UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/369,693	02/09/2012	Erich James Plondke	QC111429	7143

12371 7590 05/23/2018
Muncy, Geissler, Olds & Lowe, P.C./QUALCOMM
4000 Legato Road, Suite 310
Fairfax, VA 22033

EXAMINER
VICARY, KEITH E

ART UNIT	PAPER NUMBER
2182	

NOTIFICATION DATE	DELIVERY MODE
05/23/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

meo.docket@mg-ip.com
meo@mg-ip.com
ocpat_uspto@qualcomm.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte ERICH JAMES PLONDKE, LUCIAN CODRESCU,
CHARLES JOSEPH TABONY,
and
SWAMINATHAN BALASUBRAMANIAN[1]

Appeal 2018-000643
Application 13/369,693
Technology Center 2100

Before CARLA M. KRIVAK, HUNG H. BUI, and JON M. JURGOVAN, *Administrative Patent Judges*.

KRIVAK, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellants appeal under 35 U.S.C. § 134(a) from a Final Rejection of claims 1–3, 5, 6, 18, 20, and 21, which are all the claims pending in the application. We have jurisdiction under 35 U.S.C. § 6(b).

We reverse.

---

[1] Appellants identify the real party in interest as QUALCOMM Incorporated (App. Br. 3).

STATEMENT OF THE CASE

Appellants' invention is directed to "[s]ystems and methods for generating a floating point constant value from an instruction" that includes "a first field corresponding to a sign bit of the floating point constant value," "a second field corresponding to an exponent value of the floating point constant value," "a third field corresponding to a significand of the floating point constant value," and two additional fields indicating "first and second shift values" (Spec. ¶¶ 8, 28; Abstract). "[T]he second field and the third field . . . [are] shifted by [the] first and second shift values respectively" before "[t]he first field, the second field, and the third field are combined to form the floating point constant value" (Abstract).

Claims 1 and 18 are independent. Independent claim 1, reproduced below, is exemplary of the subject matter on appeal.

> 1. A method of generating a floating point constant value from an instruction, the method comprising:
>
> decoding a first field of the instruction as a sign bit of the floating point constant value;
>
> decoding a second field of the instruction to correspond to an exponent value of the floating point constant value;
>
> decoding a third field of the instruction to correspond to a significand of the floating point constant value;
>
> shifting the second field, based on a first shift value, and the third field, based on a second shift value, wherein a fourth field of the instruction comprises the first shift value and a fifth field of the instruction comprises the second shift value.

REFERENCES and REJECTIONS

(1) The Examiner rejected claims 1–3, 5, 6, and 18 under 35 U.S.C. § 103(a) based upon the teachings of Ishii (US 2006/0112160 A1;

published May 25, 2006) and Ford (US 2005/0154773 A1; published July 14, 2005).

(2) The Examiner rejected claims 20 and 21 under 35 U.S.C. § 103(a) based upon the teachings of Ishii, Ford, and Trissel (US 5,341,320; issued Aug. 23, 1994).

ANALYSIS

With respect to claim 1, the Examiner finds the combination of Ishii and Ford teaches "shifting the second field, based on a first shift value, and the third field, based on a second shift value, wherein a fourth field of the instruction comprises the first shift value and a fifth field of the instruction comprises the second shift value," as claimed (Final Act. 3–4; Ans. 19–22). Particularly, the Examiner finds Ishii teaches second and third fields of an instruction, the second and third fields corresponding to an exponent value and a significand, respectively, of a floating point constant value (Final Act. 3 (citing Ishii Fig. 3A)). The Examiner recognizes Ishii does not teach *shifting* instruction fields as claimed, but finds Ford teaches an instruction encoding "a rotation (i.e. a circular *shift*)" of an 8-bit *integer constant* for shifting the integer (Ans. 20; Final Act. 4 (citing Ford ¶ 9)). Based on these factual findings, the Examiner concludes a skilled artisan would have *shifted* the *significand* and *exponent fields* of an encoded *floating point* constant value as claimed, because "the basic idea included in Ford"—that "a particular value can be indirectly specified by an instruction . . . including . . . a rotate value"—is "applicable regardless of whether the particular value is an integer constant or an exponent or a significand of a floating point constant" (Ans. 17, 22). We do not agree.

We agree with Appellants that Ishii and Ford, alone or in combination, fail to teach or suggest shifting both exponent and significand fields of an instruction by "*two* shift operations to be performed in the generation of a floating point constant value based on two encoded shift value fields" of the instruction, as required by claim 1 (App. Br. 8–9). Rather, Ishii and Ford generate *floating point constant values* without "*any shifting* of immediate [i.e., instruction-encoded] values *based on fields encoded within an instruction itself*" (Reply Br. 3 (emphasis added) (citing Ford ¶¶ 40–44, Figs. 3–4); App. Br. 9).[2]

We also do not find the Examiner has provided sufficient evidence to support the finding that the skilled artisan would modify Ishii's floating point value instruction to include two encoded shift value fields, based on Ford's teaching of an integer's rotation. As Appellants point out, Ford's "described aspects of integer constants, *including the above enhancements which involve a further rotation*, do not apply to floating point constant values" (App. Br. 7 (citing Ford ¶¶ 11–12)). This is because Ford's "'rotation' has a specific meaning in the case of integer constants (described as *changing a location of the immediate value within a register* to fill the remaining bits with predetermined sequences of ones or zeros)" and "is not applicable in the case of shifts made to a significand/exponent [of a floating point constant value]" because "shifting the significand or the claimed second [exponent] field, e.g., to normalize a floating point number would not involve simply *filling* the entire register with a *predetermined* sequence of zeros or ones" (App. Br. 6; *see* Ford ¶ 11). The Examiner's Answer does

[2] We count the pages of the Reply Brief (which are not numbered) starting from the first page.

not respond to these arguments by Appellants, and merely restates that a "basic idea [of circular shifting] is applicable regardless of whether the particular value is an integer constant or an exponent or a significand of a floating point constant" (Ans. 17; *see also* Ans. 10–11, 22).

The Examiner further asserts the claimed shifting is predictable because "a rotation of a particular bit sequence by a particular value would predictably result in a particular rotated bit sequence" (Ans. 21–22). We remain unpersuaded by the Examiner's circular reasoning, which does not address Appellants' arguments regarding the particular effects of shifting as claimed (App. Br. 9; Reply Br. 3). Appellants explain the claimed shifting is not a predictable variation of Ford's integer "rotation, i.e., changing location within a register, of an integer constant"; rather, the claimed shifting of the significand (third) field "*results in changing the precision*" and shifting the exponent (second) field "*results in changing the magnitude*[] *of the floating point constant to be generated*" (App. Br. 9; Reply Br. 3 (emphases added)). *See In re Chaganti*, 554 F. App'x 917, 922 (Fed. Cir. 2014) ("It is not enough to say that . . . to do so would 'have been obvious to one of ordinary skill.' Such circular reasoning is not sufficient–more is needed to sustain an obviousness rejection.")

The Examiner has also not shown that the additional teachings of Trissel cure the above-noted deficiencies of Ishii and Ford.

Thus, for the reasons set forth above, we do not sustain the Examiner's rejection of independent claim 1 and claims 2, 3, 5, 6, and 20 dependent therefrom. We also do not sustain the Examiner's rejection of independent claim 18, argued for substantially the same reasons as claim 1, and claim 21 dependent therefrom (App. Br. 9–10).[3]

DECISION

The Examiner's decision rejecting claims 1–3, 5, 6, 18, 20, and 21 is reversed.

REVERSED

[3] In the event of further prosecution, we suggest the Examiner evaluate claims 18 and 21 for compliance with 35 U.S.C. § 101, e.g., whether claims 18 and 21 recite no more than program code, i.e., software *per se* (e.g., an abstraction) that does not fall within any of the four classes of statutory subject matter. *In re Warmerdam*, 33 F.3d 1354, 1360–61 (Fed. Cir. 1994). For example, claim 18 recites "an instruction" comprising "fields" and "code for shifting." We further suggest the Examiner evaluate claims 1 and 18 for compliance with 35 U.S.C. § 101, particularly to determine whether claims 1 and 18 recite no more than an abstract idea of a mathematical procedure for converting and expressing numbers similar to *Benson's* algorithm for converting binary coded decimal numbers to pure binary. *See Gottschalk v. Benson*, 409 U.S. 63–65, 67 (1972) ("a method for converting binary-coded decimal (BCD) numerals into pure binary numerals" is "a generalized formulation for programs to solve mathematical problems of converting one form of numerical representation to another," which is "not patentable").