (DynamIQ). There are also two clusters of ARM Cortex R5F Co processors to offload the main Kryo cores. In addition to this, there is a MMU(Memory Management Unit), Interrupt controller and a Debug controller in the microarchitecture.



Figure 15: Infotainment Compute Cluster

Features and Specifications :

- The Kryo Prime core is based on ARM Cortex X2 and ARMv9.0-A ISA. The Kryo Prime cores are a new addition to the Kryo setup. This is a result of Qualcomm adopting DynamIQ. They offer a huge power boost to the SoCs but also suffer from heating and battery drainages as byproducts of their power.

Performance : Max. CPU clock rate 2.85 GHz to 3.00 GHz

Cache

L1 cache : 128 KiB (64 KiB I-cache with parity, 64 KiB D-cache) per core

L2 cache : 512–1024 KiB per core

L3 cache : 512 KiB – 8 MiB (optional)

Architecture and classification

Microarchitecture ARM Cortex-X2

Instruction set ARMv9.0-A

- The Kryo Gold core is based on ARM Cortex A710 and ARMv9.0-A ISA. The Gold cores are known as performance (big) cores. These cores have a lot of power and can handle heavy tasks depending on their architecture. The downside here is that powerful cores tend to heat up and drain the battery pretty quickly. **Cache**
L1 cache : 64/128 KiB (32/64 KiB I-cache with parity, 32/64 KiB D-cache) per core
L2 cache : 256/512 KiB per core
L3 cache : 256 KiB – 16 MiB (optional)

Architecture and classification

Microarchitecture ARM Cortex-A710

Instruction set ARMv9.0-A

- The Kryo Silver core is based on ARM Cortex A710 and ARMv9.0-A ISA. Kryo silver cores are power-efficient cores. What this means is that these cores aren't very powerful, but they can run basic tasks, stay cool and conserve battery.

Cache

L1 cache : 64/128 KiB (32/64 KiB I-cache with parity, 32/64 KiB D-cache) per core

L2 cache : 128-512 KiB per core

L3 cache : 256 KiB – 16 MiB (optional)

Architecture and classification

Microarchitecture ARM Cortex-A510

Instruction set ARMv9-A

- Cortex R5F co-processor

The ARM Cortex-R is a family of ARM cores implementing the R profile of the ARM architecture, designed for high performance hard real-time and safety critical applications. It is similar to the A profile for applications processing but adds features which make it more fault tolerant and suitable for use in hard real-time and safety critical applications. [38]

Performance

Maximum Clock Frequency - Above 1.4GHz

Technology node - 28nm HPM

Performance 1.67 / 2.02 / 2.45 DMIPS/MHz * 3.47 CoreMark/MHz

Efficiency : From 62 DMIPS/ms **Cache**

L1 cache : 64/128 KiB (32/64 KiB I-cache with parity, 32/64 KiB D-cache) per core

L2 cache : 256/512 KiB per core

L3 cache : 256 KiB – 16 MiB (optional)

Architecture and classification

ARMv7-R architecture MPU with 16

2. Reconfigurable 3D Display

Since JLR is partnering with NVIDIA, they could use NVIDIA Graphic processors. Or they could go for an even improved version of their current Visual Processing System. JLR can incorporate heterogeneous SoC solutions here. The reason, Chiplets dedicated to graphics processing and display control can enhance the visual experience on the

infotainment system. This could include support for higher resolutions, more advanced graphics rendering, and improved support for 3D and augmented reality displays.

(a) Microarchitecture

The proposed microarchitecture arrangement accommodates Qualcomm Spectra 280 ISP, GPU (2 proposals are given- The Qualcomm Adreno 660 and NVIDIA GPU), EDMA (Enhanced Direct Memory Access) and Video codec (CODA988).

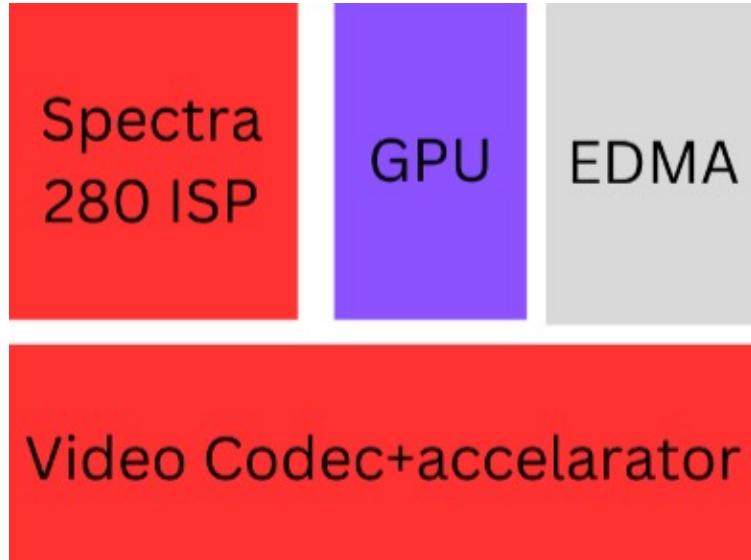


Figure 16: Display Architecture

(b) Display Features

- Up to 4K resolution supporting multiple touchscreen displays
- 4K 60 fps display over HDMI 2.0
- Up to 4K Miracast 2.0 streaming to rear seat entertainment displays
- 3:1 frame buffer compression ratio

(c) Video Codec / Accelerator IP (CODA988) [39]

CODA988 is Full HD multi-standard video IP designed to meet the strong demand for increased digital video quality, higher resolution, and frame rates for consumer multimedia applications. CODA988 is capable of decoding or encoding up to 1080p 60fps for multiple video standards and offers real-time video transcoding or communication. For AVC/H.264, it supports up to 4K(4Kx2K) resolution. Since this IP is designed to optimally share most of the sub-blocks that are used in common for video processing, it provides ultra-low power and low gate count. The CODA988 adds full decoding support for Model-View-Controller (MVC), for stereoscopic 3D experiences, Theora, and WebM (VP8) for HTML5-enabled devices with an internet connection.

Performance

- Encoder

- AVC/H.264 BP/MP/HP 4.2: Max 4096x2304; Min: 96x16; 50Mbps.
- MVC SHP L4.1: Max: 1920x1088; Min: 96x16; 50Mbps
- MPEG-4 SP L3.0: Max: 1920x1088; Min: 96x16; 20Mbps
- H.263 P3 L7: Max: 1920x1088; Min: 96x16; 20Mbps
- Decoder
 - AVC/H.264 BP/CBP/MP/HP L4.2: Max: 4096x2304; Min: 16x16; 50Mbps
 - MVC SHP L4.1: Max: 4096x2304; Min: 16x16; 50Mbps
 - MPEG-4 SP/ASP L5: Max: 1920x1088; Min: 16x16; 40Mbps
 - H.263 P3: Max: 1920x1088; Min: 16x16; 20Mbps
 - VC-1 SP/MP/AP L3: Max 1920x1088; Min: 16x16; 45Mbps
 - MPEG-1/2 MP High: Max 1920x1088; Min: 16x16; 50Mbps
 - Sorenson Spark: Max 1920x1088; Min: 16x16; 40 Mbps
 - VP8 WebM/WebP: Max 1920x1088; Min: 16x16; 20 Mbps
 - Theora: Max 1280x720; Min: 16x16; 20 Mbps
 - AVS/AVS+ Max 1920x1088; Min: 16x16; 40 Mbps

Interface

A 32-bit AMBA3 APB bus

A 64-bit AMBA3 AXI bus (with additional secondary AXI buses)

Features

- Frame buffer compression (CFrame)
- Configurable IP
- Low power consumption
- Rotation and mirroring
- Programmability
- Frame-based processing
- Multi-instances and multi-slice
- Burst Write Back (BWB)
- Downscaler (by on-the-fly mode)
- Map converter

(d) SnapDragon Spectra 280 ISP (Containing Hexagon 680 DSP)

The proposed ISP is the second-generation Spectra ISP, which is a 14-bit dual-ISP, supports up to 25MP @ 30fps with zero shutter lag, greater precision for improved image quality. It also consists of a Hexagon 680 DSP within it.

This makes it particularly adept at handling pics and videos with support for improved multi-frame noise reduction. It also implemented motion compensated temporal filtering (MCTF) and improved electronic image stabilization (EIS).

Other key features include slow-motion video, HDR recording, and high-speed performance capture, and InMotion, a feature that uses computational photography and video capture to superimpose a still image on a moving background. [40]

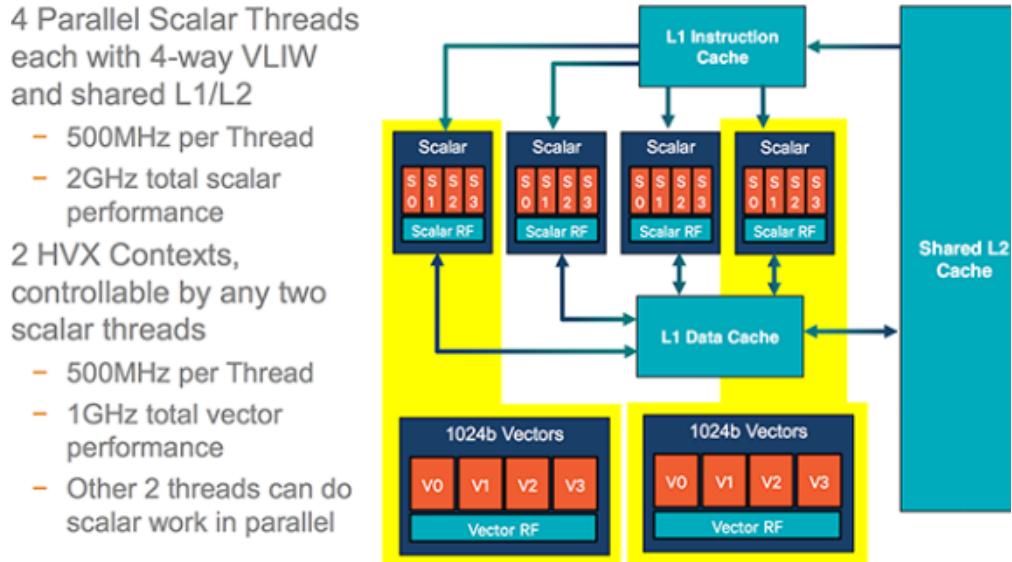


Figure 17: Spectra 280 ISP architecture

3. GPU

For 3D Graphics Rendering, there are two proposals. One is to go with the same Snapdragon's advanced technology node GPU, Adreno 660 [41]. The other proposal is that OEMs adopt the NVIDIA GeForce RTX 3060 Laptop GPU for rendering best 3D Display and Graphics.

RTX 3060 Laptop GPU

The RTX 3060 Laptop GPU incorporates the Ampere microarchitecture. This architecture incorporates 8nm technologies. Even today in 2023 being 2 years old, it would render 60+fps in demanding situations, having the following specs :

GPU Engine Specs

CUDA Cores - 3840

Tensor / AI Cores - 120

Graphics Clock (MHz) - 1283 - 1703

Memory Specs

Standard Memory Config - 6GB GDDR6 Memory Interface Width - 192-bit Memory Bandwidth (GB/sec) - 336 Gbps

Adreno 660

The Qualcomm Adreno 660 is a smartphone and tablet GPU, offers a 35 percent improved performance over its predecessor also giving energy efficiency improvement by 20 percent. The Adreno 650 supports Vulkan 1.1, DirectX 12, OpenGL ES 3.2, and OpenCL 2.0 FP. Furthermore, videos can use HDR10+ and Dolby Vision (with a supported display) [42]. Specifications are given in Table 6.

GPU name	Adreno 660
Architecture	Adreno
Generation	6xx
Uthography	5 nm
Bus interface	IGP
GPU base clock	792MHz
GPU boost clock	905MHz
Memory type	LPDDR5-6400
Memory bus	64 bit
Memory bandwidth	51.2 GB/s
Execution units	2
Shading units	512
Performance FP16 (half)	3.7 TFLOPS
Performance FP32 (float)	1.9 TFLOPS
Performance FP64 (double)	463.3 GFLOPS

Table 7: Adreno 660 specifications

2.3 Other Applications

Body Domain Controller

Premium vehicles today can already have up to 150 million lines of software code, distributed among as many as 100 electronic control units (ECUs) and a growing array of sensors, cameras, radar and light detection and ranging (lidar) devices. The expansion of the automotive industry is reflected in the expansion of functionalities and services offered by a modern vehicle. In order for all of them to be executed, it is also necessary to increase the number of physical components within the vehicle. As the entire system grows, so does the problem of connecting the entities within it. The efficient simplification of the system is achieved by introducing domain controllers, and separating the physical components based on the functionality they perform, while creating the backbone communication network between domain controllers.

Body electronics and lighting systems in a vehicle create comfort and convenience for the driver, passengers and in some cases even those outside of the vehicle. These include:

- Airbag and Crash Detection
- Electric Pumps
- Heating Ventilation and Air Conditioning
- Automotive LED Controllers
- Motor Control
- Advanced Exterior Lighting

NXP Semiconductor has developed the S32K1 Family of 32-bit MCUs for Automotive General Purpose [61]. This microcontroller is used for the above mentioned applications.

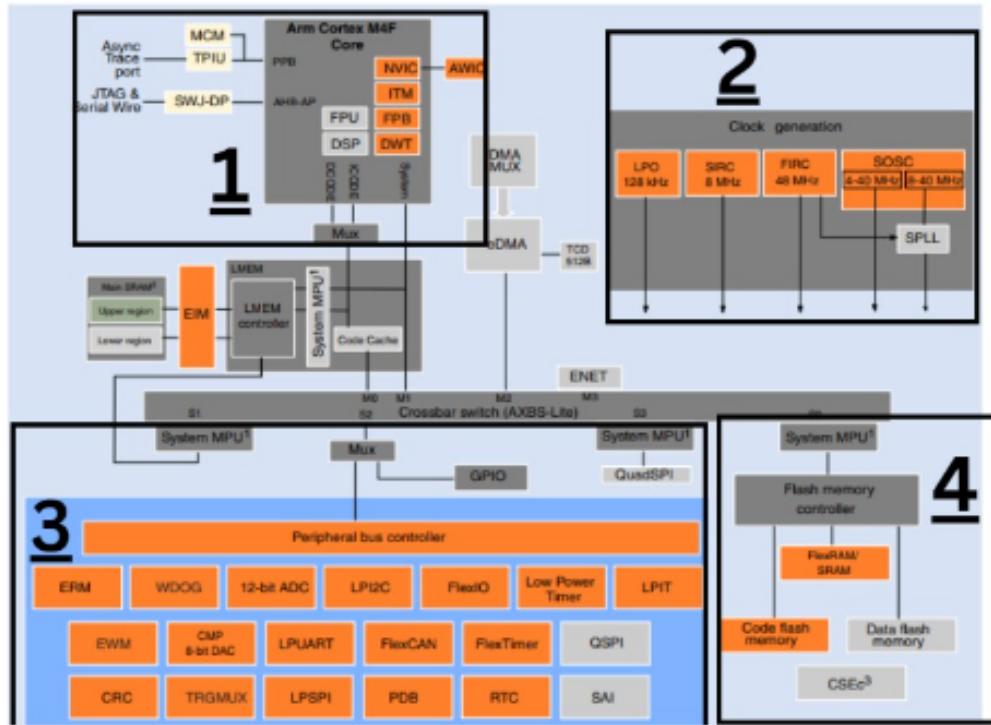


Figure 18: High-level architecture diagram for the S32K1 family [62]

S32K1 microcontroller can be used as an inspiration by JLR to design its own chiplet based Body Domain Controller (BDC). BDC can be divided into 4 chiplets as mentioned below.

- **Arm™ Cortex-M4F/M0+ core, 32-bit CPU (Block 1)**
 - Supports up to 112 MHz frequency (HSRUN mode) with 1.25 Dhrystone MIPS per MHz
 - Arm Core based on the Armv7 Architecture and Thumb®-2 ISA
 - Integrated Digital Signal Processor (DSP) – Configurable Nested Vectored Interrupt Controller (NVIC)
 - Single Precision Floating Point Unit (FPU)
- **Clock interfaces (Block 2)**
 - 4 - 40 MHz fast external oscillator (SOSC) with up to 50 MHz DC external square input clock in external clock mode
 - 48 MHz Fast Internal RC oscillator (FIRC)
 - 8 MHz Slow Internal RC oscillator (SIRC)
 - 128 kHz Low Power Oscillator (LPO)
 - Up to 112 MHz (HSRUN) System Phased Lock Loop (SPLL)

- Up to 20 MHz TCLK and 25 MHz SWD CLK
- 32 kHz Real Time Counter external clock (RTC CLKIN)

- **Communications interfaces (Block 3)**

- Up to three Low Power Universal Asynchronous Receiver/Transmitter (LPUART/LIN) modules with DMA support and low power availability
- Up to three Low Power Serial Peripheral Interface (LPSPI) modules with DMA support and low power availability
- Up to two Low Power Inter-Integrated Circuit (LPI2C) modules with DMA support and low power availability
- Up to three FlexCAN modules (with optional CAN-FD support)
- FlexIO module for emulation of communication protocols and peripherals (UART, I2C, SPI, I2S, LIN, PWM, etc).
- Up to one 10/100Mbps Ethernet with IEEE1588 support and two Synchronous Audio Interface (SAI) modules.

- **Memory and memory interfaces (Block 4)**

- Up to 2 MB program flash memory with ECC
- 64 KB FlexNVM for data flash memory with ECC and EEPROM emulation. Note: CSEc (Security) or EEPROM writes/erase will trigger error flags in HSRUN mode (112 MHz) because this use case is not allowed to execute simultaneously. The device will need to switch to RUN mode (80 MHz) to execute CSEc (Security) or EEPROM writes/erase.
- Up to 256 KB SRAM with ECC
- Up to 4 KB of FlexRAM for use as SRAM or EEPROM emulation
- Up to 4 KB Code cache to minimize performance impact of memory access latencies
- QuadSPI with HyperBus™ support

The proposal to implement domain controllers as chiplets is based on the reasons mentioned in Section 1. Since the above mentioned chiplets have generic subparts apart from the CPU, these IPs can be directly bought off the shelf and JLR can merge them in respective chiplets.

3 Chiplet Integration

Interconnects enable the chiplets to communicate among themselves.

Interconnects consist of the interface(the physical hardware) and the protocol for communicating over it. They include intra-die, inter-die and off-chip interconnections.

Intra-die communication takes place through buses or NoCs(Network on Chip). Buses are the simplest to design and NoCs are the most complex in design. On the other hand buses occupy a large volume whereas NoC takes up minimum space. The various inra-die communication protocols that are used to communicate via these buses/NoCs are:

- AMBA(Advanced Microcontroller Bus Architecture)
- AHB(Advanced High-Performance bus)

- AXI(Advanced Extensible Interface)
- Wishbone Bus
- Open Core Protocol (OCP)
- CoreConnect Bus[30]

As we break down chips into chiplets, the main requirement is to choose the appropriate inter-die communication protocol and interface. The various available inter-die technologies are[31]:

- USR SerDes(Ultra Short Range Serialiser Deserialiser)
- Apple UltraFusion
- AMD Infinity Fabric
- Intel AIB/MDIO
- TSMC LIPINCON
- OCP ODSA BOW
- NVLink-C2C

Since growth is accelerated the most by the open source community, Intel, AMD, Arm, Qualcomm, TSMC and many other industrial giants came together to develop the UCIe. Also as of now, UCIe is one of the best options for interconnects in terms of latency and power efficiency. Hence it is the best practice to use UCIe interconnect technology so that IPs from various vendors can be easily integrated and the maximum benefits of open source community can be achieved.

3.1 UCIe

In the proposed system, UCIe is used as the interconnect between all the chiplets. Universal Chiplet Interconnect Express (UCIe) is an open industry standard interconnect, offering high bandwidth, low-latency, power-efficient, and cost-effective on-package connectivity between heterogeneous chiplets. UCIe is a layered protocol, with each layer performing a distinct set of functions. It is required for every component in the UCIe stack to be capable of supporting the advertised functionality and bandwidth. The fundamental unit of UCIe's interconnect architecture is referred to as a "cluster." Each cluster is composed of N single-ended, unidirectional, full-duplex Data Lanes, where N can be 16 for the standard package or 64 for the advanced package. This composition includes essential elements such as a Valid Lane indicating data validity, a Tracking Lane for synchronization, a forwarded clock, and two lanes per direction for sideband signals. The use of clusters as building blocks provides a modular and scalable approach to chiplet communication within the UCIe framework.[32]

3.1.1 UCIE Background

UCIE consists of three layers as components:[33]

- **Physical Layer**

The physical Link of UCIE is composed of two types of connections:

- **Sideband :** This connection is used for parameter exchanges, register accesses for debug/compliance and coordination with remote partners for Link training and management. It consists of a forwarded clock pin and a data pin in each direction. Each module has its own set of sideband pins. For the Advanced Package option, a redundant pair of clock and data pins in each direction is provided for repair.
- **Mainband:** This connection constitutes the main data path of UCIE. It consists of a forwarded clock, a data valid pin, a track pin, and N Lanes of data per module. For the Advanced Package option, N=64 (also referred to as x64) or N=32 (also referred to as x32) and overall four extra pins for Lane repair are provided in the bump map

- **Physical Layer components**

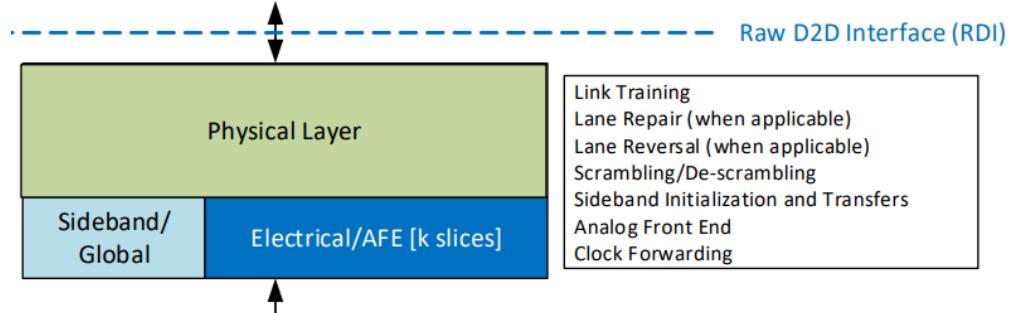


Figure 19: Physical layer components

- **Die to Die Adapter**

The D2D Adapter coordinates with the Protocol Layer and the Physical Layer to ensure successful data transfer across the UCIE Link. It minimizes logic on the main data path as much as possible, thus providing a low-latency, optimized data path for protocol Flits. When transporting CXL protocol, the ARB/MUX functionality required for multiple simultaneous protocols is performed by the D2D Adapter.

- **Protocol Layer**

This layer can be application specific, basically it has three components.[34]

- PCIe from PCIe Base Specification.
- CXL from CXL Specification.
- Streaming Protocol This offers generic modes for a user defined protocol to be transmitted using UCIE.

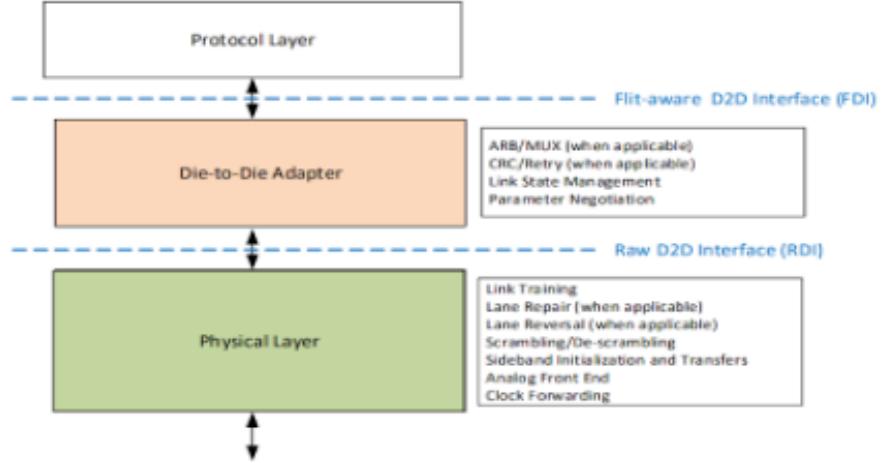


Figure 20: Layers of UCIe

3.1.2 Packaging

In the proposed design, except 3D memory stacking all the components follow the 2.5D package. UCIe Specification does not deal with 3D packaging. It allows two different packaging options: Standard Package (2D) and Advanced Package (2.5D). This covers the spectrum from lowest cost to best performance interconnects. Since the proposed system uses 2.5D packaging, advanced packaging is discussed.

Advanced Package: This packaging technology is used for performance-optimized applications. Consequently, the channel reach is short (less than 2mm, when measured from a bump on one Die to the connecting bump of the remote Die), and the interconnect is expected to be optimized for high bandwidth and low latency with the best performance and power efficiency characteristics. [34]

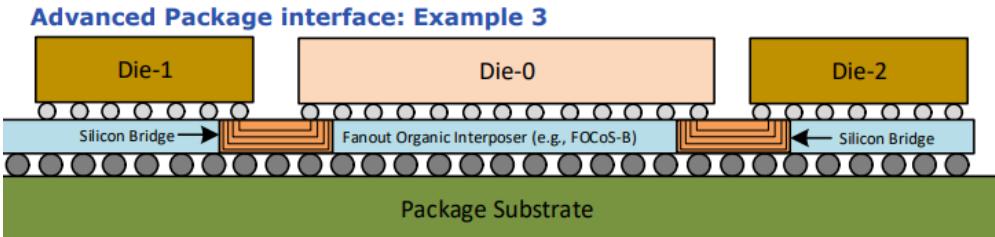


Figure 21: Advanced Package

Index	Value
Supported speeds (per Lane)	4 GT/s, 8 GT/s, 12 GT/s, 16 GT/s, 24 GT/s, 32 GT/s
Bump pitch	25 um to 55 um
Channel reach	2 mm
Raw Bit Error Rate (BER) ¹	1e-27 (<=12GT/s) 1e-15 (>=16GT/s)

Figure 22: Advanced Package Specifications

3.1.3 Connectivity of Two Dies By UCIe

The UCIe PHY is based on a silicon proven die to die PHY design. A single PHY module consists of 32 data lanes sharing one clock lane.[35] Each of the lanes is single ended. The Tx side is simple with a custom-digital serializer followed by a rough-impedance matched Tx driver. The Rx buffer is merely a sized up CMOS inverter. Electrostatic Discharge (ESD) requirements for this design are notably relaxed – for die to die inside the package, after packaging ESD events on the lanes are not possible. Hence the link needs to meet only 30V Charged Device Model (CDM) for ESD as against 500V CDM in AEC-Q100 Automotive ESD requirements. Despite the simple architecture the design is showing excellent eye margins in silicon. Even though each lane transfers 16Gbps individually [35], there is practically no signal integrity issue despite the use of high resistive interposers.

However, the UCIe standard has defined the entire stack of layers to the software level compatibility if both chiplets used CXL or PCIe at the protocol level.

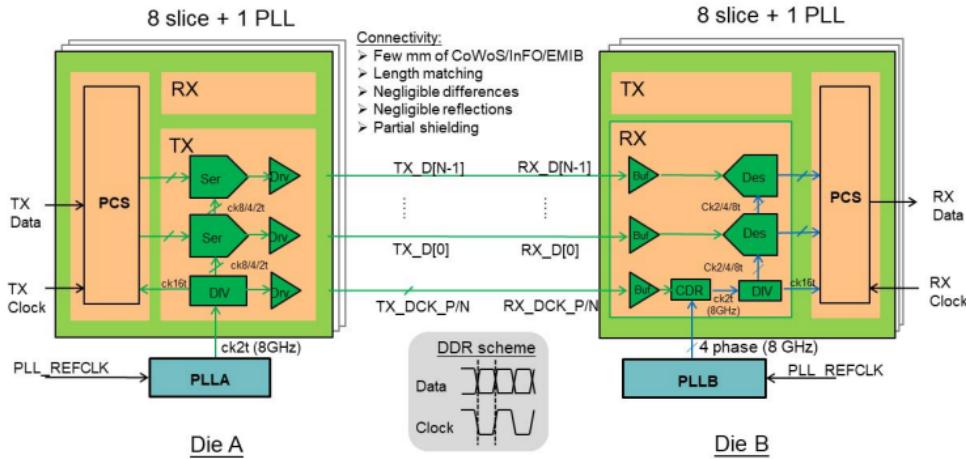


Figure 23: Internal structure of Interconnect

The figure below shows how interconnects/bump maps are connected between the two chiplets.

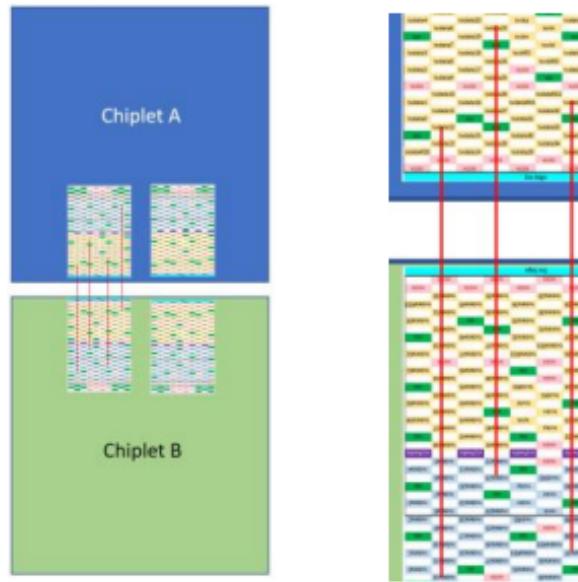


Figure 24: Connectivity of two dies

Briefly, the connections can be shown on a broader view as clear in the image below.

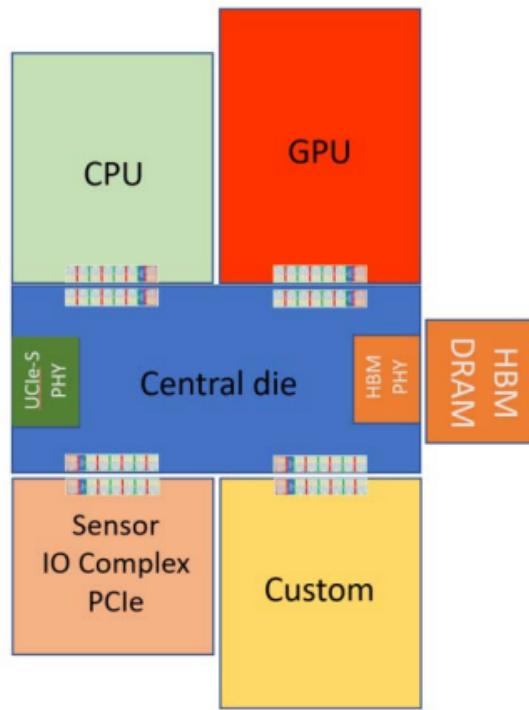


Figure 25: Overall Connections in Chiplet System

3.1.4 Specifications

For comprehensive specifications of the UCIE interconnect, the table below outlines key parameters, including data lane count, clock lane details, and specific ESD requirements. This reference table provides a concise overview of the interconnect's design characteristics for detailed analysis and evaluation. [34]

UCIE Key Performance Targets

Metric	Link Speed/ Voltage	Advanced Package (x64)
Die Edge Bandwidth Density ¹ (GB/s per mm)	4 GT/s	165
	8 GT/s	329
	12 GT/s	494
	16 GT/s	658
	24 GT/s	988
	32 GT/s	1317
Energy Efficiency ² (pJ/bit)	0.7 V (Supply Voltage)	0.5 (<=12 GT/s)
		0.6 (>=16 GT/s)
		-
	0.5 V (Supply Voltage)	0.25 (<=12 GT/s) 0.3 (>=16 GT/s)
Latency Target ³		<=2ns

Figure 26: UCIE Key Performance Targets

3.1.5 UCIE IP's

- **Synopsys UCIE PHY IP [36]-**

Specifications - Data rates up to 16 Gbps per pin , Self-contained hard macro , Self-calibrating and training ,Side band channel for initialization and parameter exchange , Built-in self-test (BIST), internal loopback, and external PHY-to-PHY link test , Flexible configuration: 64 RX and TX pins per module (advanced package) or 16 RX and TX pins per module (standard package) , Built-in test and repair functionality with redundant pins to maximize yield , NS (north, south) orientation, EW (east, west) possible.

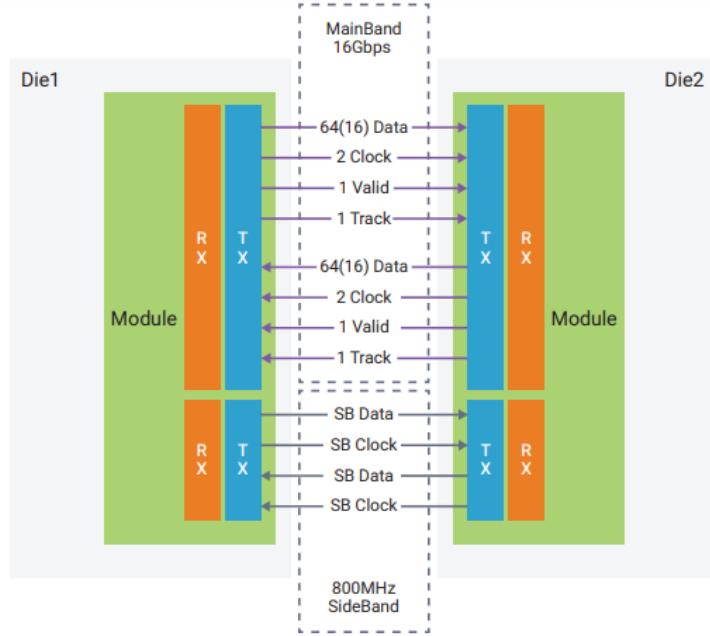


Figure 1: Synopsys UCIe PHY IP architecture (one module)

Figure 27: Synopsis UCIe IP

- **Cadence, UCIe PHY and Controller [37]-**

Specifications - Supports up to 16 Gbps per pin including 4/8/12Gbps, SerDes and DDR architecture, Forwarded clock, track, and valid pins, Sideband messaging for link training and parameter exchange , KGD (Known Good Die) testing capability, Redundant lane repair (advanced), Width degradation (standard), Lane reversal , 2-25mm wide range channel reach, Low raw BER 1e-15.

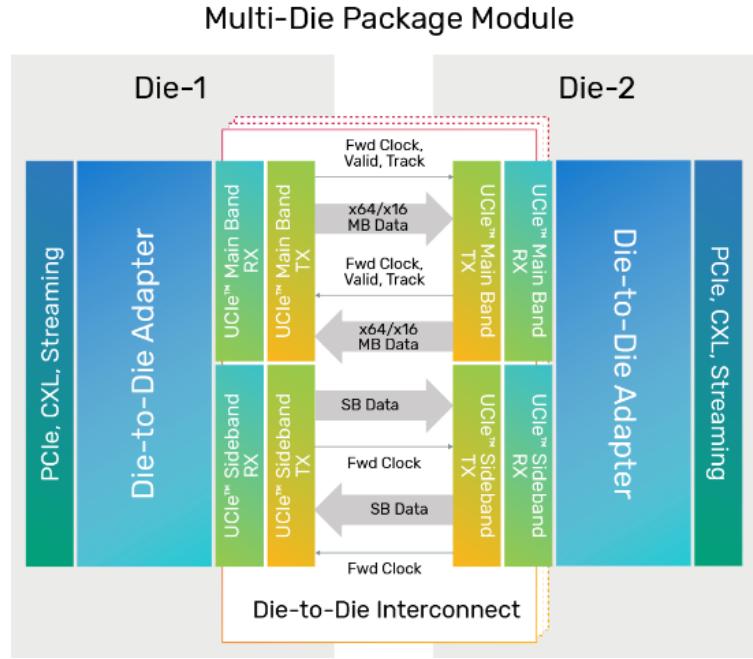


Figure 28: Cadence UCIE IP

3.2 Interposer

In order to assemble the chiplets together, various technologies have been developed and are currently available in the industry:

- Organic Substrate- Used by AMD for their EPIC family
- Passive interposer used in 2.5D packaging. This technology is used by TSMC in CoWoS
- Silicon Bridge embedded within organic substrate used by INTEL in EMIB Bridge.
- 3D Stacking- Implemented in Intel Lakefield processor using Foveros technology.

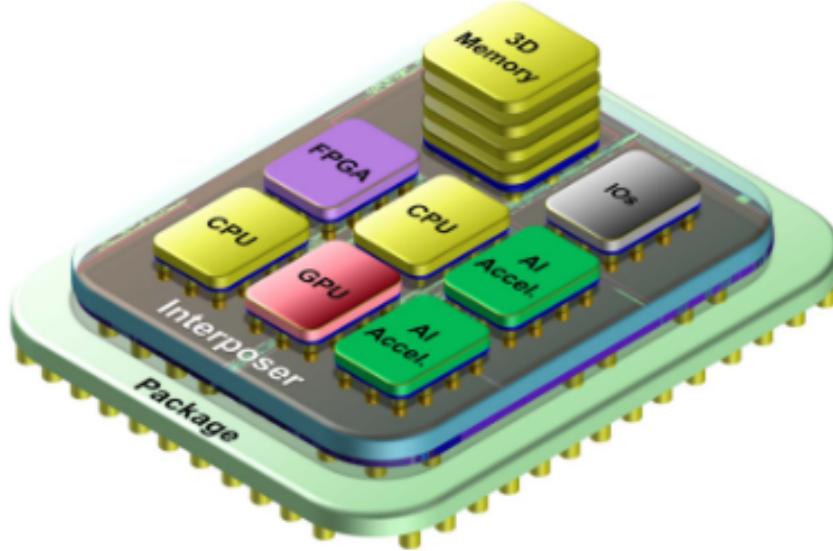


Figure 29: Interposer

But the above technology solutions have challenges:

- Inter-chiplet communication is mostly limited to side-by-side communication, due to wire-only interposer.
- Current interposer solutions do not integrate themselves with less scalable functions, such as IOs, analogs, power management, close to the chiplets.
- It is currently complex to integrate chiplets from different sources, due to missing standards.

The Active CMOS interposer suggested in [38] integrates a scalable and distributed network-on-chip (NoC), which offers the main capability of allowing any chiplet-to-chiplet traffic, without interfering with unrelated chiplets. Additional features can be integrated into the active interposer, to specialize for a given application.

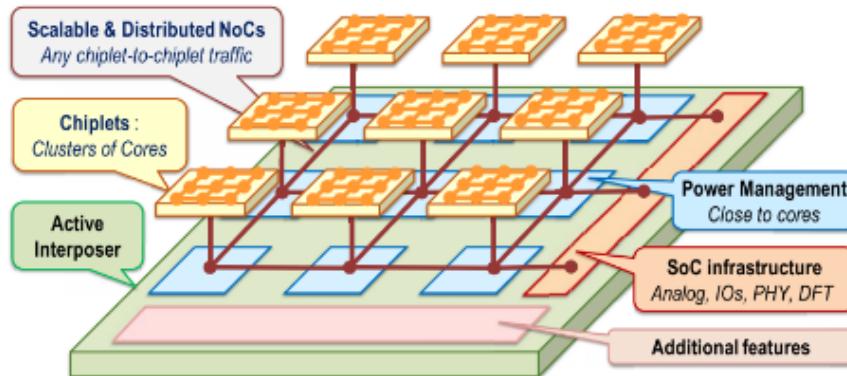


Figure 30: Active Interposer Structure

Physical and Logical components in the Active Interposer

- Power Management System- In order to provide efficient power supply to each chiplet, power management and associated power converters can be directly implemented within the active interposer, to bring power supply closer to the cores, for increased energy efficiency in the overall power distribution hierarchy, and allowing dynamic voltage and frequency scaling (DVFS) scheme at the chiplet level. Switched Capacitor Voltage Regulator has been integrated in the interposer for power management. To allow DVFS per-chiplet, fast transitions, and mitigate IR-drop effects, six integrated VRs have been included in the interposer layer. The SCVR has been chosen thanks to their fully integration capability [39].
- 3D Plug- 3D plugs are interface between chiplet and interposer. Each 3D-plug integrates both logical and physical interfaces. It contains the micro-bump array, micro-buffer cells and boundary scan logic for Design For Test(DFT). [38]
- Interconnect- Different kinds of system interconnects have been implemented between the chiplets on the interposer. More information about the interconnects has been explained in 3.1
- Other IPs- all the less-scalable functions, such as analog IPs, clock generators, and circuit IOs with SerDes and PHYs for off-chip communication, as well as the regular system-on-chip infrastructure, such as low performance IOs, test, debug, and so on, can also be implemented in the bottom die.
- Thermal management Microchannels- For thermal management, microchannels with 20 degree Celsius has deionized water flowing through them. Detailed explanation of this method is given in Section 4.1
- Interposer based Security Enforcement- TransMons are embedded in the interposer to provide security. More about this is explained in Section 3.4

Active interposer should be implemented using a mature technology, with a low logic density to achieve high yield. A difference of at least two technology nodes between the computing chiplets and the interposer should lead to an acceptable cost, while allowing enough performances in the bottom die for analog and PHYs to sustain the overall system performances.

3.2.1 Innovation (Carbon Nanotube)

When it comes to processing power hungry computations the bottleneck can come due to latency involved in communication between chips. Especially for an ADAS, when we attempt to break down the computational units into chiplets the first challenge any OEM is expected to face is in terms of uncertainty of the latency that will be introduced in carrying out the same. The processes which are affected by the latency overhead thus created can be differentiated from the others by using opensource algorithm simulators.

Pylot is an open source platform developed to further the research and development of autonomous vehicles(AV), built with the goal to allow researchers to study the effects of the latency and accuracy of their models and algorithms on the end-to-end driving behaviour of an AV[40]. Pylot allows for an easy integration with CARLA simulator with minimal amount of coding.

The parts of the ML model which gets most affected by latency will be the ones most affected by slight changes in vehicle speeds which can be distinguished by running Pylot either in CARLA or in a testing AV by using drive by wire control interfaced through ROS(Robot Operating System). Then those processes which are affected the most may be referred to as critical processes.

Carbon Nanotubes (CNT) are known for their electrical conductivity. Researchers have found that CNT holds a promising future for lowering the latency for on-chip interconnects[41]. The same can be applied to interconnect lanes as well. For long distance communication, optical cables have proved their clear advantages. But for short distances as those in semiconductors, CNTs remain to be the best option available for effectively reducing the latency along with power consumption[42]. Replacing the interconnect lanes that are quite sensitive to latency with CNTs is a viable option for tackling the latency issues and moving forward to the effective integration of chiplet technology for any mission critical system.

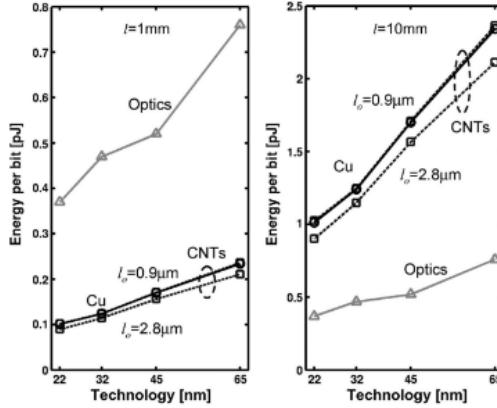


Fig. 8. Energy per bit versus technology node for two different interconnect lengths corresponding to global and semiglobal wire length scales. For CNTs, PD is 33% and the wire diameter d_t is 1 nm. For optics, the capacitance of monolithically integrated modulator/detector capacitance C_{det} is 10 fF [28], [30].

Figure 31: Energy vs Technology Node

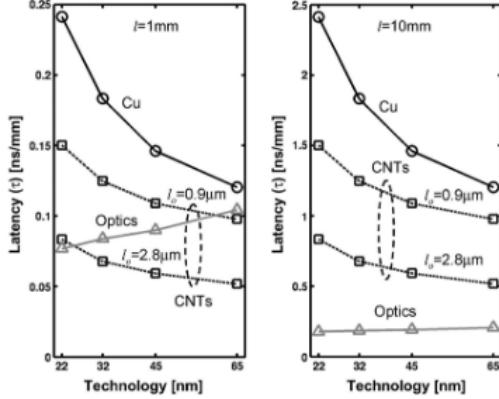


Fig. 7. Latency in terms of technology node for two different interconnect lengths. l_0 is the mean free path and PD is packing density of metallic SWCNTs in a bundle. SWCNT diameter d_t is 1 nm. For optics, the capacitance of monolithically integrated modulator/detector C_{det} is 10 fF [28], [30].

Figure 32: Latency vs Technology Node

3.3 Innovations in package of interconnects

3.3.1 Hybrid of 2.5D and 3D Packaging

2.5D and 3D technologies are the most widely used interconnect technologies alongside 2D and Silicon bridge interconnects. We propose to use a combination of 2.5D and 3D interconnects so that the memory which tends to take up a lot of space in the chip may be stacked up in a 3D stack and the other components which generate a lot of heat may be arranged in a 2.5D fashion so that efficient thermal management can be achieved.[43] It can be said that 3D stacking leads to high bandwidth which cements the use of stacked memory because Adas system requires high bandwidth in memory.

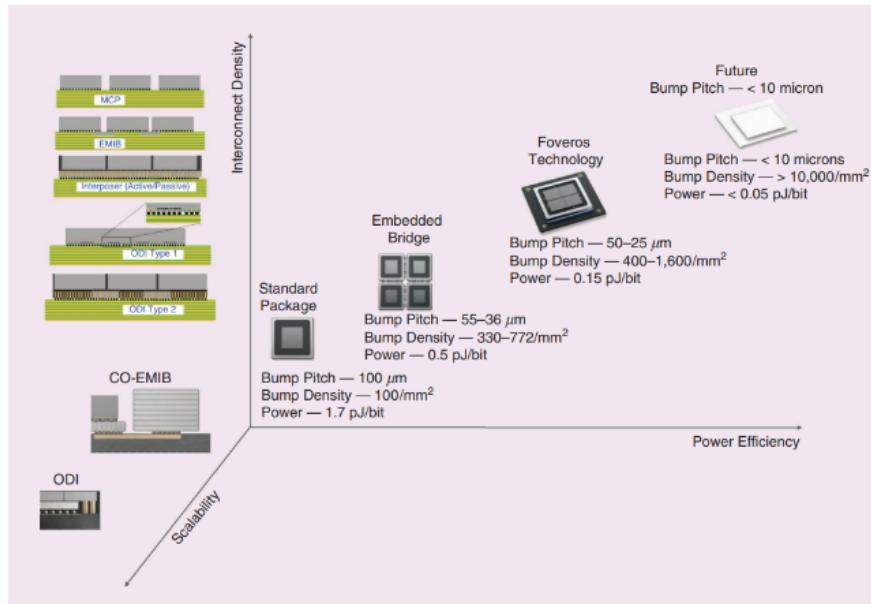


Figure 33: Increasing interconnect density, power efficiency and scalability achieved with 2D, 2.5D and 3D packaging

When shifting from standard package to 2.5D and 3D packages the various leaps in efficiency, scalability and packing density is shown in Fig.30.

TABLE 1. THE MACRO-, MICRO-, AND NANO-3-D TRADEOFFS.				
METRIC	MONOLITHIC 3D	MICRO-3D HB [WoW]	MICRO-3D HB [DoW, DoD]	MACRO-3D //BUMP (E.G., EMIB, FOVEROS)
Interstrata pitch, integration density	High	Medium	Medium	Low
Power density	High	Medium	Medium	Low
Intertier signal delay	Low	Medium	Medium	High
Known good die	No	No	Yes	Yes
Same die size required	Yes	Yes	No	No
Technology maturity	Low	Medium	Medium	High
Heterogeneous process codevelopment	Yes	No	No	No

Figure 34: Packaging Tradeoffs

Intel's Ponte Vecchio uses Co-EMIB, which is the combination of EMIB (embedded multi-die interconnect bridge) 2.5D die-to-die interconnect with Foveros 3D interconnect, to create a high-performance GPU for supercomputers[44].

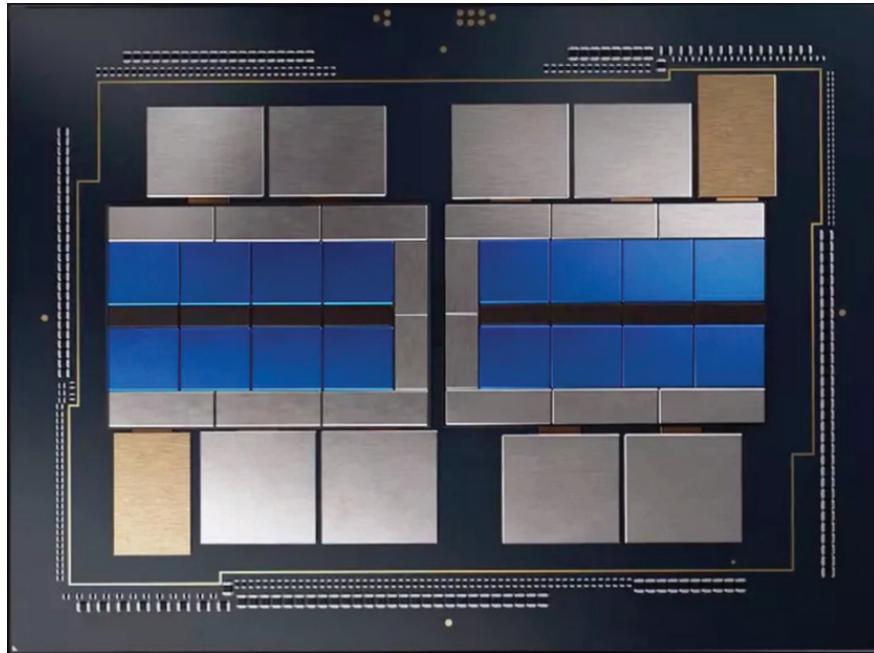


Figure 35: Intel's Ponte Vecchio high-performance GPU for high-performance computing applications utilizes both EMIB 2.5D interconnect and Foveros 3D interconnect

3.4 Safety and Reliability

In the automotive sector, ensuring secure and reliable communication is a critical differentiator, particularly due to safety-critical applications. Unlike other sectors, such as servers where communication tech is essential but doesn't impact lives directly, automotive applications demand a heightened focus on dependable communication.

In today's data-driven landscape, cybersecurity is a major concern, not only for software but also for hardware. The rise of electric cars, equipped with sophisticated technology, intensifies the need for robust cybersecurity. Protecting vehicles from online threats is pivotal for building and maintaining customer trust, influencing a company's growth.

Main Reasons for Cyber Threats:

1. **Off-the-Shelf Untrusted Chiplets:** Chiplet technology allows us to combine chiplets from different vendors. If the chiplet is inherently corrupted by trojans and eludes detection during testing, a latent compromise to its integrity persists. This "Untrusted Chiplet" can breach the security easily if it is integrated in our system [45] and thus can leak sensitive information about the system design. These Untrusted Chiplets can also read and write crucial data like coordinates of vehicle, fuel status and many more hence possessing the cyber threat to the system.
2. **Design Flaw :** Even a minute design flaw in microarchitecture can cost any system dearly. One such example is the ZombieLoad Attack [46] on Intel Processors in which data leak happened through cache memory.
3. **Insidious On-Air Data:** Data received from online weather and traffic sources may contain harmful viruses, threatening the system's integrity.
4. **Hardware-based Phishing:** Manipulating hardware components like keyboards or displays to intercept sensitive information.
5. **Corrupted Interface to System:** Malicious firmware in USB devices that can manipulate connected systems, enabling attacks like data theft or keystroke logging.

Due to these reasons there have been many cyber attacks focusing on the hardware of the system:

Meltdown and Spectre: - Exploit vulnerabilities in modern processors, allowing unauthorized access to sensitive data.

Rowhammer: - Involves repeatedly accessing a specific row of memory, causing "bit flipping" and potential unauthorized access.

Addressing hardware security threats requires a combination of secure design practices, regular software and firmware updates, and user awareness to minimize the risk of exploitation. By considering the above points we suggest Hardware Security Modules (HSM) in the ADAS, Infotainment System and Domain Controller as solutions to reliable and secure communication because as long as we are able to protect the CPU and Memory Unit we will have our data protected from cyber threats.

Threats pertaining to system-level communication and all other associated components must be taken into account while attempting to securely integrate multiple components at the system level. More precisely, the following assaults could be carried out by a malicious chiplet or a chiplet that has been contaminated by viruses[47]:

1. Passive reading also known as snooping, i.e., a chiplet illicitly reads or gathers data that is meant for/authorized to other chiplets.

2. Masquerading also known as spoofing, i.e., a chiplet disguises or poses itself as another one, to illicitly control services or request data from other chiplets.
3. Modifying a chiplet maliciously changes the data exchanged legally between other chiplets.
4. Diverting a chiplet maliciously diverts the data exchanged legally between two chiplets to a third, unauthorized chiplet.
5. Man-in-the-middle a chiplet “hijacks” the communication between two chiplets—this attack is closely related to all four above.

3.4.1 Proposed Solution

The Interposer-Based Security-Enforcing Architecture (ISEA) is a robust hardware-oriented solution designed to safeguard chiplet-based systems not only against untrusted chiplets but also various hardware-focused cyber attacks [45]. This innovative approach is crucial for ensuring the security and integrity of modern chiplet-based designs. Key Features of ISEA:

- Active Interposer

ISEA employs an active interposer, providing enhanced security features. The active interposer includes built-in security measures that effectively counter hardware-based cyber threats [45].

- Holistic Cybersecurity

ISEA is not limited to untrusted chiplets but addresses a spectrum of hardware-related cyber attacks. This comprehensive approach makes it a reliable solution for securing chiplet-based systems [45].

- Avoiding Latency Issues

In the context of security, choosing an active interposer over a passive one is advantageous. The active interposer in ISEA eliminates the need for different IPs (Intellectual Properties) for security in a passive interposer. This strategic choice helps avoid potential latency issues that could arise in a passive interposer design [48][49].

The key paradigms of ISEA are:

- 1) To physically separate commodity components (chiplets in our case) from the HWSFs
- 2) To monitor any memory-related, system level communication at runtime.

Physical Implementation of ISEA: The implementation of Hardware Security Features (HSFs) is of utmost importance, requiring strict isolation from Untrusted Chiplets. This integral security measure is achieved by physically segregating the system-level interconnect fabric, along with its interfaces and HSFs, from untrusted components. Consequently, the components or chiplets are completely oblivious to and insulated from any communication that is not explicitly directed to them or originated within their scope.

Regarding spoofing , we implement a hard-coded assignment of component identifiers (IDs) directly via the interconnect interfaces which reside exclusively in the 2.5D. Thus, a

malicious component cannot masquerade itself as another in the first place.

UCIE bus systems are associated with a master ID: The bus interface ports handle UCIE assignment. The ports that the chiplets are physically connected to are implemented in the trustworthy interposer, not in the chiplet, for ISEA purposes. As a result, when it comes to spoofing, by design, there isn't a surface on which a trojan or other malicious program operating inside the chiplets may exploit in order to change the master IDs.

Interposer-based Security-Enforcing Architecture, or ISEA constitutes the following (Hardware Security Features) HSFs:

- 1) TRANSMONs (inside the active interposer) , along with their Policy Register Spaces (PRSs) to store the various policies;
- 2) an ARM Cortex-M0 core called “PROC-0”;
- 3) Secure Interface (SI).

These HSFs are explained below:

A TRANSMON controls all transactions related to its attached memory chiplet, based on the policies stored in its PRS. A TRANSMON itself comprises three or four components:

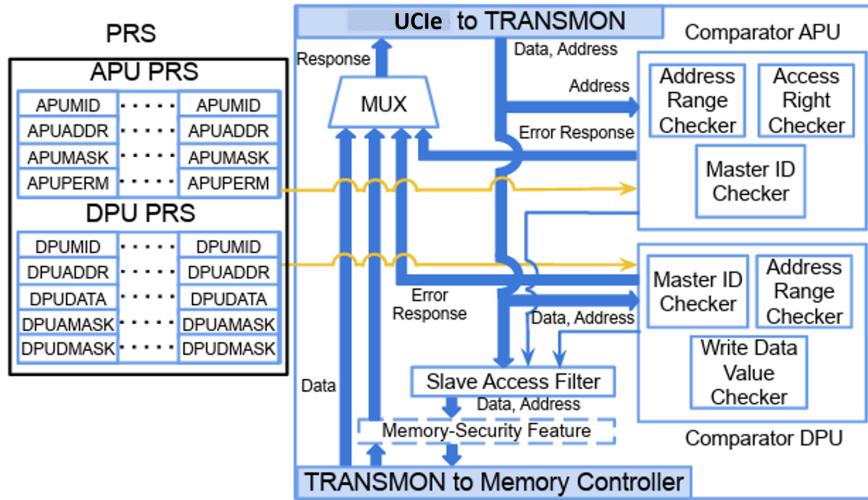


Figure 36: TransMon

- **Address Protection Unit (APU)**

An essential part of the TRANSMON is the Address Processing Unit (APU), which is in charge of carefully examining each read or write memory request and guaranteeing thorough access control over shared memory ranges. The layout guarantees policy checking throughout the UCIE's address phase, removing extra cycle delays. To maximize the effectiveness of policy enforcement, each APU uses its own Policy Rule Set (PRS) to contain policies pertaining to the memory slave it has been given.

- **Data Protection Unit (DPU)**

The DPU in the TRANSMON architecture is integral for data-level protection in the ISEA system. It prevents unauthorized overwrites and restricts the writing of sensitive data, safeguarding assets like cryptographic keys. The DPU implements policies to block transactions related to sensitive data, introducing a slight delay for thorough checks. This meticulous approach ensures the security and integrity of critical information within the ISEA system.

- Slave Access Filter (SAF)

Requests that are granted or rejected are forwarded or rejected by SAF, and harmful alterations and errors inside the memories themselves are detected by the memory-security function.

- ECC

To protect shared system-level memories from faults or malicious modifications, strategies like ECC(error code correction), CRC(cyclic redundancy check), or data mirroring can be used. ECC, using the Hamming code, adds four extra bits per memory byte with a 50 percent increase in memory cost. Despite the cost, ECC provides real-time error detection and correction without latency overhead.

ARM - Cortex M0 (Centralised Core of Management):

The distributed computation is scheduled and managed by the interposer-embedded (and so completely trustworthy) PROC-0, which also allocates and interrupts commodity cores in the untrusted chiplets as needed during runtime. In addition, PROC-0 will map the shared memory spaces at the system level and compile and update the application-specific policy sets that are stored in the PRSs. It is crucial to remember that PROC 0 is only utilized for this type of system-level control and does not function as a "bottleneck" because it is not involved in every UCIe transaction.

Secure Interface (SI):

The task of loading the application or applications and initial data onto the system, as well as collecting the final results from the system, falls to an external Trusted Configuration Unit (TCU). With special access to the UCIe, the SI is used to carry out all of these functions. Note that since we operate under the assumption of a trusted runtime environment, attacks that misuse the TCU or SI in the field are not covered. In any event, cryptographic primitives can be used to secure access to the TCU or SI. To obtain clean data, SI must process all sensor data.

3.4.2 Design of Security System:

For optimal security measures, an individual TRANSMON is strategically placed between each memory slave and the UCIe bus interface .

SI is placed before I/O ports.

ARM Cortex M0 is placed inside interposer.

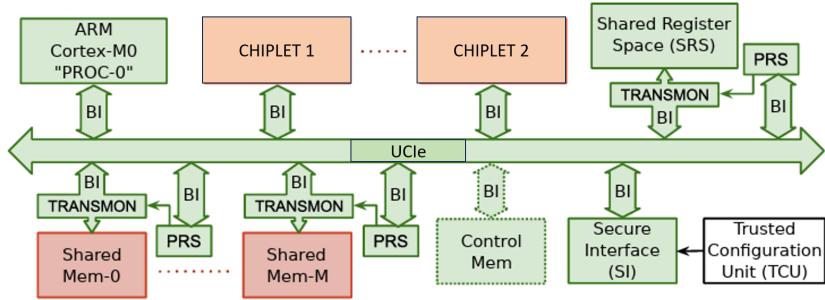


Figure 37: Design of Security System

Working Principles:

- Each APU and DPU has its own Policy Rule Set (PRS) implemented using flip-flops for efficiency. APU PRS defines policies for specific regions in the system's shared memory space, physically allocated in the connected memory slave. Each DPU PRS entry defines a policy for specific data.
- Both APU and DPU policies block read or write requests violating their rules, and TRANSMONs also block requests not matching any policy, providing protection against unauthorized requests.
- Policy verification includes master/slave ID checks, preventing spoofing, snooping, or data manipulation.
- Blocked requests trigger an error message to the initiating master and an interrupt to the trusted PROC-0. The related memory access is dropped by the SAF, ensuring it never reaches the memory.
- PROC-0 in ISEA handles mapping shared memory spaces, compiling/updating application-specific policies for each TRANSMON, and clearing related memory regions after application completion to prevent data leakage.
- The trusted end-user can implement software-level analysis and management of blocked requests, potentially isolating repetitive triggering masters to mitigate denial-of-service attacks.
- Also if any data is transferred to UCIe data bus it has to go through SI for verification.

LATENCY RESULTS:

The following are latency results based on a standard model proposed in [45]

Metrics	Non-Secure (2.5D)	Secure (2.5D)
Critical Delay (ns)	9.72	9.83
Power Consumption (mW)	266.4	300.9
Standard-Cell Area (μm^2)	24,588,292	26,844,473
Total Die Area (μm^2)	33,641,866	33,641,866
Interposer Die Area (μm^2)	6,237,600	6,237,600
Total Instance Count	569,574	745,693
Interposer Instance Count	69,742	249,085
Total Buffer Count	141,151	169,344
Total Wirelength (m)	30.5	40.5
Total Capacitance (nF)	7.92	10.89

Table 8: 2.5D Implementation Results for Non-Secure Versus Secure Designs, Chiplets in GlobalFoundries 65nm, Interposer in Synopsys SAED 90nm

Our suggestion is to make TCU on chiplet and rest of the parts,i.e. all the TRANSMONs to be included in ACTIVE INTERPOSER. The reason for doing so is to connect TCU from Connectivity Peripherals,HSM and Security Accelerator and this can be done efficiently with Chiplets while providing great yield. The issue of latency is due to TCU IP being on chiplet instead of active interposer ,is mitigated by Security Accelerator.

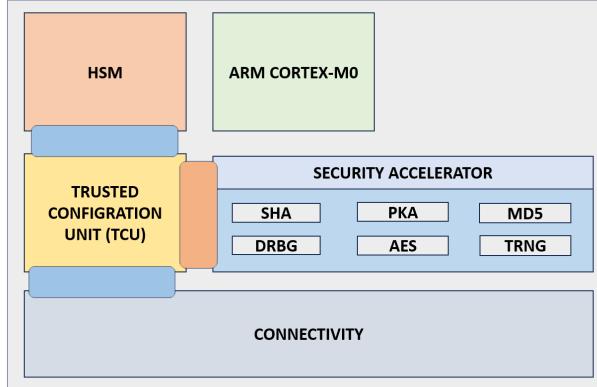


Figure 38: Chiplet Architecture

- As suggested previously, data will pass through TCU.
- Data will be Encrypted by TCU along with the help of Security Accelerators for less latency.
- Cryptographic Keys after encryption will be stored in HSM and will later be accessed at the interface of Shared Memory .
- ARM Cortex will handle all the PRSs of TRANSMONS.

4 Thermal Management

4.1 Interposer

As the integration of multiple chiplets on a single package becomes increasingly prevalent, the challenges associated with thermal management gain prominence. Efficient thermal management is crucial for ensuring the reliability and optimal performance of chiplet-based SOCs, particularly as these systems often entail diverse functional components with varying heat dissipation profiles. The upcoming sections explain the proposed solutions.

4.1.1 Microchannels

Managing heat dissipation in chiplets within 2.5D and 3D packaged systems poses a significant challenge. Stacked-die structures and varying thicknesses of dies amplify the importance of secondary heat flow paths. While using thermal interface materials (TIM) with different thicknesses can be a solution[51], it increases partial thermal resistance and elevates junction temperatures. An alternative approach is to integrate microscale cooler structures within the interposer or package substrate.

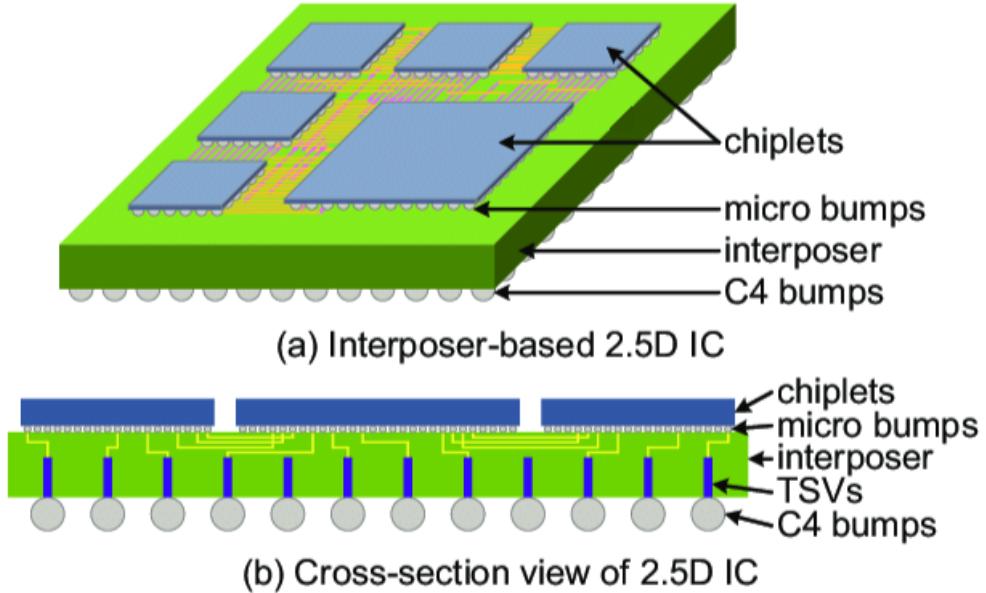


Figure 39: Interposer Design

For thermal management, G. Bognár.[50] suggests microchannels with 20 degree Celsius deionized water flowing through them. The thermal resistance of the secondary heat-flow path depends on several parameters:

- the material and dimensions of the interposer and package substrate
- the material and dimension of the joints between the chiplets and interposers
- number per unit area of the joints (bumps, c2c)

- number per unit area of through silicon via (TSV)

Moreover, while applying integrated microscale channel structures, other parameters will also influence the thermal conductivity of the secondary heat flow path:

- dimension and cross-sectional shape of the channels
- number and length of the channels
- material properties of the applied fluid, such as dynamic viscosity, density, thermal conductivity
- the volumetric or mass flow rate of the fluid

The solution is implemented as:

1. Microchannels are placed at the bottom of the interposer as some part of it contains a physical layer of UCIe.
2. Microchannels do not touch Copper vias (TSVs) but pass between them. This ultimately results in cooling and thus increasing electrical conductivity and thermal gradient between the microchannels and TSVs ,thus resulting in increased heat dissipation.
3. Components like CPU or GPU which dissipate more heat (approx 30W) are to be placed at the microchannel inlets, and components like memory which dissipate less heat (approx 5W) are to be placed at the outlet.
4. Above placement will maintain more thermal gradient between CPU, GPU and microchannels for better thermal management. But memory does not emit that much heat as of CPU or GPU, thus it can be compromised to be placed at outlet position with less thermal gradient(water will heat up from 20 degree to higher at outlet ,as it moves through interposer).

Thermal Distribution of SoC due to Microchannel: Chiplets positioned at the inlet exhibit lower temperatures than those at the outlet, which may pose occasional challenges. To handle it another parallel but opposite flow of heat can be applied to half of the channels

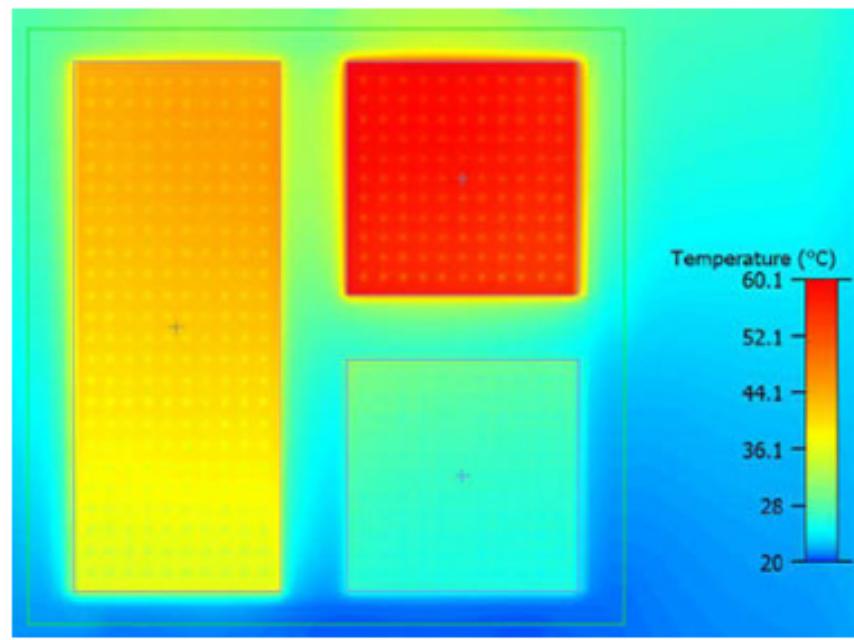


Figure 40: Temperature Distribution Along Dies ,Source:[50]

In addition, since the heat transfer efficiency is exponentially related to the volumetric flow, the temperature starts to increase significantly if the volumetric flow rate drops below a certain level.

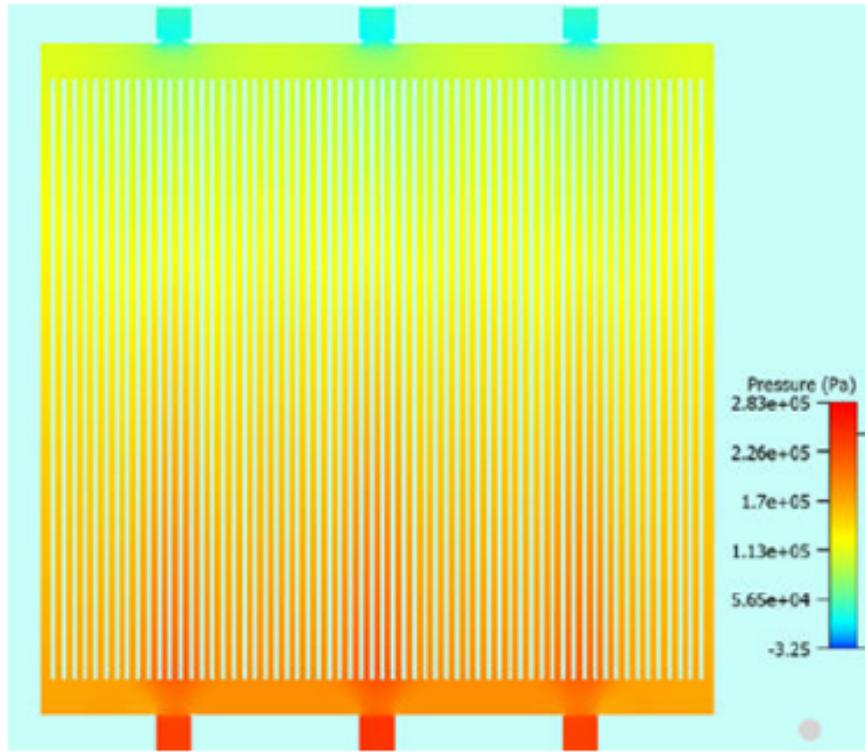


Figure 41: Pressure Drop across the Microchannels ,Source:[50]

4.2 Si-Diamond composite heat sink

The constant drive for a fully fledged autonomous vehicle has made it crucial to have high performance processors that enable large parts of self driving features to be implemented. With great processing power comes immense heat generation. Research has shown that the heat generated in processors is not uniformly distributed. Instead, there are certain areas of hotspots which rise up to 8 times the usual temperature of the other parts of the chip. The commonly deployed cooling techniques fail to properly cool these hotspots resulting in a lowered performance and lifetime of the processor.

Hence various non uniform cooling techniques like thermoelectric coolers were explored but its widespread application is limited due to the low coefficient of performance, complex design, and contact parasitic resistance.

Diamond is among the most conductive materials for heat transfer.[53] Since natural diamonds are expensive, synthetic diamond obtained via chemical vapour deposition is utilised to form a silicon-diamond composite material which serves as a heat sink for effectively cooling the hotspots.

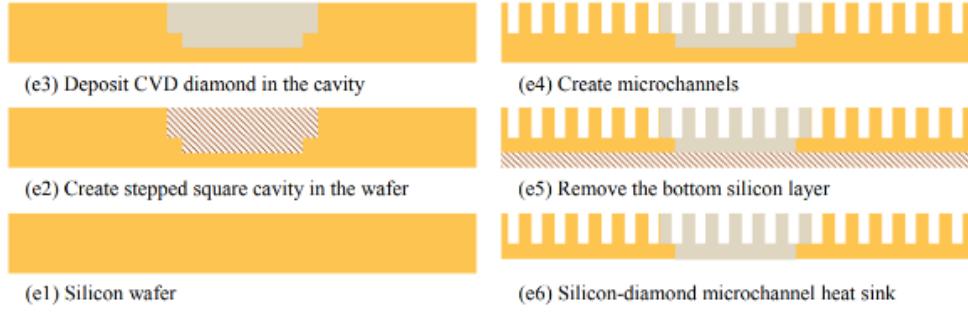


Figure 42: Manufacturing-process[52]

Fig.42 shows the manufacturing steps via the chemical vapour deposition method.(from e1 to e6)

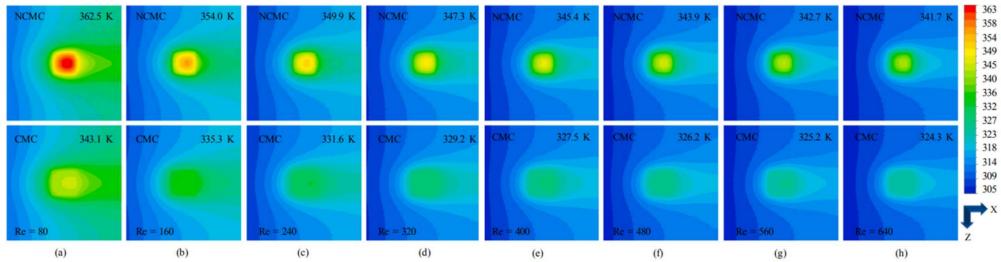


Figure 43: Advantage of Silicon Diamond composite(CMC) heatsink over Non Composite Microchannel Heatsink(NCMC)[52]

In Fig.41, the top layer consists of the results obtained from a conventional Non Composite Micro-channel(NCMC) heatsink whereas the bottom layer shows the results obtained from a Composite Micro-channel(CMC) heatsink, specifically the Silicon Diamond composite micro channel heatsink under the same working conditions.

Researchers were able to bring about a cooling of upto 1600W/squared cm.[52] The composite design exhibited a 48.0 precent reduction in the nonuniformity and 41.7 percent reduction in the thermal resistance for certain flow velocities of the coolant through the microchannel heat sinks.

5 Simulation

5.1 Simulation introduction

In an attempt to back up the proposed technology, i.e. ADAS sensor fusion chiplet system, we tried to simulate a simpler model of our system and obtained the results (latency and power consumption).

Other proposals like UCIE interconnect, Si-Diamond composite heat sink and microchannels (Innovation), carbon nanotube(Innovation), connectivity and security, etc. were backed up with experimental results.Rest all are backed up with literature data.

5.2 Gem5 and Heterogarnet

For the simulation, gem5 [Amalgamation of m5 and GEMS] simulation software was used.

- gem5 is a modular **discrete event driven simulator**. (ie, all the actions in the system are represented as events that are time stamped. These events are time stamped and are stored in an event que. As per the time, an event gets scheduled). Gem5 can execute both user level code as well as kernel codes, ie. It is a **full system simulator**.
- gem5 simulates a machine known as a “simulated machine” on the “host machine” (User Computer). It is written in C++ and python with all experiments being executed on the simulated machine.
- gem5 takes a simulation script which is written in python as input and the simulated machine’s behavior written in C++. This simulation script, ie. se.py has two parts :
 - Configuration phase : This says what is the configuration of the simulated machine, ie. simulated machine specifications.
 - Simulation Phase : This phase tells how to build the machine, ie. requirements for our host machine.
- Simulated machine specifications (in python) will be converted into C++ struct through SimObjects and passed to C++ objects. These SimObjects generally represent some physical component such as cache, main memory or dram.

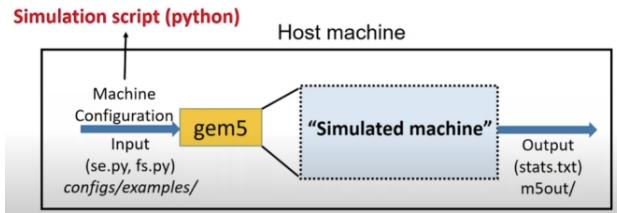


Figure 44: gem5 simulator workflow

gem5 operates on two modes : System Emulation (SE) mode, Full System (FS) mode.

gem5 can model two CPU models : In-order, Out-of-order.

gem5 can have two memory models : Classic memory model, Ruby memory model.

gem5 has two interconnection networks : Simple , Garnet

5.2.1 Garnet

Garnet2.0 is a detailed interconnection network model inside gem5. Garnet2.0 provides a cycle-accurate micro-architectural implementation of an on-chip network router. It leverages the Topology and Routing infrastructure provided by gem5’s ruby memory system model. The default router is a state-of-the-art 1-cycle pipeline. There is support to add additional delay of any number of cycles in any router, by specifying it within the topology.

Garnet provides the following features to model interconnects:

- ni-flit-size: flit size in bytes. Flits are the granularity at which information is sent from one router to the other. Default is 16 (128 bits).
- vcs-per-vnet: number of virtual channels (VC) per virtual network. Default is 4. This can also be set from the command line with `-vcs-per-vnet`.
- buffers-per-data-vc: number of flit-buffers per VC in the data message class. Since data messages occupy 5 flits, this value can lie between 1-5. Default is 4.
- buffers-per-ctrl-vc: number of flit-buffers per VC in the control message class. Since control messages occupy 1 flit, and a VC can only hold one message at a time, this value has to be 1.
- routing-algorithm: 0: Weight-based table (default), 1: XY, 2: Custom.

Garnet2.0 can also be used to model an off-chip interconnection network by setting appropriate delays in the routers and links.

5.2.2 Heterogarnet

HeteroGarnet allows simulating modern interconnection networks such as 2.5D/3D NoCs, chiplet-based architectures and photonic interconnects. It also allows to integrate the networks with the detailed CPU and GPU models of gem5 and simulate full-systems.

HeteroGarnet(Garnet 3.0) improves upon the widely-popular Garnet 2.0 network model by enabling accurate simulation of emerging interconnect systems. Specifically, HeteroGarnet adds support for:

1. **Clock-domain islands:** Garnet 3.0 introduces a Clock Domain Crossing (CDC) unit, featuring FIFO buffers for communication between different clock domains. It offers flexibility in placement, configurable delay, and dynamic latency calculation based on connected clock domains. This enhancement enhances simulation accuracy, particularly in modeling scenarios involving Dynamic Voltage and Frequency Scaling (DVFS) techniques.
2. **Network interface controllers:** The network interface controller (NIC) is an object which sits between the network end points (e.g., Caches, DMA nodes) and the interconnection system. The NIC receives messages from the controllers and converts them into fixed-length flits, short for flow control units. These flits are sized appropriately according to the outgoing physical links. The network interface also governs the flow control and buffer management for the outgoing and incoming flits. Garnet 3.0 allows multiple ports to be attached to a single end points. Thus, the NIC decides where a certain message/flit must be scheduled.
3. **Serializer-Deserializer component:** Garnet 3.0 addresses the crucial need for modeling System-on-Chips (SoCs) and heterogeneous architectures by introducing the Serializer-Deserializer (SerDes) unit. This feature is essential for supporting various interconnect widths across the system, accommodating scenarios where links between components differ in width. For instance, a link between two routers within a GPU and

a link between a memory controller and on-chip memory may have different widths. The SerDes unit, depicted in the provided figure, plays a central role in this configuration. It is designed to convert flits into appropriate widths at bit-width boundaries, allowing seamless communication between components with varying link widths. Similar to the Clock Domain Crossing (CDC) unit described earlier, the SerDes units are versatile in instantiation and can be placed anywhere in the Garnet 3.0 topology.

5.3 Garnet Synthetic Traffic simulation

In a chiplet system, the features of the interconnect such as it's topology, router's and link's latency, link's bandwidth, routing algorithms etc are what mainly influence the performance of the system.

So, simulation is necessary to gauge the performance metrics of the interconnect network and to change the adjustable parameters such as topology, link's bandwidth etc to achieve optimal performance. We use the garnet feature in the gem5 simulator to simulate the interconnect behaviour.

Garnet synthetic traffic provides the framework for simulating, testing, debugging Network on chip (NoC) and newer release garnet 3.0 has features to simulate chiplet interconnects. It is a network only simulation with synthetic traffic, no other IPs are simulated.

The configuration python files to run this simulation are provided in the gem5 library at:

```
./configs/example/garnetsynthtraffic.py
```

The user must specify parameters describing the interconnect at the command line while running the above file. At command line we provide three sets of parameters:

1. **System Parameters:** They determine the type and number of nodes in the network (i.e. how many inject and how many take data out of the network), network model, topology etc.
2. **Network Parameters:** They include parameters such as link bandwidth, latency, router latency etc.
3. **Traffic injection:** In this simulation synthetic data is injected into the network, these parameters model that model synthetic traffic are injection rate, synthetic traffic type etc.

The description of each parameter can be found at [S11].

5.4 Simulation Parameters

We ran the simulation of an simplified NoC, with the following parameters:

Parameter	value
num-cpus	2
num-dirs	2
network	garnet
topology	Mesh-XY
mesh-rows	1
sim-cycles	1000
synthetic	uniform-random
injection rate	0.01
link-latency	3
link-width-bits	128
router-latency	2
vcs-per-vnet	5

Table 9: Simulation Parameters

The simple network on chip simulated consists of 2 CPU cores and 2 cache directories connected by a network modelled by garnet model, having a MESH-XY topology. Mesh-XY topology requires the number of directories to be equal to the number of CPUs. The number of routers/switches is equal to the number of CPUs in the system. Each router/switch is connected to one L1, one L2 (if present), and one Directory. The synthetic traffic model is uniform random i.e. every cycle, each CPUs performs a Bernoulli trial with probability equal to injection rate to determine whether to generate a packet or not.

The configuration of the simulated NoC is as follows:

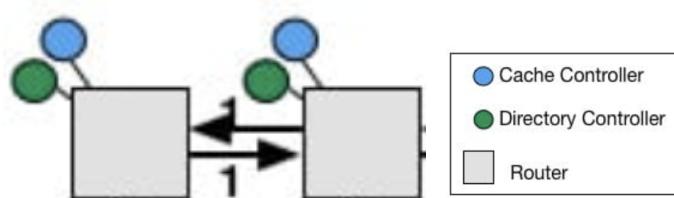


Figure 45: Model of NoC simulated

5.5 Results

The results of the simulation are dumped by the gem5 simulator in the stats.txt file and config.ini file. Please find the simulation config file and result files attached with the report.

1. config python script - garnet-synth-traffic.py
2. stats.txt
3. config.ini

5.6 Alternative Simulation attempted

We also tried to simulate the simplified version of the main functional units of the ADAS system i.e. we attempted to simulate an ARM CPU core connected to a DDR3 memory controller through heterogarnet network and run this system on a predefined workload.

5.7 Future Simulation work

We propose to simulate the interconnect network of the ADAS chiplet in a similar way in the future. Gem5 provides the ability to simulate complete computer systems, therefore we propose to simulate the main functional units of the proposed ADAS chiplet system i.e. compute cluster chiplet, vision subsystem chiplet (i.e. mainly gpu), memory interfaces and on-chip memory chiplet, with a simple predefined workload provided by the resources package of the gem5 library to gauge the performance of the proposed ADAS chiplet.

6 Future prospects

6.1 Vehicle to Everything(V2X) Chiplet

Autonomous systems are incomplete without having multiple intelligent systems which are connected to each other and sharing useful information. This information assists the ADAS and serves as a secondary sensor. Thus the future form of this connectivity is termed as Vehicle to Everything communication or V2X in short. As this is a future prospective upgrade for the ADAS communication, it is best to have a separate empty slot for the chiplet so that it can be upgraded in the future.

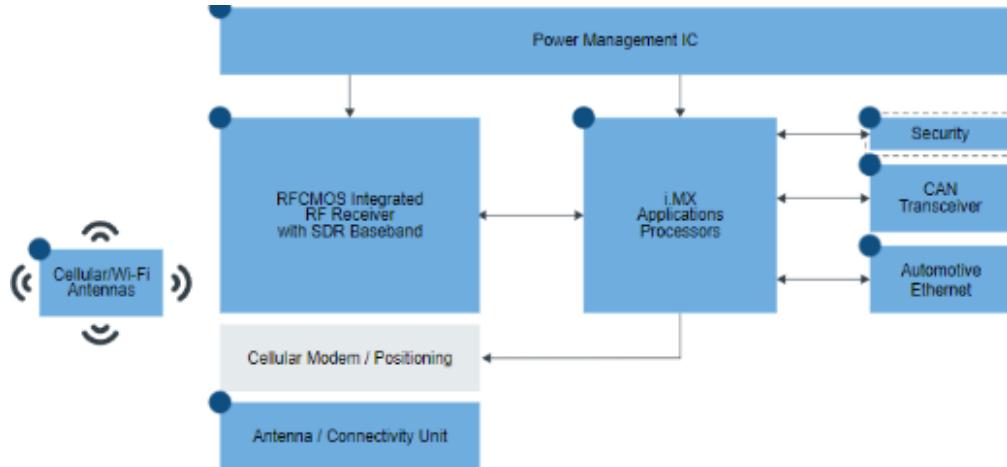


Figure 46: **V2X Block Diagram**[54]

Vehicle-to-Everything (V2X) technology enables cars to communicate with their surroundings and makes driving safer and more efficient for everyone. By making the invisible visible, V2X warns the driver of road hazards, helping reduce traffic injuries and fatalities. In addition to improving safety, V2X helps to optimize traffic flow, reduce traffic congestion

and lessen the environmental impact of transportation.[54]

Hence this enables OEMs to have various models with or without V2X support. Also since communication technology is one the fastest growing fields(proven by the fast adoption/upgradation to 5G and 6G communication technologies and frequent updates to bluetooth and wifi technologies) it is quite convenient to have a dedicated chiplet for encompassing all these technologies that help in connecting future cars so that seamless upgradations can be easily achieved.

6.2 Moving Towards Complete Autonomy

Now acknowledging the fact that JLR has its Jaguar I-Pace for the Waymo Autonomous car[57], the given proposal aims to :

- minimize the amount of sensor use (Thus cutting cost).
- Improve the efficiency by incorporating better hardware for efficient algorithms.

The proposed Idea is to design a comprehensive tile or chiplet for Advanced Driver Assistance Systems (ADAS) that moves towards full autonomy. This is a complex yet crucial endeavor in the automotive industry. Thus, this indeed needs to be developed more in the future. The integration of these advanced technologies promises to propel ADAS systems toward a future where vehicles can navigate and interact with their surroundings with heightened precision and safety.

This block would :

- Further enhance the ADAS capabilities,ie. to integrate LiDAR, RADAR, and Ultra-sound sensors, each playing a pivotal role in environmental perception.
- Recognize the significance of pre processing for better processing or Sensor Fusion,ie. incorporate specialized modules for both LiDAR and RADAR data.
- Moving toward full autonomy also encompasses a V2X block, emphasizing connectivity for vehicle-to-everything communication.

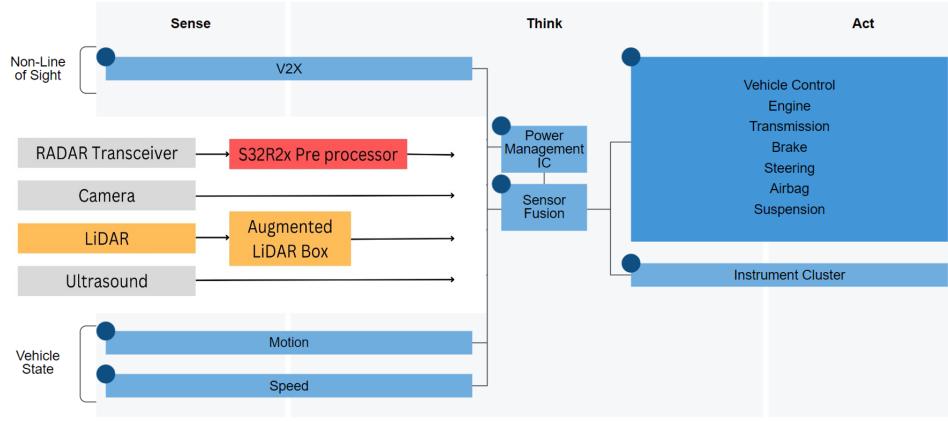


Figure 47: Complete Autonomy block diagram

6.2.1 RADAR Pre-processor

The proposed pre processor is a 32 bit-power architecture®-based microcontrollers, **S32R264** and **S32R274** from NXP [58].

Both S32R264 and S32R274 MCUs address advanced radar signal processing capabilities and merge it with microcontroller capabilities for generic software tasks and car bus interfacing. They meet the high-performance computation demands required by modern beam-forming fast chip modulation radar systems by offering signal processing acceleration together with powerful multi-core architecture.

The S32R264 and S32R274 MCUs offer more than 4x leap in performance per power vs. the previous MPC577X radar microprocessor products, increasing the level of integration available to designers of next-generation automotive radar modules.



Figure 48: S32R2x Block Diagram

6.2.2 LIDAR Pre-processor

Total of two proposals are raised

Augmented LiDAR Box :

Proposal is to go for a software pre-installed in a custom hardware device, replacing the LiDAR's interface box, and enabling a truly plug-and-play experience. The first one in the market is the Augmented LiDAR Box which is a compact, sensor agnostic device that integrates seamlessly into vehicles [59]. Real Time LiDAR pre processing is a very difficult task. This was the motivation behind this proposal.

What this would mean is that all point perception, Detection or classification task like SLAM on chip or Object Tracking would take place within the Augmented LiDAR Box and these results would directly be refined in the Sensor fusion by RADAR and Ultrasound data.

LiDAR on Chip :

This is something that is being researched very intensively. As known, LiDAR (Light Detection And Ranging) is a critical device for self-driving cars. But it's Bulky and clumsy, with 64 lasers containing moving parts. It's also very expensive and sometimes costs more than the car itself. Also it's difficult for commercialization. [54]

LIDAR on-chip is an alternative solution to commercialize the technology. This would offer Compact, integrated on a chip which is solid and durable with no moving parts. It can be produced cheaply on a large-scale and has a low prototype efficiency, 2 meters!

If JLR has plans on investing in Silicon photonics, then they could go for this approach for the future prospect. Design optimization is essential to make an on-chip LIDAR practical for the commercial market. RSoft's tools can be used to optimize design of LIDAR on an

integrated photonic chip [54].

7 Conclusion

The report presents the growing need to dive into the chiplet technology due to the slowing down and stagnation of Moore’s law. We explored domains where chiplet technology can demonstrate its effectiveness, leveraging its scalability, high yield, modularity, and cost-efficient characteristics. We discussed the advantages as well as implementation of chiplets in ADAS and infotainment segments of the automotive industry. Our approach with UCIe 1.1 specification as the interconnect technology offers compelling power-efficient, low latency and cost-effective performance compared to its counterparts. We discussed in detail on the safety and reliability aspects of chiplets from a cyber security standpoint and proposed the concept of ISEA.

We addressed the thermal management challenges associated with chiplets, employing microchannels, floor planning and Silicon Diamond heat sinks as effective solutions. Future endeavors will entail simulating the interconnect network of the ADAS and infotainment chiplet subsystem utilizing gem5.

8 References

1. “Outlook on the automotive software and electronics market through 2023” Ondrej Burkacky et al, www.mckinsey.com/industries/automotive-and-assembly/ourinsights/mapping-the-automotive-software-and-electronics-landscapethrough-2030
2. J. Quinne and B. Loferer, “Quality in 3D assembly—Is, known good die good enough?” in Proc. IEEE Int. 3D Syst. Integr. Conf. (3DIC), 3DIC’2013, pp. 1–5
3. Road accidents information <https://www.livemint.com/news/india/road-accidents-claimed-1.html>
4. Kedar Chitnis (SW Architect, ADAS Systems), Roman Staszewski (SW Architect, ADAS Systems), Gaurav Agarwal (ADAS Marketing Manager) at Texas Instruments. White Paper: “TI Vision SDK, Optimized Vision Libraries for ADAS Systems” https://www.ti.com/lit/wp/spry260/spry260.pdf?ts=1702378206576&ref_url=https%253A%252F%252Fwww.google.com%252F
5. Figures of 1 and 2 <https://uk.farnell.com/wcsstore/ExtendedSitesCatalogAssetStore/cms/asset/images/europe/common/applications/automotive/pdf/ti-adas-solution-guide.pdf>
6. <https://www.anandtech.com/show/13727/arm-announces-cortex65ae-for-automotive-first->
7. Arm Cortex-A78AE TRM IP <https://developer.arm.com/documentation/101799/0003/?lang=en>

8. Arm Cortex-A65E <https://developer.arm.com/Processors/Cortex-A65AE#Technical-Specifications>
9. CoreLink GIC-600AE <https://developer.arm.com/documentation/101206/latest/>
10. CoreLink MMU-600AE <https://developer.arm.com/Processors/CoreLink%20MMU-600AE>
11. CoreLink ELA-600 <https://developer.arm.com/ip-products/system-ip/coresight-debug-and-coresight-components/coresight-ela-600-embedded-logic-analyzer>
12. Safety Island <https://www.farnell.com/datasheets/1506395.pdf>
13. FSD Chip - Tesla [https://en.wikichip.org/wiki/tesla_\(car_company\)/fsd_chip](https://en.wikichip.org/wiki/tesla_(car_company)/fsd_chip)
14. "TMS320C66x multicore DSPs for high-performance computing" <https://www.farnell.com/datasheets/1737036.pdf>
15. "Arm® Cortex®-M4 processor- ST Electronics" <https://www.st.com/content/st-com/en/arm-32-bit-microcontrollers/arm-cortex-m4.html>
16. "Mali-C71AE for Automotive GPU" <https://developer.arm.com/Processors/Mali-C71AE>
17. "DSS7 Display Subsystem IP" https://software-dl.ti.com/jacinto7/esd/processor-sdk-linux/latest/exports/docs/linux/Foundational_Components/Kernel/Kernel_Drivers/Display/DSS7.html
18. MIPI Datasheet <https://www.st.com/resource/en/datasheet/stmipid02.pdf>
19. MIPI PHY <https://mixel.com/ip-cores/mipi-cores/d-phy/>
20. Serdes Automotive <https://www.keysight.com/blogs/inds/2021/01/28/automotive-in-vehicle-communications>
21. Computer Express linkhttps://en.wikipedia.org/wiki/Compute_Express_Link
22. Computer Area Network https://www.ti.com/lit/wp/slyy219/slyy219.pdf?ts=1701835285906&ref_url=https%253A%252F%252Fwww.google.com%252F#:~:text=For%20single%2Dpair%20Ethernet%2C%20the,faster%20than%20a%20CAN%20bus
23. PCIe Specifications <https://pcisig.com/life-fast-lane-pci-express%C2%AE-technology-autobus>
24. MOST Protocol <https://piembystech.com/most-protocol/>
25. MOST BUS https://en.wikipedia.org/wiki/MOST_Bus

26. Flexray <https://en.wikipedia.org/wiki/FlexRay>
27. TI Datasheet https://www.ti.com/lit/ds/symlink/tda4vh-q1.pdf?ts=1702026710442&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTDA4VH-Q1
28. Micromachines <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9609218/#B42-micromachine>
29. Evaluation Board <https://gettobyte.com/nxp-s32k144-evalution-board/>
30. R. P. Patil and P. V. Sangamkar, "A review of system-onchip bus protocols," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 4, no. 1, pp. 271–281, 2015.
31. Hu, Yang, Xinhuan Lin, Huizheng Wang, Zhen He, Xingmao Yu, Jiahao Zhang, Qize Yang et al. "Wafer-scale Computing: Advancements, Challenges, and Future Perspectives." arXiv preprint arXiv:2310.09568 (2023). <https://doi.org/10.48550/arXiv.2310.09568>
32. T. Jose and D. Shankar, "Performance modeling of a heterogeneous computing system based on the UCIE Interconnect Architecture," 2023 IEEE Space Computing Conference (SCC), Pasadena, CA, USA, 2023, pp. 5-10, doi: 10.1109/SCC57168.2023.00009.
33. Universal Chiplet Interconnect Express (UCIE) Announced: Setting Standards For The Chiplet Ecosystem <https://www.anandtech.com/show/17288/universal-chiplet-interconnect-express-announced-setting-standards-for-the-chiplet-ecosystem>
34. UCIE Specifications <https://www.uciexpress.org/specifications>
35. V. Agrawal, F. Piednoel, I. Elkanovich, D. Sil and M. Jahan, "Level 4 Autonomous Driving SoC, leveraging chiplet, advanced package and UCIE," 2023 IEEE Symposium on High-Performance Interconnects (HOTI), CA, USA, 2023, pp. 9-14, doi: 10.1109/HOTI59126.2023.00016.
36. Synopsis IP https://www.synopsys.com/dw/doc.php/ds/c/dwc_ucie_phy.pdf
37. Cadence IP https://www.cadence.com/en_US/home/tools/ip/design-ip/chiplet-and-d2d-conucie-phy-and-controller.html#controller
38. P. Vivet et al., "IntAct: A 96-Core Processor With Six Chiplets 3D-Stacked on an Active Interposer With Distributed Interconnects and Integrated Power Management," in IEEE Journal of Solid-State Circuits, vol. 56, no. 1, pp. 79-97, Jan. 2021, doi: 10.1109/JSSC.2020.3036341.
39. P. Meinerzhagen et al., "An energy-efficient graphics processor featuring fine-grain DVFS with integrated voltage regulators, execution-unit turbo, and retentive sleep in 14nm tri-gate CMOS," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2018, pp. 38–40.

40. I. Gog, S. Kalra, P. Schafhalter, M. A. Wright, J. E. Gonzalez and I. Stoica, "Pylot: A Modular Platform for Exploring Latency-Accuracy Tradeoffs in Autonomous Vehicles," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 8806-8813, doi: 10.1109/ICRA48506.2021.9561747. <https://ieeexplore.ieee.org/abstract/document/9561747>
41. A. Nieuwoudt, M. Mondal and Y. Massoud, "Predicting the Performance and Reliability of Carbon Nanotube Bundles for On-Chip Interconnect," 2007 Asia and South Pacific Design Automation Conference, Yokohama, Japan, 2007, pp. 708-713, doi: 10.1109/ASPDAC.2007.358070. <https://ieeexplore.ieee.org/abstract/document/4196116>
42. K. -H. Koo, H. Cho, P. Kapur and K. C. Saraswat, "Performance Comparisons Between Carbon Nanotubes, Optical, and Cu for Future High-Performance On-Chip Interconnect Applications," in IEEE Transactions on Electron Devices, vol. 54, no. 12, pp. 3206-3215, Dec. 2007, doi: 10.1109/TED.2007.909045. <https://ieeexplore.ieee.org/abstract/document/4383033>
43. F. Sheikh, R. Nagisetty, T. Karnik and D. Kehlet, "2.5D and 3D Heterogeneous Integration: Emerging applications," in IEEE Solid-State Circuits Magazine, vol. 13, no. 4, pp. 77-87, Fall 2021, doi: 10.1109/MSSC.2021.3111386. <https://ieeexplore.ieee.org/abstract/document/9621254>
44. I. Cutress, "Intel's Xe for HPC: Ponte Vecchio with chiplets, EMIB, and Foveros on 7nm, coming 2021." Nov. 17, 2019. [Online]. Available: <https://www.anandtech.com/show/15119/intels-xe-for-hpc-ponte-vecchio-with-chiplets-emib-and-foveros-on-7nm-coming-2021>
45. 2.5D Root of Trust: Secure System-Level Integration of Untrusted Chiplets. Mohammed Nabeel, Mohammed Ashraf, Satwik Patnaik, Graduate Student Member, IEEE, Vassos Soteriou, Senior Member, IEEE, Ozgur Sinanoglu, Senior Member, IEEE, and Johann Knechtel, Member, IEEE https://www.researchgate.net/publication/344056651_25D_Root_of_Trust_Secure_System-Level_Integration_of_Untrusted_Chiplets
46. https://research.chalmers.se/publication/534209/file/534209_Fulltext.pdf
47. A. Basak, S. Bhunia, T. Tkacik, and S. Ray, "Security assurance for system-on-chip designs with untrusted IPs," Trans. Inf. Forens. Sec., vol. 12, no. 7, pp. 1515–1528, 2017. <https://d1.acm.org/doi/10.1145/3439706.3446902>
48. Modular Routing Design for Chiplet-based Systems Jieming Yin* Zhifeng Lin† Muhammad Shoaib Bin Altaf* *Advanced Micro Devices, Inc. Onur Kayiran* Natalie Enright Jerger‡ †University of Southern California Matthew Poremba* Gabriel H. Loh* ‡University of Toronto <https://www.eecg.utoronto.ca/~enright/modular-isca.pdf>

49. Scalable Memory Fabric for Silicon Interposer-Based Multi-Core Systems Itir Akgun, Jia Zhan, Yuangang Wang†, and Yuan Xie Department of Electrical and Computer Engineering University of California, Santa Barbara, California, USA Email: iakgun, jzhan, yuanxie @ ece.ucsb.edu † Huawei, Shenzhen, Guangdong, China <https://web.ece.ucsb.edu/~iakgun/files/ICCD2016.pdf>
50. G. Bognár, G. Takács and P. G. Szabó, "Thermal modelling of embedded microscale channel structures realized in heterogeneous packaging," 2022 28th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC), Dublin, Ireland, 2022, pp. 1-4, doi: 10.1109/THERMINIC57263.2022.9950627.https://www.eet.bme.hu/~poppe/MTMT-DOCs/THERMINIC_2022-microchannel-2022159171.pdf
51. Heterogeneous Integration Roadmap 2021 Edition, Chapter 20., IEEE Electronics Packaging Society, <http://eps.ieee.org/hir> (2021) https://eps.ieee.org/images/files/HIR_2021/ch20_thermal1.pdf
52. Danish Ansari, Ji Hwan Jeong, "A silicon-diamond microchannel heat sink for die-level hotspot thermal management", Applied Thermal Engineering, Volume 194, 2021, 117131, ISSN 1359-4311, <https://doi.org/10.1016/j.applthermaleng.2021.117131> <https://www.sciencedirect.com/science/article/pii/S1359431121005718>
53. J. Vieira da Silva Neto, M. Amorim Fraga, V. Jesus Trava-Airoldi, Development, Properties, and Applications of CVD Diamond-Based Heat Sinks, in: Diam. Sci. Res. High Technol. Work. Title, IntechOpen, 2019. <https://doi.org/10.5772/intechopen.85349>
54. "V2X communications" <https://www.nxp.com/applications/automotive/adas-and-safe-driving/v2x-communications>:V2X-COMMUNICATIONS
55. "HVX architecture-Threading Model" <https://www.xda-developers.com/qualcomm-second-generation-hvx/>
56. "RTX 3060 Laptop" <https://www.notebookcheck.net/NVIDIA-GeForce-RTX-3060-Mobile-GPU-Edition-497453.0.html>
57. "Waymo autonomous car" https://www.linkedin.com/posts/joshuaolaiya_introducing-waymo-utmsource-share&utm_medium=member_android
58. "S32R26 and S32R27 Microcontrollers for High-Performance Radar" <https://www.nxp.com/products/processors-and-microcontrollers/s32-automotive-platform/s32r-radar-processing/s32r26-and-s32r27-microcontrollers-for-high-performance-radar>:S32R2X
59. "Augmented LiDAR Box Datasheet-v2.5" file:///C:/Users/ARJUN/Downloads/Outsight-Augmented.pdf
60. "LiDAR on Chip" https://www.synopsys.com/content/dam/synopsys/photonic-solutions/documents/pdf-demos/rsoft_simulation_methodology_for_lidar_on_chip.pdf

61 <https://www.nxp.com/products/processors-and-microcontrollers/s32-automotive-platform/s32k-auto-general-purpose-mcus/s32k1-microcontrollers-for-automotive-general-purpose:S32K1>

62 <https://gettobyte.com/nxp-s32k144-evalution-board/>