

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

REALTEK SEMICONDUCTOR CORP.,
and
TCL INDUSTRIES HOLDINGS CO., INC.

Petitioner,
v.

ATI TECHNOLOGIES ULC
Patent Owner.

Case No. IPR2023-00922

U.S. Patent No. 8,760,454¹

**PETITIONER’S REPLY TO PATENT OWNER’S
RESPONSE TO PETITION**

¹ Joinder with IPR2024-00366

TABLE OF CONTENTS

Page

I.	INTRODUCTION	1
II.	CLAIM CONSTRUCTION	1
A.	“Unified shader”	1
B.	“Operative to”	1
C.	“Various Degrees of Completion”	2
D.	ATI’s Improper “Consideration” And “Determination” Limitations ...	3
III.	GROUND 1: THE LINDHOLM REFERENCES RENDER CLAIMS 1-11 OBVIOUS.....	6
A.	ATI Fails to Show Conception of the “Complete and Operative Invention” Under Federal Circuit Precedent.....	6
B.	Dr. Mangione-Smith Offers No Opinion As to Conception of Many Limitations That ATI Argues Are Required by the Challenged Claims	9
C.	Claims 1, 3, 4, 5-10: “Considering” Available Space and “Determining”	11
D.	Claim 11: Performing Operations “At Various Degrees of Completion Based on Switching Between Instructions”	12
E.	“In Response To” (Claim 2).....	13
F.	Claim 6: Memory Separate From the “Store”	14
G.	“Control Signal” (Claims 7 and 10) and “Arbiter” (Claim 10).....	15
IV.	GROUND 2: AMANATIDES AND KOHN RENDER CLAIMS 1-11 OBVIOUS.....	15
A.	Executing Operations Based on Available Space (Claims 5-10) And Performing Operations “Until” Storage is Available (Claims 1, 3, 4).....	15
B.	Claim 11: Performing Operations “At Various Degrees of Completion Based on Switching Between Instructions”	16
C.	Claims 2 and 5: “Selected Data”	18
D.	Claims 7 and 10: “Selection Circuit” Is Within the Alleged “Unified Shader”	18
E.	Claims 7 and 10: “Control Signal”	19
F.	“Selection Circuit” (Claims 7 and 10) and “Arbiter” (Claim 10)	19

TABLE OF CONTENTS

Page

V.	GROUND 3: SELZER AND FISKE RENDER CLAIMS 1-11 OBVIOUS.....	21
A.	Executing Operations Based on Available Space (Claims 5-10) And Performing Operations “Until” Storage is Available (Claims 1, 3, 4).....	21
B.	Claim 11: Performing Operations “At Various Degrees of Completion Based on Switching Between Instructions”	24
C.	Claim 2: “In Response To”	24
D.	Claims 2, 5-7, 10, 11: “Sequencer,” “Instruction Store,” “Circuitry,” “Selection Circuit,” and “Arbiter”	24
E.	Claims 2, 5, 11: “Sequencer” and “Instruction Store” Maintain Instructions	25
F.	“Control Signal” (claims 7 and 10) and “Arbiter” (claim 10)	25
G.	Motivation to Combine Selzer and Fiske.....	26
VI.	SECONDARY CONSIDERATIONS	27

TABLE OF AUTHORITIES

Page(s)

Cases

<i>ATI Tech. ULC v. Iancu</i> , 920 F.3d 1362 (Fed. Cir. 2019)	7, 9
<i>Brand v. Miller</i> , 487 F.3d 862 (Fed. Cir. 2007)	6
<i>Burroughs Wellcome Co. v. Barr Labs., Inc.</i> , 40 F.3d 1223 (Fed. Cir. 1994)	6
<i>Dawson v. Dawson</i> , 710 F.3d 1347 (Fed. Cir. 2013)	6
<i>In re GPAC, Inc.</i> , 57 F.3d 1573 (Fed. Cir. 1995)	29
<i>In re Imes</i> , 778 F.3d 1250 (Fed. Cir. 2015)	1
<i>Info-Hold, Inc. v. Applied Media Tech. Corp.</i> , 783 F.3d 1262 (Fed. Cir. 2015)	1
<i>Milwaukee Elec. Tool Corp. v. Snap-On Inc.</i> , 271 F. Supp. 3d 990 (E.D. Wis. 2017)	7
<i>Ormco Corp. v. Align Tech., Inc.</i> , 463 F.3d 1299 (Fed. Cir. 2006)	29
<i>Pfaff v. Wells Elecs, Inc.</i> , 525 U.S. 55 (1998).....	7
<i>Textron Innovations, Inc. v. American Eurocopter Corp.</i> , 498 Fed. Appx. 23 (Fed. Cir. 2012).....	18

PETITIONER'S EXHIBIT LIST

Exhibit #	Reference Name
1001	U.S. Patent 8,760,454 to Morein et al.
1002	Prosecution History of U.S. Patent 8,760,454
1003	Declaration of Hanspeter Pfister, Ph.D.
1004	Curriculum Vitae of Hanspeter Pfister, Ph.D.
1005	Lindholm 685
1006	Lindholm 913
1007	Amanatides
1008	Kohn
1009	Selzer [Library of Congress]
1010	Fiske
1011	IEEE – Fiske
1012	Declaration of Gordon MacPherson re Authentication of Fiske (May 11, 2023)
1013	IEEE – Kohn
1014	Declaration of Gordon MacPherson re Authentication of Kohn (May 11, 2023)
1015	Initial Determination on Violation of Section 337, Certain Consumer Electronics and Display Devices With Graphics Processing and Graphics Processing Units Therein, 337-TA-932 (October 9, 2015)
1016	Selzer [University of California, Berkeley, Library]
1017	Redacted Deposition Transcript of Andrew E. Gruber, taken March 17, 2015, in ITC Case No. 337-TA-932
1018	Declaration of Hanspeter Pfister, Ph.D.
1019	Deposition Transcript of William Mangione-Smith (5/10/2024)
1020	https://en.wikipedia.org/wiki/DirectX
1021	https://en.wikipedia.org/wiki/Unified_shader_model
1022	https://en.wikipedia.org/wiki/GeForce_8_series
1023	https://www.hardwarezone.com.sg/feature-nvidia-geforce-8800-gtx-gts-g80-worlds-first-dx10-gpu/embracing-unified-shader-architecture
1024	https://en.wikipedia.org/wiki/Radeon_HD_2000_series

I. INTRODUCTION

Claims 1-11 of U.S. Patent No. 8,760,454 are unpatentable.

II. CLAIM CONSTRUCTION

Unless expressly addressed herein, claim terms should be given their plain and ordinary meaning.

A. “Unified shader”

Petitioner agrees with the Board that “unified shader” means a shader that is configured to perform both vertex and pixel operations.

B. “Operative to”

Claims 2-7 recite the term “operative to” in connection with instructions (claims 2, 5), processing unit and shared resources (claims 3, 4), circuitry (claim 6), and selection circuit (claim 7). “Operative to” or “operable to” is typically construed to mean “capable of.” *In re Imes*, 778 F.3d 1250, 1253-54 (Fed. Cir. 2015); *Info-Hold, Inc. v. Applied Media Tech. Corp.*, 783 F.3d 1262, 1265, 1268 (Fed. Cir. 2015). The ’454 specification likewise describes the claimed unified shader and its components as being “capable of” and “able to.” EX1001, 2:58-61 (“the present invention is directed to a unified shader that is **capable of** ...”), 6:29-33 (same), 6:36-41 (unified shader “**capable of**”), 5:21-36 (unified shader “of the present invention is **able to** ...”). Accordingly, “operative to” means “capable of.”

C. “Various Degrees of Completion”

Claim 11 recites that the claimed processor unit performs vertex and pixel operations “at various degrees of completion based on switching between instructions in the instruction store.” ATI *adds* the requirement of switching between “*unfinished* vertex operations and pixel operations” without support or analysis. POR 32-33. ATI cites Dr. Mangione-Smith’s declaration ¶234, but the declaration only repeats the claim language. ATI also cites Dr. Pfister’s deposition testimony, but this was in response to a hypothetical in which there are only two operations. *Id.* Dr. Pfister reasoned that to satisfy the claim requirement of multiple switches, with only two operations, the first operation must be “unfinished.” EX1018 ¶22. But claim 11 requires switching between “vertex manipulation operations [plural] and pixel manipulation operations [plural] at various degrees of completion,” and is not limited to—nor covers—switching between a total of two operations.

The plain and ordinary meaning of “various degrees of completion” would include operations that are about to start, in process (“unfinished”), or completed. EX1018, ¶¶21-22. Nothing in the ’454 patent manifests a clear disavowal of the full scope of this plain and ordinary meaning. The term requires no construction and is not limited to “unfinished” operations.

D. ATI's Improper "Consideration" And "Determination" Limitations

ATI and its expert argue that claims 1, 3, 4, and 5-10 "require consideration of the amount of available space." *See* POR 29, 32, 40, 54; EX2121, ¶¶308-313, EX1019, 85:11-86:17. ATI further argues that the unified shader must "determine which data type to execute" based on such "consideration." *Id.* ATI is wrong.²

First, the claims do not require any such "consideration" or "determination" actions. EX1018, ¶¶3-9. Dr. Mangione-Smith could not point to any term that should be construed to require "consideration" or "determination" steps. EX1019, 85:11-86:17. Claim 1 requires only that vertex and pixel operations are performed "unless the general purpose register block does not have enough available space..." As Dr. Mangione-Smith agreed, everything in claim 1 after "unless" is conditional and not required when determining invalidity. EX1019, 90:3-10. *See also*, [Ex Parte Schulhauser](#), 2016 WL 6277792 (PTAB Apr. 28, 2016) (precedential). Regardless, nothing after the "unless" recites anything more than continuing pixel operations "until" more space becomes available.

Claims 3, 4 and 5-10 also have no "consideration" or "determination" requirements. In claim 3, the processor need only perform pixel operations until enough shared resources become available to perform vertex operations. Claim 4

² ATI does not argue these additional limitations for claim 11.

is the inverse. Claim 5 requires the execution of pixel and vertex calculation operations on “selected data maintained in a store depending upon an amount of space available in the store.” No “evaluation” or “consideration” of storage space is required, and moreover, the selection of “selected data ... depending on an amount of space available” occurs *before* the sequencer “causes” the processor unit to execute operations.

Moreover, the ’454 patent teaches that any “consideration” and “determination” steps *are not carried out by the claimed unified shader*. The claims are directed solely to a *unified shader*. EX1019, 25:17-19, EX1018 ¶4. Figure 5 is a “block diagram of the unified shader”:

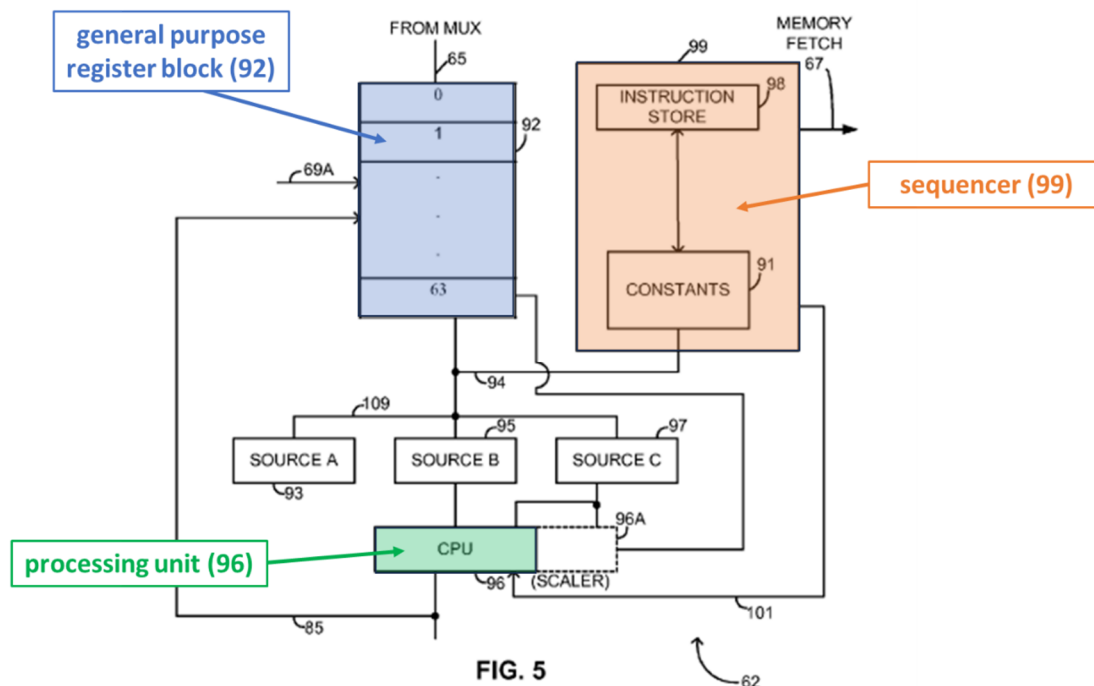


Figure 4A shows where the unified shader fits into the “graphics processor architecture”:

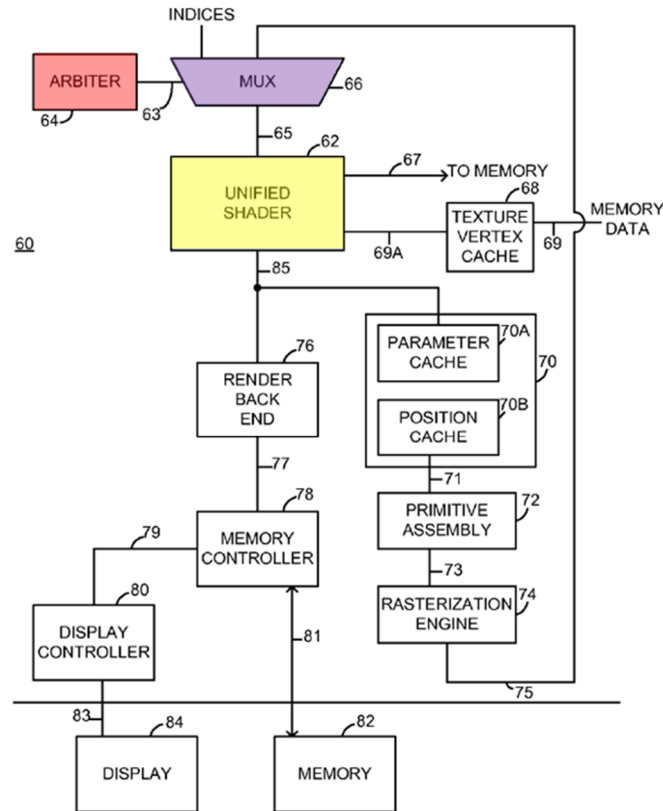


FIG. 4A

ATI and its expert agree that it is the *arbiter* that “considers” the amount of available space and then “determines” what type of vertex or pixel data to place into the general purpose register block based on its own “arbitration scheme.” EX1001, 4:18-28, 5:32-52; EX1018, ¶4; POR 7-8; EX1019, 76:6-16. The unified shader simply operates “based on information passed from the multiplexer” via the arbiter. *Id.*, 5:23-27; EX1018, ¶4. It would be improper to import an arbiter limitation into claims 1, 3, 4, and 5, which are directed solely to the unified shader.

III. GROUND 1: THE LINDHOLM REFERENCES RENDER CLAIMS 1-11 OBVIOUS

A. ATI Fails to Show Conception of the “Complete and Operative Invention” Under Federal Circuit Precedent

The Board acknowledged that the dynamic memory aspects of the R400 design may not have worked properly or been embodied into an operable product until the R600 (if at all). ID 33. Nevertheless, the Board concluded that even if “the dynamic memory aspects of the R400 design did not work properly,” the evidence showed “an idea that was definite and permanent enough that one skilled in the art could understand the invention” during the early stages of R400 development. *Id.*

But conception also requires “a complete and operative invention, as it is hereafter to be applied in practice.” *Burroughs Wellcome Co. v. Barr Labs., Inc.*, 40 F.3d 1223, 1228 (Fed. Cir. 1994). Conception is “complete and operative” when “only ordinary skill would be necessary to reduce the invention to practice, ***without extensive research or experimentation***,” and “courts require corroborating evidence of a contemporaneous disclosure that would enable one skilled in the art to make the invention.” *Id.* (emphasis added). Uncertainty revealed by “the subsequent course of experimentation, especially experimental failures,” may demonstrate that conception is not “complete and operative.” *Id.* at 1229. *Accord*, *Dawson v. Dawson*, 710 F.3d 1347, 1356 (Fed. Cir. 2013); *Brand v. Miller*, 487 F.3d 862 (Fed. Cir. 2007). A “complete” conception is shown “by instances making clear that it did

so work,” whereas failed attempts to reduce the invention to practice show a lack of a complete conception. *Pfaff v. Wells Elecs, Inc.*, 525 U.S. 55, 66, n. 12 (1998). *See, e.g., The Regents of the University of California v. The Broad Institute, Inc.*, Interference No. 106,115, Decision on Priority (PTAB Feb. 28, 2022) (experimental failures and a five-month period of research and experimentation showed the inventor did not yet possess a complete invention); *Milwaukee Elec. Tool Corp. v. Snap-On Inc.*, 271 F. Supp. 3d 990, 1019 (E.D. Wis. 2017) (experimental failures showed a lack of a “complete and operative” conception).

Petitioner cited evidence from the co-inventor, Mr. Gruber, that the “dynamic register file allocation” was never tested or found to work during the development of the R400, did not function properly when reduced to silicon for the later Xenos or R600 chip in 2005/2006, and in fact, may have never worked. Paper 8, 6-8; EX1017, 25, 103-107, 115-116, 167. Even in the R600, Mr. Gruber testified that the division of the register file may have only been a *fixed* value for vertices and pixels. *Id.* Moreover, no chip was ever built based on the R400 or R500, and there were “substantial changes” between the R500 and R600, including “substantial changes” to the sequencer. EX1017 7-9. The Xenos chip did not ship until 2005 and the R600 in 2007.

Patent Owner bears the burden of proof to show prior conception and diligence. *ATI Tech. ULC v. Iancu*, 920 F.3d 1362, 1369 (Fed. Cir. 2019). But ATI

has done nothing to prove that any conception prior to June 2003 was a “complete and operative invention” that could be reduced to practice *without extensive research or experimentation*. ATI never addresses the above testimony of Mr. Gruber and never cites any evidence that the “dynamic” memory allocation or priority ever actually worked. POR 17-29. Rather, ATI argues that “over one hundred project managers/designers diligently worked to implement and test the R400” from at least 2001 until 2003. *Id.* 25, EX2001 ¶196. And yet, as Mr. Gruber testified, the “dynamic register file allocation” never worked. The R400 Sequencer Specification itself identifies as an “Open Issue” the “[n]eed to do some testing ... on the register file allocation method (dynamic VS static)” in Versions 1.6 thru 2.11, the last of which bears an “Originate Date” of July 9, 2003, the last version of the sequencer specification. EX2119, 375-1318. This corroborates that *the issue was never resolved in the R400*.

When more than 100 engineers work on a development project from 2001 to 2005 or later, and still can’t get a claimed feature to work, this demonstrates an inability of a POSITA to reduce a claimed invention to practice “without extensive research or experimentation”—even if the issue is fixed subsequently. Dr. Mangione-Smith offers no opinion regarding actual reduction to practice, much less that the dynamic register file allocation or prioritization of vertices/pixels based on available memory or workload *actually worked*—at *any* time. EX2001 ¶¶ 71-302.

The attached Declarations from Wolfe, Watson, and Lefebvre (EX2002, 2003, 2004) were directed to the '053 patent and do not address the feature. And in *ATI*, the Federal Circuit

In *ATI*, the Federal Circuit only held that there was *diligence* followed by constructive reduction to practice. *ATI*, 920 F.3d at 1366, 1369. The Federal Circuit did not address actual reduction to practice, nor did it address conception, which was not disputed on appeal. Thus, it also never reached the question of whether ATI had a “complete and operative” conception of the dynamic memory/priority limitations claimed in the '454 Patent.

Accordingly, ATI has failed to show a complete and operative conception of the '454 Patent claims prior to June 2003.

B. Dr. Mangione-Smith Offers No Opinion As to Conception of Many Limitations That ATI Argues Are Required by the Challenged Claims

As discussed throughout this Reply, ATI attempts to add limitations to avoid the prior art. *See, e.g.*, Sections II.C-D., III.E-F. However, ATI cites no evidence—and its expert does not opine—that these requirements were conceived prior to June 2003:

- Claims 1, 3, 4, 5: ATI’s added “consideration” and “determining” requirements, *supra* II.D.. *But cf.* EX2001 ¶85 (no identification of any component or process of the “unified shader” during conception that

“considered” the amount of available space and then “determined” the type of data to execute, only that the unified shader performs operations until space becomes available).

- Claim 2: ATI adds a limitation that “receiving selected data from a data storage” is what “causes execution of instructions,” as opposed to instructions or a control signal. POR 33. *But cf.* EX2001 ¶¶112-117.
- Claims 2 and 5: ATI argues that “selected data” requires the unified shader to be “selecting data from a data store” and then “transmitting that data to the processor.” POR 44. *But cf.* EX2001 ¶¶105-117, 148-154.
- Claims 7 and 10: For conception, Dr. Mangione-Smith opines the “control signal” is just control packets with state pointer information. EX2001 ¶¶161-162, 176-177. But ATI contends the claims require “electrical signals used to control internal or external devices or processes” and that “packets” are not control signals. POR 35, 50.
- Claim 11: ATI argues that the unified shader must be able to switch back and forth between “unfinished” vertex/pixel operations. POR 10, 32-33. *But cf.* EX2001 ¶¶189-194 (not applying requirement for conception).

If the Board finds any of these added limitations are required, then ATI has failed to show prior conception for this additional reason.

C. Claims 1, 3, 4, 5-10: “Considering” Available Space and “Determining”

As discussed *supra* Section II.D., ATI improperly imports limitations into claims 1, 3, 4, and 5-10: “consideration” or “evaluation” of available space and “determination” of what to execute based on that “evaluation.” POR 29-32. The claims do not require such limitations. Nevertheless, the Lindholm references would render these limitations obvious.

Lindholm discloses prioritization of vertex or pixel operations depending on available storage space. Pet. 30-33. Lindholm discloses selection based on the availability of source data, which includes the stored “incoming” vertex data, which renders obvious that an evaluation must take place in order to select what source data to store/maintain.³ Further, Lindholm discloses using a Resource Scoreboard that tracks the status of source registers (“storage space”) scheduled to be written during processing (*i.e.*, destination registers). EX1005, 8:58-65. The Resource Scoreboard uses the location of each program instruction’s required source data to track whether that data is available, as well as computational resources. *Id.* at 8:58-60; 9:2-4, 16:53-54, 17:45-46. The scheduler uses this information from the Resource

³ Dr. Mangione-Smith agreed “maintain” means nothing more than “store.”

EX1019, 98:7-17.

Scoreboard to determine which program instructions have their source data available and can be scheduled for execution. *See, e.g.*, EX1003 at 14:20-30.

Finally, ATI adds another limitation that the unified shader must be “performing only pixel operations until there is once again room for new vertex data.” POR 32. There is no such requirement. Dr. Mangione-Smith agreed that vertex operations could continue simultaneously with pixel operations even when there was no space available for new incoming vertex data. EX1019, 96:17-97:13.

D. Claim 11: Performing Operations “At Various Degrees of Completion Based on Switching Between Instructions”

As discussed *supra* Section II.C., claim 11 is not limited to “unfinished” operations. Nevertheless, this is disclosed in Lindholm.

ATI does not dispute that Lindholm discloses multithreaded processing units that simultaneously process vertex and pixel data. EX1005, 6:11-14, 38-49; EX1006, 7:19-42. Lindholm’s multithreaded processors can switch between vertex and pixel operations by selecting different threads for execution. EX1005, 11:4-14; EX1003, ¶¶107-119; EX1018, ¶¶23-27. This includes switching instructions at various degrees of completion. For example, a POSITA would understand that Figs. 7A and B show sequential or simultaneous execution of different program instructions (“Execute First Program Instructions” 720 and 765, “Execute Second Program Instructions” 725 and 770) where the threads are allocated to different sample (data) types, so that operations on the first sample type (*e.g.*, vertex data)

occur sequentially or simultaneously with operations on the second sample type (e.g., pixel data), where either or both threads may be “unfinished.” EX1018 ¶¶28-30.

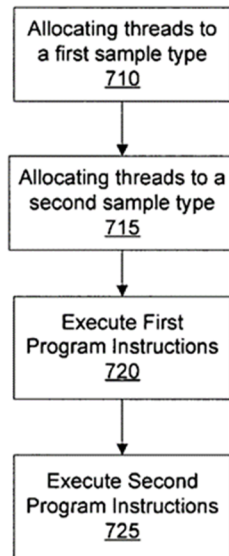


FIG. 7A

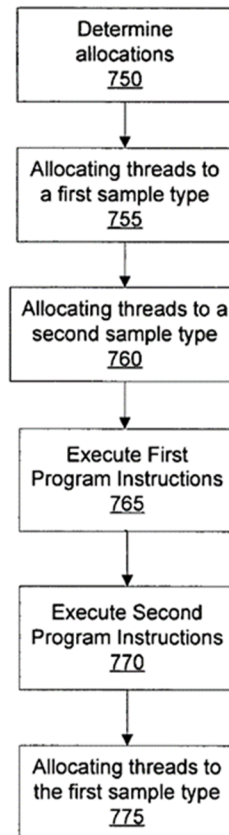


FIG. 7B

E. “In Response To” (Claim 2)

The Lindholm references satisfy the limitations of claim 2. Pet. 22-30. ATI contends, “Claim 2 requires a unified shader processor that executes certain pixel or vertex thread instructions ‘in response to’ *receiving* selected data from a data storage.” POR 33. No such requirement exists. “[G]enerate a pixel color in

response to selected data” merely means the generated pixel color will be a product of the selected data that is processed. Regardless, the ’454 patent describes how the sequencer indicates the next instruction to execute using *a signal* (not receipt of selected data). EX1001, 5:6-15. A POSITA would understand operations to commence based on the program or thread being run (receipt of instructions) and not caused by receiving data. EX1018, ¶38. *See also* EX1019, 112:15-20.

ATI also contends “the space allocated in Register File 350 is for data during thread processing, not for data that is to be processed.” POR 30. No such limitation or distinction exists in the claim. EX1018, ¶43.

F. Claim 6: Memory Separate From the “Store”

Nothing in claim 6 requires the recited memory to be separate from the “store” recited in claim 5. The claim’s plain language includes any memory, including the already recited “store.” Dr. Mangione-Smith did not opine on whether the “store” and memory must be separate. EX1019, 122:9-125:11. Regardless, Lindholm (’685) discloses that sequencer can communicate with buffers 215, 220, Texture Cache 230, and/or local memory 140 (where instruction dispatcher 440 fetches the data), any of which would satisfy the claimed “memory” under ATI’s incorrect interpretation. EX1018, ¶52.

G. “Control Signal” (Claims 7 and 10) and “Arbiter” (Claim 10)

Lindholm discloses an arbiter that provides a control signal. Pet. 37-38; EX1003, ¶¶122-123. “Thread entries,” “state data,” and “priority” are each control signals (*e.g.*, set voltage to establish a “1” or “0”). EX1018, ¶¶58-60. Even under ATI’s dictionary definition, the voltages used to transmit “thread entries,” “state data,” or “priority” would be an “electrical signal” that can control processes. *Id.*

IV. GROUND 2: AMANATIDES AND KOHN RENDER CLAIMS 1-11 OBVIOUS

**A. Executing Operations Based on Available Space (Claims 5-10)
And Performing Operations “Until” Storage is Available (Claims 1, 3, 4)**

Amanatides and Kohn each satisfy claims 1 and 3-5. Pet. 53-57, 61. ATI bases its arguments on non-existent claim requirements that the *unified shader* must (1) “consider” or “evaluate” available space, and (2) “determine” what to execute based on that “consideration.” POR 36-43. *See* Section II.D (functions not required by claims, and moreover, performed by arbiter outside unified shader).

Regardless, Amanatides discloses load balancing based on available space, as summarized by the Board:

Amanatides' priority system explicitly provides for delivering vertex data to the input the FIFOs, i.e., the claimed registers in the general register (claim 1), the claimed shared resources (claims 3, 4), or the claimed store (claims 5–10), when sufficient space becomes available to store it. Ex. 1007, 160. Amanatides notes that “[t]he worst possible scenario is when every input queue is full of GMs.” *Id.* After processing, GMs that are converted to RMs, the RMs are delivered to any input FIFO that has space for it. *Id.* “[W]hen all the output FIFOs are empty and the processors are working on the remaining RMs in their input FIFOs, the processors can begin to deliver GMs again.” *Id.* As Amanatides delivers pixel and vertex data to FIFOs based on the available space, we agree with Petitioner that Amanatides discloses this limitation.

ID 56-58; EX1018, ¶¶70-88. Amanatides selects vertex data only if space is available and after all RMs (pixel data) are processed. Ex. 1007, 159-160. ATI’s “consideration” and “determination” steps at least obvious. Amanatides provides flexible load balancing, where GM and RM packets can be processed out of order based on a decision of available storage space with respect to the FIFOs. EX1007, 157; EX1018, ¶¶80-85.

ATI’s only other argument—that the Petition did not expressly allege “what is the ‘store’” in claim 5—is wrong. POR 42. The Petition expressly identifies “DRAM, VRAM, and on-chip data cache (‘store’).” Pet. 59.

B. Claim 11: Performing Operations “At Various Degrees of Completion Based on Switching Between Instructions”

ATI’s arguments in Section V.B.2. of its POR are irrelevant, as explained *supra* Section II.C. Regardless, a POSITA would understand that pipelined

processors, such as the i860 processor described by Kohn, use pipelining to overlap execution of *instructions*, not of *data* as ATI wrongly claims. EX1018, ¶91. Kohn explains: ““The RISC core contains a pipeline consisting of four stages: fetch, decode, execute, and write.” EX1008, 17. *Instructions* are fetched from the instruction store, decoded into signals for the ALUs, and executed in the ALUs. EX1018, ¶91. The resulting data is *written* to registers. *Id.* The *instructions* move along the pipeline *at various degrees of completion*, not the data.

Kohn discloses that the floating point and integer processor units both have multiple stages. EX1008 22-23. As different operations are selected to be loaded into the pipeline, they may be at different stages of execution. EX1018, ¶¶91-92. For example, Kohn describes performing operations in parallel, in a “dual-operation mode,” where both floating-point and scalar operations could “take advantage of the pipelining capability.” *Id.* A POSITA would understand, based on the combination of Amanatides and Kohn, that the functional units within each i860 processor could not only be performing vertex operations and pixel operations simultaneously, but at various degrees of completion, based on switching between instructions in the instruction cache. EX1018, ¶92.

ATI’s attack on the motivation to combine Amanatides and Kohn has been addressed by Dr. Pfister. EX1003, ¶¶156-157. Dr. Pfister explained the motivation is provided, in part, by the common usage of the Intel i860 processor. *Id.*

C. Claims 2 and 5: “Selected Data”

Claims 2 and 5 recite “selected data” from or maintained in a general purpose register block or store. However, the claims are silent on how, when, or by what component or process the data is selected. EX1018, ¶97. Other than maintaining and executing operations on the selected data, there are no requirements for any specific component to “select” the data. *Id.* The claims also have no limitations requiring the “transmission” of the selected data to the processor unit. The processor unit may, as would be understood by a POSITA, access or retrieve the selected data. *Id.*, ¶97. ATI’s attempt to add “selecting” and “transmitting” limitations is unsupported. POR 44-46.

D. Claims 7 and 10: “Selection Circuit” Is Within the Alleged “Unified Shader”

ATI argues that the bus and FIFOs, which the Petition identifies as part of the selection circuit, are not within the alleged unified shader. POR 47. However, multiple components can be a single claimed unit. *See Textron Innovations, Inc. v. American Eurocopter Corp.*, 498 Fed. Appx. 23, 30 (Fed. Cir. 2012). The Petition clearly identifies the bus and FIFOs as part of the claimed selection circuit, a recited component for the claimed unified shader, regardless of whether they are separate components from the G/R (i860) processor. POR 47; Pet., 59.

E. Claims 7 and 10: “Control Signal”

Amanatides and Kohn disclose an arbiter providing a control signal. Pet. 58-59; EX1003, ¶¶194-195.

ATI disputes that Amanatides discloses providing priority information with the GM and RM packets because a “preset priority scheme” would be a waste of resources. POR 49-50. However, Amanatides expressly discloses that “low priority packets cannot get onto the bus if high priority packets are waiting.” EX1007, 158-159. A POSITA would recognize that low/high priority information must be passed to identify the packet status. EX1003, ¶¶194-196; EX1018 ¶¶116-118, 131-134. Thus, at a minimum, it is obvious that priority information must be part of the packets. *Id.* Further, nothing in Amanatides or Kohn suggests that these priorities are a “preset priority scheme.” Amanatides discloses a flexible graphics system, and making priorities “preset” is the opposite of being flexible. EX1018, ¶117.

Second, ATI argues that priority data is not a “control signal.” As discussed *supra* Section IV.E, priority information can be a control signal. EX1018, ¶118.

F. “Selection Circuit” (Claims 7 and 10) and “Arbiter” (Claim 10)

At least the bus and FIFOs disclosed by Amanatides satisfy the claimed selection circuit. Pet. 58-59. ATI argues that the bus and FIFOs cannot satisfy the claimed “selection circuit” because they are merely transmission lines and memory. POR 51. However, a POSITA understands that FIFOs are not “merely memory”

but typically include Memory/Registers, Read/Write Pointers, Control Logic (for managing pointers, determining when the FIFO is full or empty, and controlling the read and write operations based on status signals), and Status Flags (for identifying space issues). EX1018, ¶¶123-124. Similarly, a POSITA would know that a communication bus is not “merely transmission lines.” Buses operate based on specific protocols, which define the rules for data transfers and manage how devices on the bus request access to the data lines, how conflicts are resolved, and how data integrity is maintained during transmission. *Id.* ¶125. A POSITA would understand that the bus and FIFOs render the claimed selection circuit obvious. *Id.*

ATI argues that “[t]here is no indication in Amanatides that ‘packets’ are ever sent from the FIFOs or bus to DRAM, VRAM, or on-chip cache.” POR 51. This is false. Amanatides describes how packets are sent from FIFOs onto the bus. Pet. 58-59; EX1007, 159; EX1018 ¶127. Likewise, Amanatides describes that packets are received and processed by the processors and stored in VRAM. *Id.*, ; EX1018 ¶128. Finally, as discussed in Section V.C. above, there are no “selection” or “transmission” limitations in the claims. *See* EX1018 ¶¶135-137.

**V. GROUND 3: SELZER AND FISKE RENDER CLAIMS 1-11
OBVIOUS**

**A. Executing Operations Based on Available Space (Claims 5-10)
And Performing Operations “Until” Storage is Available (Claims
1, 3, 4)**

The only basis on which the Board found the combination of Selzer and Fiske lacking for claims 1 and 3-11 was that the Petition failed to demonstrate how Selzer’s “load balancing teachings” disclosed continuing pixel processing until shared resources become available, and vice versa, or how Fiske’s “switching between unfinished threads” and thread prioritization demonstrates switching based on the amount of space available in shared resources. ID 71-72.

First, as explained Section II.D, the claim does not require that the *unified shader* must (1) “consider” or “evaluate” available space, and (2) “determine” execution based on that “consideration.” POR 36-43. *See* Section II.D (functions not required by claims, and moreover, performed by arbiter outside unified shader).

Second, Selzer discloses “load balancing” based on availability of shared resources or storage. “The capability of automatically distributing the workload between geometry and rendering modules means inherently task switching between two jobs within the same application. Supported by the large local memory ... the switching is reduced to *the saving and restoring of all processor registers*, processing interrupt control and *inspecting the rendering FIFO status*.” EX1016, 47-48 (emphasis added). “After rendering an object any processor may run

into an idle state *if there is no rendering data in his input buffer.*” *Id.*, 47 (emphasis added). In Selzer, the rendering module is a unified shader, and its FIFO (input buffer) is a “shared resource” (claims 3 and 4), “store” (claim 5) and part of the “general purpose register block” (claim 1) because it can store either vertex or pixel tasks. EX1003, ¶¶218-220. The Board recognized (in considering Amanatides) that a FIFO can be the general register (claim 1), shared resources (claims 3-4), or store (claims 5-10). ID 57. A “balancing mechanism [can be] activated” in Selzer at any point in time to achieve load balancing, *e.g.*, activating geometry processing in the rendering modules. EX1016, 48.

If the RM “*inspects*” the FIFO and there are no pixel tasks, then that shared resource/register/store is available for vertex tasks and it switches over. [cite, Pfister] Selzer switches freely between processing vertices and pixels, and in each case, first “inspects” the FIFO. *Id.* Selzer allows this flexible processing of geometry (G) and pixel (P) jobs, such that the processing might proceed in any sequence. *Id.* It is therefore obvious to a POSITA that the flexible load balancing of the RM will switch back and forth between rendering/pixel tasks and geometry/vertex tasks based on the availability of shared resources, store, or general register. EX1018, ¶¶141-142. Selzer further discloses that its master module may direct vertex or pixel data to a rendering module in order to load balance, *i.e.*, determine what to process based consideration of available space. EX1018, ¶166.

A POSITA would understand that “the saving and restoring of all processing registers” in Selzer would include inspection of available space and shared resources. *Id.*, ¶140. A POSITA understands that when Selzer switches between geometry/vertex and rendering/pixel tasks, the RM must load the new geometry or pixel data for the new task into the registers, and thus must consider the availability and content of the registers when switching. *Id.*

A POSITA would understand that Fiske discloses processing pixel and vertex data based on storage capacity (shared resources) in connection with its hardware context switching. Pet. 63; EX1010, 211. Each thread in Fiske must have its own hardware context, *e.g.*, constant values and state data. EX1018, ¶¶144-145. When switching between vertex and pixel threads, the processor must save its current context and data to memory and load another context and data from memory. *Id.* A POSITA would have understood these switches depend on considering the availability of storage capacity. *Id.*, ¶146-147; EX1010, 211.

Claim 11 is distinct from other claims because it only requires performing pixel/vertex operations “based on vertex or pixel workload,” not the availability of memory or shared resources. As explained above, Selzer and Fiske clearly disclose switching between pixel or vertex operations based on workload. [cite Pfister]

ATI argues that claim 1 requires ceasing vertex operations and then continuing pixel operations. POR 55. This is wrong, *supra* Section III.B., and

contradicts the '454 patent's alleged improvement of flexible operations over the prior art. EX1001, 2:25-30; EX1018, ¶140.

B. Claim 11: Performing Operations “At Various Degrees of Completion Based on Switching Between Instructions”

ATI's arguments are based on its improper claim construction requiring switching between “unfinished” operations. *Supra* Section II.C. Nonetheless, Fiske discloses that its context switches can occur as a result of a “stall” with a particular thread, which is necessarily “unfinished.” EX1010, 211; EX1018, ¶¶170-171; EX2123, 86:11-15. Even if Fiske does not expressly disclose switching between “vertex and pixel threads” this is obvious based on Fiske *and* Selzer.

C. Claim 2: “In Response To”

ATI's varied constructions of “in response to” are incorrect. *Supra*, Sections III.D. and IV.C.

D. Claims 2, 5-7, 10, 11: “Sequencer,” “Instruction Store,” “Circuitry,” “Selection Circuit,” and “Arbiter”

ATI's argument that Selzer's “sequencer,” “instruction store,” “circuitry,” “selection circuit,” and “arbiter” are not “within” the alleged unified shader is a semantic argument that epitomizes form over substance. Multiple components can be a single claimed unit. *Supra* Section IV.D. The Petition identified Selzer's rendering module as “performing both vertex and pixel operations,” required of the unified shader. Pet. 66-68. As additional limitations are addressed later in the claim(s), the Petition explains how these components satisfy the claimed elements

of the unified shader and are thus part of the unified shader. *See, e.g.*, Pet. 71 (“sequencer”). Moreover, ATI ignores the fact that Petitioner asserts an obvious combination of Selzer and Fiske. Pet. 66-68. To the extent ATI is arguing that the arbiter is “within” the unified shader, that is incorrect. *See supra* Section II.D.; ’454 Patent, Fig. 4A, 4:13-28.

E. Claims 2, 5, 11: “Sequencer” and “Instruction Store” Maintain Instructions

These limitations are disclosed and rendered obvious by the combination of Selzer and Fiske. Pet. 71-72. ATI’s bluster about the geometry module is irrelevant. The Petition expressly shows that Selzer’s sequencer includes an instruction store (“fast static memory”) that maintains instructions. EX1016, 40; Pet. 78.

F. “Control Signal” (claims 7 and 10) and “Arbiter” (claim 10)

The Petition and Dr. Pfister expressly address these limitations. For example, a control signal was shown:

252. In addition, the combination of Selzer and Fiske teaches that a circuit (*e.g.*, master module in Selzer, scheduling arbiter in Fiske) provides information (*i.e.*, data) to the store (*i.e.*, memories and FIFOs) in response to a **signal** dictating whether to process vertex or pixel data. Because the modules in Selzer and Fiske

EX1003, ¶252; Pet. 76-77. ATI’s rebuttal appears to take a semantic issue with the Petition and prior art saying “signal” instead of expressly “control signal,” which is unpersuasive. Likewise, the Petition and Dr. Pfister explained that “because the

modules in Selzer and Fiske are deciding what data to select” (and sending the “signal dictating” the selection), they represent an arbiter. EX1003, ¶¶251-252; Pet. 76-77. Finally, ATI’s arguments regarding express disclosures are not relevant to Petitioner’s obviousness ground.

G. Motivation to Combine Selzer and Fiske

There is ample motivation to combine Fiske and Selzer. Pet. 64-66. Selzer is directed to “automatic load balancing” to distribute load between vertex and pixel processing. EX1016. Fiske discloses thread scheduling based on prioritization where the “goal is to devote as many of the processor resources as possible to the tasks that are known to be critical to overall performance.” EX1010, 211. The goals are the same. EX1018, ¶¶208-210.

Moreover, Fiske teaches how context switching between threads can be done very efficiently, rather than waiting for stalls to resolve. EX1010, 210. This would motivate a POSITA to combine Selzer and Fiske in order to use the multithreaded architecture of Fiske with the flexible graphics processing system of Selzer. EX1018, ¶¶176-178, 211-214.

Selzer discloses “task switching,” which is similar to context switching, and “saving and restoring of all processor registers,” which is similar to hardware context registers in a multithreaded processor. EX1016, 47-48. The similarity between task switching (Selzer) and multithreading (Fiske) provide a strong motivation to

combine because the combination would be easy to accomplish and lead to significant performance increase. Pet. 66; EX1018, ¶¶215-218. Further, a POSITA would have understood that Selzer would use its fast SRAM instruction store for context switches, because instructions of threads that are currently being processed would be stored in the internal SRAM memory of Selzer's DSPs. *Id.*, ¶181.

In sum, a POSITA would have understood the great similarity between the task switching DSP processor in Selzer and the multithreading processor of Fiske. EX1018, ¶¶220-221. Both processors contain internal local storage for data and instructions. EX1016, 40; EX1010, 212. A POSITA would immediately recognize that one routine combination of Selzer and Fiske is to replace the processor of Selzer (DSP) with the multithreaded DSP processor in Fiske. *Id.*, ¶222. It would merely be a matter of swapping the processors and upgrading the software without any additional changes to the graphics system in Selzer. *Id.*

VI. SECONDARY CONSIDERATIONS

The unified shader was neither popularized or productized for PCs by ATI. The unified shader first became prominent with the release of DirectX 10 by Microsoft in 2006. This new DirectX API was a key feature of the Windows Vista operating system. See, <https://en.wikipedia.org/wiki/DirectX> and https://en.wikipedia.org/wiki/Unified_shader_model; EX1018, ¶224.

Before ATI, in November 2006, NVIDIA launched the GeForce 8800 GTX, which was the world's first commercial GPU to use unified shaders for the PC industry and was the first to be fully compliant with DirectX. <https://www.hardwarezone.com.sg/feature-nvidia-geforce-8800-gtx-gts-g80-worlds-first-dx10-gpu/embracing-unified-shader-architecture>. Indeed, the Lindholm patents were invented and developed by NVIDIA. See, https://en.wikipedia.org/wiki/GeForce_8_series; <https://www.hardwarezone.com.sg/feature-nvidia-geforce-8800-gtx-gts-g80-worlds-first-dx10-gpu/embracing-unified-shader-architecture>; EX1018, ¶225. ATI transitioned to unified shaders with its TeraScale architecture, introduced in the Radeon HD 2000 series in 2007. The ATI Xenos GPU was only used in the Xbox 360. See, https://en.wikipedia.org/wiki/Radeon_HD_2000_series; EX1018, ¶226.

Further, ATI fails to show that the R400 (or the Xenos) actually embodied the unified shader claimed in the '454 patent, and the evidence proves the R400 was not used in the Xenos GPU. *Supra* Section III.A-B. ATI provides no analysis and fails to tie any claim limitation to either project/product. Thus, it is not entitled to any presumption. And none of ATI's arguments identify any claim limitation that is linked to any of the "praise" or "success" alleged. The prior art shows that "unified shaders" were known before the invention, and ATI has failed to establish a nexus between any limitations of the '454 patent claims and objective evidence of

secondary considerations. *See Ormco Corp. v. Align Tech., Inc.*, 463 F.3d 1299, 1311–12 (Fed. Cir. 2006); *In re GPAC, Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995).

Dated: May 20, 2024

Respectfully submitted,

/Jeffrey L. Johnson/

Jeffrey L. Johnson
Reg. No. 53,078
BAKER BOTTS, LLP
910 Louisiana Street
Houston, TX 77002-4995
Telephone: 713-229-1234
Facsimile: 713-229-1522
Email: Jeffrey.Johnson@bakerbotts.com
DLBBRealTEKIPRs@bakerbotts.com

Christopher J. Higgins
Reg. No. 66,422
ORRICK, HERRINGTON & SUTCLIFFE LLP
Columbia Center
2100 Pennsylvania Avenue NW
Washington, D.C. 20037
Telephone: (202) 339-8400
Facsimile: (202) 339-8500
Email: 0CHPTABDocket@orrick.com

Steve Baik
Reg. No. 42,281
WHITE HAT LEGAL
P. O. Box 382
San Jose, CA 95002

Telephone: (650) 618-5282
Email: sbaik@whitehat.legal

*Counsel for Petitioner Realtek
Semiconductor Corp.*

CERTIFICATE OF SERVICE

In accordance with 37 C.F.R. § 42.6(e), the undersigned certifies that on May 20, 2024, a complete and entire copy of this **PETITIONER'S REPLY TO PATENT OWNER'S RESPONSE TO PETITION** and Exhibits 1018-1024 were served on Patent Owner via electronic mail at the following addresses:

William A. Meunier
wameunier@mintz.com

Michael T. Renaud
mtrenaud@mintz.com

Adam S. Rizk
arizk@mintz.com

Reza Dokhanchy
rdokhanchy@mintz.com

AMD_IPRS@mintz.com

FOR PETITIONER
IPR2024-00366:
John Schnurer
Yun (Louise) Lu
Kevin J. Patari
Kyle R. Canavera
PERKINS COIE LLP
Email:
PerkinsServiceTCL-
AMDIPRs@perkinscoie.com

/Jeffrey L. Johnson/
Jeffrey L. Johnson (Reg. No. 53,078)
Lead Counsel for Petitioner

**CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME
LIMITATION, TYPEFACE REQUIREMENTS, AND TYPE STYLE
REQUIREMENTS**

1. This **Petitioner's Reply to Patent Owner's Response** complies with the type-volume limitation of 5,600 words, comprising 5,583 words, excluding the parts exempted by 37 C.F.R. § 42.24(a).

2. This **Petitioner's Reply to Patent Owner's Response** complies with the general format requirements of 37 C.F.R. § 42.6(a) and has been prepared using Microsoft® Word in 14-point Times New Roman.

Respectfully submitted,

BAKER BOTTS L.L.P.

/Jeffrey L. Johnson/
Jeffrey L. Johnson (Reg. No. 53,078)
Lead Counsel for Petitioner

Date: May 20, 2024