Optimizing Chiplet Disaggregation with Package Technology

Trent Uehling, Julian Pontes, Carl Culshaw, Karl Leiss
NXP Semiconductors
6501 W William Cannon Dr
Austin, Texas 78735 USA
Ph: 210-365-8058
Email: trent.uehling@nxp.com

Abstract

As the chiplet concept is making its way into products, some are declaring a "chiplet revolution" in the semiconductor industry. The term revolution is used because of the paradigm shift from large, monolithic "System on Chips (SoC's) to thoughtful disaggregation of chips into functional parts, or "chiplets," requiring packaging of these "chiplets" in a cost-effective packaging technology. Multi-chip packaging has been in use for decades for various forms of heterogeneous integration. Now, the movement is spurred largely by the slowing of Moore's law and the need to accelerate performance by methods other than semiconductor node progression. High-performance requirements are pushing the core processor to more advanced technology nodes, while other functions such as analog interfaces and memory are better served by different technologies, resulting in a need for heterogeneous integration of various "chiplets" in a single package. For example, customers may still require 5-volt interfaces and Analog-to-Digital Converters (ADCs) which are not possible in smaller geometries. On the other side, contemporary trends on communication like 10BaseT1S require upgrades of existing Microcontroller Units (MCUs). In general, this breaks down into three distinct types of chiplets: computing, interfaces (including analog and networking) and memory. Each of these have different interface bandwidth requirements and ideal technologies. Efforts are underway in the industry to develop standards that are optimized for bandwidth, latency, and power. However, the standardization and interoperability requirements will likely add some overhead in these areas. It is speculated by some that this concept may lead to some semiconductor Intellectual Property (IP) being "commodified" similar to what happened to some memory technologies in the past. Industry-wide standards can also create synergy for evolving and improving of the interfaces, contributing to the overall value of advanced packaging.

Historically considered a "necessity" to connect chip power and signals, packaging now brings more value to the system solution. As Input/Output (I/O) density and bandwidth increase, so does package cost. The key is to "right-size" technology to meet bandwidth and cost targets. Disaggregation of large multi-core devices likely require thousands of connections and no latency impact, while implementation of an I/O hub or analog interface may require tens or hundreds of connections and can possibly afford some increased latency. Clock synchronization and error correction are other considerations that contribute to area and latency.

As a SoC is disaggregated into chiplets, the SOC signals and connections to the on-chip communication bus are analyzed for mapping to a chiplet interface. In some cases, a standard interface may be the best solution. In other cases, a custom interface may be most suitable for power, area, and latency requirements. The bandwidth, latency, power, and area requirements are assessed to determine the interface requirements and packaging technology required to meet specifications.

Key words

Architecture, Chiplet, Disaggregation, Microcontroller, Microprocessor, Packaging

I. Introduction

Chiplets are a reality today as evidenced by a multitude of products built on a chiplet platform [1], [2]. The business case for the first commercial scale use of chiplets has been clear as it is predicated on large die, advanced technology defect density and reticle size limitations. That is, it became quite evident that there was not a viable alternative to separating the large system-on-chip into smaller pieces. The increased chip yield and die per wafer with the smaller chiplets easily compensate for the increased integration and package cost [1]. Once established, these platforms can be expanded to include chiplets optimized by technology for performance and cost. An example of typical partitioning blocks one might find in a chiplet system is shown in figure 1. At the heart are the Central Processing Unit (CPU) and Graphics Processing Unit (GPU) blocks which typically are designed in an advanced node for the best digital compute performance. Other functions can likely be realized in less expensive nodes. An additional application processor or AI accelerator may be added for specialized computations. While registers and processor cache reside on the processor chiplets to meet latency requirements, other forms of memory such as RAM or flash may or may not be included on the same chip. Finally, digital I/O as well as analog blocks and I/O are other extensions that can be node optimized. Once a chiplet platform is established, in theory one could create a family of products using permutations of a core set of chiplets. Next-generation products could be realized by upgrading only key chiplets while reusing those that are not necessary to revise for improved performance (I/O, for example).

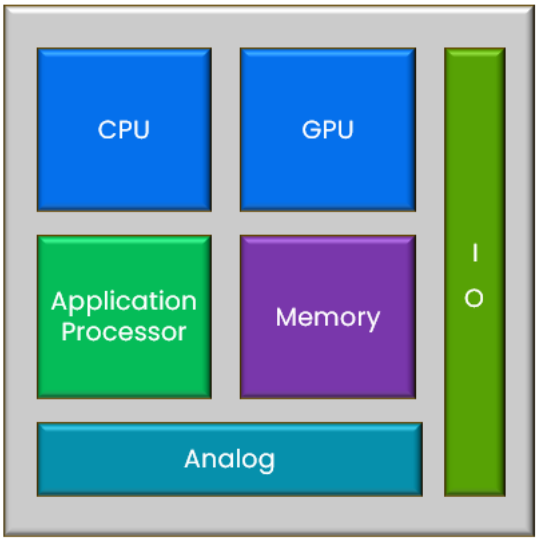

Figure 1. Hypothetical blocks found in a chiplet system

Many studies of SoC "disaggregation" have been performed [3], [4], [5]. In general, these studies analyze a high-performance compute system where significant added package cost is justified by increased silicon yield, or in some cases because no other solution exists. But what about the situations where the chiplet business case is not dominated by large die, advanced technology defect density and reticle size limitations? The defectivity and die size argument is no longer a nice benefit on which to anchor the business case, which now relies heavily on die and package overhead, design cost, reuse potential and less tangible 'time to market' benefits. For these platforms, a lightweight approach based on minimal chiplet blocks, low-overhead interfaces and economical packaging is key to success. In this study, various partitioning schemes are considered along with interface requirements and package technology needed to realize integration of the chiplets. These configurations assessed for performance, relative cost and business potential compared to monolithic solutions. Sizes or line spacing to squeeze more text into a limited number of pages.

II. Interface Requirements

First, one must understand the data movement requirements for a functional SoC and the appropriate chiplet interface to service this traffic. This type of analysis requires sophisticated models based on an established design [6], [7], [8]. At a high level, the Network on Chip (NoC) interface provides connectivity to the various blocks via the on-chip bus, e.g., Advanced eXtensible Interface (AXI), Synchronous Serial Interface (SSI), etc. By considering the compute frequency of digital processing units, as well as the maximum bandwidth of memory and analog interfaces, a "minimum" performance requirement for chiplet interface can be derived. Figure 2 shows representative frequencies for some common processor blocks in 16nm technology based on the authors' analysis of an actual product.

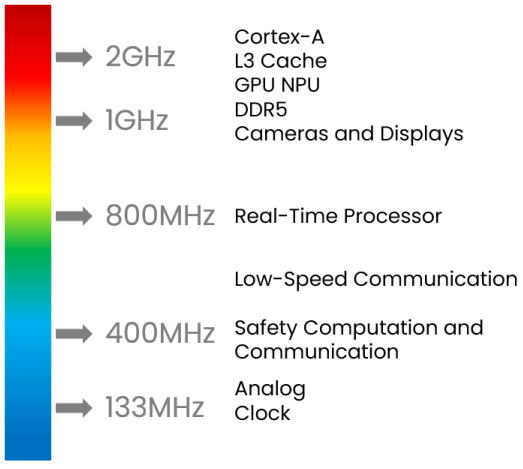

Figure 2. Representative frequencies for 16nm edge processor

While some bandwidth requirements drive an obvious need for a high-performance interface, some blocks can be connected using relatively simple, common ports. Early chiplet solutions were built upon proprietary interfaces [1], [9]. As the proposition of an "open chiplet economy" is put forth, efforts are underway to develop an interface specification to enable interoperability between chiplets from different suppliers. Notably, the Universal Chiplet Interface express (UCIe) interface specification [10] has emerged as the front-runner with the backing of many industry heavyweights. While this interface covers requirements from a broad swath of users, the efforts to make it "universal" can result in a fair amount of overhead that is not required for many use cases. UCIe is aligned to advanced nodes with high compute and data access performance requirements. Implementation in "legacy" nodes with lesser performance requirements may result in under-utilized chip area. Figure 3 summarizes bandwidth specifications for UCIe standard and advanced package interfaces. Note that the standard package implementation has sixteen lanes with 64 to 512 Gb/s capability, while the advanced package is capable of up to 2Tb/s with 64 lanes.

Characteristic	Standard Package	Advanced Package
Data Rate (Gb/s) min / max	4 / 32	4 / 32
Lanes	16	64
Bandwidth (Gb/s) min / max	64 / 512	256 / 2048
Bump Pitch (µm)	100 – 130	25 - 55

Figure 3. UCIe bandwidth for standard and advanced package [10]

Referring back to figure 2, the bus width between blocks, along with the frequency will determine the data-rate capability. Some interfaces will be capable of extremely high data-rates which would be well suited to a high-bandwidth chiplet interface, while others only require minimal bandwidth. Other requirements influence interface specification such as package technology, test, Built-in-Self-Test (BiST), timing closure, cache coherency, interface monitoring, integration and validation, and impact of data transfer latency on software. stacks. From a pure bandwidth perspective, figure 4 illustrates calculated SoC block-to-block bandwidth as a function of operating frequency and bus width. In some cases (lower left, unshaded cells) the bandwidth requirements are below the minimum performance of UCIe standard package or advanced package interfaces.

Bus Bandwidth (Gb/s)*						
	Bus Width (bits)					
Frequency (MHz)	32	64	128	256	512	1024
2000	64	128	256	512	1024	2048
1000	32	64	128	256	512	1024
800	26	51	102	205	410	819
400	13	26	51	102	205	410
133	4	9	17	34	68	136

* Bandwidth calculated by (Bus Width) x (Frequency). Overhead and Muxing not considered.

UCIe Standard Package | UCIe Advanced Package

Figure 4. UCIe standard and advanced package minimum bandwidth [10] mapped to calculated block-to-block bandwidth as a function of frequency and bus width.

As compute requirements drive processor cores to advanced nodes and high bandwidth chiplet interfaces, other system functional blocks in mature nodes may be best connected using established common interfaces that can serve both chiplet and external connection.

III. Low-End Example

Whilst there is a significant amount of activity in the higher end products for a vast amount of data movement, there are a multiple of microcontroller and edge microprocessor applications where the demands for data flow are significantly less. In these lower performing applications, the data control loop is very close to the compute engine, with the compute engine residing on the same physical silicon alongside the control / actuation. In these cases, a relatively simple source-synchronous interface is sufficient for data transport, such as the traditional Serial Peripheral Interface (SPI) or Low Power Serial Peripheral Interface (LPSPI).

For example (figure 5), two LPSPI can be configured for device-to-device communication across two devices. Whilst in this example they are being used for chip-to-chip across a printed circuit board, they could equally be used as a chiplet die-to-die connection. This has the advantage of being a simple and easy to use, well proven interface. Further, when placed inside a package, the inline capacitance is dramatically reduced, allowing the data rate to be significantly increased. For example, a quad-SPI interface (4 data lanes) running at 100MHz in Single Data Rate (SDR) mode is capable of 400Mb/s. If the same frequency is successfully used for Double Data Rate (DDR) mode, the data transfer rate could be as high as 800Mb/s

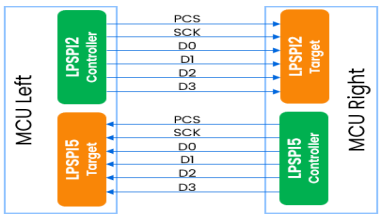

Figure 5. Example of LPSPI chip-to-chip connection [11]

As traffic requirements increase beyond what the typical SPI is capable of, something such low voltage differential signaling (LVDS) [12] or SerDes can be used which gives an MCU a fast, low-pin-count serial-communication link. One drawback can be power-hungry clock recovery blocks [13]. Work has been done to develop a configurable asynchronous SerDes architecture, targeting medium data rates, which is claimed to achieve high bandwidth low power [14]. can be used.

Figure 6 shows an example of a hypothetical scalable chiplet-based MCU platform created with xSPI and LVDS interfaces implementable in low-cost wirebond package technology. To maintain a low-cost package platform, the compute clusters would be monolithic. That is, unique compute chiplets would be required for various core counts and configurations. Other functional blocks can be scaled with chiplets based on the application requirements.

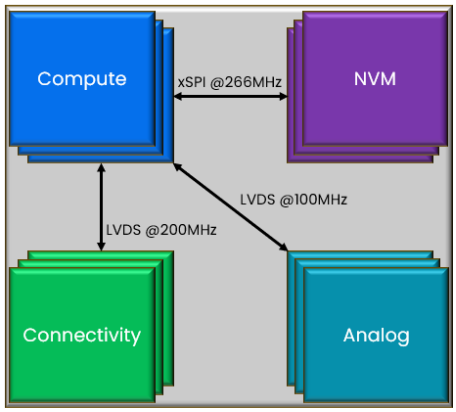

Figure 6. Hypothetical Chiplet-based MCU platform

IV. Mid-Range Example

CPUs are present in different design partitions. For example, the Arm Cortex-A cluster can be composed of different cores. More advanced products can have a cluster composed of Out-of-Order Big Cores and also Little In-Order Cores for power efficient computation. Some other variations are composed exclusively of Little Cores. Cortex-A cores have dedicated L1-L2 Cache, and the cores share a L3 Cache responsible to deal with coherency among the Cortex-A cores inside the Partition (no coherency with other elements of the device). Under this assumption, the implementation of Cortex-A partitions as chiplets would not require coherency features. From the frequency and performance point of view, the Cortex-A performance requires high operation frequencies and high-throughput. Processors are strongly impacted by latency to Double Data Rate (DDR) memory (in special In-Order CPUs – which are going to stall in case of cache misses and wait for the data from DDR). Partitioning Cortex-A as chiplet would require high performance, high-bandwidth, low-latency Chip-to-Chip interface to DDR and other cores. With this interface driving package complexity and cost, it may be beneficial in some cases to combine high performance cores and DDR interface on a single chiplet (figure 7). To benefit from high frequency, this partition would be better implemented in advanced nodes – for the sake of Performance and Power Consumption.

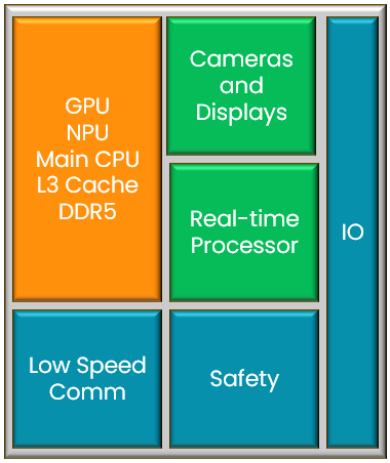

Figure 7. Example of Processor disaggregation with high-performance cores, L3 cache and DDR5 interface contained in one chiplet.

For Real Time Computation, Edge Compute processors can have one or more Arm Cortex-M cores. These cores have L1 cache and tightly coupled memories for instruction and data to ensure that critical code and data are kept close to the processor, requiring an isolated partition. As Edge Processors typically utilize real time computation, this can be a suitable candidate to be developed as a chiplet – improving reuse among products, as well as reducing time to development and integration. Since there is no coherency associated to these Cores, the interface frequency and bandwidth requirements are achievable with lower complexity package solution.

Accelerator cores such as GPU, Neural Processing Unit (NPU) and Video Processing Unit (VPU) require high datapath bandwidth along with low latency. In addition, some level of cache coherency is likely required between the accelerators and CPU, also between the accelerators themselves. Depending on product family architecture, it may be sensible to combine a minimum set of CPU plus accelerators on a single chiplet in an advanced technology node, which can be packaged with some of the lower-performance chiplets using standard package technology for an entry-to-mid-tier offering. Additional accelerators or even

CPU cores could be added using advance package technology for mid-to-high-tier products.

Camera and Display are two components that are especially important for the automotive industry. The number and the capability of displays in infotainment products is increasing constantly – even for low end cars – and this growth is expected to continue. Camera and display partitions have significative throughput requirements. Similar to accelerators, it makes sense to include a minimum set of these partitions with compute on an advanced node, while enabling expansion through advanced packaging.

There is a multitude of IPs that are responsible for off-chip communication. These IPs vary in terms of I/O operating voltages, frequency, single/double data rate, serial/parallel wiring, and synchronous/asynchronous to a clock source. The main interest in keeping all these components in the same partition is the possibility of reuse in multiple products – as most products require the off-chip communication IPs. The technology node to implement these IPs can be less advanced than that used in computing partitions and can also include safety computation and communication functions are not highly compute-intensive and also do not require high memory bandwidth . Examples of off-chip communication IPs are: PCIe, Ethernet, MIPI, USB, I2C, LPSPI, CAN, Audio, SDIO, and many more.

Committing chip area to a chiplet interface is a decision not taken likely, as this interface can only be used for chiplet applications and is “dead” space when not used in a chiplet system. The case for chiplets for some applications may only be made with the option of also using the chiplet as a monolithic packaged product. The cost is further aggravated by the high bandwidth capabilities of emerging standard chiplet interfaces which far exceed the requirements for some chiplets. A possible solution is to create a configurable chiplet interface that can be enabled with full functionality and cache coherency for high-performance applications, or can be configured as multiple, smaller interfaces for low-bandwidth interface requirements (figure 8).

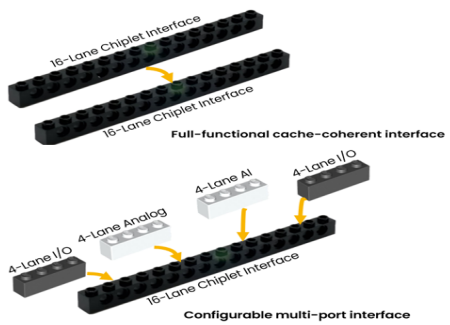

Figure 8. Configurable chiplet interface concept

III. Conclusion

Understandably, chiplet packaging cost scales with bandwidth and performance capabilities (figure 9). Higher data rate capability requires higher interconnect density. Based on system requirements, a suitable packaging technology meeting performance and cost specifications can be explored. As discussed earlier, chip disaggregation schemes leading to interface function and performance requirements will define the package technology.

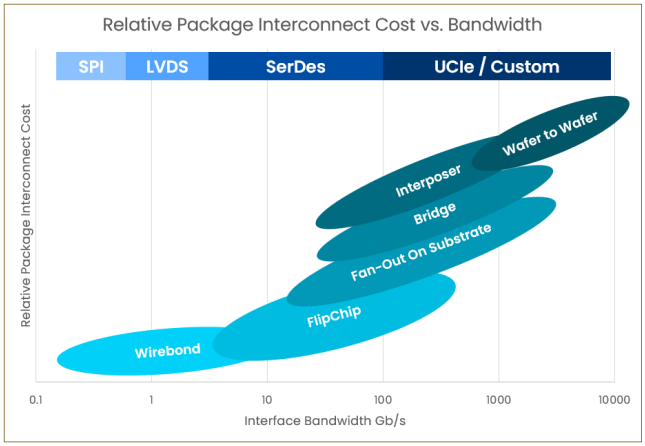

Figure 9. Package technology interface bandwidth capabilities. Cost comparison is for illustration only, and not to scale.

A chiplet strategy for microcontrollers and edge processors may be somewhat different than that for high-performance compute applications. The various processor types can be grouped with L3 cache and DDR interfaces in an advanced technology node Chip. Connections to other functional blocks in mature nodes can utilize existing off-package interfaces. Because of the close proximity of chiplets in the same package, these interfaces may be configured at a higher frequency than the off-package configuration.

Including a standard high-performance chiplet interface on CPU, GPU or NPU chiplets enables the possibility of “mixing and matching” processors for a variety of product configurations, requiring advanced flip-chip, fan-out, bridge, or interposer package technology. In this case, the enhanced device performance, as well as potential for chiplet reuse across multiple product generations may justify the advanced packaging cost. A configurable chiplet interface may enable chiplets requiring lower interface bandwidth and functionality to connect with larger, high-performance chiplets.

References

[1] S. Naffziger et al., “Pioneering Chiplet Technology and Design for the AMD EPYC™ and Ryzen™ Processor Families : Industrial Product,” 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 2021, pp. 57-70.

[2] W. Gomes, S. Morgan, B. Phelps, T. Wilson and E. Hallnor, “Meteor Lake and Arrow Lake Intel Next-Gen 3D Client Architecture Platform with Foveros,” 2022 IEEE Hot Chips 34 Symposium (HCS), Cupertino, CA, USA, 2022, pp. 1-40.

[3] 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.

[4] F. Zaruba, F. Schuiki and L. Benini, “A 4096-core RISC-V Chiplet Architecture for Ultra-efficient Floating-point Computing,” 2020 IEEE Hot Chips 32 Symposium (HCS), Palo Alto, CA, USA, 2020, pp. 1-24.

[5] A. Arunkumar et al., "MCM-GPU: Multi-chip-module GPUs for continued performance scalability," 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), Toronto, ON, Canada, 2017, pp. 320-332.

[6] C. A. Baddouh, M. Khairy, R. N. Green, M. Payer and T. G. Rogers, "Principal kernel analysis: A tractable methodology to simulate scaled GPU workloads", Proceedings of the International Symposium on Microarchitecture (MICRO), pp. 724-737, 2021.

[7] W. Liu, W. Heirman, S. Eyerman, S. Akram and L. Eeckhout, "Scale-model simulation", IEEE Computer Architecture Letters (CAL), vol. 20, no. 2, pp. 175-178, 2021.

[8] S. Eyerman, W. Heirman, Y. Demir, K. Du Bois and I. Hur, "Projecting performance for PIUMA using down-scaled simulation", Proceedings of the International High Performance Extreme Computing Conference (HPEC), pp. 1-7, 2020.

[9] S. Lakka, "Xilinx SSI technology concept to silicon development overview," 2012 IEEE Hot Chips 24 Symposium (HCS), Cupertino, CA, USA, 2012, pp. 1-22.

[10] D. D. Sharma, "System on a Package Innovations With Universal Chiplet Interconnect Express (UCIe) Interconnect," in IEEE Micro, vol. 43, no. 2, pp. 76-85, 1 March-April 2023.

[11] “Using LPSPI on KL28Z”, NXP Semiconductors Application note AN5320, Rev. 0, 08/2016

[12] “Introduction to the Zipwire Interface,” NXP Semiconductors Application Note AN5134, Rev. 0, May 2015

[13] M. Wade et al., "TeraPHY: A Chiplet Technology for Low-Power, High-Bandwidth In-Package Optical I/O," in IEEE Micro, vol. 40, no. 2, pp. 63-71, 1 March-April

[14] J. Sundaram, S. Gopal, T. P. Thomas, E. Burton and E. Ramirez, "A Reconfigurable Asynchronous SERDES for Heterogenous Chiplet Interconnects," 2021 22nd International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 2021, pp. 542-546.