

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MEDIATEK INC. and MEDIATEK USA INC., and
ARM LTD., AND ARM, INC.,
Petitioner,

v.

ADVANCED MICRO DEVICES, INC. and ATI TECHNOLOGIES ULC,
Patent Owner.

Case IPR2018-00101
Case IPR2018-01148¹
Patent 7,633,506 B1

Before JONI Y. CHANG, BRIAN J. McNAMARA, and
PAUL J. KORNICZKY, *Administrative Patent Judges*.

McNAMARA, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and
37 C.F.R. § 42.73

¹ IPR2018-00101 and IPR2018-01148 have been joined. All references in this Decision are to Papers and Exhibits in IPR2018-00101, unless otherwise noted.

BACKGROUND

On December 12, 2018, IPR2018-01148 was joined to this IPR2018-00101 in which a trial already had been instituted. *See ARM Ltd. and ARM Inc. v. Advanced Micro Devices, Inc. and ATI Techs., Inc.*, Case IPR2018-01148, (PTAB Dec. 12, 2018) (Paper 16, Decision to Institute and Grant of Motion For Joinder.) “Petitioner” refers to MediaTek Inc., MediaTek USA Inc., ARM Ltd., and ARM, Inc., collectively. “Patent Owner” refers to Advanced Micro Devices, Inc. and ATI Technologies ULC, collectively.

On April 27, 2018, we instituted an *inter partes* review of claims 1–9 (“the challenged claims”) of U.S. Patent No. 7,633,506 B1 (“the ’506 Patent”). Paper 13 (“Dec. to Inst.”). Patent Owner filed a Patent Owner Response (Paper 16, “PO Resp.”) and a contingent Motion to Amend (Paper 17, “Mot. To Amend”). Petitioner filed a Petitioner Reply (Paper 28, “Pet. Reply”) and an Opposition to Patent Owner’s Motion to Amend (Paper 29, “Opp. To Mot. To Amend”). Patent Owner filed a Reply to Petitioner’s Opposition to the Motion to Amend (Paper 34, “Reply to Opp. To Mot. To Amend”) and a Sur-reply to Petitioner’s Reply (Paper 35, “PO Sur-reply”). Petitioner filed a Sur-reply to Patent Owner’s Motion to Amend (Paper 41 “Pet. Sur-reply to Mot. To Amend”). Patent Owner also filed a Motion to Exclude (Paper 40, “Mot. To Exclude”) and Petitioner filed a Response to Patent Owner’s Motion to Exclude (Paper 42, “Resp. To Mot. To Exclude”). A transcript of an oral hearing held on January 22, 2019 (Paper 47, “H’rg. Tr.”) has been entered into the record.

We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision is issued pursuant to 35 U.S.C. § 318(a). We base our decision on the preponderance of the evidence. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d).

Having reviewed the arguments of the parties and the supporting evidence, we conclude that Petitioner has demonstrated by a preponderance of the evidence that the challenged claims are unpatentable. We also deny Patent Owner's Motion to Amend.

THE '506 PATENT (EXHIBIT 1001)

According to the '506 patent, a three dimensional (3D) image typically is displayed on a two dimensional array of pixels using a plurality of graphical objects, such as points, lines, and polygons, known as primitives that are the basis of most rendering instructions. Ex. 1001, 1:45–50. Visible primitives that are part of a scene are drawn individually by determining those pixels falling within the edges of the primitives and obtaining attributes that correspond to each of the pixels to determine the displayed color value of the applicable pixels. *Id.* at 1:52–60. The final displayed color of an individual pixel may be a blend of colors from multiple surfaces or layers. *Id.* at 1:65–67. A blending function based on an opacity value associated with each pixel of each primitive can be used to blend colors of overlapping surfaces or layers when the top surface is not completely opaque. *Id.* at 1:61–65. 3D image data represents attributes such as color, opacity, texture, depth, and perspective information. *Id.* at 2:5–6. Graphics processing is the execution of draw commands that may include X and Y coordinates for the vertices of the primitive, as well as attribute parameters for the primitive (color and depth or “Z” data) to generate a display image. *Id.* at 2:6–11.

According to the '506 patent, a graphics chip designed to carry out instruction processing to render graphics on a screen typically has a front-end and a back-end. Ex. 1001, 2:19–20. The front-end receives graphics

instructions and generates geometry defining primitives or combinations of primitives that are processed by the back-end, where they might be textured, shaded, colored, or otherwise prepared for final output. *Id.* at 2:20–26.

When back-end processing of primitives is complete, each pixel in the screen has a specific number value that defines a unique color attribute the pixel will have when drawn. *Id.* at 2:26–29. That final value is kept in a frame buffer for use at an appropriate time. *Id.* at 2:29–31. Systems have become more complex to accommodate three-dimensional (3D) data, requiring a 256-bit system that processes 512 bits in a single logic cycle. *Id.* at 2:32–40. The use of data words with a 256-bit frame buffer is a challenge for the input/output (I/O) system used by the graphics processing back-end because granularity may be too coarse. *Id.* at 2:40–45.

In the '506 patent, geometry representative data presented to the back-end of the graphics chip is divided into data words and provided to one or more parallel pipelines. *Id.* at 2:55–60. The display screen is divided into tiles and a portion of the display screen defined by one or more tiles is serviced by a pipeline. *Id.* at 2:60–64, Fig. 3. Work is allocated to pipelines based on a repeating square pixel tile pattern. *Id.* 5:23–25. The tiling pattern is based on the number of active pipelines. *Id.* at 5:50–51. As shown in Figure 5, logic 520 in set-up unit 515 intersects graphics primitives with the repeating tile pattern, such that a primitive is sent to a pipeline only if it is likely to result in generation of covered pixels. *Id.* at 5:25–28. Set-up unit 515 creates a bounding box based on X, Y coordinates for each vertex of a polygon and the bounding box is compared to the tile pattern and mapped to one or more pipelines. *Id.* at 5:51–67.

Each parallel pipeline comprises a raster back-end having (i) a scan converter to step through the geometric patterns passed to the back end, (ii) a “hierarchical-Z” component to define the borders of the geometry more precisely, (iii) a “Z-buffer” for performing three dimensional operations of the data, (iv) a rasterizer for computing texture addresses and color components for a pixel, (v) a unified shader for combining multiple characteristics for a pixel and outputting a single value, and (vi) a color buffer logic unit for taking the incoming shader color and blending it into the frame buffer using the current frame buffer blend operations. *Id.* at 2:65–3:8, Fig. 5.

ILLUSTRATIVE CLAIM

Claim 1, with claim element designations used in the Petition shown in brackets, is reproduced below:

1. [preamble] A graphics chip comprising:
 - [a] a front-end in the graphics chip configured to receive one or more graphics instructions and to output a geometry;
 - [b] a back-end in the graphics chip configured to receive said geometry and to process said geometry into one or more final pixels to be placed in a frame buffer;
 - [c] wherein said back-end in the graphics chip comprises multiple parallel pipelines;
 - [d] wherein said geometry is determined to locate in a portion of an output screen defined by a tile; and
 - [e] wherein each of said parallel pipelines further comprises a unified shader that is programmable to perform both color shading and texture shading.

GROUND OF INSTITUTION

In our Decision to Institute, we instituted trial on the following challenges to patentability (which were all of the grounds asserted):

Claims 1–3 and 5–9 as obvious under 35 U.S.C. § 103(a) over the combination of Akeley² and Rich³; and

Claim 4 as obvious under 35 U.S.C. § 103(a) over the combination of Akeley, Rich, and Greene.⁴

CLAIM CONSTRUCTION

In our Decision to Institute, we applied the ordinary and customary meaning to the terms not construed. We applied the broadest reasonable interpretation to the following terms that required construction:

Z-buffer logic unit

In our Decision to Institute, we construed “Z-buffer logic unit” to mean “a logic unit that facilitates visibility testing by comparing depth values.” Dec. to Inst. 7–8. The parties do not dispute the substance of our construction of this term.

Hierarchical Z-interface

In our Decision to Institute, we construed “Hierarchical Z-interface” to mean “an interface with a z-buffer logic unit that provides for visibility testing at a coarse level, including, for example, for an entire tile or primitive.” Dec. to Inst. 8. The parties do not dispute the substance of our construction of this term.

Early Z-interface and late Z-interface

In our Decision to Institute, we construed “early Z-interface” to mean “an interface with a z-buffer logic unit that provides for visibility prior to

² Kurt Akeley, *Reality Engine Graphics*, Computer Graphics Proceedings Annual Conference Series, 1993 (Ex. 1004)

³ U.S. Patent No. 5,808,690, issued September 15, 1998 (Ex. 1005)

⁴ U.S. Patent No. 6,646,639 B1, issued November 11, 2003 (Ex. 1006)

shading and texturing.” We construed the term “late Z-interface” to mean “an interface with a z-buffer logic unit that provides for visibility testing after shading and texturing. Dec. to Inst. 8–9. The parties do not dispute the substance of our construction of these terms.

Graphics Chip

Neither party proposed a construction of “graphics chip” prior to our Decision to Institute. The Petition states that the preamble of claim 1 is not limiting because the term “graphics chip” does not describe an essential structure or step. Pet. 32. Petitioner argues that “to the extent the Board finds the preamble is limiting,” the “graphics chip” requires only “a system with one or more chips for graphics processing.” *Id.* at 33. Patent Owner argues that whether or not the preamble is limiting, “the body of the claim requires that the claimed elements be on the same graphics chip.” PO Resp. 24. Therefore, Patent Owner proposes that the broadest reasonable interpretation of “graphics chip” in the context of the front-end is the same graphics chip in the context of the back-end, necessitating that the invention in the challenged claims of the ’506 patent exists on a single graphics chip. *Id.* at 24. Petitioner responds that the ’506 patent refers to a “graphics system” or “graphics chips” in the plural, that the invention is never described in the Specification as consisting of a single graphics chip, and that Patent Owner cites only to specific embodiments of the purported invention. Pet. Reply 10 (citing Ex. 1001, Abstract, 2:49–55, 2:65–3:6, 3:54, 3:62–63, 4:27–32).

The preamble of claim 1 provides antecedent basis for the claim limitations “a front-end in the graphics chip” and a “back-end in the graphics chip.” Noting that when complex graphics are desired in computers

additional components or chips are added to assist with complex instructions to render graphics on a screen, the Specification states “[g]raphics chips may be considered as having a front-end and a back-end,” and that “[t]he front-end typically receives graphics instructions and generates the primitives or combinations of primitives that define geometric patterns,” and “primitives are processed in the back-end where they might be textured, shaded, colored, or otherwise prepared for final output.” Ex. 1001, 1:30–31, 2:19–26. The Specification further states that “[t]he present invention relates to a parallel array graphics system” that “includes a back-end configured to receive primitives and combinations of primitives (i.e. geometry) and process the geometry to produce values to pace in a frame buffer” (*id.* at 2:49–54) and relative to Figure 5 describes “An Embodiment of a Back-End Graphics Chip” (*id.* at 4:67–7:14).

Although the antecedents in claim 1 appear to indicate the claim is drawn to a single graphics chip, the description in the ’506 patent Specification of a back-end graphics chip as a separate entity indicates that the distinction between a single graphics chip and the implementation of a graphics system using multiple chips is not material for purposes of deciding obviousness in this Decision.

Unified Shader

(a) The ITC construction

After the Petition was filed, in a parallel proceeding at the International Trade Commissions (ITC), the Administrative Law Judge (“ALJ”) construed a “unified shader” to be one that performs both color shading and “texture coordinate shading.” PO Resp. 14. Patent Owner

notes that the Commission’s Notice of Review proposed the following, even more limited, construction for “unified shader:”

a single shader circuit capable of performing color shading **and texture coordinate shading**, wherein the single shader circuit may not include separate dedicated hardware blocks that perform separate color and texture operations, and **wherein texture coordinate shading** may include texture address operations, indirect texturing, and bump mapping performed by the unified shader **to modify texture coordinates**.

Id. (citing Ex. 2008, 3) (emphases in PO Resp.). According to Patent Owner, our Decision to Institute did not consider the ALJ’s construction adequately and Patent Owner urges that we now also consider the Commission’s construction. *Id.*

Our Decision to Institute included an extensive discussion of the ITC ALJ’s construction. Dec. to Inst. 10–14. For example, we noted that in addition to “unified shader,” the ALJ construed the function actually recited in claim 1, i.e., “texture shading,” to have its plain and ordinary meaning, which the ALJ stated is “texture shading operations including coordinate *texture mapping* and *texture address operations*.” *Id.* at 11 (citing Ex. 1009, 5–6 (17:12–18:3)). Noting that the ALJ’s construction of “texture shading” includes two terms that are not otherwise defined, i.e. “coordinate texture mapping” and “texture address operations,” we were not persuaded that we could apply the ALJ’s construction in this proceeding. We also noted that the ITC staff and Patent Owner proposed “unified shader” be construed differently in the context of other patents in the ITC proceeding (U.S. Patent No. 8,760,454 (“the ’454 patent”) and U.S. Patent No. 9,582,846 (“the ’846 patent”), i.e., to mean “a single shader circuit configured to perform both vertex and pixel operations.” *Id.* at 14 (citing Ex. 2003, 10). In view of the

undefined terms and the ITC staff's different constructions of unified shader in the context of different patents, we could not determine an ordinary and customary meaning of the terms "texture shading" or "unified shader." *Id.* at 13–14.

The construction in the Commission's Notice of Review cited by Patent Owner also does not provide guidance necessary for this proceeding. The following terms in the construction articulated in the Commission's Notice of Review are not used or defined anywhere in the text of the '506 patent: texture coordinate shading, texture address operations, indirect texturing, bump mapping, and modify texture coordinates. *See* Ex. 2008. These terms appear in the U.S. Patent No. 7,796,133 B1 ("the '133 patent"), the application that purportedly was incorporated by reference into the '506 patent. As much of the construction applied by the ITC and proposed by Patent Owner comes from disclosure in the related '133 patent, we analyze the incorporation by reference below.

Furthermore, although we have considered the findings of fact and conclusions reached by the ITC, we are not bound by them. *See Nobel Biocare Servs. AG v. Intradent USA, Inc.*, 903 F.3d 1365, 1375 (Fed. Cir. 2018) (noting the Federal Circuit is not bound by its prior affirmance of an ITC decision when reviewing a final written decision of the Board because "[a]s the Board correctly observed, the evidentiary standard in its proceedings, preponderance of the evidence, is different from the higher standard applicable in ITC proceedings, clear and convincing evidence"). Here, in the instant Final Written Decision, we have made an independent determination of claim construction and patentability of the challenged claims based on the parties' contentions, the specific evidence presented in

this proceeding, and the standards applicable to *inter partes* review proceedings.

(b) Incorporation by reference of the '133 patent

Patent Owner argues that the '506 patent includes a definition of a unified shader, i.e., “the claimed unified shader must be capable of performing both color shading and texture coordinate/address shading,” “as already adopted by the ALJ, OUII, and the ITC Commission in the corresponding ITC investigation.” PO Resp. 1. Patent Owner’s reliance on the ITC construction reflects an argument that incrementally incorporates into the construction of “unified shader” features that are not found in the literal text of the '506 patent. Claim 1 recites that the unified shader performs “texture shading.” Patent Owner cites the '506 patent Specification to assert that the '506 patent specifically defines the unified shader to require a single shader circuit that can perform both color shading and texture *address* shading. PO Resp. 15 (citing Ex. 2010 (Pfister Dep. 8:10–20)); *see also*, Ex. 1001, 6:45–54 (“A unified shader is so named because the functions of a traditional color shader and a traditional texture address shader are combined into a single unified shader. The unified shader performs both color shading and texture address shading.”). Although the '506 patent does not use the term “texture coordinate shading,” Patent Owner next relies on the disclosure of texture addresses and texture coordinates in the '506 patent and a purported incorporation by reference of the application that led to issuance of the '133 patent to argue that the '506 patent equates texture address shading with texture coordinate shading. Ex. 1001, 6:60–63. Patent Owner then relies on the '133 patent and the ITC’s construction to further limit the construction of “unified shader” in the '506

patent to a shader that performs texture coordinate shading that may include texture address operations, indirect texturing, and bump mapping performed by the unified shader to modify texture coordinates. PO Resp. 14. None of these functions are described in the '506 patent. Even the Commission's construction of a "texture coordinate shader" states that texture coordinate shading *may* include these functions, but it does not require them. *See* Ex. 2008, 3.

The issue of whether the '506 patent properly incorporated the '133 patent by reference came to the Board's attention in the context of Patent Owner's Motion to Amend. H'rg. Tr. 14:18–15:6. We address Patent Owner's Motion to Amend later in this Decision. For purposes of this discussion, we note that although Patent Owner's Motion to Amend proposes to amend claim 1 to recite "texture coordinate shading" explicitly (*see* Mot. To Amend 5), Patent Owner urges us to construe "texture shading" in the context of the unified shader in unamended claim 1 to mean "texture coordinate shading." PO Resp. 15–16.

U.S. Patent Application 10/724,384 ("the '384 application") that matured into the '506 patent was filed in November 26, 2003. Ex. 1001, cover page. As to the "unified shader," the '384 application stated:

A unified shader is so named because the functions of a traditional color shader and a traditional texture address shader are combined into a single, unified shader. The unified shader performs both color shading and texture address shading. The conventional distinction between shading operations (i.e., color texture map and coordinate texture map or color shading operation and texture address operation) is not handled by the use of separate shaders. In this way, any operation, be it for color shading or texture shading, may loop back into the shader and be combined with any other operation.

The functionality of a unified shader is further described in commonly owned co-pending U.S. Patent Application entitled “Unified Shader”, with serial number 10/xxx,xxx, filed December XX, 2003, and is hereby fully incorporated by reference.

Ex. 1017, 307. Thus, as originally filed, the application that matured into the ’506 patent did not describe the functionality of the unified shader (other than to state it performs color shading and texture address shading), but instead sought to incorporate that description by reference to an application that had not been filed and could not be identified.⁵ Patent Owner took no action to address this issue until filing an Office Action Response on July 10, 2007, when Patent Owner amended the Specification’s description of the unified shader to identify U.S. Patent Application 10/730,965 filed December 8, 2003 (“the ’965 application”) as the referenced application.

Ex. 1017, 252.

The ’965 application, which matured into the ’133 patent, is identified as a continuation of abandoned U.S. Patent Application No. 10/716,946 filed on November 18, 2003 (“the ’946 application”), i.e., several days before the November 26, 2003 filing date of the ’384 application that matured into the

⁵ Similarly, in its description of set-up unit 515, the scan converter and hierarchical Z interface, and the scan converter and Early Z interface, the originally filed ’384 application that matured into the ’506 patent stated that “the functionality” of each device “is further described in commonly owned co-pending U.S. Patent Application entitled ‘Scalable Rasterizer Interpolator’, with serial number 10/xxx,xxx, filed December XX, 2003, and is hereby fully incorporated by reference.” Ex. 1017, 305, 306. In an amendment filed on July 7, 2007, Patent Owner amended the Specification to change this incorporation by reference to identify Application No. 10/730,864 (“the ’864 application”), filed December 8, 2003. *Id.* at 251–252.

'506 patent. *See* Ex. 2003. Although the '946 application has not been made of record in this proceeding, as a continuation application, the '965 application that matured into the '133 patent would have been the same as the abandoned '946 application. A review of Patent Office records indicates that before the Office took any action concerning it, the '946 application was abandoned on February 4, 2004, several weeks after the '965 application was filed. We are not aware of any explanation in the record concerning why the '946 application was abandoned and the '965 application was filed in its place.

Figures 9–15 and the corresponding subject matter that appears in the '506 patent beginning at column 9, line 27 through column 14, line 20 were not included in the '384 application as filed on November 26, 2003. *See* Ex. 1017, 312. Thus, the '384 application as originally filed did not include the '506 patent text under the headings “Unified Shader,” “Unified (Pixel) Shader Architecture,” “Shader Code Partitioning,” “Control Logic,” “Register Subsystem,” “Multiple Shaders,” and “ALU I/O Description.” *See* Ex. 1001, 9:27–14:20.

On March 31, 2008, Patent Owner filed “a Substitute Specification that includes subject matter of a co-pending application (serial no. 10/730,965) that was incorporated by reference in its entirety in the originally filed application.” Ex. 1017, 126. As discussed above, the subject matter was not incorporated by reference in the originally-filed application because Patent Owner did not identify a specific application to be incorporated by reference. *See* 37 C.F.R. § 1.57(c). Indeed, the '384 application that matured into the '506 patent included three separate incorporation by reference assertions in the form of application “10/xxx,xxx,

filed on Dec. XX, 2003” and Patent Owner’s July 10, 2007 amendment incorporated two different applications by reference, i.e., the ’965 application concerning the unified shader and the ’864 application concerning the set-up unit, scan converter, and the Early and Hierarchical Z interfaces. The existence of the subject matter to be incorporated by reference is irrelevant until Patent Owner actually identifies the subject matter to be incorporated. *See id.*, Pet. Reply to Opp. to Mot. to Amend 4. Identification of the application to be incorporated by reference occurred years after the original filing date, i.e., on July 10, 2007 at the earliest.

The Substitute Specification filed on March 31, 2008, added Figures 9–15 and the corresponding description, but did not import into the ’384 application (now the ’506 patent) the literal text using the term “texture coordinate shading” or include other material from the ’965 application (now the ’133 patent), e.g., discussions of “indirect texturing,” and “bump mapping.” There is no evidence that this issue was brought to the attention of the ITC.

(c) The description of “unified shader” in the ’506 patent

In our Decision to Institute, we declined to construe the term “unified shader” by itself. Dec. to Inst. 9–24. Rather than construe this term out of the context of the claims, we preliminarily construed the entire phrase “unified shader that is programmable to perform both color shading and texture shading” to mean “a processing mechanism that through an interface receives packets from a rasterizer and has at least one ALU/memory pair that can be programmed to adjust the color of a pixel, and issue a texture request to a texture unit or write received texture values to a memory, and outputs resultant values to a frame buffer.” *Id.* at 24 (emphasis omitted).

Our construction recognized that the recited limitation includes several elements: (i) unified shader; (ii) color shading, and (iii) texture shading.

We turn first to the “unified shader” portion of the limitation. Our construction of the “unified shader” portion of this term is based, in part, on the following description of a “unified shader” in the ’506 patent Specification as amended on March 31, 2008:

One embodiment of a unified shader is shown in the block diagram of FIG. 9. Unified shader **1100** performs per-pixel shading calculations on rasterized values that are passed from a rasterizer unit **1110**. The results of the calculations are sent to frame buffer **1120**. As part of the calculation performed by unified shader **1100**, a texture unit **1130** may receive texture lookup requests from the shader **1100**. The actual shading algorithm used may vary and may be defined by a set of instructions, such as microcode instructions.

Ex. 1001, 9:36–44.

Patent Owner proposes that we construe “unified shader” to “include ‘a single shader circuit capable of performing color shading and texture coordinate shading.’” PO Resp. 18. Patent Owner’s proposed construction is highly functional and introduces at least the following uncertainties: (i) it leaves open to question what makes up the shader circuit, rendering the term “shader circuit” a nonce term; (ii) it changes the claim term “texture shading” to “texture coordinate shading”—thus, on its face, Patent Owner proposes further limiting the claim to a particular type of texture shading, i.e., texture *coordinate* shading; and (iii) it does not define either “texture shading” or “texture coordinate shading,” thereby providing no insight into the functional being performed. In view of these uncertainties, Patent Owner’s proposed construction provides no basis upon which we can interpret the metes and bounds of the claim.

Petitioner agrees with Patent Owner that we should not import into the definition of unified shader the structural requirements that the processing mechanism “through an interface receives packets from a rasterizer and has at least one ALU/memory pair,” and “outputs resultant values to a frame buffer.” Pet. Reply 2, 6–8. The only structural limitation on the unified shader itself in this preliminary construction is that it includes at least one ALU/memory pair. *See* Ex. 1001, 9:45–10:40, Fig. 10. The remaining portions of this preliminary construction concern inputs (i.e., packets from a rasterizer) to and outputs (resultant values to a frame buffer) from the unified shader. Petitioner also proposes that we simplify the “texture shading” element of claim 1 to mean “issue a texture request or receive texture information in response to that request.” Pet. Reply 8. Thus, Petitioner proposes we construe “unified shader” as a “processing mechanism that can be programmed to adjust the color of a pixel and issue a texture request or receive texture information in response to that request.” *Id.* at 9.

Like Patent Owner’s proposed construction, Petitioner’s proposed construction is highly functional and proposes no structure for the unified shader, other than it is a “processing mechanism” programmed to perform certain functions, i.e. issue a texture request and receive texture information in response to that request. As compared to the ’506 patent’s description of operations 5 and 6 performed by the unified shader architecture shown in Figure 10 discussed below, Petitioner’s proposed construction substitutes “texture request” for the term “texture address” and “texture information” for “texture value.” *See* Ex. 1001, 10:15–19. Petitioner’s proposal is consistent with the ’506 patent’s description of operations issuing a texture

address as “texture requests” and receiving in the SRAMs returned “texture data.” *Id.* at 10:15–20.

Petitioner notes that “defining a particular claim term by its function is not improper.” Pet. Reply. 7 (citing *Hill-Rom Svcs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1374–75 (Fed. Cir. 2014)). In this case, however, the claim itself recites the particular function, i.e., “unified shader that is programmable to perform both color shading and texture shading.” Further construing “unified shader” by the function it performs is either redundant or contradictory to the recited function. That the unified shader is programmable to perform both color shading and texture shading begs the question as to what constitutes the unified shader, other than a “circuit” as Patent Owner proposes, or a “processing mechanism” as Petitioner proposes.

The term “unified shader” is not used consistently in the Specification. The Specification describes Figure 9 as “a block diagram of a unified shader according to an embodiment of the present invention.” Ex. 1001, 3:37–38, 9:36–37. The implication from this description of Figure 9 is that the elements of Figure 9 taken together make up at least one embodiment of the claimed unified shader. Figure 9 includes four elements: (i) rasterizer block 1110, shown providing an input to; (ii) unified shader block 1100 that provides inputs to and receives outputs from; (iii) texture unit block 1130; and (iv) frame buffer block 1120 that receives inputs from unified shader block 1100. The Specification states that the interface receives packets from a rasterizer and that the outputs of the unified shader are provided to a frame buffer. Ex. 1001, 9:37–40 (“Unified shader 1100 performs per-pixel shading calculations on rasterized values that are passed from a rasterizer unit 1110. The results of the calculations are sent to the

frame buffer 1120.”); *id.* at 11:14–17 (“Rasterizer 1400 generates packets of data containing information for a block of 16 pixels (4 quads). Each pixel contains one or more sets of texture coordinates (texture addresses) and one or more color values.”).

The presence of block 1100 labelled “unified shader” in Figure 9, which itself is described as being a block diagram of a “unified shader,” is only one inconsistency in the use of the term “unified shader.” In the context of describing the “Unified (Pixel) Shader Architecture,” the Specification also states that “Fig. 10 is a block diagram of a Unified Shader according to an embodiment of the present invention.” Ex. 1001, 9:45–47. Figure 10 is also described as “a unified shader architecture according to an embodiment of the present invention.” *Id.* at 3:37–38. Figure 10 is reproduced below:

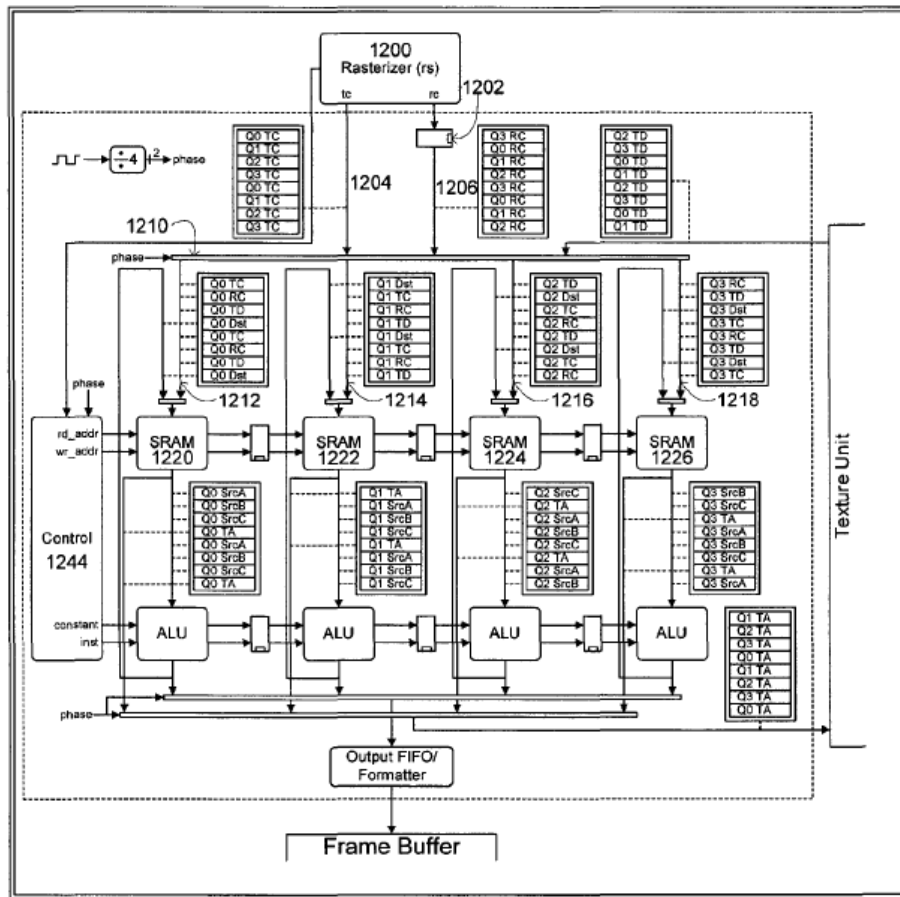


FIGURE 10

The structure of the Unified Shader in Figure 10 includes rasterizer 1200, control 1244, four ALU/SRAM pairs (1220, 1222, 1224, 1226), an un-numbered block labelled output FIFO/formatter, and an unnumbered block labelled Frame Buffer. The Output FIFO/Formatter, busses 1204, 1206, clock delay circuit 1202 connected to bus 1206 (with the clock delay further indicated by the order of Q(0-3)TC and Q(0-3)RC), a divide by four circuit, control block 1244, and the four ALU/SRAM pairs and corresponding busses 1212 (Q0 data), 1214 (Q1data), 1216 (Q2 data), 1218 (Q3 data) are shown inside an un-labeled box of dashed lines. Four way crossbar 1210 labelled “phase” and a similar line labelled “phase” indicate

bi-directional communication with an unnumbered “Texture Unit.”

Although the Specification states that Figures 9 and 10 both show a unified shader (according an embodiment of the invention), Figure 10 does not include a block labeled “unified shader,” as shown in Figure 9.

The Specification describes rasterizer 1200 of the Unified Shader illustrated in Figure 10 generating a texture address (tc) and rasterization color (rc) (in any suitable format) at a rate of one “pixel quad” (a 2x2 tile of pixels) per clock. *Id.* at 9:48–51. Rasterization color rc, delayed by one clock to provide correct interleaving, and un-delayed texture address tc are provided to respective busses 1206, 1204 that pass the packet through four way crossbar 1210 that rotates through four clocks to provide outputs 0–3 to corresponding quads 0–3. *Id.* at 9:51–59. As a result, output 0 of crossbar 1210 contains only quad 0 data, output 1 (1214) contains only quad 1 data, output 2 (1216) contains only quad 2 data, and output 3 (1218) contains only quad 3 data. Each of four identical ALU/SRAM pairs⁶ processes the rc and tc data from the rasterizer, performing the following operations in one four clock cycle:

1. Writes one rasterizer texture address to the SRAM.
2. Writes one rasterizer color value to the SRAM.
3. Reads up to three source operands from the SRAM and executes one shader instruction.
4. Writes the result from the (2nd previous) shader instruction back to the SRAM.
5. Reads one texture address from the SRAM and issues it to the texture unit.
6. Writes one return texture value to the SRAM.

⁶ Memory units other than SRAMs, with space allocated to store input values and intermediate variables needed by the shader program for multiple quads can also be used. Ex. 1001, 9:67–10:3.

Id. at 9:60–10:14. A single mux multiplexes texture requests for the four ALU/SRAM pairs into a single stream containing one texture request every clock, so that the resulting texture data is de-multiplexed and written back into the SRAMs. *Id.* at 10:15–20. Control block 1244 generates SRAM read/write addresses and generates ALU instructions for the first SRAM and ALU 1220—each SRAM/ALU pair receives the same instructions delayed by one clock cycle from the previous one. *Id.* at 10:23–26.

Claim 1 recites a graphic chip with a back-end that comprises multiple parallel pipelines. Claim 1 further recites that the parallel pipelines comprise a unified shader that is programmable to perform both color shading and texture shading. The Specification discusses the parallel pipelines in the backend in the context of Figures 1 and 3, in each case showing the geometry from the front end being routed to the pipelines and the pipelines routing the information to the frame buffer. Ex. 1001, Figs. 1, 3. Figures 1 and 3 do not mention a unified shader. Figure 5 shows a back-end chip embodiment having rasterization pipeline 520 including scan converter 540 with early and hierarchical Z interfaces providing data to rasterizer 560, where it is routed to unified shader 570. *See id.* at 4:67–7:14.

Consistent with Figures 9 and 10, Figure 5 shows unified shader 570 sending and receiving information to texture unit 585. As discussed above, Figure 9 (Ex. 1001, 9:36–37) purports to be a block diagram of a unified shader embodiment in which the unified shader includes the rasterizer, texture unit, frame buffer and a unit called “unified shader.” Figure 10, which also purports to be a unified shader embodiment (*id.* at 9:46–47), illustrates the parallel pipelines (ALU/SRAMs) receiving data from the

rasterizer and indicates that these elements communicate with the texture unit.

Claim 7 recites the parallel pipeline as further comprising a scan converter, a rasterizer, and a texture unit. Applying the doctrine of claim differentiation, we cannot construe the “unified shader” in claim 1 to include the scan converter, the rasterizer, or the texture unit, notwithstanding the descriptions of Figures 9 and 10. As the only purported description of the unified shader itself is in Figures 9 and 10 and these include the rasterizer and texture unit (and Figure 10 shows the ALU/SRAM pairs that make up the parallel pipelines as communicating with the texture unit), we understand one description of a structure that corresponds to the unified shader, i.e., block 1100 in Figure 9 to be the structure within the dashed lines in Figure 10.

Referring to Figure 5, the '506 patent Specification states that:

[U]nified shader 570 works in conjunction with texture unit 585 and applies a programmed sequence of instructions to the rasterized values. These instructions may involve simple mathematical functions (add, multiply, etc.) and may also involve requests to the texture unit. A unified shader reads in rasterized texture addresses and colors, and applies a programmed sequence of instructions. A unified shader is so named because the functions of a traditional color shader and a traditional texture address shader are combined into a single, unified shader. The unified shader performs both color shading and texture address shading. The conventional distinction between shading operations (i.e., color texture map and coordinate texture map or color shading operation and texture address operation) is not handled by the use of separate shaders. In this way, any operation, be it for color shading or texture shading, may loop back into the shader and be combined with any other operation.

Ex. 1001 6:43–59.

Although the above description of Figure 5 articulates that the unified shader is programmed to perform both color shading and texture address shading, it does not describe the program that performs those functions. The discussion of the architecture embodied in Figure 10 describes the operations that are performed by control logic block 1244 and the ALU/SRAM pairs in the unified shader. As to “texture address shading,” the description of Figure 10 states that operation 5 reads one texture address from the SRAM and issues it to the texture address unit; operation 6 writes one return texture value to the SRAM. *Id.* at 10:12–14.

In order to “app[y] a programmed sequence of instructions to the rasterized values” and to “read[] in rasterized addresses and colors” (Ex.1001, 6:44–47), the unified shader must receive packets from a rasterizer. The Specification explains that rasterizer 560 receives its data through an early and hierarchical interface (scan converter 540). *Id.* at 6:1–37. During the hearing, Petitioner conceded that a proper construction of “unified shader” requires some structure in the form of computational units, e.g., ALUs, and access to memory, and that the unified shader receives rasterized values. H’rg. Tr. 10:3–11:5. Consistent with the recitation of a parallel pipeline comprising a unified shader and the description of Figure 10 in the Specification, there must be at least one pair of ALU/SRAMs (or other memory) to carry out the instructions (one ALU/SRAM and another ALU/SRAM working in parallel). Thus, we construe the structure of the claimed unified shader to include at least one computational unit that receives rasterized values and has access to memory.

(d) The meaning of “texture shading”

We now turn our attention to the meaning of the term “texture shading” in claim 1. Petitioner acknowledges that the ’506 patent states “[t]he unified shader performs both color shading and texture address shading.” Pet. Reply 2–3 (citing Ex. 1001, 6:49–54). Petitioner argues that “the only texture operations described in the ’506 patent that are performed by the unified shader involve issuing texture requests and receiving texture information back.” Pet. Reply 3 (citing Ex. 1001, 6:47, 9:40–43, 10:12–20; Ex. 1012 (Pfister Suppl. Decl.) ¶ 10; Ex. 1018 (Wolfe Tr.) 26:4–9).

Patent Owner contends that the ’506 patent specifically defines “unified shader” as requiring texture address shading and equates texture address shading with texture coordinate shading. PO Resp. 16–17 (citing Ex. 1001, 11:14–19). Patent Owner does not claim to have invented either texture address shading or texture coordinate shading. H’rg. Tr. 32:8–10, 14–15. According to Patent Owner, “[w]hat wasn’t conventional was having a unified shader, a single shader circuit that could do that type of specific texture coordinate shading and one that could do color shading as well. That was the invention.” *Id.* at 32:10–13. Without more, however, Patent Owner’s description of the invention merely articulates an idea, i.e., providing a single device that performs both functions—but Patent Owner does not describe the structure or operation of such a circuit.

Patent Owner also argues that “[s]hading, everybody agrees, is a modification of something . . . in color shading, it’s the modification of a color of a pixel. Texture coordinate shading is the modification of a texture coordinate. Texture address shading is the modification of a texture address.” *Id.* at 31:11–15. As Petitioner notes, however, the ’506 patent

does not define texture shading, texture address shading, or texture coordinate shading. *Id.* at 18:7–17.

The text of the '506 patent cited by Patent Owner refers to “texture coordinates/texture addresses” but does not mention “texture coordinate shading.” Indeed, the term “texture coordinate shading” is not used at all in the '506 patent. Although Patent Owner criticizes the Decision to Institute as focusing on the disclosure of “texture coordinate shading” in related U.S. Patent No. 7,796,133 B1 (Ex. 2003, “the '133 patent”) (*see* PO Resp. 17–18), in its Preliminary Response, Patent Owner urged exactly that approach (Prelim. Resp. 12–14). In the Patent Owner Response, Patent Owner again cites the '133 patent “which the '506 patent explicitly incorporates by reference” to argue that the '506 patent equates texture coordinate shading and texture address shading. PO Resp. 18 (citing Ex. 2003, 3:15–29). As previously discussed, Patent Owner also urges that we adopt the ITC’s construction that includes terms from the '133 patent, but there is no indication that the late incorporation by reference or its implications for written description support relative to the '506 patent was ever disclosed to the ITC.

More importantly, in the Decision to Institute, we compared the disclosures in the '506 patent (referring to a “traditional texture address shader”) and the '133 patent (referring to a texture coordinate shader”). Dec. to Inst. 15–16. A further comparison of the '506 patent’s description of Figure 10 and the '133 patent’s description of Figure 2 indicates that the unified shader illustrated in the respective figures is programmed to perform the same six operations. Ex. 1001, 9:60–10:14; Ex. 2003, 5:24–6:9. In view of Patent Owner’s acknowledgement that texture address shading and

texture coordinate shading are equivalent, in order to determine the meaning of “programmed to perform . . . texture shading,” we focus our analysis on the disclosure in the ’506 patent (and the ’133 patent) concerning the steps carried out by the unified shader.

As discussed above, Figure 10 of the ’506 patent shows the parallel pipelines as part of the unified shader and the Specification describes the processing that takes place within the unified shader as a series of six operations performed in one four clock cycle. *Id.* at 9:60–10:14, Fig. 10. The code is partitioned so that texture operations at the same level of indirection are executed in parallel, where indirection level refers to the number of passes through the texture system. *Id.* at 10:42–67, Fig. 11. The ’506 patent provides additional disclosure of the control logic for the Unified Pixel Shader in the context of a state machine embodiment shown in Figure 12 in which a “control token” is generated as data for each block of pixels is received from the rasterizer. *Id.* at 11:1–13. Rasterizer 1400 generates packets of data containing information for a block of 16 pixels (4 quads), where each pixel “contains one or more sets of texture coordinates (texture addresses), and one or more colors.” *Id.* at 11:14–17.

For each instruction in this sequence, the level 0 Texture machine instructs the SRAM's to read a set of texture coordinates, and then issues commands to the texture unit to perform a lookup on these texture coordinates. As data is returned from the texture unit, it gets written into the SRAM's at the appropriate location. Upon receipt of the return data for the last instruction in the level 0 texture sequence, the level 0 texture machine passes the control token to the level 0 ALU machine **1450**.

Ex. 1001, 11:34–42. Patent Owner urges that the ’506 patent equates “texture coordinates” with “texture addresses.” PO Resp. 17–18 (citing Ex. 1001, 11:14–19). At this state, machine control implements

substantially the same operations discussed in the context of the unified shader shown in Figure 10, we consider the use of the term “texture coordinates” in the context of instructing the texture unit to perform a look-up to be synonymous with the term “texture addresses,” i.e., the state machine of the unified shader could perform the texture look-up using either texture coordinates or texture addresses. *See also*, Ex. 1001, 11:16 (referring to “texture coordinates (texture addresses)”). For this reason, we agree with Petitioner that a proper construction encompasses identifying pixels by both texture coordinates and texture addresses and refers more generally to issuing a “texture request” and receiving “texture information” or “texture data.” The ’506 patent consistently describes the operations associated with Figures 10 and 12 as commands to the texture unit to perform a lookup. *See* Ex. 1001, 10:12–14 (operations 5 and 6, read one texture address from SRAM and issue it to the texture unit, then write one value to the SRAM); 12:34–39 (“For each instruction in this sequence, the level 0 Texture machine instructs the SRAM's to read a set of texture coordinates, and then issues commands to the texture unit to *perform a lookup* on these texture coordinates. As data is returned from the texture unit, it gets written into the SRAM's at the appropriate location.” (emphasis added)). *See also*, Ex. 2003, 7:1–6 (the exact same language in the ’133 patent). The texture unit performs the lookup. As the texture unit is a separate element (*see* claim 7), the function performed by the unified shader is to issue texture commands (or requests) and receive responses and store them. Applying the broadest reasonable interpretation, whether or not texture address shading is equivalent to texture coordinate shading, the unified shader carries out at least those operations in the disclosed embodiment, i.e., issue a texture

request (to the texture unit), receive a response (from the texture unit), and store the response (in SRAM).

(e) *Related issues — “packets” and “programmable to”*

In our construction of “unified shader” in the Decision to Institute, we included a reference to “a processing mechanism that through an interface receives packets.” Dec. to Inst. 24 (emphasis omitted). Neither party proposes that the construction be limited to receiving packets. The description of unified shader architecture in Figure 10 states that “rasterizer 1200 generates a texture address (tc) and a rasterization color (rc) in any suitable format and order.” Ex. 1001, 9:47–48. In view of this disclosure, we do not limit the unified shader to receiving information in the form of packets.

Claim 1 recites “a unified shader is *programmable to* perform both color shading and texture shading.” Emphasis added. Although neither party proposes that we construe the term “programmable to,” Patent Owner asserts that “[w]hen a POSA refers to a programmable shader, s/he understands this to refer to newer technologies that began to replace fixed-function pipelines in order to be compatible with DirectX 8, its descendants, and other similarly capable graphics standards.” PO Resp. 56. Noting that Patent Owner’s expert, Dr. Wolfe, despite a lengthy discussion of programmable shaders in the graphics industry, does not reference the ’506 patent on this issue, Petitioner argues that nothing in the ’506 patent supports such a narrow construction. Pet. Reply. 18 (citing Ex. 2009, Wolfe Decl., ¶¶ 83–93). We agree with Petitioner. Neither the ’506 patent Specification nor the ’133 patent Specification ascribes any special meaning to the term “programmable to.” Claim 17 of the ’506 patent recites the same

language as claim 1. Claim 10 of the '506 patent recites “the unified shader performs both color shading and texture shading based on programmable instructions.” The term “programmable to” is not used elsewhere in the '506 or the '133 patents. The '506 patent states:

A unified shader 570 works in conjunction with texture unit 585 and applies a programmed sequence of instructions to the rasterized values. These instructions may involve simple mathematical functions (add, multiply, etc.) and may also involve requests to the texture unit.

Ex. 1001, 6:43–47.

(f) Summary and construction of “unified shader”

The '506 patent claims are drawn to a graphics chip (claim 1) and a method for processing computer graphics (unchallenged claim 10) with a “unified shader” that performs both color shading and “texture shading.” As originally filed, the application that led to issuance of the '506 patent referred to “texture address shading” but did not include a description of the unified shader. The application entitled “unified shader” that led to the '133 patent and was incorporated by reference into the '506 patent application in 2007 refers to “texture coordinate shading” and includes a description of the architecture of an embodiment of the unified shader. The text describing that architecture and the corresponding figures from the application that led to the '133 patent were explicitly added to the '506 patent application in 2008. The description of the architecture common to the '506 patent and the '133 patent describes an embodiment of a unified shader performing six operations. Patent Owner contends that “texture shading” should be construed to mean “texture address shading” and that “texture address shading” is equivalent to “texture coordinate shading.” Whether one refers to the terms “texture shading,” “texture address shading,” or “texture

coordinate shading,” our construction of unified shader must cover the embodiment in the common description in the ’133 patent and the ’506 patent. Therefore, a unified shader must be construed to perform those six operations described as executed by an embodiment of the architecture of the unified shader. Applying the broadest reasonable construction, other features, such as indirect shading, bump mapping, and modifying texture coordinates are not addressed by the disclosed embodiment, and therefore, are not part of the construction. This approach is consistent with the ITC construction cited by Patent Owner that states “wherein texture coordinate shading *may* include texture address operations, indirect texturing, and bump mapping performed by the unified shader to modify texture coordinates.” Ex. 2042, 15–20 (emphasis added).⁷

In consideration of the above, for purposes of this Decision, we construe the term “unified shader programmed to perform color shading and texture shading” to mean: *at least one computational unit with access to memory that receives rasterized values and executes instructions to adjust the color of a pixel, issue a texture request, and store received values.*

⁷ Notwithstanding that the term “texture coordinate shading” does not appear in the literal text of the ’506 patent, the Commission states that texture address shading is “a term used interchangeably with ‘texture coordinate shading’ in the ’506 patent.” Ex. 2042, 18–19.

ANALYSIS OF PRIOR ART CHALLENGES

Claims 1–3 and 5–9 as Unpatentable Over the Combination of Akeley and Rich

Claim 1

Akeley discloses a graphics system dubbed RealityEngine “designed primarily to render texture mapped antialiased polygons.” Ex. 1004, 11 (“Its target capability is the rendering of lighted, smooth shaded, depth buffered, texture mapped antialiased triangles.”). The Petition cites Akeley’s RealityEngine as disclosing a graphics processing system with (i) a front-end having “geometry engines” that receive instructions from a command processor to process graphics data to output primitives/geometry (claim element 1-a); and (ii) a back-end having multiple parallel pipelines, each having a “Fragment Generator” that receives primitives to be placed into a frame buddfer (claim element 1-b) and performs both pixel and texture shading operations and is therefore a unified shader. Pet. 23–24 (citing Ex. 1004 §§ 2, 4; Ex. 1003, Pfister Decl. ¶¶ 58–60). As to the claimed Z-buffer logic unit, the Petition cites Akeley’s disclosure of Image Engines that compare the depth of fragments being processed at a particular location on a screen and discard the fragments that will not be visible. Pet. 24 (citing Ex. 1004 § 2.5), *see also, id.* at 25.

The Petition cites Rich as disclosing an image generation system with a front-end and back-end having a plurality of Processing Elements (PEs) organized as panels, with each panel sharing an internal bus structure on a single chip. Pet. 24. According to Petitioner, Rich’s panels constitute parallel pipelines that perform rasterization, shading/texturing and composition, and generate final pixel values to be written to the frame buffer. *Id.* at 25–26 (citing Ex. 1005, 7:24–27, 9:40–10:44; Ex. 1003, Pfister

Decl. ¶ 62). The Petition further states that the system in Rich assigns processing of a primitive to a pipeline based on the portion of a screen to which a primitive belongs. *Id.* at 26–27 (citing Ex. 1005, 9:26–39, 28:62–64; Ex. 1003, Pfister Decl. ¶ 65). Noting that Rich’s processing elements are flexible multifunction ALUs that perform both color shading and texture shading, Petitioner states that in Rich, after primitives are assigned to pipelines, the processing elements calculate visibility information using z-buffering of the primitives prior to shading through an early Z-interface. *Id.* at 27 (citing Ex. 1005, 7:24–32, 9:43–65).

Petitioner contends that: (1) based on explicit statements in Rich concerning the desirability of offloading graphics processing from a CPU to pipelined and parallel processors with enhanced memory elements possessing a dedicated ALU and a small strip of digital memory, a person of ordinary skill would have been motivated to implement a parallel processing graphics system on a single chip using Rich’s architecture, and (2) Akeley’s RealityEngine is one such architecture using multiple processors that perform operations in parallel pipelines. Pet. 29–30 (citing Ex. 1005, 2:40–67).

Turning to the individual elements of claim 1, Petitioner identifies claim element 1-a as “a front-end in the graphics chip configured to receive one or more graphics instructions and to output a geometry.” *Id.* at 35. Citing Akeley’s “geometry board” as configured to receive one or more graphics instructions, Petitioner notes sections 2.1 and 2.2 of Akeley also disclose a graphics processor interprets each graphics command and a small register space from which the processor accesses incoming commands to be executed, such that the geometry board outputs primitives. *Id.* at 35–37.

Petitioner cites specific portions of Rich as disclosing hardware and receiving instructions that perform the front-end tasks of Akeley (geometric processing, rasterization, shading/texturing, and composition) to convert 3D objects to 2D space and output primitives. *Id.* at 36–38.

Petitioner identifies as claim element 1-b the limitation reciting “a back-end in the graphics chip configured to receive said geometry and to process said geometry into one or more final pixels to be placed in a frame buffer.” Pet. 39. Petitioner notes that Akeley discloses a back-end with Fragment Generators that receive primitives (a projected triangle ready for rasterization) broadcast on a “Triangle Bus,” and creates pixel for a framebuffer. *Id.* at 40 (citing Ex. 1004 § 2). Noting that Akeley distinguishes between pixels generated for rasterization and pixels in the frame buffer, Petitioner cites Akeley’s disclosure of Image Engines, each assigned to a subset of pixels in the framebuffer, that receive the fragments and merge depth and color sample data with the data already stored that the pixel to compute a new aggregate pixel color. *Id.*

Petitioner identifies as claim element 1-c the limitation “wherein said back-end in the graphics chip comprises multiple parallel pipelines.” Pet. 41. Citing Section 2.4 of Akeley, Petitioner argues that a person of ordinary skill “would understand that each Fragment Generator, paired with Image Engines constitutes a pipeline, within the meaning of the ’506 patent.” *Id.* at 41–42. According to Petitioner, each Fragment Generator contains a pipeline that performs “initial generation of fragments, generation of the coverage mask, texture address generation, texture lookup, texture sampling filtering, texture modulation of the fragment color, and fog computation and blending.” *Id.* at 41 (citing Ex. 1004 § 2.4). Petitioner cites Rich as

disclosing a graphics chip for performing back-end processing computations in parallel in which the heart of the system is a Processing Element Array (PEA), made up of 256 processing elements (PEs), organized in groups called “panels” that share a communications bus (parallel pipelines) and operate in parallel from a single instruction stream. *Id.* (citing Ex. 1005, 8:23–26, 12:35–40). Petitioner contends that each of Rich’s panels constitutes a pipeline for pixel processing because each panel performs rasterization, shading/texturing, and composition, and generates final pixel values to be written to the frame buffer. *Id.* at 42–43 (citing Ex. 1005, 940–10:44; Ex. 1003, Pfister Decl. ¶ 62).

Petitioner identifies as claim element 1-d, the limitation reciting “wherein said geometry is determined to locate in a portion of an output screen defined by a tile.” Pet. 44. Petitioner cites Akeley’s disclosure that each pipeline is assigned to a different portion of the framebuffer as evidence that a portion of the screen is defined by a tile. *Id.* (citing Ex. 1004 § 2). Acknowledging that Akeley does not disclose expressly that the system determines geometry to locate in a specific portion, Petitioner cites Rich’s disclosure of dividing the screen into regions, i.e. tiles, to be processed in parallel and comparing each geometry to screen regions to assigning pipelines. *Id.* (citing Ex. 1005, 9:26–39, 28:6–23, 62–64, Fig. 16A).

Petitioner identifies as claim element 1-e the limitation that recites “wherein each of said parallel pipelines further comprises a unified shader that is programmable to perform both color shading and texture shading.” As evidence that Akeley discloses a unified shader, Petitioner cites Akeley’s Fragment Generator as programmable to perform both color shading and

texture shading. Pet. 46–47. Akeley discloses that, upon receiving a projected triangle ready for rasterization from the Geometry Engine on the Triangle Bus, a Fragment Generator computes the intersection of pixels that are fully or partially covered by the triangle and the set of pixels in the framebuffer that is responsible for generating a fragment of these pixels. Ex. 1004 § 2. Color, depth and texture coordinates are then assigned to each fragment and a local copy of the texture memory is indexed by the texture coordinates. *Id.* Petitioner contends that the linear interpolation of resulting samples into a single color constitutes the claimed color shading and that texture address generation, texture lookup, and texture sample filtering performed in Akeley constitutes the claimed texture shading. Pet. 47. Petitioner cites Rich as disclosing a unified shader on a chip in which programmable ALUs (processing elements) can perform both color shading and texture shading. *Id.* at 47–48.

Patent Owner contends that the Petition fails to demonstrate unpatentability over Akeley and Rich because: (i) the Petition does not demonstrate that Akeley or Rich discloses texture address shading or texture coordinate shading (PO Resp. 35–43); (ii) the Petition does not demonstrate that a person of ordinary skill would have been motivated to combine Akeley and Rich (*id.* at 43–52); and (iii) neither Akeley nor Rich disclose programmable shaders (*id.* at 52–59).

The Unified Shader

Although claim 1 recites a unified shader that is programmable to perform color shading and texture shading, Patent Owner argues that the Petition fails because Petitioner does not allege the references disclose “texture address/texture coordinate” shading as required by the construction

of “unified shader.” PO Resp. 36. Patent Owner argues that the texture mapping function in Akeley’s RealityEngine is a color rendering function in which the texture address is used to look up a color value in a texture map and blend that color data into the fragment, but “did not disclose texture coordinate/address shading, which modifies texture coordinates.” *Id.* at 32–33. According to Patent Owner, “texture lookup” is not “texture coordinate/address shading.” *Id.* at 38. Arguing that texture coordinate/address shading is the modification of the initial texture coordinate/address to change which texture information will be retrieved and applied to a pixel, Patent Owner states that none of the operations Petitioner identifies as “texture shading” is texture coordinate or texture address shading. *Id.* at 36–37 (citing Ex. 2002, 1127:6–12; Ex. 2001, 1219:9–11, 1197:23–1198:7, 1224:17–1225:20, 1374: 8–23).

Patent Owner’s argument is premised on a claim construction that is not consistent with the disclosure of “texture shading” in the ’506 patent, as we discuss above. In contrast, the subject matter Petitioner cites in Akeley and Rich is consistent with the operations described in the ’506 patent’s discussion of the unified shader architecture shown in Figure 10. *See* Ex. 1001, 10:12–14 (operations 5 and 6, read one texture address from SRAM and issue it to the texture unit, then write one responsive value to the SRAM); 12:34–39 (“For each instruction in this sequence, the level 0 Texture machine instructs the SRAM’s to read a set of texture coordinates, and then issues commands to the texture unit to perform a lookup on these

texture coordinates. As data is returned from the texture unit, it gets written into the SRAM's at the appropriate location.”).⁸

Petitioner cites Akeley's Fragment Generators as disclosing unified shaders and Rich as disclosing programmable ALUs that implement the functions of Akeley's Fragment Generators in a chip. Pet. 47 (citing Ex. 1005, 7:24–32 (“Figure 5 illustrates the shading/texturing and compositions of the image generation system . . . processing element array 30 comprises 256 separate processing elements 32 . . . Each processing element 32 comprises an 8 bit, multifunction arithmetic logic unit (“ALU”) 33, directly coupled to its own bank of 128 bytes of memory 34.”). As to color shading, Petitioner notes that Akeley discloses interpolation and blending operations, i.e., texture modulation of the fragment color, fog computation and blending. Pet. 47.

As to texture shading, Petitioner contends that Akeley discloses texture address generation, texture lookup, and texture sample filtering. Pet. 47 (citing Ex. 1003, Pfister Decl. ¶ 94). Petitioner notes that, in Rich, processing elements that perform texture operations are also capable of performing rasterization and texture shading, and whether a particular operation is consolidated with the rasterizing processing element or the shading/texturing processing elements is matter of design choice. Pet. Reply. 17, fn 1. Patent Owner contends that “texture address generation” is a separate and distinct operation from “texture address shading” (PO Resp.

⁸ As the '133 patent discloses the same architecture and operational sequence of the unified shader, even if the '133 patent were incorporated by reference, the subject matter of Akeley and Rich is consistent with the '506 patent's description of the unified shader.

37–38), that “texture look-up” is not texture address shading (*id.* at 38–39), and that a texture sample is the texture color fetched during a texture mapping operation (*id.* at 39 (citing Ex. 1004, 110)).

Citing Rich’s description of “texture operations” in Figure 5, the Petition includes Figure 5 of Rich annotated to identify the steps that constitute “texture shading” (the term used in claim 1). Pet. 48–49. Rich states:

FIG. 5 illustrates the shading/texturing and composition functions of the image generation system . . . each processing element 32 optionally calculates one or all of lighting, fog and smooth shading values as seen in block 70 [i.e., color shading]. Texture u, v values are then generated by the processing elements 32 Texture texels are then looked up by reading the texture maps from memory through the video memory interface 44 or PCI Interface 42 and distributing the texture maps to the appropriate processing elements 32 through the central control unit 38. These texture maps are combined with lighting, fog and shading contributions to provide final contribution values as seen in block 72.

Ex. 1005, 10:6–24 (cited in Pet. 48–49), Fig. 5; *see also*, Ex. 1003, Pfister Decl. ¶ 95; Pet. Reply 11–13). Figure 5 is a flow chart of texturing and composition aspects of the image generation of Figure 1. Ex. 1005, 5:10–11. Rich’s disclosure that texture texels are looked up by reading texture maps that are distributed to the appropriate processing elements through a central control unit is similar to the ’506 patent’s description of the unified shader architecture in which a controller operates to cause ALU/SRAM pairs to send texture requests and store the received data. *See* Ex. 1001, 9:60–10:14. Rich’s statement that its Figure 5 “illustrates the shading/texturing and composition functions of the image generation system” is further evidence that Patent Owner’s proposed construction of “texture shading” is

overly narrow. Recognizing that, in Rich, each processing element calculates one or all of the shading values for its assigned contribution (Ex. 1005, 10:6–11), in view of the disclosures in Rich, we agree with Petitioner that the processing elements in Rich perform texture address shading as properly construed, including issuing texture lookup requests. Pet. Reply 10–11 (citing Pet. 47–49; Ex. 1003, Pfister Decl. ¶ 95).

We reach the same result, i.e., that Rich discloses a unified shader that performs the claimed texture shading, even if we construe “texture shading” to require the modification of texture coordinates, i.e., texture coordinate shading, as described by Patent Owner. Referring again to Figure 5, Rich discloses:

The texturing aspects of one embodiment of the present image generation system are described below.

Block 73 illustrates the next function, which is to determine if transparencies modify the contribution coverage. If the texture is transparent then the contribution function is modified and this modified contribution is utilized to modify the contribution values. After transparency determination, contributions are returned to the original processing elements for the home pixels to which they relate as seen in block 74.

Ex. 1005, 10:25–32. Rich’s disclosure of a single chip image system is consistent with the language of claim 1 that requires a unified shader programmable to perform texture shading, even if construed as Patent Owner proposes to require modification of the texture coordinates.

Programmable

Patent Owner also argues that the processing elements in Rich are not “programmable” because they implement a single instruction, multiple data (SIMD) processing array architecture in which the same processing instruction is supplied to multiple individual processors, but operates on a

different data stream. PO Resp. 33 (citing Ex. 1005, 7:19–23; Ex. 2009, (“Wolfe Decl.”) ¶ 61). According to Dr. Wolfe, this architecture “indicates that every pixel in every primitive is processed using the same rendering steps, rather than by having a shader program for each pixel or for each primitive.” *Id.* at 34 (citing Ex. 2009, Wolfe Decl. ¶ 61).

Patent Owner asserts that a person of ordinary skill would have understood Akeley’s RealityEngine architecture to be tied intimately to the known OpenGL graphics command language that uses a fixed-function modal approach for fast hardware-based graphics accelerators, but does not provide for “programmable shaders.” *Id.* at 25–26, 56–57. Patent Owner further contends that the RealityEngine modal, fixed-function pipeline for real time speeds taught by Akeley teaches away from programmable shaders and that modifying Akely to include a programmable shader would require a complete redesign. *Id.*

Patent Owner contends that Rich discloses an architecture based on an array of processing elements and a linear expression evaluator for providing coefficients to a processing element array, but the processing elements in Rich are not programmable shaders. PO Resp. 33, 57–59. According to Patent Owner, Rich ensures that every primitive in a scene is rendered using the same steps of a fixed-function graphics pipeline like Open GL (“[e]ach of the processing elements in Rich is dependent on the same program as all the others and does not run an independent program” (*id.* at 33 (citing Ex. 2009, Wolfe Decl.” ¶ 63))). Patent Owner argues that a person of ordinary skill would have understood that “shaders were engines that could apply a shader program on at least a per-primitive basis,” but “[s]ince Rich provides the exact same sequence of steps to each processing element and for every

vertex and every pixel in the scene, it is only practical and thus only enabled for a very generic rendering techniques and not for programmable shading as that term was used in the art.” *Id.* at 34.

Noting that Dr. Wolfe never references the disclosures in the ’506 patent concerning programmability (Pet. Reply 18–19), Petitioner responds that just as the ’506 patent discloses operations “defined by a set of instructions, such as microcode instructions” (*id.*, citing Ex. 1001 6:43–45; Ex. 1012, Pfister Supp. Decl. ¶¶ 62–67), Rich’s Processing Elements are programmable because “[t]he operations are programmable and determined by a set of microinstructions (*id.* at 18, citing Ex. 1005, 7:45–50; Ex. 1012, Pfister Supp. Decl. ¶¶ 62–67).

We are not persuaded by Patent Owner’s arguments concerning “programmable.” As discussed above, Patent Owner did not propose a construction of “programmable” and we find no special definition of the term in the ’506 patent. Rich expressly states that the processing elements are programmable. Ex. 1005, 7:45–50 (“The operations are programmable and determined by a set of microinstructions”). As Petitioner points out, the processing elements of Rich are programmable to adjust the color of a pixel and to issue texture requests or write texture values to memory. Pet. Reply 18 (citing Ex. 1005 at, *e.g.*, 10:6–24; 21:53–55; 4:50–54; 21:3–14; 26:40–44; 26:58–59; 25:56–60; 20:65–67; Ex. 1012, Pfister Suppl. Decl. ¶¶ 60–61).

Motivation to Combine

The Petition argues that a person of ordinary skill would have been motivated “to implement the functionality of Akeley’s RealityEngine in a single chip processor as disclosed by Rich using its flexible, programmable

processing elements as building blocks.” Pet. 28 (citing Ex. 1003, Pfister Decl. ¶ 69). Petitioner states that Akeley’s well known RealityEngine, like the Pixel Planes and Pixel Flow mentioned in Rich, discloses a system using multiple parallel pipelines to perform graphics processing and that Rich teaches a person of ordinary skill to implement the functionality of such a system in a single graphics chip. *Id.* at 30. Arguing that a person of ordinary skill would have recognized the performance advantages of a single chip implementation and the similarity of the functionality between Akeley and Rich, Petitioner states “Akeley discloses a graphics processing system with multiple parallel back-end pipelines that use *flexible* processors for graphics processing” and “Rich, in turn, discloses a similar graphics processing system that uses *flexible* ALU processing units to replace the processors of prior art systems like Akeley.” Pet. 30–31 (citing Ex. 1003, Pfister Decl. ¶¶ 68–70).

Patent Owner asserts that: (i) Akeley and Rich implement incompatible architectures (PO Resp. 43–47), (ii) Akeley and Rich could not be implemented on a single chip (*id.* at 47–50), and (iii) the Petition’s comparison of Akeley’s RealityEngine to Pixel Plane and PixelFlow is misplaced (*id.* at 50–51).

In support of its argument that Akeley and Rich implement incompatible architectures, Patent Owner describes Akeley as intimately tied to OpenGL’s support of immediate-mode operation in which polygons are drawn in response to each set of commands without waiting for the entire frame to be executed (what Patent Owner describes as a “polygon-at-a-time-in-order” architecture). PO Resp. 44 (citing Ex. 2009, Wolfe Decl., ¶ 70), 45. According to Patent Owner, Rich uses a “scene-at-a-time-out-of-order”

architecture that could not be combined with Akeley without extensive experimentation. *Id.* at 45. Patent Owner states that Rich uses a fundamentally different architecture, gathering all the polygons in a frame and processing them at once using a phased approach in which all vertex calculations are performed for each polygon before it is drawn and pixel contributions are calculated based on the entire database of polygons using a SIMD approach that applies the same operations to each pixel contribution. *Id.* at 44–45. Petitioner responds that the functionality discussed by Dr. Wolfe “has nothing to do with the ’506 patent.” *See* H’rg. Tr. 29:24–30:7.

Patent Owner also argues that, in contrast to Rich in which a central control unit performs all major timing and control functions, Akeley uses distributed control “whereby independent elements of the graphics pipeline determine their own rate of command processing and move data independently when calculations are completed.” PO Resp. 46–47 (citing Ex. 2009, Wolfe Decl. ¶¶ 73–74; Ex. 1004, Abstract. Ex. 1005, 7:57–60).

Petitioner contends that Patent Owner’s argument is irrelevant because Petitioner’s combination of the teachings of Akeley and Rich does not propose to integrate two incompatible architectures, systems, or control. Pet. Reply. 21. According to Petitioner, “[w]e are simply using the central control system of Rich, the architecture of Rich, to implement a graphics pipeline.” H’rg. Tr. 30:10–12. Petitioner argues that the proposed combination would use the components of Rich, the control system of Rich, and the teachings of Rich and that

the only teaching that would be applied to Rich to render obvious the claims of the ’506 patent is to allocate some of the PE [Rich’s processing element] panels to perform front-end processing, and other PE panels to perform back-end

processing, so both geometry and pixel processing can occur at the same time.

Pet. Reply 21 (citing Ex. 1012, Pfister Suppl. Decl. ¶¶ 107–118).

Patent Owner contends that arguments in the Petitioner Reply that the only teaching taken from Akeley is the pipeline structure, such that pipelines in Akeley are implemented in Rich's Processing Elements, improperly presents new argument not responsive to the Patent Owner Response. PO Sur-reply 21–24 (citing Pet. Reply 19–21). Noting that the Petition states a person of ordinary skill would have implemented the functions of Akeley's Fragment Generator using Rich's programmable processing elements to arrive at the unified shader, Patent Owner argues that "Petitioner plainly argued that Akeley's functionality could be shrunk and implemented in a single chip, not that Rich's PEs collectively implemented Akeley's pipeline architecture." PO Sur-Reply 22 (citing Pet. 17). Emphasizing that Akeley uses thousands of chips, Patent Owner contends a person of ordinary skill would not have been able to implement Akeley on a single chip without undue experimentation. PO Resp. 48–49.

Petitioner asserts it did not argue Rich teaches all the components of Akeley could be shrunk into a single chip, but that Rich discloses a flexible architecture using processing elements, a control system and shared memory, instead of entire processors, to implement an entire graphics pipeline on a single chip. Pet. Reply. 19–20.

This argument in the Petitioner Reply is consistent with the Petition's assertion that a person of ordinary skill would be motivated to "to implement the functionality of Akeley's RealityEngine in a single chip processor as disclosed by Rich using its flexible, programmable processing elements as building blocks." Pet. 28 (citing Ex. 1003, Pfister Decl. ¶ 69). Petitioner's

argument that Rich discloses a unified shader on a single chip is responsive to Patent Owner's argument that Akeley's unified shader could not be implemented on a single chip. *See* Pet. Reply 19 (citing PO Resp. 3, 47–50).

Petitioner states:

Rich discloses all of the limitations of Claim 1 of the '506 patent. We argued the combination of Akeley with Rich because Akeley has a more explicit disclosure of a separate front end and back end, for example, and of the multiple parallel pipelines. It has a nice picture. It shows it very explicitly.

. you're just taking Rich and using those processing elements, which can be grouped in panels and subpanels. You are implementing the functionality of a prior art system like Akeley. You're not combining any of the hardware components of Akeley with Rich . . . as Dr. Pfister explains Rich discloses a system that uses flexible ALU processing units to replace the processors of prior art systems like Akeley. Patent Owner says this a new argument. It's not. It's the same position we've had from our original Petition.

H'rg. Tr. 28:21–25, 29:12–15, 17–20.

Indeed, Petitioner's citation to the description of Figure 5 of Rich discussed above appears in the Petition, putting Patent Owner on notice of Petitioner's argument that Rich discloses a single chip graphics system.

Patent Owner also argues that the Petition's comparison of Akeley's Reality Engine to Pixel Planes and PixelFlow is flawed because Rich does not use processing elements or control structures similar to those of Pixel Planes or Pixel Flow and is wholly irrelevant to whether Rich is compatible with RealityEngine. PO Resp. 50–51. Patent Owner also argues that modifying Akeley to include programmable shaders would require a complete redesign. *Id.* at 49.

The premise of Patent Owner's argument is a physical combination of Akeley and Rich not proposed by Petitioner. The Petition cites prior art graphics pipelines, such as PixelFlow and Pixel Plane mentioned in Rich and the RealityEngine of Akeley, to establish that a person of ordinary skill would have considered pipelining and parallel processing when determining how to arrange the processing elements and panels in Rich. Pet. Reply 24.

In consideration of the above, we are persuaded that Petitioner has demonstrated a person of ordinary skill would have been motivated to combine the teachings of Akeley and Rich as proposed in the Petition to arrive at a unified shader, as recited in claim 1.

In consideration of the above, we agree that Petitioner has demonstrated by a preponderance of the evidence that claim 1 is unpatentable over the combination of Akeley and Rich.

Claim 2

Claim 2 depends from claim 1 and recites that each of the parallel pipelines comprises "a FIFO unit for load balancing said each of said pipelines." Petitioner notes that Akeley includes a FIFO before and after each of the Fragment Generators Petitioner cites as unified shaders. Pet. 50. In addition, Petitioner cites Akeley's disclosure that FIFOs are used to smooth the flow of rendering commands and to ensure that the processors are used efficiently. *Id.*

Patent Owner contends that there is no need for smoothing the flow of commands in Rich operating under a SIMD paradigm in which a single command is provided at the same time to every processing element. PO Resp. 60. Petitioner responds that Patent Owner's argument is based on the inaccurate premise that the claimed FIFOs are for state data, rather than for

load balancing, as recited in claim 2. Pet. Reply 25. Petitioner notes that the purpose of load balancing is to regulate data, as opposed to commands, to the pipelines and is equally applicable to systems such as Akeley and SIMD systems such as Rich. *Id.* at 26. Petitioner emphasizes that Akeley states the FIFOs can be used to “smooth the imbalances to data-dependent processing such as clipping” and that FIFOs are regularly used to moderate the flow of data to and from processing units. *Id.* at 25–26 (citing Ex. 1004, § 2:1). Thus, we agree that Petitioner has shown by a preponderance of the evidence that claim 2 is unpatentable over the combination of Akeley and Rich.

Claim 3

Claim 3 depends from claim 1 and recites that each of the parallel pipelines comprises “a z buffer logic unit” and “a color buffer logic unit.” As discussed above, for purposes of this Decision, we apply Petitioner’s construction of “z buffer logic unit” to mean “a logic unit that facilitates visibility testing by comparing depth values.”

As to the z buffer logic unit, Petitioner cites Akeley’s Image Engines as performing the functions of the claimed z buffer logic unit. Pet. 51. Petitioner notes that Akeley discloses its Image Engines calculate a z value for each fragment output by the Fragment Generator, compare that value to the current depth z value of the pixel at issue, and, if the fragment is closer than the stored value, replaces the stored value with the value of the fragment. *Id.* at 51 (citing Ex. 1004 § 2.5).

Petitioner also contends that Rich discloses the functionality of Akeley’s Image Engines can be implemented as part of an integrated chip. *Id.* at 52. According to Petitioner, a z buffered system that calculates visibility information and then compares that information to discard non-

visible primitives is a z buffer logic unit that was a well-known feature of such chips. *Id.* (citing Ex. 1005, 9:43–65).

As to the color buffer logic unit, i.e., logic that blends incoming colors with colors already in the frame buffer (Ex. 1001, 7:10–14), Petitioner contends that Akeley discloses the same operation as that described in the '506 patent. Pet. 54 (citing Ex. 1004 § 2.5 (“If any changes made to the pixel’s contents, the aggregate pixel color is recomputed by averaging the sub-pixel sample colors, and is immediately written to the displayable color buffer that will contain the final image” (emphasis omitted)), § 3.1 (“If pixel blending is enabled, fragments are blended directly into the color buffer. . . ”)).

Patent Owner does not respond to Petitioner’s contentions concerning claim 3. We are persuaded that Petitioner has demonstrated by a preponderance of the evidence that the elements of claim 3 are disclosed by the combination of Akeley and Rich and that claim 3 is unpatentable.

Claim 5

Claim 5 depends from claim 3 and recites that “said z buffer logic unite interfaces with said unified shader through a late Z interface.” As discussed above, in this Decision, we apply Petitioner’s proposed construction of “late Z interface” to mean “an interface with a z buffer logic unit that provides for visibility testing after shading and texturing.” Petitioner contends that Akeley’s Image Engines, which remove fragments that are unobstructed from the view of the screen, interface with its Fragment Generators through a late z interface because they perform visibility testing after shading is complete. Pet. 55– 56. Petitioner also notes that Rich discloses z buffering can be done at various points in a

graphics pipeline, e.g., after shading. *Id.* (citing Ex. 1005, 10:65–11:2; Ex. 1003, Pfister Decl. ¶ 108).

Patent Owner does not respond to Petitioner’s contentions concerning claim 5. We are persuaded that Petitioner has demonstrated by a preponderance of the evidence that the elements of claim 5 are disclosed by the combination of Akeley and Rich and that claim 5 is unpatentable.

Claim 6

Claim 6 depends from claim 1 and recites that the graphic chip comprises “a setup unit for directing said geometry into one of said multiple parallel pipelines *wherein said geometry is determined to locate in a portion of an output screen defined by a tile.*” The italicized language of claim 6 is the same as that recited in claim element 1-d of claim 1, from which claim 6 depends.

As to the setup unit, Petitioner cites the disclosures in Akeley of multiple pipelines each containing a Fragment Generator performing the functions of the claimed unified shader and each being assigned to a different set of pixels from the display screen. Pet. 56 – 57. Petitioner also cites Rich’s disclosure of a multi-pipeline system, each pipeline containing at least one processing element performing the functions of the unified shader that can be implemented in the graphics chip. *Id.* As to the setup unit, Petitioner also cites Rich’s central control unit (CCU) that determines in which section of the screen the primitive is located and directs the geometry into one of the multiple parallel pipelines thereby determining which processing elements receive the geometry for processing. *Id.* at 57–59.

Patent Owner does not respond to Petitioner's contentions concerning claim 6. We are persuaded that Petitioner has demonstrated by a preponderance of the evidence that the elements of claim 6 are disclosed by the combination of Akeley and Rich and that claim 6 is unpatentable.

Claim 7

Claim 7 depends from claim 1 and recites that the pipeline further comprises a scan converter, a rasterizer, and a texture unit. Petitioner notes that Rich discloses a graphics chip with multiple pipelines comprising processing elements each having a multifunctional ALU that can perform shading operations and backend operations including rasterization, scan conversion and texturing. Pet. 60.

Patent Owner does not respond to Petitioner's contentions concerning claim 7. We are persuaded that Petitioner demonstrated by a preponderance of the evidence that the elements of claim 7 are disclosed by the combination of Akeley and Rich and that claim 7 is unpatentable.

Claim 8

Claim 8 depends from claim 1 and recites that "the unified shader is operative to apply a programmed sequence of instructions to rasterized values and is operative to loop back to process operations for color shading and/or texture address shading." As Petitioner notes, Rich discloses programmable processing elements that perform shading and texture operations after rasterization, i.e., on rasterized values. Pet. 61–62 (citing Ex. 1003, Pfister Decl. ¶¶ 119–120). Rich also explicitly discloses that calculations are done in loops. Ex. 1005, 27:18–36. Petitioner states that, in Rich, processing elements acting as unified shaders contain their own local memory and, therefore, can loop back through the ALU multiple times. Pet.

62– 63 (citing Ex. 1003, Pfister Decl. ¶¶ 122–3). Citing Figures 19-1 and 19-2, Petitioner states that, in Rich, processing elements save an output to a local scratch pad memory in the back to use that output as an input to a subsequent shading operation. *Id.* at 63–64 (citing Ex. 1005, 30:52–31:27; Ex. 1003, Pfister Decl. ¶ 123).

Patent Owner does not respond to Petitioner’s contentions concerning claim 8. We are persuaded that Petitioner has demonstrated by a preponderance of the evidence that the elements of claim 8 are disclosed by the combination of Akeley and Rich and that claim 8 is unpatentable.

Claim 9

Claim 9 depends from claim 1 and recites that “the unified shader is operative to apply a programmed sequence of instructions such that a first set of instructions are executed to perform color shading and a second set of instructions are executed to perform texture shading.” Petitioner cites Rich as disclosing flexible processing elements to perform the functions of both color shading and texture shading in a program using sequences of instructions. Pet. 65. Citing Figure 5 of Rich, Petitioner notes that step 70 (i.e., calculating lighting, fog and smooth shading) constitutes color shading and precedes texture shading steps 71 and 72 in which texture u, v values are converted to MAP addresses, and textures are looked up and combined with the lighting and fog to produce a final contribution value. *Id.* at 66.

Patent Owner does not respond to Petitioner’s contentions concerning claim 9. We are persuaded that Petitioner has demonstrated by a preponderance of the evidence that the elements of claim 9 are disclosed by the combination of Akeley and Rich and that claim 9 is unpatentable.

Claim 4 as Obvious Over the Combination of Akeley, Rich, and Greene

Claim 4 depends from claim 3 and recites that “said z buffer logic unit interfaces with said scan converter through a hierarchical Z interface and early Z interface.” As previously discussed, for purposes of this Decision, we apply Petitioner’s proposed constructions of “early Z interface to mean “an interface with a z buffer logic unit that provides for visibility prior to shading and texturing” and “hierarchical Z interface” to mean “an interface with a z-buffer logic unit that provides for visibility testing at a coarse level, including, for example, for an entire tile or primitive.”

Petitioner notes that Rich discloses a graphics chip may be implemented with a z buffered system. Pet. 67–68. Recognizing that Rich does not discuss what type of z buffering should be used, Petitioner argues that a person of ordinary skill would have been motivated to look for efficient buffering applications, such as those taught by Greene, which discusses both the need for a z buffering graphics systems and numerous ways to implement z buffering. *Id.* at 68 (citing Ex. 1006). Petitioner cites Figure 4 of Rich, which illustrates a z buffer performs visibility testing to discard contributions obscured by nearer primitives (steps 60 and 62), prior to shading and texturing (steps 61 and 63), as evidence of buffer logic that interfaces with a scan converter through an early z-interface. *Id.* at 70–71. Recognizing that neither Akeley nor Rich disclose a hierarchical interface, Petitioner cites the disclosure in Greene of beginning visibility testing at a coarse level and incrementally moving through finer and finer levels until reaching a standard early z-buffering stage. *Id.* at 72 (citing Ex. 1006 3:14–22).

Patent Owner contends the Petition fails with respect to claim 4 because Petitioner has not shown a person of ordinary skill would have combined Green with Akeley and Rich. PO Resp. 60–61. Patent Owner contends that, although Greene may be compatible with other fixed-function pipelines, because Greene depends on a centralized pyramidal data structure that requires propagation of changes at lower levels to higher levels, Greene does not teach how central data structures can be used in a multiple pipelines or multiple processing elements, as in Akeley or Rich. *Id.* We agree with Petitioner that, in the absence of an explanation of why the use of multiple pipelines would impact the implementation of Greene’s hierarchal buffer, the preponderance of the evidence favors Petitioner’s argument that a person of ordinary skill would have combined the hierarchical z-buffering of Green with Akeley and Rich. Thus, we are persuaded that Petitioner has demonstrated by a preponderance of the evidence that claim 4 is unpatentable over the combination of Akeley, Rich, and Greene.

MOTION TO AMEND

Introduction

Patent Owner proposes to substitute amended claim 22 for independent claim 1 and to substitute dependent claims 23–30, which depend directly or indirectly from claim 22 for claims 2–9 which depend directly or indirectly from claim 1. The substance of Patent Owner’s Motion to Amend is to limit further the “unified shader” as follows:

wherein the unified shader comprises a single shader circuit capable of performing both color shading and texture coordinate shading, wherein texture coordinate shading modifies a texture coordinate.

Mot. To Amend 5–6.

Written Description Support

When filing a Motion to Amend, Patent Owner must set forth the support in the *original disclosure* of the patent for each proposed substitute claim, i.e., the patent owner must identify clearly the written description support in the disclosure corresponding to the earliest date upon which Patent Owner seeks to rely. 37 C.F.R. § 42.121(b)(1), 42.121(b)(2); *see Lectrosonics, Inc. v. Zaxcom, Inc.*, Case IPR2018-01129, -01130, slip op. at 7–8 (PTAB Feb. 25, 2019) (Paper 15) (precedential). Patent Owner contends that the proposed substitute claims are supported by the original application that became the '506 patent and that the substitute claims also have written description support in the '133 patent that was incorporated by reference in its entirety into the '506 patent and the '965 application from which the '133 patent claims priority. Mot. To Amend 4. We understand Patent Owner's assertion that the claims are supported by the original application that became the '506 patent to be the earliest date on which Patent Owner relies for written description support of the substitute claims.

Petitioner contends that because the '965 application was not part of the original disclosure and did not exist when the application that led to the '506 patent was filed, it cannot provide written support for the claims. Opp. To Mot. To Amend 6–7 (citing *Ariad Pharm. Inc. v. Eli Lilly Co.*, 598 F.3d 1336, 1355 (Fed. Cir. 2010) for the proposition that anything that occurs after the filing date of the patent is irrelevant to whether the original, as-filed specification provides written description support for the claims).

Petitioner also notes that essential material can only be incorporated by reference by way of a U.S. patent or a U.S. patent application publication.

Opp. To Mot. To Amend 9–10 (citing 37 C.F.R. § 1.57(c)). Petitioner contends, therefore, that even if the '133 patent application had existed at the time of filing, because it was filed with a nonpublication request (and issued after the '506 patent), the application that led to issuance of the '133 patent was not published before the '506 patent issued and could not provide written description support. Opp. To Mot. To Amend 10.

Petitioner also contends that, although the purported incorporation by reference is of the entire '965 application that led to the '113 patent, Patent Owner's choice to add verbatim the material in the '506 patent that now appears at column 9, line 28 through column 14, line 13 concerning the structure, architecture, and operations of a unified shader, indicates that other material concerning texture coordinate shading and indirect texturing is not applicable to the '506 patent. Opp. To Mot. To Amend 11.

Earlier in this Decision, we noted that the application that matured into the '506 patent did not describe the functionality of the unified shader (other than to state it performs color shading and texture address shading), but instead sought to incorporate that description by reference to an application that had not been filed and could not be identified. Ex. 1017, 307. The '506 patent as originally filed does not mention "texture coordinate" or "texture coordinate shading." *See* Ex. 1017, 295–326. Patent Owner also does identify where the '506 patent as originally filed describes texture coordinate shading as modifying a texture coordinate.

We extensively addressed above the purported incorporation by reference of the '133 patent into the '506 patent. The application for the '506 patent filed on November 26, 2003 included several attempts to incorporate by reference one or more applications identified as having serial

number 10/xxx,xxx filed on December XX, 2003. Thus, the application that issued as the '506 patent attempted to incorporate by reference unidentified applications that had not yet been filed. We determined that the purported incorporation by reference was inadequate at least until Patent Owner filed an amendment in 2007 identifying two such applications. Ex. 1017, 252; *See* 37 C.F.R. § 1.57(c). It is immaterial that the '965 application that led to the '133 patent was a continuation of an application filed several days before the filing date of the application that led to the '506 patent. The fact remains that no application was actually identified until 2007. Moreover, there is no evidence that the '965 application or its parent were published before issuance of the '506 patent. Thus, it is not clear that the subject matter from the applications that led to issuance of the '133 patent was incorporated by reference before the '506 patent issued. In any case, the “unified shader” subject matter could not have been incorporated into the application for the '506 patent as originally filed. As Patent Owner relies on the original filing date of the application that led to the '506 patent as support for the Motion to Amend, the amendment lacks written description support.

We also are not persuaded by Patent Owner's arguments that Petitioner acknowledged the incorporation by reference in the ITC proceeding. Reply to Opp. To Mot. To Amend 2. Whether the issue was brought to the attention of the ITC is not a matter for this proceeding, where it clearly has been made an issue.

Patent Owner does not propose substitute claims that recite the language in the '506 patent Specification, i.e., texture address shading. Instead, Patent Owner seeks to insert the term “texture coordinate shading” from the '133 patent. Patent Owner argues that the mention of texture

address shading in the original '506 patent is sufficient written description support for a claim reciting “texture coordinate shading” because “texture address shading” and “texture coordinate shading” refer to the same process. *Id.* at 3. We are unpersuaded by Patent Owner’s argument that Petitioner’s ITC expert acknowledges the equivalence of these terms. Reply to Opp. To Mot. To Amend 3, 6 (citing Ex. 2001, 1220:12–14, 17–18). Much of that testimony was offered on December 1, 2017 in the context of claim constructions already being applied in the ITC (*see* Ex. 2005 claim construction order entered November 8, 2017), where the incorporation by reference of the '965 application went unchallenged. That is not the case in this proceeding.

In view of the circumstances, we are persuaded that the substitute claims are not supported by the original application that became the '506 patent (Mot. To Amend 4) and we deny the Motion to Amend as lacking written description support.

Texture Coordinate Shading

We also consider whether “texture coordinate shading” recited in the proposed substitute claims is disclosed by the references. Petitioner asserts that texture coordinates are in texture space relative to location (0,0) in the texture image, but that texture addresses point to a specific texel in a screen and can be used to fetch a texture from memory. Opp. To Mot. To Amend 16 (citing Ex. 1012, Pfister Suppl. Decl. ¶¶ 25–32). Thus, unlike texture coordinates, texture addresses are not relative values and are not normalized to a limited range of (0, 1). *Id.* According to Petitioner, texture coordinate shading is an operation resulting in texture coordinates and not an operation

using texture addresses, as the '506 patent describes. *Id.* at 17 (citing Ex. 1012, Pfister Suppl. Decl. ¶¶ 41–47).

To the extent that texture coordinate shading involves modifying texture coordinates, Petitioner cites Rich's disclosure of perspective correction in which texture coordinates (u, v) are modified by the Processing Element by dividing the texture coordinates by a depth value w prior to using those coordinates to retrieve texture map information. Opp. To Mot. To Amend 17 (citing Ex. 1012, Pfister Suppl. Decl. ¶¶ 70–75). Patent Owner rebuts Petitioner's position by arguing that in the ITC Dr. Edwards testified that perspective correction is part of generation and cannot be texture coordinate shading. Reply to Opp. To Mot. To Amend 7 (citing Ex. 2001, 1220:22–1223:6, 1224:9–16). As discussed above, however, Dr. Edwards' testimony is offered in the context of different claim constructions and in a forum that requires clear and convincing evidence of invalidity, as opposed to this proceeding where the standard is preponderance of the evidence that a claim is unpatentable.

We further note that Patent Owner cites the Commission's opinion that texture coordinate shading broadly means any modification to texture coordinates, which can be accomplished by the unified shader through indirect texturing, bump mapping, or texture address operations. *Id.* at 6 (citing Ex. 2042, 18–19). Petitioner notes that it is undisputed that Rich's perspective correction modifies texture coordinates by performing division on them. Pet.'s Sur-Reply to Mot. To Amend 7. Petitioner points out that Figure 5 of Rich shows texture coordinates (u, v) undergoing perspective correction after color shading and before texture lookups, all of which occurs after generation of coordinates during rasterization. *Id.* at 8.

To the extent that texture coordinate shading involves modifying texture coordinates, as urged by Patent Owner, Petitioner also cites environment mapped bump mapping disclosed by Collodi⁹ as disclosing texture coordinate shading. Opp. To Mot. To Amend 22–24. Although Collodi is cited by Petitioner in related IPR2018-00102, Petitioner notes that Patent Owner does not discuss Collodi or any of the other prior art cited in IPR2018-00102 in the Motion to Amend in this proceeding. *Id.* at 15. At paragraph 18, Collodi explicitly discloses that, in a preferred embodiment, the generation of a two dimensional bump map vector to be passed from the Programmable Shading Unit to a Vector Shading Unit is not limited to texture look-up stating “[t]he Programmable Shading Unit may be programmed to generate the bump map procedurally or provide the bump map vector as a result of a texture map lookup (or a combined result of multiple texture map lookups).” Collodi’s environmental bump mapping modifies texture coordinates, thereby conforming to Patent Owner’s definition of texture coordinate shading. Pet. Sur-Reply to To Mot. To Amend 5–6. Petitioner notes that in Collodi’s disclosure of bump mapping texture coordinates (u,v) obtained from a texture map lookup are modified by a height value “h,” and that the bump map vector can be modified multiple times by additional texture map lookups. Opp. To Mot. To Amend 23–24 (citing Collodi ¶ 18). In addition, bump map vector b can be combined with an interpolated normal vector n to produce composite surface angle vector c that is used to modify the final pixel value when the vector is used in another texture map lookup. *Id.* Thus, even under Patent Owner’s

⁹ U.S. Patent Application Publ. No US2003/0076320 A1 (Ex. 1007 in *MediaTek v. Advanced Micro Devices, Inc.*, Case IPR2018-00102).

definition of “texture shading” as “texture coordinate shading,” which requires modification of texture coordinates, that feature is disclosed in Collodi.

Thus, in the context of the proposed Motion to Amend, Petitioner has demonstrated that, even if we adopt Patent Owner’s arguments that texture coordinate shading is disclosed in the ’506 patent, to the extent that such texture coordinate shading comprises modifying the coordinate as recited in the substitute claim, modification of the texture coordinates in the unified shader is disclosed at least by Rich. Therefore, we deny the Motion to Amend on this basis as well.

MOTION TO EXCLUDE

Patent Owner moves to exclude Exhibits 1037 and 1038 as irrelevant, prejudicial, and not properly authenticated. Mot. To Exclude 1–2. Patent Owner also argues that Petitioner did not cite Exhibits 1037 and 1038 in any pleading and that these exhibits were filed two weeks after the last paper in which Petitioner was allowed to submit evidence.

Petitioner states that it filed and served Exhibits 1037 and 1038 in response to Patent Owner’s objections to Exhibit 1020 and to demonstrate that Exhibit 1020 was publically accessible. Resp. to Mot. to Exclude 1. As Patent Owner has not moved to exclude Exhibit 1020, Petitioner states that Exhibits 1037 and 1038 are not necessary. *Id.* at 2. We do not refer to Exhibits 1037 and 1038 in this Decision. In view of the circumstances, we dismiss Patent Owner’s Motion to Exclude as moot.

MOTION TO SEAL

Patent Owner moves to seal the transcript of an August 30, 2018 conference call (Exhibit 2039 in each case) and requests entry of a stipulated protective order (i.e, the Board's Default Protective Order) on the grounds that pages 33:2 through 35:18 contain confidential business information. Patent Owner also notes that on September 14, 2008, it filed a redacted transcript (Exhibit 2034). Petitioner does not oppose entry of the protective order or Patent Owner's Motion to Seal. In consideration of the circumstance, Patent Owner's Motion to Seal is granted.

CONCLUSION

Having reviewed the pleadings and the evidence of record, we conclude that Petitioner has demonstrated by a preponderance of the evidence that: (i) challenged claims 1–3 and 5–9 of the '506 patent are unpatentable under 35 U.S.C. § 103 as obvious over the combination of Akeley and Rich, and (ii) claim 4 is unpatentable as obvious over the combination of Akeley, Rich, and Greene.

We also deny Patent Owner's Motion to Amend as not supported by the written description of the '506 patent application as filed and as not patentable over Akeley and Rich.

We dismiss Patent Owner's Motion to Exclude as moot.

We grant Patent Owner's Motion for entry of a protective order and to seal Exhibit 2039.

ORDER

In consideration of the above, it is
ORDERED that claims 1–9 of the '506 patent are unpatentable;
FURTHER ORDERED that Patent Owner's Motion to Amend is
DENIED;

FURTHER ORDERED that Patent Owner's Motion for entry of a
protective order and to seal Exhibit 2039 is GRANTED; and

FURTHER ORDERED, that because this is a final written decision,
parties to the proceeding seeking judicial review of the decision must
comply with the notice and service requirements of 37 C.F.R. § 90.2.

IPR2018-00101
Patent 7,633,506 B1

PETITIONER

Kevin Anderson
k panderson@duanemorris.com

Scott Felder
sfelder@wileyrein.com

PATENT OWNER

William Meunier
Michael Renaud
Adam Rizk
MINTZ LEVIN
wameunier@mintz.com
mtrenaud@mintz.com
arizk@mintz.com