

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION**

PROFESSOR MASAHIRO IIDA,

Plaintiff,

v.

INTEL CORPORATION,

Defendant.

Case No. 6:22-cv-00662-ADA

Jury Trial Demanded

INTEL'S OPENING CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

	<u>Page</u>
I. Introduction.....	1
II. Background on the '737 Patent.....	2
A. Digital Circuits.....	2
B. Look Up Tables.....	4
C. Programmable Logic Circuit Devices.....	7
D. The Alleged Invention	8
III. Construction of the Disputed Claim Terms	10
A. “look up table of M inputs and N outputs” (claims 1, 7, 13, and all claims that depend therefrom).....	10
B. “LUT units” (all asserted claims)	13
IV. Conclusion	19

TABLE OF AUTHORITIES**PAGE****FEDERAL CASES**

<i>In re Fought</i> , 941 F.3d 1175 (Fed. Cir. 2019).....	10
<i>Geneva Pharms., Inc. v. GlaxoSmithKline PLC</i> , 349 F.3d 1373 (Fed. Cir. 2003).....	1, 13, 15
<i>Halliburton Energy Servs., Inc. v. M-I LLC</i> , 514 F.3d 1244 (Fed. Cir. 2008).....	1, 15
<i>Hill-Rom Services, Inc. v. Stryker Corp.</i> , 755 F.3d 1367 (Fed. Cir. 2014).....	19
<i>Kyocera Senco Indus. Tools Inc. v. Int’l Trade Comm’n</i> , 22 F.4th 1369 (Fed. Cir. 2022)	11
<i>Nautilus, Inc. v. Biosig Instruments, Inc.</i> , 572 U.S. 898 (2014).....	13, 17
<i>O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.</i> , 521 F.3d 1351 (Fed. Cir. 2008).....	1, 10
<i>Pac. Coast Bldg. Prod., Inc. v. CertainTeed Gypsum, Inc.</i> , 816 F. App’x 454 (Fed. Cir. 2020)	18
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	13, 16
<i>Saso Golf, Inc. v. Nike, Inc.</i> , 843 F. App’x 291 (Fed. Cir. 2021)	18
<i>Teva Pharms. USA, Inc. v. Sandoz, Inc.</i> , 789 F.3d 1335 (Fed. Cir. 2015).....	18
<i>Thorner v. Sony Computer Ent. Am. LLC</i> , 669 F.3d 1362 (Fed. Cir. 2012).....	14

TABLE OF EXHIBITS

The following exhibits are attached to the Declaration of Heather E. Takahashi and also referenced in the Declaration of Dr. Stephen W. Melvin Regarding Claim Construction (“Melvin Decl.”), submitted concurrently herewith.

Exhibit	Description
1	U.S. Patent No. 6,812,737
2	<i>Principles and Structures of FPGAs</i> (Hideharu Amano, ed., Springer 2018) (97300DOC000193097) (“Amano”)
3	Satwant Singh et al., <i>The Effect of Logic Block Architecture on FPGA Performance</i> , IEEE Journal of Solid-State Circuits, vol. 27, no. 3, 281-87 (Mar. 1992) (97300DOC000182007) (“Singh – Effect”)
4	Stephen D. Brown et al., <i>Field-Programmable Gate Arrays</i> (1992) (97300DOC000181652) (“Brown – FPGAs”)
5	Narasimha B. Bhat, Novel Techniques for High Performance Field Programmable Logic Devices, Technical Report no. UCB/ERL M93/80, Electronic Research Laboratory, University of California, Berkeley (Nov. 1993), at http://www2.eecs.berkeley.edu/Pubs/TechRpts/1993/ERL-93-80.pdf (97300DOC000182060) (“Bhat - Novel Techniques”)
6	Japanese Patent Application Publication No. H09-121153A to Fujii (certified English translation) (97300DOC000178646) (“Fujii”)
7	Jason Cong et al., <i>Technology Mapping For k/m-Macrocell Based FPGAs</i> , Proceedings of the 2000 ACM/SIGDA 8th Int’l Symposium on Field Programmable Gate Arrays, 51-59 (Feb. 2000) (97300DOC000190147) (“Cong - Technology Mapping”)
8	Stephen M. Trimberger et al., <i>Field-Programmable Gate Array Technology</i> (1994) (97300DOC000183159) (“Trimberger - FPGA Technology”)
9	Stephen Brown & Jonathan Rose, <i>Architecture of FPGAs and CPLDs: A Tutorial</i> , IEEE Design and Test of Computers, vol. 13, no. 2, 42-57 (Dec. 1996) (97300DOC000184471) (“Brown & Rose”)
10	U.S. Patent No. 5,905,385 to Sharpe-Geisler et al. (97300DOC000180179) (“Sharpe”)
11	Jonathan Rose et al., <i>Architecture of Field-Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Efficiency</i> , IEEE Journal of Solid-State Circuits, vol. 25, no. 5, 1217-25 (Oct. 1990) (97300DOC000181219) (“Rose”)
12	Robert J. Francis, <i>A Tutorial on Logic Synthesis for Lookup-Table Based FPGA</i> , IEEE Int’l Conference Aided Design, Institute of Electrical Engineers Inc., 40-47 (Nov. 1992) (97300DOC000187647) (“Francis”)

Exhibit	Description
13	Japanese Patent Application Publication No. H05-252024A to Yamada (certified English translation) (97300DOC000180730) (“Yamada I”)
14	Randy H. Katz, <i>Contemporary Logic Design</i> (1993) (P-00377)
15	<i>IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries</i> (1990) (P-00398)
16	<i>IEEE The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition</i> (2000) (P-00405)
17	U.S. Patent No. 6,331,790 to Or-Bach (“Or-Bach III”) (97300DOC000179623)
18	Excerpt from Thomas H. Cormen et al., <i>Introduction to Algorithms</i> (3d ed. 2009) at 253–280 (Chapter 11 titled “Hash Tables”) (“Cormen”).

I. INTRODUCTION

The parties dispute the construction of two terms in U.S. Patent No. 6,812,737 (the '737 patent). Because the parties raise “actual dispute[s] regarding the proper scope” of these terms, the Court “must resolve” those disputes by construing the claim terms. *O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1360 (Fed. Cir. 2008)

First, the parties dispute whether the phrase “look up table of M inputs and N outputs” requires construction. Intel proposes that the Court should construe that term in accordance with the patentee’s clear lexicography. See '737 patent at 1:25–28, 1:46–54 (Exhibit 1). Iida’s proposal—no construction at all—would ignore that lexicography and invite juror confusion about the scope of the claimed invention.

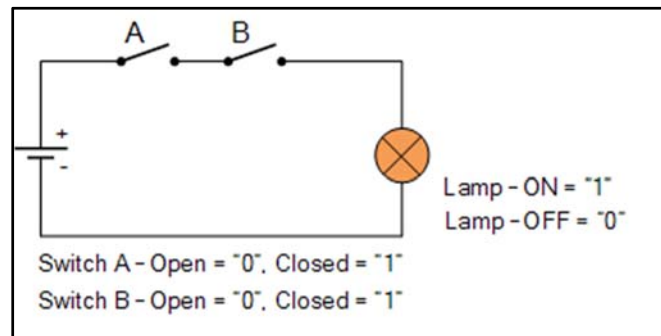
Second, the parties dispute whether the phrase “LUT units” renders the asserted claims indefinite. As Intel explains below, the claimed “LUT units” are smaller look up tables (i.e., LUTs) that, when taken together, comprise the larger “look up table of M inputs and N outputs.” The application of this definition, which is based on the plain and ordinary meaning of “units,” renders the asserted claims indefinite because “a person of ordinary skill in the art cannot **translate** the definition into meaningfully precise claim scope.” *Halliburton Energy Servs., Inc. v. M-I LLC*, 514 F.3d 1244, 1251 (Fed. Cir. 2008) (emphasis added). Identifying the “LUT units” in a given look up table is an arbitrary box-drawing exercise—nothing in the claims explains **how** to identify the LUT units in a given look up table. If you look at the device one way, identifying certain components as the “LUT units,” that device infringes. But if you look at the **same** device in another, **perfectly acceptable** way, it does not. Thus, one device can **both** infringe and not infringe the claims. “That is the epitome of indefiniteness.” *Geneva Pharms., Inc. v. GlaxoSmithKline PLC*, 349 F.3d 1373, 1384 (Fed. Cir. 2003). Iida’s counterproposal—plain and ordinary meaning—does not affect that conclusion.

II. BACKGROUND ON THE '737 PATENT

The '737 patent relates to programmable logic circuit devices (PLDs) and, more specifically, to the look up tables used to implement logic circuits in those devices. *See, e.g.*, '737 patent at Abstract. To understand look up tables and the role they play in PLDs, it is helpful to start with a few basic principles of digital circuits.

A. Digital Circuits

Digital circuits are the foundation of modern computing. Unlike analog circuits, which vary *continuously* over a voltage range, digital circuits normally present at *discrete* voltages. Most digital circuits are built to handle two different voltages: one labeled “HIGH” voltage (on, 1, or TRUE) and another labeled “LOW” voltage (off, 0, or FALSE). In the simple example below, binary inputs A and B represent the state of two switches (open or closed), and a binary output (on or off) represent the state of the Lamp (on or off):



When A and B are HIGH (1), the two switches are closed, current flows through circuit, and the Lamp turns on. Conversely, if either A or B (or both) are LOW (0), current does not flow through the circuit and the lamp does not turn on. *See* Melvin Decl. ¶ 32.

Digital circuits are often represented in a “truth table,” which shows the relationship between that circuit’s inputs and outputs. The circuit above, for example, has a four-row truth table:

Inputs		Output
Switch A	Switch B	Lamp
0	0	0
0	1	0
1	0	0
1	1	1

In general, binary circuits with K inputs have a complete (i.e., representing every input pattern) truth table with 2^K rows. *Id.* Thus, a two-input circuit has a complete truth table with 2^2 rows (i.e., 4 rows), a three-input circuit has a complete truth table with 2^3 rows (i.e., 8 rows), and so on. These truth tables describe the output for **every possible** input pattern the circuit can implement. See Melvin Decl. ¶ 33.

To construct digital circuits, circuit designers use basic building blocks called “logic gates.” Logic gates represent the fundamental Boolean logic functions—AND, OR, and NOT—and combinations of those functions. Engineers can combine these logic gates to form very complicated logic functions. But even complex logic functions can be represented in a truth table, like this one for a three-input majority-vote function:

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

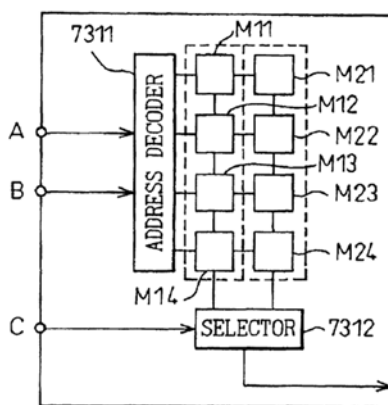
(a) Truth table

Ex. 2 at 97300DOC000193138 & FIG. 2.9. As the truth table shows, the function (f) is true (1) if, and only if, a majority of its inputs (A, B, and C) are true (1). This truth table shows the resulting output for **every** possible input pattern in this 3-input function. Thus, the truth table completely describes this more complex logic function. See Melvin Decl. ¶¶ 34–35.

B. Look Up Tables

Both simple and complex logic functions can be implemented in hardware using look up tables (often shortened to “LUTs”). A look up tables is a “digital memory” that can implement a logic function by storing the values for that function’s truth table in its memory. A digital memory contains an array of memory cells, each of which can store a single binary value (i.e., a 0 or a 1, or a “bit”). Those binary values can be read (i.e., output) from the memory using an “address,” which specifies the location (row and column) of the bit value(s) that should be output. See Melvin Decl. ¶ 36.

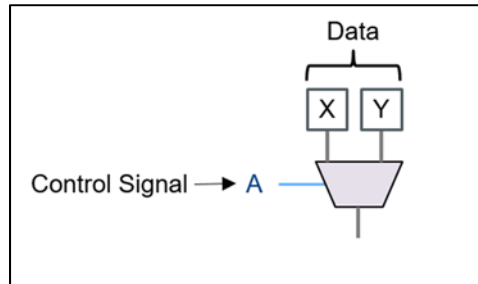
For example, consider the following digital memory described in the ’737 patent:



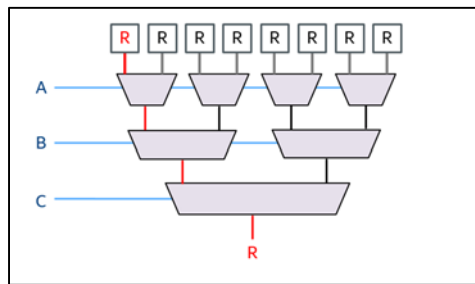
’737 patent at FIG. 7. This memory contains a four-by-two array of eight bits (M11–M14 and M21–M24) that are addressed using three binary values (A, B, and C). Together, the values for A and B identify one of four rows using the “address decoder.” Then, the value for C identifies one of two columns using the “selector.” In this way, a specific combination of values for A, B, and C identifies exactly one of the eight memory cells in the array. See Melvin Decl. ¶ 37.

This is not, however, the only configuration that can be used to address a digital memory. In general, an address decoder is used to select the rows, and a selector (or multiplexer) is used to select the columns. And if an array has many columns, those columns can be addressed using a

tree of multiplexers. A multiplexer is a circuit component (often depicted as a trapezoid) that selects one output from multiple data inputs based on binary control signals:

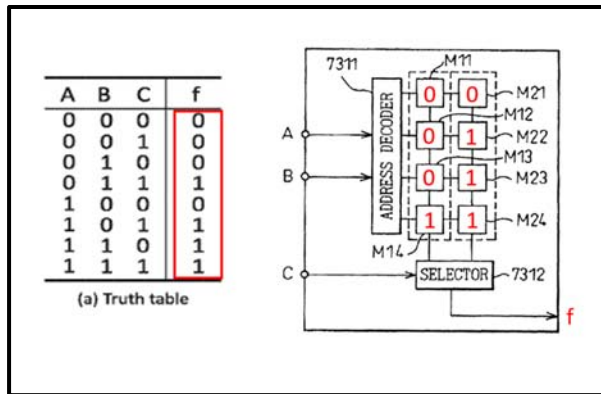


In this example, if A is HIGH (1) the multiplexer selects X, and if A is LOW (0) the multiplexer selects Y. By stacking multiplexers into a tree, one can a multiple-input multiplexer that can address a memory:



This seven-multiplexer tree implements an 8-1 multiplexer for addressing an 8-bit memory. Each level of multiplexers is controlled by a control signal (A, B, or C), which identify the single memory bit that should be output. In this example, if A, B, and C are HIGH (1), the left-most memory bit is output (shown in red). See Melvin Decl. ¶ 38.

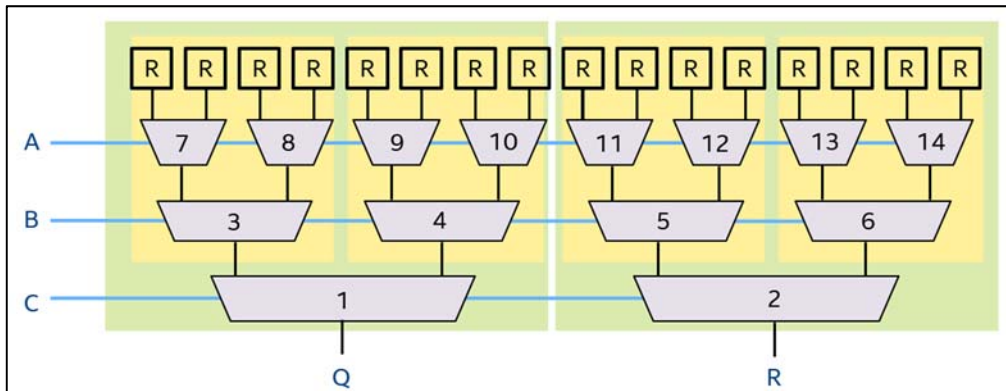
Look up tables implement logic circuits by storing the values of the circuit's truth tables in memory cells. The look up table's memory stores each possible output for a circuit and uses input signals to address that memory. As an example, consider how the memory above (*supra*, p. 4) could be used to implement the three-input function discussed above (*supra*, p. 3):



The outputs for the truth table (column f) are programmed into the memory array of the look up table (from left to right, top to bottom). Then, the values for A, B, and C are used to “look up” the correct location in the memory when the device is used. In this way, the look up table implements the majority-vote function’s relationship between inputs and outputs by storing the values of its truth table. See Melvin Decl. ¶ 39.

In general, a look up table with 2^K bits can implement *any* K -input 1-output logic function. Every possible K -input logic function can be completely represented with a truth table of exactly 2^K rows. See *supra*, p. 3. Thus, with 2^K memory bits, a look up table can store the values of one entire output column of a K -input truth table. In doing so, the look up table can store the truth table for *any* arbitrary logic function of its K inputs and one output. *E.g.*, Ex. 9 (Brown & Rose) at 97300DOC000184498; Ex. 4 (Brown – FPGAs) at 97300DOC000181665, -716; Ex. 6 (Fujii) ¶ [0005]; Ex. 10 (Sharpe) at 1:37–48. In other words, simply by reprogramming the look up table’s memory, the look up table can be used to compute any logic function of its K -inputs and one output. See Melvin Decl. ¶ 40.

Another feature of look up tables is that they often can be viewed as a collection of smaller look up tables. For example, consider this look up table with three inputs (and two outputs):



This single device may be viewed as being comprised of four smaller look up tables with the components shown inside the yellow boxes, each having *two* inputs (A and B), with the outputs from the two yellow look up tables on the left proceeding to output selector 1 (i.e., an output selector outside the look up table), and the outputs from the two yellow look up tables on the right proceeding to output selector 2 (also outside the look up table). Alternatively, this device may be viewed as having two (rather than four) smaller look up tables with the components shown in the green boxes, each having *three* inputs (A, B, and C), with *no* selectors connected to outputs Q and R from the look up tables. Thus, how one views the smaller look up tables within this device can impact whether the overall device contains (or does not contain) selectors that lie outside those smaller look up tables. See Melvin Decl. ¶ 41.

C. Programmable Logic Circuit Devices

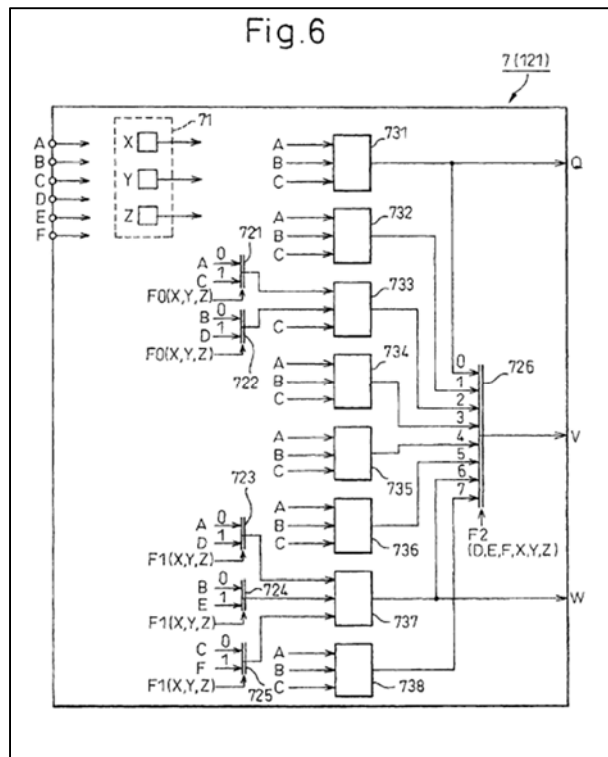
Look up tables are a key component of many PLDs. Unlike application-specific integrated circuits (ASICs), which are typically hardwired to perform a single function, PLDs can typically be configured (and reconfigured) by customers to perform a wide variety of desired functions. A customer can buy a PLD—like a field programmable gate array (FPGA)—off the shelf and

configure that device to implement logic circuits. PLDs are, in short, *field programmable*. See Melvin Decl. ¶ 42.

Look up tables are ideal for implementing this functionality. A user (with the help of software) can set the bit values of the memory cells as they want to implement a wide variety of functions. Then, if the user later wants to change the function implemented, a new set of memory bits can be determined. See Melvin Decl. ¶ 43.

D. The Alleged Invention

As noted, the '737 patent relates to PLDs and, more specifically, to look up tables (abbreviated "LUT" in the patent) used to implement logic circuits in those devices. See, e.g., '737 patent at Abstract. Figure 6 depicts one of the preferred embodiments of the alleged invention:



See also '737 patent at 5:24–61, 6:21–7:21 (describing FIG. 6). This figure shows a “6-input 3-output LUT 7 (121)” (i.e., the large box encompassing the other components) that contains “a mode changing memory 71, selectors 721~726, and 3-input 1-output LUT units (memory array)

731~738.” *Id.* at 5:25, 5:27–29. Most of the selectors (**721** through **725**) are “for selecting input signal[s]” to the “LUT units.” *Id.* at 5:33–34. But one selector (**726**) is “for selecting output signal[s]” from the “LUT units.” *Id.* at 5:55–56. “[B]y varying the values (X, Y, Z) of mode changing memory (71),” which can control the selectors 721 through 726, “the 6-input 3-output LUT can be set to different internal configurations as shown in FIGS. 3A to 3F.” *Id.* at 4:15–18; *see id.* at 4:18–47 (describing the six modes shown in FIGS. 3A to 3F); *id.* at 6:20–7:21 (describing how the “6-input 3-output LUT” in FIG. 6 can implement the six modes shown in FIGS. 3A to 3F); *see also* Melvin Decl. ¶ 44.

Iida claimed his alleged invention in three independent claims (claims 1, 7, and 13), all of which are asserted in this case. For purposes of the claim construction dispute presented here, claim 1 is representative of the independent claims:

A look up table of M inputs and N outputs, comprising:

a plurality of ***LUT units***; and

an internal configuration control circuit controlling an internal configuration of said plurality of ***LUT units***, wherein said internal configuration control circuit comprises

a plurality of selectors selecting I/O signals of said plurality of ***LUT units***, and

a selector control circuit having a memory, controlling said plurality of selectors in accordance with data stored in said memory, and defining the internal configuration of said plurality of ***LUT units***.

Id. at claim 1 (emphases added to disputed terms); *see also* claims 7, 13.

III. CONSTRUCTION OF THE DISPUTED CLAIM TERMS

A. “look up table of M inputs and N outputs” (claims 1, 7, 13, and all claims that depend therefrom)

Iida’s Proposed Construction	Intel’s Proposed Construction
Plain and ordinary meaning	“component of a programmable logic circuit device that is composed of memories and is capable of implementing an arbitrary logic circuit of M inputs on at least one of N outputs”

This term warrants construction. Intel’s proposed construction clarifies that the claimed “look up table of M inputs and N outputs” must be capable of implementing *any* arbitrary function of its M inputs and at least one of N outputs. As discussed below, this construction is required by the patent’s written description, which clearly defines a “look up table of M inputs and N outputs” as “an arbitrary logic circuit of M inputs and N outputs,” meaning the logic circuit can perform any function of its inputs. ’737 patent at 1:46–54; see Melvin Decl. ¶¶ 50–53. In contrast, Iida’s plain-and-ordinary-meaning construction would leave the jury without any meaningful guidance as to how to apply this term. Without construction, a lay juror might (mistakenly) conclude that any table of stored data is a “look up table of M inputs and N outputs.” To avoid juror confusion and *O2 Micro* issues, the Court should construe this term in accordance with the ’737 patent’s clear lexicography.¹

The ’737 patent expressly defines “a look up table of M inputs and N outputs.” It crafts that definition in two parts: (1) defining a “look up table” and (2) defining what it means for a look

¹ The claim term “a look up table of M inputs and N outputs” limits the scope of all independent claims because that phrase is relied upon for antecedent basis. See *In re Fought*, 941 F.3d 1175, 1178 (Fed. Cir. 2019). Iida has not contested this point.

up table to have “M inputs and N outputs.” Both definitions appear in the patent’s very first column:

[1] A look up table (LUT) as a **major component of a programmable logic circuit device** (PLD or FPGA) **is composed of memories**, and **is capable of implementing an arbitrary logic circuit**.

’737 patent at 1:25–28 (emphasis added);

[2] In a programmable logic circuit device, the number of bits M of input signal for the logic circuit that can be implemented corresponds to the number of bits of the address of memory composing the LUT, and the number of bits N of output signal corresponds to the number of output bits of the memory. Thus, a logic circuit that can be implemented in one LUT **is an arbitrary logic circuit of M inputs and N outputs**. Such a LUT **is** denoted in the present specification as M-input N-output LUT.

’737 patent at 1:46–54 (emphasis added). Taken together, these passages define what a “look up table of M inputs and N outputs” “**is**”—not simply what it “might be” or “could be.” *Cf. Kyocera Senco Indus. Tools Inc. v. Int’l Trade Comm’n*, 22 F.4th 1369, 1378 (Fed. Cir. 2022) (holding the patentees acted as their own lexicographers by defining “driven position” in the written description by stating “[t]his bottom position is also sometimes referred to herein as the ‘driven position’”). Indeed, the second passage makes clear that its definition applies to the entire “present specification.” That is clear definitional language, which binds the patentee.

These passages also make clear that the “look up table of M inputs and N outputs” is one that “is capable of implementing an arbitrary logic circuit” and in fact “is an arbitrary logic circuit of M inputs and N outputs.” ’737 patent at 1:25–28, 1:46–54. The claimed look up table, therefore, must be capable of being programmed to perform any logic function of its M inputs. Put differently, it must have sufficient memory—specifically, at least 2^M bits of memory—such that it is capable of implementing any logic circuit of M inputs with any combination of logic operating on those inputs on at least on output. *See* Melvin Decl. ¶¶ 50–53.

Intel’s proposed construction maps perfectly onto the express definition in the written description. In large part, that construction just copies Iida’s definitional language from the written description. *See* ’737 patent at 1:25–28, 1:46–54. Intel’s proposed construction also is consistent with all of the embodiments of a “look up table of M inputs and N outputs” in the patent. *See* Melvin Decl. ¶¶ 52–53. For example, the ’737 patent describes two “6-input 3-output” look-up tables, each capable of implementing every possible logic function of up to 6 inputs—that is, each is capable of implementing an arbitrary function of up to 6 inputs. *See id.*; ’737 patent at 5:24–7:20 & Fig. 6; *id.* 7:21–44 & Fig. 9.

Although Iida chose to explicitly define “a look up table of M inputs and N outputs” in the written description, his definition is consistent with the plain and ordinary meaning of that phrase. In the context of programmable logic devices, look up tables were at the time (and still are) generally understood as digital “memories” that could be used as the logic-implementing “component in programmable logic circuit devices.” *See supra*, p. 4 (describing look up tables); Melvin Decl. ¶¶ 36–41 (same). Moreover, a skilled artisan would have understood that a “look up table of M inputs” must be capable of performing an arbitrary function of those inputs on at least one output. For example, a 1992 textbook that both parties rely on as extrinsic evidence defines the term “Lookup Table (LUT)” as “a digital memory with K address lines that can implement **any function** of K inputs by placing the truth table into the memory.” Ex. 4 at 97300DOC000181665. The authors also explain that “[a] K-input LUT is a digital memory that can implement **any Boolean function of K variables**. The K inputs are used to address a 2^K by 1-bit memory that stores the truth table of the Boolean function.” *Id.* at 97300DOC000181716. Other publications in the field of programmable logic devices also defined “look up tables” as having sufficient memory to perform an arbitrary function of their inputs. *See, e.g.*, Ex. 12 at 97300DOC000187647 (“A K-

input lookup table is a digital memory that can implement any Boolean function of K variables. The K inputs are used to address a 2^K by 1-bit digital memory that stores the truth table of the Boolean function.”); *see generally* Melvin Decl. ¶¶ 54–58.

For all these reasons, the Court should construe “[a] look up table of M inputs and N outputs” as “a component of a programmable logic circuit device that is composed of memories and is capable of implementing an arbitrary logic circuit of M inputs on at least one of N outputs.”

B. “LUT units” (all asserted claims)

Iida’s Proposed Construction	Intel’s Proposed Construction
Plain and ordinary meaning	Indefinite Otherwise, if not indefinite, each “LUT unit” is a “component of a programmable logic circuit device that is composed of memories and is capable of implementing an arbitrary logic circuit of its inputs with at least one output”

The asserted claims are indefinite because they “fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention.” *Nautilus, Inc. v. Biosig Instruments, Inc.*, 572 U.S. 898, 901 (2014). Under any construction of “LUT units” tethered to the specification, including the plain and ordinary meaning of that phrase, a single device could be viewed as *simultaneously* infringing and not infringing the claims. “That is the epitome of indefiniteness.” *Geneva Pharms.*, 349 F.3d at 1384. The claims must be held invalid.

1. “LUT Units” Are Smaller Look Up Tables into Which a Larger “Look Up Table” May Be Divided

Half of the work in construing the phrase “LUT units” has already been done. Throughout the written description, “LUT” is used as an acronym for “look up table.” *E.g.*, ’737 patent at 1:23, 1:25, 1:66, 4:1, 4:59, 9:39. The term “LUT unit” should, therefore, be construed consistently with the term “look up table” discussed above. *See supra*, p. 10; *see generally Phillips v. AWH Corp.*,

415 F.3d 1303, 1314 (Fed. Cir. 2005) (holding “claim terms are normally used consistently throughout the patent”). Thus, the only remaining question in assessing the meaning of “LUT units” is nailing down the meaning of the word “units.”

In everyday speech, the word “units” refers to the individual parts into which a collective might be divided. The patent uses “units” in this same way. *See, e.g.*, ’737 patent at 1:56-58; *id.* 4:20-2; *id.* at 5:66-6:1; *id.* at 6:10-12; *see generally* Melvin Decl. ¶¶ 64–71. In every disclosed embodiment, the “LUT units” are individual parts of the collective “look up table.” *See, e.g.*, ’737 patent at FIGS. 6, 9, 5:26–32, 7:25–31. Those embodiments describe a 6-input 3-output look up table (64 bits) that can be divided into eight 3-input 1-output “LUT units” (8 bits each) or four 3-input 2-output “LUT units” (16 bits each). These are just different ways of dicing up the 64-bit “look up table” memory into subparts (i.e., units), each subpart forming its own “look up table.” It is similar to how a single apartment building is divided into various-sized (e.g., 1-bedroom or 2-bedroom) “apartment units” or how a measurement can be diced up into various “measurement units” (e.g., a foot, a mile).

The ’737 patent does not support, and to date Iida has not argued for, a departure from this plain and ordinary meaning of “units.” There are only two reasons that can justify adopting a construction that differs from a claim term’s plain and ordinary meaning: (1) lexicography and (2) disclaimer. *E.g., Thorner v. Sony Computer Ent. Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012). Neither is applicable here. The written description offers no definition of “units” or “LUT units,” and there is no evidence supporting a disclaimer in the specification or prosecution history.

Thus, the phrase “LUT units” should be construed in accordance with the patent’s definition of “look up table” and the ordinary meaning of “units.” In short, the “LUT units” are individual LUTs into which the larger “look up table of M inputs and N outputs” may be divided.

This is true *regardless* of how the Court chooses to construe “look up table of M inputs and N outputs.”²

2. The Meaning of “LUT Units” Renders the Asserted Claims Indefinite

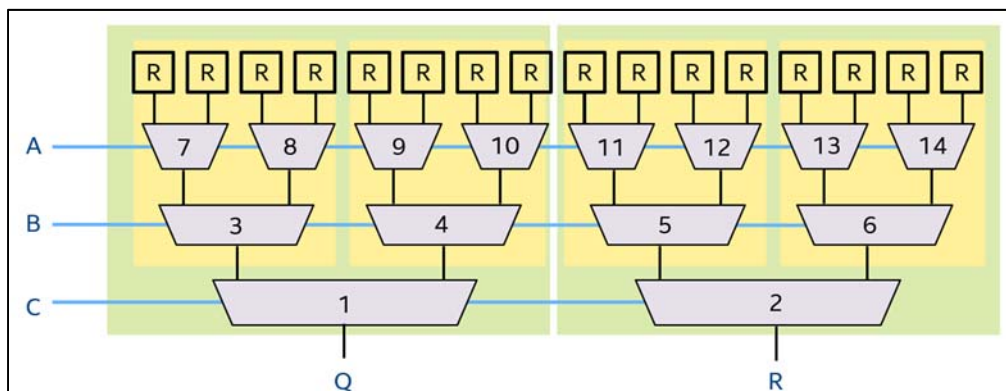
Even though the definition of “LUT units” can be reduced to words, the asserted claims are indefinite because a skilled artisan could not “translate” that definition “into meaningfully precise claim scope.” *Halliburton*, 514 F.3d at 1251. Here, the asserted claims recite “a plurality of LUT units” but place no restrictions on *how* those LUT units should be identified within the larger “look up table of M inputs and N outputs.” This is problematic because it means the same embodiment could both “simultaneously infringe and not infringe the claims” based on unclaimed conditions. *Geneva*, 349 F.3d at 1384.

The patent’s specification does not provide any limits on how one might view the “LUT units” in a particular “look up table of M input and N outputs.” In the claims, the “plurality of LUT units” is described as having inputs, outputs, and an internal configuration. *See, e.g.*, ’737 patent at claim 1 (“an internal configuration of said plurality of LUT units”), claim 2 (“an input of at least one of said LUT units . . . an output of at least one of said LUT units”). The claims also require the broader look up table to contain “a plurality of selectors” that select “I/O signals” (i.e., input/output signals) of the LUT units. *See, e.g., id.* at claim 1. But nothing about this language explains how to identify “LUT units” in a particular look up table. To be sure, there are dependent claims that require specific arrangements of LUT units within the “look up table of M inputs and N outputs.” For example, dependent claim 5 recites a “6-input 3-output look up table” that is

² As discussed above, *supra*, p. 11, the ’737 patent defines “look up table” to mean a “component of a programmable logic circuit device [that] is composed of memories and is capable of implementing an arbitrary logic circuit.” ’737 patent at 1:25-28 (emphasis added). Should the Court reject Intel’s indefiniteness argument, Intel’s alternative construction of “LUT units” is consistent with this lexicography. *See supra*, p. 10 (providing the alternative construction).

comprised of “eight 3-input 1-output LUT units,” which might inform the skilled artisan how to identify the “LUT units” in the larger “look up table of M inputs and N outputs.” *See also id.* at FIG. 6 (showing a “6-input 3-output look up table” that has been shown with “eight 3-input 1-output LUT units.”). But where the claims do not specify the size of the LUT units, a skilled artisan would not know **how** to identify them in a particular device. *See* Melvin Decl. ¶¶ 75–77. Moreover, there is no lexicography or disclaimer in the broader patent or in the prosecution history that could narrow the claims. *See, e.g., Phillips*, 415 F.3d at 1323 (“[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments.”). The claim language thus permits identifying “LUT units” in any way that meets the plain and ordinary meaning of that phrase—that is, any individual LUTs into which the larger “look up table of M inputs and N inputs” may be divided would read on the term “LUT units” as it used in the patent claims.

For a given device, this means that any smaller look up tables within the bigger “look up table of M inputs and N outputs” read on the term “LUT units.” Take the example discussed above (*see supra*, p. 7):



Under the claim language, this single device may be viewed as having four LUT units (yellow boxes), each having two inputs (A and B), with the outputs from the left LUT units proceeding to output selector 1 and the outputs from the right LUT units proceeding to output selector 2.

Alternatively, this same device may be viewed as having two (rather than four) LUT units (green boxes), each having *three* inputs (A, B, and C), with *no* selectors connected to outputs Q and R. Nothing in specification prevents or preferences either of these perspectives. See Melvin Decl. ¶ 78.

But how someone views this look up table—a **wholly** arbitrary decision—affects infringement. On one hand, viewed as a collection of four 2-input LUT units (yellow boxes), the look up table has two output selectors (selectors 1 and 2) that select outputs from LUT units (the lines exiting the yellow boxes at 3, 4, 5, and 6). This **would** meet the language of various claim limitations requiring “a plurality of selectors” selecting output signals from the “LUT units.” See, e.g., ’737 patent at claim 1 (reciting “a plurality of selectors selecting I/O signals of said plurality of LUT units”), claim 7 (same), claim 13 (reciting “selectively controlling I/O signals”); see also ’737 patent at claim 2 (requiring “an input signal selector” and “an output signal selector”); claim 8 (same), claim 15 (requiring that “an input signal” and “an output signal” be “selectively controlled”). On the other hand, viewed as a collection of two 3-input LUT units (green boxes), the very same look up table has **no** selectors that select input or output signals, because selectors 1 and 2 are now *inside* the “LUT units.” That would prevent the **same** device from meeting the language of the **same** claim limitations. See *id.*; Melvin Decl. ¶ 79.

Put simply, this single device **both** meets the language of claims and does not meet the language of the claims **simultaneously**. Whether an embodiment falls within the scope of the claims turns entirely on how the boxes around the claimed “LUT units” are drawn, yet the claims provide **no** guidance on how to identify LUT units in a particular look up table. Because of this lack of guidance, there is no “clear notice of what is claimed” in the patent. *Nautilus*, 572 U.S. at 899.

The Federal Circuit has held claims are indefinite in analogous circumstances. *See, e.g., Teva Pharms. USA, Inc. v. Sandoz, Inc.*, 789 F.3d 1335, 1341 (Fed. Cir. 2015). In *Teva*, the Federal Circuit explained that “molecular weight”—a claim limitation—could be measured in three different ways that would yield **different** results for the **same** polymer sample. *Id.* at 1344–45. This would have been **pivotal** for infringement purposes because the claims required a polymer compound “having a molecular weight of about 5 to 9 kilodaltons.” *Id.* at 1338. Thus, because the claim did not specify which measurement to use, the claims did not “inform with reasonable certainty those skilled in the art about the scope of the invention.” *Id.* at 1345. In precisely the same way, the meaning of “LUT units” allows dicing up the look up table in **different** ways for the **same** embodiment in a way that is **pivotal** for assessing infringement.

In subsequent cases, the Federal Circuit has distilled this doctrine into a three-part test. *See, e.g., Saso Golf, Inc. v. Nike, Inc.*, 843 F. App’x 291, 296 (Fed. Cir. 2021) (golf club head); *Pac. Coast Bldg. Prod., Inc. v. CertainTeed Gypsum, Inc.*, 816 F. App’x 454, 459 (Fed. Cir. 2020) (acoustical sound proofing). Claims are indefinite when (1) there are multiple methods for measuring a claimed parameter, (2) the patent does not specify which method applied, and (3) the application of the different methods results in materially different claim scope.

All three parts of that test are satisfied here. There are multiple ways of dividing a look up table into “LUT units,” the patent does not specify how it should be done, and the different divisions can be outcome-determinative for infringement. Thus, the asserted claims are indefinite and, therefore, must be held invalid.

Iida nevertheless might argue that there is some narrower interpretation of the term “LUT units” that might save the claims. To be sure, patent claims should be interpreted, *if possible*, in a manner that will preserve the validity of the claims. However, “[c]laim terms should be given their

plain and ordinary meaning to one of skill in the art at the relevant time and cannot be rewritten by the courts to save their validity.” *Hill-Rom Services, Inc. v. Stryker Corp.*, 755 F.3d 1367, 1374 (Fed. Cir. 2014). Here, there is nothing in the claims, written description, or prosecution history that would provide a basis for narrowing the plain and ordinary meaning of the term “LUT units” to save the validity of the claims.

Accordingly, the Court should find that the plain and ordinary meaning of the term “LUT unit” renders all the asserted claims invalid as indefinite.

IV. CONCLUSION

For the foregoing reasons, Intel respectfully requests that the Court hold that all asserted claims are indefinite or, in the alternative, that the Court adopt Intel’s proposed constructions.

Respectfully submitted,

Date: January 19, 2022

By: /s/ J. Stephen Ravel

J. Stephen Ravel
Texas State Bar No. 16584975
Kelly Ransom
Texas State Bar No. 24109427
KELLY HART & HALLMAN LLP
303 Colorado, Suite 2000
Austin, TX 78701
Tel: (512) 495-6429
Fax: (512) 495-6401
Email: steve.ravel@kellyhart.com
Email: kelly.ransom@kellyhart.com

Gregory P. Stone (*pro hac vice*)
Zachary M. Briers (*pro hac vice*)
Vincent Y. Ling (*pro hac vice*)
Heather E. Takahashi (*pro hac vice*)
Faye P. Teller (*pro hac vice*)
MUNGER, TOLLES & OLSON LLP
350 South Grand Avenue, Fiftieth Floor
Los Angeles, CA 90071-3426
(213) 683-9100

Andrew T. Nguyen (*pro hac vice*)
MUNGER, TOLLES & OLSON LLP
560 Mission Street, Twenty-Seventh Floor
San Francisco, CA 94105
(415) 512-4000

J. Kain Day (*pro hac vice*)
MUNGER, TOLLES & OLSON LLP
601 Massachusetts Avenue NW, Suite 500 E
Washington, D.C. 20001
(202) 220-1100

Attorneys for Defendant Intel Corporation

CERTIFICATE OF SERVICE

The undersigned hereby certifies that a true and correct copy of the above and foregoing document has been served on January 19, 2023, to all counsel of record via the Court's CM/ECF system..

/s/ J. Stephen Ravel
J. Stephen Ravel