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

DEPARTMENT: MICRO LAW

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

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

On 24 August 2024, the Court of Appeals for the Federal Circuit ("Federal Circuit") denied Plaintiff Wisconsin Alumni Research Foundation's ("WARF") appeals in two patent cases against Defendant Apple, Inc. In the first case, a jury initially awarded $234 million for Apple's infringement, which District Judge William M. Conley later increased to $506 million to account for postverdict supplemental damages, on-going royalties, costs, and prejudgment and postjudgment interest. Prior to suing Apple, WARF sued Intel Corporation for infringement of the same patent. Prior to trial, the parties settled for $110 million.[a]

This article is the first in a multipart series that reviews these three cases. These cases may be particularly interesting for the readership of *IEEE Micro* for several reasons. First, the asserted patent is a computer architecture patent, which is relatively uncommon. Second, the patent was issued to a university, which is very rare. The inventors, which are well known in computer architecture, were Andreas I. Moshovos, Scott E. Breach, Terani N. Vijaykumar, and Gurindar S. Sohi from the University of Wisconsin, and the assignee (the owner of the patent) is WARF. Third, the defendants in the three cases, Intel and Apple, are two of the world's biggest and best-known microprocessor companies. Fourth, in addition to the well-known inventors, other well-known computer architects worked as experts on this case, including David August, professor at Princeton University; Robert Colwell, Intel's former chief IA32 microprocessor architect; Thomas Conte, professor at Georgia Institute of Technology; Trevor Mudge, professor emeritus at University of Michigan; Murali Annavaram, a professor at the University of Southern California; W. Michael Johnson, author of the well-known book *Superscalar Microprocessor Design*; and Glenn Reinman, professor at University of California, Los Angeles.

Fifth, two of the best law firms for patent litigation, Irell & Manella and Wilmer Hale, led by two of the top patent litigators in the country, Morgan Chu and William F. Lee, represented the parties in the case. Sixth, due to the stakes, the quality of the lawyering, and the number of appeals, there were several interesting technical and legal issues. Seventh, due to the fact that the asserted patent is a computer architecture patent, for the readership of *IEEE Micro*, these cases are the perfect backdrop to explain how patent litigation process works and the common strategies that are used by plaintiffs and defendants.

Finally, the verdict in the *Apple* case was for a relatively large amount for a patent case, and a very large amount for a patent asserted by a university. Although patent lawsuits filed by universities are relatively rare, in the past 10–15 years, a few universities have filed a few patent lawsuits, which have resulted in very large jury verdicts. For example, in 2012, a federal jury in Pennsylvania found that Marvell Technology Group Ltd. infringed on two of Carnegie Mellon University's patents that are related to hard drive technology, and awarded the latter $1.17 billion in damages. The parties eventually settled for $750 million.[b] As a second example, in 2020, a federal jury in California found that Apple and Broadcom Ltd. infringed on three of California Institute of Technology's patents that are related to Wi-Fi data transmission, and awarded the latter $1.1 billion in damages. The Federal Circuit later overturned the award and ordered a new trial on damages. Before the new trial, the parties settled for an undisclosed amount.[c] As a third example, in 2022, a federal jury in Virginia found that Norton LifeLock Inc. and Symantec Corporation infringed on two of the patents owned by

[a]*Wisconsin Alumni Rsch. Found. v. Apple, Inc.*, 261 F.Supp.3d 900, 916 (2017)

[b]Press Release, Carnegie Mellon University, Carnegie Mellon University and Marvell Technology Group Ltd. Reach Settlement (Feb. 17, 2016), https://www.cmu.edu/news/stories/archives/2016/february/settlement.html.

[c]Caltech Settles Billion-Dollar Patent Lawsuits Against Apple and Broadcom, Pasadena News (Oct. 13, 2023), https://pasadenanow.com/main/caltech-settles-billion-dollar-patent-lawsuits-against-apple-and-broadcom

0272-1732 © 2024 IEEE
Digital Object Identifier 10.1109/MM.2024.3504881
Date of current version 11 December 2024.

The Trustees of Columbia University in the City of New York that are related to anti-malware, and awarded the latter $185 million. The district judge later increased the total amount awarded to Columbia to $577 million due to enhanced damages, supplemental damages, prejudgment interest, attorneys' fees, and costs.

U.S. PATENT NUMBER 5,781,752

The asserted patent in the WARF cases was U.S. Patent Number 5,781,752, which is titled "Table Based Data Speculation Circuit for Parallel Processing Computer." The applicant filed the patent application on 26 December 1996. In June 1997, the inventors published a paper in the International Symposium on Computer Architecture (ISCA) that appears to correspond to the '752 Patent. The title of this paper is "Dynamic Speculation and Synchronization of Data Dependences," and all four inventors of the '752 Patent were the coauthors of this paper. Because the three WARF cases were based on the '752 Patent and not the ISCA 1997 paper, this article and the remaining articles in this series are based on the patent. But to the extent that descriptions in the ISCA 1997 paper are helpful, this series will selectively cite that paper.

One interesting difference between a patent and its corresponding academic paper is the intended audience. Conference papers are typically written by, and for, professors or graduate students, or engineers who work in industry. These people typically have at least a master's degree, if not a Ph.D., in electrical engineering or computer science and perhaps decades worth of experience with research and designing microprocessors. By contrast, courts interpret what a patent means based on what a person of ordinary skill in the art ("POSITA") would understand the patent to disclose, at the time of the invention. For electrical engineering and/or computer science patents, a POSITA typically has either 1) a bachelor's degree and approximately two to three years of experience in that field or 2) a master's degree and approximately one to two years of experience. Therefore, given that papers in prestigious conferences such as ISCA are typically written for an audience with significantly more education and experience than a POSITA, there tends to be significantly less explanation of basic concepts but significantly more technical detail about the proposed idea in the paper.

Additionally, patents are generally written by an attorney who has a science, technology, or engineering background, but not necessarily in a major related to computer architecture. Even if the attorney has a background in computer architecture, he/she likely does not have an advanced degree and/or significant experience with researching and designing microprocessors. Accordingly, at least in this case, the patent provides a significant amount of background information regarding instruction-level processing, control and data dependences, speculation, mis-speculation recovery, and so on that would not be in a typical ISCA paper. Given that a POSITA would understand these concepts, it appears that the prosecuting attorney included a short description of these concepts to help a non-POSITA understand the patent.

Furthermore, the '752 Patent does not use certain terminology that is common in computer architecture, e.g., pipelining, reorder buffer, scheduling, load/store unit, and so forth, even though those terms would likely help a POSITA understand the claimed invention. Additionally, the '752 Patent uses what appears to be nonstandard terminology for other terms (e.g., *allocation circuit*, to apparently refer to issue). Finally, the '752 Patent appears to use some common terminology in nonconventional ways (e.g., the *retirement circuit* has more functionality than simply retiring/committing instructions but may also include load/store unit functionality as well as potentially other functionality). These word choices may be due to the fact that the prosecuting attorney does not have a computer architecture background.

Another difference between a patent and its corresponding academic paper is that academic papers tend to have a significant number of results that demonstrate the efficacy of the proposed idea across a range of different implementations, processors configurations, and benchmarks, whereas it is extremely rare to every find any results, simulation or otherwise, in a patent.

Finally, the fact that the invention disclosed in this patent was subsequently published in a prestigious conference indicates that experts in the field (i.e., the program committee of the conference and the peer reviewers) believe that the invention in the patent and in the paper contains a very novel idea (i.e., a technological breakthrough). This is strong evidence that the patent is valid (i.e., not anticipated by another patent/paper nor rendered obvious by combinations of patents/papers) as the paper would not have been accepted otherwise. Furthermore, this may also indicate that the invention in the patent and in the paper is significantly more valuable as only the very best ideas are accepted for publication in prestigious conferences.

At a very high level, the '752 Patent discloses a mechanism to predict whether a data dependence exists between two instructions, allowing execution if the prediction indicates that no dependence exists,

and a mechanism to recover from a misprediction. Although the abstract and claims do not expressly limit the claimed invention to loads and stores, but potentially covers any two instructions with a potential dependence, the detailed description of the invention is directed toward loads and stores. Furthermore, as a practical matter, the purported problem that the patent is trying to solve (dependences between instructions that are unknown until later in time) applies to only loads and stores as the actual memory locations that the loads access and the stores write to are unknown until they are computed.

Figure 3 (see Figure 1) of the '752 Patent appears on the cover page of the patent as the representative figure and depicts a flow chart of the operation of a "typical" implementation of the claimed invention. '752 Patent at 5:50–51.

Starting from the upper-left corner of Figure 3 (see Figure 1), the '752 Patent describes that the data speculation circuit receives an instruction and an indication that the instruction should be squashed or should execute (process block 40). If the indication is that the instruction should be squashed (decision block 242), then the data speculation circuit sends a HANDLE SQUASH signal to the predictor circuit (process block 46).

If the indication is that the instruction should execute (decision block 242), then the data speculation circuit checks whether the instruction is a load or a store (decision block 48). If the instruction is a store, then the data speculation circuit issues a store request, which "may, for example, authorize the retirement circuit 26 to perform the STORE operation for the data." *Id*. at 9:59–60.

The data speculation circuit then checks subsequent loads to see whether there was a mis-speculation, i.e., the subsequent loads read from the same memory location that an earlier, pending store was writing to (decision block 52). If so, then those loads need to be discarded. The data speculation circuit sends a HANDLE MIS-SPECULATION signal to the predictor circuit (process block 57). The '752 Patent describes that the predictor circuit uses that signal to adjust future predictions for this load/store pair. *Id*. at 9:66–7:1. The data speculation circuit then iteratively squashes all dependent loads (decision block 60 and process block 62). If the speculation was correct, then the data speculation circuit sends a HANDLE STORE signal to the predictor circuit (process block 60).

If the instruction is a load, then the data speculation circuit checks whether the load is speculative, i.e., whether there are earlier store instructions that the load instruction might have a dependence with (decision block 66). If not, then the data speculation circuit sends a HANDLE LOAD signal to the predictor circuit (process block 68) and the load instruction access memory (process block 86). If the load is speculative, then the data speculation circuit sends a HANDLE READY TO LOAD signal to the predictor circuit (process block 70). The predictor circuit responds by "making a prediction as to whether the LOAD should take place through the use of a wait flag." *Id*. at 10:18–21.

If the wait flag is one, then the data speculation circuit waits for one of three events: 1) wake-up, 2) load instruction is no longer speculative, and 3) load instruction is squashed. When the squash event occurs (decision block 82), the data speculation circuit handles the squash (process block 46). If the load instruction is no longer speculative ("NO" at decision block 84), then the data speculation circuit sends a HANDLE LOAD signal to the predictor circuit (process block 68), and the load instruction accesses memory (process block 86).

If the wake-up event occurs or if the wait flag is zero, then the speculative load accesses memory. Then

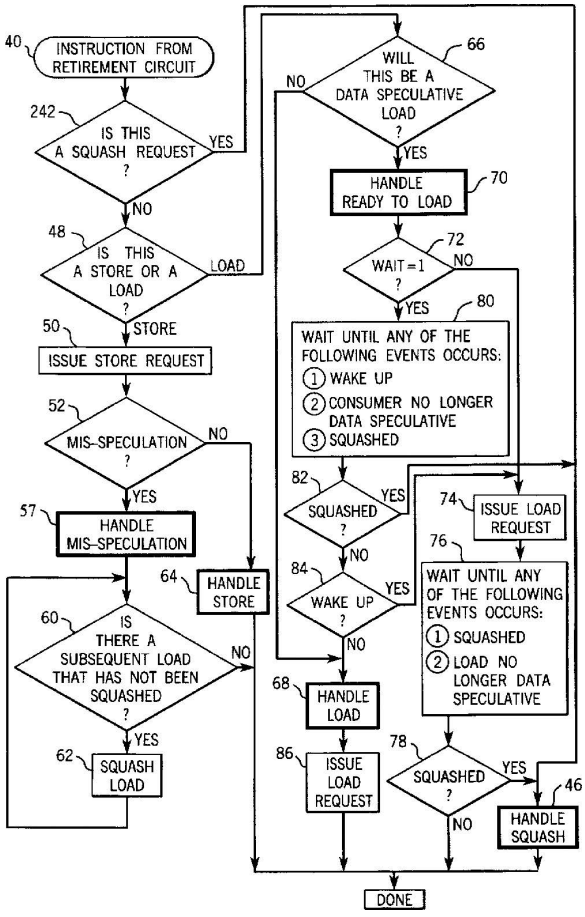

FIGURE 1. A copy of Figure 3 from the '752 Patent.

speculative load then waits for one of two events to occur: 1) load instruction is no longer speculative and 2) load instruction is squashed. If the first event occurs ("NO" at decision block 78), then the speculation that the load instruction did not depend on the prior store instruction was correct and the load instruction can commit. If the second event occurs ("YES" at decision block 78), then speculation that the load instruction did not depend on the prior store instruction was incorrect and then the load instruction needs squashed (process block 46).

The '752 Patent describes that the prediction table has at least three columns: the first identifies the load instruction, the second identifies the store instruction that the load instruction may depend on, and the third holds a prediction that indicates the likelihood of data dependence between the load and the store. *Id*. at 11:8–14, 11:27–30. Figure 5 (see Figure 2) depicts an example of the prediction table that contains static load with a logical address of 8[d] and static store with a logical address of 10 and prediction value of one.

The '752 Patent describes that the higher the value of the prediction, the greater the likelihood of mis-speculation if the load executes before the store. *Id*. at 14:3–6. The prediction "normally" starts at zero when an entry is first made in the prediction table. *Id*. at 11:33–35. The prediction is incremented when the speculation that there is no dependence between a load/store pair is incorrect. See, e.g., *id*. at 12:64–13:3. The prediction is decremented when the speculation is correct. See, e.g., *id*. at 12:14–17. The '752 Patent describes how the data speculation circuit uses the prediction:

> "A prediction threshold detector prevents data speculation for instructions that have a prediction value within a predetermined range. This prediction threshold detector may include an instruction-synchronizing circuit that instructs a processing unit to delay a later execution of the particular data consuming instruction until after the execution of the particular data producing instruction when the prediction associated with the data producing/consuming instruction pair is within a predetermined range."
>
> —*Id*. at 4:21–30.

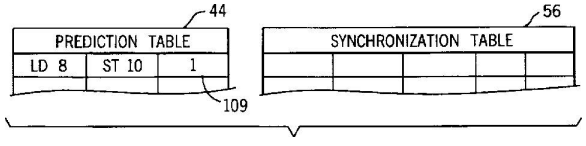

FIGURE 2. A copy of Figure 5 from the '752 Patent.

In other words, when the value of the prediction for a particular load/store pair is too high, the data speculation circuit prevents the load from speculatively executing, i.e., executing before the value for the corresponding store instruction is known or written to memory. This reduces mis-speculation costs, which concomitantly increases processor performance.

The '752 Patent has a total of nine claims: two independent and seven dependent, all of which depend on claim 1. All claims are apparatus claims, i.e., they claim a system. Claim 1 recites

> › 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 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
> - A prediction threshold detector preventing data speculation for instructions having a prediction within a predetermined range.

Claim 9 recites

> › In a processor capable of executing program instructions in an execution order differing from the program order of the instructions, the processor further having a data speculation circuit for detecting data dependence between instructions and detecting a mis-speculation where a

[d]The '752 Patent states that the "particular load instruction 8.2 identified by its physical address" is in the prediction table depicted in Figure 5 of the '752 Patent at 11:5–7. But Figure 2 of the '752 Patent depicts that the load instruction has a logical address of 8. (Figure 2 of the '752 Patent also depicts that each dynamic instance of the static load is given a decimal value, e.g., 8.1, 8.2, and 8.3). Therefore, based on the depictions in Figures 2 and 5 of the '752 Patent, it appears that the patent may have inadvertently used "physical address" here when it meant "logical address."

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 prediction table communicating with the data speculation circuit to create an entry listing a particular data consuming instruction and data producing instruction each associated with a prediction when a mis-speculation indication is received.
- An instruction synchronization circuit only instructing a processor to delay a later execution of the particular data consuming instruction if the prediction table includes an entry.

Compared to other patents, the number of claims is relatively few. Patents typically have approximately three independent claims and approximately 20 total claims. Furthermore, patents typically generally have both apparatus and method claims, the latter of which may be easier to recover damages incurred prior to the filing of a lawsuit. As such, it is very unusual to have so few claims or to have no method claims.

Likewise, the specification of the '752 Patent is relatively short, comprising only 14 total columns, and where the description of the claimed invention comprises only six columns. Although there is nothing wrong per se with a shorter specification, a shorter specification, by definition, contains less information, which may make it more difficult for the patent owner, i.e., WARF, in this case, to make specific arguments as there may be little or no support in the short specification for those arguments.

The next article in this series will begin to review the lawsuits where this patent was asserted against Intel and Apple.

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.