

**United States Court of Appeals  
for the Federal Circuit**

---

**SYNOPSIS, INC., A DELAWARE CORPORATION,**  
*Plaintiff-Appellant*

v.

**MENTOR GRAPHICS CORPORATION, AN  
OREGON CORPORATION,**  
*Defendant-Appellee*

---

2015-1599

---

Appeal from the United States District Court for the  
Northern District of California in No. 3:12-cv-06467-  
MMC, Judge Maxine M. Chesney.

---

Decided: October 17, 2016

---

CARTER GLASGOW PHILLIPS, Sidley Austin LLP, Wash-  
ington, DC, argued for plaintiff-appellant. Also represent-  
ed by RYAN C. MORRIS; M. PATRICIA THAYER, PHILIP W.  
WOO, San Francisco, CA.

JOHN D. VANDENBERG, Klarquist Sparkman, LLP,  
Portland, OR, argued for defendant-appellee. Also repre-  
sented by ANDREW M. MASON.

---

Before LOURIE, MOORE, and CHEN, *Circuit Judges*.

CHEN, *Circuit Judge*.

Synopsys, Inc. appeals the District Court for the Northern District of California’s grant of summary judgment invalidating certain claims of U.S. Patent Nos. 5,530,841; 5,680,318; and 5,748,488 (collectively, the Gregory Patents) under 35 U.S.C. § 101. *See Synopsys, Inc. v. Mentor Graphics Corp.*, 78 F. Supp. 3d 958 (N.D. Cal. 2015) (*Summary Judgment Order*). Synopsys argues that, contrary to the district court’s holding, the Gregory Patents are not directed to ineligible subject matter because they relate to complex algorithms used in computer-based synthesis of logic circuits. We disagree. A review of the actual claims at issue shows that they are directed to the abstract idea of translating a functional description of a logic circuit into a hardware component description of the logic circuit.<sup>1</sup> This idea of reviewing a description of certain functions and turning it into a representation of the logic component that performs those functions can be—and, indeed, was—performed mentally or by pencil and paper by one of ordinary skill in the art. Moreover, the claims do not call for the involvement of a computer. They therefore cannot be characterized as an improvement in a computer as a tool. The claims add nothing to the abstract idea that rises to the level of an “inventive concept” as required by precedent. We therefore *affirm* the district court’s grant of summary judgment of invalidity.

---

<sup>1</sup> For example, the claim the parties identify as representative calls for generating a schematic or netlist representation of a level sensitive latch when given a description of the logic operation of a level sensitive latch. ’841 patent, 62:61–63:12.

## BACKGROUND

### I. The Gregory Patents

The Gregory Patents are continuations of since-abandoned U.S. Patent Application No. 07/632,439 and all share a common specification.<sup>2</sup> The patents relate generally to the logic circuit design process. The logic circuit design process has evolved significantly over time. Synopsys describes the inventions of the Gregory Patents as critical steps in this evolution.

In the early days of logic circuits,<sup>3</sup> a designer was required to specify his design in great detail. He would do so in the form of a schematic diagram that identified individual hardware components and the interconnections between them or via a set of Boolean logic equations that specified the precise functionality of the design. '841 patent, 1:41–44. A fabrication facility would then build the corresponding physical circuit based on the architecture presented in the detailed design.

Over time, logic circuits became more and more complex. As complexity increased, many designers began to focus on the higher-level functionality of their designs and became less concerned with the detailed schematics or Boolean logic equations necessary to implement that functionality. *Id.* at 1:47–49. These developments created a need for a form of computer code that a designer

---

<sup>2</sup> Unless otherwise noted, all references to the common specification will be to the specification of the '841 patent. Equivalent disclosures can be found in the '318 and '488 patents.

<sup>3</sup> A “logic circuit” is an electrical circuit where all signals take the form of a logic high (also known as “true” and often represented by the binary digit “1”) or a logic low (also known as “false” and often represented by the binary digit “0”).

could use to describe a logic circuit at a functional level. This led to the advent of various functional computer languages known as hardware description languages (HDLs). *Id.* at 1:50–55. HDLs allowed designers to “describe only the desired operation of the logic circuit, i.e., the signals generated by the logic circuit,” rather than having to specify the actual individual components and interconnections of the logic circuit. *Id.* at 1:62–64; *see also id.* at 1:50–55 (describing HDLs as operating “at least one level of abstraction removed from a schematic diagram or a set of [B]oolean logic equations”).

The introduction of HDLs necessitated the development of computerized design tools that could translate the functional description of the logic circuit into a detailed design for fabrication. *Id.* at 1:64–67. Early computerized design tools, however, could only recognize and translate simple circuit elements. *Id.* at 2:1–3. “For many circuit elements, such as high impedance drivers, level sensitive latches and edge sensitive flip-flops, the designer was required first to specify [(i.e., instantiate)] the specific circuit element and then the desired connection of that element using the HDL.” *Id.* at 2:3–7.

The Gregory Patents describe constructs known as “control flow graphs,” *id.* at 2:65–3:8, and “assignment conditions,” *id.* at 3:22–30, that provide a scheme to translate HDL-based functional descriptions of logic circuits into hardware component descriptions of those same circuits without requiring the designer to instantiate any individual hardware components—not even high impedance drivers, level sensitive latches, or edge sensitive flip-flops. *Id.* at 2:27–36. The patent specification goes through several examples for different components to illustrate how control flow graphs and assignment conditions are used to translate a functional description of a logic circuit to a hardware component description of that logic circuit.

We will explore in detail one such example, which is claimed in claim 1 of the '841 patent.<sup>4</sup> But, first, one must understand the general concept of binary logic as well as the constructs introduced in the Gregory Patents—namely flow control statements, directive statements, asynchronous load functions, and asynchronous data functions—what they are and how they work. We can gain this understanding through a review of the following simple example of HDL code:

```
If(COND)
    Q: = 1;
else
    Q: = 0;
endif
```

Here, “Q” is the output of the segment of code and “COND” is a condition. The value of output Q is dictated by the line of code “If(COND),” which the specification labels as a “flow control statement.” See '841 patent, 11:20–23. This line of code asks the question “Is condition COND true?”—i.e., does it equal 1?<sup>5</sup> As the moniker “flow control statement” suggests, the answer to this question *controls the flow* of how the rest of the code runs. *Id.* at 11:18–20. In the above example, when condition COND is true (i.e., has the value “1”), the code *flows* to the immediately-following line of code, i.e., “Q: = 1.” The specification labels this line of code as a “directive statement” for it directs that output Q be assigned the value 1. *Id.* at 11:1–8. In contrast, when condition COND is false (i.e., has the value “0”), the code skips the directive statement “Q: = 1”

---

<sup>4</sup> The parties agree that claim 1 of the '841 patent is representative of all claims on appeal.

<sup>5</sup> In the field of binary logic to which the Gregory Patents belong, data is represented by “bits.” A bit can either equal 1 (also known as logic “true”) or 0 (also known as logic “false”).

and *flows* directly to the line of code “else.” Here, we find another directive statement: “Q: = 0.” Pursuant to this directive statement, Q is assigned the value 0. The relationship between condition COND and output Q can be summarized in the following table:

COND	Q
1	1
0	0

The Gregory Patents describe how the invention converts the statements from the HDL code into two constructs the specification calls “assignment conditions”: (1) an “asynchronous load function;” and (2) an “asynchronous data function.” These two assignment conditions provide another type of description of the functionality of the HDL code. The district court construed “asynchronous load function,” represented “AL( ),” as “a hardware description function for load specifying the condition or conditions under which the variable is [asynchronously<sup>6</sup>] assigned a value.” *Claim Construction Order*, 2013 WL 5957866, at \*4. In the above example, the “asynchronous load function” for output Q is “1” (i.e., AL(Q) = 1), because output Q is assigned a new value (i.e.,

---

<sup>6</sup> The district court construed “asynchronous” to mean “not triggered by a clock signal.” *Synopsys, Inc. v. Mentor Graphics Corp.*, No. C 12-6467 MMC, 2013 WL 5957866, at \*4 (N.D. Cal. Nov. 7, 2013) (*Claim Construction Order*). “Asynchronous” is the opposite of “synchronous,” which the district court construed to mean “triggered by a clock signal.” *Id.* The example HDL code is asynchronous, because it does not take a clock signal as an input.

it is “loaded”) both when condition COND is true ( $Q = 1$ ) and when it is false ( $Q = 0$ ). See ’841 patent, 4:21–23.

The district court construed “asynchronous data function,” represented “AD( ),” as “a hardware description function for data specifying the condition or conditions under which the variable is [asynchronously] assigned a value.” *Claim Construction Order*, 2013 WL 5957866, at \*3. Here, the “asynchronous data function” for output Q is “COND” because output Q is assigned the value “1” if, and only if, condition COND is true. ’841 patent, 4:23–25. Therefore,  $AD(Q) = COND$ .

The asynchronous load function for this example HDL code (i.e.,  $AL(Q) = 1$ ) is *constant*, because it always equals 1. In that way it differs from the asynchronous data function. The value of the asynchronous data function (i.e.,  $AD(Q) = COND$ ) is *non-constant* or *variable*, because it can be 1 or 0 depending on the value of condition COND. The concept of constant—as opposed to non-constant or variable—assignment conditions will be important as we next explore claim 1.

Representative claim 1 and the associated portion of the specification detail the method of using assignment conditions to translate from a functional description of a level sensitive latch into a hardware component description of that same latch. Claim 1 reads:

A method for converting a hardware independent user description of a logic circuit, that includes flow control statements including an IF statement and a GOTO statement, and directive statements that define levels of logic signals, into logic circuit hardware components comprising:

converting the flow control statements and directive statements in the user description for a logic signal Q into an assignment condition  $AL(Q)$  for an asynchronous

load function AL( ) and an assignment condition AD(Q) for an asynchronous data function AD(); and

generating a level sensitive latch when both said assignment condition AL(Q) and said assignment condition AD(Q) are non-constant;

wherein said assignment condition AD(Q) is a signal on a data input line of said flow through latch;

said assignment condition AL(Q) is a signal on a latch gate line of said flow through latch; and

an output signal of said flow through latch is said logic signal Q.

*Id.* at 62:61–63:12.

A level sensitive latch is a basic form of memory. It is a hardware component that stores a binary input (i.e., the value “1” or “0”), but only when a specified condition is true. A level sensitive latch can be described functionally using HDL code as follows:

TABLE 8

---

An Example of User Description 110

---

```

If( COND)
  Q: = D
else
endif

```

---

*Id.* at 21:49–56. Here, “D” represents the input to the latch and “Q” the output.

The relationship between input D and output Q is dictated by the “flow control statement” defined by the line of



code “If(COND).” In this example, when condition “COND” is true (i.e., has the value “1”), the code *flows* to the immediately following line of code—i.e., “Q: = D”—and output Q is assigned the value of input D. In contrast, when condition COND is false (i.e., has the value “0”), the code skips the directive statement “Q: = D” and *flows* directly to the line of code “else.” In this example, no instructions follow “else.” The value of output Q therefore remains unchanged. In sum, when condition COND is true, output Q is assigned the value of input D; when condition COND is false, output Q retains its prior value regardless of whether the value of input D remains the same or changes. The relationship between condition COND, input D, and output Q can be summarized in the following table:

COND	D	Q
1	1	1
1	0	0
0	1	Q
0	0	Q

The claimed method takes the functional description of the latch as an input. *Id.* at 62:61–62. It then converts the functional description into an equivalent description in the form of (1) an asynchronous load function; and (2) an asynchronous data function. *Id.* 62:66–63:3. Here, the asynchronous load function for output Q is COND because output Q is assigned a new value (i.e., it is “loaded”) whenever condition COND is true. The asynchronous

data function for Q is “COND\*D”<sup>7</sup> because output Q is assigned the value “1” if, and only if, both condition COND and input D are true.

The assignment conditions associated with the functional description of the latch are summarized in the table below:

TABLE 9

Variable	Assignment Conditions					
	AL()	AD()	SL()	SD()	DC()	Z()
Q	COND	COND*D	0	0	0	0

*Id.* at 21:58–65.

Claim 1 specifies that where, as here, the asynchronous load function and the asynchronous data function are non-constant,<sup>8</sup> the claimed method generates a level sensitive latch. *Id.* at 63:4–6; *see also id.* at 24:56–63. Claim 1 further specifies that the latch’s data input is the asynchronous data function (i.e., COND\*D); the latch’s gate is the asynchronous load function (i.e., COND); and

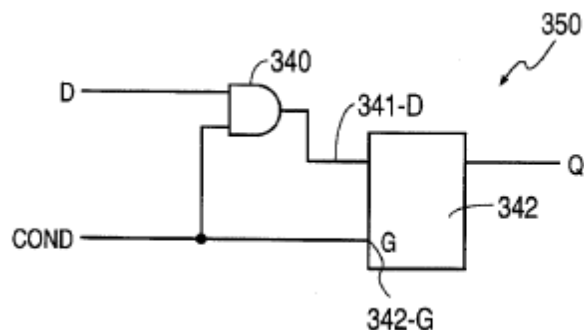
---

<sup>7</sup> “\*” symbolizes a logic AND. The logic AND of two variables is true if, and only if, both variables are true. If either variable is false, the logic AND of the variables is also false.

<sup>8</sup> The asynchronous load function and the asynchronous data function are “non-constant” (i.e., variable) because each can change—COND in the case of the asynchronous load function and COND\*D in the case of the asynchronous data function. Claim 1 does not specify what component is generated if either the asynchronous load function or the asynchronous data function (or both) were constant—for example, if one of the functions was always true (e.g., AL(Q) = 1).

the latch's output is Q. *Id.* at 63:7–12; *see also id.* at 22:12–23, 24:56–63.

A hardware component description of the level sensitive latch is shown below:



*Id.* at Fig. 8A. In this hardware component description, the rectangle marked with 342 represents the level sensitive latch, itself. Consistent with the above description, the latch's input (341-D) is the logic AND (340) of input D and condition COND; and, its gate (342-G) is condition COND. *Id.* at 22:12–23.

Importantly, the Gregory Patents make clear that HDL code existed in the prior art. *See id.* at 1:49–50 (“Hardware description language (HDL) was developed to assist such designers.”). The HDL code for the level sensitive latch shown in Table 8 was already well known by the time the claimed inventions of the Gregory Patents were conceived. The same is true of the circuit diagram for a level sensitive latch shown in Figure 8A; circuit diagrams like this existed long before the Gregory Patents. *See id.* at 1:41–44 (“Historically, a user was required typically to supply either a logic schematic diagram for use in the automated design process . . .”). What Gregory instead claims to have invented is a process for interpreting the HDL code in Table 8 that uses the assignment conditions of Table 9 to identify the circuit diagram of Figure 8A as the hardware that performs the function recited in the HDL code. At bottom, the infor-

mation provided in Table 8 (code), Table 9 (assignment conditions), and Figure 8A (circuit diagram) are all equivalent representations of the same thing: a level sensitive latch.

The Gregory Patents describe and claim additional examples relating to other circuit components, specifically high impedance drivers and edge sensitive flip-flops, that involve the use of different assignment conditions—namely synchronous load functions, synchronous data functions, don't care functions, and high-impedance functions.

## II. Procedural History

Synopsys filed suit against Mentor Graphics Corp. on December 12, 2012, in the Northern District of California alleging infringement of the Gregory Patents and U.S. Patent No. 6,836,420 (collectively, the patents-in-suit). In particular, Synopsys alleged that Mentor Graphics' "Precision" family of logic synthesis products and its "Veloce" family of emulators infringed the following claims of the patents-in-suit: claim 1 of the '841 patent; claims 32, 35, and 36 of the '318 patent; claims 1, 2, 8, and 9 of the '488 patent; and claims 1–3, 10–13, and 20 of the '420 patent.

Based on disputed issues raised by the parties, the court construed certain claim terms of the patents-in-suit on November 7, 2013. Notably, the court did not construe any claim of the Gregory Patents to require the use of a computer—general purpose or otherwise—or any other type of hardware.<sup>9</sup> *See Claim Construction Order*, 2013

---

<sup>9</sup> Perhaps more notably, none of Synopsys' proposed constructions required the use of a computer or any type of hardware. J.A. 2395–422. In particular, representative claim 1's "generating a . . . latch" means, not creating

WL 5957866, at \*2–5. Neither party challenges any of the district court’s claim constructions on appeal.

The parties subsequently cross-moved for summary judgment on Mentor Graphics’ defense that the Gregory Patents were invalid under 35 U.S.C. § 101. The court granted Mentor Graphics’ motion and invalidated all asserted claims of the Gregory Patents. *See Summary Judgment Order*, 78 F. Supp. 3d at 966. In reaching its decision, the court applied the now common two-step test described by the Supreme Court in *Alice Corp. v. CLS Bank International*, 134 S. Ct. 2347 (2014). *See Summary Judgment Order*, 78 F. Supp. 3d at 962–63. The court observed that “[e]ach of the steps in the claimed methods can be performed by a skilled designer either mentally or with pencil and paper.” *Id.* at 961. Due to the breadth of the claims, the court found, under the first step of the *Alice* test, that “the claims are directed to a mental process . . . ‘a subcategory of unpatentable abstract ideas.’” *Id.* at 963 (quoting *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1371 (Fed. Cir. 2011)); *see also id.* at 961 (noting that while the claimed method “is primarily intended for use with a computer,” “the claims themselves do not expressly call for a computer or other piece of equipment”); *id.* at 963 (“The claimed methods here at issue do not entail anything physical. Rather, as discussed above, the asserted claims are directed to the process of inference, which is fundamental to IC design and can be performed mentally.”); *id.* at 964 (“[T]he claimed methods do not require complex calculations; as noted, the claimed steps were performed mentally by the inventors and can be performed by a skilled designer either mentally or with the aid of a pencil and paper.”).

---

the physical component, but generating a representation (e.g., description, schematic, etc.) of such a component.

Turning to the second step of the *Alice* test, the court rejected Synopsys’ argument that the claims necessarily contained an inventive concept because Mentor Graphics failed to present prior art that disclosed the claimed methods. *Id.* at 964. The court then found that, while the claims were directed to a “specific” mental process, they nonetheless “preempt[ed] a building block of human ingenuity.” *Id.* at 965. Finally, it found that the claims concerned “well-understood, routine, conventional activity, previously engaged in by those in the field.” *Id.* (“As acknowledged in the specification, skilled designers had been inferring the necessary parts and connections for ICs long before the Gregory patents issued.”).

The court entered final judgment with respect to the Gregory Patents on April 20, 2015.<sup>10</sup> Synopsys appeals from this final judgment. We have jurisdiction pursuant to 28 U.S.C. § 1295(a)(1).

#### DISCUSSION

“We review a district court’s grant of summary judgment according to the law of the regional circuit, here the Ninth Circuit, where summary judgment is reviewed *de novo*.” *Kaneka Corp. v. Xiamen Kingdomway Grp. Co.*, 790 F.3d 1298, 1303 (Fed. Cir. 2015) (citations omitted). “In the Ninth Circuit, summary judgment is appropriate when, drawing reasonable inferences in favor of the non-

---

<sup>10</sup> In that same order, the court stayed further proceedings on the ’420 patent in view of a then-pending *inter partes* review. The Patent Trial and Appeal Board (Board) subsequently found the challenged claims of the ’420 patent unpatentable as obvious. *Mentor Graphics Corp. v. Synopsys, Inc.*, IPR2014-00287, 2015 WL 3637569 (PTAB June 11, 2015). We recently affirmed the Board’s decision. *Synopsys, Inc. v. Mentor Graphics Corp.*, No. 2015-2056 (Fed. Cir. Oct. 11, 2016).

moving party, there is no genuine issue of material fact.” *Id.* (citing *Comite de Jornaleros de Redondo Beach v. City of Redondo Beach*, 657 F.3d 936, 942 (9th Cir. 2011)).

A patent may be obtained for “any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof.” 35 U.S.C. § 101. The Supreme Court has “long held that this provision contains an important implicit exception: Laws of nature, natural phenomena, and abstract ideas are not patentable.” *Ass’n for Molecular Pathology v. Myriad Genetics, Inc.*, 133 S. Ct. 2107, 2116 (2013) (quoting *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1293 (2012)) (alteration omitted). First in *Mayo* and later in *Alice*, the Supreme Court set forth a two-step analytical framework to identify patents that, in essence, claim nothing more than abstract ideas. The court must first “determine whether the claims at issue are directed to a patent-ineligible concept.” *Alice Corp.*, 134 S. Ct. at 2355. If so, the court must then “consider the elements of each claim both individually and ‘as an ordered combination’ to determine whether the additional elements ‘transform the nature of the claim’ into a patent-eligible application.” *Id.* (quoting *Mayo*, 132 S. Ct. at 1298, 1297). The Supreme Court has described this second step of the analysis as “a search for an ‘inventive concept.’” *Id.*

#### I. *Alice* Step 1: Are the Asserted Claims directed to an abstract idea?

The district court based its *Alice* Step 1 analysis on a basic premise: “the claims are directed to a mental process.” *Summary Judgment Order*, 78 F. Supp. 3d at 963. We held in *CyberSource* that mental processes are “a subcategory of unpatentable abstract ideas.” 654 F.3d at 1371. As we explained:

Methods which can be performed entirely in the human mind are unpatentable not because there is anything wrong with claiming mental method

steps as part of a process containing non-mental steps, but rather because computational methods which can be performed *entirely* in the human mind are the types of methods that embody the “basic tools of scientific and technological work” that are free to all men and reserved exclusively to none.

*Id.* at 1373 (quoting *Gottschalk v. Benson*, 409 U.S. 63, 67 (1972)) (emphasis in original). While the Supreme Court has altered the § 101 analysis since *CyberSource* in cases like *Mayo* and *Alice*, we continue to “treat[] analyzing information by steps people go through in their minds, or by mathematical algorithms, without more, as essentially mental processes within the abstract-idea category.” *Elec. Power Grp., LLC v. Alstom S.A.*, 830 F.3d 1350, 1354 (Fed. Cir. 2016) (citations omitted).

Although the Asserted Claims (i.e., claim 1 of the ’841 patent; claims 32, 35, and 36 of the ’318 patent; and claims 1, 2, 8, and 9 of the ’488 patent), which are all method claims, are devoid of any reference to a computer or any other physical component, Synopsys disputes the district court’s characterization of the claims as mental processes. It suggests that the “complexity” of the claimed methods would make it implausible—if not impossible—for a skilled logic circuit designer to perform the methods mentally or with pencil and paper. Appellant’s Opening Br. 21. It distinguishes these supposedly “complex” claims from the “simple” concepts found unpatentable in cases like *Alice* and *Bilski*<sup>11</sup>. Appellant’s Opening Br. 39.

But, Synopsys’ argument is belied by the actual claims at issue. The parties agree that claim 1 of the ’841 patent, discussed above, is representative of all Asserted

---

<sup>11</sup> *Bilski v. Kappos*, 561 U.S. 593 (2010).



Claims. For convenience, we present the claim again here:

A method for converting a hardware independent user description of a logic circuit, that includes flow control statements including an IF statement and a GOTO statement, and directive statements that define levels of logic signals, into logic circuit hardware components comprising:

converting the flow control statements and directive statements in the user description for a logic signal Q into an assignment condition AL(Q) for an asynchronous load function AL( ) and an assignment condition AD(Q) for an asynchronous data function AD( ); and

generating a level sensitive latch when both said assignment condition AL(Q) and said assignment condition AD(Q) are non-constant;

wherein said assignment condition AD(Q) is a signal on a data input line of said flow through latch;

said assignment condition AL(Q) is a signal on a latch gate line of said flow through latch; and

an output signal of said flow through latch is said logic signal Q.

'841 patent, 62:61–63:12. The claim recites a method of changing one description of a level sensitive latch (i.e., a functional description) into another description of the level sensitive latch (i.e., a hardware component description) by way of a third description of that very same level sensitive latch (i.e., assignment conditions). As demonstrated above, *supra* at 8–11, and in the patent specifica-

tion itself, '841 patent, 21:45–22:23, the method can be performed mentally or with pencil and paper. The skilled artisan must simply analyze a four-line snippet of HDL code:

TABLE 8

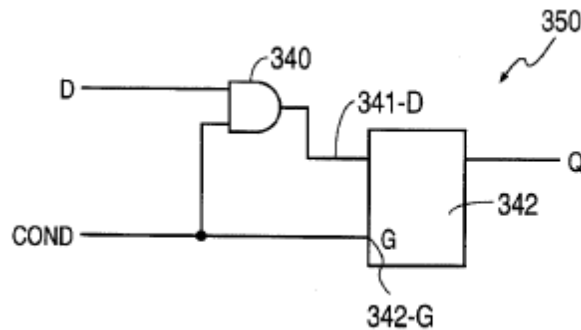
An Example of User Description 110
If( COND) Q: = D else endif

*id.* at 21:49–56; translate this short piece of code into assignment conditions:

TABLE 9

Variable	Assignment Conditions					
	AL()	AD()	SL()	SD()	DC()	Z()
Q	COND	COND*D	0	0	0	0

*id.* at 21:58–65; and further translate those two assignment conditions into a schematic representation of a level sensitive latch:



*id.* at Fig. 8A. Although an understanding of logic circuit design is certainly required to perform the steps, the

limited, straightforward nature of the steps involved in the claimed method make evident that a skilled artisan could perform the steps mentally. The inventors of the Gregory Patents confirmed this point when they admitted to performing the steps mentally themselves. *Summary Judgment Order*, 78 F. Supp. 3d at 961, 964.

Synopsys' reliance on *TQP Development, LLC v. Intuit Inc.*, No. 2:12-cv-180-WCB, 2014 WL 651935 (E.D. Tex. Feb. 19, 2014), is therefore misplaced. See Appellant's Opening Br. 39 n.8. In that case, the district court denied the defendant's motion for summary judgment that claims for a specific data encryption method for computer communication were invalid under § 101. *TQP*, 2014 WL 651935, at \*1. It distinguished the claims at issue from the mental processes found unpatentable in cases like *Gottschalk*. It explained that unlike those "simple," "basic" processes, the plaintiff's "invention involves a several-step manipulation of data that, *except in its most simplistic form, could not conceivably be performed in the human mind or with pencil and paper.*" *Id.* at \*4 (emphasis added). This case is different. Representative claim 1 is directed to generating a representation of a single specific hardware component and can be—and was—performed mentally or with pencil and paper.

Synopsys next argues that even if the Asserted Claims *could* be performed mentally they *would*, in practice, be performed on a computer. See, e.g., Appellant's Opening Br. 39 n.8 ("The methods here are designed for use by computers, and a skilled artisan would understand that the process is designed solely for computers."), Appellant's Reply Br. 9 n.6 ("Mentor's argument completely ignores that the purpose of the claimed inventions was to avoid the need to design certain circuit elements by hand and enable the increasingly necessary automation of circuit design through the use of synthesis software."). It attempts to tie the claims to those computerized design tools now common in industry. In support of this argu-

ment, counsel for Synopsys during oral argument pointed to the “200 pages of code” attached to the specifications of the Gregory Patents that he contended reveal the “true novelty” of the Asserted Claims. Oral Argument Tr. 4:25–4:37.

While Synopsys may be correct that the inventions of the Gregory Patents were intended to be used in conjunction with computer-based design tools, the Asserted Claims are not confined to that conception. The § 101 inquiry must focus on the language of the Asserted Claims themselves. *See Accenture Global Servs., GmbH v. Guidewire Software, Inc.*, 728 F.3d 1336, 1345 (Fed. Cir. 2013) (admonishing that “the important inquiry for a § 101 analysis is to look to the claim”); *see also Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat’l Ass’n*, 776 F.3d 1343, 1346 (Fed. Cir. 2014) (“We focus here on whether the claims of the asserted patents fall within the excluded category of abstract ideas.”), *cert. denied*, 136 S. Ct. 119 (2015).

On their face, the claims do not call for any form of computer implementation of the claimed methods. Synopsys stops short of arguing that the Asserted Claims must be *construed* as requiring a computer to perform the recited steps. Synopsys never sought such a construction before the district court and it does not press for such a construction here.<sup>12</sup> Its argument therefore fails. Be-

---

<sup>12</sup> While Synopsys repeatedly describes the claimed methods as implemented on a computer, *see, e.g.*, Appellant’s Opening Br. 12 (“The patents claim methods for a computer running specialized software to take ‘flow control statements’ and ‘directive statements’ in a user’s description written in HDL, and convert them into ‘assignment conditions’ for ‘hardware description functions,’ which, in turn, are used by the computer to determine the appropriate hardware and connections.” (citations omit-

cause the Asserted Claims make no mention of employing a computer or any other physical device, they are so broad as to read on an individual performing the claimed steps mentally or with pencil and paper. Just as we have held that complex details from the specification cannot save a claim directed to an abstract idea that recites generic computer parts, the Gregory Patents' incorporation of software code cannot save claims that lack any computer implementation at all. *See Accenture*, 728 F.3d at 1345 (“[T]he complexity of the implementing software or the level of detail in the specification does not transform a claim reciting only an abstract concept into a patent-eligible system or method.”).

For this reason, we need not decide whether a computer-implemented version of the invention would not be “directed to” an abstract idea. And, for the same reasons, Synopsys cannot rely on our decisions in *Enfish*<sup>13</sup> and *McRO*<sup>14</sup> to support the patentability of the Asserted Claims. In *Enfish*, we held that claims “directed to a specific improvement to the way computers operate” to store and retrieve data were not unpatentably abstract. 822 F.3d at 1336. The claims were not simply drawn to a disembodied data table. *See id.* at 1337 (“Here, the claims are not simply directed to any form of storing tabular data, but instead are specifically directed to a *self-referential* table for a computer database.” (emphasis in

---

ted)), its counsel recognized at oral argument that the words of the Asserted Claims do not require a computer and he referred instead to the patent specification and extrinsic evidence that a human would not use the methods as claimed. Oral Argument Tr. 12:26–13:01.

<sup>13</sup> *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327 (Fed. Cir. 2016).

<sup>14</sup> *McRO, Inc. v. Bandai Namco Games Am. Inc.*, --- F.3d ---, 2016 WL 4896481 (Fed. Cir. Sept. 13, 2016).

original)). In *McRO*, we similarly held that claims that recited “a specific asserted improvement in computer animation” were not directed to an unpatentable abstract idea. 2016 WL 4896481, at \*8. By their terms and the district court’s unchallenged constructions, the Asserted Claims do not involve the use of a computer in any way. See J.A. 2080 (Synopsys’ counsel stating that “computers aren’t called out” in representative claim 1); Oral Argument Tr. 12:26–12:48 (Synopsys’ counsel conceding that the claims do not “speak[]” in terms of using a computer the way the specification does). The Asserted Claims thus cannot be characterized as an improvement in computer technology.

That a human circuit designer may not use the specific method claimed when translating a functional description of a logic circuit into a hardware component description of the logic circuit as Synopsys contends does not change this result. Indeed, the Supreme Court rejected this argument in *Gottschalk*. There, the Court reviewed a claimed “method for converting binary-coded decimal (BCD) numerals into pure binary numerals.” *Gottschalk*, 409 U.S. at 64. It recognized that the claimed method had been designed for use on a computer and “varie[d] the ordinary arithmetic steps a human would use by changing the order of the steps, changing the symbolism for writing the multiplier used in some steps, and by taking subtotals after each successive operation.” *Id.* at 67. It found that the claimed method, which “c[ould] be performed without a computer,” was nonetheless not patent-eligible. *Id.*

Synopsys’ argument that “[t]he [A]sserted [C]laims . . . do not preempt *all* conversions” from functional descriptions of logic circuits to hardware component descriptions of logic circuits, Appellant’s Opening Br. 18 (emphasis in original), likewise misses the mark. “While preemption may signal patent ineligible subject matter, the absence of complete preemption does not demonstrate

patent eligibility.” *Ariosa Diagnostics, Inc. v. Sequenom, Inc.*, 788 F.3d 1371, 1379 (Fed. Cir. 2015). “Where a patent’s claims are deemed only to disclose patent ineligible subject matter under the *Mayo* framework, as they are in this case, preemption concerns are fully addressed and made moot.” *Id.*

The district court did not define the abstract idea of the Asserted Claims. Synopsys likewise makes no proposal. Mentor Graphics argues that the Asserted Claims are directed to the abstract idea of “translating a functional description of an existing, intangible logic element into its corresponding assignment-condition description, and then into yet another abstract description of the same logic element.” Appellee’s Br. 28–29.

We recognize that defining the precise abstract idea of patent claims in many cases is far from a “straightforward” exercise. *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1257 (Fed. Cir. 2014). But, here, the Asserted Claims are drawn to the abstract idea of: translating a functional description of a logic circuit into a hardware component description of the logic circuit. As detailed above, this translation is a mental process. In contrast to Mentor Graphics’ articulation of the abstract idea, which largely restates representative claim 1 in different words, we believe our definition more accurately captures the “basic thrust” of the Asserted Claims. *BASCOM Global Internet Servs., Inc. v. AT&T Mobility LLC*, 827 F.3d 1341, 1348 (Fed. Cir. 2016). And, it is wholly consistent with the Gregory Patents’ own descriptions of the invention, as laid out in the Abstract, specification, and claims:

- “A method and system are provided for generating a logic network using a hardware independent description means.” ’841 Patent, Abstract.

- “This invention relates generally to methods and systems used to convert a hardware language description to a logic circuit . . .” *Id.* at 1:30–32.
- “A method for converting a hardware independent user description of a logic circuit . . . into logic circuit hardware components . . .” *Id.* at 62:61–65.

Having now defined the abstract idea of the Asserted Claims we turn to the second step of the *Alice* analysis.

## II. *Alice* Step 2: Do the Asserted Claims include an inventive concept?

In *Alice*, the Supreme Court described an “inventive concept” as “an element or combination of elements that is ‘sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the [ineligible concept] itself.’” *Alice*, 134 S. Ct. at 2355 (quoting *Mayo*, 132 S. Ct. at 1294) (alteration in original). Synopsys equates the inventive concept inquiry with novelty and contends that the Asserted Claims contain an inventive concept because they were not shown to have been anticipated by (35 U.S.C. § 102) or obvious over (35 U.S.C. § 103) the prior art. *See* Appellant’s Opening Br. 43 (“[T]he district court ignored the fact that the methods in the asserted claims of the Gregory patents were entirely novel solutions and could not be found anywhere in the prior art.”). That position misstates the law. It is true that “the § 101 patent-eligibility inquiry and, say, the § 102 novelty inquiry might sometimes overlap.” *Mayo*, 132 S. Ct. at 1304. But, a claim for a *new* abstract idea is still an abstract idea. The search for a § 101 inventive concept is thus distinct from demonstrating § 102 novelty.

That being said, the contours of what constitutes an inventive concept are far from precise.

In *DDR Holdings*, we held that claims “directed to systems and methods of generating a composite web page that combines certain visual elements of a ‘host’ website



with content of a third-party merchant” contained the requisite inventive concept. 773 F.3d at 1248. We explained that the claims at issue involved a technological solution that overcame a specific challenge unique to the Internet. *Id.* at 1259. This distinguished the claims at issue from those claims found unpatentable in earlier cases. *Id.* And, it ensured that the claims satisfied the *Alice* Step 2 inquiry under any conceivable articulation of the claims’ underlying abstract idea. *Id.* at 1257.

In *BASCOM*, we likewise held that claims “directed to filtering content on the Internet” contained an inventive concept. 827 F.3d at 1348. We recognized that “the limitations of the claims, taken individually, recite generic computer, network and Internet components, none of which is inventive by itself.” *Id.* at 1349. We explained, however, that “an inventive concept can be found in the non-conventional and non-generic arrangement of known, conventional pieces.” *Id.* at 1350. We found that the claims at issue contained just such an inventive arrangement through “the installation of a filtering tool at a specific location, remote from the end-users, with customizable filtering features specific to each end user.” *Id.* The claimed custom filter could be located remotely from the user because the invention exploited the ability of Internet service providers to associate a search request with a particular individual account. *Id.* This technical solution overcame defects in prior art embodiments and elevated an otherwise abstract idea to a patentable invention. *Id.*

The Asserted Claims, in contrast to those at issue in *DDR Holdings* and *BASCOM*, contain no such technical solution. To the extent the Asserted Claims add anything to the abstract idea (i.e., translating a functional description of a logic circuit into a hardware component description of the logic circuit), it is the use of assignment conditions as an intermediate step in the translation process. See Appellant’s Reply Br. 21 (“The use of as-

signment conditions in converting user descriptions into specific logic circuits is, without question, an inventive concept.”). But, given that the claims are for a mental process, assignment conditions, which merely aid in mental translation as opposed to computer efficacy, are not an inventive concept that takes the Asserted Claims beyond their abstract idea.<sup>15</sup> Unlike the claims at issue in *DDR Holdings* and *BASCOM*, the Asserted Claims do not introduce a technical advance or improvement. They contain nothing that “amounts to significantly more than a patent upon the [abstract idea] itself.” *Alice*, 134 S. Ct. at 2355 (citation omitted).

#### CONCLUSION

Whether different claims otherwise supported by the written description of the Gregory Patents directed to a computerized design tool that uses assignment conditions to more efficiently identify and generate logic circuits from a functional description could pass muster under § 101 is not before us. Our analysis focuses, as it must, on the Asserted Claims. Those claims are directed to an abstract mental process and contain no inventive concept. The claims are therefore invalid under 35 U.S.C. § 101. We *affirm* the district court’s grant of summary judgment.

#### AFFIRMED

#### COSTS

Each party shall bear its own costs.

---

<sup>15</sup> The inventive concept inquiry as it relates to the Asserted Claims thus differs from the one we often face in cases under § 101, i.e., whether the claimed invention is merely an abstract idea running on a general purpose computer as opposed to a concrete improvement in how the computer itself functions.