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

DEPARTMENT: MICRO LAW

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

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 and IX 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' arguments and the court's construction for one of the five terms, the "in fact executed" claim term. The analysis of the parties' arguments and court's decision for the second term, "data speculation circuit," will be split between this article and the next due to the length of the parties' arguments. This term appears in Claim 1 of the '752 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 as shown in Table 1.

Digital Object Identifier 10.1109/MM.2026.3697951
Date of current version 17 June 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

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 mis-speculation involving load and store instructions. In general, it is unnecessary to construe this term when a term has a well-understood and well-known meaning to a person of ordinary skill in the art, which is the standard that patent law uses for claim construction. Furthermore, the meaning of this term is self-evident based on the words of the term: i.e., a "data speculation circuit" is a circuit that performs data speculation. In other words, it is unnecessary to provide a construction for a term that is well understood in the field and/or has an obvious meaning from its constituent words.

FROM THEIR PROPOSED CONSTRUCTIONS, THE PARTIES APPEAR TO AGREE THAT A "DATA SPECULATION CIRCUIT" IS A CIRCUIT THAT DETECTS A DATA DEPENDENCE AND MIS-SPECULATION 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 OPENING CLAIM CONSTRUCTION BRIEF REGARDING "DATA SPECULATION CIRCUIT"

In its opening claim construction brief, WARF did not argue that this term needed to be construed; as such, it did not provide a proposed construction for this term at that time or make any arguments. That said, the arguments that WARF provided for the "prediction" claim term in its opening brief are essentially the same for this term. As such, this section describes the relevant arguments from WARF for "predictor."

WARF argued that the language of Claim 1 (and other claims) indicates that a "prediction" is associated with a particular load and not a particular load/store pair (see page 24 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). For example, Claim 1 recites that the predictor produces a "prediction associated with the particular data consuming instruction [i.e., a load instruction] and based on the mis-speculation indication[.]" WARF argued that because other claims expressly required both loads and stores, Claim 1's failure to do so indicates that a "prediction" is associated with loads and not a load/store pair (see page 25 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

WARF argued that "[t]he reason the inventors chose to associate prediction with a LOAD instruction and not a LOAD/STORE pair is simple—data speculative execution applies only to LOAD instructions, STORE instructions are never speculatively executed" (see page 25 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF further argued that because stores do not speculatively execute, "predictions" are only associated with loads and not associated with load/store pairs (see page 25 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). As such, the "data speculation circuit" only generates mis-speculations for loads and not load/store pairs.

WARF argued that "the patent describes that predictions are for loads alone and not load/store pairs" (see pages 26–28 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). In particular, WARF cited to Figures 3 and 4 in the '752 Patent (see pages 26 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). For example, decision block 48 in

Figure 3 checks the whether the memory instruction is a load or a store, and then handles each in a different way (see Figure 1).

WARF argued that the patent describes that decision block 66 in Figure 3 checks "whether this is a data speculative LOAD, that is whether there are prior STORE instructions on which it might depend" (see page 26 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF further argued that because a single load may depend on multiple prior stores, it does not make sense to require that the prediction be directed towards load/store pairs (see page 26 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

WARF argued that "[n]otably, the inventors did not state whether the data speculation for a LOAD/STORE pair can occur or not" (see page 27 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]

WARF argued that the patent discloses that the invention could be implemented in a "three-tiered approach" with

> 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 prevents 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.*[1]) WARF argues that the claims of the patent "mirror" those three tiers (see page 28 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF asserted that Claim 1 reflects the second tier only, and that the second tier does not involve the corresponding store instruction (see page 28 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

WARF contended that the patent describes that if a load instruction should be delayed (third tier), then "a synchronization table is used to determine when the instruction is to be performed" (see page 28 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]). WARF argued that while the synchronization table contains an identifier for store instructions, this information is only needed for the third tier and not whether it is required for a prediction of mis-speculation (see page 29 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[1]).

INTEL'S OPENING CLAIM CONSTRUCTION BRIEF REGARDING "DATA SPECULATION CIRCUIT"

Intel argued that while its proposed construction requires that the data speculation circuit detects a data dependence between a load/store pair, WARF's proposed construction requires it to detect a data dependence between "load and store instructions" (see page 28 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel contended that "pairs" should be included in the court's construction for this term because 1) "the plain meaning of the claim language is that the two instructions under consideration are a load/store pair," 2) the patent describes that the invention is "identifying—and preventing the speculative execution—of load/store pairs that cause repeated mis-speculations," and 3) there is no evidence that the inventors contemplated any other situation other than load/store pairs (see page 28 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

With respect to the first reason, Intel argued that because Claim 1 describes that "detecting a mis-speculation where a data consuming instruction dependent for its data on a data producing instruction of earlier program order," this claim language describes that a specific load/store pair (i.e., the "data consuming

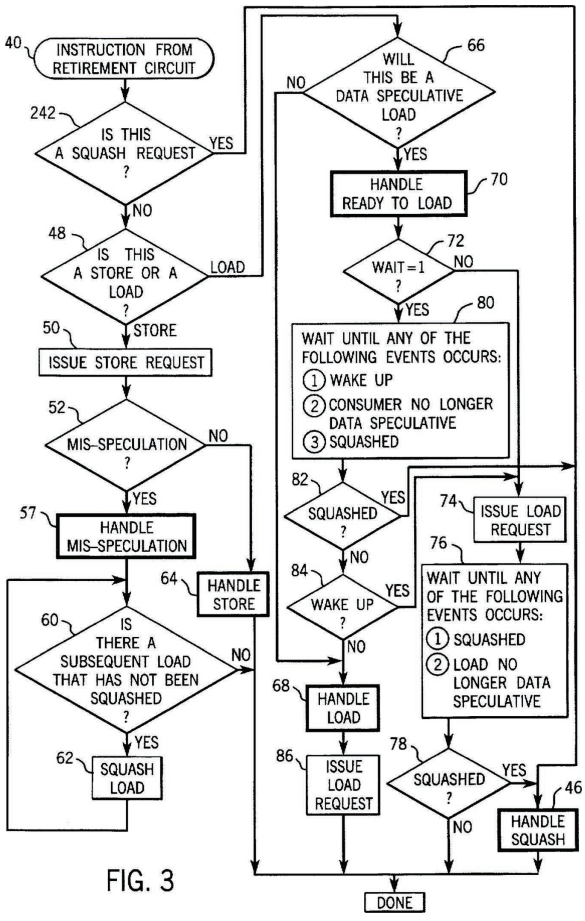

FIGURE 1. Figure 3 from U.S. Patent No. 5,781,752.

instruction" and "data producing instruction") cause a mis-speculation (see page 28 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel argued that the patent describes that "one LOAD/STORE pair causes a data mis-speculation" ('752 Patent at 3:55) and "the prediction table 44 is checked to see whether the LOAD/STORE pair causing the mis-speculation is in the prediction table 44 already" ('752 Patent at 12:65–67) (see page 29 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel argues that the word "instructions"(plural) in Limitation [b] of Claim 1 refers to the load/store pair that were described earlier in the claim (see page 29 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Based on all of this, Intel argued that because the claim language describes that a load instruction that is dependent on a store instruction and because the patent describes those loads and stores as a "load/store pair," "[a]lthough the word 'pair' does not appear in the claim itself, the plain reading of the claim is that it is directed to a load/store pair that has caused an error via mis-speculation" (see page 29 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

IN OTHER WORDS, IT IS UNNECESSARY TO PROVIDE A CONSTRUCTION FOR A TERM THAT IS WELL UNDERSTOOD IN THE FIELD AND/OR HAS AN OBVIOUS MEANING FROM ITS CONSTITUENT WORDS.

Intel also argued that the patent also supports including "load/store pair" in its proposed construction (see page 29 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). More specifically, Intel argued that the patent described that the "present invention" was to "identify the load/store pairs that mis-speculate, and prevent them from speculating" ('752 Patent at 3:51–62, 4:8–20, 14:15–21) (see page 29 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel essentially argued that load/store pairs are necessary because data speculation involves more than a load instruction; rather, "[d]ata speculation occurs when a particular instruction that uses data (a load) is executed out of order before another particular instruction that may set the value for that data (a store)" ('752 Patent at 1:67–2:4) (see page 30 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Based on that, Intel contended that "the claim must be construed to reflect the 'invention' as defined by the inventors in the specification—that is, a circuit that detects data dependence between a 'load/store pair'" (see page 30 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel further argued that the inventors did not invent or intend to claim a "processor that controlled data speculation by anything other than with load/store pairs" (see page 31 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel asserted that the patent describes that the inventors recognized that data mis-speculations were attributable to "a few static store/load instruction pairs" ('752 Patent at 3:51–55) and "the present invention" concerned mis-speculation by a "data producing/consuming instruction pair" ('752 Patent at 4:20) (see page 31 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel also argued that "the load/store pair mechanism is the only [example] disclosed in the specification" and that "[e]very figure and every description is directed toward identifying and disabling a mis-speculating load instruction that is paired with a particular store instruction" ('752 Patent at Figure 5, 11:8–14, 9:98–15; Figure 4, element 106; Figure 7, element 204; Figure 9, element 301, 1265–67; 14:3–6; 11:30–33; 14:19–21) (see page 31 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Based on that, Intel argued that the load/store pair is not just an example, but rather is a "necessary feature" of the patent (see page 32 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel next argued that there is an important difference between the phrase "load/store pair" and "load and store instructions" (see page 32 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). In particular, Intel asserted that the "purpose of the invention is not to detect mis-speculation by a load instruction per se, but rather to detect—and thereby ultimately prevent mis-speculation by a load instruction when it appears with a particular store instruction" (see page 32 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel asserted that because a speculative load may result in a mis-speculation only with some stores and not other stores, it is necessary to detect dependence between instruction pairs (see pages 32–33 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

Intel argued that the patent describes that if a load/store pair results in a mis-speculation, then a later instance of that load/store pair is highly like to result in another mis-speculation ('752 Patent at 3:51–57) (see page 31 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). Intel argued that the patent "discourages the attempt to generally identify mis-speculating loads: '[A]ny instruction loading data from memory can be data dependent on any previous instructions that writes [*sic*] to memory. Consequently, predicting and tracking ...

can easily become overwhelming" (5,781,752 Patent at 3:39–43) (see page 33 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

With respect to "detects a mis-speculation by a load instruction that is in fact executed" in its proposed construction, Intel argued that "in fact executed" is needed because the mis-speculation is caused when the load instruction in the pair executes before the store in the pair (see page 34 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]). By contrast, Intel asserted that WARF's omission of "in fact executed" is incorrect because the claim describes that the data speculation circuit detects a mis-speculation where a load is "in fact executed" (see page 34 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

With respect to the rest of WARF's proposed construction ("tracks execution of such instructions in order to detect any mis-speculations arising from the data speculative execution of LOAD instructions"), Intel conceded that there is not anything "affirmatively inaccurate," but still argued that the court should not adopt WARF's proposed construction because it makes the construction of this term longer, which might serve to confuse the jury when they try to apply this construction, and without any associated benefit (see page 34 of *Wis. Alumni Rsch. Found. v. Intel Corp.*[2]).

The next article in this series will continue to describe the parties' arguments in their responsive briefs for this term, the courts' analyses and reasoning, and my analysis.

REFERENCES

1. Wis. Alumni Res. Found. v. Intel Corp., No. 3:08-cv-00078, ECF No. 29, Jun. 26, 2008 (W.D. Wis.).
2. Wis. Alumni Res. Found. v. Intel Corp., No. 3:08-cv-00078, ECF No. 33, Jun. 27, 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.