

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

REALTEK SEMICONDUCTOR CORP.,

Petitioner

v.

ATI TECHNOLOGIES ULC

Patent Owner

Case No.: IPR2023-00564

Patent 7,742,053

**DECLARATION OF DR. MASSOUND PEDRAM IN SUPPORT OF
PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO.
7,742,053**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	8
II. QUALIFICATIONS AND PERSONAL EXPERIENCE	9
III. LEGAL PRINCIPLES	13
IV. OVERVIEW OF THE '053 PATENT	19
A. Technology Background	19
B. File History	23
V. PERSON OF ORDINARY SKILL IN THE ART	26
VI. CLAIM CONSTRUCTION	27
A. "Arbiter"	27
B. "Select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads"	27
C. "Relative priorities"	28
D. "Command processing engine"	28
E. "Command threads"	29
F. "Memory device"	29
G. "Coupled to"	29
VII. GROUND 1: STUTTARD AND THE ADMITTED PRIOR ART OF THE '053 PATENT RENDER OBVIOUS CLAIMS 1-9	30
A. Overview of Stuttard	30
B. Independent Claim 1	34
1. 1[preamble]: A graphics processing system comprising:	34
2. 1[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads; and	36

TABLE OF CONTENTS

(continued)

	Page
3. 1[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads based on relative priorities of the plurality of pixel command threads and the plurality of vertex command threads.	43
C. Dependent Claim 2: The graphics processing system of claim 1, further comprising: a command processing engine, coupled to the arbiter, wherein the arbiter is further operable to provide the command thread to the command processing engine.	44
D. Dependent Claim 3: The graphics processing system of claim 1, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.	45
E. Dependent Claim 4: The graphics processing system of claim 1, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.	46
F. Independent Claim 5 of the '053 Patent.....	46
1. 5[preamble]: A graphics processing system comprising:.....	46
2. 5[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads;.....	47
3. 5[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads; and	47
4. 5[c]: a plurality of command processing engines, coupled to the arbiter, each operable to receive and process the command thread.....	47
G. Dependent Claim 6: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one arithmetic logic unit.	48

TABLE OF CONTENTS

(continued)

	Page
H. Dependent Claim 7: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one texture processing engine.....	50
I. Dependent Claim 8: The graphics processing system of claim 5, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.....	50
J. Dependent Claim 9: The graphics processing system of claim 5, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.	51
VIII. GROUND 2: STUTTARD AND WILLIAMS RENDER OBVIOUS CLAIMS 1-9.....	51
A. Overview of Williams	51
B. Independent Claim 1	54
1. 1[preamble]: A graphics processing system comprising:.....	54
2. 1[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads; and.....	54
3. 1[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads based on relative priorities of the plurality of pixel command threads and the plurality of vertex command threads.	60
C. Dependent Claim 2: The graphics processing system of claim 1, further comprising: a command processing engine, coupled to the arbiter, wherein the arbiter is further operable to provide the command thread to the command processing engine.	61
D. Dependent Claim 3: The graphics processing system of claim 1, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.....	61

TABLE OF CONTENTS

(continued)

	Page
E. Dependent Claim 4: The graphics processing system of claim 1, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.	62
F. Independent Claim 5 of the '053 Patent.....	62
1. 5[preamble]: A graphics processing system comprising:.....	62
2. 5[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads;.....	62
3. 5[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads; and	62
4. 5[c]: a plurality of command processing engines, coupled to the arbiter, each operable to receive and process the command thread.....	63
G. Dependent Claim 6: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one arithmetic logic unit.	63
H. Dependent Claim 7: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one texture processing engine.....	63
I. Dependent Claim 8: The graphics processing system of claim 5, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.....	63
J. Dependent Claim 9: The graphics processing system of claim 5, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.	64
IX. GROUND 3: WILLIAMS AND WHITTAKER RENDER OBVIOUS CLAIMS 1-9.....	64
A. Overview of Whittaker.....	64

TABLE OF CONTENTS

(continued)

	Page
B. Independent Claim 1	66
1. 1[preamble]: A graphics processing system comprising:.....	66
2. 1[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads; and.....	68
3. 1[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads based on relative priorities of the plurality of pixel command threads and the plurality of vertex command threads.	68
C. Dependent Claim 2: The graphics processing system of claim 1, further comprising: a command processing engine, coupled to the arbiter, wherein the arbiter is further operable to provide the command thread to the command processing engine.	73
D. Dependent Claim 3: The graphics processing system of claim 1, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.....	74
E. Dependent Claim 4: The graphics processing system of claim 1, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.	74
F. Independent Claim 5 of the '053 Patent.....	75
1. 5[preamble]: A graphics processing system comprising:.....	75
2. 5[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads;.....	76
3. 5[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads; and	76

TABLE OF CONTENTS
(continued)

	Page
4. 5[c]: a plurality of command processing engines, coupled to the arbiter, each operable to receive and process the command thread.....	76
G. Dependent Claim 6: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one arithmetic logic unit.	76
H. Dependent Claim 7: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one texture processing engine.....	77
I. Dependent Claim 8: The graphics processing system of claim 5, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.....	78
J. Dependent Claim 9: The graphics processing system of claim 5, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.	78
X. DECLARATION	78

I, Dr. Massoud Pedram, declare as follows:

I. INTRODUCTION

1. I have been engaged by Realtek Semiconductor Corp. (“Petitioner”) to provide my expert opinion on invalidity related issues in connection with Petitioner’s Petition for *Inter Partes* Review in IPR2023-00564 (“Petition”) before the Patent Trial and Appeal Board (“Board”). I was asked to provide my opinions in this expert declaration (“Declaration”). I intend to provide testimony regarding my opinions if called to testify at a deposition.

2. This Declaration addresses my opinions related to the validity of the Asserted Claims of U.S. Patent Nos. 7,742,053 (the “’053 patent”) (Exhibit 1001). I understand the challenged claims to be claims 1-9 of the ’053 patent. Accordingly, my Declaration addresses these claims.

3. I am being compensated for my time working in connection with this Petition. However, my compensation is not affected by or contingent upon the substance of my opinions or on the ultimate outcome of this Petition, and I have no other interest in this Petition.

4. The opinions expressed in this Declaration are my own and are based on my personal knowledge, my education, experience, and training, and on my understanding of the information and documents referenced in this Declaration. If called as an expert in this case, I anticipate that my testimony will concern the

matters addressed in this Declaration. My testimony may be affected by the positions that ATI Technologies ULC (“Patent Owner”) and/or its expert takes on these and other related topics after this Declaration is issued, either in the form of expert declarations or deposition testimony, or any ruling that the Board may make, and therefore I reserve the right to provide testimony regarding those issues.

II. QUALIFICATIONS AND PERSONAL EXPERIENCE

5. I am a Professor of Electrical Engineering and the Charles Lee Powell Chair in Electrical Engineering and Computer Science in the Viterbi School of Engineering at the University of Southern California (“USC”).

6. I am an expert on Very Large Scale Integration (“VLSI”) design including parallel computing, distributed processing, high-speed design, power-efficient design, digital integrated circuits, and semiconductor memory. I also have expertise in neural networks and graphic processing units (“GPUs”), as well as system-on-chip designs targeting applications ranging from high-performance computing to low-power embedded computing. My current research interests are in, among others, computer-aided design methodologies and tools for CMOS circuits and systems, energy-efficient computing, computer architecture and hardware, parallel and distributed computing, neural network acceleration, and integrated circuit and system design including system-on-chip design.

7. I earned my B.S. in Electrical Engineering from the California Institute of Technology in 1986, my M.S. and Ph.D. in Electrical Engineering and Computer Sciences from the University of California at Berkeley in 1989 and 1991, respectively.

8. I was a Graduate Student Researcher in the Department of Electrical Engineering and Computer Sciences at the University of California at Berkeley from 1986 to 1991 during my graduate studies. Concurrently I had a part-time research position at the Xerox Palo Alto Research Center from 1987 to 1989. After receiving my doctorate, I joined the Department of Electrical Engineering at the University of Southern California as an Assistant Professor from 1991 to 1996. I became an Associate Professor and a Professor in 1996 and 2000, respectively. In 2013, I was named the Stephen and Etta Varra Professor in the school, which had been renamed the Viterbi School of Engineering. From 2017 to the present, I am the Charles Lee Powell Chair in Electrical Engineering and Computer Science in the Viterbi School of Engineering at USC.

9. I am a named inventor on twelve U.S. patents and two pending U.S. patent applications. These patents relate to topics ranging from pipelined architecture for logic circuits to novel address bus encoding techniques, and from charge recycling based multi-threshold CMOS design to low-leakage SRAMs. Two recently filed patent applications focus on hybrid arithmetic and logic processing of

neural networks and high-performance and scalable Montgomery modular multiplier circuits.

10. I have (co-)published five books and more than 800 archival and conference papers and book chapters on topic ranging from computer-aided design techniques and tools targeting VLSI circuit and system design to FPGA fabric design and synthesis, from low-power electronics to power distribution, conversion, and regulation.

11. I have been a member of multiple technical and scientific advisory boards for high-tech companies, including EPIC Design, Magma Design Automation, and Atrenta Inc., all of which went on to have successful initial public offerings or were acquired.

12. I have served as the (lead) principal investigator on many U.S. government (including the National Science Foundation, the Defense Advanced Research Projects Agency, the Department of Defense, and the Intelligence Advanced Research Project Activity) and private industry (including the Semiconductor Research Corporation) research projects, for some of which I have been the lead investigator of a team comprised of tens of faculty, students, and industry collaborators.

13. I was a recipient of the 1996 Presidential Early Career Award for Scientists and Engineers, and in 2000 I was selected a Fellow of the IEEE for

contributions to the theory and practice of low-power design and CAD. In 2008, I was named as an ACM Distinguished Scientist. I received the 2015 IEEE Circuits and Systems Society Charles A Desoer Technical Achievement Award for my contributions to modeling and design of low power VLSI circuits and systems, and energy-efficient computing.

14. I was recognized as the Third Most Cited Author at the 50th anniversary of the Design Automation Conference, Austin, Texas, in June 2013. I received a Top Three Author Award at the 20th Anniversary Asia and South Pacific Design Automation Conference, Chiba/Tokyo, Japan, in January 2015, and was named the Second Most Prolific and Second Most Cited Author Awards from the 20th Anniversary International Symposium on Low Power Electronics and Design, Rome, Italy, in July 2015. I have received multiple Best Paper Awards. I have also served on best or outstanding paper award committees.

15. I received the USC Viterbi School of Engineering Senior Research Award in 2017. I was named a fellow of the American Association for the Advancement of Science in early 2023.

16. Since July 2012, I have been the Executive Committee Chair of the IEEE/ACM International Symposium on Low Power Electronics and Design, a technical conference which I co-founded in 1996.

17. I served as the Vice President of Publications and Chair of the Distinguished Lecturer Program of the IEEE Circuits and Systems Society from 2005 to 2006 and 2003 to 2004, respectively. I served as a member of the Board of Governors of the IEEE Circuits and Systems Society from 2000 to 2003. I served as the Technical Program Chair of the International Symposium on Low Power Electronics and Design, Monterey, California, in August 1996. In addition, I have served on multiple advisory boards and chaired multiple conferences and symposiums.

18. I was the Editor-in-Chief of the ACM Transactions on Design Automation of Electronic Systems from June 2008 to May 2014. I was also the Inaugural Editor-in-Chief of the IEEE Journal on Emerging and Selected Topics in Circuits and Systems from January 2010 to December 2013. I have served as the editor or an associate editor for multiple publications.

19. My CV is attached as Exhibit 1008.

III. LEGAL PRINCIPLES

20. I am not a lawyer and offer no opinion of the law. Accordingly, the “Legal Principles” herein have been provided to me from counsel, but I have read and adopted it as my own. For example, I have, however, been informed by counsel of the legal principles that apply with respect to the issues for which I was asked to provide an opinion, including patent invalidity. I have applied these legal principles

in formulating my opinions and arriving at my conclusions. I have also been informed that the claims of an issued patent are presumed valid, but those claims can be found invalid if it can be proven, by clear and convincing evidence, that the claims fail to comply with one or more requirements of patentability.

21. It is my understanding that a patent may claim the benefit of the filing date of an earlier application only if the earlier application provides adequate disclosure of the patent's claims. I further understand that to satisfy the disclosure requirement, the disclosure of the earlier filed application must describe the later claimed invention in sufficient detail that one skilled in the art, as of the earlier filing date sought, could clearly conclude that the inventor invented the claimed subject matter. I understand that while the earlier application need not describe the claimed subject matter in precisely the same terms as found in the claims at issue, the prior application must convey with reasonable clarity to a person of ordinary skill in the art that, as of the filing date sought, the inventor was in possession of the invention. It is also my understanding that claims of a patent that contain new matter that is not adequately disclosed by an earlier filed application do not receive the benefit of the earlier filing date.

22. It is my understanding that a patent claim is invalid if the differences between the patented subject matter and the prior art are such that the subject matter as a whole would have been obvious at the time of the invention to a person of

ordinary skill in the art. I understand that the following factors must be evaluated to determine whether the claimed invention is obvious: (1) the scope and content of the prior art, (2) the difference or differences between each claim of the patent and the prior art, (3) the level of ordinary skill in the art at the time the invention of the patent was made, and (4) any secondary considerations or objective indicia of nonobviousness.

23. I understand that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. In other words, when a claim simply arranges prior-art elements with each performing the same function it had been known to perform and yields no more than one would expect from such an arrangement, then such a combination is obvious. Moreover, when a patent claims a structure already known in the prior art that is altered by the mere substitution of one element for another known in the field, the combination is likely to be obvious unless the combination yields an unpredictable result.

24. I understand that a patentee's admissions about the scope and content of the prior art provide a factual foundation as to what a skilled artisan would have known at the time of invention. For example, by characterizing claimed elements as conventional in the specification, the patentee effectively admits that such things would be known to a person of ordinary skill in the art. I understand that I can rely on

such admissions and/or knowledge in forming an opinion as to whether claims are obvious. Such reliance could include, for example, furnishing a motivation to combine, or supplying a missing claim limitation. Thus, even though a patentee's admissions are not themselves prior art references, they can establish the background knowledge possessed by a person of ordinary skill in the art.

25. I have been informed that when a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill in the art can implement a predictable variation, such a variation is likely unpatentable. For the same reason, if a technique has been used to improve one device, and one of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond the skill of the person of ordinary skill in the art. One question to consider is whether the improvement is more than the predictable use of prior-art elements according to their established functions.

26. I have been informed that it may often be necessary, in a validity analysis, to look to interrelated teachings of multiple patents or other publications or pieces of prior art; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by one of ordinary

skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.

27. I have been informed that a validity analysis need not seek out precise teachings directed to the specific subject matter of the challenged claim. Rather, it is appropriate to take account of the inferences and creative steps that a person of ordinary skill in the art would employ. I understand that a person of ordinary skill in the art is also a person of ordinary creativity.

28. I understand that a claim composed of several elements is not proved obvious merely by demonstrating that each element was, independently, known in the prior art. I understand that it can be important to identify a reason that would have prompted a person of ordinary skill in the art in the relevant field to combine the elements in the way the claimed invention does. Therefore, I am informed that, in determining whether a claimed combination is obvious, the correct question is whether the combination was obvious to a person of ordinary skill in the art at the time of the alleged invention. I am told that one of the ways in which subject matter can be proved obvious is by noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims. I understand that any need or problem known in the field of endeavor at the time of alleged invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.

29. I understand that one should not assume that a person of ordinary skill in the art attempting to solve a problem will be led only to those elements of prior art designed to solve the same problem. Instead, I understand that since familiar items may have obvious uses beyond their primary purposes, in many cases a person of ordinary skill in the art would be able to fit the teachings of multiple prior art references together like pieces of a puzzle.

30. I understand that, when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill in the art has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, then it is likely the product is not of innovation but of ordinary skill and common sense. I understand that, in such an instance, the fact that the combination was obvious to try may show that the combination is obvious.

31. I understand that, when determining whether a claimed combination is obvious, the correct analysis is not whether a person of ordinary skill in the art, writing on a blank slate, would have chosen the particular combination of elements described in the claim. Instead, I understand the correct analysis considers whether one of ordinary skill, facing the wide range of needs created by developments in the field of endeavor, would have seen a benefit to selecting the combination claimed.

32. I understand that, when determining whether a claimed invention is obvious, any “secondary considerations” of nonobviousness should also be considered. I further understand that there must be a nexus between any alleged secondary considerations and the patent claims. In particular, when the product that practices an invention is not co-extensive with the patented invention—for example, if the patented invention is only one component of a commercially successful product—then the patentee must show a sufficient relationship or nexus between the patented invention in a product and the secondary considerations, *e.g.*, commercial success of the product. In addition, I understand that the burden of introducing evidence of secondary considerations generally is on the Patent Owner.

IV. OVERVIEW OF THE '053 PATENT

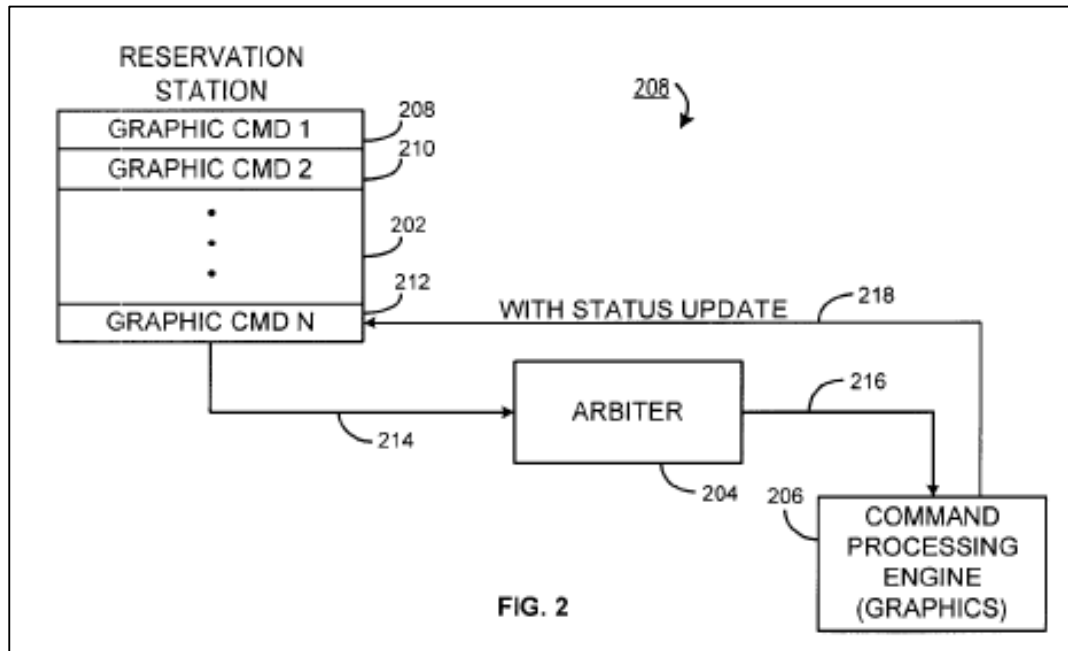
A. Technology Background

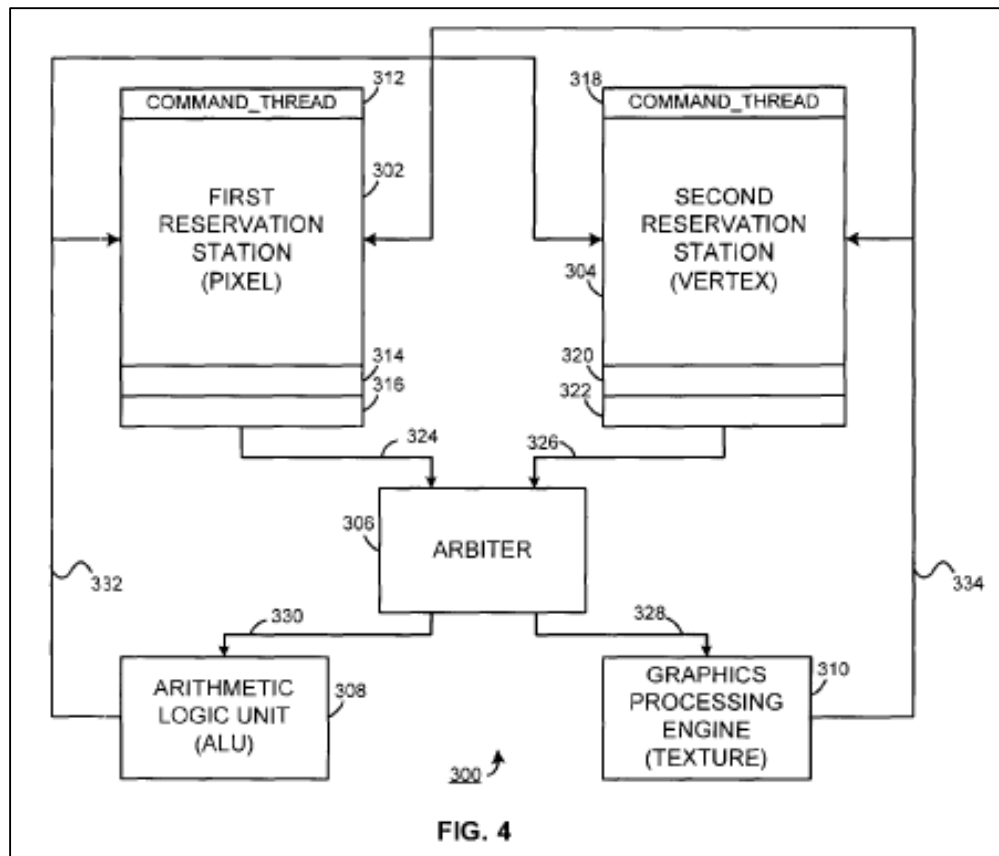
33. Computing devices often utilize a graphics processing unit (GPU) to accelerate the rendering of graphics data for display. GPUs execute a graphics processing pipeline that includes a plurality of processing units that operate together to execute graphics processing commands. Initially, GPUs included a fixed function graphics processing pipeline where each processing stage in the pipeline was implemented with fixed function. More recently, however, graphics processing pipelines have shifted to a programmable architecture where one or more processing stages in the graphics pipeline are programmable and implemented with one or more

programmable shader units. Each of the programmable shader units may be configured to execute a shader program such as a pixel (fragment) shader or a vertex (geometry) shader. Vertex shader programs are run once for each vertex in a scene or object and are responsible for how the vertex will be displayed on a 2-D display device. For example, vertex processing occurs when vertices of 3D images are transformed from one coordinate system to a screen coordinate system, and pixel processing occurs when texture mapping and colors are applied to the pixels that have been processed from the vertices. Pixel shader programs are run once for every pixel in every triangle in a scene or object and are responsible for how individual pixels will be displayed. Vertex shader programs can manipulate properties such as the position, color, texture coordinates, and lighting characteristics of a vertex. Pixel shader programs can manipulate various attributes of individual pixels such as color and translucency characteristics. A user application specifies the shader program to be executed by the programmable processing stages in a programmable graphics pipeline, thereby providing a high degree of resource reuse and flexibility.

34. In a typical graphics processing system, graphical data such as vertices and pixels are processed through multiple steps providing for the application of geometric transformations, lighting, textures, and other processing instructions. In a typical multithreaded processing system, steps for performing various graphics tasks can be broken down into and controlled by a plurality of command threads. The

alleged invention of the '053 Patent is a graphics processing system including a memory device, an arbiter, and a command processing engine. See, for example, Figures 2 and 4 of the '053 patent.





35. The memory device of the system includes a reservation station for storing a plurality of graphic command threads (208-212). Ex. 1001 ('053 patent) at 2:67-3:2. An arbiter 204 is connected to the reservation station to retrieve the command threads (208-212) from the reservation station and provide them to command processing engine 206. *Id.* at 3:08-3:11. Arbiter 204 may retrieve the command threads based on a priority scheme. *Id.* at 3:29-30. After the command processing engine 206 executes the thread command, it provides a status update 218 to the corresponding command thread in the reservation station. *Id.* at 3:12-15. Like the disclosure in the Background of the Invention section, the '053 Patent discloses two reservation stations (302, 304) one for storing command threads for pixel-based

command threads and the other for vertex-based command threads, respectively. *Id.* at 3:65-4:4. An input arbiter is provided to selectively output command threads to each of the two reservation stations (302, 304) based on whether the thread is a pixel or vertex command thread. *Id.* at 4:5-10. From each of the reservation stations, arbiter 306 selects one thread for the graphics processing engine 310 and one thread for the ALU 308. *Id.* at 4:17-21. Upon execution of the associated command of the command thread, the thread is returned to the same storage location in the reservation stations (302, 304) with its status updated. *Id.* at 4:23-27.

B. File History

36. I have reviewed the prosecution history of the '053 patent and provide a summary here. Again, I am not an attorney, but have reviewed the file history and understand that arguments and amendments made to claims during prosecution can affect claim scope and/or can provide context to a person of ordinary skill in the art as to the meaning of claim terms.

37. I understand that a patent application was filed on May 9, 2007. *See* Ex. 1001 at 1. The filed application included a set of claims numbered 1-19. Ex. 1002 at 16-20. Prior to any rejections or allowance of the proposed claims, the Examiner issued a Requirement for Restriction/Election of the claims into two groups. *Id.* at 57-59. I understand this means that the examiner concluded that there were two inventions being claimed and was asking the Applicant to choose which

invention they wanted to pursue in this application. Applicants elected Group II, which included the claims then numbered 11-19. *Id.* at 66-72.

38. On September 9, 2009, the Examiner issued what was entitled a Non-Final Office Action rejecting all elected claims (*i.e.*, those numbered 11-19) as invalid based on U.S. Patent No. 6,650,327 to Airey. *Id.* at 76-82. In response to this Office Action, Applicants argued that Airey failed to disclose any active select function as required by claims 11-19 and therefore Airey did not disclose or teach all of the features of claims 11-19. *Id.* at 107-110. Applicant did not amend any of the pending claims 11-19. *Id.* at 102-106.

39. On February 18, 2010, the Examiner issued a Notice of Allowability indicating claims 11-19 were allowed. *Id.* at 117. In the Notice of Allowability, the Examiner indicated the following as Reasons for Allowance:

Reasons for Allowance

3. The following is an examiner's statement of reasons for allowance:

The prior art taken singly or in combination does not teach or suggest, a graphics processing system, among other things, comprising:

an arbiter, coupled to the at least one memory device, operable to *select a command thread* from either of the plurality of pixel command threads and the plurality of vertex command threads *based on relative priorities* of the plurality of pixel command threads and the plurality of vertex command threads.

The closest prior art, Airey et al. (U.S. Patent No. 6,650,327) fails to teach the above features.

Id. at p. 118. On June 22, 2010, the '053 patent was issued, which included the allowed claims 11-19, now renumbered as claims 1-9.

40. An Inter Partes Review of the '053 was conducted in which the PTAB held that claims 1, 2, and 5-7 of the '053 Patent were unpatentable on the following grounds: (1) claims 5-7 as anticipated by U.S. Patent No. 6,897,871 (*Moreton*); (2) claims 1 and 2 as obvious over *Moreton* combined with U.S. Patent No. 5,968,167 (*Whittaker*); claims 1, 2 and 4-7 as obvious over U.S. Patent No. 7,015,913 (*Lindholm*) in view of Admitted Prior Art; and (4) claims 1, 2, and 5-7 as obvious over U.S. Patent No. 7,363,472 (*Stuttard*) in view of Admitted Prior Art. *Id.* at p. 282. However, the decision was appealed, and the Federal Circuit reversed, finding that *Moreton*, *Lindholm*, and *Stuttard* did not qualify as prior art. *Id.* at 323. My

understanding is that the Federal Circuit decision was based on the ability of the patentee to show that they had a date of invention prior to the U.S. filing dates of at least one prior art reference from each ground asserted during the IPR, and that the patentee was diligent in reducing to practice its invention. I further understand that the Federal Circuit decision effectively meant that those affected prior art references could no longer be considered eligible prior art references to the invention. The Federal Circuit decision did not address any of the substantive issues related to whether or not those references disclosed and/or rendered obvious the claimed subject matter.

V. PERSON OF ORDINARY SKILL IN THE ART

41. I understand that a person of ordinary skill in the art is a fictional or hypothetical person who is presumed to have known the relevant art at the time of the invention. I understand that the International Trade Commission (“ITC”) has adopted the following definition of a person of ordinary skill in the art (“POSITA”) in a related matter: “a POSITA would have a bachelor degree in electrical engineering, computer engineering, computer science, or a related field, and at least two years of experience in ‘parallel computing or graphics processing including developing, designing, or programming software or hardware for graphics processing or parallel processing units, graphics accelerators[,] or other graphics processing systems,’ with additional education substitution for experience and vice

versa.” Order No. 31 at 9 (alteration in original) (Exhibit 1010 or “*Markman* Order”).

I have applied the ITC’s definition for purposes of this Declaration.

VI. CLAIM CONSTRUCTION

42. I understand that the Board gives claim terms their ordinary and customary meaning as a POSITA would understand them in light of the patent’s specification and prosecution history. I also understand that several of the claim terms have already been construed by the ITC and by the Board in decisions regarding the ’053 patent. For the purposes of this Declaration, I have applied the narrower of the available constructions.

A. “Arbiter”

43. I understand that the Board has construed this term to mean “any computer hardware, software, or combination thereof that receives and provides a command thread.” Ex. 1011 at 7. I also understand that the ITC found that this term should be given its plain and ordinary meaning. Ex. 1010 at 17. Based on my understanding of the plain and ordinary meaning of this term, the Board’s previous construction is the narrower of the two constructions.

B. “Select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads”

44. I understand that the ITC found that this term should be given its plain and ordinary meaning as well, but it found that this term does not require an arbiter to select a command thread based on state bits associated with each command thread.

Ex. 1010 at 16. I did not consider any further limitations to this term for the purposes of this Declaration. AMD's expert in the ITC matter, argued that "[a]lthough the ALJ does not explicitly construe the term 'arbiter' to include this requirement," this limitation nevertheless means that the arbiter must select from at least four threads, two (or more) pixel threads and two (or more) vertex threads. Such a construction should be rejected because it directly contradicts the plain language of the claims, rendering the phrase "from either of" completely meaningless. Regardless, as the Board previously acknowledged, the Admitted Prior Art describes an arbiter that was known in the art to select a command thread between different types of command threads. Ex. 1011 at 99. Thus, this limitation would have been obvious even under AMD's erroneous construction.

C. "Relative priorities"

45. I understand that the ITC also found that this term should be given its plain and ordinary meaning, so I did not consider any unique limitations to this term for the purposes of this Declaration. *See* Ex. 1010 at 16-17.

D. "Command processing engine"

46. I understand that the Board has construed this term "as requiring each command processing engine be coupled to an arbiter and operable to receive and process a command thread selected by the arbiter." Ex. 1011 at 12. I also understand that the ITC deferred construction on this term in an ongoing proceeding. Ex. 1010

at 17-19. I have been informed that any construction the ITC does adopt will likely be broader than the Board's previous construction. For the purposes of this Declaration, I have used the Board's narrower construction.

E. "Command threads"

47. I understand that the Board has construed this term to "encompass a stream of instructions or a process that is part of a larger process or program." Ex. 1011 at 6. I also understand that the ITC has construed this term to mean a "sequence of commands applicable to the corresponding element" that does not require the threads to comprise state bits. Ex. 1010 at 19-20. These constructions are broad enough to not affect my conclusions in this Declaration.

F. "Memory device"

48. I understand that the ITC also found that this term should be given its plain and ordinary meaning. Ex. 1010 at 20-21. I did not consider any unique limitations to this term for the purposes of this Declaration.

G. "Coupled to"

49. I understand that the ITC also found that this term should be given its plain and ordinary meaning, but it did find that this term does not mean directly connected. Ex. 1010 at 21-22. I did not consider any further limitations to this term for the purposes of this Declaration.

VII. GROUND 1: STUTTARD AND THE ADMITTED PRIOR ART OF THE '053 PATENT RENDER OBVIOUS CLAIMS 1-9

A. Overview of Stuttard

50. I understand that PCT Publication No. WO00/62182A2 was published on October 19, 2000. *See* Ex. 1005 at 1. I refer to this publication as “Stuttard” throughout this Declaration. I have been informed that its publication date makes it eligible as prior art to the '053 patent.

51. Stuttard discloses a data processing apparatus for producing graphics images for display on a screen, comprising a SIMD (single instruction multiple data) array of processing elements, wherein the processing elements are operably divided into a plurality of processing blocks, the processing blocks being operable to process respective groups of data items. *Id.* at 2:17-23 and 4:3-9. By organizing its processing elements into a plurality of processing blocks, Stuttard system overcomes the inefficiency associated with conventional SIMD architectures when a system has to process a large number of relatively small data item groups (each data item group is associated with an instruction stream). This is because a SIMD array can only process one (small) data item group at a time, potentially leaving a large proportion of the processing elements in the array unused while data is being processed for that group. *Id.* at 1:24-2:10. With multiple SIMD processing blocks, each processing block is operable to process respective groups of data items. *Id.* at 2:25-33. In other words, Stuttard's system can simultaneously process multiple instruction streams

with each stream executing on one processing block. Stuttard discloses that the elements in its data processing apparatus are controlled to execute series of instructions in an instruction stream, and such instruction streams for the processing blocks are known as “threads.” *Id.* at 10:7-10. Stuttard explains that by supporting the concurrent execution of multiple threads in a data processing architecture, the processing element array can be kept active, by processing another thread when the current active thread is halted. *Id.* at 10:19-31. Stuttard explicitly discloses threads are assigned priorities relative to one another and that they may perform various 2D or 3D graphics processing activities. In particular, at any particular moment in time, each thread may be assigned to any one of a number of tasks. For example, thread zero may be assigned for general system control, thread 1 assigned to execute 2D (two dimensional) activities, and threads 2 to 7 assigned to executing 3D activities (such as calculating vertices, primitives or rastering). *Id.* at 11:16-25. In addition, Stuttard discloses that the graphics data processing includes performing both vertex and pixel processing instructions. *Id.* at 20:23-21:7, 22:12-32. Therefore, Stuttard’s data processing apparatus is a multithreaded graphics system, performing vertex and pixel processing tasks, including geometric transformations, lighting, shading, texture mapping, and coordinate calculations for graphics objects and shapes to be displayed on a screen. *Id.* at 1:9-22.

52. Stuttard's multithreaded graphics system includes a memory device, an arbiter, and a (command) processing core 10. This processing core, which is connected to the secondary buses 6 for the transfer of data, and to the primary bus 4 for the transfer of instructions, provides the graphics processing capability of the system. The processing core 10 includes a number of control units: thread manager 102, array controller 104, channel controller 108, a binning unit 1069 per block and a microcode store 105. These control units control the operation of processing blocks 106 of the graphics processing system. *Id.* at 7:25-8:7. *See also Id.* at 8:9-13 and Fig. 2 and 3. The thread manager 102 provides instructions to processing blocks 106 through array controller 104 and channel controller 108. *Id.* at 8:17-9:9. Processing block 106 includes processor units 1061a for executing the instructions. *Id.* at 9:11-20. Thread manager 102 manages instruction streams or threads to increase efficiency and reduce response time. *Id.* at 10:7-11:1. Stuttard's cache memory unit 1024 located inside of the thread manager stores both pixel and vertex threads. *Id.* at 11:16-25, 20:23-21:7, and 22:12-32.

53. As shown in Fig. 4 of Stuttard, thread manager 102 includes thread scheduler 1025 and thread processors 1026, as well as cache memory unit 1024 for storing instructions for each thread—having one buffer per thread. *Id.* at 11:3-10. The threads are assigned priorities relative to one another. *Id.* at 11:16-17. Stuttard also teaches that the status block 1030 within the thread manager 102 receives status

information 1036 from each of the threads. The status information is subsequently transferred to the thread scheduler 1025, which uses the status information to determine which thread should be active at any one time. *Id.* at 12:12-18. Thread processors 1026 control the issuance of instructions from the thread manager to maintain processing of simultaneously active program threads. *Id.* at 11:27-33. Stuttard's thread scheduler, working in conjunction with the thread processor, in the thread manager, arbitrates and schedules a thread based on their relative priorities and (READY vs. YIELDING) status information about each thread. *Id.* at 12:12-18, 23:35-26:32. The thread manager is coupled to the processing blocks through the controllers. *Id.* at 8:17-27, Fig. 3. The instructions for the thread are sent to processor blocks via the controllers. *Id.* at 8:22-24. The processing blocks execute the instructions received from the controllers. *Id.* at 9:11-20. Stuttard also discloses that the cache unit 1024 within the thread manager 102 could be replaced by a series of FIFO buffers, one per thread. *Id.* at 11:3-10. Fig. 9 of Stuttard depicts a flowchart illustrating data processing using Stuttard's system. In this flowchart which concerns processing and displaying primitive graphical images, the data is prepared as vertices by the host system and then loaded into the processing elements of the Stuttard's system, with each processing element in a processing block representing a vertex of a primitive (e.g., a line segment or a triangle). After some vertex processing operations, each PE transfers its data concerning its primitive to the

mathematical expression evaluator (MEE) for processing into pixel data and identifying which pixels occur inside the primitive. In this step, each PE includes data about a respective pixel i.e., data is stored on a pixel per pixel basis. *Id.* at 20:23-22-33. Evidently, since there are eight processing blocks in Stuttard's system (*Id.* at Fig. 2(b) and Fig. 4), each of which can execute one thread, a processing block may start the vertex processing for a newly arrived primitive while a different block is still finishing up the pixel processing for a previous primitive.

54. Stuttard discloses various embodiments and/or configurations of elements and/or features. A person of ordinary skill in the art would have understood that the concepts, components and/or features apply equally across each embodiment. That is, a person of ordinary skill in the art would not have understood them to be mutually exclusive and they could be "mixed and matched" within the spirit of the disclosure. Moreover, a person of ordinary skill in the art would have been motivated to mix and match them as they all are disclosed in the same reference and are directed to the same field of invention and solving the same problems.

B. Independent Claim 1

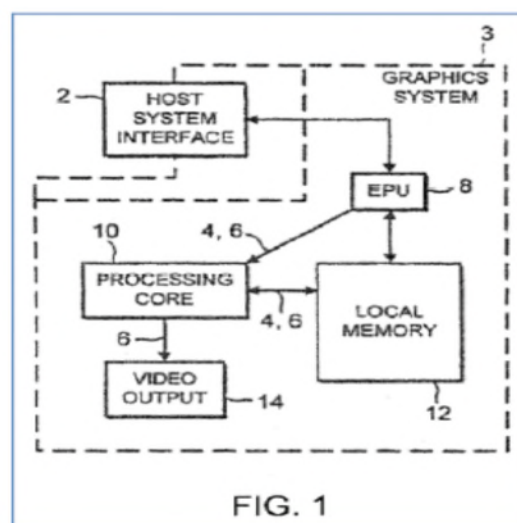
1. 1[preamble]: A graphics processing system comprising:

55. I have been informed that the preamble may not be limiting. To the extent it is limiting, it is my opinion that Stuttard discloses a graphics processing system. As I explained above, Stuttard discloses a data processing apparatus for

producing graphics images for display on a screen, comprising a SIMD (single instruction multiple data) array of processing elements, wherein the processing elements are operably divided into a plurality of processing blocks, the processing blocks being operable to process respective groups of data items. Ex. 1005 at 2:17-23 and 4:3-9. A person of ordinary skill would understand Stuttard's data processing apparatus for producing graphics images for display on a screen to be a graphics processing system.

56. Indeed, Stuttard explicitly states that it discloses a graphics processing system. For example, Stuttard discloses that its system is a “graphics processing method” *Id.* at 4:15-17, “a graphics data processing system 3” (*Id.* at 4:21-22), and/or a “graphics processing system” *Id.* at 5:8-9.

57. Stuttard further illustrates its graphics (data processing) system that includes a processing core to allow for processing of the graphical data:



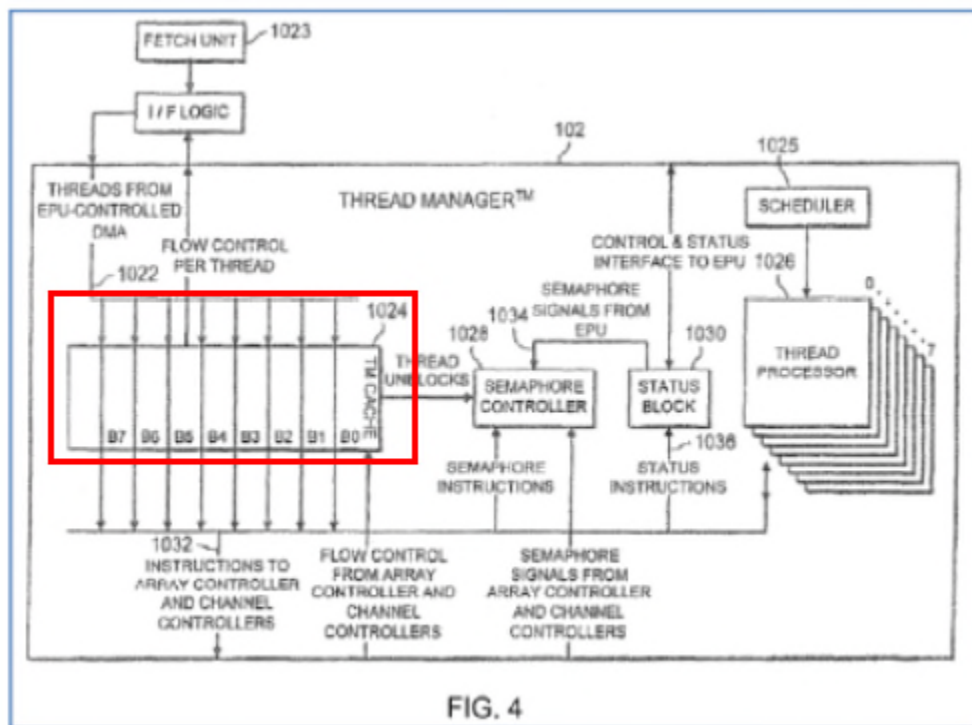
Id. at Fig. 1.

58. As depicted in Figure 1, Stuttard discloses a graphics system 3 that includes an embedded processor unit (EPU 8) that controls the overall function of the graphics system and how it interfaces with the host system, such as a personal computer or workstation. *Id.* at 4:22-30. Figure 1 further depicts a processing core 10 for processing graphical data received in the graphics system for output to a display screen. *Id.* at 4:30-33. This processing core 10 receives vertex data and pixel data at a plurality of processing blocks 106 included in the processing core 10, and these processing blocks 106 perform geometry and pixel processing, respectively. *Id.* at 8:17-31, Figure 3, 21:3-7, 22:12-32. These processing blocks 106 “perform the graphics processing itself.” *Id.* at 8:5-7. Finally, the graphics system includes a memory (local memory 12 in Figure 1 and cache unit 1024 in Figure 4) to store information as the graphics system completes its desired processing. *Id.* at 4:33-34, 11:12-14. In my opinion, these components and functions satisfy at least the requirements needed for a graphics processing system.

2. 1[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads; and

59. First, Stuttard discloses at least one memory device in the form of a cache (memory) unit 1024. *See, e.g., id.* at 11:12-14 and Figure 4. In my opinion, this cache unit 1024 is a memory device for at least the reasons explained below.

60. The processing blocks 106 that perform the graphics processing functions described above include a number of control units, including a thread manager 102, an array controller 104, and a channel controller 108. *See id.* at 8:1-7. The thread manager 102 receives control signals from the EPU 8 and gives instructions to the array controller 104 and the channel controller 108. *Id.* at 8:17-24. The structure of the thread manager 102 itself can be seen below with respect to Figure 4:



Id. at Fig. 4 (annotated).

61. As depicted in the annotated drawing, this thread manager 102 includes a cache unit 1024. *See id.* This cache unit stores a plurality of threads that are assigned to execute 2D and 3D activities (such as calculating vertices, primitives, or

rastering). *Id.* at 11:3-25. These “threads are assigned priorities relative to one another.” *Id.* at 11:16-17. This cache unit can be replaced by a series of first-in-first-out (FIFO) buffers, each for an individual thread. *Id.* at 11:3-14.

62. While Stuttard does not explicitly disclose that this cache unit 1024 stores a plurality of vertex command threads and a plurality of pixel command threads, in my opinion, it is clear that these threads are stored in the cache unit 1024. As discussed earlier, the processing blocks 106 that receive instructions from the thread manager 102 containing cache unit 1024 process both pixel and vertex data. *Id.* at 20:23-21:7, 22:4-32. As explained earlier, a processing block in Stuttard’s system may start vertex processing for some primitive while a different processing block is still doing pixel processing for another primitive. In this case, both vertex and pixel command threads corresponding to these two primitives will be present in the cache unit 1024 at the same time. As a result, it would have been obvious to a person of ordinary skill that the cache unit 1024 stores a plurality of vertex command threads and a plurality of pixel command threads.

63. Further, it would have been obvious to modify Stuttard to store the plurality of vertex command threads and the plurality of pixel command threads in different portions of the cache unit 1024. The thread manager 102 that stores the cache unit 1024 also includes a plurality of thread processors 1026 that “allow high level flow control to be performed for a thread ... and to assemble instructions to be

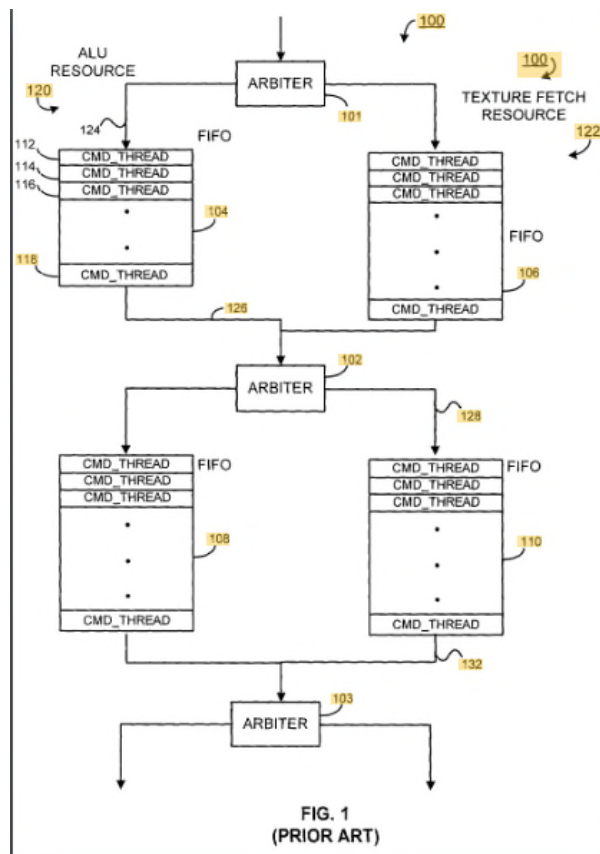
issued to the array controller and channel controller.” *Id.* at 26:3-7. Stuttard discloses that the thread processors can each control an individual thread to “reduce[] the need for context switching when changing the active thread, thereby reducing memory accesses and hence increasing the speed of operation.” *Id.* at 12:1-7. A person of ordinary skill would know that modifying Stuttard’s cache unit 1024 to separately store and process pixel and vertex threads would enhance these advantages of Stuttard’s multithreaded system by making it faster and more convenient to store and retrieve different types of command threads. A person of ordinary skill would have also considered it obvious to separately store pixel and vertex command threads since these command threads correspond to different types of graphics data processing tasks.

64. A person of ordinary skill would have also been aware of the above-described benefits for modifying Stuttard because it was very well-known at the time that pixel and vertex command threads could be stored in separate portions of the system’s memory device. *See, e.g.*, Ex. 1001 (’053 Patent) at 1:24-28, 1:42-64, and Fig. 1; Ex. 1012 (*Boyd*) at Abstract, 3:26-54, 3:58-4:10, and Figs. 1A and 1B; Ex. 1013 (*Lindholm*) at 3:57-64, 4:22-41, 5:50-68, 6:60-7:36; Ex. 1014 (*Selzer*) at pp. 37, 40-41, and Fig. 4.1. Because this design modification was well-known in the art, a person of ordinary skill would have had a reasonable expectation of success of modifying a standard memory device to store instructions corresponding to

command threads of different types in different portions of the memory. Further, because this design-option was so well-known within the art, it would have been routine for a person of ordinary skill to experiment with the memory device design in this way.

65. Further evidence that these elements are within the knowledge of one of ordinary skill in the art is that these elements are disclosed in the Admitted Prior Art.¹ The Admitted Prior Art recognizes that processing elements, such as the vertex and pixel command threads, are processed through multiple steps to apply textures and other processing instructions. Ex. 1001 at 1:24-28. The Admitted Prior Art further discloses buffers 104 and 108 that store ALU resource command threads, and buffers 106 and 110 that store texture fetch resource command threads. *Id.* at 1:42-64. A person of ordinary skill would recognize that these buffers 104 and 108 for storing ALU resource command threads and buffers 106 and 110 for storing texture fetch resource command threads may comprise first and second memory portions for storing a plurality of pixel command threads and a plurality of vertex command threads. These different memory portions are depicted in Figure 1:

¹ As described in the Legal Section in this Declaration, I understand that Admitted Prior Art, *e.g.*, a patentee's admissions, are evidence of the knowledge of the ordinary skill in the art at the time of the invention.



Id. at Fig. 1.

66. In my opinion, it would have been obvious to a person of ordinary skill to combine Stuttard's multithreaded system with the separate memory portions as known by a person of ordinary skill in the art at the time of the invention as taught in the Admitted Prior Art. Moreover, Stuttard and the Admitted Prior Art are both concerned with processing graphics. Ex. 1005 at 2:17-23 and 4:3-9; Ex. 1001 at 1:24-28. Further, storing the pixel and vertex command threads in separate memory portions was a well-known way of storing command threads, and it would have allowed faster access to command threads while simplifying storing threads in a multithreaded processor. Modifying Stuttard in view of the knowledge of a person

of ordinary skill in the art at the time of the invention, such as that taught in the Admitted Prior Art, would improve on the benefits of Stuttard's multithreaded system that reduces memory access and increases operation speed. See Ex. 1005 at 12:1-7. Such modifications would have been routine and yielded expected results. That is, a person of ordinary skill in the art would have had a reasonable, indeed high, expectation of success in adopting such techniques when practicing Stuttard.

67. In the ITC matter, AMD has argued that the memory device must store both pixel and vertex command threads "at the same time." Nothing in the claims require any such limitation. AMD's attempt to import this limitation is an effort to bolster their previously presented argument to the Board that *Stuttard* should be rejected because it performs operations in "discrete, sequential phases," i.e., processes only vertex threads followed by pixel threads, but never "at the same time." The Board previously rejected this argument. Ex. 1011 at 98-99. Regardless, the Admitted Prior Art discloses storing different types of command threads in different memory portions in a graphics processing system, rendering this limitation obvious. Ex. 1001, 1:42-46; Figure 1; Ex. 1011 at 95.

3. **1[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads based on relative priorities of the plurality of pixel command threads and the plurality of vertex command threads.**

68. Stuttard discloses a thread scheduler 1025 and thread processors 1026, which collectively are coupled to the cache unit and operable to select a pixel command thread or a vertex command thread based on relative priorities as required by this claim limitation.

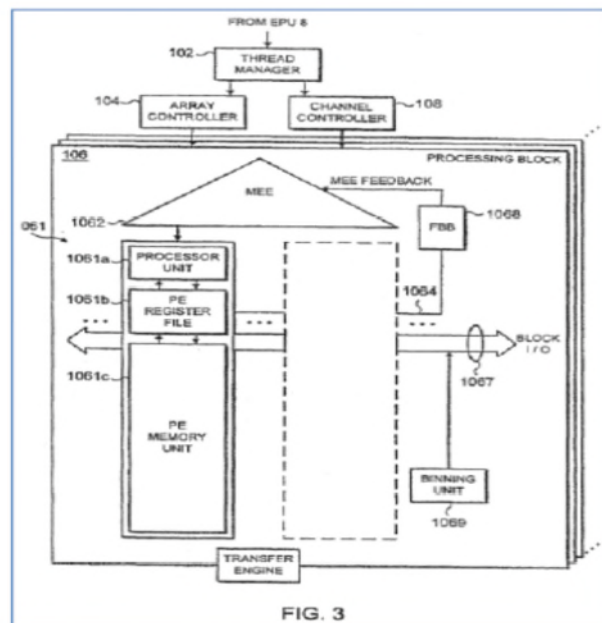
69. Stuttard discloses a thread scheduler 1025 that is coupled to the cache unit 1024. As previously discussed, the thread manager 102 includes the cache unit 1024. The thread manager 102 also includes the thread scheduler 1025 and thread processors 1026. Ex. 1005 at 11:3-10. The thread scheduler 1025 works with the thread processors 1026 to schedule and activate the highest priority thread. “Within the thread manager 102, the status block 1030 receives status information 1036 from each of the threads. The status information is transferred to the thread scheduler 1025 by the status block 1030. The status information is used by the thread scheduler 1025 to determine which thread should be active at any one time. Core instructions 1032 issued by the thread manager 102 are sent to the array controller 104 and the channel controller 108 (FIG. 3).” *Id.* at 12:13-22. Stuttard also teaches that the scheduling of the time multiplexing between threads is determined by the dynamic conditions of the program execution based on thread priority. *Id.* at 23:35-27:32. The thread

processor 1026 uses the thread scheduling to assemble instructions for the array controller 104 and channel controller 108. *Id.* at 26:3-7. This assembly is accomplished using the instruction assembler 510 inside the thread processor 1026. *Id.* at 27:21-23. The thread manager 102 utilizing the thread processors 1026 processes a plurality of pixel command threads and a plurality of vertex command threads. *Id.* at 20:23-21:7 and 22:4-32.

C. Dependent Claim 2: The graphics processing system of claim 1, further comprising: a command processing engine, coupled to the arbiter, wherein the arbiter is further operable to provide the command thread to the command processing engine.

70. Stuttard discloses a plurality of processing blocks 106 coupled to the thread scheduler 1025 and thread processors 1026, and that the thread scheduler and thread processors are further operable to provide the vertex or pixel thread to the processing blocks.

71. Stuttard discloses a processing core 10 comprising multiple processing blocks 106. Ex. 1005 at 8:9-13 and Figure 2. These processing blocks perform the actual graphics processing. *Id.* at 8:1-7. The processing blocks 106 are coupled to the thread manager 102 containing the thread scheduler 1025 and thread processors 1026 through the array controller 104 and the channel controller 108:



Id. at Fig. 3; *see also id.* at 8:17-27.

72. The processing blocks 106 further contain an array of processor elements 1061 that carry out instructions received from the thread manager 102 through the array controller 104. *See id.* at 9:11-20. These processor elements 1061 process both pixel and vertex command threads. *See id.* at 20:32-21:7, 22:26-32. In my opinion, because the processing blocks 106 and the processing elements 1061 contained therein perform the computational processing of the pixel and vertex command threads, each block comprises a “command processing engine.”

D. Dependent Claim 3: The graphics processing system of claim 1, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.

73. Stuttard discloses the cache unit (the claimed “memory device”) may be replaced by a series of FIFOs, one per thread. *See id.* at 23:3-11. The series of

FIFOs can thus be logically organized into a first and second set of FIFOs corresponding to “the first portion” and “the second portion” of the memory device, comprising “a first memory device” and “a second memory device”, respectively. Moreover, it would have been obvious from Stuttard in view of the knowledge of a person of ordinary skill in the art that the different portions of the at least one memory device discussed in element 1[a] could be contained in different memory devices. To the extent that the portions cited for element 1[a] do not explicitly disclose multiple memory devices for the multiple portions, a person of ordinary skill would have recognized that putting the portions in either one or two memory devices would be a simple design choice. These choices would have been well-known, conventional, and likely to yield predictable results.

E. Dependent Claim 4: The graphics processing system of claim 1, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.

74. As discussed earlier, Stuttard discloses a memory device in the form of a cache unit 1024. *Id.* at 11:3-25. Because this cache unit comprises a single device, this claim is satisfied for the same reasons as discussed with respect to element 1[a].

F. Independent Claim 5 of the '053 Patent

1. 5[preamble]: A graphics processing system comprising:

75. To the extent that the preamble of claim 5 is limiting, it is my opinion that Stuttard discloses a graphics processing system. This limitation is identical to

the requirements of element 1[preamble], so this limitation is disclosed for the same reasons as discussed with respect to element 1[preamble].

2. **5[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads;**

76. Element 5[a] is identical to element 1[a], so, this limitation is disclosed for the same reasons as discussed with respect to element 1[a].

3. **5[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads; and**

77. Element 5[b] is identical to element 1[b], so, this limitation is disclosed for the same reasons as discussed with respect to element 1[b].

4. **5[c]: a plurality of command processing engines, coupled to the arbiter, each operable to receive and process the command thread.**

78. Element 5[c] is nearly identical to claim 2 with multiple command processing engines, so, this limitation is disclosed for the same reasons as discussed with respect to claim 2. Further, as I explained in claim 2, Stuttard discloses multiple engines (processing blocks 106 containing processing elements 1061) that process command threads and an arbiter that selects command threads. To the extent that it is not expressly or inherently disclosed in Stuttard, a person of ordinary skill in the art would have found it obvious that command processing engines that process command threads selected by an arbiter would also receive those command threads.

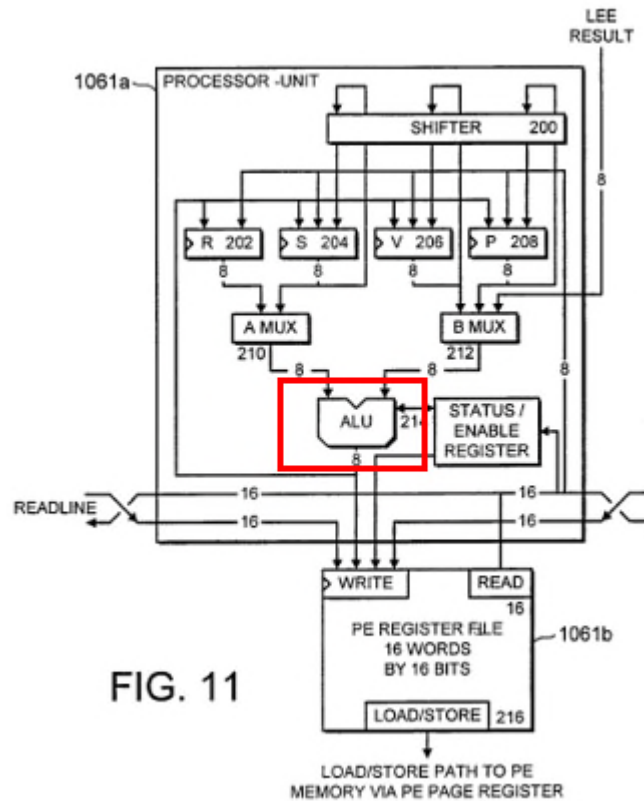
79. In the ITC matter, AMD has argued that each processing engine must be configured to process both pixel and vertex command threads. Again, nothing in the claims require importing any such restriction. Claim 5 requires only that each command processing engine be operable to receive and process “the” command thread selected by and received from the arbiter. Because the arbiter may select from “either of” pixel or vertex command threads, the claim allows the arbiter to select only pixel command threads or only vertex command threads. Thus, a command processing engine that can process a pixel or vertex command thread, respectively, would satisfy the claim. Alternatively, the “plurality of engines” may include one or more engines that process only pixel command threads when selected, with another one or more engine that processes only vertex command threads when selected. Regardless, as discussed further herein, *Stuttard* discloses both ALU and texture processing engines, which are capable of processing both vertex and pixel operations. Ex. 1011 at 101.

G. Dependent Claim 6: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one arithmetic logic unit.

80. As discussed with respect to claim 2, *Stuttard* discloses command processing engines in the form of processing blocks 106 that contain multiple processing elements 1061. *See id.* at 8:1-7 and 9:11-20.

81. Stuttard also discloses that processor units 1061a, which are contained in processing elements 1061, have “an arithmetic logic unit ... which is connected to receive data values.” *Id.* at 47:24-30. Thus, the command processing engines from Stuttard contain at least one arithmetic logic unit.

82. The arithmetic logic units (ALU) contained in the processor units 1061a is further illustrated in Figure 11:



Id. at Fig. 11 (annotated). See also *id.* at 47:24-30.

H. Dependent Claim 7: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one texture processing engine.

83. As discussed with respect to claim 2, Stuttard discloses command processing engines in the form of processing blocks that contain multiple processing elements 1061. *See id.* at 8:1-7 and 9:11-20.

84. Further, Stuttard discloses that “lighting and shading ... [are] performed to produce data depending upon the chosen viewpoint.” *Id.* at 21:3-7. In image processing, texture is defined as a function of spatial variation of the brightness intensity of the pixels. Moreover, the texture mapping/processing came about due to the requirement of sampling and transforming a flat image (as the texture map) to the correct angle and perspective (i.e., viewpoint) it would need to be in 3D space. A person of ordinary skill would have recognized that lighting and shading operations necessarily require a texture processing engine.

I. Dependent Claim 8: The graphics processing system of claim 5, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.

85. Claim 8 is identical to claim 3, so, this limitation is disclosed for the same reasons as discussed with respect to claim 3.

J. Dependent Claim 9: The graphics processing system of claim 5, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.

86. Claim 9 is identical to claim 4, so, this limitation is disclosed for the same reasons as discussed with respect to claim 4.

VIII. GROUND 2: STUTTARD AND WILLIAMS RENDER OBVIOUS CLAIMS 1-9

A. Overview of Williams

87. I understand that PCT Publication No. WO2000063770A1 was published on October 26, 2000. *See* Ex. 1006 at 1. I refer to this publication as “Williams” throughout this Declaration. I have been informed that its publication date makes it eligible as prior art to the ’053 patent.

88. Williams discloses an apparatus and method to increase the throughput of graphics commands transferred between a system bridge and a graphics subsystem, wherein the graphics subsystem 102 includes its own local processor for computing graphical transformations and has its own local graphics memory, such as texture memory and a frame buffer. *Id.* at Abstract and at 2:8-22. Williams discloses that its graphics subsystem processes various types of graphics data, including texture data and geometry data, also referred to as polygon descriptions. *Id.* at 3:1-14. Williams discloses that groups of graphics commands are ordered within a sequence of graphics commands. *Id.* 14:8-21. Although not explicitly stated

by Williams, each sequence of graphics commands constitutes a command thread as is known by a POSITA.

89. Williams in its example embodiment of Figure 2 discloses an apparatus comprising an input buffer 202, a graphics bus scheduler (GBS) 204, two or more graphics busses 116A - 116N, a graphics bus de-scheduler (GBD) 210, and an output buffer 212. *Id.* at 11:5-8. Input buffer 202, which is coupled to system bridge 106, temporarily stores graphics commands that are to be transferred from system bridge 106 to graphics subsystem 102. Input buffer 202 stores the graphics commands such that their original order is known. GBS 204 tags the multiple graphics commands that are stored within input buffer 202 to thereby indicate the specific order of the commands. GBS 204 can assign/sort the graphics commands according to the type of command. That is, GBS 204 may assign geometry related commands (i.e., vertices) to a first bus 116A while assigning texture related command (i.e., texture maps) to a second bus 116B. *Id.* at 12:3-14. The plurality of busses 116A-116N then transfer the multiple graphics commands from GBS 204 to GBD 210. GBD 210 accepts and transfers the accepted multiple graphics commands into output buffer 212, according to the tags, such that the multiple graphics commands regain the specific order (that is, their original order) within output buffer 212. *Id.* at 14:5-21.

90. Williams in its example embodiment of Figure 4 discloses an alternative apparatus where GBS 204 assigns/sorts graphics commands according to

their type, and sends the commands directly to specific functional components of graphics subsystem 102. Thus, in this embodiment, there is no need to reorder the graphics commands after they are transferred across the plurality of busses 116A-116N. Further, because the graphics commands are not reordered, there is no need to tag the graphics commands. *Id.* at 18:14-22.

91. Williams's computer system includes a memory device (e.g., a system memory and an input buffer), an arbiter (e.g., GBS which is coupled to both the system memory and the input buffer, and is operable to select between a sequence of pixel commands and a sequence of vertex commands for transfer), and a (command) processing engine i.e., graphics functional components within the graphics subsystem, which are coupled to the GBS and are operable to execute pixel and vertex processing commands. Williams discloses that multiple graphics commands are provided to specific functional components 402, 404 within graphics subsystem 102. For example, components 402 and 404 can be functional components of graphics subsystem 102 that perform geometry and pixel operations, respectively. *Id.* at 22:3-9.

92. Williams discloses various embodiments and/or configurations of elements and/or features. A person of ordinary skill in the art would have understood that the concepts, components and/or features apply equally across each embodiment. That is, a person of ordinary skill in the art would not have understood them to be

mutually exclusive and they could be “mixed and matched” within the spirit of the disclosure. Moreover, a person of ordinary skill in the art would have been motivated to mix and match them as they all are disclosed in the same reference and are directed to the same field of invention and solving the same problems.

B. Independent Claim 1

1. 1[preamble]: A graphics processing system comprising:

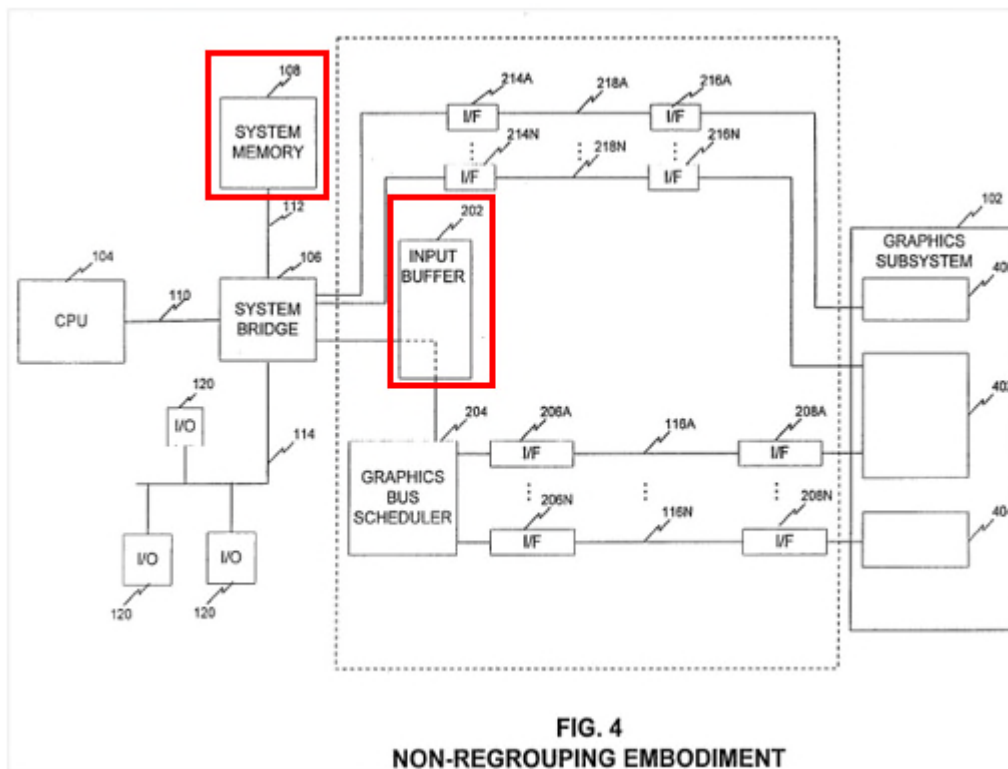
93. I have been informed that the preamble may not be limiting. To the extent that it is limiting, it is my opinion that Stuttard discloses 1[preamble] for the same reasons as I previously discussed in Ground 1.

2. 1[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads; and

94. As I discussed with respect to Ground 1, Stuttard discloses a memory device that stores a plurality of pixel command threads and a plurality of vertex command threads. Further, I have already discussed that it would have been obvious to modify Stuttard to store these different types of commands in different portions of its memory device based on the well-known benefits of storing command threads in this way.

95. To the extent that the cache unit 1024 from Stuttard does not disclose separate portions storing different types of command threads, this limitation is disclosed in Williams.

96. Williams discloses a memory either as the system memory 108 or an input buffer 202. These components are illustrated in Williams' Figure 4:



Ex. 1006 at Fig. 4 (annotated).

97. The system memory 108 can include graphics commands that may be read by the graphics subsystem 102. *Id.* at 11:10-12. Alternatively, Williams discloses an input buffer 202 that “temporarily stores graphics commands that are to be transferred from system bridge 106 to graphics subsystem 102.” *Id.* at 11:20-22. Both of these devices operate as a memory.

98. The graphics processing system in Williams further includes a system bridge 106 that couples the CPU 104 and the system memory 108 to the graphics subsystem 102. *Id.* at 11:9-19. This system bridge 106 can further be coupled to a

graphics bus scheduler (GBS) 204 through the input buffer 202. *Id.* at 18:23-25. This allows the system bridge 106 to send graphics commands to the GBS 204, which the GBS 204 then assigns to a plurality of busses 116A-116N to transfer the graphics commands from the GBS 204 to the graphics subsystem 102. *Id.* at 19:4-11. The GBS 204 assigns graphics commands to the buses based on the type of command, such that geometry commands (*i.e.*, vertex commands) can be assigned to a first bus and pixel graphics commands can be assigned to a second bus. *Id.* This allows the graphics commands to be sent to the specific functional components of the graphics subsystem 102. *Id.*

99. The order that the commands are stored in either the system memory 108 or the input buffer 202 is the order that the GBS 204 transfers the graphics commands across the plurality of busses 116A-116N. *Id.* at 11:20-24, 18:14-20. In my opinion, because the GBS 204 sends different types of commands to different busses 116A-116N, a person of ordinary skill would find it obvious that either the system memory 108 or the input buffer 202 that sends the graphics commands to the GBS 204 would store different types of commands in different portions of their respective memories. *See id.* at 18:14-20, 19:4-11. This would help accomplish Williams' goal of greater efficiency of command fetching and transfer because it was well-known at the time that commands stored in consecutive addresses can be read/written with fewer instructions than the non-consecutive alternative because

reading non-consecutive addresses involves address jumps. *Id.* at Abstract. Hardware at this time also included automatic increment addressing, which Williams could have utilized with consecutive memory addresses.

100. In addition, in the embodiment of Figure 6, Williams discloses an output buffer 212 that temporarily stores graphics commands that have gone through the busses 116B-D and the graphics bus descheduler 210 “such that the multiple graphics command regain the specific order.” *Id.* at 23:16-19. Only geometry or vertex related graphics commands are sent to the output buffer 212, whereas pixel commands are sent directly across bus 116A from the input buffer 202 to the graphics subsystem 102. *Id.* at 23:11-21. So, a person of ordinary skill would know that the vertex commands are stored in the output buffer while the pixel commands are stored in the input buffer. This is illustrated in Figure 6:

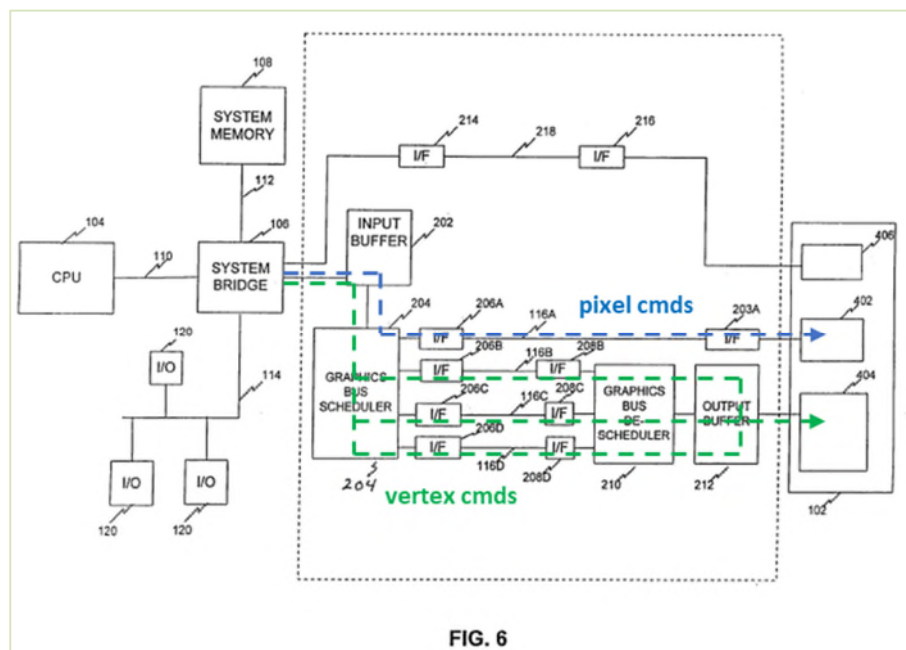


FIG. 6

Id. at Fig. 6 (annotated).

101. In my opinion, it would have been obvious to a POSITA to combine Stuttard's multithreaded system with Williams' separately stored command threads.

102. First, Williams and Stuttard are both graphics processing systems. Ex. 1005 at 4:3-5; Ex. 1006 at Abstract. As a result, a POSITA working in the field of graphics processing would have been familiar with both references. These references also have overlapping components, such as a CPU in a host system and a graphics subsystem. *See* Ex. 1005 at Fig. 1; Ex. 1006 at Fig. 1B. Both references also have the host system providing data and graphics commands to the graphics subsystem. *See, e.g.*, Ex. 1005 at 5:18-22; Ex. 1006 at 3:15-16. Further, both references discuss processing well-known, conventional graphics commands, such as vertex and pixel commands. Ex. 1005 at 1:14-17; Ex. 1006 at 6:23-24. Based on these overlapping components and processes, a POSITA would have considered these to be compatible systems such that these systems could be combined with a reasonable expectation of success. Additionally, modifying Stuttard to store different types of commands in different memory portions would simply require using existing technology in a well-known manner to separately store the command threads. A POSITA would also have been aware that vertex and pixel commands are used in separate, distinct stages of graphics processing. *See* Ex. 1005 at 20:21-22:32; Ex. 1006 at Fig. 7. As a result, it is my opinion that separately storing vertex

and pixel commands in separate portions of a memory device (or in separate memory devices), as taught by Williams, would have been an obvious design choice that a POSITA would have considered.

103. Second, there would be clear benefits that would have motivated a POSITA to combine the references. Both references seek to increase the speed of graphics processing. Ex. 1005 at 1:18-22; Ex. 1006 at Abstract. For Stuttard, this is accomplished through an innovative processing array. Ex. 1005 at Abstract. For Williams, this is done by increasing the throughput of transferring graphics commands to the graphics subsystem. Ex. 1006 at Abstract. A POSITA would have recognized these as separate solutions directed to achieving the same goal through different means. Combining these systems would increase the speed of a graphics processing system through both the innovative processing array and the higher throughput command-transfer architecture.

104. Additionally, the teachings in Williams specifically disclose improvements to the host-graphics-subsystem. Stuttard uses an AGP bus to connect the graphics subsystem and the host CPU, but Williams discloses that an AGP bus limits graphics data transfer and can be a bottleneck. Ex. 1005 at 5:9-11; Ex. 1006 at 26:10-15. Using the teachings from Williams, a POSITA would have recognized that Stuttard's system can be improved to create greater throughput without the need for extraneous local graphics memory. Ex. 1006 at 7:24-8:2.

105. Williams also discloses a system bridge that can act as a memory controller, can support address translation, and can maintain cache coherency. Ex. 1006 at 5:9-6:2. A POSITA would have been aware of these advantages from Williams' system, and it would have been motivated to apply them to Stuttard to achieve the same benefits.

106. Finally, a POSITA would have sought to use the separately stored command threads in Williams and applied it to Stuttard to improve on the benefits of Stuttard's multithreaded system that reduces memory access and increases operation speed. See Ex. 1005 at 12:1-7. Storing different types of command threads in different memory portions allows easier access by the arbiter. The system can use easily configurable pointers to quickly access the necessary portions of the system's memory. So, if the system needs access to vertex command threads, it can follow the pointer that leads to that specific portion of the memory that stores vertex command threads.

3. **1[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads based on relative priorities of the plurality of pixel command threads and the plurality of vertex command threads.**

107. Stuttard discloses element 1[b] for the same reasons I discussed in Ground 1.

C. Dependent Claim 2: The graphics processing system of claim 1, further comprising: a command processing engine, coupled to the arbiter, wherein the arbiter is further operable to provide the command thread to the command processing engine.

108. Stuttard discloses claim 2 for the same reasons I discussed in Ground 1.

D. Dependent Claim 3: The graphics processing system of claim 1, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.

109. Stuttard discloses claim 3 for the same reasons I discussed in Ground 1.

110. Additionally, Williams discloses multiple memory devices storing different types of command threads. As discussed with respect to element 1[a], Williams discloses an output buffer 212 that temporarily stores graphics commands that have gone through the busses “such that the multiple graphics command regain the specific order.” *Id.* at 23:16-19. Only geometry or vertex related graphics commands are sent to the output buffer 212 for reordering, where pixel commands are sent directly from the input buffer 202 to the graphics subsystem 102. *Id.* at 23:11-21. In my opinion, this means that the input buffer 202 and the output buffer 212 would comprise two separate memory devices for each portion claimed in element 1[a].

111. As discussed in element 1[a], a POSITA would have been motivated to combine Stuttard and Williams such that Stuttard would store different types of command threads in different portions of its memory. Whether that memory device

is a single device or, as discussed in Williams, multiple devices, would be a simple design choice.

E. Dependent Claim 4: The graphics processing system of claim 1, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.

112. Stuttard discloses claim 4 for the same reasons I discussed in Ground 1.

F. Independent Claim 5 of the '053 Patent

1. 5[preamble]: A graphics processing system comprising:

113. To the extent that the preamble of claim 5 is limiting, it is my opinion that Stuttard discloses a graphics processing system. This limitation is identical to the requirements of element 1[preamble], so this limitation is disclosed for the same reasons as discussed with respect to element 1[preamble].

2. 5[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads;

114. Element 5[a] is identical to element 1[a], so, this limitation is disclosed for the same reasons as discussed with respect to element 1[a].

3. 5[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads; and

115. Element 5[b] is identical to element 1[b], so, this limitation is disclosed for the same reasons as discussed with respect to element 1[b].

4. **5[c]: a plurality of command processing engines, coupled to the arbiter, each operable to receive and process the command thread.**

116. Element 5[c] is identical to claim 2 with multiple command processing engines, so, this limitation is disclosed for the same reasons as discussed with respect to claim 2.

- G. **Dependent Claim 6: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one arithmetic logic unit.**

117. Stuttard discloses claim 6 for the same reasons I discussed in Ground 1.

- H. **Dependent Claim 7: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one texture processing engine.**

118. Stuttard discloses claim 7 for the same reasons I discussed in Ground 1.

- I. **Dependent Claim 8: The graphics processing system of claim 5, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.**

119. Claim 8 is identical to claim 3, so, this limitation is disclosed for the same reasons as discussed with respect to claim 3.

J. Dependent Claim 9: The graphics processing system of claim 5, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.

120. Claim 9 is identical to claim 4, so, this limitation is disclosed for the same reasons as discussed with respect to claim 4.

IX. GROUND 3: WILLIAMS AND WHITTAKER RENDER OBVIOUS CLAIMS 1-9

A. Overview of Whittaker

121. I understand that US Patent No. 5,968,167 was issued on April 3, 1997. *See* Ex. 1007 at 1. I refer to this publication as “Whittaker” throughout this Declaration. I have been informed that its publication date makes it eligible as prior art to the ’053 patent.

122. Whittaker discloses a multithreaded system wherein a thread is selected for processing based on relative priorities. Whittaker is directed to a data processing management system for controlling the execution of multiple threads. The management system, which includes a media control core, a number of data processing units, and a multi-banked cache, executes the processing instructions of various threads according to the resource availability and thread priority. *See id.* at Abstract; 1:9-26. Whittaker explicitly discloses that its multithreaded system can be used for graphics processing. *Id.* at 4:49-55, 6:30-44.

123. At the center of the system is a Media Control Core (MCC) 2. As shown in Figure 1, Media Control Core (MCC) 2 includes a plurality of inputs and output

devices 4, data processing units 6, and multi-bank cache 14. The MCC 2 checks which of the possible operations it could perform have all the resources available for those tasks to be executed and, of those, which has the highest priority. *Id.* at 3:60-63. This specific resource checking and priority selection of multiple threads is further illustrated in Fig. 6. As seen, instruction buffers store threads 1, 2, and 3, as well as their various global status information. *Id.* at 8:17-23. The threads and their status information are fed into a resource checker 81, which ensures that everything required to perform a particular task is in place. *Id.* at 3:66-67. If all the information required to perform a task is present, the thread is sent to priority selection block 82, which selects for execution the thread with the highest priority. *Id.* at 8:27-31.

124. Whittaker discloses various embodiments and/or configurations of elements and/or features. A person of ordinary skill in the art would have understood that the concepts, components and/or features apply equally across each embodiment. That is, a person of ordinary skill in the art would not have understood them to be mutually exclusive and they could be “mixed and matched” within the spirit of the disclosure. Moreover, a person of ordinary skill in the art would have been motivated to mix and match them as they all are disclosed in the same reference and are directed to the same field of invention and solving the same problems.

B. Independent Claim 1

1. 1[preamble]: A graphics processing system comprising:

125. I have been informed that the preamble may not be limiting. To the extent that it is limiting, it is my opinion that Williams discloses a graphics processing system. As I explained above, Williams discloses an apparatus and method to increase the throughput of graphics commands transferred between a system bridge and a graphics subsystem, wherein the graphics subsystem 102 includes its own local processor for computing graphical transformations and has its own local graphics memory, such as texture memory and a frame buffer. Ex. 1006 at Abstract and 2:8-22.

126. One embodiment of Williams' graphics processing system is illustrated in Figure 1B:

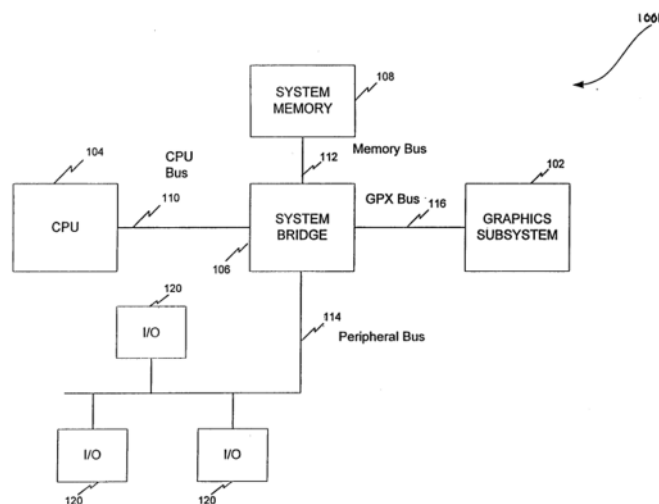
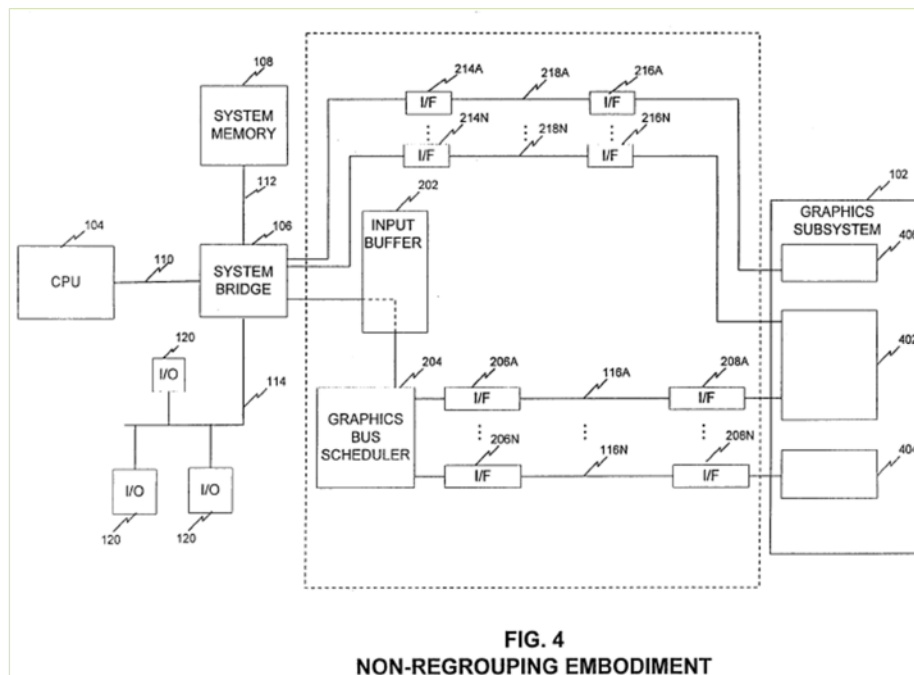


FIG. 1B

Id. at Fig. 1B.

127. As shown in Figure 1B, Williams discloses a computer system 100B with a graphics subsystem 102, a central processing unit (CPU) 104, a system memory 108, and a system bridge 106. *See id.*; *see also id.* at 2:8-22.

128. An embodiment of Williams' graphics processing system is further illustrated in Figure 4:



Id. at Fig. 4.

129. Similar to Figure 1B, Figure 4 further depicts a computer system that includes a graphics subsystem 102, a CPU 104, a system memory 108, and a system bridge 106. *See id.*; *see also id.* at 18:14-22.

130. The graphics subsystem 102 may initiate a transfer of graphics commands. *Id.* at 11:9-15. A graphics command may be transferred from a system

memory 108 to the graphics subsystem 102 via the system bridge 106. *Id.* at 11:9-19.

131. In my opinion, these components and functions mean that Williams' computer system constitutes a graphics processing system.

2. **1[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads; and**

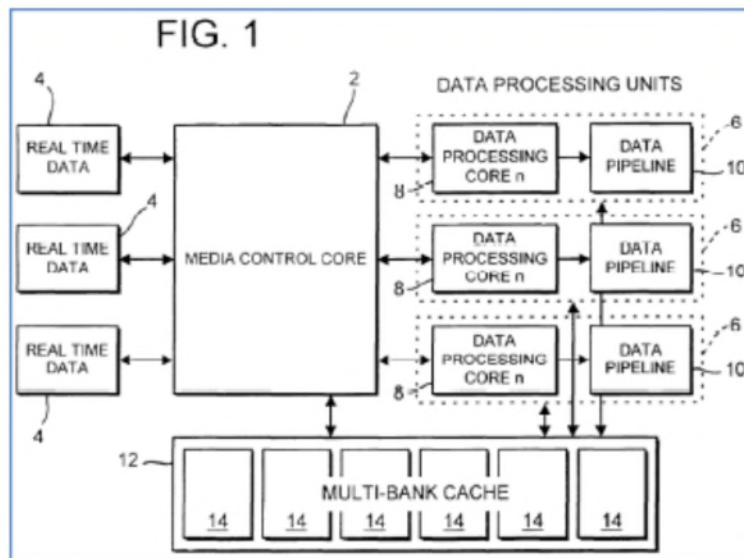
132. Williams discloses element 1[a] for the same reasons I discussed in Ground 2.

3. **1[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads based on relative priorities of the plurality of pixel command threads and the plurality of vertex command threads.**

133. In my opinion, the GBS 204 disclosed in Williams operates as an arbiter. As discussed above, the system bridge 106 couples the CPU 104 and the system memory 108 holding graphics commands to the GBS 204. Ex. 10 (Williams) 11:9-19 and 18:23-25. The system bridge 106 can further be coupled to the GBS 204 through the input buffer 202. *Id.* at 18:23-25. The GBS 204 can sort and assign graphics commands to a plurality of busses 116A-116N. *Id.* at 18:14-20. These graphics commands are assigned by the GBS 204 to the plurality of busses 116A-

116N according to the type of graphics command, *i.e.*, based on whether it is a geometry/vertex command or a pixel command. *Id.* at 18:14-20, 9:8-15.

134. To the extent that the GBS 204 in Williams is not explicitly disclosed as selecting threads “based on relative priorities,” this limitation is disclosed in Whittaker (Ex. 1007). As discussed above, Whittaker is directed to a data processing management system that controls the execution of multiple threads. Ex. 1007 at Abstract. This data processing management system can be for a graphical user interface (*i.e.*, a graphics processing system). *Id.* at 1:9-20; *see also id.* at 6:42-56. An illustration of Whittaker’s system is depicted in Figure 1:

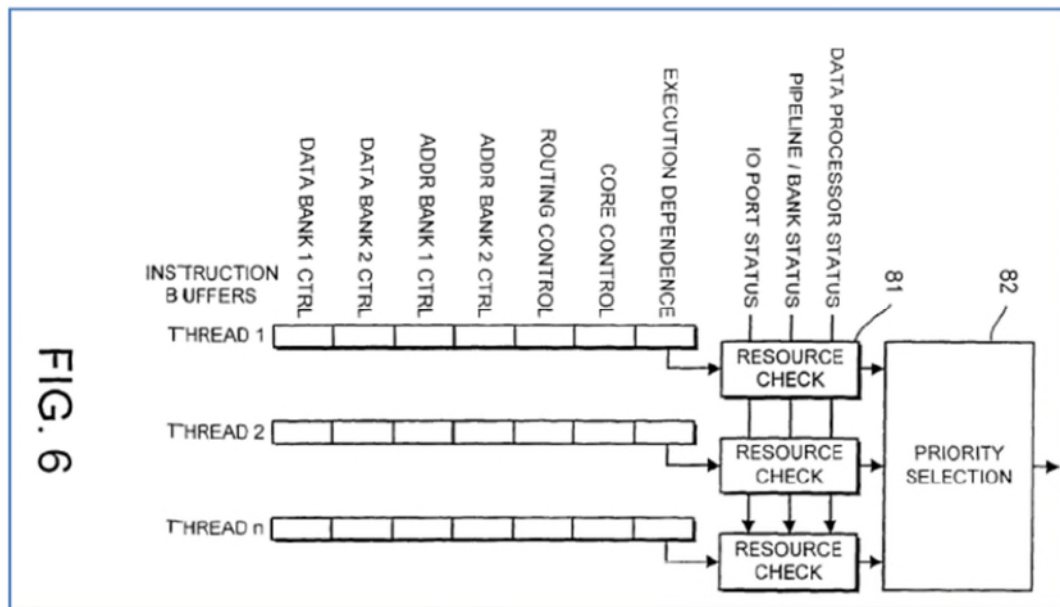


Id. at Fig. 1.

135. As shown in Figure 1, the data processing management system includes a Media Control Center Core (MCC) 2 with a plurality of inputs and output devices 4, data processing units 6, and a multi-bank cache 14. *Id.* The MCC 2 is a fine-

grained multithreading processing unit that is able to determine the possible operations that the system can perform based on the resources available, and it can determine which operations have the highest priority. *Id.* at 3:56-65.

136. As illustrated in Figure 6, Whittaker performs specific resource checking and priority selection from multiple threads:



Id. at Fig. 6.

137. The MCC 2 performs the above priority thread selection. *Id.* at 3:28-30. Instruction buffers store the multiple threads illustrated in Figure 6, including the corresponding global status information, routing control data, control status information, and execution dependency data. *Id.* at 8:17-23. This information is sent to a resource checker 81, and, for each thread, it combines that information with data “from 10 [IO] ports, the various pipeline data bank status, and the status of the

various data processing units.” *Id.* at 8:23-27. If this data processing thread can be run, then an output is generated to the priority selector 82. This has information about the priority of each of the threads and as a result can select for execution the highest priority thread. *Id.* at 8:27-31. This allows the system to determine and execute, for example, a real time data thread precedence over a background processing thread. *Id.* at 8:31-34.

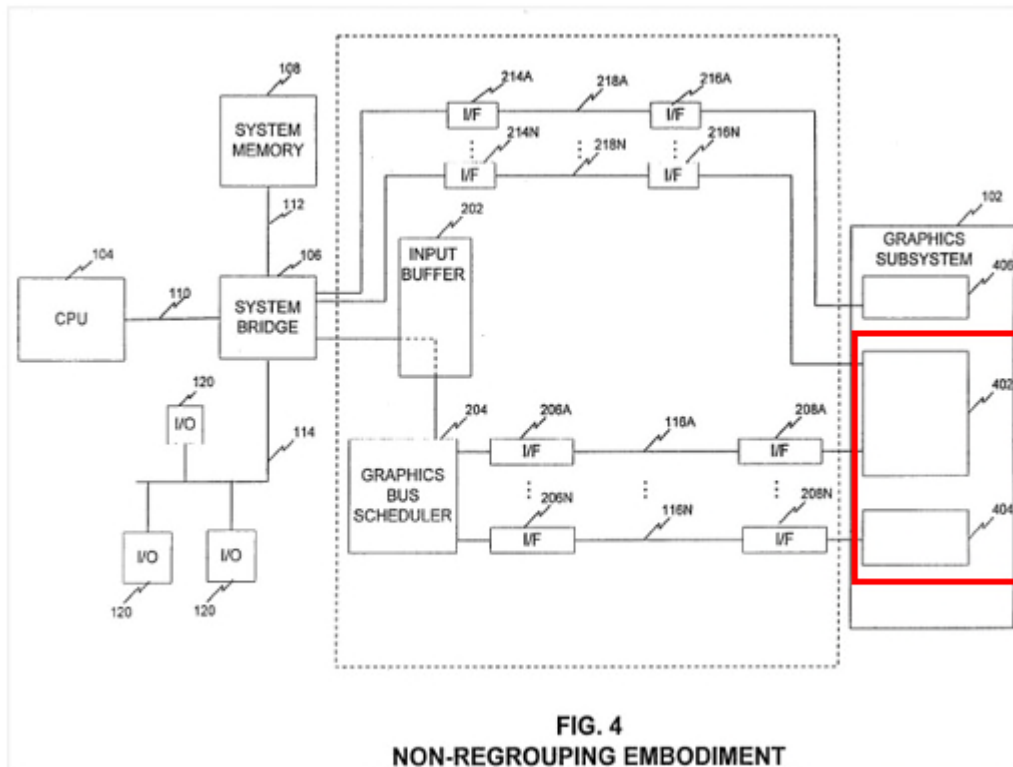
138. A person of ordinary skill would have been familiar with both Williams and Whittaker because both are directed to 3D graphics processing and rendering. Ex. 1006 at 2:7-12; Ex. 1007 at 1:9-20. Additionally, a person of ordinary skill would have recognized that the references could generally be combined to improve the systems because both process conventional 3D graphics objects and perform vertex and pixel processing. Ex. 1006 at 6:23-24; Ex. 1007 at 6:50-56. Additionally, the arbiters for both systems, the GBS 204 from Williams and the MCC 2 from Whittaker, both perform thread selection in a multi-threaded processing system. Ex. 1006 at 18:26-28 and 19:4-7; Ex. 1007 at 3:28-30. Finally, both are concerned with resource availability and seeking higher performance when selecting threads for execution. Ex. 1006 at 12:11-14; Ex. 1007 at Abstract and 1:57-63. Based on these similarities, a person of ordinary skill would also have recognized that components from these two systems can generally be combined or interchanged with a reasonable expectation of success.

139. A person of ordinary skill would have further been motivated to combine Williams and Whittaker. Both references have a common goal of improving efficiency when transferring information and commands to a graphics processing system. Ex. 1006 at 15:2-3; Ex. 1007 at 3:5-8. A person of ordinary skill seeking to further improve on the efficiency of Williams would have been motivated to combine its GBS 204 with Whittaker's priority-selection function to execute threads of the highest priority before lower-priority threads. For example, Whittaker recognizes that it is more efficient to prioritize certain data, such a real time video, which requires immediate output. Ex. 1007 at 8:29-34. A person of ordinary skill would have understood that both the GBS from Williams and the MCC from Whittaker seek to achieve higher transfer throughput to the graphics system. *Id.*, Ex. 1006, 4:22-29 (disclosing an embodiment with a graphic bus to support real-time data transfer). A person of ordinary skill would have been motivated to combine references that support this common objective through counterpart components. Additionally, it would have been advantageous to use the priority selection function that is taught in Whittaker in Williams to balance the bus load or to unevenly distribute traffic over the buses. Ex. 1006 at 12:11-14. For example, a POSITA would know that in the same way that Williams uses two different sets of buses to send pixel and vertex commands to the graphics subsystem 102, it is possible to send low vs. high priority threads over two sets of buses, with the higher priority

commands taking the direct (lower latency) path from the system bridge to the graphics subsystem.

C. Dependent Claim 2: The graphics processing system of claim 1, further comprising: a command processing engine, coupled to the arbiter, wherein the arbiter is further operable to provide the command thread to the command processing engine.

140. As discussed with respect to claim 1, Williams discloses a graphics subsystem 102 that receives graphics commands. Ex. 1006 at 19:4-11. Williams further discloses that the graphics subsystem 102 receiving these graphics commands has specific functional components 402 and 404 that can perform geometry and pixel operations. *Id.* at 22:3-7. These functional components 402 and 404 within the graphics subsystem 102 constitute command processing engines because they perform processing operations on the graphics command threads received in the graphics subsystem 102. As an illustration, these components are depicted in Williams Figure 4:



Id. at Fig. 4 (annotated).

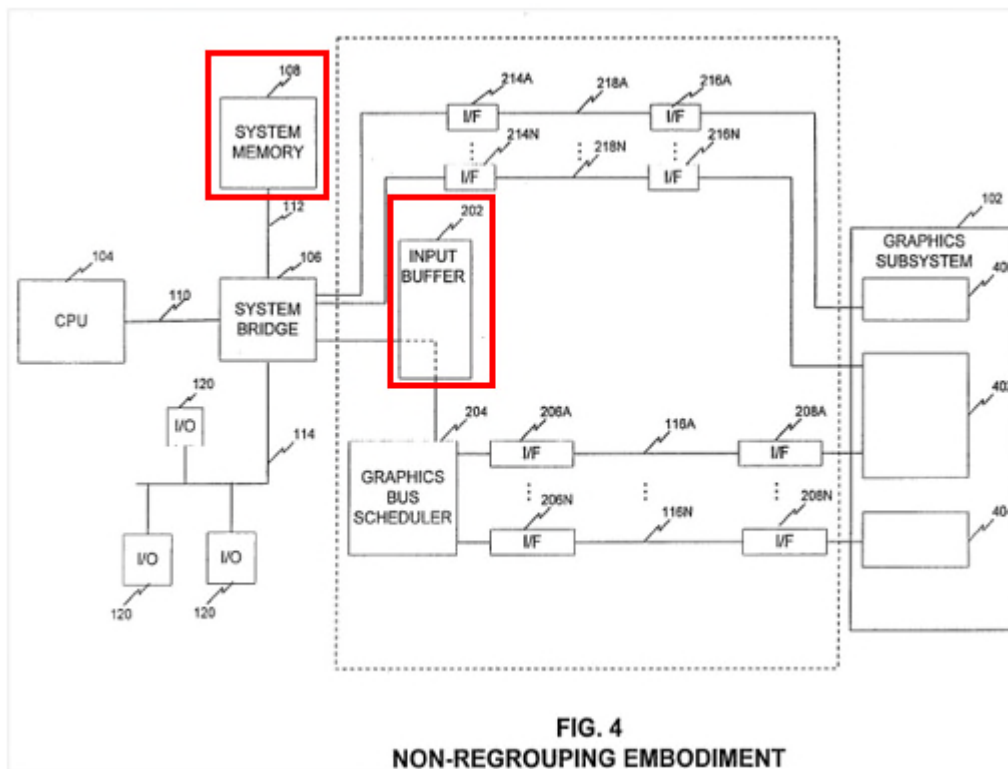
- D. Dependent Claim 3: The graphics processing system of claim 1, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.**

141. Williams discloses claim 3 for the same reasons I discussed in Ground 2.

- E. Dependent Claim 4: The graphics processing system of claim 1, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.**

142. As discussed with respect to element 1[a], Williams can disclose a memory device in the form of either a system memory 108 or an input buffer 202, and that the memory device can comprise two portions. Ex. 1006 at 11:10-12 and

11:20-22. As illustrated in Figure 1, these are illustrated in some embodiments as single devices:



Id. at Fig. 4 (annotated).

143. In my opinion, these disclosures clearly indicate that either the system memory 108 or the input buffer 202 can be single devices.

F. Independent Claim 5 of the '053 Patent

1. 5[preamble]: A graphics processing system comprising:

144. To the extent that the preamble of claim 5 is limiting, it is my opinion that Williams discloses a graphics processing system. This limitation is identical to the requirements of element 1[preamble], so this limitation is disclosed for the same reasons as discussed with respect to element 1[preamble].

2. **5[a]: at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads;**

145. Element 5[a] is identical to element 1[a], so, this limitation is disclosed for the same reasons as discussed with respect to element 1[a].

3. **5[b]: an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads; and**

146. Element 5[b] is identical to element 1[b], so, this limitation is disclosed for the same reasons as discussed with respect to element 1[b].

4. **5[c]: a plurality of command processing engines, coupled to the arbiter, each operable to receive and process the command thread.**

147. Element 5[c] is identical to claim 2 with multiple command processing engines, so, this limitation is disclosed for the same reasons as discussed with respect to claim 2.

- G. **Dependent Claim 6: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one arithmetic logic unit.**

148. As discussed with respect to claim 2, Williams discloses command processing engines in the graphics subsystem 102. Further, Williams discloses that the graphics subsystem 102 can perform geometry calculations. Ex. 1006 at 2:26-30; *see also id.* at 24:18-23 (“Evaluator 706 approximates curve and surface geometry by evaluating polynomial commands of input values. During the next stage,

per-vertex operations and primitive assembly stage 708 processes geometric primitives . . . Vertices are transformed and lit, and primitives are clipped to a viewport in preparation for the rasterization stage 710.”). Additionally, as discussed with respect to claim 2, the command processing engines (*i.e.*, the components 402 and 404) in Williams “can be functional components of graphics subsystem 102 that perform **geometry** and pixel operations, respectively.” *Id.* at 22:3-7 (emphasis added). A person of ordinary skill would recognize that this means that the graphics subsystem 102 in Williams necessarily includes an arithmetic logic unit to perform at least the necessary geometry, or vertex, operations.

H. Dependent Claim 7: The graphics processing system of claim 5, wherein the plurality of command processing engines comprises at least one texture processing engine.

149. As discussed with respect claim 2, Williams discloses command processing engines in the graphics subsystem 102. Further, Williams discloses that the graphics subsystem 102 “processes texture data in order to create lifelike surfaces and shadows within a 3D image.” *Id.* at 3:1-3. Additionally, as discussed with respect to claim 2, the command processing engines, or the components 402 and 404, in Williams “can be functional components of graphics subsystem 102 that perform geometry and **pixel** operations, respectively.” *Id.* at 22:3-7 (emphasis added). A person of ordinary skill would recognize that this means that the graphics subsystem 102 in Williams necessarily includes a texture processing engine.

- I. Dependent Claim 8: The graphics processing system of claim 5, wherein the first portion of the at least one memory device comprises a first memory device and the second portion of the at least one memory device comprises a second memory device.**

150. Claim 8 is identical to claim 3, so, this limitation is disclosed for the same reasons as discussed with respect to claim 3.

- J. Dependent Claim 9: The graphics processing system of claim 5, wherein a single memory device comprises the first portion and the second portion of the at least one memory device.**

151. Claim 9 is identical to claim 4, so, this limitation is disclosed for the same reasons as discussed with respect to claim 4.

X. DECLARATION

152. In signing this declaration, I understand that this declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the USPTO.

153. I am aware that willful false statements and the like made by me in connection with this declaration are punishable by fine or imprisonment, or both (18 U.S.C. § 1001) and may jeopardize the validity of the Petition. All statements herein made by me of my own knowledge are true and all statements made on information and belief are believed to be true. I declare under penalty of perjury that the foregoing is true and correct.

Executed this 7th day of March, 2023 at Los Angeles, California.

Respectfully Submitted,

A handwritten signature in black ink, appearing to read "M. Pedram", is written over a horizontal line.

Massoud Pedram, Ph.D.