

United States District Court
For the Northern District of California

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA

ORACLE AMERICA, INC.,
Plaintiff,

No. C 10-03561 WHA

v.

GOOGLE INC.,
Defendant.

**ORDER DENYING ORACLE'S
MOTION FOR JUDGMENT
AS A MATTER OF LAW RE
PATENT INFRINGEMENT**

INTRODUCTION

In this patent and copyright infringement action, Oracle moves for judgment as a matter of law that Google infringed Patent Nos. RE38,104 and 6,061,520. For the reasons stated below, the motion is **DENIED**.

STATEMENT

The procedural and technological background of this action has already been discussed (Dkt. Nos. 137, 433). This action began in 2010 with seven asserted patents. In May 2011, a claim construction order was issued (Dkt. No. 137). Neither party filed a motion for reconsideration of that order. In January 2012, at the request of the parties, a supplemental claim construction order for additional terms was issued (Dkt. No. 704).

By the time of trial in April 2012, Oracle had reduced the number of asserted patents and claims to two patents and eight claims. This was done as an incentive to receive an earlier trial date and because the withdrawn patents had received "final rejections" in PTO reexaminations.

1 The issue of infringement was tried to a jury and Google dropped all affirmative defenses,
2 including invalidity. The parties stipulated that indirect infringement (inducement and
3 contributory infringement) would automatically follow from a finding of direct infringement
4 (Dkt. No. 1139). Because Oracle had not disclosed an infringement theory under the doctrine of
5 equivalents, the only issue for the jury to decide was whether Android literally infringed the
6 asserted claims in the '104 and '520 patents.

7 The parties also narrowed down the claim limitations in disputed (TX 1106). For
8 the '104 patent, the dispute was whether Android met the limitation of having “instructions
9 containing one or more symbolic references.” Claim 11 is a representative example of the claim
10 language:

11 An apparatus comprising:

12 a memory containing intermediate form
13 object code constituted by a set of
14 instructions, certain of said instructions
15 containing one or more symbolic references;
16 and

17 a processor configured to execute said
18 instructions containing one or more
19 symbolic references by determining a
20 numerical reference corresponding to said
21 symbolic reference, storing said numerical
22 references, and obtaining data in accordance
23 to said numerical references.

24 (TX 1106) (for illustration, the disputed limitations are underlined). All asserted claims for
25 the '104 patent contained the limitation that the “symbolic reference” must be in the instructions.
26 Google stipulated that Android contained the non-underlined limitations in the asserted claims.
27 For the '520 patent, the dispute between the parties was whether Android satisfied the limitation
28 of “simulating execution . . . to identify the static initialization of the array.” Claim 1 is a
representative example of the claim language:

A method in a data processing system for statically
initializing an array, comprising the steps of:

compiling source code containing the array
with static values to generate a class file
with a clinit method containing byte codes
to statically initialize the array to the static
values;

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

receiving the class file into a preloader;
simulating execution of the byte codes of the clinit method against a memory without executing the byte codes to identify the static initialization of the array by the preloader;
storing into an output file an instruction requesting the static initialization of the array; and
interpreting the instruction by a virtual machine to perform the static initialization of the array.

(col. 9). All asserted claims for the '520 patent contained the “simulating execution” limitation.

After a week-long trial, followed by another week of jury deliberation, the jury rendered a verdict of non-infringement for all asserted claims. Oracle moves to set aside the jury verdict as a matter of law under Rule 50.

ANALYSIS

Judgment as a matter of law may be granted against a party if it “has been fully heard on an issue and there is no legally sufficient evidentiary basis for a reasonable jury to find for that party on that issue.” Rule 50(a). “Judgment as a matter of law may be granted only where, so viewed, the evidence permits only one reasonable conclusion, and that conclusion is contrary to the jury’s verdict.” *Wallace v. City of San Diego*, 479 F.3d 616, 624 (9th Cir. 2007).

The final charge to the jury was and remains the complete statement of the law governing the trial issues unless a timely and proper objection to the instructions was made and the Court now agrees that the objection has merit. The post-trial motions under Rule 50 must be evaluated, therefore, in light of that statement of the law and not in light of new citations, new legal theories and/or legal variations. Rule 50 is not an occasion for yet another round of summary judgment based on new slants on the case law. Of course, an appellate court might later find that an objection to an instruction was preserved, well-taken and not harmless and so require a new trial. But until then, this Court is satisfied that the jury instructions were proper and both sides are bound by the jury instructions as the exclusive statement of the governing law for the instant action.

1 Where a party does not object to the court’s jury instruction regarding the construction of
2 claim term, the verdict must be tested by the charge actually given under the ordinary meaning of
3 the language of the jury instruction. *Cordis Corp. v. Boston Scientific Corp.*, 658 F.3d 1347,
4 1355–56 (Fed. Cir. 2011).

5 [P]arties cannot reserve issues of claim construction
6 for the stage of post-trial motions. When issues of
7 claim construction have not been properly raised in
8 connection with the jury instructions, it is improper
9 for the district court to adopt a new or more detailed
10 claim construction in connection with the JMOL
11 motion. On JMOL, the issue here should have been
12 limited to the question of whether substantial
13 evidence supported the verdict under the agreed
14 instruction. In other words, where the parties and
15 the district court elect to provide the jury only with
16 the claim language itself, and do not provide an
17 interpretation of the language in the light of the
18 specification and the prosecution history, it is too
19 late at the JMOL stage to argue for or adopt a new
20 and more detailed interpretation of the claim
21 language and test the jury verdict by that new and
22 more detailed interpretation. The verdict must be
23 tested by the charge actually given and by giving
24 the ordinary meaning of the language of the jury
25 instruction.

26 *Hewlett-Packard Co. v. Mustek Systems, Inc.*, 340 F.3d 1314, 1320–21 (Fed. Cir. 2003)
27 (citations omitted).

28 **1. THE ’104 PATENT.**

 As discussed, the dispute with respect to infringement of the ’104 patent was whether
Android’s Dalvik bytecode instructions contained “symbolic references.” More specifically, the
question for the jury was whether the field indices and operands in the Dalvik bytecode
instruction stream were “symbolic references.”

A. Meaning of “Symbolic Reference.”

 The jury instructions contained the Court’s May 2011 construction of the term “symbolic
reference,” which was defined as “a reference that identifies data by a name other than the
numeric memory location of the data, and that is resolved dynamically rather than statically”
(Dkt. No. 137 at 22; Dkt. No. 1153 at 5). In formulating this definition, the order did not adopt
either parties’ proposed definition. During the May 2011 *Markman* proceeding, Oracle had

1 argued that the term “symbolic reference” did not need construction and that the plain meaning
2 meant “a reference by name.”

3 Importantly, Oracle did *not* object to the use of the term “data” in the Court’s
4 construction or request additional clarification before the claim construction’s definition went
5 into the final jury instructions. During the charging conference, the Court specifically asked
6 whether Oracle wanted to construe the term “data.” Oracle declined.

7 COURT: No one has asked for any other claim
8 constructions. If you want me to put in a sentence
9 that would say as to anything else it has its plain
10 meaning as used in the patent, I can do that. But I
11 don’t — you can’t sandbag me at the last minute
12 and say, Judge, please give us a ruling on what
13 “data” means. That would take me some time to
14 sort out. So if — you know, this was the time to
15 ask me for it. I guess you could — for good cause
16 you could ask for it later. But, it would have to be
17 really good cause to say at the last minute you want
18 me to go research a legal point and do a claim
19 construction. But I hear you saying you’re happy
20 with the plain language.

ORACLE’S COUNSEL: We’re happy with the plain
language of the claim and the claim constructions.

16 (Tr. 3710–11). Oracle’s answer was an unequivocal “no.” Therefore, the ordinary meaning of
17 the term “data” in the jury instructions controls. *See Cordis Corp.*, 658 F.3d at 1355–56.

18 Additionally, the jury asked a question regarding the meaning of symbolic reference
19 during their deliberations. The jury asked whether a reference can be both a numeric reference
20 *and* a symbolic reference. Over Oracle’s objections, the Court answered:

21 You look at whatever is in that little box and you
22 say, does that identify data by a name other than the
23 numeric memory location of the data? If the answer
24 to that is that it identifies a numeric memory
25 location of the data, then it is not a symbolic
26 reference; it’s a numeric reference. And it doesn’t
27 get transmogrified into a something else on account
28 of what happens downstream.

But you do have to consider what happens
downstream to find — to at least figure out whether
or not the item that is in that box is referring to the
numeric memory location of the data. You’ve got
to look at least that far downstream in order to see
what it’s referring to. But, once you do that — and
if you decide it’s a symbolic reference, great. If you

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

decide it's a numeric reference, that's great, too.
But it doesn't then change because the computer
marches on to one instruction after the other, and so
forth.

(Tr. 4325–26). To sum up, the Court answered that a reference cannot be a “symbolic reference” for one datum and a “numeric reference” for another datum within the meaning of the claim language.

This answer was appropriate and already inherent in the claim construction. *Cordis Corp.*, 658 F.3d at 1356. Based the patent specifications, a reference cannot be both a “symbolic reference” and a “numeric reference” within the meaning of the claims. The patent expressly stated that a reference was either a “symbolic reference *or* a numeric reference” (col. 2). The patent described two different subroutines that were invoked, depending on whether the reference was numeric or symbolic. There was nothing in the patent describing how to *first* determine whether a reference was numeric or symbolic with regard to any particular datum. The only reasonable understanding of the patent was that a reference would be recognized by an interpreter as either numeric or symbolic but not both. This understanding was inherent in the Court’s claim construction order, which stated that “[t]he ’104 patent teaches two different types of data references: numeric references and symbolic references” (Dkt. No. 137 at 20).

Furthermore, the testimony at trial of Oracle’s own expert, Dr. John Mitchell, supports this mutually exclusive characterization of a reference as either a numeric or symbolic reference. In his opening expert report, Dr. Mitchell labeled the indexes in the Dalvik bytecode instructions as “numeric references.” Realizing that this could be problematic to Oracle’s infringement contentions, in his trial testimony, Dr. Mitchell testified that his report was *mistaken* and that those indexes were *instead* symbolic references (*see, e.g.*, Tr. 3490). This 180-degree change of opinion proves the mutually exclusive characterization of a reference (within the meaning of the patent claims): if a reference could be both a numeric and a symbolic reference, then Dr. Mitchell would not have needed to repudiate his expert report at trial and instead could have supplemented his report by testifying that the indexes were both numeric and symbolic references.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

B. Sufficient Evidence to Support Jury’s Verdict.

There was sufficient evidence at trial to support the jury’s finding of non-infringement. At trial, the Dalvik bytecode instruction was the focus of both parties’ arguments. In particular, the parties focused on the IGET instruction. The IGET instruction retrieved a piece of data out of an instance field object (Tr. 3221). The IGET instruction contained an operand — field@CCCC — also known as the field index (Tr. 3221). Android resolved class, method, and string indices in the same way as the field indices and both parties agreed that the field index example was dispositive for all indices.

Oracle’s theory of infringement presented to the jury was that the field indices — operands in Davlik instructions — were symbolic references. In defense, Google argued that the same field indices were not symbolic references but were instead, numeric references, because they were pointers to table entries from the Field ID tables (a table in the constant pool of Android dex files). The jury sided with Google’s argument.

There was ample evidence to support the jury’s verdict of non-infringement. Google’s expert Dr. August testified that the term, “data” from the Court’s claim construction of “symbolic references” would have been understood to include data from Field ID tables in the dex files. Dr. August illustrated with an example of Android’s IGET instruction “52 01”, which instructed the computer to go to entry 01 in the Field ID table. Dr. August explained that the operand (“01”) for the IGET instruction qualified as a numeric reference because it referred to data by a numeric memory location: entry 01 in the Field IDs table. That is, the operand “01” identified data (the value at entry 01 in the Field ID table) in the constant pool by the numeric memory location, entry 01.

Q: The premise being that the certain of said instructions containing one or more symbolic references.

And you’re saying that — well, what is “01”? Is that, in your view, a symbolic reference?

Dr. August: No. “01” is a numeric reference because it gives you an actual numeric memory location. Right

1 here (indicating). This is the
2 location, 01.

3 There's no resolution, no search.
4 Nothing – nothing expensive about
5 figuring out what that instruction is
6 referring to when it goes to the field
7 ID table.

8 (Tr. 3865). His testimony took Android's operand "01" out of the plain meaning of the Court's
9 "symbolic reference" definition, and thus, the claim limitations of the '104 patent. The jury was
10 reasonable in crediting Dr. Parr's testimony.

11 Oracle's argued (and continues to argue) that "data" in the claim construction and claims
12 themselves referred narrowly to "the ultimate data to be obtained or used after symbolic
13 reference resolution is performed" (Dkt. No. 1189 at 1). This argument was reasonable rejected
14 by the jury. The values in the Field ID table in the Android dex file were "data" in the ordinary
15 sense of the word. The jury was reasonable to adopt this understanding of the term "data" and
16 reject Oracle's expert Dr. Mitchell's opinion that "data" was restricted to "ultimate data" in
17 Android. Google's Dr. August persuasively explained that there was no limitation regarding the
18 type of data being referenced in the plain meaning of the Court's claim construction. He
19 explained on cross-examination (Tr. 3955):

20 Q: Now, just to clean up a few other
21 things. You labeled this 'data,' but
22 this is actually the constant pool;
23 true, sir?

24 Dr. August. Constant pool is data.

25 Even adopting Oracle's belated effort to construe "data" to mean "the ultimate data to be
26 obtained or used after symbolic reference resolution is performed," a reasonable jury could still
27 find that Android did not infringe. Dr. August testified that the field data in the instance object
28 referenced by the operand (Oracle's definition of a "data" in Android) may itself be a reference
to other data. For example, that data may reference another string in the string data table.

Q. What can be in this instance object
by way of data?

Dr. August. Well, it could be a symbolic
reference. It could be a numeric

1 reference. Or it could be other non-
2 reference data.

3 * * *

4 Q. Let's talk about in Dalvik. Is the
5 data at these entries always non-
6 reference data, as we talked about in
7 your technical tutorial?

8 Dr. August. No, it's not.

9 Q. How do you know that?

10 Dr. August: Because I write programs. And you
11 can write programs that contain
12 references in instances of objects.

13 (Tr. 4002–03). Thus, the jury was reasonable to believe that the operand “01” in Android’s
14 instruction stream did not identify the “ultimate” data within the meaning of the claim language.

15 Oracle also argues Android’s operand “01” must be a symbolic reference because it was
16 resolved and only symbolic references had to be resolved. Even if this construction were
17 adopted, the jury was still reasonable to find that Android did not infringe. There was testimony
18 that Android’s field indices were *not* resolved. As Mr. Andy McFadden explained:

19 Q. And are indexes in this process being
20 resolved?

21 Dr. August. No.

22 Q. Why do you say no?

23 Dr. August. Well, resolution implies something is
24 unknown, something is ambiguous.
25 If you have an index, you know
26 exactly where you're going. You
27 have the location. For symbols, you
28 don't know where you're going and
you won't know until you have
resolved the ambiguity. So it doesn't
really make sense to say that you
“resolve an index.”

(Tr. 3650–51).

Therefore, Oracle’s motion for judgment of infringement of the ’104 patent is **DENIED**.

2. ’520 PATENT.

Oracle argues that Android SDK’s dx tool infringes claims 1 and 20 of the ’520 patent as
a matter of law. At trial, the only dispute with respect to infringement of the ’520 patent was

1 whether Android’s pattern-matching technique in its dx tool was encompassed in ’520 patent
2 “simulating execution” process limitation. There was no dispute that the dx tool used a
3 technique called “pattern matching” that looked for a repeating sequence of instructions used to
4 initialize static arrays (Tr. 3800–01). The jury was reasonable to credit the testimony of
5 Google’s expert, Professor Terence Parr, who testified that Android’s pattern-matching
6 technique was not simulating execution because it did not go through the steps required to
7 produce output showing what the static array would look like if the Java byte code instructions
8 were performed.

9 **A. Meaning of “Simulating Execution.”**

10 Neither party requested construction of the term “simulating execution.” Neither party
11 sought to clarify the term in the jury instructions. Therefore, the verdict must be tested under the
12 ordinary meaning of “simulating execution.” *Cordis Corp.*, 658 F.3d at 1355–56.

13 The purpose of the ’520 patent was to efficiently create an instruction for static
14 initialization of an array for the Java virtual machine, a stack-based machine. The specification
15 discussed static initialization only through simulating actual Java virtual machine operations:
16 manipulation of a stack by pushing, popping, and replacing values from the top of an operand
17 stack. Thus, the ordinary meaning of “simulating execution” is the simulation of actual Java
18 virtual machine operations. Google’s expert, Dr. Parr, also testified that this was the plain
19 meaning that would have been understood by those skilled in the art (Tr. 3794).

20 **B. Sufficient Evidence to Support Jury’s Verdict.**

21 It was undisputed that Android did not manipulate a stack to identify the initialization of
22 an array. At trial, Professor Parr confirmed this by looking at the source code and also
23 performing multiple experiments. As an experiment, he inserted an extra piece of bytecode that
24 put a zero into the zeroth entry in an array — where a zero already existed at the inception of the
25 array. Such an extraneous piece of code, when run through a program that simulated execution
26 of byte code instructions, would create an efficient instruction with no problem because the
27 useless instruction would end up having no effect on the eventual result (Tr. 3810). However, in
28 his experiment, the dx tool failed to create a proper instruction when it was fed that extraneous

1 code because it failed to find a particular sequence of instructions through pattern matching (Tr.
2 3809–10). The jury was reasonable to find that the dx tool code did not simulate execution of
3 the byte code within the meaning of the claim limitation.

4 In its Rule 50 motion, Oracle argues that the limitation of “simulating execution” must,
5 as a matter of law, encompass Android’s dx tool because the dx tool “examines the bytecode
6 without executing them.” This belated construction of “simulating execution” is rejected. The
7 jury was free to reject Oracle’s evidence that Android’s pattern matching was within the plain
8 meaning of “simulating execution.” The plain meaning of “simulating execution” does not
9 encompass pattern matching, and there was no mention of pattern matching in the ’520 patent
10 itself (Tr. 3521; TX 4011).

11 Therefore, Oracle’s motion for judgment of infringement of the ’520 patent is **DENIED**.

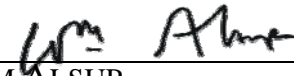
12 **CONCLUSION**

13 The foregoing is sufficient but it is worth adding that Oracle’s infringement case was
14 presented through Dr. Mitchell. A reasonable jury could have found his many “mistakes” in his
15 report merely to be convenient alterations to fix truthful admissions earlier made before he
16 realized the import of his admissions. For this reason, a reasonable jury could have rejected
17 every word of his testimony.

18 For the reasons stated, Oracle’s motion for judgment as a matter of law of infringement
19 is **DENIED**.

20
21 **IT IS SO ORDERED.**

22
23 Dated: May 30, 2012.

24 

25 WILLIAM ALSUP
26 UNITED STATES DISTRICT JUDGE
27
28