

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

THROUGHPUTER, INC.,
Patent Owner.

IPR2022-00528
Patent 9,424,090 B2

Before MICHAEL R. ZECHER, JOHN A. HUDALLA, and
MATTHEW S. MEYERS, *Administrative Patent Judges*.

MEYERS, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

A. *Background and Summary*

Microsoft Corporation (“Petitioner”) filed a Petition requesting an *inter partes* review of claims 1–7 and 9–15 (the “challenged claims”) of U.S. Patent 9,424,090 B2 (Ex. 1001, the “’090 patent”) pursuant to 35 U.S.C. § 311 *et seq.* Paper 2 (“Petition” or “Pet.”). ThroughPuter, Inc. (“Patent Owner”) filed a Preliminary Response. Paper 7 (“Prelim. Resp.”).

We have authority to determine whether to institute an *inter partes* review. *See* 35 U.S.C. § 314; 37 C.F.R. § 42.4(a). Under 35 U.S.C. § 314(a), we may not authorize an *inter partes* review unless the information in the petition “shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” For the reasons that follow, we institute an *inter partes* review as to claims 1–7 and 9–15 of the ’090 patent on all grounds of unpatentability presented.

B. *Real Parties-in-Interest*

Petitioner identifies itself as the real party-in-interest. Pet. 74. Patent Owner identifies itself as the real party-in-interest. Paper 11, 2.

C. *Related Proceedings*

The parties identify that the ’090 patent is involved in *ThroughPuter, Inc. v. Microsoft Corporation*, No. 2:22-cv-00344 (W.D. Wash). Paper 10, 1; Paper 11, 2. Patent Owner indicates that, “[a]s of August 18, 2022, the related district court litigation has been stayed pending resolution of the present *inter partes* review proceeding.” Paper 11, 2.

We also note that Petitioner has challenged other patents owned by Patent Owner in IPR2022-00527, IPR2022-00574, IPR2022-00757, and IPR2022-00758.

D. The '090 Patent

The '090 patent is titled “Scheduling Tasks to Configurable Processing Cores Based on Task Requirements and Specification.” Ex. 1001, code (54). The '090 patent pertains to “techniques for improving information processing efficiency and performance through dynamically adapting processing resource types to match processing task types.” *Id.* at 1:34–38.

According to the '090 patent, “the range of processing jobs for a given instance of processing hardware (e.g. a server blade used for computing Infrastructure-as-a-Service), especially over the lifetime of the given hardware instance, may comprise several types of jobs and their tasks, with each type best suited for its corresponding, distinct, processor type.” Ex. 1001, 1:56–61. The '090 patent describes that an array of hardware processing units, including different processor types, will likely run several types of jobs, each best suited to different processor types. *Id.* at 1:61–67. The '090 patent notes that it is “infeasible to predict ahead of time . . . what would be the optimal type of core for any given processor instance, or the optimal breakdown of core types for a given array of processors,” and as such, conventional scheduling techniques may lead to jobs “processed often by suboptimal types of processing cores.” *Id.* at 2:1–13.

The '090 patent “provides a processing task type adaptive manycore processor system for executing time variable sets of processing software program tasks of differing types on their assigned cores of matching types.”

Id. at 3:64–4:5. The '090 patent thus provides “application processing load adaptive allocation of the cores among the software applications configured for the system, as well as for dynamically reconfiguring the core slots according to the types of the processing tasks assigned to any given core slot of the multi-core fabric.” *Id.* at 8:40–49.

To accomplish this, the '090 patent utilizes for each application “capacity demand indicators 130 to the controller 140. Each of these indicators 130, referred to herein as core-demand-figures (CDFs), express how many cores 120 their associated app-task is presently able [to] utilize for its ready to execute instances.” Ex. 1001, 8:64–9:3. The system

repeatedly selects and places the to-be-executing instances 240 of the set of locally hosted app-tasks 210 to their assigned target core slots 120, as well as determines, for the core-slot reconfiguration subsystem of the processor 100 (FIGS. 3 and 4), the demanded core types for the core slots of the array 115, along with notifications of changes in the demanded core type for each given core slot.

Id. at 10:10–17.

Figure 3, reproduced below, is a block diagram of the system of the '090 patent.

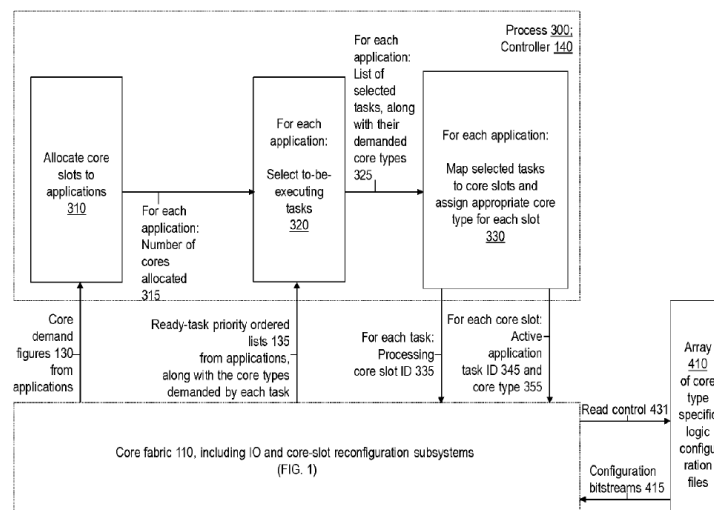


Figure 3 shows algorithm 310, which allocates core slots to applications, application selection step 320, and core mapping step 330. Ex. 1001, 14:5–67.

E. Illustrative Claim

Petitioner challenges claims 1–7 and 9–15 of the '090 patent. Pet. 2. Challenged claims 1 and 9 are independent. Claim 1 is illustrative, and is reproduced below, with Petitioner's bracketing, labels, and formatting.

[1.P] A method for assigning a set of processing tasks to an array of processor cores of configurable types, the method comprising:

[1.a] executing time variable subsets of the processing tasks of differing types on their assigned processor cores of matching types, wherein the matching type for the assigned processor core for a given processing task of the set corresponds to a type of a processor core demanded by the given processing task;

[1.b] for each of a series of core allocation periods (CAPs), selecting, from the set of processing tasks, specific tasks, referred to as selected tasks, for execution on the processor cores for a next CAP at least in part based on core capacity demand expressions associated with the processing tasks;

[1.c] assigning the selected tasks for execution on the processor cores for the next CAP in a manner to maximize, within the array, a number of processor cores whose assigned processing tasks for a present and the next CAP demands the same type of processor core; and

[1.d] configuring the array such that the type of any given processor core in the array matches the type of processing task assigned for execution on the given processor core for the next CAP.

Ex. 1001, 19:59–20:15; Pet. 31–38.

F. Prior Art and Asserted Grounds

Petitioner asserts that claims 1–7 and 9–15 of the '090 patent are unpatentable on the following grounds:

Claims Challenged	35 U.S.C. §	References
1–5, 7	103(a) ¹	Chen ² , Agrawal ³ ,
3, 4	103(a)	Chen, Agrawal, Compton ⁴
1, 2, 5–7, 9–15	103(a)	Chen, Agrawal, Brent ⁵

To support its Petition, Petitioner proffers a Declaration of Jeffrey S. Chase, Ph.D. Ex. 1003.

II. ANALYSIS

A. Legal Standards

“In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable.” *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (2012) (requiring *inter partes* review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”)). Except in limited circumstances not present here, this burden of persuasion never shifts to the

¹ The Leahy-Smith America Invents Act (“AIA”) included revisions to 35 U.S.C. §§ 102 and 103 that became effective on March 16, 2013. Because the ’090 patent claims priority as a continuation to an application filed before March 16, 2013, we apply the pre-AIA versions of the statutory bases for unpatentability. *See* Ex. 1001, code (63).

² G. Chen et al., “Configuration-Sensitive Process Scheduling for FPGA-Based Computing Platforms,” *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, Vol. 1, pp. 486–93 (2004) (Ex. 1004, “Chen”).

³ Kunal Agrawal et al., “Adaptive Scheduling with Parallelism Feedback,” *Proceedings of the 2006 ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2006 (Ex. 1005, “Agrawal”).

⁴ Katherine Compton & Scott Hauck, “Reconfigurable Computing: A Survey of Systems and Software,” *ACM Computing Surveys*, Vol. 34, No. 2, June 2002, pp. 171–210 (Ex. 1006, “Compton”).

⁵ US 2010/0131955 A1, published May 27, 2010 (Ex. 1007, “Brent”).

patent owner. *See Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (discussing the burden of proof in *inter partes* review).

A claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which that subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) objective evidence of nonobviousness when presented.⁶ *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

In analyzing the obviousness of a combination of prior art elements, it can be important to identify a reason that would have prompted one of skill in the art “to combine . . . known elements in the fashion claimed by the patent at issue.” *KSR*, 550 U.S. at 418. A precise teaching directed to the specific subject matter of a challenged claim is not necessary to establish obviousness. *Id.* Rather, “any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.” *Id.* at 420. Accordingly, a party who petitions the Board for a determination of unpatentability based on obviousness must show that “a skilled artisan

⁶ The present record does not include any evidence of secondary considerations of nonobviousness.

would have been motivated to combine the teachings of the prior art references to achieve the claimed invention, and that the skilled artisan would have had a reasonable expectation of success in doing so.” *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1381 (Fed. Cir. 2016) (quotations and citations omitted).

We analyze the challenges presented in the Petition in accordance with the above-stated principles.

B. Level of Ordinary Skill in the Art

We review the grounds of unpatentability presented in view what a person of ordinary skill in the art would have understood at the time of invention. *Graham*, 383 U.S. at 17. Petitioner asserts a person of ordinary skill in the art at the time of the invention “would have had a Master’s degree in computer science, computer engineering, or a related field, and 2–3 years of practical computer programming or engineering experience, including experience designing or researching parallel processing systems.” Pet. 19 (citing Ex. 1003 ¶ 18). Petitioner adds that “[a]dditional graduate education could substitute for professional experience, or significant experience in the field could substitute for formal education.” *Id.* at 20. Patent Owner does not address the level of ordinary skill in the art. *See, generally* Prelim. Resp.

Accordingly, for purposes of this Decision, we adopt Petitioner’s description of the level of skill in the art. On the present record, we are satisfied that this definition comports with the level of skill necessary to understand and implement the teachings of the ’090 patent and the asserted prior art.

C. Claim Construction

In an *inter partes* review, we construe patent claims using the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. § 282(b). 37 C.F.R. § 42.100(b). This rule adopts the same claim construction standard used by Article III federal courts, which follow *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*), and its progeny. Under that standard, the words of a claim are generally given their “ordinary and customary meaning,” which is the meaning the term would have to a person of ordinary skill at the time of the invention, in the context of the entire patent including the specification. *See Phillips*, 415 F.3d at 1312–13. “[W]here a party believes that a specific term has meaning other than its plain meaning, the party should provide a statement identifying a proposed construction of the particular term and where the disclosure supports that meaning.” Consolidated Trial Practice Guide,⁷ 44.

Petitioner submits that the term “an array of processor cores of configurable types” should be construed to mean “an indexed set of discrete, reconfigurable hardware elements for processing.” Pet. 20 (citing Ex. 1003 ¶¶ 62–66). Patent Owner does not comment on Petitioner’s proposed construction, or otherwise address claim construction. *See generally* Prelim. Resp.

We determine, at this stage, that no explicit construction of any terms is needed to resolve the issues presented by the arguments and evidence of record to reach a decision on institution. *See Realtime Data, LLC v. Iancu*,

⁷ Available at <https://www.uspto.gov/TrialPracticeGuideConsolidated>.

912 F.3d 1368, 1375 (Fed. Cir. 2019) (“The Board is required to construe ‘only those terms . . . that are in controversy, and only to the extent necessary to resolve the controversy.’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

D. Alleged Obviousness over Chen and Agrawal (Ground 1)

Petitioner asserts that claims 1–5 and 7 would have been obvious over Chen and Agrawal. Pet. 22–44. Petitioner also relies on the testimony of Dr. Chase to support its arguments. *Id.* (citing Ex. 1003). Patent Owner responds to Petitioner’s assertions. Prelim. Resp. 1–6.

1. Overview of Chen (Ex. 1004)

Chen is titled “Configuration-Sensitive Process Scheduling for [Field-Programmable Gate Array (“FPGA”)] FPGA-Based Computing Platforms.” Ex. 1004, 1⁸. Chen introduces an operating system “process scheduler, and demonstrate[s] how it can be customized considering the needs of reconfigurable hardware,” with a characteristic that the “process scheduler is configuration sensitive, that is, it reuses the current FPGA configuration as much as possible.” *Id.*

Chen models its “reconfigurable system as a rectangular array of configurable logic blocks (say NxM) – called CLBs.” Ex. 1004, 2. Chen describes that “[e]ach process to be scheduled in this architecture uses a ‘rectangular portion’ of this array.” Figure 1, reproduced below, “illustrates a 6x8 array and three processes scheduled on it (they occupy spaces of 2x3, 2x4, and 6x2 CLBs).” *Id.*

⁸ Exhibit 1004 has no original page numbers in the filed version. We rely on the page numbers inserted by Petitioner in the bottom, middle portion of each page in Exhibit 1004.

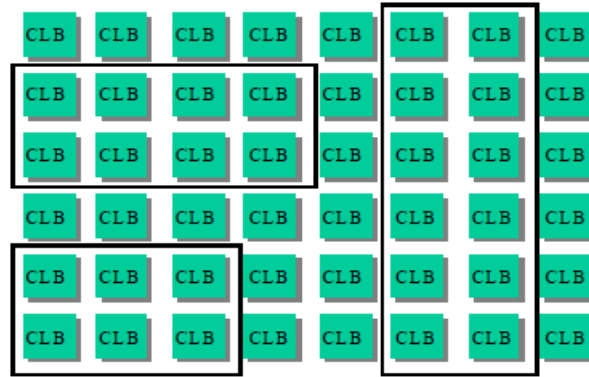


Figure 1 of Chen, illustrates a 6x8 array of configurable logic blocks (such as FPGAs), with rectangles around three running processes occupying different rectangular segments of the array.

Chen discloses that each process to be scheduled is represented by a subtask graph (“STG”) and “[e]ach node of this graph represents a process code portion (subtask) that will be executed in a single quantum of time once it gets scheduled.” Ex. 1004, 2. Chen explains that, “[s]ince one of the objectives of any FPGA-based system is to maximize FPGA utilization, the OS scheduler should be able to schedule nodes from the STGs of different applications.” *Id.* Chen’s “approach is oriented towards maximizing FPGA space utilization by parallel execution of multiple processes.” *Id.*

In Chen, “[t]he crucial step in [its] configuration-aware scheduling strategy is assigning an FPGA space for a process that is about to start executing.” Ex. 1004, 3. To that end, Chen describes that its “OS scheduler has two important tasks.” *Id.* at 3. The first is “[d]etermining a suitable schedule for the processes to be scheduled,” and the second, is “[w]hen a process need[s] to start executing, determining where (to which FPGA region(s)) its subtasks should be assigned.” *Id.* Chen’s “scheduling

algorithm is a variant of list scheduling, an algorithm frequently employed by optimizing compilers.” *Id.* at 3. More particularly, Chen discloses

[i]n this scheduling, the next STG node to be scheduled (and to be assigned an FPGA space) is the one that leads to minimum cost (considering the four cases above) among all schedulable nodes. In this way, at each step, this greedy heuristic selects the next STG node to be scheduled.

Id.

2. Overview of Agrawal (Ex. 1005)

Agrawal is titled “Adaptive Scheduling with Parallelism Feedback.” Ex. 1005, 1.⁹ Agrawal relates to multiprocessor computer system scheduling in a shared multiprogramming environment. *Id.* Agrawal presents “an adaptive task scheduler for multitasked jobs with dependencies that provides continual parallelism feedback to the job scheduler in the form of requests for processors.” *Id.* Agrawal focuses on “space-sharing” for parallel jobs, using a two-level strategy: “a kernel-level job scheduler which allots processors to jobs, and a user-level task scheduler which schedules the tasks belonging to a given job onto the allotted processors.” *Id.*

Using “parallelism feedback,” Agrawal’s scheduler “adaptively changes the allotment of processors according to the availability of processors in the current system environment and the job scheduler’s administrative policy.” Ex. 1004, 2. Agrawal’s adaptive task scheduler does not focus on job scheduling, but rather on task scheduling, and, “[i]nstead of using instantaneous parallelism,” Agrawal’s A-Greedy algorithm “provides

⁹ Exhibit 1005 has no original page numbers in the filed version. We rely on the page numbers inserted by Petitioner in the bottom, middle portion of each page in Exhibit 1004.

parallelism feedback to the job scheduler based on a single summary statistic and the job's behavior on the previous quantum." *Id.* Agrawal describes that its A-Greedy algorithm "provides parallelism feedback using the past behavior of the job" and does "not assume that the job's future parallelism is correlated with its history of parallelism." *Id.* According to Agrawal:

Between quanta $q - 1$ and q , the task scheduler determines its job's **desire** d_q , which is the number of processors the job wants for quantum q . The task scheduler provides the desire d_q to the job scheduler as its parallelism feedback. The job scheduler follows some processor allocation policy to determine the **processor availability** p_q , which is the number of processors the job is entitled to get for the quantum q .

Id. Agrawal discloses that "[t]he number of processors the job receives for quantum q is the job's **allotment** $a_q = \min \{d_q, p_q\}$, which is the smaller of its desire and the processor availability." *Id.*

Agrawal also describes "instantaneous parallelism," which relies on "the number of processors the job can effectively use at the current moment, as the parallelism feedback to the job scheduler." Ex. 1004, 2. Agrawal discloses that, "[a]lthough using instantaneous parallelism for parallelism feedback is simple, it can cause gross misallocation of processor resources."

Id. According to Agrawal,

the parallelism of a job may change substantially during a scheduling quantum, alternating between parallel and serial phases. The sampling of instantaneous parallelism at a scheduling event between quanta may lead the task scheduler to request either too many or too few processors depending on which phase is currently active, whereas the desirable request might be something in between. Consequently, the job may systematically waste processor cycles on the one hand or take too long to complete on the other.

Id. Agrawal acknowledges that A-Greedy “operates nearly as efficiently” as a “task scheduler that uses instantaneous parallelism as feedback,” and that a scheduler using instantaneous parallelism wastes no processor cycles. *Id.*

3. *Analysis of Independent Claim 1*

Petitioner asserts claim 1 would have been obvious over Chen and Agrawal. Pet. 26–39. We use Petitioner’s notations to identify the claim elements.

- a) *[1.P] A method for assigning a set of processing tasks to an array of processor cores of configurable types, the method comprising:*

Petitioner asserts that, if the preamble is treated as limiting, Chen discloses its contents. Pet. 31–33 (citing Ex. 1003 ¶¶ 71–74). More particularly, Petitioner asserts that “Chen teaches a method for assigning processes and their tasks (‘subtasks’ or ‘STG nodes’) to spaces/portions/regions (hereinafter, ‘regions’) of an FPGA, where the regions of the FPGA are configurable for the different processes.” *Id.* at 31–32 (citing Ex. 1004, 1–3, Fig. 1). According to Petitioner, “Chen proposes ‘a configuration-sensitive execution strategy; i.e., at each scheduling step, we select (among the executable process) the one that fits well in one or more regions, without needing reconfiguration (which is typically a costly operation).” *Id.* at 32 (citing Ex. 1004, 3).

- b) *Element [1.a] executing time variable subsets of the processing tasks of differing types on their assigned processor cores of matching types, wherein the matching type for the assigned processor core for a given processing task of the set corresponds to a type of a processor core demanded by the given processing task;*

Petitioner asserts that Chen discloses “a ‘configuration-aware scheduling strategy’ that includes ‘assigning an FPGA space for a process that is ***about to start executing***.” Pet. 33 (citing Ex. 1003, 3). According to

Petitioner, “Chen teaches ‘at each scheduling step,’ ‘select[ing] (among the executable process) ***the one that fits well in one or more regions***, without needing reconfiguration (which is typically a costly operation).” *Id.* at 34 (citing Ex. 1004, 3).

- c) *Element [1.b] for each of a series of core allocation periods (CAPs), selecting, from the set of processing tasks, specific tasks, referred to as selected tasks, for execution on the processor cores for a next CAP at least in part based on core capacity demand expressions associated with the processing tasks*

Petitioner asserts

[t]he combination of Chen and Agrawal discloses for each of a series of core allocation periods (CAPs), selecting, from the set of processing tasks, specific tasks, referred to as selected tasks, for execution on the processor cores for a next CAP at least in part based on core capacity demand expressions associated with the processing tasks.

Pet. 34 (citing Ex. 1003 ¶¶ 94–105; Ex. 1004, 2–3, Fig. 7; Ex. 1005, 2–3).

According to Petitioner, one of ordinary skill in the art “would have understood that a CAP is an interval or period of time for which core allocation decisions are made” (*id.* at 35 (citing Ex. 1003 ¶ 94)), and “Chen, explains that ‘[e]ach node of [a subtask] graph represents a process code portion (subtask) that will be executed ***in a single quantum of time*** once it gets scheduled.’” *Id.* (citing Ex. 1004, 2). Petitioner asserts that, in Chen,

when a new incoming process needs to be scheduled, the OS needs to find a (rectangular) space for it in the reconfigurable device (this is called space allocation). This space can be one of the spaces that have already been allocated to some other process that could not run concurrently with this incoming process (this is called configuration reuse in this paper).

Id. (quoting Ex. 1004, 2). Petitioner further asserts that Chen discloses that “each process to be scheduled [is represented] by a subtask graph

(henceforth referred to as STG),’ and ‘[e]ach node of this graph represents a process code portion (subtask) that will be executed in a single quantum of time once it gets scheduled.’” *Id.* (quoting Ex. 1004, 2). Petitioner argues that, in Chen’s configuration-sensitive execution strategy, Chen describes “select[ing] (among the executable process) the one that fits well in one or more regions, without needing reconfiguration (which is typically a costly operation).” *Id.* (quoting Ex. 1004, 3). When selecting tasks for scheduling, Chen’s scheduler “tak[es] into account, for a given ‘STG_{ij} for potential execution on the FPGAconfig-desc_{ij},’ the ‘configuration required by STG_{ij} and {(s₁,s₂),(e₁,e₂)}_{ij},’ which defines the region requested for the STG.” *Id.* at 35–36 (citing Ex. 1004, 3).

Petitioner also contends that, to the extent “Chen does not teach the use of core allocation periods,” Agrawal discloses “scheduling tasks to processors where time is broken into a sequence of equal-size scheduling quanta of length L, which is a CAP.” Pet. 36 (citing Ex. 1003 ¶ 96; Ex. 1005, 2). Petitioner asserts that, in Agrawal’s scheduling system, “time is broken into a sequence of equal-size scheduling quanta 1, 2, . . . of length L, and the job scheduler is free to reallocate processors between quanta.” *Id.* (Ex. 1005, 2). According to Petitioner, one of ordinary skill in the art “would have been motivated to combine the mapping of tasks to regions (cores) of Chen with the scheduling quanta of Agrawal to realize the benefits of selecting and using an appropriate scheduling quantum length.” *Id.* at 36–37 (citing Ex. 1003 ¶ 102; Ex. 1005, 2). Petitioner explains that one of ordinary skill in the art “would have recognized that, like Chen, Agrawal is directed to a space-sharing system for scheduling tasks to hardware by means of Greedy heuristics,” and as such “would have been motivated to

modify Chen's method with Agrawal's use of scheduling quanta for making favorable choices for space allocation across the array." *Id.* at 37 (citing Ex. 1003 ¶ 103; Ex. 1005, 3).

d) *Element [1.c] assigning the selected tasks for execution on the processor cores for the next CAP in a manner to maximize, within the array, a number of processor cores whose assigned processing tasks for a present and the next CAP demands the same type of processor core; and*

Petitioner asserts that Chen discloses "a configuration-sensitive execution strategy" that, at each scheduling step, selects the executable process "*that fits well in one or more regions, without needing reconfiguration* (which is typically a costly operation)." Pet. 38 (citing Ex. 1004, 3). Petitioner further asserts that "Chen indicates that its scheduling algorithm '*tries to reuse the current configuration as much as possible*' (*id.* (citing Ex. 1005, 1), and notes that Chen's "*main objective is to maximize configuration reuse, that is, . . . to minimize the number of reconfigurations.*" *Id.* (quoting Ex. 1005, 2). According to Petitioner, one of ordinary skill in the art

would have understood that by assigning tasks to cores in a manner that maximizes configuration reuse, or equivalently that minimizes the number of reconfigurations, Chen teaches assigning the selected tasks for execution on the processor cores for the next CAP in a manner to maximize, within the array, a number of processor cores whose assigned processing tasks for a present and the next CAP demands the same type of processor core.

Id. (citing Ex. 1003 ¶ 108).

e) Element [1.d] configuring the array such that the type of any given processor core in the array matches the type of processing task assigned for execution on the given processor core for the next CAP.

Petitioner asserts that Chen’s configuration-sensitive execution strategy “configure[s] regions (cores) of the array in advance . . . based on the scheduling information it has.” Pet. 38–39 (citing Ex. 1004, 1, 4). According to Petitioner, Chen “explains that the tasks of a process are assigned to regions (cores) and that it selects and executes a process that ‘fits well in the one or more regions’” (*id.* at 39 (citing Ex. 1004, 3)), and “Chen indicates that [‘]preconfiguration for the STG node scheduled for time *t* starts at time *t*-1.’” *Id.* (citing Ex. 1004, 4).

4. Patent Owner’s Contentions

Patent Owner argues that Petitioner’s ground of unpatentability based on Chen and Agarwal is deficient. *See* Prelim. Resp. 1–6. More particularly, Patent Owner asserts that Petitioner relies on “Chen alone” to address “the claimed steps of ‘assigning’ and ‘selecting’” because “Chen discloses selecting only a **single** subtask for execution during an upcoming epoch.” *Id.* at 1, 4.

5. Discussion

Having considered the positions of the parties, we conclude that, at this stage of the proceeding, Petitioner has shown a reasonable likelihood that it would prevail in challenging the patentability of independent claim 1. In our view, Petitioner’s obviousness approach, on this record, adequately identifies where all the elements of claim 1 are found in the prior art, and, in considering claim 1 as a whole, Petitioner demonstrates adequate reasoning to combine the teachings of Chen and Agarwal.

At this stage of the proceeding, we disagree with Patent Owner that Petitioner relies on “Chen alone” when addressing the step of “selecting,” as called for in limitation [1.b]. *See* Pet. 34–37. Instead, Petitioner relies on “[t]he combination of Chen and Agrawal” to address limitation [1.b]. *Id.* at 34. Specifically, Petitioner asserts that,

[t]o the extent that Patent Owner argues that Chen does not teach the use of core allocation periods, Agrawal discloses, for each of a series of core allocation periods (CAPs), selecting, from the set of processing tasks, specific tasks, referred to as selected tasks, for execution on the processor cores for a next CAP at least in part based on core capacity demand expressions associated with the processing tasks.

Id. at 36 (citing Ex. 1003 ¶ 96). Petitioner further asserts that one of ordinary skill in the art “would have been motivated to combine the mapping of tasks to regions (cores) of Chen with the scheduling quanta of *Agrawal* to realize the benefits of selecting and using an appropriate scheduling quantum length.” *Id.* at 36–37 (emphasis added) (citing Ex. 1003 ¶ 102; Ex. 1005, 2).

Based on the current record, we also disagree with Patent Owner that one of ordinary skill in the art would have understood Chen’s configuration-sensitive execution strategy to be limited to “selecting only a *single* subtask for execution during an upcoming epoch.” Prelim. Resp. 4. Patent Owner argues that the “Petition never explains how the selection and assignment of a *single* subtask in Chen meets the claimed recitation of selecting and assigning at least two tasks for an array for a given core allocation period,” because of the “the clear disclosure in Chen that Chen selects only a single task (or subtask) for assignment at a time.” *Id.* at 5.

However, as Petitioner points out, Chen discloses that “each ‘process to be scheduled [is represented] by a subtask graph (henceforth referred to as

STG),’ and ‘[e]ach node of this graph represents a process code portion (subtask) that will be executed in a single quantum of time once it gets scheduled.’” Pet. 35 (quoting Ex. 1004, 2). Chen describes that, because “one of the objectives of any FPGA-based system is to maximize FPGA utilization, the OS scheduler should be able to schedule nodes from the STGs of different applications.” Ex. 1004, 2. Chen also makes clear that its approach “is oriented towards maximizing FPGA space utilization by parallel execution of multiple processes.” *Id.*

In our view, the preliminary record indicates that one of ordinary skill in the art would not have understood Chen’s disclosure as limiting its scheduler to scheduling a single task for an upcoming period, but instead, to more broadly teach, scheduling more than one process if more than one process can execute in the next period.¹⁰ *See also* Ex. 1004, 3 (“The OS scheduler has two important tasks: a. Determining a suitable schedule for the processes to be scheduled. b. When a process need to start executing, determining where (to which FPGA region(s)) its subtasks should be assigned.”). This view is further supported by Chen’s disclosure that,

[i]n this scheduling, the next STG node to be scheduled (and to be assigned an FPGA space) is the one that leads to minimum cost (considering the four cases above) among all schedulable nodes. In this way, at each step, this greedy heuristic selects the next STG node to be scheduled.

Ex. 1004, 3. Here, we see no reason as to why Chen’s scheduler would stop scheduling after it schedules a single task for an upcoming period. Thus, we

¹⁰ Patent Owner “does not dispute that Chen discloses core allocation periods and instead agrees that Chen’s ‘epochs’ qualify as ‘core allocation periods.’” Prelim. Resp. 5.

are persuaded that Petitioner has made a sufficient showing, at this time, that one of ordinary skill in art would have understood the combination of Chen and Agrawal to disclose the claimed steps of “selecting” and “assigning,” as required by independent claim 1.

a) Conclusion for Claim 1

Accordingly, at this stage of the proceeding, we determine that Petitioner has established a reasonable likelihood of prevailing on its assertion that claim 1 would have been obvious over the combination of Chen and Agarwal.

6. Dependent Claims 2–5 and 7

Claims 2–5 and 7 depend directly or indirectly from claim 1. Petitioner provides a clause-by-clause analysis of where each element in dependent claims 2–5 and 7 is disclosed in, or would have been obvious in view of, Chen and Agrawal. Pet. 39–44. Throughout Petitioner’s analysis, Petitioner cites and relies on the testimony of Dr. Chase for evidentiary support. *Id.* (citing Ex. 1003 ¶¶ 114–116, 118–120, 122–124, 126–128, 130–133).

Patent Owner does not specifically address any of the dependent claims. *See* Prelim Resp. 1–6.

We find Petitioner’s arguments and evidence in support thereof sufficient at this stage of the proceeding for the reasons explained by Petitioner. *See* Pet. 39–44. Accordingly, based on the arguments and evidence on the record before us, we determine, for purposes of this Decision, Petitioner has established a reasonable likelihood of prevailing on dependent claims 2–5 and 7.

E. Alleged Obviousness over Chen, Agrawal, and Compton (Ground 2)

Petitioner asserts that claims 3 and 4 would have been unpatentable as obvious over Chen, Agrawal, and Compton. Pet. 44–50. Petitioner also relies on the testimony of Dr. Chase to support its arguments. *Id.* (citing Ex. 1003).

1. Overview of Compton (Ex. 1006)

Compton is titled “Reconfigurable Computing: A Survey of Systems and Software,” and “presents a survey of current research in hardware and software systems for reconfigurable computing, as well as techniques that specifically target run-time reconfigurability.” Ex. 1006, 171, 173.

Compton provides “an introduction to this rapidly evolving field, bringing interested readers quickly up to speed on developments from the last half-decade.” *Id.* at 173. Compton discusses systems that incorporate “some form of hardware programmability—customizing how the hardware is used using a number of physical control points. These control points can then be changed periodically in order to execute different applications using the same hardware.” *Id.* at 173–74. Compton explains that “[r]econfigurable computing systems use FPGAs or other programmable hardware to accelerate algorithm execution by mapping compute-intensive calculations to the reconfigurable substrate.” *Id.* at 175.

2. Claims 3 and 4

Claims 3 and 4 depend directly or indirectly from claim 1. Petitioner provides a clause-by-clause analysis of where each element in dependent claims 3 and 4 is disclosed in, or would have been obvious in view of, Chen, Agrawal, and Compton. Pet. 45–50. Throughout Petitioner’s analysis,

Petitioner cites and relies on the testimony of Dr. Chase for evidentiary support. *Id.* (citing Ex. 1003 ¶¶ 134–145).

Patent Owner does not specifically address any of the dependent claims challenged based on the combination of Chen, Agrawal, and Compton. *See* Prelim. Resp. 1–6.

We find Petitioner’s arguments and evidence in support thereof sufficient at this stage of the proceeding for the reasons explained by Petitioner. *See* Pet. 44–50. Accordingly, based on the arguments and evidence on the record before us, we determine, for purposes of this Decision, Petitioner has established a reasonable likelihood of prevailing on dependent claims 3 and 4.

F. Alleged Obviousness over Chen, Agrawal, and Brent (Ground 3)

Petitioner asserts that claims 1, 2, 5–7, and 9–15 would have been obvious over Chen, Agrawal, and Brent. Pet. 50–74. Petitioner also relies on the testimony of Dr. Chase to support its arguments. *Id.* (citing Ex. 1003).

1. Overview of Brent (Ex. 1007)

Brent is titled “Highly Distributed Parallel Processing on Multi-Core Device.” Ex. 1007, code (54). Brent relates to multi-core processing systems, and discloses a highly distributed multi-core system with an adaptive scheduler. *Id.* ¶¶ 3, 8.

Figure 1 of Brent, reproduced below, shows an exemplary multi-core system with an adaptive scheduler.

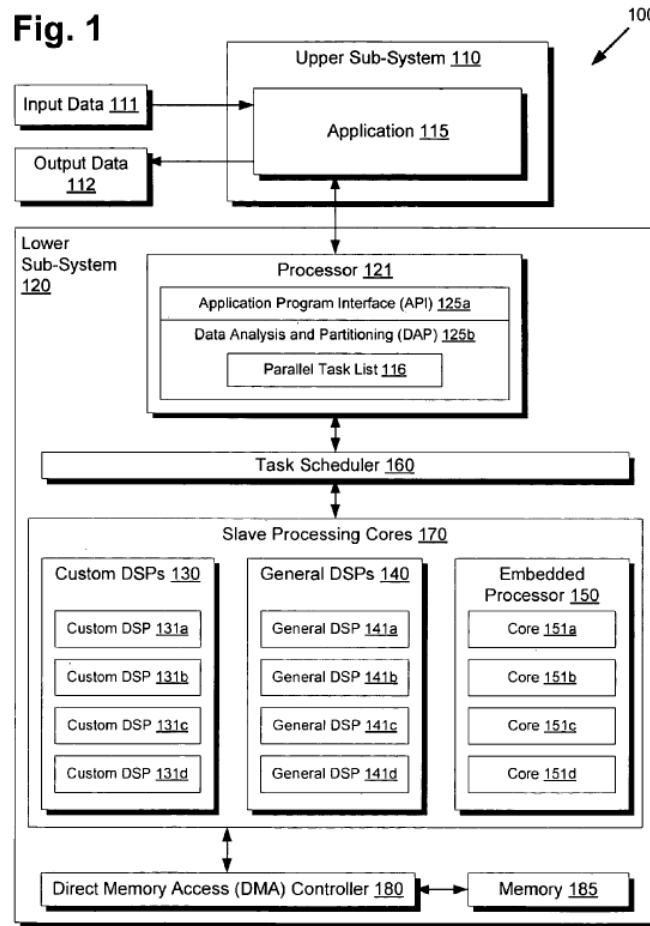


Figure 1 shows, among other things, input data 111, application 115, and processor 121 which creates task list 116, which is passed to task scheduler 160 for execution on slave processing cores 170. Brent discloses that “slave processing cores 170 may include several different types of processing cores. Custom digital signal processors (DSPs) 130 include custom DSPs 131a–131d, which may have limited instruction sets optimized for specific processing patterns.” Ex. 1007 ¶ 16. Brent describes that “each task may be assigned a preferred core type for optimal execution.” *Id.* ¶ 18.

2. Claims 1, 2, 5–7, and 9–15

Petitioner asserts claims 1, 2, 5–7, and 9–15 would have been obvious over Chen, Agrawal, and Brent. Pet. 50–74.

Patent Owner does not contest Petitioner’s challenge based on the combination of Chen, Agrawal, and Brent other than by relying on its arguments presented the asserted ground based on the combination of Chen and Agrawal, as discussed above. *See* Prelim. Resp. 3 n.1 (“The Petition includes a ground combining Brent with Chen and Agrawal. However, as explained herein, Petitioner’s ground is really a single reference obviousness ground in view of Chen and Chen alone.”).

Petitioner provides a clause-by-clause analysis of where each element in claims 1, 2, 5–7, and 9–15 is disclosed in, or would have been obvious in view of, Chen, Agrawal, and Brent. *Id.* at 50–74. Throughout Petitioner’s analysis, Petitioner cites and relies on the testimony of Dr. Chase for evidentiary support. *Id.* (citing Ex. 1003 ¶¶ 146–224). Accordingly, for the reasons set forth above, and on this preliminary record, we determine that Petitioner has established a reasonable likelihood of prevailing on its assertion that claims 1, 2, 5–7, and 9–15 would have been obvious over the combination of Chen, Agarwal, and Brent.

III. DISCRETIONARY DENIAL

Institution of *inter partes* review is discretionary. *See*, 815 F. 3d at 1367 (“[T]he PTO is permitted but never compelled to institute an [*inter partes* review (IPR)] proceeding.”); 35 U.S.C. § 314(a).

Patent Owner does not analyze each of the factors set forth in *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11, 5–6 (PTAB Mar. 20, 2020) (precedential) (“*Fintiv*”). Prelim. Resp. 6. Instead, with respect to respect to *Fintiv* Factor 1, Patent Owner simply informs us that Petitioner’s motion to stay the parallel litigation in the United States District Court for the Western District of Washington” was denied. *Id.* (citing Ex. 2002). After the

Preliminary Response was filed, however, Patent Owner informs us “that the related matter in the Western District of Washington is now stayed pending resolution of the present *inter partes* review proceeding.” Paper 11, 2.

We have not been informed by the parties that a trial date has been scheduled in the parallel litigation. In light of this development, it is not necessary to analyze the *Fintiv* factors in detail because the parallel litigation has been stayed. *See Fintiv*, Paper 11 at 6 (“A district court stay of the litigation pending resolution of the PTAB trial allays concerns about inefficiency and duplication of efforts. This fact has strongly weighed against exercising the authority to deny institution under [*NHK Spring Co. v. Intrix-Plex Techs., Inc.*, IPR2018-00752, Paper 8 (PTAB Sept. 12, 2018) (precedential, designated May 7, 2019)].”); *Precision Planting, LLC v. Deere & Co.*, IPR2019-01052, Paper 19 at 10 (PTAB Jan. 7, 2020) (finding that the district court stay of the parallel district court case rendered moot the patent owner’s argument for discretionary denial of the petition).

Accordingly, we decline to exercise our discretion to deny institution.

IV. CONCLUSION

For the foregoing reasons, we conclude that the information presented in the Petition establishes a reasonable likelihood that Petitioner would prevail in showing that at least one of claims 1–7 and 9–15 of the ’090 patent is unpatentable. Accordingly, we institute *inter partes* review of all claims on all grounds raised in the Petition.

Our determination in this Decision is not a final determination on the patentability of any challenged claims and, thus, leaves undecided any factual issues necessary to determine whether sufficient evidence supports Petitioner’s contentions by a preponderance of the evidence in the final

written decision. *See TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1068 (Fed. Cir. 2016) (noting that “there is a significant difference between a petitioner’s burden to establish a ‘reasonable likelihood of success’ at institution, and actually proving invalidity by a preponderance of the evidence at trial”) (quoting 35 U.S.C. § 314(a) and comparing § 316(e)).

V. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that, pursuant to 35 U.S.C. § 314(a), an *inter partes* review of claims 1–7 and 9–15 of U.S. Patent No. 9,424,090 B2 is instituted with respect to all grounds set forth in the Petition; and

FURTHER ORDERED that, pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial, which will commence on the entry date of this decision.

IPR2022-00528
Patent 9,424,090 B2

FOR PETITIONER:

Walter Renner
David Holt
Nicholas Stephens
Jennifer Huang
FISH & RICHARDSON P.C.
Axf-ptab@fr.com
Holt2@fr.com
nstephens@fr.com
jjh@fr.com

FOR PATENT OWNER:

Scott McKeown
ROPES & GRAY LLP
Scott.mckeown@ropesgray.com

W. Cook Alciati
Michael Dorfman
GARDELLA GRACE P.A.
calciati@gardellagrace.com
mdorfman@gardellagrace.com