NOTE: This disposition is nonprecedential.

United States Court of Appeals for the Federal Circuit

2007-1190

NAZOMI COMMUNICATIONS, INC.,

Plaintiff-Appellant,

v.

ARM HOLDINGS, PLC,
ARM LIMITED, and ARM, INC.,

Defendants-Appellees.

Thomas J. Friel, Jr., Cooley Godward Kronish LLP, of San Francisco, California, argued for plaintiff-appellant. With him on the brief were Lori R.E. Ploeger, Timothy S. Teter, and Jeffrey S. Karr, of Palo Alto, California.

Andrew Y. Piatnicia, Howrey LLP, of East Palo Alto, California, argued for defendants-appellees. With him on the brief were Robert P. Taylor, of East Palo Alto, California, and Ethan B. Andelman, of San Francisco, California.

Appealed from: United States District Court for the Northern District of California

Judge Jeremy Fogel

NOTE: This disposition is nonprecedential.

United States Court of Appeals for the Federal Circuit

2007-1190

NAZOMI COMMUNICATIONS, INC.,

Plaintiff-Appellant,

v.

ARM HOLDINGS, PLC,
ARM LIMITED, and ARM, INC.,

Defendants-Appellees.

Appeal from the United States District Court for the Northern District of California in case no. 02-CV-2521, Judge Jeremy Fogel.

DECIDED: February 21, 2008

Before RADER, SCHALL, and PROST, Circuit Judges.

SCHALL, Circuit Judge.

DECISION

Nazomi Communications, Inc. ("Nazomi") appeals the January 30, 2007 decision of the United States District Court for the Northern District of California granting summary judgment of non-infringement in favor of ARM Holdings, PLC, ARM Limited, and ARM, Inc. (collectively "ARM") in Nazomi's suit against ARM for infringement of U.S. Patent No. 6,332,215 (the "'215 patent"). Nazomi Commc'ns, Inc. v. ARM

Holdings, PLC, No. C02-2521JF (N.D. Cal. Jan. 30, 2007) ("Summary Judgment Decision").

On September 6, 2006, the district court issued an order construing the single claim term "instructions." Nazomi Commc'ns, Inc. v. ARM Holdings, PLC, No. C02-2521JF (N.D. Cal. Sept. 6, 2006) ("Claim Construction Order"). Based on that construction, Nazomi conceded that ARM's products do not infringe the '215 patent, and the district court granted ARM's motion for summary judgment of non-infringement. Summary Judgment Decision. Because we see no error in the district court's claim construction, we affirm.

DISCUSSION

I.

The invention claimed in the '215 patent is a hardware accelerator that converts stack-based Java bytecode instructions into register-based "native" instructions that are executable by a register-based central processing unit (CPU). E.g., '215 patent col.2 ll.1–10. The hardware accelerator of the claimed invention is said to significantly speed up the processing of Java bytecodes over prior art systems that used software to perform the conversion from stack-based instructions to register-based instructions. Id. col.2 ll.6–10. Claim 1 is representative of the claims of the '215 patent:

> A system comprising:
>
> a central processing unit having a register file, the central processing unit adapted to execute register-based instructions; and
>
> a hardware unit associated with the central processing unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, wherein a portion of the operand stack is stored in the register file of the central processing unit and wherein the hardware unit is adapted to produce at least one of overflow or underflow indications for the portion of the operand stack stored in the register file, wherein the

> hardware unit is adapted to swap parts of the operand stack in and out of the register file from a memory, the system including an indication of the depth of the portion of operand stack, wherein a overflow or underflow produces an operand transfer between the register file in the central processing unit and memory.

(emphases added). The district court construed the term "instructions" as "either a stack-based instruction that is to be translated into a register-based instruction, or a register-based instructions [sic] that is input to the CPU pipeline." Claim Construction Order at 11. In either case, the court found that the instruction "must be upstream of the decode stage of the CPU pipeline" and "cannot mean the control signals that are the output of the decode stage." Id.

Based on the district court's construction of the term "instructions," ARM moved for summary judgment that its "Jazelle" Revision 3 (and higher) processors do not infringe the '215 patent. ARM argued that summary judgment of non-infringement was proper because it was undisputed that ARM's processors do not translate stack-based instructions into register-based instructions upstream of the decode stage. Rather, ARM's processors merely convert stack-based instructions into "control signals" and, thus, under the district court's claim construction do not meet the limitation present in every claim of the '215 patent requiring a conversion of "stack-based instructions into register-based instructions."

Responding, Nazomi conceded that ARM's "Jazelle" Revision 3 (and higher) processors do not infringe the '215 patent under the district court's claim construction, either literally or under the doctrine of equivalents. Accordingly, the district court granted ARM's motion for summary judgment of non-infringement. Summary Judgment Decision. We have jurisdiction over Nazomi's appeal pursuant to 28 U.S.C. § 1295(a)(1).

II.

On appeal, Nazomi argues that the district court erred in construing the term "instructions" to exclude the "control signals" that are the output of the CPU's decode stage. First, Nazomi argues that the claims themselves are broad, and do not require the instruction translation to be performed at any particular location in relation to the CPU. Rather, Nazomi points out, the patent describes and claims both a co-processor embodiment, wherein the "hardware unit" is located outside of the CPU, '215 patent claim 3, and an integrated microprocessor embodiment, wherein the "hardware unit" is located within the CPU, id. claim 2. With respect to the integrated microprocessor embodiment, Nazomi contends that the plain language of the claims is broad enough to encompass a system that translates instructions not only prior to the decode stage of the CPU's pipeline but also as part of the decode stage.

Indeed, Nazomi argues that dependent claims 30 and 32 specifically contemplate the existence of "instructions" after the decode stage of the CPU's pipeline. Referring to the integrated microprocessor embodiment, claim 30 recites "[t]he system of claim 2, wherein the central processing unit includes an execution unit to execute the register-based instructions." According to Nazomi, the execution unit would never "execute" register-based "instructions" as required by claim 30 if—as the district court found—"instructions" do not exist at the execution stage of the CPU's pipeline. Nazomi makes essentially the same argument with regard to claim 32, which recites "[t]he system of claim 1, wherein register-based instructions cause the manipulation of the register file." Referring to figure 3 of the patent, Nazomi notes that the register file element (46) is connected to the CPU's "execute logic" stage (26c), which is located downstream of the

"instruction decode" stage (26b) of the CPU's pipeline. Nazomi contends that, under the district court's construction, register-based instructions would therefore never "cause" manipulation of the register file as required by claim 32. For these reasons, Nazomi argues that the language of claims 30 and 32 requires a claim construction that accommodates the existence of "instructions" after the decode stage of the CPU's pipeline. According to Nazomi, the district court erred by failing to account for the inconsistencies between its construction of the term "instructions" and the language of claims 30 and 32, and by instead simply attributing the inconsistencies to imprecise claim drafting.

In addition, Nazomi contends that the written description of the '215 patent contradicts the district court's claim construction. For example, the patent states that "[t]he execute logic 26c executes the native instructions." Id. col.5 ll.14–15. Also, discussing the integrated microprocessor embodiment, the patent states that "the central processing unit has a Java hardware accelerator subunit to translate Java bytecode into the native instructions operated on by the main portion of the CPU." Id. col.3 ll.35–41 (emphasis added). Nazomi argues that a person of ordinary skill in the art would understand "main portion of the CPU" to refer to the CPU's execute stage, and that the patent thus instructs that the CPU's execute logic "operates on" native instructions. According to Nazomi, these provisions of the written description—consistent with the language of claims 30 and 32—demonstrate that "instructions" may exist after the CPU's decode stage.

Nazomi further contends that the district court's construction improperly limits the claims to the preferred embodiment, shown in figure 3 of the patent, wherein the

translation of instructions by the hardware unit is performed prior to the CPU's decode stage. For the reasons discussed above, Nazomi argues that the patent also discloses and claims other embodiments of the invention, wherein the hardware unit is positioned differently vis-à-vis the CPU's decode stage. However, even if figure 3 represented the only embodiment disclosed in the specification, Nazomi argues, it would be improper to limit otherwise broad claim language to that single embodiment.[1]

Finally, Nazomi asserts that the district court's construction conflicts with the doctrine of claim differentiation (i.e., that claim 30 is presumed to have different scope than claim 1 from which it depends), and departs from broader uses of the term "instructions" in certain prior art patents cited by the examiner during prosecution and technical treatises.

III.

Claim construction is a question of law that we review de novo. Cybor Corp. v. FAS Techs., Inc., 138 F.3d 1448, 1454 (Fed. Cir. 1998) (en banc). In determining the meaning of a disputed claim limitation, we look primarily to the intrinsic evidence of record, including the claim language, written description, and prosecution history. Phillips v. AWH Corp., 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc). Words of a claim "are generally given their ordinary and customary meaning" as understood by a person of ordinary skill in the art. Id. at 1312–13. Claims are read in view of the

[1] In support of this argument, Nazomi cites Liebel-Flarsheim Co. v. Medrad, Inc., 358 F.3d 898, 906 (Fed. Cir. 2004) ("[T]his court has expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment. . . . Even when the specification describes only a single embodiment, the claims of the patent will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using 'words or expressions of manifest exclusion or restriction.'" (internal citations omitted)).

specification, which is the "single best guide to the meaning of a disputed term." Id. at 1315. A court "should also consider the patent's prosecution history, if it is in evidence." Id. at 1317. Finally, although it is generally less significant than the intrinsic record, extrinsic evidence can "shed useful light on the relevant art." Id. Applying this framework, we conclude that the district court properly construed the term "instructions."

We first note the specific context in which the disputed term "instructions" is used in claim 1. Claim 1 of the '215 patent requires a hardware unit "adapted to convert stack-based instructions into register-based instructions." The parties dispute whether the group of signals that forms the output of the CPU's decode stage (and, correspondingly, the input of the CPU's execute stage) may be deemed "instructions." We think that it is more appropriate for purposes of this appeal, however, to focus on the somewhat narrower question of whether those signals constitute "register-based instructions." The modifier "register-based" appears in the patent's written description only three times. Each time, it is used in conjunction with the term "native" to describe a specific type of instruction. For example, in the Summary of the Invention, the patent states:

> The hardware Java accelerator can convert the stack-based Java bytecodes into a [sic] register-based native instructions on a CPU. The hardware accelerators of the present invention are not limited for use with Java language and can be used with any stack-based language that is to be converted to register-based native instructions. Also, the present invention can be used with any language that uses instructions, such as bytecodes, which run on a virtual machine.

'215 patent col.2 ll.19–27 (emphases added). The patent generally provides for the conversion of stack-based instructions (e.g., Java bytecodes) into register-based native instructions. In most instances, the written description refers to the conversion of stack-based instructions into native instructions. See, e.g., id. col.2 ll.3–6 ("The present

invention generally relates to a Java hardware accelerator which can be used to quickly translate Java bytecodes into native instructions for a central processing unit . . . ."), col.3 ll.6–11 ("The Java hardware accelerator can do, some or all of the following tasks: . . . translating bytecodes to native instructions . . . ."). The claims only describe the conversion of stack-based-instructions into register-based instructions. Ultimately, both terms ("native instructions" in the written description and "register-based instructions" in the claims) are used as shorthand to describe the same instructions—i.e., register-based instructions that are native to a particular register-based processor. Compare id. claim 30 (claiming the inclusion of "an execution unit to execute the register-based instructions" (emphasis added)), with id. col.5 ll.14–17 (stating that "[t]he execute logic 26c executes the native instructions" (emphasis added)).

Continuing, we find the context in which the term "native instructions" is used in the written description particularly informative. In the Background of the Invention section, the patent states that "[i]n conventional programming languages, the source code of a program is sent to a compiler which translates the program into machine code or processor instructions. The processor instructions are native to the system's processor." Id. col.1 ll.13–17 (emphasis added). Referring to prior art systems that used software to translate instructions, the patent states: "To execute a Java program, a [software] bytecode interpreter takes the Java bytecode[s and] converts them to equivalent native processor instructions . . . ." Id. col.1 ll.28–30 (emphasis added); see also id. col.2 ll.6–10 ("The hardware accelerator [of the present invention] speeds up the processing of the Java bytecodes significantly because it removes the bottleneck which previously occurred when the Java Virtual Machine is run in software on the CPU to

translate Java bytecodes into native instructions." (emphasis added)). Moreover, when describing figures 5 and 6 of the patent, the written description states that "[t]he Java translating machine translates the Java bytecode into a native instruction such as the instruction ADD R1, R2. This is an instruction native to the CPU indicating the adding of value in register R1 to the value in register R2 and the storing of this result in register R2." Id. col.5 ll.55–60 (emphases added). Thus, in all cases, the '215 patent clearly and consistently uses the term "native instructions" to refer to instructions on the machine (or assembly) code level of abstraction—i.e., instructions of the type created by a native-language compiler, stored in memory, fetched upon execution, and input to the CPU's decode stage.

In contrast, the patent never describes the group of signals at the output of the CPU's decode stage as "instructions" of any type. Indeed, the prosecution history expressly contradicts the notion that the inventor intended the term "native instructions" to encompass those signals. In distinguishing over a specialized Java processor cited as prior art by the examiner, the inventor stated:

> [T]he first approach of Krall uses a specialized Java processor, such as the PicoJava microprocessor developed by Sun, which does not require any such hardware subunit. Specialized Java processors do not translate Java bytecodes into instructions native to a processor. For specialized Java processors, the Java bytecodes are the native instructions, and no translation is required.

Nazomi does not dispute that Sun's picoJava processors do, in fact, decode Java bytecodes into lower-level signals.[2] Thus, the statement that specialized Java

[2] Nazomi's expert admitted in a deposition that Sun's picoJava processors decode Java bytecodes into lower-level signals (which he referred to as "low-level instructions"). Nazomi apparently does not dispute this characterization of the picoJava processors.

processors "do not translate Java bytecodes into instructions native to a processor" demonstrates that the inventor did not use the term "native instructions" to encompass the low-level signals at the output of the processor's decode stage.

In sum, the specification and prosecution history clearly indicate that the terms "native instructions" and "register-based instructions" refer only to processor instructions prior to decoding, and do not encompass the low-level signals generated at the output of the processor's decode stage. Moreover, those terms do not merely refer to a preferred embodiment of the invention, as they are consistently used in describing the prior art, background of the invention, and invention itself.

Contrary to Nazomi's arguments, we do not think that this interpretation of "instructions" is inconsistent with the patent's claims and written description. Like the district court, we conclude that the words "execute," id. claim 30, col.5 ll.14–15, and "cause," id. claim 32, refer only to indirect causation. The patent uses the term "execute" in an indirect manner elsewhere in describing the prior art conversion of Java bytecodes using software: "To execute a Java program, a bytecode interpreter takes the Java bytecode converts them to equivalent native processor instructions and executes the Java program." Id. col.1 ll.28–32 (emphasis added). The bytecode interpreter is a software component and, as such, does not directly execute Java programs; rather, it indirectly causes the execution of Java programs.

Nor does the patent's teaching that "native instructions [are] operated on by the main portion of the CPU" compel a different result. Id. col.3 ll.35–41. Nazomi has provided no support for its assertion that one of ordinary skill in the art would understand "main portion of the CPU" to refer in particular to the CPU's execute stage.

Rather, the expert declaration to which Nazomi cites merely states that "[o]ne skilled in the computer art would understand 'main portion of the CPU' to include an Execute Logic . . . ." (emphasis added). The patent uses the phrase "main portion of the CPU" when explaining that the hardware accelerator can be incorporated into the CPU as a subunit. Id. col.3 ll.35–41. Understood in that context, we think that "main portion of the CPU" most likely refers to the components of the integrated CPU other than the hardware accelerator. In any event, it does not necessarily establish, as Nazomi argues, that "native instructions" exist at the execution stage of the CPU's pipeline.

Nazomi's argument with respect to the doctrine of claim differentiation is also inapposite. Dependent claim 30 adds the requirement that the CPU include "an execution unit to execute the register-based instructions" (emphases added). Nazomi contends that "[t]he only difference between claim 30 and the integrated microprocessor embodiment of claim 1 is that claim 30 limits its scope to the execution of register-based instructions by an execution unit, which occurs after the decode stage." However, even if we agreed with Nazomi's interpretations of "execute" (as referring to direct execution) and "instructions" (as encompassing the low-level signals output from the CPU's decode stage), this argument would still lack merit. Nazomi apparently is arguing that the scope of claim 30 is limited to an embodiment wherein instructions are translated as part of the decode stage, whereas claim 1 also encompasses the embodiment wherein instructions are translated prior to the decode stage. Under the construction advanced by Nazomi, however, "register-based instructions" would exist at the execution stage of the CPU's pipeline in both embodiments, and, accordingly, claim 30 would embrace both embodiments. The doctrine of claim differentiation thus does not motivate us to prefer a

construction of “register-based instructions” that encompasses instruction translation as part of the CPU’s decode stage.

Finally, given the clear, specific meaning attributed to “native instructions” and “register-based instructions” in the ’215 patent, we do not find it particularly helpful to consult the technical treatises and prior art patents on which Nazomi relies. At best, those references establish that some in the art refer generally to the group of signals generated by the CPU’s decode stage as a type of “instructions” (e.g., “low-level instructions” or “microinstructions”). As discussed above, however, we conclude that one of ordinary skill in the art, reading the claims of the ’215 patent in light of the written description and prosecution history, would attribute a meaning to the claim term “register-based instructions” that does not encompass those signals.

For the foregoing reasons, see no error in the district court’s construction of the term “instructions.” Because Nazomi has conceded that ARM’s products do not infringe under that construction, we affirm the district court’s grant of summary judgment of non-infringement in favor of ARM.