

Level 4 Autonomous Driving SoC, leveraging chiplet, advanced package and UCIe

Vinayak Agrawal
Intel Foundry Services
Intel Corporation
Bangalore India
vinayak.agrawal@intel.com

Francois Piednoel
North America Research
Mercedes Benz
Santa Clara, USA
francois.piednoel@mercedes-benz.com

Igor Elkanovich
CTO
Global Unichip
Jerusalem, Israel
igor@guc-asic.com

Dwaipayan Sil
Advanced Technology Development
Intel Corporation
Phoenix, USA
dwaipayan.sil@intel.com

Mirza Jahan
Intel Foundry Services
Intel Corporation
Phoenix, USA
mirza.jahan@intel.com

Abstract— 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. Smaller silicon dies, or “chiplets,” combined in a single package, with aggregate silicon area much greater than a reticle, are becoming popular and showing great promise to effectively mitigate the yield and size challenges of the traditional approach.

While chiplets solve some problems, they introduce new challenges of interoperability, higher interconnect power and latency, and the availability of a chiplet-to-chiplet IO that can meet bandwidth and reliability requirements.

In this paper, we will demonstrate how heterogeneous chiplets sourced from different vendors, and designed in diverse nodes, can leverage the UCIe interconnect standard, along with advanced packaging, to unleash unprecedented interconnect density, bandwidth, and automotive grade reliability with best-in-class power to pave the path for building a leading Level 4 AD product.

Keywords—ADAS, chiplets, advanced package, die to die, UCIe, self-driving cars, multi-chip SoC

I. INTRODUCTION

Advanced Driving Assistance Systems, or ADAS systems, are becoming increasingly ubiquitous in cars. Many automakers are selling at least some models with Level2 ADAS, and BYD in China and Mercedes in Germany are now selling models with Level3 ADAS. While estimates vary, McKinsey estimates that by 2030, over 50% of the cars will ship with at least ADAS level2, and some 2% (or about 1.6million cars per year) will ship with ADAS level4 [1]

The number and type of sensors in cars has also exploded. Over several dozen camera system in front, rear, and sides with long and short fields of view, multiple Lidar sensors, Radar sensors and ultrasonic sensors combine to provide a situational awareness view around the car in different weather, speed, parking and driving conditions. ADAS Cars now have much more sensing capacity than humans can. Most sensors are not only more capable than human faculties, but there are also strong redundancies built into them. Coupled with high-speed internet links, future car-to-car communications, and technologies like High-Definition Maps, cars are becoming massive sensors-on-wheels (Fig. 1)

As can be expected, having a sensor does not do you much good, unless you can somehow process that data and generate meaningful control signals with it. As data has exploded in service of higher ADAS level requirements, so have the compute requirements for the ADAS system. Data from multiple sensors must be cleaned for transient errors such as mud on camera, or speckle noise, fused between sensors, then segmented in computer vision and other AI cognition engines before it can be used to project what will happen to the car and other vehicles and things on the road in very near future, and then driving policy applied. All of this must be done with very little latency if the car has to travel at reasonable speeds. This has caused an explosion in amount of computation required from Level2 systems to Level4 systems (Fig.2) [7]

This explosion of processing requirements has led to its own problems. Since Moore’s law has also slowed down considerably in past few years, the exponential increase in ADAS requirements is directly now leading to increase in computer chip sizes. This has very detrimental effects on die yields and hence costs of Level3 and higher systems. Another impact is power consumption.

ADAS system power is increasing, while at the same time Auto industry is moving towards battery-based propulsion in electric vehicles. While this is a problem of cooling and costs like in datacenters and other high performance computing use-cases and additional problem is battery. Unlike in traditional internal combustion engine cars, increased ADAS energy requirements are directly at odds



Fig. 1

A Mercedes Level 4 Car with sensors

with range specifications of the vehicle.

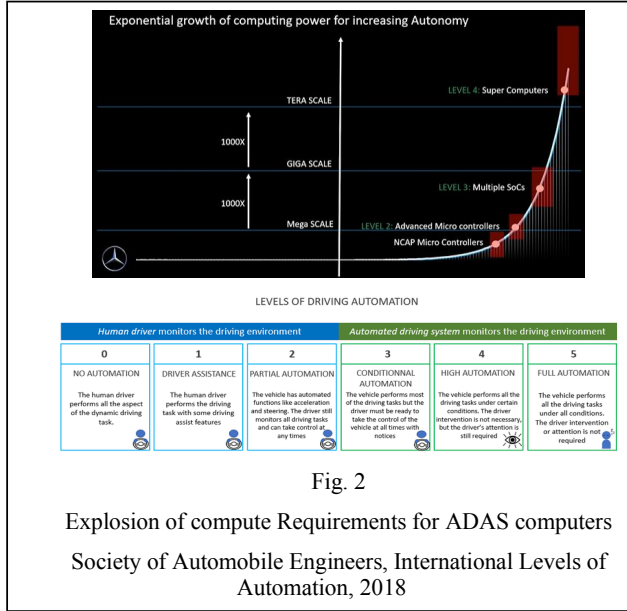


Fig. 2

Explosion of compute Requirements for ADAS computers
Society of Automobile Engineers, International Levels of Automation, 2018

In short Level3 and beyond ADAS systems start running into some fundamental constraints that make realizing such system hard and less optimal. For Level4 and above systems it may become prohibitive. Let us look at these constraints of SoC design a little more closely next before we go into how we are proposing to solve these challenges.

II. THE PROBLEM OF TODAY'S SOCs

A. Large die area required

Dennard scaling ran out of steam about a decade ago somewhere around 40nm to 28nm nodes. Prior to 40nm, logic voltages scaled in every node. In 40nm the scaling was less aggressive, and since 28nm logic gate voltages have remained in 0.75V-0.85V range, depending on the application, for nearly ten years.

Thanks to Moore's law [12] designers could expect gate density to more than double in each die area every two years, while at the same time seeing those gates become about 1.5x faster and consume less power and yield better. Today foundries give a tradeoff – you can usually improve two of the three (cost, power, and performance), but not all three, and even then, the gains are modest.

Many of today's Level3 systems, which are not yet even on the market, have over 200sq.mm. of computational silicon. Going to 1000sq.mm. and above will be required for Level4 systems and that in more advanced nodes. While cost should increase for more area, it grows faster because this necessitates use of large dies that have far poorer yields.[13]. The other option is to use several smaller chips and connecting them together for example on a PCB (Fig. 3). However, this presents its own challenges.

B. Power consumption

Obviously more data and more computation entail more power. A large fraction of power is consumed in just moving data around.

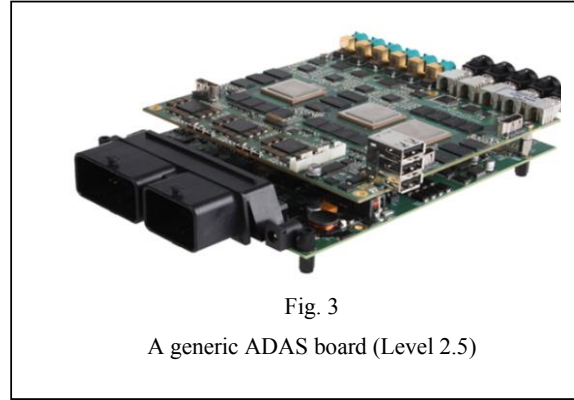


Fig. 3

A generic ADAS board (Level 2.5)

When data is transferred on a die, it usually costs about 0.1pJ/bit to 0.2pJ/bit depending on lengths of the connection. However, a typical PCIe Gen4 link consumes over 5pJ/bit transferred. What is worse – this energy consumption is when PCIe link is working at near 100% utilization. For frequent but intermittent data that does not allow the link to be put in a lower power state, the power of the link remains constant while average data rates plummet. For example, a typical PCIe Gen5 link capable of providing 500GBps of bandwidth costs about 20W, even if average usage is only a tenth of that. Other interconnects can be used, but unfortunately most are either worse than PCIe (112Gbps SerDes solutions) and often have more complicated PCB and software requirements.

C. Latency and Software Model

When the silicon is split into multiple packaged chips, typically the buses used have high latencies. Each chip also has its own Network-on-chip independent of the other. As a result, the software also must follow a divide and conquer approach. This is often a bit more challenging for developing software, especially if the hardware needs to be configurable

D. Challenges of technology and business uncertainties

Designing a large SoC is hard and time consuming. For automotive it is harder still. Automotive parts need higher reliability under harsher conditions and meet reliability standards such as AEC-Q100 [2]. Thermal ranges are higher, DPPM (defective parts per million; these are parts with defects that either escape testing or show up in chips in field due to

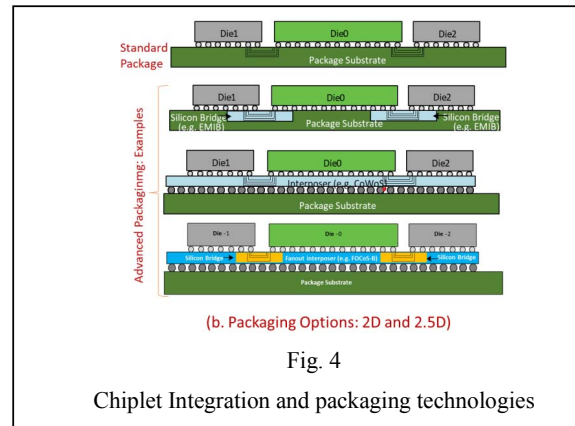


Fig. 4

Chiplet Integration and packaging technologies

aging etc.) ranges are usually hundreds of times tighter than

for datacenters for example, and there are additional requirements for shock and repeated stresses.

The SoC design, package assemblies etc. must undergo thorough validation that takes years. Then the system must be validated in actual cars for 2 years or more before it can be offered to customers. A new SoC design project for automotive chips may take as many as 5 years from design start to production.

This is not only expensive, but it also creates uncertainty. Regulatory requirements may change necessitating small changes to the hardware that may nevertheless require long re-work cycles. AI algorithms may improve such that hardware may need updates, either to add or improve features, or to replace with more efficient cheaper hardware to exploit the progress of software.

For a system with large dies, not only is the initial cost high, but every change will entail a costly re-spin.

III. CHIPLETS

The problem of large dies is yield and lack of long-term flexibility. The problem of separate chips is complex software, and very high interconnect power consumption, bandwidth constraints between different chips that occur due to power capable of 500GBps Bandwidth costs about 20W, even if the average usage of the link is only a tenth of the full bandwidth. but also because number of pins or balls on Ball Grid Array packages typically used is usually limited and more bandwidth requires using more balls.

For some years now High-Performance Computing industry has solved this problem by combining multiple chiplets in a single package (Fig. 4) This reduces size of each individual chip, improving overall yields

Chiplets can be combined with each other by using advanced packaging technologies such as EMIB technology or interposer technology [3][4]. While this approach solves large die problem, it still leaves the problems of software complexity, and die interoperability. PCIe has been in use for decades, in-package die to die is new. In addition, more

complex package technologies create new reliability problems..

To solve this problem, many interconnect technologies such as openHBI, Bunch-of-Wires (BOW) in addition to proprietary technologies have been suggested. We are choosing to use Universal Chiplet Interconnect Express or UCIe [5] in our design. UCIe defines electrical parameters to firmware and helps mitigate many of the new problems in this area.

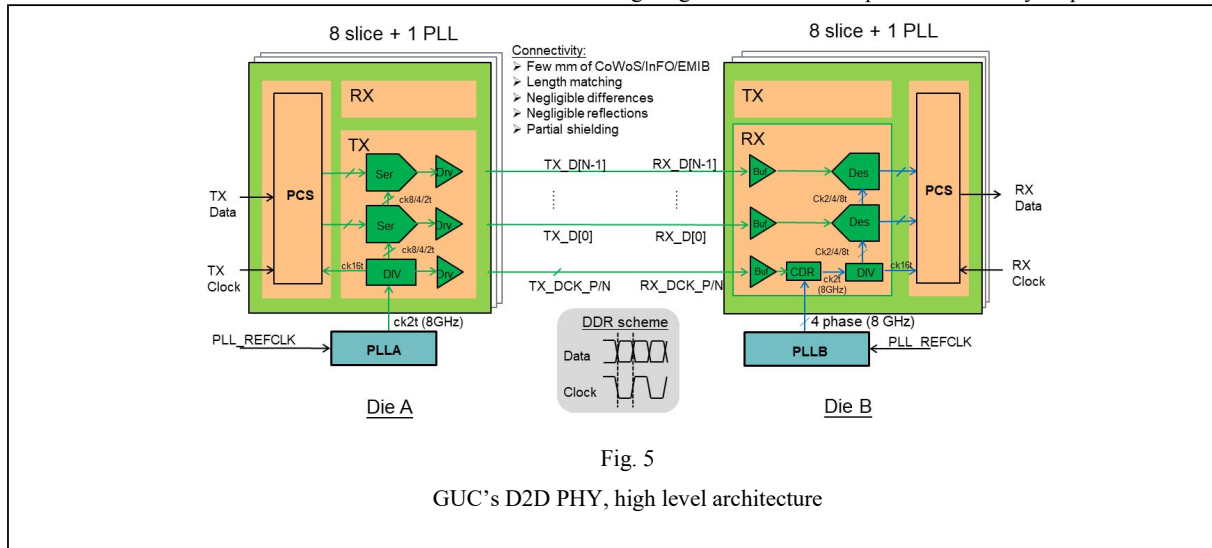
IV. UCIe AND CHOICE OF INTERCONNECT

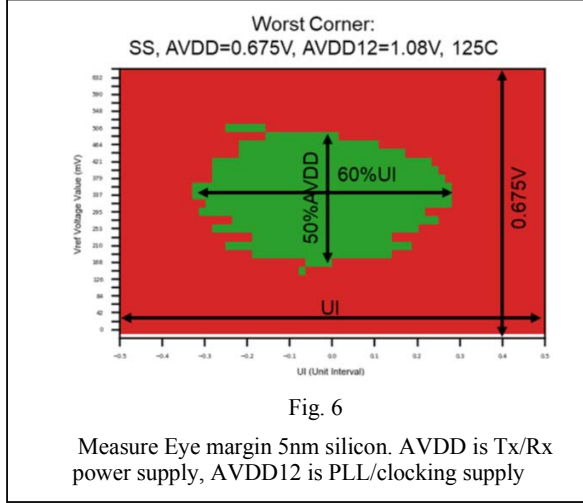
Choice of die to die interconnect is crucial for making multi die platforms. Such an interconnect should have well defined electrical, physical and logical attributes such that chips coming from different vendors can be interoperable. It should also address new problems thrown by advanced packaging. And it should have sufficient speed. We chose to use UCIe in this SoC platform and in our future platforms. [11]. In addition, we are using UCIe PHY based on Global Unichips (GUC) technology with several additional features to UCIe for the purposes of this and future projects.

The PHY is based on a silicon proven die to die PHY design (Fig. 5). A single PHY module consists of 32 data lanes sharing one clock lane. 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. ESD requirements for this design are notably relaxed – unlike for chip to chip links, for die to die inside package, after packaging ESD event on the lanes are not possible. Hence the link needs meet only 30V 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 (Fig. 6). Even though each lane transfer 16Gbps individually, there is practically no signal integrity issue despite the use of high resistive interposers.

A. Interoperability

We want to be able to re-use several chiplets in other SoC programs, much like a PCIe based chip can be used in multiple PCB based system designs without the need for another tapeout. UCIe standard released in 2022 addressed this by giving a common set of specifications. Any chiplets that meet





the specification will be interoperable with each other. More than a hundred companies have signed up as adopters of the standard, all major silicon IP vendors are designing IPs in middle 2023 in advanced nodes, with many in older nodes too.

Other standards such as BOW [9] have also come out with PHY specification similar to UCle. However the UCle standard has defined the entire stack of layers to the software level compatibility if both chiplets used CXL or PCIe at protocol level. By using a newer version with industry standard Network on Chip IPs and bus architectures such as AXI in conjunction with UCle PHY we are ensuring that our chiplets will remain compatible with future SoC requirements and later designed chiplets.

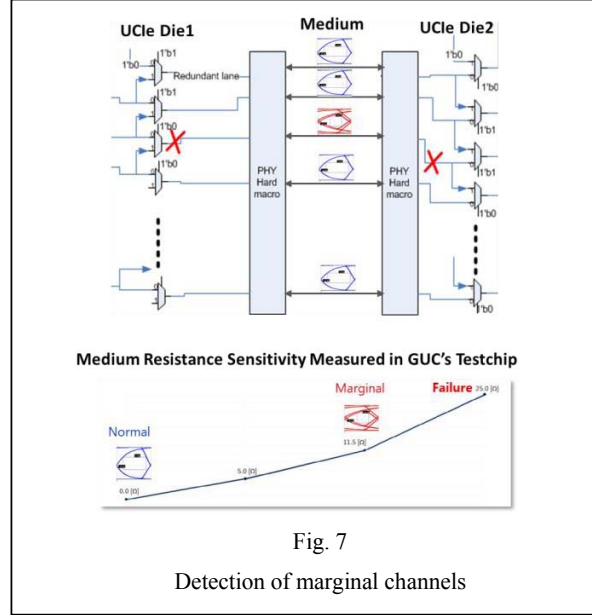
B. Power

By using a simple architecture, GUC's latest generation PHYs are now achieving better than 0.3pJ/bit energy efficiency [8]. Unlike chip to chip standards like PCIe, UCle minimizes data lane toggling during quiet data periods on the bus, saving more power. A typical 512GBps peak bandwidth capable bidirectional link will consume less than 0.7W, while a PCIe link with similar bandwidth would consume about 20W of power.

C. Link Reliability

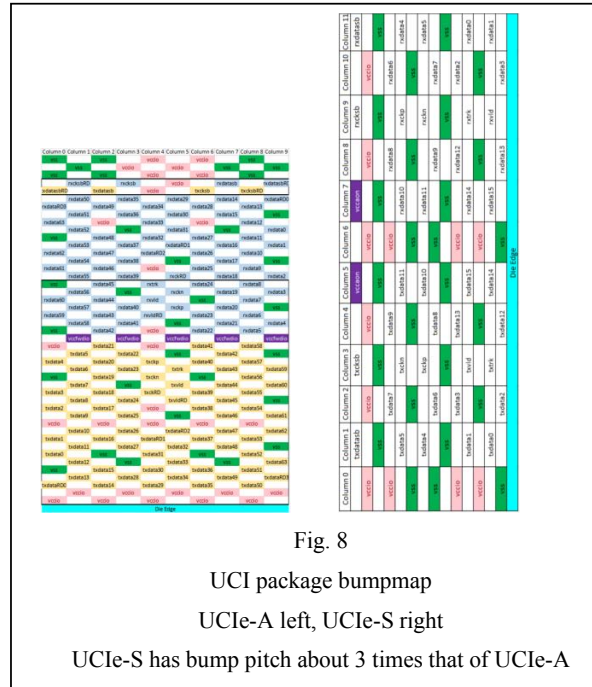
Automotive systems in particular need to have higher reliability than electronics in many applications such as Personal computers and mobile phones. ADAS systems are ASIL-D systems as per ISO26262 requirements, failures can result in injury or worse.[14]

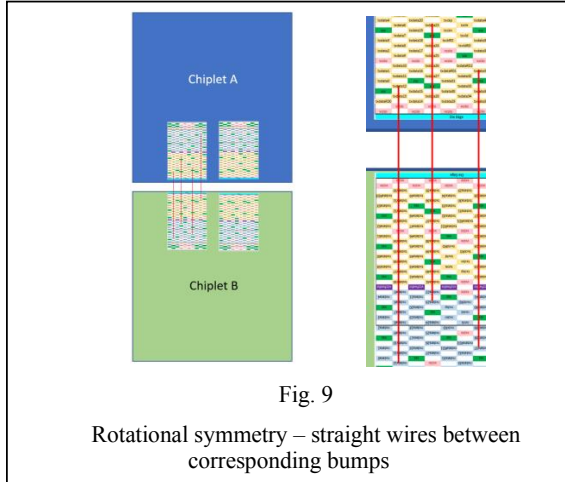
GUC's PHYs can periodically measure eye margins, and other lane parameters to detect gradual degradation of the link with time. For example one proxy for failure of microbump soldering is resistance of microbumps – by keeping track of rise times of signals such resistance, and even more directly signal eye margins can be tracked multiple times each day (Fig. 7). This data is also uploaded to cloud. If a lane starts degrading, analysis of it's own margins along with data from other lanes in the same chip as well as data from other chips in that lot can be used to predict eventual failure. UCle standard provides redundant lanes [10], In our system, in the event a potential failure is detected, such a future failure will be avoided by replacing the marginal lane with a redundant



lane. We will achieve better than 0.1FIT failure rate with this and similar measures.

Another feature implemented in our SoC design is a brute force measurement circuit that will periodically run diagnostics for several minutes each time the ADAS system is not in use (for example in a parked car). The system will keep track of failures per lane and thus bit-error-rate (BER) for not just the entire link, but also isolate a weak lane if the BER is dominated by just one or two lanes, as in often the case. Such a lane can then be kept track of and repaired if BER falls below its target rate to ensure Failure-in-time (FIT) specs are kept over the lifetime of the SoC.

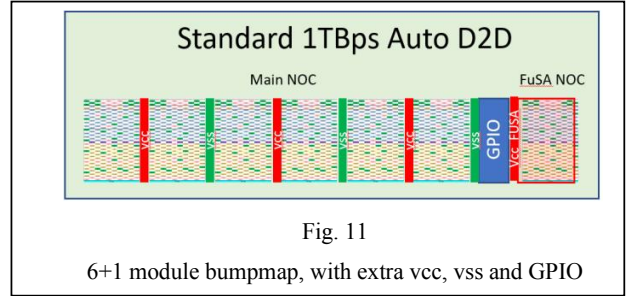
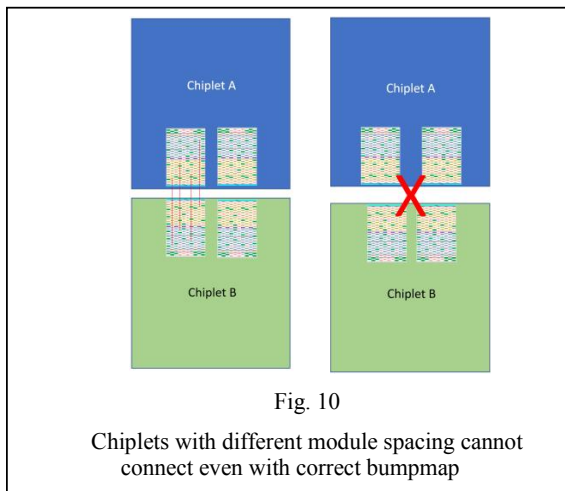




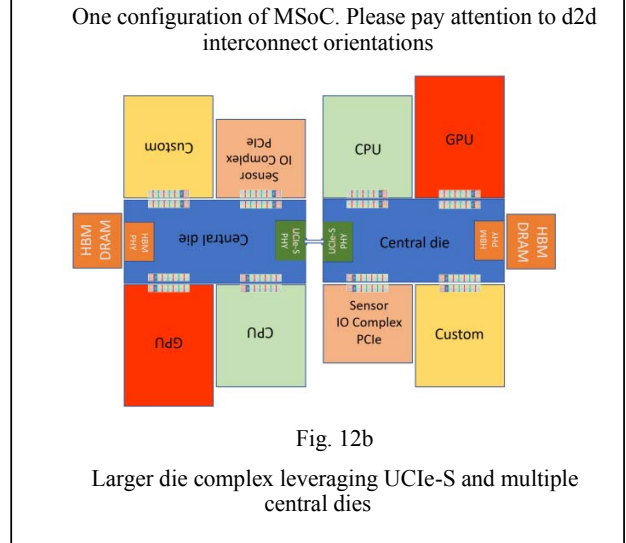
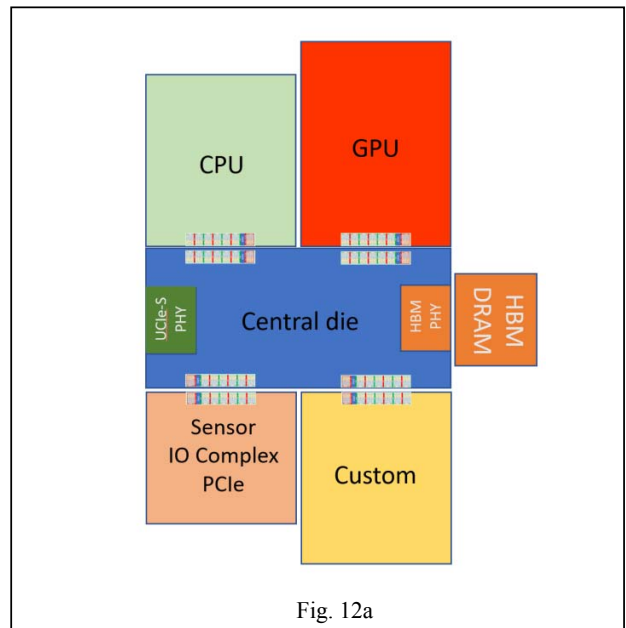
The monitored data of per-lane parameters such as eye-margins and long-term parameters such as individual BERs will be uploaded to a cloud database where it will be analyzed further at a fleet level to keep track of any systemic issues across batches of ADAS hardware, and any potential issues due to environmental factors.

D. Form factor requirements

UCIe standard defines two different PHY specifications. UCIe Advanced or UCIe-A is the PHY specification for advanced packages that use interposer (CoWoS) or EMIB technology and can have relatively smaller (~45um microbumps). UCIe Standard package PHY or UCIe-S specifies PHY which is meant to be used with cheaper packages that use standard flip-chip packages with 110um-130um bump pitch. For both of these, UCIe standard has defined bump patterns to be used in silicon die (Fig. 8)[6]. Each UCIe-A PHY or “module” consists of 64 lanes of data in each direction along with additional lanes for clocks, sidebands and redundancy. UCIe-S module on the other hand has only 16lanes. The UCIe bumpmap has rotational symmetry. Two chips (or the same chip if it is rotated version of itself) always connects corresponding bumps in straight lines. (Fig. 9). This greatly simplifies multi-die design and signal integrity design for packages.



However UCIe bumpmaps are not defined for multi-module (i.e. multi-PHY) cases. Many a chiplets needs higher bandwidth than what a single UCIe PHY module can provide, and thus uses more than one 64-lane module. However one chiplet may space the bumpmaps of the two modules more than the other chiplet. In such a case, UCIe guidelines on their



own are insufficient to guarantee routability of connections. (Fig. 10)

In addition, in automotive electronics, a separate dedicated Network-on-chip (NOC) is often required in each chiplet to connect the Functional Safety (FuSA) reliability-CPU cores and block components. In a multi-chiplet SoC all such FuSA-NOCs must be connected together across all chiplets. Any FuSA subsystems have higher reliability requirements than main systems and must have independent power supplies.

To address all of the above requirements, we have defined a 6 PHY module + 1 PHY FuSA-module configuration bumpmap for our project. At 16Gbps per lane operation, after accounting for all the overheads for AXI protocol used to connect NOCs, each single die-to-die interconnect will give over 1 Tera-Bytes-per-second of bandwidth, in addition to sufficient bandwidth for FuSA NOC subsystem. (Fig. 11). The configuration also has space for several GPIO bumps that will be used to connect debug infrastructure and configuration buses such as APB.

Finally, the dimension of the edge of each die that will have UCle interface has been specified to be 12mm. Location of UCle in a “central connect die” which contains a central NOC, several SRAM caches, CPUs, and several other PHYs and is capable of acting as a standalone Level2 ADAS processing chiplet has been specified. The central die also has HBM3 interface on one side and UCle-S interface. (Fig. 12a).

The 6+1 configuration doesn’t have rotational symmetry any more. In Fig. 12, the PHY IP is mirrored, rather than rotated (as was the case in Fig. 9).

An example configuration on Mobility-SoC (MSoC) is shown in Fig 12a. The Central die connects various different types of chiplets to one another. As long as form-factor rules are observed, the chiplets can be swapped with different chiplets and potentially newer chiplets in future. So for example, in while some systems will have 1 CPU and 1 GPU die as shown, other systems can have 2 CPU dice if needed. It will also be possible to connect two “central connect dies” together to build much larger systems for higher compute requirements. (Fig. 12b)

V. CONCLUSION

We have described several challenges in building an ADAS compute SoC for level4 systems with the goals of ease of design, re-usability, reliability, and futureproofing.

The designs will heavily leverage existing IPs and UCle’s emerging ecosystem. The approach is also already informing many industry partners developing chiplets for automotive applications and High-Performance Computing application.

While our focus was on interconnect and physical aspects of the IO, work is also ongoing in protocol compatibility, software compatibility and other issues. That will be the subject matter of a future communication.

ACKNOWLEDGMENT

We thank our multiple colleagues within Mercedes Benz, Global Unichip and Intel. We also thank colleagues from other companies that are participating in ADAS Mobility-SoC program.

REFERENCES

- [1] “Outlook on the automotive software and electronics market through 2023” Ondrej Burkacky et al, www.mckinsey.com/industries/automotive-and-assembly/our-insights/mapping-the-automotive-software-and-electronics-landscape-through-2030
- [2] AEC Q100 requirements, www.aecouncil.com/Documents/AEC_Q100_Rev_G_Base_Document.pdf.
- [3] R. Mahajan *et al.*, “Embedded Multi-Die Interconnect Bridge (EMIB) – A Localized, High Density Multi-Chip Packaging (MCP) Interconnect,” *IEEE Trans. Components Packaging and Manufacturing Technology*, vol. 9, no. 10, pp. 1952-1962, Oct. 2019.
- [4] D. Das Sharma, “Universal Chiplet Interconnect express (UCle)*: Building an open chiplet ecosystem”, [White paper](#) published by UCle Consortium, Mar 2, 2022
- [5] D. Das Sharma, “Universal Chiplet Interconnect Express (UCle)*: An Open Industry Standard for Innovations with Chiplets at Package Level”, *IEEE Micro Special Issue*, Mar-Apr 2023
- [6] D. Das Sharma et. al., “[Universal Chiplet Interconnect Express \(UCle\)*: An Open Industry Standard for Innovations with Chiplets at Package Level](#)”, invited paper, *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, Oct 2022
- [7] Francois Piedrol, “The standardization imperative for chiplets”, SNUG Silicon Valley 2023, keynote www.synopsys.com/community/snug/snug-silicon-valley/location-proceedings-2023.html
- [8] GUC die to die IP www.guc-asic.com/en/solution-ip-d2d.php
- [9] Bunch Of wires PHY Specification, opencomputeproject.github.io/ODSA-BoW/bow_specification.html
- [10] D. Das Sharma, “High-Bandwidth and Low-Latency Standardized Interconnect for an Open Chiplet Ecosystem”, Plenary talk at IEEE Heterogeneous Integration Roadmap (HIR) Workshop, Milpitas, Feb 2023. <https://r6.ieee.org/scv-eps/?p=3049>
- [11] UCle Express www.uciexpress.org
- [12] Moore, Gordon, “Cramming more components onto integrated circuits”, *Electronics*, Volume 38, Number 8, April 19, 1965, ieeexplore.ieee.org/document/4785860
- [13] J. A. Cunningham, “The use and evaluation of yield models in integrated circuit manufacturing” *IEEE Transactions on Semiconductor Manufacturing* (Volume 3, Issue 2, May 1990), ieeexplore.ieee.org/document/53188
- [14] ISO 26262, www.iso.org/standard/68383.html