

Bruce S. Sostek, admitted *pro hac vice*
bruce.sostek@tklaw.com
Richard L. Wynne, Jr., admitted *pro hac vice*
richard.wynne@tklaw.com
Adrienne E. Dominguez, admitted *pro hac vice*
adrienne.dominguez@tklaw.com
THOMPSON & KNIGHT LLP
One Arts Plaza
1722 Routh Street, Suite 1500
Dallas, Texas 75201
Telephone: (214) 969-1700
Facsimile: (214) 969-1751

John V. Picone III, Bar No. 187226
jpicone@hopkinscarley.com
HOPKINS & CARLEY
A LAW CORPORATION
The Letitia Building
70 South First Street
San Jose, CA 95113-2406

mailing address:

P.O. Box 1469
San Jose, CA 95109-1469
Telephone: (408) 286-9800
Facsimile: (408) 998-4790
Attorneys for Plaintiffs
BROADCOM CORPORATION and AVAGO
TECHNOLOGIES INTERNATIONAL SALES PTE.
LIMITED.

**United States District Court
For the Northern District of California
San Francisco Division**

BROADCOM CORPORATION, ET AL.,
Plaintiffs,
v.
NETFLIX, INC.,
Defendant.

Case No. 3:20-cv-04677-JD
Plaintiffs' Opening Claim-Construction Brief
Dept.: Courtroom 11, 19th Floor
Judge: Hon. James Donato
Hearing Date: T.B.D.
Time: 10:00 a.m.

TABLE OF CONTENTS

Table of Authorities	iii
A. '121 Patent	1
Disputed Term—"display pipeline"	1
B. '387 and '663 Patents	3
1. Technical Background	3
2. Disputed Terms	4
a. '663 Patent	4
i. Claim 12.	4
ii. Claims 18 and 19	7
b. '387 Patent	9
i. Means-Plus-Function Claiming.	10
ii. Netflix's construction improperly reads out several of the disclosed algorithms.	11
C. '283 Patent	14
1. Technical Background	14
2. Disputed Term—"wherein the second syntax element specifies the PU partition mode for the selected CU, wherein the PU partition mode is based on a size N×N PU when the selected CU is a smallest CU (SCU) of the plurality of CUs and is based on a different size PU than the size N×N PU when the selected CU is another CU than the SCU of the plurality of CUs"	17
D. '138 Patent	18
1. Technical Background	18
2. Disputed Terms	19
a. "virtual machine manager"	19
b. "rules engine"	20

TABLE OF AUTHORITIES

Cases

<i>AllVoice Computing PLC v. Nuance Commc'ns, Inc.</i> ,	
504 F.3d 1236 (Fed. Cir. 2007)	10, 11, 14
<i>Aristocrat Techs. Austl. Pty Ltd. v. Int'l Game Tech.</i> ,	
521 F.3d 1328 (Fed. Cir. 2008)	10
<i>Baldwin Graphic Sys., Inc. v. Siebert, Inc.</i> ,	
512 F.3d 1338 (Fed. Cir. 2008)	5
<i>Broadcom Corp. v. Sony Corp.</i> ,	
No. SACV 16-1052 JVS, Claim-Construction Order, ECF No. 150, at (May 18, 2017).	6
<i>Chrimar Holding Co., LLC v. ALE USA Inc.</i> ,	
732 F. App'x 876 (Fed. Cir. 2018).	19
<i>Digital-Vending Servs. Int'l, LLC v. Univ. of Phoenix, Inc.</i> ,	
672 F.3d 1270 (Fed. Cir. 2012)	2
<i>Finjan, Inc. v. Secure Computing Corp.</i> ,	
626 F.3d 1197 (Fed. Cir. 2010)	19
<i>Guzik Tech. Enters., Inc. v. W. Digital Corp.</i> ,	
No. 11-cv-03786-PSG, 2013 WL 3934892 (N.D. Cal. July 19, 2013)	14
<i>Harris v. Ericsson Inc.</i> ,	
417 F.3d 1241 (Fed. Cir. 2005)	12, 13, 14
<i>Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.</i> ,	
381 F.3d 1111 (Fed. Cir. 2004)	20
<i>Johnstech Int'l Corp. v. JF Microtechnology Sdn Bhd</i> ,	
No. 14-cv-02864-JD, 2016 WL 631936 (N.D. Cal. Feb. 17, 2016)	1
<i>Markman v. W. Instruments, Inc.</i> ,	
517 U.S. 370 (1996)	1
<i>Micro Chem., Inc. v. Great Plains Chem. Co.</i> ,	
194 F.3d 1250 (Fed. Cir. 1999)	14
<i>Netword, LLC v. Centraal Corp.</i> ,	
242 F.3d 1347 (Fed. Cir. 2001)	20
<i>Phillips v. AWH Corp.</i> ,	
415 F.3d 1303 (Fed. Cir. 2005)	1
<i>Serrano v. Telular Corp.</i> ,	
111 F.3d 1578 (Fed. Cir. 1997)	10, 13, 14

1	<i>SuperGuide Corp. v. DirecTV Enters., Inc.</i> ,	
2	358 F.3d 870 (Fed. Cir. 2004)	3
3	<i>Typhoon Touch Techs., Inc. v. Dell, Inc.</i> ,	
4	659 F.3d 1376 (Fed. Cir. 2011)	10
5	<i>Univ. of Pittsburgh of Commonwealth Sys. of Higher Educ. v. Varian Med. Sys., Inc.</i> ,	
6	561 F. App'x 934 (Fed. Cir. 2014).	12, 14

Statutes

7	35 U.S.C. § 112	10
---	---------------------------	----

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

Following the Court’s procedures, the parties were able to reach agreement on a number of terms, but some disputes remain (agreed and disputed terms are listed in Exhibit A).^{1 2}

A. ’121 Patent

Disputed Term—“display pipeline”

The ’121 Patent “relates to a network environment in an A/V system using ‘A/V decoders,’ where the A/V decoders are adapted to process, decode or decompress one or more input data streams.” ’121 Patent at 1:41–45. The systems and methods described in the patent improve the functionality of computer networks used for processing A/V data. Independent claim 1 is exemplary (disputed term in bold):

1. A network for processing data configured by a controller to form at least one **display pipeline** therein by dynamically selecting use of at least two selectable nodes from a plurality of selectable nodes and dynamically concatenating the selected at least two selectable nodes in the network together, wherein said at least one **display pipeline** has an independent data rate and a flow control module enables said independent data rate.

Broadcom contends “display pipeline” should be given its plain-and-ordinary meaning, which, in the context of the patent, is “chain of multiple nodes.” Netflix proposes that it should be construed to mean “a series of video processing modules.”

¹ The disputed terms are in the following patents: **Ex. B:** U.S. Patent No. 8,259,121 (the ’121 Patent); **Ex. C:** U.S. Patent No. 6,744,387 (the ’387 Patent); **Ex. D:** U.S. Patent No. 6,982,663 (the ’663 Patent); **Ex. E:** U.S. Patent No. 9,332,283 (the ’283 Patent); and **Ex. F:** U.S. Patent No. 8,572,138 (the ’138 Patent).

² Broadcom bases its proposed constructions of the disputed terms on the Supreme Court’s, Federal Circuit’s, and this Court’s legal standards governing claim construction. *See, e.g., Markman v. W. Instruments, Inc.*, 517 U.S. 370 (1996); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc); *Johnstech Int’l Corp. v. JF Microtechnology Sdn Bhd*, No. 14-cv-02864-JD, 2016 WL 631936, at *1–3 (N.D. Cal. Feb. 17, 2016).

Broadcom’s construction follows the specification, which consistently states that the claimed display pipeline is formed by chaining multiple nodes together:

- “the display pipeline 440 is changeably formed by *chaining*, coupling or concatenating one or more network *nodes* together,” *id.* at 6:44–46 (emphasis added);
- “the system 500 may include a number of pipelines (i.e., display pipelines) formed by *chaining* multiple *nodes* together,” *id.* at 7:36–38 (emphasis added);
- “the network 600 may include a number of display pipelines formed by *chaining* multiple *nodes* together,” *id.* at 8:47–48 (emphasis added).

When viewed in light of these references, there is no basis for modifying the scope of the claim to include *modules* that are arranged in a *series*, as Netflix proposes—indeed, the word “series” appears nowhere in the patent. A person of ordinary skill in the art, reading the claim in view of the specification, would understand that the claimed display pipeline is not just a *series* of components, but a *chain* of linked nodes. *See Digital-Vending Servs. Int’l, LLC v. Univ. of Phoenix, Inc.*, 672 F.3d 1270, 1280 (Fed. Cir. 2012) (“Our case law is clear that a claim term should be accorded the full breadth of its plain and ordinary meaning as understood by a person of ordinary skill in the art when read in the context of the specification and prosecution history.”).

Netflix’s construction also wrongly states that display pipelines are made up of *modules*. The claim, however, explicitly states that a display pipeline is formed by “concatenating the selected at least two selectable nodes in the network together.” ’121 Patent at claim 1. In fact, the claim distinguishes nodes from flow-control modules, which control data rate through the pipeline: “said at least one display pipeline has an independent data rate and a *flow control module enables said independent data rate*.” *Id.* at claim 1 (emphasis added); *see also id.* at 8:47–50 (“[T]he network 600 may include a number of display pipelines formed by chaining multiple nodes together using the *network modules 620 to switch between the nodes 616*, thus

1 varying or changing the pipeline.” (emphasis added)). Defining “data pipeline” as a chain of
 2 modules risks confusing the jury by conflating the terms node and module.

3 Finally, Netflix improperly adds a “video processing” limitation to the claim. There is no
 4 basis for limiting the display pipeline to a chain of *video processing* nodes. *See SuperGuide*
 5 *Corp. v. DirecTV Enters., Inc.*, 358 F.3d 870, 875 (Fed. Cir. 2004) (“[I]t is important not to
 6 import into a claim limitations that are not a part of the claim. For example, a particular
 7 embodiment appearing in the written description may not be read into a claim when the claim
 8 language is broader than the embodiment.”). This is particularly true here, since the patent is
 9 directed to A/V networks, through which not only visual data, but also audio data is transmitted.
 10 A construction that reads the transmission of audio data out of the claim would render the
 11 network useless as an A/V network.

12 “Display pipeline” should be given its plain-and-ordinary meaning—“chain of multiple
 13 nodes.”

14 **B. ’387 and ’663 Patents**

15 **1. Technical Background**

16 The ’387 and ’663 Patents share, in substance, a common specification and are “directed to
 17 an improved method for the binarization of data in an MPEG data stream.” ’387 Patent at
 18 Abstract; *see also id.* at 1:7–11. In the prior art, several techniques were available for binarizing
 19 data,³ including, for example, unary, binary, Golomb, and exp-Golomb binarization, each of
 20 which has certain advantages and weaknesses. *See id.* at 4:50–6:24. The patent discloses a new,
 21 hybrid binarization scheme “that retains the most valuable properties of the unary and exp-
 22 Golomb binarizations.” *Id.* at 6:26–36. In this scheme, certain portions of a codeword are filled
 23 using unary binarization and others are filled using exp-Golomb binarization.

24
 25
 26 ³ That is, representing data in a system based on the numerals 0 and 1.

Index	Unary Prefix	exp-Golomb Suffix
0	0	
1	10	
2	110	
...		
15	1...10	
16	1...110	0
17	1...110	1
18	1...1110	00
19	1...1110	01
20	1...1110	10
21	1...1110	11
22	1...11110	000
23	1...11110	001
24	1...11110	010
25	1...11110	011
26	1...11110	100
27	1...11110	101
...		

An annotated version of Table 4 of the '387 Patent (Figure 6 of the '663 Patent) provides an example of the multiple patterns and portions of such hybrid codewords. Each numbered row has a codeword in binary form, beginning with “0” in row 0, and continuing with “10” at row 1, “110” at row 2, and so forth. The blue, red, and yellow portions of the annotated figure illustrate first, second, and third portions of the codewords. The superimposed numbers in white annotate first, second, third, and fourth binarization patterns used to fill the portions.

The claims at issue are directed to methods and means for generating binarized codewords when an index value v is less than, equal to, or greater than a threshold N . If the index value is greater than or equal to the threshold value, each portion of the codeword is filled with a certain pattern to create “a modified exp-Golomb binarization for larger codewords.” *See* '387 Patent at 6:37–42. If the index value is less than the threshold, the first (blue) portion of the codeword is filled with another pattern (“a unary binarization for small codeword indices”), leaving the second (red) and third (yellow) portions empty. *Id.*

2. Disputed Terms

a. '663 Patent

i. Claim 12

Claim 12 of the '663 Patent claims a method for generating patterns in three portions of a

1 codeword. It recites:

2 12. A method for generating a codeword from an index value for digital video
3 encoding, comprising the steps of:

4 (A) generating a first pattern in a first portion of said codeword in response to said
5 index value being at least as great as a threshold;

6 (B) generating a second pattern in a second portion of said codeword following
7 said first portion representing an offset of said index value above said
8 threshold; and

9 (C) generating a third pattern in a third portion of said codeword following said
10 second portion representing a value of said index value above said offset.

11 Nothing in the express language of the claim, the specification, or the prosecution history
12 indicates, much less requires, that the three patterns must be generated in any particular order.
13 *See Baldwin Graphic Sys., Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1345 (Fed. Cir. 2008) (holding
14 method claims should not be “limited to performance of the steps in the order recited, unless
15 [they] explicitly or implicitly require[] a specific order”). Netflix, however, seeks to import an
16 order-of-steps limitation into the claim. Netflix’s construction is based on a false premise—i.e.,
17 that the later steps of the claim build on the results of the previous steps. This is not the case.
18 Each of the claim’s three steps can be performed independently.

19 The claimed “patterns” represent numeric values that describe relationships between an
20 index value (v) and a constant threshold value (N)—i.e., whether v is greater than or equal to N ;
21 the offset of v above N ; and a value of v above the offset. The numeric values (e.g., 5) and the
22 patterns that represent them (e.g., 111110) are related but distinct concepts. Accordingly, each
23 step of the claim involves two aspects: (1) determining a numeric value that describes a
24 relationship between v and N ; and (2) generating a pattern that represents that value. Because the
25 underlying values may be determined before the patterns are generated, the patterns do not have
26 to be generated in a particular order. It would be possible, for example, to determine all three
27

values before generating any patterns, and then generate the patterns all at once.

Further, it is also not necessary to determine the values in a particular order. Each can be determined independently based on v and N . The first limitation, for example, requires generating a first pattern in a first portion of a codeword, in response to an index value being at least as great as a threshold. If v is greater than or equal to N , the system can create a pattern in the first part of a codeword.

The second limitation requires generating a pattern in a second portion of the codeword, representing an offset of the index value above the threshold. Here again, all the system needs to do this are the values of v and N , because the offset can be determined by the formula $\lceil \log_2(v - (N - 2)) \rceil$, even if the first pattern has not been generated. *See* '663 Patent at 6:54–58; **Ex. G**: Declaration of Iain Richardson, Ph.D. (May 28, 2021) (Richardson Dec.), at 57.

The same is true for the third limitation, which requires generating a pattern in a third portion of the codeword, representing a value of the index value above the offset. Here, the value of the index value above the offset can be determined by the formula $v - (N - 2) - 2^{\lceil \log_2(v - (N - 2)) \rceil}$. '663 Patent at 54–61; Richardson Dec. at 57. And again, all that's necessary to perform this calculation are v and N ; the third pattern can be generated without generating the other two. In short, once the initial values v and N are known, each step of the claim can be carried out independently *and in any order* simply by substituting those values into the formulas cited above. Sequential ordering is not required by logic. Richardson Dec. at 57–58.

Nor is it required by grammar.⁴ The claim requires first, second, and third “patterns” of 1s and 0s to be placed within first, second, and third “portions” of a codeword. Although the portions “follow” one another in terms of their position within the codeword, the patterns of 1s and 0s that fill the portions are not necessarily generated in temporal order. The binary suffix, for

⁴ *Cf. Broadcom Corp. v. Sony Corp.*, No. SACV 16-1052 JVS, Claim-Construction Order, ECF No. 150, at (May 18, 2017) (finding that different language in system claims 11 and 21 of the '663 Patent required an order of steps).

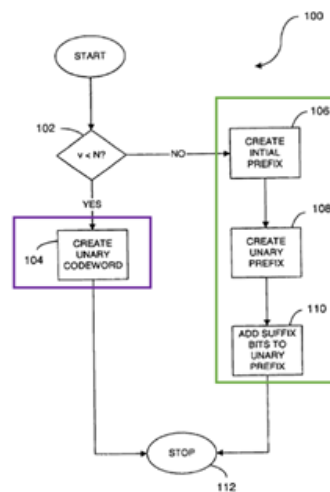
example, can be calculated before, after, or in parallel with the unary prefix and still generate the same codeword. Simply put, much like a book (or a legal brief) which has a beginning, middle, and end, codewords are read sequentially, but they do not have to be written in that same order.

ii. Claims 18 and 19

Netflix contends these dependent claims are indefinite. Netflix is wrong.

Independent claim 12 (from which claims 13, 18, and 19 depend) states that if an index value (e.g., $v = 18$) is “at least as great as a threshold” (e.g., $N = 16$), first, second, and third binarization patterns are generated to fill first, second, and third portions of a codeword. Other claims address what happens to the first, second, and third portions of the codeword if that condition is not met (e.g., $v < 16$). Claim 13 recites: “The method according to claim 12, further comprising the step of generating a fourth pattern in said first portion based on said index value in response to said index value being below said threshold.” ’663 Patent at 8:36–37.

Figure 4 of the patent (reproduced below with annotations) provides an exemplary embodiment of claims 12 and 13.



The green path in the figure provides an example of generating first, second, and third patterns when an index value v is not less than a threshold N . See ’663 Patent at 7:5–13; Richardson Dec. at 59–60. The purple path, provides an example of generating a fourth pattern when the index value is less than the threshold. See ’663 Patent at 6:66–7:4; Richardson Dec. at

60.

Referring again to the annotated version of Figure 6 of the patent, the blue, red, and yellow portions illustrate the first, second, and third portions of a codeword, and the numbers in white indicate the first, second, third, and fourth codeword patterns.

Index	Unary Prefix	exp-Golomb Suffix
0	0	
1	10	
2	110	
...		
15	1...10	
16	1...110	0
17	1...1110	1
18	1...11110	00
19	1...111110	01
20	1...1111110	10
21	1...11111110	11
22	1...111111110	000
23	1...1111111110	001
24	1...11111111110	010
25	1...111111111110	011
26	1...1111111111110	100
27	1...11111111111110	101
...		

Patterns 1, 2, and 3 are generated as recited in claim 12. Richardson Dec. at 53–55. Pattern 4 shows a “fourth pattern,” which is generated when the index value is less than the threshold, as recited in claim 13. *Id.* at 53–55, 60–61. Thus, either a *first* pattern or a *fourth* pattern is generated in the blue first portion of a codeword depending on how an index value compares to a threshold. *Id.* at 61. If $v \geq N$, the first pattern goes in the first portion; if $v < N$, the fourth pattern goes in the first portion. A person of ordinary skill in the art would understand the meaning of “generating a fourth pattern in said first portion based on said index value in response to said index value being below said threshold” and would be reasonably certain about its scope. *Id.* at 60–62. Claim 13 is not indefinite.

Claims 18 and 19 also depend from claim 12, and they address how to generate second and third portions of the codeword that either comprise a second and third pattern or are void, depending on the index value. They recite: “The method according to claim 12, wherein said [second/third] portion is void in response to said index value being below said threshold.” Recall

1 that claim 12 recites generating “a second pattern in a second portion . . . *representing an offset*
 2 *of said index value above said threshold*” and “a third pattern in a third portion . . . *representing*
 3 *a value of said index value above said offset*.”

4 A person of ordinary skill in the art would understand that an “offset” is a measure of the
 5 relative distance (or displacement) from a starting point. Richardson Dec. at 63. If the index
 6 value is below the threshold, there is no offset above the threshold and thus, no need to generate
 7 patterns in portions 2 and 3 of the codeword. *Id.* at 64. Claims 18 and 19 therefore clarify that the
 8 second and third portions are “void in response to said index value being below said threshold.”
 9 *Id.*

10 If a portion is void, it does not contribute any bins to the generated codeword. Accordingly,
 11 a person of ordinary skill in the art would understand that “wherein said [second/third] portion is
 12 void in response to said index value being below said threshold” means “wherein said
 13 [second/third] portion does not contribute any bins to the codeword in response to said index
 14 value being below said threshold.” *Id.* Claims 18 and 19 are not indefinite.

15 **b. ’387 Patent**

16 Claim 3 of the ’387 Patent recites:

17 A binarization system comprising:

18 means for determining if a code symbol index value is less than a threshold value;

19 means for constructing a codeword using a unary binarization if said code symbol

20 index value is less than a threshold value; and

21 means for constructing a codeword using a exp-Golomb binarization if said code

22 symbol index value is [not] less than a threshold value.

23 The parties agree that each of the three elements of this claim are means-plus-function
 24
 25
 26
 27

1 limitations governed by 35 U.S.C. § 112 ¶ 6 and that their functions are stated by the claim.⁵ The
 2 parties further agree that: (i) the structure in which these limitations are performed is binarization
 3 module 62, shown in Figure 2 of the patent; and (ii) the specification provides a general
 4 algorithm for performing them at Figure 4, as well as a more detailed algorithm at column 7,
 5 lines 40–60. The dispute is whether the claim is limited to the more detailed algorithm. It is not.

6 **i. Means-Plus-Function Claiming**

7 Under 35 U.S.C. § 112 ¶ 6, an element in a patent claim may be drafted as a generic
 8 “means” for performing a function without including a description of the physical structure by
 9 which the element performs the function. Construction of a means-plus-function limitation
 10 involves two steps. “First, the court must determine the claimed function. Second, the court must
 11 identify the corresponding structure in the written description of the patent that performs the
 12 function.” *AllVoice Computing PLC v. Nuance Commc’ns, Inc.*, 504 F.3d 1236, 1240 (Fed. Cir.
 13 2007) (quotation marks omitted). When the disclosed structure of a means-plus-function
 14 limitation is a computer or microprocessor programmed to carry out a function, the claim and the
 15 specification must disclose an algorithm that makes the computer perform that particular
 16 function. *See Aristocrat Techs. Austl. Pty Ltd. v. Int’l Game Tech.*, 521 F.3d 1328, 1333 (Fed.
 17 Cir. 2008).

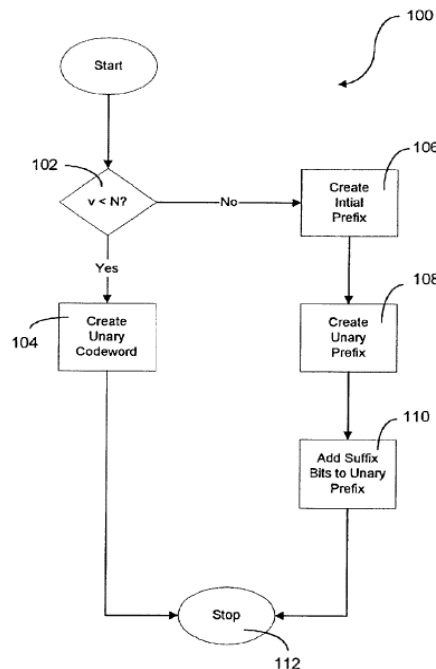
18 “Precedent and practice permit a patentee to express [an] algorithm in any understandable
 19 terms including as a mathematical formula, in prose, or as a flow chart, or in any other manner
 20 that provides sufficient structure.” *Typhoon Touch Techs., Inc. v. Dell, Inc.*, 659 F.3d 1376, 1385
 21 (Fed. Cir. 2011) (quotation marks omitted). If a specification discloses multiple structures,
 22 including algorithms, that may correspond to the function of a means-plus-function limitation,
 23 the limitation is properly construed as covering any of the structures and their equivalents.

25 ⁵ The ’387 Patent application was filed on July 10, 2002, so the pre-AIA version of the
 26 statute applies.

Serrano v. Telular Corp., 111 F.3d 1578, 1583 (Fed. Cir. 1997) (refusing to endorse a construction that included only one of multiple structures disclosed in the specification).

ii. Netflix’s construction improperly reads out several of the disclosed algorithms.

Figure 4 of the ’387 Patent “illustrates the steps of the invention,” ’387 Patent at 7:62–63, and the parties agree that it discloses algorithms for each of the limitations at issue.



The specification explains that “[p]rocess 100 begins at step 102 where a test is made to determine if the value of the code symbol index is less than the value of the threshold.” *Id.* at 7:63–66. A person of ordinary skill in the art would understand that the condition operation shown at 102 is a fundamental logic operation that is an essential building block in both software and hardware. Richardson Dec. at 24; see *AllVoice Computing*, 504 F.3d at 1245 (“[A]lgorithms in the specification need only disclose adequate defining structure to render the bounds of the claim understandable to one of ordinary skill in the art.”).

If the index value is less than the threshold value, “processing moves to step 104 were [sic]

a unary codeword is constructed comprising a series of v 1's terminated with a 0," and ends at step 112. *Id.* at 7:66–8:1. If the index value is not less than the threshold value:

- a) "processing moves to step 106, where an initial prefix of N 1's is created," *id.* at 8:2–3;
- b) "[p]rocessing then moves to step 108 where the most significant bits of the value $v-(N-2)$ are extracted and converted to a unary representation [and] appended to the initial prefix to create a unary prefix," *id.* at 8:3–7; and
- c) "[p]rocess 100 then moves to step 110 where the binary representation of the least significant bits of the value of $v-(N-2)$ are appended to the unary prefix to create the codeword," *id.* at 8:7–10.

The parties agree that this represents an algorithm for: (i) determining if a code symbol index value is less than a threshold value (step 102); (ii) constructing a codeword using a unary binarization if the code symbol index value is less than a threshold value (step 104); and (iii) constructing a codeword using an exp-Golomb binarization if the code symbol value is not less than a threshold value (steps 106–110). They also agree that the formula described at column 7, lines 40–60, is a detailed algorithm that describes one way to carry out the steps shown in Figure 4. Netflix, however, contends that because this second algorithm provides more detail, it is the only algorithm that satisfies § 112 ¶ 6. Netflix is mistaken.

The Federal Circuit has consistently held that an algorithm does "not need to include every possible implementation of the function, so long as it [is] linked to and encompass[es] the function." *Univ. of Pittsburgh of Commonwealth Sys. of Higher Educ. v. Varian Med. Sys., Inc.*, 561 F. App'x 934, 941 (Fed. Cir. 2014) (limiting the algorithm to its broadest description and declining to include more detailed iterations). *Harris v. Ericsson Inc.*, 417 F.3d 1241 (Fed. Cir. 2005), is instructive in this regard. In *Harris*, the patent disclosed a "time domain processing means," which the Federal Circuit construed as a microprocessor programmed to perform an algorithm. *Id.* at 1253–55. The court found that the algorithm was a two-step algorithm, even

though the specification illustrated and described—in written words in multiple portions of the specification—a series of calculations to be performed by a “support processor” and a “fast array processor.” *Id.* at 1254–55. The Federal Circuit condensed the multiple calculations into a two-step algorithm, with “each processor perform[ing] one of the steps.” *Id.* at 1254.

Here, too, there is no reason to limit the structure to the algorithm disclosed at column 7, lines 40–60. In fact, the patent explains that the detailed algorithm is just one way to process the general algorithm shown in Figure 4. Specifically, it states that the detailed algorithm is a “description of the method for constructing [the] hybrid bin[a]rization” illustrated in Tables 3 and 4, which “demonstrate particular instances” of the code that are “particularly appropriate” for certain magnitudes of the MPEG-AVC/H.264 standard. *Id.* at 6:43–7:35, 7:40–41.

In contrast, the specification states that the general algorithm disclosed in Figure 4 “illustrates the steps of *the present invention*.” *Id.* at 7:62–63 (emphasis added). Further, the patentee stated that the claimed binarization scheme is not restricted to the MPEG H.264 standard, and that “[a]s one skilled in the art can appreciate any system utilizing BAC [binary arithmetic coders] may make use of the present invention to improve binarization.” *Id.* at 8:11–16. Thus, the specification clearly evidences that the invention is not limited to the detailed algorithm described at column 7, lines 40–60. *See Serrano*, 111 F.3d at 1583 (holding structure was not limited to the physical circuitry of the preferred embodiment where the specification stated that “it should be recognized to those of ordinary skill in the art that a microprocessor-based system could also be used”).

Finally, the detailed algorithm described at column 7, lines 40–60, is not the only implementation of the algorithm shown in the Figure 4. As explained in detail by Dr. Richardson, Tables 1, 3, and 4 and original claims 1 and 3 also disclose algorithms for “constructing a codeword *using a unary binarization* if said code symbol index value *is less than a threshold value*.” Richardson Dec. at 29–36; *see* ’387 Patent at 6:43–7:15. And Tables 1, 2, 3, and 4 and original claims 1 and 3 disclose algorithms for “constructing a codeword *using exp-*

1 *Golomb binarization* if said code symbol index value *is not less than a threshold value.*”

2 Richardson Dec. at 36–51; *see* ’387 Patent at 6:43–7:15. Netflix’s construction improperly reads
3 each of these algorithms, as well as the algorithm shown in Figure 4, out of the claim. *See Guzik*
4 *Tech. Enters., Inc. v. W. Digital Corp.*, No. 11-cv-03786-PSG, 2013 WL 3934892, at *7 (N.D.
5 Cal. July 19, 2013) (“[I]f the specification identifies more than one possible structure, the
6 construction should not be limited to a preferred embodiment.”) (citing *Micro Chem., Inc. v.*
7 *Great Plains Chem. Co.*, 194 F.3d 1250, 1258 (Fed. Cir. 1999)).

8 In sum, Figure 4 of the ’387 Patent, standing alone, discloses algorithms that “disclose
9 adequate defining structure to render the bounds of the claim understandable to one of ordinary
10 skill in the art.” *See Allvoice Computing*, 504 F.3d at 1245; Richardson Dec. at 24, 27–29, 38–
11 41. No more is necessary. *Id.* Netflix’s attempt to limit the algorithm to just one of the more
12 detailed algorithms described in the specification should be rejected.⁶

13 The terms at issue should be construed broadly enough to cover the algorithms disclosed in
14 Figure 4; Tables 1, 2, 3, and 4; column 6, line 26, through column 8, line 23; and original claims
15 1 and 3. *See Serrano*, 111 F.3d at 1583.

16 **C. ’283 Patent**

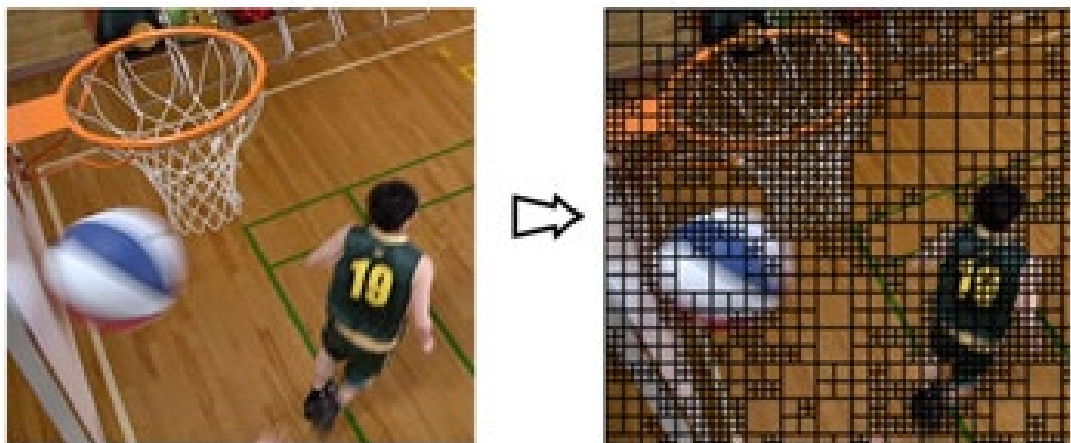
17 **1. Technical Background**

18 The ’283 Patent is directed to a process for compressing data needed to transmit video. On
19 a high level, compression takes advantage of the fact that many images in a video stream do not

21 ⁶ In *Broadcom Corp. v. Sony Corp.*, the court held that the detailed algorithm disclosed at
22 column 7, lines 40–60, was the corresponding structure based on a finding that “Figure 4 is a
23 *general description* of the same algorithm that the patent describes in column 7:40–60.” Claim-
24 Construction Order at 17 (emphasis in original). The court, however, did not cite or distinguish
25 the Federal Circuit cases holding that constructions should not be limited to detailed iterations of
26 a structure otherwise broadly described in the specification. *See Allvoice Computing*, 504 F.3d. at
27 1245; *Varian Med. Sys.*, 561 F. App’x at 941; *Harris*, 417 F.3d at 1253–55.

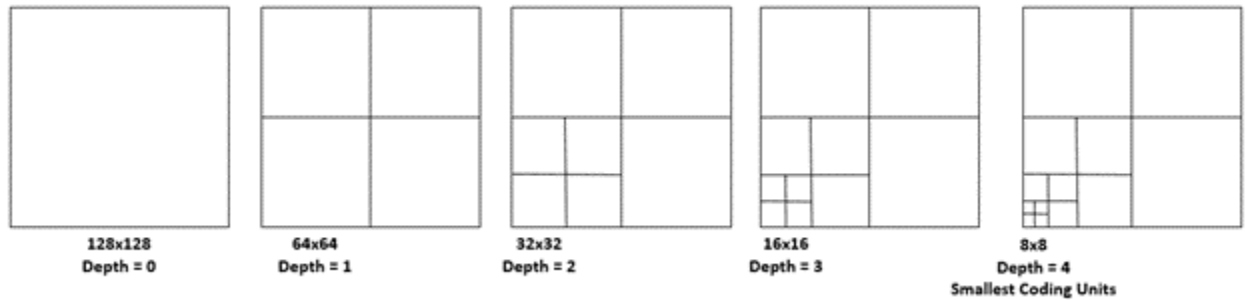
change from frame to frame, and if they do change, the differences from one frame to the next are often minor. Transmitting the differences between frames, instead of entire frames, results in considerable savings in data transmission.

To accomplish this, each frame in a video is divided into coding units (CUs), which provide reference areas for comparing the pictures. As the patent explains, coding units “can include a number of pixels typically arranged in a square shape.” ’283 Patent at 6:18–19. The figure below provides an example of a picture divided into coding units of various sizes.

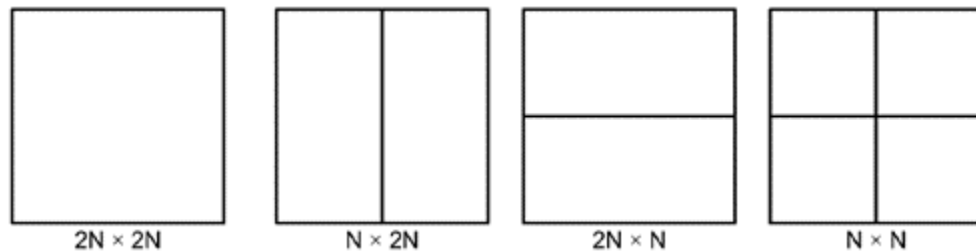


As the patent explains, a coding unit can be subdivided into four smaller squares, each of which can be further divided into four more squares, and this recursive nesting can continue to a certain depth (i.e., a certain number of times). *Id.* at Fig. 11, 16:41–17:18; Richardson Dec. at 10–13. Thus, as illustrated in the figure below, a 128×128 pixel coding unit can be divided into four 64×64 pixel coding units, each of which can, in turn, be divided into four 32×32 coding units, and so on to the depth of 4, at which point the coding units are 8×8 pixels. In this example, the depth level is limited to 4, so no more division can occur.⁷ The patent refers to coding units formed at the lowest depth level as smallest coding units (SCUs). *Id.* at 16:58–62.

⁷ A system can have different depth limits; the 4-depth limit is just one embodiment of the recursive nesting described in the patent. In any event, the CUs at the lowest depth are smallest coding units.



Each coding unit in a video frame contains prediction units (PUs), which are used to describe changes within the video frames. Prediction units can be arranged within a coding unit in patterns, called PU partition modes, that vary based on the relative size of the coding unit. *Id.* at 15:32–35. A smallest coding unit, for example, can be partitioned in several ways. Figure 12 of the patent, reproduced below, shows four possible PU partition modes for a size $2N \times 2N$ smallest coding unit. *Id.* at Fig. 12, 17:19–23.



The partition mode on the left comprises a single prediction unit that occupies the entire coding unit. The next two modes each have a pair of partitions, $N \times 2N$ and $2N \times N$. And the fourth has four $N \times N$ partitions. Prediction units in a smallest coding unit can be arranged in at least any of these configurations. *Id.* at 17:19–23.

For coding units other than a smallest coding unit—e.g., coding units formed at depth levels 0–3 in the example above—the inventors recognized that the amount of data needed to transmit video can be significantly decreased by eliminating the possibility of partitioning into four $N \times N$ prediction units. *Id.* at 18:34–54; Richardson Dec. at 17. The invention exploits this by using two codebooks to set partition modes. One codebook sets partition modes for smallest coding units and another sets them for other coding units. '283 Patent at 18:34–54; Richardson

Dec. at 17. $N \times N$ partition modes can be set only by the codebook for smallest coding units. When a coding unit is not a smallest coding unit, the invention uses a codebook that cannot create $N \times N$ partition modes. '283 Patent at 18:34–54; Richardson Dec. at 17. Doing this reduces coding complexity and results in as much as a 33% savings in codeword size for certain PU partition modes. Richardson Dec. at 17.

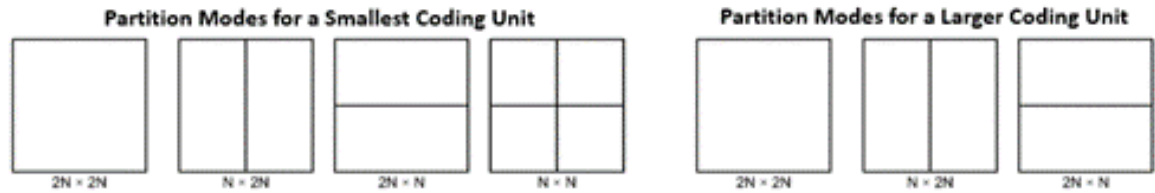
2. Disputed Term—“wherein the second syntax element specifies the PU partition mode for the selected CU, wherein the PU partition mode is based on a size $N \times N$ PU when the selected CU is a smallest CU (SCU) of the plurality of CUs and is based on a different size PU than the size $N \times N$ PU when the selected CU is another CU than the SCU of the plurality of CUs”

The issue is whether this term is indefinite. It is not. Viewed in light of the specification, the term describes a system that uses at least two codebooks: one sets partition modes on smallest coding units, and another sets partition modes on coding units other than smallest coding units.

The term begins by stating that a syntax element, which is an agreed term,⁸ “specifies the PU partition mode for [a] selected coding unit.” It then distinguishes the way partition modes are specified based on whether the selected coding unit can be partitioned into $N \times N$ prediction units. If the selected coding unit is “a smallest CU,” the partition mode is “based on a size $N \times N$ PU.” Otherwise, the partition mode is “based on a different size PU than the size $N \times N$ PU.” A person of ordinary skill in the art would understand that this means a codebook that can set $N \times N$ partition modes is used for a smallest coding unit, and a codebook that cannot set $N \times N$ partition modes is used for a larger coding unit. Richardson Dec. at 18. Thus, for the embodiment shown

⁸ The parties agree that “syntax unit” should be construed to mean “a value that is encoded to form part of the output bitstream.”

in Figure 12, all four partition modes would be available for a smallest coding unit, but only the first three for a larger coding unit.



In short, the term “inform[s] a person of ordinary skill about the scope of the invention with reasonable certainty.” See *Tinnus Enters., LLC v. Telebrands Corp.*, 733 F. App’x 1011, 1020 (Fed. Cir. 2018). It is not indefinite.

D. ’138 Patent

1. Technical Background

The ’138 Patent is directed to problems associated with distributed computing systems. ’138 Patent at 1:11–12. As the patent explains, an enterprise environment—such as a large business organization—often includes several business groups, and each group may have competing and variable computing needs that require connecting separate, independent computing devices to each other on the network. *Id.* at 1:16–33. The patented invention solves the problem of managing such a diversity of competing and variable computing requirements with an infrastructure management facility (IMF). *See id.* at 8:32–36. The IMF uses virtual machine managers to generate virtual machines and other “container services” that provide application services independent of the type of computing infrastructure. *See id.* at 9:59–10:5. The virtual machines, provided by the virtual-machine managers, appear on the network as available resources as if they were independent computing resources that can be accessed by various groups and utilized to suit their highly-diverse and specialized computing needs. *See id.* at 35:36–47.

1 **2. Disputed Terms**

2 **a. “virtual machine manager”**

3 Representative claim 1 of the ’138 Patent recites, in pertinent part (disputed term in bold):

4 A distributed computing system comprising . . . a plurality of image instances
5 of a **virtual machine manager** that is executable on a plurality of application
6 nodes, wherein when executed on the applications [sic] nodes, the image
7 instances of the **virtual machine manager** provide a plurality of virtual
8 machines, each of the plurality of virtual machine operable to provide an
9 environment that emulates a computer platform.

10 Broadcom contends the term “virtual machine manager” needs no construction because its
11 meaning is readily understandable to a lay jury when viewed in the context of the patent. Netflix
12 contends it should be construed to mean a “software program that manages one or more virtual
13 machines.” The issue is whether the claim requires that the virtual machine manager must
14 actually engage in managing one or more virtual machines, as Netflix contends. It does not. It
15 requires only that the virtual machine manager must be *able to* manage the virtual machines
16 “when executed.” ’138 Patent at claim 1. The patentee made this clear by expressly defining
17 “virtual machine manager” as “a software program that is *capable of* managing one or more
18 virtual machines,” not a program that *actually* does so. *Id.* at 32:60–62 (emphasis added).

19 As the Federal Circuit held in *Chrimar Holding Co., LLC v. ALE USA Inc.*, 732 F. App’x
20 876 (Fed. Cir. 2018), as amended (June 1, 2018), “it is hardly uncommon for an apparatus or
21 system claim, as a claim to a product rather than a process (or a forbidden mix), to be directed to
22 capability, instead of actual operation.” *Id.* at 885. And “to infringe a claim that recites capability
23 and not actual operation, an accused device need only be capable of operating in the described
24 mode.” *Finjan, Inc. v. Secure Computing Corp.*, 626 F.3d 1197, 1204 (Fed. Cir. 2010) (citation
25 and quotation marks omitted). Netflix’s attempt to import an actual-management limitation into
26 the claim should be rejected.

1 **b. “rules engine”**

2 Representative claim 9 recites: “The distributed computing system of claim 1, wherein the
3 control node further comprises one or more *rules engines* that provide autonomic deployment of
4 the software applications to the virtual machines in accordance with a set of one or more rules.”
5 *Id.* at claim 9 (emphasis added). Broadcom contends “rules engine” does not need to be
6 construed because its meaning is readily understandable to a lay jury when viewed in the context
7 of the patent. According to Netflix, a rules engine is “software that performs logical reasoning
8 using ‘if/then’ rules.”

9 Netflix imports limitations into the claim from an embodiment with rules engines that “*may*
10 *be* viewed as software that performs logical reasoning using knowledge encoded in high-level
11 condition-action rules.” *Id.* at 22:56–58 (emphasis added). Netflix’s construction is flawed
12 because the language in the specification that Netflix relies on does not demonstrate a clear
13 intention to limit the claim scope. *See Innova/Pure Water, Inc. v. Safari Water Filtration Sys.,*
14 *Inc.*, 381 F.3d 1111, 1117 (Fed. Cir. 2004) (“[E]ven where a patent describes only a single
15 embodiment, claims will not be read restrictively unless the patentee has demonstrated a clear
16 intention to limit the claim scope.” (quotation marks omitted)). On the contrary, the specification
17 uses permissive language and merely states how rules engines “may be viewed.” That they *may*
18 *be* viewed in a certain way does not mean *must be* viewed in that way.

19 Netflix’s construction should be rejected. *See Network, LLC v. Centraal Corp.*, 242 F.3d
20 1347, 1358 (Fed. Cir. 2001) (declining to limit claim term where the specification used
21 permissive language); *Mount Hamilton Partners, LLC v. Groupon, Inc.*, No. C 12-1700 SI, 2014
22 WL 1047408, at *7 (N.D. Cal. Mar. 14, 2014) (same).

1 Dated: July 2, 2021

THOMPSON & KNIGHT LLP

2 By: *Bruce S. Sostek*

Bruce S. Sostek

3 Attorney for Plaintiffs

4 BROADCOM CORPORATION, AVAGO

5 TECHNOLOGIES INTERNATIONAL

6 SALES PTE. LIMITED

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28