UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ARM LIMITED,
Petitioner,

v.

ICPILLAR LLC,
Patent Owner.

Case IPR2024-00476
Patent 8,924,899

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

Page

I. INTRODUCTION 1

II. THE PETITION FAILS TO SHOW A REASONABLE LIKELIHOOD OF PREVAILING AGAINST ANY OF THE CHALLENGED CLAIMS 2

A. The Banerjee Ground Fails (All Claims, Ground 2). 2

1. The '899 Patent Discloses A Different Tool Than Banerjee. 4

a. The '899 Patent Outputs A Hardware Design And Associated Software To Be Executed On The Hardware. 5

b. Banerjee's Cited Design Tool Does Not Output Software To Be Executed On Its Outputted Hardware Design. 7

2. Petitioner Fails To Show That Banerjee Discloses Or Renders Obvious "Outputting ... Software Programming Commands To Be Executed By The Hardware Integrated Circuit For Controlling The Electronic Device…" (Limitations 1[d] & 8[e]) 13

a. The Limitation Is Not Shown To Be Disclosed. 14

i. Banerjee Is Not Shown To Deliver Software. 16

ii. The "Output" Petitioner Cites Is Data Transmitted Between Internal Modules Of Banerjee's Tool, Not Executable Software Outputted From Its Tool. 19

b. The Limitation Is Not Shown To Be Obvious. 24

B. The Rompaey Ground Fails (All Claims, Ground 1) 25

1. The Patent Discloses A Different Tool Than Rompaey. 26

2. Petitioner Fails To Show That Rompaey Discloses Or Renders Obvious "Outputting ... Software Programming Commands To Be Executed By The Hardware Integrated Circuit For Controlling The Electronic Device…" (Limitations 1[d] & 8[e]) 32

a. The Limitation Is Not Shown To Be Disclosed. 33

b. Rompaey Models Software, But Does Not Output It. 36
c. "SYMPHONY" Does Not Output The Claimed Software. 39
d. The Limitation Is Not Shown To Be Obvious. 43
III. THIS PETITION SHOULD BE DENIED UNDER *FINTIV* DUE TO THE ADVANCED STATE OF PARALLEL DISTRICT COURT LITIGATION. 44
A. *Fintiv* Factor 1: The District Court Has Not Granted A Stay And There Is Zero Evidence That A Stay Might Be Granted. 47
B. *Fintiv* Factor 2: The District Court Trial Will Occur Over Five Months Before The Deadline For A Final Written Decision In This Proceeding. 49
C. *Fintiv* Factor 3: By The Time Of A Decision On Institution, A Claim Construction Order Is Expected, Final Infringement And Validity Contentions Will Be Complete, And Fact Discovery Will Be Approaching Completion. 57
D. *Fintiv* Factor 4: There Is Perfect Overlap Between This IPR And The District Court Proceeding. 60
E. *Fintiv* Factor 5: Petitioner Is A Defendant In District Court. 64
F. Fintiv Factor 6: Other Circumstances Weigh Against Institution, Including The Weakness Of The Petition. 65
IV. CONCLUSION 66

TABLE OF AUTHORITIES

Page(s)

COURT DECISIONS

Cal. Inst. of Tech. v. Broadcom Ltd.,
25 F.4th 976 (Fed. Cir. 2022) 63

Lone Star SCM Systems, LTD. v. Bluebird Inc.,
6:21-cv-00844-ADA (W.D. Tex. Aug. 1, 2023) 48

Smart Mobile Technologies v. Apple Inc.,
6:21-cv-00603-ADA (W.D. Tex. Aug. 8, 2023) 48

AGENCY DECISIONS

3Shape Trios A/S v. Densys Ltd.,
IPR2021-00236, Paper 12 (May 20, 2021).....63

Apple Inc. v. Fintiv, Inc.,
IPR2020-00019, Paper 11 (Mar. 20, 2020)
(**precedential**)..... 45, 48, 49, 55, 57, 58, 60, 62, 64, 65

Apple Inc. v. Fintiv, Inc.,
IPR2020-00019, Paper 15 (May 13, 2020)
(**informative**)..... 45, 49, 51

BOE Tech. Grp. Co., Ltd. et al. v. Element Capital Commercial Co.,
IPR2023-00808, Paper 9 (Nov. 15, 2023).....59

EClinicalWorks, LLC v. Decapolis LLC,
IPR2022-00229, Paper 10 (Apr. 13, 2022)..... 51, 61

Ericsson Inc. et al. v. Collision Comms., Inc.,
IPR2022-01233, Paper 12 (Jan. 19, 2023).....60

Int'l Bus. Machs. Corp. v. Digital Doors, Inc.,
IPR2023-00970, Paper 7 (Jan. 12, 2024).....50

NHK Spring Co. v. Intri-Plex Techs., Inc.,
IPR2018-00752, Paper 8 (Sept. 12, 2018)
(**precedential**).....45

Nine Energy Serv. Inc. v. NCS Multistage Inc.,
IPR2020-01615, Paper 18 (Mar. 24, 2011); 56

NXP USA, Inc. v. Impinj, Inc.,
PGR2022-00005, Paper 18 (May 2, 2022) 49

NXP USA, Inc. v. Impinj, Inc.,
PGR2022-00005,Paper 20 (Aug. 25, 2022).....51

Roku, Inc. v. IOEngine, LLC,
IPR2022-01551, Paper 11 (May 5, 2023).....59

Samsung Elecs. Co., Ltd. v. Mojo Mobility Inc.,
IPR2023-01094, Paper 11 (Feb. 9, 2024).....59

Vector Flow, Inc. v. HID Global Corp.,
IPR2023-00353, Paper 8 (July 17, 2023) 61

STATUTES

35 U.S.C. § 314(a) 25, 44

REGULATIONS

37 C.F.R. § 42.100(c).....49

OTHER AUTHORITIES

USPTO, Director Memo re Interim Procedure For Discretionary
Denials In AIA Post-Grant Prcdgs. With Parallel Dist. Ct. Litig.
(June 21, 2022)..... 44, 45, 49, 52

EXHIBIT LIST	
2001	Declaration of Joseph C. M^{c}Alexander ^{III}, P.E. [McAlexander]
2002	Welsim | Quantify the Uncertain, *GDSII, the data format for chip and integrated circuit design*, Medium (Nov. 2, 2023), *available at* https://getwelsim.medium.com/gdsii-the-data-format-for-chip-and-integrated-circuit-design-012352970a54 (last visited May 8, 2024) [GDSII-Data Format].
2003	Layout Editor, *GDSII*, *available at* https://layouteditor.org/layout/file-formats/gdsii (last visited May 8, 2024) [GDSII-Layout Editor].
2004	Wikipedia The Free Encyclopedia, *GDSII*, *available at* https://en.wikipedia.org/wiki/GDSII (last visited May 8, 2024) [Wikipedia-GDSII].
2005	U. Waseem, *RTL Design: A Comprehensive Guide to Understanding and Implementing Register-Transfer Level Design*, Wevolver (Jan. 12, 2024), *available at* https://www.wevolver.com/article/rtl-design-a-comprehensive-guide-to-unlocking-the-power-of-register-transfer-level-design (last visited May 8, 2024) [RTL-Design].
2006	IBM, Documentation, *Header Files*, updated Mar. 24, 2023, *available at* https://www.ibm.com/docs/en/aix/7.2?topic=reference-header-files) (last visited May 8, 2024) [IBM-Header Files].
2007	IBM, Documentation, *SWITCH statement*, updated Apr. 3, 2024, *avail. at* https://www.ibm.com/docs/en/netcoolomnibus/8.1?topic=files-switch-statement (last visited May 8, 2024) [IBM-SWITCH].
2008	*A Short Introduction to Makefile*, Univ. of Notre Dame, *available at* https://www3.nd.edu/~zxu2/acms60212-40212/Makefile.pdf (last visited May 8, 2024) [UND-Makefile].
2009	Example, *Learn Makefiles*, *available at* https://makefiletutorial.com/ (last visited May 4, 2024) [Learn Makefiles].
2010	Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation, Katherine K. Vidal (USPTO June 21, 2022) [Interim Discretionary Denial Memo].
2011	*ICPillar LLC v. ARM Ltd. et al*, 1:23-CV-00282-ADA, Defendants' [Petitioner's] Final Invalidity Contentions (Apr. 23, 2024) [ARM-Final Invalidity-Contentions].
2012	*ICPillar LLC v. ARM Ltd. et al*, 1:23-CV-00282-ADA, Civil Docket,

EXHIBIT LIST	
	May 7, 2024 [District-Court-Docket].
2013	United States District Courts – National Judicial Caseload Profile Trial Statistics [Trial-Statistics].
2014	DocketNavigator, U.S. District Courts Statistics for Judge Albright, *obtained at* https://search.docketnavigator.com/patent/judge/16961/15 [Albright-Statistics].
2015	*Sable Networks, Inc. et al v. Cloudflare, Inc.*, 6:21-cv-00261-ADA, Civil Docket, May 7, 2024 [*Sable*-Docket].
2016	*Demaray LLC v. Samsung Elecs. Co., Ltd.*, 6:20-cv-00636-ADA, Civil Docket, May 7, 2024 [*Demaray*-Docket].
2017	*ICPillar LLC v. ARM Ltd. et al*, 1:23-CV-00282-ADA, Defendants' [Petitioner's] Preliminary Invalidity Contentions (Nov. 7, 2023) [Preliminary-Invalidity-Contentions].
2018	*ICPillar LLC v. ARM Ltd. et al*, 1:23-CV-00282, Final Infringement Contentions (Apr. 23, 2024) [ARM-Final-Infringement-Contentions].
2019	*ICPillar LLC v. ARM Ltd. et al*, 1:23-CV-00282-ADA, Claim Construction Order (Mar. 26, 2024) [Markman-Order].

I. INTRODUCTION

The Petition should be denied. Its grounds fail to show any likelihood of unpatentability because, at least, neither Banerjee nor Rompaey satisfies the outputting of "***software programming commands...***" that every claim requires. The '899 patent's approach significantly reduces the time needed to design an integrated circuit, by standardizing and designing an integrated circuit's hardware ***and executable software*** using standard components based on device functionality. In contrast, Banerjee's and Rompaey's tools only output hardware description files, but not software programming commands to be executed by the designed hardware integrated circuit for controlling the device. Simply put, the software programming command output limitations of the claims are not shown to be either disclosed or obvious.

The Petition should also, independently, be denied as a matter of discretion. The District Court litigation is moving swiftly to trial, with all *Fintiv* factors weighing in Patent Owner's favor, over ***five months*** before any IPR trial is reasonably likely to be held.[1]

For both these reasons, institution of the requested IPR should be denied.

[1] Emphases are added, and paragraph numbers omitted, unless otherwise stated.

II. THE PETITION FAILS TO SHOW A REASONABLE LIKELIHOOD OF PREVAILING AGAINST ANY OF THE CHALLENGED CLAIMS.

This case involves one of two IPR petitions in which Petitioner challenges the two patents Patent Owner asserts against Petitioner in District Court: IPR2024-00476 (patent no. 8,924,899) and IPR2024-00566 (patent no. 9,367,657).

Both patents have the same invention disclosure, as one issued from a continuation of the other's application. Both petitions present essentially the same grounds: alleged obviousness based on Banerjee alone, and alleged obviousness based on Rompaey alone or in combination with CoWare I. Petitioner presents the Banerjee ground and the Rompaey ground in a different order in the two petitions. In the Petition in this case, Petitioner presents Banerjee second and Rompaey first, while in the other case against the other patent it presents Banerjee first and Rompaey second.

To simplify the Board's consideration of these two cases, Patent Owner addresses the Banerjee ground first, and the Rompaey ground second.

A. The Banerjee Ground Fails (All Claims, Ground 2).

The Banerjee Ground (ground 2 in the present Petition) alleges that the challenged claims are obvious in view of Banerjee alone. Pet., 57-84. The Petition, however, fails to demonstrate any reasonable likelihood that Banerjee discloses or renders obvious all the limitations of the challenged claims, including

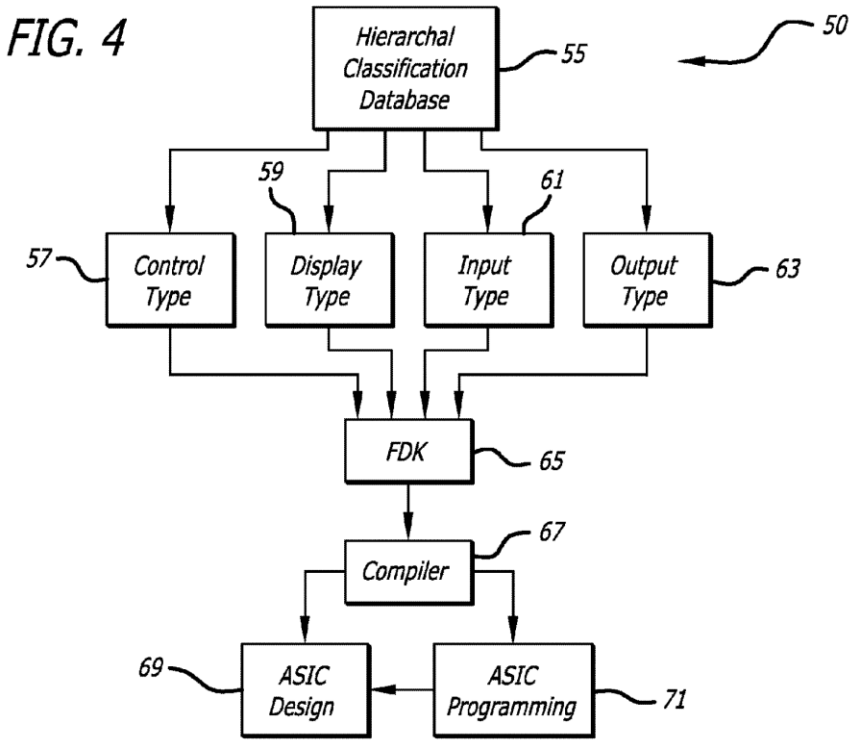
at least limitation 1[d]: "***outputting*** by the processor ***software programming commands to be executed by the hardware integrated circuit*** for controlling the electronic device in accordance with the selection of categorized input and output characteristics of the electronic device," and similar limitation 8[e].

Thus, limitations 1[d] and 8[e] require outputting software programming commands to be executed by the hardware integrated circuit ("IC") for controlling the device. The Petition contends that these limitations are disclosed via Banerjee's discussion of attributes and performance during the optimization steps of designing a hardware IC. Banerjee, however, instead discloses intermediate processing steps that only pass between internal modules of Banerjee's design tool in the course of designing the IC. The tool does not output software programming commands to be executed by the hardware IC. Rather, the only output that Banerjee delivers is a hardware design: specifically, a description file for a hardware integrated circuit. Thus, Banerjee does not disclose limitations 1[d] and 8[e]. This is fatal to the Banerjee ground, as the Petition does not even suggest that it would be obvious to modify Banerjee to add the missing limitations 1[d] and 8[e].

1. The ’899 Patent Discloses A Different Tool Than Banerjee.

The Petition challenges independent claims 1 and 8 and dependent claims 5, 6, and 14 of the ’899 patent. The claims are directed towards designing an integrated circuit and associated software to be executed on that integrated circuit.

In the claimed invention, first, as part of a system or a method to design a hardware integrated circuit and associated software programming commands executed by the integrated circuit to control an electronic device, a memory of a processor receives a selection of categorized input and output characteristics of that device. *See, e.g.*, Ex.-1001, cl. 1, limitation 1[a]. After analysis, based on this selection, the processor outputs a hardware integrated circuit design. *Id., e.g.*, cl. 1, limitations 1[b]-[c]; cl. 6; *see id., e.g.*, FIG. 4, 9:19-10:14; *see also, e.g., id.*, 13:6-13:17 (“The FDK may provide a software representation[] of a complete IC design. In one embodiment, the software representation may be a schematic and/or layout.”); cl. 8. Ex.-2001 [McAlexander] ¶¶25-26.



'899 FIG. 4 (showing, e.g., outputs 69, 71)

The present Preliminary Response will focus on the last limitation of each independent claim, which requires a further output: outputting software programming commands to be executed by the hardware integrated circuit designed above for controlling the electronic device. Ex.-1001, claims 1, 8. Banerjee is not shown to disclose, and is not argued to render obvious, this limitation. Pet., 78-80.

a. *The '899 Patent Outputs A Hardware Design And Associated Software To Be Executed On The Hardware.*

The '899 patent is directed toward, *inter alia*, "simplif[ying] and vertically integrat[ing] the development of the hardware and software for a universal control

system," performing unified design of integrated circuit hardware and executable software. Ex.-1001 ['899 Pat.] 3:20-23. The '899 patent further "provide[s] an IC design based on the selection of input and output characteristics of the device[] and provide[s] software for controlling the device through the IC." *Id.*, 3:21-3:22, 5:26-5:28, 5:50-5:52.

More specifically, the '899 patent describes and claims a method and system for designing a hardware integrated circuit and associated software programming commands to be executed by the hardware integrated circuit for controlling the device. *Id.*, *e.g.*, cl. 1; cl. 8; 18:38-18:40; *see, e.g., id.*, 5:3-5:4 ("In still another aspect, the invention includes providing software for controlling the device ...."), 9:61-9:64 ("ASIC Programming **71** may be software to be programmed and/or stored in a fabricated ASIC and/or a controller to establish interoperability between the controller, ASIC, and/or a device that the ASIC is integrated with."), 13:20-13:23 ("the FDK may also provide accompanying software to run with the IC (assuming it is a programmable IC) and/or on a computer to enable remote control."), 13:36-13:38; Ex.-2001 [McAlexander] ¶¶27-28.

Among other benefits, the '899 patent's approach significantly reduces the time needed to design such an integrated circuit, by standardizing and designing an integrated circuit and the associated software to be executed by the integrated circuit using standard components based on device functionality. *See* Ex.-1001

['899 Pat.], *e.g.*, 4:39-4:45. The claimed inventions are directed towards a computer implemented design tool for designing an integrated circuit which controls the device when it is integrated into that device and software programming commands to be executed by the hardware integrated circuit, for controlling that device in accordance with the received selection of the device's characteristics. *Id.*; *see, e.g., id.*, cl. 1, limitation 1[d], 4:56-5:2 (explaining aspects of the invention including "providing an IC design based on the selection of input and output characteristics of the device[] and providing software for controlling the device through an IC based on the IC design"), 5:26-5:28 (similar), 5:50-5:53 (similar); *see also, e.g., id.*, 4:56-4:57, 5:3-5:8, 5:13-5:14, 9:61-9:62; 13:20-13:23; FIG. 4. *See* Ex.-2001 [McAlexander] ¶¶29-30.

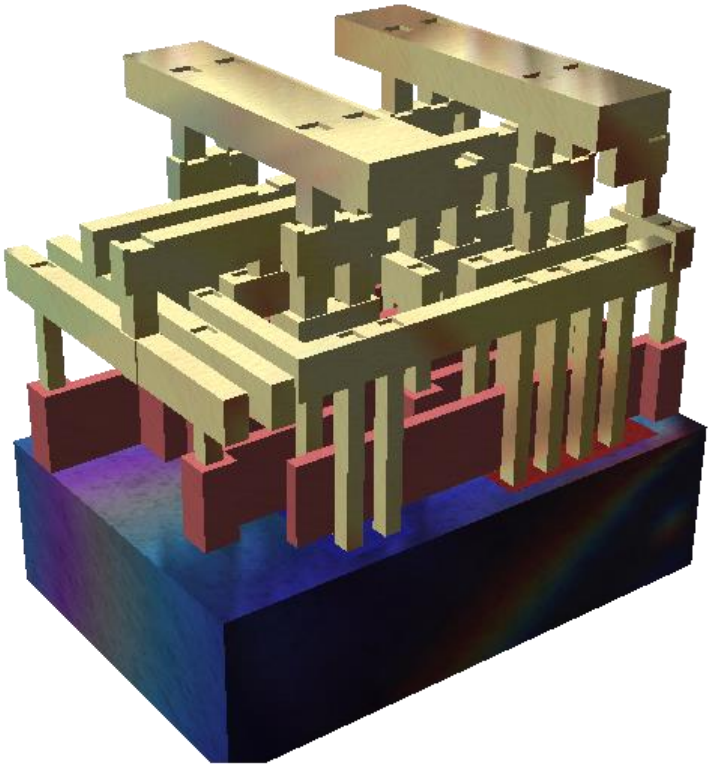
b. *Banerjee's Cited Design Tool Does Not Output Software To Be Executed On Its Outputted Hardware Design.*

Banerjee discloses a tool that fundamentally differs from the '899 patent's invention. Banerjee only outputs from its tool a hardware design. Critically, its tool does not output software, and certainly not software programming commands to be executed by the hardware integrated circuit for controlling the electronic device in accordance with the selection of categorized input and output characteristics of the electronic device. Ex.-2001 [McAlexander] ¶31.

Banerjee describes a design tool "to automatically produce a hardened GDSII description or other representation of" an integrated circuit (IC). Ex.-1005

[Banerjee] Abstract. "Designers select contractual terms for incorporating third party intellectual property and then design and initiate manufacture of the IC by way of a network portal." *Id.* This reference to a "hardened GDSII description" of an IC refers to a hardware description file describing the IC design.[2] This data file for representing the IC artwork is not software that could be executed on the resulting IC. *Id.* Banerjee describes a system that outputs this "hardened" IC design in the form of a *hardware* IC description file. *Id.* It says nothing about the *creation* of *software* to be executed on the hardware of the IC. Ex.-2001 [McAlexander] ¶32.

[2] As an ordinary artisan at the time of the invention would have known, GDSII, which currently is a de facto industry standard for electronic design automation data exchange of IC or IC layout artwork, is a binary database file format, representing in hierarchical form information about the IC layout such as text labels or shapes such as planar geometric shapes, which may be used for creating photomasks, transferring artwork between tools, reconstructing in whole or in part the artwork when sharing layouts. Ex.-2001 [McAlexander] ¶32 (citing Exs. 2002 [GDSI-Data Format], 2003 [GDSII-Layout Editor]), 2004 [Wikipedia-GDSII]).



Example of GDSII hardware design (source: Ex.-2004)

As Joseph McAlexander, P.E., a distinguished expert in integrated circuit and associated software design, explains:

> Banerjee discloses a design tool that "a designer can use to develop the [hardware IC] description file." Ex.-1005 [Banerjee] [0033], Figure 2. Banerjee's tool includes functional description module 21, parameter description module 22, workload analysis module 23, OS optimization module 24, and tools (simulation, formal verification) module 25.
>
> The functional description module 21 of Banerjee's tool guides the user, via a user interface, to develop a functional description of the IC to be designed, and also allows the user to upload design scripts for

customized elements. *Id*., [0033], Figures 2-3. Parameter description module 22 guides the user in defining performance parameters for the functional blocks, which module 21 uses "to optimize the selection of [hardware] functional block components." *Id*., [0034].

Banerjee's workload analysis module 23 accepts algorithms or other specifications that describe the workload targets for the IC the user is designing. *Id.*, [0036]. It then performs an analysis of workload targets to analyze the algorithm's processing loads and identify potential bottlenecks. *Id*., [0036]-[0037]. This "[i]nformation from module 23 is passed to module 21 and module 22 to optimize the selection of the [hardware] functional block components." *Id*., [0036].

The workload target algorithm, loosely referred to in Banerjee as "application programming code" more precisely is a script or algorithm containing workload targets used by module 23 for workload analysis in designing an IC. *Id*. The format used by Banerjee to describe the algorithm can be a flowchart, a script written in C or C++ or other format to specify the IC workload targets. *Id*. The result of the workload analysis module is passed to modules 21 and 22 for optimizing the hardware IC design and is never outputted from the design tool. *Id*.; Figure 2. In short, this workload target algorithm is not an output from the Banerjee design tool. It is a data point used within Banerjee's design tool for optimization of the hardware IC design. More particularly, it certainly is not software programming commands to be executed by the hardware integrated

circuit for controlling the electronic device, like the output software commands expressly required by the '899 Patent's claims' language.

Banerjee's OS optimization module 24 is used to optimize the footprint and performance of the operating system by analyzing the workload from the perspective of the operating system. *Id.*, [0038]; Figure 2. Banerjee's module 24 is also used to analyze what the "typical[]" operating system storage and other footprint needs will be for the IC that is being designed, so that these typical requirements can guide the design of the hardware IC. *Id.*, [0072]. Said another way, module 24, "based on a prototype description," compiles the characteristics of the typical or "baseline" "operating system configuration," in order to analyze its expected "footprint" on the IC and how it is expected to be instantiated on the device. *Id.*, [0072]-[0074], [0083]. As Banerjee puts it: "Module 24 enables the configuration of the operating system to minimize its footprint in memory in the finished product," to avoid "inefficient use of memory resources" in the IC design. *Id.*, [0073].

As Banerjee explains, module 24's intermediate analysis is not an output from its tool, contrary to what the Petition repeatedly suggests. Module 24 "interfaces with modules 23-21 to determine whether it is preferable to implement certain functions [of the IC] in software rather than hardware." *Id.*, [0038]. For example, if a function requires performing a significant number of multiplication operations, then a hardware multiplication implementation may be deemed preferred to minimize the need for the operating system to include a software multiplication implementation. Conversely, if

multiplication is seldom performed, then it may be area and power efficient to implement the multiplication function in software, even though it will take longer to perform the multiplication. *Id.*, [0039]. As illustrated in Figure 2, the result of analysis of module 24 is passed back to modules 21-23. Banerjee's module 24 does not provide any output from Banerjee's design tool.

The tool's module 25 (simulation, formal verification) assembles and outputs a hardware description file. Ex.-1005 [Banerjee] [0040], Figure 2. It specifies the hardware components and operational characteristics in RTL ("Register Transfer Level"). *Id.*, [0040]. As an ordinary artisan would have understood at the time of the '899 Patent invention, RTL refers to a "design abstraction" which models a synchronous digital circuit in terms of the flow of digital signals (data) "between hardware registers," and logical "operations performed on" those signals. Ex.-2005 [RTL-Design] 1-2. "Module 25 also performs logic synthesis to compile and optimize the … circuit description and generate a gate-level netlist representation" of the IC design. Ex.-1005 [Banerjee] [0044].

In sum, the Banerjee design tool (computer program 15, *see id.*, [0045]) gathers information regarding the functional description and design parameters of the IC, optimizes the hardware IC design through workload targets and tradeoff analyses between hardware and software implementation and then, outputs an HDL file descriptive of the hardware IC design. *Id.*, [0045], Figure 2, *see*, *e.g.*, *id.*, Figure 6 (showing step of "Send netlist to foundry" (67) if "Design acceptable" (66)).

> In this way, according to Banerjee's teachings, expected characteristics of an operating system may be used to output a more optimal design for the hardware IC; the operating system "software," referred to in the cited portions of Banerjee, is an optional input in the hardware design. Banerjee does not disclose outputting software for execution on the newly designed IC. Nor does Banerjee disclose writing or outputting software programming commands for controlling any device with the designed IC.

Ex.-2001 [McAlexander] ¶¶33-41.

2. Petitioner Fails To Show That Banerjee Discloses Or Renders Obvious "Outputting ... Software Programming Commands To Be Executed By The Hardware Integrated Circuit For Controlling The Electronic Device…" (Limitations 1[d] & 8[e])

Although Banerjee's teachings differ from the claimed invention in numerous ways, the present Preliminary Response will focus on the Petition's failure to show that Banerjee discloses or renders obvious limitation 1[d]: the output "software programming commands to be executed by the hardware integrated circuit for controlling the electronic device in accordance with the selection of categorized input and output characteristics of the electronic device," and materially similar limitation 8[e].[3] Banerjee's tool outputs only a hardware description file. It does not output software programming commands to be

[3] For convenience, the Petition's numbering of the limitations is used herein.

executed by the IC, to control the electronic device into which it is integrated. What the Petition points to in Banerjee in reference to this limitation are internal analyses in Banerjee's tool during the hardware design process. These intermediate analyses do not produce any output from the tool and certainly not executable software commands. Ex.-2001 [McAlexander] ¶44.

a. *The Limitation Is Not Shown To Be Disclosed.*

The Petition asserts that Banerjee discloses the "software programming commands" of limitation 1[d] (or 8[e]) because Banerjee supposedly "outputs the hardware circuit design and the related software to implement the designed system." Pet., 8; *see also, e.g., id.*, 78-80.[4] These assertions misapprehend and mischaracterize Banerjee's disclosure, which teaches outputting a hardware integrated circuit design in the form of a hardware description file.

The Petition asserts that limitation 1[d] is met because:

> Banerjee's tool outputs the software specific to the designed system and application thereof and based on the selections identified in 1[a]. EX-1003, ¶¶375-380. The output software includes: (1) the operating system; (2) application software; (3) application specific instruction

[4] The Petition does not raise substantial additional arguments for the corresponding limitation of the other challenged independent claim, 8[e]. Pet., 83-84.

extensions; and (4) "middleware" (such as I/O device driver software).

Pet., 78. The Petition cites Banerjee at [0038], [0039], [0045], [0046], [0056], [0067], [0078], and [0084] for these assertions. The Petition further cites Banerjee Figure 5, asserting that supposedly "Fig. 5 exemplifies how the operating system software is compiled and output," and that "OS software is generated ***and output***" (emphasis added), citing Banerjee at [0075], [0081], and [0083]. Pet., 78-79. Finally, the Petition cites Banerjee at [0038] and [0086] for the proposition that "Banerjee's tool generates and outputs software instructions related to the application specific instructions." *Id.*, 79-80.

As discussed below, none of these portions of Banerjee discloses the output of software as claimed. In accordance with the discussion above, Section II.A.2.b, they only describe Banerjee's process of using the desired attributes of what software will be typically executed on the desired IC design in order to optimize the outputted hardware design of the IC, ***without*** producing any software to run on that IC after it is fabricated. Even assuming, *arguendo*, that Banerjee outputs a hardware IC design meeting the recited output in limitation 1[c], Banerjee outputs no software that meets limitation 1[d]. *See* Ex.-2001 [McAlexander] ¶¶45-46.

The descriptions of the use of application software and operating system in Banerjee may address the attributes and requirements of executing such software

on the resulting IC, but they do not describe any software ***output*** by Banerjee that is "executed on the hardware integrated circuit for controlling the electronic device[.]" The only output described in Banerjee is a hardware description file: in other words, a file containing the hardware IC design. Banerjee does not disclose, or render obvious, the output of software executable on the designed IC to control the device. *See id.*, ¶¶47-48.

i. Banerjee Is Not Shown To Deliver Software.

Relying on its declarant and citations to broad swaths of Banerjee's disclosure, the Petition repeatedly characterizes Banerjee as providing "output software" that is "specific to the designed system and application thereof," including "the operating system" and "application software" for the IC being designed. Pet., 58-59, 78, 79. These unsupported assertions misapprehend or mischaracterize what Banerjee does. Banerjee does not output software, let alone software programming commands, let alone the "software programming commands to be executed by the hardware integrated circuit for controlling the electronic device" as claimed.

As Mr. McAlexander explains:

> The Petition quotes the following statement in Banerjee as supposedly disclosing outputting software as claimed:
>
> > [M]odule 24 outputs C or C++ code that configures the Windows, Linux or RTOS operating system by

> outputting header files, switches, and a modified makefile. …. Module 24 then determines the optimal configuration for the design by selecting the processor that most closely matches the specified parameters and providing any additional functionality in software either at the application, middleware or operating system level.

Pet., 78 (quoting Ex.-1005 [Banerjee] [0078]). This disclosure in Banerjee does not describe outputting any software programming commands executable on the designed IC recited in the claims.

The claims require (referring specifically here to claim 1) not only "outputting by the processor of a hardware integrated circuit design" (limitation 1[c]), but also "outputting ... software programming commands to be executed by the hardware integrated circuit for controlling the electronic device" (limitation 1[d]). Ex.-1001 ['899] cls. 1, 8. In contrast, and as discussed previously, Banerjee only delivers a hardware description file, but not software to be executed on that IC. *See* Ex.-1005 [Banerjee], *e.g.*, [0040]; Figures 2 and 6.

Banerjee's descriptions of outputting its IC design are never accompanied by any suggestion that Banerjee is also, with that hardware design, outputting executable software for that designed hardware IC. Banerjee's disclosure of what it outputs includes only a hardware description file or "netlist," and does not output any software.

For example, Banerjee describes at [0044] module 25 of Figure 2 compiling and optimizing to "generate a gate-level netlist

representation of the device." (Here, "the device" clearly refers to the IC being designed by the tool: it is not referring to an integrated circuit that will be integrated into the designed IC device to control that device.). Notably, Banerjee's description "of a generalized method of the" tool in Banerjee, at Figure 6 and [0085]-[0093], does not mention outputting software. *Id.*, [0086], [0093]. When Banerjee here states that "the design process is completed," it has described only a hardware "device" design that is suitable to be sent "to a foundry." *Id.*, [0091] ("Preferably, only a hardened design is exported to a foundry at the conclusion of the design process as indicated at 67."), [0093].

The Petition quotes Banerjee's statement that Banerjee's "module 24 outputs C or C++ code that configures the Windows, Linux or RTOS operating system by outputting header files, switches, and a modified makefile." Pet., 78. But this statement in Banerjee merely confirms in even more specific terms that what Banerjee's module 24 produces—for internal use in the tool only, in any event—is not software, as the Petition appears to mistakenly assume. As the ordinary artisan at the time of the '899 Patent invention would have recognized, Banerjee's references here to "header files," switches" and "makefiles" are not references to software, because "header files," "switches," and "makefiles" do not refer to software. Rather, they refer to information to facilitate developing software. To summarize these briefly:

"***Header files***"—Header files contain information that is needed by several different files or functions. A header file contains C-

language definitions and structures. Centralizing information into a header file facilitates the creation and update of programs. *See, e.g.*, Ex.-2006 [IBM-Header Files] 1.

"***Switches***"— "Switches" here refers to switch statements, which transfer control to different statements within a switch body depending on the value of the switch expression. They may be thought of as a concept or piece of information akin to an "if…else…if" ladder. *See, e.g.*, Ex.-2007 [IBM-SWITCH] 1.

"***Makefile***"—A makefile sets rules, typically a set of rules, to determine which parts of a program need to be recompiled, and issues commands to recompile them. Such a makefile is a way to automate software building procedures and other complex tasks with dependencies. *See, e.g.*, Ex.-2008 [UND-Makefile], *e.g.*, 1-3; *see also, e.g.*, Ex.-2009 [Learn Makefiles] 1 ("Makefiles are used to help decide which parts of a large program need to be recompiled.").

Ex.-2001 [McAlexander] ¶¶49-57.

ii. The "Output" Petitioner Cites Is Data Transmitted Between Internal Modules Of Banerjee's Tool, Not Executable Software Outputted From Its Tool.

When the Petition quotes Banerjee's use of the term "outputs," it appears to insinuate that Banerjee thereby discloses "output[ting]" software alongside the hardware design, as recited in the claims. *E.g.*, Pet., 78 ("'Module 24 outputs C or C++ code...'") (quoting Ex.-1005 [Banerjee] [0078]). However, such "outputs" refer to information produced in Banerjee's optimization process that is never

output from the tool and is "output" only from one internal module into other internal modules as part of the hardware IC design process.

As Mr. McAlexander explains:

> Banerjee's statement that "module 24 outputs C or C++ code" (emphasis altered from Petition) refers to the optimization module itself during the process of designing the hardware IC. Banerjee specifically says (and is even quoted in the Petition as saying) that ***after*** this module "outputs" this "code," Module 24 "***then*** determines the optimal configuration for the design by selecting the processor that most closely matches the specified parameters and providing any additional functionality in software either at the application, middleware or operating system level." Ex.-1005 [Banerjee] [0078]. Here, "the design" refers specifically to the IC for which the hardware description file is being created, as shown by Banerjee's statement that it is using these "specified parameters" of a functional specification in "selecting the processor," which is hardware, not software.
>
> To repeat, the claims of the '899 Patent require, in summary, outputting ***both*** (i) a hardware IC design, ***and*** (ii) software programming commands executable on the designed IC to control the electronic device whose characteristics formed a basis for the IC's design. *See* Ex.-1001 ['899] cls. 1, 8; *see also id.*, 5:3-5:8, 5:13-5:14, 9:61-9:62, 13:20-13:23. In contrast, Banerjee does not disclose outputting or delivering any executable software.

The reference in Banerjee to outputting "code" does not mean that Banerjee is disclosing outputting from its tool executable software programming commands as recited in and required by the claims. Rather, it is a reference to, specifically, "outputting header files, switches, and a modified makefile" corresponding to the "operating system." Ex.-1005 [Banerjee] [0078]. These files contain information to facilitate the creation of programs in the future. They are not software programs themselves. It is a reference to establishing the necessary "parameters" for the software functionality that the IC must be designed to support, as part of "the optimization process" for designing the IC. *Id.*

This is further confirmed by Banerjee's next paragraph, which clarifies that this is a hardware design selection process, not an outputting of executable software for the IC. As Banerjee explains in the next paragraph, this "optimization process" is based on parameters of an "operating system footprint" used to optimize the IC design:

> To illustrate the optimization process, consider the process of multiplying two numbers. In one design, the processor can include a high performance hardware multiply unit. In another embodiment, the design may simply bring in a floating point emulation repository that the user could explicitly access by selecting a floating point unit or not. In yet another embodiment the designer may simply specify performance (rate of megaflop), or go a step farther and give computer program 15 an

> application to profile and to select the optimal configuration.

Ex.-1005 [Banerjee] [0079]; *see also id.*, [0078] ("By way of illustration, the designer can provide module 24 with a range of possibilities such as 1 megabyte of memory, restrict the processor to be no bigger than 3 mm^2 while meeting certain performance metrics."). Each of these descriptions in Banerjee concern optimizing the IC's hardware design, not writing or outputting software that will run on the IC to control a device after the IC is fabricated.

The Petition's citations to Banerjee at [0038], [0045], [0046], [0056], [0067], [0075], [0081], [0083], and [0086], Pet., 78-79, do not change this conclusion, but, rather, confirm and reinforce it. Banerjee at [0038] and [0086] describes how to optimize ***hardware*** "processor design" (such as "reduc[ing] the processor die size and gate count") based on desired hardware and software characteristics, such as "determin[ing] whether it is preferable to implement certain functions in software rather than hardware." It does not describe writing, or outputting from its tool, software programming commands.

Meanwhile, Banerjee at [0045]-[0046] and [0056] describes hardware design choices such as "cache line size" and generating an "HDL file descriptive of the design," where the "H" stands for hardware, not software, *id.*, [0004], and modifying this HDL file based on the desired "features and functionality" and "storage requirements" of the expected "application" software and operating system code. Here too, there is no description of outputting software to execute on the IC thus designed. Banerjee at [0067], [0075], and

[0081] describes how Banerjee's optimization process "provides quantitative information on the application software including code structure, critical components, risk areas etc." by selecting attributes based on the configuration of software which the hardware will need to "operate in conjunction with"; again, existing software requirements are quantified or characterized for optimization purposes, but no creation or producing of a tool output of software for execution on the new IC is disclosed. At [0067]-[0069], Banerjee further refers to workload analysis for identifying "instructions" that may be executed by the to-be-designed processor to, for example, "achieve the intended results quickly and efficiently." Yet Banerjee makes clear that this "identif[ication]" of such "instructions" is not the generation or outputting of executable software programming commands; rather, this simply describes an analysis of "the optimal configuration" of the hardware, such as its DSP "blocks," to achieve potential "savings in power and die area," reflected in "the description file" in hardware description language. *Id.*, [0069].

Next, Banerjee at [0083], describes how Banerjee's optimization module, "based on a prototype description," compiles the characteristics of the typical or "baseline" "operating system configuration," to be executed on the IC once it is fabricated, in order to analyze its expected "footprint" on the IC and how it is expected to be instantiated on the device. Ex.-1005 [Banerjee] [0083]; *see also id.*, [0072]-[0074].

Petitioner's characterization of these disclosures, as output of software programming commands to be executed by the hardware

> integrated circuit for controlling the electronic device in accordance with the selection of categorized input and output characteristics of the electronic device, is not correct. Pet., 78-80. Banerjee uses the attributes and requirements of software that is expected to be run on the IC under design to optimize that design prior to outputting the hardware related design. But Banerjee only outputs a hardware description file. *See* Ex.-1005 [Banerjee], *e.g.*, [0044], [0069]. It does not output newly written (or rewritten) software. *See id.*, [0084] (explaining that Banerjee's tool helps avoid "having to rewrite" software that is already available).

Ex.-2001 [McAlexander] ¶¶58-67.

In short, Banerjee's tool examines the attributes of existing software to help optimize its outputted hardware design. It does not output software programming commands to be executed on that hardware design when it is fabricated.

For this reason, Banerjee is not shown to disclose limitation 1[d] of claim 1, or the materially similar limitation 8[e] of claim 8, the other challenged independent claim.

b. *The Limitation Is Not Shown To Be Obvious.*

Petitioner's Banerjee Ground alleges obviousness based on a single reference: Banerjee. Pet., 57-60 (contending that "Banerjee discloses all features of the Challenged Claims" and "discloses, and/or renders obvious, the Challenged Claims") (capitalization altered).

The Petition presents no argument that it would be obvious to modify Banerjee, alone, to add limitation 1[d]/8[e] if Banerjee does not disclose that limitation. Pet., 78-80. Instead, Petitioner only contends that limitations 1[a], 1[b], and 8[a] would be obvious. *Id.*, 69-72 (alleging that limitation 1[a] is "obvious"), 76-77 (same, 1[b]), 80 (same, 6), 81 (same, 8[a]). Therefore, the Petition makes no showing, and does not even attempt to make a showing, that Banerjee renders obvious missing limitation 1[d]/8[e] if it is not disclosed.

Accordingly, the Banerjee ground is not shown to disclose or render obvious limitations 1[d] and 8[e], and for at least this reason, Petitioner's Banerjee Ground is not shown to have sufficient merit to support institution under 35 U.S.C. § 314(a). Ex.-2001 [McAlexander] ¶¶68-69.

B. The Rompaey Ground Fails (All Claims, Ground 1)

The Rompaey Ground fails for, at least, fundamentally similar reasons that the Banerjee Ground does. Just as the Banerjee Ground relies solely on Banerjee for limitations 1[d] and 8[e], the Rompaey Ground relies solely on Rompaey for these limitations. Pet., 48-50. And like Banerjee, Rompaey is not shown to output by its tool software programming commands to be executed on an outputted hardware design, as claimed.

1. The Patent Discloses A Different Tool Than Rompaey.

The Rompaey Ground (ground 1 in the present Petition) alleges that the challenged claims are obvious in view of Rompaey alone, or in the alternative solely as to the preamble of Claim 1, obvious in view of Rompaey in combination with CoWare I. Pet., 16-60; *see id.*, 37-40 (citing CoWare I as to cl. 1 preamble). But Rompaey, an antiquated reference from 1996, contains no description of now-common types of ICs such as SoCs. Rompaey and the '899 patent, which dates from approximately two decades later, differ, if anything, even more fundamentally than Banerjee and the '899 patent.

As already discussed above, the '899 patent describes and claims a method for designing a hardware integrated circuit, which controls an electronic device when integrated into the device, and associated software executed by the integrated circuit to control the device. *Id., e.g.,* cl. 1; cl. 8; *see supra* Section II.A.1 (describing more details and advantages of the '899 invention).

In a way similar to Banerjee above, Rompaey discloses a tool that fundamentally differs from the '899 patent's invention. Rompaey's design tool outputs a hardware design based on a different design methodology than the '899 patent but, most critically here, does not output software, and certainly not software programming commands to be executed by the hardware integrated circuit for controlling the electronic device in accordance with the selection of

categorized input and output characteristics of the electronic device, as claimed. Ex.-2001 [McAlexander] ¶¶70-72.

Rompaey discloses a design tool that uses a convoluted "encapsulation" scheme, in connection with "a data-model," to design for heterogeneous systems with heterogeneous specifications and outputs a "VHDL" file. Ex.-1004 [Rompaey] Figures 2, 6, 1:54-60, 2:42-46, 6:9-18; Ex.-2001 [McAlexander] ¶73. The references in Rompaey to a "VHDL" type of file refers to a hardware description file describing the IC design. Ex.-2001 [McAlexander] ¶73. Indeed, Rompaey expressly states that "VHDL" refers to "Very High Speed Integrated Circuit (VHSIC) Hardware Description Language," which, as Rompaey explains, is a "hardware description language." Ex.-1004 [Rompaey] 1:54-1:60; Ex.-2001 [McAlexander] ¶73.

Multiple figures of Rompaey depict this "VHDL" as the sole output from the design tool.

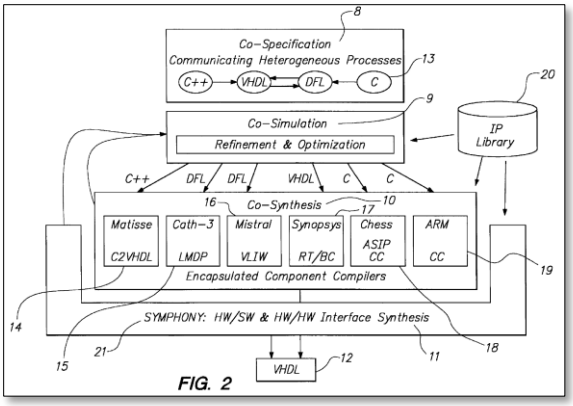


FIG. 2

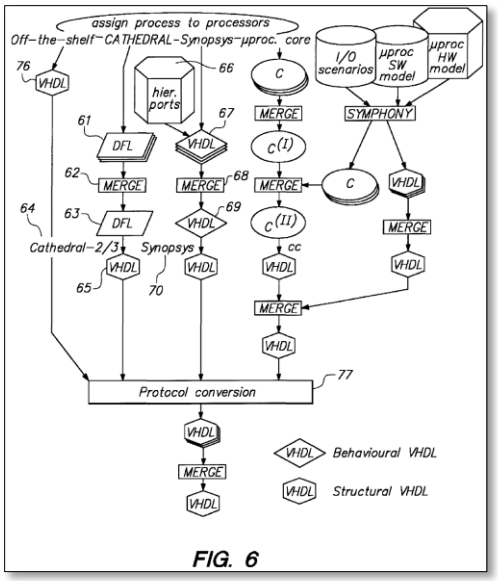


FIG. 6

Ex.-1004 [Rompaey] Figures 2, 6

As Mr. McAlexander explains:

Rompaey defines a hardware/software co-design as a digital system containing both high and low data-rate blocks in the signal path, where the high data-rate blocks are synthesized in hardware and low data-rate blocks are implemented on programmable processors. Ex.-1004 [Rompaey] 4:9-14. Rompaey also discloses that "hardware subsystems comprise any one or more of processor cores, off-the-shelf components, custom components, ASIC, processors, and boards." *Id.*, 6:49-53. The software subsystems comprise machine instructions for the off-the-shelf programmable processors. *Id.*, 5:35-37, 6:49-53, 18:25-27. Machine instruction or machine code is an instruction for the processor to perform a specific task in a digital binary form, *e.g.*, long sequence of zeros and ones. Thus, throughout Rompaey's disclosure, references to software subsystems and software generally

concern machine instructions or codes for off-the-shelf programmable processors.

Rompaey explains that hardware and software may be "designed separately" or else "co-design[ed]," and that when they are, the "hardware components" may be designed "in a specific hardware description language such as Very High Speed Integrated (VHSIC) Hardware Description Language (VHDL) or VERILOG for the hardware components and a software description language such as C or assembler for the software components." *Id.*, 1:12-16, 1:54-62. Rompaey is directed to a "hardware/software co-design environment" involving "a simulation of the digital system" to assist creation of a "specification" for "an heterogeneous essentially digital system," "the specification having hardware and software aspects." *Id.*, 6:6-9, 6:18-36, 8:48-58, 9:14-15. Rompaey further discloses that its "design environment ... includes means for creating processor models of the hardware components as objects in the database, the models comprising software models representing the software views on the hardware components and hardware models representing the hardware views on the hardware components." *Id.*, 8:4-10….

Rompaey describes a "specification having hardware and software aspects," in which "the implementation comprises hardware and software subsystems of the system, where the software subsystems are executed on one or more of the hardware subsystems." *Id.*, 8:48-58, 9:16-24. Here, the "software subsystems" do not refer to application software, operating system, or software programming commands, but to machine instructions to program a programmable

> processor, which is a type of processor having hardware that is programmable by machine instructions to operate in different ways. *Id.*, 6:51-53, 9:25-33. While Rompaey notes that this "implementation comprises hardware and software subsystems," Rompaey's tool does not output both a hardware design and software executable on that hardware, as the claimed invention requires. Rather, Rompaey outputs only "a net list comprising the layout information of the implementation"—which refers to a hardware IC design, rather than software that may be executed on a designed IC to control a device. *Id.*, 9:16-24.

Ex.-2001 [McAlexander] ¶¶74-76.

Rompaey refers to a "software tool" called "SYMPHONY." The Petition repeatedly cites "SYMPHONY" in connection with Rompaey's tool supposedly outputting software programming commands. Pet., *e.g.*, 48 (asserting that "The discussion in *[sic]* element 1[b] above regarding "SYMPHONY" highlights this output of software specific to the designed software *[sic]*."), 49. Rompaey discloses, however, that "SYMPHONY" provides "a methodology for interface synthesis whereby every communication channel is refined by selection of a communication scenario." Ex.-1004 [Rompaey] 12:36-39. It is part of the hardware/software co-design tool running a "model" of the target processor and a library of possible I/O scenarios for that processor concerning drivers and hardware interfaces for programmable processor designs. As Mr. McAlexander explains:

> Rompaey discloses that, where the processor is "a programmable processor, the implementation of inter-process communication comprises the steps of generating device drivers and of generating hardware interfaces. A software tool is used to achieve this. This tool is called SYMPHONY[.]" *Id.*, 17:59-63. SYMPHONY runs a "model" of "the target processor and a library of I/O scenarios ... for that processor," allowing the processor to be simulated. *Id.*, 17:65-18:12. Rompaey teaches that this software modeling tool, by supplying "performance figures," "will allow the designer or SYMPHONY to make a decision about what I/O scenario to use for which port." *Id.*, 19:13-16. In other words, SYMPHONY will model the software subsystem and I/O scenarios to provide performance figures that can help inform the hardware design of the IC, although, as Rompaey explains, "the user can always override the proposed assignment." *Id.*, 19:54-55. Further, "[t]he result of SYMPHONY is a refinement of the original host language encapsulation into a CoWare encapsulation of which the behavioral interface is identical to that of the original encapsulation. SYMPHONY effectively replaces a software encapsulation by a hardware encapsulation that has equivalent functionality." Rompaey, 20:26-31, Figure 7.

Ex.-2001 [McAlexander Decl.] ¶¶77-78.

In sum, Rompaey's cited mentions of "software" do not suggest, contrary to what the Petition seems to assert, that Rompaey outputs software to execute on a new hardware IC design. Rompaey outputs a hardware IC design in VHDL. All

that the cited teachings of Rompaey describe is delivering a hardware description. The references to "software" are not references to generating software programming commands executable on a designed hardware IC. Adds Mr. Alexander:

> Contrary to what the Petition suggests, Rompaey does not disclose outputting software for execution on a newly designed IC. Nor does Rompaey disclose writing or outputting software programming commands for controlling any device with the designed IC.

Id. ¶80.

As already described above, the challenged claims of the '899 patent recite, briefly put, outputting a hardware IC design ***and*** software programming commands executable on that IC. Since Rompaey delivers only a hardware description file, Rompaey, in ways similar to Banerjee, fails to disclose or render obvious all limitations of the claims.

2. Petitioner Fails To Show That Rompaey Discloses Or Renders Obvious "Outputting ... Software Programming Commands To Be Executed By The Hardware Integrated Circuit For Controlling The Electronic Device…" (Limitations 1[d] & 8[e])

Although Rompaey differs from the claimed invention in many significant ways, the present Preliminary Response will again focus on the software output limitations 1[d] and 8[e], and the Petition's failure to show that it is disclosed or rendered obvious.

a. *The Limitation Is Not Shown To Be Disclosed.*

The Petition asserts that Rompaey discloses the output "software programming commands" of limitations 1[d] and 8[e] because supposedly "Rompaey discloses that CoWare outputs software programming commands to be executed by the hardware circuit," referring to the "SYMPHONY" tool discussed in Rompaey. Pet., 48-49.[5] As with Banerjee, the Petition's arguments misapprehend or mischaracterize Rompaey's cited disclosures. Even assuming, *arguendo*, that Rompaey outputs a hardware IC design that may meet the recited IC design output, Rompaey outputs no software that meets limitation 1[d]/8[e].

As Mr. McAlexander explains:

> The descriptions of the use of "SYMPHONY" in Rompaey are not referring to the output of the design process, but the "modeling," partway through the hardware design process, which is intended by Rompaey to inform decisions how to assign, *e.g.*, the ports of programmable processors.
>
> Although the Petition contains partial quotations of Rompaey that refer, for example, to "software," Pet., 49 (quoting Ex.-1004 [Rompaey] 25:29-26:49), when these portions of Rompaey are read in full, they make it perfectly clear to artisans that when Rompaey refers

[5] The Petition relies on no substantial additional arguments for the corresponding limitation of the other challenged independent claim, limitation 8[e]. Pet., 54.

to the "software" on the chip to be designed, it is referring to such "software" in the context of what is "in" a "model." *See* Ex.-1004 [Rompaey], *e.g.*, 11:12-35.

Rompaey's "model" includes, *inter alia,* "processes," providing "modularity" in the model, which "are used to describe the system components." *Id.*; *see also, e.g., id.*, 8:4-8:38. "A process is a container for a number of host language encapsulations of a component." *Id.*, 11:40-11:41. Rompaey's use of software-sounding terms such as "processes," are not references to executable software programming commands, like the claims require; rather, they refer to parts of a "model" that Rompaey teaches can create, at an abstract level, "a description of the component," facilitating among other things simulation of "the communication behavior of the system," including modeling of the programming of these components in particular programming "languages," such as "C", and how they will communicate using so-called "ports." *See, e.g., id.*, 8:4-8:38, 11:39-11:57, 12:57-61, 14:41-14:46, 16:61-16:65, 17:3-17:7.

Rompaey teaches:

> By using primitive channels, ports, and protocols, the designer first concentrates on the functionality of the system while abstracting from terminals, signals, and handshakes. Once the designer is convinced that the processes of the system are functionally correct, the communication behavior of the system can be refined.

Id., 14:41-46; *see also id.*, 16:42-65. In sum, Rompaey's references to "channels," "ports," "protocols," and "processes" are all references to

what is being modeled in Rompaey partway through its design process, and not to actual hardware or executable software.

In the model, these "ports" are not actual physical ports, any more than the "processes" are actual software processes. Rather, in Rompaey's model, the "[p]orts are objects through which processes communicate." *Id.*, 12:57; *see, e.g.*, *id.*, 8:4-8:14 (describing Rompaey's "design environment" as including "processor models of the hardware components as objects in the database," with "I/O scenario models of the ports as objects in the database, the I/O scenario models representing the implementation of the ports on the hardware components"). "A primitive port is characterized by a protocol and a data type parameter." *Id.*, 12:57-59; *see also id.*, 12:64-67 ("Protocols define the communication semantics of a port.... Each primitive protocol indicates another way of data transport."). "For example, the primitive protocol used to model the memory port of a processor will have an index to model the address of the data that is put on the memory port." *Id.*, 13:15-18; *see also, e.g., id.*, 15:31-37 ("A primitive protocol determines the communication semantics, but not the communication implementation; it does not fix the timing diagram used in the communication.... Hierarchical protocols are high level models for alternative implementations of a primitive protocol: they preserve both data direction and control direction of the primitive protocol."). And, for example, "[a] channel is a point-to-point connection of a master port and a slave port." *Id.*, 13:41-42; *see also id.*, Figure 7, 17:65-18:46 (further describing Rompaey's use of SYMPHONY "software model," discussed further below).

Ex.-2001 [McAlexander] ¶¶83-88.

b. *Rompaey Models Software, But Does Not Output It.*

Importantly, none of this modeling of the designs in Rompaey leads to any output of executable software, let alone software programming commands to be executed on a designed hardware IC to control a device.

As Mr. McAlexander further explains:

> Rompaey's "model" can be used to create "a representation of the system" and its results can be used "to analyze, manipulate, and refine the system description." Ex.-1004 [Rompaey] 15:50-57. Once this modeling, simulation, or "synthesis" of the hardware and software behavior is complete, "[t]he final step is to link all the synthesis and hardware descriptions to drive commercial back-end tools to generate layout." *Id.*, 12:44-49. "Layout" denotes hardware description, not software programming commands. This final step is shown in Rompaey's Figure 2: the output deliverable is an IC design in hardware description language (feature 12).
>
> This modeling in Rompaey's design process is a "model," not the real thing. *See id., e.g.*, 13:42-51 ("Two ports that are connected by a channel can exchange data.... A primitive channel provides for unbuffered communication.... In software it is implemented with a (possibly in-lined) function call. In this way, primitive channels model the basic communication primitives found back in software ...."). Rompaey does not output or deliver any software programming commands executable by a hardware IC being designed by the system.

Ex.-2001 [McAlexander] ¶¶89-91.

The Petition cites a series of disclosures in Rompaey as allegedly showing that Rompaey meets this limitation, but when one looks past the Petition's selective quotations of these disclosures and reads what Rompaey actually says, it becomes clear that each of the descriptions cited by the Petition are of software modeling or hardware description language, not executable software like that recited in limitations 1[d]/8[e]. *Id.*, ¶92; Pet., 48-49. As Mr. McAlexander explains:

> Leaving aside for the moment the Petition's references to Rompaey's "SYMPHONY" tool, which I will discuss below, the disclosures in Rompaey that the Petition, at 48-49, cites as supposedly describing the output of executable software programming commands, as recited in the claims, all refer to either hardware descriptions or models of potential characteristics of software that might be used on the IC that is to be designed. None of them refers to executable software delivered as an output from the tool.
>
> For example, I will list specific citations the Petition makes to Rompaey, and what they actually describe:
>
> i. "Fig. 7" and "6:45-53" (Pet., 48)
>
> Figure 7 of Rompaey literally shows a software ("SW") "model." As for "6:45-53," it describes the tool or tools disclosed by Rompaey, which includes "hardware subsystems and software subsystems," which may include "machine instructions," or binary

programming for programmable processors. But this does not describe what outputs the tool delivers and does not change the fact that Rompaey ultimately outputs only a hardware description file. Ex.-1004 [Rompaey], *e.g.*, Figures, 2, 6.

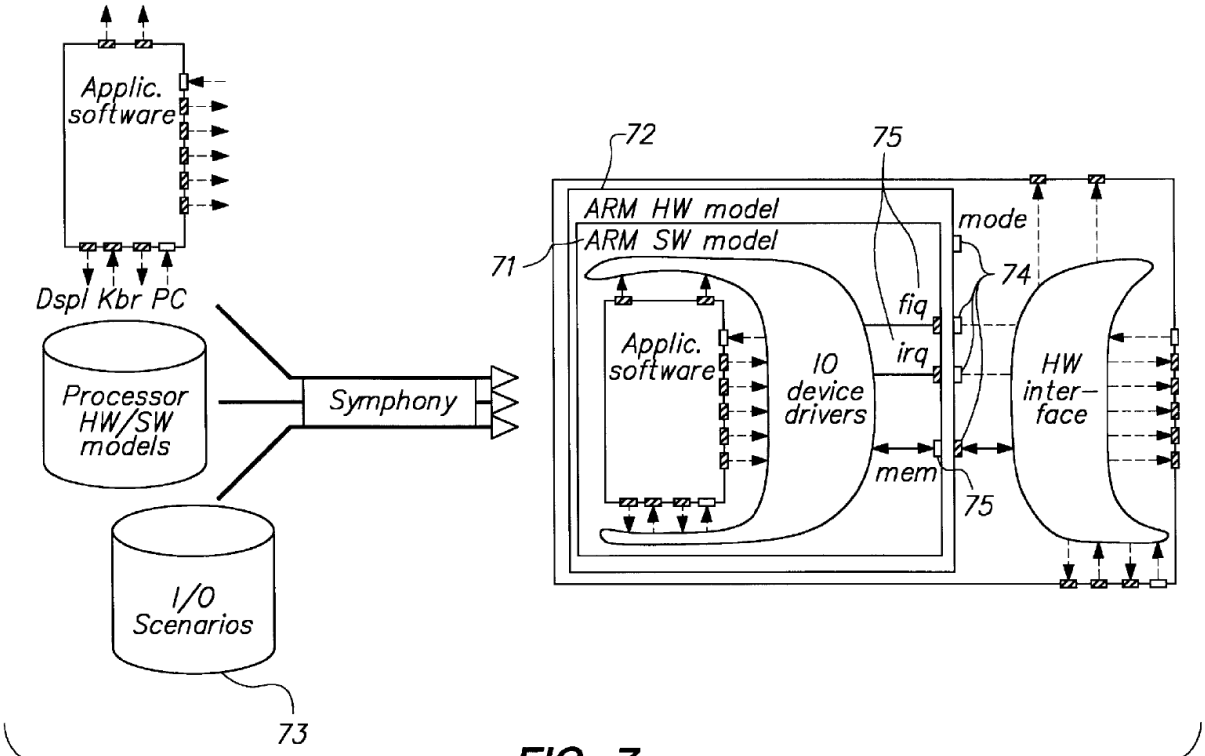


FIG. 7

ii. "Figs. 2, 6-7, 9, 18, 25:29-26:49" (Pet., 49)

Figure 7 was discussed above, and 25:29-26:49 discusses simulation or modeling with "SYMPHONY," discussed below. Figures 2 and 6 clearly show hardware description language as the only output of the process shown. Figure 9 is a "co-simulation" that elaborates on "the method of FIG. 2," describes a "library" rather than executable software and is not shown to disclose outputting software of any kind. *Id.*, *e.g.*, 10:12-14. Figure 18 shows a feedback "loop," based on modeling of the system, with the end result being "a gate

level net list (for a VHDL simulation), or a net list," all of which describe hardware, and are not the software executable on the designed hardware required by the claims. *Id.*, 26:50-28:29.

iii. "8:4-38; 26:14-30; claims 6-8, 27-37; TITLE; Abstract" (Pet., 49)

Rompaey at 8:4-38 describes modeling of ports, not executable software outputs. Rompaey at 26:14-30 is discussed above. Rompaey's cited claims are describing Rompaey's tool, not the ***outputs*** of that tool. Rompaey's title and abstract, like the rest of its written description, nowhere state that Rompaey generates or outputs software programming commands executable on the hardware designed in light of Rompaey's modeling and simulation of what that hardware might be used for.

Ex.-2001 [McAlexander] ¶¶93-97.

In short, none of the assorted citations to Rompaey in the Petition for limitation 1[d] or 8[e] disclose outputting software programming commands to be executed on the hardware designed in Rompaey's process, let alone for controlling a device whose characteristics formed a basis for that design. *Id.*, ¶98.

c. *"SYMPHONY" Does Not Output The Claimed Software.*

The Petition asserts that Rompaey's discussions of its "co-design" tool, "SYMPHONY," highlight[] this output of software specific to the designed software." Pet., 48. No such "output of software" occurs.

The Petition's reference to "software specific to the designed software" is difficult to make sense of. The limitation concerns outputting software executable on the IC, not the "designed software." Petitioner's statement is therefore irrelevant. And, even if we were to generously assume that Petitioner meant to say "designed hardware" rather than "designed software," such an assertion would have been unsupported by Rompaey's disclosure. These disclosures do not describe any output software programming commands by Rompaey that could be executed on the designed IC.

The only output described in Rompaey is a hardware description file: in other words, a file describing the hardware IC design. *See* Ex.-2001 [McAlexander] ¶99. As Mr. McAlexander explains, SYMPHONY does not output the software required by the claims:

> The Petition quotes Rompaey's statement that "In addition, SYMPHONY, adds RAM and ROM blocks to store the program code and data …." Pet., 48 (quoting Ex.-1004 [Rompaey] 20:22-25). This disclosure in Rompaey describes the use of SYMPHONY, partway through the hardware design process, to make decisions about adding hardware components; it says nothing about outputting software. "RAM and ROM blocks to store the program code and data" can store information, but they are ***hardware***, not software programming commands.

Nowhere does Rompaey say that it specifies or outputs "the program code and data" that will actually be stored in these components once the described designed IC has been fabricated. Nor could it: the only output described by Rompaey of this design process is a file in ***hardware*** design language, which could not specify such ***software*** program code and data even if that were desired.

The Petition further points to Rompaey's Figure 7, which it erroneously asserts "shows the output of hardware and the associated 'applic. Software' and 'IO device drivers' to interface with the display, keyboard, and program controls." Pet., 48. This interpretation of Rompaey's Figure 7 is not correct and seems to confirm that the Petition has misapprehended Rompaey's disclosure of modeling to refer to actual software.

Figure 7 of Rompaey clearly depicts these features as being within, and part of, the Rompaey process's "SYMPHONY" "modeling":

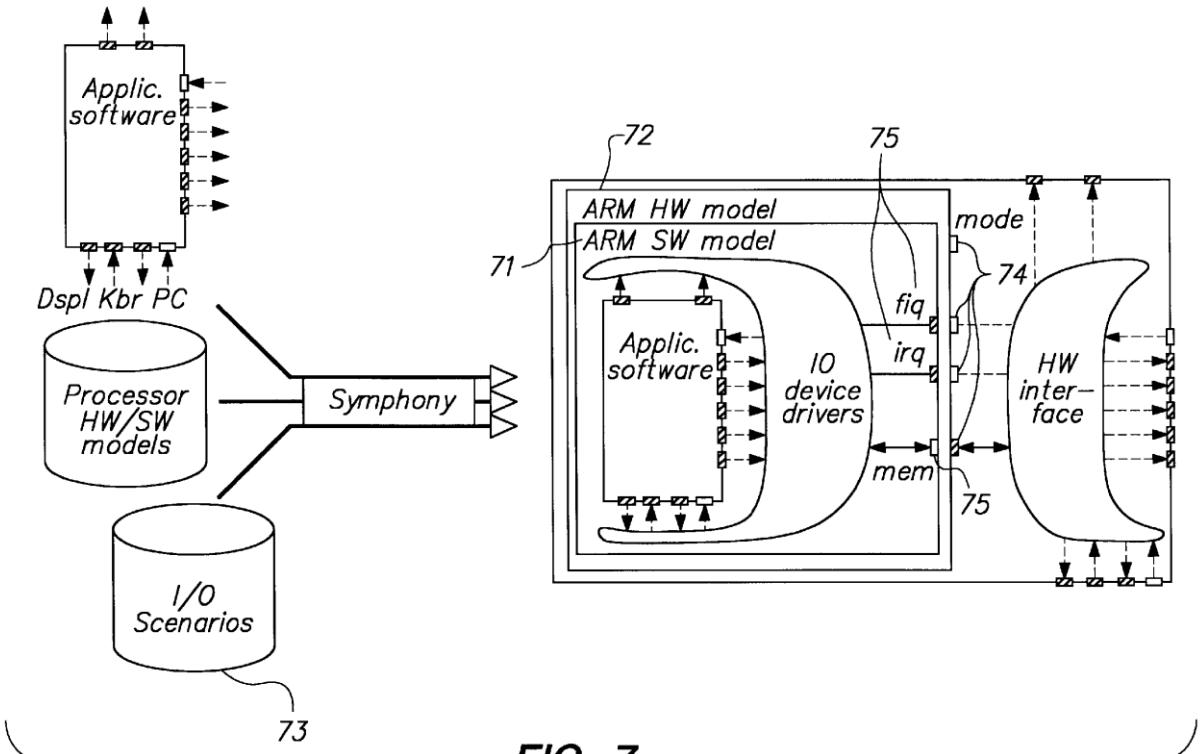


FIG. 7

As already mentioned above, Section II.B.1, this "software modeling" in Rompaey is performed to supply "performance figures" that "will allow the designer or SYMPHONY to make a decision about what I/O scenario to use for which port." Ex.-1004 [Rompaey] 19:13-16. The result of SYMPHONY is to replace the software encapsulation with a hardware encapsulation that has the same functionality. *Id.*, 20:26-31. In other words, SYMPHONY models the system and provides performance figures that can help inform the hardware design of the IC. It is not the end or output of the design process, and this is a model of the system, not the actual executable software itself.

The same goes for Rompaey's statement cited in the Petition of concerning "I/O device driver software." Pet., 49. Contrary to the Petition's insinuation, this is only a step toward generating what is

> actually the output of the tool: the hardware description file. It is not generated software programming commands to be executed on the described, designed IC for controlling a device.
>
> In sum, for each of the reasons above, the Petition's assertions that these disclosures in Rompaey are directed to outputted software that may be executed on the designed IC to control an electronic device are contrary to Rompaey's disclosure.

Ex.-2001 [McAlexander] ¶¶100-106.

d. *The Limitation Is Not Shown To Be Obvious.*

The Petition briefly argues that, if "Rompaey does not expressly disclose this element, Rompaey renders obvious this element," based on a single sentence of argument, that "Rompaey's teachings are devoted to converting the design of an 'application' (such as the 'pager application') into the hardware integrated circuits and software implementing that design." Pet., 49. The Petition does not suggest that adding CoWare I to the combination affects this question, but relies solely on Rompaey. This single-sentence argument is unsupported, and the Petition therefore fails to show that Rompaey renders obvious this missing limitations 1[d]/8[e].

As Mr. McAlexander explains,

> Rompaey does not disclose any outputting by the design tool of any software at all, let alone any software that is to be executed by the fabricated IC that is being designed. And more specifically, Rompaey

> neither discloses nor suggests outputting by the processor, as claimed, software programming commands to be executed by the hardware IC being designed for controlling the electronic device in accordance with the selection of categorized input and output characteristics of the electronic device.

Ex.-2001 [McAlexander] ¶108. Therefore, the Petition makes no showing that Rompaey renders obvious missing limitations 1[d]/8[e] if it is not disclosed. *Id.* ¶109.

Accordingly, the Rompaey ground, like the Banerjee ground, is not shown to disclose or render obvious limitations 1[d]/8[e], and for at least this reason, is not shown to have sufficient merit to support institution under 35 U.S.C. § 314(a).

III. THIS PETITION SHOULD BE DENIED UNDER *FINTIV* DUE TO THE ADVANCED STATE OF PARALLEL DISTRICT COURT LITIGATION.

As shown above, the petitioned grounds' merits do not meet the minimum threshold needed under 35 U.S.C. § 314(a). Moreover, institution should also, for a separate reason, be denied as a matter of discretion to "minimize potential conflict between the Patent Trial and Appeal Board (PTAB) and district court proceedings" and avoid counterproductive "inefficiency and gamesmanship in AIA proceedings, given the existence of parallel proceedings between the Office and district courts." USPTO, Director Memo re Interim Procedure For Discretionary Denials In AIA Post-Grant Prcdgs. With Parallel Dist. Ct. Litig. (June 21, 2022)

(Ex.-2010), 1. Exercise of this discretion is governed by, *inter alia*, the precedential decision in *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11, 5-6 (Mar. 20, 2020) (precedential) ("*Fintiv*"). Here, the District Court action as filed in March 2023, fifteen months ago. Arm waited ten months before initiating this IPR. By then, each side had already submitted a claim construction brief and the Markman hearing was just a month away. Since the IPR filings, predictably, the Markman order, the final contentions, discovery, and much more has occurred in the District Court.

Assigned PTAB panels may in their discretion deny *inter partes* review in light of advanced progression of parallel district court litigation regarding the same patent and similar issues. *See NHK Spring Co. v. Intri-Plex Techs., Inc*., IPR2018-00752, Paper 8, 19-21 (Sept. 12, 2018) (precedential). Precedent sets forth six non-exclusive factors for assessing whether discretionary denial in light of such parallel litigation is appropriate. *Fintiv* (precedential), 5-6. In analyzing them, the Board "takes a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review." *Id.*; *see also Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 15, 7-17 (May 13, 2020) (informative) ("*Fintiv II*"); Ex.-2010 [Interim Discretionary Denial Memo] 2.

Each of *Fintiv*'s six factors, and the collection of them all together, weighs in favor of denying the requested review in this case. For Factor 1, no stay of the

parallel District Court litigation has been granted (or even sought) and a stay by the court is extremely unlikely especially given the advanced stage of the case. For Factor 2, the District Court trial is scheduled to be tried long before—over ***five months*** before—the due date for a final written decision in this proceeding, and Petitioner has failed to show any significant likelihood that the Court's lengthy head start will disappear. For Factor 3, the parties and the District Court have already invested significantly in preparing for the court trial and that significance will only increase: the District Court held a Markman Hearing on February 27th and issued the claim construction order on March 26th, final infringement and invalidity contentions were served on April 23rd, the time to amend pleadings will soon end, and before any decision will be made on institution the parties and court will be many months into fact discovery—in fact, each party has already served and responded to a round of written discovery, each party has already served three third party subpoenas, and scheduling of June depositions is already underway. *See* Ex.-1015 [Docket-Scheduling-Order] 3. For Factor 4, there is perfect overlap in both the claims and art at issue, and Petitioner's limited *Sand Revolution* stipulation does little or nothing to reduce the concern of inefficiency and inconsistent rulings. For Factor 5, Petitioner challenges the patent both here and there. And for Factor 6, as Section II confirms, the Petition fails to present a challenge so "compelling" as to compel institution.

These factors are addressed in more detail, factor by factor, below.

A. *Fintiv* Factor 1: The District Court Has Not Granted A Stay And There Is Zero Evidence That A Stay Might Be Granted.

In the District Court proceeding involving the same claims, same patent, and same challenger, the District Court has not granted a stay pending this IPR, no stay has been requested, and there is zero evidence a stay might be granted.

On the contrary, the facts indicate that it is unlikely that the District Court would ever grant a stay given the advanced stage of the parallel District Court case. In short: the parties have exchanged final infringement contentions and invalidity contentions. *See*, *e.g.*, Ex.-2011 [ARM-Final-Invalidity-Contentions]. The *Markman* hearing was completed months ago, and the District Court issued a claim construction order on March 26, 2024. Ex.-1015 [Docket-Scheduling-Order] 3; Ex.-2012 [District-Court-Docket].[6] By the institution deadline (August 9), fact discovery will be over four months underway, the deadline to amend pleadings will have passed, and opening expert reports will be about to be filed in the following month. Ex.-1015 [Docket-Scheduling-Order] 3-4. Moreover, all other pretrial matters will occur long before any decision in this proceeding. The District Court jury trial is set for March 3, 2025—over five months before the expected date of

[6] A copy of the March 26 Markman Order, which addresses certain claim limitations of this Patent and the related '657 Patent, is submitted. Ex.-2019.

any final written decision, August 9, 2025. *Id.*, 5; *see* Section III.B, *infra*. Thus, even if Petitioner were to file a motion for a stay pending this proceeding in the future, it is highly unlikely Judge Albright would grant a stay.

This non-likelihood of any future stay request being granted is amply confirmed by the history of decisions denying stays by the district judge. *See, e.g.*, *Lone Star SCM Systems, LTD. v. Bluebird Inc.*, 6:21-cv-00844-ADA, 4-5 (W.D. Tex. Aug. 1, 2023) (Albright, J.) (denying motion to stay pending IPR where fact discovery recently opened and *Markman* hearing had yet to occur) (later stayed through joint stipulation); *Smart Mobile Techs. v. Apple Inc.*, 6:21-cv-00603-ADA, 3 (W.D. Tex. Aug. 8, 2023) (Albright, J.) (denying motion to stay pending IPR where *Markman* and open of discovery had yet to occur).

In sum, the District Court litigation has not been stayed, no stay has even been sought, and any stay is speculative and unlikely. Under *Fintiv*, these facts weigh in favor of finding that allowing this IPR to proceed in parallel with the well-advanced, soon to be tried District Court proceeding would create wasteful "inefficiency and duplication of efforts." *Fintiv* (precedential), 6. And, as noted below in Factor 3, the patentability disputes before the court should resolve all of "the patentability disputes between the parties" in this case as to this patent. *Id.*, 9. Factor 1 weighs against institution.

B. *Fintiv* Factor 2: The District Court Trial Will Occur Over Five Months Before The Deadline For A Final Written Decision In This Proceeding.

"If the court's trial date is earlier than the projected statutory deadline, the Board generally has weighed this fact in favor of exercising authority to deny institution under *NHK*." *Fintiv* (precedential), 9.

This factor weighs strongly against institution because the record shows that the court's trial date is well earlier than the projected statutory deadline in this case. There is an over ***five-month*** gap between the District Court's earlier jury trial date, and any expected later final written decision in this IPR. Here, the deadline for a final written decision will be approximately August 9, 2025, which may be extended to February 9, 2026, for good cause. 37 C.F.R. § 42.100(c). Meanwhile, the District Court litigation is set for trial on March 3, 2025. *See* Ex.-1015 [Docket-Scheduling-Order] 5; Pet., 86.

When applying this factor, "the PTAB has taken the 'courts' trial schedules at face value absent some strong evidence to the contrary." Ex.-2010 [Interim Discretionary Denial Memo] 8 (citing *Fintiv II* (informative), at 13); *see also NXP USA, Inc. v. Impinj, Inc.*, PGR2022-00005, Paper 18, 9 (May 2, 2022) ("*NXP*") ("We generally take courts' trial schedules at face value absent some strong evidence to the contrary.") (quoting *Fintiv II* (informative), 13). There is no "strong evidence to the contrary." *Id.* The timeframe set by the District Court is

consistent with the Federal Court Management Statistics for the Western District of Texas. *See* Ex.-2010 [Interim Discretionary Denial Memo] 3 & n.4 (encouraging use of these statistics in evaluating this factor). Those statistics indicate that, over the past six years, a civil case in the district was brought to trial on average 25 months after filing. Ex.-2013 [Trial-Statistics] 37 (indicating median times to trial for civil cases for the twelve-month period ending in December for 2018-2023 as 20.4, 27.9, 18.5, 25.5, 29.5, 28.3). Here, trial is set just under 24 months after the cases were filed, entirely consistent with this body of statistical evidence except that it is a month earlier than that median schedule; and even if the trial date were to move to the median value it would still be four months earlier than the earliest expected date of any final decision in this IPR. *See* Ex.-1015 [Docket-Scheduling-Order] 5 (trial set for March 3, 2025); Ex.-2012 [District-Court-Docket] (parallel case filed March 14, 2023).

In short, the District Court trial is set to occur ***five months*** before the deadline for a final written decision, and possibly nearly a year before the deadline if extended. Precisely such a five-month gap, and even shorter gaps, have been found sufficient for a denial of institution in the very recent past, confirming that *Fintiv* is fully consistent with this weighing of factor 2. *Int'l Bus. Machs. Corp. v. Digital Doors, Inc.*, IPR2023-00970, Paper 7, 10-11 (Jan. 12, 2024) (denying institution where "trial would occur about ***five months*** before the statutory

deadline"); *see Fintiv II* (informative), 13 (finding this factor weighed in favor of discretionary denial and denying institution where the district court trial was scheduled to occur ***two months*** before the deadline for the Board to reach a final written decision); *see also EClinicalWorks, LLC v. Decapolis LLC*, IPR2022-00229, Paper 10, 9 (Apr. 13, 2022) ("*EClinicalWorks*") (finding this factor weighed in favor of discretionary denial and denying institution where "the beginning of the jury trial in the WDTX Cases is roughly ***one or two months*** before any final decision would have been due had *inter partes* review been instituted"); *NXP*, 9 (finding this factor weighed in favor of discretionary denial "[b]ecause the currently scheduled trial in the parallel proceeding is scheduled to begin over two months before our deadline to reach a final decision"), *reh'g denied*, Paper 20 (Aug. 25, 2022).

Petitioner argues that the Board should ignore this set trial date of March 3, 2025 because "that is not a date certain." Pet., 86. This attorney argument is mere speculation by Petitioner. Petitioner offers no "strong evidence to the contrary" that the trial date as currently set will not go forward as scheduled, which, as noted above, is consistent with the most current time-to-trial statistics for the Western District of Texas. Ex.-2010 [Interim Discretionary Denial Memo] 8 (citing *Fintiv II* (informative), 13).

The only evidence Petitioner offers of a supposed likelihood that the trial date will meaningfully shift is weak. Instead of offering any statistics as the Director suggests might suffice to show such a probability, Ex.-2010 [Interim Discretionary Denial Memo] at 3 & n.4, Petitioner only relies on anecdote: pointing to two cases which were before the same District Court judge, Judge Albright, where the initial trial date was moved. Pet., 86; *see* Exs. 1016-1017. But Petitioner's proffered evidence is anecdotal (at best) and does not show this trial date is likely to be continued. Appropriate evidence to attempt to show the trial date is likely to be continued to after the likely date of a decision in this case would have included "the most recent statistics on median time-to-trial for civil actions in the district court in which the parallel litigation resides" and "additional supporting factors such as the number of cases before the judge in the parallel litigation and the speed and availability of other case dispositions." Ex.-2010 [Interim Discretionary Denial Memo] 8-9. But Petitioner points to no median time-to-trial statistics or evidence of the number of cases currently before the judge. Instead, it focuses on these two cherry-picked cases before Judge Albright which have already gone to trial and been disposed of in district court. Exs. 2015-2016. These case dispositions are not representative of the median time-to-trial in the district and there is no evidence that they at all approximate such a statistic. Instead, they are two outliers, selected by Petitioner to bolster its speculation. In fact, the *Sable*

and *Demaray* cases Petitioner cites vastly exceeded the median time-to-trial in the district. *Sable* (Ex.-1016) took 35 months to trial, while *Demaray* (Ex.-1017) took 47 months to trial. Given that the median time-to-trial in the Western District of Texas is approximately 25 months, no evidence even suggests that Petitioner's pair of cases represent a statistically typical timetable. Thus, not only does Petitioner fail to address the appropriate time-to-trial statistics, but the cases it cites fail to show "strong evidence to the contrary" that the current trial date would not go forward as scheduled, and so are irrelevant.

Nor are the cases Petitioner cites representative of the typical timeframe of trial before District Judge Albright. Statistics on the patent case management website Docket Navigator indicate that Judge Albright's median time to trial is 25.5 months, only slightly above the median time-to-trial of the Western District of Texas. *See* Ex.-2014 [Albright-Statistics].[7] There is no evidence that the parallel proceeding will be unduly delayed simply because it is before Judge Albright.

The reason Petitioner did not cite the median time-to-trial statistics is because the statistics do not support Petitioner's position; they support Patent Owner's position, strongly confirming that the District Court is very likely to go to trial many months before this IPR would issue any decision.

[7] https://search.docketnavigator.com/patent/judge/16961/15.

Even if this anecdotal-at-best evidence could be—and it is not—cognizable as "strong" evidence that this District Court case's large lead on this IPR petition might diminish, the two IPRs Petitioner cites are also factually distinguishable from the proceedings here, and those marked distinctions provide the obvious explanation for their out-of-the-ordinary times to trial, and make them all but irrelevant here. In *Sable Networks, Inc. v. Cloudflare, Inc.*, trial was continued multiple times, due to, *inter alia*, (i) a district court motion by *Sable*'s IPR petitioner for venue transfer (*see, e.g.*, Ex.-1016, Dkts. 71, 79), which if granted would inevitably have caused the trial date to move (since it would have moved the case to an entirely new district); (ii) delays in the district court *Markman* hearing due to the aforementioned venue transfer motion (*id.*, Dkts. 46, 54, 66, 71, 123, 125, 132); (iii) motions for leave to amend infringement contentions (*id.*, Dkts. 158, 161, 163); (iv)-(v) two separate motions for dismissal due to lack of subject matter jurisdiction (*id.*, Dkts. 306, 315); (vi) extensions on final contention and pleading deadlines (*id.*, Dkts. 78, 119, 157, 165), and (vii) multiple joint motions to correct scheduling order deadlines (*id.*, Dkts. 166, 170, 195, 201, 325). There are no such facts in the parties' District Court action here. No venue transfer motion has been filed or is expected; the *Markman* hearing has already been held and the order already issued last month; final infringement and invalidity contentions were served on April 23rd; no subject matter jurisdiction question has been raised; and

the parties have stipulated to the current trial date and neither has attempted to move that mutually-agreed date.

Moreover, unlike this case, *Sable* was ***never*** a case for *Fintiv* Factor 2. In *Sable* the deadlines for decision in all the related IPRs were three or four months before—not after—the originally scheduled trial date in its parallel district court action. So in *Sable*, the district court trial ***trailed*** the decision in the IPRs from the start—and every continuation of the scheduled trial only increased the IPR's lead. *See* IPR2021-00909, IPR2021-01005, IPR2021-00969, IPR2021-01067. That means that even under *Sable*'s originally scheduled district court trial date, *Fintiv* ***never*** favored denial under Factor 2—which, as seen above, expressly requires that "the court's trial date is ***earlier*** than the projected statutory deadline." *Fintiv* (precedential), 9. That entirely distinguishes *Sable* from the present case, where trial is currently set to begin five months before, rather than three to four months after, any expected decision in this IPR.

Petitioner's only other cited case, *Demaray v. Samsung Electronics*, featured a district court pretrial record even more slow and complicated than *Sable*'s, and even more unlike the record of the parallel district court action in the present case. *Demaray*'s parallel district court trial was continued multiple times due to, *inter alia*, (i) a venue transfer motion (*see, e.g.*, Ex.-1017, Dkts. 40, 82), a postponed *Markman* hearing (*id.*, Dkts. 33, 82), (ii) a need for multiple *Markman* hearings

(*id.*, Dkts. 119, 167, 263, 272, 287, 304, 313, 315, 333, 337, 341), (iii) multiple discovery hearings (*id.*, Dkts. 214, 216, 221, 288, 309, 341, 353), and (iv) multiple motions to amend the case schedule (*id.*, Dkts. 125, 177, 218, 259, 354). Once again, the parallel District Court action here is very different, and the facts which led to multiple trial continuations in *Demaray* are not shown. No transfer motions are pending, a *Markman* hearing has already been held with no indication that more are necessary, no hearings to resolve discovery disputes have been shown to be likely, and the trial date was stipulated by the parties with no evidence that later disagreement has arisen. There is no evidence that there will be similar delay here.

Here, "based on the evidence of record, there is no evidence to suggest that trial will be delayed." *Nine Energy Serv. Inc. v. NCS Multistage Inc.*, IPR2020-01615, Paper 18, 9 (Mar. 24, 2011) (finding that Factor 2 favored denial of IPR). The District Court trial is currently set to occur over five months before the earliest deadline for a final written decision in this IPR, consistent with the court's time-to-trial statistics over the last several years, and there is no evidence that this date is likely to change. This factor weighs heavily against institution. *See Fintiv II* (informative), 13.

C. *Fintiv* Factor 3: By The Time Of A Decision On Institution, A Claim Construction Order Is Expected, Final Infringement And Validity Contentions Will Be Complete, And Fact Discovery Will Be Approaching Completion.

Factor 3 relates to the "amount and type of work already completed in the parallel litigation by the court and the parties ***at the time of the institution decision***." *Fintiv* (precedential), 9. For example, "if, at the time of the institution decision, the District Court has issued substantive orders related to the patent at issue in the petition," factor 3 favors denial. *Id.*, 9-10. "Likewise, district court claim construction orders may indicate that the court and parties have invested sufficient time in the parallel proceeding to favor denial." *Id.*, 10.

The record shows that Factor 3 weighs against institution, because pretrial investment in the District Court will be far advanced by the time a decision is made whether to institute review in this case. The District Court's docket control order, entered August 25, 2023 and setting forth stipulated pretrial deadlines up to and including the March 3, 2025 scheduled trial, has set a fast schedule. The District Court already held a *Markman* hearing as scheduled on February 27, 2024—more than two months before this preliminary response. Petitioner has already served its final invalidity contentions. Ex.-1015 [Docket-Scheduling-Order] 3; Ex.-2012 [District-Court-Docket] 6; Ex.-2011 [Arm-Final-Invalidity-Contentions]. The claim construction order (which clearly qualifies as a "substantive order[]" under *Fintiv*, *see Fintiv* (precedential), 9-10) issued March 26, many months before a

decision on institution, which is due by statute August 9, 2024. Ex.-2012 [District-Court-Docket]. By that time, scheduled fact discovery will be approaching completion—nearly six months old with only one month remaining—and expert reports will be due the following month, adding large further investments by the parties under Factor 3. *See* Ex.-1015 [Scheduling-Order] 3-4. Furthermore, Petitioner's final invalidity contentions overlap with the grounds in this IPR, raising all the obviousness references herein (Rompaey, CoWare I, and Banerjee), as further discussed below. *See* Section III.D, *infra*. All of this investment weighs strongly against institution and ensures that this IPR will not be an efficient alternative to District Court litigation. The parties and the District Court will have invested significant time and resources in the District Court litigation even by the time the Board decides whether to institute.

As many panels have found, this *Fintiv* factor favors denial when parallel District Court litigation is this advanced, or even when it is less advanced than the one here. *See Fintiv* (precedential), 10; *Fintiv II* (informative), 13-14. *Fintiv II*, which is "informative"-designated precedent applying the *Fintiv* analysis, denied institution based in part on the finding that Factor 3 favored discretionary denial where "the level of investment and effort already expended on claim construction and invalidity contentions" was that the parties had exchanged final contentions, a *Markman* hearing had been held, the court had entered a claim construction order,

and discovery had begun. *Fintiv II*, 13-14. All of these things are also expected or already true here. *See also, e.g.*, *BOE Tech. Grp. Co., Ltd. et al. v. Element Capital Commercial Co.*, IPR2023-00808, Paper 9, 23-24 (Nov. 15, 2023) (denying institution in part because claim construction was complete and fact discovery was nearly complete); *Roku, Inc. v. IOEngine, LLC*, IPR2022-01551, Paper 11, 12 (May 5, 2023) (same); *Samsung Elecs. Co., Ltd. v. Mojo Mobility Inc.*, IPR2023-01094, Paper 11, 9 (Feb. 9, 2024) (same).

Of note, Petitioner filed this Petition over five months after it stipulated to, and the Court ordered, the present pretrial and trial schedule in the District Court, and close to three months after it served invalidity contentions raising all the references, theories, and combinations now raised in the Petition. On August 25, 2023, the District Court ordered the unopposed schedule to which Petitioner had stipulated. On November 7, 2023, Petitioner served invalidity contentions raising all the references raised in the Petition (Rompaey, CoWare I, and Banerjee), and alleging that Rompaey and Banerjee "render obvious, alone or in combination with each other or other references", the same claims—indistinguishable from the grounds in the Petition. *Compare* Ex.-2017 [Arm-Preliminary-Invalidity-Contentions] 84-85, *with* Pet., 9; *see* Section VI.D, *infra*. Petitioner offers no reason why, despite knowing and having stipulated to this schedule, it did not file the Petition earlier than January 26, 2024. The Board has found that petitioners'

delay in filing petitions may add to the weight of this *Fintiv* factor in favor of denial. *See, e.g., Ericsson Inc. v. Collision Comm'ns, Inc.*, IPR2022-01233, Paper 12, 15 (Jan. 19, 2023).

There has been substantial investment in the parallel District Court proceedings, and by the time of any decision on institution in this case final contentions will have been exchanged and claim construction and fact discovery will be complete or nearly complete. Factor 3 weighs against institution. *See Fintiv II* (informative), 13-15.

D. *Fintiv* Factor 4: There Is Perfect Overlap Between This IPR And The District Court Proceeding.

Factor 4 turns on "whether all or some of the claims challenged in the petition are also at issue in district court" and whether the "petition includes the same or substantially the same claims, grounds, arguments, and evidence" as the parallel District Court case. *Fintiv* (precedential), 12-13. Here, the claims challenged in the petition are the exact same claims at issue in the District Court, and the petition includes the same claims, grounds, arguments, and art as the invalidity contentions in the District Court case. Because the overlap is complete or essentially complete, Factor 4 strongly favors denial.

Every single claim challenged in the Petition is "also at issue in district court." *Id.*; *compare* Pet., 12 (challenging claims 1, 5, 6, 8, and 14) *with* Ex.-2017 [Arm-Preliminary-Invalidity-Contentions] 1-2. Needless to say, under *Fintiv* as

seen above, this overlap weighs entirely against institution, and the Board's decisions agree. *E.g., Vector Flow, Inc. v. HID Global Corp.*, IPR2023-00353, Paper 8, 22 (July 17, 2023) (institution denied); *EclinicalWorks, supra*, 12 (same).

Petitioner offers nothing but speculation to challenge that common-sense conclusion. Petitioner argues that this factor does not favor institution because "there will not be complete overlap in the issues raised in the IPR and the Related Litigation," and "it is highly unlikely the district court will address the validity of all Challenged Claims." Pet., 87. Why? Because "PO has asserted 13 claims from two patents," which, Petitioner says, "likely exceeds the number of claims allowed at trial," so "the patentee will likely be required to reduce the asserted claims." Pet., 87. No evidence supporting any of these suppositions is offered. Petitioner impliedly asks the Board to forge the following chain of inferences: because (i) Patent Owner has asserted in court several claims from a second patent in addition to the five claims challenged here, (ii) it is "likely" that the court will require Patent Owner to cease asserting some of these claims of its two patents, and then (iii) Patent Owner will drop some of the five claims challenged in the Petition from the District Court case.

Petitioner cites to no evidence or authority supporting any links in this chain. And even if Petitioner's speculations were accurate—and there is no evidence offered for this—Petitioner's theory ignores both common sense and *Fintiv* itself.

Fintiv expressly instructs that, even if "a petition involves the same prior art challenges but challenges claims in addition to those that are challenged in the district court, ***it may still be inefficient to proceed*** because the district court may resolve validity of enough overlapping claims to resolve key issues in the petition." *Fintiv* (precedential), 13. And this is just common sense. Petitioner does not present any evidence that dropping any of these five claims would substantially reduce the overlap between the validity questions at issue.

As for the unpatentability arguments and art in the two cases, the record shows that the Petition raises "the same or substantially the same claims, grounds, arguments, and evidence" as the parallel case in district court. *Fintiv* (precedential), 12-13. The Petition raises two ground sets: (1)-(2) obviousness over Rompaey alone or in combination with CoWare I; and (3) obviousness over Banerjee alone. Pet., 12. As already explained, Petitioner's latest invalidity contentions in the District Court raise and combine these same references and assert the same theories of obviousness, including the theories that Rompaey or Banerjee "alone" allegedly show obviousness. Ex.-2017 [Arm-Preliminary-Invalidity-Contentions] 84-85; Ex.-2011 [Arm-Final-Invalidity-Contentions] 84-86. There is, thus, complete or essentially complete overlap in the prior art invalidity theories presented to the Board and the District Court.

Petitioner fails to point to any substantial difference between the grounds, arguments and evidence raised in the Petition and those that it has raised in the District Court. Pet., 87. What Petitioner's argument amounts to is an offer of a limited "*Sand Revolution*" stipulation, not the more expansive "*Sotera*" stipulation. "Petitioner stipulates that it will not pursue invalidity against the asserted claims in the district court using the combination of prior art references in the grounds in this Petition for establishing obviousness." Pet., 87; *see* Ex.-2010 [Interim Discretionary Denial Memo] 3. However, this narrow stipulation does little or nothing to alleviate inefficiency concerns. With this *Sand Revolution* stipulation, as opposed to a *Sotera* stipulation, Petitioner is agreeing to far less than what even would be the natural consequences of the case proceeding to a final written decision. *See, e.g.*, *Cal. Inst. Of Tech. v. Broadcom Ltd.*, 25 F.4th 976, 991 (Fed. Cir. 2022) ("overrul[ing] *Shaw* and clarify[ing] that estoppel applies not just to claims and grounds asserted in the petition and instituted for consideration by the Board, but to all grounds not stated in the petition but which reasonably could have been asserted against the claims included in the petition."); *see id.*, 990; *3Shape Trios A/S v. Densys Ltd.*, IPR2021-00236, Paper 12, 11-12 (May 20, 2021) (denying institution, rejecting a similar stipulation because it "is too narrow to clarify that it will not pursue any ground that could have reasonably been raised in

an IPR," and noting that "[a] broader stipulation might better address concerns regarding duplicative efforts and potentially conflicting decisions").

And this stipulation is especially meaningless here. Specifically, under this stipulation, the only grounds for which Petitioner would be estopped is Rompaey or Banerjee as a single-reference obviousness ground, or Rompaey/CoWare I as an obviousness ground, but in the District Court ligation, Petitioner has also alleged that all three references may also anticipate the asserted claims and that the three references in combination with other references render those claims obvious. Ex.-2011 [Arm-Preliminary-Invalidity-Contentions] 84-85. In fact, of the proposed combinations listed in Petitioner's invalidity contentions, all of them include each of Rompaey, Banerjee, and CoWare I. *Id.*, 88-92.

In short, the Petition challenges the same claims asserted in the parallel litigation, and uses the same arts as relied upon in the District Court. This raises substantial "concerns of inefficiency and the possibility of conflicting decisions," and weighs against institution. *See Fintiv* (precedential), 12.

E. *Fintiv* Factor 5: Petitioner Is A Defendant In District Court.

Petitioner is a defendant in the parallel District Court litigation. "Because the petitioner and the defendant in the parallel proceeding are the same party, this factor weighs in favor of discretionary denial." *Fintiv II* (informative), 15.

F. Fintiv Factor 6: Other Circumstances Weigh Against Institution, Including The Weakness Of The Petition.

This factor looks at "other circumstances that impact the Board's exercise of discretion, including the merits." *Fintiv* (precedential), 6. Such other circumstances, and therefore this sixth factor, weigh against institution. Specifically, the merits of the petition are not only less than sufficiently "compelling" to compel institution, they are relatively weak. As shown above at length in Section II, the challenged claims recite outputting of (i) a hardware IC design and (ii) software programming commands to be executed on that designed hardware IC. Section II.A.1.a, *supra*. In contrast, the two single-reference obviousness theories Petitioner proposes each rely on a reference that does not disclose outputting of any executable software to be executed on an output hardware design: they only at most disclose outputting of a hardware design. Sections II.A.1.b, II.A.2.a, II.B.2.a.-.c. In both grounds, the Petition relies—to a striking degree—on multiple unexplained citations to discussions in the references of "software" seemingly without apprehending, and certainly without stating, that the cited teachings are discussions of internal analyses of functional specifications, or literally "modeling," and are not descriptions of what is actually output from the tool. Sections II.A.2.a.ii, II.B.2.b.-.c. Finally, the Petition does not seriously argue that these references, alone or in combination with others, renders it obvious to add this missing output. Sections II.A.2.b, II.B.2.d.

* * *

Each of the factors of the precedential *Fintiv* analysis, and all of the factors collectively, weigh in favor of exercising discretion to deny institution.

IV. CONCLUSION

For the foregoing reasons, the Petition should be denied.

Respectfully submitted,

____/ Kenneth J. Weatherwax /_____

Kenneth J. Weatherwax, Reg. No. 54,528
LOWENSTEIN & WEATHERWAX LLP

Date: May 9, 2024

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITS

This Patent Owner Preliminary Response (the "POPR") consists of 13,996 words, excluding table of contents, table of authorities, certificate of service, this certificate, or table of exhibits. The POPR complies with the type-volume limitation of 14,000 words as mandated in 37 C.F.R. § 42.24. In preparing this certificate, counsel has relied on the word count of the word-processing system used to prepare the paper (Microsoft Word).

Respectfully submitted,

/ Kenneth J. Weatherwax /

Date: May 9, 2024

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the following documents were served by electronic service, by consent between the parties (Pet. at 90), on the date signed below:

PATENT OWNER'S PRELIMINARY RESPONSE

EXHIBITS 2001-2019

The names and address of the parties being served are as follows:

Kevin P. Anderson	kpanderson@duanemorris.com
Patrick D. McPherson	PDMcPherson@duanemorris.com

Respectfully submitted,

/ Robert Pistone /

Date: May 9, 2024