

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NVIDIA Corporation,
Petitioner,

v.

Onesta IP, LLC,
Patent Owner.

Case IPR2026-00011
U.S. Patent No. 11,841,803

**DECLARATION OF DR. VIVEK SUBRAMANIAN REGARDING CLAIMS
1-4, 7-10 OF U.S. PATENT NO. 11,841,803**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS AND EXPERIENCE	1
III.	LEGAL PRINCIPLES	4
A.	Scope of Conclusions	5
B.	Anticipation	5
C.	Obviousness	5
D.	Claim Dependency	9
IV.	MATERIALS CONSIDERED	9
V.	OVERVIEW OF THE GROUNDS FOR FINDING CLAIMS 1-4, 7-10 OF THE '803 PATENT INVALID	9
VI.	OVERVIEW OF THE TECHNOLOGY	11
A.	Basic Structure of a Semiconductor Package	11
B.	The Development of “Chiplets”	14
C.	Chiplet-to-Substrate Connections	23
D.	The Use of Buses to Couple CPUs and GPUs	23
E.	Coherent Cache Hierarchies	24
VII.	OVERVIEW OF THE '803 PATENT	25
A.	Overview of the '803 Patent	26
B.	Overview of the Prosecution History for the '803 Patent	32
VIII.	OVERVIEW OF THE PRIOR ART	39
A.	Kelleher	40
B.	Collins	48
C.	Koker	51
D.	Sistla	54
IX.	LEVEL OF ORDINARY SKILL IN THE ART	56
X.	CLAIM CONSTRUCTION	58
A.	“Chiplet”	60
B.	“Dedicated for inter-chiplet communications”	61

C.	“A passive crosslink, wherein the passive crosslink is dedicated for inter-chiplet communication”	67
XI.	SPECIFIC GROUNDS FOR DEMONSTRATING THAT CLAIMS 1-4 AND 7-10 ARE INVALID	68
A.	GROUND I: Claims 1-4, 7, and 10 Would Have Been Obvious In View of the Combined Teachings of Kelleher and Collins	68
1.	Independent Claim 1	68
2.	Dependent Claim 2	117
3.	Dependent Claim 3	118
4.	Dependent Claim 4	125
5.	Dependent Claim 7	130
6.	Dependent Claim 10	131
B.	GROUND II: Claims 8 and 9 Would Have Been Obvious In View of the Combined Teachings of Kelleher, Collins, Koker, and Sistla	142
1.	Dependent Claim 8	142
2.	Dependent Claim 9	193
XII.	AVAILABILITY FOR CROSS-EXAMINATION	202
XIII.	RIGHT TO SUPPLEMENT	203
XIV.	JURAT.....	203

I, Vivek Subramanian, Ph.D., declare as follows:

I. INTRODUCTION

1. My name is Vivek Subramanian.
2. I am making this Declaration at the request of NVIDIA Corporation (“Petitioner”) as an independent expert in this proceeding before the United States Patent and Trademark Office and to provide testimony pertinent to Petitioner’s Petition for *Inter Partes* Review of U.S. Patent No. 11,841,803 (the “’803 patent”). I understand that Petitioner is challenging claims 1-4, 7-10 of the ’803 patent.

II. QUALIFICATIONS AND EXPERIENCE

3. My qualifications for setting forth the conclusions in this Declaration are summarized below. My curriculum vitae, which describes inter alia my education, experience, professional activities, my publications, and patents, is attached as **Appendix A**.

4. I received my Bachelor of Science in Electrical Engineering from Louisiana State University (in 1994). I also received my Master of Science (in 1996), and Ph.D. (in 1998), both in Electrical Engineering, from Stanford University.

5. As an undergraduate, I performed research on various aspects of microelectronics. My research as an undergraduate resulted in several publications in international journals. As a graduate student at Stanford University, I received an

Eastman Kodak fellowship from Kodak. While pursuing my Masters and Ph.D degrees, I worked as a research assistant in the Center for Integrated Systems at Stanford.

6. I have over two decades of experience in the semiconductor industry, including working on microprocessor and microcontroller systems including integrated memory. My work has included co-founding Matrix Semiconductor, where I helped pioneer high-density memory technology, and Locix Inc. Both of these companies produced packaged semiconductor products and I was directly involved in the development therein. For example, Matrix developed and successfully commercialized a memory with controller, and the latter involved implementing on-chip RAM and off-chip nonvolatile memory, and interconnect routing both intra-chip and off-chip for the same. At Locix, we developed and commercialized a wireless sensor product that integrated a multi-core CPU with on-chip and off-chip RAM and off-chip nonvolatile memory. I led all technology efforts in the development of this, including direct technical involvement in chip definition, place and interconnect routing, validation, and testing.

7. I have also been serving as a consultant to leading semiconductor companies on topics such as advanced memory design, silicon process integration, and RFID systems. I have advised multiple startups in printed electronics and energy conversion devices.

8. In addition to my industry work, I have held long-standing academic appointments. I have been a faculty member in the Department of Electrical Engineering & Computer Sciences at the University of California, Berkeley since 2000, progressing from Assistant Professor to Professor, and currently serving as Adjunct Professor. During my time as a Professor at the University of California, Berkeley, I was responsible for all the semiconductor device and technology courses offered within the department. I taught these courses routinely, including both undergraduate and graduate courses. I also actively performed research in the area of semiconductor packaging and have published journal articles and conference proceedings on the same.

9. In August 2018, I became a professor at École polytechnique fédérale de Lausanne (EPFL), where I direct the Micromanufacturing Science and Engineering Center (M2C) and serve as Associate Director of the Institute of Electrical and Micro Engineering. My teaching has covered integrated circuit fabrication, device physics, and emerging electronics, and I have mentored students and faculty worldwide through visiting appointments, thesis committees, and international collaborations.

10. I have also authored or co-authored over 200 technical papers in international journals and conferences and have been named inventor or coinventor on more than 60 patents covering several aspects of electronic systems, including

packaging. My work has been recognized with numerous best paper awards at leading conferences such as the IEEE Device Research Conference and IMAPS Microelectronics Conference. My contributions have also been honored with the IEEE Kiyo Tomiyasu Award, the NSF CAREER Award, and the Paul Rappaport Award for best paper in an IEEE EDS journal.

11. I have also held leadership roles in the field, including serving on the Executive Committee of the International Electron Device Meeting, chairing scientific boards, and contributing to program committees for major conferences in microelectronics and printed electronics.

12. I am being compensated for my work in this matter at my standard consulting rate of \$750 per hour. However, my conclusions are based on my own understanding of the prior art, the knowledge of a person of ordinary skill in the art, and the patent at issue. My compensation is not dependent on and in no way affects the content of my statements or conclusions in this Declaration, nor does it affect any testimony I give, nor is it dependent on the outcome of this proceeding.

13. I have no financial interest in the Petitioner. I similarly have no financial interest in the '803 patent and have had no contact with the named inventors of the '803 patent.

III. LEGAL PRINCIPLES

14. I am not an attorney. For the purposes of this Declaration, I have been informed about certain aspects of the law that are relevant to my conclusions. My understanding of the law was provided to me by Petitioner's attorneys.

A. Scope of Conclusions

15. I have been informed and understand that the petitioner in an *inter partes* review may request cancellation of claims as unpatentable only on grounds that such claims are anticipated or would have been obvious to a person of ordinary skill in the art before the earliest filing date (the "effective filing date") of the claim in the patent, and only on the basis of prior art. I have been informed and understand that a patent is prior art if it has a priority date before or was published or issued before the earliest claimed priority date of the '803 patent, which I understand is the filing date of the '803 patent (June 28, 2019). I have been informed and understand that a publication is prior art if it was published before the earliest claimed priority date of the '803 patent.

B. Anticipation

16. I understand that, for a patent claim to be "anticipated" by the prior art, each and every requirement of the claim must be found, expressly or inherently, in a single prior art reference as recited in the claim.

C. Obviousness

17. I understand that a claim is unpatentable if it would have been obvious to a person of ordinary skill in the art before the earliest filing date (the “effective filing date”) of the claim in the patent. I understand that a claim could have been obvious based on the teachings or suggestions of a single prior art reference or from a combination of two or more prior art references. I understand that an obviousness analysis requires an understanding of the scope and content of the prior art, any differences between the alleged invention and the prior art, and the level of ordinary skill in evaluating the pertinent art.

18. I further understand that a claim would have been obvious if it unites old elements with no change to their respective functions or alters prior art by mere substitution of one element for another known in the field, and that combination yields predictable results. Also, I understand that obviousness does not require physical combination/bodily incorporation, but rather consideration of what the combined teachings would have suggested to persons of ordinary skill in the art at the time of the alleged invention.

19. In connection with obviousness, I have been informed that there must have been some reason or motivation that would have led a person of ordinary skill in the art to combine or modify the relevant teachings in the prior art to obtain the claimed invention, and one of ordinary skill in the art must have had a reasonable expectation of success in doing so.

20. While it may be helpful to identify a reason for this combination, I understand that there is no rigid requirement of finding an express teaching, suggestion, or motivation to combine within the references. If a person of ordinary skill in the art can implement a predictable variation, obviousness likely bars its patentability. For the same reason, if a technique has been used to improve one device and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique would have been obvious. I understand that a claim would have been obvious if a person of ordinary skill in the art would have had reason to combine multiple prior art references or add missing features to reproduce the alleged invention recited in the claims.

21. In sum, I understand that a determination of whether a claim would have been obvious is based upon a variety of considerations, including, among others:

- whether the teachings of the prior art references disclose known concepts combined in familiar ways, which, when combined, would yield predictable results;
- whether a person having ordinary skill in the art could implement a predictable variation, and would see the benefit of doing so;
- whether the claimed elements represent one of a limited number of known design choices, and would have a reasonable expectation of success by those skilled in the art;

- whether a person having ordinary skill would have recognized a reason to combine known elements in the manner described in the claim;
- whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- whether the alleged invention applies a known technique that had been used to improve a similar device or method in a similar way.

22. I also understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the patent being considered. I also understand that one of ordinary skill in the art has ordinary creativity, and is not an automaton.

23. I further understand that certain factors, commonly referred to as secondary considerations, may support or rebut the obviousness of a claim. I also understand that secondary considerations cannot be used to overcome anticipation. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art before the effective filing date of the patent, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus, that is, a connection between any such secondary considerations and the alleged invention. I

also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

24. I reserve the right to respond to any allegations that may be put forth by Patent Owner in this proceeding that any secondary considerations tend to rebut the obviousness of any claim of the '803 patent discussed in this Declaration.

D. Claim Dependency

25. I understand that a patent may include two types of claims, independent claims, and dependent claims. An independent claim stands alone and includes only the limitations it recites. A dependent claim can depend from an independent claim or another dependent claim. I understand that a dependent claim includes all the limitations that it recites in addition to all of the limitations recited in the claim or claims from which it depends.

26. The analysis in this Declaration is in accordance with the above-stated legal principles.

IV. MATERIALS CONSIDERED

27. In preparing this Declaration, I reviewed and considered the materials cited in this Declaration and in **Appendix B**, as well as any document cited in this Declaration.

V. OVERVIEW OF THE GROUNDS FOR FINDING CLAIMS 1-4, 7-10 OF THE '803 PATENT INVALID

28. It is my conclusion that the claims 1-4 and 7-10 are invalid on the following grounds.

- **Ground I:** The combination of Kelleher and Collins makes Claims 1-4, 7, and 10 obvious under 35 U.S.C. §103.
- **Ground II:** The combination of Kelleher, Collins, Koker, and Sistla makes Claims 8 and 9 obvious under 35 U.S.C. §103.

29. These grounds demonstrate in detail that claims 1-4 and 7-10 are unpatentable for being obvious in view of the prior art. A copy of these claims is attached as **Appendix C**.

30. I provide more explanation as to the specific grounds in Section XI (Specific Grounds for Finding Claims 1-4, 7-10 of the '803 patent Invalid). Below I provide below a brief overview of the references in the grounds.

31. **Kelleher:** U.S. Patent No. 7,598,958, entitled “Multi-Chip Graphics Processing Unit Apparatus, System, and Method” issued on October 6, 2009, to Brian M. Kelleher (“Kelleher”). EX1004-Kelleher, Cover. Kelleher is directed to “[a] multi-chip graphics system” that “includes a master chip and a slave chip coupled by an interlink. The slave chip performs a graphics processing operation in parallel with the master chip.” EX1004-Kelleher, Abstract. I discuss Kelleher in greater detail in Section XI.A (Ground I).

32. **Collins:** U.S. Patent No. 11,610,862, entitled “Semiconductor Packages with Chiplets Coupled to a Memory Device” was filed on September 28,

2018, to Andrew Collins and Jianyong Xie (“Collins”). EX1005-Collins, Cover. Collins is directed to “a semiconductor package” that includes “an interconnect structure.” EX1005-Collins, Abstract. I discuss Collins in greater detail in Section XI.A (Ground I).

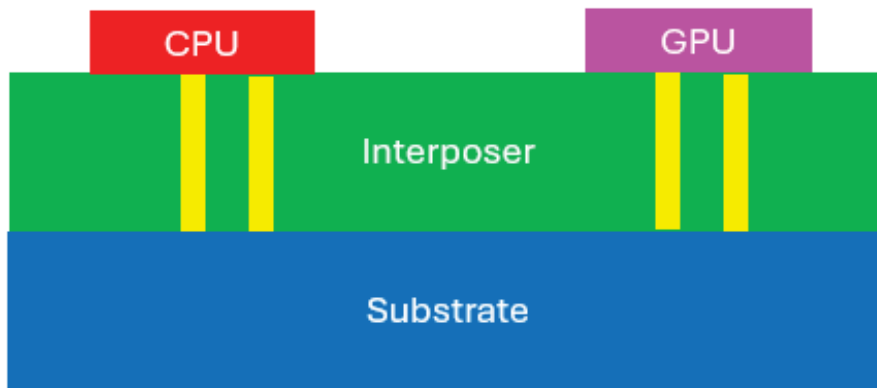
33. **Koker:** U.S. Patent No. 10,909,652, entitled “Enabling Product SKUs Based On Chiplet Configurations” was filed on March 15, 2019 to Atlug Koker et. al. (“Koker”). EX1031-Koker, Cover. Koker is directed to “[a] disaggregated processor package” having “a standard physical interconnect for chiplets that can enable the chiplet to interface with a fabric or bridge interconnect.” EX1031-Koker, Abstract. I discuss Koker in greater detail in Section XI.B (Ground II).

34. **Sistla:** U.S. Published Application No. 2009/0037658, entitled “Providing an Inclusive Shared Cache Among Multiple Core-Cache Clusters” was filed on July 31, 2007, and published on February 5, 2009 to Krishnakanth Sistla (“Sistla”). EX1047-Sistla, Cover. Sistla is directed to “a core-cache unit [that] is used as a building block and multiple such units are joined to increase the number of cores in the processor. The core-cache unit can contain two or more cores with an appropriately sized shared cache.” EX1047-Sistla, [0008]. I discuss Sistla in greater detail in Section XI.B (Ground II).

VI. OVERVIEW OF THE TECHNOLOGY

A. Basic Structure of a Semiconductor Package

35. As shown and discussed below, a typical semiconductor package includes several layers and components.



36. A **central processing unit (“CPU”)**, shown above in red, is widely regarded as the primary computational engine of a computer system. The CPU interprets and processes instructions from software, performs arithmetic and logical operations, and orchestrates the activities of peripheral subsystems. Since its commercial introduction in the 1970s, CPUs have evolved from single-core, fixed-function processors into highly integrated, multi-core architectures capable of executing billions of instructions per second.

37. A **graphics processing unit (“GPU”)**, shown above in pink, is a specialized processor optimized for highly parallel, data-intensive workloads. GPUs were originally developed to accelerate rendering in computer graphics, and are architected with a large number of smaller, efficient cores capable of executing thousands of threads concurrently. This parallelism makes GPUs particularly well-

suited for modern high-performance computing applications, including artificial intelligence (AI) model training and inference, large-scale data analytics, and scientific simulations. GPUs entered broad commercial awareness in 1999, when NVIDIA introduced the first mass-market GPU chip, establishing a new class of processors distinct from general-purpose CPUs. EX1044-McClanahan, 2.

38. One known packaging approach, acknowledged in the patent, involved mounting multiple integrated circuit dies side-by-side on an **interposer**, which is shown above in green. An interposer is an intermediate substrate positioned between active dies (e.g., CPU and GPU) and the package substrate, shown in blue. It provides fine-pitch conductive routing channels that enable high-bandwidth, low-latency communication between the dies. In addition to lateral interconnects, interposers frequently incorporate vertical conductive structures known as vias (shown above in yellow), which transmit power and signals between the dies and the underlying package substrate. When fabricated from silicon and extending completely through the interposer layer, these structures are referred to as through-silicon vias (“TSVs”). This technology enables dense, three-dimensional integration of components, reducing interconnect length, improving signal integrity, and supporting the high data rates required in advanced heterogeneous computing systems.

39. Interconnection between semiconductor components has long been a fundamental aspect of system design. As I explain throughout this Section, prior to June 2019, numerous interconnect technologies—such as buses, bridges, and high-speed links—were well established.

40. This basic structure of a semiconductor package that I describe above was well known within the art by 2019.

B. The Development of “Chiplets”

41. The semiconductor industry historically relied on Moore’s Law to deliver performance gains by scaling transistor dimensions. As Kannan explains, “[i]ndustry has pushed to keep the scaling of transistors to be in line with Moore’s law... every 18 months, chip performance would double.” EX1011-Kannan, 3. This scaling allowed designers “to pack more transistors and hence more logic within the same area” and even improve frequency without increasing power. EX1011-Kannan, 3. However, “around the years of 2005–2007,” the industry “could no longer scale transistors down and utilize all of the additional transistors without sacrificing an increase in power consumption.” EX1011-Kannan, 3. These limitations drove the shift to multi-core processors and, eventually, to disaggregated chiplet-based designs.

42. Larger dies suffer from yield loss because “[d]efects can appear on the wafer during the manufacturing process... [i]f a defect is present within the

boundaries of the die, it renders the die inoperable.” EX1011-Kannan, 10. This economic pressure created strong incentives to “replace larger chips with combinations of smaller chips” and integrate them into a single system. EX1011-Kannan, 10.

43. As such, the move away from monolithic packages to chiplets was driven by economics and scaling realities widely discussed in the industry. EX1021-ODSA, 6-7. For example, Lau explains:

Unfortunately, the end of Moore’s law is fast approaching and it is more and more difficult and costly to reduce the feature size (to do the scaling) to make the SoC. Heterogeneous integration contrasts with SoC. Heterogeneous integration uses packaging technology to integrate dissimilar chips, photonic devices, or components (either side-by-side, stack, or both) with different materials and functions, and from different fabless design houses, foundries, wafer sizes, feature sizes and companies into a system or subsystem. ...

EX1010-Lau, at v.

44. The Open Domain-Specific Architecture WorkGroup (ODSA) in 2018 explained that “[h]eterogeneous integrated systems offer a new design option” in which components “are designed and implemented on separate die referred to as chiplets.” EX1021-ODSA, 5-7. As shown below, CPU chiplets were already widely known.

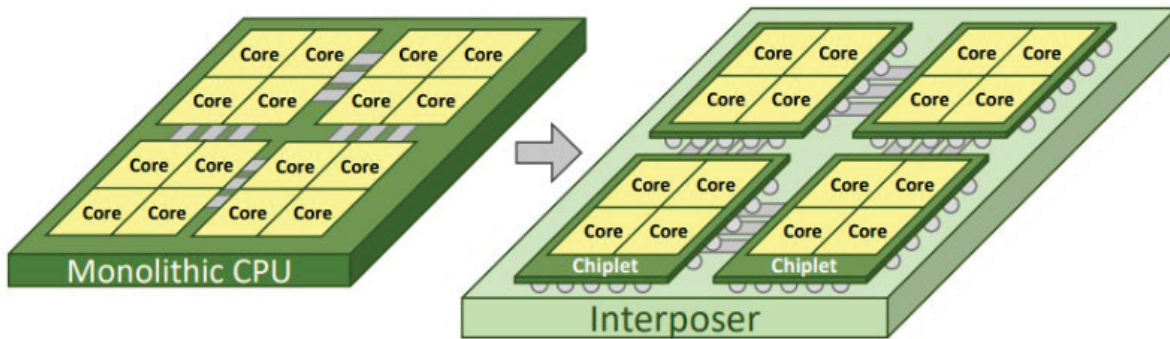


Fig. 1. Transition from monolithic manycore CPU to interposer-based 2.5D system with multiple chiplets

EX1020-Stow, Figure 1.

45. Indeed, chiplets were well-known as a practical way to build larger systems with improved yield and flexible scaling. EX1005-Collins, 2:62-7:10; EX1031-Koker, 62:11-25. Collins describes “a plurality of chiplets” mounted on a substrate and explains that “[u]tilizing chiplets may provide for more good dies per wafer... and greater flexibility to scale the die complex size.” EX1005-Collins, 2:62-7:10. Koker shows that “a disaggregated compute system can include a parallel processor in which the various components of the parallel processor SOC are distributed across multiple chiplets,” and adds that “[e]ach chiplet can be separately manufactured using different process technologies,” which confirms heterogeneous integration across nodes as routine design practice. EX1031-Koker, 62:11-25.

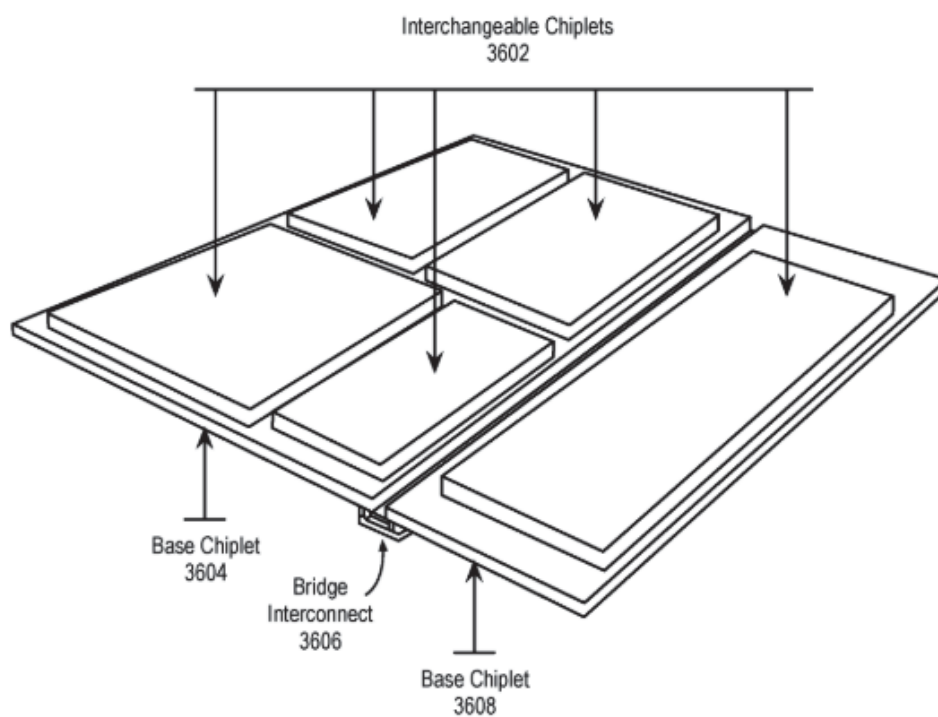


FIG. 36

EX1031-Koker, Figure 36.

46. Prior to 2019, companies were already developing chiplet-based architectures. For example, ODSA reported that “Intel, AMD and Xilinx address a complete stack” for “multi-chiplet system[s]”. EX1021-ODSA, 6-7; *see also* EX1010-Lau, 19-20.

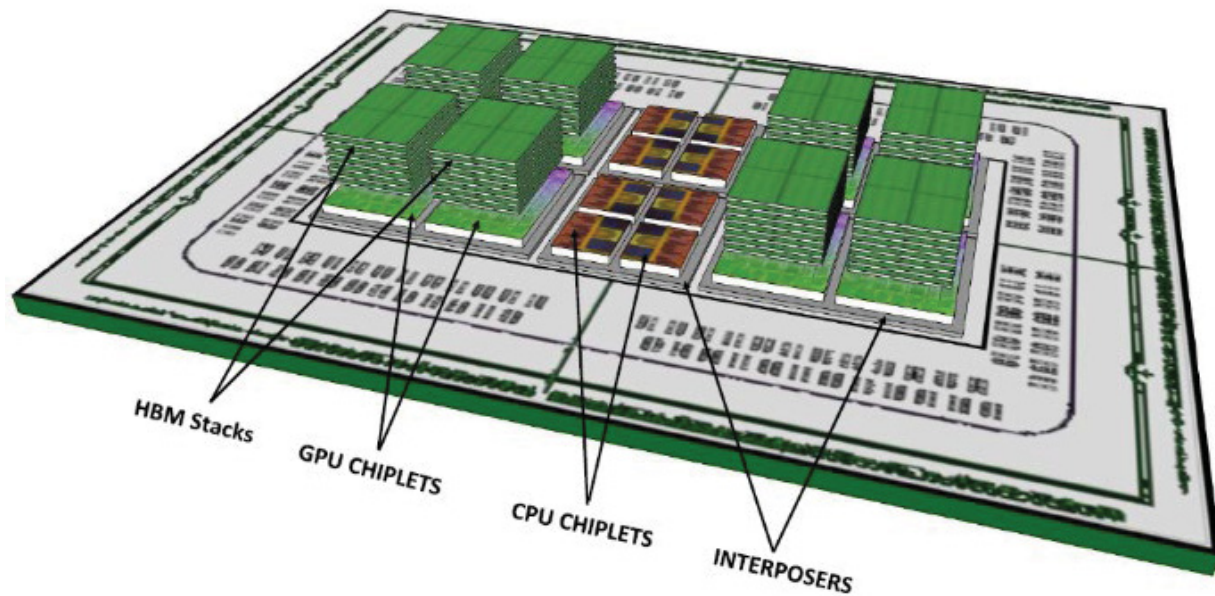


Figure 2. Exascale Heterogeneous Processor (EHP)

EX1045-Vijayaraghavan, 86.

A. Chiplet-to-Chiplet Connections

47. Chiplets were commonly coupled to each other through known interconnect elements. EX1005-Collins, 4:47-59, 6:23-35. Manufacturing techniques and package structures enabling these systems were widely known as well. EX1004-Kelleher, 5:65-6:23; EX1014-Elsherbini, [0096]; EX1010-Lau, 19-21; EX1021-ODSA, 6-7. Kannan notes that “[a] cheaper and more feasible way of approaching multi-chip integration is using a silicon interposer to connect each of the components. This approach is known as 2.5D stacking.” EX1011-Kannan, 1. “A silicon interposer is essentially a large piece of silicon die with transistors and metal

layers that serves as a base for the package,” where “[m]ultiple chips are placed face-down... [and] connected to the interposer through micro-bumps.” EX1011-Kannan, 1. The interposer’s wiring layers “are used to make chip-to-chip connections,” enabling high-bandwidth, low-latency communication without the overhead of traditional off-chip I/O. EX1011-Kannan, 1, 7. Unlike 3D stacking, which “has several obstacles... the main one being thermal issues,” 2.5D integration avoids severe thermal constraints. EX1011-Kannan, 1, 7. Below is an example of a 2.5D assembly.

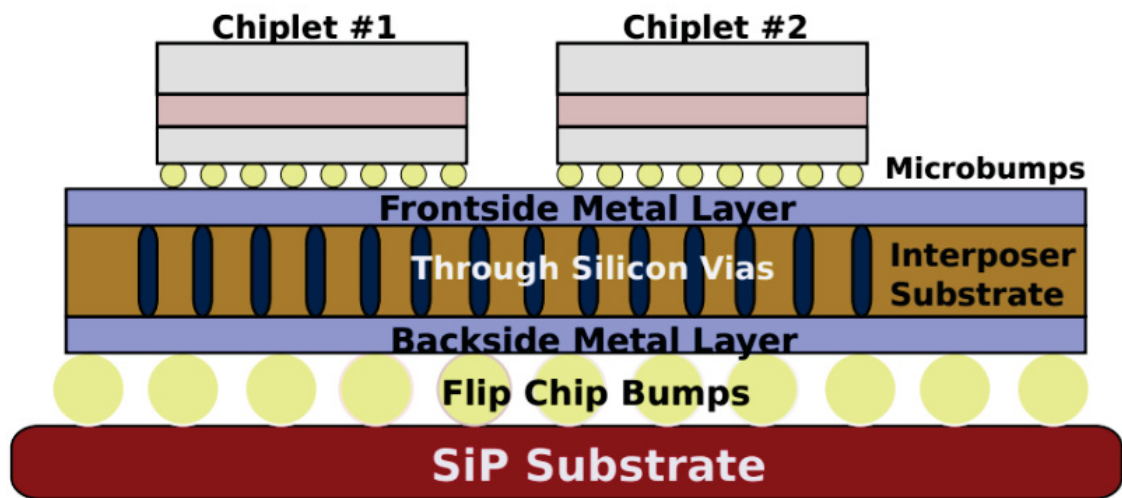


Figure 1: The cross-section of a 2.5D system.

EX1006-Coskun, Figure 1; *see also* EX1006-Coskun, *1.

48. In addition, Figure 3 of Kelleher shows “an edge-to-edge multi-chip module 313” with two GPU chips, and Figure 4 of Kelleher shows GPU chips that are “flip-chip mounted” on a package. EX1004-Kelleher, 5:39-42, 5:66-6:23.

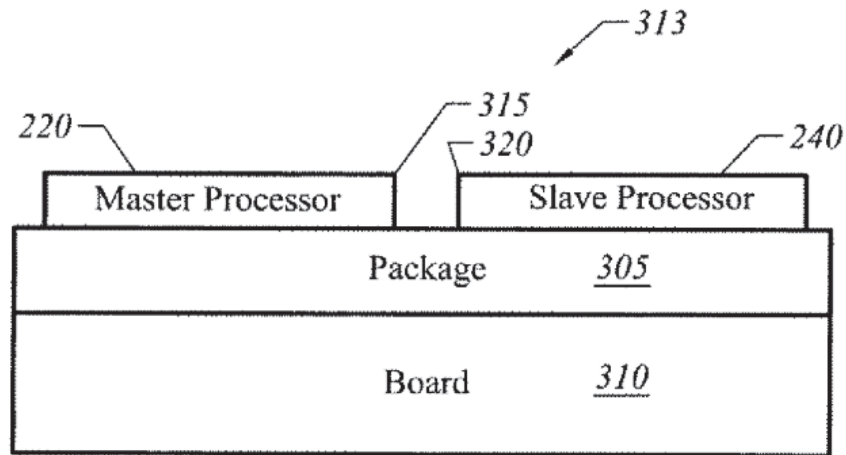


FIG. 3

EX1004-Kelleher, Figure 3.

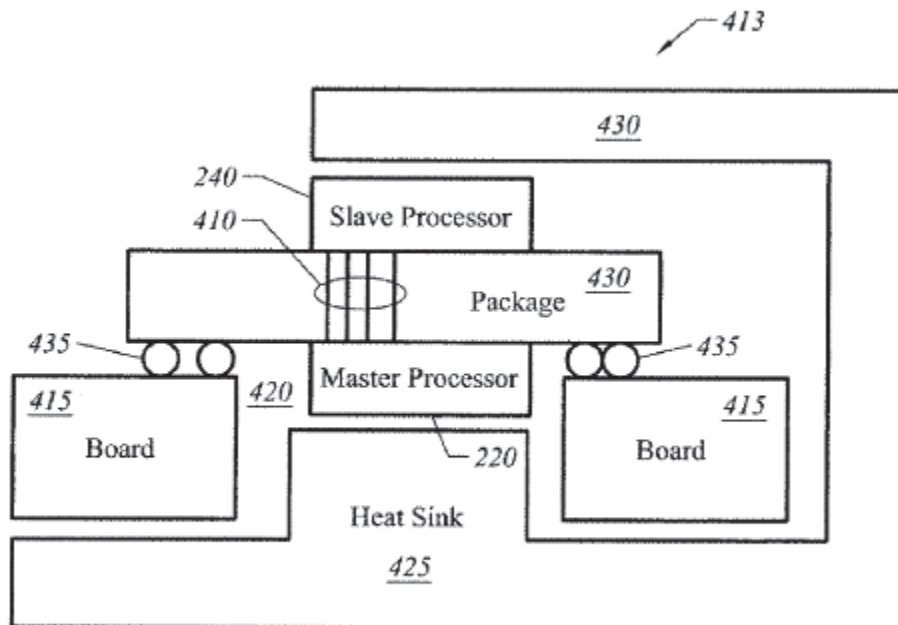


FIG. 4

EX1004-Kelleher, Figure 4.

49. One known connection between chiplets was Intel's "embedded multi-die interconnect bridge...to replace the TSV-interposer in heterogeneous integration

systems.” EX1010-Lau, 19-20. Intel’s embedded multi-die interconnect bridge (EMIB) was known to allow for communication between chips in a multi-chip package. *See Generally* EX1037-Liu. Collins also discloses the use of EMIBs to connect chiplets, noting that its “interconnect structure can further couple one or more of the chiplets to one or more of the other chiplets.” EX1005-Collins, 6:23-35.

Collins also explains that:

[T]he interconnect structure 108 includes one or more embedded multi-die interconnect bridges (EMIBs), where each of the EMIBs are embedded within the substrate assembly 102. The substrate assembly 102 includes conductive elements that couple the chiplets 104 and the memory device 106 to the EMIBs. Each of the EMIBs include a piece of passive silicon with conductive elements located within the passive silicon for conducting signals.

EX1005-Collins, 3:50-59.

50. The choice between whether an inter-chiplet connection was passive or active was also known and considered in the art. For example, Coskun emphasized consideration of “both passive and active links” and “five different link types,” indicating that there was a well-known set of inter-chiplet link choices. EX1006-Coskun, Abstract, *1.

51. Passive and active interposers and their tradeoffs were also widely discussed. EX1006-Coskun, *3 §3.3; EX1020-Stow, 728, 731-732. For example, Coskun stated that “[p]assive interposers cost less due to their lower manufacturing

cost and higher yield,” and that “[a]ctive interposers allow for repeaters and flip-flops” that enable “better link bandwidth and latency.” EX1006-Coskun, *3 §3.3. Stow further explained that passive interposers are used for “interconnect-only” whereas active interposers include “embedded routers and low latency repeated wires.” EX1020-Stow, 731-732.

52. In 2015, Kannan explained that interposers were “passive, i.e. they do not provide transistors on the interposer silicon layer. Only metal routing between chips and TSVs for signals entering/leaving the chip are provided.” EX1011-Kannan, 6. Passive interposers were already widely used in commercial products, such as “AMD’s High-Bandwidth Memory” systems, which “combine 3D-stacked memory with a CPU/GPU SoC die with an interposer.” EX1011-Kannan, 6.

53. Several techniques for implementing inter-chiplet communications were also widely known. For example, Coskun introduced a “gas station link” that can “‘refuel’ a passive link using repeaters and/or flip-flops that are inside other chiplets along the way,” which demonstrates a known technique to extend reach over passive routes. EX1006-Coskun, *3 §3.3. Coskun reported “two implementation schemes for a chiplet-to-chiplet link,” including a path that “uses Chiplet #2 as a gas station” and a direct path. EX1006-Coskun, *3 §3.3.

54. Moreover, commercial products documented the use of passive silicon interposers to attach chips to a stacked memory at very high bandwidth.

EX1046-NVIDIA-P100, 16. For example, it was known that “[a] passive silicon interposer” could be “used to connect...memory stacks and [a] GPU die.” EX1046-NVIDIA-P100, 16.

55. Researchers also described active interposer stacks and chiplet-rich APUs that combine CPU and GPU resources with in-package memory. EX1045-Vijayaraghavan, 86. Vijayaraghavan reported “a balanced mix of CPU and GPU compute resources integrated with in-package 3D DRAM” and “a modular ‘chiplet’ design” with “chiplets stacked on other active interposer chips.” EX1045-Vijayaraghavan, 86.

C. Chiplet-to-Substrate Connections

56. Chiplets were commonly coupled to the package through known structures. EX1005-Collins, 4:47-59, 6:23-35. For example, Collins states that “[t]he interconnect elements couple the chiplet[s]...to the conductive elements of the substrate assembly,” including “a ball grid array, a land grid array, a column grid array, or some combination,” which reflects known packaging options to attach the chiplets to a substrate. EX1005-Collins, 4:47-59. Further, Elsherbini details the use of copper “conductive pillars” to connect components to the package substrate. EX1014-Elsherbini, [0096].

D. The Use of Buses to Couple CPUs and GPUs

57. Industry practice has long recognized heterogeneous systems that couple CPUs and GPUs over high-speed buses. EX1004-Kelleher, 3:6-27. For example, Kelleher teaches that “[a] central processing unit is coupled to a graphics processing unit chip through a suitable bus interface” and further explains that “a bridge such as a bridge chipset, is used to couple CPU to GPU” with “a front side bus” to the bridge and a “(PCI)-Express bus” to the GPU. EX1004-Kelleher, 3:6-27.

E. Coherent Cache Hierarchies

58. It was well known that multi-chiplet systems required mechanisms to ensure fast and consistent data sharing across chiplets in a multi-chip architecture. EX1031-Koker, 17:46-58, 18:18-36; EX1022-Milic, 129-130. This need was particularly important in systems involving GPU chiplets, where large volumes of data are processed in parallel and must be accessed and updated across different chiplets in real time. One well-understood solution to this problem was the use of a cache memory hierarchy, in which each chiplet maintains its own local cache for fast access to frequently used data while also participating in a shared cache system that spans multiple chiplets. Indeed, Coherent cache hierarchies and coherence protocols were well understood in multi-core and heterogeneous systems.

59. In fact, industry and academic sources consistently modeled such hierarchies. For example, Arunkumar explains: “We model a multi-level cache hierarchy with a private L1 cache per SM and a shared L2 cache. Caches are banked

such that they can provide the necessary parallelism to saturate DRAM bandwidth.

We model software based cache coherence in the private caches, similar to state-of-the-art GPUs.” EX1033-Arunkumar, *4-*5. Similarly, Koker describes a “cache memory hierarchy” that includes “at least one level of instruction and data cache within each processor core and one or more levels of shared mid-level cache, such as a Level 2 (L2), Level 3 (L3), Level 4 (L4), or other levels of cache, where the highest level of cache before external memory is classified as the LLC.” EX1031-Koker, 40:29-38. Koker further notes that “cache coherency logic maintains coherency between the various cache units.” EX1031-Koker, 40:29-38. As such, combining local caches with shared higher-level caches was a standard design approach.

60. It was well-known that such hierarchies improve performance by reducing the time it takes to access data and by minimizing the need to fetch data from slower, off-chip memory. Further, techniques like cache snooping and directory-based protocols were well understood and widely applied. For example, Koker states that “[c]oherency is maintained for data and instructions stored in the various caches and system memory via inter-core communication over a coherence bus,” and that “a cache snooping protocol is implemented.” EX1031-Koker, 17:46-58.

VII. OVERVIEW OF THE '803 PATENT

A. Overview of the '803 Patent

61. The '803 patent describes a conventional semiconductor package having GPU chiplets. EX1001-'803, 1:6-9, 1:45-46, 1:62-67, Figure 1. However, the '803 patent purports to have claimed an “improve[d] system performance by using GPU chiplets while preserving the current [CPU] programming model.” EX1001-'803, 1:12-2:4.

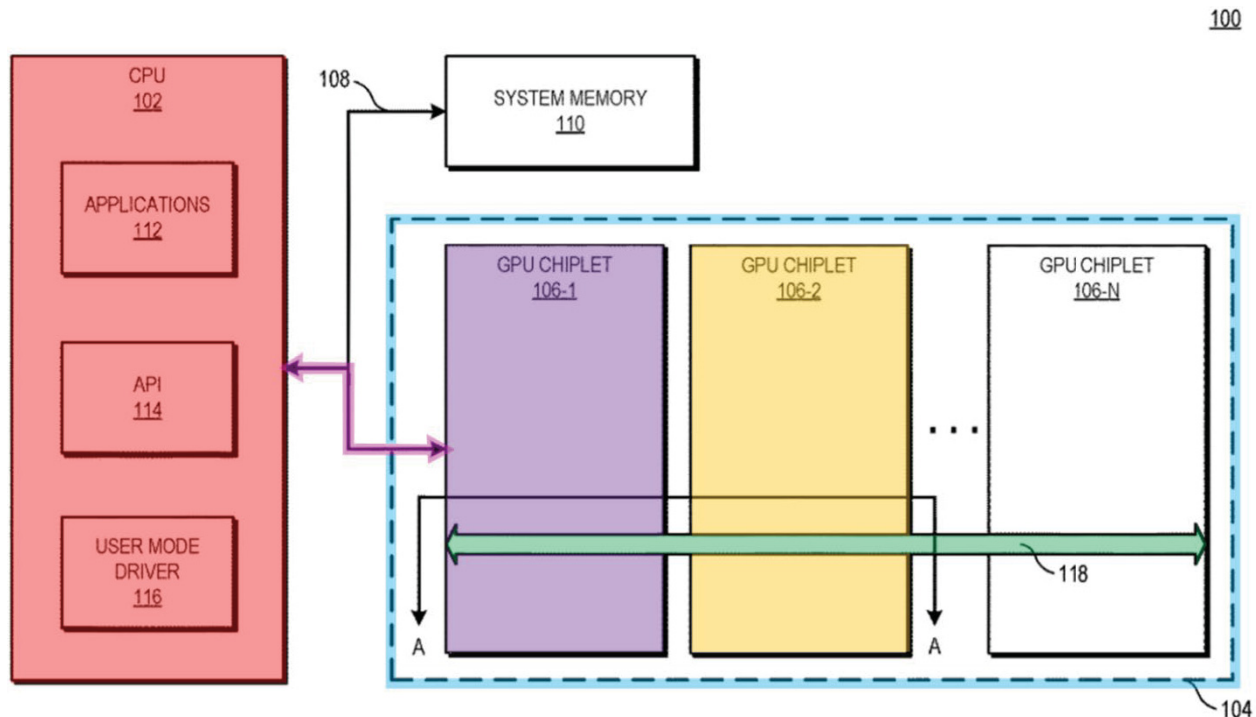
62. In particular, the '803 patent describes its invention as solving problems associated with “bring[ing] chiplet design methodology to GPU architectures.” In particular, the patent notes:

Conventional monolithic die designs are becoming increasingly expensive to manufacture. Chiplets have been used successfully in CPU architectures to reduce cost of manufacture and improve yields, as its heterogeneous computational nature is more naturally suited to separate CPU cores into distinct units that do not require much inter-communication. GPU work by its nature includes parallel work. However, the geometry that a GPU processes includes not only sections of fully parallel work but also work that requires synchronous ordering between different sections. Accordingly, a GPU programming model that spreads sections of work on different threads is often inefficient because the parallelism is difficult to distribute across multiple different working groups and chiplets as it is difficult and expensive computational to synchronize the memory contents of shared resources throughout the entire system to provide a coherent view of the memory to applications. Additionally, from a logical point of view, applications are written with the view that the system only has a single GPU. That is, even though a conventional GPU includes many GPU cores, applications

are programmed as addressing a single device. So, it has been historically challenging to bring chiplet design methodology to GPU architectures.

EX1001-'803, 1:45-67.

63. Below is an annotated version of Figure 1 of the '803 patent.¹



EX1001-'803, Figure 1.

64. As shown above, the system in the '803 patent includes multiple **GPU chiplets 106-1, 106-2, ... 106-N** (shown purple, gold, and white) instead of one monolithic GPU. The specification lists three requirements for a component to be considered a “chiplet”:

¹ All color annotations and emphasis in this Declaration have been added unless noted otherwise.

- 1) a chiplet includes an active silicon die containing part of the computational logic used to solve a full problem (i.e., the computational workload is distributed across multiples of these active silicon dies);
- 2) chiplets are packaged together as a monolithic unit on the same substrate; and
- 3) the programming model preserves the concept that these separate computational dies are a single monolithic unit (i.e., each chiplet is not exposed as a separate device to an application that uses the chiplets for processing computational workloads).

EX1001-'803, 2:41-53.

65. As shown above, the system in the '803 patent uses a **“dedicated” passive crosslink** (shown in green) to transmit signals between the GPU chiplets. The '803 patent notes that the “passive crosslink 118 does not contain any through silicon vias (TSVs) [(shown in brown below)]. In this manner, the passive crosslink 118 is a passive interposer die that communicably couples and routes communications between the GPU chiplets 106, thereby forming a passive routing network.” EX1001-'803, 4:16-21.

66. For example, the cross-sectional image of Figure 2, reproduced below, shows more details regarding the passive crosslink.

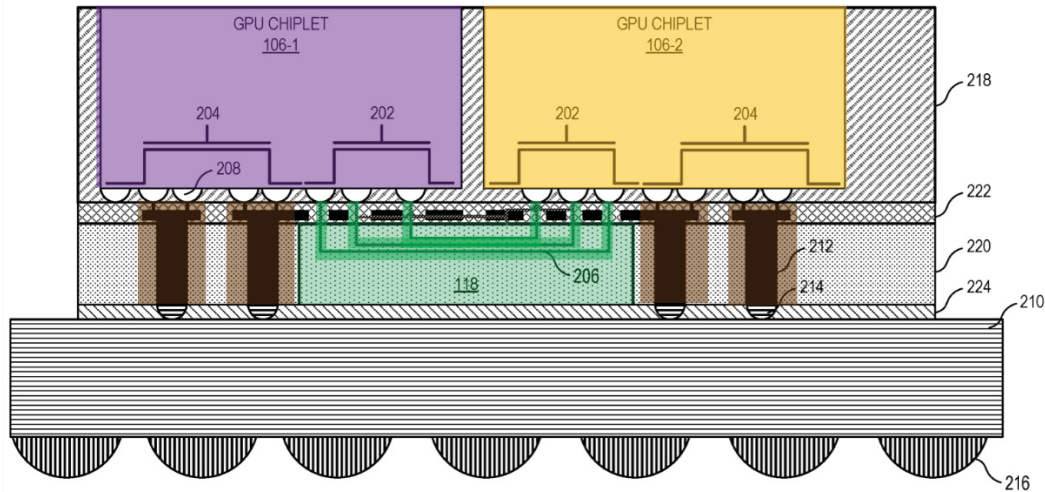


FIG. 2

EX1001-'803, Figure 2.

67. The '803 patent explains that the passive crosslink connects the two chiplets' physical device ("PHY") regions together to transmit chiplet-to-chiplet signals. In particular, the '803 patent states that "each GPU chiplet 106 is constructed with a physical device (PHY) region 202, which has various internal and external conductor structures dedicated to the transmission of chiplet-to-chiplet signals, and a non-PHY region 204, which has conductor structures that are tailored more to the conveyance of power and ground and/or chiplet-to-circuit board signals." EX1001-'803, 3:62-4:42. In this way, the '803 patent characterizes the passive crosslink as providing a "dedicated" communication pathway for chiplet-to-chiplet signals, as it notably does not include pathways for the transmission of

chiplet-to-non-chiplet signals, such as for connecting the chiplets to the base substrate. EX1001-'803, Abstract, 2:11-14, 3:57-4:2, 4:16-21, Claim 1.

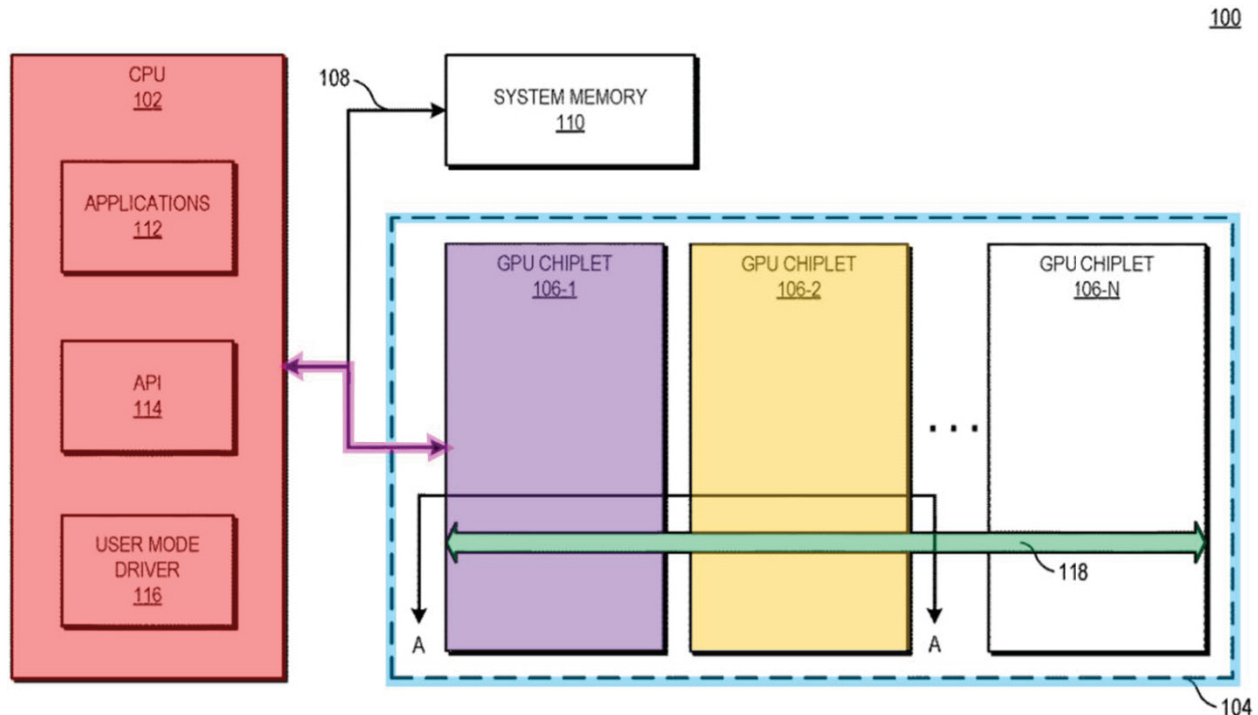
68. For example, the '803 patent explains that the chiplets are connected to the base substrate via conductive pillars (shown above in brown) and that power is routed around the passive crosslink:

The conductive pillars 212 connect signals between the GPU chiplets 106 and the substrate in areas where the HDCL die is not present (e.g., areas with vertical discrepancy between the GPU chiplets 106 and the circuit board 210), with empty spaces filled with epoxy or other gap fill materials. ***In this manner, power and input/output (I/O) lines of the non-PHY regions 204 are routed around the interposer die (i.e., passive crosslink 118) using redistribution (RDL) technology, thereby replacing usage of conventional TSVs.*** For example, as illustrated in the embodiment of FIG. 2, the GPU chiplets 106 and passive crosslink 118 are at least partially encapsulated in a molding material, such as the two molding layers 218 and 220 of FIG. 2. The conductive pillars 212 traverse a plurality of insulating layers, such as the molding layer 220 and a polymer layer 222. In various embodiments, the polymer layer 222 is a RDL layer designed to act as a stress buffer and/or an isolation film for enabling redistribution layer routing. The conductive pillars 212 include various conductor materials, such as copper and the like. Similarly, the solder interconnects 208 and interconnect structures 214 include materials using various solder compositions, such as tin-silver, tin-silver-copper, and the like.

EX1001-'803, 4:36-58.

69. As originally shown in Figure 1 (reproduced again below), the CPU (shown in red) communicates with the first GPU chiplet (shown in purple) via a bus

(shown in pink). That first chiplet then passes information received from the CPU to the remaining chiplets, for example, including the second chiplet shown above in gold. As I explained above, the first chiplet communicates with the second chiplet via the dedicated, passive crosslink. EX1001-'803, 3:41-49.



EX1001-'803, Figure 1.

70. According to the '803 patent, the architecture described above (having a CPU communicate with only one chiplet, and in turn, having that chiplet communicate with the rest of the chiplets) allows the CPU to view the separate GPU chiplets as a single monolithic GPU. EX1001-'803, 3:49-56. For example, the '803 patent states that “[f]rom the CPU’s point of view, it appears to be addressing a

“single die, monolithic GPU.” This allows for use of a large-capacity, multi-chiplet GPU that appears as a single device to an application.” EX1001-’803, 2:30-36.

71. Claim 1 is a representative claim describing the purported invention of the ’803 patent. Claim 1 recites:

1. A system, comprising:
a central processing unit (CPU) communicably coupled to
a first graphics processing unit (GPU) chiplet of a GPU
chiplet array, wherein the GPU chiplet array includes:

the first GPU chiplet communicably coupled to the CPU
via a bus; and

a second GPU chiplet communicably coupled to the first
GPU chiplet via a passive crosslink, wherein the passive
crosslink is dedicated for inter-chiplet communications.

EX1001-’803, Claim 1.

72. But as I describe below, all of the features of claims 1-4, 7-10 of the ’803 patent were well-known before the earliest claimed priority date.

B. Overview of the Prosecution History for the ’803 Patent

73. The ’803 patent issued from U.S. Application No. 16/456,287 (“’287 application”), which was filed on June 28, 2019. EX1001-’803, Cover.

74. On September 11, 2020, the Examiner rejected the filed claims. EX1003-FH, 874-875. First, the Examiner rejected the claims as anticipated by U.S. Publication No. 2019/0123022 (“Teig”). *See Generally* EX1027-Teig. EX1003-FH, 876-877 (9/11/2020 Non-Final Rejection, at 2-3). Second, the Examiner

rejected the claims as obvious over Teig in view of U.S. Publication No. 2001/0005873 (“Yasuda”). EX1003-FH, 877-878 (9/11/2020 Non-Final Rejection, at 3-4). *See Generally* EX1040-Yasuda.

75. On February 25, 2021, the Applicant argued that Teig did not disclose several aspects of the claims and that Yasuda did not remedy the deficiencies of Teig. In particular, the Applicant argued that Teig did not disclose “GPU chiplets” and a “passive crosslink [that] is dedicated for inter-chiplet communications.” EX1003-FH, 847 (2/25/2021 OA Response, at 8); EX1003-FH, 848 (2/25/2021 OA Response, at 9). I have reproduced the Applicant’s arguments below:

As shown, Teig generally mentions GPU cores. Nowhere does Teig disclose that the GPU cores are GPU chiplets. Also, even assuming arguendo that Teig’s GPU cores are comparable to the GPU chiplets of claim 1, nowhere does Teig disclose both that a first GPU core is communicably coupled to the CPU via a bus and a second GPU core is coupled to the first GPU chiplet via a passive crosslink. The cited sections of Teig fail to provide any details regarding how the GPU cores are coupled to the CPU and each other. Also, regarding the passive crosslink of claim 1, this claim requires that “the passive crosslink is dedicated for interchiplet communications”. In contrast, paragraph [0148] of Teig merely states, “the first IC die is face-to-face mounted with a passive interposer that electrically connects the die to circuits outside of the 3D chip or to other dies that are face-to-face mounted or back-to-face mounted on the interposer.” Nowhere in paragraph [0148] does Teig disclose that the passive interposer is dedicated for inter-chiplet communications.

EX1003-FH, 847 (2/25/2021 OA Response, at 8); *see also* EX1003-FH, 848 (2/25/2021 OA Response, at 9).

76. On May 17, 2021, the Examiner again rejected the filed claims. EX1003-FH, 745-746. The Examiner rejected the claims on the same grounds as the previous September 11, 2020 rejection. EX1003-FH, 747-751. In response to the Applicant's remarks regarding whether Teig disclosed "GPU chiplets," the Examiner noted that "there is no patentable distinction between ... GPU chiplets and the GPU cores disclosed in figure 9 of Teig." EX1003-FH, 749 (5/17/2021 Final Rejection, at 4). *See Generally* EX1027-Teig.

77. On April 28, 2022, the Examiner again rejected the filed claims.² EX1003-FH, 641-642 (4/28/2022 Non-Final Rejection, at 2). Specifically, the Examiner rejected the claims as obvious over Teig in view of U.S. Patent No. 10,475,147 ("Price"). EX1003-FH, 643-648. *See Generally* EX1041-Price.

78. On October 4, 2022, the Applicant argued that Teig did not disclose several aspects of the claims and that Price did not remedy the deficiencies of Teig. In particular, the Applicant argued that Teig and Price did not disclose "GPU chiplets" as defined by the specification because the alleged chiplets were "exposed

² I understand that there were intervening procedural mechanisms that allowed prosecution to re-open after the May 17, 2021 Final Rejection.

as separate devices to applications.” I have reproduced the Applicant’s argument below:

Third, Teig and Price do not teach or suggest a “GPU chiplet” as defined by Applicant’s specification. In particular, Applicant’s specification states “the term ‘chiplet’ refers to any device including, but is not limited to, the following characteristics: ... 3) the programming model preserves the concept that these separate computational dies are a single monolithic unit (i.e., each chiplet is not exposed as a separate device to an application that uses the chiplets for processing computational workloads).” Applicant’s Specification at [0012]. Teig is silent regarding such a programming model. The GPUs of Price are explicitly exposed as separate devices to applications. *See* Price at col. 2, 11. 60-66.

EX1003-FH, 611 (10/4/2022 OA Response, at 10). *See Generally* EX1027-Teig.

79. The Applicant also argued that Teig and Price did not disclose a “passive crosslink [that] is dedicated for inter-chiplet communication” because Teig’s interposer connected the CPU and GPU cores with the cache memory for those cores. I have reproduced the Applicant’s argument below:

There is no teaching in Teig or Price regarding “wherein the passive crosslink is dedicated for inter-chiplet communications.” The passive interposer of Teig is a physical layer placed between two dies, such as dies 905 and 910 of FIG. 9. *See* Teig at [0148] and FIG. 9. Such an interposer from Teig would not be considered by one of ordinary skill to be dedicated for inter-chiplet communications, as the passive interposer would be the connection between the CPU and GPU cores 922 and 924 and the LI caches 932 and 934 for those cores. *See* Teig at FIG. 9 and [0090]. If the communications were dedicated

to only such communications, then it appears that the CPU and GPU cores 922 and 924 of Teig would be unable to access the LI caches 932 and 934 for memory operations, which would result in a nonfunctional system.

EX1003-FH, 610-611 (10/4/2022 OA Response, at 9-10). *See Generally* EX1027-Teig.

80. On January 20, 2023, Examiner again rejected the filed claims. EX1003-FH, 555-556 (01/20/2023 Non-Final Rejection, at 2). Specifically, the Examiner rejected the claims as obvious over U.S. Publication No. 2007/0273699 (“Sasaki”) in view of Teig.³ EX1003-FH, 557-562. *See Generally* EX1042-Sasaki.

81. On July 5, 2023, the Applicant argued that Sasaki and Teig did not disclose several aspects of the claims. In particular, the Applicant “invok[ed] their right to be their own lexicographer,” and argued that Sasaki and Teig did not disclose “GPU chiplets” as defined by the specification. I have reproduced the Applicant’s argument below:

First, Sasaki and Teig do not teach or suggest a “GPU chiplet” as defined by Applicant’s specification in accordance with MPEP § 2111.01(III)(A). During the interview, the Examiner explained that he was interpreting the claim language using the broadest reasonable interpretation standard. However, the MPEP states that “An applicant is entitled to be their own lexicographer and

³ I understand that there was a typographical error in the Examiner’s rejection where the Examiner initially listed the rejection as unpatentable over Sasaki in view of Price (instead of Teig), which was later corrected. EX1003-FH, 336 (05/25/2023 Applicant-Initiated Interview Summary, at 2); EX1003-FH, 317 (7/5/2023 OA Response, at 7).

may rebut the presumption that claim terms are to be given their ordinary and customary meaning by clearly setting forth a definition of the term that is different from its ordinary and customary meaning(s) in the specification at the relevant time” and “Where an explicit definition is provided by the applicant for a term, that definition will control interpretation of the term as it is used in the claim.” MPEP § 2111.01(III)(A) (emphasis added). Because Applicant is invoking their right to be their own lexicographer, any interpretation of the term “GPU chiplet” from Applicant’s claims must be compatible with Applicant’s definition of the term and any broader interpretation, such as the one advanced by the Office Action, is improper.

In particular, Applicant’s specification states “the term ‘chiplet’ refers to any device including, but is not limited to, the following characteristics: ... 3) the programming model preserves the concept that these separate computational dies are a single monolithic unit (i.e. each chiplet is not exposed as a separate device to an application that uses the chiplets for processing computational workloads).” Applicant’s Specification at [0012]. As discussed during the interview, “any device including, but is not limited to, the following characteristics:” refers to a minimum set of characteristics which must be present to consider a device a “chiplet.” However, devices may include other characteristics. In Sasaki, the CPU addresses and communicates with both the first GPU and the second GPU. *See Sasaki*, Abstract. In other words, the GPUs of Sasaki are exposed as separate devices to applications. Teig is silent regarding the programming model of a chiplet. Therefore, the cited references, individually or in combination, do not teach or suggest a “GPU chiplet,” as in claim 1.

EX1003-FH 7/5/2023, 318 (OA Response, at 9).

82. The Applicant also argued that Sasaki and Teig did not disclose a “passive crosslink [that] is dedicated for inter-chiplet communication” because Teig’s interposer connected the CPU and GPU cores with the cache memory for those cores, and because in Sasaki, data is sent between the CPU and the second GPU, rather than between GPUs. I have reproduced the Applicant’s argument below:

Second, Sasaki and Teig individually or in combination, do not teach or suggest at least “a second GPU chiplet communicably coupled to the first GPU chiplet via a passive crosslink, wherein the passive crosslink is dedicated for inter-chiplet communications,” as in claim 1. The Office Action cites a passive interposer from paragraph [0148] of Teig regarding the claimed passive crosslink and states that “it would have been obvious to a person of ordinary skill in the art to modify Sasaki by coupling components via a passive interposer as taught by Teig in paragraph 148.” *See* Office Action at 2 and 3. However, this rejection ignores the language of the claim and appears to contradict the teachings of Sasaki. The passive interposer of Teig is a physical layer placed between two dies, such as dies 905 and 910 of FIG. 9. *See* Teig at [0148] and FIG. 9. Such an interposer from Teig would not be considered by one of ordinary skill to be dedicated for inter-chiplet communications, as the passive interposer would be the connection between the CPU and GPU cores 922 and 924 and the LI caches 932 and 934 for those cores. *See* Teig at FIG. 9 and [0090]. If the communications were dedicated to only such communications, then it appears that the CPU and GPU cores 922 and 924 of Teig would be unable to access the LI caches 932 and 934 for memory operations, which would result in a nonfunctional system. Further, in Sasaki, data is “sent from the CPU 300 to the second GPU 200”

“by the second-GPU interface 140 to the second GPU 200 via the second GPU connection bus 190.” *See* Sasaki at [0116]. One of ordinary skill would not understand this communication from Sasaki to be an inter-chiplet communication. Although the first GPU 100 of Sasaki is used as an intermediate step in the communication, the first GPU core 120 is not involved in the communication and instead this is a communication between the CPU 300 and the second GPU 200. If a communication link dedicated for inter-chiplet communications were inserted in place of second GPU connection bus 190 of Sasaki as proposed by the Examiner during the interview, such an arrangement would not allow this communication, potentially causing erroneous operation to the system of Sasaki.

EX1003-FH, 319 (7/5/2023 OA Response, at 10). *See Generally* EX1027-Teig.

83. On July 31, 2023, the Examiner allowed the claims. EX1003-FH, 285 (7/31/2023 NOA, at 2). The Examiner noted that the “[t]he primary reasons for allowance...is the combination with the inclusion in these claims that ‘wherein the GPU chiplet array includes: the first GPU chiplet communicably coupled to the CPU via a bus; and a second GPU chiplet communicably coupled to the first GPU chiplet via a passive crosslink, wherein the passive crosslink is dedicated for inter-chiplet communications’ (interpreted in view of paragraph 12 of the specification).” EX1003-FH, 285 (7/31/2023 NOA, at 2).

84. However, the Examiner did not have the benefit of the references I describe below in Grounds I-II.

VIII. OVERVIEW OF THE PRIOR ART

A. Kelleher

85. Kelleher, issued on October 6, 2009, discloses a multi-chip module having multiple GPU chiplets. EX1004-Kelleher, Abstract. In particular, Kelleher discloses:

[a] graphics processing unit chip...that may be...packaged as a multi-chip module having a master/slave pair with improved processing performance. A packaged multi-chip master/slave pair includes a high bandwidth interlink for transferring data from the master chip to the slave chip and returning processed data from the slave chip to the master chip. The master chip receives a graphics command string and generates a graphics output. The master chip utilizes the slave chip to process, in parallel, data for one or more stages of the graphics pipeline of the master chip.

EX1004-Kelleher, 2:7-17.

86. Kelleher's Figure 2 discloses a system very similar to the system that is claimed in the '803 patent:

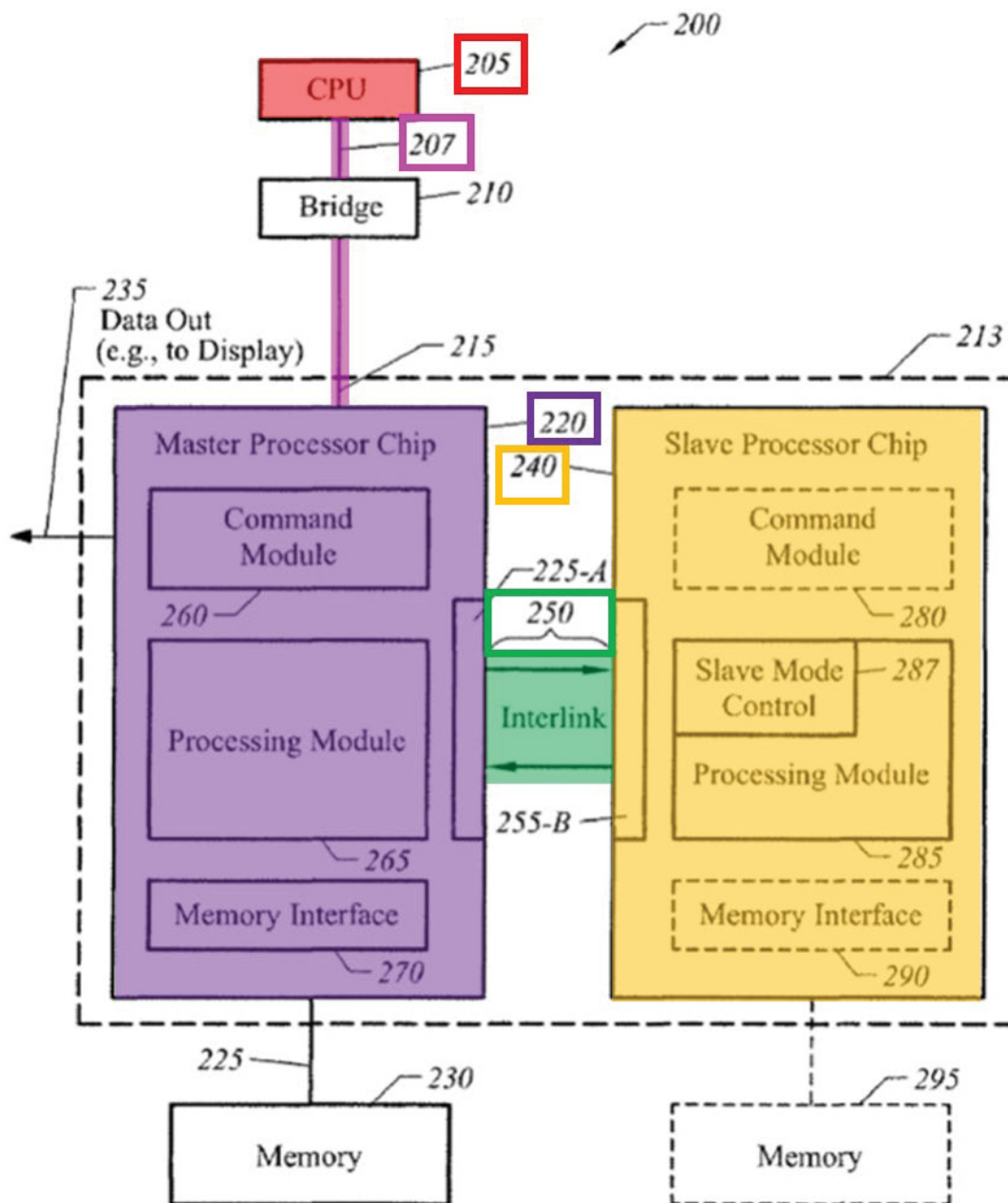


FIG. 2

EX1004-Kelleher, Figure 2. *See, e.g.*, EX1001-'803, 1:45-67; Figure 1.

87. Kelleher's "master GPU" chip and a "slave GPU" chip meet the '803 patent's 3-part definition of "chiplets." EX1004-Kelleher, Figure 2, 4:20-35; EX1001-'803, 2:41-53.

88. **"Chiplet" Definition - Part 1:** "a chiplet includes an active silicon die containing part of the computational logic used to solve a full problem (i.e., the computational workload is distributed across multiples of these active silicon dies)." EX1001-'803, 2:41-53.

89. Kelleher's master chip and slave chip both meet this requirement. For example, Kelleher explains:

GPU chip 220 includes a command module 260. Command module 260 receives a graphics command stream from CPU 205, reads and parses the command string, and distributes work to the other stages within GPU chip 220. **At least one of the stages in the graphics pipeline of GPU chip 220 is a processing module 265 that performs a processing operation that may be performed in parallel.** One example is a shader stage. Shader stages in high performance GPUs are typically arranged as groups of shader pipelines along with logic that permits a load balancing operation in which the shading work is distributed between the shader pipelines. Slave GPU chip 240 is packaged with GPU chip 220 as a multi-chip module 213 that may be used as a replacement for an individual GPU chip. ...

Slave GPU chip 240 includes a processing module 285 capable of performing a processing function that is compatible with that performed by processing module 265. This permits, for example, two GPU's from the same or related product lines to be used in system 200. Thus, for

example, slave GPU chip 240 may have its own command module 280 and memory interface 290. ...

GPU chip 220 acts as the master processor and utilizes processing module 285 as an additional processing resource that is used in parallel with processing module 265. In one embodiment slave GPU chip 240 also receives clock signals from GPU chip 220 to facilitate synchronization of slave GPU chip 240 to GPU chip 220. In the slave mode, a load balancing module (not shown) within processing module 265 determines a distribution of work to be performed by processing modules 265 and 285 and transfers a portion of the graphics data that it receives to processing module 285. Thus for example, **if processing modules 265 and 285 are substantially identical, about half of the processing that would ordinarily be performed by processing module 265 is off-loaded to processing module 285,** permitting an approximately two-fold improvement in performance (e.g., speed or quality), assuming that processing modules 265 and 285 have a similar number of operable processing elements.

EX1004-Kelleher, 3:35-4:19.

90. **“Chiplet” Definition - Part 2:** “chiplets are packaged together as a monolithic unit on the same substrate.” EX1001-’803, 2:41-53.

91. Kelleher’s master chip and slave chip both meet this requirement. For example, Kelleher states that “[s]lave GPU chip 240 is packaged with GPU chip 220 as a multi-chip module 213 that may be used as a replacement for an individual GPU chip.” EX1004-Kelleher, 3:46-49. Kelleher also states that “multi-chip module 213 may include a package adapted to be mounted onto a board.” EX1004-Kelleher,

3:49-51. Additionally, Figure 3 (reproduced below) shows both a master and slave chip mounted on the same substrate.

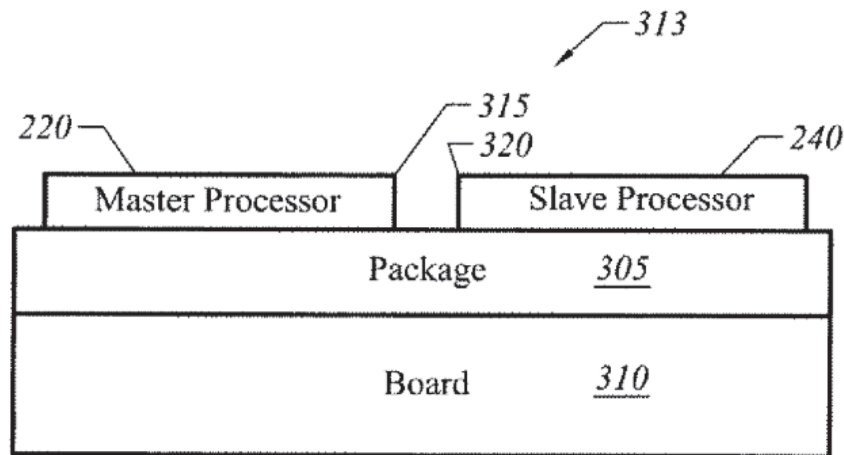


FIG. 3

EX1004-Kelleher, Figure 3.

92. **“Chiplet” Definition - Part 3:** “the programming model preserves the concept that these separate computational dies are a single monolithic unit (i.e., each chiplet is not exposed as a separate device to an application that uses the chiplets for processing computational workloads).” EX1001-’803, 2:41-53.

93. Kelleher’s master chip and slave chip both meet this requirement. For example, Kelleher states that “in the multi-chip module 213 only the master GPU 220 receives graphics instructions from CPU 205. As result, replacing a single GPU chip with a multi-chip module having a master/slave pair of GPUs is a seamless change to driver software within CPU 205. That is, the driver software [in the CPU] perceives the master/slave pair as a single GPU.” EX1004-Kelleher, 5:9-15.

94. Also, like the '803 patent, Kelleher's discloses using a crosslink—an “interlink” that is used for communication between the master and slave chips. For example, Figure 2 below shows interlink 250 connecting the two chips.

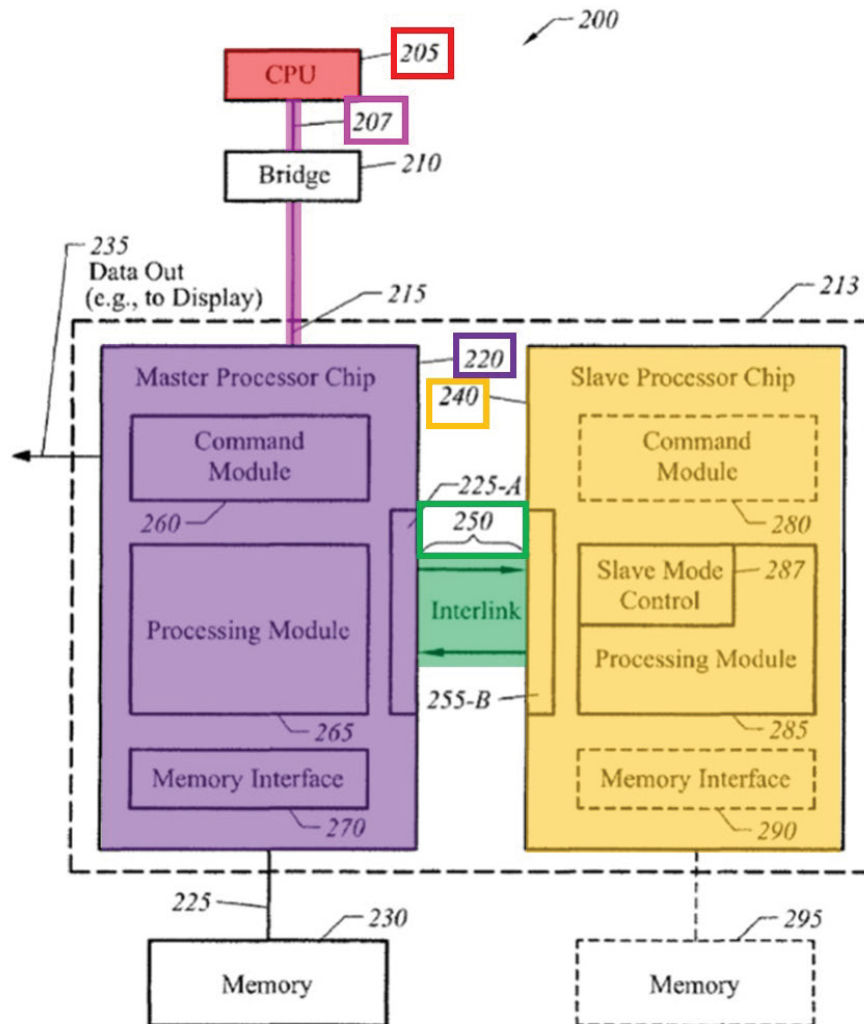


FIG. 2

EX1004-Kelleher, Figure 2.

95. Kelleher explains:

Slave GPU chip 240 is coupled to GPU chip 220 via an interlink 250. Interlink 250 is preferably a low latency, high bandwidth link. Interlink 250 should have a sufficiently low latency that the round trip delay time associated with sending and receiving data on interlink 250 does not exceed a timing margin for processing data. Additionally, the aggregate bandwidth of interlink 250 is preferably comparable to that of internal buses used to transmit graphics data of the type processed by graphics processing module 265. For example, the bandwidth of interlink 250 is preferably sufficiently large to accommodate sending and receiving data corresponding to the full input/output data width of processing module 285. These attributes of interlink 250 facilitate seamlessly offloading processing tasks to slave GPU chip 240 as if processing module 285 of slave GPU chip 240 was an internal extension of processing module 265 of GPU chip 220.”

EX1004-Kelleher 4:20-35.

96. This is similar to the '803 patent's teaching that “the processing system 500 utilizes a master-slave topology in which a single GPU chiplet in direct communication with the CPU 102 (i.e., GPU chiplet 106-1) is designated as the master chiplet.... The other GPU chiplets communicate with the CPU 102 indirectly via the passive crosslink 118 and are designated as the slave chiplets.”

EX1001-'803, 6:55-62.

97. In addition, further like the '803 patent, Kelleher discloses a bus that allows the CPU to communicate with only the master GPU chip. For example, Kelleher explains that “[a] central processing unit (CPU) 205 is coupled to a graphics processing unit (GPU) chip 220 through a suitable bus interface.” EX1004-Kelleher,

3:6-13. Kelleher also notes that “in the multi-chip module 213 only the master GPU 220 receives graphics instructions from CPU 205.” EX1004-Kelleher, 4:63-5:18. Then, using the interlink, the master GPU chip relays the information from the CPU to the slave GPU chip. For example, Kelleher explains:

Slave GPU chip 240 is coupled to GPU chip 220 via an interlink 250. Interlink 250 is preferably a low latency, high bandwidth link. Interlink 250 should have a sufficiently low latency that the round trip delay time associated with sending and receiving data on interlink 250 does not exceed a timing margin for processing data. Additionally, the aggregate bandwidth of interlink 250 is preferably comparable to that of internal buses used to transmit graphics data of the type processed by graphics processing module 265. For example, the bandwidth of interlink 250 is preferably sufficiently large to accommodate sending and receiving data corresponding to the full input/output data width of processing module 285. These attributes of interlink 250 facilitate seamlessly offloading processing tasks to slave GPU chip 240 as if processing module 285 of slave GPU chip 240 was an internal extension of processing module 265 of GPU chip 220.”

EX1004-Kelleher, 4:20-35.

98. With this configuration, the CPU considers the master and slave chips to be a “monolithic” GPU. “As result, replacing a single GPU chip with a multi-chip module having a master/slave pair of GPUs is a seamless change to driver software within CPU 205. That is, the driver software perceives the master/slave pair as a single GPU.” EX1004-Kelleher, 5:9-15. This is similar to the ’803 patent’s disclosure that “the CPU 102 is communicably coupled to a single GPU chiplet (i.e.,

GPU chiplet 106-1) through the bus 108. CPU-to-GPU transactions...is received at the GPU chiplet 106-1. Subsequently, any inter-chiplet communications is routed through the passive crosslink 118 as appropriate to access memory channels on other GPU chiplets 106. In this manner, the GPU chiplet-based system 100 includes GPU chiplets 106 that are addressable as a single, monolithic GPU from a software developer's perspective (e.g., the CPU 102 and any associated applications/drivers are unaware of the chiplet-based architecture), and therefore avoids requiring any chiplet-specific considerations on the part of a programmer or developer.” EX1001-'803, 3:42-56.

B. Collins

99. Collins, filed on September 28, 2018, discloses a “semiconductor package” that “include[s] an interconnect structure.” EX1005-Collins, Abstract.

100. Figure 1 of Collins (reproduced bellow) illustrates interconnects referred to as embedded multi-die interconnect bridges, or EMIBs (shown in green), which are described as “passive.” Specifically, Collins explains:

[T]he interconnect structure 108 includes one or more embedded multi-die interconnect bridges (EMIBs), where each of the EMIBs are embedded within the substrate assembly 102. The substrate assembly 102 includes conductive elements that couple the chiplets 104 and the memory device 106 to the EMIBs. Each of the EMIBs include **a piece of passive silicon** with conductive elements located within the passive silicon for conducting signals.

EX1005-Collins, 3:50-59.

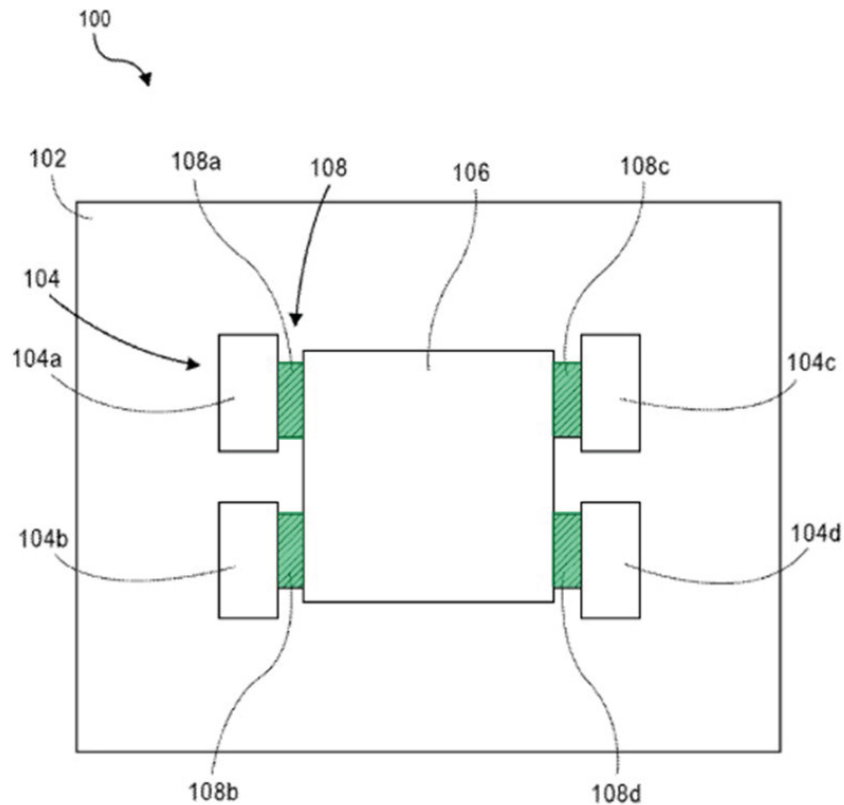


FIG. 1

EX1005-Collins, Figure 1.

101. Although Figure 1 depicts EMIBs connecting chiplets to “memory device 106,” Collins expressly discloses that “the EMIBs 108 may couple two or more of the chiplets 104 together in some embodiments. The chiplets 104 coupled with each other via the EMIBs 108 may utilize the EMIBs 108 for communication between the chiplets 104....” EX1005-Collins, 4:24–39. Collins further explains

that “the interconnect structure further couples a physical layer (PHY) of the first chiplet to a PHY of the second chiplet.” EX1005-Collins, 16:24–27.

102. Figure 2 of Collins, reproduced below, shows the structure of Collins’s interconnects used in an embodiment connecting a chiplet to a memory device.

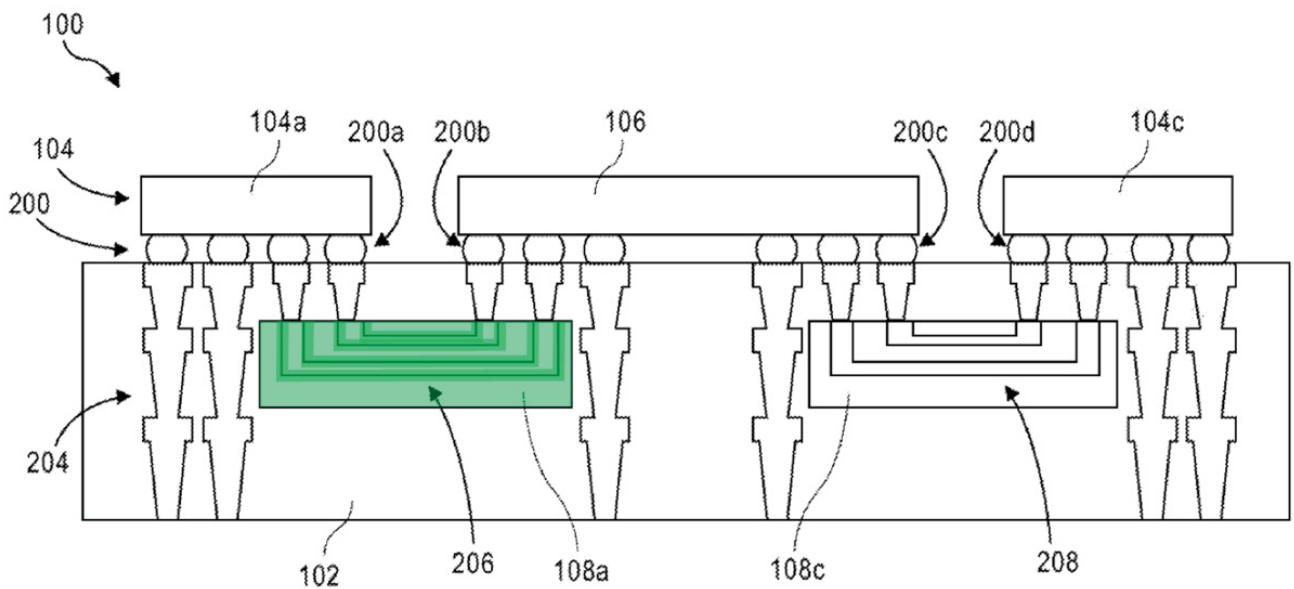


FIG. 2

EX1005-Collins, Figure 2.

103. However, when applied to an embodiment in which two chiplets are connected—illustrated in a modified version of Figure 2, reproduced below—the memory device is replaced with a second chiplet, as described by Collins which I explained above. EX1005-Collins, 4:24–39, 16:24–27.

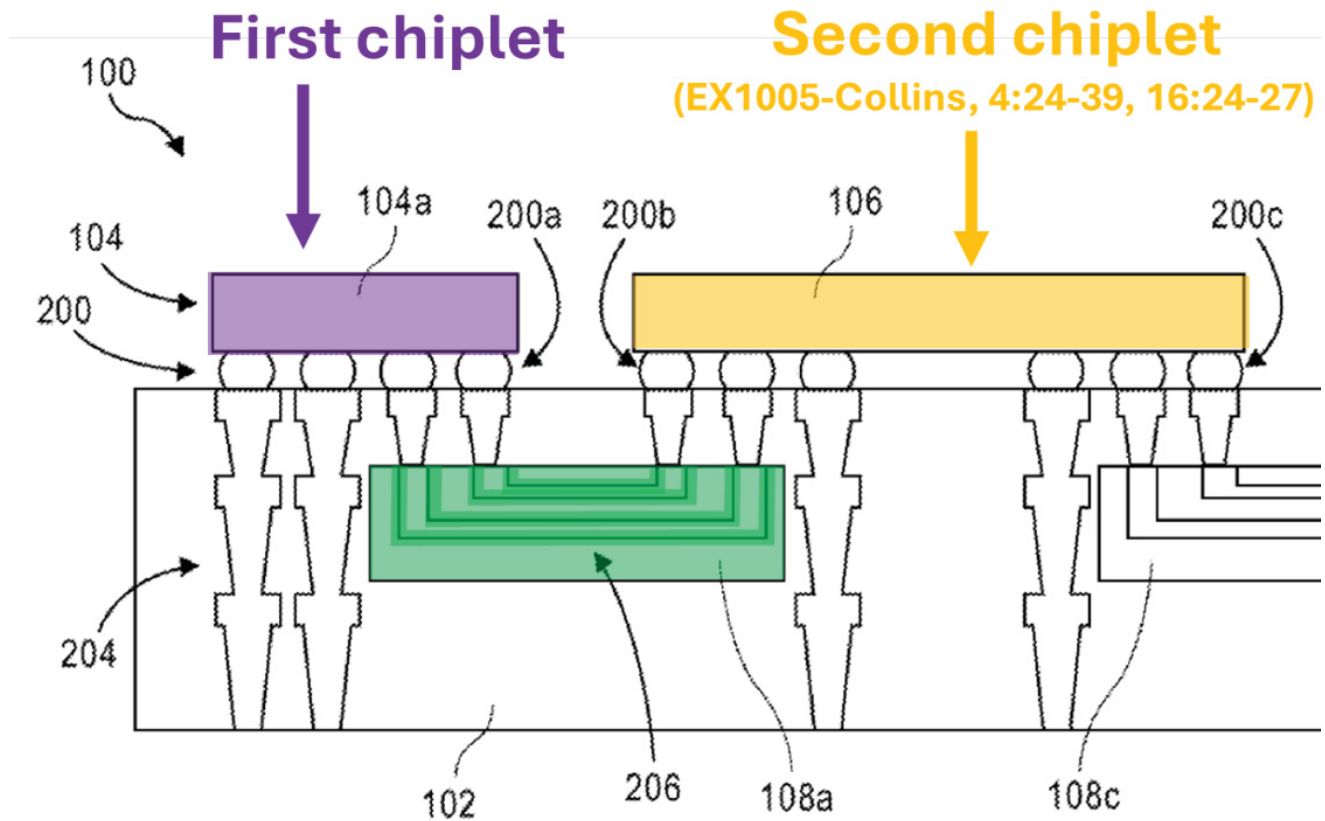


FIG. 2

EX1005-Collins, Modified Figure 2 (replacing “memory device 106” with a second chiplet according to Collins’s disclosure in 4:24-39 and 16:24-27).

104. In this configuration, the EMIB includes a passive interposer that lacks any signaling pathways that connect the chiplets to other components, and therefore the EMIB serves exclusively as a conduit for inter-chiplet communications.
EX1005-Collins, 4:24–39, 16:24–27.

C. Koker

105. Koker, filed on March 15, 2019, discloses “[a] disaggregated processor package” having “a standard physical interconnect for chiplets that can enable the chiplet to interface with a fabric or bridge interconnect.” EX1031-Koker, Abstract.

106. Figure 3C of Koker, reproduced below, describes a graphics processing unit (GPU 380) that “includes dedicated sets of graphics processing resources arranged into multi-core groups [MCGs] 365A-365N.” EX1031-Koker, 12:18-27, 13:36-42.

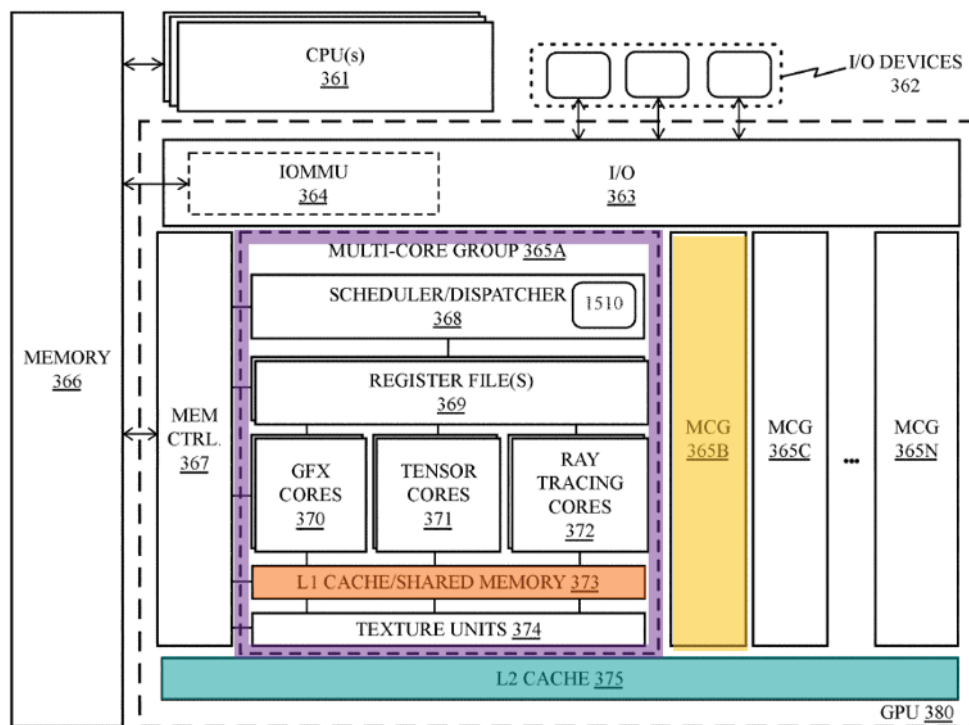


FIG. 3C

EX1031-Koker, Figure 3C.

107. Koker further explains that the GPU architecture incorporates a cache memory hierarchy consisting of multiple levels of cache. The “level 1 (L1) caches

and shared memory units 373 store graphics data such as texture data, vertex data, pixel data, ray data, bounding volume data, etc., locally within each multi-core group,” such as MCG 365A and MCG 365B. EX1031-Koker, 13:55–60. In contrast, the “[l]evel 2 (L2) cache 375 is shared by all or a subset of the multi-core groups 365A-365N” and is used to “store[] graphics data and/or instructions for multiple concurrent graphics threads.” EX1031-Koker, 13:60-66. As shown below in Figure 3C, the first multi-core group (MCG 365A) (shown in purple) contains a L1 cache (shown in orange). The second multi-core group (MCG 365B) (shown in gold) also contains a L1 cache (not pictured). Both of these groups share the L2 cache (shown in teal).

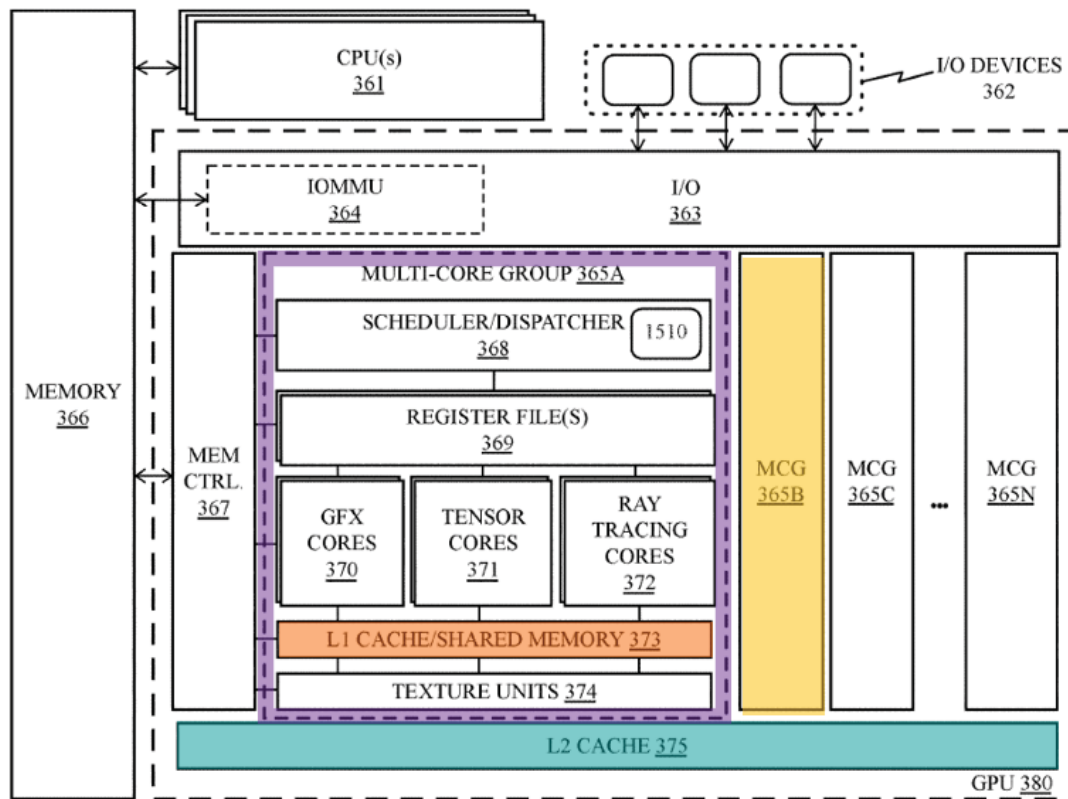


FIG. 3C

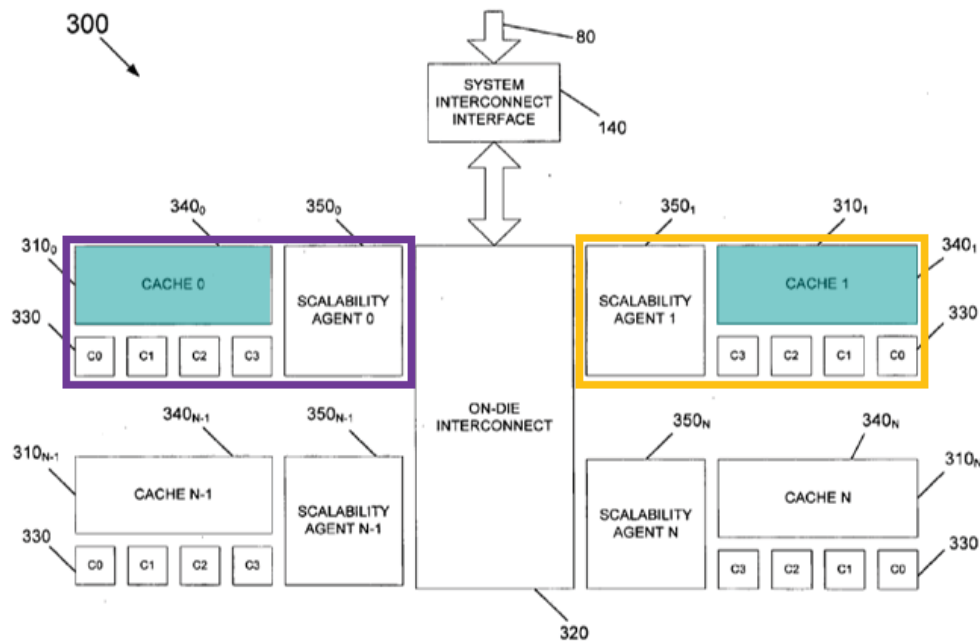
EX1031-Koker, Figure 3C.

D. Sistla

108. Sistla, filed on July 31, 2007, and published on February 5, 2009, discloses “a core-cache unit [that] is used as a building block[,] and multiple such units are joined to increase the number of cores in the processor. The core-cache unit can contain two or more cores with an appropriately sized shared cache.” EX1047-Sistla, [0008].

109. In the embodiment set forth in Figure 1 (shown below), a processor 300 includes a first “core-cache cluster” 310₀ (in purple) that has a first cache 340₀ (in

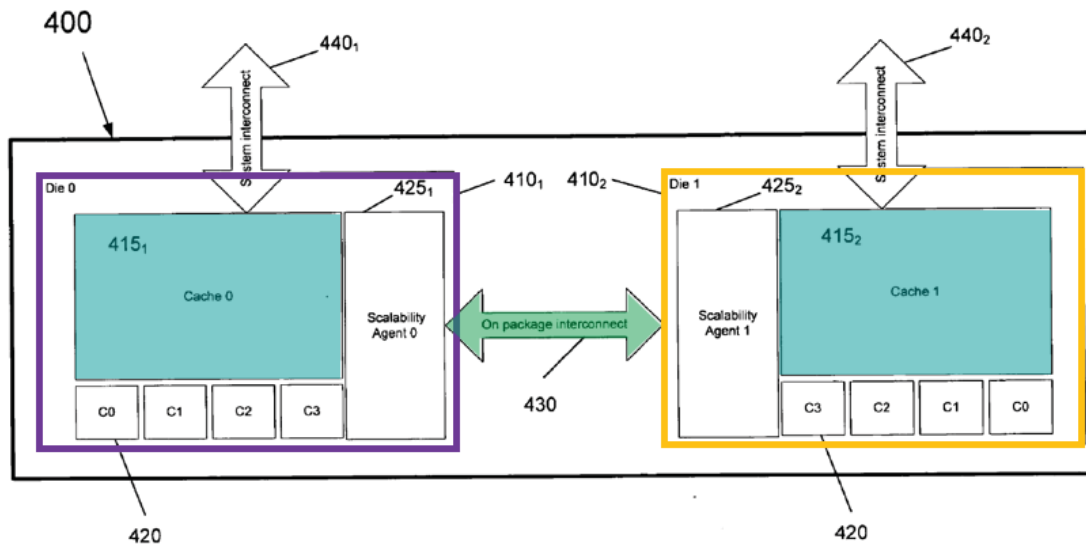
teal) and a second “core-cache cluster” 310₁ (in gold) that has a second cache 340₁ (in teal).



EX1047-Sistla, Figure 1, [0011]-[0013]. Sistla further explains that the processor 300 has a “distributed shared cache architecture” in which the first cache 340₀ (in teal) and the second cache 340₁ (in teal) “act together as a shared last level cache (LLC).” EX1047-Sistla, [0011]-[0012]. In this way, the two caches form a single last level cache that is shared by the processing cores (C0 to C3) of both the first core-cache cluster 310₀ (in purple) and the second core-cache cluster 310₁ (in gold). EX1047-Sistla, [0011]-[0012].

110. Sistla discloses another embodiment in Figure 2 (shown below) in which a first core-cache cluster of a processor 400 is contained within a first die 410₁

(in purple), and a second core-cache cluster of the processor 400 is contained within a second die 410₂ (in gold).



EX1047-Sistla, Figure 2, [0011]-[0013]. In addition, a first portion 415₁ of the last level cache (in teal) is included in the first die 410₁ (in purple), a second portion 415₂ of the last level cache (in teal) is included in the second die 410₂ (in gold), and the first die 410₁ (in purple) and the second die 410₂ are connected via a “point-to-point interconnect[.]” (in green). EX1047-Sistla, [0012]-[0013].

IX. LEVEL OF ORDINARY SKILL IN THE ART

111. It is my understanding that when considering the claims of the '803 patent and the prior art, I must do so based on the perspective of one of ordinary skill in the art at the relevant effective priority date.

112. As I detail in Section VII.B (Overview of the Prosecution History for the '803 Patent), the '803 patent was filed on June 28, 2019. EX1001-'803, Cover. Therefore, I understand the claimed priority date is June 28, 2019.

113. I have been informed and understand that several factors are to be considered in determining who would be a person of ordinary skill in the art. These factors include: (1) the types of problems encountered in the art; (2) the prior art solutions to those problems; (3) the rapidity with which innovations are made; (4) the sophistication of the technology; and (5) the educational level of active workers in the field.

114. Based on these factors as well as my experience and expertise, a person of ordinary skill in the art as of the claimed priority date would have had a bachelor's degree in computer science, computer engineering, electrical engineering, or equivalent field, and three to five years of experience with semiconductor layout design, or a master's degree or Ph.D. in the same fields, and one to two years of experience with semiconductor layout design. Less work experience may be compensated by a higher level of education, and vice versa.

115. Before June 28, 2019, which is the claimed priority date of the '803 patent, I would have qualified at least as a person of ordinary skill in the art under this standard.

116. Although my qualifications and experience exceed those of a person of ordinary skill in the art as defined above, my analysis and conclusions herein regarding the '803 patent have been based on the knowledge and perspective a person of ordinary skill in the art would have had as of the relevant time of the '803 patent.

X. CLAIM CONSTRUCTION

117. I was informed and I understand that, in my analysis, I should give claim terms their ordinary and customary meaning, as understood by a person of skill in the art at the time of the earliest filing date (the “effective filing date”) of the claim in the patent. I was informed and I understand that the words of the patent’s claims and the context in which the term is used in the claims are highly instructive in ascertaining the term’s meaning. I further understand that the terms of a claim are to be interpreted in the context of the entire patent, including the patent’s claims, its specification, and its figures.

118. I have been further informed and understand that a patent’s prosecution history may often provide helpful evidence about how the U.S. Patent and Trademark Office and the applicant understood the patent and its claims. Therefore, I have been informed and understand that claim terms should also be interpreted in light of the patent’s prosecution history.

119. In addition to the claims, the specification, the figures, and the prosecution history, I have been informed and understand that certain “extrinsic evidence,” such as scientific or technical references, may also help inform the way in which a person of ordinary skill in the art understands the claim term. I have been informed and understand that extrinsic evidence must always be considered in the context of the intrinsic evidence.

120. I have been informed and understand that a patent specification may provide a special definition for a claim term. When that is the case, the inventor’s own “lexicography” controls the claim term’s meaning.

121. I have also been informed and understand that a patent specification may contain a disavowal of claim scope by the inventor. I understand that such a disavowal must be clear and unmistakable. When that is the case, the inventor’s disavowal controls, even if the plain meaning of the claim term might otherwise include the disavowed subject matter.

122. I understand that claim constructions have been proposed in another proceeding in the International Trade Commission (“ITC”) involving the ’803 patent— *Certain Integrated Circuits, Electronic Devices Containing the Same, and Components Thereof*, Inv. No. 337-TA-1450 (“ITC litigation”). I understand that in the ITC litigation, NVIDIA, Onesta IP, and staff at the Office of Unfair Import

Investigations (“the Staff”) disagree on the construction of certain terms. EX1043-Claim-Construction-Chart.

123. For the purposes of this Declaration, I will give the claim terms their plain and ordinary meaning, except for the terms noted below.

A. “Chiplet”

124. I understand that in the ITC litigation, NVIDIA, Onesta IP, and the ITC Staff Attorney (“Staff”) agree that “chiplet” should be interpreted as:

a device that includes the following characteristics: 1) a chiplet includes an active silicon die containing part of the computational logic used to solve a full problem (i.e., the computational workload is distributed across multiples of these active silicon dies); 2) chiplets are packaged together as a monolithic unit on the same substrate; and 3) the programming model preserves the concept that these separate computational dies are a single monolithic unit (i.e., each chiplet is not exposed as a separate device to an application that uses the chiplets for processing computational workloads)

EX1043-Claim-Construction-Chart at 2. This construction is the same as the definition of “chiplet” set forth in the ’803 patent:

In various embodiments, and as used herein, the term “chiplet” refers to any device including, but is not limited to, the following characteristics: 1) a chiplet includes an active silicon die containing part of the computational logic used to solve a full problem (i.e., the computational workload is distributed across multiples of these active silicon dies); 2) chiplets are packaged together as a monolithic unit on the same substrate; and 3) the programming model preserves the concept that these

separate computational dies are a single monolithic unit (i.e., each chiplet is not exposed as a separate device to an application that uses the chiplets for processing computational workloads).

EX1001-'803, 2:41-53.

125. I have used this agreed-upon construction of “chiplet” in my analysis in this Declaration.

B. “Dedicated for inter-chiplet communications”

126. Both NVIDIA and the ITC Staff propose construing the term “dedicated for inter-chiplet communication” as “used for transmission of only chiplet-to-chiplet signals.” EX1043-Claim-Construction-Chart, 4. I agree with this proposed construction.

127. The meaning of “dedicated,” as defined by the Dictionary of Computer Science, is “[c]ommitted entirely to a single purpose or device.” EX1034-Dictionary-of-Computer-Science, at 149. Additionally, the Dictionary of Multimedia defines “dedicated” as “[d]escrib[ing] any computing system that performs one function exclusively.” EX1035-Dictionary-of-Multimedia, at 118. Therefore, these dictionaries support construing “dedicated for inter-chiplet communication” as “used for transmission of only chiplet-to-chiplet signals.”

128. The '803 patent also describes “PHY regions” that are dedicated to communications between chiplets—i.e., communication of only “chiplet-to-chiplet

signals”—and “non-PHY regions” that are used to transmit other signals, such as power signals, and to communicate signals between chiplets and a circuit board. For example, with reference to Figure 2 (reproduced below), the ’803 patent states:

In various embodiments, each GPU chiplet 106 is constructed with *a physical device (PHY) region 202, which has various internal and external conductor structures dedicated to the transmission of chiplet-to-chiplet signals, and a non-PHY region 204, which has conductor structures that are tailored more to the conveyance of power and ground and/or chiplet-to-circuit board signals.*

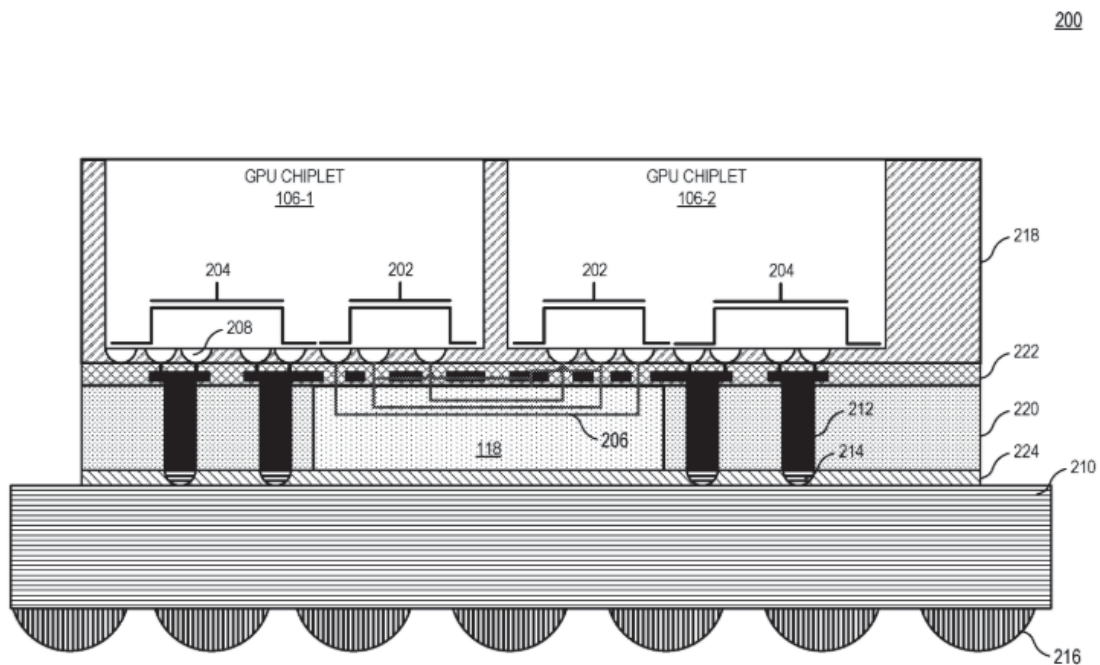


FIG. 2

EX1001-’803, Figure 2, 3:63-4:2 (emphasis added).

129. The patent further explains that conductive traces 205 of a passive crosslink 118, which are used only for chiplet-to-chiplet communications, electrically interface with the PHY regions:

The passive crosslink 118 includes a plurality of internal conductor traces, which may be on a single level or multiple levels as desired. Three of the traces are **illustrated in FIG. 2 and labeled collectively as traces 206. The traces 206 interface electrically with conductor structures of the PHY regions 202 of the GPU chiplets 106 by way of conducting pathways.**

EX1001-'803, 4:8-16.

130. On the other hand, the patent explains that conductive pillars, which are used for communicating chiplet-to-circuit board or chiplet-to-substrate signals—and not for chiplet-to-chiplet communications—electrically interface with the non-PHY regions and are routed around the passive crosslink that is dedicated to transmitting only chiplet-to-chiplet signals:

The non-PHY regions 204 of the GPU chiplets 106 interface electrically with a circuit board 210 (or any other substrate) by way of a plurality of conductive pillars 212. Each conductive pillar 212 is connected electrically to the GPU chiplets 106 by way of a solder interconnect 208, which may include a solder bump, micro bump, and the like. In various embodiments, the circuit board 210 interfaces electrically with other electrical structure, such as another circuit board or other structures, by way of a plurality of interconnect structures 214 (e.g., solder balls and the like).

EX1001-'803, 4:22-31.

The conductive pillars 212 connect signals between the GPU chiplets 106 and the substrate in areas where the HDCL die is not present (e.g., areas with vertical discrepancy between the GPU chiplets 106 and the circuit board 210), with empty spaces filled with epoxy or other gap fill materials. **In this manner, power and input/output (I/O) lines of the non-PHY regions 204 are routed around the interposer die (i.e., passive crosslink 118) using redistribution (RDL) technology, thereby replacing usage of conventional TSVs.**

EX1001-'803, 4:36-44.

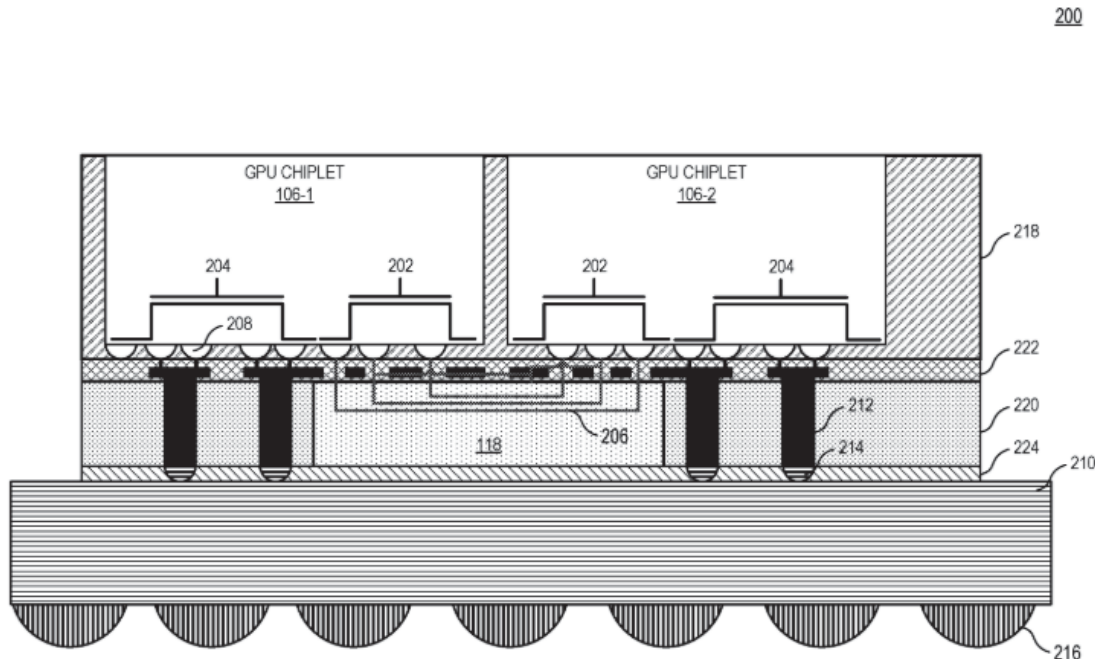


FIG. 2

EX1001-'803, Figure 2,

131. The '803 patent's description of separate PHY and non-PHY regions for transmitting signals confirms that “dedicated for inter-chiplet communication” means “used for transmission of only chiplet-to-chiplet signals.”

132. Additionally, the '803 patent distinguishes the “passive crosslink” from the regions containing through silicon vias (TSVs) by disclosing that the “passive crosslink” is free from vias or signals that are not used for “inter-chiplet communications.” For example, the '803 patent explains:

As previously noted, the GPU chiplets 106 are communicably coupled by way of the passive crosslink 118. ... The passive crosslink 118 includes a plurality of internal conductor traces, which may be on a single level or multiple levels as desired. Three of the traces are illustrated in FIG. 2 and labeled collectively as traces 206. The traces 206 interface electrically with conductor structures of the PHY regions 202 of the GPU chiplets 106 by way of conducting pathways. ***It is noted that the passive crosslink 118 does not contain any through silicon vias (TSVs).*** In this manner, the passive crosslink 118 is a passive interposer die that communicably couples and routes communications between the GPU chiplets 106, thereby forming a passive routing network.

The non-PHY regions 204 of the GPU chiplets 106 interface electrically with a circuit board 210 (or any other substrate) by way of a plurality of conductive pillars 212. Each conductive pillar 212 is connected electrically to the GPU chiplets 106 by way of a solder interconnect 208, which may include a solder bump, micro bump, and the like. In various embodiments, the circuit board 210 interfaces electrically with other electrical structure, such as another circuit board or other structures, by way of a plurality of interconnect structures 214 (e.g., solder balls and the like). ...

The conductive pillars 212 connect signals between the GPU chiplets 106 and the substrate in areas where the HDCL die is not present (e.g., areas with vertical discrepancy between the GPU chiplets 106 and the circuit board 210), with empty spaces filled with epoxy or other

gap fill materials. *In this manner, power and input/output (I/O) lines of the non-PHY regions 204 are routed around the interposer die (i.e., passive crosslink 118) using redistribution (RDL) technology, thereby replacing usage of conventional TSVs.*

EX1001-'803, 4:4-44.

133. Moreover, Figure 2 of the '803 patent, reproduced below, does not show any TSVs, such as the conductive pillars 212, contained within the passive crosslink 118.

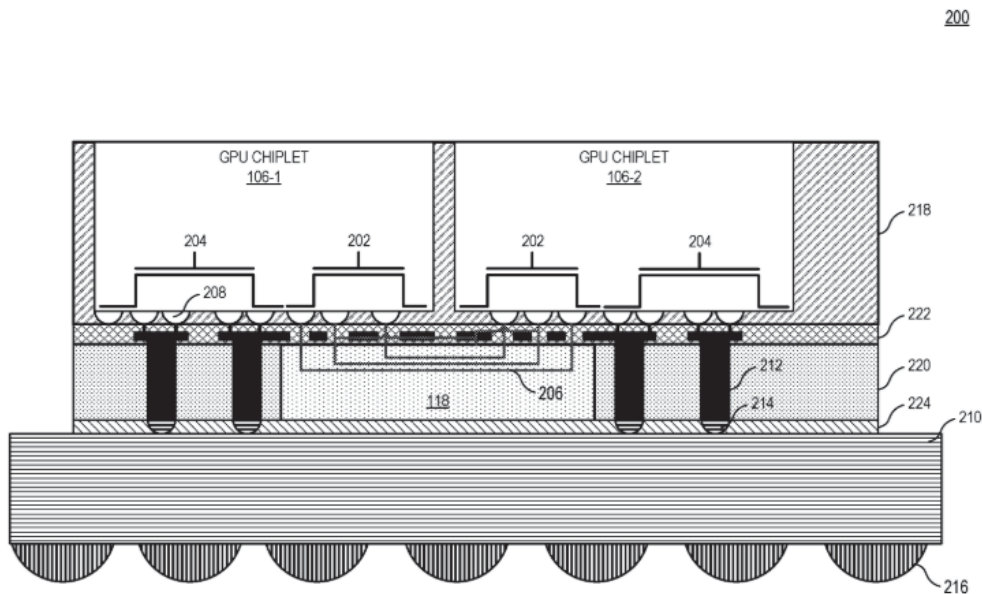


FIG. 2

EX1001-'803, Figure 2.

134. Lastly, as I explained in Section V.II.B [Overview of the Prosecution History for the '803 Patent] above, the Applicant argued that Teig's interposer was not "dedicated" for inter-chiplet communications because Teig's interposer connected the CPU and GPU cores on one die, with the cache memory for those

cores located on a die on the opposite side of the interposer. *See generally* EX1027-Teig, [0148]; EX1003-FH, 847.

135. I understand that in the co-pending ITC litigation, Onesta IP proposes construing “dedicated for inter-chiplet communication” as “configured to provide communication only between GPU chiplets.” EX1043-Claim-Construction-Chart at 4. It is my opinion that Onesta IP’s proposed construction improperly allows the passive crosslink to convey signals other than inter-chiplet communications. For example, Onesta IP’s construction would allow ground and power signals between a chiplet and other components to be conveyed via the passive crosslink. But, as I have discussed above, this construction is inconsistent with the plain meaning of the claim language, the specification of ’803 patent, and its file history. For example, as noted above, the ’803 patent explains that power and ground signals are “routed around” the passive crosslink. EX1001-’803, 4:4-44. As noted below, the prior art invalidates the challenged claims under NVIDIA’s proposed construction, Onesta IP’s proposed construction, or any reasonable construction.

C. “A passive crosslink, wherein the passive crosslink is dedicated for inter-chiplet communication”

136. I understand that the Staff and Onesta IP have further proposed construing the term “a passive crosslink, wherein the passive crosslink is dedicated for inter-chiplet communication.” EX1043-Claim-Construction-Chart at 4. I

understand that the Staff has proposed construing this term as “a non-active semiconductor die, which die does not include any through silicon vias and which die is used for transmission of only chiplet-to-chiplet signals.” EX1043-Claim-Construction-Chart at 4. I understand that Onesta IP has proposed construing “passive crosslink” as “a non-active connecting structure.” EX1043-Claim-Construction-Chart at 4. Onesta IP also proposes construing the term “dedicated for inter-chiplet communication” as “configured to provide communication only between GPU chiplets,” as noted above. EX1043-Claim-Construction-Chart at 4. It is my opinion that this term does not need to be construed for the purposes of my analysis. In any event, claims 1-4 and 7-10 are rendered obvious under all proposed constructions set forth above or under any reasonable construction.

XI. SPECIFIC GROUNDS FOR DEMONSTRATING THAT CLAIMS 1-4 AND 7-10 ARE INVALID

A. GROUND I: Claims 1-4, 7, and 10 Would Have Been Obvious In View of the Combined Teachings of Kelleher and Collins

1. Independent Claim 1

(a) Element 1[pre]: A system, comprising:

137. Kelleher teaches Element 1[pre].

138. Kelleher teaches a “[a] multi-chip graphics system [which] includes a master chip and a slave chip coupled by an interlink.” EX1004-Kelleher, Abstract. Kelleher’s system can also be seen in Figure 2, reproduced below. Kelleher explains

that “FIG. 2 is a block diagram of a multi-chip graphics system 200 in accordance with one embodiment of the present invention. A central processing unit (CPU) 205 is coupled to a graphics processing unit (GPU) chip 220 through a suitable bus interface. As used in this application a “chip” is an integrated circuit that resides on a die, with the die packaged to permit electrical power and signal connections to be made to the integrated circuit.” EX1004-Kelleher, 3:6-13, 3:52.

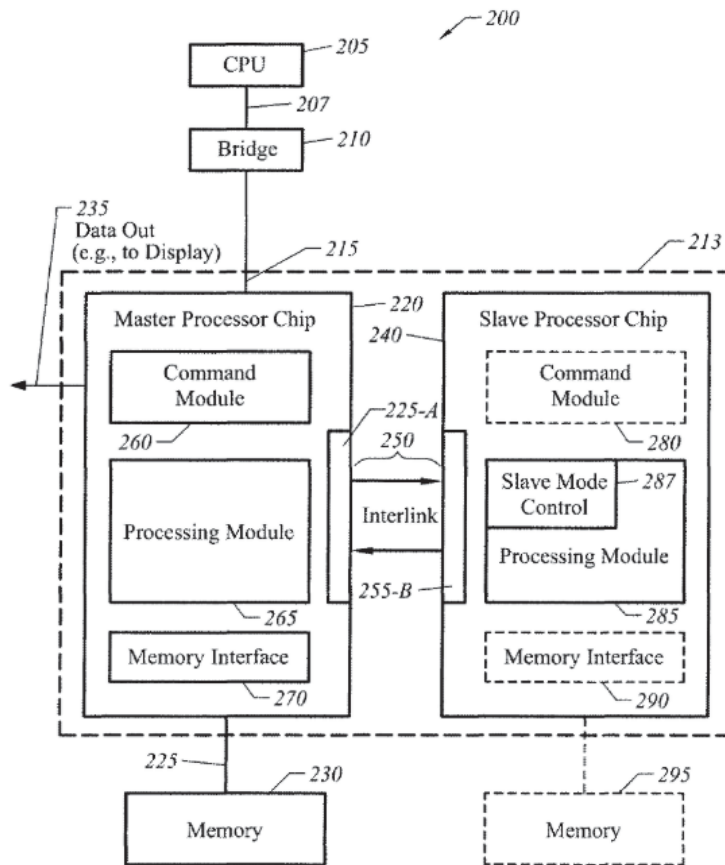


FIG. 2

EX1004-Kelleher, Figure 2

139. I provide more details regarding Kelleher's system below with respect to Elements 1[a]-1[d].

**(b)Element 1[a]: a central processing unit (CPU)
communicably coupled to a first graphics processing unit
(GPU) chiplet of a GPU chiplet array,**

140. Kelleher teaches Element 1[a].

141. Kelleher teaches *a central processing unit (CPU)* (CPU 205, shown in red in Figure 2 below) *communicably coupled to a first graphics processing unit (GPU) chiplet* (master GPU chip 220, shown in purple in Figure 2 below) *of a GPU chiplet array* (multi-chip module 213, shown in blue in Figure 2 below).

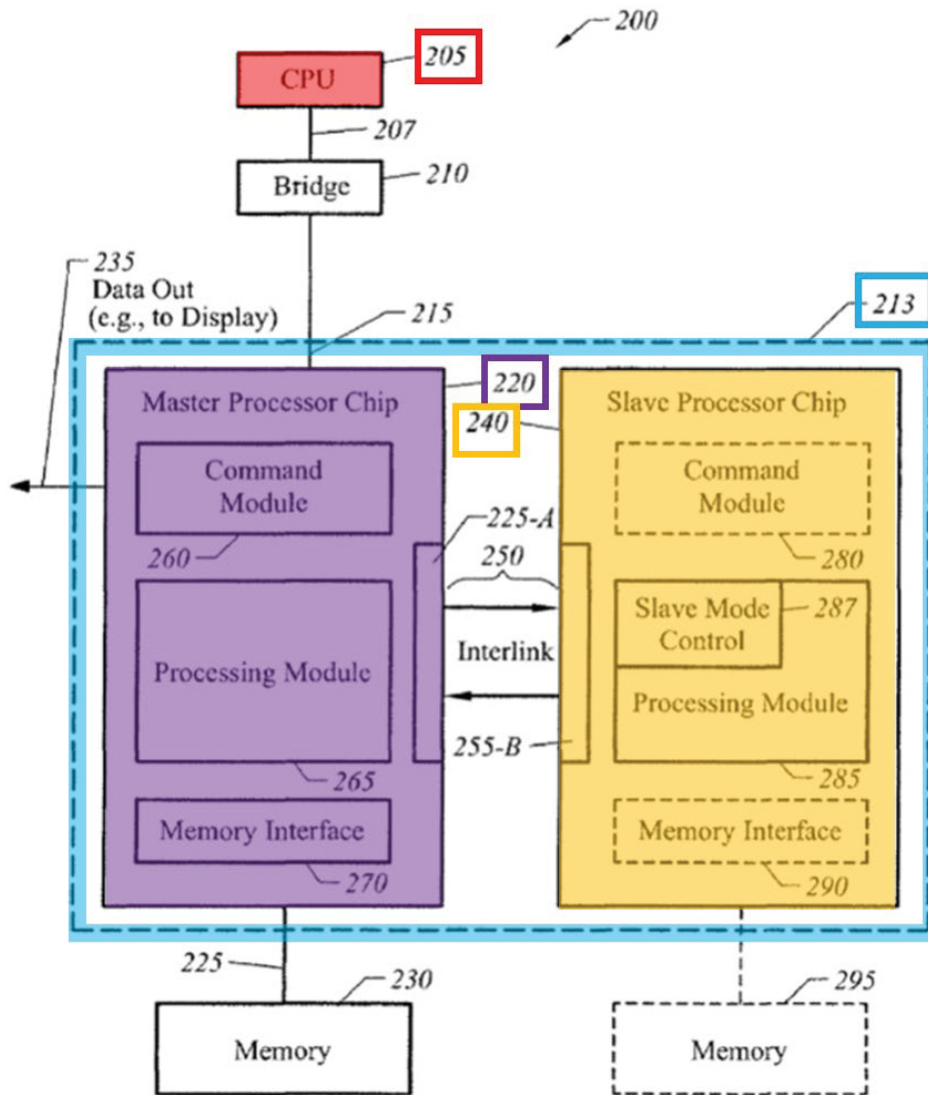


FIG. 2

EX1004-Kelleher, Figure 2.

142. First, Kelleher teaches a central processing unit (CPU) (CPU 205, shown in red) that is “communicably coupled” to a first graphics processing unit (GPU) chiplet (master GPU chip 220, shown in purple).

143. Kelleher describes a “central processing unit (CPU 205)” that is “coupled to a graphics processing unit (GPU) chip 220 through a suitable bus interface” EX1004-Kelleher, 3:6-13. The GPU chip 220 is configured to “receive[] a graphics command stream from CPU 205.” EX1004-Kelleher, 3:29-34; 5:9-11, 7:1-8, 8:35-47.

144. To facilitate this connection between the CPU 205 and the master GPU chip 220, Kelleher explains that “a bridge 210, such as a bridge chipset, is used to couple CPU 205 to GPU chip 220.” EX1004-Kelleher, 3:14-20. Kelleher further explains that “a front side bus 207 may be used to communicatively couple CPU 205 to bridge 210 whereas a performance component interface (PCI)-Express bus 215 may be used to communicatively couple GPU chip 220 to bridge 210.” EX1004-Kelleher, 3:14-20. A “PCI-Express is “a high speed interface standard promoted by the PCI-SIG organization and defined by the PCI-Express specification published by PCI-SIG.” EX1004-Kelleher, 3:20-23.

145. I further discuss the connection between the CPU and the master GPU chip 220 with respect to Element 1[b], which I incorporate herein by reference.

146. Second, Kelleher teaches *a first graphics processing unit (GPU) chiplet* (master GPU chip 220, shown in purple Figure 2 below) *of a GPU chiplet array* (multi-chip module 213, shown in blue in Figure 2 below).

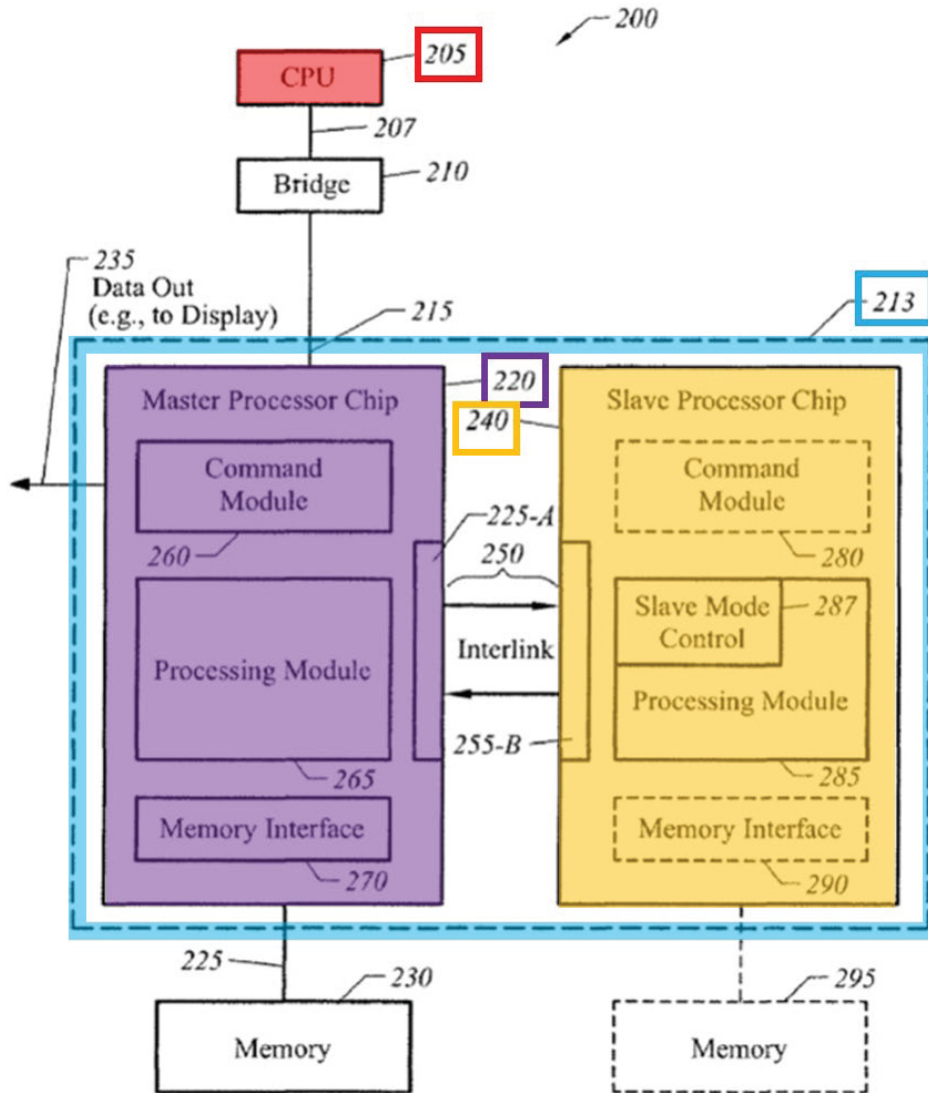


FIG. 2

EX1004-Kelleher, Figure 2.

147. Figure 2 of Kelleher shows that the *first GPU chiplet* (GPU chip 220) is part of a *GPU chiplet array* (multi-chip module 213). The multi-chip module 213 includes both a master GPU chip 220 and a slave GPU chip 240. Kelleher explains that “[s]lave GPU chip 240 is packaged with GPU chip 220 as a multi-chip module

213 that may be used as a replacement for an individual GPU chip.” EX1004-Kelleher, 3:45-50. Kelleher also notes that “multi-chip module 213 may include a package adapted to be mounted onto a board.” EX1004-Kelleher, 3:45-50.

148. Referring to Figure 3 below, Kelleher also describes an embodiment in which “GPU chip 240 and GPU chip 220 are packaged in an edge-to-edge multi-chip module 313 with the interlink formed by fabricating signal leads between the chips.” EX1004-Kelleher, 5:39-65.

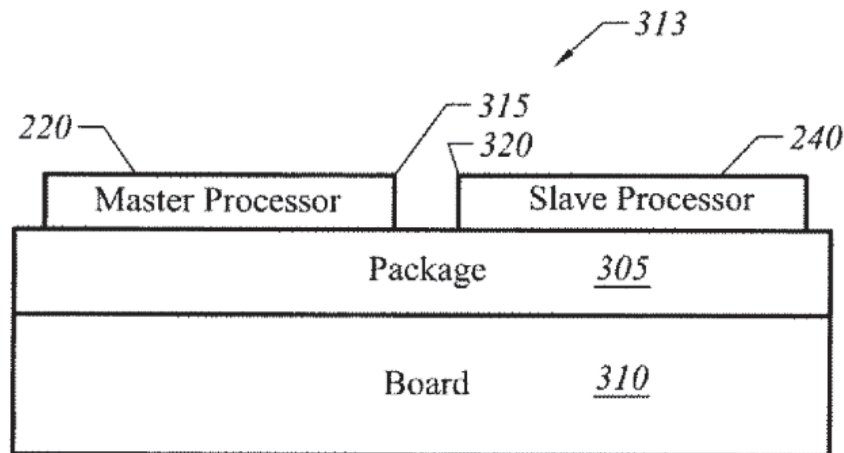


FIG. 3

EX1004-Kelleher, Figure 3.

149. As I explained in Section X (Claim Construction), a “chiplet” has three requirements according to the specification of the ’803 patent. EX1001-’803, 2:41-53. Kelleher’s master GPU chip 220 and slave GPU chip 240 meet each of these requirements for the below reasons, in addition to the reasons I explained in Section

VIII.A (Overview of the Prior Art—Kelleher), which I incorporate herein by reference.

150. The three requirements of a “chiplet” are reproduced below from the specification of the ’803 patent. I address each of these in turn.

In various embodiments, and as used herein, the term “chiplet” refers to any device including, but is not limited to, the following characteristics: 1) a chiplet includes an active silicon die containing part of the computational logic used to solve a full problem (i.e., the computational workload is distributed across multiples of these active silicon dies); 2) chiplets are packaged together as a monolithic unit on the same substrate; and 3) the programming model preserves the concept that these separate computational dies are a single monolithic unit (i.e., each chiplet is not exposed as a separate device to an application that uses the chiplets for processing computational workloads).

EX1001-’803, 2:41-53.

151. **“Chiplet” Definition - Part 1:** “a chiplet includes an active silicon die containing part of the computational logic used to solve a full problem (i.e., the computational workload is distributed across multiples of these active silicon dies).”

EX1001-’803, 2:41-53.

152. Regarding the first requirement, both Kelleher’s GPU chip 220 and GPU chip 240 include an active silicon die that contains a “processing module.” Kelleher explains that “a ‘chip’ is an integrated circuit that resides on a die, with the die packaged to permit electrical power and signal connections to be made to the

integrated circuit.” EX1004-Kelleher, 3:10-13. Within the GPU chip 220, “one of the stages in the graphics pipeline of GPU chip 220 is a processing module 265 that performs a processing operation.” EX1004-Kelleher, 3:38-41. Similarly, the “[s]lave GPU chip 240 includes a processing module 285 capable of performing a processing function.” EX1004-Kelleher, 3:51-53. Kelleher further notes that the “GPU chip 220 acts as the master processor and utilizes processing module 285 as an additional processing resource that is used in parallel with processing module 265.” EX1004-Kelleher, 4:3-19.

153. A person of ordinary skill in the art would have understood Kelleher’s reference to “dies” to mean silicon dies, which were standard in the semiconductor industry long before 2019. For example, Koker explains that “[b]uilding larger and larger silicon dies is challenging for a variety of reasons,” and describes embodiments that “provide techniques to disaggregate an architecture of a system on a chip integrated circuit into multiple distinct chiplets,” where “a graphics processing unit or parallel processor is composed from diverse **silicon chiplets** that are separately manufactured.” EX1031-Koker, 61:15–50.

154. This understanding is further supported by additional references in the field. For example, a 1997 article explaining Moore’s Law mentions that “[a] simple observation, made over 30 years ago, on the growth in the number of devices per **silicon die** has become the central driving force of one of the most dynamic of the

world's industries.” EX1029-Schaller, 53-54, 57. Additionally, a 2015 article explaining 3D and 2.5D stacking notes that “[d]ie-stacking technology enables the combination of multiple distinct silicon chips within a single package. In the past several years, the industrial adoption of die-stacking technologies has been accelerating. Vertical or 3D stacking takes **multiple silicon die** and places one on top of the other with through-silicon vias (TSVs) providing inter-layer connectivity. **Another die-stacking approach takes multiple silicon die**, and ‘stacks’ them side-by-side on a silicon interposer carrier...[which is] [k]nown as ‘2.5D stacking.’” EX1008-Jerger, *1-*2. Another 2015 article explains how in “multi-chip modules (MCMs), “chips are placed horizontally on the package substrate (generally a ceramic flatpack) and connected to the substrate using C4 bumps. Wires between chips are routed using metal layers on the substrate. This allows the designer to pack the chips together much closer than with a multi-socket system, thereby reducing the interconnect delay significantly. ... While this alleviates the limitations of pin connections, the bandwidth and latency are restricted by the density of C4 bumps and the substrate routing that connects **the silicon die**.” EX1011-Kannan, ii, 1-2, 6-13, 18, 44. *See also* EX1030-Deng & Maly, 450-452.

155. Kelleher also references Ziptronix, Inc., which “provides three dimensional wafer bonding and interconnect services that permit a non-adhesive bond to be formed between a die and a corresponding interconnect on another chip.”

EX1004-Kelleher, 6:41-46. An article by Ziptronix further describes the use of “silicon oxide coated silicon wafers” in such bonding processes, which a person of ordinary skill in the art would understand to be a silicon die. EX1028-Ziptronix, 308-315.

156. In addition, both Kelleher’s GPU chip 220 and GPU chip 240 contain part of the computational logic necessary to solve a complete problem because the computational workload is distributed between the processing module 265 of the GPU chip 220 and the processing module 285 of the GPU chip 240. Kelleher explains that the “GPU chip 220 acts as the master processor and utilizes processing module 285 [of the GPU chip 240] as an additional processing resource that is used in parallel with processing module 265 [of the GPU chip 220].” EX1004-Kelleher, 4:3-19. Kelleher also notes that “a load balancing module (not shown) within processing module 265 determines a distribution of work to be performed by processing modules 265 and 285 and transfers a portion of the graphics data that it receives to processing module 285. Thus for example, if the processing modules 265 and 285 are substantially identical, about half of the processing that would ordinarily be performed by processing module 265 [of the GPU chip 220] is off-loaded to processing module 285 [of the GPU chip 240].” EX1004-Kelleher, 4:3-19.

157. **“Chiplet” Definition - Part 2:** “chiplets are packaged together as a monolithic unit on the same substrate.” EX1001-’803, 2:41-53.

158. Kelleher's GPU chip 220 and GPU chip 240 are integrated as a unified, monolithic unit on the same substrate. For example, Figure 3 of Kelleher, which is set forth below, shows that the GPU chip 220 (in purple) and the GPU chip 240 (in gold) integrated as a single, monolithic unit on the package 305 and circuit board 310. Kelleher explains that the "[s]lave GPU chip 240 is packaged with GPU chip 220 as a multi-chip module 213 that may be used as a replacement for an individual GPU chip." EX1004-Kelleher, Figure 3, 3:46-50. Kelleher also notes that "multi-chip module 213 may include a package [i.e. package 305] adapted to be mounted onto a board [i.e. circuit board 310]." EX1004-Kelleher, 3:46-50. Kelleher further states that the "GPU chip 240 and GPU chip 220 are packaged on top of a common multi-chip package 305." EX1004-Kelleher, 5:39-44.

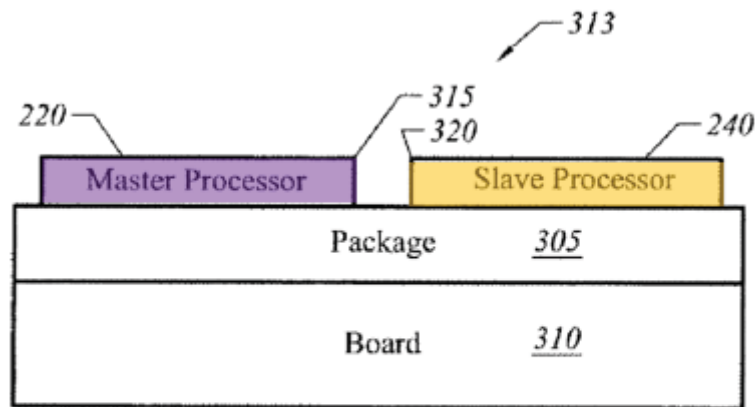


FIG. 3

EX1004-Kelleher, Figure 3.

159. **“Chiplet” Definition - Part 3:** “the programming model preserves the concept that these separate computational dies are a single monolithic unit (i.e., each chiplet is not exposed as a separate device to an application that uses the chiplets for processing computational workloads).” EX1001-’803, 2:41-53.

160. Kelleher discloses that applications running on CPU 205 treat GPU chip 220 and GPU chip 240 as a unified, monolithic unit rather than as separate devices. In particular, Kelleher explains that “in the multi-chip module 213 [with the GPU chips 220 and 240] only the master GPU 220 receives graphics instructions from CPU 205. As result, replacing a single GPU chip with a multi-chip module having a master/slave pair of GPUs is a seamless change to driver software within CPU 205. That is, the driver software perceives the master/slave pair as a single GPU.” EX1004-Kelleher, 4:63-5:18. Kelleher also notes various programming techniques that were well-known to a person of ordinary skill in the art.

A variety of techniques can be used to establish a master-slave relationship between GPU chip 220 and slave GPU chip 240 in a packaged multi-chip module 213. For example, one or both of the chips may be programmed prior to packaging (e.g., by programming a bit in a permanent memory). Alternatively, one or both of the chips may determine that they are the master or the slave by detecting whether they are receiving command instructions from a command module and determining whether interlink 250 is active. Additionally, a messaging technique may be used. For example, GPU chip 220 and slave GPU chip 240 may exchange messages to set up a master-slave relationship. Hardware or software within

each processing module may be included to support the establishment of a master-slave relationship. For example, a slave mode control module 287 may be included in slave GPU chip 240 to establish a slave condition if one or more slave conditions are present, such as detecting a load on interlink 250. Similarly a master mode control module (not shown) within in GPU chip 220 may be included to establish a master condition if one or more master conditions are present.

EX1004-Kelleher, 5:19-38.

**(c) Element 1[b]: wherein the GPU chiplet array includes:
the first GPU chiplet communicably coupled to the CPU
via a bus; and**

161. Kelleher teaches Element 1[b].

162. Kelleher teaches that *the GPU chiplet array* (multi-chip module 213) *includes: the first GPU chiplet* (GPU chip 220 shown in purple in Figure 2, below) *communicably coupled to the CPU* (CPU 205 shown in red in Figure 2, below) *via a bus* (PCI-Express bus 215 and/or front side bus 207 shown in pink in Figure 2, below).

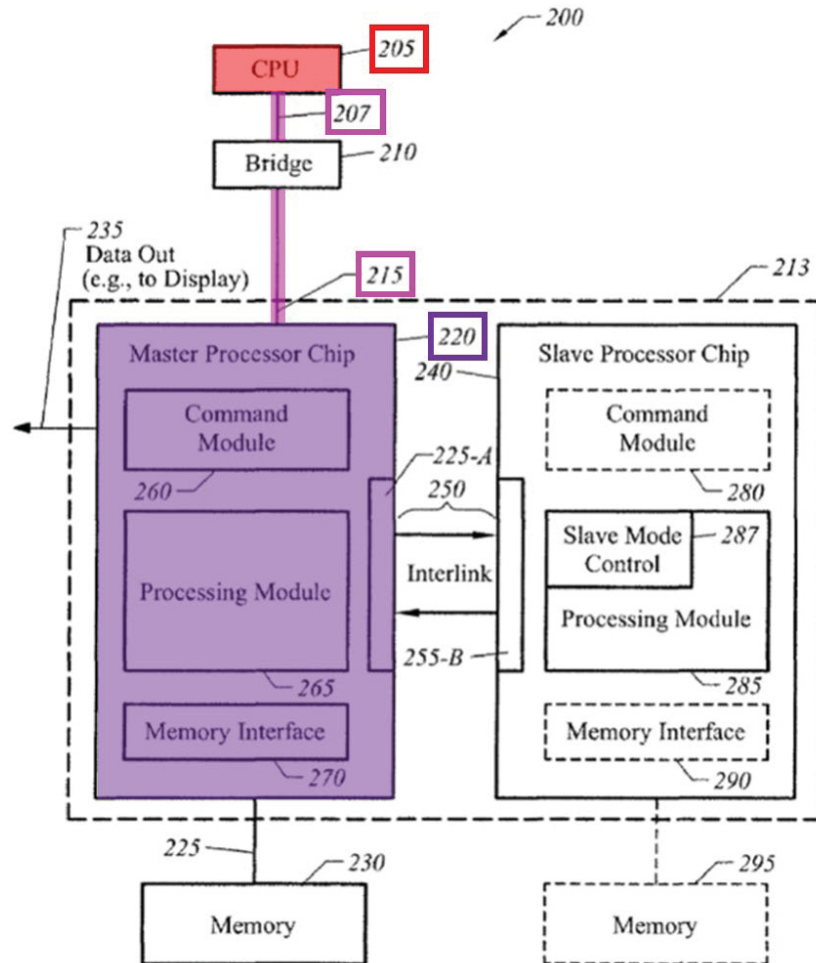


FIG. 2

EX1004-Kelleher, Figure 2.

163. Kelleher describes that CPU 205 is coupled to GPU chip 220 through a series of connections, including a front side bus 207, a bridge 210, and a PCI-Express bus 215. EX1004-Kelleher, 3:14-24. Kelleher explains that “a bridge 210, such as a bridge chipset, is used to couple CPU 205 to GPU chip 220. For example, a front side bus 207 may be used to communicatively couple CPU 205 to bridge 210

whereas a performance component interface (PCI)-Express bus 215 may be used to communicatively couple GPU chip 220 to bridge 210.” EX1004-Kelleher, 3:14-24. Kelleher also notes that “[a] central processing unit (CPU) 205 is coupled to a graphics processing unit (GPU) chip 220 through a suitable bus interface.” EX1004-Kelleher, 3:6-13.

164. The ’803 patent similarly discloses that, in some embodiments, the first GPU chiplet and CPU are connected using a bridge and buses. Namely, the ’803 patent states that “system 100 includes the GPU chiplet 106-1 communicating with the CPU 102 over a direct connection or via other buses, bridges, switches, routers, and the like.” EX1001-’803, 2:59-67.

**(d)Element 1[c]: [wherein the GPU chiplet array includes:
a second GPU chiplet communicably coupled to the first
GPU chiplet via a passive crosslink,**

165. Kelleher and Collins render obvious Element 1[c].

166. Kelleher and Collins teach that *the GPU chiplet array* (multi-chip module 213) *includes: a second GPU chiplet* (GPU chip 240 shown in gold in Figure 2, below) *communicably coupled to the first GPU chiplet* (GPU chip 220 shown in purple in Figure 2, below) *via a passive crosslink* (Kelleher’s interlink 250 as modified by Collins’s teaching of an interconnect structure/embedded multi-die bridge (EMIB) with conductive paths).

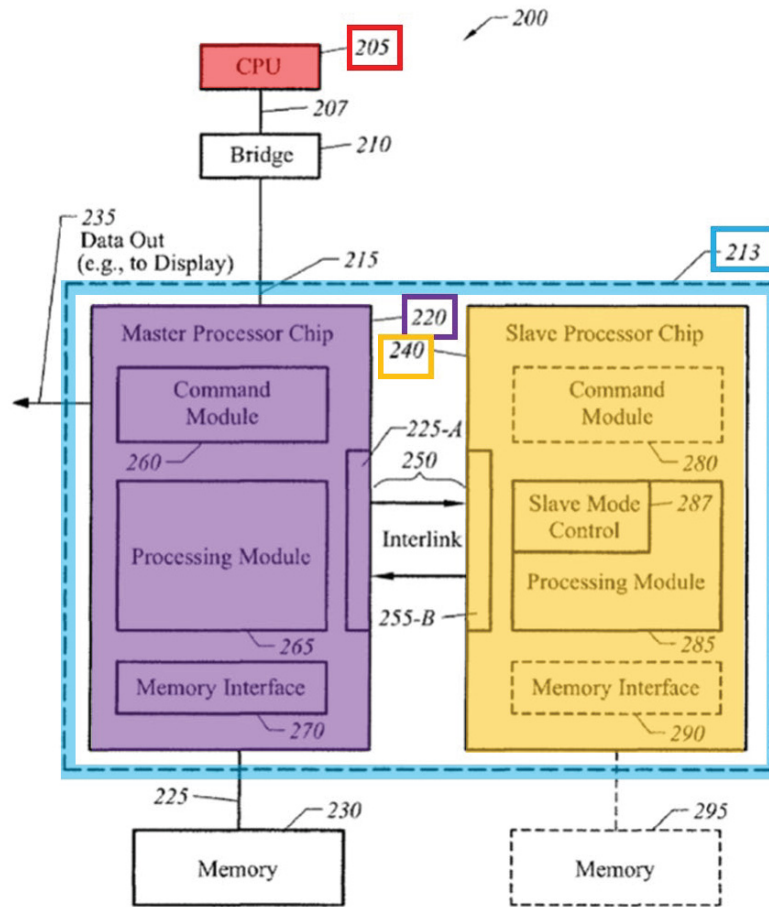
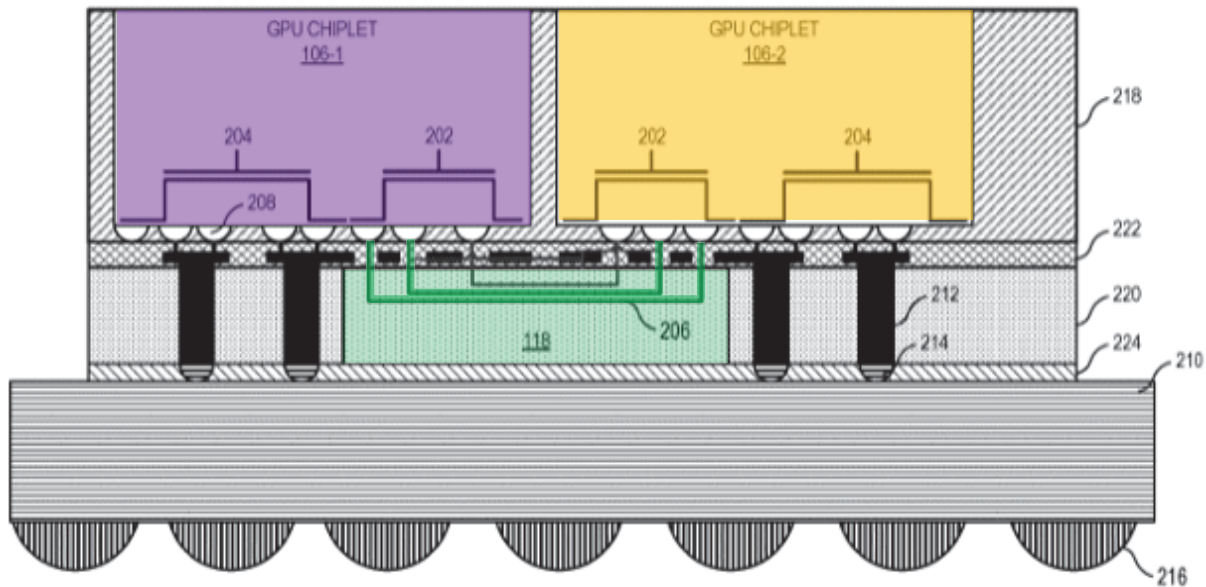


FIG. 2

EX1004-Kelleher, Figure 2.

167. Before explaining the relevant disclosures of Kelleher and Collins, I am including the following brief overview of how the '803 patent describes a "passive crosslink."

168. Figure 2 of the '803 patent shows a passive crosslink 118 that couples the first chiplet 106-1 (shown in purple) and the second chiplet 106-2 (shown in gold).



EX1001-'803, Figure 2.

169. The '803 patent states that Figure 2 “provides a sectional view of GPU chiplets 106-1, 106-2, and passive crosslink 118 of FIG. 1 taken at section A-A.” EX1001-'803, 3:57-4:2. Although the patent does not provide a lot of details regarding the “passive crosslink,” it explains that the passive crosslink 118 contains a passive interposer with conductive traces 206 for communicably coupling the chiplet 106-1 and the chiplet 106-2. EX1001-'803, 4:3-21. Specifically, the '803 patent notes that:

[T]he GPU chiplets 106 are communicably coupled by way of the passive crosslink 118. In various embodiments, **the passive crosslink 118 is an interconnect chip** constructed of silicon, germanium or other semiconductor materials and may be bulk semiconductor, semiconductor on insulator or other designs. **The passive crosslink 118 includes a plurality of internal conductor traces**, which

may be on a single level or multiple levels as desired. Three of the traces are illustrated in FIG. 2 and labeled collectively as traces 206. **The traces 206 interface electrically with conductor structures of the PHY regions 202 of the GPU chiplets 106 by way of conducting pathways. It is noted that the passive crosslink 118 does not contain any through silicon vias (TSVs). In this manner, the passive crosslink 118 is a passive interposer die that communicably couples and routes communications between the GPU chiplets 106, thereby forming a passive routing network.**

EX1001-'803, 4:3-21.

170. In Kelleher, “[s]lave GPU chip 240 is packaged with GPU chip 220 as a multi-chip module 213 that may be used as a replacement for an individual GPU chip.” EX1004-Kelleher, 3:46-50. Kelleher further explains that “[s]lave GPU chip 240 is coupled to GPU chip 220 via an interlink 250...[which] facilitate[s] seamlessly offloading processing tasks to slave GPU chip 240 as if processing module 285 of slave GPU chip 240 was an internal extension of processing module 265 of GPU chip 220.” EX1004-Kelleher, 4:20-35, 7:1-8.

171. In Collins, Figure 2 shows an interconnect structure / embedded multi-die interconnect bridge (EMIB) with conductive paths, and Collins discloses that this structure/EMIB is “passive.” For example, Collins explains that

The semiconductor package 100 further includes an **interconnect structure 108**. The interconnect structure 108 couples the chiplets 104 to the memory device 106 **[or to another chiplet according to EX1005-Collins, 4:24-29, 16:24-27]. In some embodiments, the interconnect structure 108 includes one or more embedded multi-**

die interconnect bridges (EMIBs), where each of the EMIBs are embedded within the substrate assembly 102. The substrate assembly 102 includes conductive elements that couple the chiplets 104 and the memory device 106 **[or to another chiplet according to EX1005-Collins, 4:24-29, 16:24-27]** to the EMIBs. Each of the EMIBs include a piece of **passive silicon** with conductive elements located within the passive silicon for conducting signals. In other embodiments, the interconnect structure 108 may include active EMIBs with repeaters designed into the bridge, interposers (such as silicon interposers, organic interposers, or other interposers known in the art), fan out wafer level structures, or some combination thereof.

EX1005-Collins, 3:49-64.

The semiconductor package 300 further includes an interconnect structure 304. In the illustrated embodiment, the interconnect structure 304 comprises an interposer. The interposer is mounted to a surface of the substrate assembly 302. The interposer can be passive or active silicon with one or more conductive paths located within the silicon.

EX1005-Collins, 5:42-47.

172. Collins provides embodiments in Figures 1 and 2 where “the interconnect structure 108 includes a first EMIB 108*a*, a second EMIB 108*b*, a third EMIB 108*c*, and a fourth EMIB 108*d*.” EX1005-Collins, 3:65-4:23. Collins notes that in those embodiments, “[t]he first EMIB 108*a* couples the first chiplet 104*a* to the memory device 106, the second EMIB 108*b* couples the second chiplet 104*b* to the memory device 106, the third EMIB 108*c* couples the third chiplet 104*c* to the

memory device 106, and the fourth EMIB 108*d* couples the fourth chiplet 104*d* to the memory device 106.” EX1005-Collins, 3:65-4:23.

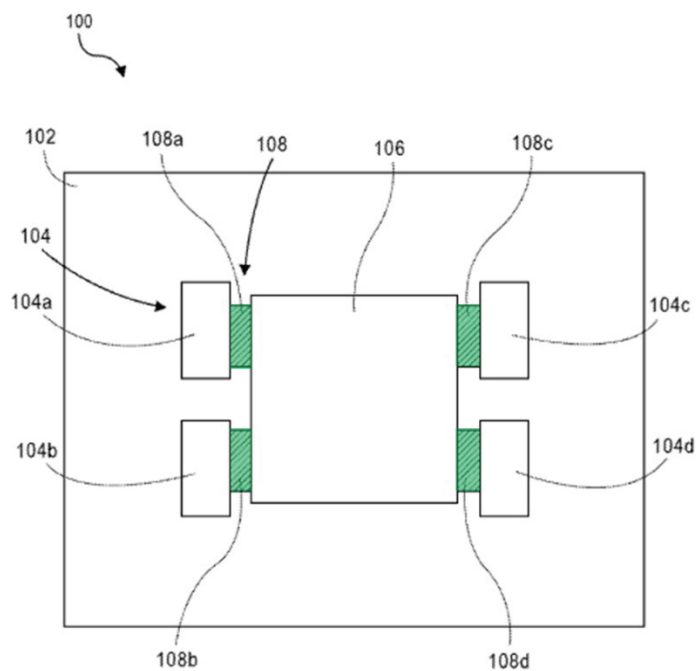


FIG. 1

EX1005-Collins, Figure 1.

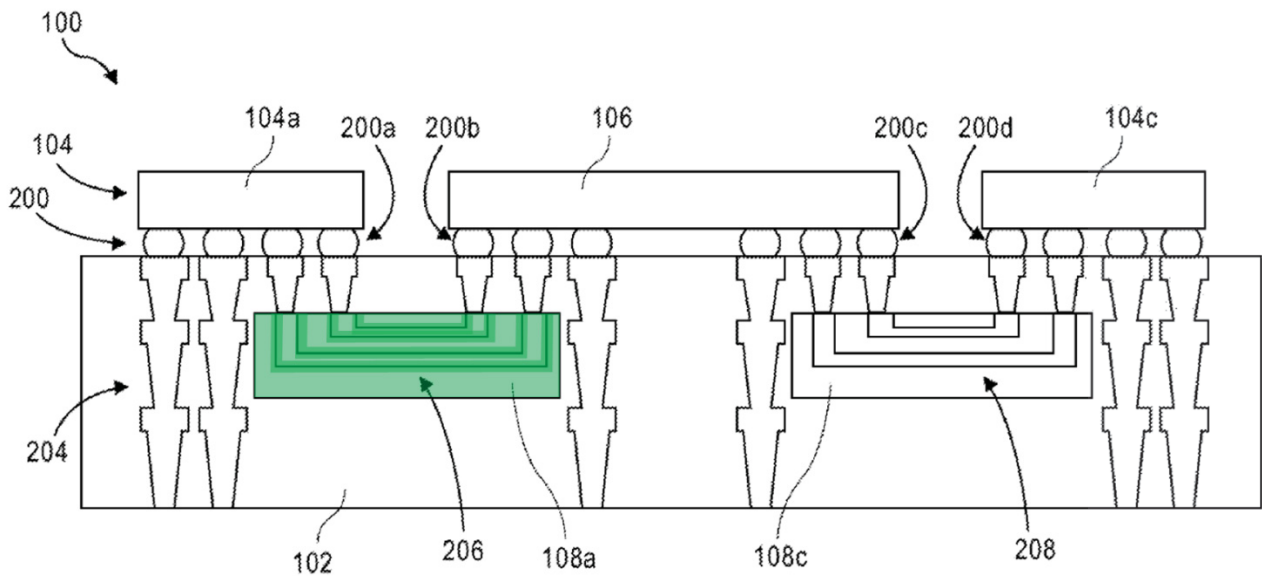


FIG. 2

EX1005-Collins, Figure 2.

173. As I have previously explained in Section VIII.B [Overview of the Prior Art—Collins], although Figure 1 and Figure 2 depict EMIBs connecting chiplets to a “memory device 106,” Collins expressly notes that in some embodiments, “the EMIBs 108 may couple two or more of the chiplets 104 together... The chiplets 104 coupled with each other via the EMIBs 108 may utilize the EMIBs 108 for communication between the chiplets 104....” EX1005-Collins, 4:24-39. Collins further explains that “the interconnect structure further couples a physical layer (PHY) of the first chiplet to a PHY of the second chiplet.” EX1005-Collins, 16:24-27.

174. I have produced a modified version of Figure 2 below after applying Collins's teachings of chiplet-to-chiplet embodiments to the chiplet-to-memory device embodiment of original Figure 2. EX1005-Collins, 4:24-39, 16:24-27.

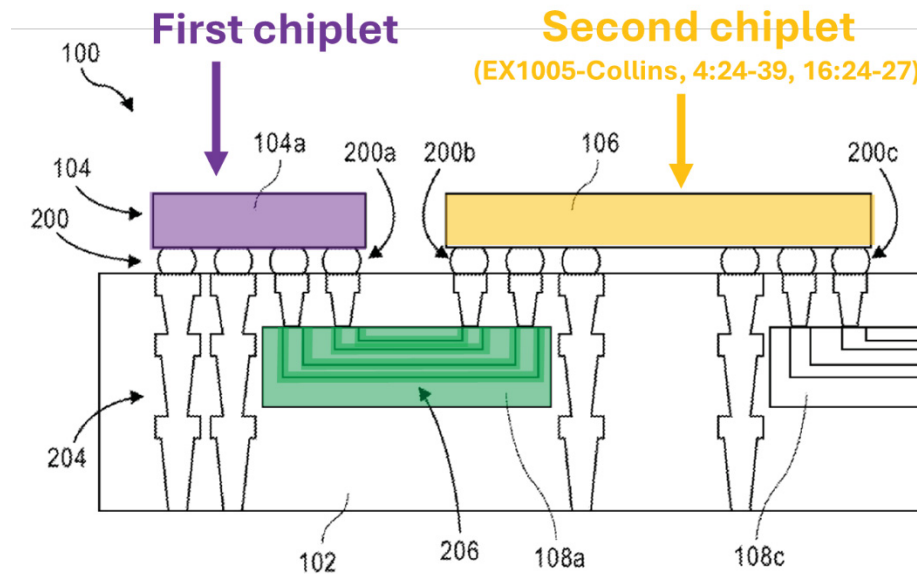


FIG. 2

EX1005-Collins, Modified Figure 2 (replacing memory device 106 with another chiplet according to Collins's chiplet-to-chiplet embodiment set forth, for example, in 4:24-39 and 16:24-27).

175. Furthermore, in Figure 2 of Kelleher shown below, Kelleher teaches that the GPU chiplet array (multi-chip module 213 shown in blue) includes a second GPU chiplet (GPU chip 240 shown in gold) that is communicably coupled to the first GPU chiplet (GPU chip 220 shown in purple) via an interlink 250 (shown in green).

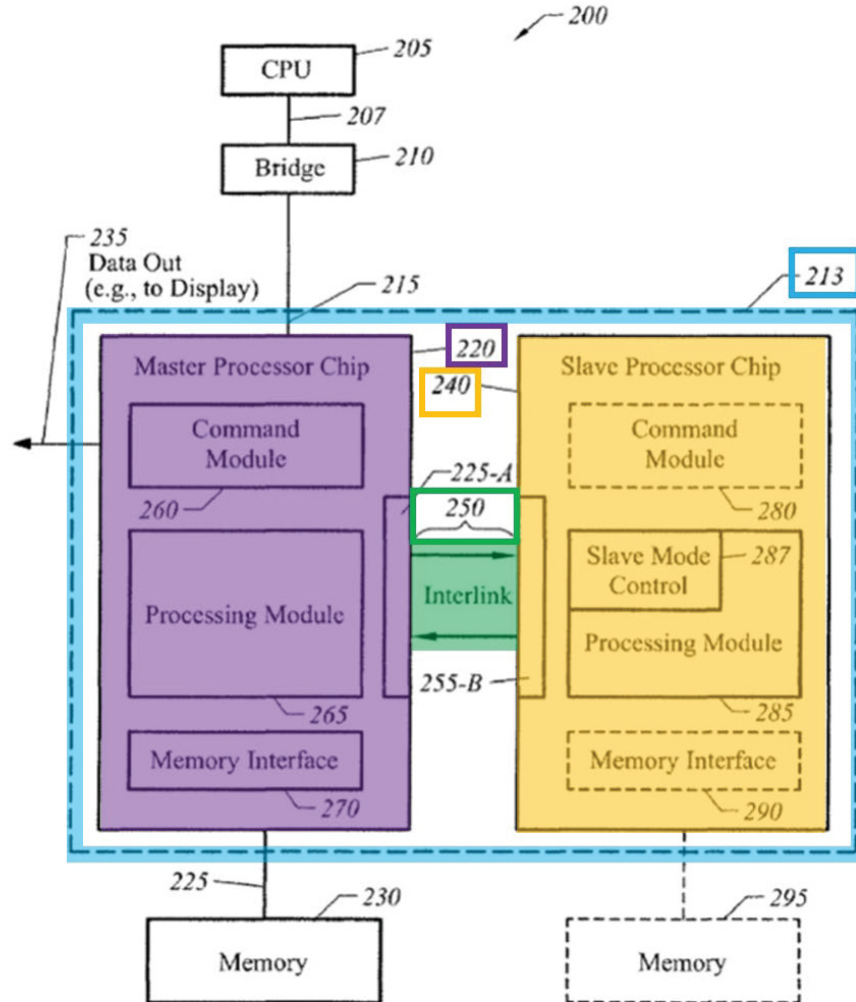


FIG. 2

EX1004-Kelleher, at Figure 2. For example, Kelleher states that the “[s]lave GPU chip 240 is packaged with GPU chip 220 as a multi-chip module 213 that may be used as a replacement for an individual GPU chip.” EX1004-Kelleher, at 3:46-50.

Further, Kelleher states:

Slave GPU chip 240 is coupled to GPU chip 220 via an interlink 250. Interlink 250 is preferably a low latency, high bandwidth link. Interlink 250 should have a

sufficiently low latency that the round trip delay time associated with sending and receiving data on interlink 250 does not exceed a timing margin for processing data. Additionally, the aggregate bandwidth of interlink 250 is preferably comparable to that of internal buses used to transmit graphics data of the type processed by graphics processing module 265. For example, the bandwidth of interlink 250 is preferably sufficiently large to accommodate sending and receiving data corresponding to the full input/output data width of processing module 285. ***These attributes of interlink 250 facilitate seamlessly offloading processing tasks to slave GPU chip 240 as if processing module 285 of slave GPU chip 240 was an internal extension of processing module 265 of GPU chip 220.***

EX1004-Kelleher, at 4:20-35; *see also* EX1004-Kelleher, at 7:7-8 (“Slave GPU chip 740 is coupled to master GPU 720 by interlink 750.”)

176. In sum, Kelleher and Collins renders obvious that *the GPU chiplet array* (multi-chip module 213 shown in blue in Figure 2 of Kelleher) *includes: a second GPU chiplet* (GPU chip 240 shown in gold in Figure 2 of Kelleher) *communicably coupled to the first GPU chiplet* (GPU chip 220 shown in purple in Figure 2 of Kelleher) *via a passive crosslink* (Collins’s teaching of an interconnect structure/embedded multi-die bridge (EMIB) with conductive paths coupling chiplets). Moreover, as explained above, since Collins discloses that its EMIB with conductive paths coupling chiplets is “passive” (EX1005-Collins, 3:49-4:64), such structure constitutes the passive crosslink under NVIDIA’s proposed construction (“passive crosslink”) and under Onesta IP’s proposed construction (“a non-active

connecting structure”). Furthermore, Collins’s EMIB with conductive paths similarly discloses the passive crosslink under the Staff’s proposed construction, because it discloses “a non-active [passive] semiconductor die [EMIB/interposer (EX1005-Collins, 3:49-64)], which die does not include any though silicon vias.” EX1043-Claim-Construction-Chart at 4. I also explain this in my discussion of Claim 2, Element 1[d], which I incorporate here by reference.

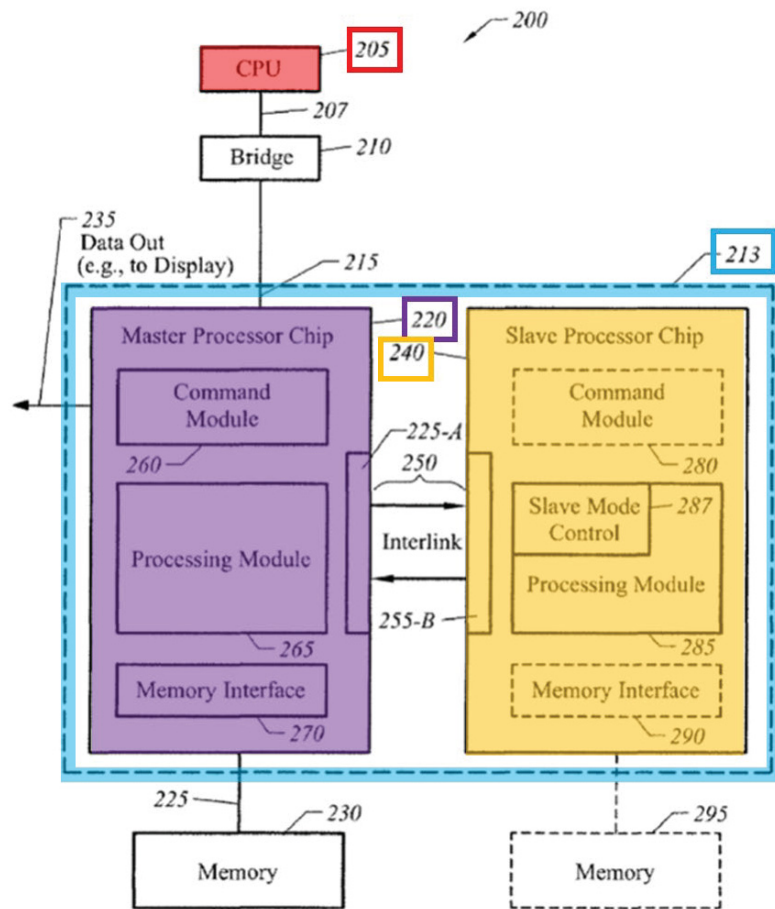


FIG. 2

EX1004-Kelleher, Figure 2.

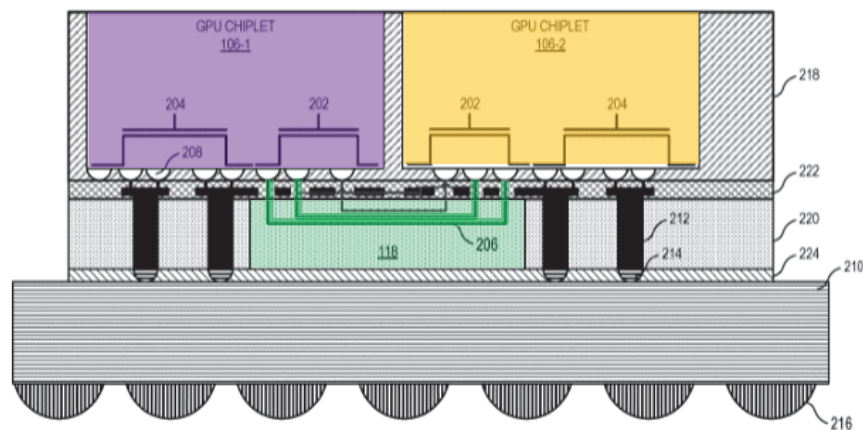
177. Below I explain reasons why a person of ordinary skill in the art would have been motivated to combine the teachings of Kelleher and Collins, which I incorporate herein by reference.

(e) The Kelleher and Collins Combination

178. A person of ordinary skill in the art would have been motivated to combine Kelleher's and Collins's teachings by using the teachings of Collins's interconnect structure/embedded multi-die bridge (EMIB) to communicably couple Kelleher's GPU chip 220 and GPU chip 240.

179. Kelleher and Collins are analogous art to each other and claims 1-4, 7-10 of the '803 patent because each of them relates to chiplets that are communicably coupled to each other and that are mounted on a common substrate.

180. Figure 2 of the '803 patent, which I have reproduced below, shows two chiplets mounted on the same substrate and an interposer communicably couples the chiplets.



EX1001-'803, Figure 2, 4:22-58. Further, the patent explains that:

[T]he GPU chiplets 106 are communicably coupled by way of the passive crosslink 118. In various embodiments, the passive crosslink 118 is an interconnect chip constructed of silicon, germanium or other semiconductor materials and may be bulk semiconductor, semiconductor on insulator or other designs. The passive crosslink 118 includes a plurality of internal conductor traces, which may be on a single level or multiple levels as desired. Three of the traces are illustrated in FIG. 2 and labeled collectively as traces 206. The traces 206 interface electrically with conductor structures of the PHY regions 202 of the GPU chiplets 106 by way of conducting pathways. It is noted that the passive crosslink 118 does not contain any through silicon vias (TSVs). In this manner, the passive crosslink 118 is a passive interposer die that communicably couples and routes communications between the GPU chiplets 106, thereby forming a passive routing network.

EX1001-'803, 4:3-21.

181. Figure 2 and Figure 3 of Kelleher, which is reproduced below, also show two chiplets (GPU chips 220 and 240) mounted on the same substrate (package 305 and circuit board 310), with an interlink communicably coupling the chiplets. Kelleher states that “[s]lave GPU chip 240 is coupled to GPU chip 220 via an interlink 250.... Interlink 250 should have a sufficiently low latency that the round trip delay time associated with sending and receiving data on interlink 250 does not exceed a timing margin for processing data.” EX1004-Kelleher, 4:20-35. Kelleher further explains that “GPU chip 240 and GPU chip 220 are packaged on top of a

common multi-chip package 305 that is then placed on a board 310.” EX1004-Kelleher, 5:39-57.

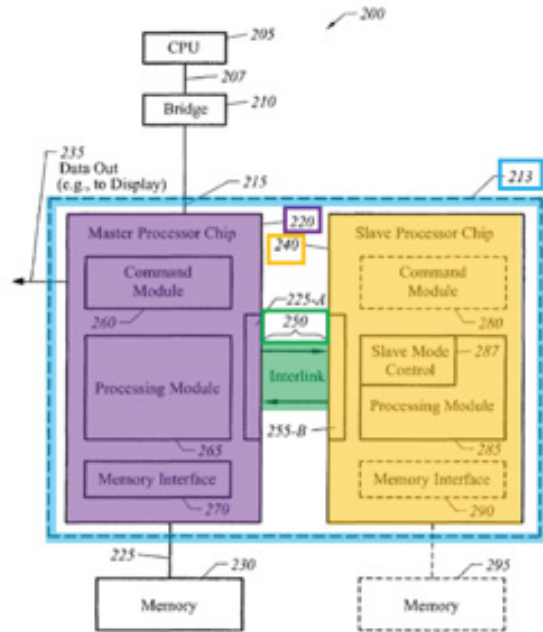


FIG. 2

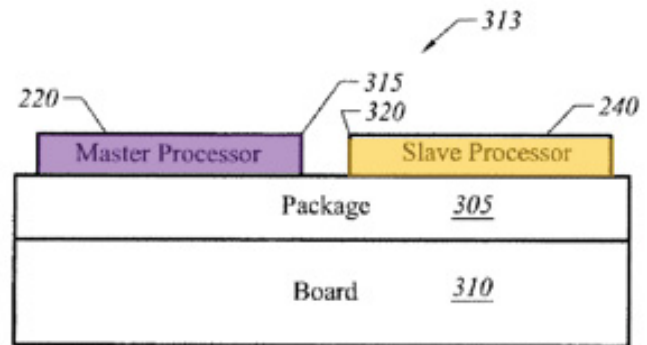


FIG. 3

EX1004-Kelleher, Figure 2 and Figure 3.

182. Collins teaches that “the EMIBs 108 may couple two or more of the chiplets 104 together... The chiplets 104 coupled with each other via the EMIBs 108 may utilize the EMIBs 108 for communication between the chiplets 104.” EX1005-Collins, 4:24-39.

183. A person of ordinary skill in the art also would have been motivated to combine the teachings of Kelleher and the teachings of Collin teachings by using the passive interconnect structure/EMIB of Collins to communicably couple the GPU chips 220 and 240 of Kelleher. As I will explain further below, incorporating

Collins's passive interconnect structure teachings into Kelleher's system of GPU chips would be beneficial because it would allow for more bandwidth when transferring data and would also eliminate the need for additional components, such as heat sinks, that are typically included in such systems to address heat issues.

184. For example, Figure 3 of Kelleher shows an "edge-to-edge" configuration of GPU chips 240 and 220 on package 305, where sub-links of the interlink extend along the top of the package 305 from the edge 315 of the chip 220 (shown in purple in Figure 3 below) to the edge 320 of the chip 240 (shown in gold in Figure 3 below). EX1004-Kelleher, at 5:39-57.

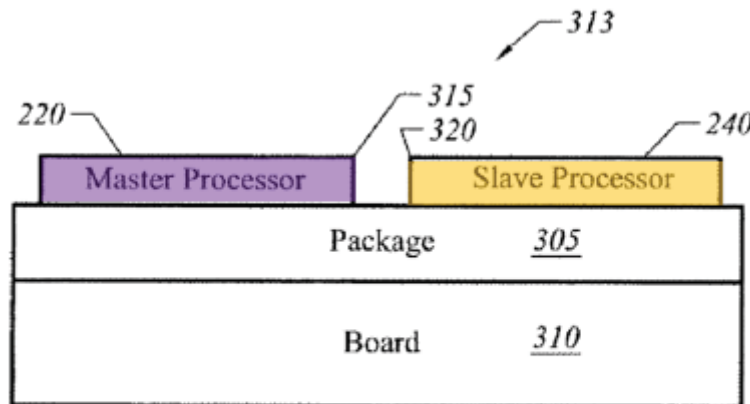


FIG. 3

EX1004-Kelleher, Figure 3. Namely, Kelleher states:

Referring to FIG. 3, in one embodiment **GPU chip 240 and GPU chip 220 are packaged in an edge-to-edge multi-chip module 313** with the interlink formed by fabricating signal leads between the chips. GPU chip 240 and GPU chip 220 are packaged on top of a common multi-chip package 305 that is then placed on a board 310.

However, high performance GPUs process large amounts of data, particularly for the shader stage. An **edge-to-edge configuration permits only a comparatively small number of low latency sub-links to be formed between edges 315 and 320.**

EX1004-Kelleher, at 5:39-57.

185. However, Kelleher cautions that “high performance GPUs process large amounts of data, particularly for the shader stage,” and that “an edge-to-edge configuration permits only a comparatively small number of low latency sub-links to be formed between edges 315 and 320.” EX1004-Kelleher, 5:39-57.

186. Additionally, this configuration introduces “an associated internal (chip) propagation delay for signals to propagate between the periphery of the chip to internal source or destination points.” EX1004-Kelleher, 5:39-65. As a result, Kelleher identifies a drawback: “an edge-to-edge packaging configuration ... may not permit an interlink 250 to be formed having sufficient latency and bandwidth attributes required to support parallel processing of stages in high performance GPUs requiring a large bandwidth for data transfer, such as shader stages.” EX1004-Kelleher, 5:39-57.

187. To address this limitation, Kelleher describes an alternative embodiment in which “multi-chip module 213 has slave GPU chip 240 and GPU chip 220 packaged parallel and in close proximity to each other across a substantial portion of the area of each chip.” EX1004-Kelleher, at 5:58-65. This arrangement

“facilitates using leads, such as vias, to form a substantial number of parallel sublinks that are distributed across the area of each chip with a low latency per sublink and with a high total aggregate bandwidth.” EX1004-Kelleher, at 5:39-65. As shown below in Figure 4, Kelleher proposes mounting the master GPU chip (in purple) and slave GPU chip (in gold) on opposite sides of the package in parallel.

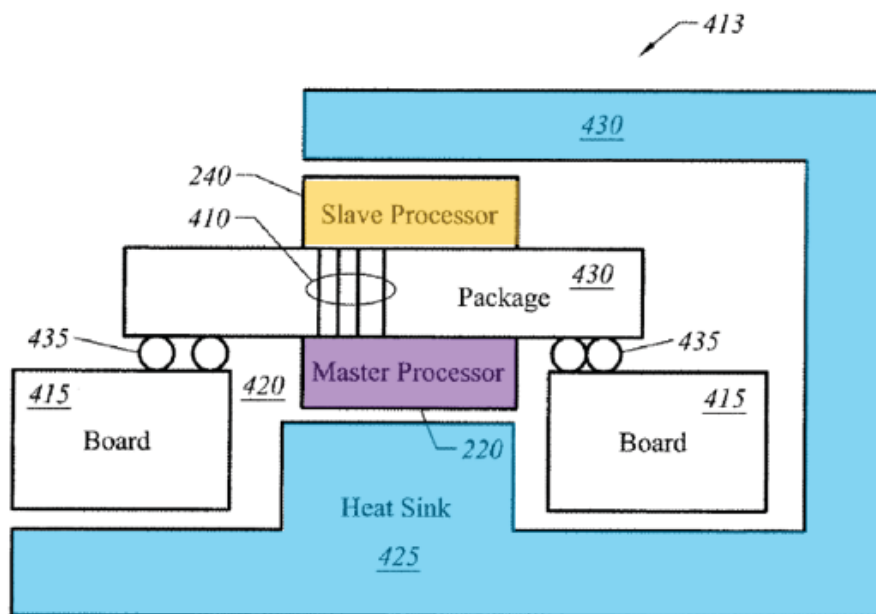


FIG. 4

EX1004-Kelleher, Figure 4.

188. As shown, in Figure 4 above, the sub-links 410 of the interlink span from the bottom of slave GPU chip (in gold), through the package, to corresponding portions of the top area of the master GPU chip (in purple).

189. In this arrangement, more sub-links can be formed across the top area of the chip 220 and bottom area of the chip 240 than along edges of the chips 220

and 240, as in the “edge-to-edge” configuration of Figure 3. As a result, Kelleher explains that this “parallel” configuration provides greater bandwidth for transferring data:

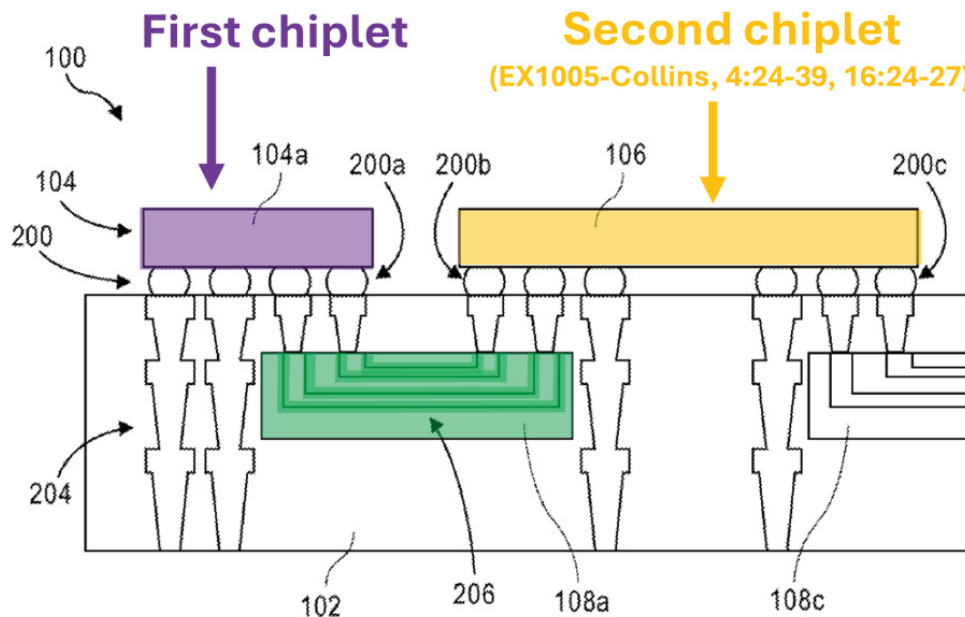
FIG. 4 shows an embodiment of a multi-chip module 413 for implementing a low latency, high bandwidth interlink 250. In this embodiment GPU chip 220 and slave GPU 240 are flip-chip mounted to opposite sides of a package 405. The package 405 may, for example, comprise a coreless substrate. Core-less substrates utilize a stacked build-up process to form the substrate. A core-less substrate utilizes only the build up layers to form interconnects and hence lacks a conventional core. Core-less substrates utilize micro-vias that permit an increased density (pitch) of vias, have less signal attenuation due to the lower package thickness, have reduced inductance, facilitate high speed data transfer, and provide other benefits. Conductive vias 410 within package 405 are used to form interlink 250. The vias 410 may be distributed across the area of GPU chip 220 and slave GPU 240 to provide a combined (total) bandwidth that scales with the number of vias. The transit distance for signals to travel from one processor to another corresponds to the thickness of package 405. In one embodiment, the board 415 onto which the package is mounted has a hole 420 to permit both GPU chip 220 and slave processor 240 to be cooled to a heat sink 425. In one embodiment heat sink 425 includes a wrap-around section 430 for cooling slave GPU 240. Package 405 may be mounted to board 415 using conventional bonding processes, such as by using solder balls 435.

EX1004-Kelleher, 5:66-6:23.

190. However, Kelleher’s “parallel” configuration still has disadvantages, such as requiring additional components to ensure proper heat dissipation. Because GPU chips generate a substantial amount of heat, when the large surface area of the

top of the master GPU chip and the large surface area of the bottom of the slave GPU chip are spaced close together and face each other, additional measures—such as the heat sinks shown in Figure 4 of Kelleher—are required to ensure proper operation by cooling the chips. EX1004-Kelleher, 6:53-55, 6:16-21. Kelleher notes that “[h]igh performance GPU chips dissipate a substantial amount of heat. ... Consequently, packaging two GPU chips in a multi-chip module 213 will increase heat dissipation. Thermal modeling is desirable to optimize the heat sinking and cooling for multi-chip module 213.” EX1004-Kelleher, 6:53-59. Kelleher further explains that “the board 415 onto which the package is mounted has a hole 420 to permit both GPU chip 220 and slave processor 240 to be cooled to a heat sink 425. In one embodiment heat sink 425 includes a wrap-around section 430 for cooling slave GPU 240.” EX1004-Kelleher, 6:16-21. Adding heat sinks, however, increases both the system’s size and the system’s cost.

191. Due to Kelleher’s disadvantages set forth above, a person of ordinary skill in the art would have been motivated to communicably couple Kelleher’s master and slave GPU chips (shown in purple and gold) with Collins’s interconnect structure/EMIB (shown in green), as illustrated in Collins’s Modified Figure 2 reproduced below.



EX1005-Collins, Modified Figure 2 (replacing memory device 106 with another chiplet according to Collins's disclosure in 4:24-39 and 16:24-27).

192. By applying Collins's EMIB teaching, the number of sub-links for communicably coupling two chips is not restricted to the number that can be formed along the edges of the chips, as is the case of Kelleher's Figure 3 embodiment. Instead, a greater number of sub-links can be formed along the bottom surfaces of the chips, thereby increasing the bandwidth for transferring data between them. EX1004-Kelleher, at 5:39-57.

193. In addition, by applying Collins's EMIB teaching, only the small opposing edges of the first and second chips face each other, rather than the large

top and bottom surfaces of the chips as in Kelleher's Figure 4 embodiment. This arrangement significantly reduces the need for heat sinks and other cooling techniques, which in turn substantially decreases the size and cost of a multi-chip module containing the chips. EX1005-Collins, 4:24-39. Thus, Collins discloses the benefits of applying the teachings of its interconnect to Kelleher's system.

194. Moreover, the advantages of using an EMIB were well-known as of the '803 patent's June 28, 2019 filing date. Thus, a person of ordinary skill in the art would have recognized that an EMIB is a "dedicated ultra-high-density interconnection between dies within [a] package." EX1037-Liu, [0028]. For example, because EMIBs are positioned close to the chips, the wires used are significantly smaller than those required to traverse a full interposer. EX1037-Liu, [0028]. This reduced wire length "help[s] to significantly reduce loading and directly boost[s] performance." EX1037-Liu, [0028]. Additionally, EMIBs are less susceptible to unintended bending or distortion of the interposer, known as "warping." EX1037-Liu, [0029]. Liu explains:

In order to facilitate communications between two chips on a multi-chip package, the package may include an embedded multi-die interconnect bridge (EMIB) that is designed and patented by INTEL Corporation. An EMIB is a small silicon die that is embedded in the underlying substrate of a multi-chip package and that offers dedicated ultra-high-density interconnection between dies within the package. EMIBs generally include wires of minimal

length, which help to significantly reduce loading and directly boost performance.

EMIB solutions may be advantageous over other multi-chip packaging schemes that use a silicon interposer, which is prone to issues such as warpage and requires a comparatively large number of microbumps and through-silicon vias (TSVs) to be formed on and within the interposer, thereby reducing overall yield and increasing manufacturing complexity and cost. The number of dies that can be integrated using an interposer is also limited to that supported by EMIB technology.

EX1037-Liu, [0028]-[0029].

195. A 2014 article by Yazdani, entitled “Design and Direct Assembly of 2.5D/3D Rigid Silicon Interposer on PCB,” also describes the disadvantages of using a large silicon interposer, such as increased costs, decreased reliability, and increased “warpage, cracks, and thermal stresses.” EX1038-Yazdani, Abstract, 13-16.

196. In addition, a person of ordinary skill in the art would have known that EMIBs require fewer microbumps and TSVs, making them less complex and more cost-effective to manufacture compared to large interposers. EX1037-Liu, [0029].

197. A person of ordinary skill in the art also would have had a reasonable expectation of success of implementing Collins’s teachings of a passive EMIB to communicably couple Kelleher’s master and slave GPU chips. Collins teaches that the EMIB “may couple two or more of the chiplets 104....” EX1005-Collins, 4:32-39. Collins also provides implementation details for such coupling, explaining that

“the interconnect structure ... couples a physical layer (PHY) of the first chiplet to a PHY of the second chiplet.” EX1005-Collins, 16:24-27.

198. A person of ordinary skill in the art would have readily appreciated that the EMIB’s described capability of coupling two chiplets could readily be incorporated into Kelleher’s system to communicably couple the PHY region of Kelleher’s master GPU chip to the PHY region of Kelleher’s slave GPU chip. This combination would have been well within the skill of a person of ordinary skill in the art, who would have had a reasonable expectation of success in using Collins’s teaching of its interconnect structure/EMIB to couple Kelleher’s chiplets without undue experimentation. EX1005-Collins, 16:24-27.

199. A person of ordinary skill in the art therefore would have been motivated to use Collins’s teachings of a passive interconnect structure/EMIB to implement the interlink in Kelleher’s system because doing so would have yielded predictable results that are aligned with Kelleher’s performance objectives. The combination involves incorporating Collins’s well-known passive interconnect structure/EMIB teachings to Kelleher’s well-known chiplet system that was ready for improvement, and it would have been a matter of routine design choice within the skill of a person of ordinary skill in the art.

200. It is therefore my conclusion that a person of ordinary skill in the art would have been motivated to combine the teachings of Kelleher and Collins.

(f) Element 1[d]: wherein the passive crosslink is dedicated for inter-chiplet communications.

201. Kelleher and Collins renders obvious Element 1[d].

202. Kelleher and Collins render obvious that *the passive crosslink* (Collins’s teaching of an interconnect structure/embedded multi-die bridge (EMIB) with conductive paths) *is dedicated for inter-chiplet communications* (includes communication pathways only between Kelleher’s master GPU chip and slave GPU chip).

203. As I explained in Section X (Claim Construction), the phrase “dedicated for inter-chiplet communications” should be interpreted as “used for transmission of only chiplet-to-chiplet signals.” In Figure 2 of Kelleher, which I have reproduced below, Kelleher’s interlink (in green) is used for transmission of signals between the master GPU chip (in purple) and slave GPU chip (in gold).

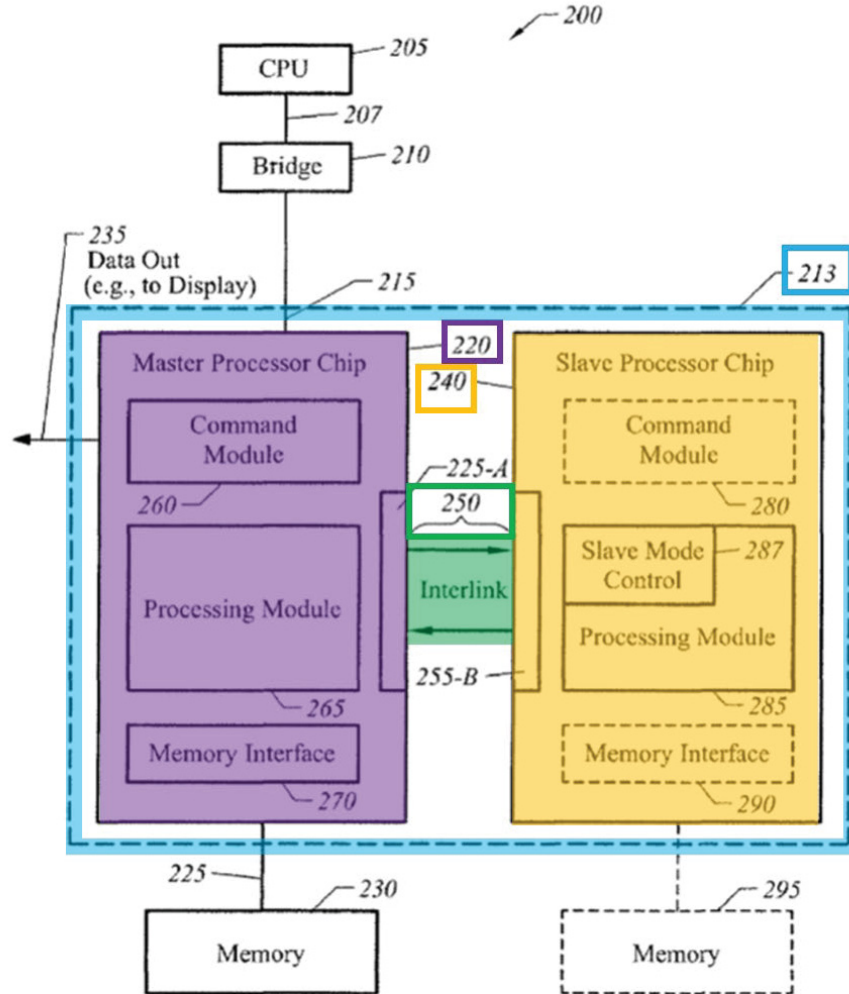


FIG. 2

EX1004-Kelleher, Figure 2.

204. Kelleher describes that “[s]lave GPU chip 240 is coupled to GPU chip 220 via an interlink 250.... [A]ttributes of interlink 250 facilitate seamlessly offloading processing tasks to slave GPU chip 240 as if processing module 285 of slave GPU chip 240 was an internal extension of processing module 265 of GPU chip 220.” EX1004-Kelleher, 4:20-35.

205. Figures 4–11 (reproduced below) of Kelleher further show the interlink being used to transmit signals between the master and slave GPU chips.

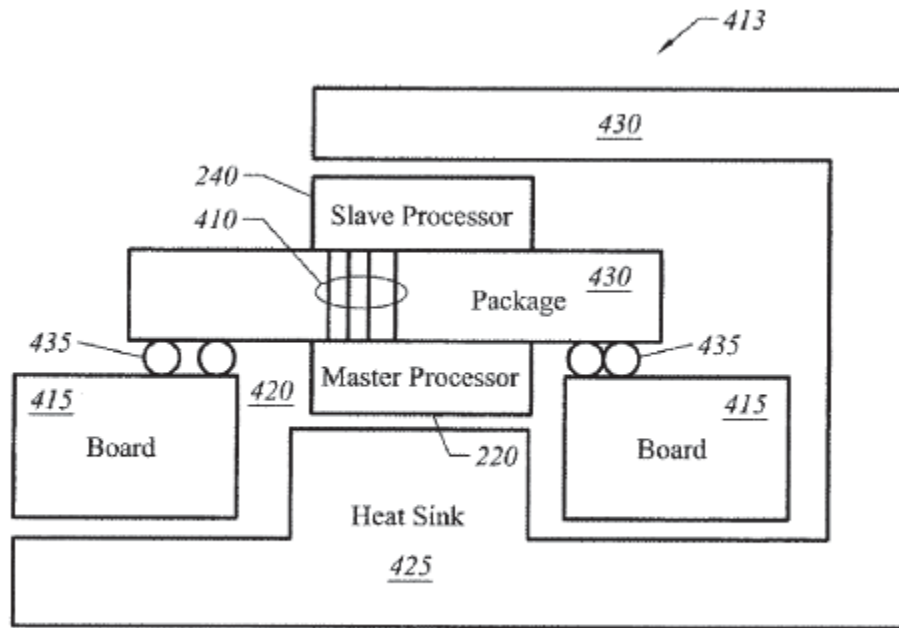


FIG. 4

EX1004-Kelleher, Figure 4.

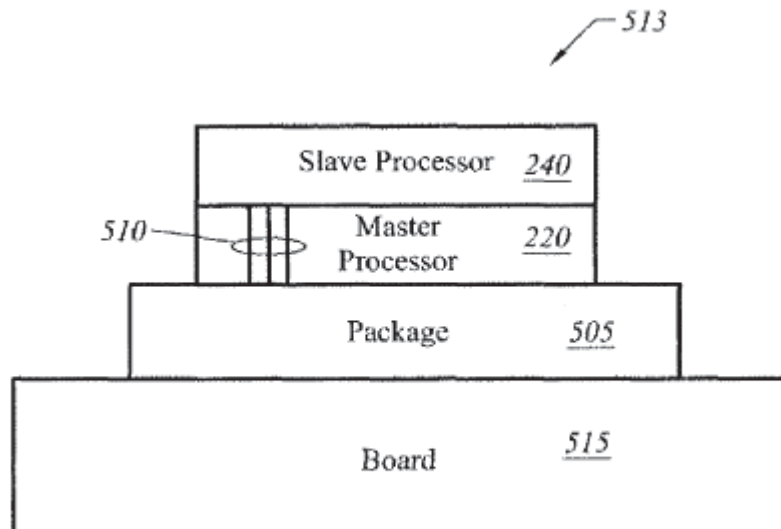


FIG. 5

EX1004-Kelleher, Figure 5.

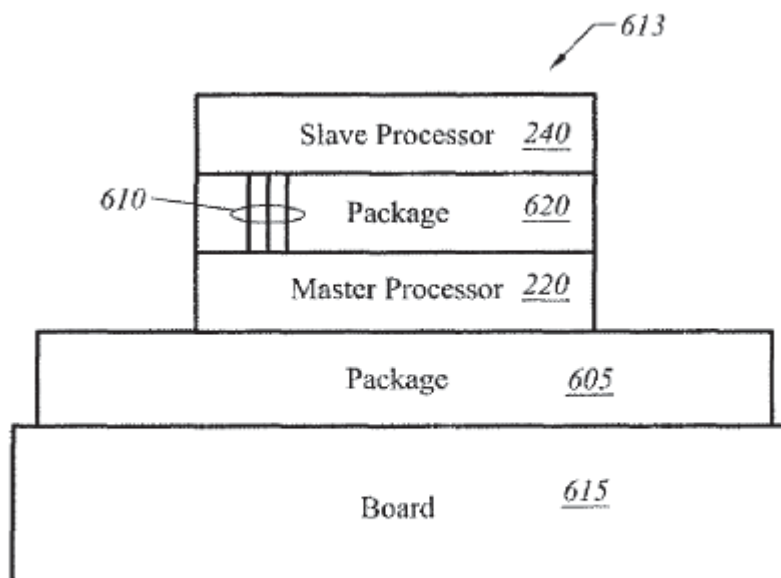


FIG. 6

EX1004-Kelleher, Figure 6.

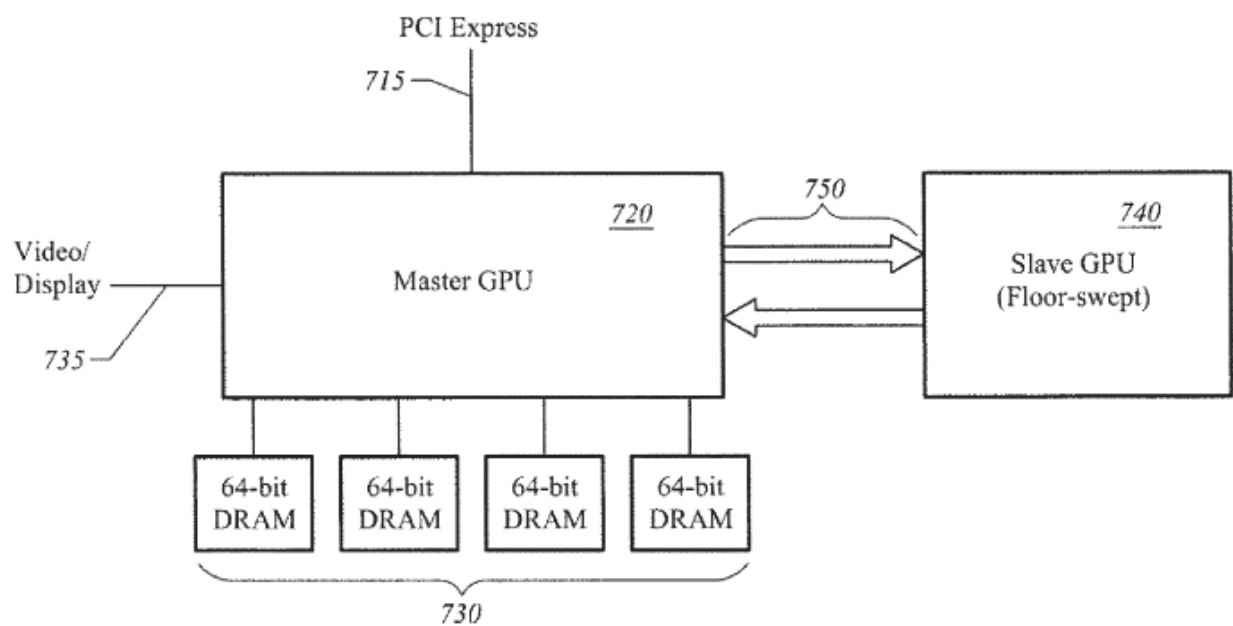


FIG. 7

EX1004-Kelleher, Figure 7.

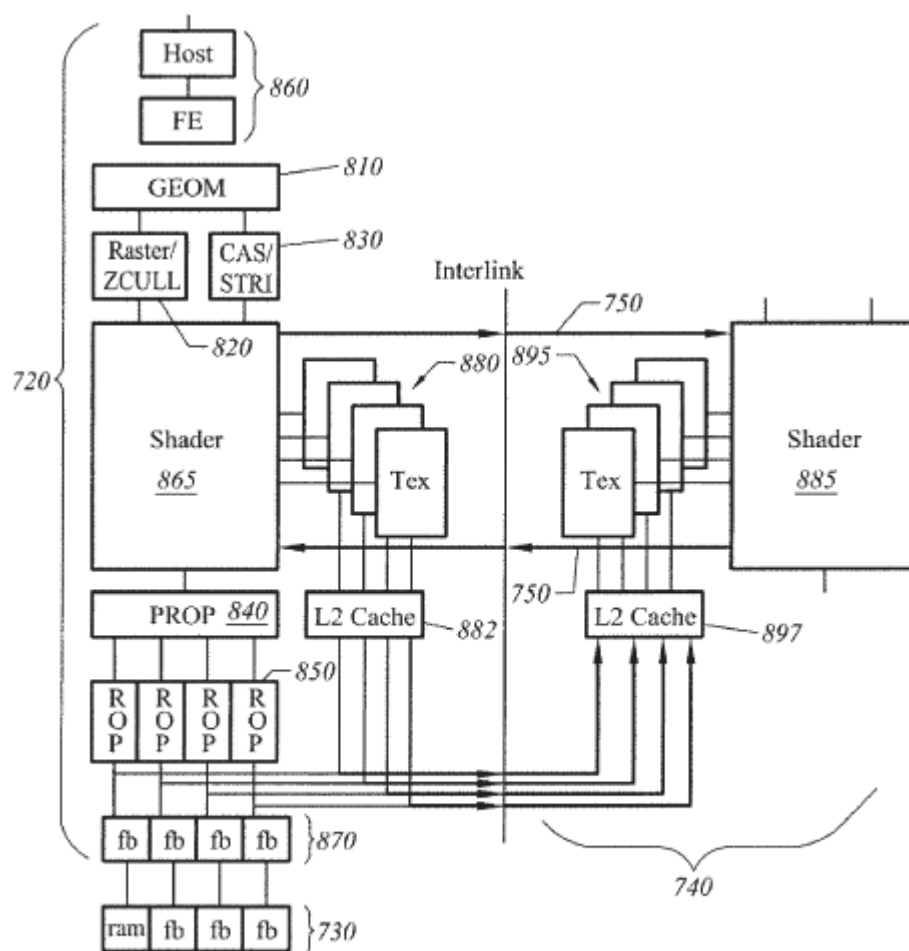


FIG. 8

EX1004-Kelleher, Figure 8.

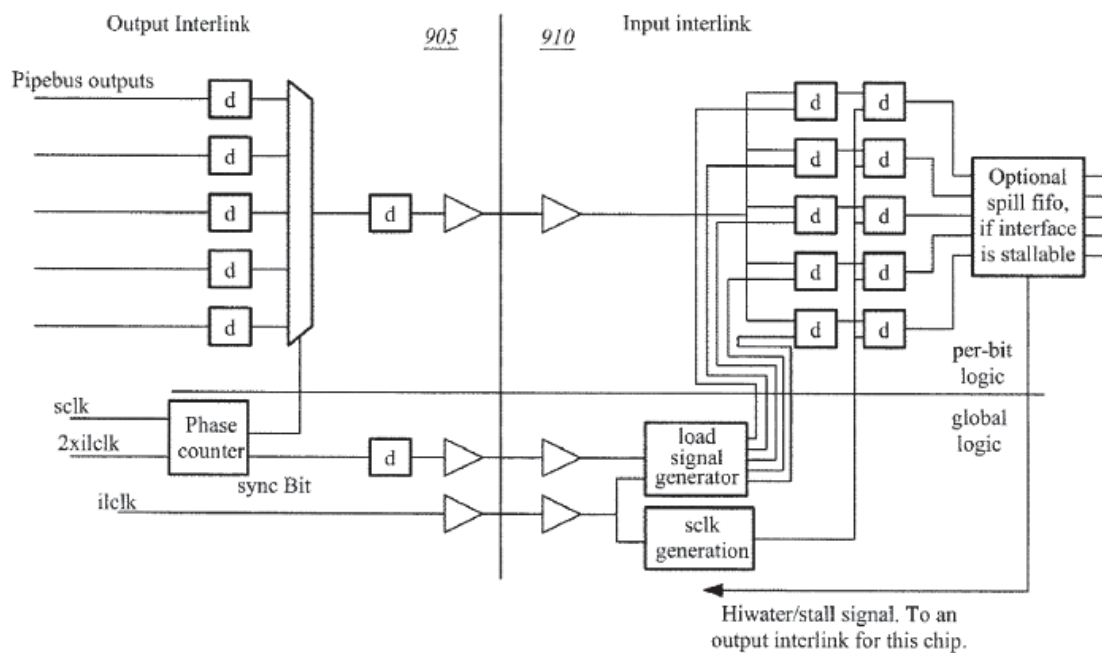


FIG. 9

EX1004-Kelleher, Figure 9.

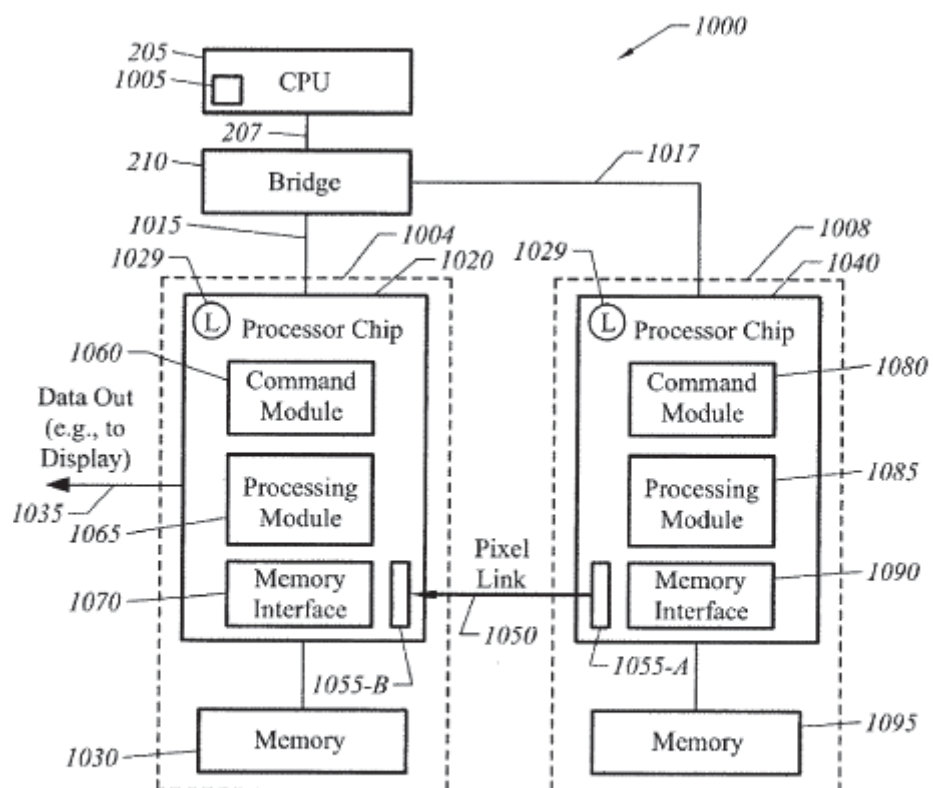


FIG. 10

EX1004-Kelleher, Figure 10.

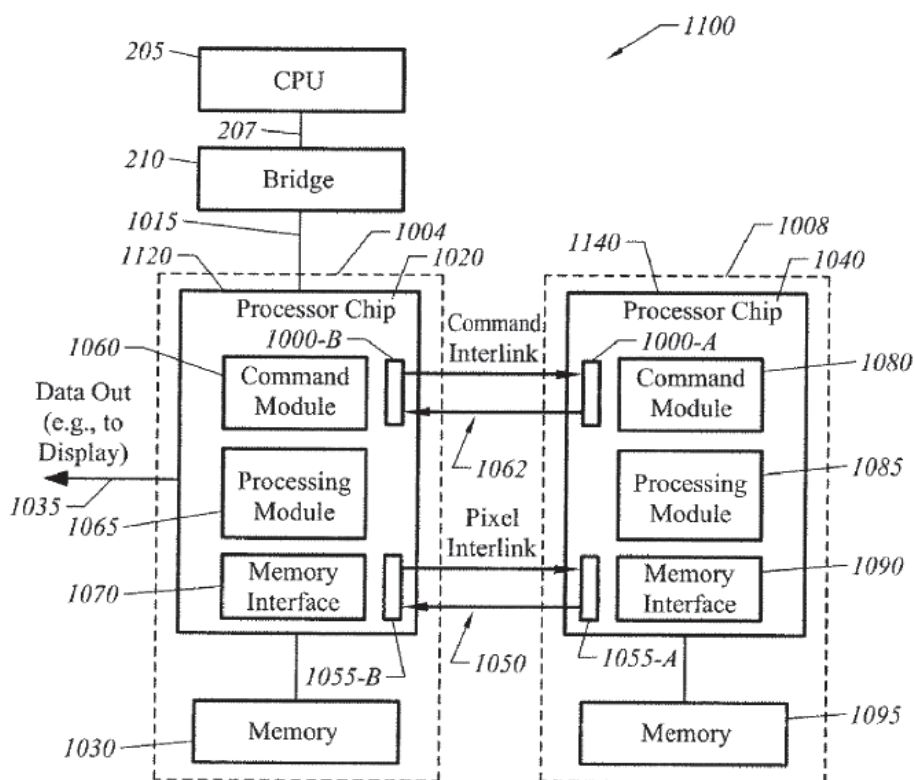


FIG. 11

EX1004-Kelleher, Figure 11.

206. As I will explain further below, by using the passive EMIB of Collins to couple the master and slave GPU chips of Kelleher, the Collins's passive EMIB, which corresponds to the claimed passive crosslink, is dedicated for inter-chiplet communications. In other words, Collins's passive EMIB is used only for transmitting chiplet-to-chiplet signals between Kelleher's master and slave GPU chips, which correspond to the claimed first and second chiplets.

207. In particular, in Modified Figure 2 of Collins, the passive EMIB (shown in green) would be used for transmission of only signals between the first chiplet (shown in purple) and the second chiplet (shown in gold). Namely, as noted above, Collins explains that “[t]he chiplets 104 coupled with each other via the EMIBs 108 may utilize the EMIBs 108 for communication between the chiplets 104” and that “the interconnect structure couples ... the first chiplet to ... the second chiplet.” EX1005-Collins, 4:24-39, 16:24-27, 18:29-32, Claim 5.

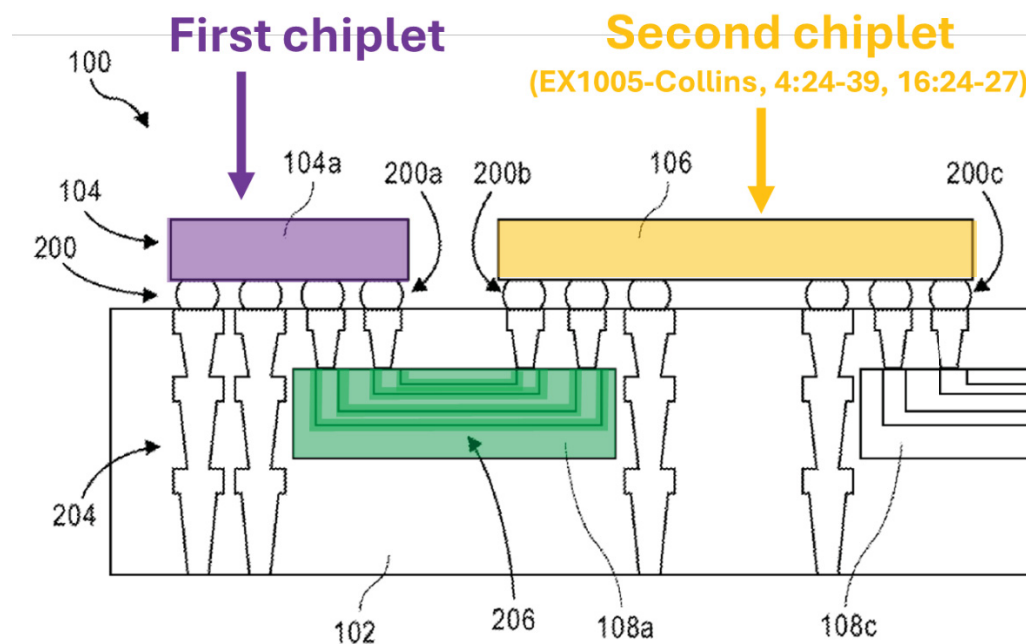


FIG. 2

EX1005-Collins, Modified Figure 2 (replacing memory device 106 with another chiplet according to Collins’s disclosure in 4:24-39 and 16:24-27). Moreover, as shown above, in this configuration, the EMIB (shown in green) is free of signaling

pathways, or conductive elements (such as traces, vias, and/or conductive planes) linking the first chiplet (shown in purple) and the second chiplet (shown in gold) to other components.

208. Furthermore, it would have been obvious to a person of ordinary skill in the art to only route chiplet-to-chiplet communications along the EMIB and use TSVs for power and/or ground. This is because TSVs have a larger cross-section than the traces in the EMIB and therefore are more suited than the EMIB to conduct for power and ground distribution with less resistive loss and less heating. EX1050-Ghaffarian, at 23.

209. Thus, when Collins's passive EMIB is used to couple Kelleher's master and slave GPU chips, the resulting combination teaches a passive crosslink (Collins's passive EMIB) that is dedicated solely to inter-chiplet communication between the two GPU chips. This is consistent with the '803 patent's teaching that the "passive crosslink 118 does not contain any through silicon vias (TSVs). In this manner, the passive crosslink 118 is a passive interposer die that communicably couples and routes communications between the GPU chiplets 106, thereby forming a passive routing network." EX1001-'803, 4:16-21.

210. For reasons that are similar to those present above, the combined teachings of Kelleher and Collins discloses a passive crosslink that is "dedicated for inter-chiplet communications" under the Staff's and Onesta IP's proposed

constructions. EX1043-Claim-Construction-Chart at 2. Namely, the combined teachings disclose that Collins's passive EMIB is configured to provide communication only between Kelleher's master GPU chip and slave GPU chip (*i.e.*, only between GPU chiplets) (per Onesta IP's proposed construction) and that the passive EMIB/interposer is a "die used for transmission of only chiplet-to-chiplet signals" (per the Staff's proposed construction). EX1043-Claim-Construction-Chart at 2. *See also* Claim 2.

211. It is therefore my opinion that Kelleher and Collins render obvious that *the passive crosslink* (Collins's teaching of an interconnect structure/embedded multi-die bridge (EMIB) with conductive paths) *is dedicated for inter-chiplet communications* (includes communication pathways only between Kelleher's master GPU chip and slave GPU chip).

2. Dependent Claim 2

(a) The system of claim 1, wherein the passive crosslink comprises a passive interposer die.

212. Claim 2 depends from claim 1. Therefore, I incorporate by reference my discussion of claim 1.

213. Collins teaches the additional requirement of claim 2.

214. Collins teaches that passive interposers may be used in place of EMIBs. Specifically, Collins explains:

The semiconductor package 100 further includes an interconnect structure 108. The interconnect structure 108 couples the chiplets 104 to the memory device 106. In some embodiments, the interconnect structure 108 includes one or more embedded multi-die interconnect bridges (EMIBs), where each of the EMIBs are embedded within the substrate assembly 102. The substrate assembly 102 includes conductive elements that couple the chiplets 104 and the memory device 106 to the EMIBs. Each of the EMIBs include a piece of passive silicon with conductive elements located within the passive silicon for conducting signals. In other embodiments, **the interconnect structure 108 may include** active EMIBs with repeaters designed into the bridge, **interposers (such as silicon interposers**, organic interposers, or other interposers known in the art), fan out wafer level structures, or some combination thereof.

EX1005-Collins, 3:49-64.

215. Collins further states that “[t]he interposer **can be passive** or active silicon with one or more conductive paths located within the silicon.” EX1005-Collins, 5:42-47.

216. It is therefore my opinion that Collins discloses that the passive crosslink—interconnect structure 108a—comprises a passive interposer die.

3. Dependent Claim 3

217. Claim 3 depends from claim 1. Therefore, I incorporate by reference my discussion of claim 1.

218. Kelleher and Collins render obvious the additional requirements of Claim 3.

(a) Element 3[a]: The system of claim 1, wherein the first GPU chiplet comprises a first PHY region including conductor structures for chiplet-to-chiplet communications,

219. Kelleher and Collins renders obvious the additional requirement of Element 3[a].

220. The combined teachings of Kelleher and Collins disclose that *the first GPU chiplet* (GPU chip 220 of Kelleher) *comprises a first PHY region* (a physical layer (PHY) input/output (I/O) region) *including conductor structures* (solder balls of a ball grid array) *for chiplet-to-chiplet communications* (communications between GPU chips 220 and 240 of Kelleher).

221. Before explaining the relevant disclosures of Kelleher and Collins, I am including the following brief overview of how the '803 patent describes a "PHY regions."

222. As shown in Figure 2 of the '803 patent, the first GPU chiplet (in purple) includes a first PHY region 202 (in blue) with conductor structures (in blue)

for communicating with the second chiplet (in gold) via the passive crosslink (conductive pathways of the passive crosslink are shown in green).⁴

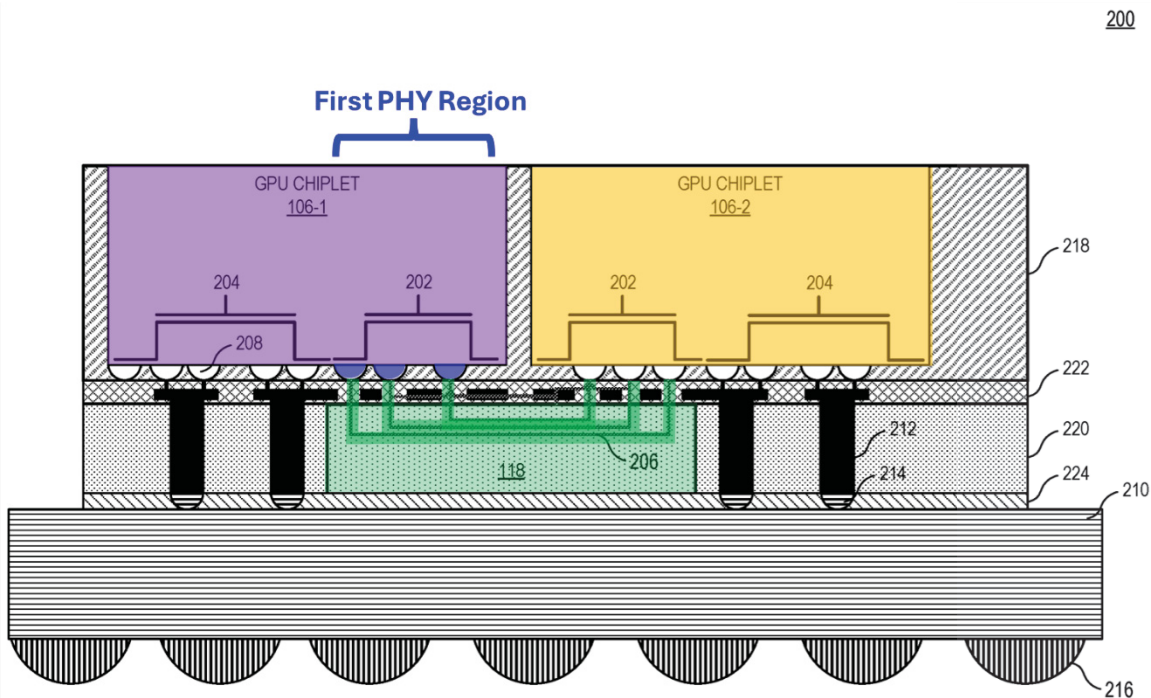


FIG. 2

EX1001-'803, Figure 2.

223. The '803 patent explains:

In various embodiments, **each GPU chiplet 106 is constructed with a physical device (PHY) region 202, which has various internal and external conductor structures dedicated to the transmission of chiplet-to-**

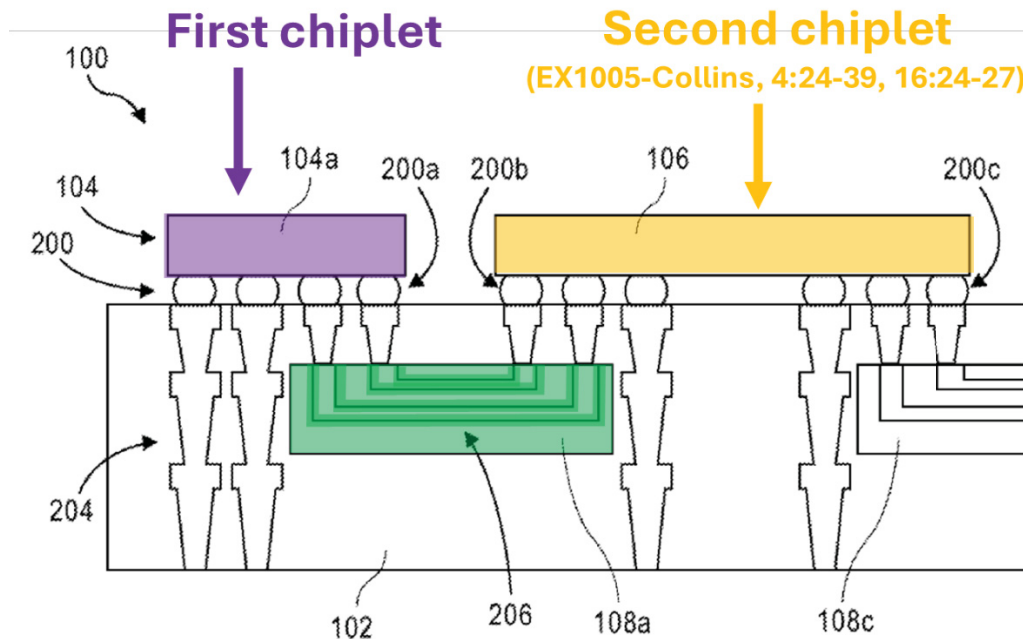
⁴ A person of ordinary skill in the art would understand that the symbols corresponding to 202 and 204 in Figure 2 of the '803 patent denote a PHY region and a non PHY region, respectively. Although these symbols may look like transistor symbols, a person of ordinary skill in the art would understand that the symbols conceptually represent electrical connections delineating the PHY and non-PHY regions. EX1001-'803, 3:57-4:2; EX1004-Kelleher, 3:64-4:2.

chiplet signals, and a non-PHY region 204, which has conductor structures that are tailored more to the conveyance of power and ground and/or chiplet-to-circuit board signals.

As previously noted, the GPU chiplets 106 are communicably coupled by way of the passive crosslink 118. In various embodiments, the passive crosslink 118 is an interconnect chip constructed of silicon, germanium or other semiconductor materials and may be bulk semiconductor, semiconductor on insulator or other designs. **The passive crosslink 118 includes a plurality of internal conductor traces**, which may be on a single level or multiple levels as desired. Three of the traces are illustrated in FIG. 2 and labeled collectively as traces 206. **The traces 206 interface electrically with conductor structures of the PHY regions 202 of the GPU chiplets 106 by way of conducting pathways.**

EX1001-'803, 3:57-4:21. The '803 patent further notes that the external conductive structures "may include solder bump[s], micro bump[s], and the like." EX1001-'803, 4:22-35.

224. Modified Figure 2 of Collins shows that the first chiplet (shown in purple) likewise includes a first PHY region (PHY I/O) for inputting signals from, and outputting signals to, a second PHY region (PHY I/O) of the second chiplet (shown in gold) via the passive crosslink (passive interconnect/EMIB (shown in green)).



EX1005-Collins, Modified Figure 2 (replacing memory device 106 with another chiplet according to Collins’s disclosure in 4:24-39 and 16:24-27).

225. Collins explains that each chiplet includes a physical layer (PHY) input/output (I/O) region. For example, Collins refers to an “active die size memory physical layer (PHY) input/output (I/O).” EX1005 Collins, 1:20-24. Collins also explains:

The memory device 106 [or another chiplet according to Collins’s disclosure in 4:24-39 and 16:24-27] includes a PHY I/O. ... For centralized PHY I/O, a single PHY I/O may be utilized by all the memory channels of the memory device 106 [or another chiplet according to Collins’s disclosure in 4:24-39 and 16:24-27]. For distributed PHY I/O, the memory device 106 [or another chiplet according to Collins’s disclosure in 4:24-39 and 16:24-27] includes

a plurality of PHY I/Os, where each of the PHY I/Os may be utilized by one or more of the memory channels... For example, ... the PHY I/O are separated into I/O quadrants, where the quadrants can match a floor plan of the memory device 106 [or another chiplet according to Collins's disclosure in 4:24-39 and 16:24-27].

EX1005 Collins, 3:34-48.

226. Collins teaches that its “interconnect structure further couples a physical layer (PHY) of the first chiplet to a PHY of the second chiplet.” EX1005-Collins, 16:24-27.

227. Collins further teaches that “each of the EMIBs can couple each of the chiplets 104 to a centralized PHY I/O or to different distributed PHY I/Os [of the second chiplet according to Collins's disclosure in 4:24-39 and 16:24-27]. For example, the first EMIB 108a can couple the first chiplet 104a to a first distributed PHY I/O [of the second chiplet according to Collins's disclosure in 4:24-39 and 16:24-27], [and] the second EMIB 108b can couple the second chiplet 104b to a second distributed PHY I/O [of the second chiplet according to Collins's disclosure in 4:24-39 and 16:24-27]....” EX1005 Collins, 3:65-4:23.

228. In another embodiment, “[e]ach of the chiplets 502 have a PHY I/O that is one quarter of the PHY I/O of the memory device 504. For example, each of the PHY I/O of the chiplets 502 may be the same size of PHY I/O associated with each of the channel clusters [of the second chiplet according to Collins's disclosure in

4:24-39 and 16:24-27]. Further, the chiplets 502 may be coupled to each other.”

EX1005 Collins, 8:6-11; *see also* EX1005-Collins, 11:4-18, 12:32-46.

229. Figure 2 of Collins further illustrates that the PHY I/Os of the first and second chiplets have external conductor structures—interconnect elements 200a and 200b. Collins states that “[t]he interconnect elements 200 couple the first chiplet 104a, the third chiplet 104c, and the memory device 106 [or second chiplet according to Collins’s disclosure in 4:24-39 and 16:24-27] to conductive elements 204 of the substrate assembly 102.... In the illustrated embodiment, the interconnect elements 200 are illustrated as a ball grid array.” EX1005-Collins, 4:46-59. A person of ordinary skill in the art would have understood that the balls in a ball grid array are the same as the solder bumps shown in Figure 2 of the ’803 patent. EX1001-’803, 4:25-27. For example, an article by Joel Mearing and Bill Goers entitled “An Overview of Manufacturing BGA Technology” equates balls in a ball grid array with solder bumps, explaining that Ball Grid Array (“BGA”) “components, by design, have all soldered connections under the component body.” EX1036-Mearig-&-Goers, 434, 436.

230. Therefore, it is my opinion that the combination of Kelleher and Collins teaches that the first GPU chiplet (GPU chip 220 of Kelleher) includes a first PHY region (PHY I/O region) with conductor structures—specifically, solder balls of a

ball grid array—dedicated to chiplet-to-chiplet communications between the GPU chips 220 and 240 of Kelleher.

(b)Element 3[b]: and further wherein the second GPU chiplet comprises a second PHY region including conductor structures for chiplet-to-chiplet communications.

231. Kelleher and Collins renders obvious the additional requirement of Element 3[b].

232. For the same reasons that I have explained with respect to element 3[a], which I incorporate herein by reference, the combined teachings of Kelleher and Collins disclose that *the second GPU chiplet* (GPU chip 240 of Kelleher) *comprises a second PHY region* (PHY I/O) *including conductor structures* (solder balls of a ball grid array) *for chiplet-to-chiplet communications* between the GPU chips 220 and 240 of Kelleher.

4. Dependent Claim 4

233. Claim 4 depends from claim 1. Therefore, I incorporate by reference my discussion of claim 1.

234. Kelleher and Collins renders obvious the additional requirements of Claim 4.

(a)Element 4[a]: The system of claim 3, further comprising: a third GPU chiplet communicably coupled to the first GPU chiplet via the passive crosslink,

235. Kelleher and Collins renders obvious the additional requirement of Element 4[a].

236. The combined teachings of Kelleher and Collins disclose *a third GPU chiplet* (GPU chip 240 of Kelleher) *communicably coupled to the first GPU chiplet* (GPU chip 220 of Kelleher) via *the passive crosslink* (interconnect structure/EMIB of Collins).

237. For example, while Figure 2 of Kelleher depicts two chiplets—master and slave GPU chips—connected via an interlink, Kelleher further teaches that the master GPU chip may be communicably coupled with multiple slave GPU chips.

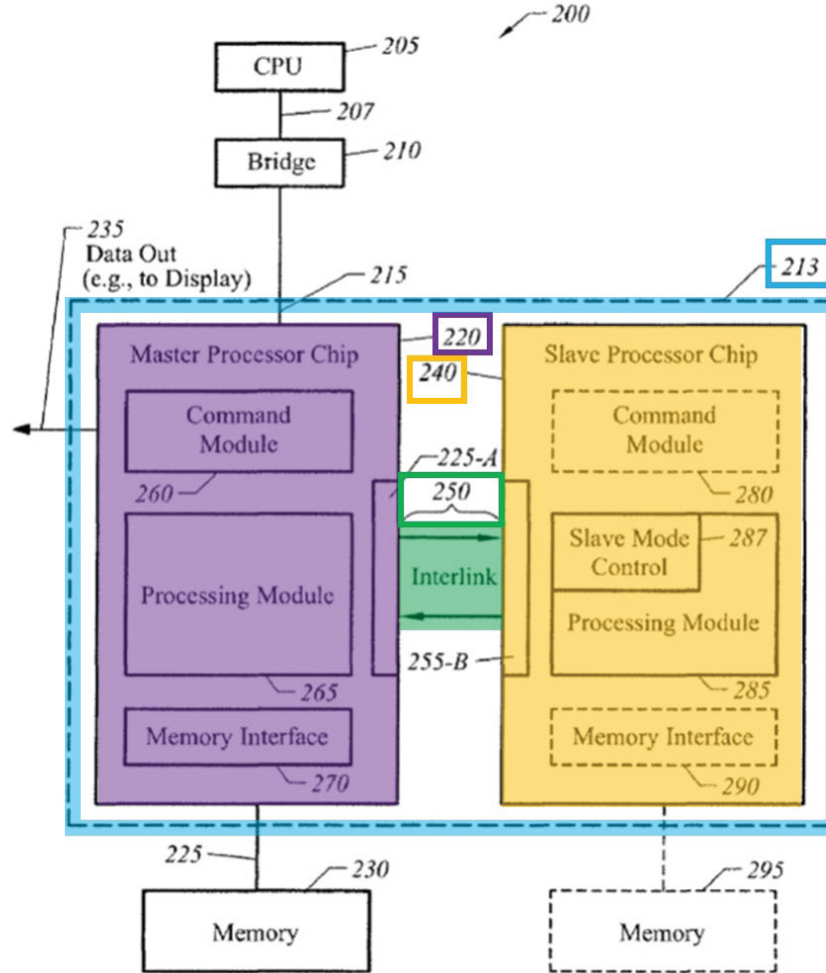


FIG. 2

EX1004-Kelleher, Figure 2.

238. For example, Kelleher states:

While an exemplary embodiment of the present invention has been described as having a single master GPU chip and a single slave GPU chip, it will be understood that numerous extensions are contemplated. **For example, in some applications it may be desirable to utilize two or more slave GPUs for each master GPU.** Moreover, while examples have been described in which the slave GPU performs a processing operation in parallel for one

of the stages of the master GPU, more generally one or more operable stages within slave GPU may be used to extend the performance of corresponding stages within the master GPU.

EX1004 Kelleher, 8:8-18; *see also* EX1004 Kelleher, 10:45-59.

239. In such an arrangement, a person of ordinary skill in the art would have understood that both the first and second slave GPU chips would be communicably coupled to the master GPU chip via the interlink. I also refer to my analysis of Element 1[a], which I incorporate herein by reference.

240. Collins also discloses that its interconnect structure/EMIB may be used to connect three chiplets. For example, Collins explains:

Further, the **EMIBs 108 may couple two or more of the chiplets 104 together** in some embodiments. The chiplets 104 coupled with each other via the EMIBs 108 may utilize the EMIBs 108 for communication between the chiplets 104, the communication including clock distribution communications, register control communications, interface testing function communications, or some combination thereof.

EX1005 Collins, 4:32-39.

241. Therefore, it is my opinion that the combined teachings of Kelleher and Collins discloses a third GPU chiplet (second slave GPU chip of Kelleher) communicably coupled to the first GPU chiplet (master GPU chip 220 of Kelleher) via the passive crosslink (interconnect structure/EMIB of Collins). I also refer to my analysis of Element 1[c], which I incorporate herein by reference.

(b)Element 4[b]: wherein the third GPU chiplet comprises a third PHY region including conductor structures for chiplet-to-chiplet communications.

242. Kelleher and Collins renders obvious the additional requirement of Element 4[b].

243. The combined teachings of Kelleher and Collins disclose *that the third GPU chiplet* (second slave GPU chip of Kelleher) *comprises a third PHY region* (PHY I/O) *including conductor structures* (solder balls of a ball grid array) *for chiplet-to-chiplet communications* (communications between the master and second slave GPU chips of Kelleher).

244. As I explained with respect to Element 4[a], incorporated herein by reference, the combined teachings of Kelleher and Collins discloses that the master and second slave GPU chips of Kelleher engages in chiplet-to-chiplet communications via the passive crosslink—implemented as Collins’s interconnect structure/EMIB.

245. Namely, as noted in Element 4[a], Collins explains:

Further, the **EMIBs 108 may couple two or more of the chiplets 104 together** in some embodiments. The chiplets 104 coupled with each other via the EMIBs 108 may utilize the EMIBs 108 for communication between the chiplets 104, the communication including clock distribution communications, register control communications, interface testing function communications, or some combination thereof.

EX1005-Collins, 4:32-39.

246. As discussed for Elements 3[a] to 3[b], the combined teachings of Kelleher and Collins disclose that the master and first slave GPU chips of Kelleher each includes a first PHY region and a second PHY region (PHY I/Os) with conductor structures—specifically, solder balls of a ball grid array—dedicated to chiplet-to-chiplet communications. For the same reasons, the combined teachings of Kelleher and Collins likewise discloses that the second slave GPU chip of Kelleher includes a third PHY region (PHY I/O) with conductor structures—again, solder balls of a ball grid array—dedicated to chiplet-to-chiplet communications.

5. Dependent Claim 7

(a) The system of claim 1, wherein the passive crosslink communicably couples all GPU chiplets in the GPU chiplet array.

247. Claim 7 depends from claim 1. Therefore, I incorporate by reference my discussion of claim 1.

248. Kelleher and Collins renders obvious the additional requirement of claim 7.

249. The combined teachings of Kelleher and Collins disclose that *the passive crosslink*—implemented as the interconnect structure/EMIB of Collins—*communicably couples all GPU chiplets* (the master and slave GPU chips of Kelleher) *in the GPU chiplet array* (the multi-chip module 213 of Kelleher).

250. As I previously explained with respect to Element 1[c], incorporated herein by reference, the interconnect structure/EMIB of Collins provides the communication link between the master and slave GPU chips of Kelleher. Further, as I previously explained with respect to Element 1[a], incorporated herein by reference, in the embodiment shown in Figure 2 of Kelleher, the master and slave GPU chips are all of the GPU chiplets contained within the multi-chip module 213.

6. Dependent Claim 10

(a) The system of claim 1, further comprising: a plurality of conductive pillars coupling a circuit board substrate to a first non-PHY region of the first GPU chiplet and a second non-PHY region of the second GPU chiplet.

251. Claim 10 depends from claim 1. Therefore, I incorporate by reference my discussion of claim 1.

252. Kelleher and Collins renders obvious the additional requirement of claim 10.

253. The combined teachings of Kelleher and Collins disclose *a plurality of conductive pillars* (conductive elements of Collins) *coupling a circuit board substrate* (printed circuit board (PCB) of Collins/circuit board of Kelleher) *to a first non-PHY region of the first GPU chiplet* (master GPU chip of Kelleher) *and a second non-PHY region of the second GPU chiplet* (slave GPU chip of Kelleher).

254. Before explaining the relevant disclosures of Kelleher and Collins, I am including the following brief overview of the relevant disclosure of the '803 patent.

255. Figure 2 of the '803 patent shows that conductive pillars (in brown) couple a circuit board substrate (in red) to the non-PHY regions 204 (indicated by a blue bracket) of the first GPU chiplet (in purple) and the second GPU chiplet (in gold).

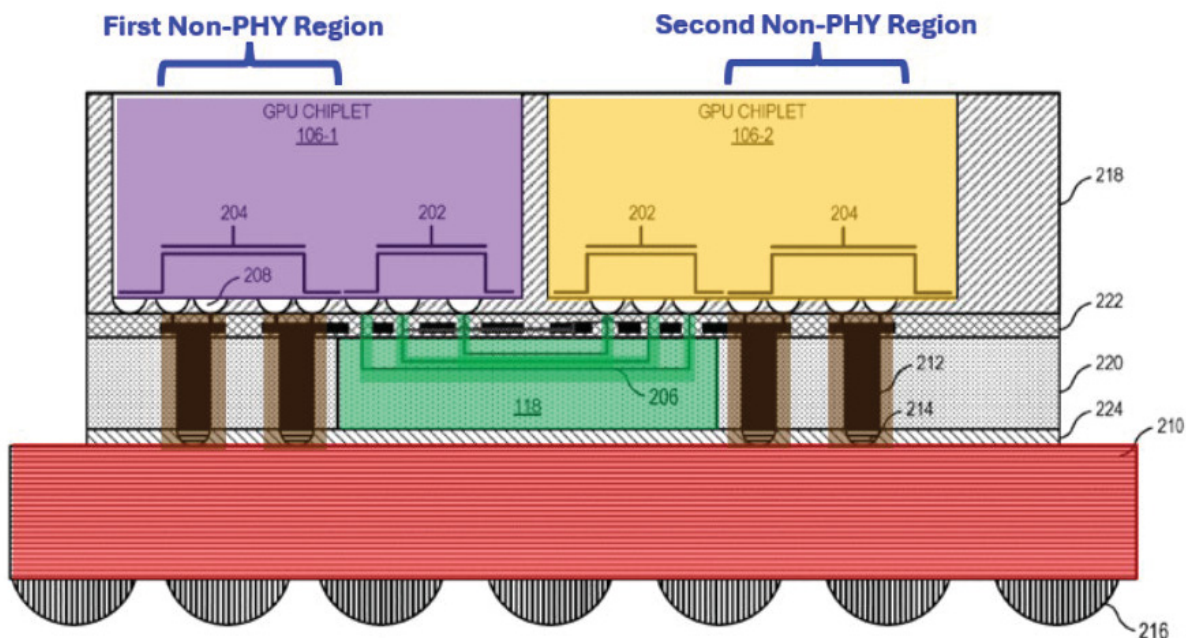


FIG. 2

EX1001-'803, Figure 2.

256. The '803 patent explains:

In various embodiments, **each GPU chiplet 106 is constructed** with a physical device (PHY) region 202, which has various internal and external conductor structures dedicated to the transmission of chiplet-to-

chiplet signals, and **a non-PHY region 204, which has conductor structures that are tailored more to the conveyance of power and ground and/or chiplet-to-circuit board signals....**

The non-PHY regions 204 of the GPU chiplets 106 interface electrically with a circuit board 210 (or any other substrate) by way of a plurality of conductive pillars 212. Each conductive pillar 212 is connected electrically to the GPU chiplets 106 by way of a solder interconnect 208, which may include a solder bump, micro bump, and the like. In various embodiments, the circuit board 210 interfaces electrically with other electrical structure, such as another circuit board or other structures, by way of a plurality of interconnect structures 214 (e.g., solder balls and the like). However, those skilled in the art will appreciate that various types of interconnect structures such as pins, land grid array structures, other interconnects, and the like may be used without departing from the scope of this disclosure.

EX1001-'803, 3:63-4:36.

257. As I explained for Elements 3[a] and 3[b], incorporated herein by reference, the first and second chiplets of Collins have a PHY regions for communicating with each other via the interconnect structure/EMIB. EX1005-Collins, 4:24-29, 16:24-27. As shown in Modified Figure 2 of Collins below, the first chiplet (in purple) and second chiplet (in gold) also have non-PHY regions configured to interface electrically with a printed circuit board (PCB) (not shown in Figure 2 but depicted in red) via conductive pillars (conductive elements 204 (in brown) passing through a substrate assembly 102.

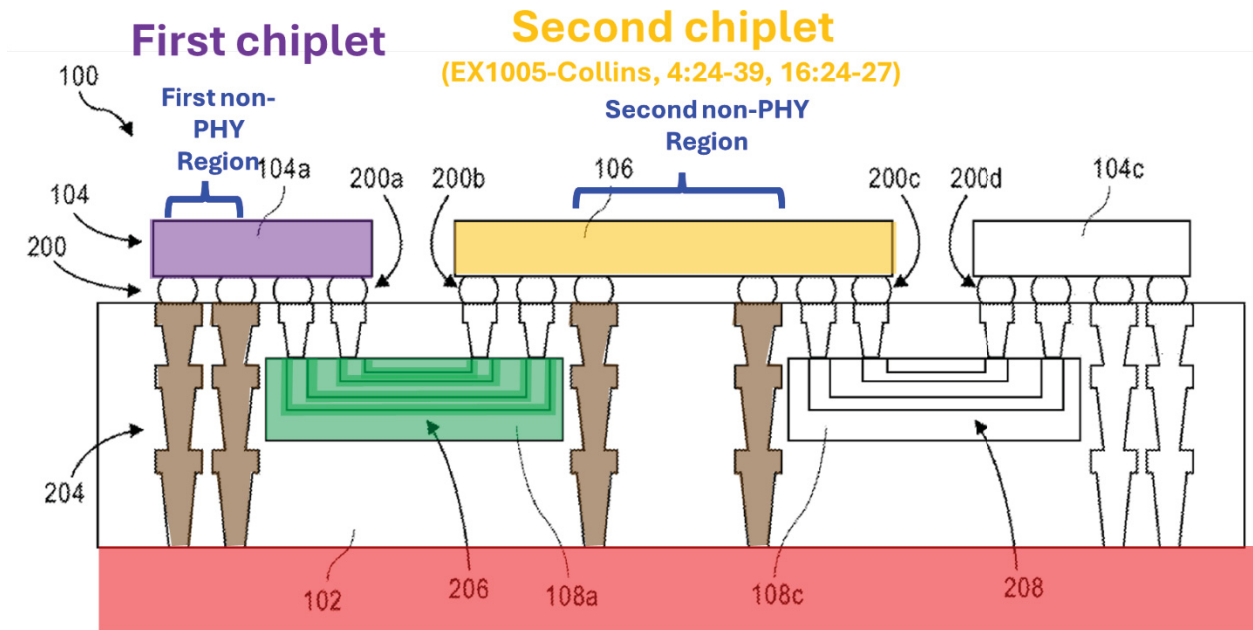


FIG. 2

EX1005-Collins, Modified Figure 2 (replacing memory device 106 with another chiplet according to Collins's disclosure in 4:24-39 and 16:24-27).

258. In Figure 10 of Collins shown below, a semiconductor package 1004 includes a chiplet 1010 and a memory device 1012 (or another chiplet according to Collins's disclosure in 4:24-39 and 16:24-27) that are connected to a substrate assembly 1006 with an EMIB 1008.

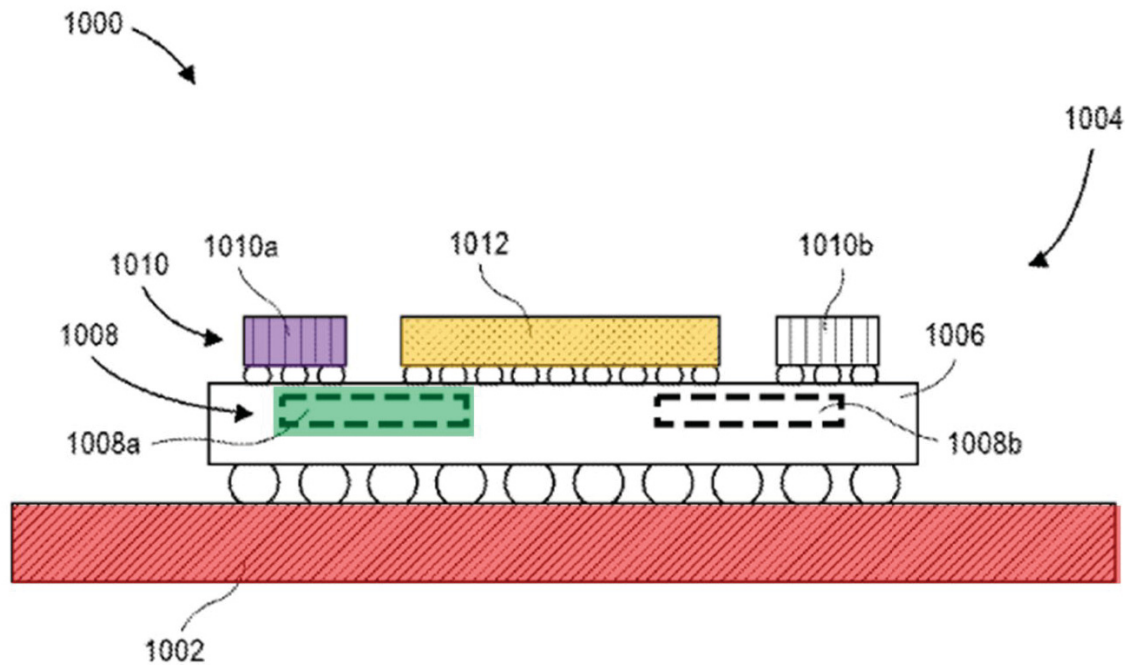


FIG. 10

EX1005-Collins, Figure 10.

259. Collins explains that:

The semiconductor package 1004 includes a substrate assembly 1006. The substrate assembly 1006 includes one or more of the features of the substrate assembly 102 (FIG. 1) and/or the substrate assembly 302 (FIG. 3). The substrate assembly 1006 is mounted and coupled to the PCB 1002.

The semiconductor package 1004 further includes an interconnect structure 1008. In the illustrated embodiment, the interconnect structure 1008 is illustrated as a first EMIB 1008a and a second EMIB 1008b embedded within the substrate assembly 1006.

EX1005-Collins, 13:26-35.

260. Collins further explains:

The interconnect structure 1008 provides the coupling among the chiplets 1010 and the memory device 1012 [or second chiplet according to Collins's disclosure in 4:24-39 and 16:24-27] according to the couplings among the chiplets and the memory devices described throughout this disclosure. For example, the interconnect structure 1008 may couple the chiplets 1010 with the memory device 1012, the chiplets 1010 with each other, the substrate assembly 1006 with the chiplets 1010 and the memory device 1012, or some combination thereof.

EX1005-Collins, 14:8-16.

261. Figure 10 of Collins also illustrates that the substrate assembly 1006 (shown in white), which is similar to substrate assembly 204 in Figure 2, is mounted to a printed circuit board 1002.

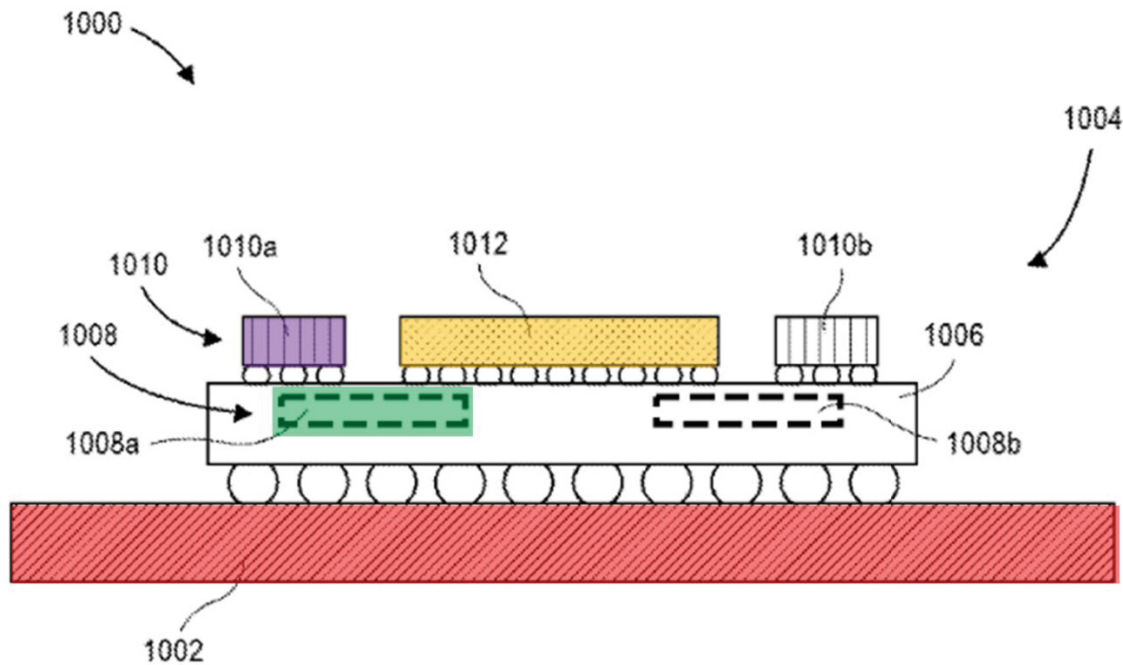


FIG. 10

EX1005-Collins, Figure 10. Further, Collins states:

The semiconductor package 1004 is mounted to a surface of the PCB 1002. The semiconductor package 1004 may implement one or more of the features the semiconductor packages (such as the semiconductor package 100 (FIG. 1) and/or the semiconductor package 300 (FIG. 3)) and/or the representations of semiconductor package arrangements (such as the representation 500 (FIG. 5), the representation 600 (FIG. 6), the representation 700 (FIG. 7), the representation 800 (FIG. 8), and/or the representation 900 (FIG. 9)), as described herein.

The semiconductor package 1004 includes a substrate assembly 1006. The substrate assembly 1006 includes one or more of the features of the substrate assembly 102 (FIG. 1) and/or the substrate assembly 302

(FIG. 3). The substrate assembly 1006 is mounted and coupled to the PCB 1002.

The semiconductor package 1004 further includes an interconnect structure 1008. **In the illustrated embodiment, the interconnect structure 1008 is illustrated as a first EMIB 1008a and a second EMIB 1008b embedded within the substrate assembly 1006.**

EX1005-Collins, 13:11-35.

262. A person of ordinary skill in the art would have understood that, Figure 10 of Collins teaches the non-PHY regions of the first chiplet 1010a (shown in purple) and the memory device 1012 [or second chiplet (shown in gold) according to Collins's disclosure in 4:24-39 and 16:24-27] are electrically coupled to the printed circuit board 1002 (shown in red) via conductive pillars (conductive elements 204) (shown in brown), as taught by the embodiment in Figure 2 of Collins, with the pillars passing through substrate assembly 1006. For example, Collins explains that in Figure 10, "[t]he semiconductor package 1004 is mounted to a surface of the PCB 1002. The semiconductor package 1004 may implement ... the features [of] the semiconductor packages (such as the semiconductor package 100 (FIG. 1)..." EX1005-Collins, 13:15-25. Collins further notes that "FIG. 2 illustrates a cross-sectional, side view of the example semiconductor package of FIG. 1."

EX1005-Collins, 1:35-37.

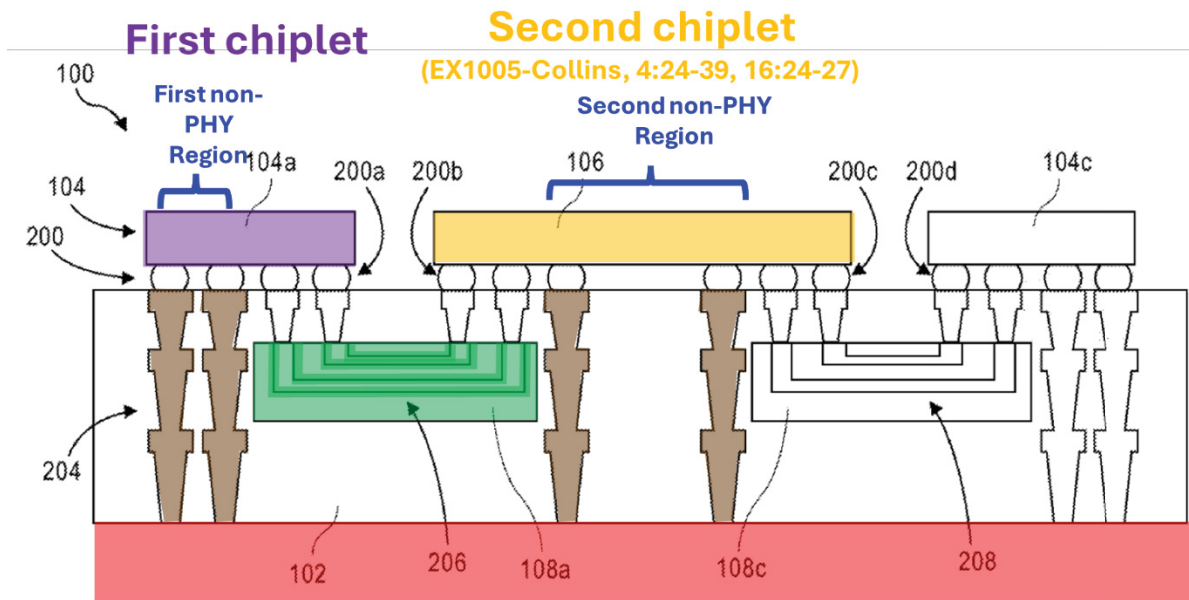


FIG. 2

EX1005-Collins, Figure 2.

263. Figure 2 of Kelleher likewise illustrates that the master GPU chip (shown in purple) and the slave GPU chip (shown in gold) are coupled to a circuit board (shown in red) via a substrate (package) (shown in white).

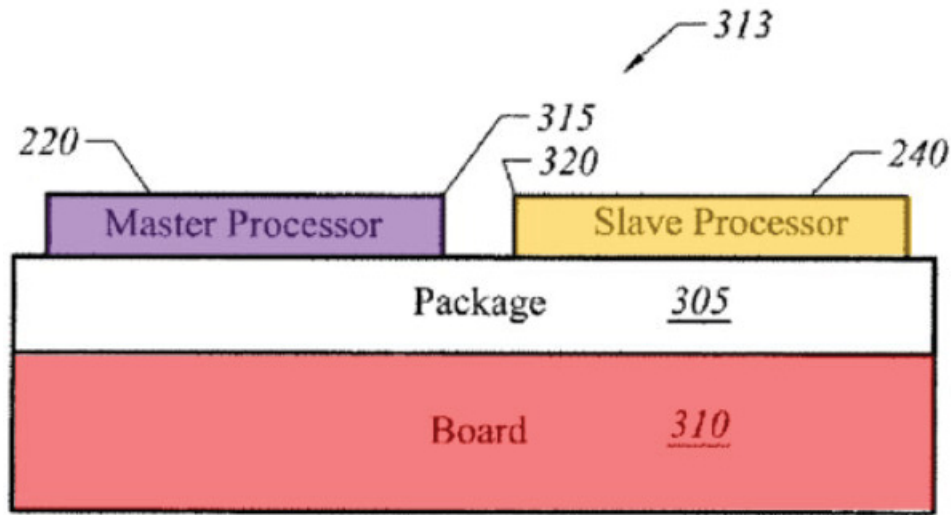


FIG. 3

E1005-Collins, Figure 3.

264. Kelleher explains that “GPU chip 240 and GPU chip 220 are packaged on top of a common multi-chip package 305 that is then placed on a board 310.”

EX1004-Kelleher, 5:39-44.

265. A person of ordinary skill in the art would have known that Kelleher’s master and slave GPU chips 220 and 240 (shown in purple and gold) receive power from the circuit board via the package to operate. As a result, electrical connections between the chips and the board through the package must exist. EX1004-Kelleher, 3:35-50, 5:39-57, 5:66-23, 6:24-52. As I have explained throughout this Ground, Collins teaches that forming conductive elements through the package/substrate is a standard way of providing such electrical connections to chiplets via a package/substrate, as shown in Figure 2 of Collins.

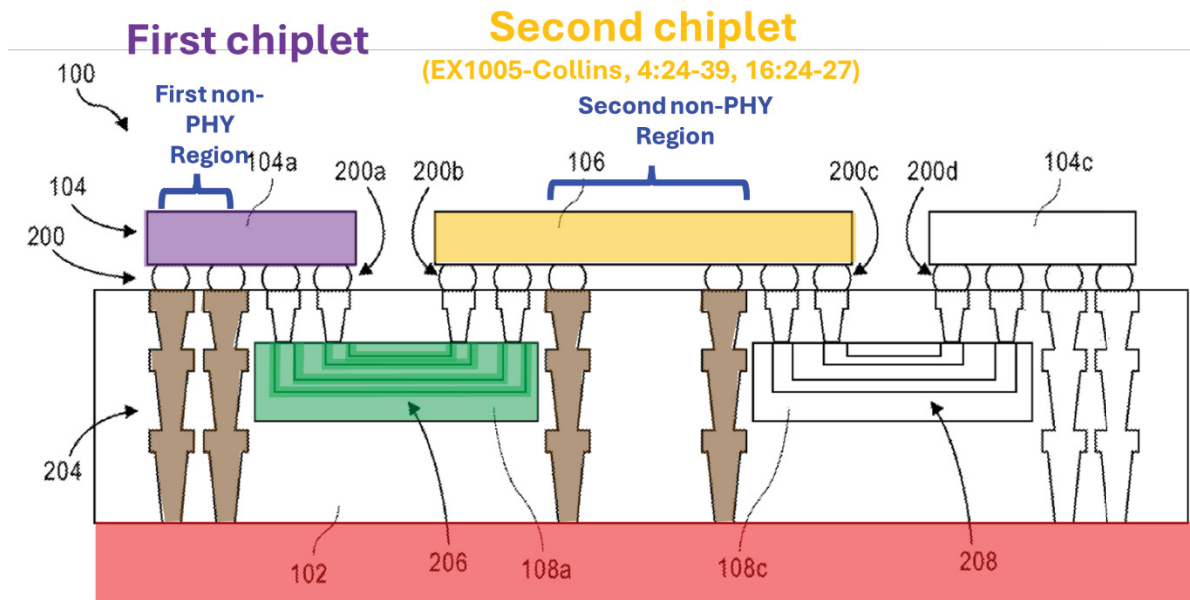


FIG. 2

EX1005-Collins, Figure 2.

266. It is therefore my opinion that, as shown in the annotated figure below, the combined teachings of Kelleher and Collins discloses using conductive pillars (conductive elements of Collins shown in brown) to couple a circuit board substrate (board of Kelleher shown in red) to a first non-PHY region (in blue) of the first GPU chiplet (master GPU chip of Kelleher shown in purple) and a second non-PHY region (in blue) of the second GPU chiplet (slave GPU chip of Kelleher shown in gold).

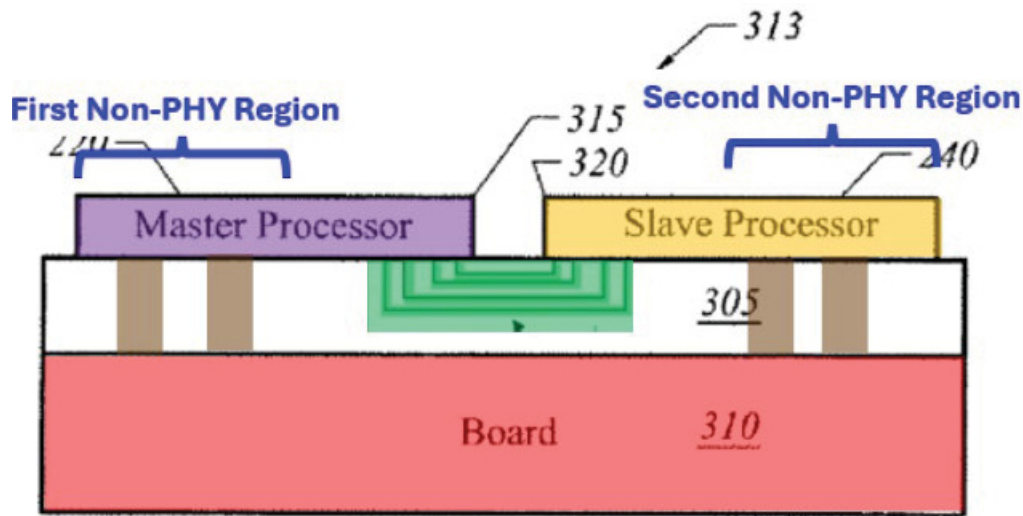


FIG. 3

EX1004-Kelleher, Modified Figure 3 (incorporating both Kelleher and Collins's teachings).

B. GROUND II: Claims 8 and 9 Would Have Been Obvious In View of the Combined Teachings of Kelleher, Collins, Koker, and Sistla

1. Dependent Claim 8

267. Claim 8 depends from claim 1. Therefore, I incorporate by reference my discussion of claim 1.

268. The combined teachings of Kelleher, Collins, Koker, and Sistla teach the additional requirements of Claim 8.

**(a) Element 8[a]: The system of claim 1, further comprising:
a first cache memory hierarchy at the first GPU chiplet,**

269. The combined teachings of Kelleher, Collins, Koker, and Sistla render obvious the additional requirement of Element 8[a].

270. The combined teachings of Kelleher, Collins, Koker, and Sistla disclose *a first cache memory hierarchy* (L1 cache/shared memory 373 and L2 cache 375 of Koker (as modified by the teachings of Sistla)) *at the first GPU chiplet* (master GPU chip 220 of Kelleher, which corresponds to multi-core group 365A of Koker).

271. Before explaining the relevant disclosures of Kelleher, Collins, Koker, and Sistla, I am including the following brief overview of how the '803 patent describes its cache memory hierarchies.

272. In Figure 3 below, the '803 patent illustrates that the first and second chiplets 106-1 and 106-2 include a “cache memory hierarchy,” which is a tiered arrangement of memories having multiple levels, such as a Level 1 (L1) cache (in brown), a Level 2 (L2) cache (in orange), and a Level 3 (L3) cache (in teal).



FIG. 3

EX1001-'803, Figure 3. Namely, the '803 patent explains:

FIG. 3 is a block diagram illustrating a cache hierarchy of GPU chiplets coupled by a passive crosslink in accordance with some embodiments. The view 300 provides a hierarchical view of GPU chiplets 106-1, 106-2, and passive crosslink 118 of FIG. 1. Each of the GPU chiplets 106-1, 106-2 include a plurality of workgroup processors 302 (WGP) and a plurality of fixed function blocks 304 (GFX) that communicate with a given channel's L1 cache memory 306. Each GPU chiplet 106 also includes a plurality of L2 cache memory 308 banks that are individually accessible, a plurality of L3 cache memory 310 channels, and a plurality of memory PHY 312 (denoted as graphics double data rate (GDDR) in FIG. 3 for indicating connection to GDDR memory) channels mapped to the L3 channels. The L2 level of cache is coherent within a single chiplet and the L3 level (or other last level) of cache is unified and coherent across all of the GPU chiplets 106.

A graphics data fabric 314 (GDF) of each GPU chiplet 106 connects all of the L1 cache memories 306 to each of the channels of the L2 cache memory 308, thereby allowing each of the workgroup processors 302 and fixed function blocks 304 to access data stored in any bank of the L2 cache memory 308. Each GPU chiplet 106 also includes a scalable data fabric 316 (SDF) (also known as a SOC memory fabric) that routes across the graphics core (GC) and system on chip (SOC) IP cores to the passive crosslink 118. The GC includes CUs/WGPs, fixed function graphics blocks, caches above L3, and the like. Portions of the GPU used for traditional graphics and compute (i.e., the GC) are differentiable from other portions of the GPU used for handling auxiliary GPU functionality such as video decode, display output, and various system supporting structures that are contained on the same die. The passive crosslink 118 routes to both local L3 cache memory 310 of a chiplet (e.g., of GPU chiplet 106-1) and the L3 cache memory 310 of all

other external GPU chiplets (e.g., GPU chiplet 106-2 in FIG. 3). In this manner, a memory address request is routed to the appropriate lanes on the passive crosslink 118 to access L3 cache memory 310 locally or at L3 cache memory 310 of a different GPU chiplet 106 (as discussed in more detail with respect to FIG. 5).

EX1001-'803, 5:1-42. This hierarchy provides progressively larger and slower memory levels to optimize data access speed and efficiency within each chiplet.

273. Similar to the '803 patent, Koker, in Figure 3C, discloses “multi-core groups” or “MCGs”, where each MCG (group A shown in purple and group B shown in gold) is associated with a cache memory hierarchy with multiple levels of cache—specifically, a Level 1 (L1) cache (in orange) and a Level 2 (L2) cache (in teal).

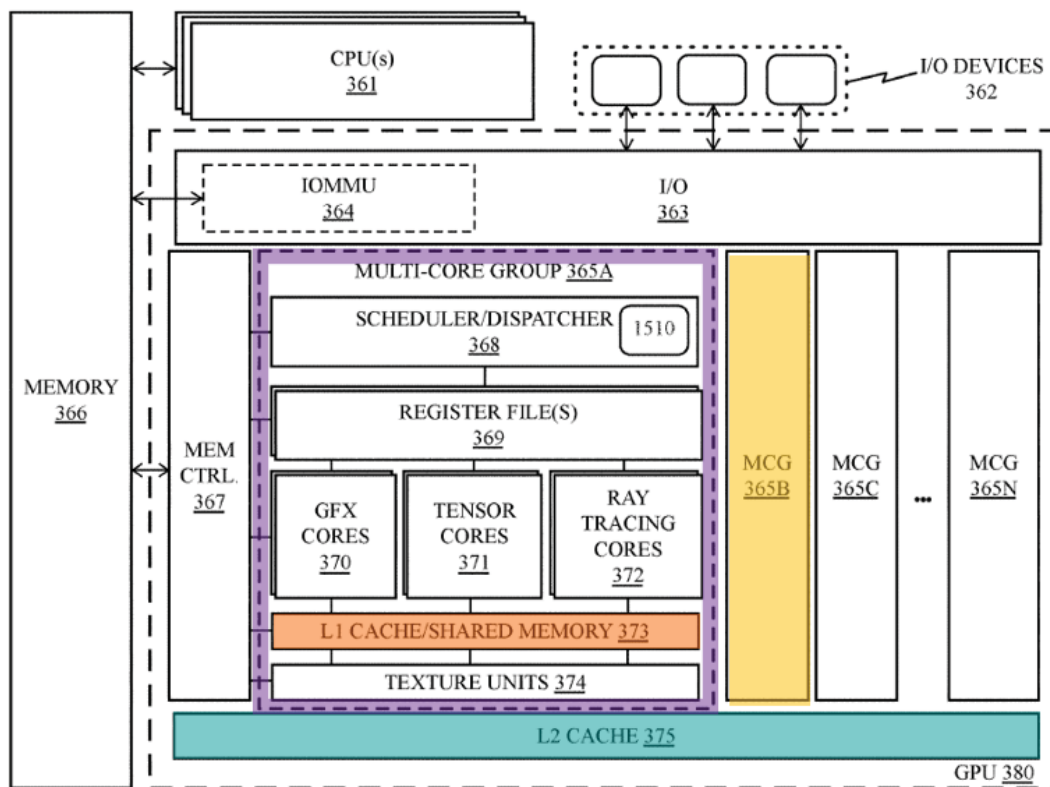


FIG. 3C

EX1031-Koker, Figure 3C.

274. Koker explains:

FIG. 3C illustrates a **graphics processing unit (GPU) 380** which includes dedicated sets of **graphics processing resources arranged into multi-core groups 365A-N**. While the details of only a single multi-core group 365A are provided, **it will be appreciated that the other multi-core groups 365B-365N may be equipped with the same or similar sets of graphics processing resources.**

EX1031-Koker, 13:36-42.

275. Koker further explains:

One or more combined level 1 (L1) caches and shared memory units 373 store graphics data such as texture data, vertex data, pixel data, ray data, bounding volume data, etc., locally within each multi-core group 365A. One or more texture units 374 can also be used to perform texturing operations, such as texture mapping and sampling. **A Level 2 (L2) cache 375 shared by all or a subset of the multi-core groups 365A-365N stores graphics data and/or instructions** for multiple concurrent graphics threads. As illustrated, **the L2 cache 375 may be shared across a plurality of multi-core groups 365A-365N.** One or more memory controllers 367 couple the GPU 380 to a memory 366 which may be a system memory (e.g., DRAM) and/or a dedicated graphics memory (e.g., GDDR6 memory).

EX1031-Koker, 13:55-14:2.

276. Additional disclosures in Koker confirm this cache hierarchy structure:

The caches 462A-462D may comprise level 1 (L1) and level 2 (L2) caches. In addition, one or more shared caches 456 may be included in the caching hierarchy

and shared by sets of the cores 460A-460D. For example, one embodiment of the processor 407 includes 24 cores, each with its own L1 cache, twelve shared L2 caches, and twelve shared L3 caches. In this embodiment, one of the L2 and L3 caches are shared by two adjacent cores.

EX1031-Koker, 17:36-44.

The internal cache units 1504A-1504N and shared cache units 1506 represent a cache memory hierarchy within the processor 1500. The cache memory hierarchy may include at least one level of instruction and data cache within each processor core and one or more levels of shared mid-level cache, such as a Level 2 (L2), Level 3 (L3), Level 4 (L4), or other levels of cache, where the highest level of cache before external memory is classified as the LLC. In some embodiments, cache coherency logic maintains coherency between the various cache units 1506 and 1504A-1504N.

EX1031-Koker, 40:29-38.

277. In the '803 patent, the L3 cache (in teal) is a “last level cache” (“LLC”), and different portions of the LLC are contained within the first and second chiplets 106-1 and 106-2, respectively, as shown in Figure 3 below.

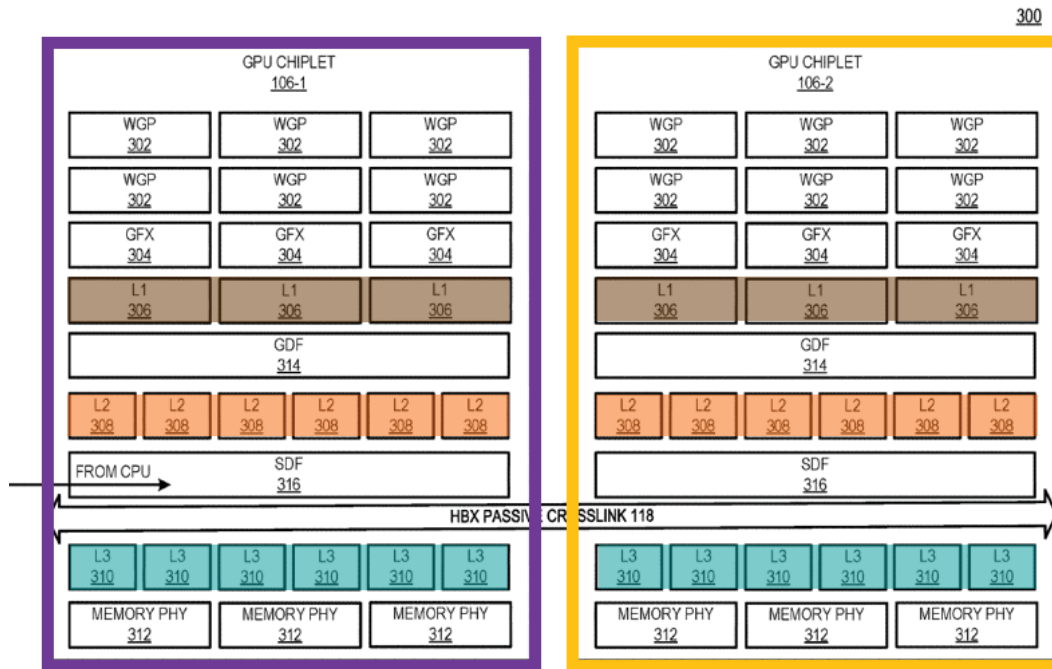


FIG. 3

EX1001-'803, Figure 3, 5:14-18 (“The L2 level of cache is coherent within a single chiplet and the L3 level (or other last level) of cache is unified and coherent across all of the GPU chiplets 106.”)

278. As shown in Figure 3C of Koker (below), the L2 cache (in teal) is a last level cache (LLC).

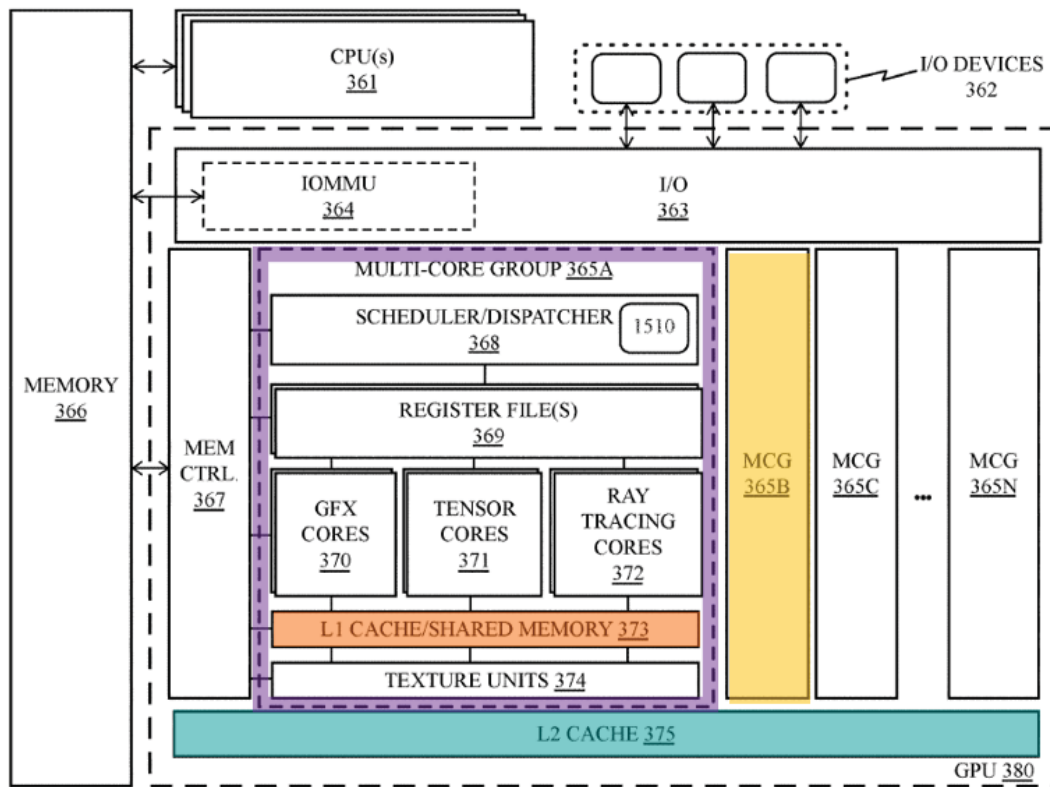


FIG. 3C

Namely, the L2 cache (in teal) is the highest level of cache before the external system (or graphics) memory 366, and a person of ordinary skill in the art would have understood that this cache is a last level cache (“LLC”).

279. Specifically, Koker explains:

The cache memory hierarchy may include at least one level of instruction and data cache within each processor core and one or more levels of shared mid-level cache, such as a Level 2 (L2), Level 3 (L3), Level 4 (L4), or other levels of cache, where **the highest level of cache before external memory is classified as the LLC**. In some embodiments, cache coherency logic maintains coherency between the various cache units 1506 and 1504A-1504N.

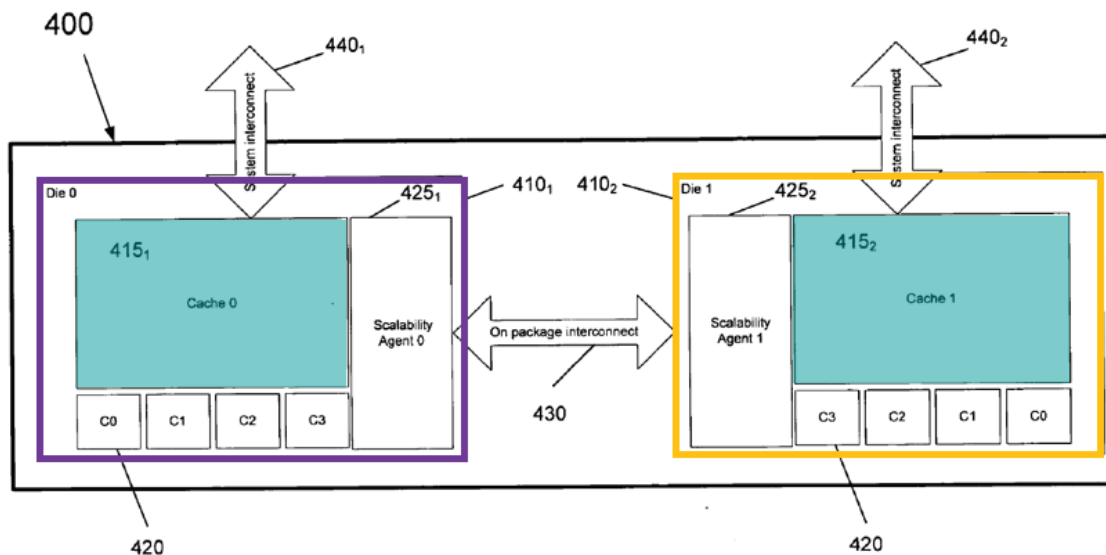
EX1031-Koker, 40:29-42.

As illustrated, the L2 cache 375 may be shared across a plurality of multi-core groups 365A-365N. One or more memory controllers 367 couple the GPU 380 to a **memory 366 which may be a system memory (e.g., DRAM) and/or a dedicated graphics memory (e.g., GDDR6 memory).**

EX1031-Koker, 13:64-14:2.

280. Although Figure 3C of Koker shows that the L2 cache is external to the “multi-core groups,” a person of ordinary skill in the art would have known to divide up Koker’s last level L2 cache and include different portions of the L2 cache within the respective multi-core groups 365A to 365N.

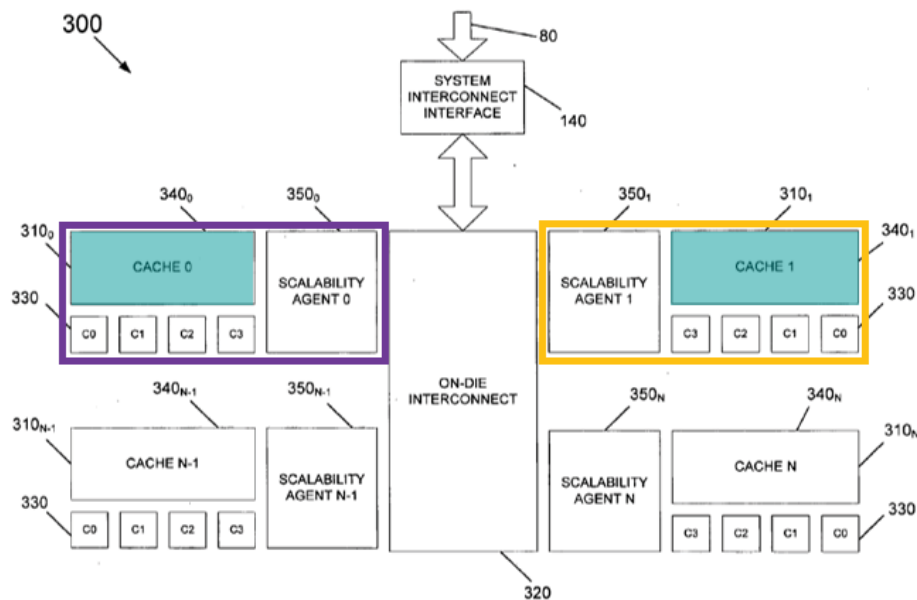
281. For example, Figure 2 of Sistla (below) shows a processor 400 that contains a first die 410₁ (corresponding to a first chiplet) (in purple) and a second die 410₂ (corresponding to a second chiplet) (in gold).



EX1047-Sistla, Figure 2, [0011]-[0013].

282. As further shown in Figure 2 and explained in Sistla, the processor 400 includes a last level cache (in teal), and a first portion 415₁ of the last level cache is included in the first die/chiplet (in purple) and a second portion 415₂ of the last level cache is included in the second die/chiplet (in gold).

283. Specifically, Sistla explains that, in the embodiment in Figure 1 (shown below), the processor 300 includes a first “core-cache cluster” 310₀ (in purple) that has a first cache 340₀ (in teal) and a second “core-cache cluster” 310₁ (in gold) that has a second cache 340₁ (in teal).



EX1047-Sistla, Figure 1, [0011]-[0013]. Moreover, the processor 300 has a “distributed shared cache architecture” in which the first cache 340₀ (in teal) and the second cache 340₁ (in teal) “act together as a shared last level cache (LLC)” that is shared by the processing cores (C0 to C3) 300 of both the first “core-cache cluster”

310₀ (in purple) and the second “core-cache cluster” 310₁ (in gold). EX1047-Sistla, Figure 2, [0011]-[0013].

284. Specifically, Sistla explains:

Referring now to FIG. 1, an exemplary embodiment of a multi-core processor 300 such as a clustered CMP having a scalability agent is shown. **Multi-core processor 300 includes a plurality of core-cache clusters 310₁-310_n (generically cluster 310) in communication with each other over an on-die interconnect 320.** Multi-core processor 300 is in communication with externally located devices over a system interconnect interface 140 and via an interconnect 80. According to one embodiment of the invention, on-die interconnect 320 is configured as a ring interconnect, but may be configured as an interconnect mesh (e.g., 2D mesh). **Each core-cache cluster 310 includes one or more cores 330 that share one of a plurality of caches 340₁-340_N (generically cache 340).** The architecture of core-cache clusters 310 may be in accordance with a caching bridge architecture or a **distributed shared cache architecture**. The transactions involving core-cache clusters 310 are controlled by corresponding scalability agents (SAs) 350₁-350_n (generically scalability agent 350) as described below. According to this architecture, multi-core processor 300 enables the latency of a first shared cache 340₁ to remain substantially constant despite increases in the number of cores in processor 300, ensuring that the scalar performance of threads with no or limited sharing remains constant.

EX1047-Sistla at [0011].

A processor in accordance with one embodiment of the present invention can be aggregated to increase its overall performance and support next generation processor designs. For example, if the core-cache cluster is using the

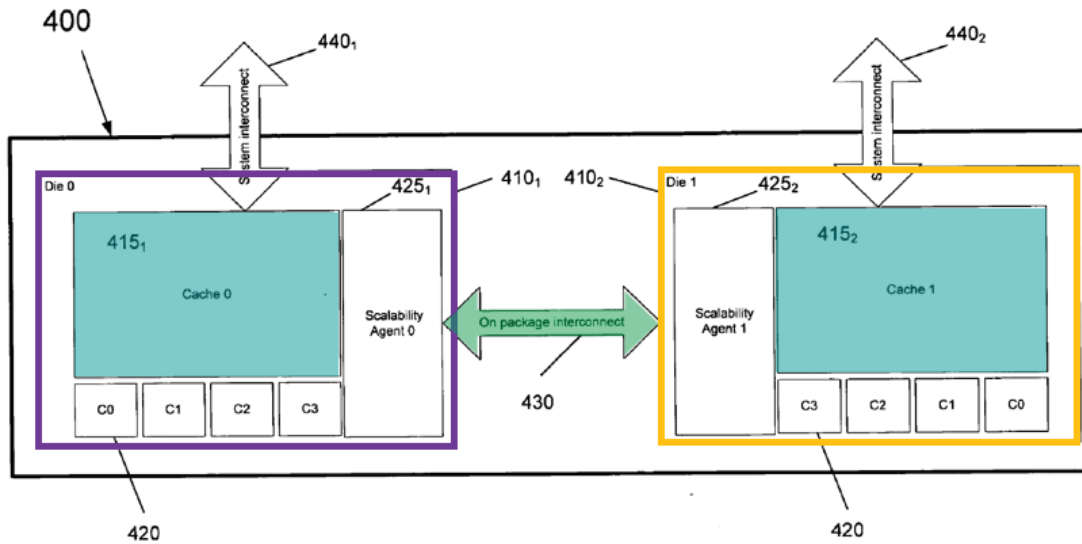
caching bridge style architecture, better performance may be realized by aggregating two (4-core) core-cache clusters to produce an eight core (8-core) multi-core processor. Also, two 4-core clusters can be used to build an 8-core processor in one generation, a 12-core processor in the next generation and 16-core processor in a subsequent generation. The appropriate number “N” of core-cache clusters 310 and the number of cores in each core-cache cluster may be determined to achieve optimum performance. This offers flexibility and the option to choose a simpler implementation. **In addition to the above mentioned advantages, SAs enable the individual caches to act together as a shared last level cache (LLC), and allows data that is private to each core-cache cluster to stay in its own cache, resulting in lower latency to private data. Thus further shown in FIG. 1, SA units 350₁-350₄ (generically SA units 350) (N=4) uniquely correspond to core-cache clusters 310 and may be address partitioned into independent SA units 350, each of which is responsible for a subset of an address space.**

EX1047-Sistla at [0012].

285. Moreover, while the Figure 1 embodiment discloses the core-cache clusters on a single die, in the embodiment shown in Figure 2 (reproduced again below), the processor comprises a “multi-chip package” in which the core-cache cluster 410₁ is contained within a first “Die 0” (in purple) (corresponding to a first chiplet) and the core-cache cluster 410₂ is contained within a second “Die 1” (in gold) (corresponding to a second chiplet). Moreover, the dies/chiplets are connected by a “point-to-point []interconnect” (in green), and the last level cache 415 (in teal) is distributed across the two dies/chiplets such that a first portion 415₁ of the cache

415 is contained within the first die/chiplet and a second portion 415₂ of the cache

415 is contained within the second die/chiplet.



EX1047-Sistla, Figure 2, [0011]-[0013]. In particular, Sistla discloses:

While shown in the embodiment of FIG. 1 as including a processor formed on a single die, in other embodiments a multi-chip package may be implemented. Referring now to FIG. 2, shown is a block diagram of a processor in accordance with another embodiment of the present invention. **As shown in FIG. 2, processor 400, which may be a multi-chip processor package, includes a first die 410₁ and a second die 410₂ connected by an on-package interconnect 430, which may be a point-to-point (PTP) interconnect. Each die 410 (generically) may include at least one core-cache cluster including a cache 415₁ and 415₂ and a plurality of cores 420. Each die 410 further includes a scalability agent 425₁ and 425₂. Processor 400 may communicate with other components of a system using a pair of system interconnects 440₁ and 440₂ both of which may be PTP links, in some embodiments. Of course, other implementations are possible. For example, each die may include multiple core-cache clusters and so forth. By**

implementing the SA on each of the dies, the dies can be joined at the interconnect, thus doubling the number of cores. In addition **the caches 415₁ and 415₂ will now behave like a shared cache of the size of both, providing a performance improvement. For example, if each die were a 4-core die with 8 megabytes (M) cache, then the combined 8-core processor may behave as a 16M cache.**

EX1047-Sistla at [0013].

286. As explained below in Section XI.B.1(d), a person of ordinary skill in the art would have been motivated by the teachings of Sistla to modify the teachings of Koker by dividing up Koker's last level L2 cache and including different portions of the L2 cache within the respective multi-core groups 365A to 365N.

287. Therefore, it is my opinion that Koker teaches that the first multi-core group 365A includes a cache memory hierarchy comprising L1 cache/shared memory 373 and L2 cache 375. When these teachings are incorporated into Kelleher (as discussed below in Section XI.B.1(c)) and modified based on the teachings of Sistla (as discussed below in Section XI.B.1(d)), the combined teachings of Kelleher, Collins, Koker, and Sistla disclose a first cache memory hierarchy (L1 cache/shared memory 373 and L2 cache 375 of Koker) at the first GPU chiplet (master GPU chip 220 of Kelleher, which corresponds to the MCG 365A of Koker).

288. Therefore it is my opinion that this limitation is obvious in view the combined teachings of Kelleher, Collins, Koker, and Sistla.

(b)Element 8[b]: wherein a first level of the first cache memory hierarchy is coherent within the first GPU chiplet; and

289. The combined teachings of Kelleher, Collins, Koker, and Sistla render obvious the additional requirement of Element 8[b].

290. The combined teachings of Kelleher, Collins, Koker, and Sistla disclose *wherein a first level (L1 cache of Koker) of the first cache memory hierarchy (L1 cache/shared memory 373 and L2 cache 375 of Koker (as modified by the teachings of Sistla)) is coherent within the first GPU chiplet* (master GPU chip 220 of Kelleher, which corresponds to MCG 365A of Koker).

291. Before explaining the relevant disclosures of Kelleher, Collins, Koker, and Sistla, I am including the following brief overview of how the '803 patent describes the first level of cache in the first cache memory hierarchy.

292. Figure 3 of the '803 patent (below) illustrates that multiple components—for example, workgroup processors (WGPs) 302 and fixed function blocks (GFXs) 304—of the first chiplet 106-1 (in purple) and the second chiplet 106-2 (in gold) access a local first level (L2 cache in orange) of the hierarchy (which consists of a three level cache: L1 cache in brown, L2 cache in orange, and L3 cache in teal). The patent further states that the L2 cache (in orange) is coherent within the first and second GPU chiplets (in purple and gold).

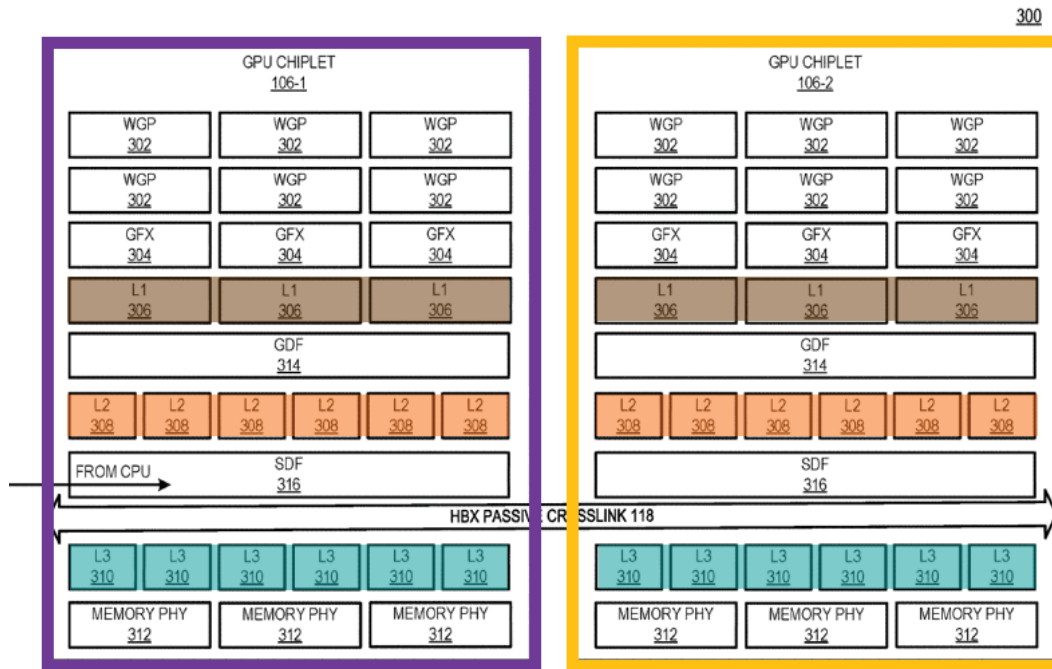


FIG. 3

293. Specifically, the '803 patent explains:

FIG. 3 is a block diagram illustrating a cache hierarchy of GPU chiplets coupled by a passive crosslink in accordance with some embodiments. The view 300 provides a hierarchical view of GPU chiplets 106-1, 106-2, and passive crosslink 118 of FIG. 1. **Each of the GPU chiplets 106-1, 106-2 include a plurality of workgroup processors 302 (WGP) and a plurality of fixed function blocks 304 (GFX) that communicate with a given channel's L1 cache memory 306. Each GPU chiplet 106 also includes a plurality of L2 cache memory 308 banks that are individually accessible, a plurality of L3 cache memory 310 channels, and a plurality of memory PHY 312 (denoted as graphics double data rate (GDDR) in FIG. 3 for indicating connection to GDDR memory) channels mapped to the L3 channels. The L2 level of cache is coherent within a single chiplet and the L3 level (or other last level) of cache is unified and coherent across all of the GPU chiplets 106.**

A graphics data fabric 314 (GDF) of each GPU chiplet 106 connects all of the L1 cache memories 306 to each of the channels of the L2 cache memory 308, thereby allowing each of the workgroup processors 302 and fixed function blocks 304 to access data stored in any bank of the L2 cache memory 308. Each GPU chiplet 106 also includes a scalable data fabric 316 (SDF) (also known as a SOC memory fabric) that routes across the graphics core (GC) and system on chip (SOC) IP cores to the passive crosslink 118. The GC includes CUs/WGPs, fixed function graphics blocks, caches above L3, and the like. Portions of the GPU used for traditional graphics and compute (i.e., the GC) are differentiable from other portions of the GPU used for handling auxiliary GPU functionality such as video decode, display output, and various system supporting structures that are contained on the same die. The passive crosslink 118 routes to both local L3 cache memory 310 of a chiplet (e.g., of GPU chiplet 106-1) and the L3 cache memory 310 of all other external GPU chiplets (e.g., GPU chiplet 106-2 in FIG. 3). In this manner, a memory address request is routed to the appropriate lanes on the passive crosslink 118 to access L3 cache memory 310 locally or at L3 cache memory 310 of a different GPU chiplet 106 (as discussed in more detail with respect to FIG. 5).

EX1001-'803, 5:1-42.

294. A person of ordinary skill in the art would have understood that the L2 cache in the '803 patent is coherent within the first chiplet because the workgroup processors (WGPs) 302 and fixed function blocks (GFXs) 304 of the first chiplet can access the L2 cache. EX1001-'803, 5:1-42. A coherent L2 cache ensures that the data stored in the cache remains valid and not stale (i.e., remains "up-to-date") when different components (such as the WGPs 302 and GFXs 304) within the first

chiplet write data to, and read data from, the same memory location of the L2 cache.

See EX1001-'803, 5:1-42.

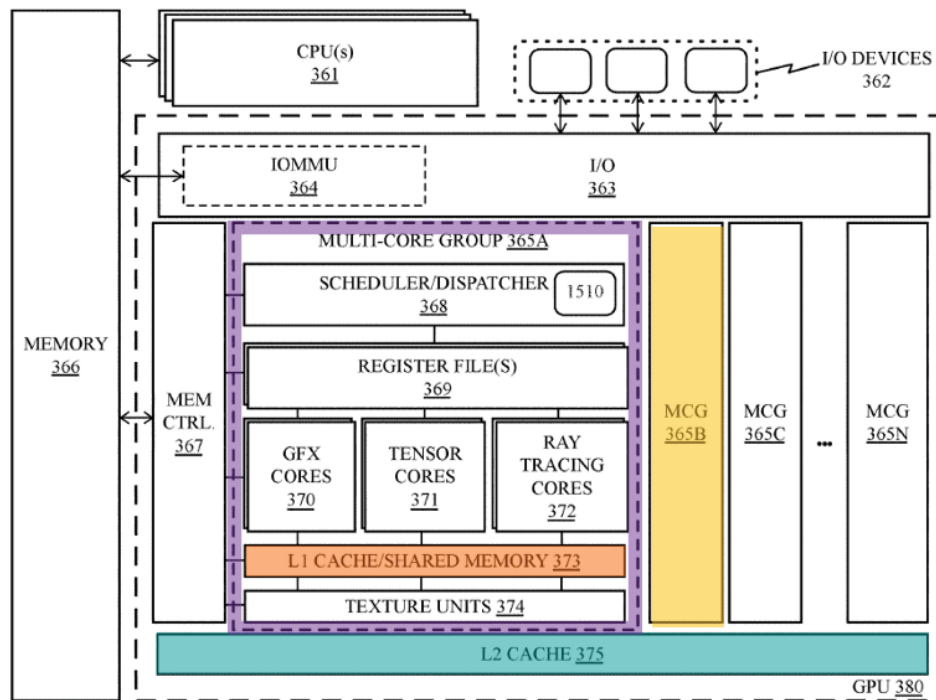
295. Indeed, the concept of cache coherency, as well as its associated challenges in multi-chip architecture systems were well-known in the art before 2019. For example, a 1993 paper explains:

By providing private (local) memory to each microprocessor we have created a very serious problem. At any given instant of time, multiple processors connected to the same bus would maintain local cached copies of a unique shared memory location. Each processor would then modify its local copy at one time or another. As a result, an inconsistent view of this particular shared memory location is projected across the system. This is commonly referred to as the cache coherence problem.

Cache coherence schemes (protocols) ensure that each request from a processor gets the most up-to-date copy of the block. There are a lot of different ways one can achieve this end result. Each strategy has distinct advantages and limitations.

EX1048-Nayyar, 32-33.

296. Like the '803 patent, Figure 3C of Koker, which I have reproduced below, illustrates multiple components belonging to the first MCG 365A (in purple), including graphics cores 370, tensor cores 371, and ray tracing cores 372. Each of these components access the first level (L1 cache/shared memory (in orange)) of the first cache memory hierarchy (which consists of a two-level cache: L1 cache/shared memory (in orange) and L2 cache (in teal) (as modified by the teachings of Sistla).



EX1031-Koker, Figure 3C.

297. For example, Koker explains:

As illustrated, a multi-core group 365A may include a set of graphics cores 370, a set of tensor cores 371, and a set of ray tracing cores 372. A scheduler/dispatcher 368 schedules and dispatches the graphics threads for execution on the various cores 370, 371, 372. A set of register files 369 store operand values used by the cores 370, 371, 372 when executing the graphics threads. These may include, for example, integer registers for storing integer values, floating point registers for storing floating point values, vector registers for storing packed data elements (integer and/or floating point data elements) and tile registers for storing tensor/matrix values. In one embodiment, the tile registers are implemented as combined sets of vector registers.

One or more combined level 1 (L1) caches and shared memory units 373 store graphics data such as texture

data, vertex data, pixel data, ray data, bounding volume data, etc., locally within each multi-core group

365A. One or more texture units 374 can also be used to perform texturing operations, such as texture mapping and sampling. A Level 2 (L2) cache 375 shared by all or a subset of the multi-core groups 365A-365N stores graphics data and/or instructions for multiple concurrent graphics threads. As illustrated, the L2 cache 375 may be shared across a plurality of multi-core groups 365A-365N. One or more memory controllers 367 couple the GPU 380 to a memory 366 which may be a system memory (e.g., DRAM) and/or a dedicated graphics memory (e.g., GDDR6 memory).

EX1031-Koker, 13:43-14:2.

298. Since the multiple cores—graphics cores 370, tensor cores 371, and ray tracing cores 372—share the local L1 cache/shared memory 373 and read and write data from and to the local L1 cache/shared memory, a person of ordinary skill in the art would have understood that Koker's L1 cache/shared memory is coherent within each multi-core group.

299. In fact, Koker explains that cache coherency techniques were well known in the art:

Coherency is maintained for data and instructions stored in the various caches 462A-462D, 456 and system memory 441 via inter-core communication over a coherence bus 464. For example, each cache may have cache coherency logic/circuitry associated therewith to communicate to over the coherence bus 464 in response to detected reads or writes to particular cache lines. In one implementation, a cache snooping protocol is implemented over the coherence bus 464 to snoop cache accesses. Cache snooping/coherency techniques are

well understood by those of skill in the art and will not be described in detail here to avoid obscuring the underlying principles of the invention.

EX1031-Koker, 17:46-58.

300. Koker also notes with respect to another embodiment:

The internal cache units 1504A-1504N and shared cache units 1506 represent a cache memory hierarchy within the processor 1500. The cache memory hierarchy may include at least one level of instruction and data cache within each processor core and one or more levels of shared mid-level cache, such as a Level 2 (L2), Level 3 (L3), Level 4 (L4), or other levels of cache, where the highest level of cache before external memory is classified as the LLC. In some embodiments, **cache coherency logic maintains coherency between the various cache units 1506 and 1504A-1504N.**”

EX1031-Koker, 40:29-38.

301. Koker’s teachings further confirms that a person of ordinary skill in the art would have understood that there were several standard coherency techniques, including techniques that would ensure that the local L1 cache in Koker is coherent within the first MCG 365A.

302. As set forth above, Koker teaches that a first level (L1 cache/shared memory 373 (in orange)) of the first cache memory hierarchy (L1 cache/shared memory 373 (in orange)/L2 cache 375 (in teal) (as modified by the teachings of Sistla)) is coherent within the first MCG 365A (in purple). As a result, when teachings of Kelleher and the teachings of Koker are combined, as set forth in

Section XI.B.1(c) below), the combined teachings disclose that a first level (L1 cache/shared memory 373 of Koker) of the first cache memory hierarchy (L1 cache/shared memory 373/L2 cache 375 of Koker (as modified by the teachings of Sistla) is coherent within the first GPU chiplet (master GPU chip 220 of Kelleher (corresponding to the first MCG 365A of Koker).

303. Therefore it is my opinion that this limitation is obvious in view the combination of Kelleher, Collins, Koker, and Sistla.

(c) The Kelleher, Collins, and Koker Combination

304. As I explained in Section XI.A.1.(e) (The Kelleher and Collins Combination), which I incorporate herein by reference, a person of ordinary skill in the art would have been motivated to combine Kelleher's and Collins's teachings by communicably coupling Kelleher's GPU chips 220 and 240 with Collins's interconnect structure/embedded multi-die bridge (EMIB).

305. A person of ordinary skill in the art would also have been motivated, with a reasonable expectation of success, to incorporate Koker's cache memory hierarchy teachings into the system of Kelleher and Collins.

306. Kelleher, Collins, and Koker are analogous art to each other and claims 1-4 and 7-10 of the '803 patent because each of them relates to communicably coupling chiplets or multi-core groups within a GPU.

307. Kelleher and Collins relate to communicably coupling chiplets for the reasons presented above in XI.A.1.(e) [The Kelleher and Collins Combination].

308. Regarding Koker, I have reproduced Figure 3 below, which illustrates “dedicated sets of graphics processing resources arranged into multi-core groups 365A-365N” that make up a GPU 380. EX1031-Koker, 12:21-24. *See also* EX1031-Koker, 13:36-35, 13:61-66.

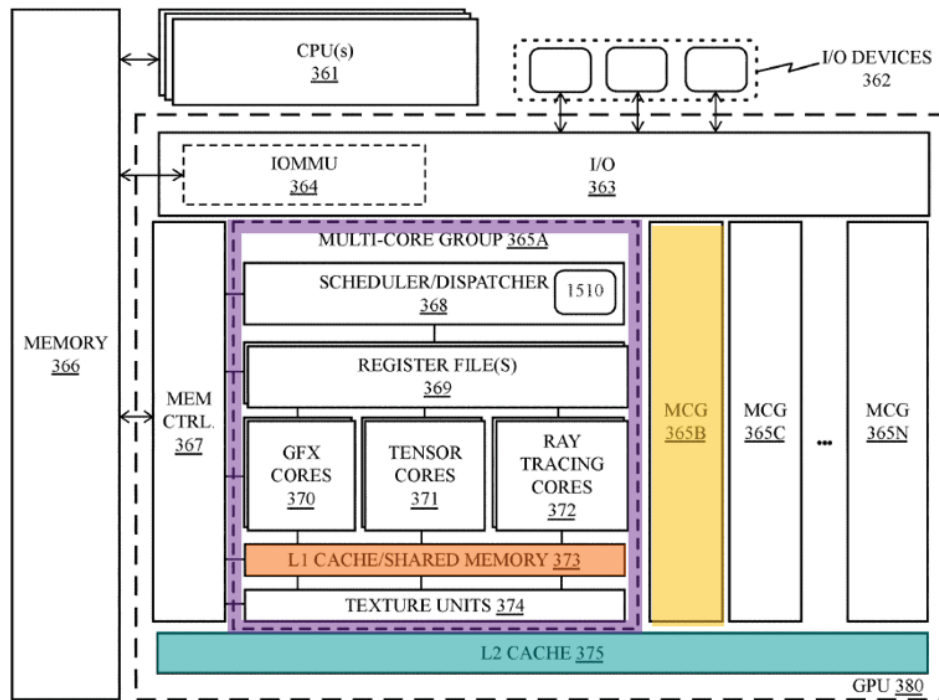


FIG. 3C

EX1031-Koker, Figure 3C.

309. A person of ordinary skill in the art would have recognized that both Kelleher’s and Koker’s systems divide processing resources into chiplets, or in the case of Koker, multi-core groups, to perform GPU operations in parallel and improve overall performance.

310. For example, in Figure 2 of Kelleher, Kelleher discloses a master GPU chiplet 220 (shown in purple), which includes processing module 265, and a slave GPU chiplet 240 (shown in gold), which includes a processing module 285. Each these chiplets 220 and 240 executes tasks in parallel to improve GPU performance.

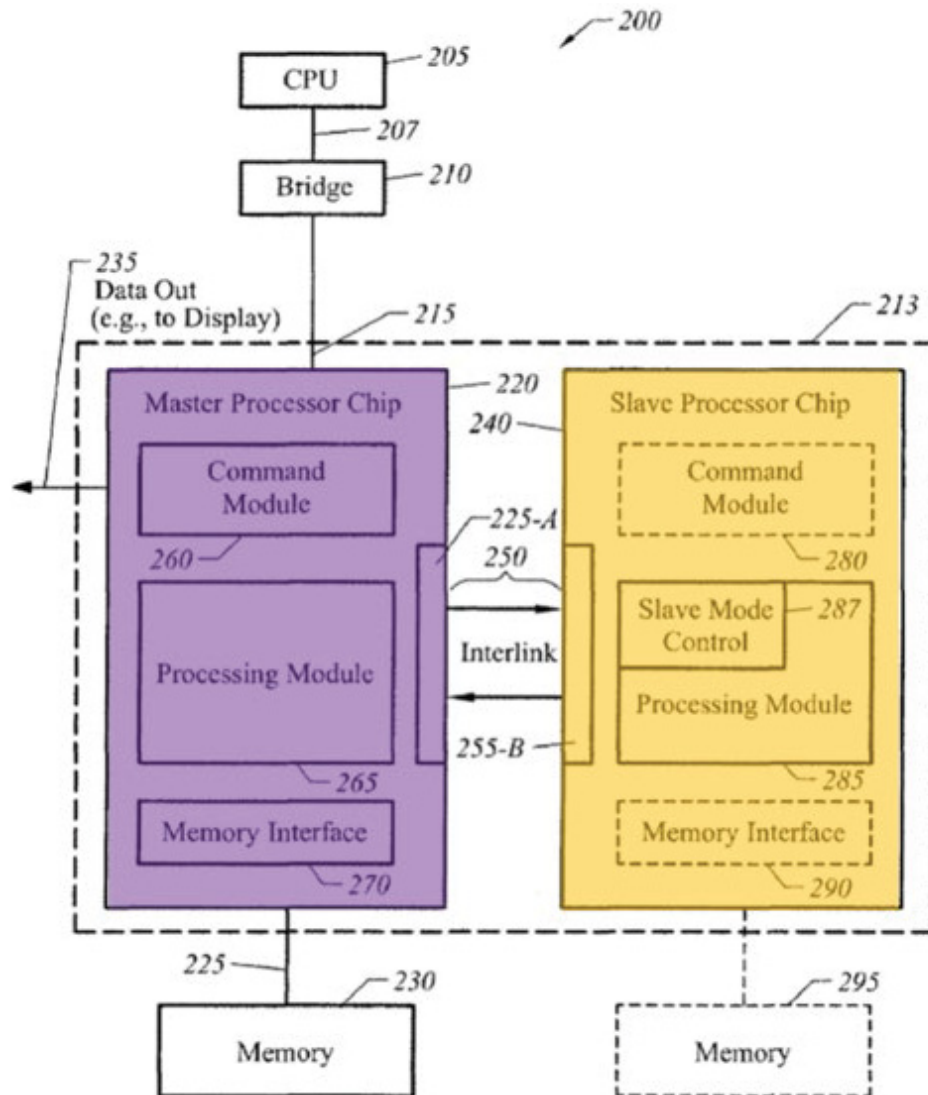


FIG. 2

EX1004-Kelleher, Figure 2.

311. In particular, Kelleher explains:

GPU chip 220 includes a command module 260. Command module 260 receives a graphics command stream from CPU 205, reads and parses the command string, and distributes work to the other stages within GPU chip 220. **At least one of the stages in the graphics pipeline of GPU chip 220 is a processing module 265 that performs a processing operation that may be performed in parallel.** One example is a shader stage. Shader stages in high performance GPUs are typically arranged as groups of shader pipelines along with logic that permits a load balancing operation in which the shading work is distributed between the shader pipelines. Slave GPU chip 240 is packaged with GPU chip 220 as a multi-chip module 213 that may be used as a replacement for an individual GPU chip. ...

Slave GPU chip 240 includes a processing module 285 capable of performing a processing function that is compatible with that performed by processing module 265. This permits, for example, two GPU's from the same or related product lines to be used in system 200. Thus, for example, slave GPU chip 240 may have its own command module 280 and memory interface 290. ...

GPU chip 220 acts as the master processor and utilizes processing module 285 as an additional processing resource that is used in parallel with processing module 265. In one embodiment slave GPU chip 240 also receives clock signals from GPU chip 220 to facilitate synchronization of slave GPU chip 240 to GPU chip 220. In the slave mode, a load balancing module (not shown) within processing module 265 determines a distribution of work to be performed by processing modules 265 and 285 and transfers a portion of the graphics data that it receives to processing module 285. Thus for example, **if processing modules 265 and 285 are substantially identical, about half of the processing that would ordinarily be performed by processing module 265 is**

off-loaded to processing module 285, permitting an approximately two-fold improvement in performance (e.g., speed or quality), assuming that processing modules 265 and 285 have a similar number of operable processing elements.

EX1004-Kelleher, 3:35-4:19.

312. Further, Figure 8 of Kelleher provides additional details for the processing resources of the two GPU chiplets—namely, the master GPU chip 720 (in purple) and the slave GPU chip 740 (in gold).

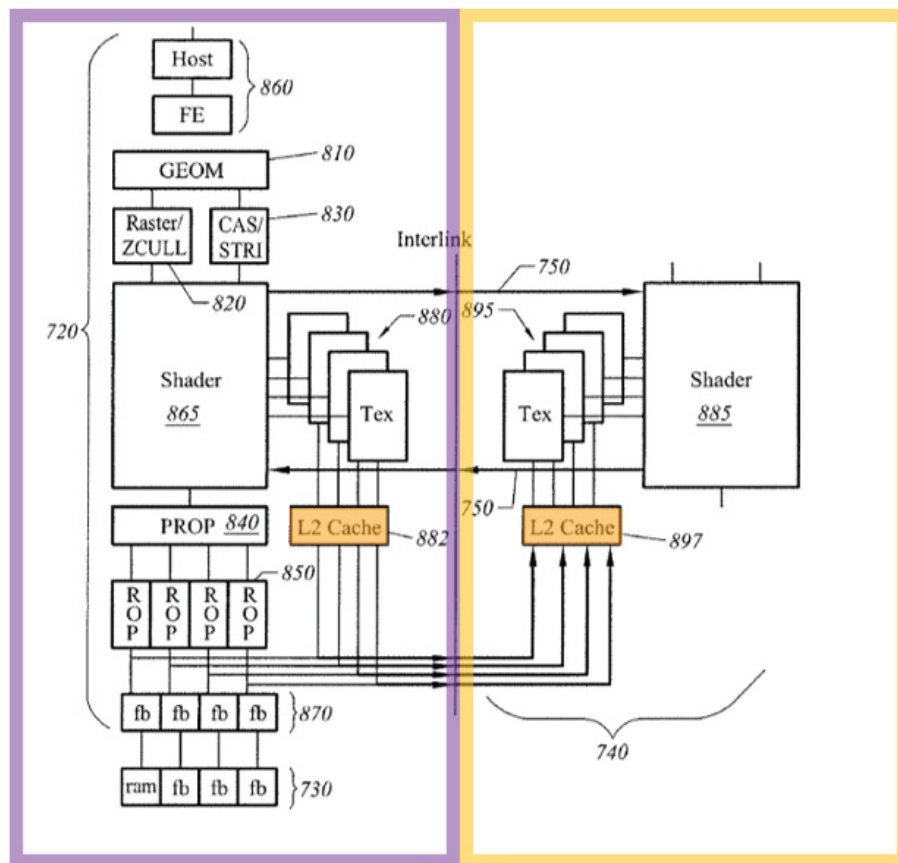


FIG. 8

EX1004-Kelleher, Figure 8.

313. More specifically, each of the GPU chips has texture modules 880/895, a shader stage 865/885, an L2 cache 882/897 (shown above in orange), and other processing resources. For example, Kelleher explains:

FIG. 8 is a block diagram showing some of the components within master GPU chip 720 and slave GPU chip 740 in more detail. Master GPU chip 720 includes a host and front end stage acting as a command module 860. A geometry stage 810 performs geometry processing. A raster stage 820 rasterizes primitives and also performs a Z culling operation to cull out occluded primitives. A CAS/STRI module is a color assembly/shader triangle module that receives attributes, such as colors, from the geometry state and calculates plane equations for interpolation by the shader stage 865. **Shader stage 865 includes a plurality of shader pipelines for pixel shading. In one embodiment shader stage 865 includes 24 pixel shader pipelines.** Raster operations (ROP) functions are performed in a pre raster operations stage 840 and associated raster operation stages 850. A frame buffer interface 870 serves as a memory interface to memory 730. **Texture modules 880 and cache memory 882 are utilized by shader 865 for texture mapping operations.**

Slave GPU chip 740 includes a shader stage 885 having a plurality of shader pipelines, texture modules 895, and cache 897. In one embodiment, shader stage 885 includes 24 shader pipelines such that slave GPU chip 740 doubles the shading power of a master GPU chip 720 having 24 shader pipelines.

EX1004-Kelleher, 7:9-47.

314. Similar to Kelleher's teachings regarding its two GPU chiplets (master GPU chiplet in purple and slave GPU chiplet in gold below), where each has its own

As illustrated, a **multi-core group 365A** may include a **set of graphics cores 370**, a **set of tensor cores 371**, and a **set of ray tracing cores 372**. A **scheduler/dispatcher 368** schedules and dispatches the graphics threads for execution on the various cores 370, 371, 372. A set of register files 369 store operand values used by the cores 370, 371, 372 when executing the graphics threads. These may include, for example, integer registers for storing integer values, floating point registers for storing floating point values, vector registers for storing packed data elements (integer and/or floating point data elements) and tile registers for storing tensor/matrix values. In one embodiment, the tile registers are implemented as combined sets of vector registers.

One or more combined level 1 (L1) caches and shared memory units 373 store graphics data such as texture data, vertex data, pixel data, ray data, bounding volume data, etc., locally within each multi-core group 365A. One or more texture units 374 can also be used to perform texturing operations, such as texture mapping and sampling. A Level 2 (L2) cache 375 shared by all or a subset of the multi-core groups 365A-365N stores graphics data and/or instructions for multiple concurrent graphics threads. As illustrated, the L2 cache 375 may be shared across a plurality of multi-core groups 365A-365N. One or more memory controllers 367 couple the GPU 380 to a memory 366 which may be a system memory (e.g., DRAM) and/or a dedicated graphics memory (e.g., GDDR6 memory).

EX1031-Koker, 13:43-14:2;

316. While Figure 8 of Kelleher, which I have reproduced again below, shows that its master GPU chiplet (in purple) (as well as its slave GPU chiplet (in gold) has a local L2 cache (in orange), Kelleher does not set forth a lot of details about the local L2 cache within the chiplets. For example, Kelleher does not explain

how its local L2 cache functions, operates, and integrates with other caches in a cache memory hierarchy or how its local L2 cache remains coherent.

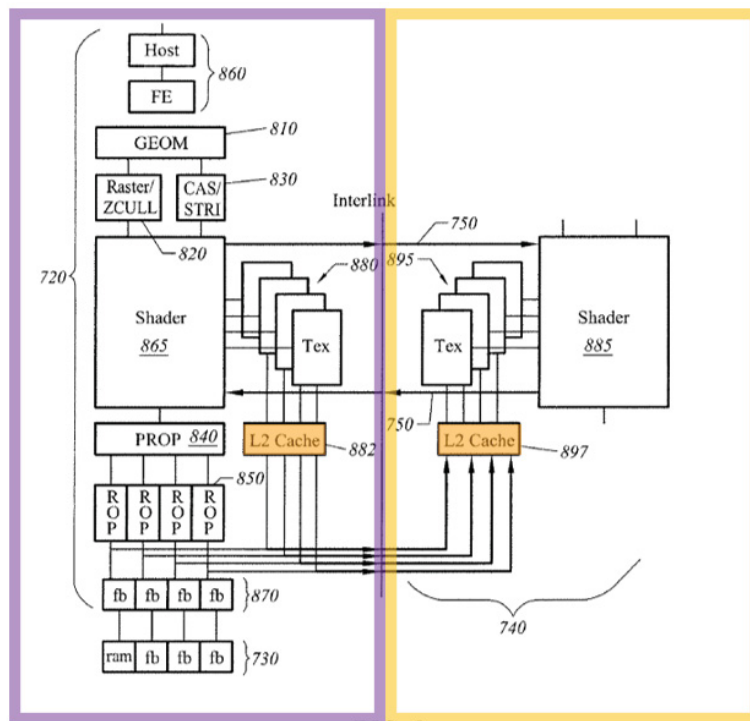


FIG. 8

EX1004-Kelleher, Figure 8. I note that Kelleher only briefly mentions the word “cache” four times at column 7, lines 25, 28, 33, and 40, but it does not provide enough detail about the cache for a person of ordinary skill in the art to determine how to configure a cache system at each chiplet.

317. Accordingly, a person of ordinary skill in the art would have turned to other prior art references, such as Koker, to fill in those missing details.

318. For example, as shown below in Figure 3C, Koker teaches that each multi-core group (MCG) 365A and 365B has a local cache, such as the L1 cache, to store data that is used *locally within* each multi-core group and that a higher-level

cache, for example, the L2 cache, is *shared among all* multi-core groups so that the multi-core groups can execute multiple concurrent software instruction threads.

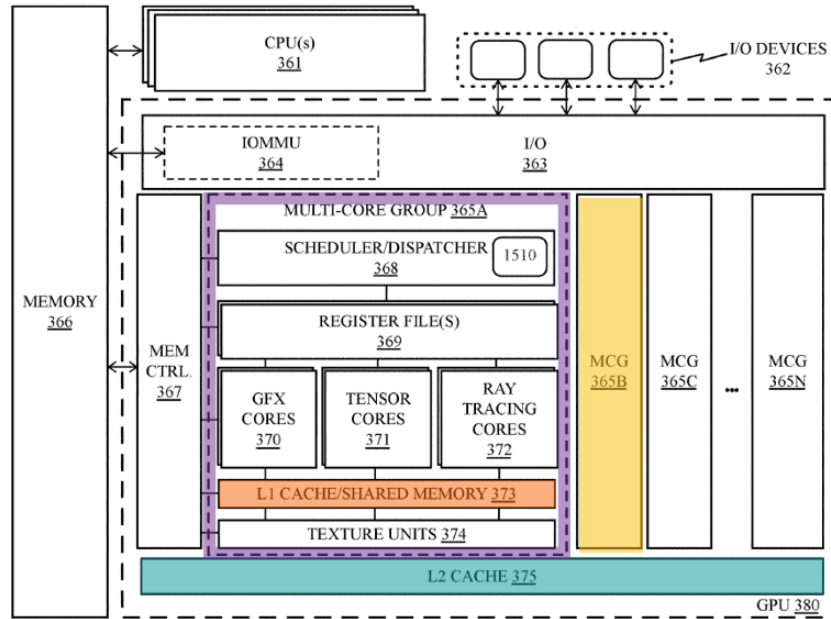


FIG. 3C

EX1031-Koker, Figure 3C.

319. For example, Koker explains:

One or more combined level 1 (L1) caches and shared memory units 373 store graphics data such as texture data, vertex data, pixel data, ray data, bounding volume data, etc., locally within each multi-core group 365A. One or more texture units 374 can also be used to perform texturing operations, such as texture mapping and sampling. A Level 2 (L2) cache 375 shared by all or a subset of the multi-core groups 365A-365N stores graphics data and/or instructions for multiple concurrent graphics threads. As illustrated, the L2 cache 375 may be shared across a plurality of multi-core groups 365A-365N. One or more memory controllers 367 couple the GPU 380 to a memory 366

which may be a system memory (e.g., DRAM) and/or a dedicated graphics memory (e.g., GDDR6 memory).

EX1031-Koker, 13:55-14:2.

320. With respect to another embodiment, Koker notes that “[t]he caches 462A-462D may comprise level 1 (L1) and level 2 (L2) caches... [and] one or more shared caches 456 may be included in the caching hierarchy and shared by sets of the cores 460A-460D.” EX1031-Koker, 17:36-40. Likewise, “[t]he internal cache units 1504A-1504N and shared cache units 1506 represent a cache memory hierarchy within the processor 1500... includ[ing] at least one level of instruction and data cache within each processor core and one or more levels of shared mid-level cache, such as a Level 2 (L2), Level 3 (L3), Level 4 (L4), or other levels of cache....” EX1031-Koker, 40:29-38.

321. A person of ordinary skill in the art would have recognized that, by sharing a level 2 cache among Kelleher’s chiplets, as Koker teaches with respect to its multi-core groups, Kelleher’s chiplets would efficiently share the latest, up-to-date data among the chiplets. This would allow the chiplets in Kelleher’s system to simultaneously process software instruction threads more quickly. *See* Ex1049-Balasubramanian, Section 1.1.1, 1-7.

322. Koker also provides a way to maintain coherency in its caches—such as its local cache L1 and its shared cache L2—so that the data in the caches remain valid and not stale (*i.e.*, up-to-date). This is important when different components

(such as the processing resources in one of Kelleher's chiplet or Kelleher's multiple chiplets) read data from and write data to the same memory locations within the caches. Namely, Koker discloses that cache coherency techniques were well known in the art:

Coherency is maintained for data and instructions stored in the various caches 462A-462D, 456 and system memory 441 via inter-core communication over a coherence bus 464. For example, each cache may have cache coherency logic/circuitry associated therewith to communicate to over the coherence bus 464 in response to detected reads or writes to particular cache lines. In one implementation, a cache snooping protocol is implemented over the coherence bus 464 to snoop cache accesses. Cache snooping/coherency techniques are well understood by those of skill in the art and will not be described in detail here to avoid obscuring the underlying principles of the invention.

EX1031-Koker, 17:46-58.

323. Koker's teachings regarding the caches contained locally within and shared by its multi-core groups can be applied to Kelleher's chiplets. Just like Koker's multi-core groups, Kelleher's chiplets are dedicated, modular processing units that would benefit from both local caches and a shared cache to coordinate data across the entire device and ensure that every component of the system that needs that data is accessing the most up to date version of that data. By incorporating Koker's cache memory hierarchy into Kelleher's chiplet-based architecture, a person of ordinary skill in the art would have achieved several benefits, such as a higher

cache capacity, reduced latency for local workloads, and reliable data sharing across processing units. For example, a 1993 paper explains:

[A] hierarchical, shared memory multiprocessor system ... is a scalable architecture, capable of delivering very high performance and is also easy to program (due to the shared main memory). “The shared-memory paradigm has the advantage that the programmer is not burdened with the issues of data partitioning, and accessibility of data from all processors simplifies the task of dynamic load distribution.”

.... Another salient feature of this architecture is that it uses an efficient snoopy cache protocol to maintain coherence across the caches in the system.

EX1048-Nayyar, at 46 (citations omitted).

324. A person of ordinary skill in the art also would have had a reasonable expectation of success of implementing Koker’s teachings regarding its cache memory hierarchy within Kelleher’s chiplet-based system. As I have explained, both Kelleher’s master and slave GPU chiplets and Koker’s first and second multi-core groups include dedicated processing resources that executing graphics tasks in parallel at the same time. Koker discloses the way to use a cache hierarchy having a local cache within each multi-core group (or chiplet, as applied to Kelleher’s system) and a shared cache for all multi-core groups (or chiplets, as applied to Kelleher’s system). EX1031-Koker, 13:55-66. Additionally, as Koker notes, cache coherency techniques were routine and implementing them within a system such as

Kelleher's system was well-within the skill of a person of ordinary skill in the art.

EX1031-Koker 17:55-58.

325. Therefore, it is my opinion that a person of ordinary skill in the art would have been motivated to combine the teachings of Koker with the teachings of Kelleher and Collins, because the cache memory hierarchy teachings of Koker would enable the Kelleher-Collins system to maintain coherency within its caches and effectively use its caches. It would also result in improved performance across chiplets, ensuring that information stored in memory remains accurate and can be shared across chiplets despite the fact that the chiplets are physically separated. The combination would have been a predictable and advantageous design choice, with a reasonable expectation of success, resulting in a multi-chiplet GPUs that is coherent and higher-performing.

(d) The Kelleher, Collins, Koker, and Sistla Combination

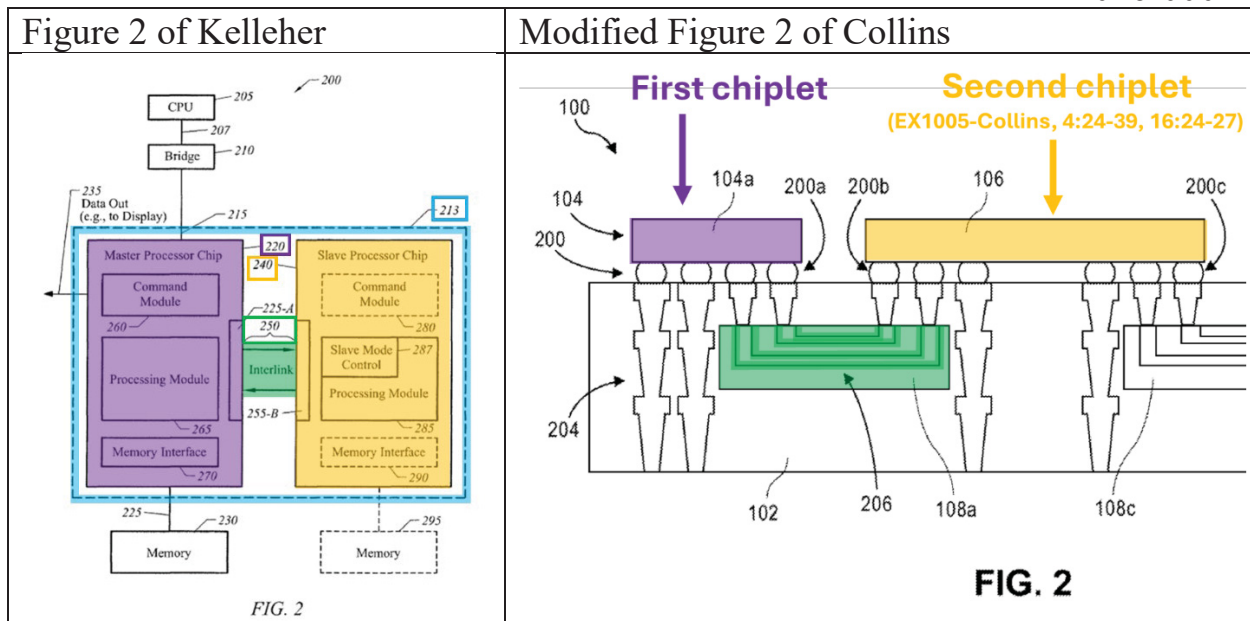
326. As I explained in Section XI.A.1.(e) (The Kelleher and Collins Combination) above, which I incorporate herein by reference, a person of ordinary skill in the art would have been motivated to combine the teachings of Kelleher and Collins by communicably coupling Kelleher's GPU chip 220 (corresponding to a first chiplet) and GPU chip 240 (corresponding to a second chiplet) with Collins's interconnect structure/embedded multi-die bridge (EMIB).

327. As I further explained in Section IX.B.1(c) (The Kelleher, Collins, and Koker Combination) above, which I incorporate herein by reference, a person of ordinary skill in the art would have been motivated to incorporate Koker's cache memory hierarchy teachings into the system of Kelleher and Collins.

328. A person of ordinary skill in the art would have further been motivated to combine the teachings of Kelleher, Collins, Koker, and Sistla.

329. Kelleher, Collins, Koker, and Sistla are analogous art to each other and claims 1-4 and 7-10 of the '803 patent because each of them relates to communicably coupling chiplets or multi-core groups within a GPU or CPU.

330. Kelleher and Collins relate to communicably coupling chiplets for the reasons presented above in Section XI.A.1.(e) (The Kelleher and Collins Combination) and for Elements 1[c] and 1[d] above. Moreover, as I have explained for Elements 1[c] and 1[d] (as well as in Section XI.A.1.(e)) above, the combined teachings of Kelleher and Collins disclose communicably coupling a first chiplet (in purple) and a second chiplet (in gold) together via a passive crosslink (in green) that is dedicated for inter-chiplet communications. Although a more detailed explanation is set forth above in Elements 1[c] and 1[d] and in Section XI.A.1.(e), for convenience, I am reproducing Figure 2 of Kelleher and Modified Figure 2 of Collins again below as a reminder of how the teachings of the references would have been combined.



EX1004-Kelleher, Figure 2; EX1005-Collins, Modified Figure 2 (replacing memory device 106 with another chiplet according to Collins's disclosure in 4:24-39 and 16:24-27).

331. Regarding Koker, as shown in Figure 3C below, it also relates to communicably coupling a first multi-core group 365A (in purple corresponding to a first chiplet) and a second multi-core group 365B (in gold corresponding to a second chiplet).

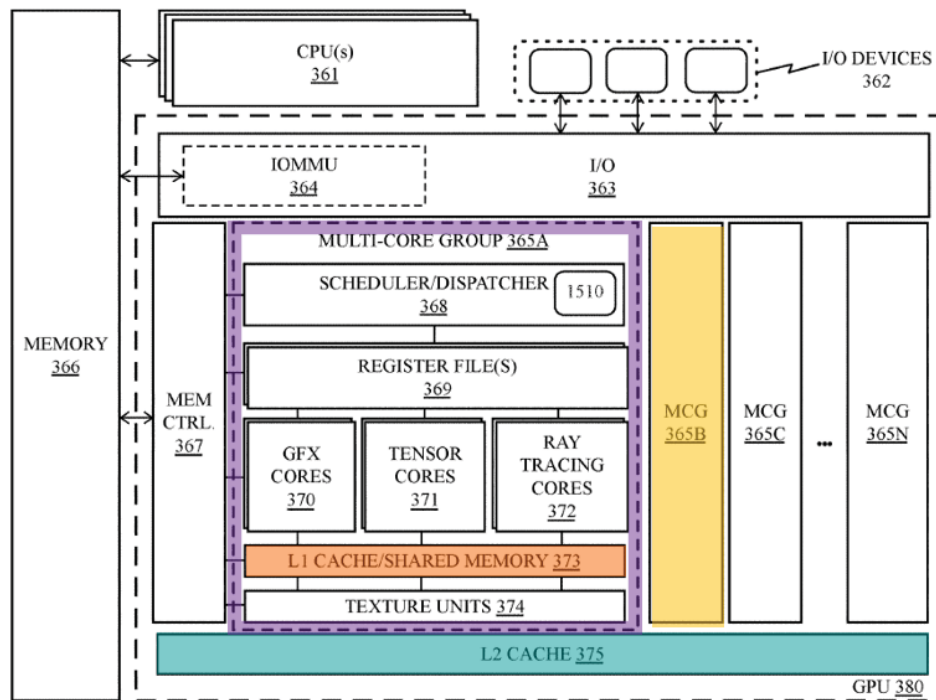


FIG. 3C

EX1031-Koker, Figure 3C. Both the first multi-core group 365A (in purple) and the second multi-core group 365B (in gold) are at least indirectly communicably coupled together because they share the same L2 cache 375 (in teal).

332. For example, Koker discloses:

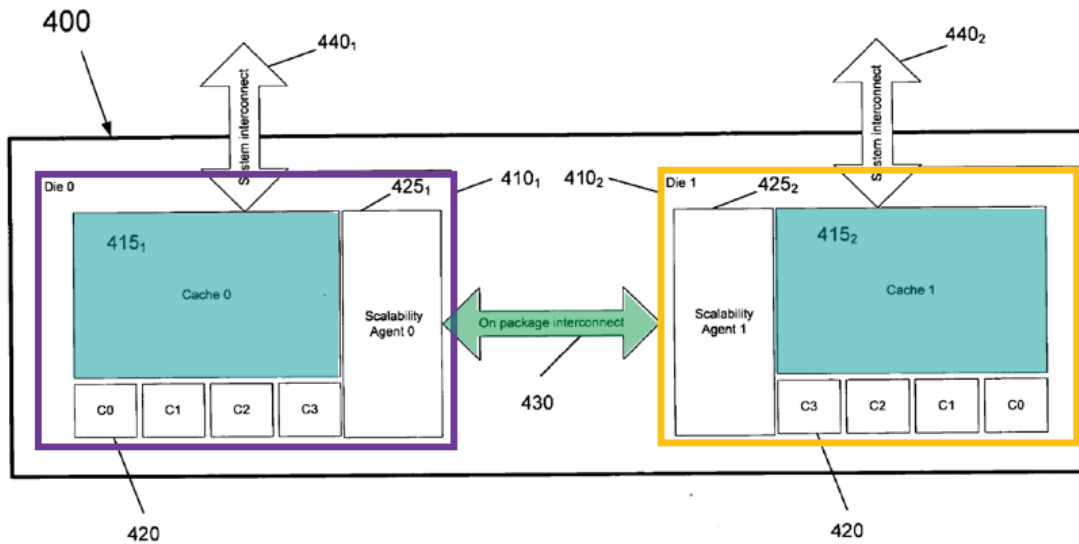
One or more combined level 1 (L1) caches and shared memory units 373 store graphics data such as texture data, vertex data, pixel data, ray data, bounding volume data, etc., locally within each multi-core group 365A. One or more texture units 374 can also be used to perform texturing operations, such as texture mapping and sampling. **A Level 2 (L2) cache 375 shared by all or a subset of the multi-core groups 365A-365N stores graphics data and/or instructions for multiple concurrent graphics threads. As illustrated, the L2 cache 375 may be shared across a plurality of multi-core groups 365A-365N.** One or more memory controllers 367 couple the GPU 380 to a memory 366

which may be a system memory (e.g., DRAM) and/or a dedicated graphics memory (e.g., GDDR6 memory).

EX1031-Koker, 13:55-14:2.

333. As noted above for Elements 8[a] and 8[b] and in Section XI.B.1(c) (The Kelleher, Collins, and Koker Combination), Koker teaches certain details for implementing a cache memory hierarchy. Nonetheless, a person of ordinary skill in the art would have looked to the teachings of Sistla to more efficiently incorporate Koker's last level L2 cache (in teal), which is external to the first multi-core group 365A (in purple) and the second multi-core group 365B (in gold), into the combined Kelleher and Collins system, in which the first chiplet (in purple) and the second chiplet (in gold) communicate via a dedicated passive crosslink (in green).

334. As shown in Figure 2 of Sistla below, Sistla teaches a first die 410₁ (corresponding to a first chiplet) (in purple) and a second die 410₂ (corresponding to a second chiplet) (in gold) that are communicably coupled via a "point-to-point (PTP) interconnect" 430 (in green).



EX1047-Sistla, Figure 2, [0011]-[0013]. In this configuration, Sistla teaches that the last level cache (LLC) (in teal), which is shared by the first die/chiplet (in purple) and the second die/chiplet (in gold), is divided into a first portion 415₁ (in teal) and a second portion 415₂ (in teal). The first portion 415₁ of cache is contained within the first die/chiplet, and the second portion 415₂ of cache is contained within the second die/chiplet.

335. For example, Sistla explains:

While shown in the embodiment of FIG. 1 as including a processor formed on a single die, in other embodiments a multi-chip package may be implemented. Referring now to FIG. 2, shown is a block diagram of a processor in accordance with another embodiment of the present invention. As shown in FIG. 2, **processor 400, which may be a multi-chip processor package, includes a first die 410₁ and a second die 410₂ connected by an on-package interconnect 430, which may be a point-to-point (PTP) interconnect. Each die 410 (generally) may include at least one core-cache cluster including a cache 415₁ and**

415₂ and a plurality of cores 420. Each die 410 further includes a scalability agent 425₁ and 425₂. Processor 400 may communicate with other components of a system using a pair of system interconnects 440₁ and 440₂ both of which may be PTP links, in some embodiments. Of course, other implementations are possible. For example, each die may include multiple core-cache clusters and so forth. By implementing the SA on each of the dies, the dies can be joined at the interconnect, thus doubling the number of cores. **In addition the caches 415₁ and 415₂ will now behave like a shared cache of the size of both, providing a performance improvement. For example, if each die were a 4-core die with 8 megabytes (M) cache, then the combined 8-core processor may behave as a 16M cache.**

EX1047-Sistla, [0013].

In addition to the above mentioned advantages, SAs enable the individual caches to act together as a shared last level cache (LLC)....

EX1047-Sistla, [0012].

336. A person of ordinary skill in the art would have been motivated, with a reasonable expectation of success, to modify Koker's cache memory hierarchy in light of Sistla's teachings by dividing up Koker's last level L2 cache (in teal) and including portions of the cache in the first chiplet (in purple) and the second chiplet (in gold) of the combined Kelleher and Collins system. *See, e.g.,* EX1049-Balasubramanian, Chapter 1, 1 ("LLCs have a better chance of optimizing miss rates, latency, and complexity if they are implemented as shared caches").

337. For example, Sistla teaches that, by using the “aggregated core-cache cluster” configuration, in which portions of the last level cache are included in respective dies, a device (such as a CPU) coupled to the core-cache clusters views the aggregated core-cache clusters as a single caching agent. Ultimately, such an implementation is beneficial in the Kelleher-Collins-Koker system because it allows the CPU to view the aggregated system as a single, monolithic device, and therefore allows for seamless programming of the CPU.

Based on such configurations, **a scalability agent is adapted to support the scalability agent protocol for aggregating core-cache clusters but the aggregation appears as a single caching agent to devices communicatively coupled to a system interconnect.** The purpose for shielding the number of core-cache clusters is two-fold. First, the shielding mitigates interconnect saturation issues caused by repetitive traffic over an on-die (or package) interconnect. Second, the shielding avoids repeated reconfiguration of a home agent. More specifically, if each processor core constituted a caching agent, an “N” cluster, “M” socket system would be perceived as a N*M caching agent system to the home agents in the system. The SA essentially functions by assuming the burden of local coherency and distributing coherency responsibilities to the core-cache clusters themselves. Otherwise, each time a core-cache cluster is modified or added, the home agent would need to be re-designed. Besides the shielded aggregation functionality, the scalability agent protocol is designed for handling local coherence between core-cache clusters on the same IC.

EX1047-Sistla, [0014].

Increased silicon efficiencies are now providing new opportunities for adding additional functionality into the processor silicon. As an example, applications are taking advantage of increased multi-threading capabilities realized from an increased number of processing cores in the same processor. **Performance of a multi-core processor may be optimized by mitigating system interconnect latency issues and ensuring that aggregated processor cores appear as one caching agent in order to avoid scalability issues and interconnect saturation.** Systems that include caches each associated with multiple cores typically are structured as split caches in which each cache acts as an independent cache domain. This increases the total number of coherence misses, thus increasing the misses per instruction. Split-caches also lose capacity due to replication of shared code/data in all the independent caches.

EX1047-Sistla, [0002].

A “scalability agent” is hardware and/or software that manages the flow of outgoing and incoming transactions into a socket associated with the core-cache cluster and supports the scalability agent protocol. According to one embodiment of the invention, **the scalability agent (i) aggregates core-cache clusters to appear as one caching agent,** (ii) handles local cache coherence between core-cache clusters on the same IC, and (iii) support scalability so that the operations of a core-cache cluster are not significantly effected if other core-cache clusters are added.

EX1047-Sistla, [0010].

338. This objective is similar to Kelleher’s objective in which the driver software of a CPU (in red) perceives Kelleher’s master processor chip

(corresponding to the first chiplet) (in purple) and Kelleher's slave processor chip (corresponding to the second chiplet) (in gold) as one monolithic GPU.

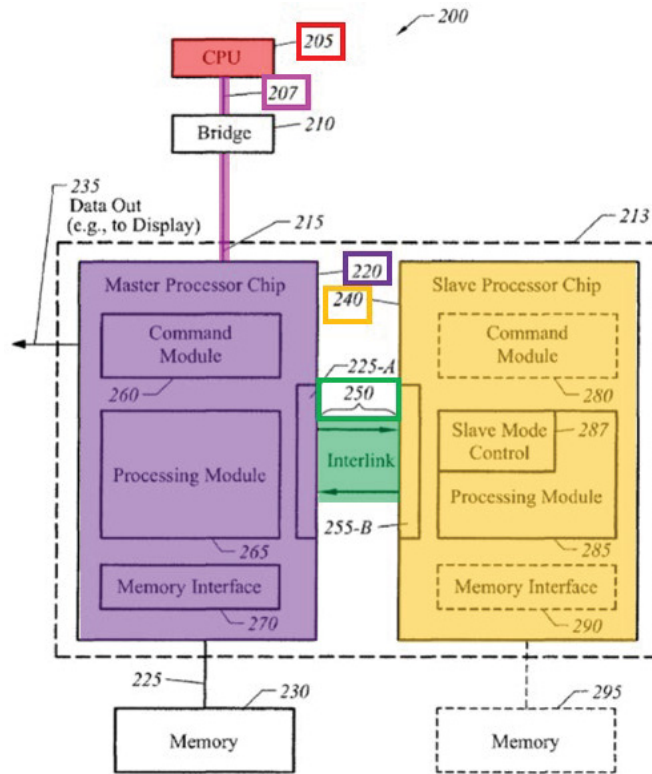


FIG. 2

EX1031-Kelleher, Figure 2. For example, Kelleher states:

In one embodiment GPU chip 220 and slave GPU chip 240 are formed from the same product design or with product designs having compatible processing modules 265 and 285. In this embodiment, a manufactured GPU chip is designed to support a normal mode of operation, a master mode of operation, and a slave mode of operation. In the normal mode of operation a GPU chip is operable as an individual processor and may be packaged as an individual processor. **However, two GPU chips may be packaged as a multi-chip module 213 in which one of the GPU chips acts as the master and the other acts as the slave.** The multi-chip module 213, which utilizes the resources of processor module 285 in parallel with processor module 265, has

improved performance compared with an individual GPU chip. Note that in the multi-chip module 213 only the master GPU 220 receives graphics instructions from CPU 205. **As result, replacing a single GPU chip with a multi-chip module having a master/slave pair of GPUs is a seamless change to driver software within CPU 205. That is, the driver software perceives the master/slave pair as a single GPU.** Thus with one type of GPU chip two different markets can be served by packaging either a single GPU chip or by packaging a master/slave pair of GPU chips for higher performance.

EX1004-Kelleher, 4:63-5:19.

339. One of ordinary skill in the art would have been motivated to incorporate Sistla's teachings into the system of Kelleher, Collins, and Koker because Sistla's configuration of aggregated core-cache clusters provides additional advantages. For example, Sistla's configuration improves the scalability of the system by allowing core-cache clusters to be efficiently added to the system without degrading the system's operation:

According to one embodiment of the invention, the "scalability agent protocol" is a communication scheme that enables the **aggregating of core-cache clusters operating independently from each other, and provides "graceful" scalability where the responsibility for maintaining memory coherency is approximately shared equally among the core-cache clusters** despite any increase or decrease in number, as the scalability agent protocol is partitionable among address regions in the system memory address space. **The latencies to the cache in the core-cache cluster substantially remain unchanged as the number of cores increases.**

EX1047-Sistla, [0009].

According to one embodiment of the invention, the scalability agent (i) aggregates core-cache clusters to appear as one caching agent, (ii) handles local cache coherence between core-cache clusters on the same IC, and (iii) support scalability so that the operations of a core-cache cluster are not significantly effected if other core-cache clusters are added.

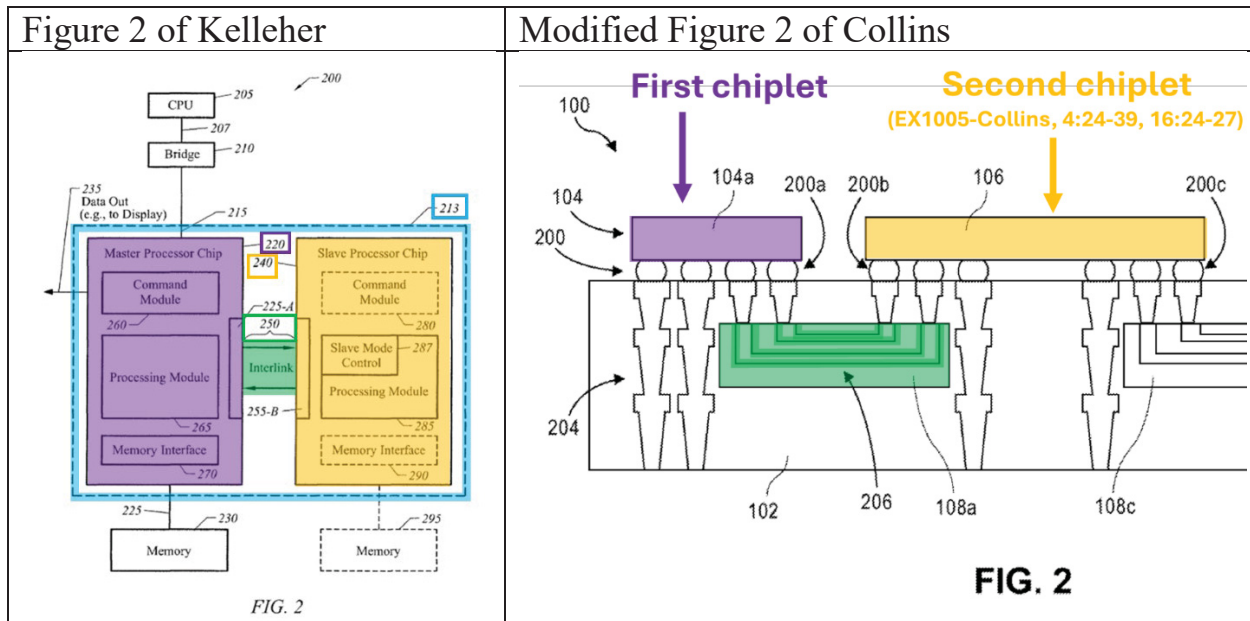
EX1047-Sistla, [0010].

The architecture of core-cache clusters 310 may be in accordance with a caching bridge architecture or a **distributed shared cache architecture**. The transactions involving core-cache clusters 310 are controlled by corresponding scalability agents (SAs) 3501-350_n (generically scalability agent 350) as described below. According to this architecture, **multi-core processor 300 enables the latency of a first shared cache 340₁ to remain substantially constant despite increases in the number of cores in processor 300**, ensuring that the scalar performance of threads with no or limited sharing remains constant.

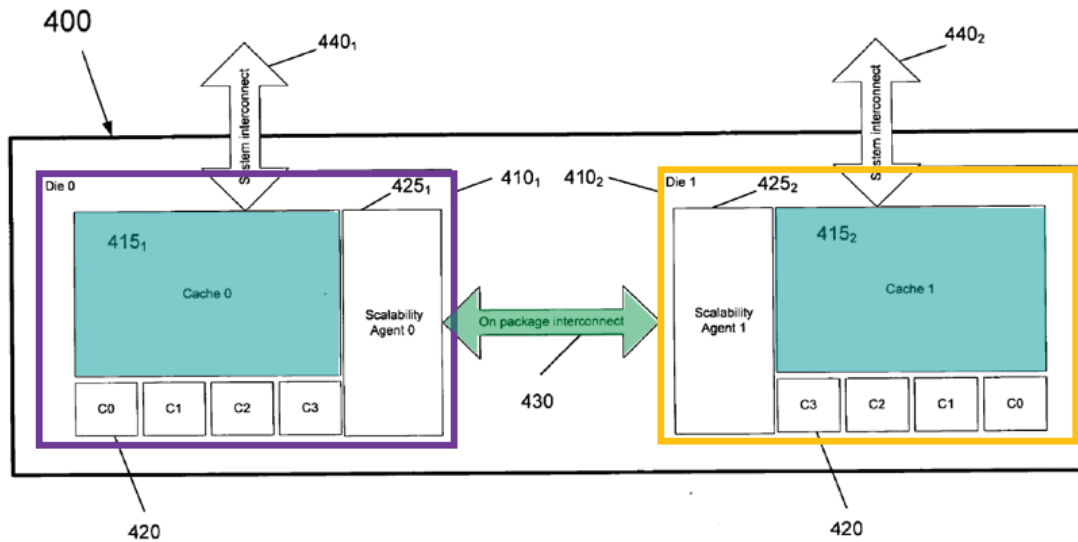
EX1047-Sistla, [0011].

340. Moreover, incorporating portions of Koker's last level L2 cache (in teal) in the first chiplet (in purple) and the second chiplet (in gold) of the combined Kelleher and Collins system, rather than using an external last level cache, reduces the number of components that needs to be connected to the first chiplet and second chiplet with relatively large external connective structure, which in turn, reduces the overall size of the system.

341. A person of ordinary skill in the art also would have had a reasonable expectation of success of modifying the combined system of Kelleher, Collins, and Koker by including portions of Koker's last level L2 cache into the respective first and second chiplets of the combined system, as taught by Sistla. For example, as shown below, the combined teachings of Kelleher and Collins disclose a first chiplet (in purple) communicably coupled to a second chiplet (in gold) via a dedicated passive crosslink (in green).



342. Similarly, as shown below, Sistla discloses a first die/chiplet (in purple) communicably coupled to a second die/chiplet (in gold) via a “point-to-point” link (in green).



EX1047-Sistla, Figure 2, [0011]-[0013]. Moreover, Sistla teaches the implementation details of designing an aggregated core-cluster system, in which portions of the shared last level cache (in teal) are contained within the first die/chiplet (in purple) and the second die/chiplet (in gold) and teaches that such a system achieves the advantages above.

343. A person of ordinary skill in the art would have readily appreciated how to incorporate portions of the shared last level cache (in teal) within the first die/chiplet (in purple) and the second die/chiplet (in gold) and doing so would have been well within the skill of a person of ordinary skill in the art, without undue experimentation. *See, e.g.*, EX1049-Balasubramonian, Section 1.2, 7-10 (discussing various implementations of distributed shared caches).

344. A person of ordinary skill in the art therefore would have been motivated to use Sistla's teachings of its aggregated core-cache cluster configuration

to implement the last level cache of Koker in the first and second chiplets of the combined Kelleher and Collins system, because doing so would have yielded predictable results that are aligned with Kelleher's performance objectives, and it would have been a matter of routine design choice within the skill of a person of ordinary skill in the art. *See, e.g.*, EX1049-Balasubramanian, Section 1.2, 7-10 (discussing various implementations of distributed shared caches).

345. It is therefore my conclusion that a person of ordinary skill in the art would have been motivated to combine the teachings of Kelleher, Collins, Koker, and Sistla.

(e) Element 8[c]: a second cache memory hierarchy at the second GPU chiplet,

346. The combined teachings of Kelleher, Collins, Koker, and Sistla render obvious the additional requirement of Element 8[c].

347. For the same reasons I explained with respect to Element 8[a], incorporated herein by reference, the combined teachings of Kelleher, Collins, Koker, and Sistla disclose a *second cache memory hierarchy* (L1 cache/shared memory 373 (in orange) and L2 cache 375 (in teal) of Koker (as modified by Sistla) *at the second GPU chiplet* (slave GPU chip 240 of Kelleher (in gold), which corresponds to multi-core group 365B of Koker (in gold)).

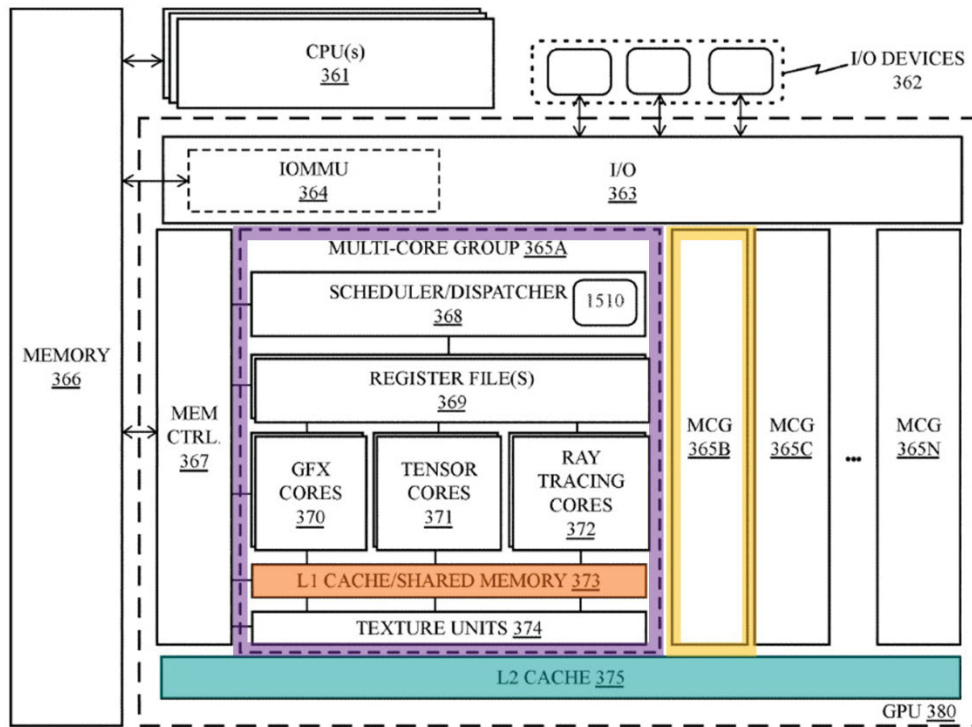


FIG. 3C

EX1031-Koker, Figure 3C.

348. Koker confirms that the details of its first multi-core group 365A (in purple) applied to its second multi-core group 365B (in gold), noting that “[w]hile the details of only a single multi-core group 365A are provided, ... the other multi-core groups 365B-365N may be equipped with the same or similar sets of graphics processing resources.” EX1031-Koker, 13:38-51.

349. Moreover, as explained for Element 8[a] and in Section XI.B.1(d), a person of ordinary skill in the art would have been motivated to divide up Koker’s L2 cache (based on the teachings of Sistla) and include portions of the L2 cache in the respective multi-core groups 365A to 365N (corresponding to the first chiplet and second chiplet of the combination of Kelleher and Collins).

350. Therefore it is my opinion that this limitation is obvious in view the combination of Kelleher, Collins, Koker, and Sistla.

(f) Element 8[d]: wherein a first level of the second cache memory hierarchy is coherent within the second GPU chiplet.

351. The combined teachings of Kelleher, Collins, Koker, and Sistla render obvious the additional requirement of Element 8[d].

352. For the same reasons I explained with respect to Element 8[b], incorporated herein by reference, the combined teachings of Kelleher, Collins, Koker, and Sistla disclose *wherein a first level (L1 cache/shared memory 373 (in orange) of the second cache memory hierarchy (L1 cache/shared memory 373 (in orange) and L2 cache 375 (in teal) of Koker (as modified by Sistla)) is coherent within the second GPU chiplet (slave GPU chip 240 of Kelleher (in gold), which corresponds to multi-core group 365B of Koker (in gold)).*

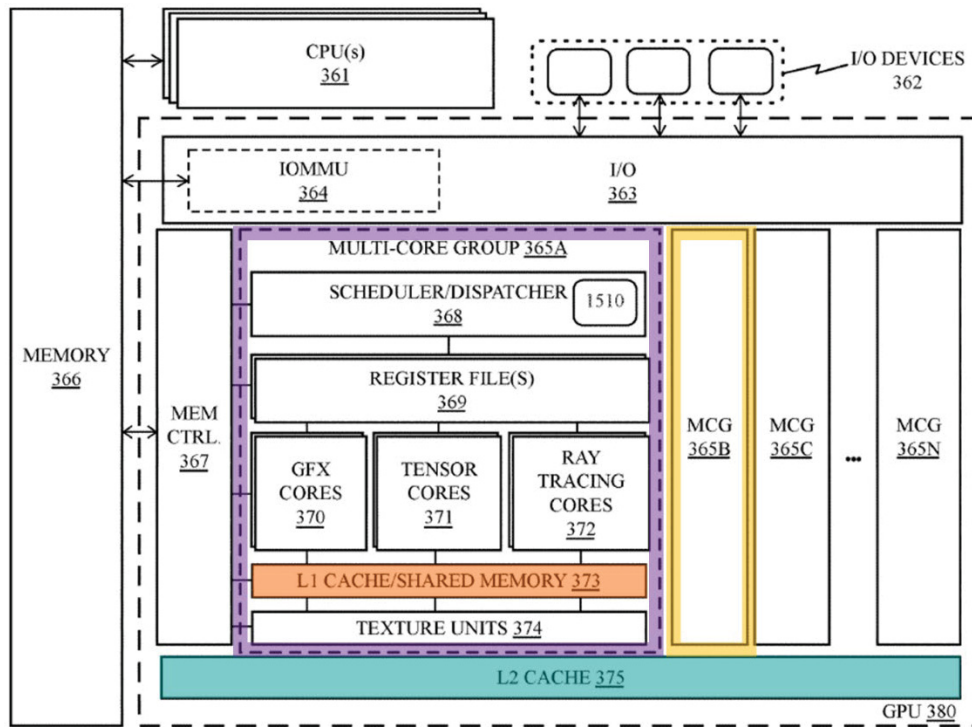


FIG. 3C

EX1031-Koker, Figure 3C.

353. Again, Koker confirms that the details of its first multi-core group 365A (in purple) applied to its second multi-core group 365B (in gold), noting that “[w]hile the details of only a single multi-core group 365A are provided, ... the other multi-core groups 365B-365N may be equipped with the same or similar sets of graphics processing resources.” EX1031-Koker, 13:38-51.

354. Therefore it is my opinion that this limitation is obvious in view the combination of Kelleher, Collins, and Koker.

2. Dependent Claim 9

355. Claim 9 depends from claim 1. Therefore, I incorporate by reference my discussion of claim 1.

356. The combined teachings of Kelleher, Collins, Koker, and Sistla disclose the additional requirements of Claim 9.

**(a) Element 9[a]: The system of claim 8, further comprising:
a unified cache memory including both a last level of the
first cache memory hierarchy and a last level of the
second cache memory hierarchy,**

357. The combined teachings of Kelleher, Collins, Koker, and Sistla render obvious the additional requirement of Element 9[a].

358. Therefore it is my opinion that this limitation is obvious in view the combined teachings of Kelleher, Collins, Koker, and Sistla.

359. Kelleher, Collins, Koker, and Sistla teach that *a unified cache memory* (L2 cache 375 of Koker (as modified by Sistla)) *including both a last level* (L2 cache 375 of Koker (as modified by Sistla)) *of the first cache memory hierarchy* (Koker's L1 cache/shared memory 373/L2 cache 375 (as modified by Sistla to be located within the first multi-core group 365A)) implemented in the master GPU chip 220 of Kelleher (which corresponds to the first multi-core group 365A of Koker)) *and a last level* (L2 cache 375 of Koker (as modified by Sistla)) *of the second cache memory hierarchy* (Koker's L1 cache/shared memory 373/L2 cache 375 (as modified by Sistla to be located within the second multi-core group 365B)) implemented in the slave GPU chip 240 of Kelleher (which corresponds to the second multi-core group 365B of Koker)).

360. Before explaining the relevant disclosures of Kelleher, Collins, Koker, and Sistla, I am including the following brief overview of how the '803 patent describes its unified cache memory.

361. As shown below, in Figure 3 of the '803 patent, each of the first and second chiplets 106-1 and 106-2 includes a “cache memory hierarchy”—a tiered arrangement of memory subsystems having multiple levels, such as Level 1 (L1) (in brown), Level 2 (L2) (in orange), and Level 3 (L3) caches (in teal). The L3 cache is the last level of the first memory hierarchy at the first chiplet 106-1 (in purple), and the L3 cache is the last level of the second memory hierarchy at the second chiplet 106-2 (in gold).



FIG. 3

EX1001-'803, Figure 3.

362. The '803 patent further explains:

FIG. 3 is a block diagram illustrating a cache hierarchy of GPU chiplets coupled by a passive crosslink in accordance with some embodiments. The view 300 provides a hierarchical view of GPU chiplets 106-1, 106-2, and passive crosslink 118 of FIG. 1 . **Each of the GPU chiplets 106-1, 106-2 include** a plurality of workgroup processors 302 (WGP) and a plurality of fixed function blocks 304 (GFX) that communicate with a given channel's L1 cache memory 306. Each GPU chiplet 106 also includes a plurality of L2 cache memory 308 banks that are individually accessible, a plurality of **L3 cache memory 310 channels**, and a plurality of memory PHY 312 (denoted as graphics double data rate (GDDR) in FIG. 3 for indicating connection to GDDR memory) channels mapped to the L3 channels. The L2 level of cache is coherent within a single chiplet and **the L3 level (or other last level) of cache is unified and coherent across all of the GPU chiplets 106.**

A graphics data fabric 314 (GDF) of each GPU chiplet 106 connects all of the L1 cache memories 306 to each of the channels of the L2 cache memory 308, thereby allowing each of the workgroup processors 302 and fixed function blocks 304 to access data stored in any bank of the L2 cache memory 308. Each GPU chiplet 106 also includes a scalable data fabric 316 (SDF) (also known as a SOC memory fabric) that routes across the graphics core (GC) and system on chip (SOC) IP cores to the passive crosslink 118. The GC includes CUs/WGPs, fixed function graphics blocks, caches above L3, and the like. Portions of the GPU used for traditional graphics and compute (i.e., the GC) are differentiable from other portions of the GPU used for handling auxiliary GPU functionality such as video decode, display output, and various system supporting structures that are contained on the same die. **The passive crosslink 118 routes to both local L3 cache memory 310 of a chiplet (e.g., of GPU chiplet 106-1) and the L3 cache memory 310 of all other external GPU chiplets**

(e.g., GPU chiplet 106-2 in FIG. 3). In this manner, a memory address request is routed to the appropriate lanes on the passive crosslink 118 to access L3 cache memory 310 locally or at L3 cache memory 310 of a different GPU chiplet 106 (as discussed in more detail with respect to FIG. 5).

EX1001-'803, 5:1-42.

363. A person of ordinary skill in the art would recognize that because the L3 cache is shared by and coherent across the first chiplet 106-1 and the second chiplet 106-2, it is a “unified” cache memory. EX1001-'803, 5:1-42. For example, the '803 patent explains:

[V]arious architectures already have at least one level of cache (e.g., L3 or other last level cache (LLC)) that is coherent across the entire conventional GPU die. **Here, the chiplet-based GPU architecture positions those physical resources (e.g., LLC) on different dies and communicably couples those physical resources such that the LLC level is unified and remains cache coherent across all GPU chiplets.**

EX1001-'803, 2:18-26; *see also* EX1001-'803, 5:1-42 (quoted above).

364. Figure 3C of Koker, which I have reproduced below, similarly depicts the L2 cache (in teal) as the last level cache (LLC) of the first memory hierarchy (which includes L1 cache/shared memory 373 and L2 cache 375) for the first multi-core group, MCG 365A (in purple), and the L2 cache is also the last level cache (LLC) of the second memory hierarchy (which includes L1 cache/shared memory 373 and L2 cache 375) for the second multi-core group, MCG 365B (in gold).

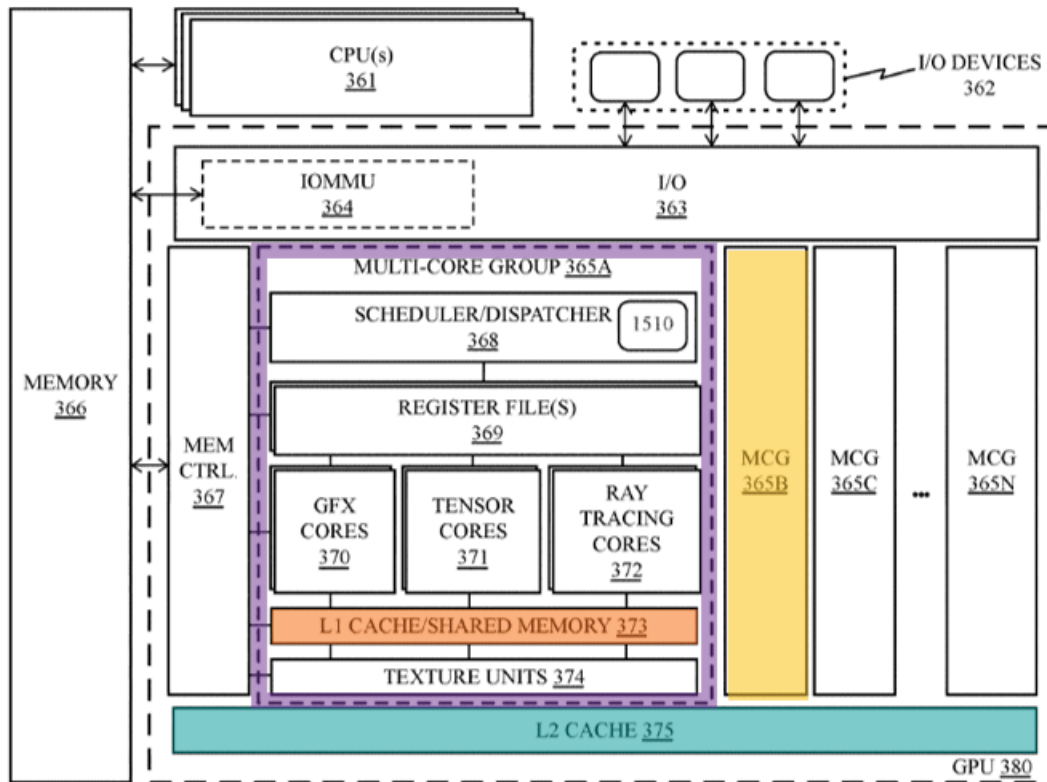


FIG. 3C

EX1031-Koker, Figure 3C.

365. Koker confirms that the teachings regarding multi-core group 365A may be applied to multi-core group 365B, noting that “[w]hile the details of only a single multi-core group 365A are provided, ... the other multi-core groups 365B-365N may be equipped with the same or similar sets of graphics processing resources.” EX1031-Koker, 13:38-51. Koker also notes that “[t]he caches 462A-462D may comprise level 1 (L1) and level 2 (L2) caches... [and] one or more shared caches 456 may be included in the caching hierarchy and shared by sets of the cores 460A-460D.” EX1031-Koker, 17:36-40. Likewise, “[t]he internal cache units 1504A-1504N and shared cache units 1506 represent a cache memory

hierarchy within the processor 1500... includ[ing] at least one level of instruction and data cache within each processor core and one or more levels of shared mid-level cache, such as a Level 2 (L2), Level 3 (L3), Level 4 (L4), or other levels of cache, where the highest level of cache before external memory is classified as the LLC [Last Level Cache].” EX1031-Koker, 40:29-38.

366. Because Koker’s L2 cache is shared by and coherent across the two multi-core groups, a person of ordinary skill in the art would have recognized that the L2 cache is a “unified” cache memory. I provided more details regarding Koker’s teachings with respect to Elements 8[a] to 8[d] which I incorporate herein by reference. Koker confirms:

One or more combined level 1 (L1) caches and shared memory units 373 store graphics data such as texture data, vertex data, pixel data, ray data, bounding volume data, etc., locally within each multi-core group 365A. One or more texture units 374 can also be used to perform texturing operations, such as texture mapping and sampling. **A Level 2 (L2) cache 375 shared by all or a subset of the multi-core groups 365A-365N stores graphics data and/or instructions for multiple concurrent graphics threads. As illustrated, the L2 cache 375 may be shared across a plurality of multi-core groups 365A-365N.** One or more memory controllers 367 couple the GPU 380 to a memory 366 which may be a system memory (e.g., DRAM) and/or a dedicated graphics memory (e.g., GDDR6 memory).

EX1031-Koker, 13:55-14:2.

367. Koker also confirms that cache coherency techniques were well known in the art:

Coherency is maintained for data and instructions stored in the various caches 462A-462D, 456 and system memory 441 via inter-core communication over a coherence bus 464. For example, each cache may have cache coherency logic/circuitry associated therewith to communicate to over the coherence bus 464 in response to detected reads or writes to particular cache lines. In one implementation, a cache snooping protocol is implemented over the coherence bus 464 to snoop cache accesses. Cache snooping/coherency techniques are well understood by those of skill in the art and will not be described in detail here to avoid obscuring the underlying principles of the invention.

EX1031-Koker, 17:46-58.

368. A person of ordinary skill in the art would have further understood that Koker's L2 cache is a "unified" cache memory because it is located on different multi-core groups (MCG 365A and MCG 365B) (as modified based on the teachings of Sistla (as explained for Elements 8[a] and 8[c] and in Section XI.B.1(d)) is shared by the multi-core groups, and communicably couples the resources of the multi-core groups.

369. The '803 patent similarly teaches that the "chiplet-based GPU architecture positions those physical resources (e.g., LLC) on different dies [chiplets 106-1 and 106-2] and communicably couples those physical resources such that the

LLC level is unified and remains cache coherent across all GPU chiplets.”

EX1001-'803, 2:18-24.

370. Therefore it is my opinion that this limitation is obvious in view the combination of Kelleher, Collins, Koker, and Sistla.

(b)Element 9[b]: wherein the unified cache memory is coherent across all GPU chiplets of the GPU chiplet array.

371. The combined teachings of Kelleher, Collins, Koker, and Sistla render obvious the additional requirement of Element 9[b].

372. For the same reasons I explained with respect to Elements 8[a], 8[b], 8[c], 8[d], and 9[a], which I incorporate herein by reference, the combined teachings of Kelleher, Collins, Koker, and Sistla teach that *the unified cache memory* (Koker's L2 cache 375 (as modified by Sistla)) *is coherent across all GPU chiplets* (Kelleher's GPU chips 220 and 240, which correspond to Koker's multi-core groups 365A and 365B, respectively) *of the GPU chiplet array* (Kelleher's multi-chip module 213).

373. For example Koker explains:

One or more combined level 1 (L1) caches and shared memory units 373 store graphics data such as texture data, vertex data, pixel data, ray data, bounding volume data, etc., locally within each multi-core group 365A. One or more texture units 374 can also be used to perform texturing operations, such as texture mapping and sampling. A Level 2 (L2) cache 375 shared by all or a

subset of the multi-core groups 365A-365N stores graphics data and/or instructions for multiple concurrent graphics threads. As illustrated, the L2 cache 375 may be shared across a plurality of multi-core groups 365A-365N. One or more memory controllers 367 couple the GPU 380 to a memory 366 which may be a system memory (e.g., DRAM) and/or a dedicated graphics memory (e.g., GDDR6 memory).

EX1031-Koker, 13:55-14:2.

374. Sistla also explains that its shared last level cache is coherent. EX1047-Sistla, [0021]-[0042].

375. Therefore it is my opinion that this limitation is obvious in view the combination of Kelleher, Collins, and Koker.

XII. AVAILABILITY FOR CROSS-EXAMINATION

376. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

XIII. RIGHT TO SUPPLEMENT

377. I reserve the right to supplement my opinions and conclusions to respond to any arguments that the Patent Owner raises, and to take into account new information as it becomes available to me.

XIV. JURAT

378. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: October 15, 2025



Vivek Subramanian, Ph.D

APPENDIX A

Vivek Subramanian

@EPFL:

Rue de la Maladière 71
CH-2000 Neuchâtel,
Switzerland
Vivek.subramanian@epfl.ch
Ph: +41 21 695 42 65

@ Virtual Mailbox

13809 Research Blvd Suite 500
PMB 91298
Austin, TX 78750, USA
viveksub94563@gmail.com
Ph: +1 510 842 7763

@ Berkeley

viveks@eecs.berkeley.edu

Education

3 / 96 - 6 / 98

Ph.D. in Electrical Engineering, Stanford University
Stanford, California

Honors earned: Graduate Fellowship, Eastman Kodak Company

9 / 94 - 3 / 96

MS in Electrical Engineering, Stanford University
Stanford, California

8 / 90 - 5 / 94

BS in Electrical Engineering, Louisiana State University
Baton Rouge, Louisiana

Honors earned: Summa cum laude, College Honors, Outstanding Senior, Junior and Sophomore, Honors College

Professional Experience

2018 - present

École polytechnique fédérale de Lausanne, Switzerland

2021 – present: Director, Micromanufacturing Science and Engineering Center (M2C)

2021 – present: Associate Director, Institute of Electrical and Micro Engineering (IEM)

2021 – present: Scientific Board, Swiss National Strategic Focus Area – Advanced Manufacturing (SFA-AM)

8/18 – present: Professor

2000 - present

Department of Electrical Engineering & Computer Sciences, University of California
Berkeley, CA

7/20 to date : Adjunct Professor

7/11 – 6/20: Professor

7/05 – 7/11: Associate Professor

7/00-7/05: Assistant Professor

Honors Earned:

2015 IEEE Kiyo Tomiyasu Award

Best in session and best in track awards, 2013 IMAPS Microelectronics Conference

Outstanding paper award, 2012 IMAPS Microelectronics Conference

2008 Printed Electronics Champion, Printed Electronics USA Conference

Outstanding Teaching Award, EECS Department, UC Berkeley, 2005

Best paper award, 2004 IEEE Device Research Conference

Nominated to MIT's Technology Review top 100 young innovators list (TR100), 2002

National Science Foundation Young Investigator (CAREER) Award, FY2002,

Nominated to National Academy of Engineering's Frontiers of Engineering, 2002

2002 Paul Rappaport Award for best paper in an IEEE EDS Journal

2009 - 2018

Professor, World Class University Program, Sunchon National University
Sunchon, Korea

Principal Investigator leading large research program focused on developing a new department in the area of printed electronics including 9 faculty covering multiple disciplines and 3 different countries

2000 - present

Independent Consultant

Consultant to the semiconductor industry and its associated fields in the following areas:

Memory technology and design

Silicon Process Technology

Display / Imager and Flexible Electronics Technology

RFID Technology

Intellectual Property Consulting

Technology Evaluation / Venture Capital due diligence

2014 – 2022	Founder and CTO, Locix Inc. San Bruno, CA Senior leadership role in venture-funded startup company working on wireless networking
2004 – 2013	Founding Scientific Advisor, Kovio, Inc. Sunnyvale, CA Scientific advising to printed electronics startup company
2008 – 2011	Chief Technical Advisor, QuSwami, Inc. San Francisco, CA Scientific advising to energy conversion device startup company
1998-2000	Consulting Assistant Professor, Electrical Engineering Department, Stanford University. Stanford, CA Advisory role for research group of Prof. Krishna C. Saraswat
1998-2000	Visiting Research Engineer, Electrical Engineering Department, University of California. Berkeley, CA Research into 25nm MOSFET technologies for giga-scale integration
1998 - 2000	Founder, Matrix Semiconductor, Inc. Santa Clara, CA Co-founder and technical advisor of startup company working on high-density memory technology Honors Earned: Nominated to Scientific American's SA50 List for Visionary Technology Finalist for 2003 World Technology Award for Information Technology Hardware Winner, 2005 EDN Innovation Award
1998	Co-instructor, Electrical Engineering Department, Stanford University. Stanford, CA Co-teaching of EE311, Advanced Integrated Circuit Fabrication Processes
1997	Intern, Advanced Product Research and Development Laboratory, Motorola Inc. Austin, TX Research into process development issues affecting SiGe SEMFET devices
1996	Head Teaching Assistant EE410: IC Fabrication Laboratory, Stanford University Stanford, CA Coordination and instruction of EE410, graduate level laboratory course.
1994-1998	Research Assistant, Electrical Engineering, Stanford University Stanford, CA Research into crystallization of amorphous Si and SiGe films using low thermal budget processes. Professional Affiliations and Activities Program Committee, IEEE Electronic Components and Technology Conference, 2014 to 2016 Chair, Scientific Board, iPACK, Royal Institute of Technology (KTH), Sweden, 2011 to 2016 Tampere Institute of Technology Faculty Search Committee, Finland, 2013 University of Oulu Faculty Search Committee, Finland, 2013 Served as external thesis committee member for several universities world-wide, including University of Cape Town (South Africa), Tampere Institute of Technology (Finland), Technical University of Eindhoven (Netherlands), Indian Institute of Science (India), Indian Institute of Technology (India), 2009 to date Associate Editor, IEEE Journal of Display Technology, 2008 to date Technical Program Chair, Large Area, Organic, and Printed Electronics Conference, 2012-2013 Scientific Committee, Large Area, Organic, and Printed Electronics Conference, 2009-2011 Scientific Committee, International Conference on Printed and Flexible Electronics, 2009-2013 IEEE Electron Devices Society Organic Electronics Committee, 2003 - 2005 Executive Committee, International Electron Device Meeting, 2003 to 2009 Technical Program Committee, International Electron Device Meeting, 2001-2002 Technical Program Committee member, Device Research Conference, 2000-2002 Technical Program Committee member, VLSI-TSA Conference, 2005 Member, Institute of Electrical and Electronic Engineers

List of cases with testimony / deposition by Vivek Subramanian

- Lexar Media v. Pretec Electronics Corp., et al (N.D. Cal., Case No. 00-CV-4770 MJJ, deposition) – retained by Pretec Electronics Corp, PNY Electronics, Memorex
- Motorola, Inc. v. Analog Devices, Inc., Civil Action No. 1:03-CV-0131 – retained by Analog Devices
- *Sandisk Corporation vs. Micron*, Case No. CV-01-3855 CW – retained by Micron
- In the Matter of CERTAIN NAND FLASH MEMORY CIRCUITS AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-526, deposition and trial) – retained by STMicroelectronics
- In the Matter of CERTAIN NAND FLASH MEMORY DEVICES AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-553, deposition and trial) – retained by Hynix
- In the Matter of CERTAIN NOR AND NAND FLASH MEMORY DEVICES AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-560, deposition and trial) – retained by STMicroelectronics
- STMicroelectronics, Inc. v. SanDisk Corporation (E.D. Texas, Civil Action No. 4:05-CV-45, deposition) – retained by STMicroelectronics
- Sandisk Corporation v. Memorex Products, Inc. et al, CV 01-4063 – retained by Memorex.
- Seagate Technology LLC et al v. STEC, Inc., Case No. 5:08-CV-01950 JW (HRL) – retained by STEC.
- In the Matter of CERTAIN NAND FLASH MEMORY DEVICES AND COMPONENTS THEREOF, AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-592) – retained by Hynix
- In the Matter of CERTAIN FLASH MEMORY CONTROLLERS, DRIVES, MEMORY CARDS, AND MEDIA PLAYERS AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-619, deposition and trial) – retained by Phison Electronics, Silicon Motion, Kingston, and Apacer
- Harari and Mehrotra v. Hollmer and Cleveland (Board of Patent Appeals and Interferences, Interference# 105,606, deposition) – retained by Spansion

- In the Matter of CERTAIN SEMICONDUCTOR INTEGRATED CIRCUITS USING TUNGSTEN METALLIZATION AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-648) – retained by STMicroelectronics
- In the Matter of CERTAIN SEMICONDUCTOR CHIPS HAVING SYNCHRONOUS DYNAMIC RANDOM ACCESS MEMORY CONTROLLERS AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-661, deposition and trial) – retained by NVidia
- Fast memory erase, LLC. v. Spansion, Inc. et al, Civil Action No. 3:08-CV-00977-M, retained by Spansion
- In the Matter of CERTAIN MOBILE TELEPHONES AND WIRELESS COMMUNICATIONS DEVICES FEATURING DIGITAL CAMERAS AND COMPONENTS THEREOF (ITC, Investigation No. 337-TA-663, deposition and trial) – retained by LG Electronics and Samsung Electronics
- In the Matter of CERTAIN MLC FLASH MEMORY DEVICES AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-683, deposition and trial) – retained by Samsung
- In the Matter of CERTAIN MOBILE TELEPHONES WIRELESS COMMUNICATION DEVICES FEATURING DIGITAL CAMERAS, AND COMPONENTS THEREOF (ITC, Investigation No. 337-TA-703, deposition and trial) – retained by Apple
- Vizio, Inc. v. L Electronics, Inc., et al, In the United States District Court For The District of Maryland, Civil action No. 1:09-CV-1481 (BEL), deposition – retained by LG
- In the Matter of CERTAIN ELECTRONIC DEVICES, INCLUDING MOBILE PHONES, PORTABLE MUSIC PLAYERS, AND COMPUTERS (ITC, Investigation No. 337-TA-701, deposition and trial) – retained by Apple
- Nokia v. Apple, 09-CV-791 and v09-CV-1002 – retained by Apple
- Innovative Memory Systems, Inc. v. Micron Tech, Inc. 14-cv-01480 and associated IPRs, retained by Micron (deposition)
- SanDisk Corporation v. Phison Electronics Corp., et al, In the United States District Court For The Western District of Wisconsin, Civil action No. 07-C-0605-C and 07-C-0607-C, deposition – retained by Phison Electronics Corp., Silicon Motion, Skymedi, and Kingston
- Microchip Technology Inc v. LSI Corporation and Agere Systems Inc., Case No. 2:11-cv-01326-DGC, retained by Microchip

- In the Matter of CERTAIN INTEGRATED CIRCUITS, CHIPSETS, AND PRODUCTS CONTAINING SAME INCLUDING TELEVISIONS, MEDIA PLAYERS AND CAMERAS (ITC, Investigation No. 337-TA-709, deposition and trial) – retained by Freescale Semiconductor, Inc.
- In the Matter of CERTAIN SEMICONDUCTOR PRODUCTS MADE BY ADVANCED LITHOGRAPHY TECHNIQUES AND PRODUCTS CONTAINING SAME, (ITC, Investigation No. 337-TA-729) – retained by Samsung
- In the Matter of CERTAIN MOBILE DEVICES AND RELATED SOFTWARE (ITC, Investigation No 337-TA-750, deposition and trial) – retained by Apple, Inc.
- In the Matter of CERTAIN SEMICONDUCTOR CHIPS AND PRODUCTS CONTAINING SAME (ITC, Investigation No 337-TA-753) – retained by NVidia
- In the Matter of CERTAIN MOBILE TELEPHONES AND MODEMS (ITC, Investigation No. 337-TA-758) – retained by LG.
- In the Matter of CERTAIN VIDEO GAME SYSTEMS AND WIRELESS CONTROLLERS AND COMPONENTS THEREOF (ITC, Investigation No. 337-TA-770, deposition and trial) – retained by Nintendo
- In the Matter of CERTAIN RANDOM ACCESS MEMORY DEVICES AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-595) retained by Renesas
- In the Matter of CERTAIN MICROPROCESSORS, COMPONENTS THEREOF, AND PRODUCTS CONTAINING SAME (ITC, Investigation No. 337-TA-781, deposition and trial) – retained by Intel
- In the Matter of CERTAIN DIGITAL TELEVISIONS CONTAINING INTEGRATED CIRCUIT DEVICES AND COMPONENTS THEREOF (ITC, Investigation No. 337-TA-806, deposition) – retained by Renesas
- In the Matter of CERTAIN INTEGRATED CIRCUITS, CHIPSETS, AND PRODUCTS CONTAINING SAME INCLUDING TELEVISIONS (ITC, Investigation No. 337-TA-786, deposition and trial) – retained by Freescale
- In the Matter of CERTAIN SEMICONDUCTOR INTEGRATED CIRCUIT DEVICES AND PRODUCTS CONTAINING SAME (ITC Investigation No. 337-TA-840) – retained by Microchip.
- In the Matter of CERTAIN DYNAMIC RANDOM ACCESS MEMORY AND NAND FLASH MEMORY DEVICES AND PRODUCTS CONTAINING THE SAME (ITC, Investigation No. 337-TA-803, deposition) – retained by Hynix

- AVM Technologies LLC v. Intel Corporation, Civil Action No. 10-610-RK – retained by Intel
- Round Rock Research LLC v. Dell Inc., No. 11-CV-976-RGA – retained by Dell
- Kingston Technology Company v. Polaris Innovations Ltd, IPRs – retained by Kingston (deposition)
- Solid State Storage Solutions, Inc. v. STEC, Inc., OCZ Technology, Inc. Corsair Memory, Texas Memory Systems, Inc. PNY Technologies, Inc., Patriot Memory LLC, Fusion-IO, Inc., Other World Computing, Inc., and Mushkin, Inc., Civil Action No 2:11-CV-391, retained by Corsair, PNY, and Fusion-IO.
- Grail Semiconductor, Inc., v. Renesas Electronics America, Inc., Case No. C 11-03847, retained by Renesas.
- In The Matter of Certain Wireless Consumer Electronics Devices and Components Thereof (ITC, Inv. No. 337-TA-853), retained by Amazon.com, Acer, Barnes & Noble, Garmin, HTC, Huawei, Kyocera, LG, Nintendo, Novatel, Samsung, and ZTE.
- Invensas Corporation v Renesas Electronics Corporation, Civil Action No. 11-448-GMS-CJB, retained by Renesas (deposition)
- In the Matter of Certain Non-volatile Memory Chips and Products Containing Same (ITC, Inv. No. 337-TA-916), retained by Macronix (deposition)
- Sandisk Corporation v. PNY Technologies, Inc., Case No. 1-11-CV-205928, retained by PNY (deposition)
- *TPK Touch Solutions, Inc. v. Wintek Electro-Optics Corporation, et al.*, 3:13-cv-02218-NC (N.D. Cal.), and associated issues in front of the patent board, retained by Wintek (deposition)
- GSI Technology, Inv. V. Cypress Semiconductor Corporation, IPR2014-00121 and associated issues in front of the patent board, retained by Cypress (deposition)
- Grail Semiconductor, Inc. v Mitsubishi, Case No. 1-07-CV-098590, retained by Mitsubishi
- AVM Technologies, LLC v. Intel Corporation, C.A. No. 15-33-RGA, retained by Intel (deposition and testimony)
- Intellectual Ventures LLV v. Toshiba, C.A. No. 13-453-SLR-SRF, retained by Toshiba (deposition and testimony)

- In the Matter of CERTAIN MEMORY MODELS AND COMPONENTS THEREOF, AND PRODUCTS CONTAINING SAME, US ITC 337-TA-1023, retained by Hynix (deposition and testimony)
- In the Matter of CERTAIN INTEGRATED CIRCUITS WITH VOLTAGE REGULATORS AND PRODUCTS CONTAINING THE SAME, US ITC 337-TA-1024, retained by Intel (deposition)
- Technology Properties Ltd, LLC v. LG et al, Case No. 3:12-cv-03880-VC, retained by defendants.
- Longitude Licensing Ltd. V Apple, Case 3:14-CV-04275-EDL, retained by Apple.
- KAIST IP v. Samsung et al, 2:16-cv-01314-JRG-RSP – retained by Samsung (deposition and trial).
- Toshiba Corporation v Lone Star Silicon Innovations LLC, IPR – retained by Toshiba
- Samsung Electronics Co Ltd. Vs PROMOS Technologies Inc, IPRs – retained by Samsung (deposition)
- Samsung Electronics Co. Ltd vs. UUSI LLC d/b/a Narton, IPRs – retained by Samsung (deposition)
- Carl Zeiss AG and ASML v Nikon, 2:17-cv-03221 RGK – retained by ASML and Zeiss (deposition and trial testimony)
- In the matter of CERTAIN MEMORY MODULES AND COMPONENTS THEREOF (ITC, Inv. No. 337-TA-1089), retained by SK Hynix (deposition and trial testimony)
- In the Matter of CERTAIN SOLID STATE STORAGE DRIVES, STACKED ELECTRONICS COMPONENTS, AND PRODUCTS CONTAINING SAME, US ITC 337-TA-1097, retained by Samsung and Hynix (deposition)
- In the Matter of CERTAIN INFOTAINMENT SYSTEMS, COMPONENTS THEREOF, AND AUTOMOBILES CONTAINING THE SAME, US ITC 337-TA-1119, retained by Renesas, Denso, Pioneer, Panasonic, Socionext, and Toyota (deposition and trial testimony)
- In the Matter of CERTAIN MOVABLE BARRIER OPERATOR SYSTEMS AND COMPONENTS THEREOF, US ITC 337-TA-1118, retained by Chamberlain (deposition and trial testimony)
- X2Y ATTENUATORS, LLC v. INTEL CORPORATION, 3:18-cv-1394-HZ, retained by Intel (deposition)

- In the Matter of CERTAIN INTEGRATED CIRCUITS AND PRODUCTS CONTAINING THE SAME, US ITC 337-TA-1148, retained by Intel (deposition and trial testimony)
- In the Matter of CERTAIN TOUCH-CONTROLLED MOBILE DEVICES, COMPUTER AND COMPONENTS THEREOF, US ITC 337-TA-1162, retained by Samsung, HP, Dell, Microsoft, Lenovo, and Motorola (deposition)
- CONTINENTAL AUTOMOTIVE GMBH v. ATMEL CORP., ICC CASE NO. 22521/FS, retained by Atmel
- INTEL CORPORATION, v. TELA INNOVATIONS, INC., CASE NO. 18-CV-02848-WHO, retained by Intel (deposition)
- RENESAS ELECTRONICS CORPORATION, Petitioner v. BROADCOM CORPORATION Patent Owner, Inter Partes Review No. IPR2019-01041, retained by Renesas (deposition)
- Applied Materials et al. v. Demaray LLC Inter Partes Review Matters IPR2021-00103, PIR2021-00104, retained by Applied Materials et al (deposition)
- SAMSUNG ELECTRONICS CO., LTD., v. TRENCHANT BLADE TECHNOLOGIES LLC, IPR2021-00258, retained by Samsung (deposition)
- KIOXIA CORPORATION, KIOXIA AMERICA, INC., WESTERN DIGITAL TECHNOLOGIES, INC., AND DELL TECHNOLOGIES, INC. v. SONRAI MEMORY LIMITED, Various IPRs, retained by Kioxia and Western Digital (deposition).
- In the Matter of CERTAIN MOBILE PHONES, TABLET COMPUTERS, SMART WATCHES, SMART SPEAKERS, AND DIGITAL MEDIA PLAYERS, AND PRODUCTS CONTAINING SAME, US ITC, 337-TA-1300, retained by Apple (deposition)
- IMPINJ, INC., v. NXP B.V., IPR2020-01630 and Case No. 4:19-cv-03161-YGR, retained by NXP (depositions and trial)
- VLSI Technology LLC v. Intel Corporation, 1:18-cv-00966-CFC and associated matters, retained by Intel (deposition)
- Intel Corporation v. ParkerVision, Inc. Inter Partes Review Case IPR2020-01265 and associated matters, retained by Intel (deposition)
- In the Matter of CERTAIN POWER SEMICONDUCTORS, AND MOBILE DEVICES AND COMPUTERS CONTAINING SAME, US ITC Inv. No. 337-TA-1308, retained by Apple (deposition and trial)

- In the Matter of CERTAIN INTEGRATED CIRCUITS, COMPONENTS THEREOF, AND PRODUCTS CONTAINING THE SAME, US ITC Inv. No. 337-TA-1350 and related matters, retained by AMD (deposition and trial)
- SAMSUNG ELECTRONICS CO., LTD., v. NETLIST, INC., IPR2022-00063 and associated matters, retained by Samsung (deposition).

APPENDIX B

Appendix B: Materials Considered

Exhibit	Description
1001	U.S. Patent No. 11,841,803 (“’803 patent”)
1003	File History for U.S. Patent No. 11,841,803 (“FH”)
1004	U.S. Patent No. 7,598,958 (“Kelleher”)
1005	U.S. Patent No. 11,610,862 (“Collins”)
1006	Coskun et. al, A Cross-Layer Methodology for Design and Optimization of Networks in 2.5D Systems, PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN (2018) (“Coskun”)
1008	Jerger et al., NoC Architectures for Silicon Interposer Systems, IEEE/ACM INTERNATIONAL SYMPOSIUM ON MICROARCHITECTURE (2015) (“Jerger”)
1010	John Lau, Heterogeneous Integrations, Springer (2019) (“Lau”)
1011	Ajaykumar Kannan, Enabling Interposer-Based Disintegration of Multi-Core Processors, University of Toronto Department of Electrical & Computer Engineering (2015) (“Kannan”)
1014	International Patent Publication No. WO2019132971 (“Elsherbini”)
1020	Dylan Stow, et. al., Cost-Effective Design of Scalable High-Performance Systems Using Active and Passive Interposers, IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN (ICCAD) (2017) (“Stow”)
1021	The Open Domain-Specific Architecture WorkGroup, A Chiplet-Based Open Architecture, Draft v0.9 (2018) (“ODSA”)
1022	Ugljesa Milic et. al., BEYOND THE SOCKET: NUMA-AWARE GPUS, <i>in</i> PROCEEDINGS OF MICRO-50 (2017) (“Milic”)
1023	Complaint, <i>In the Matter of</i> CERTAIN INTEGRATED CIRCUITS, ELECTRONIC DEVICES CONTAINING THE SAME, AND COMPONENTS THEREOF, Inv. No. 337-TA-1450 (“ITC-Complaint”)

Subramanian Declaration: Appendix B (Materials Considered)

Exhibit	Description
1024	Complaint, <i>Onesta IP, LLC v. NVIDIA Corp</i> , No. 1:25-cv-00586 (W.D. Tex.) (“District-Court-Complaint”)
1025	ITC Exhibit 50, <i>In the Matter of</i> CERTAIN INTEGRATED CIRCUITS, ELECTRONIC DEVICES CONTAINING THE SAME, AND COMPONENTS THEREOF, Inv. No. 337-TA-1450 (“ITC-Ex50-Infringement”)
1026	ITC Exhibit 19, <i>In the Matter of</i> CERTAIN INTEGRATED CIRCUITS, ELECTRONIC DEVICES CONTAINING THE SAME, AND COMPONENTS THEREOF, Inv. No. 337-TA-1450 (“ITC-Ex19-Domestic-Industry”)
1027	U.S. Patent Publication No. 2019/0123022 (“Teig”)
1028	Paul Enquist, Room Temperature Direct Wafer Bonding for Three Dimensional Integrated Sensors, <i>SENSORS AND MATERIALS</i> , Vol. 17, No. 6 (2005) (“Ziptronix”)
1029	Robert R. Schaller, Moore’s Law: Past, Present, and Future, <i>IEEE SPECTRUM</i> (June 1997) (“Schaller”)
1030	Yangdong (Steven) Deng & Wojciech Maly, 2.5D System Integration: A Design Driven System Implementation Schema, <i>IEEE</i> (2004) (“Deng-&-Maly”)
1031	U.S. Patent No. 10,909,652 (“Koker”)
1033	Akhil Arunkumar et. al, MCM-GPU: Multi-Chip-Module GPUs for Continued Performance Scalability <i>in Proceedings of ISCA ’17</i> (June 2017) (“Arunkumar”)
1034	Dictionary of Computer Science (7th ed.), Oxford University Press (2016) (“Dictionary-of-Computer-Science”)
1035	Dictionary of Multimedia (4th ed.), Franklin, Beedle & Associates Inc. (2005) (“Dictionary-of-Multimedia”)
1036	Joel Mearig & Bill Goers, An Overview of Manufacturing BGA Technology, <i>IEEE/CPMT Int’l Electronics Manufacturing Technology Symposium</i> (1995) (“Mearig-&-Goers”)
1037	U.S. Patent Publication No. 2018/0240778 (“Liu”)

Exhibit	Description
1038	Farhang Yazdani, Design and Direct Assembly of 2.5D/3D Rigid Silicon Interposer on PCB, IMAPS Proceedings (2014) (“Yazdani”)
1040	U.S. Patent Publication No. 2001/0005873 (“Yasuda”)
1041	U.S. Patent No. 10,475,147 (“Price”)
1042	U.S. Patent Publication No. 2007/0273699 (“Sasaki”)
1043	Amended Joint Proposed Claim Construction Chart, <i>In the Matter of</i> CERTAIN INTEGRATED CIRCUITS, ELECTRONIC DEVICES CONTAINING THE SAME, AND COMPONENTS THEREOF, Inv. No. 337-TA-1450 (“Claim-Construction-Chart”)
1044	Chris McClanahan, History and Evolution of GPU Architecture, Georgia Tech. (2010) (“McClanahan”)
1045	Thiruvengadam Vijayaraghavan et.al, Design and Analysis of an APU for Exascale Computing, IEEE International Symposium on High Performance Computer Architecture (2017) (“Vijayaraghavan”)
1046	NVIDIA Tesla P100, Whitepaper, The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World’s Fastest GPU (2016) (“NVIDIA-P100”)
1047	U.S. Patent Publication No. 2009/0037658 (“Sistla”)
1048	Raman Nayyar, “Performance Analysis of a Hierarchical, Cache-Coherent, Shared Memory Based, Multi-processor System” (1993) (“Nayyar”)
1049	Rajeev Balasubramonian et. al, “Multi-Core Cache Hierarchies,” Morgan & Claypool Publishers (2011) (“Balasubramonian”)
1050	Reza Ghaffarian, Book of Knowledge (BOK) for NASA Electronic Packaging Roadmap (2015) (“Ghaffarian”)

APPENDIX C

Appendix C (Challenged Claims)

Ground	35 U.S.C. Section	Claims	References
I	103	1-4, 7, 10	Kelleher, Collins
II	103	8, 9	Kelleher, Collins, Koker, Sistla

Claim No.	Challenged Claim Language
[1.pre]	A system, comprising:
[1.a]	a central processing unit (CPU) communicably coupled to a first graphics processing unit (GPU) chiplet of a GPU chiplet array,
[1.b]	wherein the GPU chiplet array includes: the first GPU chiplet communicably coupled to the CPU via a bus; and
[1.c]	[wherein the GPU chiplet array includes:] a second GPU chiplet communicably coupled to the first GPU chiplet via a passive crosslink,
[1.d]	wherein the passive crosslink is dedicated for inter-chiplet communications.
2	The system of claim 1, wherein the passive crosslink comprises a passive interposer die.
[3.a]	The system of claim 1, wherein the first GPU chiplet comprises a first PHY region including conductor structures for chiplet-to-chiplet communications,
[3.b]	and further wherein the second GPU chiplet comprises a second PHY region including conductor structures for chiplet-to-chiplet communications.
[4.a]	The system of claim 3, further comprising: a third GPU chiplet communicably coupled to the first GPU chiplet via the passive crosslink,
[4.b]	wherein the third GPU chiplet comprises a third PHY region including conductor structures for chiplet-to-chiplet communications.
7	The system of claim 1, wherein the passive crosslink communicably couples all GPU chiplets in the GPU chiplet array.

Claim No.	Challenged Claim Language
[8.a]	The system of claim 1, further comprising: a first cache memory hierarchy at the first GPU chiplet,
[8.b]	wherein a first level of the first cache memory hierarchy is coherent within the first GPU chiplet; and
[8.c]	a second cache memory hierarchy at the second GPU chiplet,
[8.d]	wherein a first level of the second cache memory hierarchy is coherent within the second GPU chiplet.
[9.a]	The system of claim 8, further comprising: a unified cache memory including both a last level of the first cache memory hierarchy and a last level of the second cache memory hierarchy,
[9.b]	wherein the unified cache memory is coherent across all GPU chiplets of the GPU chiplet array.
10	The system of claim 1, further comprising: a plurality of conductive pillars coupling a circuit board substrate to a first non-PHY region of the first GPU chiplet and a second non-PHY region of the second GPU chiplet.