EXHIBIT 803-A-04

U.S. Patent No. 10,909,652 to Altug Koker et al. ("Koker")

U.S. Patent No. 10,909,652 to Altug Koker et al. ("Koker") anticipates and/or renders obvious (alone or in combination with other references), including at least under Complainant's apparent infringement theory:

- claims 1-3, 7-10 of U.S. Patent No. 11,841,803 ("the '803 patent") under 35 U.S.C. §§ 102 and/or 103.

NVIDIA has charted the portions of the reference that correspond to each element, including at least under Complainant's apparent interpretation of the claims as reflected in Complainant's infringement contentions. Where the chart below states that the reference "discloses," "teaches," or "suggests" a claim limitation, such disclosure, teaching, or suggestion may be express, implied, or inherent. The cited evidence is merely illustrative, and NVIDIA reserves the right to cite alternative or additional evidence.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the asserted patents. In addition, nothing stated in this chart shall be treated as an admission or suggestion that NVIDIA agrees with Complainant regarding either the scope of any of the asserted claims or Complainant's apparent interpretation of the scope of the asserted claims, or that NVIDIA's accused products meet any limitations of the claims.

To the extent that Complainant contends that Koker does not disclose one or more limitations of the asserted claims, it would have been obvious to combine the teachings of Koker with (1) the knowledge of one of ordinary skill in the art, and/or (2) the teachings of any of the prior art set forth in NVIDIA's other invalidity charts and cover pleading, to show all the limitations of the asserted claims, including for the reasons set forth in NVIDIA's Invalidity Contention Cover Pleading, which are incorporated herein.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
1[Pre]: A system, comprising:	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See* evidence cited for '803 patent claims 1-3, 7-10. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
1[A]: a central processing unit (CPU) communicably coupled to a first graphics processing unit (GPU) chiplet of a GPU chiplet array,	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See, e.g.*, Koker at Fig. 1.

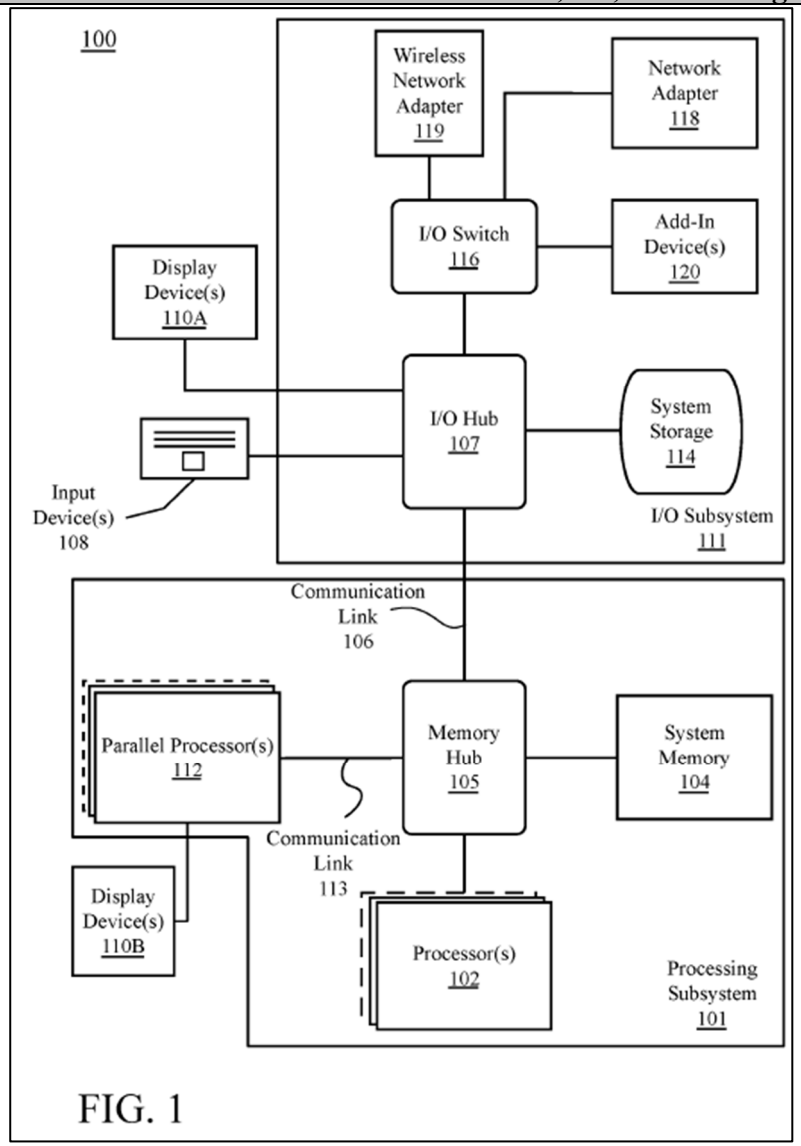
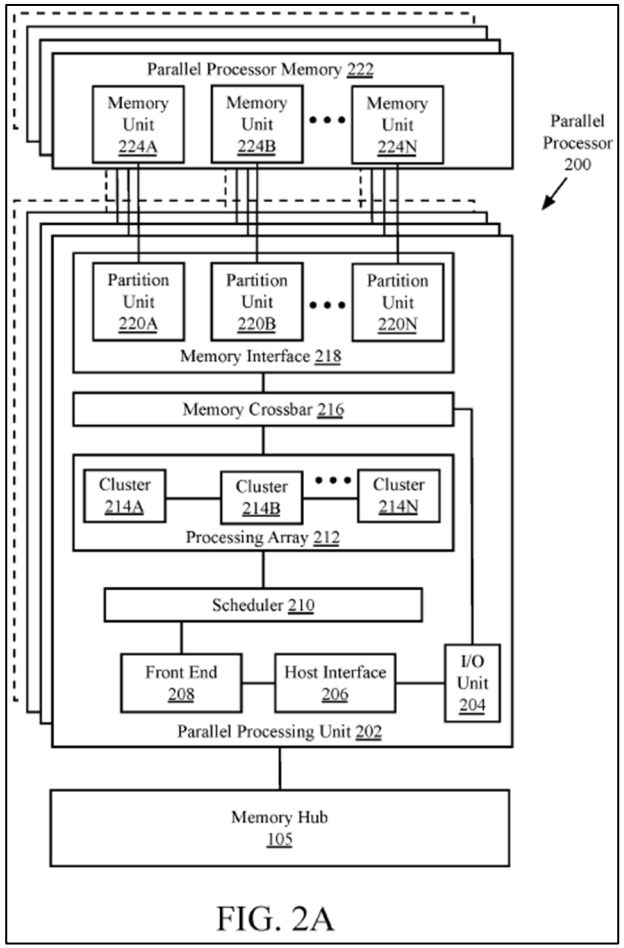
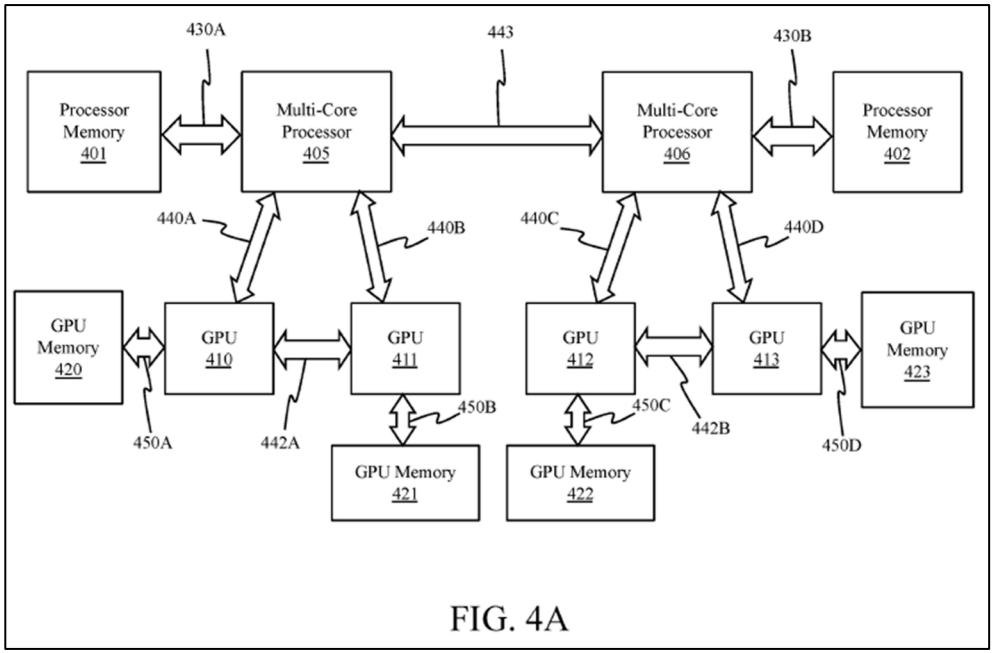
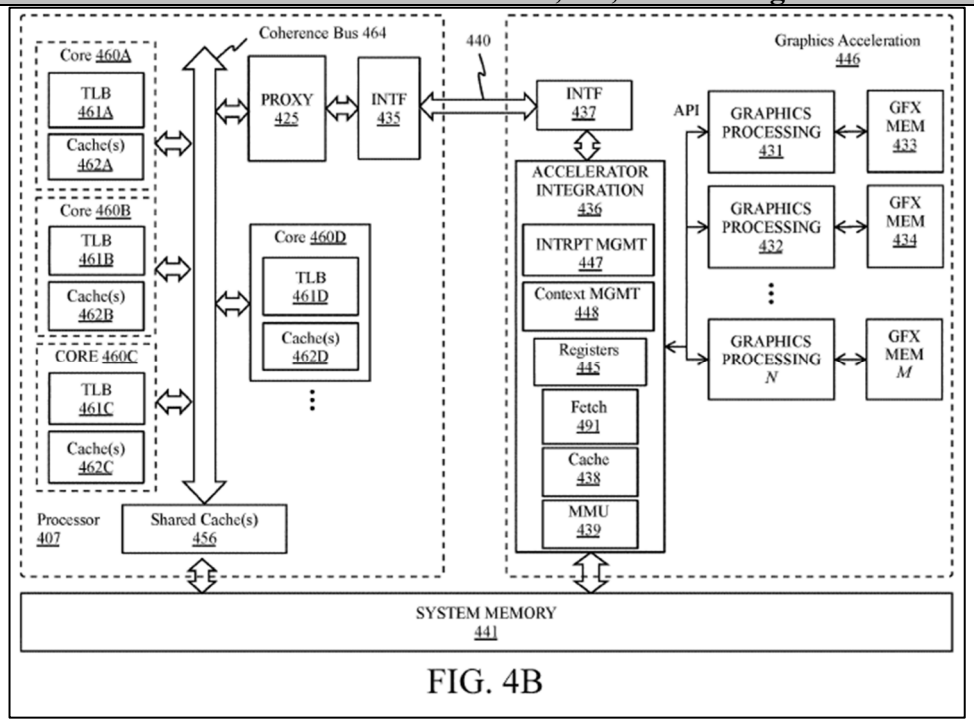
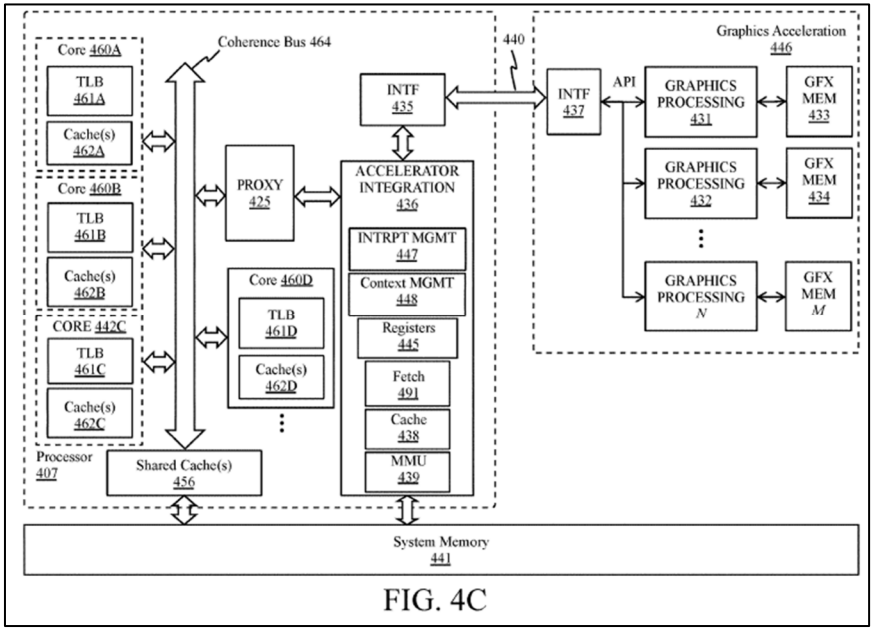
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	

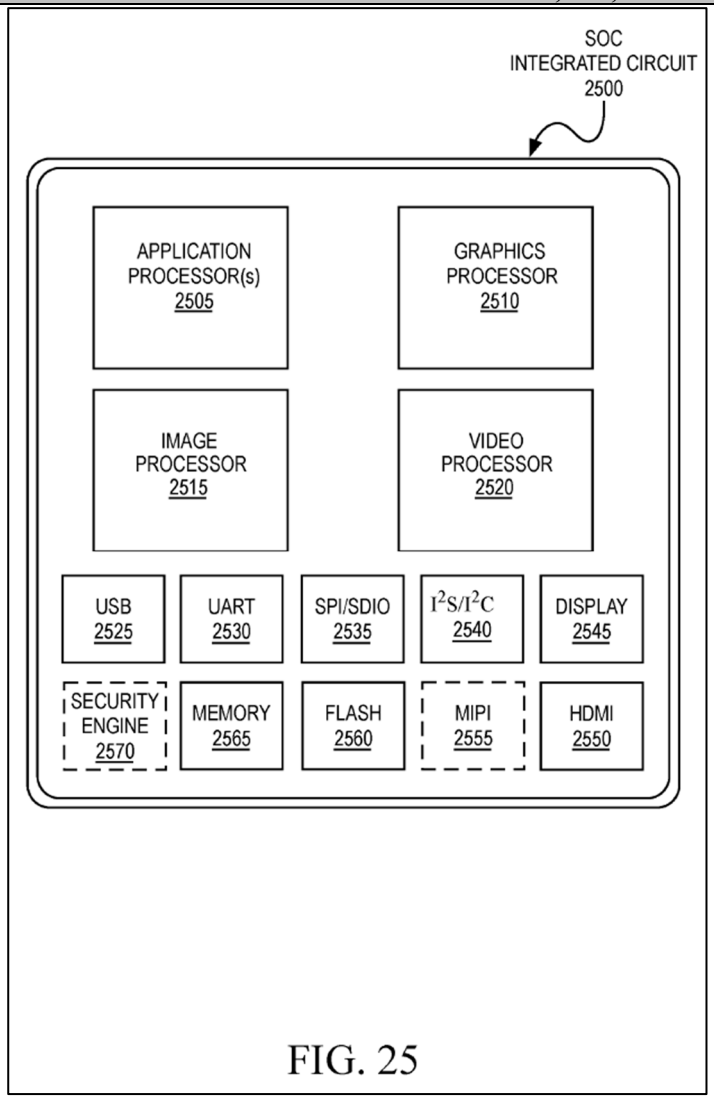
FIG. 1

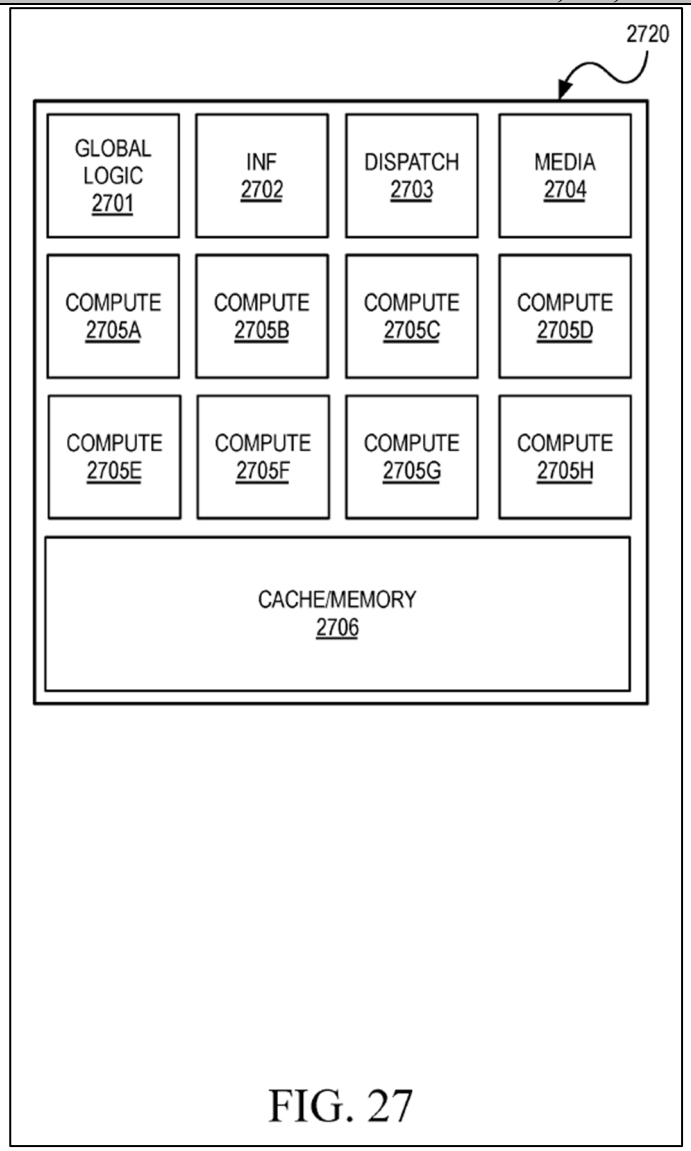
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at Fig. 2A.  FIG. 2A Koker at Fig. 4A.

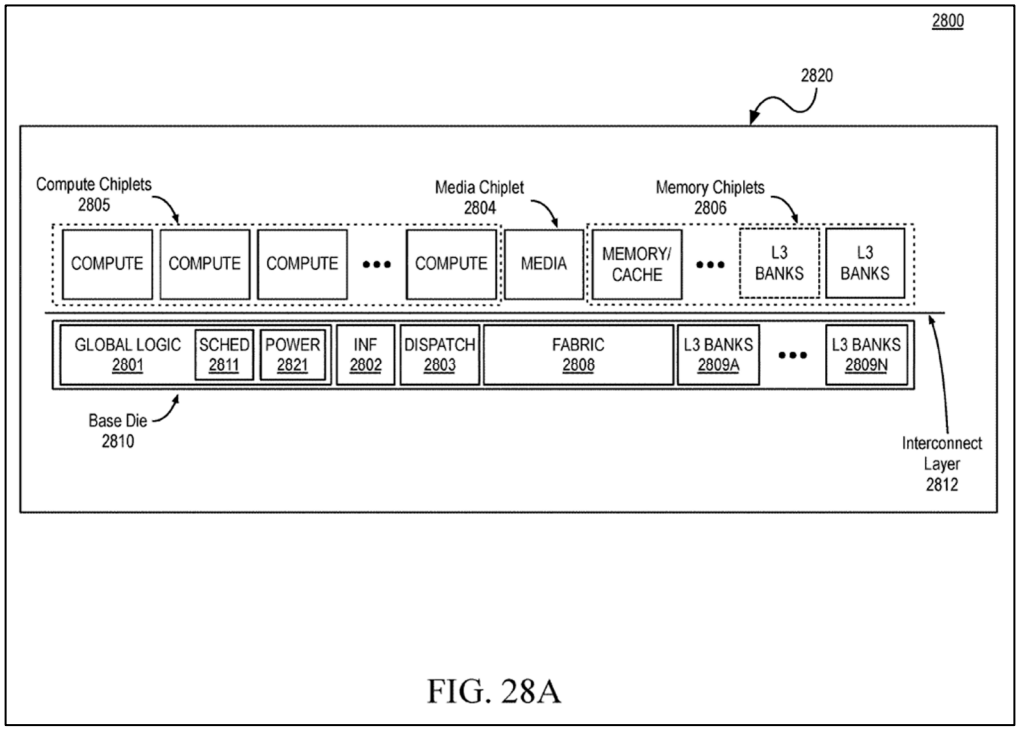
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4A Koker at Fig. 4B.

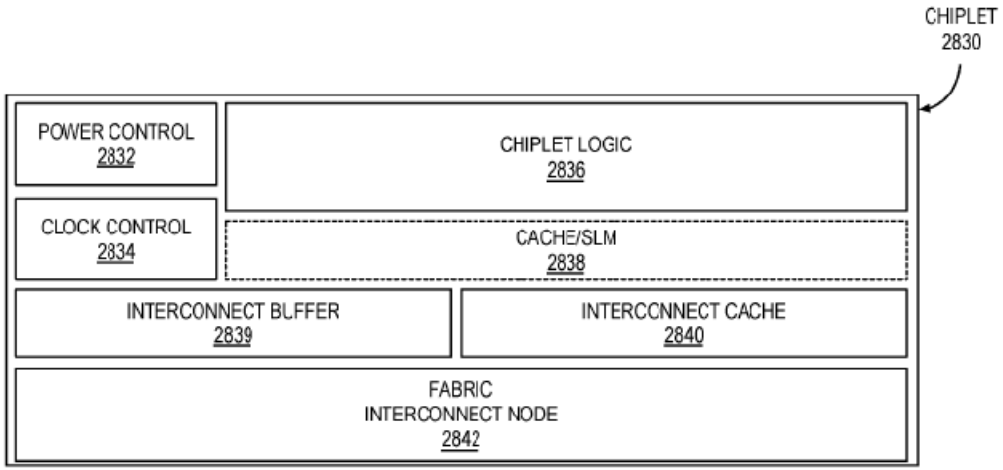
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4B Koker at Fig. 4C.

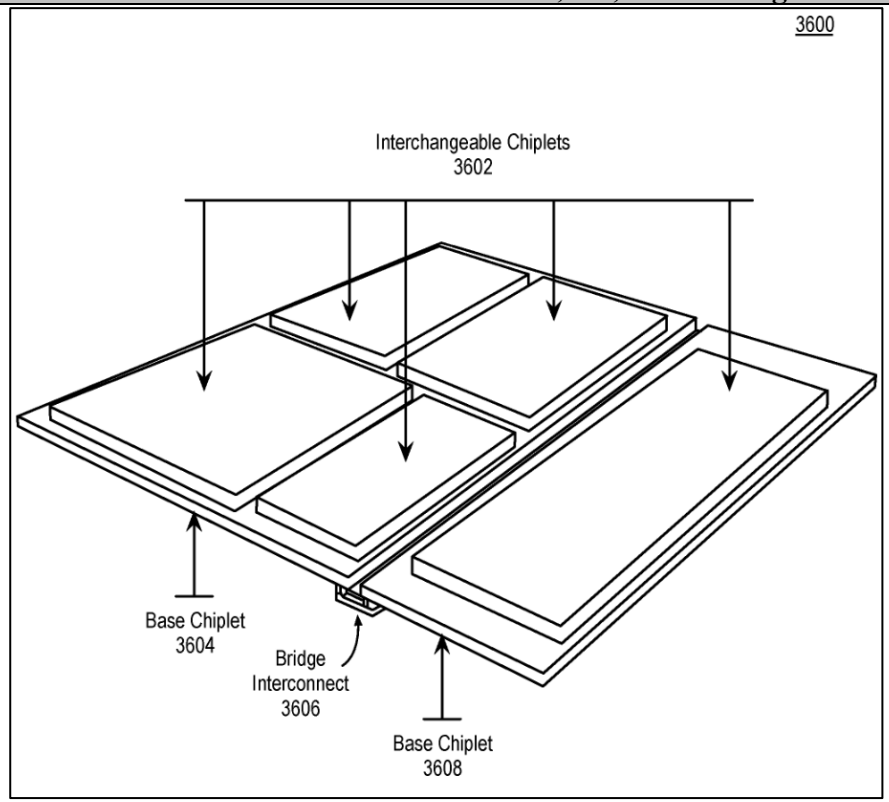
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4C Koker at Fig. 25.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 25 Koker at Fig. 27.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 27

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at Fig. 28A.  FIG. 28A Koker at Fig. 28B.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 28B Koker at Fig. 35

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 Koker at Abstract ("A disaggregated processor package can be configured to accept interchangeable chiplets. Interchangeability is enabled by specifying a standard physical interconnect for chiplets that can enable the chiplet to interface with a fabric or bridge interconnect. Chiplets from different IP designers can conform to the common interconnect, enabling such chiplets to be interchangeable during assembly. The fabric and bridge interconnects logic on the chiplet can then be configured to confirm with the actual interconnect layout of the on-board logic of the chiplet. Additionally, data from chiplets can be transmitted across an inter-chiplet fabric using encapsulation, such that the actual data being transferred is opaque to the fabric, further enable interchangeability of the individual chiplets. With such an interchangeable design, higher or lower density memory can be inserted into memory chiplet

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	slots, while compute or graphics chiplets with a higher or lower core count can be inserted into logic chiplet slots.") Koker at 3:20-35 ("In some embodiments, a graphics processing unit (GPU) is communicatively coupled to host/processor cores to accelerate graphics operations, machine-learning operations, pattern analysis operations, and various general-purpose GPU (GPGPU) functions. The GPU may be communicatively coupled to the host processor/cores over a bus or another interconnect (e.g., a high-speed interconnect such as PCIe or NVLink). In other embodiments, the GPU may be integrated on the same package or chip as the cores and communicatively coupled to the cores over an internal processor bus/interconnect (i.e., internal to the package or chip). Regardless of the manner in which the GPU is connected, the processor cores may allocate work to the GPU in the form of sequences of commands/instructions contained in a work descriptor. The GPU then uses dedicated circuitry/logic for efficiently processing these commands/instructions.") Koker at 3:44-61 ("FIG. 1 is a block diagram illustrating a computing system 100 configured to implement one or more aspects of the embodiments described herein. The computing system 100 includes a processing subsystem 101 having one or more processor(s) 102 and a system memory 104 communicating via an interconnection path that may include a memory hub 105. The memory hub 105 may be a separate component within a chipset component or may be integrated within the one or more processor(s) 102. The memory hub 105 couples with an I/O subsystem 111 via a communication link 106. The I/O subsystem 111 includes an I/O hub 107 that can enable the computing system 100 to receive input from one or more input device(s) 108. Additionally, the I/O hub 107 can enable a display controller, which may be included in the one or more processor(s) 102, to provide outputs to one or more display device(s) 110A. In one embodiment the one or more display device(s) 110A coupled with the I/O hub 107 can include a local, internal, or embedded display device.") Koker at 4:27-37 ("The computing system 100 can include other components not explicitly shown, including USB or other port connections, optical storage drives, video capture devices, and the like, may also be connected to the I/O hub 107. Communication paths interconnecting the various components in FIG. 1 may be implemented using any suitable protocols, such as PCI (Peripheral Component Interconnect) based protocols (e.g., PCI-Express), or any other bus or

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	point-to-point communication interfaces and/or protocol(s), such as the NV-Link high-speed interconnect, or interconnect protocols known in the art.") Koker at 4:38-57 ("In one embodiment, the one or more parallel processor(s) 112 incorporate circuitry optimized for graphics and video processing, including, for example, video output circuitry, and constitutes a graphics processing unit (GPU). In another embodiment, the one or more parallel processor(s) 112 incorporate circuitry optimized for general purpose processing, while preserving the underlying computational architecture, described in greater detail herein. In yet another embodiment, components of the computing system 100 may be integrated with one or more other system elements on a single integrated circuit. For example, the one or more parallel processor(s) 112, memory hub 105, processor(s) 102, and I/O hub 107 can be integrated into a system on chip (SoC) integrated circuit. Alternatively, the components of the computing system 100 can be integrated into a single package to form a system in package (SIP) configuration. In one embodiment at least a portion of the components of the computing system 100 can be integrated into a multi-chip module (MCM), which can be interconnected with other multi-chip modules into a modular computing system.") Koker at 5:43-63 ("When the host interface 206 receives a command buffer via the I/O unit 204, the host interface 206 can direct work operations to perform those commands to a front end 208. In one embodiment the front end 208 couples with a scheduler 210, which is configured to distribute commands or other work items to a processing cluster array 212. In one embodiment the scheduler 210 ensures that the processing cluster array 212 is properly configured and in a valid state before tasks are distributed to the processing clusters of the processing cluster array 212. In one embodiment the scheduler 210 is implemented via firmware logic executing on a microcontroller. The microcontroller implemented scheduler 210 is configurable to perform complex scheduling and work distribution operations at coarse and fine granularity, enabling rapid preemption and context switching of threads executing on the processing array 212. In one embodiment, the host software can prove workloads for scheduling on the processing array 212 via one of multiple graphics processing doorbells. The workloads can then be automatically distributed across the processing array 212 by the scheduler 210 logic within the scheduler microcontroller.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 5:64-6:12 ("The processing cluster array 212 can include up to "N" processing clusters (e.g., cluster 214A, cluster 214B, through cluster 214N). Each cluster 214A-214N of the processing cluster array 212 can execute a large number of concurrent threads. The scheduler 210 can allocate work to the clusters 214A-214N of the processing cluster array 212 using various scheduling and/or work distribution algorithms, which may vary depending on the workload arising for each type of program or computation. The scheduling can be handled dynamically by the scheduler 210, or can be assisted in part by compiler logic during compilation of program logic configured for execution by the processing cluster array 212. In one embodiment, different clusters 214A-214N of the processing cluster array 212 can be allocated for processing different types of programs or for performing different types of computations.") Koker at 7:64-8:15 ("While a single instance of the parallel processing unit 202 is illustrated within the parallel processor 200, any number of instances of the parallel processing unit 202 can be included. For example, multiple instances of the parallel processing unit 202 can be provided on a single add-in card, or multiple add-in cards can be interconnected. The different instances of the parallel processing unit 202 can be configured to inter-operate even if the different instances have different numbers of processing cores, different amounts of local parallel processor memory, and/or other configuration differences. For example, in one embodiment some instances of the parallel processing unit 202 can include higher precision floating point units relative to other instances. Systems incorporating one or more instances of the parallel processing unit 202 or the parallel processor 200 can be implemented in a variety of configurations and form factors, including but not limited to desktop, laptop, or handheld personal computers, servers, workstations, game consoles, and/or embedded systems.") Koker at 12:18-27, ("FIG. 3A-3C illustrate additional graphics multiprocessors, according to embodiments. FIG. 3A-3B illustrate graphics multiprocessors 325, 350, which are variants of the graphics multiprocessor 234 of FIG. 2C. FIG. 3C illustrates a graphics processing unit (GPU) 380 which includes dedicated sets of graphics processing resources arranged into multi-core groups 365A-365N. The illustrated graphics multiprocessors 325, 350 and the multi-core groups 365A-365N can be streaming multiprocessor (SM) capable of simultaneous execution of a large number of execution threads.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 13:5-18 ("Persons skilled in the art will understand that the architecture described in FIGS. 1, 2A-2D, and 3A-3B are descriptive and not limiting as to the scope of the present embodiments. Thus, the techniques described herein may be implemented on any properly configured processing unit, including, without limitation, one or more mobile application processors, one or more desktop or server central processing units (CPUs) including multi-core CPUs, one or more parallel processing units, such as the parallel processing unit 202 of FIG. 2A, as well as one or more graphics processors or special purpose processing units, without departure from the scope of the embodiments described herein."). Koker at 13:19-35 ("In some embodiments a parallel processor or GPGPU as described herein is communicatively coupled to host/processor cores to accelerate graphics operations, machine-learning operations, pattern analysis operations, and various general purpose GPU (GPGPU) functions. The GPU may be communicatively coupled to the host processor/cores over a bus or other interconnect (e.g., a high speed interconnect such as PCIe or NVLink). In other embodiments, the GPU may be integrated on the same package or chip as the cores and communicatively coupled to the cores over an internal processor bus/interconnect (i.e., internal to the package or chip). Regardless of the manner in which the GPU is connected, the processor cores may allocate work to the GPU in the form of sequences of commands/instructions contained in a work descriptor. The GPU then uses dedicated circuitry/logic for efficiently processing these commands/instructions.") Koker at 14:3-14 ("Input/output (I/O) circuitry 363 couples the GPU 380 to one or more I/O devices 362 such as digital signal processors (DSPs), network controllers, or user input devices. An on-chip interconnect may be used to couple the I/O devices 362 to the GPU 380 and memory 366. One or more I/O memory management units (IOMMUs) 364 of the I/O circuitry 363 couple the I/O devices 362 directly to the system memory 366. In one embodiment, the IOMMU 364 manages multiple sets of page tables to map virtual addresses to physical addresses in system memory 366. In this embodiment, the I/O devices 362, CPU(s) 361, and GPU(s) 380 may share the same virtual address space.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 14:30-39 ("In one embodiment, the CPUs 361, GPUs 380, and I/O devices 362 are integrated on a single semiconductor chip and/or chip package. The illustrated memory 366 may be integrated on the same chip or may be coupled to the memory controllers 367 via an off-chip interface. In one implementation, the memory 366 comprises GDDR6 memory which shares the same virtual address space as other physical system-level memories, although the underlying principles of the invention are not limited to this specific implementation.") Koker at 15:6-24 ("In one embodiment, the ray tracing cores 372 accelerate ray tracing operations for both real-time ray tracing and non-real-time ray tracing implementations. In particular, the ray tracing cores 372 include ray traversal/intersection circuitry for performing ray traversal using bounding volume hierarchies (BVHs) and identifying intersections between rays and primitives enclosed within the BVH volumes. The ray tracing cores 372 may also include circuitry for performing depth testing and culling (e.g., using a Z buffer or similar arrangement). In one implementation, the ray tracing cores 372 perform traversal and intersection operations in concert with the image denoising techniques described herein, at least a portion of which may be executed on the tensor cores 371. For example, in one embodiment, the tensor cores 371 implement a deep learning neural network to perform denoising of frames generated by the ray tracing cores 372. However, the CPU(s) 361, graphics cores 370, and/or ray tracing cores 372 may also implement all or a portion of the denoising and/or deep learning algorithms."). Koker at 16:34-44 ("FIG. 4A illustrates an exemplary architecture in which a plurality of GPUs 410-413 are communicatively coupled to a plurality of multi-core processors 405-406 over high-speed links 440A-440D (e.g., buses, point-to-point interconnects, etc.). In one embodiment, the high-speed links 440A-440D support a communication throughput of 4 GB/s, 30 GB/s, 80 GB/s or higher, depending on the implementation. Various interconnect protocols may be used including, but not limited to, PCIe 4.0 or 5.0 and NVLink 2Ø However, the underlying principles of the invention are not limited to any particular communication protocol or throughput.") Koker at 17:21-28 ("FIG. 4B illustrates additional details for an interconnection between a multi-core processor 407 and a graphics acceleration module 446 in accordance with one embodiment. The graphics acceleration module 446 may include one or more GPU chips integrated on a line

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	card which is coupled to the processor 407 via the high-speed link 440. Alternatively, the graphics acceleration module 446 may be integrated on the same package or chip as the processor 407.") Koker at 19:36-46 ("FIG. 4C illustrates another embodiment in which the accelerator integration circuit 436 is integrated within the processor 407. In this embodiment, the graphics processing engines 431-432, N communicate directly over the high-speed link 440 to the accelerator integration circuit 436 via interface 437 and interface 435 (which, again, may be utilize any form of bus or interface protocol). The accelerator integration circuit 436 may perform the same operations as those described with respect to FIG. 4B, but potentially at a higher throughput given its close proximity to the coherency bus 464 and caches 462A-462D, 456.") Koker at 24:45-25:9 ("FIG. 5 illustrates a graphics processing pipeline 500, according to an embodiment. In one embodiment a graphics processor can implement the illustrated graphics processing pipeline 500. The graphics processor can be included within the parallel processing subsystems as described herein, such as the parallel processor 200 of FIG. 2A, which, in one embodiment, is a variant of the parallel processor(s) 112 of FIG. 1. The various parallel processing systems can implement the graphics processing pipeline 500 via one or more instances of the parallel processing unit (e.g., parallel processing unit 202 of FIG. 2A) as described herein. For example, a shader unit (e.g., graphics multiprocessor 234 of FIG. 2C) may be configured to perform the functions of one or more of a vertex processing unit 504, a tessellation control processing unit 508, a tessellation evaluation processing unit 512, a geometry processing unit 516, and a fragment/pixel processing unit 524. The functions of data assembler 502, primitive assemblers 506, 514, 518, tessellation unit 510, rasterizer 522, and raster operations unit 526 may also be performed by other processing engines within a processing cluster (e.g., processing cluster 214 of FIG. 2A) and a corresponding partition unit (e.g., partition unit 220A-220N of FIG. 2A). The graphics processing pipeline 500 may also be implemented using dedicated processing units for one or more functions. In one embodiment, one or more portions of the graphics processing pipeline 500 can be performed by parallel processing logic within a general purpose processor (e.g., CPU). In one embodiment, one or more portions of the graphics processing pipeline 500 can access on-chip memory (e.g., parallel processor memory

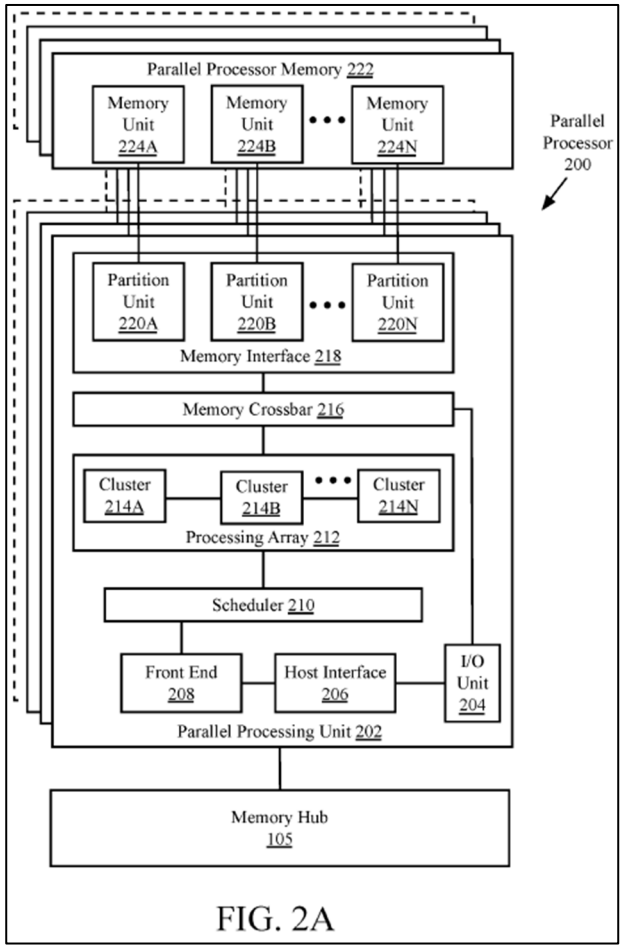
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	222 as in FIG. 2A) via a memory interface 528, which may be an instance of the memory interface 218 of FIG. 2A."). Koker at 29:52-30-11 ("FIG. 8 illustrates a multi-GPU computing system 800, according to an embodiment. The multi-GPU computing system 800 can include a processor 802 coupled to multiple GPGPUs 806A-806D via a host interface switch 804. The host interface switch 804, in one embodiment, is a PCI express switch device that couples the processor 802 to a PCI express bus over which the processor 802 can communicate with the set of GPGPUs 806A-806D. Each of the multiple GPGPUs 806A-806D can be an instance of the GPGPU 700 of FIG. 7. The GPGPUs 806A-806D can interconnect via a set of high-speed point to point GPU to GPU links 816. The high-speed GPU to GPU links can connect to each of the GPGPUs 806A-806D via a dedicated GPU link, such as the GPU link 710 as in FIG. 7. The P2P GPU links 816 enable direct communication between each of the GPGPUs 806A-806D without requiring communication over the host interface bus to which the processor 802 is connected. With GPU-to-GPU traffic directed to the P2P GPU links, the host interface bus remains available for system memory access or to communicate with other instances of the multi-GPU computing system 800, for example, via one or more network devices. While in the illustrated embodiment the GPGPUs 806A-D connect to the processor 802 via the host interface switch 804, in one embodiment the processor 802 includes direct support for the P2P GPU links 816 and can connect directly to the GPGPUs 806A-806D.") Koker at 39:34-52 ("For example, circuit boards ("sleds") can be used on which components such as CPUs, memory, and other components are placed are designed for increased thermal performance. In some examples, processing components such as the processors are located on a top side of a sled while near memory, such as DIMMs, are located on a bottom side of the sled. As a result of the enhanced airflow provided by this design, the components may operate at higher frequencies and power levels than in typical systems, thereby increasing performance. Furthermore, the sleds are configured to blindly mate with power and data communication cables in a rack, thereby enhancing their ability to be quickly removed, upgraded, reinstalled, and/or replaced. Similarly, individual components located on the sleds, such as processors, accelerators, memory, and data storage drives, are configured to be easily upgraded due to their increased

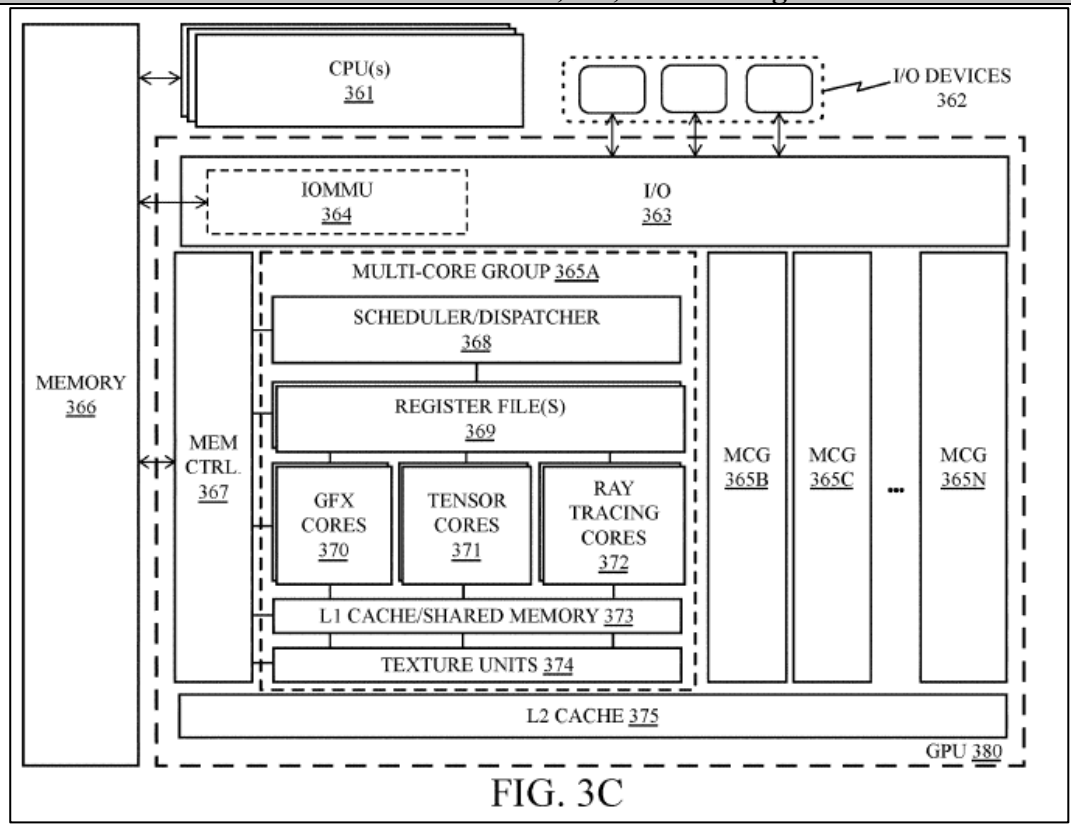
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	spacing from each other. In the illustrative embodiment, the components additionally include hardware attestation features to prove their authenticity.") Koker at 45:11-35 ("In one embodiment the SoC interface 1837 enables the graphics processor core 1800 to communicate with general-purpose application processor cores (e.g., CPUs) and/or other components within an SoC, including memory hierarchy elements such as a shared last level cache memory, the system RAM, and/or embedded on-chip or on-package DRAM. The SoC interface 1837 can also enable communication with fixed function devices within the SoC, such as camera imaging pipelines, and enables the use of and/or implements global memory atomics that may be shared between the graphics processor core 1800 and CPUs within the SoC. The SoC interface 1837 can also implement power management controls for the graphics processor core 1800 and enable an interface between a clock domain of the graphic core 1800 and other clock domains within the SoC. In one embodiment the SoC interface 1837 enables receipt of command buffers from a command streamer and global thread dispatcher that are configured to provide commands and instructions to each of one or more graphics cores within a graphics processor. The commands and instructions can be dispatched to the media pipeline 1839, when media operations are to be performed, or a geometry and fixed function pipeline (e.g., geometry and fixed function pipeline 1836, geometry and fixed function pipeline 1814) when graphics processing operations are to be performed.") Koker at 45:36-58 ("The graphics microcontroller 1838 can be configured to perform various scheduling and management tasks for the graphics processor core 1800. In one embodiment the graphics microcontroller 1838 can perform graphics and/or compute workload scheduling on the various graphics parallel engines within execution unit (EU) arrays 1802A-1802F, 1804A-1804F within the sub-cores 1801A-1801F. In this scheduling model, host software executing on a CPU core of an SoC including the graphics processor core 1800 can submit workloads one of multiple graphic processor doorbells, which invokes a scheduling operation on the appropriate graphics engine. Scheduling operations include determining which workload to run next, submitting a workload to a command streamer, pre-empting existing workloads running on an engine, monitoring progress of a workload, and notifying host software when a workload is complete. In one embodiment the graphics microcontroller 1838 can also facilitate low-power or idle states

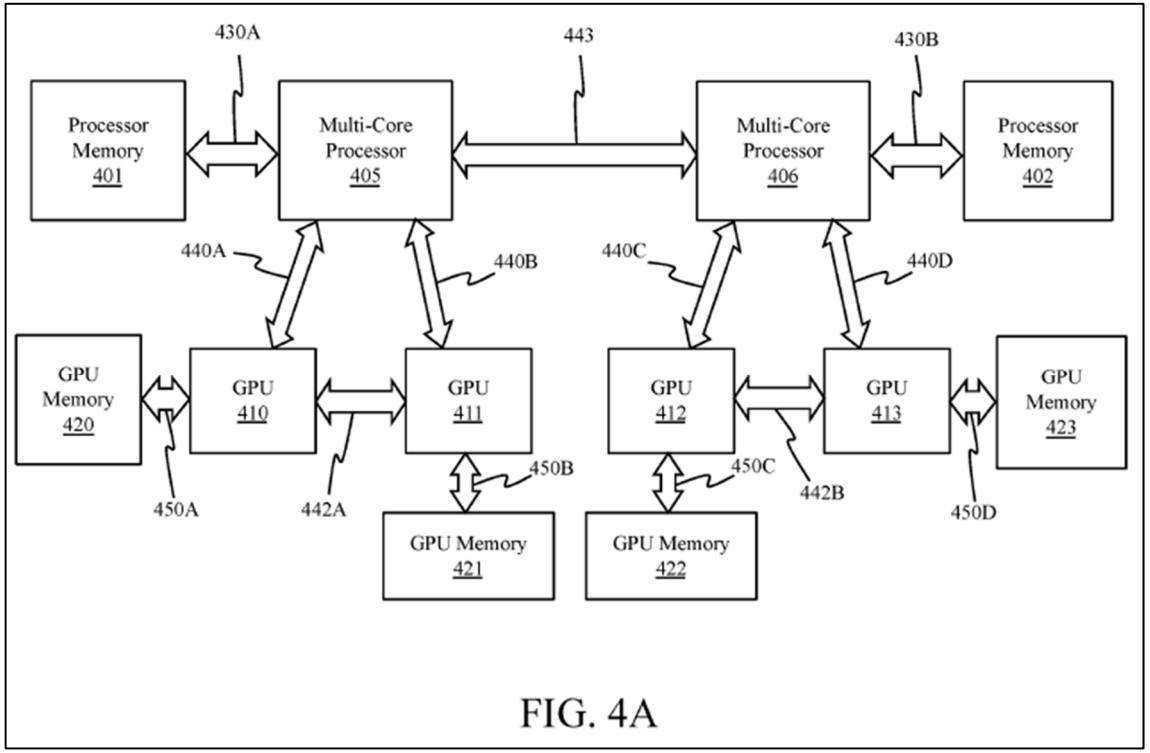
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	for the graphics processor core 1800, providing the graphics processor core 1800 with the ability to save and restore registers within the graphics processor core 1800 across low-power state transitions independently from the operating system and/or graphics driver software on the system.") Koker at 59:53-60:6 ("FIG. 25 is a block diagram illustrating an exemplary system on a chip integrated circuit 2500 that may be fabricated using one or more IP cores, according to an embodiment. Exemplary integrated circuit 2500 includes one or more application processor(s) 2505 (e.g., CPUs), at least one graphics processor 2510, and may additionally include an image processor 2515 and/or a video processor 2520, any of which may be a modular IP core from the same or multiple different design facilities. Integrated circuit 2500 includes peripheral or bus logic including a USB controller 2525, UART controller 2530, an SPI/SDIO controller 2535, and an I^2S/I^2C controller 2540. Additionally, the integrated circuit can include a display device 2545 coupled to one or more of a high-definition multimedia interface (HDMI) controller 2550 and a mobile industry processor interface (MIPI) display interface 2555. Storage may be provided by a flash memory subsystem 2560 including flash memory and a flash memory controller. Memory interface may be provided via a memory controller 2565 for access to SDRAM or SRAM memory devices. Some integrated circuits additionally include an embedded security engine 2570.") Koker at 61:15-23 ("Building larger and larger silicon dies is challenging for a variety of reasons. As silicon dies become larger, manufacturing yields become smaller and process technology requirements for different components may diverge. On the other hand, in order to have a high-performance system, key components should be interconnected by high speed, high bandwidth, low latency interfaces. These contradicting needs pose a challenge to high performance chip development.") Koker at 61:24-41 ("Embodiments described herein provide techniques to disaggregate an architecture of a system on a chip integrated circuit into multiple distinct chiplets that can be packaged onto a common chassis. In one embodiment, a graphics processing unit or parallel processor is composed from diverse silicon chiplets that are separately manufactured. A chiplet is an at least partially packaged integrated circuit that includes distinct units of logic that can be

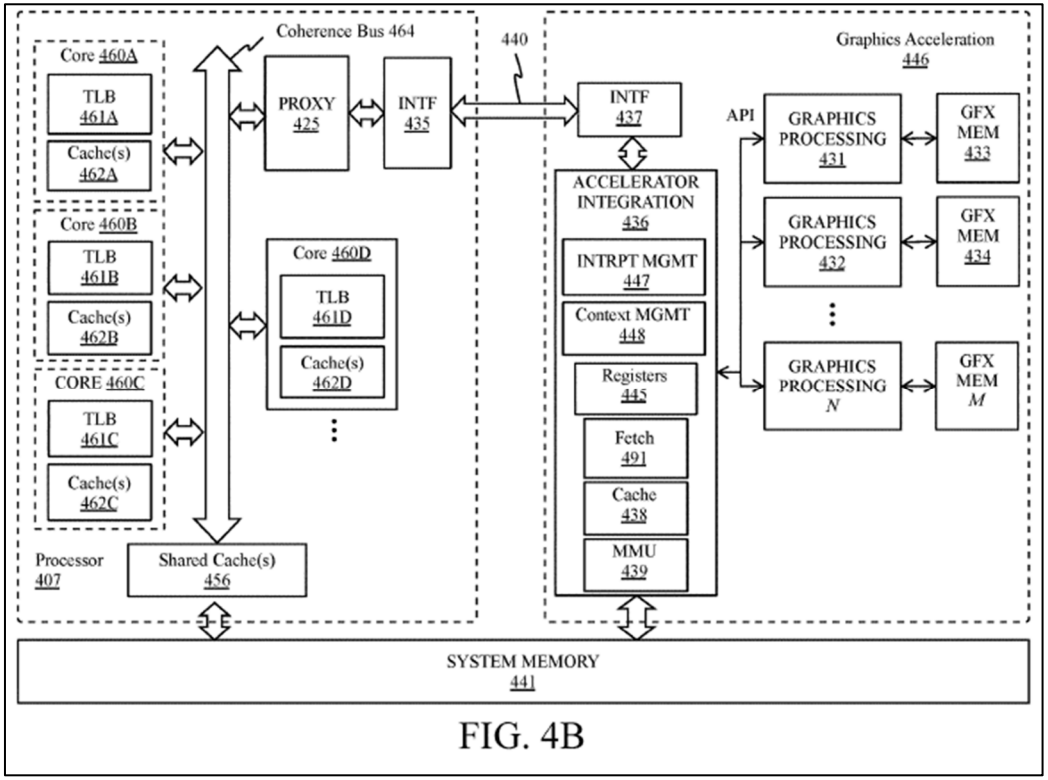
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	assembled with other chiplets into a larger package. A diverse set of chiplets with different IP core logic can be assembled into a single device. Additionally the chiplets can be integrated into a base die or base chiplet using active interposer technology. The concepts described herein enable the interconnection and communication between the different forms of IP within the GPU. The development of IPs on different process may be mixed. This avoids the complexity of converging multiple IPs, especially on a large SoC with several flavors IPs, to the same process.") Koker at 61:42-50 ("Enabling the use of multiple process technologies improves the time to market and provides a cost-effective way to create multiple product SKUs. For customers, this means getting products that are more tailored to their requirements in a cost effective and timely manner. Additionally, the disaggregated IPs are more amenable to being power gated independently, components that are not in use on a given workload can be powered off, reducing overall power consumption.") Koker at 61:51-62:5 ("FIG. 27 shows a parallel compute system 2700, according to an embodiment. In one embodiment the parallel compute system 2700 includes a parallel processor 2720, which can be a graphics processor or compute accelerator as described herein. The parallel processor 2720 includes a global logic 2701 unit, an interface 2702, a thread dispatcher 2703, a media unit 2704, a set of compute units 2705A-2705H, and a cache/memory units 2706. The global logic unit 2701, in one embodiment, includes global functionality for the parallel processor 2720, including device configuration registers, global schedulers, power management logic, and the like. The interface 2702 can include a front-end interface for the parallel processor 2720. The thread dispatcher 2703 can receive workloads from the interface 2702 and dispatch threads for the workload to the compute units 2705A-2705H. If the workload includes any media operations, at least a portion of those operations can be performed by the media unit 2704. The media unit can also offload some operations to the compute units 2705A-2705H. The cache/memory units 2706 can include cache memory (e.g., L3 cache) and local memory (e.g., HBM, GDDR) for the parallel processor 2720. Koker at 62:6-10 ("FIG. 28A-28B illustrate a hybrid logical/physical view of a disaggregated parallel processor, according to embodiments described herein. FIG. 28A illustrates a

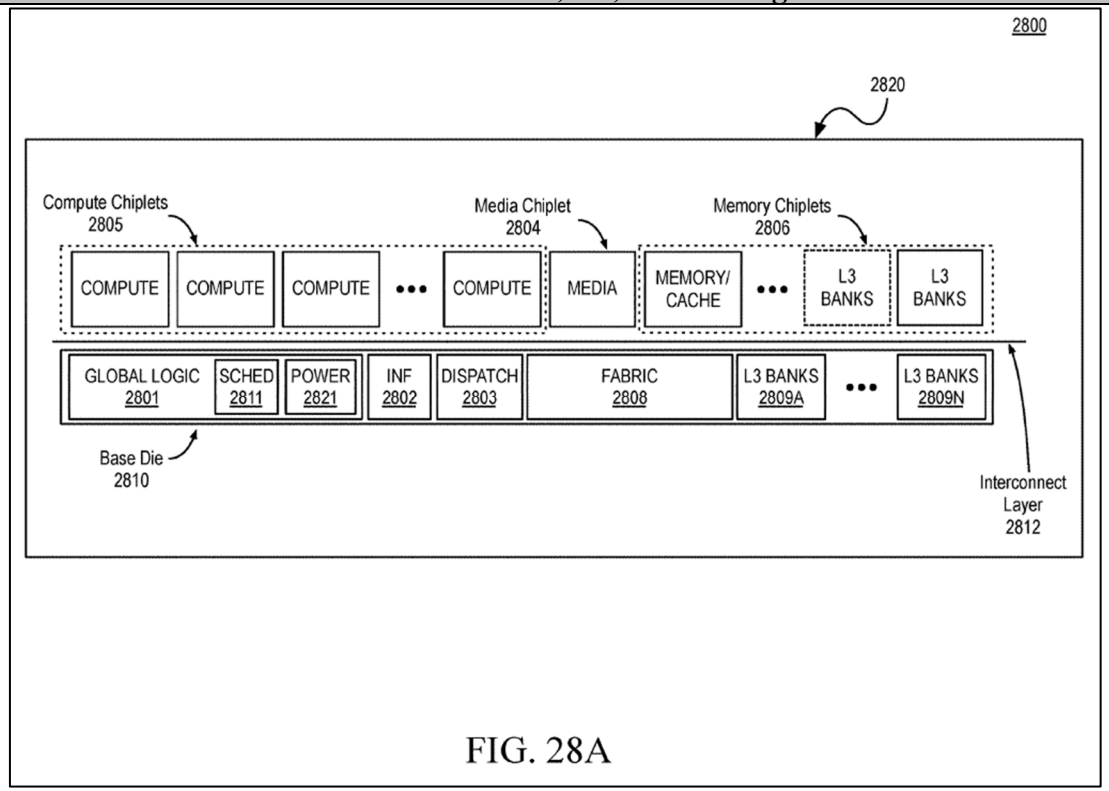
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	disaggregated parallel compute system 2800. FIG. 28B illustrates a chiplet 2830 of the disaggregated parallel compute system 2800.") Koker at 62:11-25 ("As shown in FIG. 28A, a disaggregated compute system 2800 can include a parallel processor 2820 in which the various components of the parallel processor SOC are distributed across multiple chiplets. Each chiplet can be a distinct IP core that is independently designed and configured to communicate with other chiplets via one or more common interfaces. The chiplets include but are not limited to compute chiplets 2805, a media chiplet 2804, and memory chiplets 2806. Each chiplet can be separately manufactured using different process technologies. For example, compute chiplets 2805 may be manufactured using the smallest or most advanced process technology available at the time of fabrication, while memory chiplets 2806 or other chiplets (e.g., I/O, networking, etc.) may be manufactured using a larger or less advanced process technologies.") Koker at 62:52-62 ("The various chiplets of the parallel processor 2820 can be designed to perform specific functionality that, in existing designs, would be integrated into a single die. A set of compute chiplets 2805 can include clusters of compute units (e.g., execution units, streaming multiprocessors, etc.) that include programmable logic to execute compute or graphics shader instructions. A media chiplet 2804 can include hardware logic to accelerate media encode and decode operations. Memory chiplets 2806 can include volatile memory (e.g., DRAM) and one or more SRAM cache memory banks (e.g., L3 banks).") This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
1[B]: wherein the GPU chiplet array includes: the first GPU chiplet communicably coupled to the CPU via a bus; and	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See, e.g.*,

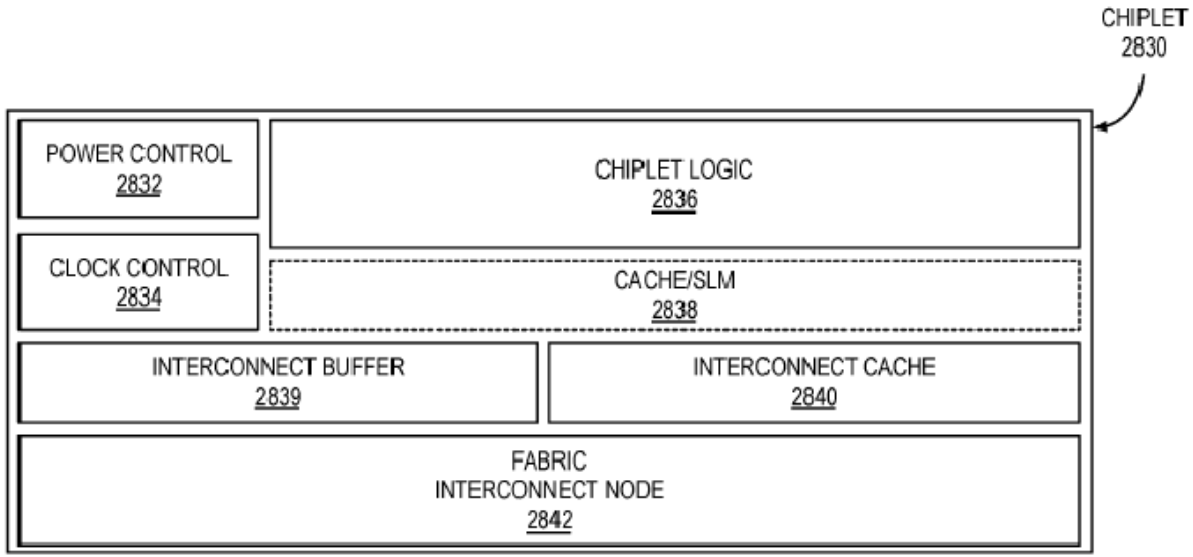
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at Fig. 2A.  FIG. 2A Koker at Fig. 3C.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 3C Koker at Fig. 4A.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4A Koker at Fig. 4B.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4B Koker at Fig. 28A.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 28A Koker at Fig. 28B.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 28B Koker at 3:20-35 ("In some embodiments, a graphics processing unit (GPU) is communicatively coupled to host/processor cores to accelerate graphics operations, machine-learning operations, pattern analysis operations, and various general-purpose GPU (GPGPU) functions. The GPU may be communicatively coupled to the host processor/cores over a bus or another interconnect (e.g., a high-speed interconnect such as PCIe or NVLink). In other embodiments, the GPU may be integrated on the same package or chip as the cores and communicatively coupled to the cores over an internal processor bus/interconnect (i.e., internal to the package or chip). Regardless of

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	the manner in which the GPU is connected, the processor cores may allocate work to the GPU in the form of sequences of commands/instructions contained in a work descriptor. The GPU then uses dedicated circuitry/logic for efficiently processing these commands/instructions.") Koker at 3:44-61 ("FIG. 1 is a block diagram illustrating a computing system 100 configured to implement one or more aspects of the embodiments described herein. The computing system 100 includes a processing subsystem 101 having one or more processor(s) 102 and a system memory 104 communicating via an interconnection path that may include a memory hub 105. The memory hub 105 may be a separate component within a chipset component or may be integrated within the one or more processor(s) 102. The memory hub 105 couples with an I/O subsystem 111 via a communication link 106. The I/O subsystem 111 includes an I/O hub 107 that can enable the computing system 100 to receive input from one or more input device(s) 108. Additionally, the I/O hub 107 can enable a display controller, which may be included in the one or more processor(s) 102, to provide outputs to one or more display device(s) 110A. In one embodiment the one or more display device(s) 110A coupled with the I/O hub 107 can include a local, internal, or embedded display device.") Koker at 4:27-37 ("The computing system 100 can include other components not explicitly shown, including USB or other port connections, optical storage drives, video capture devices, and the like, may also be connected to the I/O hub 107. Communication paths interconnecting the various components in FIG. 1 may be implemented using any suitable protocols, such as PCI (Peripheral Component Interconnect) based protocols (e.g., PCI-Express), or any other bus or point-to-point communication interfaces and/or protocol(s), such as the NV-Link high-speed interconnect, or interconnect protocols known in the art.") Koker at 4:38-57 ("In one embodiment, the one or more parallel processor(s) 112 incorporate circuitry optimized for graphics and video processing, including, for example, video output circuitry, and constitutes a graphics processing unit (GPU). In another embodiment, the one or more parallel processor(s) 112 incorporate circuitry optimized for general purpose processing, while preserving the underlying computational architecture, described in greater detail herein. In yet another embodiment, components of the computing system 100 may be integrated with one or more other system elements on a single integrated circuit. For example, the one or more

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	parallel processor(s) 112, memory hub 105, processor(s) 102, and I/O hub 107 can be integrated into a system on chip (SoC) integrated circuit. Alternatively, the components of the computing system 100 can be integrated into a single package to form a system in package (SIP) configuration. In one embodiment at least a portion of the components of the computing system 100 can be integrated into a multi-chip module (MCM), which can be interconnected with other multi-chip modules into a modular computing system.") Koker at 12:18-27, ("FIG. 3A-3C illustrate additional graphics multiprocessors, according to embodiments. FIG. 3A-3B illustrate graphics multiprocessors 325, 350, which are variants of the graphics multiprocessor 234 of FIG. 2C. FIG. 3C illustrates a graphics processing unit (GPU) 380 which includes dedicated sets of graphics processing resources arranged into multi-core groups 365A-365N. The illustrated graphics multiprocessors 325, 350 and the multi-core groups 365A-365N can be streaming multiprocessor (SM) capable of simultaneous execution of a large number of execution threads.") Koker at 13:19-35 ("In some embodiments a parallel processor or GPGPU as described herein is communicatively coupled to host/processor cores to accelerate graphics operations, machine-learning operations, pattern analysis operations, and various general purpose GPU (GPGPU) functions. The GPU may be communicatively coupled to the host processor/cores over a bus or other interconnect (e.g., a high speed interconnect such as PCIe or NVLink). In other embodiments, the GPU may be integrated on the same package or chip as the cores and communicatively coupled to the cores over an internal processor bus/interconnect (i.e., internal to the package or chip). Regardless of the manner in which the GPU is connected, the processor cores may allocate work to the GPU in the form of sequences of commands/instructions contained in a work descriptor. The GPU then uses dedicated circuitry/logic for efficiently processing these commands/instructions.") Koker at 29:52-30-11 ("FIG. 8 illustrates a multi-GPU computing system 800, according to an embodiment. The multi-GPU computing system 800 can include a processor 802 coupled to multiple GPGPUs 806A-806D via a host interface switch 804. The host interface switch 804, in one embodiment, is a PCI express switch device that couples the processor 802 to a PCI express bus over which the processor 802 can communicate with the set of GPGPUs 806A-806D. Each

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	of the multiple GPGPUs 806A-806D can be an instance of the GPGPU 700 of FIG. 7. The GPGPUs 806A-806D can interconnect via a set of high-speed point to point GPU to GPU links 816. The high-speed GPU to GPU links can connect to each of the GPGPUs 806A-806D via a dedicated GPU link, such as the GPU link 710 as in FIG. 7. The P2P GPU links 816 enable direct communication between each of the GPGPUs 806A-806D without requiring communication over the host interface bus to which the processor 802 is connected. With GPU-to-GPU traffic directed to the P2P GPU links, the host interface bus remains available for system memory access or to communicate with other instances of the multi-GPU computing system 800, for example, via one or more network devices. While in the illustrated embodiment the GPGPUs 806A-D connect to the processor 802 via the host interface switch 804, in one embodiment the processor 802 includes direct support for the P2P GPU links 816 and can connect directly to the GPGPUs 806A-806D.") Koker at 38:9-24 ("In some embodiments, one or more processor(s) 1402 are coupled with one or more interface bus(es) 1410 to transmit communication signals such as address, data, or control signals between processor 1402 and other components in the system 1400. The interface bus 1410, in one embodiment, can be a processor bus, such as a version of the Direct Media Interface (DMI) bus. However, processor busses are not limited to the DMI bus, and may include one or more Peripheral Component Interconnect buses (e.g., PCI, PCI Express), memory busses, or other types of interface busses. In one embodiment the processor(s) 1402 include an integrated memory controller 1416 and a platform controller hub 1430. The memory controller 1416 facilitates communication between a memory device and other components of the system 1400, while the platform controller hub (PCH) 1430 provides connections to I/O devices via a local I/O bus.") Koker at 62:6-10 ("FIG. 28A-28B illustrate a hybrid logical/physical view of a disaggregated parallel processor, according to embodiments described herein. FIG. 28A illustrates a disaggregated parallel compute system 2800. FIG. 28B illustrates a chiplet 2830 of the disaggregated parallel compute system 2800.") Koker at 62:11-25 ("As shown in FIG. 28A, a disaggregated compute system 2800 can include a parallel processor 2820 in which the various components of the parallel processor SOC are

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	distributed across multiple chiplets. Each chiplet can be a distinct IP core that is independently designed and configured to communicate with other chiplets via one or more common interfaces. The chiplets include but are not limited to compute chiplets 2805, a media chiplet 2804, and memory chiplets 2806. Each chiplet can be separately manufactured using different process technologies. For example, compute chiplets 2805 may be manufactured using the smallest or most advanced process technology available at the time of fabrication, while memory chiplets 2806 or other chiplets (e.g., I/O, networking, etc.) may be manufactured using a larger or less advanced process technologies.") Koker at 62:26-41 ("The various chiplets can be bonded to a base die 2810 and configured to communicate with each other and logic within the base die 2810 via an interconnect layer 2812. In one embodiment, the base die 2810 can include global logic 2801, which can include scheduler 2811 and power management 2821 logic units, an interface 2802, a dispatch unit 2803, and an interconnect fabric module 2808 coupled with or integrated with one or more L3 cache banks 2809A-2809N. The interconnect fabric 2808 can be an inter-chiplet fabric that is integrated into the base die 2810. Logic chiplets can use the fabric 2808 to relay messages between the various chiplets. Additionally, L3 cache banks 2809A-2809N in the base die and/or L3 cache banks within the memory chiplets 2806 can cache data read from and transmitted to DRAM chiplets within the memory chiplets 2806 and to system memory of a host.") Koker at 62:42-51 ("In one embodiment the global logic 2801 is a microcontroller that can execute firmware to perform scheduler 2811 and power management 2821 functionality for the parallel processor 2820. The microcontroller that executes the global logic can be tailored for the target use case of the parallel processor 2820. The scheduler 2811 can perform global scheduling operations for the parallel processor 2820. The power management 2821 functionality can be used to enable or disable individual chiplets within the parallel processor when those chiplets are not in use") Koker at 62:52-62 ("The various chiplets of the parallel processor 2820 can be designed to perform specific functionality that, in existing designs, would be integrated into a single die. A set of compute chiplets 2805 can include clusters of compute units (e.g., execution units, streaming multiprocessors, etc.) that include programmable logic to execute compute or graphics

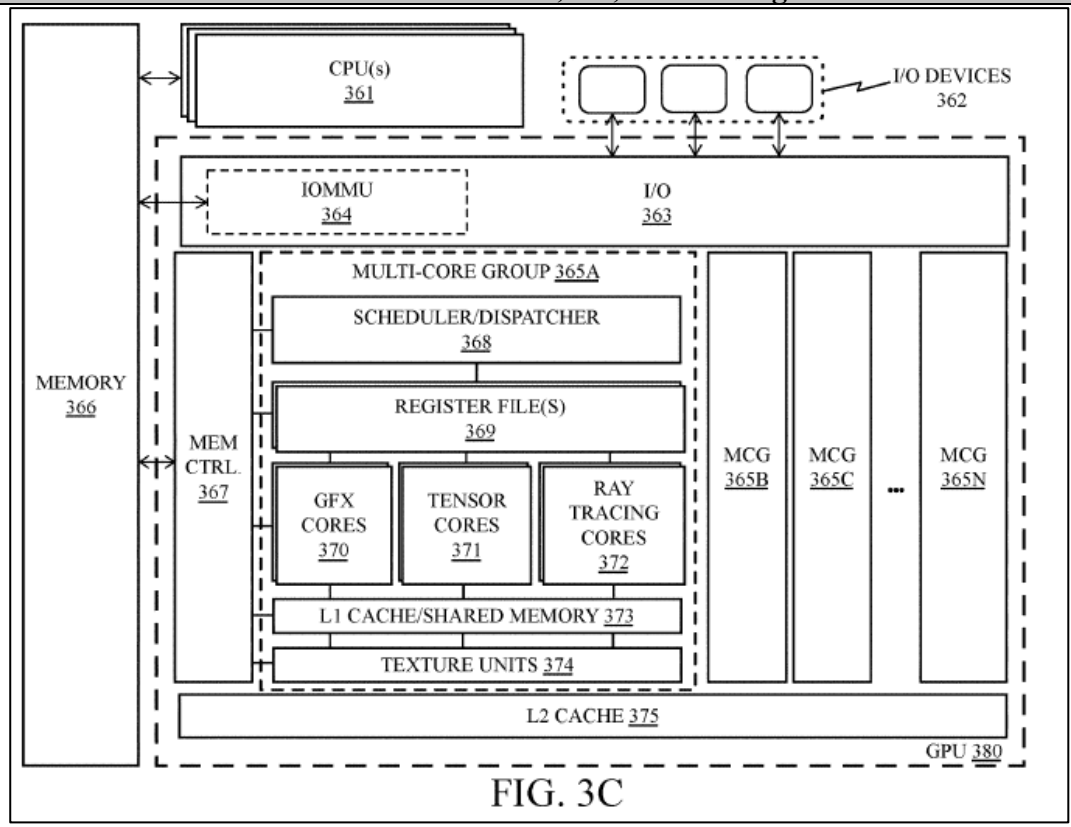
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	shader instructions. A media chiplet 2804 can include hardware logic to accelerate media encode and decode operations. Memory chiplets 2806 can include volatile memory (e.g., DRAM) and one or more SRAM cache memory banks (e.g., L3 banks).") Koker at 62:63-63:17 ("As shown in FIG. 28B, each chiplet 2830 can include common components and application specific components. Chiplet logic 2836 within the chiplet 2830 can include the specific components of the chiplet, such as an array of streaming multiprocessors, compute units, or execution units described herein. The chiplet logic 2836 can couple with an optional cache or shared local memory 2838 or can include a cache or shared local memory within the chiplet logic 2836. The chiplet 2830 can include a fabric interconnect node 2842 that receives commands via the inter-chiplet fabric. Commands and data received via the fabric interconnect node 2842 can be stored temporarily within an interconnect buffer 2839. Data transmitted to and received from the fabric interconnect node 2842 can be stored in an interconnect cache 2840. Power control 2832 and clock control 2834 logic can also be included within the chiplet. The power control 2832 and clock control 2834 logic can receive configuration commands via the fabric can configure dynamic voltage and frequency scaling for the chiplet 2830. In one embodiment, each chiplet can have an independent clock domain and power domain and can be clock gated and power gated independently of other chiplets.") Koker at 63:18-25 ("At least a portion of the components within the illustrated chiplet 2830 can also be included within logic embedded within the base die 2810 of FIG. 28A. For example, logic within the base die that communicates with the fabric can include a version of the fabric interconnect node 2842. Base die logic that can be independently clock or power gated can include a version of the power control 2832 and/or clock control 2834 logic.") Koker at 63:26-30 ("FIG. 29A-29B illustrate a package view of a disaggregated parallel processor, according to embodiments. FIG. 29A illustrates a physical layout of a package assembly 2920. FIG. 29B illustrates an interconnect between multiple chiplets 2904, 2906 and an interconnect fabric 2940.") Koker at 63:31-43 ("As shown in FIG. 29A, a package assembly 2920 can include multiple units of hardware logic chiplets connected to a substrate 2910 (e.g., base die). The hardware logic

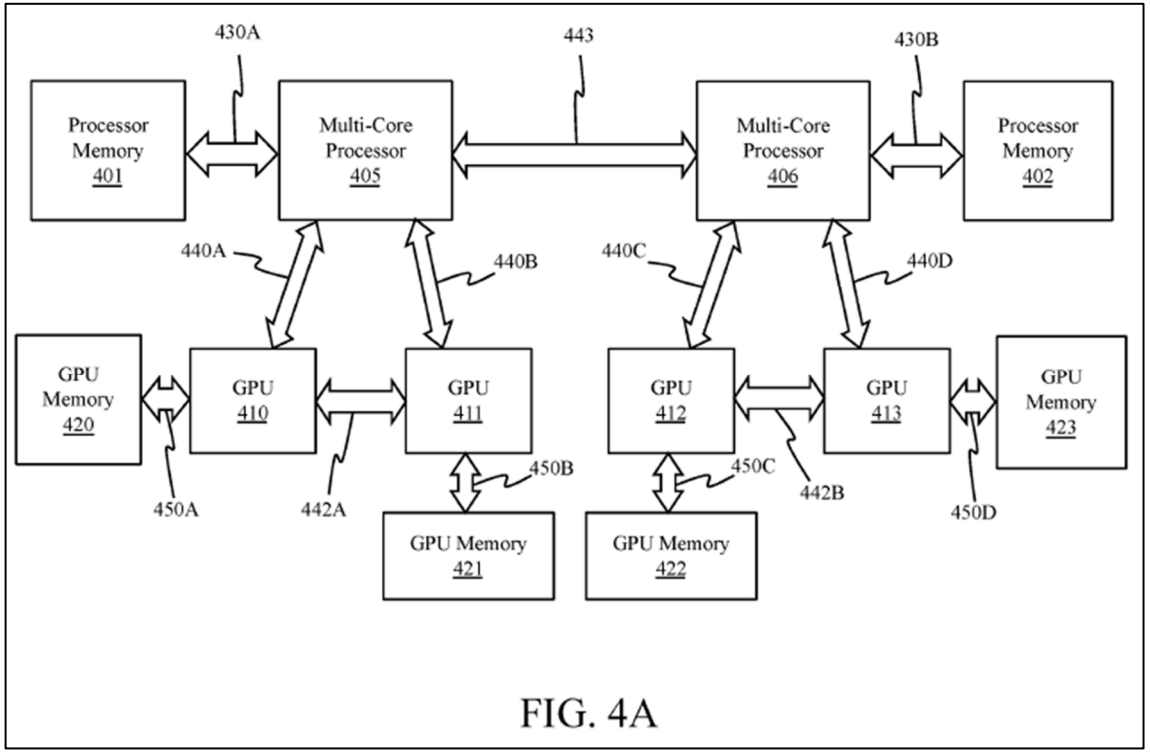
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	chiplets can include special purpose hardware logic chiplets 2902, logic or I/O chiplets 2904, and/or memory chiplets 2905. The hardware logic chiplets 2902 and logic or I/O chiplets 2904 may be implemented at least partly in configurable logic or fixed-functionality logic hardware and can include one or more portions of any of the processor core(s), graphics processor(s), parallel processors, or other accelerator devices described herein. The memory chiplets 2905 can be DRAM (e.g., GDDR, HIBM) memory or cache (SRAM) memory, as described and illustrated herein.") Koker at 63:44-54 ("Each chiplet can be fabricated as separate semiconductor die and coupled with the substrate 2910 via an interconnect structure 2903. The interconnect structure 2903 may be configured to route electrical signals between the various chiplets and logic within the substrate 2910. The interconnect structure 2903 can include interconnects such as, but not limited to bumps or pillars. In some embodiments, the interconnect structure 2903 may be configured to route electrical signals such as, for example, input/output (I/O) signals and/or power or ground signals associated with the operation of the logic, I/O and memory chiplets.") Koker at 63:55-63 ("In some embodiments, the substrate 2910 is an epoxy-based laminate substrate. The substrate 2910 may include other suitable types of substrates in other embodiments. The package assembly 2920 can be connected to other electrical devices via a package interconnect 2914. The package interconnect 2914 may be coupled to a surface of the substrate 2910 to route electrical signals to other electrical devices, such as a motherboard, other chipset, or multi-chip module.") Koker at 63:64-64:12 ("In some embodiments, a logic or I/O chiplet 2904 and a memory chiplet 2905 can be electrically coupled via a bridge 2917 that is configured to route electrical signals between the logic or I/O chiplet 2904 and a memory chiplet 2905. The bridge 2917 may be a dense interconnect structure that provides a route for electrical signals. The bridge 2917 may include a bridge substrate composed of glass or a suitable semiconductor material. Electrical routing features can be formed on the bridge substrate to provide a chip-to-chip connection between the logic or I/O chiplet 2904 and a memory chiplet 2905. The bridge 2917 may also be referred to as a silicon bridge or an interconnect bridge. For example, the bridge 2917, in some

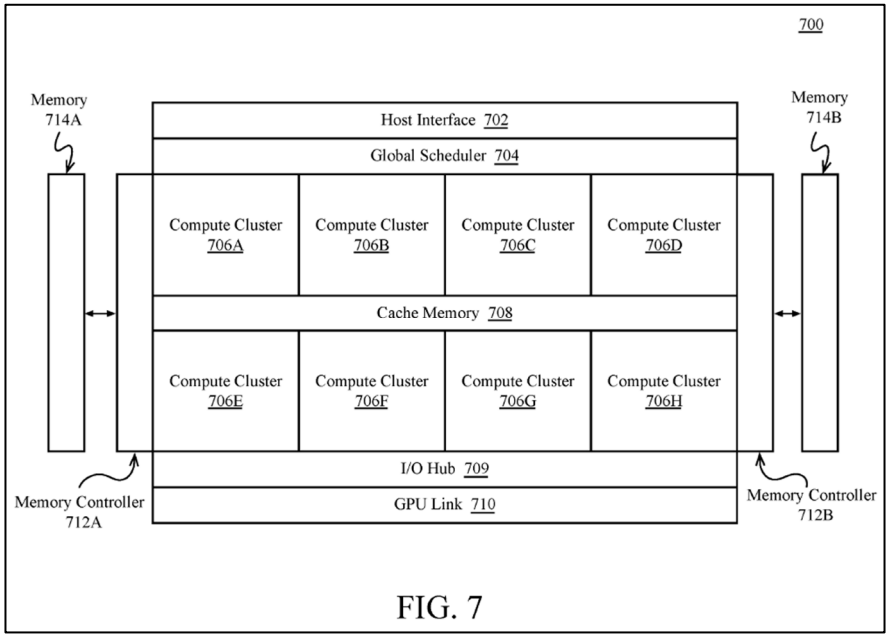
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	embodiments, is an Embedded Multi-die Interconnect Bridge (EMIB). In some embodiments, the bridge 2917 may simply be a direct connection from one chiplet to another chiplet.") Koker at 64:13-18 ("The substrate 2910 can include hardware components for I/O 2911, cache memory 2912, and other hardware logic 2913. A fabric 2915 can be embedded in the substrate 2910 to enable communication between the various logic chiplets and the logic 2911, 2913 within the substrate 2910.") Koker at 64:19-32 ("In various embodiments a package assembly 2920 can include fewer or greater number of components and chiplets that are interconnected by a fabric 2915 or one or more bridges 2917. The chiplets within the package assembly 2920 may be arranged in a 3D or 2.5D arrangement. In general, bridge structures 2917 may be used to facilitate a point to point interconnect between, for example, logic or I/O chiplets and memory chiplets. The fabric 2915 can be used to interconnect the various logic and/or I/O chiplets (e.g., chiplets 2902, 2904, 2911, 2913). with other logic and/or I/O chiplets. In one embodiment, the cache memory 2912 within the substrate can act as a global cache for the package assembly 2920, part of a distributed global cache, or as a dedicated cache for the fabric 2915.") Koker at 64:33-41 ("As shown in FIG. 29B, a memory chiplet 2906 can connect with a logic or I/O chiplet 2904 via chiplet interconnects 2935 routed though the interconnect bridge 2947. The interconnect bridge 2947 can be a variant of the bridge 2917 embedded in the substrate 2910 of the package assembly 2920 shown in FIG. 29A. The logic or I/O chiplet 2904 can communicate with other chiplets via the interconnect fabric 2940, which is a variant of the fabric 2915 of FIG. 29A.") Koker at 64:42-67 ("In one embodiment, the memory chiplet 2906 includes a set of memory banks 2931 that correspond with the memory technology provided by the chiplet. The memory banks 2931 can include any of the types of memory described herein, including but not limited to DRAM, SRAM, or flash, or 3D XPoint memory. A memory control protocol layer 2932 can enable control of the memory banks 2931 and can include logic for one or more memory controllers. An interconnect bridge protocol layer 2933 can relay messages between the memory control protocol 2932 and the interconnect bridge I/O layer 2934. Interconnect bridge I/O layer

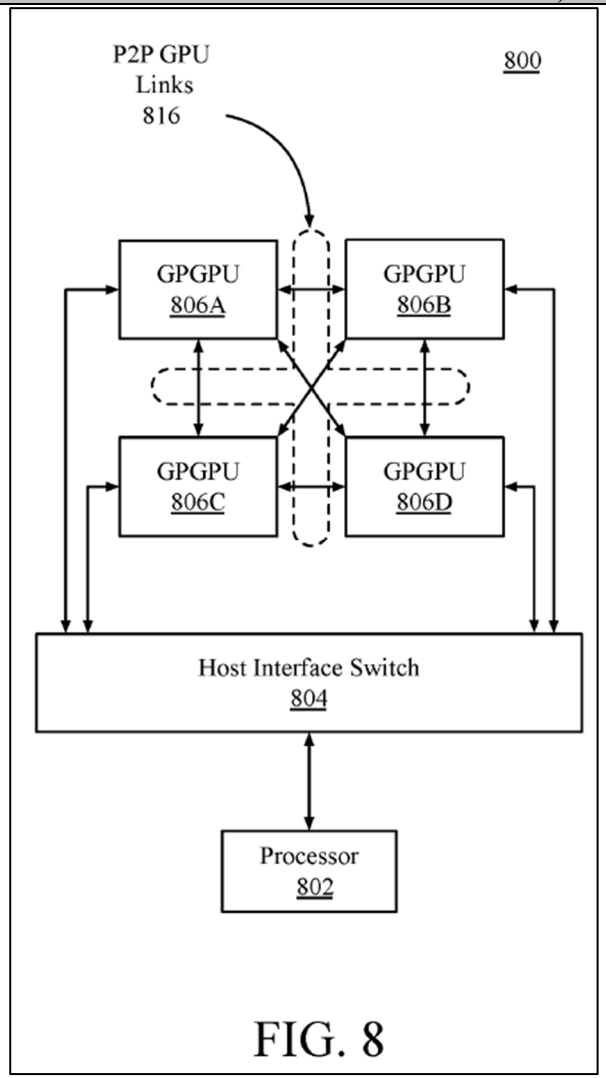
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	2934 can communicate with interconnect bridge I/O layer 2936 over the chiplet interconnects 2935. Interconnect bridge I/O layers 2934, 2936 can represent physical layers that transmit signals to or receive signals from a corresponding interconnect point over the chiplet interconnects 2935. The physical I/O layers may include circuitry to drive signals over the chiplet interconnects 2935 and/or receive signals from the chiplet interconnects 2935. An interconnect bridge protocol layer 2937 within the logic or I/O chiplet 2904 can translate signals from interconnect bridge I/O layer 2936 in to messages or signals that can be communicated to the compute or I/O logic 2939. In one embodiment a digital adapter layer 2938 can be used to facilitate the translation of signals into messages or signals for use by the compute or I/O logic 2939.") Koker at 65:1-6 ("The compute or I/O logic 2939 can communicate with other logic or I/O chiplets via the interconnect fabric 2940. The compute or I/O logic 2939, in one embodiment, includes integrated fabric node logic 2939 that can communicate with the interconnect fabric 2940, such as the fabric interconnect node 2942 of FIG. 28B.") Koker at 65:7-16 ("In one embodiment, a control layer 2968 in the memory chiplet 2906 can be in communication with a control layer 2970 in the logic or I/O chiplet. These control layers 2968, 2970 can be used to propagate or transmit certain control signals in an out-of-band manner, for example, to send power and configuration messages between interface bus protocol layer 2937 of the logic or I/O chiplet 2904 and interface bus protocol layer 2933, memory control protocol 2932 and/or memory banks 2931 of the memory chiplet 2906.") Koker at 65:17-33 ("FIG. 30 illustrates a message transportation system 3000 for an interconnect fabric, according to an embodiment. The message transportation system 3000 can be configured to handle traffic at different rates depending on the interface widths available. The specific interface widths can vary on a chiplet-by-chiplet basis and the speed or configuration of the fabric can be adjusted to enable data to be transmitted at a rate that is appropriate for the various functional units to be interconnected. The transport layer can also cross one or more clock domains via the use of clock-domain-crossing FIFOs at a boundaries between clock domains to enable data to be transferred across the clock domains. The transport layer can also be divided into one or more sub-layers, where each sub-layer includes one or more clock domains. While

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	transport layers are described and illustrated, in some embodiments the illustrated operations can be performed at a data-link layer of the interconnect fabric.") Koker at 65:34-44 ("In one embodiment a first functional unit 3001A in an origin layer 3010 can communicate via a second functional unit 3001B in a destination layer 3013 via one or more transport layers 3011, 3012. The origin layer 3010 can be logic or I/O within a chiplet or within the substrate of a package assembly, such as the substrate 2910 of package assembly 2920 as in FIG. 29A. The destination layer 3013 can also be logic or I/O within a chiplet or within the substrate. In one embodiment, the origin layer 3010 and/or destination layer 3013 can be associated with cache memory within a chiplet or the substrate/base die.") *See also* evidence cited for '803 patent claim 1[A]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
1[C]: a second GPU chiplet communicably coupled to the first GPU chiplet via a passive crosslink,	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See, e.g.*, Koker at Fig. 3C.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 3C Koker at Fig. 4A.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4A Koker at Fig. 7.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 7 Koker at Fig. 8.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 8 Koker at Fig. 21.

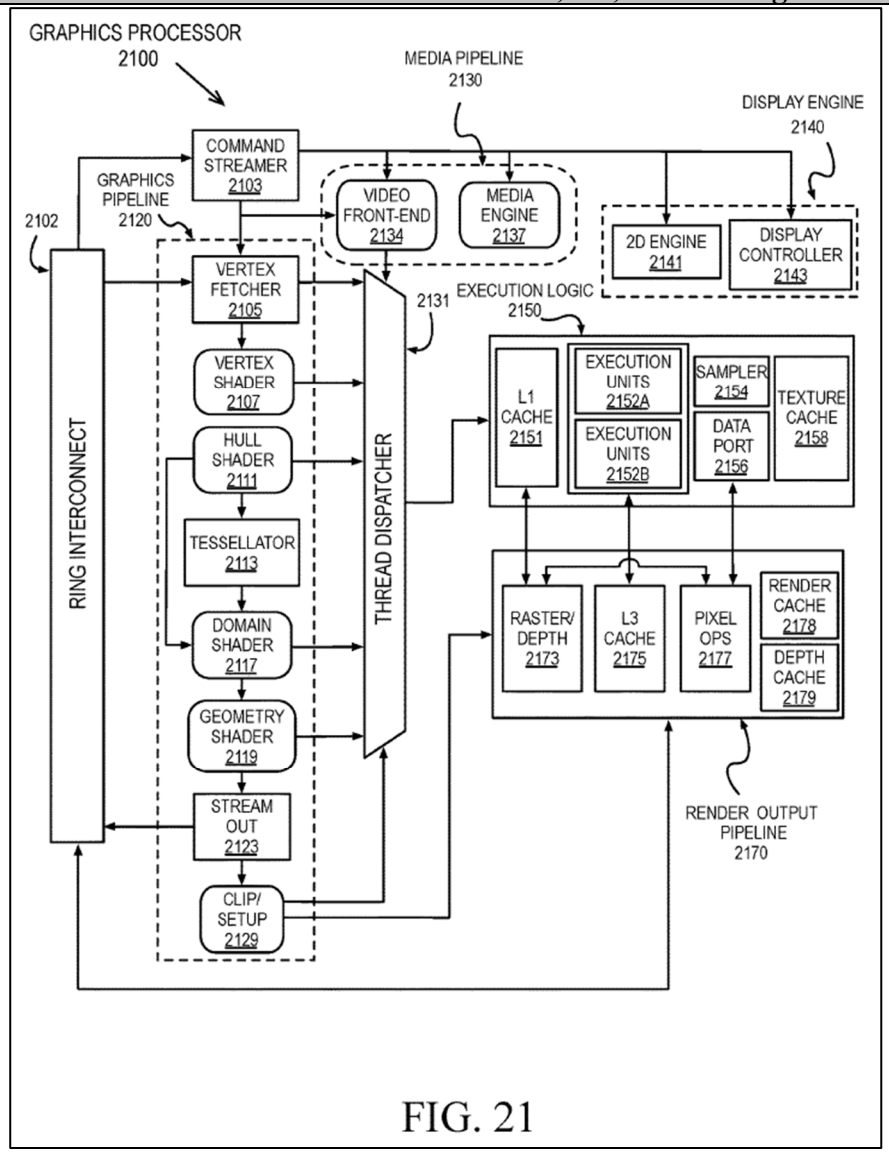
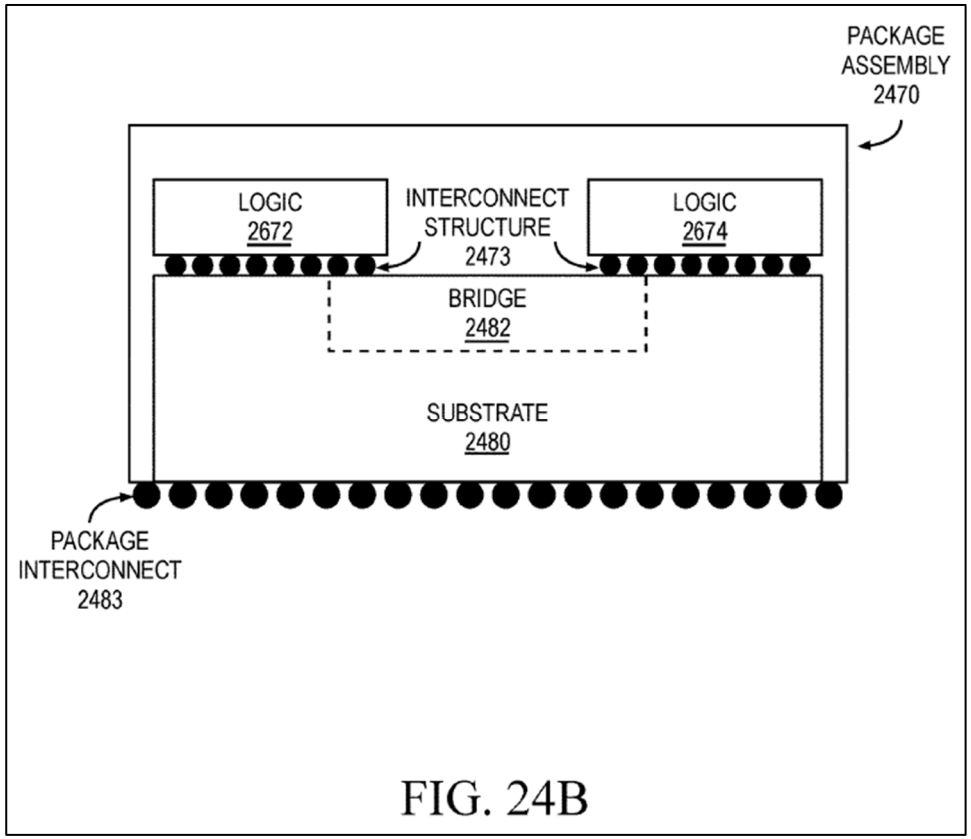
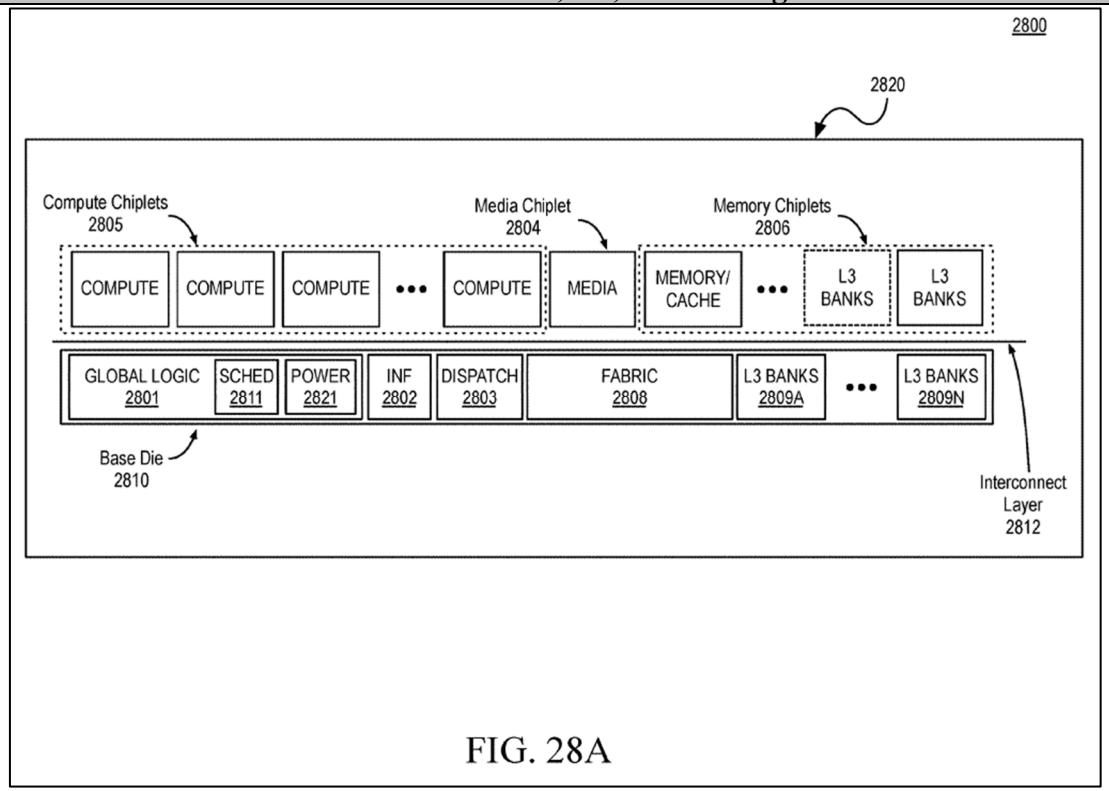
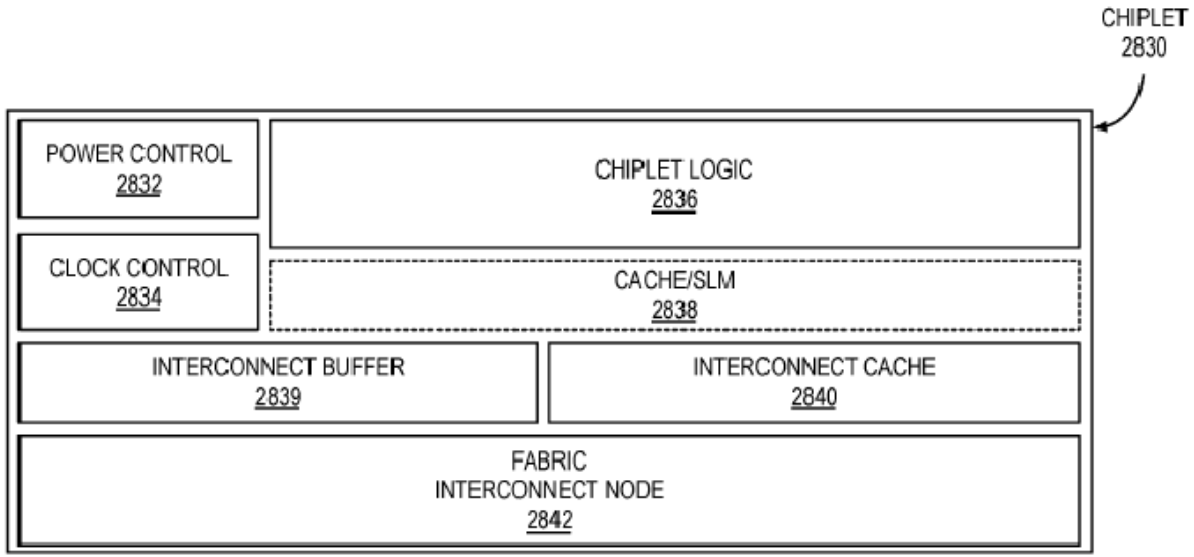
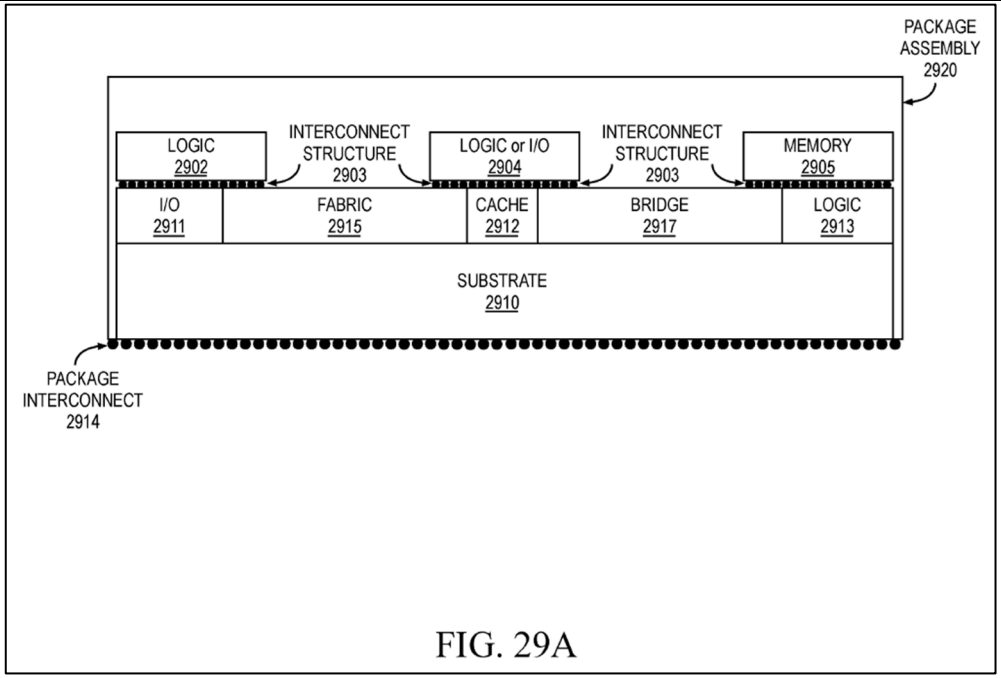
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	

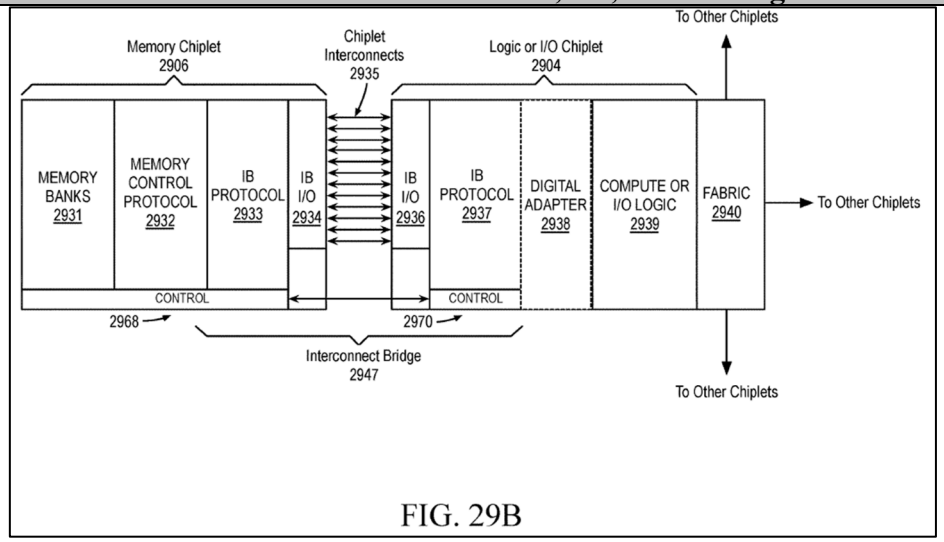
FIG. 21

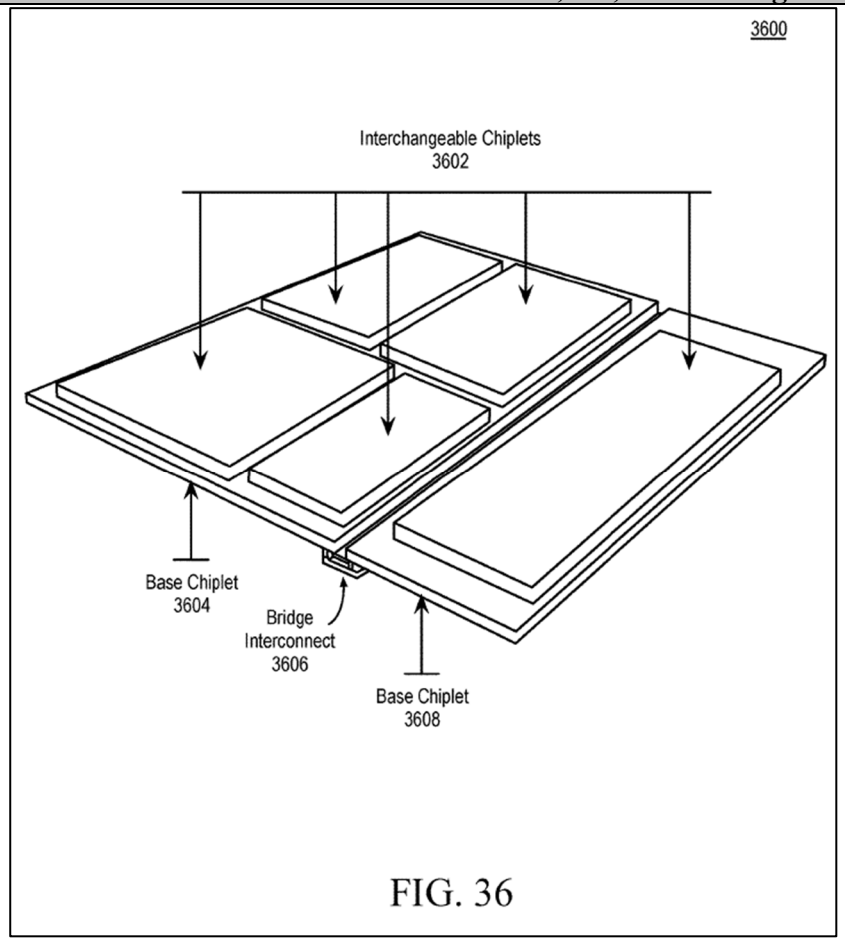
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker Fig. 24B.  FIG. 24B Koker at Fig. 28A.

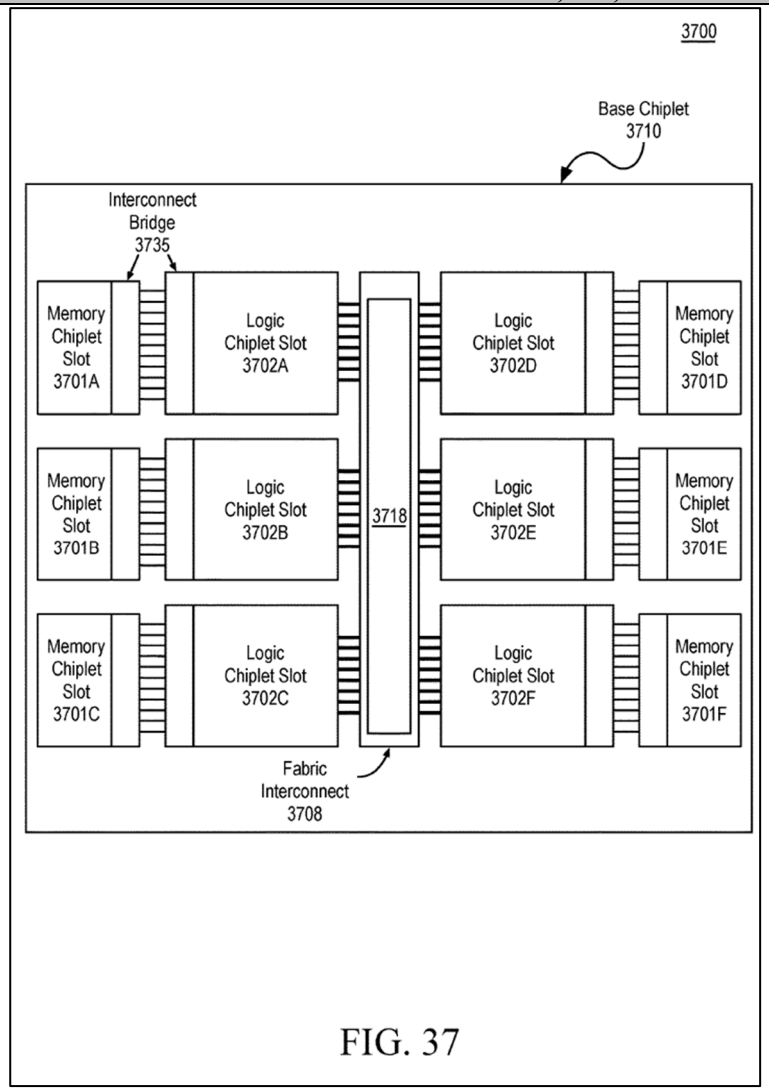
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 28A Koker at Fig. 28B.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 28B Koker at Fig. 29A.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 29A Koker at Fig. 29B.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 29B Koker at Fig. 36.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 36 Koker at Fig. 37.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 37 Koker at 12:42-56 ("The various components can communicate via an interconnect fabric 327. In one embodiment the interconnect fabric 327 includes one or more crossbar switches to enable

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	communication between the various components of the graphics multiprocessor 325. In one embodiment the interconnect fabric 327 is a separate, high-speed network fabric layer upon which each component of the graphics multiprocessor 325 is stacked. The components of the graphics multiprocessor 325 communicate with remote components via the interconnect fabric 327. For example, the GPGPU cores 336A-336B, 337A-337B, and 3378A-338B can each communicate with shared memory 346 via the interconnect fabric 327. The interconnect fabric 327 can arbitrate communication within the graphics multiprocessor 325 to ensure a fair bandwidth allocation between components."). Koker at 16:45-58 ("In addition, in one embodiment, two or more of the GPUs 410-413 are interconnected over high-speed links 442A-442B, which may be implemented using the same or different protocols/links than those used for high-speed links 440A-440D. Similarly, two or more of the multi-core processors 405-406 may be connected over high speed link 443 which may be symmetric multi-processor (SMP) buses operating at 20 GB/s, 30 GB/s, 120 GB/s or higher. Alternatively, all communication between the various system components shown in FIG. 4A may be accomplished using the same protocols/links (e.g., over a common interconnection fabric). As mentioned, however, the underlying principles of the invention are not limited to any particular type of interconnect technology.") Koker at 17:21-28 ("FIG. 4B illustrates additional details for an interconnection between a multi-core processor 407 and a graphics acceleration module 446 in accordance with one embodiment. The graphics acceleration module 446 may include one or more GPU chips integrated on a line card which is coupled to the processor 407 via the high-speed link 440. Alternatively, the graphics acceleration module 446 may be integrated on the same package or chip as the processor 407.") Koker at 18:1-17 ("In one implementation, an accelerator integration circuit 436 provides cache management, memory access, context management, and interrupt management services on behalf of a plurality of graphics processing engines 431, 432, N of the graphics acceleration module 446. The graphics processing engines 431, 432, N may each comprise a separate graphics processing unit (GPU). Alternatively, the graphics processing engines 431, 432, N may comprise different types of graphics processing engines within a GPU such as graphics execution units,

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	media processing engines (e.g., video encoders/decoders), samplers, and blit engines. In other words, the graphics acceleration module may be a GPU with a plurality of graphics processing engines 431-432, N or the graphics processing engines 431-432, N may be individual GPUs integrated on a common package, line card, or chip.") Koker at 29:16-36 ("Multiple instances of the GPGPU 700 can be configured to operate as a compute cluster. The communication mechanism used by the compute cluster for synchronization and data exchange varies across embodiments. In one embodiment, the multiple instances of the GPGPU 700 communicate over the host interface 702. In one embodiment the GPGPU 700 includes an I/O hub 709 that couples the GPGPU 700 with a GPU link 710 that enables a direct connection to other instances of the GPGPU. In one embodiment the GPU link 710 is coupled to a dedicated GPU-to-GPU bridge that enables communication and synchronization between multiple instances of the GPGPU 700. In one embodiment the GPU link 710 couples with a high-speed interconnect to transmit and receive data to other GPGPUs or parallel processors. In one embodiment the multiple instances of the GPGPU 700 are located in separate data processing systems and communicate via a network device that is accessible via the host interface 702. In one embodiment the GPU link 710 can be configured to enable a connection to a host processor in addition to or as an alternative to the host interface 702.") Koker at 29:52-30-11 ("FIG. 8 illustrates a multi-GPU computing system 800, according to an embodiment. The multi-GPU computing system 800 can include a processor 802 coupled to multiple GPGPUs 806A-806D via a host interface switch 804. The host interface switch 804, in one embodiment, is a PCI express switch device that couples the processor 802 to a PCI express bus over which the processor 802 can communicate with the set of GPGPUs 806A-806D. Each of the multiple GPGPUs 806A-806D can be an instance of the GPGPU 700 of FIG. 7. The GPGPUs 806A-806D can interconnect via a set of high-speed point to point GPU to GPU links 816. The high-speed GPU to GPU links can connect to each of the GPGPUs 806A-806D via a dedicated GPU link, such as the GPU link 710 as in FIG. 7. The P2P GPU links 816 enable direct communication between each of the GPGPUs 806A-806D without requiring communication over the host interface bus to which the processor 802 is connected. With GPU-to-GPU traffic directed to the P2P GPU links, the host interface bus remains available for system memory access or to communicate with other instances of the multi-GPU computing system 800, for

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	example, via one or more network devices. While in the illustrated embodiment the GPGPUs 806A-D connect to the processor 802 via the host interface switch 804, in one embodiment the processor 802 includes direct support for the P2P GPU links 816 and can connect directly to the GPGPUs 806A-806D.") Koker at 41:3-10 ("In some embodiments, a ring based interconnect unit 1512 is used to couple the internal components of the processor 1500. However, an alternative interconnect unit may be used, such as a point-to-point interconnect, a switched interconnect, or other techniques, including techniques well known in the art. In some embodiments, graphics processor 1508 couples with the ring interconnect 1512 via an I/O link 1513.") Koker at 52:35-50 ("In some embodiments, graphics processor 2100 includes a geometry pipeline 2120, a media pipeline 2130, a display engine 2140, thread execution logic 2150, and a render output pipeline 2170. In some embodiments, graphics processor 2100 is a graphics processor within a multi-core processing system that includes one or more general-purpose processing cores. The graphics processor is controlled by register writes to one or more control registers (not shown) or via commands issued to graphics processor 2100 via a ring interconnect 2102. In some embodiments, ring interconnect 2102 couples graphics processor 2100 to other processing components, such as other graphics processors or general-purpose processors. Commands from ring interconnect 2102 are interpreted by a command streamer 2103, which supplies instructions to individual components of the geometry pipeline 2120 or the media pipeline 2130.") Koker at 53:39-51 ("The graphics processor 2100 has an interconnect bus, interconnect fabric, or some other interconnect mechanism that allows data and message passing amongst the major components of the processor. In some embodiments, execution units 2152A-2152B and associated logic units (e.g., L1 cache 2151, sampler 2154, texture cache 2158, etc.) interconnect via a data port 2156 to perform memory access and communicate with render output pipeline components of the processor. In some embodiments, sampler 2154, L1 cache 2151, texture cache 2158, and execution units 2152A-2152B each have separate memory access paths. In one embodiment the texture cache 2158 can also be configured as a sampler cache.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 58:63-59:25 ("FIG. 24B illustrates a cross-section side view of an integrated circuit package assembly 2470, according to some embodiments described herein. The integrated circuit package assembly 2470 illustrates an implementation of one or more processor or accelerator devices as described herein. The package assembly 2470 includes multiple units of hardware logic 2472, 2474 connected to a substrate 2480. The logic 2472, 2474 may be implemented at least partly in configurable logic or fixed-functionality logic hardware, and can include one or more portions of any of the processor core(s), graphics processor(s), or other accelerator devices described herein. Each unit of logic 2472, 2474 can be implemented within a semiconductor die and coupled with the substrate 2480 via an interconnect structure 2473. The interconnect structure 2473 may be configured to route electrical signals between the logic 2472, 2474 and the substrate 2480, and can include interconnects such as, but not limited to bumps or pillars. In some embodiments, the interconnect structure 2473 may be configured to route electrical signals such as, for example, input/output (I/O) signals and/or power or ground signals associated with the operation of the logic 2472, 2474. In some embodiments, the substrate 2480 is an epoxy-based laminate substrate. The substrate 2480 may include other suitable types of substrates in other embodiments. The package assembly 2470 can be connected to other electrical devices via a package interconnect 2483. The package interconnect 2483 may be coupled to a surface of the substrate 2480 to route electrical signals to other electrical devices, such as a motherboard, other chipset, or multi-chip module.") Koker at 59:26-34 ("In some embodiments, the units of logic 2472, 2474 are electrically coupled with a bridge 2482 that is configured to route electrical signals between the logic 2472, 2474. The bridge 2482 may be a dense interconnect structure that provides a route for electrical signals. The bridge 2482 may include a bridge substrate composed of glass or a suitable semiconductor material. Electrical routing features can be formed on the bridge substrate to provide a chip-to-chip connection between the logic 2472, 2474.") Koker at 59:35-44 ("Although two units of logic 2472, 2474 and a bridge 2482 are illustrated, embodiments described herein may include more or fewer logic units on one or more dies. The one or more dies may be connected by zero or more bridges, as the bridge 2482 may be excluded when the logic is included on a single die. Alternatively, multiple dies or units of logic can be connected by one or more bridges. Additionally, multiple logic units, dies, and bridges can be

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	connected together in other possible configurations, including three-dimensional configurations.") Koker at 61:15-23 ("Building larger and larger silicon dies is challenging for a variety of reasons. As silicon dies become larger, manufacturing yields become smaller and process technology requirements for different components may diverge. On the other hand, in order to have a high-performance system, key components should be interconnected by high speed, high bandwidth, low latency interfaces. These contradicting needs pose a challenge to high performance chip development.") Koker at 61:24-41 ("Embodiments described herein provide techniques to disaggregate an architecture of a system on a chip integrated circuit into multiple distinct chiplets that can be packaged onto a common chassis. In one embodiment, a graphics processing unit or parallel processor is composed from diverse silicon chiplets that are separately manufactured. A chiplet is an at least partially packaged integrated circuit that includes distinct units of logic that can be assembled with other chiplets into a larger package. A diverse set of chiplets with different IP core logic can be assembled into a single device. Additionally the chiplets can be integrated into a base die or base chiplet using active interposer technology. The concepts described herein enable the interconnection and communication between the different forms of IP within the GPU. The development of IPs on different process may be mixed. This avoids the complexity of converging multiple IPs, especially on a large SoC with several flavors IPs, to the same process.") Koker at 61:42-50 ("Enabling the use of multiple process technologies improves the time to market and provides a cost-effective way to create multiple product SKUs. For customers, this means getting products that are more tailored to their requirements in a cost effective and timely manner. Additionally, the disaggregated IPs are more amenable to being power gated independently, components that are not in use on a given workload can be powered off, reducing overall power consumption.") Koker at 62:6-10 ("FIG. 28A-28B illustrate a hybrid logical/physical view of a disaggregated parallel processor, according to embodiments described herein. FIG. 28A illustrates a

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	disaggregated parallel compute system 2800. FIG. 28B illustrates a chiplet 2830 of the disaggregated parallel compute system 2800.") Koker at 62:11-25 ("As shown in FIG. 28A, a disaggregated compute system 2800 can include a parallel processor 2820 in which the various components of the parallel processor SOC are distributed across multiple chiplets. Each chiplet can be a distinct IP core that is independently designed and configured to communicate with other chiplets via one or more common interfaces. The chiplets include but are not limited to compute chiplets 2805, a media chiplet 2804, and memory chiplets 2806. Each chiplet can be separately manufactured using different process technologies. For example, compute chiplets 2805 may be manufactured using the smallest or most advanced process technology available at the time of fabrication, while memory chiplets 2806 or other chiplets (e.g., I/O, networking, etc.) may be manufactured using a larger or less advanced process technologies.") Koker at 62:26-41 ("The various chiplets can be bonded to a base die 2810 and configured to communicate with each other and logic within the base die 2810 via an interconnect layer 2812. In one embodiment, the base die 2810 can include global logic 2801, which can include scheduler 2811 and power management 2821 logic units, an interface 2802, a dispatch unit 2803, and an interconnect fabric module 2808 coupled with or integrated with one or more L3 cache banks 2809A-2809N. The interconnect fabric 2808 can be an inter-chiplet fabric that is integrated into the base die 2810. Logic chiplets can use the fabric 2808 to relay messages between the various chiplets. Additionally, L3 cache banks 2809A-2809N in the base die and/or L3 cache banks within the memory chiplets 2806 can cache data read from and transmitted to DRAM chiplets within the memory chiplets 2806 and to system memory of a host.") Koker at 63:26-30 ("FIG. 29A-29B illustrate a package view of a disaggregated parallel processor, according to embodiments. FIG. 29A illustrates a physical layout of a package assembly 2920. FIG. 29B illustrates an interconnect between multiple chiplets 2904, 2906 and an interconnect fabric 2940.") Koker at 63:31-43 ("As shown in FIG. 29A, a package assembly 2920 can include multiple units of hardware logic chiplets connected to a substrate 2910 (e.g., base die). The hardware logic chiplets can include special purpose hardware logic chiplets 2902, logic or I/O chiplets 2904,

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	and/or memory chiplets 2905. The hardware logic chiplets 2902 and logic or I/O chiplets 2904 may be implemented at least partly in configurable logic or fixed-functionality logic hardware and can include one or more portions of any of the processor core(s), graphics processor(s), parallel processors, or other accelerator devices described herein. The memory chiplets 2905 can be DRAM (e.g., GDDR, HIBM) memory or cache (SRAM) memory, as described and illustrated herein.") Koker at 63:44-54 ("Each chiplet can be fabricated as separate semiconductor die and coupled with the substrate 2910 via an interconnect structure 2903. The interconnect structure 2903 may be configured to route electrical signals between the various chiplets and logic within the substrate 2910. The interconnect structure 2903 can include interconnects such as, but not limited to bumps or pillars. In some embodiments, the interconnect structure 2903 may be configured to route electrical signals such as, for example, input/output (I/O) signals and/or power or ground signals associated with the operation of the logic, I/O and memory chiplets.") Koker at 63:55-63 ("In some embodiments, the substrate 2910 is an epoxy-based laminate substrate. The substrate 2910 may include other suitable types of substrates in other embodiments. The package assembly 2920 can be connected to other electrical devices via a package interconnect 2914. The package interconnect 2914 may be coupled to a surface of the substrate 2910 to route electrical signals to other electrical devices, such as a motherboard, other chipset, or multi-chip module.") Koker at 63:64-64:13 ("In some embodiments, a logic or I/O chiplet 2904 and a memory chiplet 2905 can be electrically coupled via a bridge 2917 that is configured to route electrical signals between the logic or I/O chiplet 2904 and a memory chiplet 2905. The bridge 2917 may be a dense interconnect structure that provides a route for electrical signals. The bridge 2917 may include a bridge substrate composed of glass or a suitable semiconductor material. Electrical routing features can be formed on the bridge substrate to provide a chip-to-chip connection between the logic or I/O chiplet 2904 and a memory chiplet 2905. The bridge 2917 may also be referred to as a silicon bridge or an interconnect bridge. For example, the bridge 2917, in some embodiments, is an Embedded Multi-die Interconnect Bridge (EMIB). In some embodiments, the bridge 2917 may simply be a direct connection from one chiplet to another chiplet.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 64:19-32 ("In various embodiments a package assembly 2920 can include fewer or greater number of components and chiplets that are interconnected by a fabric 2915 or one or more bridges 2917. The chiplets within the package assembly 2920 may be arranged in a 3D or 2.5D arrangement. In general, bridge structures 2917 may be used to facilitate a point to point interconnect between, for example, logic or I/O chiplets and memory chiplets. The fabric 2915 can be used to interconnect the various logic and/or I/O chiplets (e.g., chiplets 2902, 2904, 2911, 2913). with other logic and/or I/O chiplets. In one embodiment, the cache memory 2912 within the substrate can act as a global cache for the package assembly 2920, part of a distributed global cache, or as a dedicated cache for the fabric 2915.") Koker at 65:1-6 ("The compute or I/O logic 2939 can communicate with other logic or I/O chiplets via the interconnect fabric 2940. The compute or I/O logic 2939, in one embodiment, includes integrated fabric node logic 2939 that can communicate with the interconnect fabric 2940, such as the fabric interconnect node 2942 of FIG. 28B.") Koker at 66:51-61 ("FIG. 33 illustrates a method 3300 of configuring a fabric connection for a functional unit within a disaggregated parallel processor. The interconnect fabric of the disaggregated parallel processor described herein is configurable to transmit messages and/or signals for a variety of diverse components, which may be IP cores that have different designers and/or manufactures. The specific type of link used by each functional unit within a chiplet or base-die logic component can be configurable. In one embodiment the link is configurable at the fabric interconnect node used by the functional unit.") Koker at 68:30-46 ("Semiconductor dies are be tested during manufacturing to evaluate the integrated circuits formed on the dies. Standard tests for gross functionality are can be performed by probe testing the die at the wafer. Burn-in testing can be performed after the dies have been divided and packaged, or using a test harness for the bare die. Defective dies may be discarded. However, dies that pass initial testing but fail subsequent test at maximum frequency may operate correctly at a lower frequency. This binning-out process can select dies that under or over-perform and target those dies for higher or lower performance products with differing stock keeping units (SKUs). For monolithic system on a chip integrated circuits, the bin-out process is a coarse-grained process. While a processor with a few defective compute or graphics cores may

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	be down-binned, a minimum number of non-defective components must be present to meet minimum product requirements.") Koker at 68:47-55 ("With a disaggregated SoC architecture described herein, individual chiplets can be tested and binned at the chiplet level and SKU level can be determined for a product during assembly based on demand for a given product SKU. During assembly, different product SKUs with different configurations can be assembled with different amounts of memory, different functionality, and different performance, by specifying specific chiplets or different bins of the same chiplet design.") Koker at 68:56-69:6 ("A disaggregated processor package can be configured to accept interchangeable chiplets. Interchangeability is enabled by specifying a standard physical interconnect for chiplets that can enable the chiplet to interface with a fabric or bridge interconnect. Chiplets from different IP designers can conform to the common interconnect, enabling such chiplets to be interchangeable during assembly. The fabric and bridge interconnects logic on the chiplet can then be configured to confirm with the actual interconnect layout of the on-board logic of the chiplet. Additionally, data from chiplets can be transmitted across an inter-chiplet fabric using encapsulation, such that the actual data being transferred is opaque to the fabric, further enable interchangeability of the individual chiplets. With such an interchangeable design, higher or lower density memory can be inserted into memory chiplet slots, while compute or graphics chiplets with a higher or lower core count can be inserted into logic chiplet slots.") Koker at 69:7-18 ("Functionality can also be determined during assembly. For example, media chiplets can be added or excluded based on product specification and demand. In some products, on-package networking or other communication chiplets can be added. In some products, different types of host connections can be enabled using different chiplets. For example, where a host interconnect version change involves a change to interconnect logic without a change in physical form factor, an upgrade to the new interconnect version may be performed by changing host interconnect chiplets during assembly, without requiring a re-design of the SoC to insert new interconnect logic into a monolithic die.")

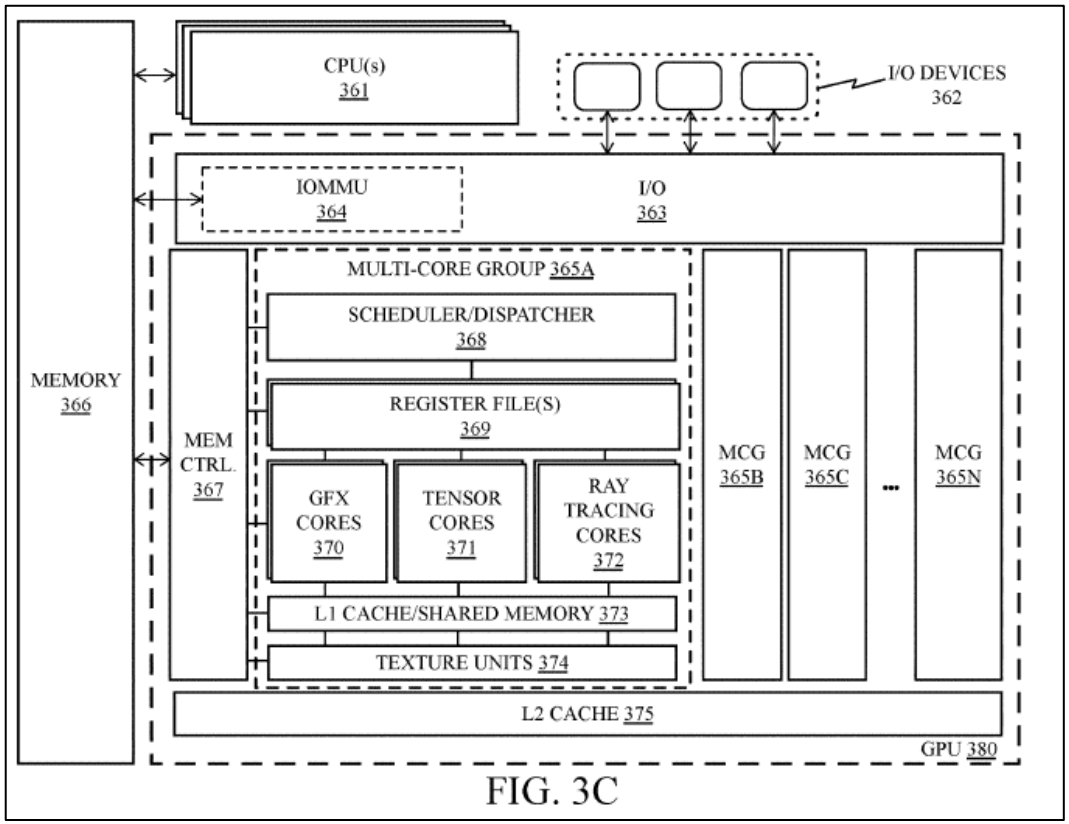
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 69:19-22 ("Chiplet binning can be further enabled by the provision of a chiplet testing harness that conforms the standardized chassis interfaces. This chiplet testing harness can enable rapid testing and binning of chiplets for different SKUs.") Koker at 69:23-40 ("FIG. 36 illustrates a parallel processor assembly 3600 including interchangeable chiplets 3602. The interchangeable chiplets 3602 can be assembled into standardized slots on one or more base chiplets 3604, 3608. The base chiplets 3604, 3608 can be coupled via a bridge interconnect 3606, which can be similar to the other bridge interconnects described herein. Memory chiplets can be connected to logic or I/O chiplets via a bridge interconnect. I/O and logic chiplets can communicate via an interconnect fabric. The base chiplets can each support one or more slots in a standardized format for one of logic or I/O or memory/cache. Different memory densities can be assembled into a chiplet slot based on the target SKU of the product. Additionally, logic chiplets with a different number of type of functional units can be selected at time of assembly based on the target SKU of the product. Additionally, chiplets containing IP logic cores of differing types can be inserted into the interchangeable chiplet slots.") Koker at 69:41-56 ("FIG. 37 illustrates an interchangeable chiplet system 3700, according to an embodiment. In one embodiment the interchangeable chiplet system 3700 includes at least one base chiplet 3710 including multiple memory chiplet slots 3701A-3701F and multiple logic chiplet slots 3702A-3702F. A logic chiplet slot (e.g., 3702A) and a memory chiplet slot (e.g., 3701A) can be connected by an interconnect bridge 3735, which can be similar to other interconnect bridges described herein. Logic chiplet slots 3702A-3702F can be interconnected via a fabric interconnect 3708. The fabric interconnect 3708 includes switching logic 3718 that can be configured to enable the relay of data packets between logic chiplets slots in a manner that is agnostic to the data by encapsulating the data into a fabric packet. The fabric packet can then be switched to the destination slot within the fabric interconnect 3708.") Koker at 69:57-70:11 ("The fabric interconnect 3708 can include one or more physical data channels. One or more programmable virtual channels can be carried by each physical channel. The virtual channels may be arbitrated independently, with channel access negotiated separately per virtual channel. Traffic over the virtual channels may be classified into one or more traffic

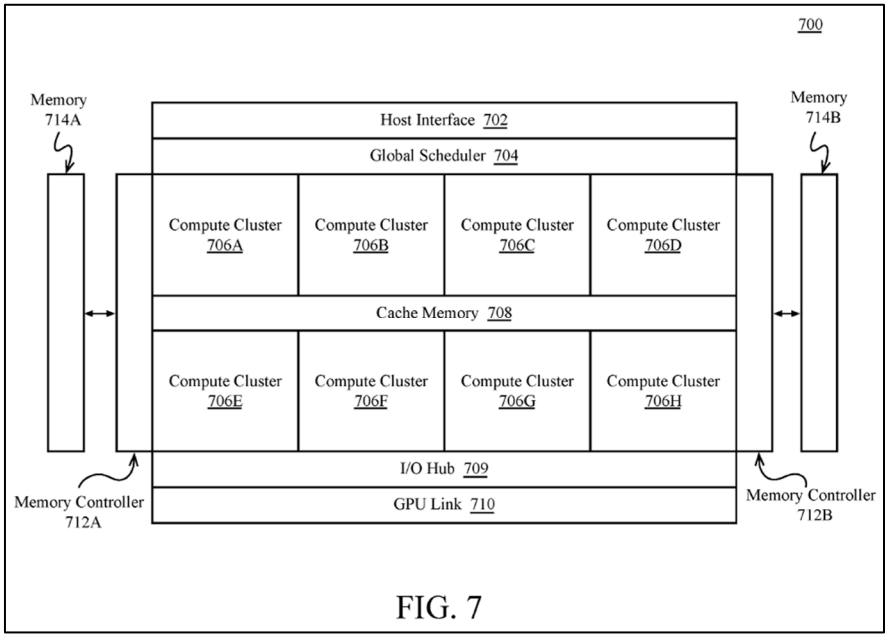
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	classes. In one embodiment, a prioritization system allows virtual channels and traffic classes to be assigned a relative priority for arbitration. In one embodiment, traffic balancing algorithms operate to maintain substantially equal bandwidth and throughput to each node coupled to the fabric. In one embodiment, the fabric interconnect logic operates at a higher clock rate than the nodes couples to the fabric, allowing reduction in interconnect width while maintaining the bandwidth requirements between nodes. Where certain nodes require higher bandwidth, multiple physical links can be combined to carry a single virtual channel, as in FIG. 31 above. In one embodiment, each physical link is separately clock gated when idle. An early indication upcoming activity can be used as a trigger to wake the physical link before data is to be transmitted.") Koker at 73:59-74:16 ("One embodiment provides for a general-purpose graphics processor comprising a base die including an interconnect fabric; and one or more chiplets coupled with the base die and the interconnect fabric via an interconnect structure, the interconnect structure to enable electrical communication between the one or more chiplets and the interconnect fabric. The one or more chiplets can include a first chiplet and a second chiplet, where the first chiplet is coupled with the base die and connected to the interconnect fabric via a first interconnect structure and the second chiplet is coupled with the base die and connected to the interconnect fabric via a second interconnect structure. The chiplets can include functional units configured to perform general-purpose graphics processing operations, media encode or decode operations, matrix operation acceleration, and/or ray-tracing. In one embodiment the chiplets include a network processor and a physical network interface (e.g., network port, wireless radio, etc.). The chiplets can additionally include memory, which may be cache memory or DRAM. Each chiplet may be separately and independently power gated. Additionally, logic or memory can be included in the base die. In one embodiment the base die includes cache memory. The cache memory in the base die can be a processor-wide cache. Base die cache memory can be configured to work in concert with cache memory in a chiplet.") Koker at 74:17-35 ("One embodiment provides for a data processing system comprising a general-purpose graphics processor comprising a base die including an interconnect fabric and multiple chiplets coupled with the base die and the interconnect fabric via multiple interconnect structures, the multiple interconnect structures to enable electrical communication between the

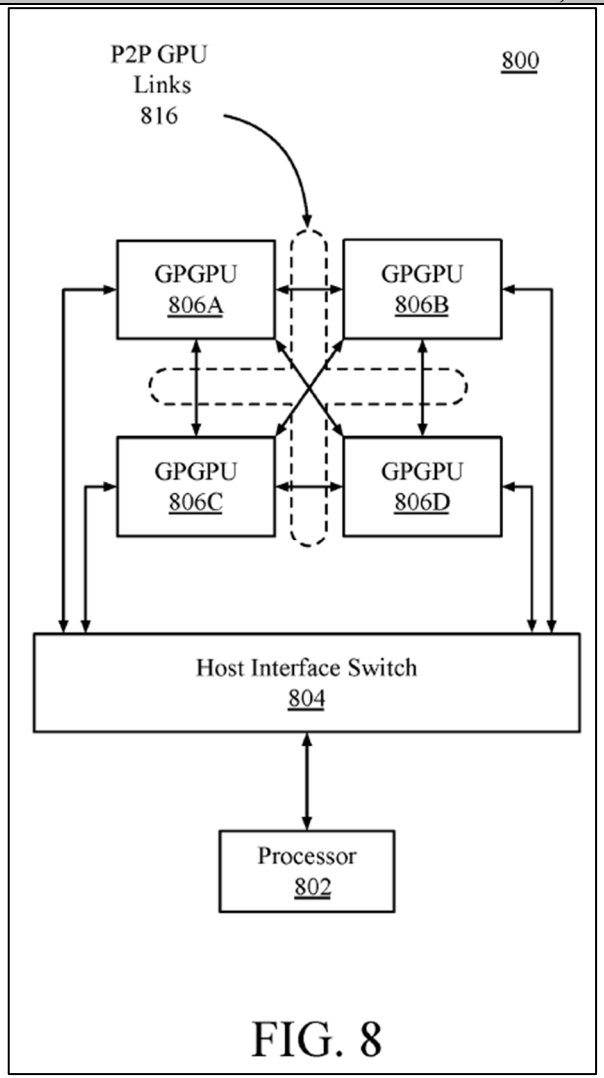
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	multiple chiplets and the interconnect fabric, wherein the interconnect fabric is to receive a message or signal from a first fabric interface node associated with a first chiplet of the multiple chiplets and relay the message or signal to a second fabric interface node associated with a second chiplet of the multiple chiplets. The interconnect fabric can transmit messages or signals via multiple virtual channels over multiple physical links of the interconnect fabric. In one embodiment multiple virtual channels can be transmitted across a single physical link. In one embodiment, a single virtual channel can be transmitted across multiple physical links. Idle physical links can be separately power gated.") Koker at 75:27-46 ("One embodiment provides for a general-purpose graphics processor comprising a base die including an interconnect fabric and one or more chiplets coupled with the base die and the interconnect fabric via an interconnect structure, the interconnect structure to enable electrical communication between the one or more chiplets and the interconnect fabric, wherein the one or more chiplets are interchangeable during assembly of the general-purpose graphics processor. The one or more chiplets include a memory chiplet having memory cells associated with a memory device. The memory chiplet is coupled to a first memory chiplet slot. The one or more chiplets can additionally include a first logic chiplet and a second logic chiplet. The first logic chiplet can be coupled with the base die and connected to the interconnect fabric via a first interconnect structure. The first interconnect structure is bonded to a first logic chiplet slot. The second logic chiplet is coupled with the base die and connected to the interconnect fabric via a second interconnect structure. The second interconnect structure is bonded to a second logic chiplet slot.") Koker at 76:5-28 ("One embodiment provides for a data processing system comprising a general-purpose graphics processor comprising a base die including an interconnect fabric and multiple chiplets coupled with the base die and the interconnect fabric via multiple interconnect structures. The multiple interconnect structures enable electrical communication between the multiple chiplets and the interconnect fabric. The interconnect fabric can receive a fabric packet from a first fabric interface node associated with a first logic chiplet of the multiple chiplets and relay the fabric packet to a second fabric interface node associated with a second logic chiplet of the multiple chiplets. The interconnect fabric can transmit the fabric packet via multiple virtual channels over multiple physical links of the interconnect fabric. The interconnect fabric can

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	transmit fabric packets associated with a single virtual channel across multiple physical links of the interconnect fabric. The interconnect fabric can also transmit fabric packets for multiple virtual channels across a single physical link of the interconnect fabric. A physical link of the multiple physical links can be power gated when the physical link is idle. In one embodiment, a fabric packet for a virtual channel can be associated with one or more traffic classes. A virtual channel and a traffic class can each have an associated priority.") Koker at 76:41-53 ("Of course, one or more parts of an embodiment may be implemented using different combinations of software, firmware, and/or hardware. Throughout this detailed description, for the purposes of explanation, numerous specific details were set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the embodiments may be practiced without some of these specific details. In certain instances, well-known structures and functions were not described in elaborate detail to avoid obscuring the inventive subject matter of the embodiments. Accordingly, the scope and spirit of the invention should be judged in terms of the claims that follow.") Koker at Claim 1 ("1. A general-purpose graphics processor comprising: a base die including an interconnect fabric; and one or more chiplets coupled with the base die and the interconnect fabric via an interconnect structure, the interconnect structure to enable electrical communication between the one or more chiplets and the interconnect fabric, wherein the one or more chiplets are interchangeable during assembly of the general-purpose graphics processor, the one or more chiplets include a first logic chiplet coupled with the base die and connected to the interconnect fabric via a first interconnect structure, the first interconnect structure is bonded to a first logic chiplet slot, and the first logic chiplet includes functional units configured to perform general-purpose graphics processing operations; and wherein the first logic chiplet slot is configurable to accept a logic chiplet selected from a group including the first logic chiplet and a third logic chiplet that includes functional units configured to perform matrix acceleration operations.").

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at Claim 5 ("5. The general-purpose graphics processor as in claim 4, wherein the second logic chiplet is coupled with the base die and connected to the interconnect fabric via a second interconnect structure, the second interconnect structure bonded to a second logic chiplet slot."). Koker at Claim 6 ("6. The general-purpose graphics processor as in claim 5, wherein the interconnect structure is a silicon interconnect structure.") Koker at Claim 12 ("12. A data processing system comprising: a general-purpose graphics processor comprising a base die including an interconnect fabric and one or more chiplets coupled with the base die and the interconnect fabric via an interconnect structure, the interconnect structure to enable electrical communication between the one or more chiplets and the interconnect fabric, wherein the one or more chiplets are interchangeable during assembly of the general-purpose graphics processor, the one or more chiplets include a first logic chiplet coupled with the base die and connected to the interconnect fabric via a first interconnect structure, the first interconnect structure bonded to a first logic chiplet slot, the first logic chiplet slot is configurable to accept a logic chiplet selected from a group including the first logic chiplet and a third logic chiplet, the first logic chiplet includes functional units configured to perform general-purpose graphics processing operations, and the third logic chiplet includes functional units configured to perform matrix acceleration operations; and a memory coupled with the general-purpose graphics processor.") This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
1[D]: wherein the passive crosslink is dedicated for inter-chiplet communications.	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation.

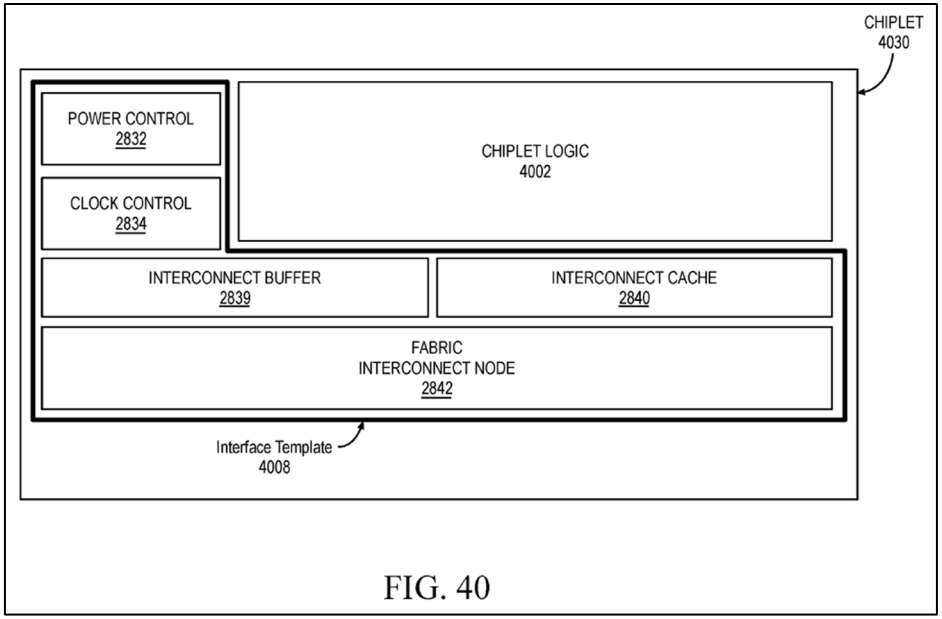
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	See, e.g., Koker at Fig. 3C.  FIG. 3C Koker at Fig. 7.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 7 Koker at Fig. 8.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 8

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 29:16-36 ("Multiple instances of the GPGPU 700 can be configured to operate as a compute cluster. The communication mechanism used by the compute cluster for synchronization and data exchange varies across embodiments. In one embodiment, the multiple instances of the GPGPU 700 communicate over the host interface 702. In one embodiment the GPGPU 700 includes an I/O hub 709 that couples the GPGPU 700 with a GPU link 710 that enables a direct connection to other instances of the GPGPU. In one embodiment the GPU link 710 is coupled to a dedicated GPU-to-GPU bridge that enables communication and synchronization between multiple instances of the GPGPU 700. In one embodiment the GPU link 710 couples with a high-speed interconnect to transmit and receive data to other GPGPUs or parallel processors. In one embodiment the multiple instances of the GPGPU 700 are located in separate data processing systems and communicate via a network device that is accessible via the host interface 702. In one embodiment the GPU link 710 can be configured to enable a connection to a host processor in addition to or as an alternative to the host interface 702.") Koker at 29:52-30-11 ("FIG. 8 illustrates a multi-GPU computing system 800, according to an embodiment. The multi-GPU computing system 800 can include a processor 802 coupled to multiple GPGPUs 806A-806D via a host interface switch 804. The host interface switch 804, in one embodiment, is a PCI express switch device that couples the processor 802 to a PCI express bus over which the processor 802 can communicate with the set of GPGPUs 806A-806D. Each of the multiple GPGPUs 806A-806D can be an instance of the GPGPU 700 of FIG. 7. The GPGPUs 806A-806D can interconnect via a set of high-speed point to point GPU to GPU links 816. The high-speed GPU to GPU links can connect to each of the GPGPUs 806A-806D via a dedicated GPU link, such as the GPU link 710 as in FIG. 7. The P2P GPU links 816 enable direct communication between each of the GPGPUs 806A-806D without requiring communication over the host interface bus to which the processor 802 is connected. With GPU-to-GPU traffic directed to the P2P GPU links, the host interface bus remains available for system memory access or to communicate with other instances of the multi-GPU computing system 800, for example, via one or more network devices. While in the illustrated embodiment the GPGPUs 806A-D connect to the processor 802 via the host interface switch 804, in one embodiment the processor 802 includes direct support for the P2P GPU links 816 and can connect directly to the GPGPUs 806A-806D.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 61:24-41 ("Embodiments described herein provide techniques to disaggregate an architecture of a system on a chip integrated circuit into multiple distinct chiplets that can be packaged onto a common chassis. In one embodiment, a graphics processing unit or parallel processor is composed from diverse silicon chiplets that are separately manufactured. A chiplet is an at least partially packaged integrated circuit that includes distinct units of logic that can be assembled with other chiplets into a larger package. A diverse set of chiplets with different IP core logic can be assembled into a single device. Additionally the chiplets can be integrated into a base die or base chiplet using active interposer technology. The concepts described herein enable the interconnection and communication between the different forms of IP within the GPU. The development of IPs on different process may be mixed. This avoids the complexity of converging multiple IPs, especially on a large SoC with several flavors IPs, to the same process.") *See also* evidence cited for '803 patent claim 1[C]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
2. The system of claim 1, wherein the passive crosslink comprises a passive interposer die.	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this claim. *See* evidence cited for '803 patent claim 1[C]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.

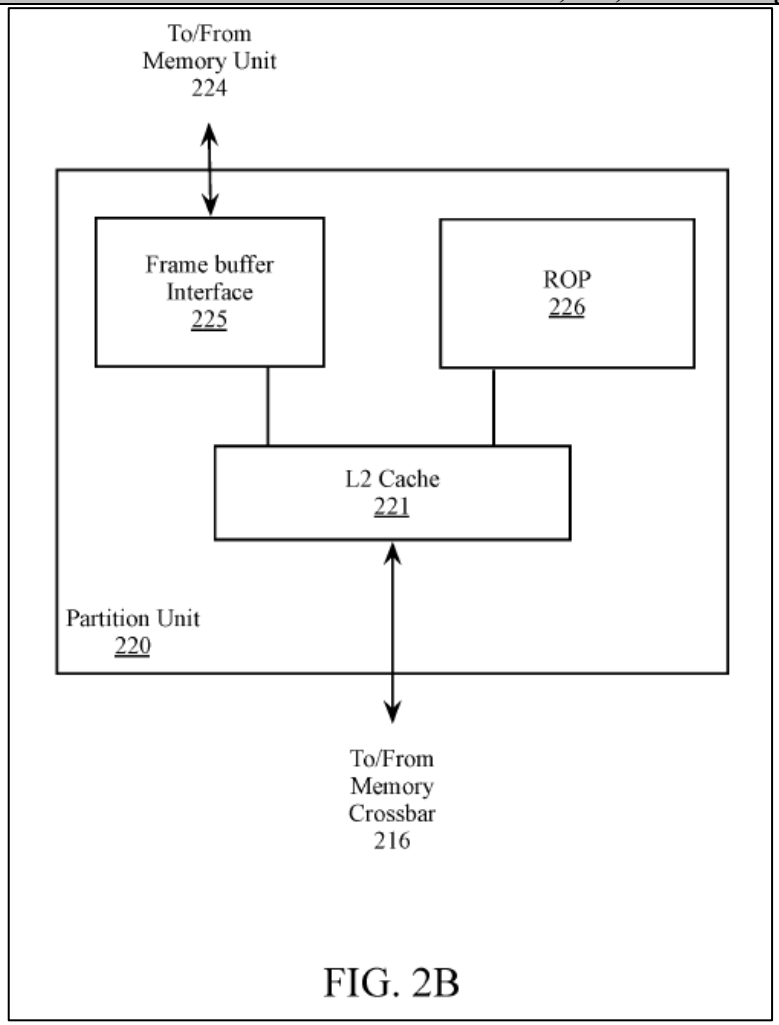
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
3[A]: The system of claim 1, wherein the first GPU chiplet comprises a first PHY region including conductor structures for chiplet-to-chiplet communications,	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See, e.g.*, Koker at Fig. 40.  FIG. 40 Koker at 66:62-67:8 ("In one embodiment a fabric interconnect node can receive bandwidth configuration data for a functional unit that is to be configured to communicate via an interconnect fabric within a parallel processor package (block 3302). The configuration data can be provided statically during initial assembly and provisioning for the disaggregated parallel processor of can be configured dynamically during initialization of the disaggregated parallel processor. For dynamic initialization, the fabric interconnect node can receive a bandwidth configuration request from the functional unit that specifies the physical width and frequency of

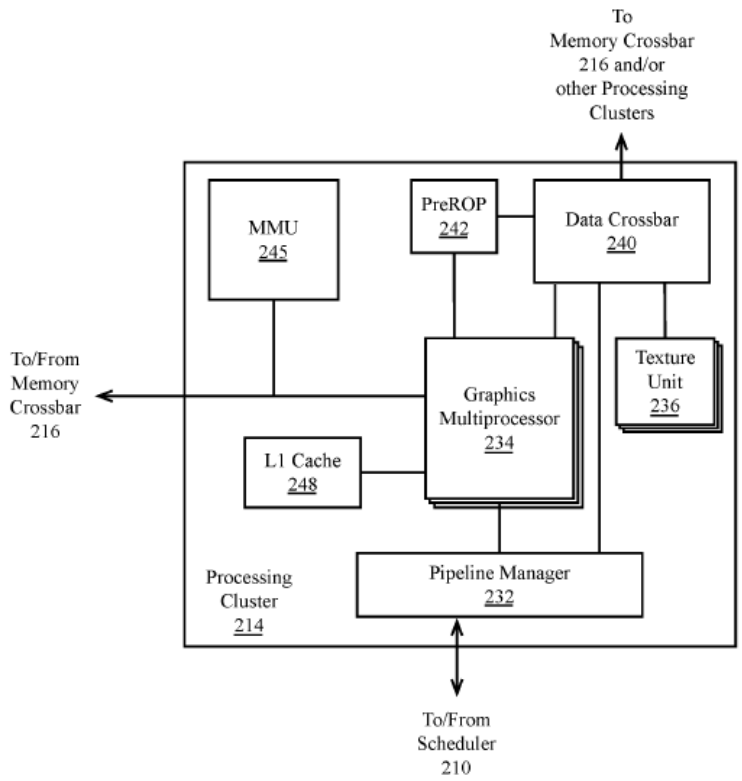
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	the interconnect between the functional unit and the fabric interconnect node, as well as the bandwidth requirement of the functional unit.") Koker at 67:9-15 ("The fabric interconnect node can then analyze configured interconnect width and frequency for the functional unit (block 3304). The fabric interconnect node can then configure convergent and/or divergent links for the functional unit (block 3306). Once configured, the fabric interconnect node can relay messages and/or signals for the functional unit across configured links (block 3308).") Koker at 67:16-22 ("FIG. 34 illustrates a method 3400 of relaying messages and/or signals across an interconnect fabric within a disaggregated parallel processor. The interconnect fabric of the disaggregated parallel processor described herein can relay messages and/or signals across one or more layers of the disaggregated parallel processor, while crossing multiple clock domains.") Koker at 67:23-38 ("In one embodiment a first functional unit within a chiplet or base die of a processor can generate data in the form of a message or signal to be transmitted (block 3402). The first functional unit can transmit the message or signal to the interconnect fabric via a first fabric interface node (block 3404). The fabric interface node can converge or diverge virtual channels into physical transport links (block 3406), as shown in FIG. 31 and FIG. 32. The message or signal to be transmitted can be associated with a virtual channel and transmitted via the associated virtual channel. How the fabric interconnect performs forwarding and/or switching operations for the message or signal can be influenced by the virtual channel assigned to the message or signal. Additionally, multiple virtual channels can be combined into a single physical link or a virtual channel can be carried by multiple physical links.") Koker at 67:39-50 ("In one embodiment the interconnect fabric can transport the message or signal across multiple clock domains within one or more transport and/or data-link layers (block 3408). One or more clock crossing modules including high-speed memory and domain crossing FIFOs can be used to cross the multiple clock domains. In one embodiment, each chiplet can have a separate clock domain relative to the interconnect fabric. The interconnect fabric can also have multiple clock domains. Transmitting the message or signal across the multiple clock

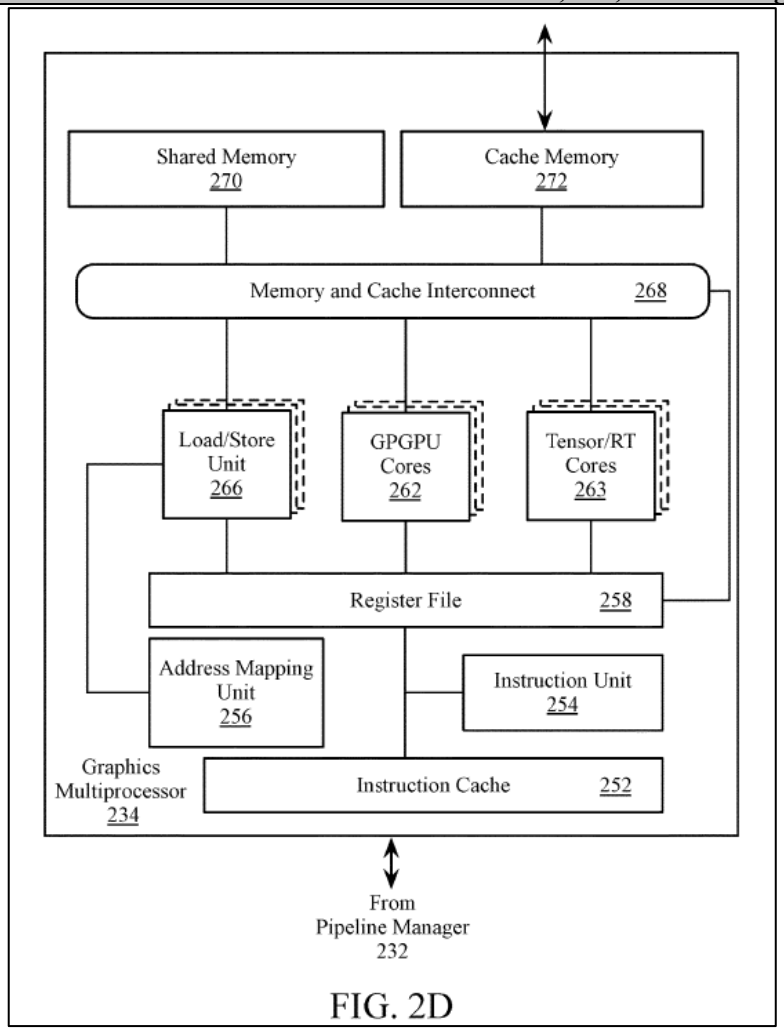
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	domains can include switching the message or signal using switching logic within the interconnect fabric.") Koker at 67:51-61 ("A second fabric interface node can receive the message or signal (block 3410). The second fabric interface node can then diverge or converge virtual channels from physical transport links at the second fabric interface node (block 3412). Multiple virtual channels can be diverged from a single physical link or a virtual channel can be converged from multiple physical links. A second functional unit within a chiplet or base die of the processor can then receive data in the form of the message or signal at second hardware logic (block 3414). The second functional unit can then perform an operation based on the received data.") Koker at 72:23-41 ("FIG. 40 illustrates a modular architecture for interchangeable chiplets, according to an embodiment. In one embodiment a chiplet design 4030 can be made interchangeable by adapting the chiplet logic 4002 for interoperability with an interface template 4008. The interface template 4008 can include standardized logic such as power control 2832 and clock control 2834 logic, and interconnect buffer 2839, interconnect cache 2840, a fabric interconnect node 2842, as in chiplet 2830 of FIG. 28B. The IP designer can then provide the chiplet logic 4002, which is designed to interface with the interface template 4008. The specifics of the chiplet logic 4002 can vary and can include execution units, compute units, or streaming multiprocessors as described herein. The chiplet logic 4002 can also include media encode and/or decode logic, matrix acceleration logic, or ray-tracing logic. For memory chiplets, the chiplet logic 4002 can be replaced with memory cells and the fabric interconnect node can be replaced with interconnect bridge I/O circuits, for example, as illustrated in the memory chiplet 2906 of FIG. 29B.") Koker at 75:9-26 ("Further embodiments provide for a disaggregated processor package that can be configured to accept interchangeable chiplets. Interchangeability is enabled by specifying a standard physical interconnect for chiplets that can enable the chiplet to interface with a fabric or bridge interconnect. Chiplets from different IP designers can conform to the common interconnect, enabling such chiplets to be interchangeable during assembly. The fabric and bridge interconnects logic on the chiplet can then be configured to confirm with the actual interconnect layout of the on-board logic of the chiplet. Additionally, data from chiplets can be transmitted

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	across an inter-chiplet fabric using encapsulation, such that the actual data being transferred is opaque to the fabric, further enable interchangeability of the individual chiplets. With such an interchangeable design, higher or lower density memory can be inserted into memory chiplet slots, while compute or graphics chiplets with a higher or lower core count can be inserted into logic chiplet slots.") *See* evidence cited for '803 patent claim 1[C]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
3[B]: and further wherein the second GPU chiplet comprises a second PHY region including conductor structures for chiplet-to-chiplet communications.	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See* evidence cited for '803 patent claim 3[A]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
7. The system of claim 1, wherein the passive crosslink communicably couples all GPU chiplets in the GPU chiplet array.	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this claim. *See* evidence cited for '803 patent claim 1[D]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
8[A]: The system of claim 1, further comprising: a first cache memory hierarchy at the first GPU chiplet,	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See, e.g.*, Koker at Fig. 2B.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 2B Koker at Fig. 2C.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 2C Koker at Fig. 2D.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 2D Koker at Fig. 3A.

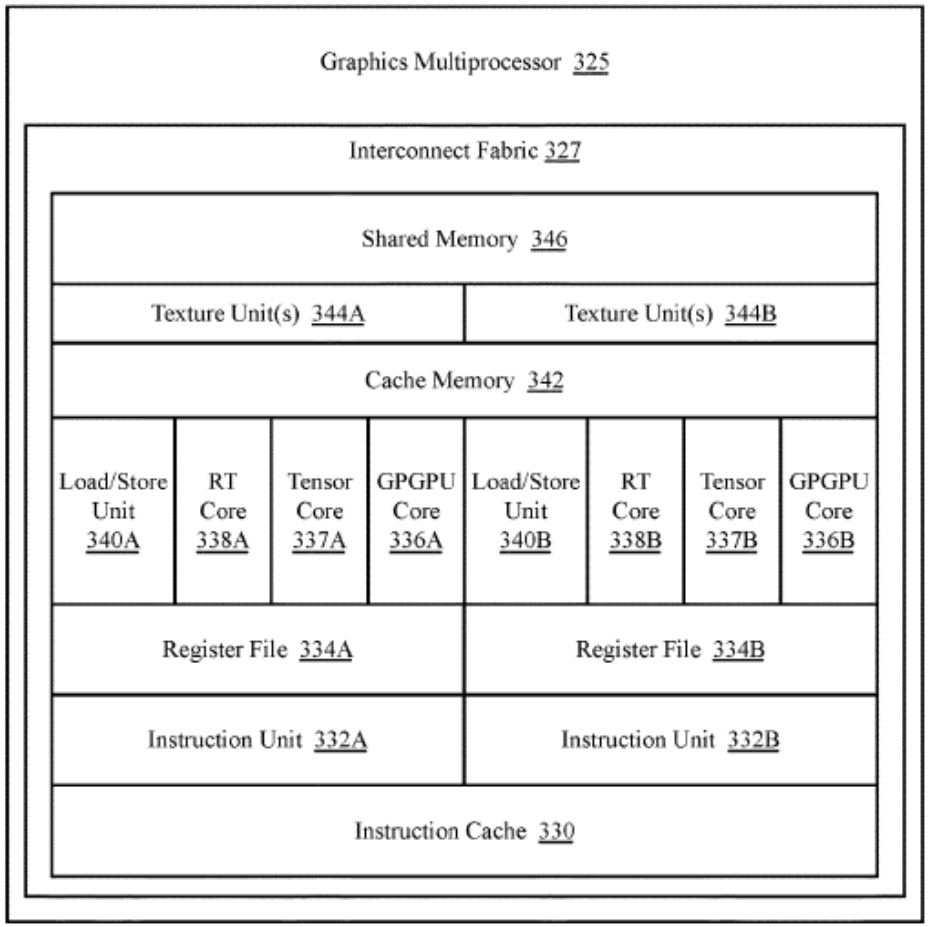


FIG. 3A

Onesta IP, LLC, Exhibit No. 2105
Case No. IPR2026-00011
Page 78 of 115

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at Fig. 3B.

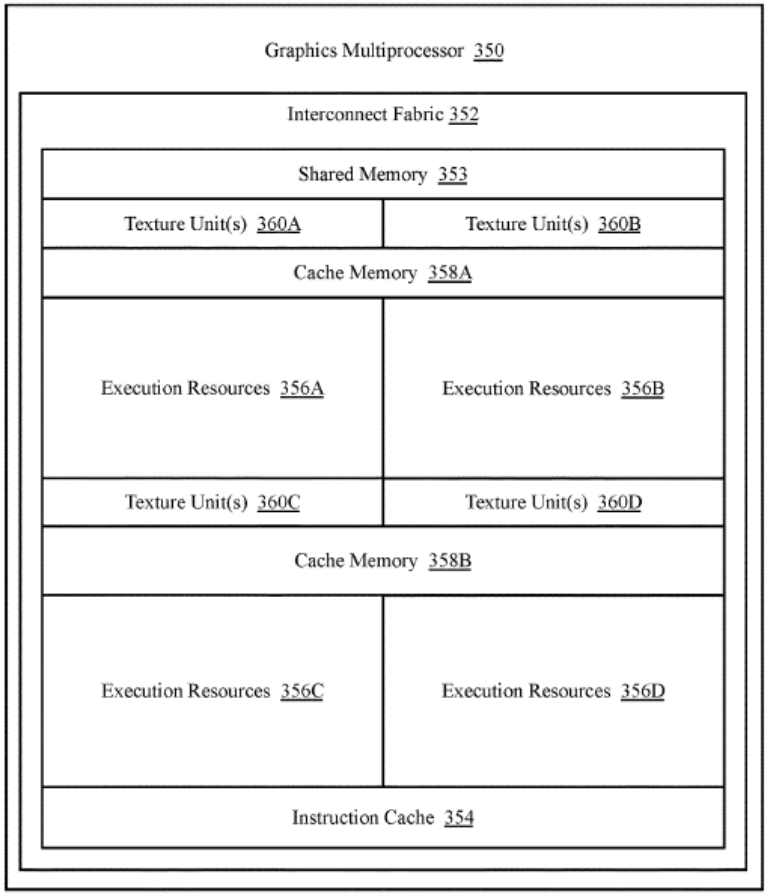
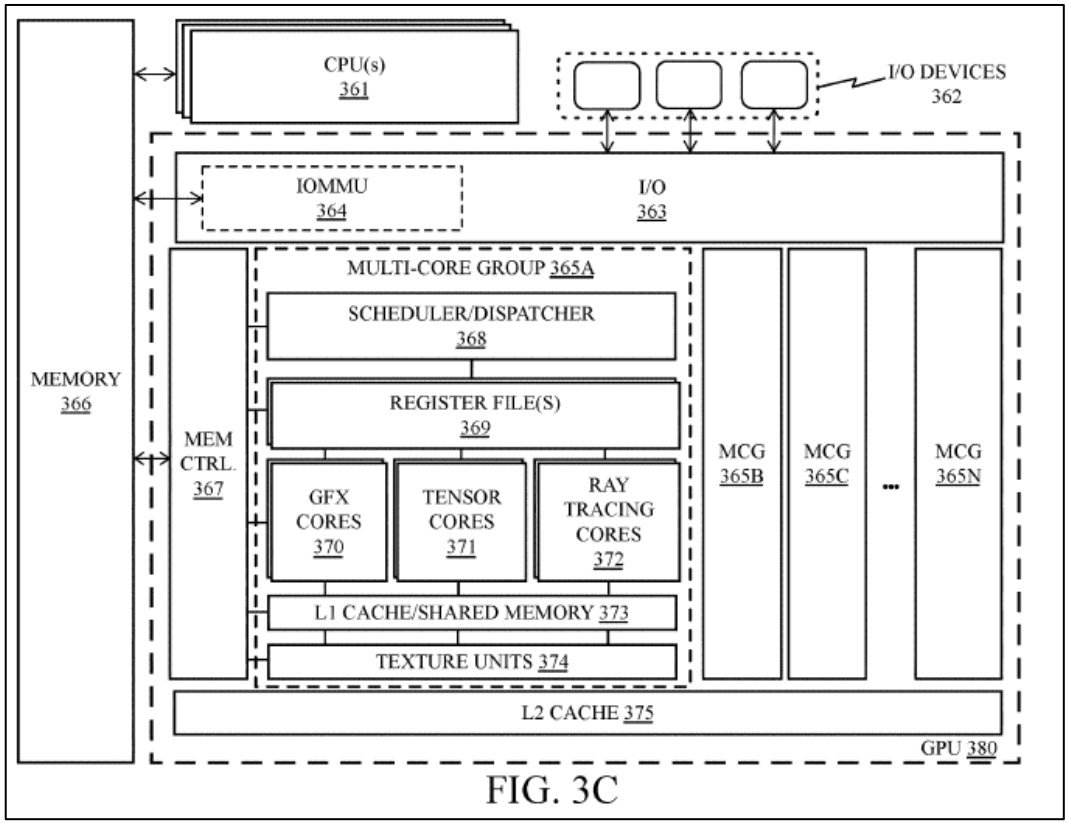
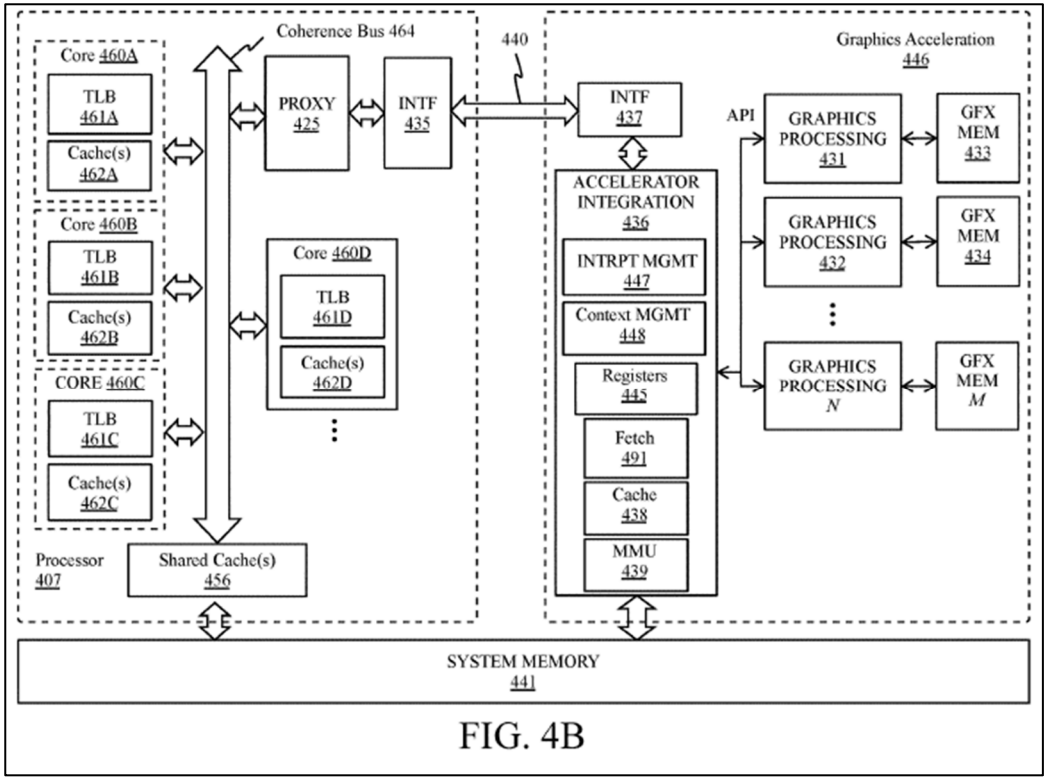
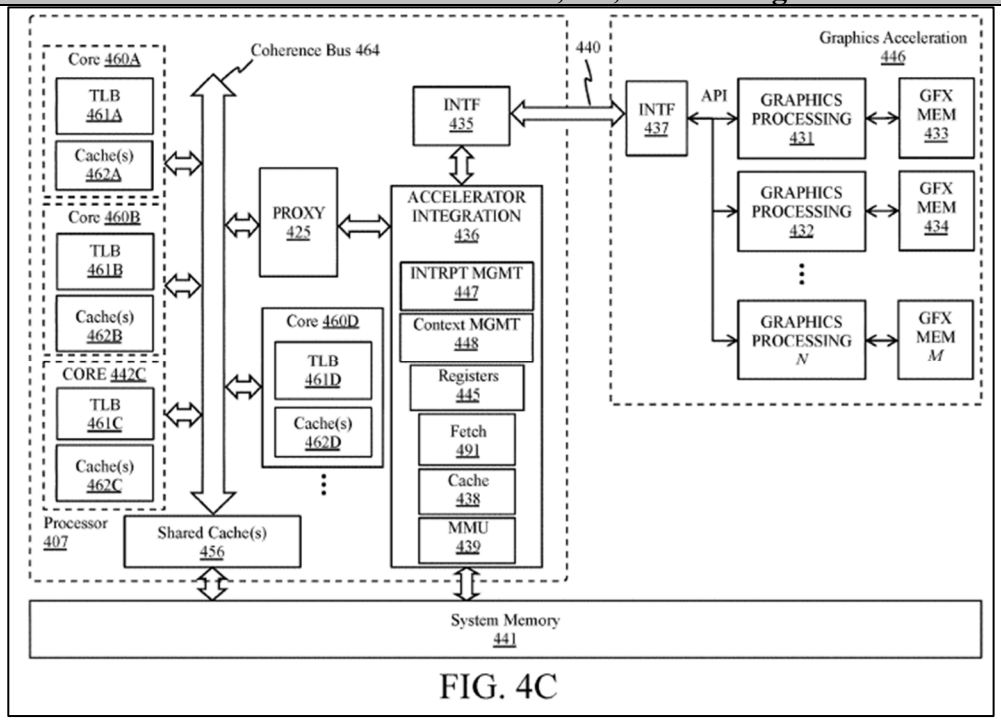
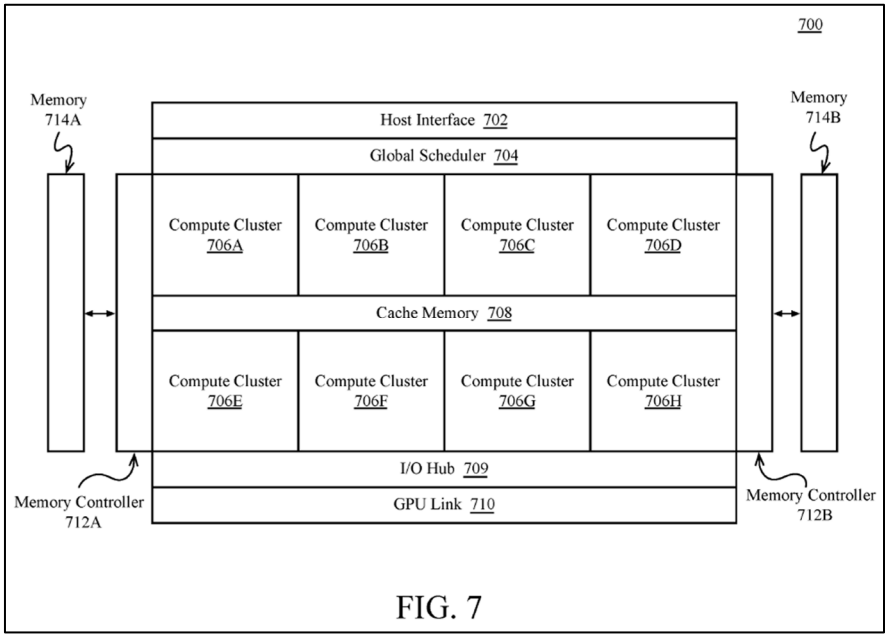


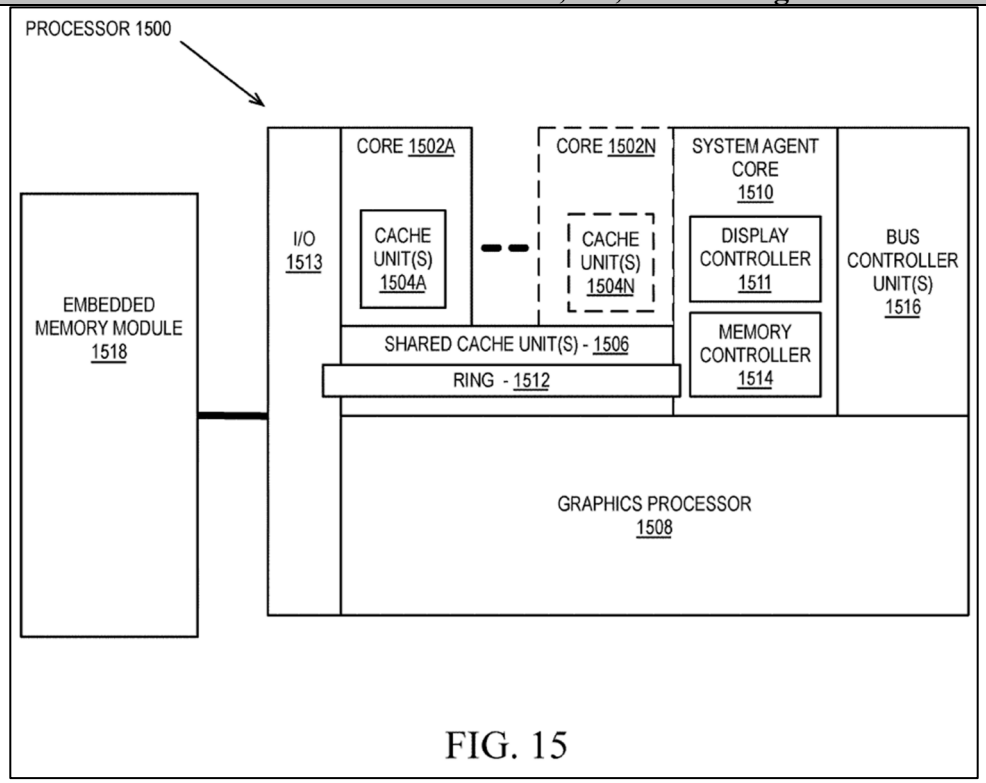
FIG. 3B

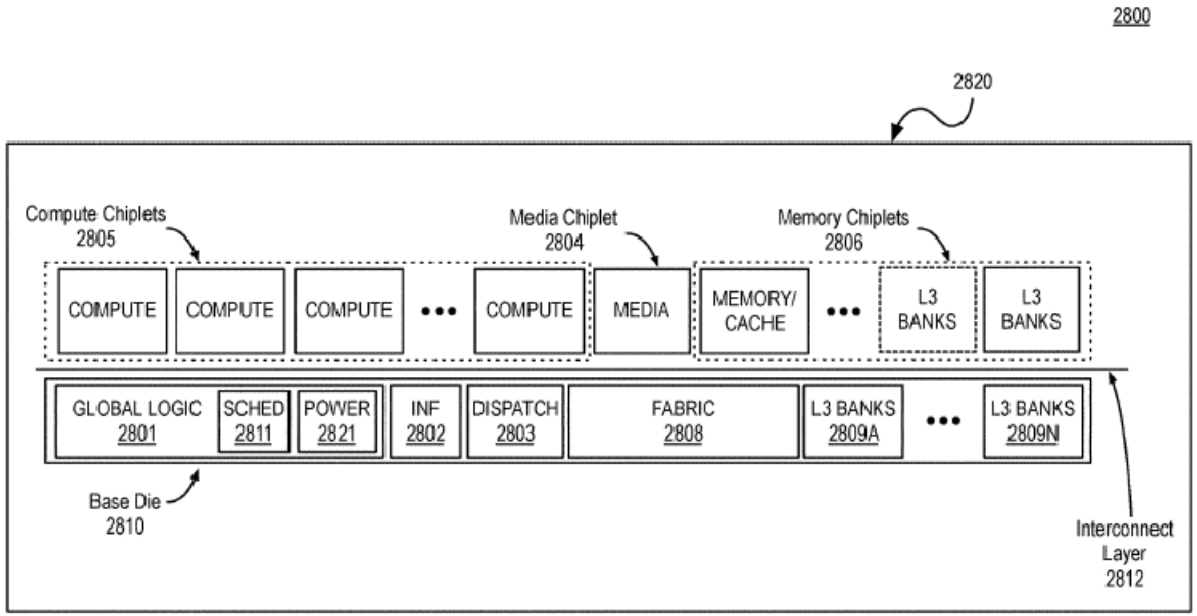
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at Fig. 3C.  FIG. 3C Koker at Fig. 4B.

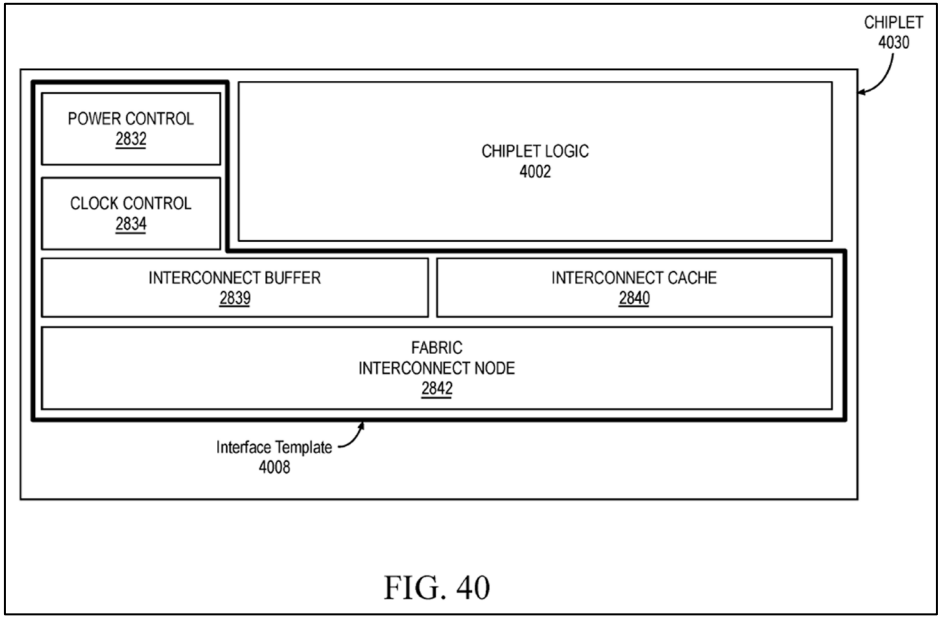
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4B Koker at Fig. 4C.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4C Koker at Fig. 7.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 7 Koker at Fig. 15.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 15 Koker at Fig. 28A.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 28A Koker at Fig. 40.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 40 Koker at 9:63-10:12 ("In one embodiment the graphics multiprocessor 234 includes an internal cache memory to perform load and store operations. In one embodiment, the graphics multiprocessor 234 can forego an internal cache and use a cache memory (e.g., L1 cache 248) within the processing cluster 214. Each graphics multiprocessor 234 also has access to L2 caches within the partition units (e.g., partition units 220A-220N of FIG. 2A) that are shared among all processing clusters 214 and may be used to transfer data between threads. The graphics multiprocessor 234 may also access off-chip global memory, which can include one or more of local parallel processor memory and/or system memory. Any memory external to the parallel processing unit 202 may be used as global memory. Embodiments in which the processing cluster 214 includes multiple instances of the graphics multiprocessor 234 can share common instructions and data, which may be stored in the L1 cache 248.") Koker at 10:60-11:7 ("FIG. 2D shows a graphics multiprocessor 234, according to one embodiment. In such embodiment the graphics multiprocessor 234 couples with the pipeline manager 232 of the processing cluster 214. The graphics multiprocessor 234 has an execution

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	pipeline including but not limited to an instruction cache 252, an instruction unit 254, an address mapping unit 256, a register file 258, one or more general purpose graphics processing unit (GPGPU) cores 262, and one or more load/store units 266. The GPGPU cores 262 and load/store units 266 are coupled with cache memory 272 and shared memory 270 via a memory and cache interconnect 268. In one embodiment the graphics multiprocessor 234 additionally includes tensor and/or ray-tracing cores 263 that include hardware logic to accelerate matrix and/or ray-tracing operations.") Koker at 12:18-27, ("FIG. 3A-3C illustrate additional graphics multiprocessors, according to embodiments. FIG. 3A-3B illustrate graphics multiprocessors 325, 350, which are variants of the graphics multiprocessor 234 of FIG. 2C. FIG. 3C illustrates a graphics processing unit (GPU) 380 which includes dedicated sets of graphics processing resources arranged into multi-core groups 365A-365N. The illustrated graphics multiprocessors 325, 350 and the multi-core groups 365A-365N can be streaming multiprocessor (SM) capable of simultaneous execution of a large number of execution threads.") Koker at 13:36-42 ("FIG. 3C illustrates a graphics processing unit (GPU) **380** which includes dedicated sets of graphics processing resources arranged into multi-core groups **365**A-N. While the details of only a single multi-core group **365**A are provided, it will be appreciated that the other multi-core groups **365**B-**365**N may be equipped with the same or similar sets of graphics processing resources.") Koker at 13:43-55 ("As illustrated, a multi-core group **365**A 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.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 13:56-14:2 ("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).") Koker at 17:29-46 ("The illustrated processor 407 includes a plurality of cores 460A-460D, each with a translation lookaside buffer 461A-461D and one or more caches 462A-462D. The cores may include various other components for executing instructions and processing data which are not illustrated to avoid obscuring the underlying principles of the invention (e.g., instruction fetch units, branch prediction units, decoders, execution units, reorder buffers, etc.). 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. The processor 407 and the graphics accelerator integration module 446 connect with system memory 441, which may include processor memories 401-402.") Koker at 17:47-58 ("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.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 17:59-67 ("In one embodiment, a proxy circuit 425 communicatively couples the graphics acceleration module 446 to the coherence bus 464, allowing the graphics acceleration module 446 to participate in the cache coherence protocol as a peer of the cores. In particular, an interface 435 provides connectivity to the proxy circuit 425 over high-speed link 440 (e.g., a PCIe bus, NVLink, etc.) and an interface 437 connects the graphics acceleration module 446 to the high-speed link 440."). Koker at 18:18-36 ("In one embodiment, the accelerator integration circuit 436 includes a memory management unit (MMU) 439 for performing various memory management functions such as virtual-to-physical memory translations (also referred to as effective-to-real memory translations) and memory access protocols for accessing system memory 441. The MMU 439 may also include a translation lookaside buffer (TLB) (not shown) for caching the virtual/effective to physical/real address translations. In one implementation, a cache 438 stores commands and data for efficient access by the graphics processing engines 431-432, N. In one embodiment, the data stored in cache 438 and graphics memories 433-434, M is kept coherent with the core caches 462A-462D, 456 and system memory 411. As mentioned, this may be accomplished via proxy circuit 425 which takes part in the cache coherency mechanism on behalf of cache 438 and memories 433-434, M (e.g., sending updates to the cache 438 related to modifications/accesses of cache lines on processor caches 462A-462D, 456 and receiving updates from the cache 438).") Koker at 23:30-40 ("In one embodiment, bias/coherence management circuitry 494A-494E within one or more of the MMUs 439A-439E ensures cache coherence between the caches of the host processors (e.g., 405) and the GPUs 410-413 and implements biasing techniques indicating the physical memories in which certain types of data should be stored. While multiple instances of bias/coherence management circuitry 494A-494E are illustrated in FIG. 4F, the bias/coherence circuitry may be implemented within the MMU of one or more host processors 405 and/or within the accelerator integration circuit 436.") Koker at 23:41-63 ("One embodiment allows GPU-attached memory 420-423 to be mapped as part of system memory, and accessed using shared virtual memory (SVM) technology, but without suffering the typical performance drawbacks associated with full system cache

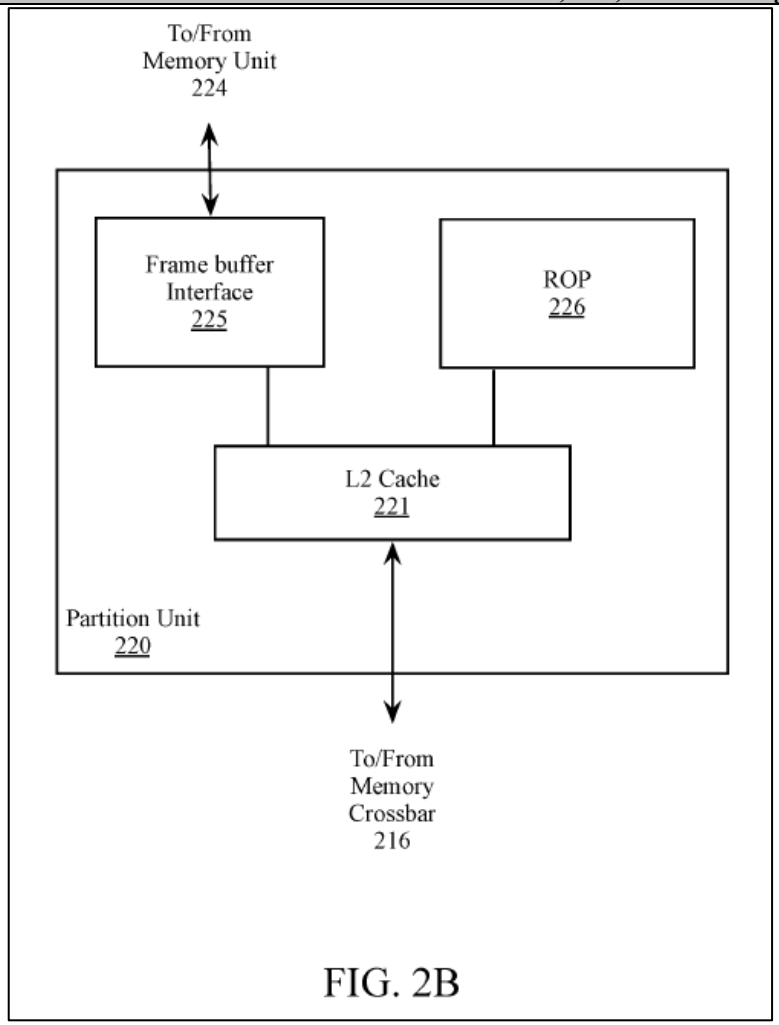
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	coherence. The ability to GPU-attached memory 420-423 to be accessed as system memory without onerous cache coherence overhead provides a beneficial operating environment for GPU offload. This arrangement allows the host processor 405 software to setup operands and access computation results, without the overhead of tradition I/O DMA data copies. Such traditional copies involve driver calls, interrupts and memory mapped I/O (MMIO) accesses that are all inefficient relative to simple memory accesses. At the same time, the ability to access GPU attached memory 420-423 without cache coherence overheads can be critical to the execution time of an offloaded computation. In cases with substantial streaming write memory traffic, for example, cache coherence overhead can significantly reduce the effective write bandwidth seen by a GPU 410-413. The efficiency of operand setup, the efficiency of results access, and the efficiency of GPU computation all play a role in determining the effectiveness of GPU offload.") Koker at 24:35-43 ("In one embodiment, cache coherency is maintained by temporarily rendering GPU-biased pages uncacheable by the host processor 405. To access these pages, the processor 405 may request access from the GPU 410 which may or may not grant access right away, depending on the implementation. Thus, to reduce communication between the host processor 405 and GPU 410 it is beneficial to ensure that GPU-biased pages are those which are required by the GPU but not the host processor 405 and vice versa.") Koker at 28:48-59 ("The GPGPU 700 includes a host interface 702 to enable a connection with a host processor. In one embodiment the host interface 702 is a PCI Express interface. However, the host interface can also be a vendor specific communications interface or communications fabric. The GPGPU 700 receives commands from the host processor and uses a global scheduler 704 to distribute execution threads associated with those commands to a set of compute clusters 706A-706H. The compute clusters 706A-706H share a cache memory 708. The cache memory 708 can serve as a higher-level cache for cache memories within the compute clusters 706A-706H.") Koker at 37:60-38:8 ("In some embodiments, the processor 1402 includes cache memory 1404. Depending on the architecture, the processor 1402 can have a single internal cache or multiple levels of internal cache. In some embodiments, the cache memory is shared among various components of the processor 1402. In some embodiments, the processor 1402 also uses an

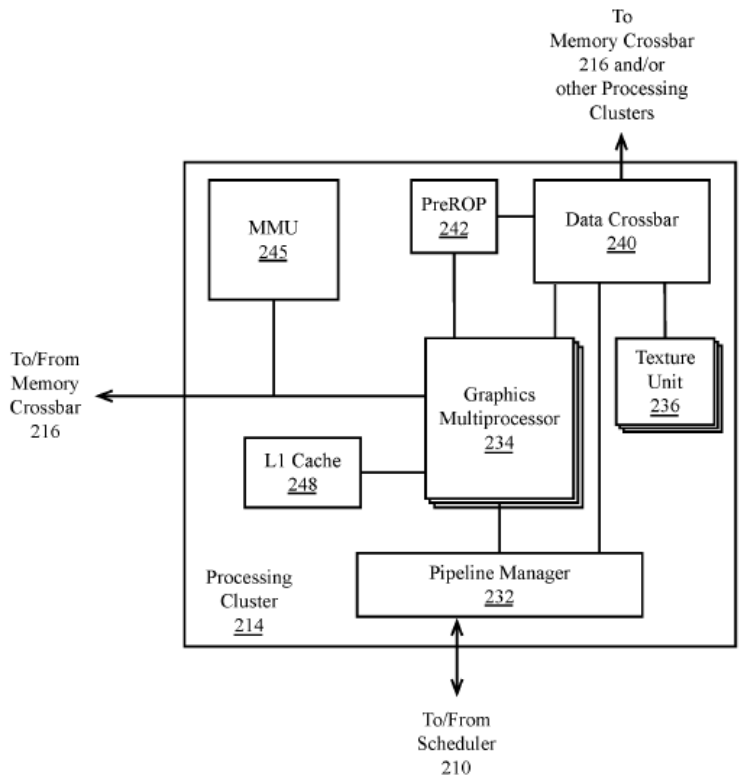
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	external cache (e.g., a Level-3 (L3) cache or Last Level Cache (LLC)) (not shown), which may be shared among processor cores 1407 using known cache coherency techniques. A register file 1406 can be additionally included in processor 1402 and may include different types of registers for storing different types of data (e.g., integer registers, floating point registers, status registers, and an instruction pointer register). Some registers may be general-purpose registers, while other registers may be specific to the design of the processor 1402.") Koker at 40:15-28 ("FIG. 15 is a block diagram of an embodiment of a processor 1500 having one or more processor cores 1502A-1502N, an integrated memory controller 1514, and an integrated graphics processor 1508. Those elements of FIG. 15 having the same reference numbers (or names) as the elements of any other figure herein can operate or function in any manner similar to that described elsewhere herein, but are not limited to such. Processor 1500 can include additional cores up to and including additional core 1502N represented by the dashed lined boxes. Each of processor cores 1502A-1502N includes one or more internal cache units 1504A-1504N. In some embodiments each processor core also has access to one or more shared cached units 1506.") Koker at 40:29-38 ("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.") Koker at 45:59-46:11 ("The graphics processor core 1800 may have greater than or fewer than the illustrated sub-cores 1801A-1801F, up to N modular sub-cores. For each set of N sub-cores, the graphics processor core 1800 can also include shared function logic 1810, shared and/or cache memory 1812, a geometry/fixed function pipeline 1814, as well as additional fixed function logic 1816 to accelerate various graphics and compute processing operations. The shared function logic 1810 can include logic units associated with the shared function logic 1720 of FIG. 17 (e.g., sampler, math, and/or inter-thread communication logic) that can be shared by

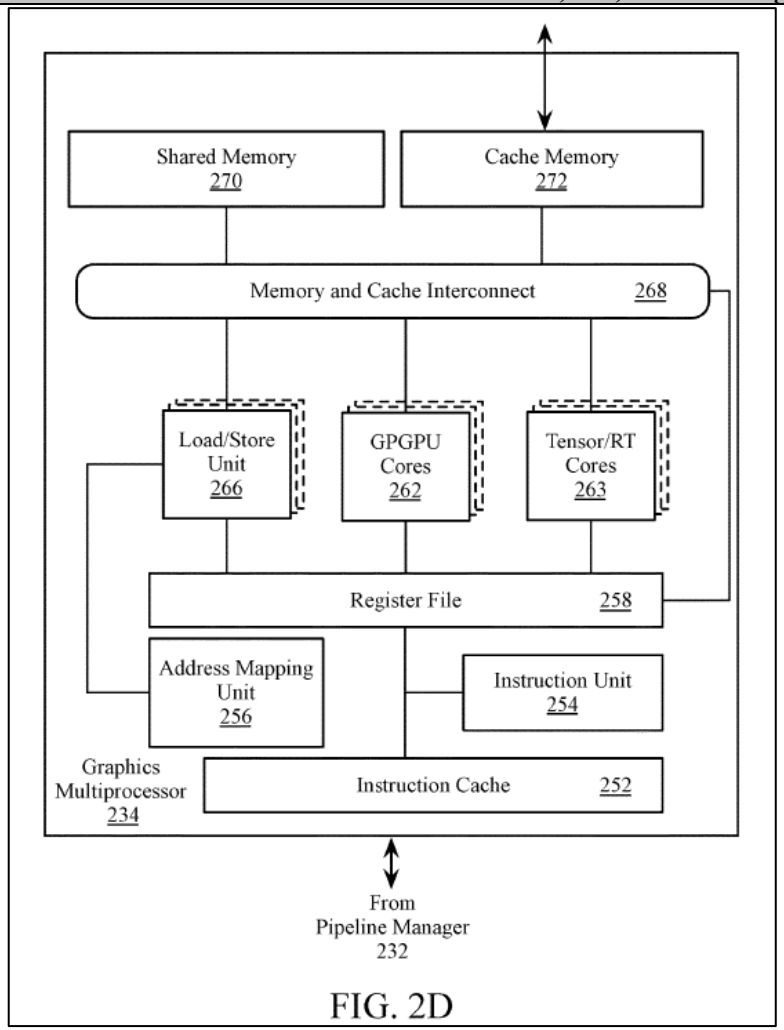
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	each N sub-cores within the graphics processor core 1800. The shared and/or cache memory 1812 can be a last-level cache for the set of N sub-cores 1801A-1801F within the graphics processor core 1800, and can also serve as shared memory that is accessible by multiple sub-cores. The geometry/fixed function pipeline 1814 can be included instead of the geometry/fixed function pipeline 1836 within the fixed function block 1830 and can include the same or similar logic units.") Koker at 52:62-53:2 ("In some embodiments, execution units 2152A-2152B are an array of vector processors having an instruction set for performing graphics and media operations. In some embodiments, execution units 2152A-2152B have an attached L1 cache 2151 that is specific for each array or shared between the arrays. The cache can be configured as a data cache, an instruction cache, or a single cache that is partitioned to contain data and instructions in different partitions.") Koker at 62:63-63:17 ("As shown in FIG. 28B, each chiplet 2830 can include common components and application specific components. Chiplet logic 2836 within the chiplet 2830 can include the specific components of the chiplet, such as an array of streaming multiprocessors, compute units, or execution units described herein. The chiplet logic 2836 can couple with an optional cache or shared local memory 2838 or can include a cache or shared local memory within the chiplet logic 2836. The chiplet 2830 can include a fabric interconnect node 2842 that receives commands via the inter-chiplet fabric. Commands and data received via the fabric interconnect node 2842 can be stored temporarily within an interconnect buffer 2839. Data transmitted to and received from the fabric interconnect node 2842 can be stored in an interconnect cache 2840. Power control 2832 and clock control 2834 logic can also be included within the chiplet. The power control 2832 and clock control 2834 logic can receive configuration commands via the fabric can configure dynamic voltage and frequency scaling for the chiplet 2830. In one embodiment, each chiplet can have an independent clock domain and power domain and can be clock gated and power gated independently of other chiplets.") Koker at 64:14-18 ("The substrate 2910 can include hardware components for I/O 2911, cache memory 2912, and other hardware logic 2913. A fabric 2915 can be embedded in the substrate

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	2910 to enable communication between the various logic chiplets and the logic 2911, 2913 within the substrate 2910.") Koker at 64:19-32 ("In various embodiments a package assembly 2920 can include fewer or greater number of components and chiplets that are interconnected by a fabric 2915 or one or more bridges 2917. The chiplets within the package assembly 2920 may be arranged in a 3D or 2.5D arrangement. In general, bridge structures 2917 may be used to facilitate a point to point interconnect between, for example, logic or I/O chiplets and memory chiplets. The fabric 2915 can be used to interconnect the various logic and/or I/O chiplets (e.g., chiplets 2902, 2904, 2911, 2913). with other logic and/or I/O chiplets. In one embodiment, the cache memory 2912 within the substrate can act as a global cache for the package assembly 2920, part of a distributed global cache, or as a dedicated cache for the fabric 2915.") Koker at 72:23-41 ("FIG. 40 illustrates a modular architecture for interchangeable chiplets, according to an embodiment. In one embodiment a chiplet design 4030 can be made interchangeable by adapting the chiplet logic 4002 for interoperability with an interface template 4008. The interface template 4008 can include standardized logic such as power control 2832 and clock control 2834 logic, and interconnect buffer 2839, interconnect cache 2840, a fabric interconnect node 2842, as in chiplet 2830 of FIG. 28B. The IP designer can then provide the chiplet logic 4002, which is designed to interface with the interface template 4008. The specifics of the chiplet logic 4002 can vary and can include execution units, compute units, or streaming multiprocessors as described herein. The chiplet logic 4002 can also include media encode and/or decode logic, matrix acceleration logic, or ray-tracing logic. For memory chiplets, the chiplet logic 4002 can be replaced with memory cells and the fabric interconnect node can be replaced with interconnect bridge I/O circuits, for example, as illustrated in the memory chiplet 2906 of FIG. 29B.") This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
8[B]: wherein a first level of the first cache memory hierarchy is coherent within the first GPU chiplet; and	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See, e.g.*, Koker at Fig. 2B.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 2B Koker at Fig. 2C.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 2C Koker at Fig. 2D.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 2D Koker at Fig. 3A.

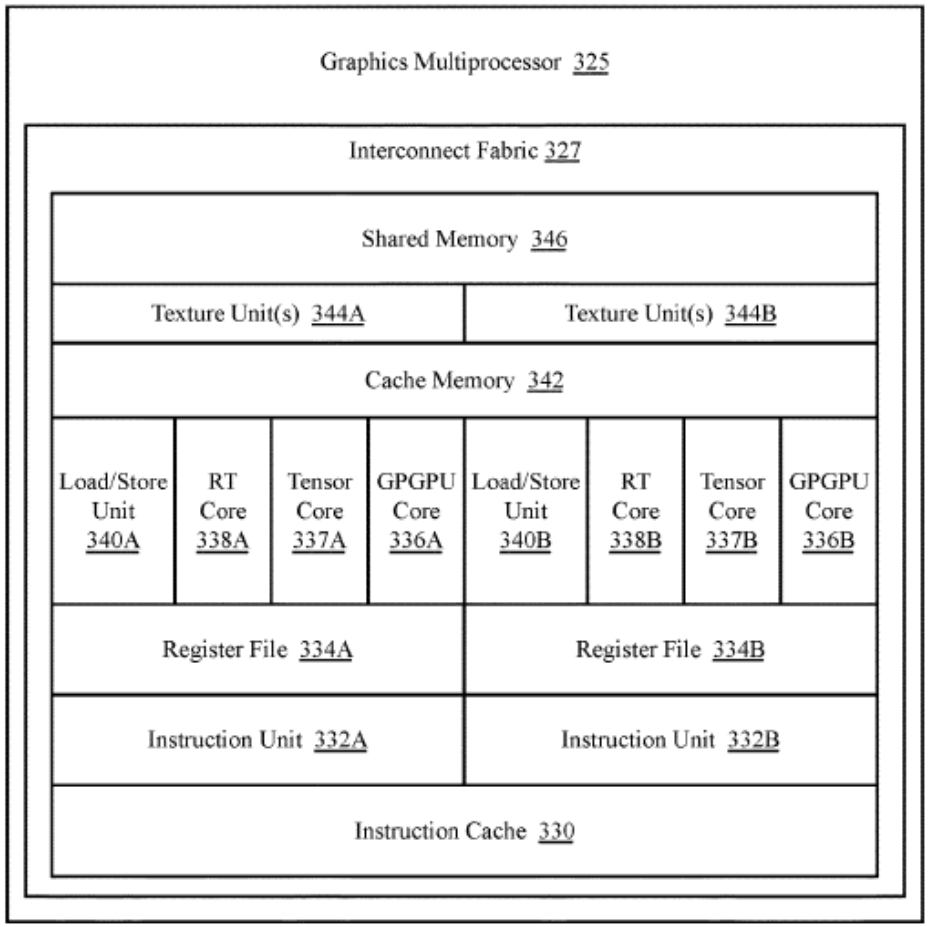


FIG. 3A

Onesta IP, LLC, Exhibit No. 2105
Case No. IPR2026-00011
Page 98 of 115

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at Fig. 3B.

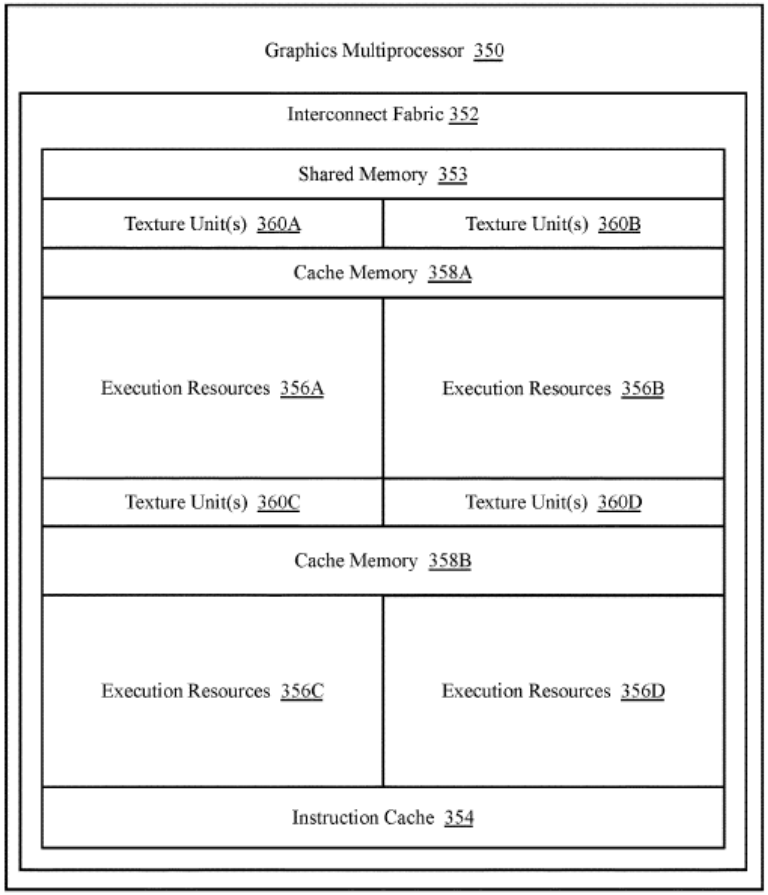
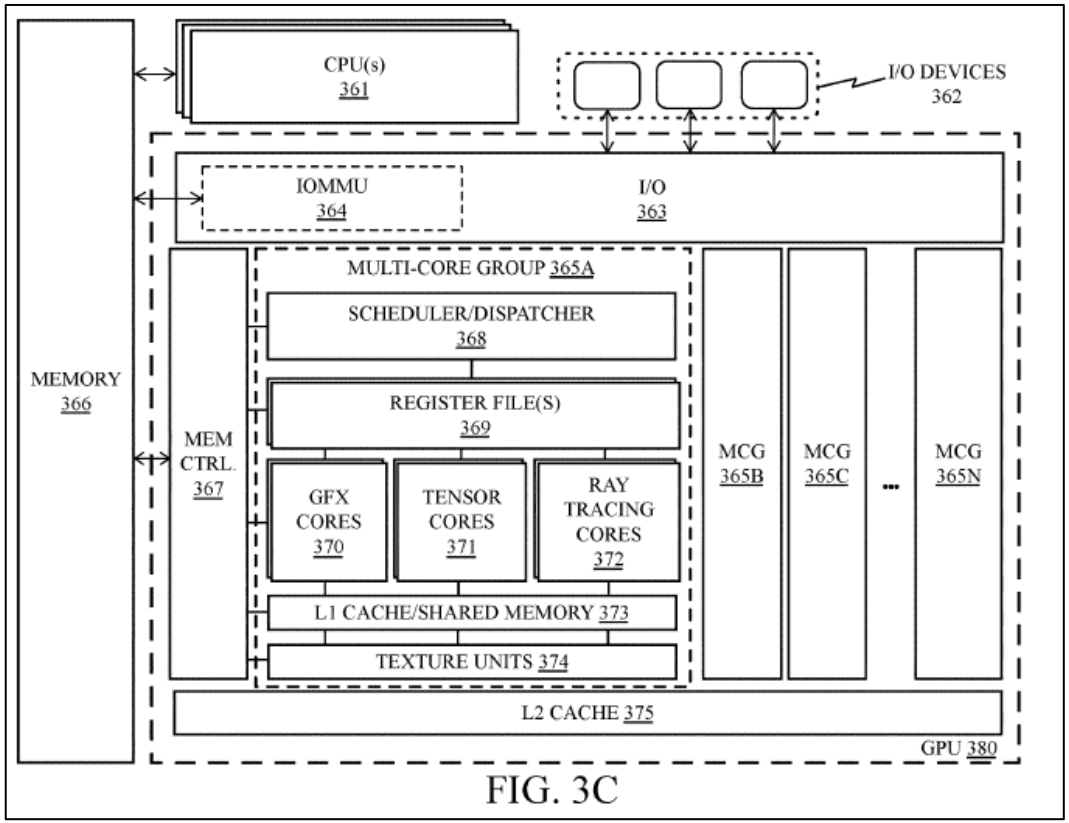
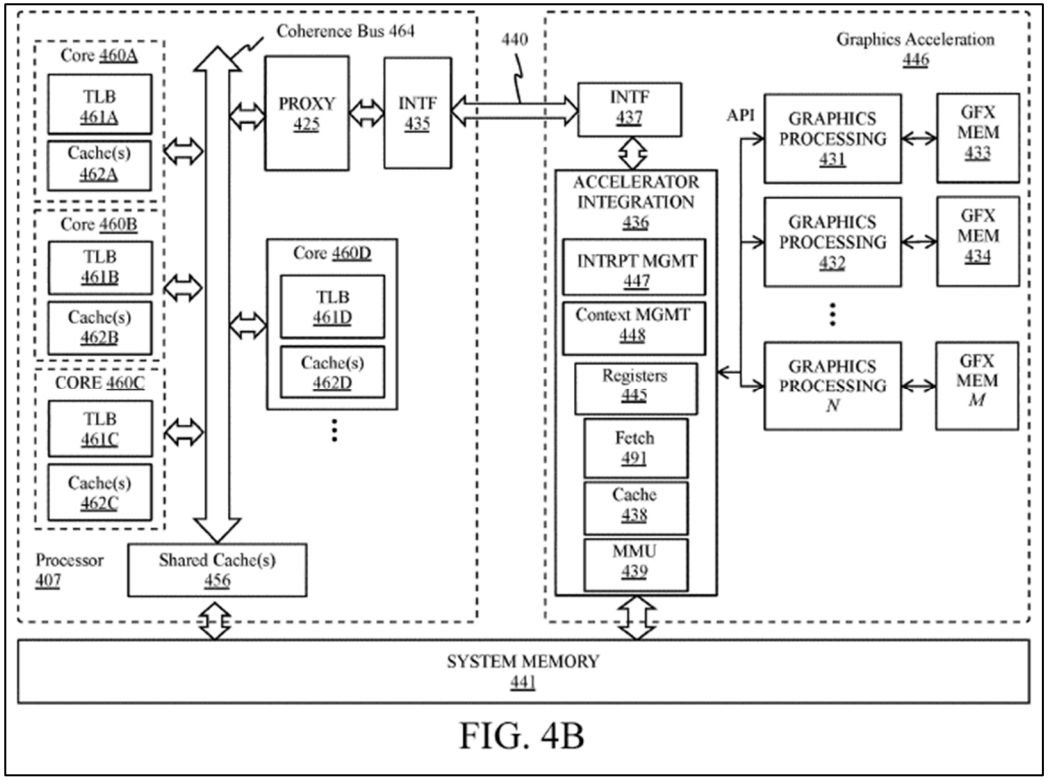
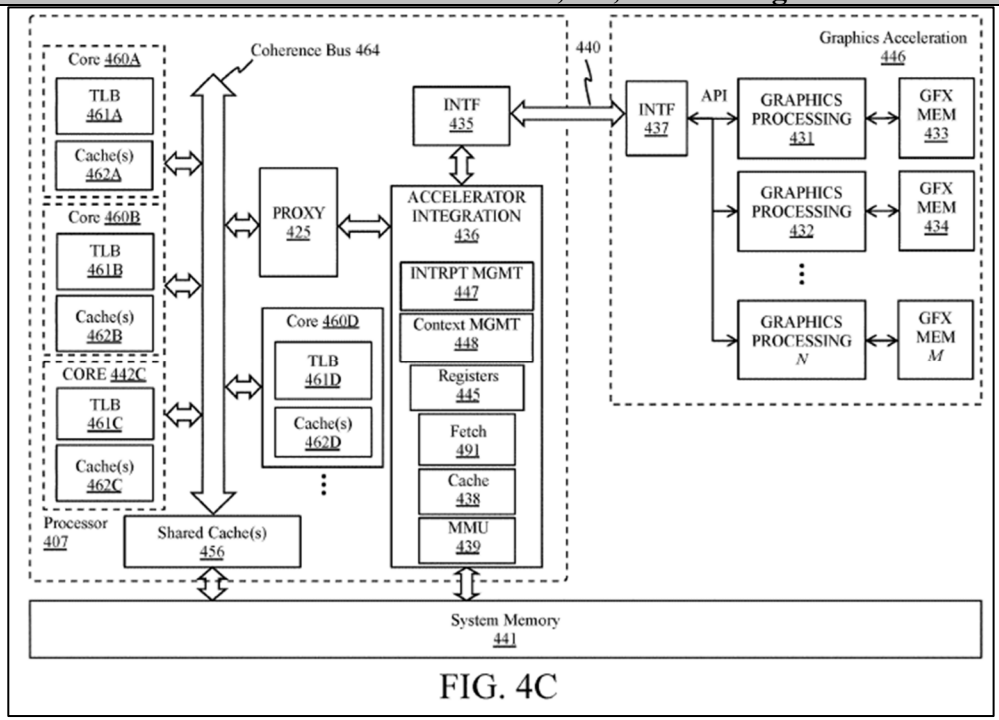
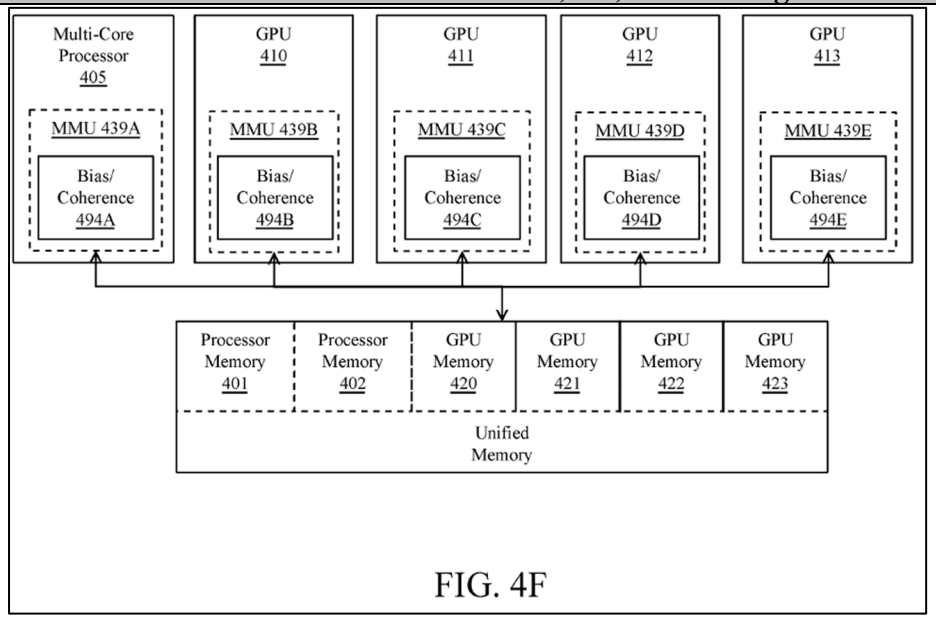


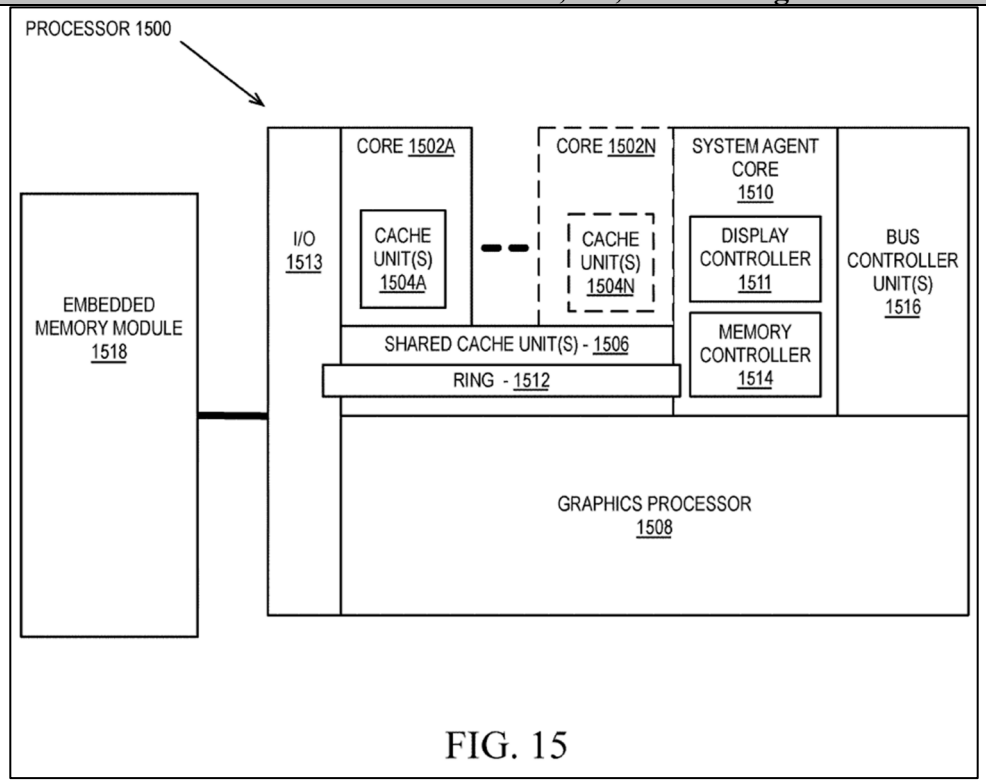
FIG. 3B

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at Fig. 3C.  FIG. 3C Koker at Fig. 4B.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4B Koker at Fig. 4C.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4C Koker at Fig. 4F.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 4F Koker at Fig. 15.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	 FIG. 15 Koker at 12:18-27, ("FIG. 3A-3C illustrate additional graphics multiprocessors, according to embodiments. FIG. 3A-3B illustrate graphics multiprocessors 325, 350, which are variants of the graphics multiprocessor 234 of FIG. 2C. FIG. 3C illustrates a graphics processing unit (GPU) 380 which includes dedicated sets of graphics processing resources arranged into multi-core groups 365A-365N. The illustrated graphics multiprocessors 325, 350 and the multi-core groups 365A-365N can be streaming multiprocessor (SM) capable of simultaneous execution of a large number of execution threads.") Koker at 13:36-42 ("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

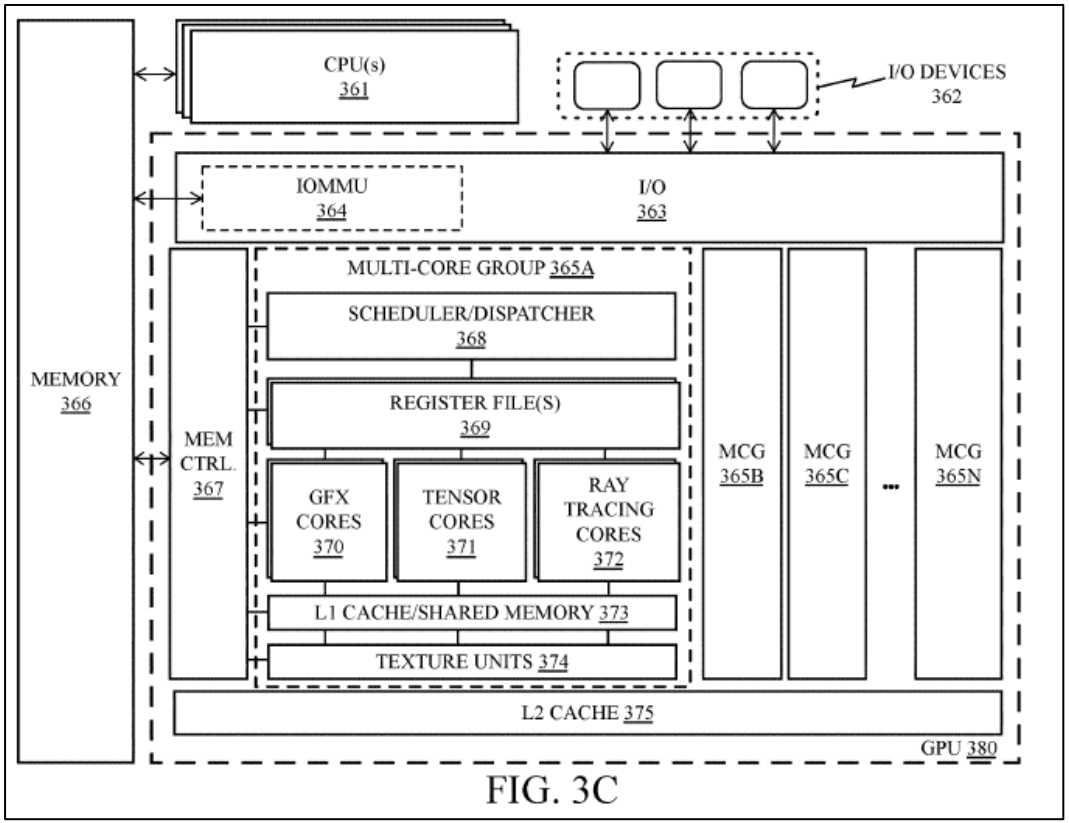
U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	the details of only a single multi-core group 365A are provided, it will be appreciated that the other multi-core groups 365B-365N may be equipped with the same or similar sets of graphics processing resources.") Koker at 13:43-55 ("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.") Koker at 13:56-14:2 ("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).") Koker at 17:47-58 ("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.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	Koker at 17:59-67 ("In one embodiment, a proxy circuit 425 communicatively couples the graphics acceleration module 446 to the coherence bus 464, allowing the graphics acceleration module 446 to participate in the cache coherence protocol as a peer of the cores. In particular, an interface 435 provides connectivity to the proxy circuit 425 over high-speed link 440 (e.g., a PCIe bus, NVLink, etc.) and an interface 437 connects the graphics acceleration module 446 to the high-speed link 440."). Koker at 18:18-36 ("In one embodiment, the accelerator integration circuit 436 includes a memory management unit (MMU) 439 for performing various memory management functions such as virtual-to-physical memory translations (also referred to as effective-to-real memory translations) and memory access protocols for accessing system memory 441. The MMU 439 may also include a translation lookaside buffer (TLB) (not shown) for caching the virtual/effective to physical/real address translations. In one implementation, a cache 438 stores commands and data for efficient access by the graphics processing engines 431-432, N. In one embodiment, the data stored in cache 438 and graphics memories 433-434, M is kept coherent with the core caches 462A-462D, 456 and system memory 411. As mentioned, this may be accomplished via proxy circuit 425 which takes part in the cache coherency mechanism on behalf of cache 438 and memories 433-434, M (e.g., sending updates to the cache 438 related to modifications/accesses of cache lines on processor caches 462A-462D, 456 and receiving updates from the cache 438).") Koker at 23:30-40 ("In one embodiment, bias/coherence management circuitry 494A-494E within one or more of the MMUs 439A-439E ensures cache coherence between the caches of the host processors (e.g., 405) and the GPUs 410-413 and implements biasing techniques indicating the physical memories in which certain types of data should be stored. While multiple instances of bias/coherence management circuitry 494A-494E are illustrated in FIG. 4F, the bias/coherence circuitry may be implemented within the MMU of one or more host processors 405 and/or within the accelerator integration circuit 436.") Koker at 23:41-63 ("One embodiment allows GPU-attached memory 420-423 to be mapped as part of system memory, and accessed using shared virtual memory (SVM) technology, but without suffering the typical performance drawbacks associated with full system cache

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	coherence. The ability to GPU-attached memory 420-423 to be accessed as system memory without onerous cache coherence overhead provides a beneficial operating environment for GPU offload. This arrangement allows the host processor 405 software to setup operands and access computation results, without the overhead of tradition I/O DMA data copies. Such traditional copies involve driver calls, interrupts and memory mapped I/O (MMIO) accesses that are all inefficient relative to simple memory accesses. At the same time, the ability to access GPU attached memory 420-423 without cache coherence overheads can be critical to the execution time of an offloaded computation. In cases with substantial streaming write memory traffic, for example, cache coherence overhead can significantly reduce the effective write bandwidth seen by a GPU 410-413. The efficiency of operand setup, the efficiency of results access, and the efficiency of GPU computation all play a role in determining the effectiveness of GPU offload.") Koker at 24:35-43 ("In one embodiment, cache coherency is maintained by temporarily rendering GPU-biased pages uncacheable by the host processor 405. To access these pages, the processor 405 may request access from the GPU 410 which may or may not grant access right away, depending on the implementation. Thus, to reduce communication between the host processor 405 and GPU 410 it is beneficial to ensure that GPU-biased pages are those which are required by the GPU but not the host processor 405 and vice versa.") Koker at 28:48-59 ("The GPGPU 700 includes a host interface 702 to enable a connection with a host processor. In one embodiment the host interface 702 is a PCI Express interface. However, the host interface can also be a vendor specific communications interface or communications fabric. The GPGPU 700 receives commands from the host processor and uses a global scheduler 704 to distribute execution threads associated with those commands to a set of compute clusters 706A-706H. The compute clusters 706A-706H share a cache memory 708. The cache memory 708 can serve as a higher-level cache for cache memories within the compute clusters 706A-706H.") Koker at 37:60-38:8 ("In some embodiments, the processor 1402 includes cache memory 1404. Depending on the architecture, the processor 1402 can have a single internal cache or multiple levels of internal cache. In some embodiments, the cache memory is shared among various components of the processor 1402. In some embodiments, the processor 1402 also uses an

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	external cache (e.g., a Level-3 (L3) cache or Last Level Cache (LLC)) (not shown), which may be shared among processor cores 1407 using known cache coherency techniques. A register file 1406 can be additionally included in processor 1402 and may include different types of registers for storing different types of data (e.g., integer registers, floating point registers, status registers, and an instruction pointer register). Some registers may be general-purpose registers, while other registers may be specific to the design of the processor 1402.") Koker at 40:15-28 ("FIG. 15 is a block diagram of an embodiment of a processor 1500 having one or more processor cores 1502A-1502N, an integrated memory controller 1514, and an integrated graphics processor 1508. Those elements of FIG. 15 having the same reference numbers (or names) as the elements of any other figure herein can operate or function in any manner similar to that described elsewhere herein, but are not limited to such. Processor 1500 can include additional cores up to and including additional core 1502N represented by the dashed lined boxes. Each of processor cores 1502A-1502N includes one or more internal cache units 1504A-1504N. In some embodiments each processor core also has access to one or more shared cached units 1506.") Koker at 40:29-38 ("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.") Koker at 52:62-53:2 ("In some embodiments, execution units 2152A-2152B are an array of vector processors having an instruction set for performing graphics and media operations. In some embodiments, execution units 2152A-2152B have an attached L1 cache 2151 that is specific for each array or shared between the arrays. The cache can be configured as a data cache, an instruction cache, or a single cache that is partitioned to contain data and instructions in different partitions.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	See evidence cited for '803 patent limitation 8[A]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
8[C]: a second cache memory hierarchy at the second GPU chiplet,	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See* evidence cited for '803 patent limitation 8[A]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
8[D]: wherein a first level of the second cache memory hierarchy is coherent within the second GPU chiplet.	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See* evidence cited for '803 patent limitation 8[B]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
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	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See, e.g.*,

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
level of the second cache memory hierarchy,	Koker at Fig. 3C.  FIG. 3C Koker at Fig. 15.

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.

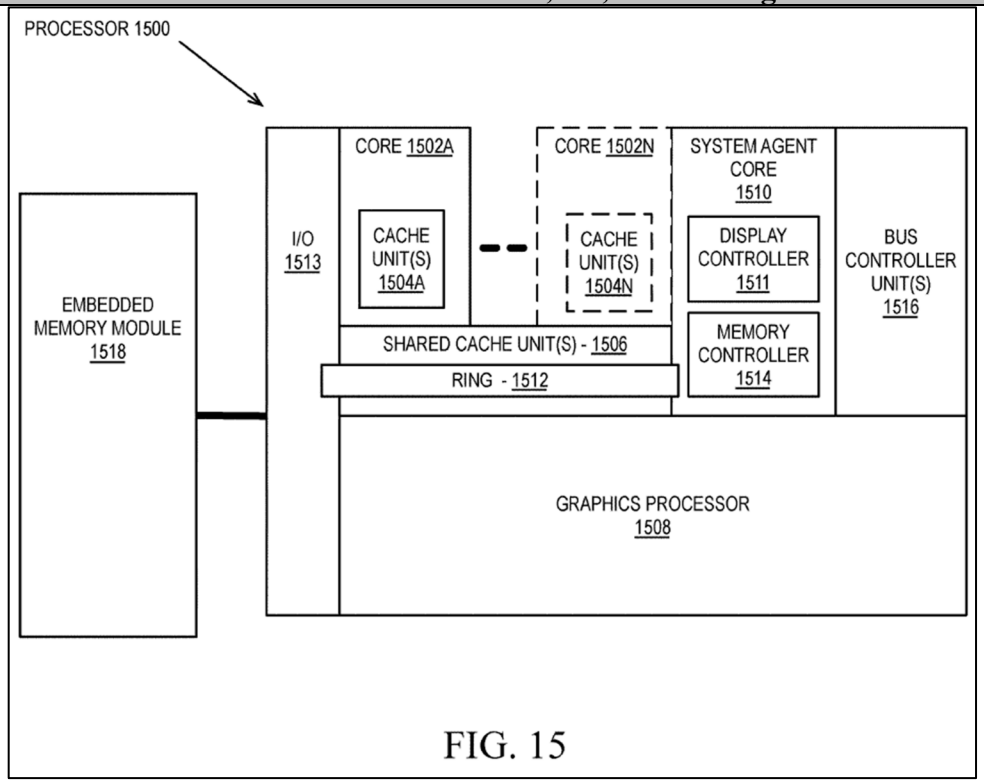


FIG. 15

Koker at 12:18-27, ("FIG. 3A-3C illustrate additional graphics multiprocessors, according to embodiments. FIG. 3A-3B illustrate graphics multiprocessors 325, 350, which are variants of the graphics multiprocessor 234 of FIG. 2C. FIG. 3C illustrates a graphics processing unit (GPU) 380 which includes dedicated sets of graphics processing resources arranged into multi-core groups 365A-365N. The illustrated graphics multiprocessors 325, 350 and the multi-core groups 365A-365N can be streaming multiprocessor (SM) capable of simultaneous execution of a large number of execution threads.")

Koker at 17:59-67 ("In one embodiment, a proxy circuit 425 communicatively couples the graphics acceleration module 446 to the coherence bus 464, allowing the graphics acceleration

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	module 446 to participate in the cache coherence protocol as a peer of the cores. In particular, an interface 435 provides connectivity to the proxy circuit 425 over high-speed link 440 (e.g., a PCIe bus, NVLink, etc.) and an interface 437 connects the graphics acceleration module 446 to the high-speed link 440."). Koker at 37:60-38:8 ("In some embodiments, the processor 1402 includes cache memory 1404. Depending on the architecture, the processor 1402 can have a single internal cache or multiple levels of internal cache. In some embodiments, the cache memory is shared among various components of the processor 1402. In some embodiments, the processor 1402 also uses an external cache (e.g., a Level-3 (L3) cache or Last Level Cache (LLC)) (not shown), which may be shared among processor cores 1407 using known cache coherency techniques. A register file 1406 can be additionally included in processor 1402 and may include different types of registers for storing different types of data (e.g., integer registers, floating point registers, status registers, and an instruction pointer register). Some registers may be general-purpose registers, while other registers may be specific to the design of the processor 1402.") Koker at 40:15-28 ("FIG. 15 is a block diagram of an embodiment of a processor 1500 having one or more processor cores 1502A-1502N, an integrated memory controller 1514, and an integrated graphics processor 1508. Those elements of FIG. 15 having the same reference numbers (or names) as the elements of any other figure herein can operate or function in any manner similar to that described elsewhere herein, but are not limited to such. Processor 1500 can include additional cores up to and including additional core 1502N represented by the dashed lined boxes. Each of processor cores 1502A-1502N includes one or more internal cache units 1504A-1504N. In some embodiments each processor core also has access to one or more shared cached units 1506.") Koker at 40:29-38 ("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

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	the LLC. In some embodiments, cache coherency logic maintains coherency between the various cache units 1506 and 1504A-1504N.") Koker at 41:11-19 ("The exemplary I/O link 1513 represents at least one of multiple varieties of I/O interconnects, including an on package I/O interconnect which facilitates communication between various processor components and a high-performance embedded memory module 1518, such as an eDRAM module. In some embodiments, each of the processor cores 1502A-1502N and graphics processor 1508 can use embedded memory modules 1518 as a shared Last Level Cache.") Koker at 45:59-46:11 ("The graphics processor core 1800 may have greater than or fewer than the illustrated sub-cores 1801A-1801F, up to N modular sub-cores. For each set of N sub-cores, the graphics processor core 1800 can also include shared function logic 1810, shared and/or cache memory 1812, a geometry/fixed function pipeline 1814, as well as additional fixed function logic 1816 to accelerate various graphics and compute processing operations. The shared function logic 1810 can include logic units associated with the shared function logic 1720 of FIG. 17 (e.g., sampler, math, and/or inter-thread communication logic) that can be shared by each N sub-cores within the graphics processor core 1800. The shared and/or cache memory 1812 can be a last-level cache for the set of N sub-cores 1801A-1801F within the graphics processor core 1800, and can also serve as shared memory that is accessible by multiple sub-cores. The geometry/fixed function pipeline 1814 can be included instead of the geometry/fixed function pipeline 1836 within the fixed function block 1830 and can include the same or similar logic units.") Koker at 52:62-53:2 ("In some embodiments, execution units 2152A-2152B are an array of vector processors having an instruction set for performing graphics and media operations. In some embodiments, execution units 2152A-2152B have an attached L1 cache 2151 that is specific for each array or shared between the arrays. The cache can be configured as a data cache, an instruction cache, or a single cache that is partitioned to contain data and instructions in different partitions.") Koker at 62:63-63:17 ("As shown in FIG. 28B, each chiplet 2830 can include common components and application specific components. Chiplet logic 2836 within the chiplet 2830 can

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	include the specific components of the chiplet, such as an array of streaming multiprocessors, compute units, or execution units described herein. The chiplet logic 2836 can couple with an optional cache or shared local memory 2838 or can include a cache or shared local memory within the chiplet logic 2836. The chiplet 2830 can include a fabric interconnect node 2842 that receives commands via the inter-chiplet fabric. Commands and data received via the fabric interconnect node 2842 can be stored temporarily within an interconnect buffer 2839. Data transmitted to and received from the fabric interconnect node 2842 can be stored in an interconnect cache 2840. Power control 2832 and clock control 2834 logic can also be included within the chiplet. The power control 2832 and clock control 2834 logic can receive configuration commands via the fabric can configure dynamic voltage and frequency scaling for the chiplet 2830. In one embodiment, each chiplet can have an independent clock domain and power domain and can be clock gated and power gated independently of other chiplets.") Koker at 64:19-32 ("In various embodiments a package assembly 2920 can include fewer or greater number of components and chiplets that are interconnected by a fabric 2915 or one or more bridges 2917. The chiplets within the package assembly 2920 may be arranged in a 3D or 2.5D arrangement. In general, bridge structures 2917 may be used to facilitate a point to point interconnect between, for example, logic or I/O chiplets and memory chiplets. The fabric 2915 can be used to interconnect the various logic and/or I/O chiplets (e.g., chiplets 2902, 2904, 2911, 2913). with other logic and/or I/O chiplets. In one embodiment, the cache memory 2912 within the substrate can act as a global cache for the package assembly 2920, part of a distributed global cache, or as a dedicated cache for the fabric 2915.") Koker at 70:58-71:2 ("Coherency is maintained during data transmission between fabric interconnect nodes. In one embodiment, data for an originating thread on a GPGPU or parallel processor is routed within the same traffic class and the traffic classes are assigned to a specific virtual channel. Data within a single traffic class on a single virtual channel is transmitted in FIFO order. Thus, data from a single thread is strictly ordered when transmitted via the fabric and per-thread coherency is maintained to avoid read-after-write or write-after-read data hazards. In one embodiment, thread group coherency is maintained via a global synchronization mechanism between resource nodes.")

U.S. Patent No. 11,841,803	U.S. Patent No. 10,909,652 to Altug Koker et al.
	See evidence cited for '803 patent claim 8. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
9[B]: wherein the unified cache memory is coherent across all GPU chiplets of the GPU chiplet array.	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this limitation. *See* evidence cited for '803 patent claim 8. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.
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.	Including at least under Complainant's apparent infringement theory, Koker discloses and/or renders obvious this claim. *See* evidence cited for '803 patent claim 1[C]. This limitation also would have been obvious to a person of ordinary skill in the art based on (1) Koker alone; (2) Koker in combination with the general knowledge of a person of ordinary skill in the art; and/or (3) Koker in combination with the teachings with respect to this claim element as detailed in NVIDIA's other invalidity charts and cover pleading.