UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

NAZOMI COMMUNICATIONS, INC., Plaintiff, v. NOKIA CORPORATION et al., Defendants.	Case No.: 10-CV-04686-RMW Case No:: 10-CV-05545 RMW ORDER CONSTRUING DISPUTED CLAIM LANGUAGE; GRANTING MOTION FOR SUMMARY JUDGEMENT OF NON-INFRINGEMENT
NAZOMI COMMUNICATIONS, INC., Plaintiff, v. SAMSUNG TELECOMMUNICATIONS AMERICA, L.L.C. et al., Defendants.	

Plaintiff Nazomi Communications, Inc. ("Nazomi") brings this patent infringement action against various technology companies (collectively "Defendants"), alleging infringement of claims 48 and 74 of U.S. Patent No. 7,080,362 (the "'362 patent") and claims 1 and 5 of U.S. Patent No. 7,225,436 (the "'436 patent"). The parties now move for a "preliminary" construction of all four asserted claims. Defendants Western Digital Corporation, Western Digital Technology

(collectively "WD") and Sling Media ("Sling") (collectively the "Moving Defendants") also move for summary judgment that their accused products, the WD "MyBook World Edition" and the Sling "Slingbox Pro-HD," do not infringe Nazomi's patents. After consideration of the claims, specifications, prosecution history and argument of the parties, the court construes the disputed claim language as described below. In addition, the court grants the moving defendants' motion for summary judgment of non-infringement.

I. BACKGROUND

A. The Patented Invention

This case concerns technology for processing different types of code, or "instruction sets," within the hardware of a computer: (1) "stack-based" instructions, which include those written in the popular programming language Java, and (2) "register-based" instructions. A "stack" and a "register" are different ways of storing data in memory. Therefore, Java "bytecodes" and other stack-based instructions will not operate on a register-based system unless they are first "translated" into register-based instructions.

Many register-based systems use software to create a Java Virtual Machine ("JVM") that performs the necessary translation. This approach provides flexibility, but slows overall execution speed. The patented invention "removes the bottleneck which previously occurred when the Java Virtual Machine is run in software … [by implementing] at least part of the Java Virtual Machine in hardware as the hardware Java accelerator." '362 Patent at 2:10-17. While the patents include both method and apparatus claims, the instant dispute focuses solely on claims directed to a hardware apparatus designed to process Java bytecodes and register-based instructions.

B. The Accused Products

Defendant-intervenor ARM develops and licenses processor core "designs." Other companies use ARM designs to build complete processors. Consumer product manufacturers, including the moving defendants, then incorporate those processors into their products.

In 2000, ARM developed a design geared towards accelerating the processing of Java bytecodes. The hardware component of the design is called "Jazelle." ARM included Jazelle

circuitry in the ARM 926EJ-S processor core (the "ARM core"), which was in turn licensed by chip makers, including non-parties Texas Instruments and PLX. It is the Jazelle hardware within the ARM core that Nazomi asserts infringes its patents.

Importantly, while Jazelle is "physically" present in every ARM core, the circuitry is "dormant" in its default state. Ex. A-6 (ARM 30(b)(6) Depo.) at 73. As defendants explained at oral argument, because processors are designed to be used in a wide variety of products, they often contain optional functionality not utilized by every manufacturer. To reduce power consumption, ARM designs its cores so that any unused functionality does not unnecessarily drain resources. *See* Ex. A-6 at 164. As a result, unless Jazelle is "enabled," it is "not receiving Java bytecodes into its decoders. There is no activity in the logic. It is not doing anything …" *Id.*

In order to activate Jazelle, a manufacturer must license from ARM a software package called the Java Technology Enabling Kit ("JTEK"). Ex. A-3 (Steel Decl.) ¶ 16. ARM describes JTEK as performing a series of "complicated steps" that are kept "secret" from the public. *Id.* When a manufacturer installs JTEK onto a device, the software is incorporated into a "JTEK-enabled JVM" that can use Jazelle to process Java bytecodes. There is no evidence that anyone has ever utilized Jazelle functionality without first taking a JTEK license from ARM. Ex. A-6 at 168.

1. The MyBook World Edition

The MyBook World Edition (the "MyBook") is an external hard drive that can be connected to a computer to store information. The MyBook contains a PLX processor, which incorporates an ARM core. Early in the MyBook's development, WD found that a Jazelle-based JVM was "not functional" because it would "lock up" and "was not stable enough" to reliably process Java bytecodes on the device. Ex. A-5 (WD 30(b)(6) Depo.) at 56-57. WD therefore incorporated a software-based JVM and Java accelerator called Just-in-Time ("JIT") that does not utilize Jazelle. *See id.* at 39. WD never included a Jazelle-based JVM in any MyBook shipped to consumers. *See id.* at 62.

The MyBook JVM is located on a "hidden" portion of the product's disk drive, which consumers are prohibited from modifying under WD's warranty. *See id.* at 100. While WD could

update the MyBook's software to include a JVM that utilizes Jazelle, it has never done so. *See id.* at 83. The software update mechanism was not designed to allow end-users to put additional software on the system, and there is no evidence that any end-user has installed JTEK onto a MyBook or otherwise made use of the product's Jazelle circuitry. *See id.* at 83-85.

2. The Slingbox Pro-HD

The Slingbox Pro-HD (the "SlingBox") is a device used to deliver video content over the internet to a laptop, tablet or smartphone. The Slingbox uses a Texas Instruments Da Vinci processor, which includes an ARM core. The Slingbox does not run Java-based applications or contain software that enables a consumer to download Java-based applications. *See* Ex. D (Ansis Decl.) ¶ 3. Sling thus did not license JTEK from ARM. *See* Ex. E (Shah Decl.) ¶ 5. In addition, the Slingbox does not include an interface to enable manipulation of the product's existing software. *See* Ansis Decl. ¶ 4. According to Sling, in order to install a Java-based application on the Slingbox, a consumer would have to "hack" the device and manually add the program. *Id.* There is no evidence that any consumer has ever taken such an action, which would "likely render the product useless and unable to function as designed." *Id.*

II. ANALYSIS

A. Claim Construction

As noted above, Nazomi asserts infringement of four distinct apparatus claims, each of which includes numerous terms. However, the present motions seek a "preliminary" construction of the four claims as a whole, rather than any particular claim term. The moving defendants argue that the asserted claims "require that the processor of the claimed apparatus, when operated, will perform the recited functions related to the processing of stack-based instructions without modification." Dkt. No. 373 at 2. Nazomi, on the other hand, seeks a construction requiring "only that the claimed apparatus be capable of performing the recited functions." Dkt. No. 380 at 24. Nazomi implies that the asserted claims cover any product that *could* process stack-based instructions in hardware, regardless of whether the device ever actually does so.

Claim construction analysis begins with the words of the claim. *See Nystrom v. Trex Co., Inc.*, 424 F.3d 1136, 1142 (Fed. Cir. 2005). A particular claim term is generally given "the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005). In determining how one of ordinary skill in the art would define a claim term, the court looks to "the words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art.'" *Id.* at 1314 (quoting *Innova /Pure Water, Inc. v. Safari Water Filtration Systems, Inc.*, 381 F.3d 1111, 1116 (Fed. Cir. 2004)).

Claim 48 of the '362 Patent, which is similar in most respects to the other asserted claims, reads:

> 48. A central processing unit (CPU) capable of executing a plurality of instruction sets comprising:
>
> An execution unit and associated register file, the execution unit to execute instructions of a plurality of instruction sets, including a stack-based and a register-based instruction set; a mechanism to maintain at least some data for the plurality of instruction sets in the register file including maintaining an operand stack for the stack-based instructions in the register file and an indication of a depth of the operand stack; a stack control mechanism that includes at least one of an overflow and underflow mechanism, wherein at least some of the operands are moved between the register file and memory; and a mechanism to generate an exception in respect of selected stack-based instructions.

First, the court notes that each of the structural elements of Claim 48 is defined by its functionality. More specifically, Claim 48 describes its structural limitations by explicitly referencing actions used to process stack-based instructions. For example, the claim does not recite simply an "execution unit," but rather "an execution unit to execute … a stack-based … instruction set." Likewise, the claim does not describe a generic "mechanism," but instead "a mechanism to maintain … an operand stack for the stack-based instructions;" "a stack control mechanism … wherein at least some of the operands are moved…;" and "a mechanism to generate an exception in respect of selected stack-based instructions." The other asserted claims define structural elements in a similar manner. *See* Claim 74, '362 Patent ("A [CPU] comprising: a decoding mechanism to

decode … a stack-based instruction set … an execution unit that processes … the stack-based instruction set …"); Claim 1, '436 Patent ("A CPU for executing stack and register-based instructions, comprising: … a hardware accelerator to process stack-based instructions …"); Claim 5, '436 Patent (A [CPU] comprising: … a hardware accelerator to process stack-based instructions …"). By defendants' count, of the 678 words in the asserted claims, at least 478 relate to processing stack-based instructions.

The patents' specifications also consistently describe the invention as a hardware apparatus used to process Java bytecodes. *See, e.g.*, '362 Patent at 4:8-9 ("The instructions translations unit is used to convert Java bytecodes to native instructions."); *id.* at 5:28-29 ("The instruction buffer stores the bytecodes from the instruction cache."); *id.* at 5:64-65 ("The Java translating machine translates the Java bytecode into a native instruction …"); '436 Patent at 2:6-7 ("The present invention generally relates to Java hardware accelerators used to translate Java bytecodes into native instructions …"); *id.* at 2:43-37 ("[A]nother embodiment of the present invention comprises a hardware accelerator operable connected to a central processing unit, the hardware accelerator adapted to convert stack-based instructions into register-based instructions …"). The specifications further emphasize that the primary advantage of the patented invention over the prior art is that the processing of stack-based instructions in hardware, rather than software, results in "an improved system for implementing *Java programs*." '362 Patent at 2:1-2; '436 Patent at 2:1-2 (emphasis added). Indeed, even the patents' titles, "Java Virtual Machine Hardware For RISC and CISC Processors" and "Java Hardware Accelerator Using Microcode Engine," demonstrate the centrality of processing Java bytecodes to the claimed invention.

By defining structural elements in functional terms and highlighting the implementation of Java-based programming, Nazomi plainly added limitations that would not be present had it used generic, structural language. *See K-2 Corp. v. Salomon S.A.*, 191 F.3d 1356, 1363 (Fed. Cir. 1999) ("The functional language is, of course, an additional limitation in the claim.") (citing *Wright Med. Tech., Inc. v. Osteonics Corp.*, 122 F.3d 1440, 1443-44 (Fed. Cir. 1997)). On the other hand, limiting the claims, as defendants suggest, to "processors that actually perform the recited functions

when placed in operation," would impermissibly convert Nazomi's apparatus claims into mixed method-apparatus claims. *See IPXL Holdings, L.L.C. v. Amazon.com, Inc.*, 430 F.3d 1377 (Fed. Cir. 2005) ("[R]eciting both an apparatus and a method of using that apparatus renders a claim indefinite under section 112, paragraph 2."); *Hewlett-Packard Co. v. Bausch & Lomb*, 909 F.2d 1464, 1468 (Fed. Cir. 1990) ("[A]pparatus claims cover what a device *is*, not what a device *does*.") (emphasis in original). Courts have typically resolved this issue by construing functional language to describe an apparatus' capabilities, rather than its performance. *See, e.g., Yodlee, Inc. v. Cashedge, Inc.*, No. 05-01550 SI, 2006 U.S. Dist. LEXIS 86699, at *14 (N.D. Cal. Nov. 29, 2006) ("All of the claims challenged by defendant … place functional limitations on the apparatuses by describing the capabilities of the apparatuses."); *Apple, Inc. v. Samsung Elecs. Co.*, No. 11-01846 LHK, 2012 U.S. Dist. LEXIS 90943, at *32-33 (N.D. Cal. June 29, 2012) ("The disputed claims here … use functional language to describe the capabilities of the claimed apparatus …."). Further, at least as to Claim 48, any construction requiring actual processing would ignore the term "capable of" found in the claim's preamble. The court therefore agrees with Nazomi that the functional language of the asserted claims recites capabilities of the claimed structures.

"Capability," however, is not a blank check. As the Federal Circuit has made clear, the scope of an apparatus claim phrased in functional terms is generally limited to products "configured" to perform the recited functions without "modification" by an end-user. *See Typhoon Touch Techs., Inc. v. Dell, Inc.*, 659 F.3d 1376, 1381 (Fed. Cir. 2011) (upholding a construction of the term "memory for storing [a data collection application]" to mean "memory [that] is actually programmed or configured to store the data collection application"); *Silicon Graphics, Inc. v. ATI Techs., Inc*., 607 F.3d 784, 794 (Fed. Cir. 2010) ("[A]n apparatus claim directed to a computer that is claimed in functional terms is nonetheless infringed so long as the product is designed 'in such a way as to enable a user of that [product] to utilize the function . . . without having to modify [the product].'") (quoting *Fantasy Sports Props., Inc. v. Sportsline.com, Inc.*, 287 F.3d 1108, 1117-18 (Fed. Cir. 2002)). The emphasis on an accused product's default configuration is consistent with the language of the patents-in-suit, which describes an invention designed primarily for a single

purpose—processing Java bytecodes—rather than operation in multiple modes or customization by end-users. *Compare Intel Corp. v. United States Int'l Trade Comm'n*, 946 F.2d 821, 831 (Fed. Cir. 1991) (finding infringement of a patent disclosing "memory with a programm*able* selection means for selecting *alternative* addressing modes" despite the fact that the defendant did not intend the accused product to operate in an infringing mode) (emphasis added). Accordingly, the court construes the asserted claims to cover a hardware apparatus configured to process Java bytecodes without modification by an end-user.

B. Summary Judgment of Non-Infringement

In order to establish a prima facie case of direct infringement, Nazomi must show that the moving defendants make, use, sell, offer to sell or import a product that infringes at least one asserted claim. *See* 35 U.S.C. § 271(a).[1] Where a defendant seeks summary judgment of non-infringement, "nothing more is required than the filing of a … motion stating that the patentee had no evidence of infringement and pointing to the specific ways in which accused [products] did not meet the claim limitations." *Exigent Tech. v. Atrana Solutions, Inc.*, 442 F.3d 1301, 1309 (Fed. Cir. 2006). The burden of production then shifts to the patentee to "identify genuine issues that preclude summary judgment." *Optivus Tech., Inc. v. Ion Beam Applications S.A.*, 469 F.3d 978, 990 (Fed. Cir. 2006).

Here, the court finds that the moving defendants have met their initial burden. It is undisputed that Jazelle cannot process Java bytecodes in the absence of JTEK software. It is also undisputed that the accused products do not use a JTEK-enabled JVM. The accused products are thus not configured to process Java bytecodes in hardware because their Jazelle circuitry is "dormant" by default.

Further, it is clear that the moving defendants did not "intend[] or anticipate[]" their products to be configured in an infringing manner. *High Tech Med. Instrumentation, Inc. v. New*

[1] Although Nazomi asserted claims under 35 U.S.C. § 271 (b)-(c) in its complaint, it did not respond to moving defendants' arguments that it had failed to provide any evidence showing indirect infringement and focused exclusively on direct infringement claims at oral argument. The court therefore grants summary judgment in favor of moving defendants as to Nazomi's claims for indirect infringement.

Image Indus., Inc., 49 F.3d 1551, 1555 (Fed. Cir. 1995). Neither product is designed to allow a user to download software that would "activate" Jazelle. *See* Ansis Decl. ¶ 3; Ex. A-5 at 100; *compare Finjan, Inc. v. Secure Computing Corp.*, 626 F.3d 1197, 1205 (Fed. Cir. 2010) (finding infringement of a software claim directed towards capability where consumers could purchase from the defendant a "key" to unlock the infringing functionality). Moreover, enabling Jazelle on the accused products would serve no "functional purpose not already accomplished by other means." *High Tech Med.*, 49 F.3d at 1556. WD produced uncontroverted testimony that a hardware-based JVM "was not stable enough" for use on the MyBook, and that it elected to employ a software-based solution instead. Ex. A-5 at 39; 56-57. The Slingbox does not use Java programs at all, and thus has no need for *any* JVM, let alone one that utilizes Jazelle. On the whole, the record indicates that although the accused devices contain hardware that could, in theory, be utilized to process Java bytecodes, that functionality was not sought by the moving defendants or intended to be accessible to their customers. *Compare Fantasy Sports*, 287 F.3d at 1118 (a software designer is liable for infringement if software is "written in such a way as to enable a user of that software to utilize the [infringing] function … without having to modify that code").

Nazomi argues that the mere presence of Jazelle circuitry is sufficient to find infringement, relying on *Silicon Graphics, Inc. v. ATI Techs., Inc.*, 607 F.3d 784 (Fed. Cir. 2010). In *Silicon Graphics*, the patents-in-suit related to a system used to render graphics for animated movies, including claims covering a "processor that rasterizes" and a "frame buffer … for storing" data. *Id.* at 795. The defendant made graphics processors, and was sued for indirect infringement. It argued that because its processors could not "store" or "rasterize" absent an operating system, licensed users of the Windows operating system were not liable for direct infringement under a provision in the Windows license agreement that provided immunity for any use that did not infringe "separate and apart from the combination" with the operating system. *Id.* at 794. The court disagreed, explaining that "[e]ven if the products cannot rasterize or store absent an operating system, they may include a rasterization circuit and a frame buffer for doing so. If they do, they infringe

separate and apart from the operating system, and the Immunity Provision does not apply." *Id.* at 795.

The court finds that *Silicon Graphics* is distinguishable for two reasons. First, because *Silicon Graphics* focused solely on the issue of infringement by end-users, it did not address whether the manufacturer could be held independently liable under Section 271(a). Second and more importantly, *Silicon Graphics* reaffirmed the rule that "an apparatus claim directed to a computer that is claimed in functional terms is nonetheless infringed so long as the [accused] product is designed in such a way as to enable a user of that [product] to utilize the function . . . without having to modify [the product]." *Id.* at 794 (quoting *Fantasy Sports*, 287 F.3d at 1118). Put another way, in determining whether an accused product is "configured" to perform a claimed function, a court must consider the manner in which *that product* is actually used. *See id.* ("[I]n every infringement analysis, the language of the claims, as well as the nature of the *accused product*, dictates whether an infringement has occurred.") (emphasis added). Unlike the processor in *Silicon Graphics*, which was designed to be used in conjunction with an operating system that activated the infringing functionality, the MyBook and Slingbox cannot process Java bytecodes in hardware when sold to consumers, nor can consumers enable Jazelle without first modifying the accused devices in a manner not contemplated by the moving defendants. Accordingly, *Silicon Graphics* does not support Nazomi's position here. The court therefore concludes that Nazomi has failed to establish that there are genuine factual issues precluding summary judgment.

III. ORDER

For the foregoing reasons, the court construes the asserted claims as described above, and grants the moving defendants' motion for summary judgment of non-infringement.

Dated: __8/14/12_____

Ronald M. Whyte

RONALD M. WHYTE
United States District Judge