

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

OASIS TOOLING, INC.,)	
)	
Plaintiff,)	
)	
v.)	Civil Action No. 22-151-CJB
)	
SIEMENS INDUSTRY SOFTWARE,)	
INC.,)	
)	
Defendant.)	
<hr/>		
OASIS TOOLING, INC.,)	
)	
Plaintiff,)	
)	
v.)	Civil Action No. 22-312-CJB
)	
GLOBALFOUNDRIES U.S., INC.,)	
)	
Defendant.)	

Philip A. Rovner and Jonathan A. Choa, POTTER ANDERSON & CORROON LLP, Wilmington, DE; Paul J. Andre, Lisa Kobialka, James Hannah and Timothy Layden, KRAMER LEVIN NAFTALIS & FRANKEL LLP, Redwood Shores, CA; Aaron M. Frankel and Cristina L. Martinez, KRAMER LEVIN NAFTALIS & FRANKEL LLP, New York, NY, Attorneys for Plaintiff.

Karen Jacobs and Cameron P. Clark, MORRIS, NICHOLS, ARSHT & TUNNELL LLP, Wilmington, DE; John D. Vandenberg, Kristin L. Cleveland and Mark W. Wilson, KLARQUIST SPARKMAN, LLP, Portland, OR; Kristina R. Cary, KIRKLAND & ELLIS LLP, Boston, MA; Gregg F. LoCascio, P.C., Michael A. Pearson, Jr. and Matthew J. McIntee, KIRKLAND & ELLIS LLP, Washington, D.C., Attorneys for Defendant Siemens Industry Software, Inc.

Brian E. Farnan and Michael J. Farnan, FARNAN LLP, Wilmington, DE; Clement Naples, LATHAM & WATKINS LLP, New York, NY; Gabriel K. Bell, LATHAM & WATKINS LLP, Washington, D.C.; Thomas W. Yeh, LATHAM & WATKINS LLP, Los Angeles, CA; Brett M. Sanford and Daniel S. Todd, LATHAM & WATKINS LLP, San Francisco, CA, Attorneys for Defendant GlobalFoundries U.S. Inc.

MEMORANDUM OPINION AND ORDER

November 17, 2023
Wilmington, Delaware

Christopher J. Burke
BURKE, United States Magistrate Judge

In these two related actions filed by Plaintiff Oasis Tooling, Inc. (“Oasis” or “Plaintiff”) against Defendants Siemens Industry Software, Inc. (“Siemens”) and GlobalFoundries U.S., Inc. (“GF” and collectively with Siemens, “Defendants”), Oasis alleges infringement of United States Patent Nos. 7,685,545 (the “’545 patent”) and 8,266,571 (the “’571 patent” and collectively with the ’545 patent, “the asserted patents”). Presently before the Court is the matter of claim construction. (Civil Action No. 22-151-CJB, D.I. 76; D.I. 77; Civil Action No. 22-312-CJB, D.I. 72; D.I. 73)¹ The Court² hereby adopts the constructions as set forth below.

I. BACKGROUND

Oasis filed its Complaint against Siemens on February 1, 2022 in Civil Action No. 22-151-CJB. (D.I. 1) Oasis accuses Siemens’ Calibre Design Solutions suite of infringing at least claims 1 and 16 of the ’571 patent, and at least claim 1 of the ’545 patent. (*Id.* at ¶¶ 33, 49, 63, 71, 84)

Oasis filed its Complaint against GF on March 9, 2022 in Civil Action No. 22-312-CJB. (Civil Action No. 22-312-CJB, D.I. 1) Oasis accuses GF’s DRC+ tool and its open process technology platforms of infringing at least claim 16 of the ’571 patent, and at least claim 14 of the ’545 patent. (*Id.* at ¶¶ 58, 82, 99, 107, 125)

¹ For simplicity’s sake, hereafter the Court will refer to the “D.I.” number in Civil Action No. 22-151-CJB, unless otherwise indicated.

² On May 11, 2022, the parties consented to the Court’s jurisdiction to conduct all proceedings in these actions, including entry of a final judgment. (D.I. 26; Civil Action No. 22-312-CJB, D.I. 19)

The two patents-in-suit, both titled “Methods and Devices for Independent Evaluation of Cell Integrity, Changes and Origin in Chip Design for Production Workflow,” share a common specification.³ The '545 patent issued on March 23, 2010 from U.S. Appl. No. 12/536,413, which was filed on August 5, 2009. ('545 patent at 1)⁴ The '571 patent issued on September 11, 2012 from U.S. Appl. No. 12/482,296, which was filed on June 10, 2009. ('571 patent at 1) The patents relate to systems and methods for the granular analysis of design data, which is used to prepare chip designs for manufacturing and to identify similarities and differences among design data residing in files. (*Id.*, Abstract; *see also* D.I. 1 at ¶ 15) Further details regarding the asserted patents will be provided below in Section III.

On March 23, 2023, the parties filed their joint claim construction brief. (D.I. 88) The Court conducted a *Markman* hearing on April 20, 2023. (D.I. 161 (hereinafter, “Tr.”)) On May 24, 2023, Defendants submitted a notice of supplemental authority. (D.I. 117)

II. STANDARD OF REVIEW

The Court has often set out the relevant legal standards for claim construction, including in *Vytacera Bio, LLC v. CytomX Therapeutics, Inc.*, Civil Action No. 20-333-LPS-CJB, 2021 WL 4621866, at *2-3 (D. Del. Oct. 7, 2021). The Court hereby incorporates by reference its discussion in *Vytacera Bio* of these legal standards and will follow them herein. To the extent consideration of the disputed terms here necessitates discussion of other, related legal principles, the Court will address those principles in Section III below.

III. DISCUSSION

³ As such, the Court will cite below only to the '571 patent, unless otherwise noted.

⁴ The patents-in-suit are attached as exhibits to the relevant Complaints. (D.I. 1, exs. 1-2; Civil Action No. 22-312-CJB, D.I. 1, exs. 1-2) Herein, the Court will cite to the patents by their patent number.

The parties set out eight disputed terms for the Court’s review.⁵ This Memorandum Opinion first addresses the six terms that were argued at the *Markman* hearing, in the order in which they were argued. The Court then takes up the remaining two terms, which were submitted on the papers.

A. “canonical forms”

The first disputed term, “canonical forms[.]” appears in, *inter alia*, claims 1, 4, 5-6, 8-11 and 14 of the '545 patent and claims 1, 2, 4, 15 and 16 of the '571 patent. Exemplary claim 1 of the '545 patent recites:

1. A computer-implemented method of evaluating similarities and/or differences between design data for circuits, the design data residing in at least two files stored in computer memory, the method including:

using a computer, identifying cells within design data residing in first and second files, wherein the cells correspond to portions of design for a physical circuit;

parsing syntax of and normalizing the design data within the cells into *canonical forms*, wherein the *canonical forms* reduce sensitivity of data analysis to non-functional variations in the design data within a particular cell;

partitioning functionally significant design data from non-significant data within the *canonical forms*, wherein the design data is functionally significant when a change in the design data would result in a change in a circuit generated from the design data;

calculating and storing digests of at least selected design data in the *canonical forms*, producing at least one digest per cell;

⁵ There was a ninth disputed term, “syntax trees,” discussed in the briefing. (D.I. 88 at 53-57) However, by the end of briefing, Oasis accepted Defendants’ construction: “a hierarchical data structure representing the syntax of design data with connected nodes, each node having exactly one parent, except a single root node which has no parent[.]” (*Id.* at 56-57; D.I. 103, ex. 2 at 2) The Court will therefore adopt Defendants’ proposed construction for “syntax trees.”

wherein the selected design data in the *canonical forms* used to calculate the digests includes at least the functionally significant design data;

comparing the digests of the cells in the first file to the digests of the cells in the second file; and

summarizing at least some results of the comparing of the digests.

('545 patent, cols. 81:56-82:15 (emphasis added)) There is no dispute here that the patents tell us that a “canonical form” transforms functionally equivalent data into the same representation, so that their equivalence will be detected by comparing their canonical form representations. (*See, e.g.,* D.I. 88 at 6, 15-16 (citations omitted))

The parties’ competing proposed constructions for “canonical forms” are set out in the chart below:

Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“canonical forms”	<p>“A normalized form of a body of design and/or manufacturing data that reduces the sensitivity of data comparison analysis to variations in the data that have no functional impact on the design.</p> <p>For design data including orientation data, either (1) the canonical form reduces sensitivity to different expressions of the orientations of the design data or (2) the digest of the canonical form includes data corresponding to different orientations of the canonical form.”</p>	<p>“a standardized format in which all functionally equivalent instances of a given unit of design data are transformed into the same representation”</p> <p>Also, Indefinite.</p>

(D.I. 103, ex. 2 at 1)

The parties have four disputes regarding this term. The Court will address them in turn.

1. Are “canonical forms” required to eliminate all false negatives?

The first dispute is whether canonical forms are required to merely “*lessen* the likelihood of a false negative” (as Oasis contends) or whether they are required to eliminate *all* false negatives (as Defendants contend). (D.I. 88 at 8 (emphasis added); Tr. at 40)⁶ Here, the Court concludes that canonical forms are not required to eliminate all false negatives (and, for this reason, declines to adopt Defendants’ proposed construction).

The outcome here is driven primarily by the plain language of the claims. (D.I. 88 at 8) As Oasis points out, all of the asserted independent claims recite that the canonical forms “*reduce* sensitivity of data analysis to non-functional variations in the design data within a particular cell[.]” (See, e.g., ‘545 patent, col. 81:64-66 (emphasis added); ‘571 patent, col. 81:53-56 (emphasis added)) The patentee did not include a requirement that the canonical forms *eliminate all* sensitivity of data analysis to non-functional variations—it instead chose the term “reduce.” (D.I. 88 at 8, 21; Tr. at 9, 11) It is not disputed that the plain and ordinary meaning of “reduce” is to lessen, but not necessarily to eliminate entirely. (D.I. 88 at 21; Tr. at 16-17; see also D.I. 88, ex. C-3 at 3 (defining “reduce” to mean “to diminish in size, amount, extent, or number”); D.I. 88, ex. C-4 at 1 (defining “reduce” to mean “to bring down to a smaller extent, size, amount, number, etc.”))

⁶ In other words, under Oasis’ view of the term, the canonical forms may reduce sensitivity of data analysis with regard to at least *some* nonfunctional variations in the design data, but may not catch *all* nonfunctional variations in a particular cell. (Tr. at 10) Meanwhile, under Defendants’ view of the term, the canonical forms must reduce sensitivity of data analysis with regard to *all* nonfunctional variations in the design data within a particular cell. (*Id.* at 10, 50-51)

In addition to the import of this claim language, the shared specification is also helpful to Oasis. (D.I. 88 at 8-9; Tr. at 11-12) The Abstract teaches that “[o]rganizing the design data into canonical forms *generally reduces* the sensitivity of data analysis to variations in data that have no functional impact on the design.” (‘571 patent, Abstract (emphasis added)) This summary of the patented inventions underscores that the claimed canonical forms need not strictly eliminate all sensitivity of data analysis, all of the time; instead, it merely requires a general lessening of such sensitivity. Other portions of the specification further reinforce this notion. (*Id.*, col. 69:23-24 (“The canonical forms reduce sensitivity of data analysis to non-functional variations in the design data.”); *see also id.*, cols. 1:32-33, 3:21-24)⁷

Defendants’ arguments to the contrary are not persuasive. Defendants primarily contend that “the ordinary technical meaning of ‘canonical form[s]’” is what supports their position that “canonical forms” must entirely eliminate all sensitivity of data analysis to nonfunctional variations. (D.I. 88 at 12, 17; Tr. at 38-39) To that end, they point to the declaration of their expert, Dr. Majid Sarrafzadeh, who opines that “multiple definitions and uses of ‘canonical form’ in various disciplines” boil down to the same meaning—i.e., Defendants’ proposed construction (“a standardized format in which *all* functionally equivalent instances of a given unit of design data are transformed into the same representation”). (D.I. 88 at 17 (citing D.I. 90 at ¶ 30))

⁷ Oasis’ expert, Dr. Stephan Athan, opines that a person of ordinary skill in the art (“POSITA”) reading the claims in view of the specification would further understand that the invention need not eliminate all such sensitivity, because certain embodiments describe design files that include “unconstrained textual data” relating to the five major steps of circuit development, and because such data “may be indirectly coupled to geometric data, [such that] it would be difficult to reliably convert them into a canonical form that would guarantee elimination of all sensitivity to non-functional variation.” (D.I. 92 at ¶¶ 10-11)

However, as Oasis retorts, at least certain of these uses of “canonical form” actually seem to support *Oasis*’ position. (*Id.* at 23-24; Tr. at 18-19) For example, Dr. Sarrafzadeh cites to United States Patent No. 5,832,480 (“Byrd”), which relates to the use of canonical forms to develop a dictionary of names in a text. (D.I. 90 at ¶ 31.a; D.I. 88, ex. C-5 at 1) Byrd explains that “[i]n spell checking applications, identifying proper names and treating all variants as instances of [the] same canonical form *will reduce the false alarms* for misspelling that the checker normally outputs for the user.” (D.I. 88, ex. C-5 at col. 25:17-20 (emphasis added)) This explanation of the use of the canonical form in Byrd is similar to that here, in that the canonical form will work to *reduce* false alarms—without necessarily completely eliminating them. (D.I. 88 at 23-24) As another example, Dr. Sarrafzadeh cites to an article entitled “Detecting Co-Derivative Source Code — An Overview” (the “Elo reference”) which discusses canonical forms in the context of source code. (D.I. 90 at ¶ 31.c)⁸ The article notes that such canonization facilitates straightforward comparisons, while recognizing that “[u]nfortunately, it is unrealistic to expect to attain perfect canonization for a typical generic-purpose programming language[.]” (D.I. 91, ex. D-7 at 33-34)⁹

⁸ The Elo reference was cited during prosecution of the patents, so it constitutes intrinsic evidence. (Tr. at 47-48); *see also, e.g., Bayer Pharma AG v. Watson Labs., Inc.*, Civil Action No. 12-1726-LPS-CJB, 2014 WL 4954617, at *9 (D. Del. Sept. 30, 2014).

⁹ Defendants additionally argued in their briefing that Oasis’ position (i.e., that merely reducing non-functional variations would be sufficient to create a canonical form) would “improperly erase a portion of the claim language”—specifically, the claim phrase “sensitivity of data analysis to.” (D.I. 88 at 31-32; Tr. at 49-50) The Court does not understand this argument, (D.I. 88 at 22), and it does not see how any aspect of Oasis’ construction “erase[s]” this portion of the claim language. In any event, whatever the meaning of “sensitivity of data analysis to non-functional variations,” the claims still contain the requirement that the canonical forms *reduce* this sensitivity. And it is Defendants’ proposal, not Oasis’ proposal, that is improperly erasing *that* requirement. Indeed, when pressed about “what work is the word ‘reduce’ doing in the claim” if their proposal was correct, Defendants’ response was only to say that the language

Defendants also suggest that if you only reduce (but do not eliminate) sensitivity to nonfunctional variations, that “doesn’t help the designer at all.” (Tr. at 91; *see also* D.I. 88 at 18 (“Removal of some, but not all, non-functional differences between two data items being compared does not solve the false-negative problem addressed by the patents.”)) Defendants did not provide record support for this assertion. The Court notes that even were that assertion to be accurate, the United States Court of Appeals for the Federal Circuit has explained that in circumstances where claim language would lead to a “nonsensical result[,]” courts still must not redraft claims. *See, e.g., Chef Am., Inc. v. Lamb-Weston, Inc.*, 358 F.3d 1371, 1373-74 (Fed. Cir. 2004) (concluding that claim language requiring “heating the resulting batter-coated dough to a temperature in the range of about 400° F. to 850° F” meant that the dough itself is to be heated to that temperature, even where that meant the dough would burn to a crisp instead of becoming suitable for cooking to a light, flaxy, crispy texture—a result that was intended by the patentee). That said, the Court does not really understand how an invention that, for example, provides canonical forms that reduce sensitivity of data analysis to four out of five types of nonfunctional variations would not still be helpful to some degree—especially as compared to methods that were sensitive to all variations in design data, whether functional or not. (Tr. at 20-21, 56-57, 60-61) Indeed, during the *Markman* hearing, even Defendants’ counsel appeared to acknowledge that such an invention could, in fact, be helpful to the public. (*Id.* at 45, 62)

2. Is “canonical forms” indefinite?

of the claim could be clearer—and that the way the patentees used the word “reduce” “muddies” things up. (Tr. at 51-54)

The parties' second dispute is whether the term is indefinite, as Defendants contend.¹⁰ Here, Defendants' position is that if the first issue were to come out Oasis' way (such that the canonical forms are required to merely reduce sensitivity to nonfunctional variations)—which it has—then the intrinsic record does not provide sufficient guidance as to “what type or percentage of non-functional differences need to be removed to create a ‘canonical form.’” (D.I. 88 at 18; Tr. at 46, 77) Indeed, according to Defendants, the prosecution history further complicates this question, because, in a “Concise Statement of Utility,” the patentee explained that “[t]he broadest method and device claims are useful to chip design managers because they enable the design managers to determine differences and/or similarities between a pair of complex design files *with a low level of noise* related to non-functional differences between the

¹⁰ Section 112 of the Patent Act requires that a patent claim “particularly point[] out and distinctly claim[] the subject matter which the inventor or a joint inventor regards as the invention.” 35 U.S.C. § 112(b). If it does not, the claim is indefinite and therefore invalid. *Nautilus, Inc. v. Biosig Instruments, Inc.*, 572 U.S. 898, 902 (2014). The primary purpose of the definiteness requirement is to ensure that patent claims are written in such a way that they give notice to the public of what is claimed, thus enabling interested members of the public (e.g., competitors of the patent owner) to determine whether they infringe. *All Dental Prodx, LLC v. Advantage Dental Prods., Inc.*, 309 F.3d 774, 779-80 (Fed. Cir. 2002). Even so, “absolute precision is unattainable” and is not required. *Nautilus*, 572 U.S. at 910. In the end, “a patent is invalid for indefiniteness if its claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention.” *Id.* at 901. As long as claims satisfy the test for definiteness, “relative terms and words of degree do not render patent claims invalid.” *One-E-Way, Inc. v. Int’l Trade Comm’n*, 859 F.3d 1059, 1063 (Fed. Cir. 2017). Definiteness is to be evaluated from the perspective of a POSITA at the time the patent was filed. *Nautilus*, 572 U.S. at 908.

Like claim construction, definiteness is a question of law for the court. *H-W Tech., L.C. v. Overstock.com, Inc.*, 758 F.3d 1329, 1332 (Fed. Cir. 2014). The Federal Circuit has stated that “[a]ny fact critical to a holding on indefiniteness . . . must be proven by the challenger by clear and convincing evidence.” *Intel Corp. v. VIA Techs., Inc.*, 319 F.3d 1357, 1366 (Fed. Cir. 2003); *see also Tech. Licensing Corp. v. Videotek, Inc.*, 545 F.3d 1316, 1338 (Fed. Cir. 2008).

design files.” (D.I. 75, ex. B-5 at 31 (emphasis added)) Defendants’ argument here seems to be that if the claim term “canonical forms” requires a reduction in sensitivity to nonfunctional variations such that a “low level” of noise is achieved, then the term is one of degree and is necessarily indefinite, because the specification “provides no guide as to how to measure the level of noise or what level of noise is too high to be ‘low.’” (D.I. 88 at 17-18; *see also id.* at 19, 34)¹¹

In the Court’s view, however, the intrinsic record provides the necessary objective baseline through which a person of ordinary skill in the art (“POSITA”) can ascertain the boundaries of the claims with reasonable certainty. (*Id.* at 27-29; Tr. at 32-33) As discussed above, the claimed canonical forms must reduce—i.e., lessen—sensitivity of data analysis to non-functional variations in the design data. The intrinsic record does not spell out a minimum amount of reduction that must be achieved in order to practice the claims. (D.I. 88 at 27; Tr. at 32) Thus, as Oasis explains, the POSITA would know how to objectively determine whether an accused product satisfied the “canonical forms” limitation: (1) one would compare the data as it existed before it was normalized into a canonical form (the “un-normalized data”) to the canonical form; and (2) if the canonical form reduced sensitivity to nonfunctional variations as compared to the data as it existed before, then the claim limitation is met. (D.I. 88 at 28-29; Tr. at 32-33 (“the normalized data has to be less sensitive than the un-normalized data”)) In other

¹¹ In their briefing, Defendants at times suggested that the patentee’s reference to achieving “a low level of noise related to nonfunctional differences” amounted to a kind of definitional statement as to the meaning of “canonical forms.” (D.I. 88 at 34) At the *Markman* hearing, however, Defendants’ counsel acknowledged that this statement did not amount to a disclaimer. (Tr. at 76) The Court agrees with Oasis that while the patentee was here setting out the utility of the invention, the statement of utility does not impose some more specific “threshold level” for reduction with respect to the claim term “canonical forms.” (*Id.* at 36, 79-80)

words, for example, if analysis of the un-normalized data would pick up five nonfunctional variations in the data of a particular cell, and analysis of the normalized data would pick up three of those five nonfunctional variations, the claim limitation is met—the canonical form has reduced sensitivity of data analysis to nonfunctional variations in the design data. (Tr. at 80) The claim term “canonical forms” is definite. *See, e.g., Liberty Ammunition, Inc. v. United States*, 835 F.3d 1388, 1395-96 (Fed. Cir. 2016) (concluding that term “reduced area of contact” was not indefinite where the intrinsic record provided a sufficient baseline for determining whether the area of contact had been reduced); *DSM IP Assets, B.V. v. Lallemand Specialties, Inc.*, 16-cv-497-wmc, 2018 WL 1433850, at *18 (W.D. Wis. Mar. 22, 2018) (“[T]here is nothing inherently indefinite about the requirement that there be ‘reduced’ activity.”).

3. Can the design data included in the canonical forms include manufacturing data?

The third dispute is whether the design data included in the canonical forms can include manufacturing data. (D.I. 88 at 13, 24; Tr. at 30, 38) Oasis asserts that it can (and thus includes “and/or manufacturing data” in its proposed construction); Defendants disagree.

The Court will not adopt Oasis’ proposed language here. The claim language is clear that it is “design data” that is contained within the canonical forms. (*See* D.I. 88 at 32) And to the extent that what is disputed here is whether *design data* can include manufacturing data, that really seems to be more of a dispute about the meaning of the claim term “design data,” not a dispute about the meaning of “canonical forms.” (Tr. at 30, 32-33 (Oasis’ counsel explaining that the “canonical form is generated from the design data. Our position is that . . . can include processing the manufacturing data that’s within the design data”)) And such a claim construction dispute is not well teed up here; it would need to be addressed in a more fulsome fashion later in this case (if necessary). (*See, e.g.,* D.I. 88 at 8-11 (Oasis’ opening claim

construction brief failing to even address the manufacturing data issue); Tr. at 38 (Defendants' counsel asserting that this dispute really concerns "how broad is [] design data[,] " which was not an issue addressed in the parties' briefing))

4. Does "canonical forms" require a reduction in sensitivity to nonfunctional variations when it comes to orientation/rotation data?

The final dispute has to do with whether the construction should include Oasis' second sentence. This sentence is meant to indicate that if the design data at issue includes geometric data, then the claims require a reduction of sensitivity to rotational/orientation issues. (Tr. at 22-23; D.I. 88 at 10-11 ("[A] POSITA would understand that either the canonical form or the digest of the canonical form must address rotational issues."), 25)¹² In other words, Oasis' position is that while, as a general matter, the claims merely require some reduction in sensitivity of data analysis to nonfunctional variations, to the extent that the design data includes geometric data, then there *must* be a reduction of sensitivity to rotational/orientation issues. (Tr. at 22-23) Oasis acknowledges that it is the party seeking a narrower construction with respect to this aspect of its proposal. (*Id.*) The Court finds that the construction should not include Oasis' second sentence.

This conclusion is dictated by the plain language of the claims. The claim language does not itself call out any type of design data that should get special treatment. (*See* D.I. 88 at 14 (Defendants noting that "no asserted claim refers to 'orientations' or even to geometric shapes"); Tr. at 64-65))

Oasis hangs its hat here on a snippet from the specification. (D.I. 88 at 10, 26) The specification explains that:

Design tools have considerable freedom to choose the OASIS® elements used to represent the geometry of a layout cell. . . . Such

¹² Oasis views "orientation" and "rotation" synonymously. (Tr. at 86) Defendants disagree that orientation data is the same thing as rotation data. (*Id.* at 41, 69)

a tool might also change the “winding direction” of a POLYGON from counterclockwise to clockwise. To avoid these issues, *all geometric elements are converted to a canonical representation* for canonical cell digest calculation[.]

(’571 patent, col. 34:7-17 (emphasis added)) But this excerpt does not do the work that Oasis claims it does. It is found in a description of an embodiment of the invention. (*See id.*, col. 32:10-15 (introducing examples 5-6 relating to OASIS® and GDSII file types); Tr. at 25) The Federal Circuit has “repeatedly warned against confining the claims to [the specific] embodiments” set out in the specification. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005); *see also Kara Tech. Inc. v. Stamps.com Inc.*, 582 F.3d 1341, 1345 (Fed. Cir. 2009). It is the claims that claim, and again, they do not specifically call out orientation/rotational data for any specialized role here.

Oasis also asserts that it “confirmed this requirement to address rotational issues” during prosecution, when it discussed United States Patent No. 7,386,819 (the “Yuan reference”). (D.I. 88 at 10-11; Tr. at 29-30) This argument too is not a winner. Yuan was identified as a closely related reference to the subject matter of the patents, and the patentee explained that “Y[ua]n normalizes the LUT of the FPGA into a canonical minimum LUTmask by rotating or permuting inputs, as describe[d] above.” (D.I. 75, ex. B-5 at 13; *see also id.* at 6-7) The Court does not understand how this general description of Yuan could amount to a definitional statement about the asserted claims, let alone one that would import a requirement with respect to a particular type of data into the meaning of canonical forms.

In sum, the portions of the record cited by Oasis do not support adoption of a construction for “canonical forms” that is narrower than the term’s plain and ordinary meaning. *See Tonal Sys., Inc. v. iFit Inc.*, Civil Action No. 20-1197-GBW-CJB, 2023 WL 3089920, at *6 (D. Del. Apr. 26, 2023) (citing cases).

5. Construction

With the parties' disputes resolved, that leaves the question of how "canonical forms" should be construed in light of such resolution. Some of the language in Oasis' first sentence of its proposal is redundant, in that it spells out how the canonical form "reduces the sensitivity of data comparison analysis to variations in the data that have no functional impact on the design"; this job is already taken care of by the claims' "wherein" clause that follows "canonical forms." (See D.I. 88 at 12) And while Oasis' proposal uses the phrase "normalized," there seems to be no dispute that this means "standardized[;]" the Court believes that "standardized" is a more helpful word to use for the jury's benefit. (See *id.* at 16; '571 patent, col. 75:4 ("By canonical, we mean in a standardized format."))

For all of the above reasons, "canonical forms" should be construed to mean "a standardized form of a body of design data."

B. "digest[s]"

The next disputed term, "digest[s]" appears in, *inter alia*, claims 1-11, 13-14 and 18-20 of the '545 patent and claims 1-3, 5-12, 14-16 and 20 of the '571 patent. Exemplary claim 1 of the '571 patent recites:

1. A computer-implemented method of evaluating similarities and/or differences between design data for circuits, the design data residing in at least two files stored in computer memory, the method including:

using a computer, identifying cells within design data residing in first and second files, wherein the cells have logical names as parts of the design data and the cells correspond to portions of design for a physical circuit;

parsing syntax of and normalizing the design data within the cells into canonical forms, wherein the canonical forms reduce sensitivity of data analysis to non-functional variations in the design data within a particular cell;

calculating and storing *digests* of at least selected design data in the canonical forms, producing at least one *digest* per cell that uniquely *digests* functional data design within the cell;

comparing the *digests* of the cells in the first file to the *digests* of the cells in the second file; and

summarizing at least some results of the comparing of the *digests*.

('571 patent, col. 81:44-64 (emphasis added))

The parties' competing proposed constructions for "digest[s]" are set out in the chart below:

Term	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"digest[s]"	<p>"The archivable collection of result(s) generated from the application of hash function(s) to at least one canonical form.</p> <p>For design data including orientation data, either (1) the canonical form reduces sensitivity to different expressions of the orientations of the design data or (2) the digest of the canonical form includes data corresponding to different orientations of the canonical form."</p>	<p>"output of a hash function, including, e.g., CRC or MD5"</p>

(D.I. 103, ex. 2 at 1)

The parties agree that, as a general matter, a digest is the output of a hash function. (D.I. 88 at 39, 40 n.7; Tr. at 96; Defendants' *Markman* Presentation, Slide 38)¹³ And so they have one

¹³ The specification notes that "[a] variety of hash functions can be used to create the digests, such as CRC, MD5 and others." ('571 patent, col. 6:16-17)

key dispute remaining regarding this term that the Court has not already addressed¹⁴: whether a digest is the output of a *single* hash value (as Defendants assert), or whether a digest can be a “collection of multiple hashing functions” (as Oasis contends). (D.I. 88 at 39, 41-42 (emphasis added); Tr. at 95-96, 113-14) The Court sides with Defendants here.¹⁵

Both parties agree that the following excerpt from the specification is key with respect to this dispute:

Here is a portion of a digest report for an OASIS® file:

Cell “Structure_1” (not sorted)	
f64ce851	Cell with Comments
f64ce851	Cell without Comments
(none)	Cell Comments
20b84e54	Cell Body non-Layer
e03bf9d2	Cell Body Layer 3
fda35715	Cell Body Layer 42
cb6e08e2	Cell Body non-Geometric Data Layer 3

(’571 patent, col. 33:24-35; *see also* Tr. at 106, 108) For its part, Oasis claims that the specification recites “numerous embodiments where the digest is a collection of multiple hashing functions[,]” and it then calls out this example. (D.I. 88 at 41) It asserts that here the specification is describing the collection of “the result of 7 different hash functions in a single *digest report* for a cell, with each hash corresponding to different canonical forms” generated to capture different variations of the same cell (such as, for example, a canonical form of the cell with comments, without comments, and non-geometric data within the cell). (*Id.* (emphasis added)) From this, Oasis argues that a POSITA would understand that this embodiment involves

¹⁴ While Oasis also seeks to include in the construction for “digest[s]” the same second sentence that it sought to include in the construction for “canonical forms,” (*see* D.I. 88 at 37), for the same reasons as discussed above, the Court declines to include this language in its construction for “digest[s][,]” (*id.* at 39; Tr. at 95-96).

¹⁵ Defendants acknowledge that there can be a *collection* of digests, but their point is that *each* digest must be a *single* value. (D.I. 88 at 39)

“collections of the results of multiple hashes generated by looking at different aspects of a cell”—i.e., a digest. (*Id.*)

But this specification excerpt cannot bear the weight that Oasis puts on it. Had the excerpt said something like “here is a digest for an OASIS® file,” then Oasis would have a much stronger argument. But it does not. Instead, as Oasis had to acknowledge, the excerpt refers to the collection of results from multiple hash functions as a “digest *report*.” (Tr. at 109, 111, 122; *see also id.* at 102-03 (Defendants’ counsel explaining that this digest report is a report of multiple digests that have been run)) And Oasis points to nothing in the specification telling us that a digest report is synonymous with a digest (nor does it point to anything in the specification that makes clear that a digest is a collection of multiple hash functions). (*See id.* at 103)

Indeed, as Defendants note, other portions of the specification confirm that a digest is a single value (rather than a collection of multiple values). (D.I. 88 at 42; Tr. at 126) For example, the specification refers to “(none)” in the excerpt of the digest report above as a “digest[.]” (’571 patent, col. 33:37-39) Another portion of the specification describes the generation of “a 32-bit file-level digest of db7be73c (hexadecimal).” (*Id.*, col. 9:46-48) Elsewhere, the specification notes that “[i]f sorting is requested, different digests are generated for some parts of the cell[.]” with an exemplary table then listing out a collection of hash values. (*Id.*, col. 10:50-51)

For all of the above reasons, “digest[s]” should be construed to mean “output of a hash function, including, e.g., CRC or MD5.”

C. “cell”

The next disputed term, “cell[.]” appears, *inter alia*, in claims 1-12, 14 and 18-20 of the ’545 patent and claims 1, 3-13, 15-16 and 20 of the ’571 patent. The term’s use in claim 1 of

each of the '545 and '571 patents, depicted above, is representative. The parties' competing proposed constructions for "cell" are set out in the chart below:

Term	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"cell"	<p>"Cell" is a technical term and should be given its plain and ordinary technical meaning to a POSITA/no construction required.</p> <p>In the alternative, the plain and ordinary meaning of the term could be articulated as "a subset of design data that can be referenced as a whole representation and expression of: an object, a set of instructions, properties, or comments."</p>	<p>"A design unit having a view and an artifact, encompassing, for example, VHDL constructs including entities, configurations, architectures, procedures, functions, components, types, and subtypes."</p> <p>Also, Indefinite.</p>

(D.I. 103, ex. 2 at 1-2)

Both parties agree that "cell" is a technical term. (D.I. 88 at 43, 46) The parties' disputes regarding this term largely flow from Defendants' position that, nevertheless, Oasis gave the term "cell" a special definition in the specification that differs from its plain and ordinary meaning. (*Id.* at 46; Tr. at 138-39, 146-47; *see also* Defendants' *Markman* Presentation, Slide 53) Oasis, for its part, disagrees, asserting that the patents utilize "cell" in line with its plain and ordinary meaning. (D.I. 88 at 43-44; Tr. at 127-28)¹⁶

¹⁶ It is beyond dispute that a rule of claim construction is that "[t]he words of a claim are generally given their ordinary and customary meaning as understood by a person of ordinary skill in the art when read in the context of the specification and prosecution history." *Thorner v. Sony Comput. Ent. Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012). There are two exceptions to this rule: "1) when a patentee sets out a definition and acts as his own lexicographer, or 2) when the patentee disavows the full scope of a claim term either in the specification or during prosecution." *Id.* The standards for finding lexicography and disavowal are "exacting." *Pacing Techs., LLC v. Garmin Int'l, Inc.*, 778 F.3d 1021, 1024 (Fed. Cir. 2015) (internal quotation marks and citation omitted).

According to Defendants, the specification provides three definitions for “cell” that all must be incorporated into the Court’s construction. (D.I. 88 at 46, 52; Tr. at 139-40, 146-47; Defendants’ *Markman* Presentation, Slide 53) Specifically, Defendants point to the following excerpts from the specification as constituting these three purported definitional statements:

- “[T]he words ‘design units’ could be substituted for ‘cells[.]’” (’571 patent, col. 75:18-19);
- “Cells have views and artifacts.” (*Id.*, col. 13:58)
- “Many VHDL language constructs can be considered to meet the definition of a ‘cell’, including entities, configurations, architectures, procedures, functions, components, types, and subtypes.” (*Id.*, col. 28:57-60; *see also id.*, col. 30:20-23)

Moreover, Defendants argue that while “cell” is construable, it is indefinite due to internal inconsistencies in the specification. (D.I. 88 at 48-49; Tr. at 148) The Court will take up these four disputes (i.e., whether a “cell” is a design unit (and relatedly, whether “cell” is indefinite); whether a cell has a view and an artifact; and whether the exemplary VHDL language constructs should be included in the construction) in turn.

First, Defendants contend that a “cell” must be construed to mean, *inter alia*, a “design unit” because the specification, in a number of places, purportedly equates “cell” and “design unit.” (D.I. 88 at 48, 53; Tr. at 140) Defendants certainly have some material to work with here. Twice the specification notes that “[o]ne should understand, more generally, that the method applies to design units of data and the words ‘design units’ could be substituted for ‘cells’ in the description that follows and in the originally filed claims.” (’571 patent, col. 75:17-20; *see also id.*, col. 78:43-46) It also explains that:

Canonical digests for a file are computed by analyzing the file and categorizing sections by type. Many files have a header which may include global information about the file. *There may also be*

cells or modules, individual design units which are combined to form a design.

(*Id.*, col. 14:29-33 (emphasis added))

Yet, as Defendants acknowledge, other portions of the patent tell us that a “cell” is distinguishable from a “design unit” because a design unit can include more than just a cell. (D.I. 88 at 49, 53) This is also how Oasis compares a design unit and a cell—that is, that while the two terms “have a similar scope” because both “generally refer to portions of design data,” a design unit is a more general term that refers “to a larger object or a combination of design data with file header data.” (*Id.* at 50-51) For instance, the specification notes that “[c]anonical *cell* digests and, *more generally*, canonical *design unit* digests, are outputs of a new tool that will be useful in the IC design process.” (’571 patent, col. 5:49-51 (emphasis added)) The patent further explains that cells are something different from file headers, and a design unit can include both header data and cell data. (*Id.*, col. 7:30-31, 42-56) Claim 15 of the ’571 patent also underscores that a “design unit” is not necessarily the same thing as a cell, as it can also include header data. (*Id.*, col. 84:5-7) So too does the prosecution history, where it teaches that ““design units[]” [is] a term that encompasses the headers and cells of designs.” (D.I. 91, ex. D-1 at 29-30) In light of these disclosures, the Court is not persuaded that Oasis clearly expressed an intent to define “cell” to mean “design unit.”

The Court next takes up Defendants’ related argument that “cell” is indefinite. Here Defendants assert that this is so because the specification sometimes seems to equate “cell” with “design unit” and other times differentiates between the terms. (D.I. 88 at 49, 53; *see also* D.I. 90 at ¶¶ 103-04) According to Defendants and their expert, these “mixed messages” would leave the POSITA confused as to which definition governs and which does not. (D.I. 88 at 53)

The Court is not persuaded. Instead it agrees with Oasis and its expert that, considering the portions of the intrinsic record described above in context, the patent is telling us that: (1) a design unit is something that is more general than a cell, because it can be broader and encompass both the cell and header data; and (2) the specification passages noting that “‘design units’ could be substituted for ‘cells’ in the description that follows” means that the methods at issue can be applied to both design units and to cells. (*Id.* at 51; *see also* D.I. 92 at ¶ 29; Tr. at 152-54)¹⁷ The term “cell,” therefore, is not indefinite.

Next, with regard to Defendants’ position that a “cell” must be construed as having a view and an artifact, the Court agrees. (D.I. 88 at 48; Tr. at 142) As referenced above, the specification clearly and expressly states that “[c]ells have views and artifacts.” (’571 patent, col. 13:57) So that must be so. The patent then goes on to explain that a view “is one of the physical, functional or electrical representations of a cell” and an “artifact” is “typically a file that results from the creation of a cell view[.]” (*Id.*, col. 13:58-64)

¹⁷ During oral argument, with respect to this term, Defendants’ counsel pointed for the first time to a portion of the specification explaining that “[t]he digester can generate separate digests for *header* and body parts of a cell and generate digests by layer within a cell.” (Tr. at 147-48 (citing ’571 patent, col. 6:19-20) (emphasis added)) Counsel’s point seemed to be that: (1) here the patent is indicating that a cell includes “header” data; but (2) as noted above, sometimes the patent distinguishes cells from design units on the ground that cells do *not* include certain header data, while design units do; and (3) this is confusing, and such confusion on this point bolsters Defendants’ indefiniteness arguments. (*Id.* at 149-50) But a close look at the specification appears to provide an explanation that would help dispel any such confusion. The patent indicates that data in a cell could be divided into “*cell header . . . data*” and “*cell body data*.” (’571 patent, col. 75:13-15 (emphasis added); *see also, e.g., id.*, col. 76:36-40; *id.*, cols. 76:67-77:3) Meanwhile, as discussed above, the patent also talks about “*file header data*” that is “outside of a[] cell.” (*Id.*, col. 7:26-31 (emphasis added); *see also id.*, col. 7:42-44; *id.*, col. 10:3-9) The patent teaches that a “design unit” refers to *file header* data (i.e., the data that is outside of a cell—something different from *cell header* data) and cell data. (*Id.*, col. 7:51-53) So assuming that the “header data” in the portion of the specification cited by Defendants’ counsel at argument refers to *cell header* data, then this excerpt from the patent would not support Defendants’ indefiniteness argument.

Oasis argues against this inclusion by asserting that artifacts, views and cells are “different concepts” and that an artifact and a view can be distinct objects from the cell itself. (D.I. 88 at 45-46; Tr. at 128) Indeed, Oasis asserts that just because a cell could be associated with an artifact and/or a view “does not mean that every cell must ‘have’ or ‘contain’ both.” (D.I. 88 at 50) Yet with regard to Oasis’ point that artifacts and views might not constitute the cell itself, in the Court’s view, requiring a cell to “have” an artifact and a view allows for this. After all, it is not as if Defendants are proposing that a “cell” be construed to mean “a cell is a view” or “a cell is an artifact.” (*Id.* at 48 (“That the artifact may be a separate attribute of a ‘cell’ . . . does not mean it is optional.”))¹⁸

Finally, with respect to the dispute regarding whether the construction of “cell” should include Defendants’ exemplary language, the Court sides with Oasis. Defendants acknowledge that “cells” are not required to be VHDL constructs. (*Id.*) The patent clearly tells us that the invention is not limited to VHDL language and “can readily be extended to other formats” including OASIS, GDSII, Verilog and Cadence Library Exchange Format. (’571 patent, col. 7:7-22; *see also* D.I. 88 at 45) Oasis argues that in light of this, “there is no . . . reason to include in the construction Defendants’ now open-ended list of possible types of cells.” (D.I. 88 at 51) The Court agrees that Defendants did not really explain why this proposed language should be included in any construction of “cell”; they did not, for example, make a case that this language

¹⁸ During oral argument, Oasis’ counsel suggested that it would be wrong to require a “cell” to *have* an artifact and a view because, for a cell to exist, it does not *need* to have a view, and it does not *need* to have an artifact—in fact, an artifact will only be generated if a cell is viewed. (Tr. at 133-35) In other words, Oasis seems to be saying that while a cell is *capable of having* a view and an artifact, it need not actually *have* these attributes to exist. That may be so, and here the Court will not attempt to further interpret what it means for a cell to “hav[e]” an artifact and a view. In line with the patent, the Court will require cells to have views and artifacts, whatever it is the patentee meant by that (which may be a fight for a different day).

would be helpful to the jury in some way. (*Id.* at 48, 52) Thus, the Court has not been persuaded that it must be adopted. *See, e.g., Keithley v. Homestore.com, Inc.*, No. C03-04447 MJJ, 2007 WL 2701337, at *8 (N.D. Cal. Sept. 12, 2007) (declining to include an exemplary list in a construction where the defendant failed to explain why it should be included).

With the parties' disputes regarding "cell" now resolved, the Court turns to the appropriate construction for the term. The Court will largely adopt Oasis' proposed construction with additional language to make clear that a cell has a view and an artifact. Defendants posit that the plain and ordinary meaning of a "cell" is "[a] collection (or 'unit') of design elements that together perform a function[,]" (D.I. 88 at 6), which is similar to at least the beginning of Oasis' construction (i.e., "a subset of design data"). And in their briefing, Defendants do not dispute that a cell *can* contain objects, properties and comments. (*Id.* at 52; *see also* '571 patent, cols. 7:32-34, 8:23-29, 32:24-27)¹⁹

For these reasons, "cell" should be construed to mean "a subset of design data, having a view and an artifact, that can be referenced as a whole representation and expression of: an object, properties, or comments."

D. "normalizing the design data within the [cells / design units] into canonical forms" and "normalizer logic [. . .] cooperating with the parser that organizes the syntax trees to produce canonical forms"

The next disputed term set includes two terms: "normalizing the design data within the [cells / design units] into canonical forms" and "normalizer logic [. . .] cooperating with the parser that organizes the syntax trees to produce canonical forms" (together, the "normalizer

¹⁹ Oasis' proposed construction also indicated that a cell could be an expression of a set of instructions. (D.I. 103, ex. 2 at 2) However, it did not show where the specification supported this inclusion. (*See* D.I. 88 at 49-50 (Oasis showing where the specification indicates that a cell can contain, in addition to design language, objects, properties and comments)) Thus, the Court does not include this language in its construction.

logic term”), which are found in, *inter alia*, claims 1, 4-6, 8-11 and 14 of the '545 patent and claims 1, 15 and 16 of the '571 patent. With respect to the first term, as seen above, exemplary claim 1 of the '545 patent recites a method of evaluating similarities and/or differences between design data for circuits that includes, *inter alia*, “parsing syntax of and *normalizing the design data within the cells into canonical forms*, wherein the canonical forms reduce sensitivity of data analysis to non-functional variations in the design data within a particular cell[.]” ('545 patent, col. 81:63-67 (emphasis added)) And with respect to the second term, exemplary claim 16 of the '571 patent recites:

16. An article of manufacture including:

a non-transitory computer readable storage medium that stores program code configured to be run on a computer and to evaluate similarities and/or differences between design data for circuits, the design data residing in at least two files stored in computer memory, the program code including:

a parser that parses a file containing design data representing aspects of a design for a physical circuit and creates one or more syntax trees in the memory;

normalizer logic cooperating with the parser that organizes the syntax trees to produce canonical forms, wherein the normalizer logic includes:

a partitioning module that partitions the file into at least one header and, depending on rules of a design language used to encode the file, into multiple cells of design data and organizes the syntax trees to represent the header and cell partitions;

a canonical forming module that interprets the syntax trees to produce canonical forms of the design data, wherein the canonical forms reduce sensitivity of data analysis to non-functional variations in the design data;

a digester that receives the canonical forms for at least selected partitions and calculates and stores in the memory at least one digest per selected partition;

a comparer module that receives and compares the digests of at least a first file and a second file, which contain design data; and

a reporter module coupled to the digester that summarizes at least some of the matches and/or differences detected by the comparisons of digests.

('571 patent, col. 84:24-56 (emphasis added)) The parties' competing proposed constructions for the normalizer logic term are set out in the chart below:

Term	Plaintiff's Proposed Construction	Defendants' Proposed Construction
<p>"normalizing the design data within the [cells / design units] into canonical forms"</p> <p>"normalizer logic [. . .] cooperating with the parser that organizes the syntax trees to produce canonical forms"</p>	<p>See proposed constructions for "canonical forms" and "cells."</p> <p>For the remainder of the term: plain and ordinary meaning/no construction required.</p> <p>In the alternative, the plain and ordinary meaning of the term "normalizing the [cells / design units] design data into canonical forms" could be articulated as "organizing data into a standardized form as part of the generation of canonical forms."</p> <p>In the alternative, the plain and ordinary meaning of the term "normalizer logic [. . .] cooperating with the parser that organizes the syntax trees to produce canonical forms" could be articulated as "organizing data into a rule-based hierarchical form as part of the generation of canonical forms."</p> <p>This limitation is not subject to 35 U.S.C. 112(6). In the</p>	<p>Sec. 112/6.</p> <p><u>Function</u>: "generating a canonical form"</p> <p><u>Structures</u>: "the techniques described at '545 [patent] 71:22-25, 75:14-76:4, 78:59-79:15 or '571 [patent] 71:22-25, 75:2-59, 78:47- 79:3"</p> <p>Dispute additional structures proposed by Plaintiff</p>

	<p>alternative, to the extent 112(6) applies:</p> <p><u>Function</u>: “generating a canonical form”</p> <p><u>Structures</u>: Elements 533 in FIG. 5 and 613 in FIG. 6 of the Asserted Patents, the techniques described at '571 Patent 6:8-17, 69:20-24; 71:20-25; 75:1-59; 78:47-79:14 and '545 Patent 69:22-26; 71:20-25; 75:13-76:4; 78:59-79:15.</p>	
--	---	--

(D.I. 103, ex. 2 at 2-3) The primary dispute regarding the normalizer logic term—and a challenging dispute at that—is whether it is a means-plus-function term subject to 35 U.S.C. § 112(6).²⁰ (D.I. 88 at 64, 67; Tr. at 160)

²⁰ The parties agree that the pre-AIA version of 35 U.S.C. § 112 applies. Section 112(6) provided that “[a]n element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.” 35 U.S.C. § 112(6). The “means-plus-function” technique of claim drafting is a “convenience” that allows a patentee to express a claim limitation in functional terms “without requiring the patentee to recite in the claims all possible structures” that could perform that function. *Med. Instrumentation & Diagnostics Corp. v. Elekta AB*, 344 F.3d 1205, 1211 (Fed. Cir. 2003) (internal quotation marks and citation omitted). In exchange for getting the benefit of this drafting convenience, however, patentees must disclose, in the written description of the patent, a corresponding structure for performing the claimed function. *Noah Sys, Inc. v. Intuit Inc.*, 675 F.3d 1302, 1318 (Fed. Cir. 2012); *see also Elekta*, 344 F.3d at 1211 (“[T]he price that must be paid for use of that convenience is limitation of the claim to the means specified in the written description and equivalents thereof.”) (internal quotation marks and citation omitted). A patentee satisfies this requirement “only if the specification or prosecution history *clearly links or associates* that structure to the function recited in the claim.” *In re Aoyama*, 656 F.3d 1293, 1297 (Fed. Cir. 2011) (emphasis added) (quoting *Elekta*, 344 F.3d at 1210); *see also Elekta*, 344 F.3d at 1220. “If the specification does not contain an adequate disclosure of the structure that corresponds to the claimed function, the patentee will have failed to particularly point out and distinctly claim the invention as required by . . . section 112[], which renders the claim invalid for indefiniteness.” *Blackboard, Inc. v. Desire2Learn Inc.*, 574 F.3d 1371, 1382 (Fed. Cir. 2009) (internal quotation marks and citation omitted).

Given that the claims do not use the traditional “means” language often found in means-plus-function claims, “there is a rebuttable presumption that [Section 112(6)] does not apply” as to the normalizer logic term. *Diebold Nixdorf, Inc. v. Int’l Trade Comm’n*, 899 F.3d 1291, 1298 (Fed. Cir. 2018). “[T]he presumption can be overcome and [Section 112(6)] will apply if the challenger [here Defendants] demonstrates that the claim term fails to recite sufficiently definite structure or else recites function without reciting sufficient structure for performing that function.” *Id.* (internal quotation marks and citations omitted). The challenger must meet this burden by a preponderance of the evidence. *See Apex Inc. v. Raritan Comput., Inc.*, 325 F.3d 1364, 1372 (Fed. Cir. 2003). “To determine whether a claim recites sufficient structure, it is sufficient if the claim term is used in common parlance or by [POSITAs] to designate structure, even if the term covers a broad class of structures and even if the term identifies the structures by their function.” *Skky, Inc. v. MindGeek, s.a.r.l.*, 859 F.3d 1014, 1019 (Fed. Cir. 2017) (internal quotation marks and citation omitted).

1. “normalizer logic [. . .] cooperating with the parser that organizes the syntax trees to produce canonical forms”

The Court first assesses the normalizer logic term as it appears in claim 16 of the '571 patent. Defendants argue that the term fails to recite sufficient and definite structure to perform the claimed function, which triggers Section 112(6). (D.I. 88 at 66-69) As seen above, claim 16 recites the normalizer logic term as an element of “a non-transitory computer readable medium that stores program code configured to be run on a computer[.]” ('571 patent, col. 84:25-26) It is

Construing a means-plus-function limitation is a two-step process. The first step is determining the claimed function of the limitation. *Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1351 (Fed. Cir. 2015); *Medtronic, Inc. v. Advanced Cardiovascular Sys., Inc.*, 248 F.3d 1303, 1311 (Fed. Cir. 2001). The second step is identifying the corresponding structure disclosed in the specification and equivalents thereof. *Williamson*, 792 F.3d at 1351; *Medtronic, Inc.*, 248 F.3d at 1311.

not in dispute that the normalizer logic term performs the function of generating a canonical form. (D.I. 88 at 50-61, 66) Below, the Court will first describe the state of the law for assessing how Section 112(6) applies to claim limitations directed to software; it will then explain why Defendants have not met their burden to demonstrate, by a preponderance of the evidence, that a POSITA would not associate a sufficiently definite structure with “normalizer logic” for performing the claimed function (i.e., generating a canonical form).

How can a claim limitation for computer “code” provide sufficient *structure* for performing the claimed function to a POSITA? *See XR Commc’ns, LLC v. ARRIS Sols., Inc.*, 2022-1125, 2022-1141, 2023 WL 3529830, at *2 (Fed. Cir. May 18, 2023) (explaining that the proper question to ask in determining whether a claim term has invoked Section 112(6) is “whether a POSITA would understand the disputed term not just as structure, but as sufficient structure for performing the claimed function”) (internal quotation marks, brackets and citation omitted). Unlike in the mechanical arts, a search for traditional “physical structure” in a claim to computer software is not the proper inquiry, because software is not made up of physical structures. *Apple Inc. v. Motorola, Inc.*, 757 F.3d 1286, 1298 (Fed. Cir. 2014), *overruled on other grounds by Williamson v. Citrix Online, LLC*, 792 F.3d 1339 (Fed. Cir. 2015);²¹ *see also Dyfan, LLC v. Target Corp.*, 28 F.4th 1360, 1368 (Fed. Cir. 2022). Instead, the structure of software “is understood through, for example, an outline of an algorithm, a flowchart, or a specific set of instructions or rules.” *Apple*, 757 F.3d at 1298; *see also, e.g., M2M Sols., LLC v.*

²¹ While *Williamson* overruled *Apple* in altering the strength of the presumption that applies when a claim term does not use “means,” “*Apple* remains on point for computer-implemented inventions that do not employ means-plus-function language.” *Sci. Telecommc’ns, LLC v. Adtran, Inc.*, Civ. No. 15-647-SLR, 2016 WL 6872311, at *3 n.16 (D. Del. Nov. 21, 2016).

Sierra Wireless Am., Inc., Civil Action No. 14-cv-01102-RGA, Civil Action No. 14-cv-01103-RGA, 2019 WL 6328119, at *4 (D. Del. Nov. 26, 2019).

More specifically, a claim limitation to code has sufficient structure if, read in light of the specification, it: (1) recites a claim term with a structural definition that is generally known in the art; or (2) recites a known or generic term along with a sufficient description of the limitation's operation, such as its input, output or connections. *Apple*, 757 F.3d at 1296-97, 1299-300. A claim limitation's operation is not simply the limitation's function; rather, it is "*how the function is achieved* in the context of the invention." *Id.* at 1299 (emphasis added); *see also, e.g., Rain Computing, Inc. v. Samsung Elecs. Am., Inc.*, 989 F.3d 1002, 1006 (Fed. Cir. 2021) (concluding that "user identification module" being configured to control access was subject to Section 112(6), where the term had no commonly understood meaning and the claim language and specification failed to provide any structure for performing the claimed function); *M2M Sols., LLC*, 2019 WL 6328119, at *5 (holding that Section 112(6) applied to a "processing module" limitation with functions of (1) determining a change in status and (2) determining whether something otherwise indicates an alarm condition, where the surrounding claim language failed to "describe *how* the processing module carries out" the claimed functions) (emphasis added). Accordingly, if a patent indicates that a claim limitation for computer code

refers to existing code known in the art,²² or alternatively, “disclose[s] what that code is[,]”²³ it will avoid application of Section 112(6). *See Tracktime, LLC v. Amazon.com, Inc.*, C.A. No. 18-1518 (MN), 2021 WL 2823163, at *6 (D. Del. July 7, 2021). Such a claim limitation would not amount to the type of term wherein the applicant sought to “capture any possible means for achieving [an] end.” *Noah Sys., Inc. v. Intuit Inc.*, 675 F.3d 1302, 1317 (Fed. Cir. 2012); *see also M2M Sols. LLC v. Sierra Wireless Am., Inc.*, Civil Action No. 12-30-RGA, Civil Action No. 12-32-RGA, Civil Action No. 12-33-RGA, 2015 WL 5826816, at *5 (D. Del. Oct. 2, 2015).

Here, the evidence does not demonstrate that “normalizer logic” has a structural definition generally known to a POSITA. (D.I. 88 at 67) Defendants’ expert opines that the term does not have a “definite structural meaning” in the field of electronic design automation (“EDA”)—that is, that it does not “refer to a limited set of existing, conventional software offerings sharing a mostly common set of software structures.” (D.I. 90 at ¶ 131) Rather, a POSITA would understand “normalizer logic” as a generic, functional term that refers to *any* software or hardware performing the function of normalizing and creating canonical forms. (*Id.* at ¶ 132) Indeed, Oasis does not attempt to argue that “normalizer logic” has a known structural

²² *See also, e.g., Dyfan, LLC*, 28 F.4th at 1369 (“[B]ecause the recited functions can be performed by conventional off-the-shelf software, a person of ordinary skill in the art would have understood the alleged means-plus-function ‘code’ limitations in the asserted claims to connote structure.”); *Zeroclick, LLC v. Apple Inc.*, 891 F.3d 1003, 1008 (Fed. Cir. 2018) (finding that the code limitations at issue did not trigger Section 112(6) because they “are not used as generic terms or black box recitations of structure or abstractions, but rather as specific references to conventional graphic user interface programs or code, existing in prior art at the time of the inventions”).

²³ *See e.g., M2M Sols. LLC v. Sierra Wireless Am., Inc.*, Civil Action No. 12-30-RGA, Civil Action No. 12-32-RGA, Civil Action No. 12-33-RGA, 2015 WL 5826816, at *3-5 (D. Del. Oct. 2, 2015) (finding that the claim term “processing module” with the recited function of authenticating a received incoming transmission was not subject to Section 112(6), where the surrounding claim language expressly conveyed algorithmic structure for how to accomplish this function).

meaning to a POSITA.²⁴ In light of this, without more, the normalizer logic term would be subject to Section 112(6). This is so even if a POSITA would know how to program a computer to perform the function at issue, because “the fact that one of skill in the art could program a computer to perform the recited functions cannot create structure where none otherwise *is disclosed*.” *Williamson*, 792 F.3d at 1351 (emphasis added); *see also, e.g., WSOU Invs., LLC v. Xilinx, Inc.*, C.A. No. 20-1228-CFC-JLH (Consolidated), 2022 WL 2093066, at *6 (D. Del. June 10, 2022), *report and recommendation adopted*, 2022 WL 16707078 (D. Del. Nov. 4, 2022).²⁵

Nevertheless, Oasis argues that Section 112(6) does not apply here because the claim language, read in light of the specification, otherwise sufficiently identifies the claim limitation’s operation—including its relationship to other elements—so as to connote specific, sufficient structure to a POSITA. (D.I. 88 at 62-65, 72-73; Tr. at 189-90) As discussed above, in order to sufficiently describe the operation of a challenged claim term, the patent cannot simply parrot the claim term’s function (here, “generating a canonical form”). Rather, there must be sufficient description of *how* generating a canonical form²⁶ *is achieved* in the context of the invention.

²⁴ In that sense, the facts here are different than those at issue in cases like *Dyfan* and *Zeroclick*, *see supra* n.22, where the record demonstrated that the limitations at issue were references to known, conventional computer code. (See D.I. 88 at 69-70; Tr. at 168-69)

²⁵ At a few points in its briefing and during oral argument, Oasis seemed to suggest that if you have a claim limitation directed to computer code, and a POSITA would know how to program the code to perform the recited function, then that claim limitation is not subject to Section 112(6)—full stop. (D.I. 88 at 62; Tr. at 185-88, 190 (“[W]hen a programmer is looking at this, one of skill in the art, and they see code operating on a processor that has normalizing logic . . . the programmer knows exactly what that is and what it’s supposed to be doing.”)) That is not correct under the law. That a POSITA would know how to program code alone does not constitute sufficient structure to save a claim limitation from Section 112(6).

²⁶ As discussed above, “canonical forms” will be construed to mean “a standardized form of a body of design data.” *See supra* at 15.

So what does Oasis point to in this regard? It turns first to the language in claim 16 itself. There it notes that in the claim, “normalizer logic” is described as “cooperating with the parser that organizes the syntax trees to produce canonical forms[.]” (D.I. 88 at 63 (quoting '571 patent, col. 84:35-36 (emphasis omitted))) And Oasis asserts that from this, a POSITA would understand “normalizer logic” to mean “commands or statements in a computer program or routine that can be directly executed by a processor to perform the recited objective of cooperating with the parser to organize syntax trees to produce the canonical forms.” (*Id.*) In other words, Oasis seems to be saying that this surrounding claim language helps provide sufficient structure to the normalizer logic limitation, in that the language explains how the normalizer logic works together with the parser to generate a canonical form. (*Id.*)

Next, the language that comes after the normalizer logic claim term also provides a bit more structural detail. It specifies that the normalizer logic includes: (1) a *partitioning module* that “partitions the file into at least one header and, depending on rules of a design language used to encode the file, into multiple cells of design data and organizes the syntax trees to represent the header and cell partitions;” and (2) a *canonical forming module* that “interprets the syntax trees to produce canonical forms of the design data.” ('571 patent, col. 84:37-45)

Other sources amplify this description of how the normalizer logic is structured and how it operates to generate canonical forms. For example, the specification further underscores that the normalizer logic: (1) is computer code; that (2) generates canonical forms by cooperating with the parser that organizes syntax trees. (*Id.*, col. 78:47-52) And Oasis points to the declaration of its expert, Dr. Stephan Athan, for further support. Dr. Athan opines that based on the language of claim 16, a POSITA would understand that normalizer logic is “software code that, cooperating with the parser, organizes syntax trees to produce canonical forms [which]

includes organizing data into a rule-based hierarchical form as part of the generation of canonical forms[.]” (D.I. 89 at ¶ 74) In a follow-up declaration, Dr. Athan further opines that the specification sets out specific structural requirements for the normalizer logic that “provide substantial structural certainty” to a POSITA, including that it: (1) runs on a processor; (2) cooperates with the parser, which organizes syntax trees, to produce canonical forms; (3) includes a partitioning module that partitions the file into at least one header and organizes the syntax trees to represent the header and cell partitions; (4) includes a canonical forming module that interprets the syntax trees to produce canonical forms of design data; and that (5) these canonical forms reduce the sensitivity of data analysis to non-functional variations in the design. (D.I. 92 at ¶ 32)

In light of the above, this is not a circumstance like that in *Williamson*, for example, where the claim said *nothing* about *how* the distributed learning control model performed the recited functions of (1) receiving communications transmitted between computer systems and (2) relaying the communications to an intended receiving computer system and (3) coordinating the operations of the streaming data module. 792 F.3d at 1344, 1351.²⁷ Instead, here the patent does not leave us completely guessing as to how canonical forms are generated. Claim 16’s language (read in light of the specification and the presented extrinsic evidence), sufficiently explains: (1) what algorithmic components the normalizer logic must “include” in order to perform the

²⁷ Cf. *Tracktime, LLC*, 2021 WL 2823163, at *5-6 (agreeing with the defendants that certain “executable program code” limitations were subject to Section 112(6), where “the claims are silent as to *how* [the executable program code] achieve[s] the claimed functions”) (internal quotation marks omitted, emphasis in original); *M2M Sols., LLC*, 2019 WL 6328119, at *5 (“The words immediately following ‘processing module’ in claims 28 and 30 do not provide algorithmic structure as they do not describe *how* the processing module carries out the ‘change in status’ or ‘alarm condition’ determinations. . . . Therefore, [Section] 112[(6)] applies.”) (emphasis added).

function of generating canonical forms (i.e., a partitioning module and a canonical forming module); (2) how those components do their work; and (3) what other claim elements the normalizer logic must work with in this regard (i.e., the parser). (D.I. 88 at 64, 73-74; *see also* '571 patent, cols. 78:47-52, 84:35-46; *id.*, FIG. 5) Figure 5 of the patent, as well as the specification, further makes it clear that canonical forms are produced by organizing syntax trees. (*Id.*, FIG. 5; *id.*, col. 78:47-52) So the patent does not seem to be attempting to tie up *any* possible means for generating a canonical form.²⁸

The Court thus concludes that Defendants have not met their burden²⁹ to demonstrate, by a preponderance of the evidence, that claim 16 of the '571 patent (and the other applicable claims) fail to recite sufficiently definite structure for the normalizer logic term. *See M2M Sols., LLC*, 2019 WL 6328119, at *4 (concluding that the use of “processing module” in certain claims did not implicate Section 112(6), where the claims contained additional language informing the reader how the processing module performed the recited function (i.e., authenticating one or

²⁸ If the only way to generate a canonical form is by organizing and interpreting syntax trees, this has not been made clear to the Court in this record.

²⁹ In their briefing, Defendants make it seem as if the patent would allow for “*whatever* software structure can perform the function.” (D.I. 88 at 69 (emphasis in original); Tr. at 164, 167-68 (“If it [can be] any software [as it is with respect to the terms at issue here], if it’s any code to do the function without specifying how, without specifying particular algorithms, then you have a purely functional claim”)) Similarly, Defendants’ expert, Dr. Sarrafzadeh, opines that the patent uses the normalizing logic term “generically to refer to any software that can perform the function.” (D.I. 90 at ¶ 135; *see also id.* at ¶ 136 (opining that the patent defines “‘normalizer logic’ as *any* set of instructions in software . . . that can perform the function of normalizing without restricting it to a particular class of software structures”) (emphasis in original)) As for the patent’s teaching that the normalizer logic generates canonical forms by cooperating with the parser and by organizing the syntax trees and interpreting syntax trees, Dr. Sarrafzadeh opines that these are just “functions[—]not definite structures for performing the function” and that the claims fail to tell us how “organizing the syntax trees” is actually performed. (*Id.* at ¶ 142) For the reasons set forth above, however, the Court disagrees with Defendants as to the state of the record here, and it disagrees that this record is sufficient to rebut the presumption against means-plus-function claiming.

more wireless transmissions sent from a programming transmitter and received by the programmable communicator device), in that they explained that this occurred by “determining if at least one transmission contains a coded number”) (internal quotation marks and citations omitted). Defendants present no alternative proposed construction, and so no further construction is necessary.

2. “normalizing the design data within the [cells / design units] into canonical forms”

The Court next assesses the normalizer logic term as it appears in representative claim 1 of the '545 patent. This is a method claim that entails, *inter alia*, “parsing syntax of and *normalizing the design data within the cells into canonical forms*, wherein the canonical forms reduce sensitivity of data analysis to non-functional variations in the design data within a particular cell[.]” ('545 patent, col. 81:63-67 (emphasis added))

For method claims, Section 112(6) is implicated only when a claim element therein recites a step for performing a specified function without the recital of acts in support of the function. *O.I. Corp. v. Tekmar Co.*, 115 F.3d 1576, 1582-83 (Fed. Cir. 1997); *Techno View IP, Inc. v. Facebook Techs., LLC*, Civil Action No. 17-386-CFC-CJB, 2018 WL 5077898, at *7 (D. Del. Oct. 18, 2018). Unless method claims include “step for” type language, such claims are presumed not to be step-plus-function terms. *Epcon Gas Sys., Inc. v. Bauer Compressors, Inc.*, 279 F.3d 1022, 1028 (Fed. Cir. 2002); *see also, e.g., Techno View IP, Inc.*, 2018 WL 5077898, at *8. Here, the claim does not recite “steps for,” and so Defendants must make a showing that the limitation contains nothing that can be construed as an act in order for Section 112(6) to be implicated. *Techno View IP, Inc.*, 2018 WL 5077898, at *8.

The parties did not provide much in the way of additional analysis regarding this method claim term. Defendants assert that the term is subject to Section 112(6) because “normalizing”

identifies a function without also reciting particular acts sufficient to perform that function. (D.I. 88 at 68) Oasis’ response is that the method claims at issue do not include “step for” type language, and they “expressly recite the required *action*—converting the design data into canonical forms—such that the term is outside the ambit of [Section] 112(6).” (*Id.* at 65 (emphasis added))

Here, the Court agrees with Oasis. Defendants have not overcome the presumption that Section 112(6) does not apply. The term “normalizing” amounts to a specific act that itself describes how the method organizes design data within the cells into canonical forms. *See, e.g., Serrano v. Telular Corp.*, 111 F.3d 1578, 1580, 1583 (Fed. Cir. 1997) (claim limitation “automatically determining at least the last-dialed number of the telephone number dialed on the telephone communications-type device” was not subject to Section 112(6), because it did not recite a function, but “recites only the act of determining a last-dialed digit”); *Fraunhofer-Gesellschaft Zur Förderung der angewandten Forschung e.V. v. Sirius XM Radio Inc.*, Civil Action No. 17-184-JFB-SRF, 2020 WL 549801, at *7 (D. Del. Feb. 4, 2020) (finding that the limitation “the step of transmitting being carried out by a first transmitter and a second transmitter spaced part from the first transmitter” was not subject to Section 112(6), because it “conveys the affirmative act of ‘transmitting’”), *report and recommendation adopted*, 2020 WL 1969508 (D. Del. Apr. 24, 2020); *VPS, LLC v. SmugMug, Inc.*, No. 10 CV 2142, 2012 WL 5471012, at *18 (N.D. Ill. Nov. 9, 2012) (finding that the claim limitation “storing a high resolution and a low resolution copy of each of a first [second] plurality of digital images provided by a first [second] asset [image] provider user in an electronically searchable format on a storage device” did not implicate Section 112(6), because “[t]he claim element is clear that the step of *storing a high resolution and a low resolution copy* is an action”) (emphasis in original);

Metraflex Co. v. Flex-Hose Co., No. 10 C 302, 2011 WL 4001144, at *6-7 (N.D. Ill. Sept. 8, 2011) (rejecting the defendant’s argument that a step-plus function analysis should apply to “generating a composite image of said selected product” where “the term ‘generating,’ in the sense of causing a composite image to be displayed from stored images of individual permutations, contains an act because it describes how the function of generating a composite image occurs”). The specification provides some further information as to the act of normalizing, when it teaches that “[s]ome design data files are normalized or transformed into a canonical format by passing them through a parser and applying parsing rules. Other files may require semantic analysis of a syntax tree generated by the parsing or other manipulations to normalize the file.” (’571 patent, col. 75:5-9); *see also, e.g., Metraflex Co.*, 2011 WL 4001144, at *6-7 (noting that the specification provided additional description of how the act of generating a composite image is accomplished).

The Court thus concludes that the term “normalizing the design data within the cells into canonical forms” is not subject to Section 112(6). Defendants present no alternative proposed construction, so no further construction is necessary.

E. “a comparer module [. . .] that receives and compares the digests of at least a first file and a second file, which contain design data”

The next disputed term, “a comparer module [. . .] that receives and compares the digests of at least a first file and a second file, which contain design data” (the “comparer module”) is found in, *inter alia*, claim 14 of the ’545 patent and claim 16 of the ’571 patent. Its use in claim 16 of the ’571 patent, recited above, is representative. The parties’ competing proposed constructions for the comparer module term are set out in the chart below:

Term	Plaintiff's Proposed Construction	Defendants' Proposed Construction
<p>“a comparer module [. . .] that receives and compares the digests of at least a first file and a second file, which contain design data”</p>	<p>See proposed constructions for “digest(s)” and “cell(s).”</p> <p>For the remainder of the term: plain and ordinary meaning/no construction required.</p> <p>This limitation is not subject to 35 U.S.C. 112(6). In the alternative, to the extent 112(6) applies:</p> <p><u>Function</u>: “comparing digests of [cells / design units] in the first file to digests of [cells / design units] in the second file”</p> <p><u>Structures</u>: Elements 536 in FIG. 5 and 615 in FIG. 6 of the Asserted Patents, the techniques described at '571 Patent 6:36-56; 56:20-59:10; 59:58-62:44; 63:42-65:60; 68:55-69:13; 70:13-21; 70:43-53; 71:20-23; 75:63-78:23; 79:7-42 and '545 Patent 6:27-6:47; 56:23-59:10; 59:58-62:44; 63:42-65:60; 68:56-69:15; 70:15-23; 70:45-55; 71:20-25; 76:8-78:34; 79:19-54.</p>	<p>Sec. 112/6.</p> <p><u>Function</u>: “comparing digests of [cells / design units] in the first file to digests of [cells / design units] in the second file to detect exact matches, partial matches, or differences”</p> <p><u>Structures</u>: “the techniques described at '571 [patent] 76:1-76:27, 79:15-42, or '545 [patent] 76:13-76:39, 79:27-54”</p> <p>Dispute additional structures proposed by Plaintiff</p>

(D.I. 103, ex. 2 at 4) The parties’ initial dispute here is the same as it was for the normalizer logic term—is this a means-plus-function term subject to Section 112(6)? (D.I. 88 at 65, 83; Tr. at 184)

In representative claim 16 of the '571 patent, the comparer module term is a limitation to “program code” and it recites the function of comparing “the digests of at least a first file and a

second file, which contain design data.” (’571 patent, col. 84:51-53) This claim term does not use the word “means,” and so this triggers a rebuttable presumption that Section 112(6) does not apply. The term “module[,]” however, is a “well-known nonce word that can operate as a substitute for ‘means’ in the context of” Section 112(6). *Williamson*, 792 F.3d at 1350. The inquiry here is whether a POSITA would understand the comparer module term as sufficient structure for performing the claimed function (i.e., comparing the digests). *XR Commc’ns, LLC*, 2023 WL 3529830, at *2. And again, we assess this by looking to whether the comparer module, read in light of the specification: (1) has a structural definition that is generally known in the art; or (2) constitutes a known or generic term along with a sufficient description of how the function is achieved in the context of the invention. *See, e.g., Apple*, 757 F.3d at 1299-1300; *Rain Computing, Inc.*, 989 F.3d at 1006.

Oasis only provides one argument specific to this term as to why it is not subject to Section 112(6). And that argument is an incorrect one. Oasis asserts that because the comparer module is program code, a POSITA “would understand the plain language of this element to describe software code that ‘compares digests of cells’ from a ‘first and second file[]’ [which] has a specific meaning to a POSITA, who would know how to replicate this straightforward function in software.” (D.I. 88 at 83 (citing D.I. 89 at ¶ 81)) But as discussed above, if the claim term at issue, read in light of the specification, does not connote sufficient structure for performing the claimed function, then the term is subject to Section 112(6), even if a POSITA “could program a computer to perform the recited function[]”; this is because the POSITA’s knowledge “cannot create structure where none otherwise is disclosed.” *Williamson*, 792 F.3d at 1351; *see also, e.g., M2M Sols, LLC*, 2019 WL 6328119, at *5.

The Court concludes that comparer module invokes Section 112(6). No one is arguing here that a “comparer module” has a structural definition that is generally known to a POSITA. Indeed, on that score, Dr. Sarrafzadeh opines that the term is one with “no definite structural meaning in the EDA arts.” (D.I. 90 at ¶ 134) Therefore, in order to avoid application of Section 112(6), there must be a sufficient description in the claim of *how* the comparer module achieves the claimed function (i.e., compares the digests of at least a first file and a second file). Oasis’ briefing is silent on this point—it makes no attempt to explain how the comparer module accomplishes this task, other than to simply posit that the POSITA would know how to replicate this function in software. (D.I. 88 at 83) The claim language is not helpful either, as it does not say anything about how the comparer module does its work. (*See* '571 patent, col. 84:51-53) Nor does the single portion of the specification that Oasis highlights as to this issue; there, the specification simply reiterates that “[a] comparer module [] runs on a processor [] and compares the digests of canonical forms.” (*Id.*, col. 79:7-8) Subsequently, the specification acknowledges that “[t]here are *multiple* ways in which the comparer can be structured.” (*Id.*, col. 79:15-16 (emphasis added)) Therefore, in light of this record, it is clear that the claim does not recite to a POSITA sufficient structure to perform the claimed function. (D.I. 88 at 66-67, 76); *see also*, e.g., *Avocent Huntsville, LLC v. ZPE Sys., Inc.*, Case No. 17-cv-04319-WHO, 2018 WL 4677437, at *10 (N.D. Cal. Aug. 23, 2018) (“The claims merely recite the function[] of the ‘management module’ . . . without connoting sufficiently definite structure to convey to one skilled in the art how to accomplish th[is] function[].”).

Next, the Court must: (1) determine the claimed function of the comparer module; and (2) identify the corresponding structure disclosed in the specification. *Williamson*, 792 F.3d at 1351. The parties have disputes with respect to both of these steps. (*See* D.I. 88 at 84)

As for the claimed function, the parties agree on most of the language: “comparing digests of [cells/design units] in the first file to digests of [cells/design units] in the second file[.]” (D.I. 103, ex. 2 at 4) Defendants, however, further contend that the comparer module compares digests “to detect exact matches, partial matches, or differences” and that such language should also be included in the function. (*Id.*; *see also* D.I. 88 at 84-85)

The Court will not adopt Defendants’ additional function language. Defendants contend that the language is supported by the preamble of the claims,³⁰ which tells us that the invention “evaluate[s] similarities and/or differences between design data for circuits[.]” (’571 patent, col. 84:27-28; ’545 patent, col. 85:40-41; *see also* D.I. 88 at 84-85; Tr. at 176) According to Defendants, the specification teaches that such similarities and/or differences between data are identified by the comparer module detecting exact matches or partial matches between the data. (D.I. 88 at 85 (citing ’571 patent, col. 76:21-24 (“When there is no complete match between the cell of interest and any of the candidate cells, a threshold or ratio may be applied for a number of digest matches that causes a pair of cells to be considered similar.”))) Oasis retorts that the preamble does not support Defendants’ “overly rigid” additional language, and that the claims encompass a “more flexible, scoring-based analysis.” (*Id.*; *see also* ’571 patent, col. 76:1-12 (noting that “[t]here are multiple ways in which [the claimed] comparison can be performed” including by “comparing and scoring multiple digests”))

Section 112(6) does not permit limitation of a means-plus-function claim “by adopting a function different from that explicitly recited in the claim.” *Generation II Orthotics Inc. v. Med. Tech. Inc.*, 263 F.3d 1356, 1363 (Fed. Cir. 2001) (internal quotation marks and citation omitted)). It does not appear that the patent ever even expressly recites Defendants’ proposed additional

³⁰ The parties apparently agree that the preamble is limiting. (Tr. at 176)

language, and Defendants have not otherwise convinced the Court that it would be appropriate to append this additional verbiage onto the functional description. The claim elsewhere already makes clear that the purpose of the claimed invention is evaluating similarities and/or differences in design data. And Defendants have not pointed the Court to any place in the specification that says that the only way this can be done is to “detect exact matches, partial matches, or differences”—whatever that means, exactly. (See D.I. 89 at ¶ 82 (Oasis’ expert opining that evaluating similarities and/or differences between design data is not *limited* to only exact or partial matches)) The Court therefore adopts Oasis’ proposed construction as to the function.

Lastly, the Court must identify the relevant corresponding structure that can perform the claimed function. Here, Defendants cite to several portions of the specification referencing such structure—citations that Oasis does not dispute.³¹ (See D.I. 103, ex. 2 at 4) Oasis, however, identifies several additional sections of the specification that it contends also constitute structure for the comparer module term.

The Court, however, agrees with Defendants, (D.I. 88 at 84, 86), that a great many of the disputed citations do not provide meaningful insight into *how* the comparer module performs the claimed function. For example, Oasis asserts that column 6, lines 36-56 of the '571 patent is corresponding structure. However, this portion of the specification begins with a high-level reiteration of the function, followed by a description of *what* gets compared, and ends with a high-level description of a different function (a reporter module that summarizes some of the matches and/or differences). ('571 patent, col. 6:36-56; *see also id.*, cols. 75:63-67, 79:7-14)

³¹ Specifically, it is undisputed that “the techniques described at '571 [patent cols.] 76:1-76:27, 79:15-42, or '545 [patent cols.] 76:13-76:39, 79:27-54” constitute structure for the comparer module term. (D.I. 103, ex. 2 at 4) These citations indeed describe “multiple ways in which this comparison can be performed.” ('571 patent, cols. 76:1-2; *see also id.*, col. 79:15-16)

And at least a large portion of some of Oasis’ other citations also seem to discuss examples of what data is compared (along with, at times, a description of what is subsequently reported). (*Id.*, cols. 56:20-59:10, 59:58-62:44, 63:42-65:60, 76:28-78:23)³² Some passages describe different uses for comparing digests in the design of integrated circuits. (*Id.*, cols. 68:55-69:13, 70:13-21, 70:43-53) Yet these passages do not appear to be discussing *how* it is that the comparer module compares the digests. Thus, they do not refer to sufficient structure, and the Court will not include them in the construction. *See, e.g., Tracktime, LLC*, 2021 WL 2823163, at *7 (finding that portions of the patent that “simply repeat the function” and “describe applications and embodiments of the claimed annotation function” failed to disclose sufficient structure).

For these reasons, the Court concludes that “a comparer module [. . .] that receives and compares the digests of at least a first file and a second file, which contain design data” is subject to Section 112(6). The function is “comparing digests of [cells / design units] in the first file to digests of [cells / design units] in the second file[.]” The corresponding structure is: “the techniques described at '571 [patent, cols.] 56:42-51, 58:36-39, 62:1-9, 65:1-9, 71:20-23, 76:1-76:27, 79:15-42, or '545 [patent, cols.] 56:44-52, 58:35-37, 62:1-9, 65:1-9, 71:20-23, 76:13-76:39, 79:27-54[.]”

³² The Court was hindered here by the fact that the parties did not do a deep dive in their briefing on all of the portions of the specification that reference disputed structure, such that they did not (on a line-by-line or paragraph-by-paragraph basis) explain why certain content does or does not constitute relevant structure. The Court has done its best to carefully review the portions of the specification cited above. In doing so, it notes that there are a few portions—i.e., those at '571 patent, cols. 56:42-51, 58:36-39, 62:1-9, 65:1-9, 71:20-23 and at '545 [patent, cols.] 56:44-52, 58:35-37, 62:1-9, 65:1-9, 71:20-23—that seem to be discussing something more than just what data is compared or what is reported, and instead appear to describe *how* the comparison is accomplished. In light of that, the Court will include these portions of the specification in the construction.

F. “a [digester module / digester] . . . that receives the canonical forms for at least selected partitions and calculates and stores in the memory at least one digest per selected partition”

The next disputed term, “a [digester module / digester] . . . that receives the canonical forms for at least selected partitions and calculates and stores in the memory at least one digest per selected partition” (the “digester module”) is found in, *inter alia*, claim 14 of the '545 patent and claim 16 of the '571 patent. Its use in claim 16 of the '571 patent, recited above, is representative. The parties’ competing proposed constructions for the digester module are set out in the chart below:

Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“a [digester module / digester] . . . that receives the canonical forms for at least selected partitions and calculates and stores in the memory at least one digest per selected partition”	<p>See proposed constructions for “canonical form(s)” and “digest(s).”</p> <p>For the remainder of the term: plain and ordinary meaning/no construction required.</p> <p>In the alternative, the plain and ordinary meaning of the term “digester module” and “digester” could be articulated as “the portion of the software that generates the digest.”</p> <p>This limitation is not subject to 35 U.S.C. 112(6). In the alternative, to the extent 112(6) applies:</p> <p><u>Function</u>: “creating a digest for canonical form(s)”</p> <p><u>Structures</u>: Elements 534 in FIG. 5 and 614 in FIG. 6 of the Asserted Patents, the</p>	<p>Sec. 112/6.</p> <p><u>Function</u>: “calculate and store at least one digest for the canonical forms of each selected partition of the data that already has been transformed into a canonical form”</p> <p><u>Structures</u>: “applying a hash function to the canonical form of the data to generate a string output, including, e.g. CRC or MD5 hash that is 32 to 128 bits, as described at '545 [patent] 6:5-9, '571 [patent] 6:13-17”</p> <p>Dispute additional structures proposed by Plaintiff</p>

	techniques described at '571 Patent 6:4-25; 79:4-6; 79:43-65; 80:64-67; 81:13-16 and '545 Patent 5:63-6:16; 79:16-18; 79:55-80:10; 81:9-11; 81:25-28.	
--	---	--

(D.I. 103, ex. 2 at 5) The parties’ primary dispute here is whether this term is a means-plus-function term subject to Section 112(6). (D.I. 88 at 87)

Like the normalizer logic and comparer module terms, the digester module is found in representative claim 16 of the '571 patent; it is thus a limitation to “program code[,]” and it expressly recites the function of calculating and storing in the memory at least one digest per selected partition. ('571 patent, col. 84:48-50) The analysis here is the same as described above: there is a presumption that the term is not a means-plus-function term (due to the lack of the use of the word “means”), and the inquiry is whether a POSITA would understand the digester module as sufficient structure for performing the claimed function (i.e., calculating and storing digests). *XR Commc’ns, LLC*, 2023 WL 3529830, at *2.

Oasis does not assert that a digester module has a structural definition that is generally known to a POSITA. And Defendants’ expert opines that “digester module” does not have a definite structural meaning in the EDA arts. (D.I. 90 at ¶ 134; *see also* D.I. 88 at 68) Therefore, in order to avoid application of Section 112(6), there must be a sufficient description of *how* the digester module achieves the claimed function of calculating and storing digests.

Here, just as with the comparer module, there is not. The claim language does not speak to how the digester module calculates and stores the digests. (*See* '571 patent, col. 84:47-49) Oasis’ briefing asserts that a POSITA would understand that the digester is the portion of the software that “generates the digest”—which is really just another way of stating the claimed function—“and *encompasses* condensing and hashing functions.” (D.I. 88 at 87 (emphasis

added); *see also* D.I. 89 at ¶ 84) The specification teaches that “[a] variety of hash functions can be used to create the digests, such as CRC, MD5 and others.” (’571 patent, col. 6:16-17)

Defendants retort that if Oasis were taking the position that the digester module is *limited* to the MD5 and CRC algorithms disclosed in the patent, then that would convey a particular structure for calculating the digest. (Tr. at 169-70; D.I. 90 at ¶ 144(e)) However, this is not Oasis’ position (as seen by looking to its proposed plain and ordinary meaning construction for this term —“the portion of the software that generates the digest”). (D.I. 103, ex. 2 at 5; Tr. at 169-70, 197-98 (Defendants’ counsel asserting that Oasis’ counsel is “saying as long as it performs the function, it’s covered by our claim, which is exactly what Congress prohibited and enacted [Section 112(6)] to prevent”)) On this record, the Court concludes that the claim does not recite to a POSITA sufficient structure for the digester module to perform the claimed function. (D.I. 88 at 68)

Having made this determination, the Court must now: (1) determine the claimed function of the digester module; and (2) identify the corresponding structure disclosed in the specification. *Williamson*, 792 F.3d at 1351.

As for the claimed function, while Oasis proposes a different function than Defendants—one that does not track the claim language—by the time it filed its reply brief, Oasis did not seem to be still actively contesting Defendants’ proposed function. (D.I. 88 at 88) Defendants’ proposal tracks more closely with the express claim language, and so the Court adopts it here.

With respect to the corresponding structure, Defendants again dispute some of the structure that Oasis identifies. (*Id.* at 87-89) Defendants’ proposed corresponding structure is focused on how the digester module performs the claimed function: “[t]he digests . . . may be 32 or 64 bit codes generated from canonical output of the parser and normalizer logic. A variety of

hash functions can be used to create the digests, such as CRC, MD5 and others.” (’571 patent, col. 6:14-17; D.I. 88 at 88) Defendants assert that Oasis’ additional proposed structures merely constitute empty boxes, high-level descriptions of the digesting and storing function (not *how* those functions are performed), or descriptions of a different function. (D.I. 88 at 88, 89)

The Court agrees with Defendants here. For example, one portion of the specification that Oasis wants to include as structure states: “[i]n one embodiment, a parser **531** running on a processor **530**, normalizer logic **533** running cooperatively with the parser, and a digester **534** running on the processor generated syntax trees **532** and canonical cell digests that are stored in memory **415**.” (’571 patent, col. 6:8-12; *see also, e.g., id.*, col. 79:4-6) This description tells us nothing about how the digester calculates and stores the digests. Another disputed portion tells us that “[t]he partition module may further partition functionally significant design data from non-significant data within the canonical forms before the digester calculates and stores the digests.” (*Id.*, col. 79:43-46) This description relates more to the partitioning module step of claim 16, and again tells us nothing about how the digester calculates and stores the digests. Accordingly, the Court adopts Defendants’ proposed corresponding structure for the digester module.

For the foregoing reasons, the Court concludes that “a [digester module / digester] . . . that receives the canonical forms for at least selected partitions and calculates and stores in the memory at least one digest per selected partition” is subject to Section 112(6). The function is “calculate and store at least one digest for the canonical forms of each selected partition of the data that already has been transformed into a canonical form[.]” The corresponding structure is: “applying a hash function to the canonical form of the data to generate a string output, including,

e.g. CRC or MD5 hash that is 32 to 128 bits, as described at '545 [patent, col.] 6:5-9, '571 [patent, col.] 6:13-17[.]”

G. “parses a file containing design data” and “parsing syntax of [. . .] design data”

The next disputed term, “parses a file containing design data” and “parsing syntax of [. . .] design data” appears in, *inter alia*, claims 1, 4-6, 8-11 and 14 of the '545 patent and claims 1, 12 and 15-16 of the '571 patent. The term’s use in claim 1 of each of the '545 and '571 patents, depicted above, is representative. The parties’ competing proposed constructions for this term are set out in the chart below:

Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“parses a file containing design data” “parsing syntax of [. . .] design data”	<i>See</i> proposed constructions for “syntax tree.” For the remainder of the term: plain and ordinary meaning/no construction required. In the alternative, the plain and ordinary meaning of the term could be articulated as “analyzing the design data into logical components”	“recogniz[es/ing] design data roles by applying syntax rules”

(D.I. 103, ex. 2 at 2) It is not entirely clear to the Court what the parties’ dispute is regarding this term. The briefing on this term was short and largely focused on language that Oasis has since omitted from its proposed construction. (D.I. 88 at 57-60)

Defendants seem to suggest that there may be a remaining dispute as to whether “parsing” must recognize the “basic structure of code” (which Defendants say is reflected in their proposed construction), or instead whether “parsing” involves simply scanning or analyzing data. (*Id.* at 60) If that is the case, though, it is not clear to the Court: (1) how scanning or analyzing

data is different from recognizing the basic structure of code; and (2) what Oasis’ response is.

For these reasons, the Court will not construe this term at this time. To the extent that the parties continue to have a dispute over the scope of “parsing[,]” they can raise the issue in their summary judgment briefing or otherwise later in the case.

G. “a canonical forming module that interprets the syntax trees to produce canonical forms of the design data”

The final disputed term, “a canonical forming module that interprets the syntax trees to produce canonical forms of the design data[,]” (the “canonical forming module”) appears, *inter alia*, in claim 14 of the '545 patent and claim 16 of the '571 patent. The term’s use in claim 16 of the '571 patent, set out above, is representative. The parties’ competing proposed constructions for the canonical forming module are set out in the chart below:

Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“a canonical forming module that interprets the syntax trees to produce canonical forms of the design data”	<p><i>See</i> proposed constructions for “canonical form(s)” and “syntax tree.”</p> <p>For the remainder of the term: plain and ordinary meaning/no construction required.</p> <p>This limitation is not subject to 35 U.S.C. 112(6). In the alternative, to the extent 112(6) applies:</p> <p><u>Function</u>: “interprets design data that has been parsed into syntax trees in order to produce canonical forms of that data”</p> <p><u>Structures</u>: Within Elements 533 in FIG. 5 and 613 in FIG. 6 of Asserted Patents, the</p>	<p>112/6.</p> <p><u>Function</u>: “interprets design data that has been parsed into syntax trees in order to produce canonical forms of that data”</p> <p><u>Structures</u>: “the techniques described at '545 [patent] 34:10-16, 71:22-25, 75:14-76:4, 78:59-79:15 or '571 [patent] 33:64-34:3, 71:22-25, 75:2-59[,] 78:47-79:3”</p> <p>Dispute additional structures proposed by Plaintiff.</p>

	techniques described at '571 Patent 33:64-34:3, 35:20-69; 71:20-25; 75:1-59; 76:28-35; 78:47-79:14; 79:66-80:13 and '545 Patent 34:10-16; 35:33-36:8; 69:22-26; 71:20-25; 75:13-76:4; 76:40-76:64; 78:59-79:15, 80:11-25	
--	--	--

(D.I. 103, ex. 2 at 3-4) The parties’ primary dispute here is whether this term is a means-plus-function term subject to Section 112(6). (D.I. 88 at 67, 83)

Oasis argues that the canonical forming module is not subject to Section 112(6) for the same reasons as the normalizing module. (*Id.* at 79) Its expert, Dr. Athan, opines that the claim language itself “conveys specific structural meaning to a POSITA”—i.e., it produces canonical forms of design data by interpreting design data that has been parsed into syntax trees. (D.I. 89 at ¶ 79)

The Court agrees with Oasis here, for the same reasons as articulated above with respect to the normalizing module. The claim language itself tells us how the canonical module is going to produce canonical forms of design data. Indeed, for their proposed corresponding structure, Defendants point to some portions of the specification that further describe this “how” that is reflected in the claim language (i.e., generating canonical forms by interpreting design data that has been parsed into syntax trees). (D.I. 103, ex. 2 at 3) For example, Defendants point to column 75 of the '571 patent, which teaches that “[s]ome design data files are normalized or transformed into a canonical format by passing them through a parser and applying parsing rules. Other files may require semantic analysis of a syntax tree generated by the parsing or other manipulations to normalize the file.” ('571 patent, col. 75:5-9)

Thus, the Court concludes that Defendants have not met their burden to demonstrate, by a preponderance of the evidence, that claim 16 of the '571 patent (and the other applicable claims)

fails to recite sufficiently define structure for the canonical forming module. Defendants present no alternative proposed construction, and so no further construction is necessary.

IV. CONCLUSION

For the foregoing reasons, the Court adopts the following constructions:

1. “canonical forms” should be construed to mean “a standardized form of a body of design data”
2. “digest[s]” should be construed to mean “output of a hash function, including, e.g., CRC or MD5”
3. “cell” should be construed to mean “a subset of design data, having a view and an artifact, that can be referenced as a whole representation and expression of: an object, properties, or comments”
4. No construction is necessary for “normalizer logic [. . .] cooperating with the parser that organizes the syntax trees to produce canonical forms” and “normalizing the design data within the cells into canonical forms[.]”
5. “a comparer module [. . .] that receives and compares the digests of at least a first file and a second file, which contain design data” is subject to Section 112(6). The function is “comparing digests of [cells / design units] in the first file to digests of [cells / design units] in the second file[.]” The corresponding structure is: “the techniques described at '571 [patent, cols.] 56:42-51, 58:36-39, 62:1-9, 65:1-9, 71:20-23, 76:1-76:27, 79:15-42, or '545 [patent, cols.] 56:44-52, 58:35-37, 62:1-9, 65:1-9, 71:20-23, 76:13-76:39, 79:27-54[.]”
6. “a [digester module / digester] . . . that receives the canonical forms for at least selected partitions and calculates and stores in the memory at least one digest per selected partition” is subject to Section 112(6). The function is “calculate and store at least one digest for

the canonical forms of each selected partition of the data that already has been transformed into a canonical form[.]” The corresponding structure is: “applying a hash function to the canonical form of the data to generate a string output, including, e.g. CRC or MD5 hash that is 32 to 128 bits, as described at ’545 [patent, col.] 6:5-9, ’571 [patent, col.] 6:13-17[.]”

7. The Court declines to construe “parses a file containing design data” or “parsing syntax of [. . .] design data” at this time.

8. No construction is necessary for “a canonical forming module that interprets the syntax trees to produce canonical forms of the design data[.]”

9. “syntax trees” should be construed to mean “a hierarchical data structure representing the syntax of design data with connected nodes, each node having exactly one parent, except a single root node which has no parent”