

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE, INC.,
Petitioner,

v.

WISCONSIN ALUMNI RESEARCH FOUNDATION,
Patent Owner.

Case IPR2014-01567
Patent 5,781,752

Before JUSTIN T. ARBES, BRIAN J. McNAMARA, and
J. JOHN LEE, *Administrative Patent Judges*.

McNAMARA, *Administrative Patent Judge*.

DECISION
Denial of Institution of *Inter Partes* Review
37 C.F.R. § 42.108

BACKGROUND

Apple, Inc. (“Petitioner”) filed a Petition, Paper 2 (“Pet.”), to institute an *inter partes* review of claims 1–9 (the “challenged claims”) of U.S. Patent No. 5,781,752 (“the ’752 Patent”). 35 U.S.C. § 311. Wisconsin Alumni Research Foundation (“WARF” or “Patent Owner”) timely filed a Preliminary Response, Paper 11 (“Prelim. Resp.”),¹ contending that the Petition should be denied as to all challenged claims. We conclude that Petitioner has not shown, under 35 U.S.C. § 314(a), that there is a reasonable likelihood that it will prevail with respect to at least one of the challenged claims.

For the reasons discussed below, we decline to institute an *inter partes* review in this proceeding.

PENDING LITIGATION

The ’752 Patent is the subject of a civil case styled *WARF v. Apple Inc.*, Civil Action No. 3:14-cv-00062-bbc, pending in U.S. District Court for Western District of Wisconsin. The ’752 Patent previously was asserted in *WARF v. Intel Corp.*, Civil Action No. 3:08-cv-00078-bbc,”) in the U.S. District Court for the Western District of Wisconsin (the “Intel case”), which was dismissed on October 20, 2009.

¹ Patent Owner filed Redacted Preliminary Response (Paper 11) and Unredacted Preliminary Response (Paper 10). Because we need not refer to the Unredacted Patent Owner Preliminary Response for purposes of this Decision, all references to “Prelim Resp.” herein are to Redacted Preliminary Response (Paper 11).

THE '752 PATENT (EXHIBIT 1001)

The '752 Patent recognizes that in processing systems, instructions execute correctly only if earlier instructions using the same data (i) do not change data common to the instructions or (ii) have completed changing common data. Ex, 1001, col. 2, ll. 2–5. The '752 Patent concerns those systems, such as an instruction level parallel (ILP) processing unit, in which it is possible to execute an instruction that loads data into a processing unit before the appropriate data for that instruction has been stored or updated, resulting in an error that cannot be ignored. *Id.* at col. 1, l. 50–col. 2, l. 12.

Noting the difficulty in predicting and tracking data dependencies in such systems, *id.* at col. 3, ll. 42–44, the '752 Patent distinguishes between unambiguous and ambiguous dependencies, *id.* at col. 2, ll. 5–12. An instruction has an “unambiguous” data dependency when the dependent instruction necessarily produces an error if the instruction executes before the instruction from which it depends. *Id.* at col. 2, ll. 5–7. Recognizing unambiguous dependencies is useful because the time consumed in squashing (i.e., discarding) results of premature execution of instructions reduces the benefits achieved by parallel processing. *Id.* at col. 2, ll. 46–57.

The '752 Patent discloses that dependencies are often ambiguous, i.e., it cannot be determined whether a particular instruction will, in fact, be dependent upon earlier instructions that have not completed their execution, without actually executing that particular instruction. *Id.* at col. 2, ll. 9–22. The '752 Patent discloses that, because many ambiguous dependencies resolve to no dependency, it was known for some ILP processors to employ “speculation” in which an instruction with an ambiguous dependency executes as if no dependency exists. *Id.* at col. 2, ll. 26–30. “Data

speculation” involves reading from memory to obtain data for a later instruction, even though earlier stores to the relevant memory location have not been completed and the data may change. *Id.* at col. 2, ll. 36–40.

The ’752 Patent notes that in an ILP processor, data speculation circuits detect data dependencies and report to a retirement circuit that writes final results to memory or squashes the result of any mis-speculation, i.e., an instruction that executed out of program order and produced an erroneous result because the instruction was in fact dependent on an earlier instruction in the execution order. *Id.* at col. 2, l. 60–col. 3, l. 8. According to the ’752 Patent, data speculation circuitry known at the time of the invention speculated on all memory accesses or none at all. *Id.* at col. 3, ll. 8–11.²

The ’752 Patent states that the invention is based on the recognition that most data dependence mis-speculations can be attributed to a few static STORE/LOAD instruction pairs that exhibit “temporal locality” and that if one LOAD/STORE pair of instructions causes a data mis-speculation at a given point in time, it is highly likely the same pair will soon cause another mis-speculation. *Id.* at col. 3, ll. 54–57. Based on previous mis-speculations, data dependent instructions likely to be a source of mis-speculation are stored in a memory, using a table based approach. *Id.* at col. 3, ll. 57–62. Thus, if an instruction has no history of mis-speculation, it is executed without further inquiry; but if there has been a mis-speculation for a given load instruction, a predictor based on the past history of mis-speculation determines whether the instruction should be executed or delayed. *Id.* at col. 3, l. 64–col. 4, l. 4, col. 11, ll. 19–24.

² Petitioner disputes this characterization, citing Hesson (Ex. 1003). Pet. 7.

The '752 Patent describes modifications to the operation of a conventional data speculation circuit to accommodate its prediction and synchronization circuitry. *Id.* at col. 9, ll. 40–41. A retirement circuit provides to the data speculation circuit an instruction and an indication of whether that instruction should be squashed or is about to execute. *Id.* at col. 9, ll. 46–49. The data speculation circuit communicates with a predictor circuit. *Id.* at col. 7, ll. 66–67. The predictor circuit, which is updated based on historical mis-speculations detected by the data speculation circuit, provides the data speculation circuit a dynamic indication about whether the data speculation should be performed. *Id.* at col. 8, ll. 1–12. The prediction circuit includes a prediction table in which each row identifies a physical address of the instruction that is ready for its operation to be performed, the instruction on which the first instruction may be dependent, and a prediction. *Id.* at col. 11, ll. 8–13. If there is no entry in the prediction table, it is assumed the instruction can proceed. *Id.* at col. 11, ll. 19–23. Entries are made and incremented or decremented based on the history of mis-speculation. If the prediction table indicates there is a likelihood of mis-speculation, a synchronization table, which indicates whether there is a pending LOAD instruction awaiting its dependent STORE instruction, is examined. *Id.* at col. 11, ll. 38–47. A flag in the synchronization table indicates whether a STORE associated with a particular LOAD has occurred or is required, thereby indicating whether the LOAD must wait or can be allowed to proceed. *Id.*

If a mis-speculation occurs and there is already an entry in the prediction table, the prediction is updated and incremented toward synchronization so that this mis-speculation may be avoided in the future.

Id. at col. 12, l. 61–col. 13, l. 3. If not, the prediction is decremented toward the do not synchronize direction, and if after decrementing the prediction is below a predetermined limit, it is removed from the table. *Id.* at col. 13, ll. 5–17.

ILLUSTRATIVE CLAIM

1. In a processor capable of executing program instructions in an execution order differing from their program order, the processor further having a data speculation circuit for detecting data dependence between instructions and detecting a mis-speculation where a data consuming instruction dependent for its data on a data producing instruction of earlier program order is in fact executed before the data producing instruction, a data speculation decision circuit comprising:

- a) a predictor receiving a mis-speculation indication from the data speculation circuit to produce a prediction associated with the particular data consuming instruction and based on the mis-speculation indication; and
- b) a prediction threshold detector preventing data speculation for instructions having a prediction within a predetermined range.

BASIS OF PETITION

Petitioner asserts that claims 1–9 are unpatentable under 35 U.S.C. § 103 over U.S. Patent No. 5,666,506 filed on May 12, 1995 and issued on September 9, 1997, to Hesson (Ex. 1003, “Hesson”) in view of U.S. Patent No. 5,619,662 filed on August 12, 1994 and issued on April 8, 1997, to Steely (Ex. 1004, “Steely”). Petitioner does not assert any other challenges in the Petition.

CLAIM CONSTRUCTION

Citing the Declaration of its expert, Dr. Robert Colwell (“Colwell Decl.”), Ex. 1002 ¶ 45, Petitioner proposes that “data speculation circuit” be construed to mean “a circuit that detects data dependence between data consuming and data producing instructions and that detects mis-speculation.” Pet. 12. Citing the claim Construction Order in the Intel case (Ex. 1005, 9–10), Petitioner contends that its proposed construction is consistent with that applied by the district court, except that Petitioner’s proposal is not limited to “load” and “store” instructions. For purposes of this Decision, we adopt Petitioner’s proposed construction.

Petitioner proposes that “mis-speculation” is “when a data consuming instruction that is dependent for its data on a data producing instruction appearing earlier in the program order is in fact executed before the data producing instruction.” Pet. 13. Petitioner contends that this construction is consistent with that applied by the district court in the Intel case, except that the instructions are not limited to load and store instructions. *Id.* Petitioner contends, however, that even if the terms are limited to load, rather than data consuming instructions, and store, rather than data producing instructions, the prior art discloses such load and store instructions. *Id.* at 13 n.6. Patent Owner contends that the terms load and store are interchangeable with data consuming instructions and data producing instructions, respectively, in the ’752 Patent, and that there is no support for Petitioner’s contention that load and store are narrower than data consuming and data producing instructions. Prelim. Resp. 18–19. There is little, if any dispute that the prior art concerns load and store instructions, and neither party has disputed the application of the prior art on the basis that a data consuming instruction is a broader term

than a load instruction or that a data producing instruction is a broader term than a store instruction. Thus, for purposes of our analysis in this Decision, we need not determine whether there is any difference in the scope of the terms.

Petitioner further contends that “in fact executed” in its proposed construction means “when a data consuming instruction has actually accessed a memory address that has not yet been updated by a data producing instruction appearing earlier in the program order,” but does not mean “completed” because some instructions that have executed are squashed. Pet. 13–14. Patent Owner does not comment on Petitioner’s proposed construction. The specification of the ’752 Patent states that the processing unit has a “data speculation circuit for detecting data dependence between instructions and detecting a mis-speculation where a data consuming instruction dependent for its data on a data producing instruction is, in fact, executed before the data [producing] instruction.” Ex. 1001, col. 4, ll. 13–16. We apply this description from the specification as the meaning of the term “mis-speculation” and, thus adopt Petitioner’s proposed construction for purposes of this Decision. The specification does not use the term “in fact” in a way that requires further construction.

Petitioner contends that “predictor” should be construed as “a circuit that receives a mis-speculation indication from the data speculation circuit to produce a prediction.” Pet. 14. Petitioner contends that “prediction” is a “value that indicates the likelihood that the data speculative execution of a data consuming instruction will result in a mis-speculation.” *Id.* at 15. Noting that the district court in the Intel case initially applied a similar construction and later limited it to “capable of receiving ongoing updates,”

Petitioner argues that the district court's modification does not reflect the broadest reasonable construction. *Id.* at 15–16.

Patent Owner contends that the broadest reasonable interpretation of “prediction” is “a **variable** that indicates the likelihood that the data speculative execution of a load instruction will result in a mis-speculation.” Prelim. Resp. 13. Patent Owner disputes Petitioner's construction as inconsistent with the '752 Patent because it proposes a single unchanging value that indicates only the existence of a single instance of past mis-speculation, but is not capable of changing to reflect the likelihood of future mis-speculation. *Id.* According to Patent Owner, Petitioner's proposed construction writes out of the claim the temporal locality and on-going monitoring of mis-speculation history that prevents data speculation for instructions having a prediction within a predetermined range. *Id.* at 13–18. Patent Owner notes that the specification of the '752 Patent also describes the prediction being incremented or decremented under certain conditions. *Id.* at 18. We agree that in the '752 Patent, the mis-speculation prediction at any point in time is a function of the mis-speculation history of load-store instruction pairs. Pet. 16. Thus, the prediction is a variable. The fact that the prediction has a particular value at each point in time is merely an indication of its functional relationship and does not change the nature of the prediction from a variable to a constant value. Thus, we construe “prediction” as “a variable that indicates the likelihood that the data speculative execution of a load instruction will result in a mis-speculation.”

ART CITED IN PETITION

Hesson discloses an apparatus that permits load and store instructions to execute out of order. Ex. 1003, col. 1, l. 67–col. 2, l. 1. Hesson defines a

store violation condition as the situation wherein a load instruction that follows a store instruction in program order executes ahead of the store and produces the same real address. *Id.* at col. 4, ll. 12–16. A store barrier cache is used to predict dynamically whether a store violation condition is likely to occur and, if so, to restrict the issue of instruction to the load/store unit until the store instruction has been executed and it is once again safe to proceed with out-of-order execution. *Id.* at col. 2, ll. 1–7. The store barrier cache is accessed in parallel with an instruction cache and contains history bits used to predict the condition where a load instruction has executed ahead of a store instruction in program order and that the load and store instructions have the same address. *Id.* at col. 3, ll. 59–63. If the store barrier cache predicts a store-load conflict, a bit barrier bit is issued and no loads in the program order are permitted to execute ahead of the store that is predicted to be violated. *Id.* at col. 3, l. 63–col. 4, l. 1, col. 6, ll. 4–29. History bits record the persistence condition for each entry on the store barrier cache. *Id.* at col. 4, ll. 39–40. When a store instruction writes back formatted store data, the persistence of a store violation condition is monitored by comparing the store address against all newer load instructions that have completed. If the store violation condition persists, the store barrier bit is maintained, but if the violation condition is not detected when the store instruction data is about to be written to memory, the store barrier bits are updated. *Id.* at col. 6, ll. 42–45. Preferably, two history bits are used so that the store barrier bit is de-asserted after two successive write backs exhibit no store violation condition, although other approaches are possible. *Id.* at col. 6, ll. 44–50.

Steely discloses a pipelined processor with memory reference tagging with an instruction scheduler that can reorder the issuance of instructions from the instruction processor in which tags indicate whether memory reference instructions previously reordered and executed experienced a collision. Ex. 1004, Abstract, col. 2, ll. 64–66. The tags are compared during subsequent fetches of the instruction to ascertain whether the memory reference instructions can be reordered. *Id.* at col. 2, l. 66–col. 3, l. 1. In one embodiment, when a pair of load and store instructions cause a problem the first time through a write buffer, i.e., the write buffer determines a load instruction, e.g., instruction 1011, should not have been executed before a store, e.g., instruction 1007, a portion of the address in memory which resulted in the load-store collision, e.g., the five least significant bits, is stored in a memory tag store at a location determined by a line predictor address. *Id.* at col. 48, ll. 1–26. A write buffer tag circuit will associate these bits with load instruction 1011 and store instruction 1007. *Id.* at col. 48, ll. 26–30. The next time the instructions are called, the memory reference tag circuit provides the tag bits to the instruction scheduler and, if they are the same, the scheduler will not reorder the instructions. *Id.* at col. 48, ll. 30–50. As a result, the second time a troublesome load-store pair is encountered, the system will not reorder the load in front of the store. *Id.* at col. 48, ll. 51–53.

ANALYSIS OF PETITIONER’S PRIOR ART CHALLENGES

Claim 1 contains an extensive preamble written in the form “In a processor capable of executing program instructions in an execution order differing from their program order, the processor further having a data speculation circuit” There appears to be little, if any, dispute between

the parties that the elements in the preamble of claim 1 were known in the art. Claim 1 recites a predictor that receives from the data speculation circuit a mis-speculation indication associated with a particular data consuming instruction. The second limitation of claim 1 recites a prediction threshold detector that prevents data speculation for instructions having a prediction within a predetermined range.

Petitioner argues that, like the '752 Patent, both Hesson and Steely relate to reordering of instructions in pipelined processors. Pet. 28–29. Petitioner further contends that Hesson and Steely address the same problem because, at the time instructions are reordered, it is not known whether the instructions will access the same address, and therefore be dependent. *Id.* at 29 (citing Colwell Decl., Ex. 1002 ¶ 68).

Petitioner contends that both Hesson and Steely disclose a technique for determining whether to allow a load instruction to execute before a store instruction. *Id.* at 29–30. Petitioner also contends that Hesson and Steely approach the same problem using similar solutions, i.e., by storing information in a memory about instructions that have mis-speculated in the past and using that stored information to determine whether to allow speculative execution of currently pending instructions. *Id.* at 30. Petitioner concedes that Hesson does not describe expressly a process that associates a prediction “with the particular data consuming [*e.g.*, load] instruction” as recited in claim 1 of the '752 Patent. *Id.* at 27 (brackets in original). Although Petitioner contends that a person of ordinary skill would have considered that distinction to have been an obvious variation of the processor described in Hesson in view of the teaching in Steely, *id.* at 27–28 (citing Colwell Decl., Ex. 1002 ¶ 64), Patent Owner contends that the

inventors of the '752 Patent recognized that the goal of prediction in such systems need not be constrained, as in Hesson, or abandoned, as in Steely. Prelim. Resp. 9.

Petitioner notes each of the predictions (history bits) in Hesson is associated with a particular store instruction that previously contributed to a mis-speculation. Pet. 28 (citing Hesson's description of Table 1, Ex. 1003, col. 4, ll. 21–38, in which each line of a store barrier cache stores an address tag of a store instruction and its associated history bits). Petitioner notes that if Hesson's store barrier cache predicts a pending store instruction is likely to cause a store violation condition, Hesson delays all pending load instructions, whether or not a load instruction contributes to a store violation, until the store instruction completes. *Id.* at 34 (citing Ex. 1003, col. 6, ll. 3–8, 22–25; Colwell Decl., Ex. 1002 ¶ 74). Petitioner then cites Steely's disclosure that for each STORE that arrives, logic circuits in a write buffer search for all logically following LOADs that have preceded it and, if a matching physical address is found, a STORE-LOAD error is identified, indicating the occurrence of a mis-speculation that requires the load instruction to be re-executed. *Id.* at 30–31. Thus, Petitioner argues that, “unlike Hesson, Steely does not delay **all** load instructions upon detection of a problematic store instruction. Rather, Steely delays (or stops speculation for) **only** load instructions associated with problematic store instructions.” *Id.* at 35 (citing Colwell Decl., Ex. 1002 ¶ 75) (emphasis in original).

Noting Petitioner's concession that Hesson does not disclose “a prediction associated with the particular data consuming instruction,” as recited in claim 1 of the '752 Patent, Patent Owner emphasizes another acknowledgement by Petitioner, i.e., that each of Hesson's predictions is

associated with a data producing instruction, rather than a data consuming instruction, as recited in claim 1. Prelim. Resp. 19 (citing Pet. 16–17, 28). Patent Owner also notes that, unlike Hesson’s predictive technique, Steely permanently blocks from further execution load/store pairs that have been involved in a mis-speculation. *Id.* at 19–20. Patent Owner argues a person of ordinary skill would not have had a reasonable expectation of success in combining Steely and Hesson. *Id.* at 20.

Thus, our inquiry focuses on the propriety of Petitioner’s proposed combination, which substitutes the store-load pairs from Steely’s all-or-nothing approach, for the data producing store instructions in Hesson’s predictive approach. For the reasons discussed below, we are not persuaded that Petitioner has established it is reasonably likely to succeed in demonstrating that the combination would have been obvious to one of ordinary skill in the art.

Motivation to Combine – Available Alternatives

Petitioner contends that one reason a person of ordinary skill would have been motivated to incorporate Steely’s techniques of basing speculation on load-store pairs into Hesson’s predictor is that only a limited number of alternative approaches were available at the time of the invention claimed in the ’752 Patent. Pet. 35–36. According to Petitioner only three promising ways existed to track, and thus associate, predictions with instructions involved in load-store mis-speculations, i.e., (1) associating predictions with store instructions alone, as taught by Hesson; (2) associating predictions with the load instructions alone; and (3) associating predictions with load and store instructions as pairs, as taught by Steely. *Id.* (citing Colwell Decl., Ex. 1002 ¶ 77). Petitioner maintains that one of ordinary skill would have

recognized that “any of these three approaches could have been used in a predictor for predicting data dependencies in an out-of-order processor.” *Id.* However, Patent Owner provides evidence that persuasively disputes Petitioner’s assertions concerning a limited number of available alternative approaches to data dependency mis-speculation and determining whether instructions can be executed out of program order.

Patent Owner provides documentary evidence of contemporaneously known additional alternatives that speculate based on the actual value of the data, rather than the storage location of the data, Prelim. Resp. 21 (citing Ex. 2035³), and techniques that speculate based on a specific attribute of the storage location that a load will access, e.g., the address of the storage location that a load will access, *id.* (citing Ex. 2016⁴). Patent Owner identifies non-predictive approaches that focus on resolving, instead of predicting, whether load and store instructions are dependent. *Id.* at 22 (citing Ex. 1016⁵). Patent Owner notes methods that predict when a store instruction will have completed and a load instruction can proceed safely. *Id.* at 23 (citing Ex. 1009, 82–83, discussion of U.S. Patent No. 5,555,432 (“Hinton”) in file history of ’752 Patent). Patent Owner also identifies non-

³ Mikko H. Lipasti, Christopher B. Wilkerson, and John Paul Sheen, *Value Locality and Value Load Prediction*, pre-print of paper to be appear in ASPLOS-VII in Oct. 1996. We recognize the evidentiary issues concerning this Preprint and do not rely upon this document as prior art, but only for the proposition that alternatives other than those stated by Petitioner were available to be considered by persons of ordinary skill on the filing date of the application that matured into the ’752 Patent.

⁴ Mikko H. Lipasti and John Paul Shen, *Exceeding the Dataflow Limit via Value Prediction*, Proceedings of the 29th Annual ACM/IEEE International Symposium on Microarchitecture (1996).

⁵ U.S. Patent No. 6,463,523 B1, issued Oct. 8, 2002, col. 2, ll. 14–15.

speculative techniques, where no prediction is made on data dependency, *id.* at 24–26 (citing Ex. 2020⁶, describing dynamic memory disambiguation during program execution, using preload and check instructions), and techniques in which it is assumed that all prior stores are likely to access the same address as the subsequent block, so that there is no need to track, predict, or associate a prediction with a particular load or store instruction, Ex. 2008, 2.⁷

In view of the alternatives Patent Owner has demonstrated were available on the filing date of the application that matured into the '752 Patent, we are unpersuaded by Petitioner's argument that only three alternatives would be considered by a person of ordinary skill.

Reason to Combine – Performance

Petitioner also asserts that a person of ordinary skill would recognize that associating predictions with load-store pairs as taught by Steely could be advantageous in Hesson's processor. Pet. 36 (citing Cowell Decl., Ex. 1002 ¶ 78). According to Petitioner, a person of ordinary skill would recognize that halting speculation for only some load instructions, rather than all load instructions, could increase processing speed in Hesson. *Id.*

Noting that Hesson boasts performance within 1% of what is theoretically possible and that Hesson states that the frequency of mis-speculation is low, Patent Owner disputes any assertion that Petitioner has

⁶ David D. Gallagher et al., *Dynamic Memory Disambiguation Using the Memory Conflict Buffer*, ASPLOS-VI Proceedings (1994).

⁷ Adi Yoaz et al., *Speculation Techniques For Improving Load Related Scheduling*, Proceedings of 26th International Symposium on Computer Architecture (1999). Although the article was published in 1999, the relevant text discusses the approach taken in the P6 processor family [Intel96].

demonstrated one of ordinary skill would be motivated to incorporate the teachings of Steely into Hesson's near-perfect system. Prelim. Resp. 28–29. Citing Petitioner's acknowledgment that increased storage space for memory dependence management "may or may not be worth any concomitant improvement in processor performance," Patent Owner argues that one of ordinary skill would not have been motivated to risk impacting performance by increasing the complexity of Hesson's design. *Id.* at 27 (citing Pet. 37). In response to Petitioner's contention that a person of ordinary skill would seek to improve processing speed in Hesson, Patent Owner cites the disclosure in Hesson that "all load instructions are held up . . . a minimum amount of time." *Id.* at 29 (citing Ex. 1003, col. 6, ll. 22–23). Describing the use of store violation and persistence history information to control the dispatch of instructions, so as to prevent all load instructions *from* issuing until the store instruction that previously had been violated has proceeded successfully through the execution stage of the instruction pipeline, Hesson states:

The store instruction is permitted to execute once all its dependencies including the store data are resolved. Load instructions are once again permitted to proceed after the violated store instruction has executed and, as a result, not request the cache memory location that the store instruction is about to updated (sic). . . . Permitting *all* load instructions to proceed after the store instruction has executed versus after it has written back has two benefits. First, *all load instructions are held up at the instruction issue stage a minimum amount of time* and, second, history information on the persistence or lack of persistence of this store violation condition is provided. By dynamically controlling the issue and execution of load instructions according to a single violation condition and its history information using the store barrier cache 11, significant performance advantages are achieved over the prior art.

Ex. 1003, col. 6, ll. 11–30 (emphasis added). Hesson states specifically that permitting *all load instructions* to proceed only after the violated store has executed provides the advantage that *all load instructions* are held up for a minimum amount of time. *Id.* Thus, as Patent Owner points out, Hesson does not support Petitioner’s contention that a person of ordinary skill would be motivated to incorporate Steely’s approach, in which only some load instructions are held up based on store-load pairing, into Hesson to achieve improvements in processing speed.

Petitioner asserts that one of ordinary skill would have been aware of techniques for determining whether associating load-store pairs, as in Steely, could be beneficial and would improve processor performance. Pet. 37. However, as further evidence of the lack of motivation to combine, Patent Owner notes that initial publication of the invention in the ’752 Patent subsequently generated substantial skepticism about whether the increased hardware complexity the inventors were advocating and the constraints on table size would result in improved performance. Prelim. Resp. at 29 (citing Ex. 1015, col. 2, ll. 31–46,⁸ which references memory tagging described by Steely (*id.* at col. 1, l. 51–col. 2, l. 12) and Hesson (*id.* at col. 2, ll. 13–30) and criticizes a 1997 paper (Moshovos et al., *Dynamic Speculation and Synchronization of Data Dependence*) by the inventors proposing a solution based on a set of small, fully associative tables which hold load/store pairs that have caused memory violation in the past and are used to synchronize the execution of instructions to avoid future violations because the tables are complex and cumbersome to generate and maintain as well as being slow to

⁸ U.S. Patent No. 6,108,770, filed June 24, 1998, and issued on Aug. 22, 2000.

access and, given hardware constraints on table size, performance is significantly less than a perfect memory predictor would provide). Patent Owner also cites the testimony of Dr. William Dally, which discusses cost/benefit issues raised in Steely and states:

Increasing the complexity of a method for controlling data speculation adds die area and may increase cycle time. . . . [A] design that seems [feasible] on paper may be either infeasible or not worth implementing because it costs too much in terms of resources. Likewise, a design whose benefit is unclear is unlikely to be implemented. . . . Using what to a person of ordinary skill would have viewed as a complex technique like that disclosed in the Wisconsin Patent [the '752 Patent] is not obvious given cost, performance, and implementation considerations.

Ex. 2006 (“Dally Decl.”) ¶ 106. We further note that, citing deposition testimony by Simon Steely, Dr. Dally comments that Steely confirmed that in 1996 there was resistance to slowing down the critical path of a processor, as would result from adding a feature that might increase the number of cycles necessary to issue a LOAD, and that adding a complex table that has to be accessed before issuing a LOAD would have been viewed as potentially increasing the number of cycles necessary to issue a load. *Id.* ¶ 108.

Petitioner’s expert states that a person of ordinary skill would be motivated to consider associating predictions with load-store pairs rather than a store instruction alone because a larger instruction window, i.e., the number of instructions that can be executed out-of-order relative to one another, generally could include more load instructions that could execute speculatively without mis-speculating and thus could benefit from not being delayed unnecessarily. Ex. 1002 ¶ 79. According to Patent Owner, Hesson

itself suggests otherwise because Hesson states that it protects against unanticipated dependencies between a known problematic store instruction, while holding up impacted load instructions only a “minimum amount of time,” thereby avoiding a “penalty” from mis-speculation that is “quite severe.” Prelim. Resp. 32 (citing Ex. 1003, col. 1, ll. 44–46, col. 6, ll. 2–8, 22–29). Patent Owner further argues that, because speculation that would not be allowed by Hesson would be allowed for a previously un-encountered load that is ambiguously dependent on a known problematic store, under Petitioner’s proposed modification of Hesson the risk of a “quite severe” “penalty” resulting from mis-speculation actually increases. *Id.* at 32–33. *See also id.* at 38–41 (describing in more detail increased risks that would discourage a person of ordinary skill from Petitioner’s proposed combination). Patent Owner also argues that a person of ordinary skill would not have been motivated to make the asserted combination to reduce “false dependencies” because evidence demonstrates that removing such “false dependencies” does not improve performance. *Id.* at 36, 41 (citing Ex. 1011, 9 (col. 1, § 6.2)).

In consideration of the above, we are not persuaded by Petitioner’s arguments that a person of ordinary skill would recognize that significant processing speed or other performance improvements could be achieved by combining the teachings of Hesson and Steely.

Reason to Combine – Integrating Steely Into Hesson

Petitioner argues that one of ordinary skill readily would have integrated Steely into Hesson by adding into Hesson’s store barrier entry cache for the violated store condition a single field identifying the load instruction address. Pet. 38. Petitioner states that, by modifying Hesson’s

store barrier cache, a problematic store instruction would delay only the load instructions stored in the same entry of the cache. *Id.* Although Petitioner acknowledges that other modifications may be necessary to implement this addition to the store barrier cache, Petitioner contends such additional modifications would have been routine, *id.* (citing Colwell Decl., Ex. 1002 ¶ 83), and that adding a field to produce a store-load association would have been a “simple and obvious modification of Hesson’s predictor” well within the ability of one of ordinary skill, *id.* at 39. Citing U.S. Patent No. 5,584,001 (Ex. 1012, “Hoyt”), which relates to branch prediction, Petitioner contends that persons of ordinary skill would already have been aware of the instances where two addresses and a prediction were stored in an entry of a cache for purposes of performing speculation in an out-of-order processor. *Id.* However, Petitioner does not assert Hoyt as a basis of its challenge in this proceeding.

Patent Owner argues that Petitioner fails to address the unique difficulties associated with modifying Hesson in the store/load context. Noting that loads occur far more often than stores, Patent Owner emphasizes that Petitioner fails to explain how a person of ordinary skill modifying Hesson would address multiple loads depending on a single store. Prelim. Resp. 42–44.

Patent Owner contends that, in contrast to the simple binary branching in Hoyt, the tracking hardware and data structures required for data speculation would be far more complex. *Id.* at 43. Patent Owner further points out that Petitioner does not discuss factors that would dissuade a person of ordinary skill, including whether to implement the table with additional columns or rows to accommodate multiple loads depending from

a single store. *Id.* According to Patent Owner, even if the modification involves simply creating a new row in the prediction table for each colliding store/load pair, the number of rows in Hesson's store barrier entry cache would increase significantly given the greater population of loads over stores. *Id.* at 44. Patent Owner notes that neither the corresponding additional storage space nor the corresponding longer search time through the store cash would be desirable to a person of ordinary skill. *Id.* Although Hesson discloses that the incidence of dependencies is quite low, Ex. 1003, col. 1, ll. 42–44, making it difficult to estimate the corresponding increase in the size of the store entry barrier cache, Patent Owner cites the testimony of the inventor of the Steely reference that increasing the number of cycles it takes to issue a load in the critical path is viewed as undesirable. Prelim. Resp. 44–45 (citing Ex. 2006 ¶¶ 107–109).

In view of the above, we are not persuaded by Petitioner's arguments that a person of ordinary skill would combine the teachings of Hesson and Steely because they can be integrated readily.

Therefore, we are not persuaded that Petitioner has demonstrated it is reasonably likely to succeed in its challenge to claim 1 based on the combination of Hesson and Steely and we do not institute *inter partes* review on this challenge to claim 1.

Remaining Claims

Claims 2–8 depend from claim 1. Because we are unpersuaded that Petitioner has demonstrated it is reasonably likely to prevail in demonstrating claim 1 is unpatentable under 35 U.S.C. § 103, we decline to institute an *inter partes* review on Petitioner's challenges to claims 2–8.

Independent claim 9 recites a prediction table that communicates with the data speculation circuit to create an entry listing a particular data producing instruction and data consuming instruction each associated with a prediction when a mis-speculation indication is received. Petitioner cites the same combination of Hesson and Steely as disclosing this limitation. Pet. 56–57. However, as discussed above, we are not persuaded that Petitioner has demonstrate a person of ordinary skill would be motivated to combine Hesson’s approach, which is based on varying predictions based on mis-speculations associated with store instructions, with Steely’s approach that permanently prevents further executing loads when a mis-speculation occurs for a load-store pair. Therefore, we are not persuaded that Petitioner has demonstrated it is reasonably likely to succeed in its challenge to claim 9 based on the combination of Hesson and Steely and we do not institute *inter partes* review on this challenge to claim 9.

MOTIONS TO SEAL

Patent Owner’s First and Second Motions To Seal

Patent Owner filed a Motion to Seal, Paper 12 (“Patent Owner’s First Motion to Seal”), in which it moved to seal Exhibits 2006, 2021, and 2025, which are subject to a protective order entered in the Intel case. Patent Owner’s First Motion to Seal 1. Patent Owner’s First Motion to Seal also requested that portions of its Patent Owner Preliminary Response (Paper 10, “Unredacted Patent Owner Preliminary Response”) be maintained confidential because it quotes from Ex. 2021. Patent Owner’s First Motion to Seal 2–3. We have not relied upon the Unredacted Patent Owner Preliminary Response in this Decision. Therefore, we dismiss Patent Owner’s Motion to Seal its Unredacted Patent Owner Preliminary Response

as moot. We grant Patent Owner's Motion to Seal Ex. 2021 and 2025, because they are subject to the protective order in the Intel case (Ex. 2043), Patent Owner has explained why the material constitutes "confidential information" under 35 U.S.C. § 316(a)(7), and we have not cited them in our Decision.

Ex. 2006, the Dally Declaration, presents another issue, because we cite Ex. 2006 in this Decision. Redacted Patent Owner Preliminary Response, Paper 11, which we have cited extensively in the Decision, includes certain references to Ex. 2006. Patent Owner's First Motion to Seal states that Ex. 2006 is "[a]n excerpt of a rebuttal . . . report[] generated in the Intel Action." Patent Owner's First Motion to Seal 2. Patent Owner's First Motion to Seal also states that "[r]edacted portions of Exhibit 2006 include direct quotes from confidential recommendation letters, email correspondences and/or peer-reviews. These should be maintained under seal because those communications were made in confidence." *Id.* at 3. Patent Owner, however, has not provided a redacted version of Exhibit 2006, and has not indicated which portions correspond to this allegedly confidential material. Although Patent Owner presents Dr. Dally's expert testimony in support of its positions in the Redacted Patent Owner Preliminary Response, Patent Owner designated the entirety of Dr. Dally's testimony in Ex. 2006 as confidential. Patent Owner has not provided a sufficient basis to determine whether the portions of Ex. 2006 that we cite in our Decision are confidential. Confidential information relied upon in a decision to grant or deny a request to institute ordinarily will be made public. Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,760 (Aug. 14, 2012). Therefore, because Patent Owner has designated the

entirety of Ex. 2006 confidential, Patent Owner's Motion to Seal Ex. 2006 is denied. However, in view of the circumstances, Patent Owner may file a redacted version of Ex. 2006 that does not redact those portions of the Ex. 2006 cited in this Decision and request that the unredacted version of Ex. 2006 be expunged, as discussed below.

We authorized Patent Owner to file a Supplemental Patent Owner Preliminary Response (Paper 14) that includes material the parties contended is confidential. Patent Owner has moved to seal certain deposition transcript excerpts, i.e., Exhibits 2041 and 2042, and portions of its Unredacted Supplemental Patent Owner Preliminary Response. Paper 16 ("Patent Owner's Second Motion to Seal"), 1. We do not rely on these documents in this Decision, and, therefore, dismiss Patent Owner's Second Motion to Seal as moot.

Petitioner's Motion To Seal

We also authorized Petitioner to file a Reply to the Supplemental Patent Owner Preliminary Response ("Petitioner's Reply"). Paper 18. Petitioner filed an unredacted Petitioner's Reply (Paper 20), which Petitioner designated confidential because it cites to unredacted versions of Exhibits 1020 and 1021, which Petitioner also designated confidential. Petitioner moves to seal these materials. Paper 19 ("Petitioner's Motion to Seal"). Petitioner also filed a redacted Petitioner's Reply. Paper 21.

We do not rely on these documents in this Decision, and therefore, dismiss Petitioner's Motion to Seal as moot.

Expunging Confidential Information Not Relied Upon in a Decision

We remind the parties of the expectation that confidential information relied upon in a decision to grant or deny a request to institute or identified

in a final written decision will be made public. Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,761 (Aug. 14, 2012). Confidential information that is subject to a protective order ordinarily becomes public 45 days after denial of a petition to institute or 45 days after final judgment in a trial. A party seeking to maintain the confidentiality of the information may file a motion to expunge the information from the record prior to the information becoming public. 37 C.F.R. § 42.56.

SUMMARY

We do not authorize an *inter partes* review on any of the grounds asserted in the Petition.

ORDER

In consideration of the foregoing, it is hereby:

ORDERED that the Petition is DENIED;

FURTHER ORDERED that Patent Owner's First Motion to Seal is GRANTED-IN PART, DISMISSED-IN-PART, AND DENIED-IN-PART. Patent Owner's First Motion To Seal is GRANTED with respect to Exhibit 2021 and Exhibit 2025, DISMISSED with respect to Patent Owner's Unredacted Patent Owner Preliminary Response (Paper 10), and DENIED with respect to Exhibit 2006, with leave to file a redacted version of Ex. 2006 with a certification that no material cited in this Decision has been redacted;

FURTHER ORDERED that Patent Owner's Second Motion to Seal is DISMISSED;

FURTHER ORDERED that Petitioner's Motion to Seal is DISMISSED; and

IPR2014-01567
Patent 5,781,752

FURTHER ORDERED that Papers 10, 14, 20, and Exhibits 1020, 1021, 2041, and 2042, are expunged from the record of this proceeding.

PETITIONER:

Richard Goldenberg
Lauren B. Fletcher
WILMER CUTLER PICKERING HALE & DORR LLP
richard.goldenberg@wilmerhale.com
lauren.fletcher@wilmerhale.com

PATENT OWNER

Gary N. Frischling
Hong A. Zhong
IRELL & MANELLA LLP
gfrischling@irell.com
Apple-WARF_IPRGroup@irell.com