May 1995
Volume 77, No. 5
Pages 349–422
$4.00 US

JOURNAL of the PATENT and TRADEMARK OFFICE SOCIETY

In This Issue

Toward a Fact-Based Standard for Determining Whether Programmed Computers are Patentable Subject Matter: The Scientific Wisdom of *Alappat* and Ignorance of *Trovato*

James R. Goodman,[1] *Todd E. Marlette,*[2] *Peter K. Trzyna*[3]

I. Overview

A. *Building on the Prior JPTOS Article*

This article builds on our previous article "The *Alappat* Standard for Determining that Programmed Computers are Patentable Subject Matter."[4] In that article we elucidated the *Alappat* statement that "programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed."[5] Our elucidation involved a description of the science of what happens when a computer is programmed to reach the conclusion that a computer program literally makes a new machine by making an enormous number of electrical connections.[6] We also pointed out that any implementation carried out by programming a computer can also

1 James R. Goodman is a Professor of Computer Science, University of Wisconsin, Madison.

2 Todd E. Marlette is an associate in Washington, D.C.

3 Peter K. Trzyna is a partner at Keck, Mahin & Cate, Chicago, Illinois.

4 Goodman et al., "The *Alappat* Standard for Determining that Programmed Computers are Patentable Subject Matter," J. Pat. & Tm. Off. Soc., 10/94, pp. 771–786. We would like to acknowledge the assistance of Robert E. Astley, Vice President for Business Development for Doctor Design, Inc., 5415 Oberlin Drive, San Diego, California 92121-1716; 619-457-45445. We would also like to thank IBM, Motorola, VLSI, and Michael, Best & Friedrich for their kind letters to the Editor about our prior article which were published in J. Pat. & Tm. Of. Soc., 1/95, pp. 69–70 and 78–80.

5 *Alappat* at 1558.

6 The number of possible machines is on the order of $2^{16,000,000}$ (*i.e.*, 1 followed by more than 5 million zeros).

be carried out in hardware, so that premising patentability on one of the two makes no scientific sense.

More specifically, our prior article chronicled how one of the first programmable computers, the ENIAC, was programmed with telephone patch cords such that the only difference between a programmable computer and a hard wired computer was solder; how over time, the patch cords were replaced with soldered wires having switches to make programming easier; eventually, the number of possible electrical connections were increased but the scale of the circuitry was reduced by placing the circuitry on chips; but how in the end, programming is still making circuitry. In other words, programming a computer is just another way of making circuitry, and the exact same circuitry or equivalent circuitry[7] can be hard wired. We therefore concluded that the *Alappat* decision was scientifically correct.

B. *Applying the Rationale of* Alappat *and Our Prior JPTOS Article to* In re Trovato[8]

In this article we continue the view that a fact-based approach to understanding patentable subject matter is necessary for credibility. More specifically, the instant article analyses the scientific underpinnings of the subsequent *Trovato* decision.

1. Logic Circuits, Shifters, and Read Only Memory

Generally, *Trovato* held that claims directed to a programmed computer[9] were not patentable subject matter and distinguished the holding from that in *Alappat* as follows:

> Our result here comports with our recent decision in *Alappat* (cites omitted). Although the claims of the inventor in *Alappat* were also drafted in means format, unlike the disclosure here, his application disclosed a specific hardware embodiment. There, we extensively relied upon the hardware listed in the specification, including *arithmetic logic circuits, barrel shifters*[10] *and a read only memory*[11] *in*

7 An equivalent circuit receives the same input and produces the same output as another circuit.

8 33 U.S.P.Q.2d 1194 (Fed. Cir. 1994).

9 Claim 41, for example, is directed a "computer apparatus. . . ."

10 A shifter is part of an arithmetic logic circuit, or unit, that moves bits to the left or right.

11 In our prior article we criticized the idea that patentability can be premised on the selection of the storage device—such as a read-only memory (ROM)—for the computer program. We stated the following:

> *Diamond v. Bradley*, 450 U.S. 381 (1981) can be interpreted as establishing that ROM-implemented processes are patentable even if the same process, computer program implemented, would not. However, at least one legal commentator, D. Davidson, observed the following:
>
> This would be silly. The ROM in *Bradley* could have been replaced in the computer with a volatile RAM (read-

reaching the result that the claimed invention constituted patent eligible subject matter (cites omitted). Specific note was also made of the combination of claimed elements from which the inventor formed a machine. *Id.* at 1544, 31 U.S.P.Q.2d at 1557. As we have noted, however, *a search through Trovato's application for the combination of similar apparatus is unavailing.*

(Emphasis added).

However, virtually all personal computers have "*arithmetic logic circuits, barrel shifters and a read only memory*," as a simple review of, say, a Motorola microprocessor data book would readily show.[12] Even a check of the *Encyclopedia of Computer Science and Engineering*[13] would show the same.

Arithmetic logic circuits are the heart of a computer and must be present for the circuitry to qualify as a digital computer. A barrel shifter, or a functionally equivalent circuit, is also present in any digital computer. All digital computers also contain a memory device, which differs from ROM only in that other memory device has additional functionality—the ability to be written.

Although the *Trovato* decision should be criticized for showing so little a grasp of the technology at issue that it reflects badly on the Federal Circuit,[14] more interesting insights can be gleaned from the reasoning of the decision.

Over the past year, there have been at least four Federal Circuit decisions involving the patentability of programmed computers:[15] *In re*

---

write) memory chip which looks like a ROM; the software could first be read into the RAM, then the process performed.

The *Bradley* decision is indefensible from a factual, *i.e.*, scientific, point of view.

Nonetheless, *Trovato* relies heavily on *Bradley* and looks for the existence of a ROM.

12 See, for example, the Motorola Microprocessor Data Book MC 68HC05C3 *Microprocessor, Microcontroller, and Peripheral Data*, Vol. 1, Motorola Inc. (1988), wherein at page 3-819 a block diagram shows the arithmetic and logic units and a read-only memory; FIG. 17 on page 3-839 shows a barrel shifter arrangement. Apple computers use chips from Motorola.

13 (2d Ed), (Ralston et al. Eds., Van Nostrand Reinhold Co., 1983) at 103–106 (discussing the ALU of a computer and shifters in particular at 105) and at 1264–1265 (discussing the use of ROMs in computers). See also, N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, (Addison-Wesley 1985) p. 366 ("Barrel shifters are important elements in many microprocessor designs").

14 It is ironic that *Trovato* contends that the patent applications "provide no grasp of the underlying physical process" because it is clearly the panel that shows the lack of grasp.

We have previously criticized the carelessness in the dissenting opinion in *Alappat*: "Actually, Judges Archer and Nies are mistaken. Alappat's brief makes no such statement." Goodman, et al. at 781. Nies authored *Trovato*.

15 One could find the beginning of this line of cases in *Arrhythmia Research Technology, Inc. v. Corazonix Corp.* 958 f.2d 1053 (Fed. Cir. 1992), which was cited favorably *in banc* in *In re Donaldson* 16 F.3d 1189 (Fed. Circ. 1994).

Alappat, *In re Warmerdam*,[16] *In re Lowry*,[17] and *In re Trovato*. Only *Trovato* concluded that the subject matter was unpatentable, and thus only *Trovato* had to distinguish itself from the other decisions. But they all involve programmed computers. One programmed computer cannot be patentable as a machine and another not: they are all programmed computers. There is no scientifically credible basis for distinguishing one from another,[18] particularly with a test to determine which one is a machine.[19] Thus, while the attempt to make such a distinction in *Trovato* is blatantly ignorant of technical facts, no other attempt would be more credible. Any such attempt would result in a "scientifically untenable decision," as we pointed out in our previous article.

2. Paucity of Structure

Trovato refers to "a paucity of structure disclosed in Trovato's specification." Elsewhere, *Trovato* states the following:

> . . . the specifications involved here provide no grasp of any underlying physical process. Although cursory references to such diverse apparatus as robots, dynamic emergency exit routs and electronic maps are present, *no computer architecture is provided, no circuit diagram is revealed, and no hardware at all receives more than brief mention*. When questioned during oral argument before this Court, counsel for Trovato admitted that neither specification includes a hardware enablement of the claimed invention. Instead, the entire *disclosure consists of flow charts and program code* computing the least cost path from starting to goal states based upon the configuration space. We therefore conclude that Trovato claims nothing more than the process for numerical calculation. Simply stated, viewing the claims absent the algorithm, and as a whole, no statutory subject matter is present.
>
> (Emphasis added.)

16 33 F.3d 1354 (Fed. Cir. 1994).

17 32 F.3d 1379 (Fed. Cir. 1994). Note that *Lowry* was a printed matter rejection rather than a mathematical algorithm rejection. the holding was that a data structure in an electrical computer is a physical structure, and therefore implicitly that the structure is patentable subject matter under Sec. 101.

18 While they are all programmed computers, one differs from another in the specific programming and thus the specific circuitry. This difference has to do with novelty and unobviousness, not with the existence of a machine. All programmed computers or hard wired computers are machines, and there is no factually credible way of concluding that some are and some are not.

19 35 U.S.C. § 101 provides that "Whoever invents or discovers any new and useful process, machine . . . or any new and useful improvement thereof, may obtain a patent therefor. . . ."

a. Circuitry is Literally Formed

Trovato overlooks that programming a computer makes new circuitry as a matter of fact, as detailed in our prior article.[20] This comes about from making electrical connections in such components as 16 Mbit (16 million memory cell)[21]—on the order of $2^{16,000,000}$ distinct circuits (*i.e.*, 1 followed by more than 5 million zeros). Programming a computer with software makes these connections in the same way that programming the ENIAC with telephone patch cords makes new circuitry.

It is no wonder that in the patent applications of *Trovato*, "*no computer architecture is provided, no circuit diagram is revealed, and no hardware at all receives more than brief mention.*" These are completely unnecessary for programming a computer, and programming a computer inherently makes the circuitry—this is how a computer operates.[22]

b. Computer Architecture/Diagrams

If *Trovato* can be understood to require a disclosure of computer architecture and hardware of the programmed computer, then the way to obtain a patent is to simply convert the software implementation into a hardware implementation of the programmed computer, obtain a patent on the hardware, and show infringement under the doctrine of equivalents.[23]

Converting from software to hardware is routine, either with a ROM-burner, or one could simply use commercially available software

20 Goodman et al., "The *Alappat* Standard for Determining that Programmed Computers are Patentable Subject Matter," J. Pat. & Tm. Off. Soc., 10/94, pp. 771–786. Note too the statement in *Alappat* that "programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed. . . ."

21 As shown by Bakoglu in Table 4.1. See also, Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Chapter 4.3, Random Access Memory (RAM), p. 145 (Addison-Wesley 1990).

22 The program unambiguously defines the machine. If anyone really wanted to know the architecture and circuit diagrams of a particular programmable computer, they could review the data books for the equipment and study the registered chip mask works.

23 In our prior article, we discussed cases finding infringement under the doctrine of equivalents applied to hardware/software embodiments, including *The Magnavox Co. and Sanders Assoc., Inc. v. Mattel, Inc.*, 216 U.S.P.Q. 28 (N.D. Ill. 1982), *Magnavox Co., Inc. v. Chicago Dynamic Industries*, 201 U.S.P.Q. 25 (N.D. Ill. 1977), *Decca Ltd. v. United States*, 554 F.2d 1070 (1976), *aff'd in part, modified in part, and rev'd in part*, 640 F.2d 1156 (Ct. Cl. 1980), *cert. denied*, 454 U.S. 819 (1981), *Hughes Aircraft Co. v. United States*, 717 F.2d 1351 (Fed. Cir. 1983), *Lockheed Aircraft Corp. v. United States*, 553 F.2d 69 (Ct. Cl. 1977), *Arshal v. United States*, 202 U.S.P.Q. 749 (1979), *modified*, 621 F.2d 421 (Ct. Cl. 1980), *Dynamics Corp. of America v. United States*, 5 Cl. Ct. 591 (1984), and *IVAC Corporation v. Terumo Corp.*, Case No. 87-0413-B(M) (S.D. Cal. 1989).

tools[24] for designing circuitry based on a "*disclosure [that] consists of flow charts and program code.*" For example, Synopsys, Inc.[25] is one of many vendors of such circuit design software, and companies like Doctor Design, Inc.[26] can do the work for you. Thus, requiring a hardware disclosure would be a mindless increase in the cost for a patent application, particularly where the best mode of the invention is a software implementation.

c. Software/Hardware Conversion

i. ROM-burner

A ROM-burner is a commercially available machine used to create a ROM from a computer program. To make a ROM with a ROM-burner, a computer program is written and stored in the RAM of a computer. The stored computer program is then dumped into the ROM-burner, which then uses electricity to burn out connections and thereby embed the program in the ROM. Copying a computer program from a RAM indelibly into the ROM is just another way of storing the exact, same computer program. Then, one simply replaces the RAM with the ROM (which has the same number of pins,[27] pin spacing, and electrical requirements as the RAM) in a trivial process.

If the Federal Circuit in *Trovato* is going to require a ROM for patentability—no problem. The cost to have a computer program stored in a ROM is about three hundred dollars, which is less than the cost for a small entity to file the patent application.

ii. Circuit Design Software Tools

Another approach is to use circuit design software tools make circuitry by employing circuit modules, which can yield more efficient designs than those developed by merely storing the computer program in ROM. These modules can be simple functions, like gates, or more complex functions like registers, buffers, and ring counters. In the circuit design software tools, these and other functions have a standard cell definition so that one can use cell logic to design circuitry, much

24 Using software tools is a largely automated process of going from the code and flow charts to circuit designs. The use of the software tools takes less time than manually doing schematic design by hand, particularly for very large gate count devices.

25 Synopsys, Inc., 1505 LBJ Freeway, Suite 340, Dallas, Texas 75234: 1-800-344-0004. Synopsis products include its Behavioral Compiler and the ModelSource family of products.

26 Robert E. Astley, Doctor Design, Inc., 5415 Oberlin Drive, San Diego, California 92121-1716: 619-457-4545.

27 However, the write pin on a ROM is no longer needed.

like one can build a house out of Lego building blocks.[28] All the standard cell components are plugged together to expediently produce designs for circuitry having arrays with vast numbers of gates. For a skilled person using the software tools, a hard wired version of a computer program can be gracefully completed in less than a few weeks.[29]

More specifically, the conversion process begins by specifying the input requirements, the output requirements, and the flow chart definitions of the way that input would be used to generate the output.[30] The circuit design tools use this information to produce depictions of a series of digital logic gates in a field programmable gate array (FPGA) or in an application specific integrated circuit (ASIC).

A field programmable gate array is a device which is already largely configured; it just needs final specifications to connect the gates. A FPGA is a relatively expensive device. For example, to individually "program" each 5,000 gate FPGA will cost several hundred dollars per FPGA, but it may replace several thousand dollars worth of less integrated hardware.

One can replace a FPGA with an ASIC, *i.e.*, a customized silicon chip. The software tools generate a chip mask work having the gates, and the chip mask work is used to make the chip. Initially, producing an ASIC is expensive, *e.g.*, $100,000. But for a high volume, perhaps for some mass produced device, in the long run, it would be cost effective to change from a FPGA costing several hundred dollars to an ASIC implementation, which then may cost several dollars per component.

The ASIC can be inserted in place of the FPGA, which in turn can be inserted to replace the programmed microprocessor—all to perform exactly the same functions specified by the software.

---

28 The approach is an automated version of the manual approach hardware designers employing 7400 Series logic would follow a few years ago.

29 Robert E. Astley, referring to the services of Doctor Design, Inc. Mr. Astley is the United Kingdom equivalent of a U.S. Registered Electrical Engineer and is the Vice President of Business Development for Doctor Design, Inc., 5415 Oberlin Drive, San Diego, California 92121-1716: 619-457-4545.

30 Converting from a software implementation to a hard wired implementation is routine because the operations of a programmed digital electrical computer literally are switching circuitry operations, and thus, are replicatable. A programmed computer operates by executing functions sequentially or however they are programmed within the memory of the computer. Thus, the executing is a purely digital and logical sequence, which can be recreated by a hard wired structure of gates controlled by timers. From a technical point of view, it is therefore a routine task to go from flow charts and code to hard wired circuitry.

Of course the process can be done in the reverse—that is, going from a hard wired implementation to a programmed implementation.[31] The process involves deriving the functions of the hardware, which can easily be done because every piece of hardware has specifications: A set of inputs, a set of outputs, a functional capability. Every aspect of that hardware can be generated in its equivalent flow chart form by a programmer working in whatever language he or she prefers, *e.g.*, C, Fortran, or Pascal. In sum, one takes the specifications of the device and its function, and represents them in a flow chart with whatever timing characteristics and whatever input/output (I/O) devices are used. Then it is a straightforward process of writing code according to the specifications of the flow chart.

In one example, converting hardware containing a field programmable gate array (for which there was no definition except for the way it was connected to the rest of the circuitry) into computer program code took less than two weeks.[32]

Converting a programmed computer implementation into a hardware implementation, or vice versa, is a routine undertaking that is not considered in the *Trovato* decision. Patentability should not be premised on a routine technicality (a disclosure in hardware rather than in software).[33]

31 Although the circuitry created by the tools could be identical to the programmed circuitry, it probably is equivalent the circuitry. (Two different circuits that produce identical output from identical input are known in electrical engineering as equivalent circuits.) That is, the logical definition for performing, say, multiplication in either hardware or software is identical, so hard wired circuitry must perform the same function as programmed circuitry. However, the hard wired circuitry would probably be somewhat differently structured to reflect the technology employed in building the circuit—*e.g.*, if the circuit is not put on a chip, it would operate at a higher voltage, etc. However, the use of different but equivalent structures to performing the same functions is not unique to converting from software to hardware. Two circuit designers could implement the same function with different hard wired circuitry.

32 Robert E. Astley, referring to the services of Doctor Design, Inc., 5415 Oberlin Drive, San Diego, California 92121-1716: 619-457-4545.

33 Another disturbing aspect of *Trovato* is that 35 U.S.C. 112 requires disclosing how to make and use the invention and the best mode of the invention known by the inventor at the time the application is filed, but § 112 does not require architecture or circuit diagrams where they are not the best mode. From the disclosure of flow charts and code provided in the Trovato patent applications, to make and use the invention requires little more than typing in the code into a programmable digital electrical computer. Providing architecture and circuit diagrams is not the duty of the inventor seeking to disclose how to make and use the best mode of the invention, the requirement in Sec. 112.

C. *Denying Hardware/Software Equivalence Leads to Inconsistencies and Scientifically Untenable Decisions*[34]

Trovato muddies the water recently clarified by *Alappat*, *Warmerdam*, and *Lowry*. Because all these inventions involve programmed computers, no factual criteria can distinguish which one is any more of a machine than another. It is to be expected that the attempt to do so in *Trovato* would be incomprehensible from a scientific point of view.

As it stands today, the answer to the question of whether a programmed computer is patentable subject matter as a machine depends on a random draw of judges on the Federal Circuit, and the random draw may or may not produce a factually credible decision.

1. Warmerdam

Consistent with the fact-based approach used by the majority in *Alappat* and detailed in our prior article, *Warmerdam* makes factual sense because it recognizes that a programmed computer is a machine.[35] In *Warmerdam*, method claims 1–4 were unpatentable based on § 101 because they were not limited to § 101 subject matter. However, the court found patentability with the additional requirement specified in claim 5:

> A machine having a memory which contains data representing a bubble hierarchy generated by the method of any of the Claims 1 through 4.

The requirement of "a machine" was satisfied by

> any machine (presumably including a known computer) having a memory which contains any data representing a bubble hierarchy determined by any of the method claims 1–4.[36]

Thus, merely requiring that the method be carried out on a programmed computer was sufficient to make the invention patentable subject matter under *Warmerdam*.

Although *Warmerdam* skates over any technical detail of its patentable subject matter reasoning, the decision inherently makes scientific sense. Directing the claims to a programmed computer inherently directs the claims to circuitry. As a matter of fact, claim 5 in *Warmerdam* does require the circuitry of a "machine."

---

34 Actually, this is a subheading from our prior Article and our reasoning is equally applicable here.

35 As a machine, it is patentable subject matter under § 101.

36 *Warmerdam* at 1361.

Trovato cannot be reconciled with *Warmerdam*. The discussion of *Warmerdam* in *Trovato* is as follows:

> For the purposes of the determination of statutory subject matter, we find these claims scarcely distinguishable from those before the Court in its recent decision in *In re Warmerdam* (cite omitted). In *Warmerdam*, this Court held that claims reciting a method for creating a data structure which controlled the motion of objects did not constitute patent eligible subject matter. Citing the difficulties in determining the proper boundaries of the nonstatutory category of mathematical algorithms, *Warmerdam* did not proceed by employing the latter term. The court instead reasoned that the claimed method was nothing more than the manipulation of abstract ideas, rather than speaking of a mathematical algorithm. (cite omitted). *See also In re Alappat* (cite omitted) "the Supreme Court never intended to create any overly broad, fourth category of subject matter [mathematical algorithms] excluded from § 101").
>
> As in *Warmerdam*, Trovato's claims operate merely in the domain of abstract ideas. The methodical application of arithmetic operations to data placed within a numerical configuration in order to determine the least cost path through a mathematically structured graph amounts only to a generality or disembodied concept, outside the subject matter listed in § 101. Without further application or connection to a technical art, we cannot say that Trovato's claims pass muster under the alternative analysis of statutory subject matter expressed in *Warmerdam*.

The reasoning in *Trovato* sidesteps what made claim 5 patentable in *Warmerdam*—the requirement of a machine "including a known computer." *Trovato*'s claims all explicitly require this apparatus. But the lack of an explanation as to why a programmed computer implementation is patentable subject matter in *Warmerdam* but not in *Trovato*, again, is perhaps not merely another technical lapse or oversight in judicial craftsmanship. The lack of a coherent explanation is a revelation that there is no subject matter difference: both require programmed computers, and thus there is no coherent way to factually distinguish the subject matter.

2. Lowry

Lowry too is consistent with *Alappat* and the fact-based approach of our prior article, but *Lowry* is irreconcilable with *Trovato*. *Lowry* involved a printed matter rejection of claims directed to a data structure in the memory of a programmed computer—the U.S. Patent and Trademark Office (PTO) contended that the claims covered the electronic equivalent to printed matter. However, the Federal Circuit found that memory in a programmed electrical computer involves physical struc-

ture, thereby implicitly holding that such structure is patentable subject matter under § 101.

In our prior article, we pointed out the following:

To better understand the physical structure of a Random Access Memory, a single memory cell is examined. In an illustrative example,[37] a J-K flip flop may be formed from a plurality of NAND gates (as previously discussed).[38] A J-K flip flop has essentially two input lines, J and K, an output Q and a clock input or trigger. For each combination of predetermined input values along input lines J and K at the time that the clock signal is triggered, a predetermined electrical pathway is formed which corresponds to the values of each of the NAND gates contained therein. Thus, just as each of the individual NAND gates forms a distinct machine in response to predetermined input, a combination of NAND gates, in the form of a J-K flip-flop, will form a distinct machine in response to its input.

A second kind of memory cell is the CMOS static memory cell which may be formed directly from six (6) separate transistors.[39] Each of the transistors may be characterized as an open circuit or as a short circuit in response to the input data lines or "BIT" lines. Thus, the same way that a NAND gate forms a new circuit based upon the A and B inputs, the CMOS static memory cell forms a new circuit based upon the "BIT" line inputs.

Most memory cells, including J-K flip flops and CMOS static memory cells, incorporate a pair of cross-coupled, self-driving circuits.[40] Each circuit remains indefinitely in either of two distinct states to retain information.[41] Each RAM cell, through its transistors, forms one of two distinct and separate circuits.

Lowry correctly recognized that

. . . the stored data exist as a collection of bits having information about relationships between [attributive data objects]. Yet this is the essence of electronic structure.[42]

Trovato does not mention *Lowry*, which may seem odd because the claims of both are explicitly directed to data structured in memory.[43]

37 An example of a memory cell is shown in Figure 10-12(a) of Pucknell et al., *Basic VLSI Design Systems and Circuits*, Chapter 10, Memory, Registers, and Aspects of System Timing, p. 220 (Prentice-Hall 1988).

38 The J-K flip flop as shown in Pucknell, et al. is constructed from 9 NAND gates.

39 See Figure 4.7(a) of Bakoglu, *supra*, at 144. Although this section is a direct quote from our prior article, it may be more accurate to say that only 4 of the 6 transistors can be so characterized, as the other 2 transistors are used passively.

40 Accordingly, *Lowry* is correct as a matter of fact that a configuration of memory cells is patentable subject matter (as a new machine); the Patent and Trademark Office view, to the contrary, that it is printed matter, is scientifically untenable.

41 Mead et al., *Introduction to VLSI Systems*, Chapter 1.14, Properties of Cross-Coupled Circuits, p. 26 (Addison-Wesley 1980.).

42 *Lowery* at 1583.

43 Compare *Lowery*'s claim 1 with *Trovato*'s claim 22.

Instead the *Trovato* panel states that it is unconvinced that "the data structure is a physical entity, consisting of electrical or magnetic signals and requiring interaction between the processing and memory apparatus."[44]

Unfortunately, from a factual point of view, as outlined in the above-quoted portion of our article, the *Trovato* panel is simply wrong as a matter of electrical engineering fact.

Again, a memory structure cannot be patentable subject matter in *Lowry* but not in *Trovato*. There is no subject matter difference—both require memory structures in programmable computers, and thus as a matter of fact, both require new circuitry.

Claim 1 (of *Lowery*) A memory for storing data for access by an application program being executed on a data processing system, comprising:

a data structure stored in said memory, said data structure including information resident in a database used by said application program and including:

a plurality of attribute data objects stored in said memory, each of said attribute data objects containing different information from said database;

a single holder attribute data object for each of said attribute data objects, each of said holder attribute data objects being one of said plurality of attribute data objects, a being-held relationship existing between each attribute data object and its holder attribute data object, and each of said attribute data objects having a being-held relationship with only a single other attribute data object, thereby establishing a hierarchy of said plurality of attribute data objects;

a referent attribute data object for at least one of said attribute data objects, said referent attribute data object being nonhierarchically related to a holder attribute data object for the same at least one of said attribute data objects and also being one of said plurality of attribute data objects, attribute data objects for which there exist only holder attribute data objects being called element data objects, and attribute data objects for which there also exist referent attribute data objects being called relation data objects; and

an apex data object stored in said memory and having no being-held relationship with any of said attribute data objects, however, at least one of said attribute data objects having a being-held relationship with said apex data object.

Claim 22 (of *Trovato*) Apparatus for determining motion of an object [sic, the] apparatus comprising:

a. *a memory for storing a discretized configuration space in the form of a configuration space data structure which includes a plurality of states corresponding to a discretized subset of all physical poses of the object in a physical task space, the states being arranged so that each state has a plurality of neighboring states situated along respective permissible transition directions;*

b. means for initializing the states;

c. means for storing goal information, corresponding to a physical goal pose, in a respective goal state;

d. means for measuring cost of transition from each state to its neighboring states along the permissible directions according to a space variant metric function;

e. means for propagating cost waves from the goal state using the measuring means, the propagating means assigning to each state a cost value and a direction of travel corresponding to a least cost path from the physical pose corresponding to the sate of the physical goal pose;

f. means for determining a series of discrete states along the path, the determining means determining the path states by starting at a start state corresponding to a physical state pose and following the cost and direction of travel values assigned to the start state and succeeding states; and

g. means for transforming the series into an electronic form usable by the object to follow the path.

44 *Trovato* at 1198.

The *Trovato* panel seeks to distinguish the purpose of the circuitry from that in the other above-cited Federal Circuit cases by reasoning that the *Trovato* claims recite a mathematical algorithm.[45] But as recognized in *Alappat*, the issue posed by Congress in § 101 is whether there is "any" machine, not whether the machine is defined by an algorithm. The proper test under § 101 was articulated in *Alappat*:

> . . . the Supreme Court explained that there are three categories of subject matter for which one may not obtain patent protection, namely "laws of nature, natural phenomena, and abstract ideas."[46]

By this standard, *Alappat* and *Trovato* equally define patentable subject matter—regardless of the purpose of the circuitry or whether one or the other involves an algorithm.[47] As in *Alappat*, *Trovato*'s claims are not "a disembodied mathematical concept which may be characterized as an 'abstract idea,' " but are instead "a specific machine."[48]

The *Trovato* panel sidestepped *Lowry* because there is no scientifically credible basis to distinguish it.

3. Summary

Trovato cannot be credibly explained as lacking the subject matter in *Alappat*, *Warmerdam*, and *Lowry* because they all involve exactly the same subject matter—programmed computers. Either programmed computers are machines or they are not, and therefore they are patentable subject matter or they are not.

From a factual point of view, as there is no distinction between hard wired circuitry and circuitry formed by setting the switches in a programmable computer. Every digital electrical computer uses hardware to different degrees, before the computer is programmed. Whatever is not done in hardware can be done in software. For example, although it is easy to create, multiply and divide functions in software, many computers use a chip or a portion of a chip to provide these same functions in hardware.[49] There is literally no difference between performing these functions either way.

---

45 *Trovato* disputes that the claims cover an algorithm. Of course, as pointed out in our prior article "it is rather common in electrical engineering to define digital circuitry functionally by the mathematical operation it performs. As stated by Judge Rader in his concurring opinion in *Alappat*, 'Mathematics is simply a form of expression—a language.' "

46 *Alappat* at 1542.

47 Again, Trovato disputes that the claims even involve an algorithm.

48 *Alappat* at 1557.

49 See, for example, Motorola data books, generally.

As an even simpler example, consider the wiring for a light bulb. The same circuit is formed by either soldering wires to the light bulb or by soldering the wires to the light bulb via a switch turned "on" (and all the software does is turn the switch "on"). Either approach makes no difference to an electron floating along the circuitry. Premising the patentability decision on a hardware/software distinction involves viewing the same circuit as patentable if the electricity comes directly from the wires, but unpatentable if the electricity comes from the wires via the switch—because the same circuitry exists in either approach, the attempted distinction only reflects badly on the scientific credibility of the patentability decision.

Further, as to the *Trovato* panel's "algorithm" thinking, the purpose of the circuitry has no bearing on the existence of the circuitry. Put another way, it does not matter whether the illuminated light bulb represents the solution to a mathematical equation[50] or whether the light bulb represents a dot produced by a rasterizer,[51] in either case, circuitry exists.

Trovato makes no factual sense because its reasoning fails to comprehend what is elementary to an electrical engineer—programming a microprocessor is just another way of making circuitry.[52]

50 This example also shows the inherent factual failing of the so-called "*Freeman—Walter—Abele* test."

51 Compare the claims in *Alappat*.

52 Recently, in an attempt to block a patent on a programmed computer, the PTO revived the rejection that a programmed computer invention is inherently obvious over any generic computer. (Ser. No. 08/123,312) One can hardly say that a computer inherently renders obvious all possible computer programs. For a given moment, a 16 Mbit (16 million memory cell) capacity can form a number of electrical pathways on the order of $2^{16,000,000}$ (*i.e.*, 1 followed by more than 5 million zeros, a number that dwarfs the number of possibilities in *In re Baird* 29 U.S.P.Q.2d (Fed. Cir. 1994) by millions fold). Given this number of possibilities, one cannot believably contend that the existence of a microprocessor renders obvious all programmed settings. Otherwise all prime numbers, cryptography and game theory, for example, would be known, and computer science departments would be all but moot. Further, under this reasoning, no computer program could be patented subsequent to the development of a generic computer.

In making circuitry to execute instructions, whether via hard wiring or via programming, designing the circuitry is not as difficult as knowing what instructions to execute. For example, articulating and balancing every principle involved in winning a chess game is quantum more difficult than writing a computer program that carries out the balanced principles.

Accordingly, programming a microprocessor is merely another way of making circuitry, and regardless of how the circuitry is formed, executing instructions in circuitry is usually easier than knowing what instructions should be executed.

C. *Conclusion*

Our prior article elucidated the fact-based approach of *Alappat* to determining that programmed computers are patentable subject matter and concluded:

> The view of *In re Alappat* that "programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed. . . ." is the key to understanding how patentable subject matter requirements apply to computer program-related inventions. It is simply an application of the plain meaning of 35 U.S.C. § 101 to the facts. The Federal Circuit majority is correct in its approach.

Applying the same fact-based approach to *Trovato* leads to only one conclusion: From a scientific point of view, *Trovato* is an embarrassment to the Federal Circuit. Virtually all personal computers have "*arithmetic logic circuits, barrel shifters and a read only memory*," and this provides no basis for distinguishing *Trovato* from *Alappat*. But there is no more credible way to attempt to explain why "programming creates a new machine" in *Alappat*[53] but not in *Trovato*, which reveals the true failing of the thinking in *Trovato*. The subject matter is the same, they are both digital electrical computers, and as a matter of fact, regardless of whether the respective circuitry is formed by programming or by hard wiring, the circuitry is the same or is equivalent. There can be no coherent distinction because the subject matter is the same.

As we stated in our prior article, scientifically untenable decisions result from not using a fact-based approach (like that used in *Alappat*) to determine patentability of programmed computers.[54] Hopefully, the Federal Circuit will grant the Petition for Rehearing[55] in *Trovato* to restore some credibility and predictability to the law.

53 As pointed out above, the same reasoning was inherently controlling in *Warmerdam* and *Lowery*.

54 It is indeed ironic that *Trovato* would conclude with the following statement: "The presence of patent eligible subject matter must always be determined upon the individual facts of each case."

55 Filed January 30, 1995.