EDITOR: Joshua J. Yi, josh@joshuayipatentlaw.com

DEPARTMENT: MICRO LAW

# A Review of *Wisconsin Alumni Research Foundation v. Apple*—Part XI

Joshua J. Yi, *The Law Office of Joshua J. Yi, PLLC, Austin, TX, 78750, USA*

Part I of this series introduced the *Wisconsin Alumni Research Foundation v. Apple* cases and described the asserted patent (U.S. Patent No. 5,781,752). That article also summarized some recent large verdicts for patents asserted by academic institutions and provided several reasons why this series may be of interest to the readership of *IEEE Micro*, most notably because the inventors are well-known and several well-known computer architects worked as experts on this case. Part II described the complaints; namely, it described the plaintiff, Wisconsin Alumni Research Foundation ("WARF"), the inventors, and WARF's allegations as to how the Apple's products infringed WARF's patent. Part III described Apple's answer to the allegations in WARF's complaint, Apple's counterclaims, and WARF's response to those counterclaims. Part IV examined Apple's allegation of inequitable conduct by the inventors, a technical analysis of that allegation, and Judge William M. Conley's legal analysis of the sufficiency of Apple's allegations. Parts V, VI, and VII reviewed three of WARF's motions to compel Apple to produce 1) executable versions of the simulators for its A6, A7, and A8 processors (hereinafter "accused products"), 2) documentation regarding future (unreleased) processors, and 3) a version of each accused processor that had the accused feature disabled, and the hearings that the court had on those motions. Parts VIII, IX, and X described what claim construction is, what claim terms were disputed both in these cases and in the prior *Intel* case, and an analysis of select claim terms.

## INTRODUCTION

The purpose of the claim construction is to interpret the meaning of claim terms, which helps define the scope of the claims. Then, based on that scope, the parties can prepare their arguments so the jury (or the court in some cases) can determine if the accused products infringe the asserted claims and/or are invalid.

Judge Barbara B. Crabb presided over the prior *Intel* case, and the early part of the *Apple* case. As part of the *Intel* case, she construed five terms from this patent. Because those claim terms were argued in the *Intel* case, the arguments were, obviously, made by WARF and Intel (and not Apple).

The last article in this series reviewed the parties' opening arguments for one of the five claim terms, the "data speculation circuit" term. This article reviews the parties' arguments in their responsive briefs. This term appears in Claim 1 of the Asserted Patent as follows:

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.

The parties proposed constructions for this term are shown in Table 1.

## SECOND DISPUTED CLAIM TERM: "DATA SPECULATION CIRCUIT"

From their proposed constructions, the parties appear to agree that a "data speculation circuit" is a circuit that detects a data dependence and misspeculation


Digital Object Identifier 10.1109/MM.2026.3716097
Date of current version 18 August 2026.

**TABLE 1.** Disputed claim term discussed in this article.

| Term | WARF's Proposed Construction | Intel's Proposed Construction |
|---|---|---|
| #2: Data speculation circuit | A circuit that detects data dependence between LOAD and STORE instructions and tracks execution of such instructions in order to detect any mis-speculations arising from the data speculative execution of LOAD instructions | A circuit that detects data dependence between load/store pairs and detects a mis-speculation by a load instruction that is in fact executed |

involving load and store instructions. There are two key differences between the parties' proposed construction. First, Intel's proposed construction requires that the detected data dependence is between a "pair" of load and store instructions, whereas WARF's proposed construction only requires a data dependence between load and store instructions. Second, Intel's proposed construction requires that the load and store pair be "in fact executed," whereas WARF's proposed construction does not contain that phrase.

## WARF'S RESPONSIVE CLAIM CONSTRUCTION BRIEF REGARDING "DATA SPECULATION CIRCUIT"

WARF said that the parties agree that the Asserted Patent "identifies a conflict between a LOAD instruction[a] and a STORE instruction (i.e., it takes two to mis-speculate, a STORE and a LOAD)" but disagree to what degree the "data speculation circuit" will prevent speculation: 1) for all prior pending stores (WARF's position) based on a prediction indicator or 2) only when there is a conflict between a specific load/store pair (Intel's position) (see pages 5 to 6 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

WARF first argues that the inventors made two observations regarding the execution of speculative loads: 1) Only a few load/store pairs cause the majority of data misspeculations and 2) if a load conflicted with a store in the past, that load is very likely to conflict with that store again in the future (see page 7 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). Based on that, WARF argues that "[t]hese two observations led the inventors to conceive a broad invention that generally seeks to delay the LOAD instructions that are causing the majority of data mis-speculations and then seeks to release the delayed LOAD instructions as soon as possible" (see page 7 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

[a]This article uses lowercase for "load" and "store" unless the parties capitalize them, at which point this article follows the parties' formatting when quoting the parties' briefs.

WARF explains that the Asserted Patent discloses both a prediction table and a synchronization table, wherein each row of the former contains a load instruction and a store instruction, and a value that indicates the likelihood of a conflict between the two and is used to potentially delay the speculative execution of the load (see page 8 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF explains that the Asserted Patent describes that the synchronization table indicates that the load instruction is still waiting on a prior pending store instruction (see page 8 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

WARF contends that the inventors "specifically contemplated that the [Asserted Patent] could be selectively implemented in a three-tiered manner where each tier would utilize different portions of the prediction and the synchronization table[s]" and where different claims "mirror" the various tiers (see page 9 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). In particular, WARF asserts that (see pages 9 to 10 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]):

- **First Tier**: "The processor checks whether the LOAD instruction has mis-speculated in the past. If the LOAD instruction does not have a history of mis-speculation, it is executed speculatively."
- **Second tier**: For loads with a history of misspeculation, "the processor delays the LOAD instruction if the prediction indicates so."
- **Third tier**: When a load is delayed, the processor puts it in the synchronization table and allows the load instruction to execute after the related store instruction executes.

With respect to the third tier, WARF argues that even if the related store instruction executes, the load instruction may still depend on another prior store, which could potentially cause a misspeculation (see page 13 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF asserts that while Claim 1 uses the First and Second Tiers, because it fails to mention the synchronization table, it does not use the Third Tier (see page 10 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). By contrast, WARF asserts that Claim 5 uses all three tiers

(see page 10 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF argues that Intel is trying to require that Claim 1 also use the Third Tier despite the fact that Claim 1 does not mention the synchronization table and despite the fact that case law prohibits limiting claims to embodiments (i.e., examples) in the specification unless the inventors evinced a clear intent to limit the claims to those embodiments (see pages 10 to 11 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

WARF argues that, contrary to Intel's position that the Asserted Patent delays a load instruction based on a particular store instruction, the Asserted Patent discloses delaying a load based only on the associated prediction indicator (see pages 11 to 14 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF argues that Figure 3 (see Figure 1) from the Asserted Patent supports its argument by showing that while block 66 checks whether the load instruction is a data speculative load, i.e., one where there is a prior store instruction that the load instruction may conflict with, the circuit does not equivalently check the store in the load/store pair (see page 12 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

Based on the above, WARF argues that Intel is "mistaken in arguing that the [Asserted Patent] operates only on the 'load/store pair' basis" and that it is legally improper to limit an invention to the embodiments disclosed in the patent (see page 14 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

WARF further argues that block 80 in Figure 3 (see Figure 1) depicts how the data speculation circuit handles delayed loads; namely, that the loads wait until they are squashed, no longer data speculative, i.e., the pending load does not conflict with any prior pending stores, or are woken up (see pages 15 to 16 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF argues that for the loads that are woken up, block 76 depicts that those loads wait until they either are squashed or are no longer data speculative (see pages 16 to 18 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). In other words, delayed loads are either squashed or access the memory hierarchy once they are no longer data speculative.

WARF contends that Claim 1 does not even use the word "pair" and that Intel is incorrect that Claim 1 does (see page 25 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). More specifically, WARF argues it is well-known that a load can potentially depend on several stores, which the preamble describes by requiring "a data consuming instruction dependent for its data on a data producing instruction of earlier program order," where "a" in patent law generally means "one or more" (see page 25 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). In other words, the use of "a" before "data producing instruction of earlier program order" means that the load

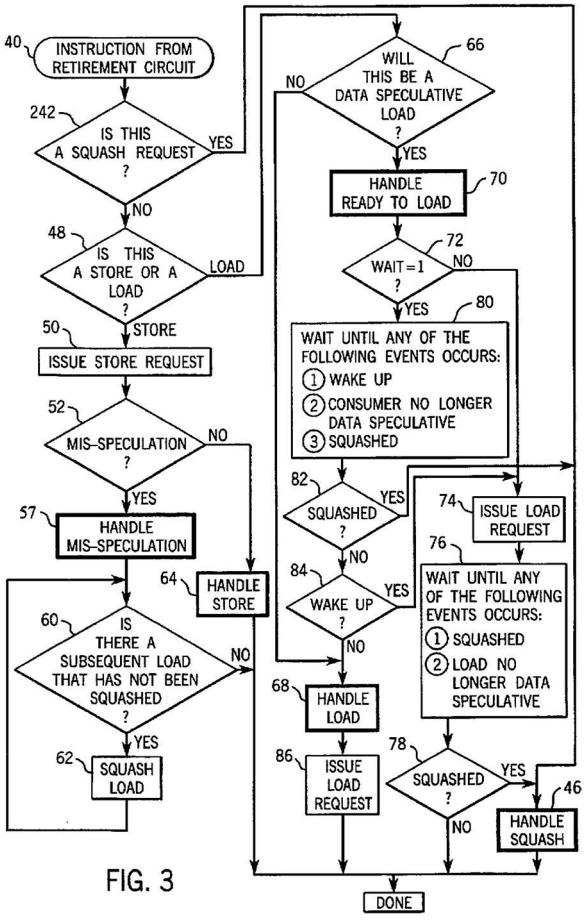


**FIGURE 1.** Figure 3 from U.S. Patent No. 5,781,752.

instruction is dependent on one or more stores, which means that Intel is incorrect that the data speculation circuit operates on specific load/store pairs (see page 25 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

## INTEL'S RESPONSIVE CLAIM CONSTRUCTION BRIEF REGARDING "DATA SPECULATION CIRCUIT"

Intel first argues that while the word "pair does not appear in Claim 1, the plain reading of the claim shows that the instructions at issue are a store instruction and a dependent load instruction, i.e., what the specification calls a 'pair'" (see page 8 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

With respect to WARF's argument that the claim language recites "prediction associated with the particular data consuming instruction," Intel argues that WARF ignores other claim language, such as the preamble (which are the words immediately following the claim number, i.e., "[i]n a processor capable of executing

program instructions"), that describes that the load is paired with an earlier store ("detecting a mis-speculation where a data consuming instruction dependent for its data on a data producing instruction of earlier program order") (see page 8 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel argues that the patent repeatedly describes that "pairs" are part of "the invention" (see page 9 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel further argues that load/store pairs are not only an embodiment, but the only embodiment in the specification, and that the binding case law requires that "claim terms should be limited to what the specification describes as the invention" (see pages 9 to 10 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel argues that WARF acknowledged that pairs are part of the invention on two occasions (see page 10 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel first asserts that WARF argued that the Asserted Patent is better than a prior art patent because the Asserted Patent blocks speculative execution of load instruction that conflict with a prior store (see page 10 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel next asserts that WARF described that the Asserted Patent was "intelligent[]" because it identified load/store pairs that are likely to misspeculate again in the future and prevents the load from speculatively executing (see page 10 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel argues that WARF is mistaken that the Asserted Patent discloses that the invention can be divided into multiple "tiers" (see page 10 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). In its opening claim construction brief, WARF argued that there were

> three decision points associated with the execution of LOAD instructions: (1) whether the LOAD instruction is the prediction table; (2) if it is, whether the LOAD instruction should be delayed, i.e., whether data speculation should be prevent[ed] for the LOAD; and (3) if it is delayed, how long should the LOAD instruction be delayed.

(See page 28 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[3]) Based on that, WARF argued that Claim 1 was only directed toward the First Tier and the Second Tiers, while load/store pairs were limited to the Third Tier (see pages 28 to 29 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[3]).

Intel contends that while the patent mentions once there is a "three-tier approach," it does not explain what that approach is, let alone indicate that Claim 1 is directed toward the First Tier and the Second Tier, and that load/store pairs are limited only to the Third Tier (see page 12 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Rather, Intel asserts that the fact that the patent discloses that the synchronization table is used in the Third Tier does not exclude its use in the Second Tier (see page 12 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel further asserts that the synchronization table is the only structure that the Asserted Patent discloses to prevent load misspeculation (see page 12 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel argues that WARF's description of how the invention works ignores the store half of the pair (see page 13 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). More specifically, Intel argues that "[a]ccording to WARF, each time a load is encountered, the prediction table is searched for **that load**, and a determination is made with respect to speculative execution of that load" (see page 13 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel argues that the "system is **not** concerned with a load instruction that mis-speculates too often; the system is concerned with a load instruction that mis-speculates too often **when paired with a particular store**" (see page 13 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel asserts that "[i]t is only a mis-speculation by a load/store pair that is logged in the prediction table and assigned a prediction value[,]" which is the "central insight" of the Asserted Patent, namely, that "most data dependence mis-speculations can attributed to a few static STORE/LOAD pairs" so it only makes sense to track the pairs (see page 14 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel argues that while the Asserted Patent "refers to a load instruction that mis-speculates, and to a prediction value for a load instruction[,]" the patent also repeatedly describes that the load instruction is handled as part of a **pair** of instructions (see pages 14 to 15 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Based on the above, Intel contends that "while it is correct, in part, to say that a load instruction mis-speculates, the full context of the patent reveals that, from the perspective of the [Asserted Patent], the load mis-speculates **as part of a load/store pair**"(see page 15 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel contends that

> WARF's attempt to expand claim 1 to treatment only of a load instruction: (a) finds no support in the patent specification, (b) is contrary to the language of claim 1 (which refers to both a load and the store on which it depends), and (c) perhaps most importantly, contradicts the alleged insight that animated the patent.

(See page 15 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]) Intel asserts that WARF's argument that the "data speculation circuit" should not be limited to load/store

pairs "rests on a faulty premise" (see page 20 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). More specifically, Intel asserts that WARF's argument is that while prior art based its predictions, in part, on stores, the Asserted Patent is broader than that and should not be limited to the embodiment that discloses predictions based on load/store pairs (see page 20 to 21 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). But Intel asserts WARF's argument is faulty because while the claims could have been written more broadly, that is not a reason to expand the claims in light of the claims and the description in the Asserted Patent (see page 20 to 21 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

## JUDGE CRABB'S CLAIM CONSTRUCTION ORDER

After holding a hearing, Judge Crabb entered a claim construction order where she decided that the meaning of "data speculation circuit" meant "a circuit that detects data dependence between load and store instructions and that detects mis-speculation by load instructions" (see page 4 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]).

Judge Crabb first found that Intel is "mistaken" that the Asserted Patent's description of load/store pairs is description of the invention as a whole for a few reasons (see page 4 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]). First, Judge Crabb held that while the inventors clearly understood the importance of load/store pairs, they chose not to use the word "pair" in the claim language (see page 5 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]).

Second, Judge Crabb held that while there is an embodiment in the Asserted Patent directed toward load/store pairs, the description in the Asserted Patent does not support the conclusion that that is the only possible embodiment (see pages 4 to 5 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]). Furthermore, based on the description in the Asserted Patent that the predictor looks at the prediction table for "*the* particular instruction 8.2 identified by *its* physical address," Judge Crabb held that the focus is on one instruction and not a particular load/store pair (see page 6 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]).

Third, Judge Crabb held that because Claim 3, which depends on Claim 1, first introduces a "prediction table," Claim 1 cannot also require a prediction table or that would make the language in Claim 3 introducing a prediction table to be meaningless, which courts generally disfavor (see page 7 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]). Based on that, Judge Crabb found that "the use of load/store pairs in a prediction table is not evidence that data speculation circuit is limited to load/store pairs" (see page 7 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]).

Judge Crabb also found that the Asserted Patent describes "checking load instructions, and not load/store pairs, when a mis-speculation occurs" (see page 8 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]).

Finally, Judge Crabb wrote that "one purpose of the invention is to 'identify data dependencies on an on-going or dynamic basis,' which would not 'be furthered by focusing on only specific load/store pairs' because there may be mis-speculations with other store instructions" (see page 8 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]).

Judge Crabb concluded that the phrase "that requires the circuit to track instruction execution" in WARF's proposed construction should not be included (see page 9 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]). More specifically, Judge Crabb concluded that while the Asserted Patent mentions this phrase, because the claim does not, it would improper to limit the claim language to this particular requirement (see page 9 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[4]).

## AUTHOR'S COMMENTS AND ANALYSIS

I agree with Judge Crabb's conclusion and reasons not to limit a "data speculation circuit" to load/store pairs as Intel proposed. That said, I generally disagree with her construction (i.e., definition) for a couple of reasons. First, the term "data speculation circuit" is one that a person of ordinary skill in the art would easily understand. Patent law does not require that the judge provide a construction for terms of art such as this one. Rather, what other judges frequently do is just provide a construction of "plain-and-ordinary meaning," which is the meaning that a person of ordinary skill in the art would understand this term to have, i.e., a circuit for data speculation.

Second, the plain-and-ordinary meaning of a "data speculation circuit" is not limited to detecting a data dependence between a load and a store only, but it can also detect data dependences between other types of instructions, such as arithmetic and logical instructions. Therefore, it is incorrect to limit the type of instructions that a "data speculation circuit" could operate on to loads and stores.

While Judge Crabb's construction was based on the parties' proposed constructions and the description in the Asserted Patent, which was focused on memory instructions, it is legally improper for a judge provide a construction for term that redefines the plain-and-ordinary ordinary meaning of the term unless the inventors chose to clearly and unmistakably redefine the

claim term (i.e., to be limited to memory instructions). Here, however, neither side argued that the inventors did so and Judge Crabb also did not conclude that the inventors did so.

While the inventors chose to use the term "data speculation circuit" in the Asserted Patent, a more accurate phrase for this term, based on how the Asserted Patent discusses this term, would be "memory dependence predictor" or "speculative load execution predictor."

Given that "data speculation circuit" is not limited to loads and stores, why did Intel pick this term to be construed and attempt to limit it to load/store pairs? With respect to the latter question, Intel likely proposed their construction because the "data speculation circuit" in their products does not operate on load/store pairs, but operates on loads only without considering stores and/or the circuit recognized that a load that could depend on multiple stores. As such, if Judge Crabb adopted Intel's construction, then Intel's products would not infringe as they do not operate in the same manner that is covered by Claim 1. With respect to the former question, Intel likely picked this term as Claim 1 is very short, so Intel did not have a lot of terms that they could choose from to introduce the concept of load/store pairs into a term, and thus into Claim 1.

The next article in this series will continue describe the parties' arguments for the next term, the court's analyses and reasoning, and my analysis.

## REFERENCES


1. Wis. Alumni Res. Found. v. Intel Corp., No. 3:08-cv-00078, ECF No. 54, Jul. 24, 2008 (W.D. Wis.).
2. Wis. Alumni Res. Found. v. Intel Corp., No. 3:08-cv-00078, ECF No. 51, Jul. 24, 2008 (W.D. Wis.).
3. Wis. Alumni Res. Found. v. Intel Corp., No. 3:08-cv-00078, ECF No. 29, Jun. 26, 2008 (W.D. Wis.).
4. Wis. Alumni Res. Found. v. Intel Corp., No. 3:08-cv-00078, ECF No. 65, Sep. 18, 2008 (W.D. Wis.).


**JOSHUA J. YI** is a solo practitioner at The Law Office of Joshua J. Yi, PLLC, Austin, TX, 78750, USA. Contact him at josh@joshuayipatentlaw.com.