UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

DYNETIX DESIGN SOLUTIONS INC., a California corporation, Plaintiff, v. SYNOPSYS INC., a Delaware corporation, and DOES 1-50, Defendants.	Case No.: CV 11-05973 PSG **ORDER GRANTING-IN-PART DEFENDANT'S MOTION FOR SUMMARY JUDGMENT** **(Re: Docket No. 62)**

In this patent infringement case, Defendant Synopsys Inc. ("Synopsys") moves for summary judgment of non-infringement on a variety of issues.[1] Plaintiff Dynetix Design Solutions Inc. ("Dynetix") opposes the motions, and seeks a summary judgment of its own.[2] The parties have appeared for multiple hearings on the motions. This order deals only with Synopsys' first motion for summary judgment of non-infringement. A further order addressing the remaining motions will issue. Having reviewed the papers and considered the arguments of counsel, the court GRANTS-IN-PART Synopsys' first summary judgment motion.

[1] See Docket No. 62, 136, 141.

[2] See Docket No. 87.

I. BACKGROUND

The reasoning that the court applies to the pending summary judgment motion is straightforward, and so the court will provide only a limited background.

On December 5, 2011, Dynetix filed this suit, alleging that various Synopsys products and in particular its VCS multicore technology infringe Dynetix's patent, United States Patent 6,466,898 ("the '898 patent").[3] Dynetix and Synopsys are both electronic design automation ("EDA") companies, involved in creating software tools to design and test integrated circuits.[4] The '898 patent discloses a multithread HDL logic simulator that can process both VHDL and Verilog languages in a single program, and uses special algorithms to accelerate performance on multiprocessor systems.[5] The VCS product is an EDA tool and a logic simulator.[6] VCS Multicore features two levels of parallelism: Design Level Parallelism ("DLP") and Application Level Parallelism ("ALP").[7] DLP allows the user to run a parallel simulation by dividing DUT into multiple partitions, then simulating those partitions on different threads.[8] ALP allows the user to run simulations in parallel with other applications.[9]

On September 3, 2012, before claim construction had taken place, Synopsys moved for partial summary judgment of non-infringement.[10] Synopsys challenges that Dynetix cannot prove

[3] See Docket No. 1. Synopsys filed an answer and cross-complaint denying infringement of Dynetix's patent and claiming that Dynetix's products infringe two of Synopsys's patents. See Docket No. 58.

[4] See Docket No. 1 ¶ 8; Docket No. 64 ¶ 3.

[5] '898 Patent at 1. Dynetix has asserted 18 claims of the '898 patent: claims 1-3, 5-7, 19-23, 36, 37, 39, 44, 45, 48, and 53. See Docket No. 143, Ex. B.

[6] See Docket No. 142 ¶ 8.

[7] See id. ¶ 9-10.

[8] See Docket No. 64 ¶ 2.

[9] See id. ¶ 17.

[10] See Docket No. 62.

VCS Multicore infringes claims 1-3, 5-7, 36, 37, 39, 44, 45, 48, and 53 ("the parallel simulation claims").[11] On October 10, 2012 the court held a claim construction hearing[12] and construed the term "to create a master thread and a plurality of slave threads" as "creating one thread for each processor where the master thread is executed on one processor and each of the slave threads is executed on a separate remaining processor."[13]

II. LEGAL STANDARDS

Summary judgment is appropriate only if there is "no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law."[14] The moving party bears the initial burden of production by identifying those portions of the pleadings, discovery and affidavits which demonstrate the absence of a triable issue of material fact.[15] If, as here, the moving party is the defendant, he may do so in two ways: by proffering "affirmative evidence negating an element of the non-moving party's claim," or by showing the non-moving party has insufficient evidence to establish an "essential element of the non-moving party's claim."[16] If met by the moving party, the burden of production then shifts to the non-moving party, who must then provide specific facts showing a genuine issue of material fact for trial.[17] The ultimate burden of persuasion, however, remains on the moving party.[18] In reviewing the record, the court must construe the evidence and

[11] Claims 19-23 ("the remote access claims") are not challenged in the present motion.

[12] See Docket No. 119.

[13] Docket No. 121 at 175-76.

[14] See Fed. R. Civ. P. 56(a).

[15] See Fed. R. Civ. P. 56(c)(1); Celotex Corp. v. Catrett, 477 U.S. 317, 323 (1986).

[16] Celotex Corp., 477 U.S. at 331.

[17] See id. at 330; *T.W. Elec. Service, Inc. v. Pac. Elec. Contractors Ass'n*, 809 F.2d 630, 630 (9th Cir. 1987).

[18] Id.

the inferences to be drawn from the underlying evidence in the light most favorable to the non-moving party.[19]

To infringe a claim, each claim limitation must be present in the accused product, literally or equivalently.[20] Patent infringement is a two-step process: first, the court must construe the asserted claims; then, the court must compare the accused products with the construed claims and determine whether the products contain each limitation of the claims, either literally or equivalently.[21] A product literally infringes if it contains each element and limitation of the patent claim as construed.[22] A product may also infringe under the doctrine of equivalents, which applies if the element in the accused device performs substantially the same function, in substantially the same way, to obtain substantially the same result as the element claimed in the patent.[23]

III. DISCUSSION

Of the parallel simulation claims challenged by Synopsys, only claims 1, 36, and 45 are independent. Claim 1 includes the following language:

> automatically detecting the number of microprocessors (CPUs) available on the multiprocessor platform to create a master thread and a plurality of slave threads for concurrent execution of the multithreaded event driven simulation of the design to achieve linear to super-linear scalable performance speedup as according to the number of CPUS on the multiprocessor platform.[24]

[19] See Anderson, 477 U.S. at 248; Matsushita Elec. Indus. Co., Ltd. v. Zenith Radio Corp., 475 U.S. 574, 587 (1986).

[20] See Dawn Equip. Co. v. Kentucky Farms, Inc., 140 F.3d 1009, 1014 (Fed. Cir. 1998).

[21] See Freedman Seating Co. v. American Seating Co., 420 F.3d 1350, 1356-57 (Fed. Cir. 2005).

[22] See id. at 1357.

[23] See Abbott Laboratories v. Sandoz, Inc., 566 F.3d 1282, 1296-97 (Fed. Cir. 2009).

[24] ‘898 Patent, col. 23, ll. 21-27.

Claims 36 and 45 contain the following language:

> creating a master thread and a plurality of slave threads, based on the number of available CPUs on the multiprocessor platform, prior to the start of simulation.[25]

All three independent claims thus require (1) automatic detection of the number of microprocessers available (the "Auto Detection Component") and (2) creation of threads based on the number of CPUs (the "Purpose Component").

Synopsys contends that both parallel features of VCS Multicore, DLP and ALP, do not automatically detect the number of available processors to create threads. As support, Synopsys submits the declaration of Pallab Dasgupta ("Dasgupta"), Director of Research and Development of the Verification Group.[26]

According to Dasgupta, DLP creates threads based on the characteristics of the Design Under Testing ("DUT").[27] DLP breaks up the DUT into partitions, then simulates those partitions in parallel.[28] Each partition is simulated on a different thread, and any part of the circuit not attributed to a partition is simulated on a remaining thread.[29] The DUT can be partitioned in two ways. The user can input a configuration file, which would inform DLP on how to partition the circuit design.[30] Alternatively, if the user is unsure how to partition the DUT, the user can use the DLP "autopartition" feature, which "analyze[s] the DUT and inform[s] the user how to partition the circuit in a way that will maximize parallelism."[31] DLP thread creation thus depends solely on the number of partitions in the DUT, whether generated by the configuration file or the

[25] Id.at col. 28, ll. 33-37; col. 29, ll. 43-44 (emphasis added).

[26] See Docket No. 64 ¶ 2.

[27] See id. ¶ 5, 11.

[28] See id. ¶ 12.

[29] See id. ¶ 15.

[30] See id.

[31] See id. ¶ 14.

autopartitioning feature, but never on the number of processors available as required by the parallel simulation claims.[32]

With respect to ALP, Dasgupta explains that ALP allows the user to run various applications alongside the simulation.[33] It does this by creating additional "processes" (essentially equivalent to a thread) for each additional application.[34] ALP thread creation thus depends on the number of applications being run, and not the number of processors available on the user's hardware platform.[35]

Dynetix, in turn, submits expert testimony from Minesh B. Amin ("Amin"), informed by Amin's review of VCS Multicore source code, user manuals, and release notes.[36] Amin focuses on a portion of the source code in VCS Multicore's autopartitioning feature that detects the number of available CPUs.[37] The autopartitioning element then limits the number of partitions to be created to the total number of CPUs minus one.[38] Because the number of threads equals the number of slave partitions, the total number of threads generated (both slave and one master thread) is at least sometimes equal to the number of CPUs available.[39] In this scenario, although VCS uses an additional step of first partitioning the design at the compile time, converting the partitions into slave threads at the runtime stage practices both the Auto Detection and Purpose Components.

[32] See id. ¶ 16.

[33] See id. ¶ 17.

[34] See id. ¶ 18.

[35] See id. ¶ 20.

[36] See Docket No. 168-2 ¶ 31.

[37] See id. ¶ 47.

[38] See id. ¶ 48.

[39] See id.

In its reply, Synopsys argues that the autopartitioning source code relied upon by Amin is "blocked" by other code in the program and therefore can never be executed.[40]

The net result of all this is that, at least with respect to DLP, there is a genuine issue of material fact as to whether the autopartitioning feature infringes the parallel simulation claims. While Synopsys has presented evidence showing the accused product does not practice a key limitation of the claims in question, Dynetix has presented competent evidence to counter that assertion. In particular, Dynetix's expert Amin points to portions of the source code that indicate the user does not supply a variable, the program launches into the autopartioning mode described as infringing.[41] This is a classic "battle of the experts" on a material issue of fact.[42] It is the jury's province to resolve such issues, not the court's.[43]

As for ALP, even if it could show evidence that ALP has been used, Dynetix offers no evidence or even argument to rebut Synopsys's assertion that ALP does not practice the particular claim limitations at issue in this motion. All of Amin's testimony regarding autopartitioning appears to be directed to DLP exclusively. Under Fed. R. Civ. P. 56(c), a party asserting that a fact cannot be genuinely disputed must support this assertion by "citing to particular parts of materials in the record," or else "showing that the materials cited do not establish the absence or presence of

[40] See Docket No. 205 at 7.

[41] See Docket No. 168-2 ¶ 38 (citing Ex. 2). To be sure, at least with respect to method claims, "[i]t is not enough to simply show that a product is capable of infringement; the patent owner must show evidence of specific instances of direct infringement." See Fujitsu Ltd. v. Netgear Inc., 620 F.3d 1321, 1329 (Fed. Cir. 2010). But Dynetix has presented an affidavit showing these facts are unavailable to it and the court has issued not just one but two orders compelling Synopsys to produce evidence relating to this issue. See, e.g., Docket No. 256 (ordering production of simulation results and data, which may demonstrate use of the autopartitioning feature). Two other motions to compel are pending. See Docket No. 262, 284. Under such circumstances, it would be unjust to penalize Dynetix for failing to tender this very same evidence. See Fed. R. Civ. P. 56(d)(1); Docket No. 72-1.

[42] In re Gabapentin Patent Litig., 503 F.3d 1254, 1260 (Fed. Cir. 2007).

[43] See Regents of Univ. of California v. Dako N. Am., Inc., Case No. 05-03955 MHP, 2009 WL 1083446, at *15 (N.D. Cal. Apr. 22, 2009) (explaining that a "battle of the experts" is appropriately left to the trier of fact to resolve).

United States District Court
For the Northern District of California

a genuine dispute." Dynetix has done neither in its opposition. Therefore, summary judgment regarding ALP's non-infringement of the claim limitations at issue in this motion is appropriate.

IV. CONCLUSION

Synopsys' first motion for summary judgment of non-infringement is GRANTED-IN-PART.

IT IS SO ORDERED.

Dated: March 31, 2013

PAUL S. GREWAL
United States Magistrate Judge